

# Arquitetura e Organização de Computadores

Caros alunos, as videoaulas desta disciplina encontram-se no AVA  
(Ambiente Virtual de Aprendizagem).



# Abertura

## **Apresentação**

A disciplina *Organização e Arquitetura de Computadores* visa apresentar e discutir alguns conceitos e princípios básicos que envolvem a organização interna de um sistema computacional (computador), seus componentes e interconexões. É muito importante que o profissional de computação saiba como funciona os componentes integrantes de um computador, pois tendo esse conhecimento, saberá entender, por exemplo, a capacidade e a incapacidade de um computador.

## **Apresentação do Professor**

- Meu nome é Kleber Marcio de Souza, possuo graduação em Engenharia da Computação pela Universidade Norte do Paraná (2004). Concluí o mestrado no curso de Engenharia Elétrica na Universidade Estadual de Londrina na área de controle avançado (2014); sou licenciado em Matemática pela Universidade Estadual de Ponta Grossa (2017) e possuo Especialização em Informática na Educação pela Universidade Estadual de Londrina (2007).

## **Objetivos**

- Entender os componentes fundamentais de um sistema de computador;
- Apresentar uma visão geral sobre a organização hierárquica dos computadores descrevendo sua estrutura e funcionamento.

# Unidade 1

**Fundamentos da Computação.**

# Introdução da Unidade

**Olá, amigo(a) discente! Seja bem-vindo(a)!**

Acreditando que exista um grande relacionamento entre hardware e software, pretendemos nessa unidade discutir os principais conceitos relacionados a Arquiteturas e organização de computadores. Essa teoria se faz necessária para quem pretende ser um profissional da área tecnológica.

Nesta unidade, vamos estudar os conceitos gerais dos fundamentos da computação, como foi organizado um computador e como ele tornou-se essa máquina tão importante nos dias atuais. Para isso, estudaremos conceitos como: Hierarquias de níveis de computador, barramentos e modelo de organização.

## Objetivos

- Ter uma visão geral da arquitetura do computador.
- Entender como se organizam os sistemas de computador.

## Conteúdo Programático

**Aula 1:** Conceitos gerais da computação.

**Aula 2:** Organização de sistemas de computador.



Você poderá também **assistir às videoaulas** em seu celular! Basta apontar a câmera para os **QRCodes** distribuídos neste conteúdo.

Pode ser necessário instalar um aplicativo de leitura QRcode no celular e efetuar login na sua conta Gmail.



### **Apresentação da Disciplina**

Utilize o QRcode para assistir!



### **Minicurrículo**

Utilize o QRcode para assistir!



# Aula 1 Conceitos gerais da computação

Olá, estudante, como vai? Vamos começar agora um estudo muito importante sobre os computadores. Você pode estar pensando: mas para que eu vou estudar hardware de computadores? Como usuário final talvez você não precisaria pensar nos detalhes dessa máquina, porém como profissional da área tecnológica é essencial esse aprendizado. Entender os hardwares nos ajuda a entender muitos erros que acontecem em nível de software. Como por exemplo, entender por que em determinadas máquinas o software tem uma certa lentidão de execução e em outras máquinas não. Outro exemplo bem legal de ser mostrado é que, para desenvolver compiladores, é necessário entender o ambiente de hardware específico para o qual você estará projetando. Poderíamos citar aqui diversos exemplos justificando o estudo dessa matéria, porém espero que você tire as suas dúvidas lendo o nosso material.

Tá, tudo bem. Mas o que é então **Arquitetura e Organização de Computadores**?

**Vamos separar esses dois conceitos para entender:**

- **Arquitetura:** refere-se aos atributos de um sistema visíveis a um programador, ou seja, atributos que possuem influência direta sobre a execução lógica de um programa (STALLINGS, 2011). Vamos para alguns exemplos então: Atributos arquiteturais podem incluir os conjuntos de instruções, números de bits utilizados para representar diversos tipos de dados (números e caracteres) e mecanismos de entrada e saída.
- **Organização:** trata de questões de hardware como sinais de controle (exemplo: frequência de clock), métodos de sinalização e tipos de memória; compreende então a aspectos físicos dos sistemas de computação (NULL, 2010).

Para fechar nosso raciocínio, vamos pensar de uma maneira prática. É uma questão de projeto arquitetural se um computador terá uma instrução de multiplicação, por exemplo. É uma questão organizacional se essa instrução será implementada por uma unidade de multiplicação especial ou por algum outro método que chegue ao mesmo resultado.

Muitos fabricantes de computadores oferecem uma grande variedade de famílias de modelos de computador, todos com a mesma arquitetura, mas com diferenças nas suas organizações.



## Videoaula 1

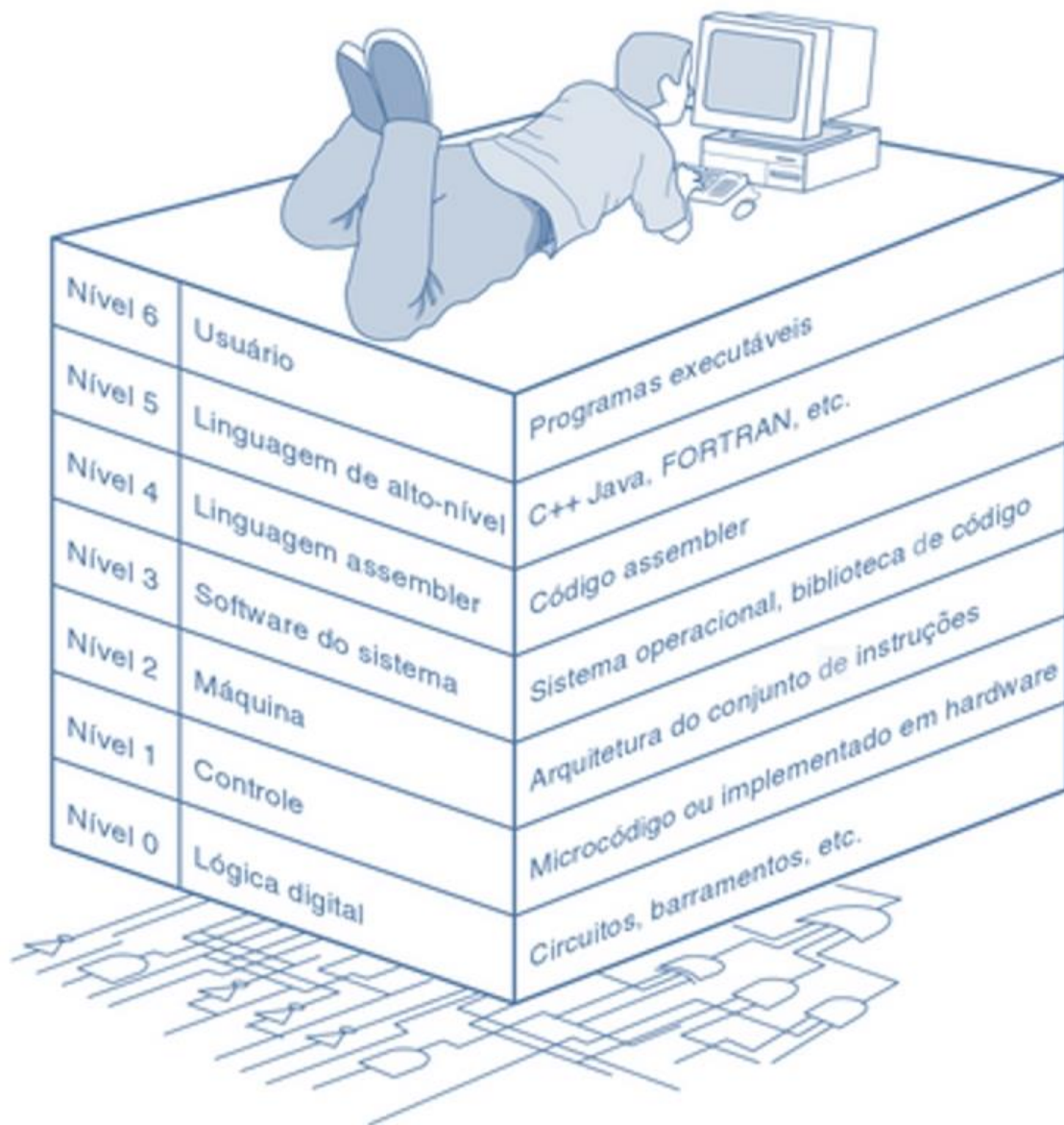
Utilize o QRcode para assistir!

Assista ao vídeo para entender melhor o assunto.



## A Hierarquia de níveis do computador

Depois de identificar os conceitos de arquitetura e organização, vamos agora aprender outro conceito importante sobre o computador. Autores como Stallings (2011), Tanenbaum (2013) e Null (2010), destacam que a organização dos sistemas de computação pode ser pensado de uma maneira abstrata, onde imaginamos a máquina como sendo construída em uma hierarquia de níveis, onde cada nível tem uma função específica dentro do sistema. Observe a figura 1, onde mostra as camadas comumente aceitas dos modernos sistemas de computação.



**Figura 1:** Níveis abstratos dos modernos sistemas de computação.

Fonte: NULL,2010.

Notando a figura acima no seu topo, vemos o nível 6, o nível do usuário, é composto de aplicações utilizadas pelos usuários. É nesse nível que executamos programas como processadores de texto, jogos e navegadores de internet. O nível 5 é onde podemos encontrar as linguagens de programação como C, java e C++. Essas linguagens são traduzidas para uma linguagem conhecida da máquina, que são as linguagens simbólicas. Note que nesse nível, o usuário começa a enxergar pouco os níveis inferiores.

No nível 4, já estamos no lugar das linguagens simbólicas. Como notamos anteriormente, as linguagens de alto nível são traduzidas para as linguagens simbólicas que por sua vez é traduzida para linguagem de máquina.

O nível 3, o Nível de software do sistema, trata das instruções do sistema. É nesse nível que encontramos o sistema operacional, cuja função é: Multiprogramação, proteção da memória, sincronização de processos e outras funções importantes.

O nível 2, a Arquitetura do conjunto de instruções, consiste na linguagem de máquina reconhecida pela arquitetura particular do sistema de computação. As linguagens de máquina são executadas diretamente pelos circuitos eletrônicos.

O nível 1, o Nível de controle, é onde a unidade de controle se assegura de que as instruções sejam decodificadas e executadas adequadamente, e também direcionando os fluxos de dados para os seus devidos destinos. Unidades de controle são projetadas de duas maneiras: projetadas em hardware ou microprogramadas. Em unidades de controle implementadas em hardware os sinais de controle partem diretamente de componentes de lógica digital e por esse motivo também, essas unidades são muito rápidas, porém uma vez implementadas, são difíceis de serem removidas.

A microprogramação é um programa escrito em uma linguagem de baixo nível que é **implementada diretamente pelo hardware**. As instruções de máquina produzidas no nível 2 são alimentadas nesse programa, que interpreta as instruções pela ativação do hardware capaz de executar a instrução original (NULL, 2010).

Por fim, amigos, estamos no nível 0, nas profundezas! O nível da lógica digital é onde encontramos os sistemas físicos da computação: portas lógicas, registradores e fios.



### Videoaula 2

Utilize o QRcode para assistir!

Assista ao vídeo para fixar o seu estudo.





## Indicação de Leitura

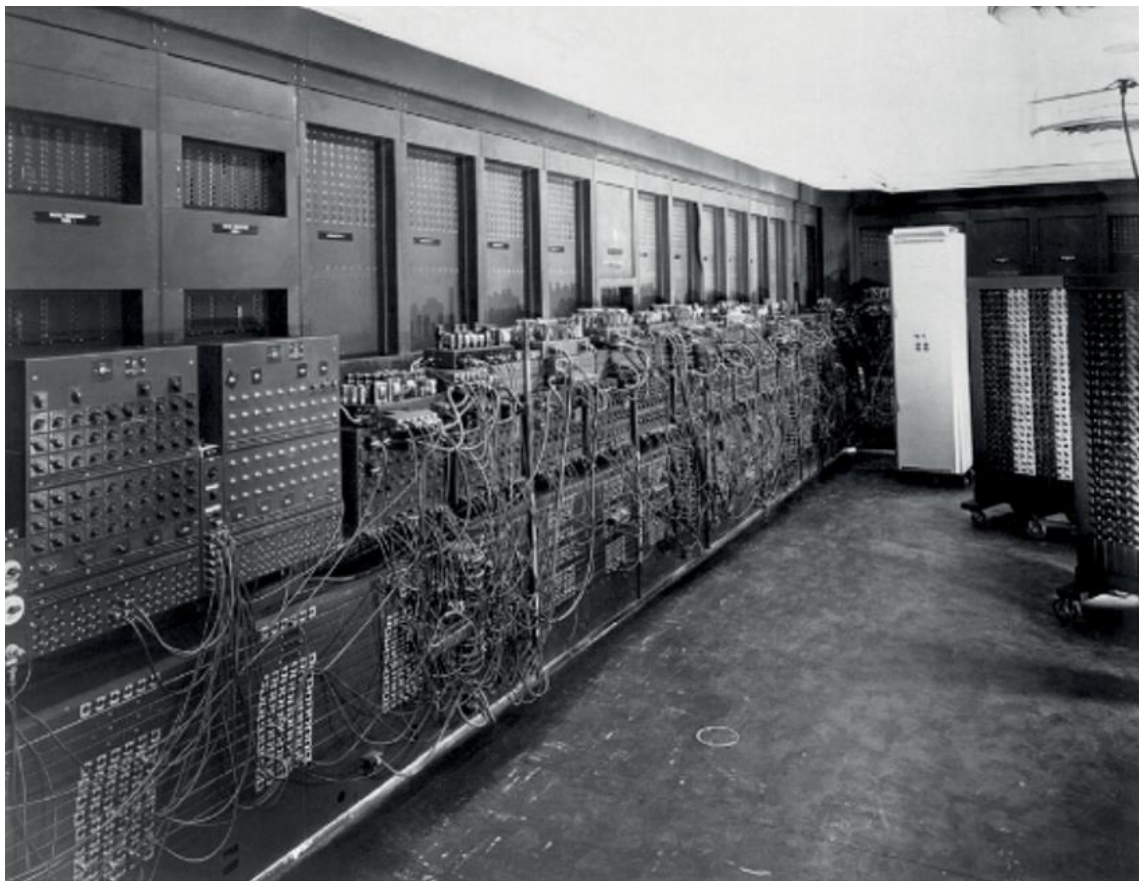
### Leitura Obrigatória

Para complementar seus estudos até aqui, leia o livro de TANENBAUM os itens 1.1 a 1.2. TANENBAUM é um escritor muito conhecido na área de arquitetura.

< <https://plataforma.bvirtual.com.br/Acervo/Publicacao/355>>. Acesso em: 08.04.2020.

## Marco da Arquitetura de Computadores

Quando falamos em computação não tem como fugir da história dos computadores, pelo fato de que se entendermos sua evolução estaremos também entendendo o seu presente, basta lembrar que o impacto dessa tecnologia na nossa sociedade é imenso e nossa dependência dela fica cada vez maior.



**Figura 2:** ENIAC – Primeiro computador digital eletrônico.

Fonte: <<https://pt.wikipedia.org/wiki/ENIAC>>. Acesso em: 20.04.2020.

Observe a figura 2: esse imenso computador pertenceu à **primeira geração** de computadores valvulados (1945-1955), seu nome era ENIAC (*Eletronic Numerical Integrator and computer*), ocupava um espaço muito grande e para a época era uma grande invenção. Suas

especificações técnicas eram: 18.000 válvulas e 1500 relés, pesava 30 toneladas e consumia 140 kW de energia. Hoje qualquer calculadora de bolso faz o que ENIAC fazia.

Depois tivemos a **segunda geração** de computadores (1954-1965) conhecida como geração dos transistores, isso mesmo, aquele componente estudado nas aulas de eletrônica. Os transistores provocaram mudança na eletrônica da época e, no fim da década de 50 os computadores a válvula ficaram obsoletos Côrrea (2016, p. 11). A propósito, você sabe dizer quantos transistores um processador Intel Core i7 tem? Faça uma pesquisa rápida pela internet e descobrirá que passa dos 4 bilhões de transistores, fantástico não? Continuando, em 1961 foi lançado o primeiro computador comercial, conhecido como PDP-1 (Programmed Data Processor) como mostrado na figura 03. Assim nascia a indústria de minicomputadores.

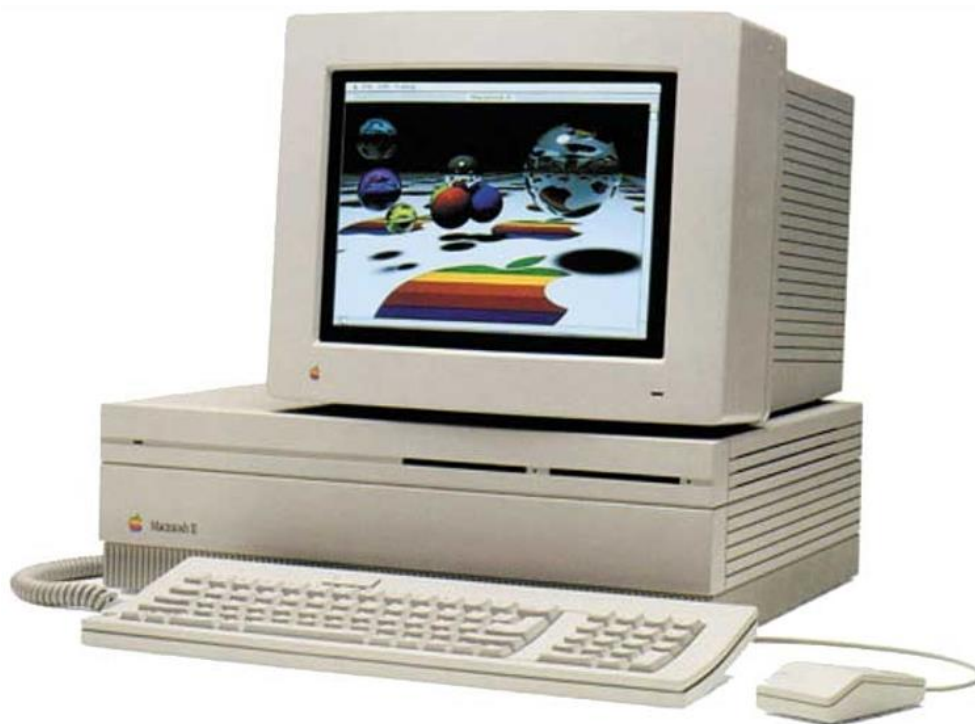


**Figura 03:** Primeiro computador comercial PDP-1.

Fonte: <<https://en.wikipedia.org/wiki/PDP-1>>. Acesso em: 20.04.2020.

Seguindo a linha de evolução, tivemos a **terceira geração** de computadores (1965-1980), nessa época desenvolveu-se uma tecnologia onde dezenas ou centenas de transistores fossem colocados em um único chip, ficando conhecido como circuitos integrados. A partir disso, vários modelos de PC's começaram a ser fabricados. Note que com o avanço da eletrônica, os computadores foram ficando cada vez menores.

A época que realmente marcou a evolução dos computadores foi a **quarta geração** (1980 – atual). Essa época foi conhecida como era do computador pessoal, a tecnologia VLSI (*Very Large Scale Integration*) colocava milhões de transistores em um único chip. Essa tecnologia produziu cada vez mais computadores menores e também a indústria do software foi crescendo junto com hardware, tanto que em 1980 a Microsoft lança seu primeiro sistema operacional o *Windows*. Parece um texto nostálgico, mas é interessante saber a evolução da tecnologia e como ela mudou a vida no planeta terra. Já pensou em um mundo sem computadores? Por fim, hoje a computação está em tudo, acredito que você esteja lendo esse texto por um dispositivo tecnológico. Abaixo uma foto de um *personal computer*, um *macintosh* da década de 80.



**Figura 04:** Macintosh década de 80.

Fonte: <<http://macmais.com.br>>. Acesso em: 20.04.2020.

#### Indicação de Vídeo

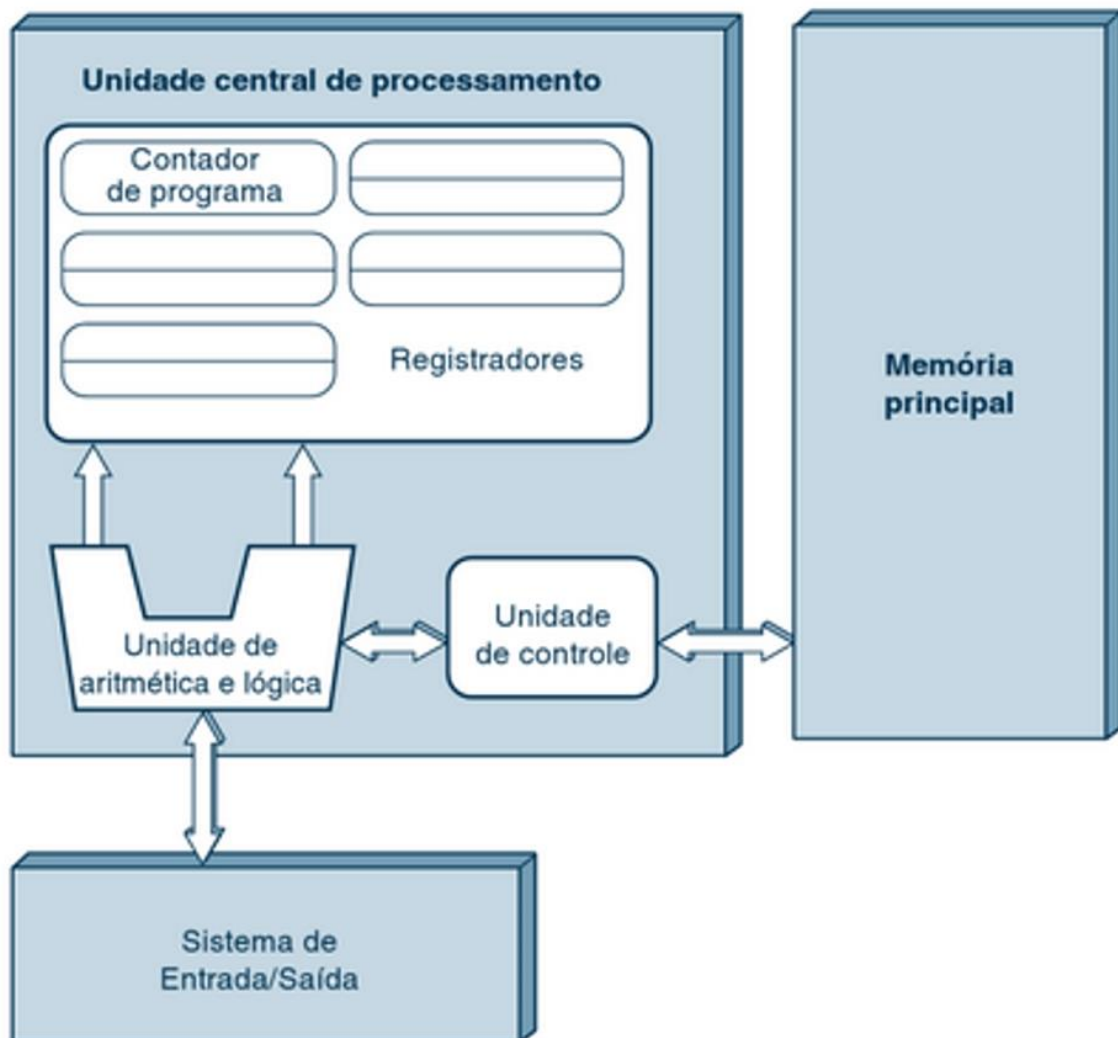
**Assista esse vídeo: Breve História do Computador – 4:36 Min**

Disponível em: <<https://www.youtube.com/watch?v=uQuYPRYPiM>>. Acesso em: 20.04.2020.  
- 13 min.

## O modelo de Von Neumann

Primeiramente é interessante saber que Von Neumann era um famoso matemático húngaro que naturalizou-se americano e foi um brilhante matemático do século XX. Foi um dos construtores do gigante computador ENIAC, citado no tópico anterior. Neumann achava que programar o ENIAC era muito trabalhoso e cansativo. Então percebeu que dados e programas deviam ser guardados em memória, e por isso, projetou uma máquina que poderia implementar tal ideia.

A máquina proposta por Von Neumann é composta por três sistemas de hardware: Uma UCP (unidade central de processamento) juntamente com uma unidade de controle, registradores (pequenas memórias) um contador de programa e uma ULA, um sistema de memória principal que armazena programas para serem executados; e um sistema de E/S(entrada e saída). A figura 5 mostra como se organiza a máquina von Neumann.



**Figura 05:** Arquitetura de von Neumann.

Fonte: NULL (2010).

As partes dessa arquitetura têm as seguintes funções:

- **Memória:** armazena os dados e os programas durante o processamento.
- **Unidade Lógica e aritmética (ULA):** com auxílio de registradores, essa parte realiza operações lógicas e aritméticas. Note na figura acima que o sistema faz passar através da ULA toda sua E/S.
- **Sistema de entrada e saída:** recebe as entradas dos dados e dos programas que posteriormente fornece os resultados dos processamentos.
- **Unidade de Controle:** sua função é controlar os três itens acima (memória, ULA, Equipamentos de entradas e saídas).

Na arquitetura de Von Neumann o programa utilizado deve ser alocado na memória. Isso difere em relação aos primeiros computadores, onde somente os dados eram armazenados na memória e os programas eram implementados pela manipulação de comutadores ou fios.



### Videoaula 3

Utilize o QRcode para assistir!

Assista agora ao vídeo para fixar o seu estudo.



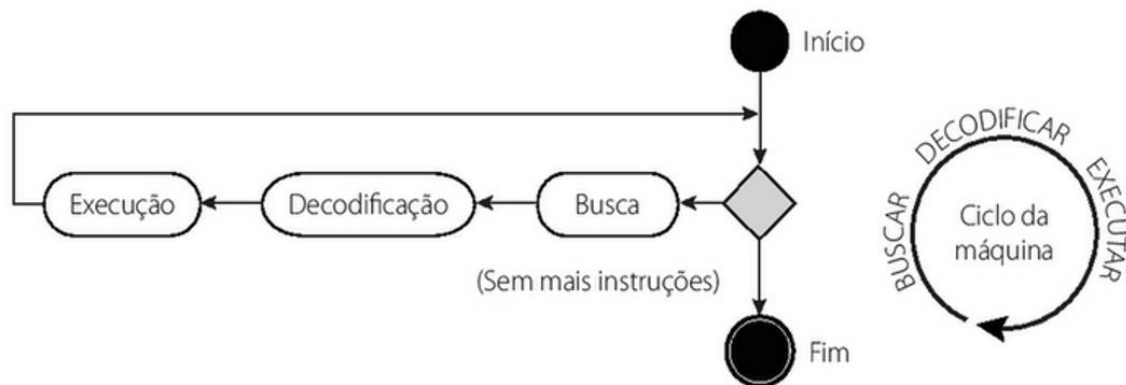
Ainda nessa arquitetura, os programas executam programas no que é conhecido como ciclo de execução de von Neumann (conhecido **como ciclo de busca-decodificação-execução**). A execução de um programa de computador deve seguir alguma ordem, pois caso contrário diversos tipos de erros podem acontecer. Para executar instruções, a CPU utiliza ciclos de máquina – basicamente um ciclo de máquina são procedimentos para o processamento de dados. O ciclo de máquina possui três fases:

1. **Busca:** para controle desse ciclo a unidade de controle contém dois registradores: o contador de instruções e o registrador de instruções. No contador é armazenado o endereço da próxima instrução a ser executada, dessa forma a unidade de controle mantém-se informada sobre a posição do programa em execução. Então o registrador guarda as instruções de máquina que está em execução.
2. **Decodificação:** a unidade de controle decodifica o que está no registrador de instruções. Desse procedimento resulta em um código binário para uma operação que o sistema vai realizar.
3. **Execução:** circuitos específicos da ULA são acionados para que receba os dados de entrada e a tarefa indicada na instrução seja executada. Por exemplo: A CPU ou UCP



pode comandar a ULA somar o conteúdo de dois registradores de entrada, e depois colocar o resultado no registrador de saída.

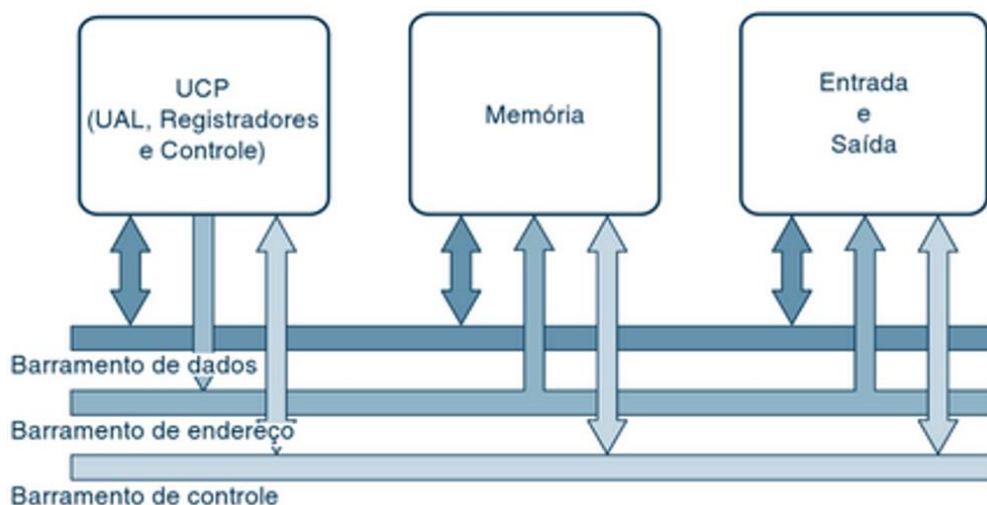
Ao final da execução, a unidade de controle recomeça o ciclo, partindo da fase de busca. Observe a figura abaixo.



**Figura 06:** Ciclo de máquina.

Fonte: CÔRREA (2016, p.27).

O modelo de von Neumann foi evoluindo e estendido de modo que programas e dados que até então estava sendo armazenados em meios de armazenamento lentos, tal como o disco rígido, pudessem ser copiados para meios de memória volátil de acesso mais rápido, tal como a RAM, antes da execução. Dessa forma foi criado o modelo de barramentos, que é mostrado na figura 7.



**Figura 07:** Arquitetura de von Neumann, adicionando um barramento de sistema.

Fonte: NULL (2010).

**Vamos verificar as funções de cada barramento então:**

- **Barramento de dados:** são compostos por diversas linhas de comunicação. Cada linha transporta apenas um bit de cada vez, então os números de linhas de um barramento dependem do tamanho da palavra da arquitetura de um computador. Por exemplo: um computador que usa palavra de 16 bits, precisa de 16 linhas para comunicação.
- **Barramento de endereços:** permite o acesso físico na memória do computador para leitura de dados que é solicitado pelo processador ou outro dispositivo.
- **Barramento de controle:** controla a comunicação entre a CPU e a memória. Vamos usar um exemplo do uso desse barramento: para buscar uma instrução na memória, primeiramente a CPU coloca o endereço da memória em seu barramento de endereços. Então ela ativa o barramento de controle para informar a memória que quer ler uma instrução.

### **Indicação de Vídeo**

**Agora assista o vídeo que explica sobre a arquitetura de Von Neumann.**

Disponível em: <<https://www.youtube.com/watch?v=V5qE-u6jGo4>>. Acesso em: 20.04.2020.

### **Modelos diferentes do modelo de Von Neumann**

Até em tempos atuais, quase todos os computadores de uso gerais seguem o projeto de von Neumann, porém a transmissão de dados entre a CPU e a memória leva o que chamamos de **gargalo de von Neumann**. Nos computadores modernos a troca de dados entre o processador e a memória é muito menor do que a taxa com o processador pode trabalhar.

sistemas mais rápidos [...] de baixo custo e compatíveis com a grande quantidade de software comercialmente disponível[...]. Vários diferentes subcampos caem na categoria de não von Neumann, incluindo redes neurais (usando ideias de modelos do cérebro como paradigma de computação), algoritmos genéticos (explorando ideias de biologia e da evolução de DNA), computação quântica, computação de fluxo de dados e computadores paralelos. Todos estes possuem algo em comum – a computação é distribuída entre diferentes unidades de processamento que atuam em paralelo. Destes a computação paralela é a mais popular. (NULL,2010, p.66-67).

Como vê, aluno(a), sempre surgem estudos novos para melhorar o desempenho dos computadores. O paralelismo é uma tecnologia comum nos computadores, onde sua função é dividir um problema em problemas menores, que então são resolvidos em forma paralela.

Amigo(a) aluno(a)! Nesta aula começamos vendo um pouco da evolução dos computadores e também alguns aspectos conceituais da arquitetura. Você pode notar que o mais importante desenvolvimento isolado na computação que nos afeta como desenvolvedores tenha sido a introdução do conceito de programa armazenado da máquina von Neumann. Embora existam

outros modelos arquitetônicos, a arquitetura de von Neumann predomina nos atuais computadores de uso geral atuais.

Para um aprofundamento melhor dos conteúdos estudados até aqui, leia os apontamentos dos livros indicados na **leitura obrigatória**, porque são livros que aprofundam os conceitos e também servem como referência para nossas futuras avaliações.

### **Indicação de Leitura**

Leia toda **unidade 01 do livro da escritora Ana Grasielle**, ele aprofunda os conteúdos e está disponível na sua biblioteca Virtual.

Corrêa, Ana Grasielle Dionísio. **Organização e Arquitetura de Computadores**. Pearson Educational do Brasil. 1 ed. 2016.

Disponível em: <<https://plataforma.bvirtual.com.br/Acervo/Publicacao/124147>>. Acesso em: 20.04.2020.



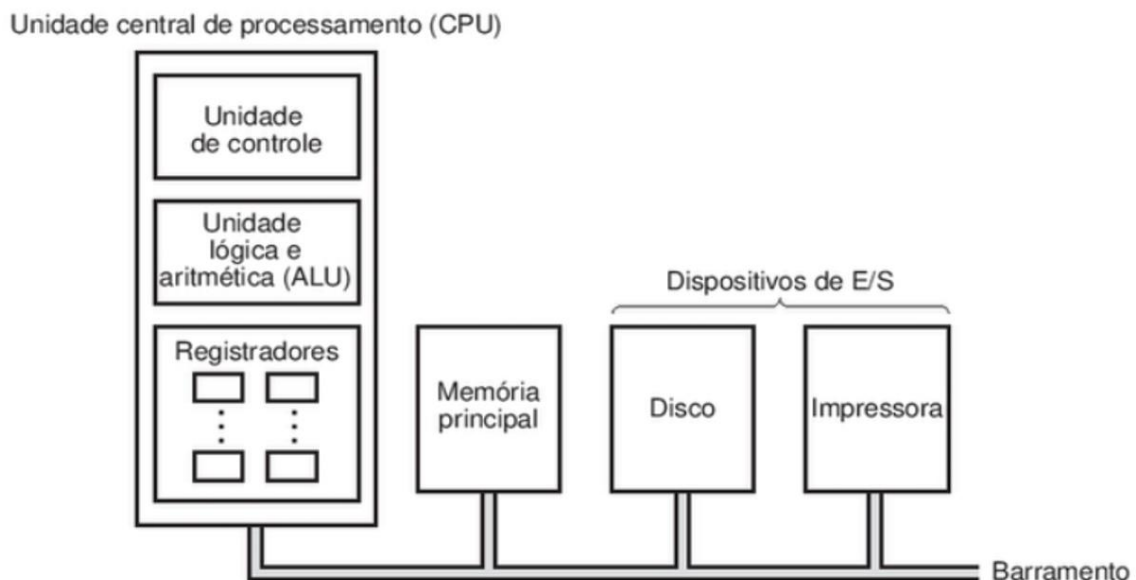
## Aula 2 Organização de sistemas de computador

Olá, aluno(a)! Vamos começar uma nova aula importante. Na aula anterior estudamos os conceitos fundamentais que estruturaram o desenvolvimento da computação. Nessa aula estudaremos os componentes principais na estrutura de um computador.

### Organização da CPU

Apesar de que os atuais computadores apresentam arquiteturas e organizações diferenciadas, vamos aqui, caro(a) aluno(a), estudar um sistema mais simples, porém que é base de todo sistema computacional atual (CÔRREA, 2016).

A organização de um computador simples com barramento é mostrada na figura 1 a CPU é o componente pensante do sistema todo. Sua principal função é executar os programas que estão na memória principal. Cada instrução é buscada na memória pelo processador, que a examina e executa uma após a outra.



**Figura 1:** Organização de um computador muito simples, com 1 processador e 2 dispositivos de E/S.

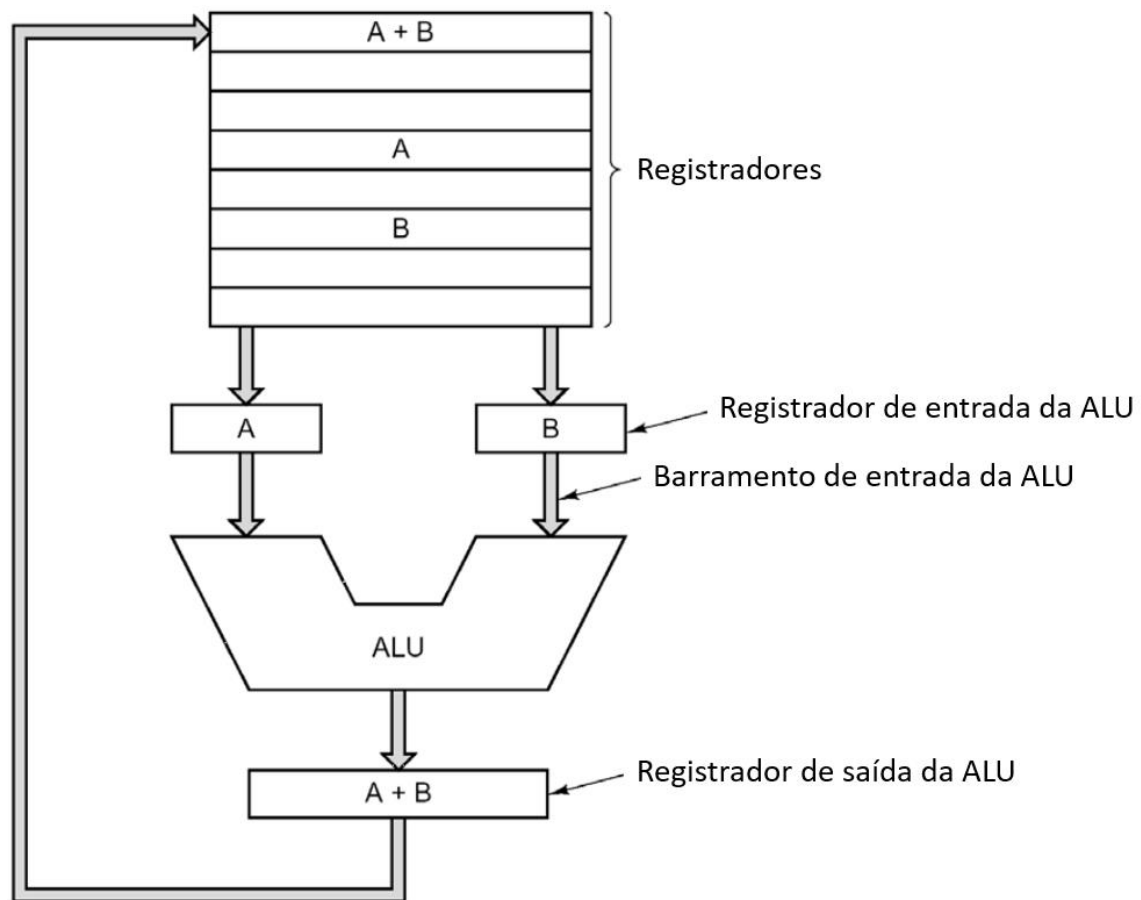
Fonte: TANEBAUM (2013, p.29).

Como podem notar pela figura, os componentes são conectados por um barramento-conjunto de fios paralelos que transmitem endereços, dados e sinais de controle. Os barramentos podem ser externos a CPU conectando-se a memória e aos dispositivos E/S e também pode ser

interna conectando componentes internos da CPU. Atualmente os computadores contêm vários barramentos.

A CPU é composta de alguns componentes: a unidade de controle é responsável pela busca de informações na memória e identificar seu tipo; a ULA efetua operações matemáticas (SOMA, MULTIPLICAÇÃO) e booleanas (AND, OR) para executar as instruções. Por fim, os registradores são usados como memória para armazenagem de dados temporários, esse componente tem uma execução de leitura e escrita muito rápida pelo fato de estar internamente a CPU.

Internamente a CPU no modelo Von Neumann pode ser mostrada com mais detalhes na figura 2.



**Figura 2:** Caminho de dados em uma CPU.

Fonte: TANENBAUM (2013).

Mas, afinal, como os dados trafegam pela CPU? A figura acima nos mostra o que é conhecido como caminho de dados (TANENBAUM, 2013). O caminho é composto de registradores, ALU e barramentos.

Como pode ser visto, os registradores alimentam dois registradores de entrada da ALU que são representados por A e B na figura. A ALU executa diversas operações sobre suas entradas produzindo um resultado no registrador de saída. Essa saída pode ser armazenada em um

outro registrador que posteriormente pode ser armazenada em uma memória. Temos que lembrar queridos alunos, que nem todo projeto tem os registradores A e B de saída.

### Fique atento!

Em computação uma palavra é um grupo de 8, 16, 32 ou 64 bits. Uma palavra de 8 bits é chamada de *byte*.

As instruções podem ser divididas em duas categorias: registrador-memória ou registrador-registrador (TANENBAUM, 2013). Instruções registrador-memória permitem que “palavras” sejam buscadas nos registradores e trazidas para as entradas da ALU, onde podem ser usadas como instruções. Palavras são unidades de dados movimentadas entre registradores e memória. Uma palavra, por exemplo, pode ser um número inteiro. Também existem outras instruções registrador-memória que permitem que registradores voltem a memória para armazenagem.

O outro tipo de instrução é a registrador-registrador. Essa instrução busca dois operandos nos registradores, trazem os dois para as entradas da ALU, efetua alguma operação com eles, e armazena o resultado em um dos registradores. O processo de passar dois operandos pela ALU e armazenar o resultado é que chamamos de caminho de dados e é utilizado na maioria das CPU's. Então, por isso, quanto mais rápido for o ciclo do caminho de dados, mais rápido será o funcionamento da máquina.



#### Videoaula 1

Utilize o QRcode para assistir!

Assista agora ao vídeo para fixar o seu estudo.



**A CPU executa cada instrução em uma série de pequenas etapas (TANENBAUM, 2013). Veja as etapas:**

1. Trazer a próxima instrução da memória até o registrador.
2. Alterar o contador de programa para indicar a próxima instrução.
3. Determinar o tipo de instrução trazida.
4. Se a instrução usar uma palavra na memória, determinar onde essa palavra está.
5. Trazer a palavra para dentro de um registrador da CPU, se necessário.

6. Executar a instrução.
7. Voltar a etapa 1 para iniciar a execução da instrução seguinte.

Essa sequência de etapas costuma ser denominada de ciclo **buscar-decodificar-executar**, visto na aula anterior.

### Os registradores

Os registradores são memórias muito rápidas da CPU, os quais desenvolvem determinadas funções. Eles guardam dados temporariamente e podem ser lidos e escritos em alta velocidade. Basicamente existem três tipos registradores:

1. **Contador de programa:** Sendo uns dos mais importantes presentes na CPU, indica a próxima instrução a ser buscada para execução.
2. **Registrador de instrução:** Contém a instrução a que está sendo executada no momento.
3. **Registrador de dados:** Conserva os dados de entrada, os resultados de entrada e os resultados finais.

Os computadores também têm diversos outros registradores, alguns de uso geral, outros de uso específico.

### A Unidade lógica e aritmética (ULA ou ALU)

A função da unidade lógica e aritmética é efetuar basicamente as operações de adição, subtração e outras operações simples sobre suas entradas, produzindo um resultado no registrador de saída.

#### A ULA realiza três tipos de operações (CÔRREA, 2016):

- **Operações lógicas:** tratam das entradas de dados e os resultados como um padrão binário.
- **Operações aritméticas:** realizam operações básicas de matemática (adição, subtração, divisão e multiplicação).
- **Operações de deslocamento** – compostas por dois grupos diferentes:
  - **Deslocamento lógico** - tem a função de deslocar padrões binários para esquerda ou direita.
  - **Deslocamento aritmético** – é aplicado a números inteiros e tem como objetivo dividir ou multiplicar números inteiros por dois.



## Videoaula 2

Utilize o QRcode para assistir!

Assista agora ao vídeo para fixar o seu estudo.



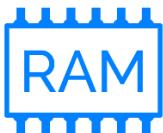
## Memória Principal

A memória é a parte do computador onde podemos armazenar dados e programas. Memórias são formadas por uma quantidade de células, onde cada uma pode armazenar uma informação (TANENBAUM, 2013). Se a memória tiver  $n$  células, então ela terá de 0 a  $n-1$  de endereços. Por exemplo, uma memória com  $2^{12}$  células de 8 bits cada e uma memória com  $2^{12}$  de 64 bits cada, ambos vão precisar de endereços de 12 bits.

No começo, a indústria de computadores padronizou células de 8 bits, que é denominada **byte**. **Bytes** são agrupados em palavras. Um computador com uma palavra de 32 bits tem 4 bytes/palavra, por exemplo. Sem uma memória da qual os processadores possam ler, gravar ou escrever informações não haveria computadores digitais.

Vamos aproveitar esse momento, e estudar um pouco dos tipos de memórias existentes.

## TIPOS DE MEMÓRIA



### Memória RAM

- A memória RAM caracteriza-se pelo fato de poder ser lida e escrita durante o processamento e é conhecida como memória principal do computador. Pode ser de dois tipos (FLOYD, 2007):
  - **Estática – SRAM (Static Random Access Memory)** – memória muito rápida e volátil, todas as RAMs estáticas são caracterizadas por células de memória *latch*. Enquanto a tensão de alimentação estiver aplicada numa célula de memória estática, ela retém indefinidamente o estado 0 ou 1. Se a alimentação for removida, o bit de dado armazenado é perdido.
  - **Dinâmica -DRAM (Dynamic random access memory)** - As RAMs dinâmicas usam capacitores como elementos de armazenamento e não podem reter os dados por muito tempo sem que o capacitor seja recarregado por um processo

denominado renovação (*refresh*). Assim como as SRAM, se a alimentação for removida, o bit de dado armazenado é perdido.



## Memória ROM

Uma memória ROM contém dados permanentes ou semipermanentes que podem ser lidos da memória, porém em alguns casos não podem ser apagados ou não podem ser modificados sem o auxílio de um equipamento especializado. Esse tipo de memória armazena dados que são utilizados de maneira repetida em aplicações como as instruções programadas para a inicialização e operação de um sistema. A principal característica das ROMs é que ela conserva os dados armazenados quando a alimentação é desligada, sendo, portanto, memórias não voláteis (FLOYD, 2007).

Vamos ver então alguns tipos de ROMs existentes:

- **PROM:** as memórias ROMs são geralmente programadas na indústria. A diferença é que as PROMs vêm do fabricante sem programação e são programadas pelo usuário de acordo com a sua necessidade.
- **EPROM:** é programável eletricamente pelo usuário, porém os dados armazenados têm que ser apagados através da exposição da memória à luz ultravioleta por um período de vários minutos.
- **EEPROM:** é uma PROM que pode ser tanto apagada, quanto programada com pulsos elétricos. A EEPROM pode ser rapidamente programada e apagada no próprio circuito (*in-circuit*) para uma reprogramação.

Atualmente notamos o uso de uma outra extraordinária memória, que tem uma alta capacidade de armazenamento, é não-volátil, tem capacidade de leitura e escrita no sistema, é compatível com operações rápidas e tem um custo baixo. Comercialmente é conhecida como memória **flash**.

As memórias flash são memórias de leitura/escrita de alta densidade (alta densidade se traduz em alta capacidade de armazenamento de bits) não-voláteis, o que significa que os dados podem ser mantidos armazenados indefinidamente sem alimentação. Algumas vezes elas são usadas no lugar de drives de disquetes ou de discos rígidos de pequenas capacidades em computadores portáteis (FLOYD, 2007).

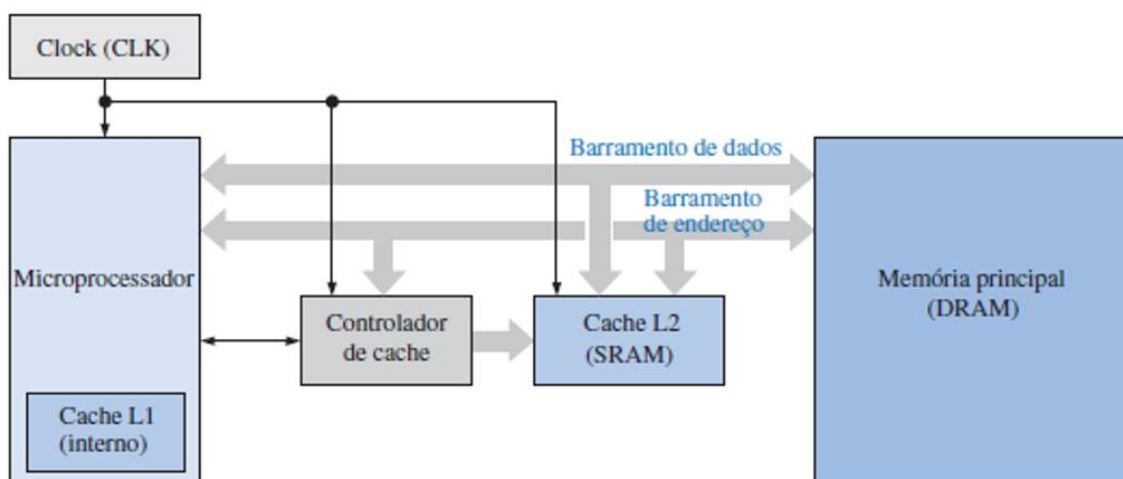


## Memória cache

Em um computador o processador é muito rápido, e de forma constante fica buscando informações na memória, o que por consequência, significa que frequentemente tenha de esperar para que a informação chegue, pois, sistemas de memória são mais lentos do que a velocidade do processador. A memória cache é uma memória pequena, temporária, porém rápida, que o processador utiliza para acessar informações que, provavelmente, serão utilizadas a curto prazo (NULL, 2010).

O conceito de memória cache é baseado na ideia de que os programas de computadores tendem a obter instruções ou dados de maneira sequencial a partir de uma área da memória principal. Basicamente, o controlador de cache “supõe” que o processador precisará das informações que estão na sequência na área da memória, sendo assim, move para a memória cache os dados de forma que esses estarão prontos quando necessário. Se o controlador de cache supor corretamente, os dados estarão imediatamente disponíveis para o microprocessador. Caso o controlador de cache faça uma dedução errada, a CPU terá que acessar a memória principal (FLOYD, 2007).

De uma maneira mais comum, os sistemas computacionais são montados com cache conhecidas como: L1 que é integrada ao chip de processamento e é conhecida cache primária. Tem também as cache (L2) que é um conjunto de chips externos ao processador e geralmente tem uma capacidade maior do que a cache L1. A L2, é conhecida como cache secundária. A figura 3 ilustra as memórias caches L1 e L2 em um sistema de computador.



**Figura 3:** Diagrama em bloco mostrando as memórias cache L1 e L2 em um sistema de computador.

Fonte FLOYD (2007, p.565).

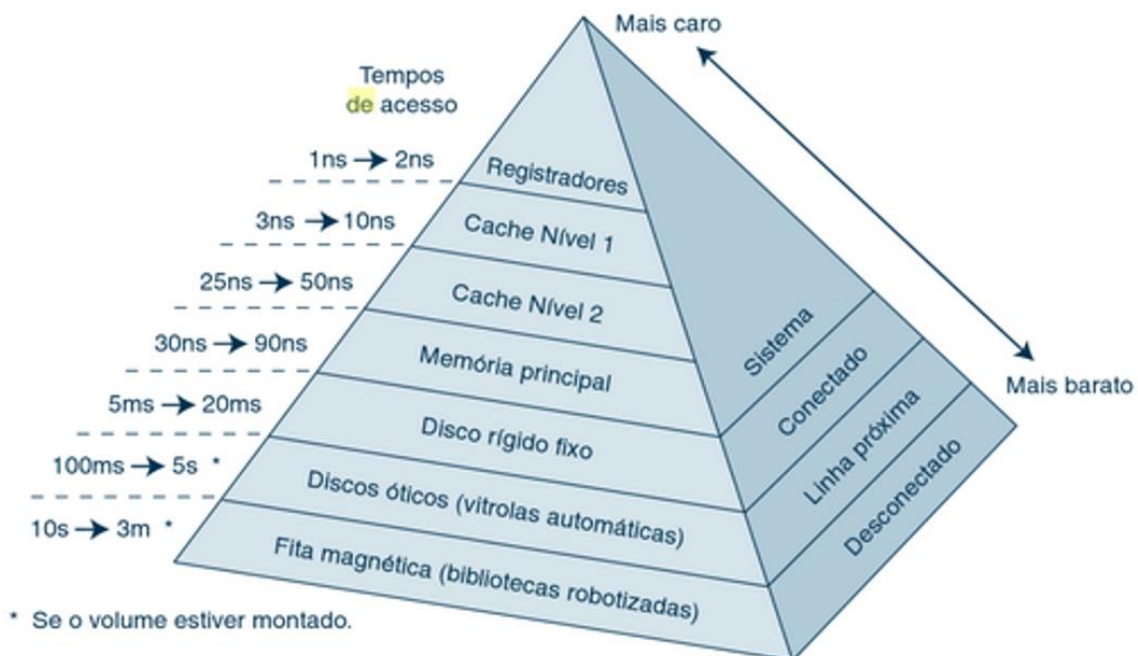
## Hierarquia de Memória

Após estudarmos um pouco sobre as memórias, precisamos agora entender um aspecto importante sobre o desempenho do processador. Notamos que nos sistemas computacionais existem vários tipos de memória: Existem algumas eficientes, porém, muito caras. E existem outras que não são tão eficientes, porém baratas. O que fazer então para que o processador tenha uma boa performance utilizando-se de memórias distintas? A resposta então é a utilização de hierarquia de memórias.

Então, para lidar com essa disparidade, os sistemas de computação atuais usam uma combinação de tipos de memória para fornecer um alto desempenho ao melhor custo. Os tipos básicos que normalmente compõe o sistema de memória hierárquica incluem: Registradores, cache, memória principal e memória secundária (NULL, 2010).

A hierarquia de memória é ilustrada na figura 4. A figura é desenhada na forma de pirâmide, pois, pode ajudar a indicar os tamanhos relativos dessas diversas memórias. Você, aluno(a), pode notar que memórias próximas ao topo são menores no tamanho, tem uma boa performance e, portanto, um custo mais elevado do que as memórias encontradas na base da pirâmide. Os dados do lado esquerdo da pirâmide indicam tempos de acessos típicos (NULL, 2010).

Quando o processador precisa de alguma informação, ele envia uma requisição para um nível mais baixo na hierarquia, no caso a memória cache (normalmente a cache, porque os registradores tendem a ser de propósito especial). Se o dado for encontrado na cache, ele pode ser carregado rapidamente para a UCP, caso contrário, a requisição é enviada a um nível mais abaixo na hierarquia que no caso é a memória principal (NULL,2010).



**Figura 4:** A Hierarquia de memória.

Fonte: NULL (2010, p.313).



**À medida que descemos na hierarquia, três parâmetros aumentam:**

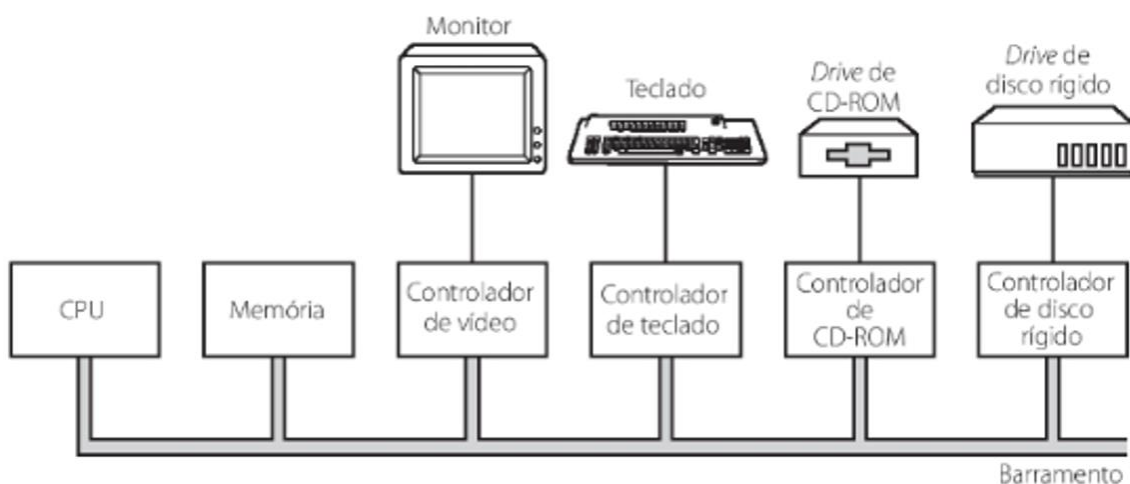
- O tempo de acesso fica maior, o acesso aos registradores fica na casa dos nanossegundos, enquanto o acesso ao disco, por exemplo, fica na casa dos milissegundos.
- A capacidade de armazenagem aumenta à medida que descemos a hierarquia, discos magnéticos tem maior capacidade de armazenamento do que registradores.
- O número de bits por dólar gasto aumenta descendo a hierarquia na pirâmide.

### **Entrada e Saída(E/S)**

Como estudado anteriormente, notamos que o computador é composto basicamente por três componentes: CPU, Memória, Entrada e Saídas. Vamos então estudar agora um pouco sobre entradas e saídas.

Um computador é composto de uma grande placa de circuito impresso, denominada placa - mãe. Na placa mãe se localiza o chip da CPU, encaixes para memórias e diversos chips de funções variadas. Tem também barramentos, os quais são usados para conectar dispositivos internos e externos ao computador.

Em Tanenbaum (2013), é mostrado um exemplo de um simples computador para nos dar entendimento de como se conecta os dispositivos que pertencem a um computador. Observe a figura 5 abaixo.



**Figura 5:** Estrutura de um computador simples.

Fonte: TANENBAUM (2013, p.86).

Observando a figura 5 acima, notamos que o simples computador é composto apenas de um barramento que conecta todos os dispositivos, porém, atualmente, ele é composto de dezenas deles.

Cada dispositivo de E/S é composto de duas partes: uma que contém a parte total da eletrônica, chamado de controlador. E a outra parte que é o dispositivo de E/S em si, tal como *drive de disco-rígido*. A função do controlador é controlar seu dispositivo de E/S e manipular seu acesso ao barramento (TANENBAUM, 2013). Quando um programa quer dados do disco, por exemplo, ele envia um comando a controlador de disco, que então emite comandos de busca e outros comandos para o drive de disco. Por isso, quando estiver fazendo manutenção de PC's é necessário instalar *driver's* corretamente.

Existem diversos tipos de dispositivos de E/S disponíveis: Teclados, monitores, mouses, microfones, *web-cams*. Lembre-se! Para todos eles são necessários controladores. Para saber mais sobre periféricos, verifique a indicação de leitura.

### Indicação de Leitura

Para complementar seus estudos até aqui, leia o capítulo 2 do livro de TANENBAUM os <<https://plataforma.bvirtual.com.br/Acervo/Publicacao/355>>. Acesso em: 20.04.2020.

### O que é RISC e CISC?

O projeto do Conjunto de Instruções inicia com a escolha de uma entre duas abordagens: a abordagem RISC e a CISC. O termo RISC é a abreviação de *Reduced Instruction Set Computer*, ou Computador de Conjunto de Instruções Reduzido e CISC vem de *Complex Instruction Set Computer*, ou Computador de Conjunto de Instruções Complexo.

De uma maneira geral, máquinas CISC se apoiam em microcódigo para lidar com a complexidade de instruções. Fica a cargo do microcódigo dizer ao processador como executar cada instrução. Apesar de serem eficientes os microcódigos, eles se limitam pelas instruções de tamanho variável, que tornam o processo de decodificação mais lento, e pelo número variável de ciclos de *clock* por instrução, que torna mais difícil implementar *pipelines* de instruções.

Ainda na plataforma CISC, todo dado armazenado na memória deve ser interpretado pelo microcódigo. Este processo de tradução adiciona mais tempo no processamento.

Arquiteturas RISC diferem um pouco da CISC. Na arquitetura RISC, a maior parte das instruções são executadas em um ciclo de *clock*. Para alcançar essa velocidade, ao invés de usar a micro programação, a arquitetura RISC utiliza-se de um controle implementado em hardware que é mais rápido na execução de instruções. Dessa forma, fica mais fácil fazer *pipelining* de instruções, porém, fica mais difícil lidar com a complexidade em nível de hardware (NULL, 2010).

Sempre surge a pergunta então: Qual é a melhor? Na literatura sobre computadores não existe uma resposta exata. A melhor resposta seria: depende do uso que se quer fazer do processador. As duas arquiteturas tem suas vantagens e desvantagens. Um computador RISC parte do pressuposto de que um conjunto simples de instruções vai resultar numa Unidade de Controle simples, barata e rápida. Já os computadores CISC visam criar arquiteturas complexas o bastante a ponto de facilitar a construção dos compiladores, assim, programas complexos são compilados em programas de máquina mais curtos. Atualmente é muito difícil categorizar os processadores RISC e CISC, pelo fato dos processadores usar uma tecnologia híbrida, ou

seja, processadores que contenha as duas arquiteturas. A tabela 1 mostra algumas características de máquinas RISC e CISC.

RISC	CISC
Vários conjuntos de registradores, muitas vezes contendo mais de 256 registradores	Um conjunto de registradores, geralmente 6 a 16 registradores no total
Permite três operandos em registradores por instrução (p. ex., add R1, R2, R3)	Permite um ou dois operandos em registradores por instrução (p. ex., add R1, R2)
Passagem de parâmetros eficiente através de janelas de registradores no chip	Passagem de parâmetros ineficiente em memória fora do chip
Instruções de um ciclo (exceto para load e store)	Instruções de vários ciclos
Controle implementado em hardware	Controle microprogramado
Muito pipeline	Pouco pipeline
Pequeno número de instruções simples	Muitas instruções complexas
Instruções de tamanho fixo	Instruções de tamanho variável
Complexidade no compilador	Complexidade no microcódigo
Apenas instruções de carga e armazenamento podem acessar a memória	Muitas instruções podem acessar a memória
Poucos modos de endereçamento	Muitos modos de endereçamento

Tabela 1 – Características de máquinas RISC versus CISC.

Fonte: NULL (2010).



**Videoaula 3**

Utilize o QRcode para assistir!

Assista agora ao vídeo para fixar o seu estudo.



**Paralelismo**

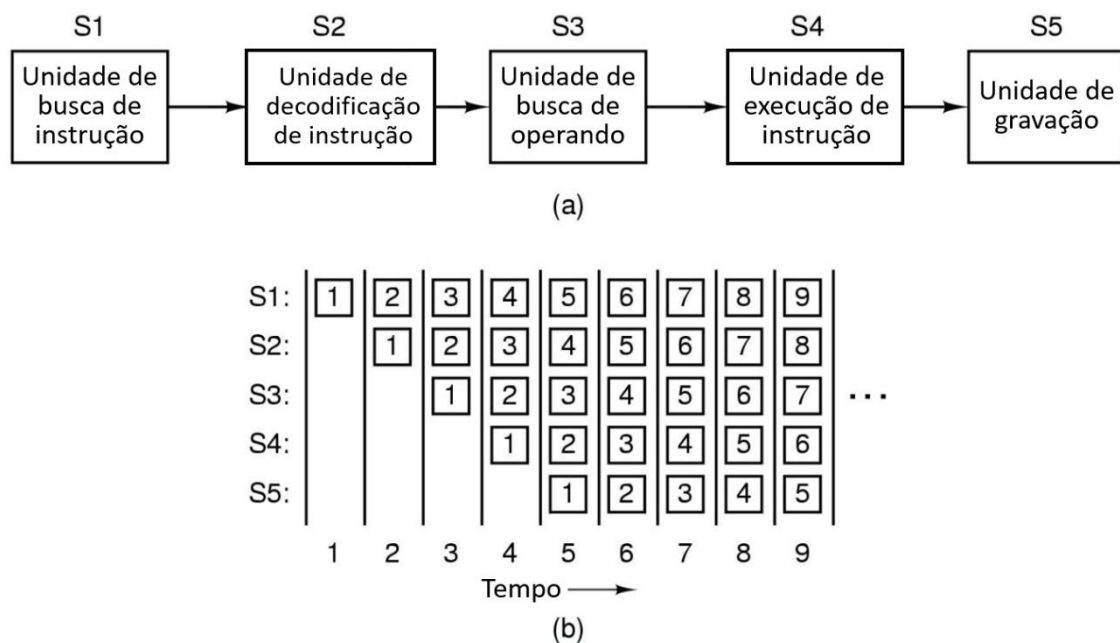
Pensando em conseguir um melhor desempenho no processamento dos dados, foi implementado nos sistemas computacionais o paralelismo, ou seja, a execução de mais de

uma instrução ao mesmo tempo. Existem diversos tipos de paralelismo: Paralelismo por Múltiplas CPUs; Paralelismo a nível de CHIP; Paralelismo a nível de Processador e entre outros.

A busca de instruções na memória era um grande gargalo na execução de uma instrução alguns tempos atrás. Assim, os computadores tinham a capacidade de buscar as instruções antecipadamente na memória e armazenar em um conjunto de registradores denominado **buffer de busca antecipada**. Dessa forma, quando necessário, a instrução poderia ser buscada de forma antecipada no **buffer** sem esperar pela conclusão de uma leitura na memória (TANENBAUM, 2013). A busca antecipada divide a execução da instrução em duas partes: **Busca e a instrução** propriamente dita.

Depois de certo tempo, foi implementado o *pipelining* que consiste em dividir a execução da instrução em várias parte, onde cada uma é manipulada por uma parte dedicada do hardware e todas elas podem executar em paralelo. Esse tipo de implementação é conhecido também como paralelismo a **nível de instrução**.

Para ilustrar a ideia de *pipelining*, vamos nos basear na ilustração abaixo onde mostra esse mecanismo. Essa ilustração é dividida em (a) e (b) e foi pesquisada em (CÔRREA, 2016, p.32). Observe a figura 6 abaixo:



**Figura 6:** ilustra um pipeline em 5 estágios.

Fonte: Côrrea (2016, p.32).

**Vejamos, temos os estágios na figura 6(a) com suas funções definidas.**

1. **S1:** Busca a instrução na memória e a coloca em um *buffer*, até que ela seja necessária.
2. **S2:** Decodifica a instrução, determina seu tipo e de quais operandos ela necessita.
3. **S3:** Localiza e busca os operandos, tanto nos registradores como na memória.

4. **S4:** Realiza o trabalho de executar a instrução.
5. **S5:** Escreve o resultado de volta no registrador adequado.

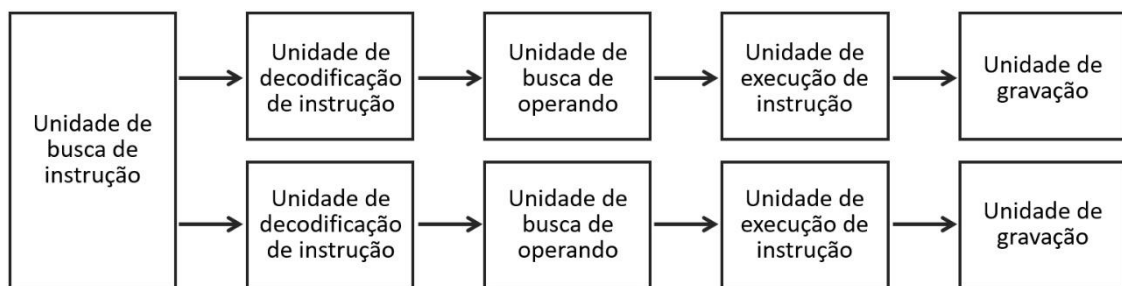
A figura 6(b) mostra como o **pipelining** trabalha em função do tempo. Observe na figura o que acontece durante cada ciclo de clock.

1. **Ciclo 1** – O estágio S1 trabalha na instrução 1(Quadrado), buscando-a na memória.
2. **Ciclo 2** - O estágio S2 Decodifica a instrução 1, enquanto S1 busca a instrução 2.
3. **Ciclo 3** – Durante esse ciclo, o estágio S3 busca os operandos para instrução 1, enquanto S2 decodifica a instrução 2 e o S1 busca a terceira instrução.
4. **Ciclo 4** – No ciclo 4, o estágio S4 executa a instrução 1. Enquanto S3 busca os operandos para a instrução 2, S2 decodifica a instrução 3 e S1 busca a instrução 4.
5. **Ciclo 5** – Nesse último ciclo, S5 grava o resultado da instrução 1 de volta no registrador, enquanto os outros estágios trabalham nas instruções seguintes.

Acredito que agora deu para entender um pouco mais sobre paralelismo! Para encerrarmos essa ideia, então, suponha que o ciclo dessa máquina seja de 2 ns. Dessa forma, uma instrução leva 10 ns para percorrer todo o caminho do *pipeline* de cinco estágios.

### Pipelines Duplos

Como vimos anteriormente, o *pipeline* único foi uma grande invenção que ajudou na performance dos computadores. Agora pense, e se tivéssemos um projeto com dois *pipelines*? Muito bom, não é mesmo? Foi isso que foi projetado. Observe figura 7 abaixo.



**Figura 7:** Pipelines duplos de cinco estágios.

Fonte: TANENBAUM (2013).

Nesse caso, em um ciclo de busca de instrução, busca-se pares de instruções ao mesmo tempo. Para que haja execução em paralelo, as duas instruções não devem ter conflitos de utilização de recursos e nenhuma deve depender do resultado da outra (TANENBAUM, 2013).

*Pipelines* simples ou duplos são utilizados na maioria das máquinas RISC, a *Intel* começou a acrescentar, a partir do computador 486, *pipelines* de dados em suas CPUs. O 486 tinha apenas um pipeline e o *Pentium* original tinha pipelines de cinco estágios. Atualmente existem diversas tecnologias visando o aumento de performance das máquinas.

### **Indicação de Leitura**

Para complementar seus estudos até aqui, leia o capítulo 2 do livro de TANENBAUM

Disponível em: <<https://plataforma.bvirtual.com.br/Acervo/Publicacao/355>> Acesso em: 20.04.2020.

## **Encerramento**

Chegamos ao final da nossa primeira unidade, onde você teve contato com os primeiros conceitos computacionais. Com certeza, muito do que foi mostrado aqui você já tinha uma pequena noção. Espero que você, aluno(a), tenha aprendido um pouco mais sobre esse incrível mundo da computação.

No mundo computacional, existem dezenas de conceitos, termos técnicos e diversas teorias. Por ser uma disciplina introdutória, foi melhor apresentar essa teoria em termos gerais para que você, caro(a) estudante, não se perca nesse emaranhado de ideias digitais. O principal objetivo da aula 1, foi de mostrar os conceitos principais que são as bases da computação, sem aprofundar em detalhes. O aprofundamento irá ocorrer na medida que você for lendo as referências utilizadas no desenvolvimento desse material. Na aula 1, discutimos dois grandes importantes conceitos: Hierarquia de níveis nos computadores e um modelo de máquina utilizado até os dias atuais.

Na aula 2, foi discutido a organização de sistemas de computador. Estudamos algumas partes do computador dando uma compreensão do papel de cada parte e de como se dá a comunicação entre elas. Nessa aula estudamos: a composição de uma unidade central de processamento, tipos de memórias e paralelismo.

Juntando as duas aulas teremos um embasamento básico sobre organização e arquitetura de computadores. Não pense, amigo(a) aluno(a), que essas teorias são apenas para computadores de uso geral. Ao contrário, são utilizadas em todos dispositivos digitais até a data atual. Até a próxima e bons estudos!

**Bom trabalho!**

## Referências

- CORRÊA, Ana Grasielle Dionísio. **Organização e Arquitetura de Computadores**. Pearson Educational do Brasil. 1 ed. 2016. Disponível em:  
< <https://plataforma.bvirtual.com.br/Acervo/Publicacao/124147>>. Acesso em: 20.04.2020.
- FLOYD, Thomas L. **Sistemas digitais: fundamentos e aplicações**. 9. ed. Porto Alegre, RS: Bookman, 2007. 888 p. ISBN 978-85-60031-93-1.
- NULL, Linda; LISBÔA, Maria Lúcia B.; LISBÔA, Carlos Arthur L. (Rev. téc.). **Princípios básicos de arquitetura e organização de computadores**. 2. ed. Porto Alegre: Bookman, 2010. 821 p. ISBN 978-85-7780-737-6 – Acervo físico da Universidade.
- STALLINGS, William; VIEIRA, Daniel; BOSNIC, Ivan; PANNAIN, Ricardo (Rev. téc.). **Arquitetura e organização de computadores**. 8. ed. São Paulo: Pearson, 2011. 624 p. ISBN 978-85-7605-564-8. Disponível em: <<https://plataforma.bvirtual.com.br/Acervo/Publicacao/151479>>. Acesso em: 20.04.2020.
- TANENBAUM, Andrew S.; AUSTIN, Todd. **Organização estruturada de computadores**. 5. ed. São Paulo: Pearson, 2013. ISBN 9788581435398. Disponível em:  
<<https://plataforma.bvirtual.com.br/Acervo/Publicacao/355>>. Acesso em: 20.04.2020.
- TOCCI, Ronald J.; WIDNER, Neal S.; MOSS, Gregory L. **Sistemas digitais: princípios e aplicações**. 10. ed. São Paulo: Pearson, 2008. 804 p. ISBN 978-85-7605-095-7.

Esperamos que este guia o tenha ajudado compreender a organização e o funcionamento de seu curso. Outras questões importantes relacionadas ao curso serão disponibilizadas pela coordenação.

Grande abraço e sucesso!

