

# ENSINO MÉDIO INTEGRADO - INFORMÁTICA

## LINGUAGEM DE PROGRAMAÇÃO

### *Aula 02: Organização e Arquitetura de Computadores / Lógica Digital (Parte I)*

*O conteúdo deste documento tem por objetivo apresentar uma visão geral sobre a Organização e Arquitetura de Computadores, e é baseado no livro “Princípios Básicos de Arquitetura e Organização de Computadores” – Linda Null e Julia Labur.*

## **Introdução**

Entender o que faz um sistema de computação funcionar é necessário para que possamos pensar em otimizá-lo, corrigi-lo, ou apenas, utilizá-lo adequadamente.

### **O que é Organização de Computadores?**

Para compreendermos o funcionamento de um computador, faz-se necessária a familiarização com os diversos circuitos eletrônicos e componentes, que ajustados, criam um sistema computacional. Isso é feito através do estudo da organização de computadores. A organização de computadores trata das questões de controle (como o computador é controlado), tipos de memória; compreende todos os aspectos físicos do sistema de computação.

### **O que é Arquitetura de Computadores?**

O estudo da arquitetura de computadores tem seu foco na estrutura e o comportamento do sistema de computação. A arquitetura de computadores inclui elementos como: processador (conjuntos de instruções, tipos de dados, quantidade e tipos de registradores) memória (modos de endereçamento, métodos de acesso à memória principal) e diversos mecanismos de E/S. A arquitetura de um sistema afeta diretamente a execução lógica de programas. O estudo da arquitetura de computadores nos ajuda a compreender como é projetado um computador.

## **Principais Componentes de um Computador**

No nível mais básico, um computador é um dispositivo que consiste de três partes:

1. Um processador para interpretar e executar programas;
2. Uma memória para armazenar os dados e as instruções dos programas;
3. Um mecanismo para transferir dados vindos do mundo externo e enviar dados para o mundo externo (E/S);

O funcionamento de um computador pode ser análogo a muitos comportamentos humanos: imagine um estudante sentado em sua cadeira representa os três componentes de um computador – o cérebro do estudante é o processador, as anotações que estão sendo feitas representam a memória e a caneta usada para tomar notas é o mecanismo de E/S.

## Terminologia de Medidas na Computação

Na computação para que grandezas possam ser representadas, com intuito de informar o quão grande ou quão rápido algo é, são usadas unidades de medida. Nos anos 60, alguém decidiu que, como as potências de 2 eram similares às potências de 10, os mesmos nomes de prefixos poderiam ser usados para ambas. Portanto,  $2^{10}$  é próximo a  $10^3$ , e, assim, “kilo” é usado para referir-se a ambos. Por exemplo: se o seu modem antiquado transmite a 28,8 kb/s, então ele transmite 28.800 bits por segundo (ou  $28,8 \times 10^3$ ).

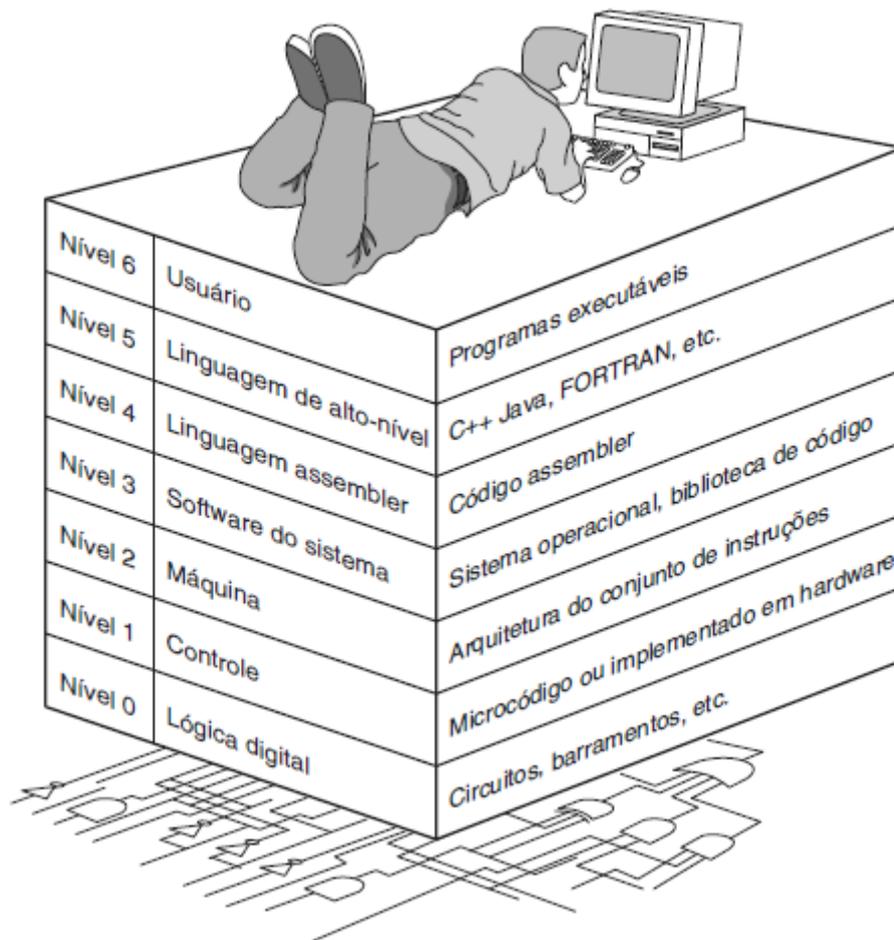
Importante notar que o uso da letra minúscula “k” para representar  $10^3$  e da letra minúscula “b” para se referir a bits. Uma letra maiúscula “K” é usada para se referir ao prefixo de potência de 2, ou 1024. Note que a letra maiúscula “B” se refere a byte. Se um disco contém 1MB, então ele contém  $2^{20}$  bytes (ou um megabyte) de informação.

**TABELA 1.1** Prefixos comuns associados com arquitetura e organização de computadores

Pref.	Símb.	Potência de 10	Potência de 2	Pref.	Símb.	Potência de 10	Potência de 2
Kilo	K	1 mil = $10^3$	$2^{10} = 1024$	Mili	m	1 milésimo = $10^{-3}$	$2^{-10}$
Mega	M	1 milhão = $10^6$	$2^{20}$	Micro	μ	1 milionésimo = $10^{-6}$	$2^{-20}$
Giga	G	1 bilhão = $10^9$	$2^{30}$	Nano	n	1 bilionésimo = $10^{-9}$	$2^{-30}$
Tera	T	1 trilhão = $10^{12}$	$2^{40}$	Pico	p	1 trillionésimo = $10^{-12}$	$2^{-40}$
Peta	P	1 quadrilhão = $10^{15}$	$2^{50}$	Femto	f	1 quadrilionésimo = $10^{-15}$	$2^{-50}$
Exa	E	1 quintilhão = $10^{18}$	$2^{60}$	Atto	a	1 quintilionésimo = $10^{-18}$	$2^{-60}$
Zetta	Z	1 sextilhão = $10^{21}$	$2^{70}$	Zepto	z	1 sextilionésimo = $10^{-21}$	$2^{-70}$
Yotta	Y	1 setilhão = $10^{24}$	$2^{80}$	Yocto	y	1 setilionésimo = $10^{-24}$	$2^{-80}$

## Hierarquia de Níveis

A organização de um sistema computacional pode ser abordada em camadas. Podemos imaginar a máquina como sendo construída em uma hierarquia de níveis, onde cada nível tem uma função específica.



Hierarquia de Níveis

- **Nível 6:** Nível do Usuário, composto pelas aplicações, é o nível com o qual todos estamos familiarizados. Neste nível executamos programas como processadores de textos, pacotes gráficos ou jogos, sendo os níveis inferiores praticamente invisíveis ao nível do usuário.
- **Nível 5:** Nível das Linguagens de Alto Nível, como C, C++, FORTRAN, essas linguagens devem ser traduzidas (usando um compilador ou um interpretador) para uma linguagem que a máquina possa entender. Linguagens compiladas são traduzidas para a linguagem simbólica do nível abaixo.
- **Nível 4:** Nível da Linguagem Simbólica, são a tradução das linguagens compiladas de alto nível para linguagem simbólica, que em níveis inferiores, será diretamente traduzida para linguagem de máquina. Esta tradução ocorre um a um, ou seja, uma instrução em linguagem simbólica é traduzida exatamente para uma instrução em linguagem de máquina (zeros e uns).
- **Nível 3:** Nível do Software de Sistema, trata das instruções de operação do sistema. Este nível é responsável por multiprogramação, proteção de memória, sincronização de processos e várias outras funções importantes. Muitas vezes as instruções traduzidas da linguagem simbólica para a linguagem de máquina são passadas por meio deste nível sem modificação.
- **Nível 2:** Arquitetura do Conjunto de Instruções (ISA), ou Nível de Máquina, consiste da linguagem de máquina reconhecida pela arquitetura particular do

sistema de computação, podem ser executados diretamente pelos circuitos eletrônicos sem quaisquer interpretadores, tradutores ou compiladores.

- **Nível 1:** Nível de Controle, é onde a unidade de controle se assegura de que as instruções sejam decodificadas e executadas adequadamente e de que os dados sejam movidos para onde e quando eles devem ser. A unidade de controle interpreta as instruções de máquina passadas para ela, uma de cada vez, fazendo com que as ações requeridas sejam executadas.
- **Nível 0:** Nível de Lógica Digital, onde encontramos os componentes físicos do sistema de computação: as portas e os fios. Estes são os blocos básicos de construção, as implementações da lógica matemática, que são comuns a todos os sistemas de computação.



***Prática: Lógica Digital – Nível 0 (Software Digital Works – Windows)***

[http://www.gileduardo.com.br/ifpr/lp/downloads/lp\\_pratica02.pdf](http://www.gileduardo.com.br/ifpr/lp/downloads/lp_pratica02.pdf)

# TECNÓLOGO EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

## PROGRAMAÇÃO DE COMPUTADORES I

### *Aula 02: Arquitetura Von-Neumann / Memórias e Tipos de Memórias (Parte II)*

#### **Arquitetura Von-Neumann**

Nas primeiras máquinas de computação eletrônicas, programar era sinônimo de conectar fios a pinos, sendo assim trocar a programação, ou seja, o programa que uma máquina devia executar era uma tarefa quase impossível, porém com a proposta de Arquitetura do matemático Von Neumann, foi possível criar computadores com programas armazenados, que ficaram conhecidos como sistemas de Von Neumann.

Arquitetura Von Neumann consiste em três sistemas de hardware:

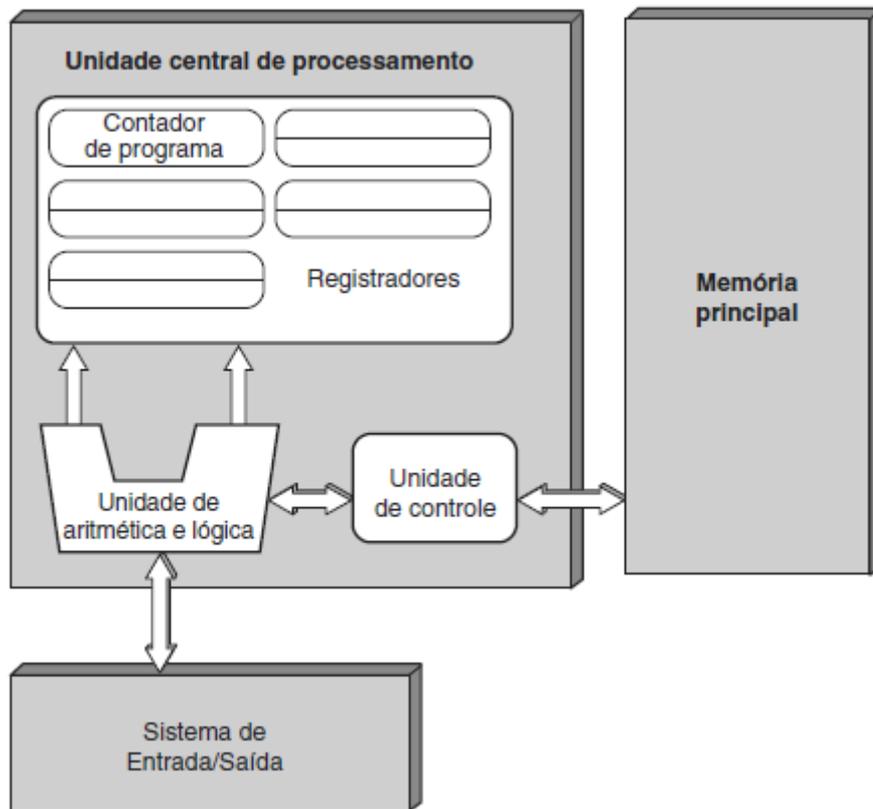
- Uma unidade central de processamento (UCP), como unidade de controle, formada pela unidade lógica aritmética (ULA), registradores (pequenas memórias) e um contador de programa;
- Um sistema de memória principal, que armazena programas que controlam a execução do computador;
- Um sistema de Entrada e Saída.

Seu funcionamento agrega os seguintes tópicos:

- Capacidade de executar processamento sequencial de instruções;
- Possui um único caminho, físico ou lógico, entre o sistema de memória principal e a unidade de controle da UCP, forçando a alternância entre ciclos de instrução e execução.

Um ciclo de instrução (ou ciclo de busca → decodifica → executa) consiste em:

1. A unidade de controle busca a próxima instrução do programa na memória usando o contador de programa para determinar onde a instrução está localizada.
2. A instrução é decodificada para uma linguagem que a ULA possa entender.
3. O operando de dados requerido para executar a instrução é carregado da memória e colocado em registradores dentro da UCP.
4. A ULA executa a instrução e coloca os resultados em registradores ou na memória.



**Arquitetura Von Neumann**



**Prática: Máquina / Instruções – Nível 1 e 2 (Software SAVON – Windows)**  
[http://www.gileduardo.com.br/ifpr/lp/downloads/lp\\_pratica02.pdf](http://www.gileduardo.com.br/ifpr/lp/downloads/lp_pratica02.pdf)

## Memória e Tipos de Memória

Os computadores, em sua grande maioria, são construídos utilizando o modelo Von Neumann, centrado na memória, tendo seus programas armazenados nela, quando estão em execução. As memórias são estruturadas como *arrays* (vetores) lineares de posições, com endereços de iniciando em 0 e indo até o tamanho máximo da memória que o processador pode endereçar.

### Tipos de Memórias

Os vários tipos de memória existentes em um computador estão ligados ao fato de que a tecnologia evoluiu ao longo dos anos e novos tipos de memória mais avançados foram criadas com isso, aumentando sua capacidade e velocidade de acesso. Embora exista um grande número de tecnologias de memória, existem somente dois tipos básicos: RAM (*random access memory*) e ROM (*ready-only memory*).

A memória do tipo RAM é dita de acesso aleatório, porém é um termo errôneo, ela seria mais bem definida como memória de leitura e escrita. A memória RAM é aquela a qual as especificações de um computador se referem, portanto se um computador possui 128 megabytes de memória, isso significa que ele tem 128 MB de RAM. Ela também é conhecida como memória principal.

Na memória de escrita-leitura temos armazenados os programas e dados que o computador necessita durante a sua execução. Contudo a RAM é uma memória volátil, perdendo suas informações quando a energia do computador é desligada. Existem dois tipos gerais de chips usados para construir a maior parte das memórias RAM nos computadores atuais: SRAM (*static random access memory*) e DRAM (*dynamic random access memory*).

A DRAM é construída com minúsculos capacitores que deixam escapar eletricidade, a DRAM requer uma recarga a cada poucos milissegundos para manter os seus dados. A tecnologia estática (SRAM), em contraste, mantém o seu conteúdo enquanto a alimentação estiver disponível. A SRAM é muito mais rápida e cara que a DRAM, entretanto os projetistas usam DRAM porque ela é muito mais densa (pode armazenar muito mais bits por chip), usando menos energia e gerando menos calor que a SRAM. Frequentemente ambas as tecnologias são combinadas: DRAM para memória principal e SRAM para memória cache.

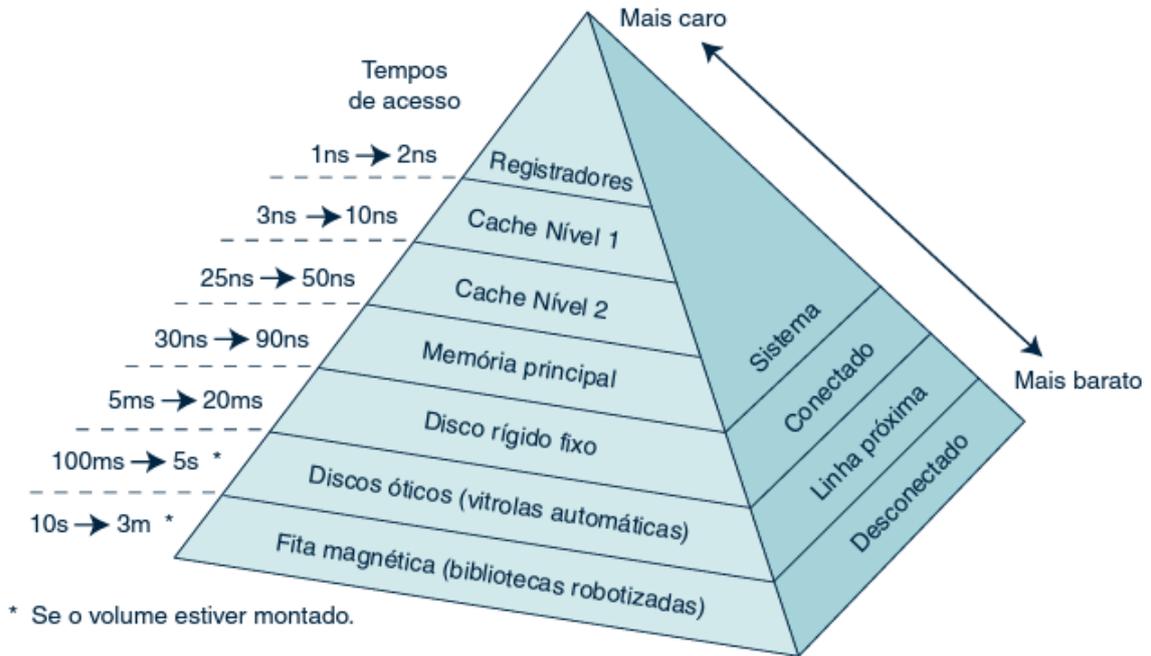
Além da RAM, a maioria dos computadores contém uma pequena quantidade de ROM que armazena informações importantes, necessárias para operar o sistema, tal como o programa necessário para a carga inicial do computador. A ROM não é volátil e sempre retém seus dados.

### **Hierarquia de Memória**

Um das considerações mais importantes para a compreensão da capacidade de desempenho de um processador moderno é a hierarquia da memória. Como existem tipos e tipos de memória (velocidade / custo), os sistemas de computação atuais usam uma combinação de tipos de memória para fornecer um melhor desempenho ao melhor custo. Essa abordagem é designada memória hierárquica. Como regra, temos: quanto mais rápida a memória for, mais cara ela se torna por bit armazenado. Os tipos básicos que normalmente constituem o sistema de memória hierárquica incluem registradores, cache, memória principal e memória secundária.

Atualmente cada computador possui uma pequena quantidade de uma memória rápida, chamada cache. Esta memória está conectada a uma memória principal muito maior, que é geralmente uma memória de média velocidade. Esta memória é complementada por uma memória secundária bastante grande, composta por um disco rígido e por diversos meios removíveis. Ao usar tal esquema é possível melhorar a velocidade de acesso efetivo à memória usando chips rápidos e caros, permitindo aos projetistas criar um computador com um desempenho aceitável.

A memória é classificada com base na sua “distância” do processador, sendo medida pelo número de ciclos de máquina necessários para o acesso. Quanto mais próxima do processador mais rápida ela deve ser. Portanto tecnologias mais lentas são usadas para essas memórias e tecnologias mais rápidas são usadas para memórias mais próximas da UCP. Sendo assim memórias mais rápidas tendem a ter menos capacidade em relação a memórias mais lentas devido ao alto custo.



Hierarquia de Memória



**Prática: Memória (Códigos-fonte C – Linux)**

[http://www.gileduardo.com.br/ifpr/lp/downloads/lp\\_pratica02.pdf](http://www.gileduardo.com.br/ifpr/lp/downloads/lp_pratica02.pdf)