

# Componentes de hardware

Prof. Fabio Henrique Silva

## Descrição

A importância e a utilização dos componentes de hardware dos computadores.

## Propósito

Aprender que um sistema de computação é um conjunto interconectado e inter-relacionado de componentes principais e subcomponentes.

# Objetivos

---

Módulo 1

## Estrutura básica de um computador

Identificar a estrutura básica de um computador.

---

Módulo 2

## Subsistemas de processamento, memória e entrada e saída

Examinar os princípios básicos dos subsistemas de processamento, memória e entrada e saída.

---

Módulo 3

## Sistema operacional

Registrar o que é mais importante sobre o sistema operacional.

---



## Introdução

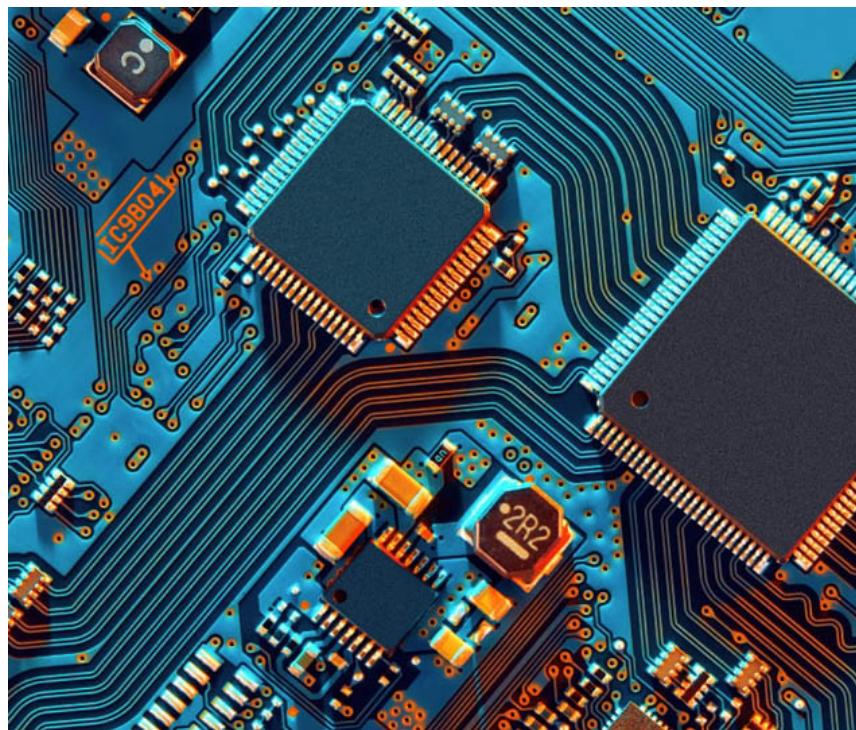
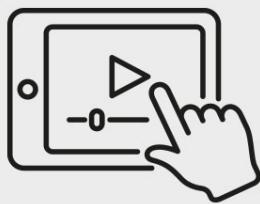
Um sistema de computação é um conjunto de um conjunto de componentes principais e subcomponentes, que são fundamentais para o funcionamento do computador. Dessa forma, há uma estrutura básica, comum a todos eles, mas que possuem especificações distintas, de acordo com a intenção de uso do computador.

Compreender a importância e a correta utilização dos componentes de hardware dos computadores é de grande relevância para que sejam escolhidos os sistemas primários e subsistemas adequados para a necessidade de uso do usuário.

Assim, neste conteúdo, você compreenderá quais são os componentes vitais de um computador. Também aprenderá quais são os principais subsistemas de hardware de um computador e seus princípios básicos.

Por fim, compreenderá o básico sobre o que é um sistema operacional e qual a sua função em um computador.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



## 1 - Estrutura básica de um computador

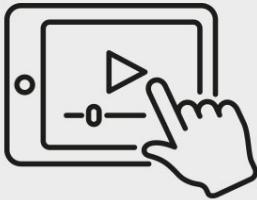
Ao final deste módulo, você será capaz de identificar a estrutura básica de um computador.

## Sistemas de computação

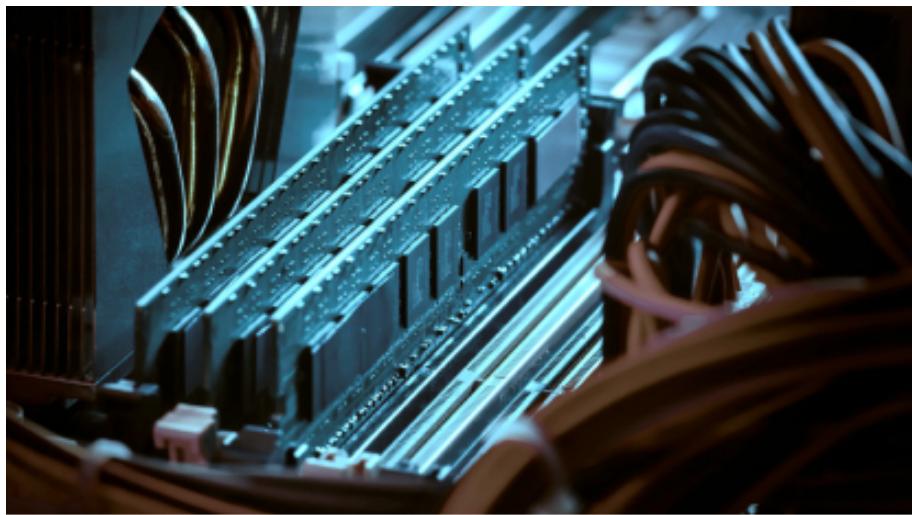


# O que é um Sistema?

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



A área conhecida como **implementação de computadores** se relaciona, em geral, com a abordagem de aspectos que são desnecessários ao programador. Por exemplo, a tecnologia usada na construção da memória, a frequência do relógio, sinais de controle para iniciar as micro-operações etc.



A implementação de computadores difere do conceito de **arquitetura de computadores**, cujo termo se relaciona, em geral, ao tratamento de pontos que são de interesse do programador, a saber, conjunto de instruções do processador, tamanho da palavra, modos de endereçamento de instruções, entre outros. A implementação de computadores e a arquitetura de computadores estão inseridas no domínio mais geral dos **sistemas de computação**. Podemos citar como alguns exemplos de sistema:

## sistemas

Conjuntos de partes coordenadas que concorrem para a realização de um determinado objetivo (MONTEIRO, 2007).



## Sistema jurídico

●

---

**Sistema nervoso**

●

---

**Sistema de informação**

●

---

**Sistema familiar**

●

---

**Sistema social**

●

---

**Sistema solar**

A **definição de computação** pode ser entendida como a **realização de cálculos de forma ordenada** ou apenas a **manipulação de valores**.

Partindo dessa definição, como você definiria um **sistema de computação**?

### Resposta

Um sistema de computação é um **conjunto de partes coordenadas** que concorrem para a realização do **objetivo de computar** (dados).

Antes de avançarmos, é necessário que você conheça algumas **definições de termos importantes**:



## Dados

Constituem um conjunto de fatos em estado bruto a partir dos quais conclusões podem ser tiradas.



## Informação

É a inteligência e o conhecimento derivados dos dados.

Outro termo importante que devemos incluir no vocabulário é o *data processing* ou, **processamento de dados** que consiste em uma série de atividades **ordenadamente realizadas** (receita de bolo), com o objetivo de produzir um arranjo determinado de **informações** a partir de outras obtidas inicialmente.

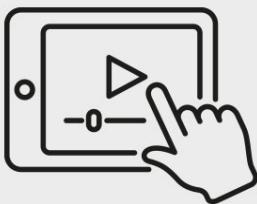
Veja a seguir as **etapas** de um processamento de dados:



Etapas do processamento de dados.

## Linguagem de programação

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



## Sistemas, dados e informação

Um **algoritmo** pode ser formalizado em **comandos** de uma **linguagem de programação**, entendida pelo sistema de computação. Por exemplo, um algoritmo para soma de 100 números (1 a 100) está exemplificado, a seguir - exemplo adaptado de Monteiro (2007, p. 9):

1

Escrever e guardar N = 0 e SOMA = 0

2

Ler número da entrada

3

Somar valor do número ao de SOMA e guardar resultado como SOMA

4

Somar 1 ao valor de N e guardar resultado como novo N

5

Se valor de N for menor que 100 então passar para item 2

6

Se não for, imprimir valor de SOMA

7

Parar

# Tipos de linguagens

Um programa pode ser escrito em diferentes tipos de linguagens, por exemplo, Assembly, Pascal, C, Cobol, Basic etc.

Uma linguagem de máquina (código de máquina) é formada por sequências de bits que representam as operações. Dessa forma, temos a seguinte comparação:

## Linguagem de baixo nível de abstração

Usa os chamados **mнемônicos** ao invés de bits. Está relacionada diretamente à arquitetura do processador. Pode ser conhecida como linguagem de montagem e Assembly.



## Linguagem de alto nível de abstração

Possui um nível de abstração relativamente elevado, mais afastado da linguagem de montagem e mais próximo à linguagem humana.

Exemplos: Pascal, Fortran, C++, Delphi.

## nemônicos

Mnemônicos são sintaxes que designam as instruções de operações.

# Organização de um sistema de computação



# Organização básica de um sistema de computação

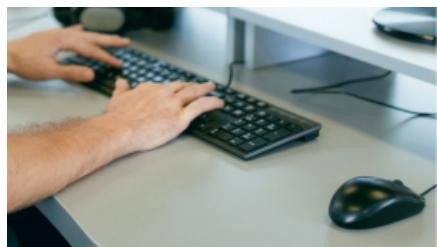
Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



A organização funcional de um **sistema de computação (S.C.)** possui os seguintes componentes:



## Dispositivo de entrada



Exemplos dispositivo de entrada.



## Dispositivo de saída

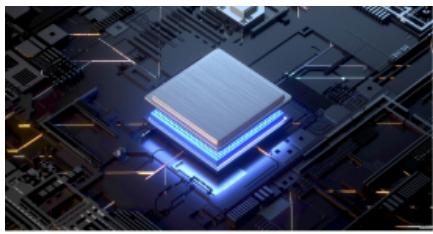


Exemplos dispositivo de saída.



## Processador





Exemplos processador.

---



## Memória principal (primária)



Exemplos Memória principal.

---



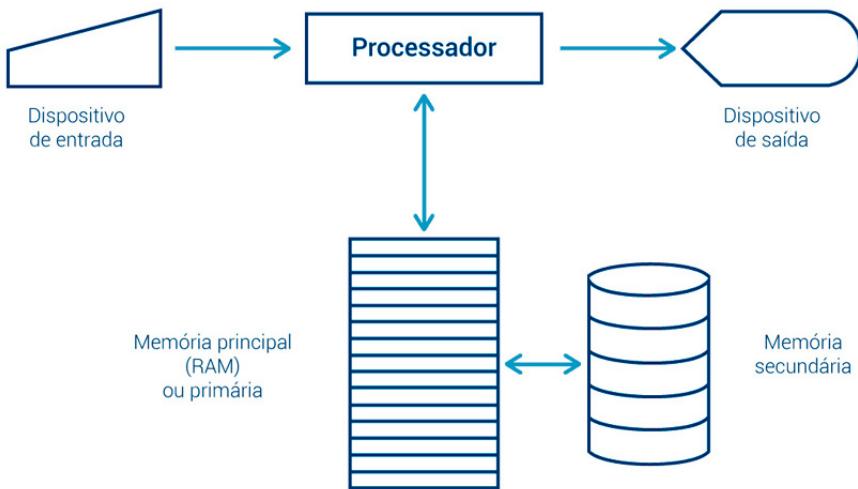
## Memória secundária



Exemplos Memória secundária.

---

Veja a seguir qual é a **relação de funcionamento** entre estes **componentes**:



Componentes de um sistema de computação.

A **Arquitetura de John von Neumann** (pronuncia-se *fon Noiman*) foi concebida a partir de 1946, precursora da arquitetura que conhecemos hoje. Ela possibilita a uma máquina digital **armazenar seus programas** no mesmo **espaço de memória** que os **dados**, permitindo, assim, a manipulação de tais programas.

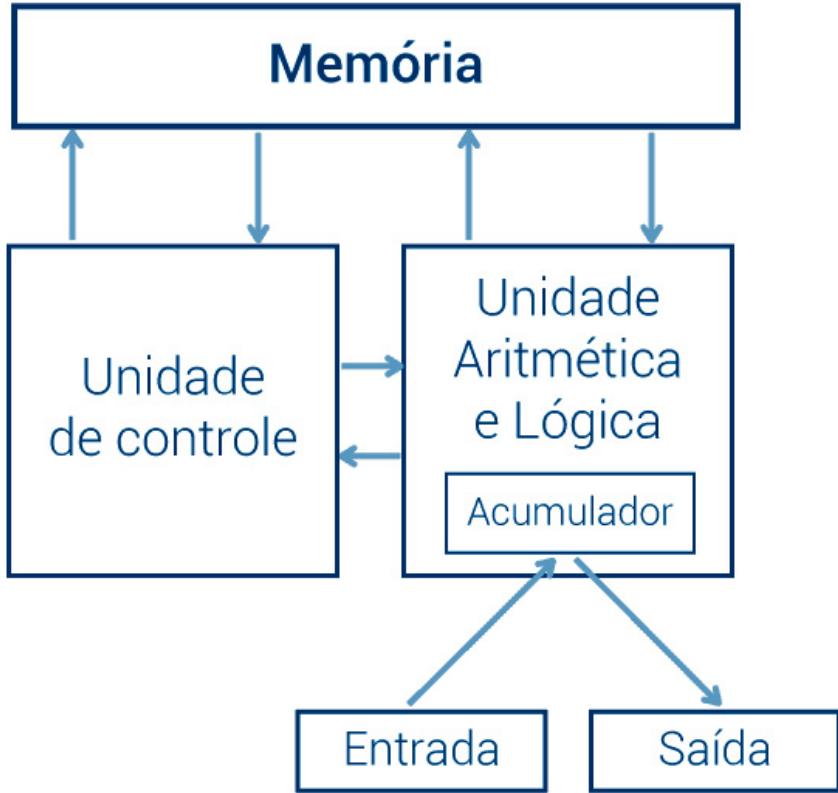
**John von Neumann** (1903-1957) foi um matemático húngaro, considerado um dos grandes gênios da humanidade.

Possui contribuições em diversas áreas do conhecimento, desde a Economia, Teoria dos Jogos, Computação até a Física Nuclear.



John von Neumann.

A arquitetura a seguir é um projeto modelo de um computador digital de programa armazenado que utiliza uma unidade de processamento (CPU) e uma de armazenamento (memória) para comportar, respectivamente, instruções e dados, conforme ilustrado.



Projeto modelo de um computador digital.

---

## Barramento

Fundamentalmente, todo sistema de computação (computador) é organizado (funcionalmente) em **três** grandes módulos ou subsistemas:

**1**

Processador

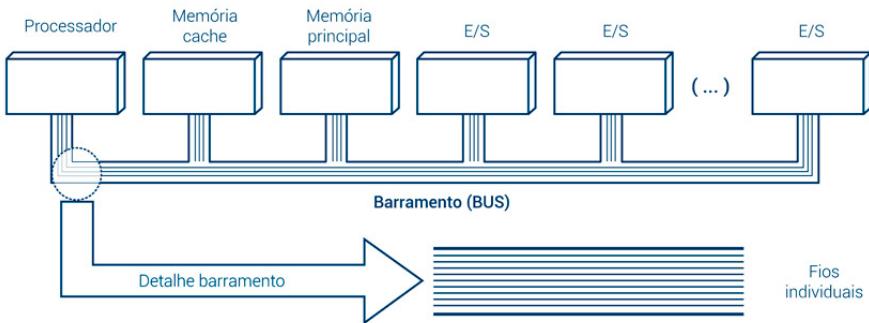
**2**

Memória

**3**

Entrada/saída (E/S)

Como se trata de componentes eletrônicos, a comunicação e o controle entre eles realiza-se por sinais elétricos que percorrem fios. Estes fios são chamados, em conjunto, de **barramento**.



Organização funcional de um S.C.

A partir do que foi apresentado e com base em seus conhecimentos, você sabe definir qual é a **função do barramento?**

### Resposta

Barramento é um conjunto de fios que têm por função transportar sinais de informação e sinais de controle e comunicação entre os componentes interligados.

## Sinais de informação

São bits, transportados entre o processador e demais componentes (memória ou periféricos), vice-versa.

## Sinais de controle e comunicação

Pulsos, que surgem em duração e intervalo de tempo distintos conforme sua função. Cada um deles serve a um propósito diferente.

Veja a seguir os tipos de barramento:

### Barramentos de dados (BD)

São bidirecionais, transportam bits de **dados** entre o processador e outro componente, vice-versa.

### Barramentos de endereços (BE)

São unidirecionais, transportam bits de um **endereço de acesso** de memória ou de um dispositivo de E/S, do processador para o controlador do barramento.

## Barramentos de controle (BC)

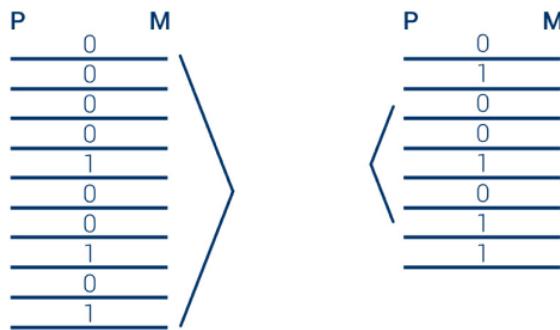
Possuem fios que enviam sinais específicos de **controle e comunicação** durante uma determinada operação.

A soma dos fios do BC, do BD e do BE é igual ao total de pinos do processador ou total de furos do soquete, ou seja: **Totalpinos = BD + BE + BC**.

Em uma operação de transferência ou acesso (seja para leitura ou para escrita, exemplificado na ([figura A](#)), o barramento é único, embora dividido em grupos de fios que realizam funções diferentes ([figura B](#)):

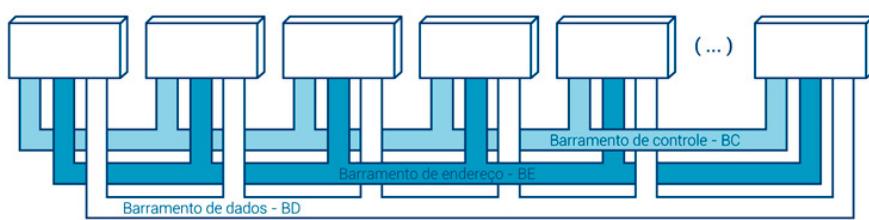
**Figura A**

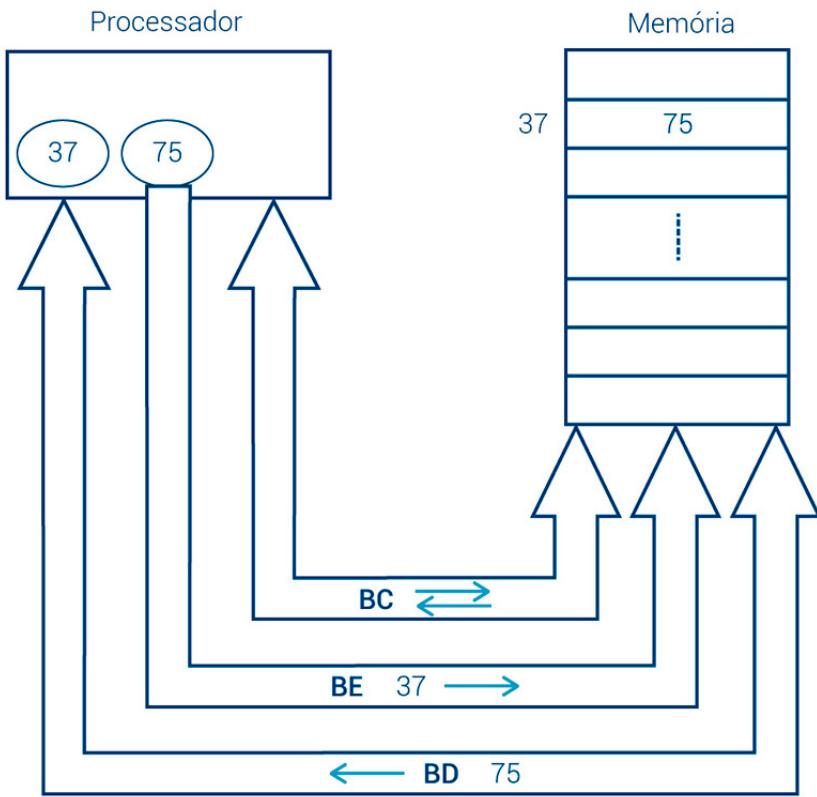
Uma operação de acesso do processador à memória principal.



**Figura B**

Divisão do barramento em grupos de fios com funções diferentes.

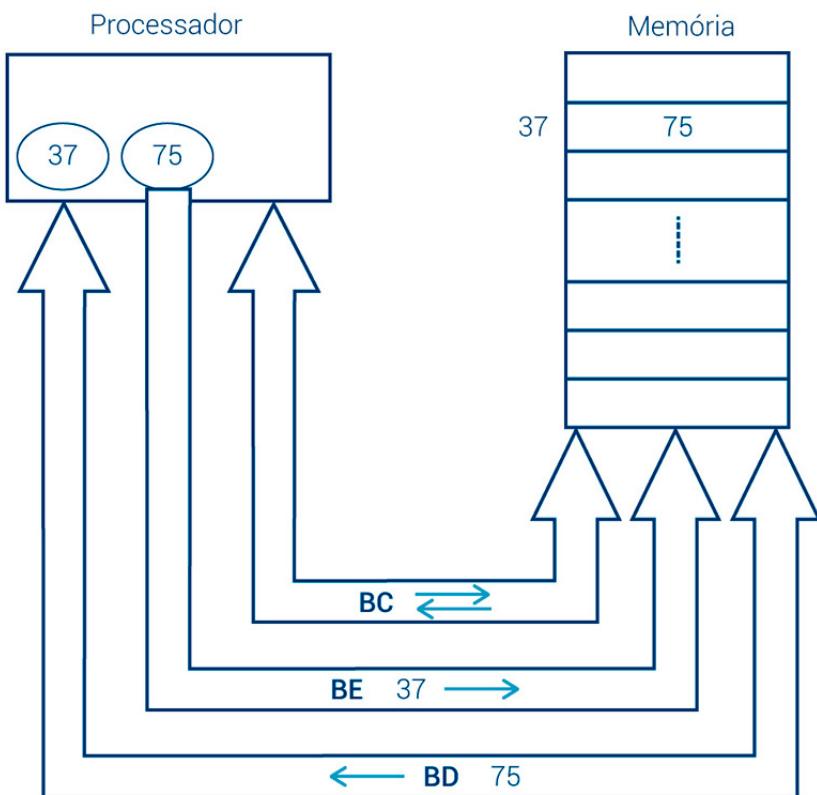




Operação de transferência.

Se o processador precisar de um dado específico ao longo da execução de uma instrução, ele saberá o endereço dele, que, por exemplo, é o endereço 37 em decimal, 0000100101 em binário [com o barramento de endereços (BE) possuindo 10 fios]. Ao acessar o endereço especificado através do barramento de endereços, o processador, então, realizará uma **operação de leitura**, transferindo o dado, por exemplo, 7510, que se encontra no interior da célula de memória, pelo barramento de dados (BD). O barramento de controle (BC) será responsável pelos sinais de controle (exemplificados a seguir).

Após o dado chegar ao processador e ser processado, um dado resultante desse processamento poderá agora seguir pelo BD, para ser armazenado em um endereço de memória, em uma operação denominada **operação de escrita**. O exemplo aqui descrito encontra-se ilustrado na figura a seguir:



Funcionamento dos BE, BD e BC.

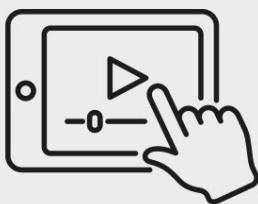
Com a finalidade de complementar os estudos a respeito de barramentos, vejamos o vídeo a seguir.



## Tipos de barramentos

Assista a uma breve explanação das características de cada um dos tipos de barramentos.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.

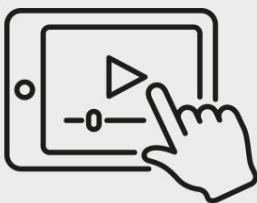


## Processadores



# Funções básicas de um processador

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Os processadores são projetados com a capacidade de realizarem diretamente (pelo hardware) pequenas e simples (primitivas) **operações**, tais como:

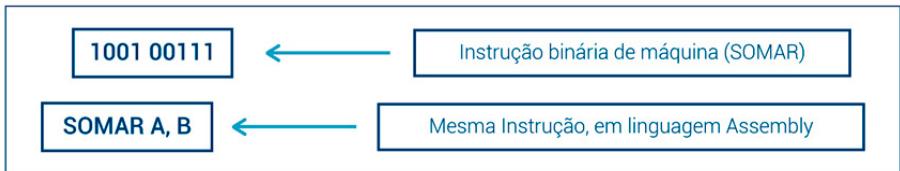
- Executar operações aritméticas com 2 números (somar, subtrair, multiplicar, dividir)

- Mover um número (dado) de um local para outro

- Mover um número (dado) de dispositivo de entrada ou de saída

- Desviar a sequência de controle

A execução de um comando em linguagem de alto nível (por exemplo, Pascal), como  $X = A + B$  requer, primeiro, sua conversão para instruções de máquina e, em seguida, sua execução propriamente dita (figura), ou seja, **somar** o valor indicado por A com o valor indicado por B e armazenar o resultado no local indicado por A.



Uma mesma instrução em linguagens diferentes.

Os processadores, então:

## Interpretam

O que fazer (qual a operação – no exemplo anterior, a operação era SOMAR).

## Executam

A operação (como fazer – algoritmo para completar a operação propriamente dita).

Uma **instrução de máquina** consiste no conjunto de bits que identifica uma determinada **operação primitiva** a ser realizada diretamente pelo hardware, por exemplo, 1001 00111 00001.

Podemos citar como **exemplos** de operações primitivas:

- Operações aritméticas – Somar, subtrair, multiplicar e dividir;
- Operações lógicas – AND, OR, XOR;
- Operações de entrada e saída de dados;
- Operações de desvio de controle;
- Operações de movimentação de dados.

## Instrução de máquina

Ações que podem ser realizadas por meio de instruções de máquina de acordo com Monteiro (2007, p.181):

Transferir uma palavra de acordo de uma célula para outra;

Efetuar a soma entre dois operandos, guardando o resultado em um deles ou em um terceiro operando;

Desviar incondicionalmente para outro endereço fora da sequência;

Testar uma condição. Se o teste for verdadeiro, então desviar para outro endereço fora da sequência;

Realizar uma operação lógica AND entre dois valores;

Parar a execução de um programa;

Adicionar 1 ao valor de um operando;

Transferir um byte de dados de uma porta de E/S para a MP;

Transferir um byte de dados da MP para uma porta de E/S;

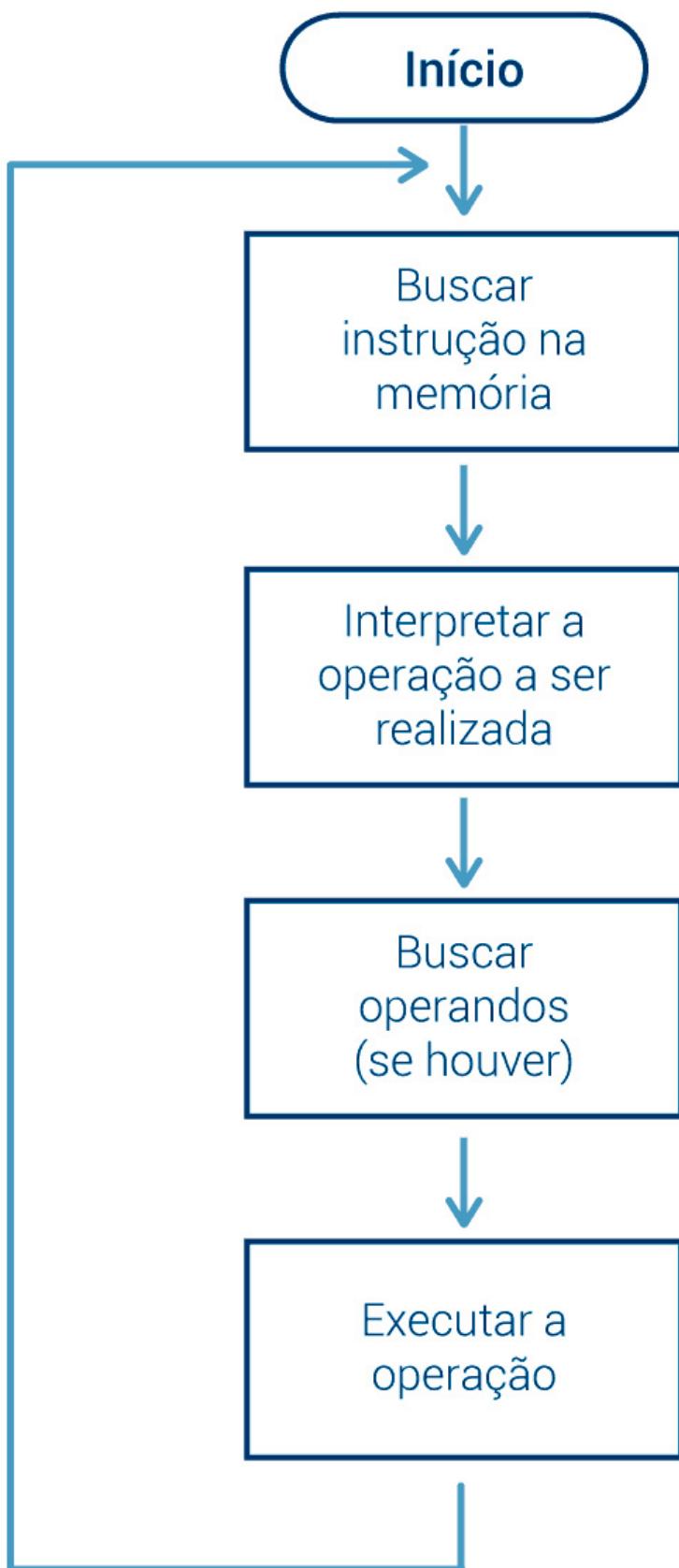
Substituir o operando por seu valor absoluto.

## Ciclo de instruções

Você sabe a diferença entre conjunto de instrução e ciclo de instrução?

**Conjunto de instruções** são todas as possíveis instruções que podem ser interpretadas e executadas por um processador. Por exemplo, o Intel 8080 tinha 78 instruções de máquina, o Pentium 4 tinha 247;

**Ciclo de instruções** é um conjunto de instruções de máquina sequencialmente organizadas para a execução de um programa.

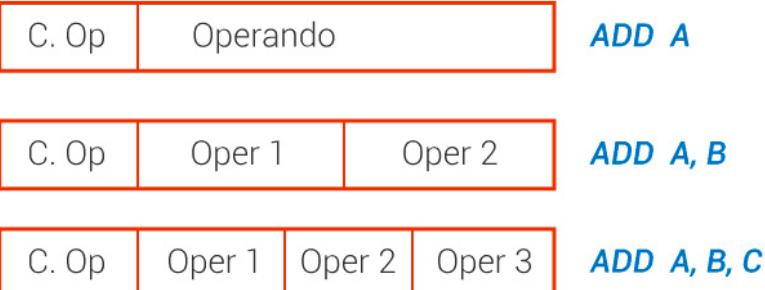


Ciclo de instrução.

O formato básico de uma instrução de máquina é constituído de duas partes. Vamos conferir!

**Código de operação (C.Op.):** Identificação da operação a ser realizada.

**Operando(s) (Op.):** Pode ter 1, 2 ou 3.

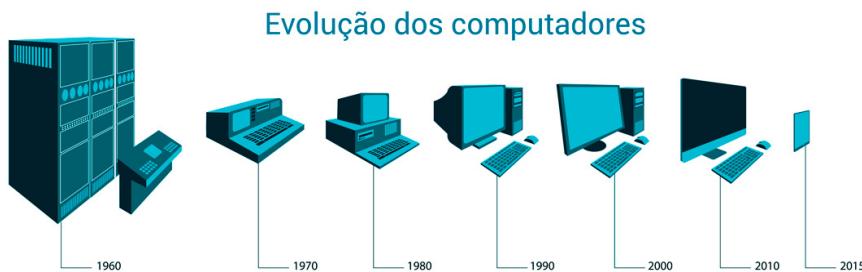


Instruções de máquina com um, dois e três operandos.

## Tendências

Desde a sua criação, o progresso tecnológico da computação foi um dos fatos mais extraordinários da humanidade. Hoje, por menos de R\$1.000,00 é possível comprar um telefone celular com um desempenho equivalente ao computador mais rápido do mundo comprado em 1993 por US\$50 milhões. Esse rápido progresso veio dos avanços na tecnologia usada para construir computadores e das inovações no design de computadores.

Reflita sobre o que virá no futuro, e sobre que tipos de conhecimentos devem ser adquiridos para almejar a vanguarda da atuação na área de computação.



Evolução no design de computadores.

**Falta pouco para atingir seus objetivos.**

# Vamos praticar alguns conceitos?

## Questão 1

A sequência básica de execução de operações primitivas é a definição de:

A Instrução de máquina.

B Conjunto de instruções.

C Ciclo de instrução.

D Mnemônico.

E Software básico.

Parabéns! A alternativa C está correta.

As etapas básicas de um ciclo de instrução podem ser simplificadas para o acrônimo BDE: Buscar instrução na memória. Decodificar a operação a ser realizada e buscar operando, se houver. Executar a operação.

## Questão 2

Um sistema de computação (S.C.) possui um processador que endereça 4 Mega (M) de endereços de memória principal no máximo. Qual é a largura de seu barramento de endereços (BE) em bits?

A 4096 bits.

B 512 bits.

C 32.768 bits.

D 22 bits.

E 32 bits.

Parabéns! A alternativa D está correta.

A quantidade de endereços de memória a serem endereçados pelo barramento de endereços é obtida da seguinte forma:  $N = 2^L$

Sendo:

$N$  = Quantidade de endereços.

$L$  = Largura (quantidade) de bits do BE ou de cada endereço.

Temos:

$N = 4$  Mega endereços (não estamos considerando o conteúdo de cada célula, apenas a quantidade de células existentes).

Cálculo de  $L$ :

A tabela a seguir expressa alguns prefixos usados para abreviar valores em computação, nos valores em potência de 2 e em potência de 10.

Unidade	Valor em potência de 2	Valor unitário	Valor em potência de 10
1k (quilo)	$2^{10}$	1024	$10^3$
1M (mega)	$2^{20}$	1.048.576	$10^6$

1G (giga)	$2^{30}$	1.073.741.824	$10^9$
1T (tera)	$2^{40}$	1.099.511.627.776	$10^{12}$

Tabela: Prefixos usados para abreviar valores em computação.

Fabio Henrique Silva

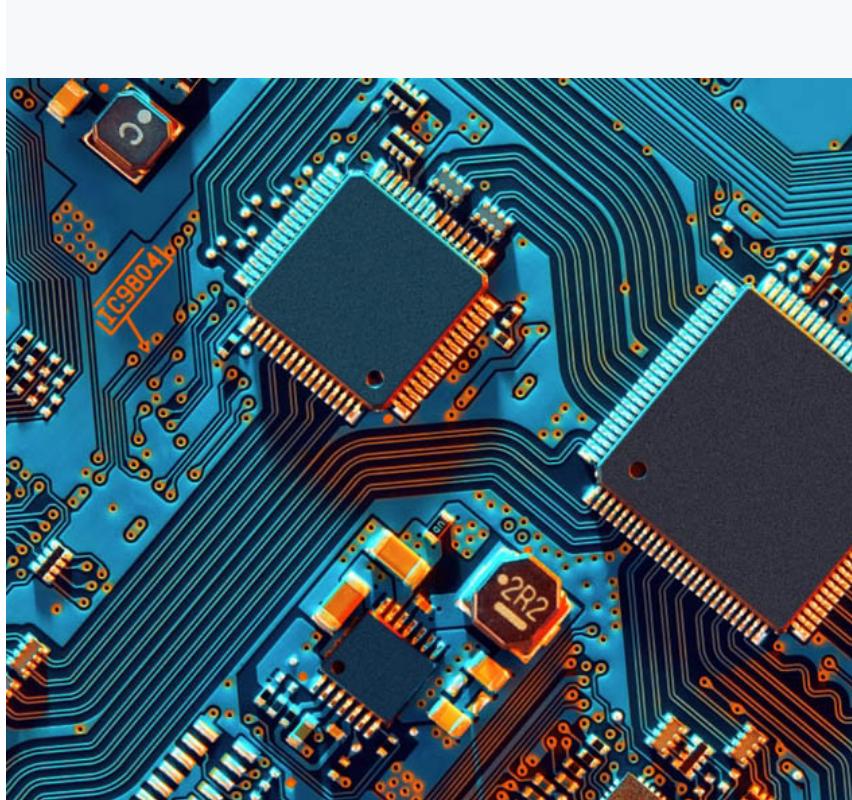
Desmembrando o valor 4 do prefixo **M**, podemos escrever o valor 4 da seguinte maneira:  $2^2 = 4$

Podemos escrever o prefixo **Mega (M)** da seguinte maneira:  $2^{20} = \text{Mega}$

Juntando: 4 Mega endereços =  $2^2 \times 2^{20}$

~~Repetindo a base e somando os expoentes:  $2^{22}$~~

Assim:  $4 \text{ M} = 2^{22} = 22 \text{ bits}$



## 2 - Subsistemas de processamento, memória e entrada e saída

Ao final deste módulo, você será capaz de examinar os princípios básicos dos subsistemas de processamento, memória e entrada e saída.

# Subsistemas de processamento

Um processador ou Unidade Central de Processamento – UCP (*Central Process Unit – CPU*) possui basicamente duas funções principais:

## Função processamento

Responsável pelo processamento dos dados.

## Função controle

É a parte funcional que realiza as atividades de buscar a instrução; interpretar as ações; gerar os sinais de controle para ativar as atividades requeridas (dentro ou fora do processador).



## Processador

Veja a seguir um vídeo que fala mais detalhadamente sobre os componentes dessas funções de processamento.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



# Memória

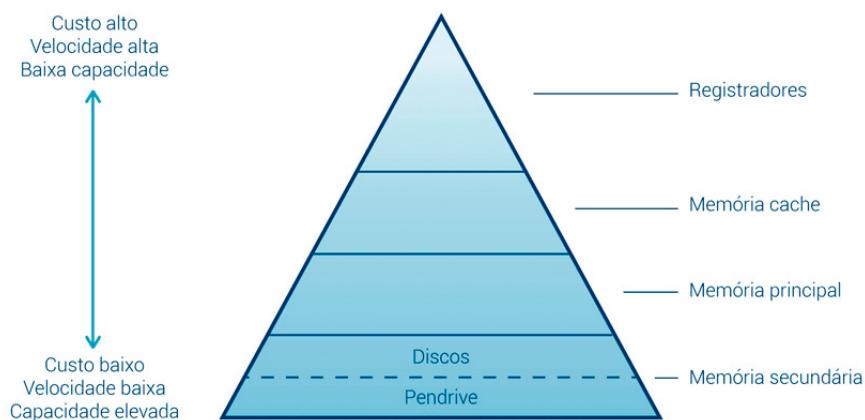
## Tipos de memória

A memória é um sistema constituído de vários componentes, cada um com velocidades, custos e capacidades diferentes. Todos, no entanto, com mesma função – **armazenar e recuperar valores**, quando desejado.

Existem diferentes tipos de memória, para diferentes finalidades, no que é conhecido como [hierarquia de memórias](#):

- Registradores;
- Memória cache;
- Memória principal– MP (ex.: RAM);
- Memória secundária (ex.: HDs, Pendrive);
- Memória virtual.

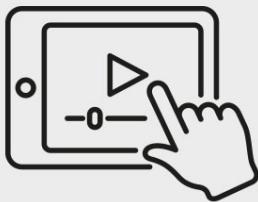
## Hierarquia de memórias





# Hierarquia de Memória

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



## Registradores

Os registradores **são memórias** com as **características do topo da pirâmide**, localizadas dentro do **processador**.

Existem:

### Registradores de dados

Armazenam os dados que serão processados pelas unidades de cálculo, separados em unidades para números inteiros e números de ponto flutuante.

### Registrador de dados de memória - RDM (*Memory Buffer Register - MBR*)

Para transferências externas de dados.

### Registrador de endereço - REM (*Memory Address Register - MAR*)

Para transferências externas de endereços de memória.

### Contador de instrução ou contador de programa - CI (*Program Counter - PC*)



Para buscar a próxima instrução.

### Registrador de instrução - RI (*Instruction Register - IR*)



Armazena instrução.

### Segmentos



Para armazenar endereços de Segmentos [apontam para determinados segmentos (programa, dados, pilha, etc.)].

### Flags



Podem ser usadas para indicar o resultado de certas instruções.

## Memória cache

A **memória cache** é uma memória de pequena capacidade, situada entre a memória principal (MP) e o processador. Essa memória armazena certa quantidade dos dados que estão sendo utilizados no momento, e que são transferidos para o processador em alta velocidade.

**Uma pergunta que muitos podem fazer é:  
Por que muitas memórias e não apenas uma?**

O ideal seria haver apenas uma memória nos computadores, com os seguintes requisitos:

- **Tempo de acesso muito curto (semelhante ao do processador)**

- **Grande capacidade de armazenamento**

- **Armazenamento permanente (não volátil)**

- **Baixo custo**

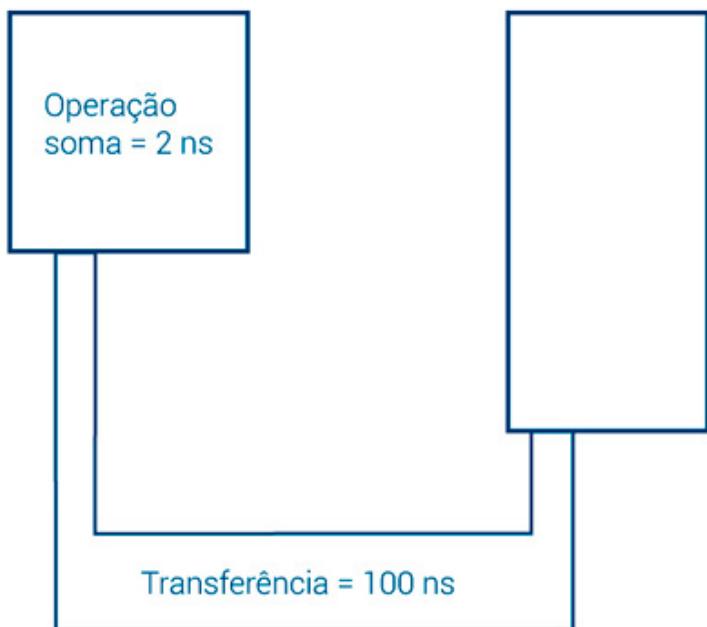
Como, em termos práticos, hoje ainda não é possível a existência desse tipo de memória, faz-se necessária a adoção de soluções mais viáveis para compensar a diferença de velocidades que existe entre o processador e a MP.

A figura ao lado ilustra a comparação de velocidades que existem entre o processador e a MP (apenas processador e MP).

Há uma grande diferença de tempo entre a transferência da MP para o processador e este usar os dados. No exemplo, o processador gasta 2 nanosegundos para somar e espera 100 nanosegundos para receber novos dados.

Processador

Memória Principal



Comparação de velocidades entre o processador e a MP.

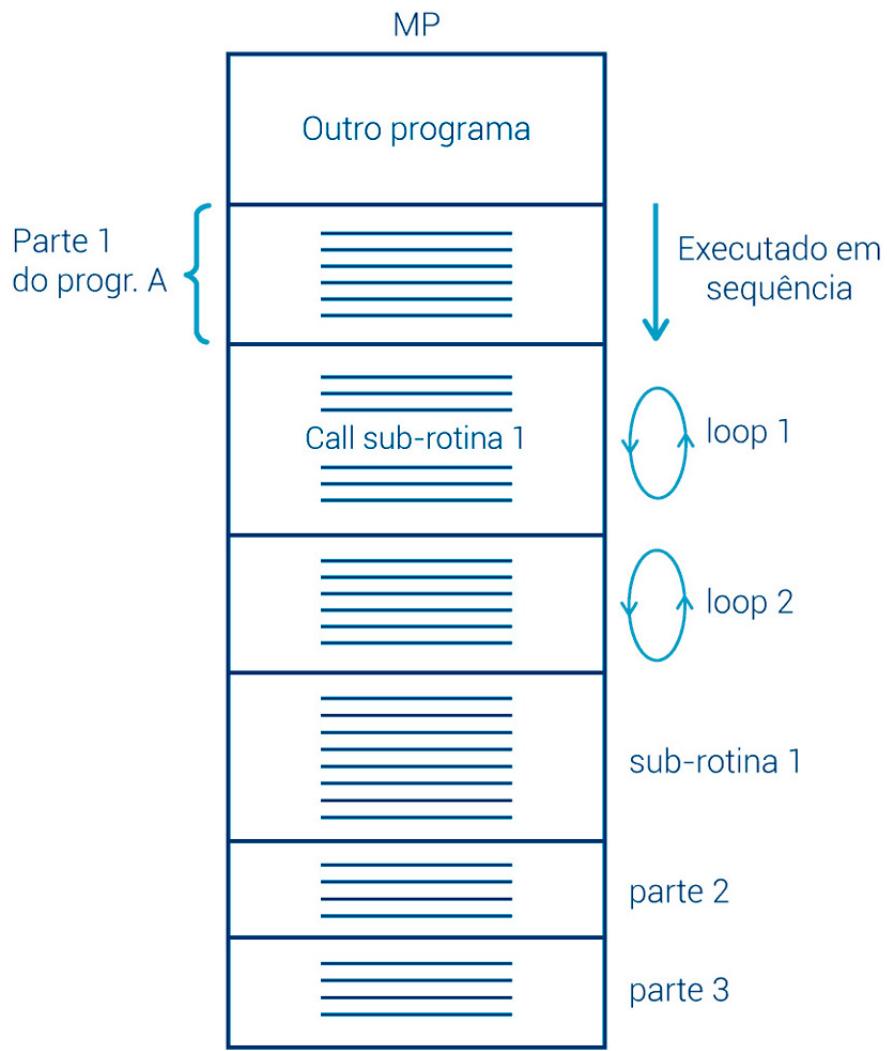
---

## Princípio da localidade

O **princípio da localidade** é um princípio de programação que determina o modo como as instruções são executadas (em sequência, durante certo tempo).

Nele, os programas são organizados de modo que as linhas de código costumam ser executadas em sequência.

Apenas em alguns momentos a sequência é interrompida e o processo desvia da sequência, sendo esta retomada em seguida.



O modo como as instruções são executadas na MP.

O princípio da localidade é dividido em:

### Localidade espacial

Sempre que o processador realiza um acesso a um endereço de memória, é provável que o próximo acesso seja ao endereço contíguo seguinte.



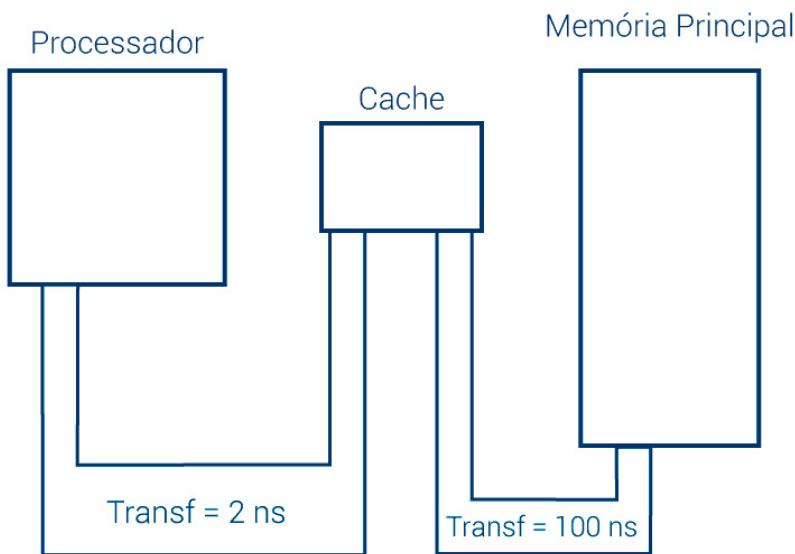
### Localidade temporal

Sempre que o processador realiza um acesso a um endereço de memória, é provável que, em curto tempo, ele acesse novamente o mesmo endereço.

Devido ao princípio da localidade (espacial), é possível incluir uma **memória de pequena capacidade**, chamada **memória cache**, entre a MP e o processador.

A figura a seguir ilustra novamente a comparação de velocidades que existem entre o processador e a MP, mas, agora, utilizando memória cache como intermediária.

Caso seja usada uma memória intermediária de alta velocidade entre a MP e o processador (que armazena uma cópia dos dados sendo imediatamente usados), este irá esperar 2 nanosegundos pelos dados, ao invés de 100 nanosegundos.



Comparação de velocidades entre o processador e a MP usando a memória cache intermediária.

## Memória principal (MP)

A **memória principal (MP)** é a memória básica, na qual o programa que será executado e seus dados são armazenados, para que o processador busque cada instrução ao longo do tempo de processamento.

Memórias muito antigas usavam o método de acesso sequencial, em que o endereço de cada acesso era sempre relativo ao endereço inicial. Exemplo de acesso sequencial é o dos sistemas VHS (videocassete) e das fitas magnéticas.

[Saiba mais](#)

Em 1968, um cientista da IBM criou uma memória constituída apenas de componentes eletrônicos e cujo acesso dependia apenas de seu endereço, sendo independente dos demais. Por isso, ele chamou-a de **memória de acesso aleatório ou RAM** (*Random-Access Memory*). Estas memórias (logo chamadas de DRAM, cujo D vem da palavra **dinâmica**) passaram progressivamente a ser o tipo usado para acesso pelo processador e, daí, tornaram-se a memória principal de praticamente todos os sistemas de computação.

A memória principal (RAM) permite a realização de duas operações:

## Escrita (armazenar)

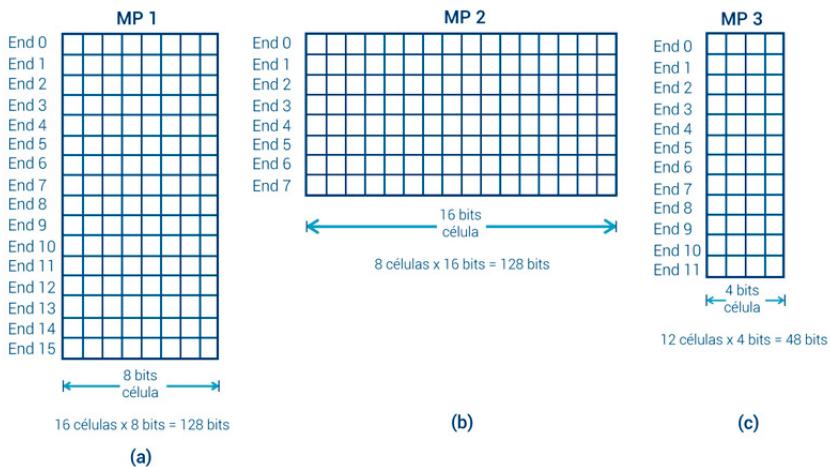
O dado anteriormente armazenado é apagado.

## Leitura (recuperar)

Normalmente se recupera uma cópia do dado.

A memória é organizada como um conjunto de **N partes** iguais, com cada parte possuindo um conteúdo fixo de **M bits**.

O valor de M depende do tipo de memória. Usualmente é 8 bits (1 Byte) nas memórias RAM, mas existem valores maiores para outras memórias.



Diferentes tamanhos de memórias.

Cada parte (chama-se **célula** ou, em alguns casos, **palavra** nas memórias RAM, **linha** nas memórias cache, **setor** nos HDs etc.) é identificada por um número, chamado **endereço**. Todos os **N** endereços têm mesma **largura de endereço**, de **E** bits. Vejamos um breve exemplo.

## Exemplo

Para o endereço 1011101, a largura de endereço **E** será 7 bits.

Uma memória com **N** partes também possui **N** endereços. O cálculo de **N** no endereçamento pode ser realizado da seguinte maneira:

$$2^E = N$$

Onde:

**E** = Largura de cada endereço;

**N** = Quantidade de endereços (partes endereçáveis).

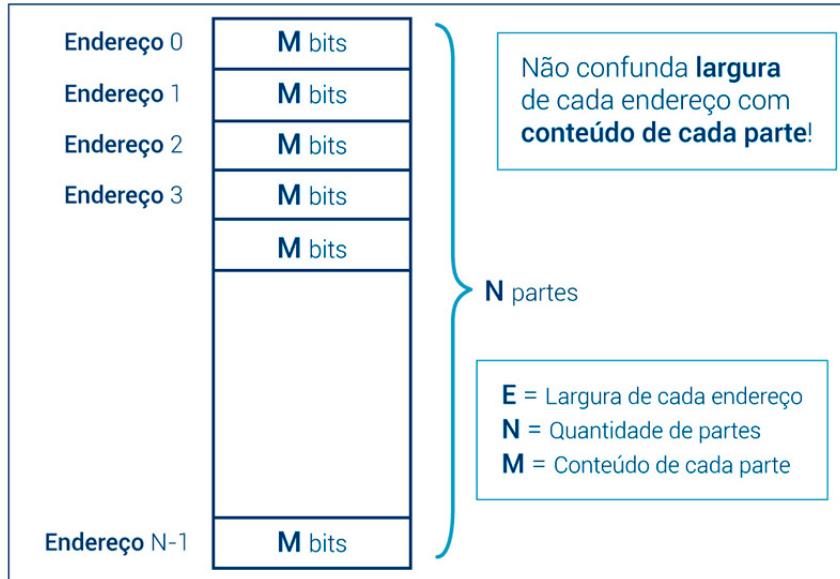
Rotacione a tela. ☺

Vejamos um breve exemplo:

### Exemplo

Se a largura de endereços é igual a 6 bits, a quantidade de endereços **N** será:  $2^6 = 64$  endereços

Esta figura esquematiza simplificadamente os termos apresentados anteriormente:



Organização básica de uma memória.

As memórias eletrônicas que empregam o acesso aleatório podem ser fabricadas para permitir **duas aplicações**:

## Para leitura e escrita

Read/Write – **R/W**

# Somente para leitura

*Read Only Memory – ROM*

/W

O termo R/W nunca foi adotado, usa-se RAM para representar memórias voláteis para leitura e escrita, e ROM para representar memórias não voláteis somente para leitura.

As memórias RAM são constituídas de **dois tipos**:

## SRAM (*Static Random-Access Memory*)

- Cada bit é constituído de 5 a 7 transístores;
- Não requer recarregamento, sendo, por isso, mais rápidas, mas ocupam mais espaço e são mais caras;
- Usadas como memória cache.

## DRAM (*Dynamic Random-Access Memory*)

- Cada bit é constituído por 1 capacitor e 1 transístor;
- O capacitor serve para representar o valor do bit e o transístor para ser usado nas leituras/escritas;
- Como o capacitor se descarrega, é preciso recarregar periodicamente (sinal de refresh – gasta tempo);
- Usadas como memória principal.

As memórias dinâmicas podem ser de dois tipos, vejamos a seguir:

## Memórias dinâmicas assíncronas

Não são sincronizadas com o processador, por exemplo, Dynamic RAM (DRAM), Fast Page Mode (FPM), Extended Data Out DRAM (EDO), Burst Extended Data Out DRAM (BEDO).

## Memórias Dinâmicas Síncronas

Sincronizadas com o processador, evitam que o processador espere os dados, por exemplo, Synchronous DRAM (SDRAM), Double Data Rate (DDR), Double Data Rate 2 (DDR2).

Hoje em dia, é comum o uso de memórias DDR SDRAM, pois as memórias **Single Data Rate** (SDRAM) só transferem dados na subida do sinal de clock; Já as memórias **Double Data Rate** (DDR-SDRAM) transferem dados na subida e na descida do sinal de clock, dobrando a taxa de transferência de dados (data rate); Assim, uma **memória DDR-SDRAM** operando num clock de 100MHz (real) consegue desempenho equivalente a 200MHz (efetivo).

Também existe a **classificação** quanto ao **tipo de encapsulamento** das memórias (formatos dos módulos):

## SIMM (*Single In Line Memory Module*)

O contato elétrico de um lado é igual ao do outro lado.

## DIMM (*Dual In Line Memory Module*)

Os contatos dos dois lados são independentes.

## Memória secundária

Terminando a pirâmide da hierarquia de memória, a **memória secundária** objetiva o armazenamento persistente (permanente) aos programas de usuário e seus dados.

O emprego de diferentes tecnologias para compor os diferentes tipos de memórias da hierarquia pode ser feito através de parâmetros para análise, tais como:

**Tempo de acesso:** Também conhecido como tempo de acesso para leitura ou tempo de leitura.

**Ciclo de memória:** É outro parâmetro (apenas para memórias eletrônicas), indica o tempo entre 2 operações sucessivas de leitura ou escrita.

**Capacidade.**

**Volatilidade.**

**Tecnologia de fabricação:** Memórias de semicondutores, memórias de meio magnético, memória de meio óptico.

**Temporariedade:** Permanente, transitório

**Custo.**

Para os **tipos de memória da hierarquia**, podem ser exemplificados alguns **parâmetros de análise**, conforme exibido na tabela:

	Internamente no processador			
	Registrador(es)	Cache L1	Cache L2	
Tecnologia de fabricação	Eletrônica	Eletrônica	Eletrônica	E
Volatilidade	Volátil	Volátil	Volátil	V
Tempo de acesso	Ex.: 1 a 2ns	Ex.: 2 a 6ns	Ex.: 2 a 6ns	E
Capacidade	32 ou 64 bits	8KB, 256KB, ...	2MB, 8MB, ...	4

Tabela: Parâmetros de análise das memórias.

Fabio Henrique Silva

A sequência de transferência de dados realizada entre o processador e as memórias em um sistema computacional é hierárquica, conforme mostrado na imagem a seguir.

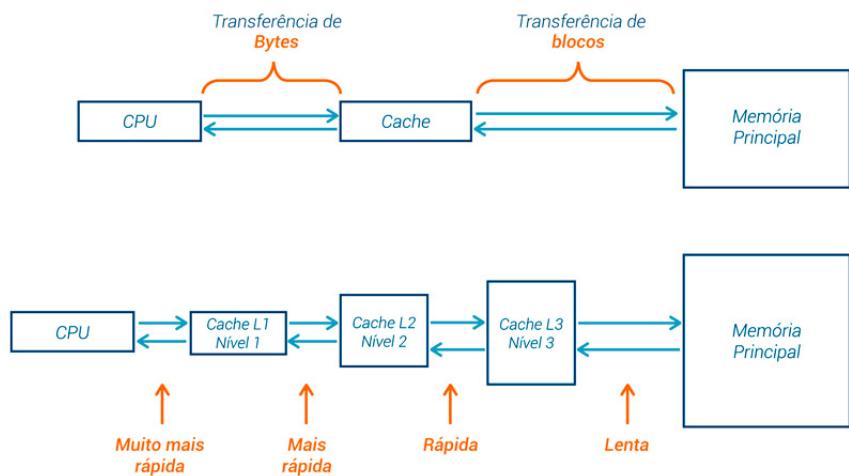
Ou seja, grosso modo, em uma operação de leitura, o processador:

Irá verificar primeiro se o dado está localizado na cache L1.

Caso não esteja, verificará se o dado se encontra na cache L2 e L3 (se houver).

Finalmente, irá buscar o dado na memória principal, caso o dado não esteja localizado em nenhuma das memórias cache consultadas.

O mesmo raciocínio pode ser aplicado na operação de escrita, na qual o processador escreverá sempre na cache mais próxima dele, mas o dado precisa estar atualizado na memória principal (RAM) para, em seguida, ser armazenado na memória permanente (HD, por exemplo).



Diferentes tamanhos de memórias.



## Memória

Confira agora com mais detalhes os conceitos apresentados.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



## Subsistema de entrada e saída (E/S)

### Objetivos e funções

O subsistema de entrada e saída (E/S) tem por função interligar o mundo exterior (o nosso mundo) ao mundo interior (processador-memória). Os dispositivos de entrada e saída também são chamados **periféricos**, pois estão na periferia do núcleo processador/memória principal.

São **funções** do subsistema de E/S:

● Receber ou enviar informações do/para o meio exterior..

● Converter as informações (de entrada ou de saída) em uma forma inteligível para a máquina (se estiver recebendo) ou para o operador (estar enviando).

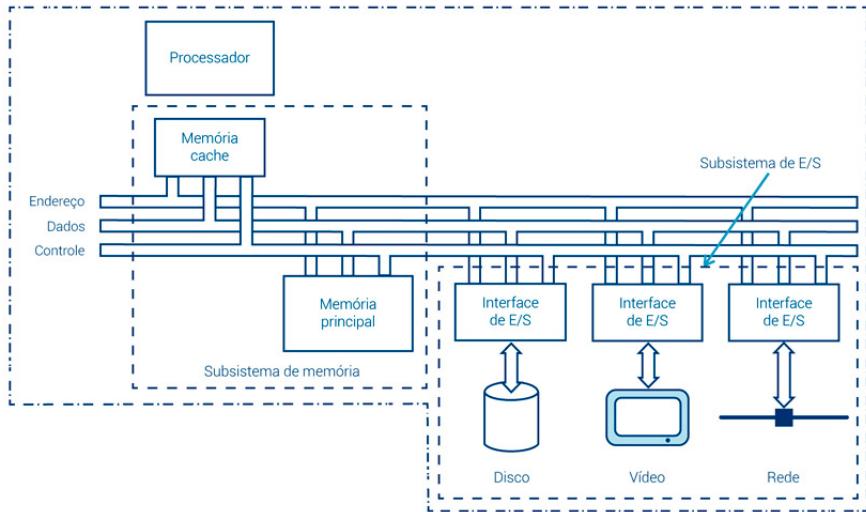
### Dispositivo e interface

Todo componente de E/S é constituído de 2 partes: O **dispositivo** propriamente dito; um componente denominado **interface**. Vejamos alguns exemplos:

#### Exemplo

Monitor de vídeo (dispositivo) e placa de vídeo (interface); disco magnético (composto pelo dispositivo hard disk e pelo controlador ou interface).

Agora, vejamos como o subsistema de E/S se comunica dentro de um sistema de computação a partir do esquema a seguir:



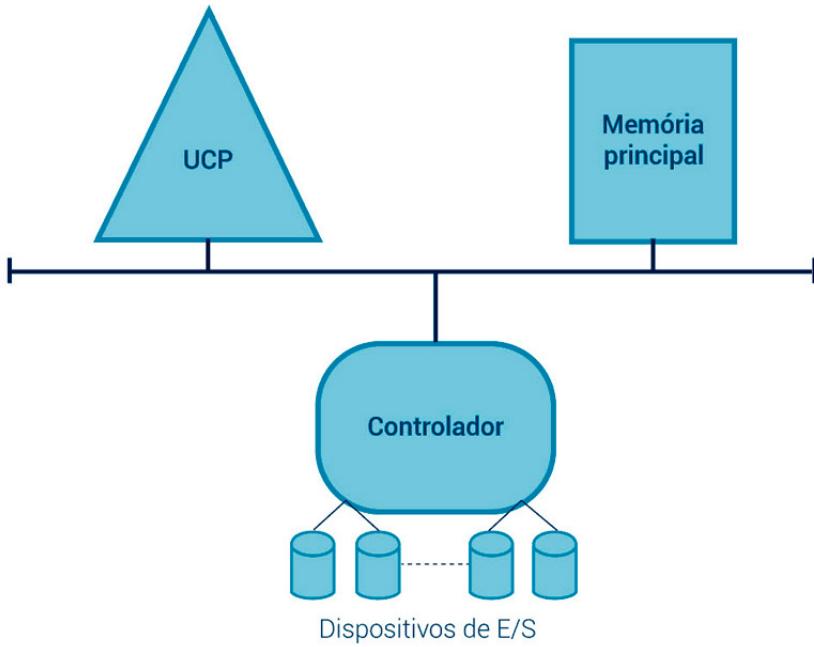
Como o subsistema de E/S se comunica dentro de um sistema de computação.

**Interface ou controlador** serve para compatibilizar as diferentes características entre o processador/memória e o dispositivo que controla, bem como controlar o funcionamento do referido dispositivo.

### Exemplo

Uma placa de vídeo é interface do monitor, assim como a placa controladora de um HD.

Confira um esquema visual da localização do controlador:



Esquema ilustrativo da localização do controlador.

A necessidade do emprego de interfaces tem origem em diversos fatores:

1

Cada dispositivo possui suas próprias características. Exemplo: velocidade de transferência de dados, quantidade de bits enviados em cada instante, formato do dado a ser transferido etc.

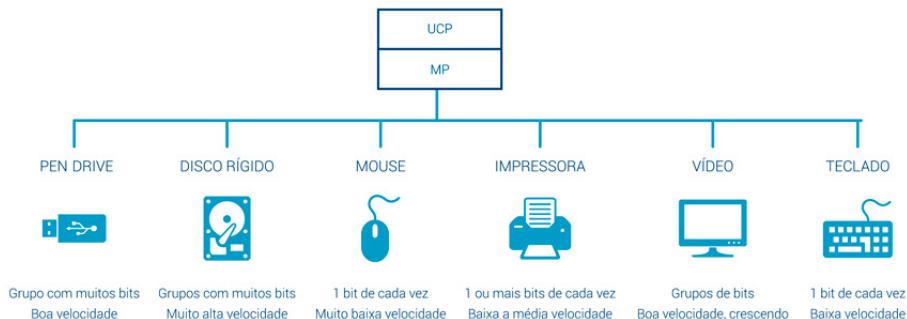
2

As atividades de E/S são assíncronas, isto é, não são sincronizadas pelos pulsos do relógio interno. Nunca se sabe quando uma tecla será pressionada no teclado, nem quando termina o movimento de braço de leitura e gravação dentro de um disco rígido. Há necessidade de um acordo para a comunicação fluir corretamente.

3

Podem ocorrer ruidos e outras interferências, pois os fios externos (geralmente cabos) têm comprimento apreciável.

Em relação aos dispositivos de E/S devemos ressaltar que cada um possui suas próprias características. Veja melhor na imagem a seguir:



Características dos dispositivos de E/S.

Exemplos de **dispositivos ou periféricos** são:

### Dispositivos de entrada

Teclado, mouse, mesa digitalizadora, scanner etc.

**Exemplo:** O teclado possui o seguinte processo de funcionamento:

- Detecção do pressionamento de uma tecla;
- Confirmação do pressionamento;
- Geração do código de identificação da tecla;
- Sinal de interrupção (grosso modo, incluir o pedido de processamento da tecla no meio de outras execuções que estão sendo realizadas);
- O programa de controle (BIOS) processa o significado daquela tecla (por exemplo, um caractere), e envia o resultado para a aplicação que está em execução; utilizará o resultado desse processamento.

### Dispositivos de saída

Impressora, caixa de som, monitor.

**Exemplo:** A impressora pode ser dos tipos:

- Impacto (esfera, matricial);
- Sem impacto (jato de tinta, laser, sublimação de tinta).

## Dispositivos de entrada e saída

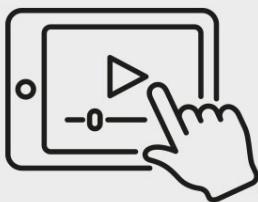
Disco magnético, pen drive, SD Card, disco SSD etc.

## Disco magnético



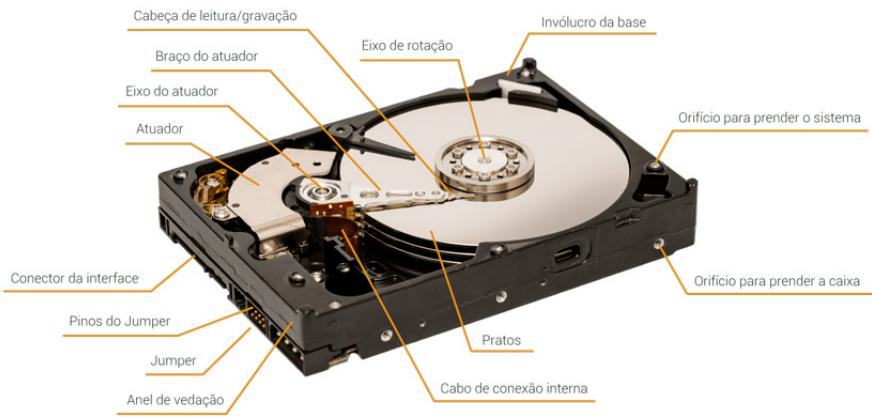
## Funcionamento básico do disco magnético

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Um **disco magnético** é composto por um **prato circular** construído de **material não magnético**, chamado **substrato**, revestido com um **material magnetizável**.

Os dados são gravados e posteriormente recuperados do disco por meio de uma bobina condutora chamada **cabeça**. Em muitos sistemas, existem duas cabeças, uma de leitura e uma de gravação. Durante uma operação de leitura ou gravação, o cabeçote fica parado enquanto o prato gira embaixo dele.



Disco magnético ou Hard Disk.

A organização dos dados no prato é feita em um conjunto concêntrico de anéis, chamados **trilhas**. As trilhas adjacentes são separadas por **intervalos**. Isso impede ou minimiza erros devido ao desalinhamento da cabeça ou simplesmente interferência de campos magnéticos.

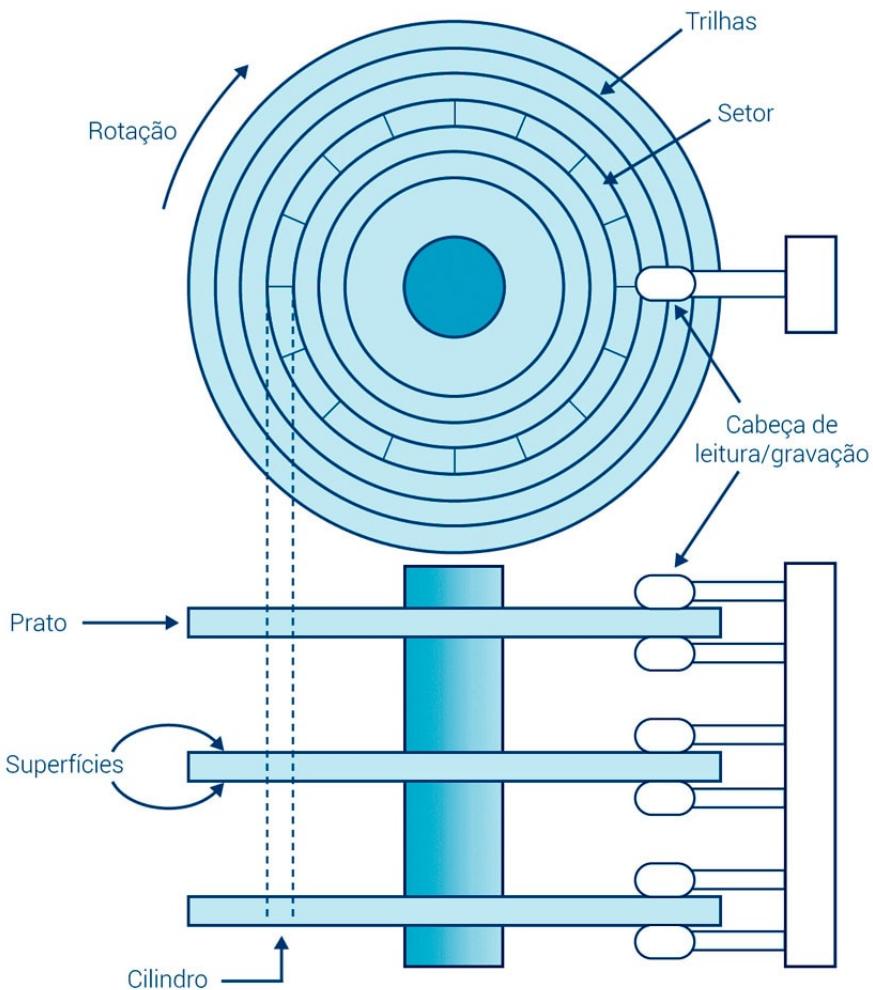
### Atenção!

Os dados são transferidos para o disco em setores. Normalmente, existem centenas de setores por faixa, e estes podem ter comprimento fixo ou variável.

Algumas unidades de disco acomodam vários pratos empilhados verticalmente com uma fração de polegada de distância. Os discos de múltiplos pratos empregam uma cabeça móvel, com uma cabeça de leitura e gravação por superfície do prato.

Todas as cabeças são fixadas mecanicamente, para que todas fiquem à mesma distância do centro do disco e se movam juntas. Assim, a qualquer momento, todas as cabeças são posicionadas sobre trilhos que estão a uma distância igual do centro do disco. O conjunto de todas as faixas na mesma posição relativa no prato é chamado de **cilindro**.

Confira alguns dos elementos da estrutura de um disco magnético:



Estrutura de um disco magnético.

Nos primeiros sistemas de computação, a CPU e os periféricos se comunicavam por **instruções de E/S** executadas **pelo próprio processador**. Essas instruções continham detalhes específicos de cada periférico, por exemplo, em qual trilha e em qual setor de um disco deveria ser lido ou gravado um bloco de dados. Existia uma forte dependência entre processador e dispositivos de E/S.

Com o surgimento do controlador ou interface, estes passaram a agir independentemente dos dispositivos. Uma técnica de ação se chama **E/S controlada por programa**, no qual o processador fica ocupado até o término da operação de E/S. Outra técnica é denominada **E/S controlada por interrupção**, na qual o processador permanece livre para processar outras tarefas.

A técnica de E/S controlada por interrupção é bem eficiente, porém, na ocorrência da transferência de um grande volume de dados, o processador tem que intervir mais vezes. A solução adotada para isso veio na forma da técnica conhecida como **DMA (Direct Access Memory)**.

### Saiba mais

No DMA, um bloco de dados pode ser transferido entre memória principal e dispositivos de E/S sem a intervenção do processador, exceto no início e no final da transferência.

# Driver de dispositivo

Cada dispositivo de E/S ligado ao computador precisa de algum **código específico** do dispositivo para controlá-lo. A esse código dá-se o nome de **driver de dispositivo**.

## Exemplo A

Um driver de disco deve saber sobre setores, trilhas, cilindros, cabeçotes, movimento do braço etc.

## Exemplo B

Um driver de mouse deve aceitar informações dizendo o quanto se moveu e qual botão foi pressionado.

Os dispositivos podem transmitir dados em grupos de bits (paralela) ou bit por bit em série (ou serial):

### Transmissão paralela

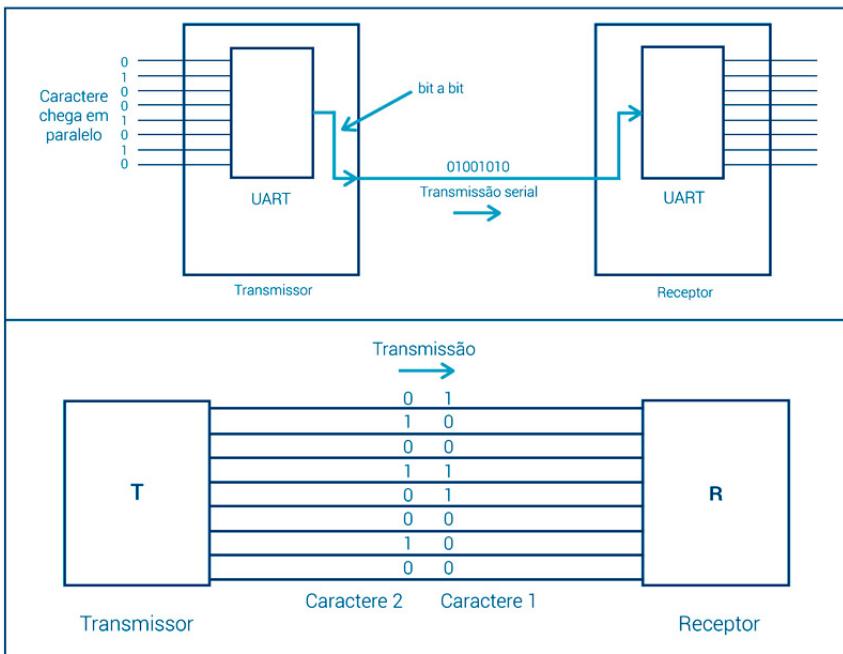
Na transmissão paralela, um grupo de bits é transmitido de cada vez, cada um sendo enviado por uma linha separada de transmissão.



### Transmissão serial

Na transmissão serial, o periférico é conectado ao dispositivo controlador por uma única linha de transmissão de dados, um bit de cada vez.

A transmissão em série utiliza menos condutores e, por isso, tem menor custo que a paralela. A transmissão paralela foi usada muito tempo para conexão de periféricos (vídeo, impressoras etc.) ao processador/memória. No entanto, ela possui um problema chamado **deslizamento** (em altas taxas). Isso significa que, se houver um mínimo atraso em qualquer um dos sinais enviados simultaneamente pelas várias linhas, o receptor não captará o dado. Hoje em dia a maioria das transmissões paralelas foram substituídas por serial, tais como USB, SATA etc. A transmissão paralela é usada apenas nas conexões internas do processador e placas-mãe para a ligação entre processador, cache e memória principal.



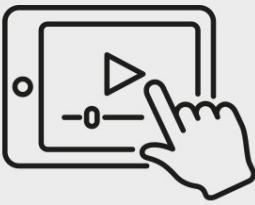
Transmissão serial e transmissão paralela.



## Dispositivos de entrada e saída

Confira agora mais detalhes sobre os conceitos trabalhados.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



## Tendências

Segundo Patterson (2011), no que tange aos futuros processadores, eles incluirão vários núcleos específicos de domínio, que executam apenas uma classe de cálculos, mas o fazem notavelmente melhor do que os núcleos de uso geral, visando a melhoria do custo de desempenho energético. É a especialização.

Quanto às memórias, em todas as camadas da pirâmide de hierarquia de memória, temos significativos avanços ocorridos:

A memória cache possui velocidades e tamanhos cada vez maiores, que auxiliam no desempenho dos processadores.

As memórias DDR SDRAM possuem latência (atraso que existe para que o dado seja acessado pelo processador na memória) cada vez menor e maior capacidade de armazenamento, sendo utilizadas inclusive em placas gráficas de alto desempenho.

Os dispositivos de memória secundária, principalmente discos, estão sendo fabricados com novas tecnologias diferentes da tecnologia dos antigos discos rígidos, por exemplo, Solid State Disks (SSDs), o que confere uma maior velocidade de acesso aos dados.

Além disso, cabe ressaltar o uso de serviços de recursos remotos, ou seja, em nuvem, que empregam recursos compartilhados com alta disponibilidade, capacidade e qualidade de experiência para os usuários.

## Falta pouco para atingir seus objetivos.

### Vamos praticar alguns conceitos?

Questão 1

Qual é e em que área da UCP (processador) se localiza o registrador cujo conteúdo controla a sequência de processamento das instruções de um programa?

A Unidade de controle – UC.

B Registrador de instrução – RI.

C Contador de instrução – CI.

D Registrador de dados de memória– RDM.

E Cache L1.

Parabéns! A alternativa C está correta.

O registrador é o CI, Contador de Instruções, que armazena o endereço da próxima instrução a ser executada. Fica localizado na área de controle.

## Questão 2

Qual é a função dos registradores de dados?

A Realizar as operações aritméticas e lógicas existentes no conjunto de instruções do processador.

B Armazenar os dados a serem manipulados pelas unidades de cálculo.

C Realizar a movimentação de dados e de instruções de E para o processador.

D

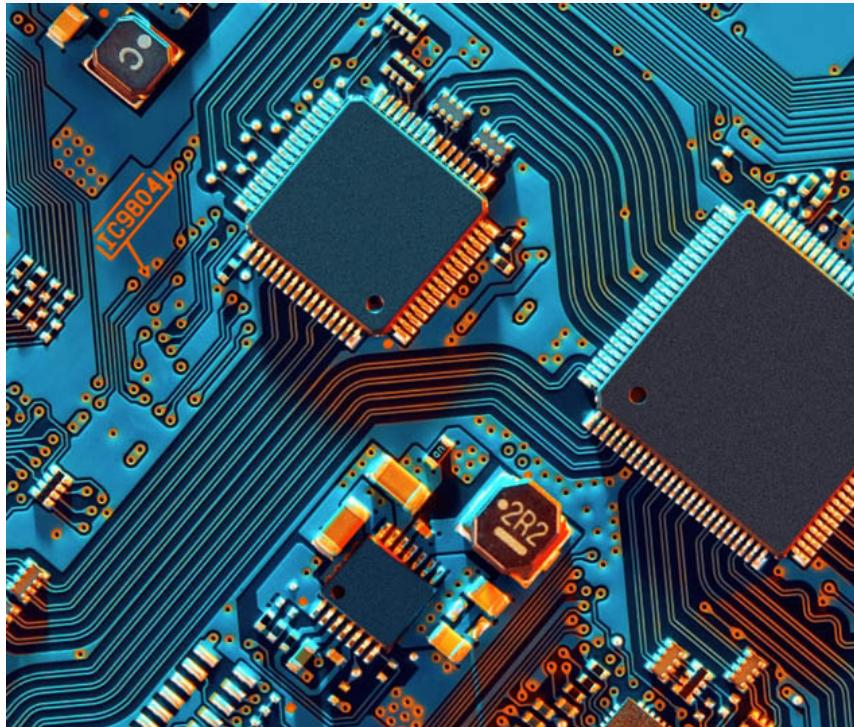
Determinar o período de duração de cada uma de suas atividades e controlar o sincronismo entre elas.

E

Determinar a cadência da execução das instruções do programa.

Parabéns! A alternativa B está correta.

Não confunda armazenamento dos dados a serem manipulados pelas unidades de cálculo com armazenamento da instrução a ser executada (ex.: ADD).



## 3 - Sistema operacional

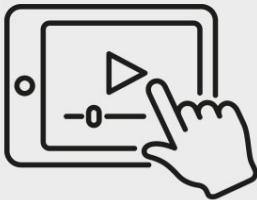
Ao final deste módulo, você será capaz de registrar o que é mais importante sobre o sistema operacional.

### Sistema operacional



# Definições de Sistema Operacional

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



## Definição

Um **sistema operacional** (SO) é um **conjunto de rotinas**, assim como um programa, porém funciona de maneira diferente de programas tradicionais, atuando como intermediário entre o usuário e o hardware de um computador, tornando sua utilização mais simples, rápida e segura.

Desse modo, o sistema operacional é:

O único programa executado em modo **núcleo (kernel)**, possuindo acesso completo ao hardware e execução de qualquer instrução possível.

Um programa de controle que comanda a execução dos programas do usuário e as operações dos dispositivos de E/S.

Um gerenciador de recursos (hardware) que dirige e aloca as partes de todo um sistema complexo.

## Único (kernel)

O **núcleo (kernel)** consiste no **conjunto de rotinas** que oferece serviços aos usuários, às suas aplicações e também ao próprio sistema.

O computador pode ser compreendido como uma **máquina de níveis** ou **máquina de camadas**, onde existem inicialmente **duas camadas**:

## Hardware

É a camada física, refere-se aos componentes físicos, tais como placas, fios, componentes eletrônicos.

## Software

É a camada lógica, refere-se a todas as abstrações do sistema de computação, tais como aplicativos e seus dados.

O usuário interage diretamente com o sistema operacional, como se o hardware não existisse. Diz-se, então, que a visão do usuário é **modular** e **abstrata**. Na verdade, não existem somente duas camadas, mas tantas quantas forem necessárias.

Nos primeiros computadores, a programação era feita através de painéis, tinha-se um maior conhecimento do hardware, porém a complexidade era muito grande.

O hardware em si deve ter pouca utilidade para o usuário, já o software permite oferecer melhor os serviços, e isso foi uma das motivações para o surgimento do Sistema Operacional.

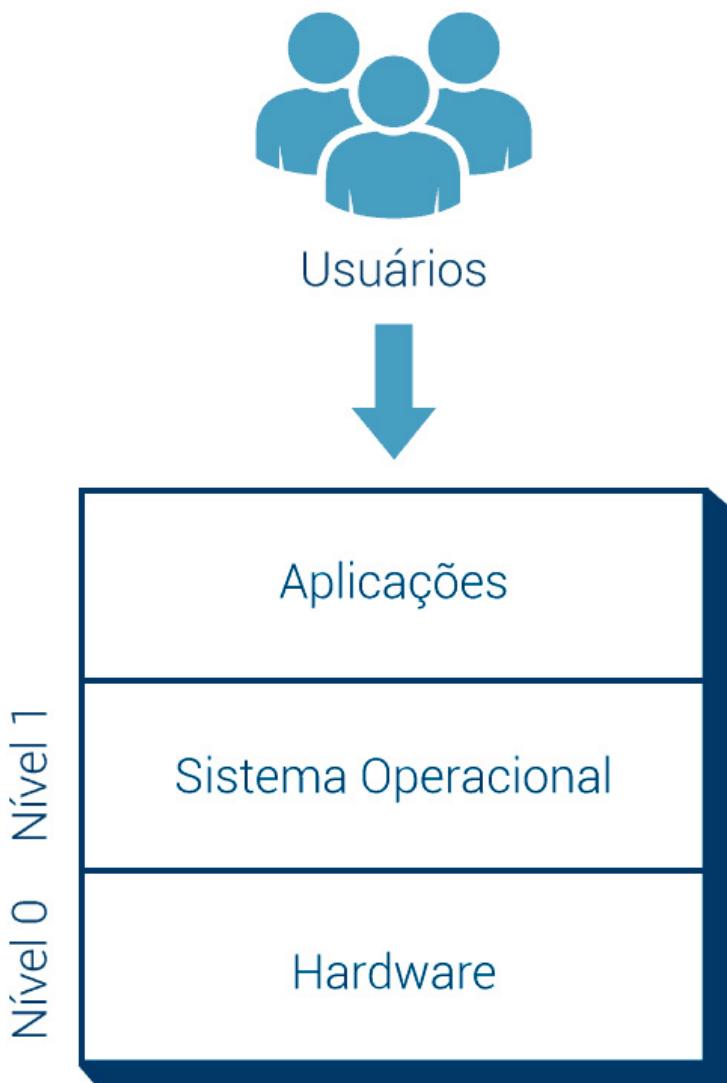
A partir da imagem ao lado, podemos dizer que:

**Os usuários** são as pessoas, máquinas, outros computadores.

**Os aplicativos** definem as maneiras como os recursos são usados, para resolver os problemas de computação dos usuários. Exemplo: Compiladores, BDs, videogames, programas comerciais.

**O sistema operacional** controla e coordena o uso do hardware entre os vários programas de aplicação, para os diversos usuários.

**O hardware** fornece recursos básicos de computação CPU, memória, dispositivos de E/S.



Máquina de camadas.

---

## Tipos de sistemas operacionais

Os **sistemas operacionais** podem ser enquadrados nos seguintes **tipos**:



Tipos de sistemas operacionais.

---

## Sistemas monoprogramáveis/monotarefas

O processador, memória e periféricos permanecem dedicados exclusivamente à execução de somente um programa. Um exemplo é o MS-DOS.

## Sistemas multiprogramáveis/multitarefas

Os recursos computacionais são compartilhados entre os diversos usuários e aplicações. Exemplos são Windows, Linux, macOS.

## Sistemas com múltiplos processadores

Possuem duas ou mais UCPs (CPUs) interligadas e trabalhando em conjunto. Sistemas como Linux e Windows oferecem esse suporte.

Vantagens de sistemas com múltiplos processadores:



### **Escalabilidade**



### **Disponibilidade**



### **Balanceamento de carga**

## Sistemas multiprogramáveis / multitarefas

Na imagem a seguir apresentaremos os sistemas multiprogramáveis:



Tipos de sistemas multiprogramáveis/multitarefas.

## Sistemas batch

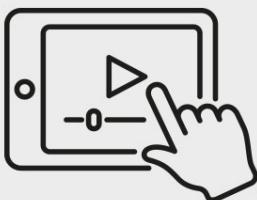
Processam tarefas de rotina sem a presença interativa do usuário. Exemplo: Processamento de apólices de companhia de seguro; relatório de vendas de uma cadeia de lojas.

## Sistemas de tempo compartilhado



## Sistema de Tempo Compartilhado

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



Permitem que múltiplos usuários remotos executem suas tarefas simultaneamente no computador.

Exemplo: Realização de consultas a um banco de dados.

## Sistemas em tempo real

Possuem o tempo como parâmetro fundamental. Exemplo: Linha de montagem de um carro.

Veja a seguir a diferença entre **sistemas forte e fracamente acoplados**:

## Sistemas fortemente acoplados

Existem vários processadores compartilhando uma única memória física e dispositivos de E/S sendo gerenciados por apenas um SO.



## Sistemas fracamente acoplados

Existem dois ou mais sistemas computacionais conectados através de linhas de comunicação.

Antigamente, os sistemas fortemente acoplados eram restritos aos sistemas de grande porte, mas hoje em dia os computadores pessoais e estações de trabalho possuem múltiplos processadores, com suporte dos respectivos sistemas operacionais.

Um exemplo típico de sistema fracamente acoplado atual é o modelo de redes de computadores.

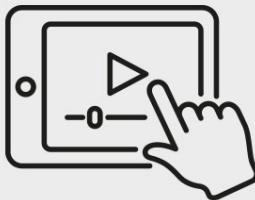
Em um sistema operacional distribuído, todos os terminais na rede são tratados como um sistema único. Já um cluster é constituído de servidores interligados de modo transparente ao usuário, na web, por exemplo.



## Sistema operacional

Veja agora mais detalhes sobre os sistemas operacionais.

Para assistir a um vídeo sobre o assunto, acesse a versão online deste conteúdo.



## Tendências

Sistemas operacionais tendem a ser **virtualizados** (disponibilizados na web, sem necessidade de instalação de software) e oferecidos para os usuários usarem em uma infraestrutura na **nuvem**.

A virtualização de servidores reproduz um computador inteiro em software, que executa um sistema operacional inteiro. O sistema operacional executa um aplicativo.

Já os contêineres, conforme a IBM (2020), adotam uma abordagem alternativa. Eles compartilham um kernel do SO subjacente (ou seja, o núcleo), executando apenas o aplicativo e tudo o que dele depende, como bibliotecas de software e variáveis de ambiente. Isso os torna menores e mais rápidos de implantar.

## Falta pouco para atingir seus objetivos.

### Vamos praticar alguns conceitos?

#### Questão 1

Na literatura, encontram-se diversas classificações de sistemas operacionais (SO). Uma delas classifica os SO nos seguintes tipos: Em lotes (batch), de tempo compartilhado (time sharing) e de tempo real (real time). Sobre esse assunto, considere as assertivas abaixo:

- I. Sistemas operacionais de tempo compartilhado e tempo real são dependentes do tempo de execução de cada programa, porém se baseiam em diferentes parâmetros de eficiência.
- II. Sistemas do tipo lote (batch) podem ser multitarefa.
- III. Sistemas operacionais de tempo real são mais adequados para executar rotinas do tipo lote (batch), se for desejado obter resultados no tempo mais curto possível.
- IV. Ao executar um programa do tipo batch, um sistema operacional de tempo compartilhado se comporta como um sistema monotarefa, desativando as rotinas de alternância de programas em execução.

Estão corretas as assertivas:

A

I e IV.

B

II, somente.

C

III, somente.

D

I, II, III e IV.

E

somente a IV.

Parabéns! A alternativa B está correta.

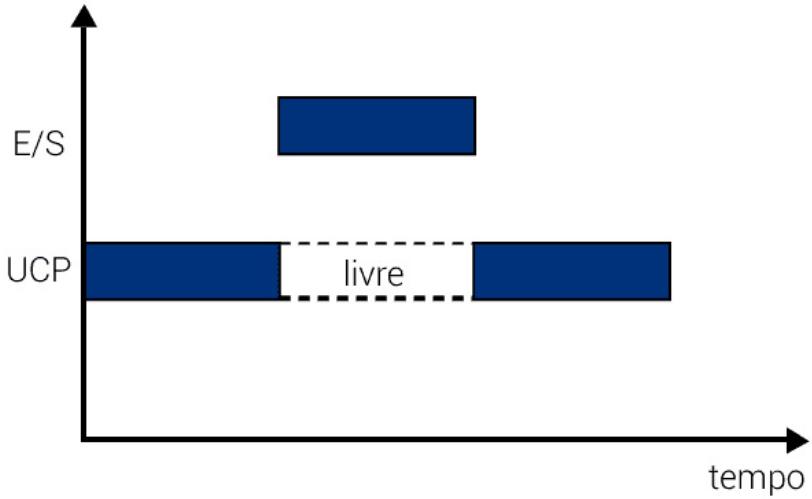
Sistemas de tempo compartilhado permitem que múltiplos usuários remotos executem suas tarefas simultaneamente no computador. Sistemas de tempo real possuem o tempo como parâmetro fundamental.

III: O sistema de processamento em lote (batch) processa tarefas de rotina sem a presença interativa do usuário, independentemente do tipo de sistema operacional que o executa.

IV: A afirmação da execução do sistema operacional de tempo compartilhado não procede.

## Questão 2

A ilustração gráfica a seguir representa um sistema que utiliza uma técnica na qual as solicitações de entrada ou saída de dados e a execução de uma única tarefa devem ser executadas pela CPU em alternância de tempo. Apenas uma tarefa pode utilizar os recursos disponíveis até que ela seja encerrada, dando lugar a outra tarefa.



Essa técnica é conhecida como:

- A** Multiprogramação.
- B** Monousuário.
- C** Multiusuário.
- D** Monoprogramação.
- E** Multitarefa.

Parabéns! A alternativa D está correta.

Nos Sistemas monoprogramáveis, o processador, memória e periféricos permanecem dedicados exclusivamente à execução de um programa.

# Considerações finais

Continuando com os estudos em organização e arquitetura de computadores, o estudante aqui pôde aprender sobre os componentes *vitais* de um computador, que são o processador, a memória e os dispositivos de entrada e saída.

Se o estudante quiser realizar uma grosseira aproximação com a *máquina humana*, o processador seria a parte do cérebro que executa tarefas, a memória seria a parte do cérebro que *guarda as informações* e os dispositivos de entrada e saída seriam os diversos órgãos que possibilitam captar recursos externos e oferecer um resultado após a transformação no interior do corpo.

Essa analogia, apesar de um tanto simplista e até discutível, serve para uma primeira tentativa de aproximação da premissa de que podemos extrair melhores resultados e expandir suas possibilidades ao conhecermos melhor como são e como funcionam tanto a nossa máquina, de carne e osso, quanto a máquina que constitui um sistema de computação.

A seguir, abordaremos o tema por meio de perguntas e respostas.

Para ouvir o áudio, acesse a versão online deste conteúdo.



## Explore +

Para reforçar os conhecimentos adquiridos, realize mais [exercícios](#).

## Referências

HARRIS, D.; HARRIS, S. **Digital Design and Computer Architecture**. 2. ed. San Francisco: Morgan Kaufmann, 2012.

HENNESSY, JOHN L. **Organização e projeto de computadores: A interface hardware/software**. 2. ed. Rio de Janeiro: LTC, 2000.

IBM. **What is Virtualization?** Consultado na internet em: 18 mar. 2020.

MACHADO, F. B.; MAIA, L. P. M. **Arquitetura de sistemas operacionais**. 4. ed. Rio de Janeiro: LTC, 2007.

MONTEIRO, M. **Introdução à organização de computadores**. 5. ed. Rio de Janeiro: LTC, 2007.

MURDOCCA, M. J.; HEURING, V. P. **Introdução à Arquitetura de Computadores**. Rio de Janeiro: Campus, 2000.

PATTERSON, D. A. et al. **Computer Architecture, a quantitative approach**. 5. ed. San Francisco: Morgan Kaufmann, 2011.

PATTERSON, D. A. et al. **Organização e projeto de computadores**. 2. ed. Rio de Janeiro: LTC, 2000.

POUNDSTONE, William. **John von Neumann – Biography, Accomplishments, Inventions & Facts**. Consultado na internet em: 18 mar. 2020.

STALLINGS, W. **Arquitetura e organização de computadores**. 8. ed. São Paulo: Pearson Prentice Hall, 2010.

TANEMBAUM, A. S.; BOS, H. **Sistemas Operacionais Modernos**. 3. ed. São Paulo: Pearson, 2010.

TANENBAUM, A. S. **Organização estruturada de computadores**. 5. ed. Rio de Janeiro: LTC, 2007.



## Material para download

Clique no botão abaixo para fazer o download do conteúdo completo em formato PDF.



Download material

O que você achou do conteúdo?



Relatar problema