

Para ser computador tem que ter entrada, processamento, saída (*programas também*)

### **Apresentação da informação:**

- bit
- caractere
- byte
- palavra

**BIT** → Menor unidade de medida. É binário.

**BYTE** → agrupamento ordenado de 8 bits

**CARACTERE** → São letras. Na prática, é um conjunto de BITS

**PALAVRA** → É um valor fixo (32 ou 64 bits). alguns fabricantes estabelecem o tamanho dos registradores internos a CPU igual ao da palavra

### **GRANDEZA**

todo valor é indicado em base 2

- K == 1024 ou  $2^{10}$
- M == 1024 x 1024 ( $2^{10} \times 2^{10} = 2^{20}$ )
- G == 1024 x 1024 x 1024 ( $2^{10} \times 2^{10} \times 2^{10} = 2^{30}$ )

Então:

$$512 \text{ kbytes} == 512 \times 1024 = 524.288 \text{ bytes}$$

**hardware** → Parte física do computador

**software** → Parte lógica (programas, SO, etc)

**UCP** → Unidade Central de Processamento

- Mais conhecida como CPU (Central Process Unit)
- Responsável pelo processamento e execução dos programas
- Executa instruções internas
- O computador é controlado pela CPU

**CPU é constituída de:**

- **UC - unidade de controle** → Controla os I/O, executa operações, comanda o funcionamento da ULA, informa as outras unidades o que e quando fazer
- **ULA - Unidade Lógica Aritmética** → Realiza instruções aritméticas e logicas, opera os dados de acordo com as ordens do programa. não armazena nenhum dado
- **CI - Contador de Instruções**(PC - Program Counter) → Armazena o endereço da próxima instrução a ser executada
- **RI - Registrador de Instrução** (IR - Instruction Register) → Faz o que o nome diz
- **ACC - Acumulador**(accumulator) → Armazena os dados para as operações na ULA, é um dos principais elementos que definem o tamanho da palavra do computador - o tamanho da palavra e igual ao tamanho do acumulador

**REGISTRADORES** → São as memórias mais rápidas do computador, que auxiliam a UC e ULA, faz apenas armazenamento temporário dos dados. Processadores de 32 ou 64 bits indica que o tamanho dos registradores e de 32 ou 64 bits

**LINGUAGEM DE MAQUINA** → É composta de códigos binários, que representam instruções, endereços e dados.

A CPU contém a lógica e os circuitos para fazer o computador funcionar, mas nao possui espaço para armazenar programas e dados.

**MEMORIAS** → registradores; cache, memoria principal, memoria secundaria

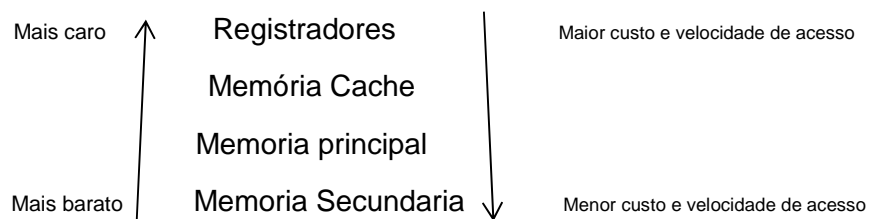
**principal** → RAM(Random Access Memory) e ROM(Read Only Memory)

**cache** → volátil, de alta velocidade e pequena capacidade de armazenamento. Melhora a performance do sistema quando executa cálculos com os mesmos valores, repetidas vezes.

- cache hit: quando o dado solicitado está na memória cache
- cache miss ou fault: quando o dado nao está na memora cache

**secundaria** → armazena grandes quantidades de informação, não volátil (o dado nao e perdido ao desligar), é mais lenta do que a principal

**memória virtual e de paginação** → É quando seu computador nao tem muita memória RAM, ai enche tudo e ele começa a usar o HD que e mais lento e o computador fica lentão.



---

## SISTEMAS DE NUMERAÇÃO

binário = base 2

decimal = base 10

octal = base 8

hexadecimal = base 16

## CONVERSAO

**Binário → octal** => separa de 3 em 3 e olha na tabela

**Binário → hexadecimal** => separa de 4 em 4

**octal/hexadecimal → binário** => é só olhar na tabela a representação de cada dígito e trocar por binário

**octal → hexadecimal** => Converte para binário depois para hexadecimal

**hexadecimal → octal** => Converte para binário e depois para octal

**decimal → binário** => vai dividindo por dois, como se não houvesse amanhã

**binário → decimal** => Inverte a posição dos dígitos e faça ele  $\times 2$  elevado a posição - 1, ou seja, sempre começando do 0:

$$1101 \rightarrow 1011 \rightarrow 1 \times 2^0 + 0 \times 2^1 + 1 \times 2^2 + 1 \times 2^3$$
$$1 + 0 + 4 + 8 = 13$$

---

## Categorias de Software:

**Software de sistema ou básico** → integrados ao hardware, se dividem em:

- SO
- Utilitários de Sistema: auxilia o SO

**Software Aplicativo** → Realiza algum trabalho para o usuário

**LP** → software utilizado para criação de software

## Aquisição de software:

**freeware** → distribuído gratuitamente, não abre o código fonte.

**Free Software** → distribuição pode ser gratuita ou paga, e é permitido seu uso, modificação e redistribuição

**Open source** → distribuído sob licença de open-source, código fonte e de domínio público ou com copyright

**shareware** → distribuído gratuitamente, mas requer pagamento depois de um período(teste)

**adware** → distribuído gratuitamente, mas requer que o usuário visualize propagandas

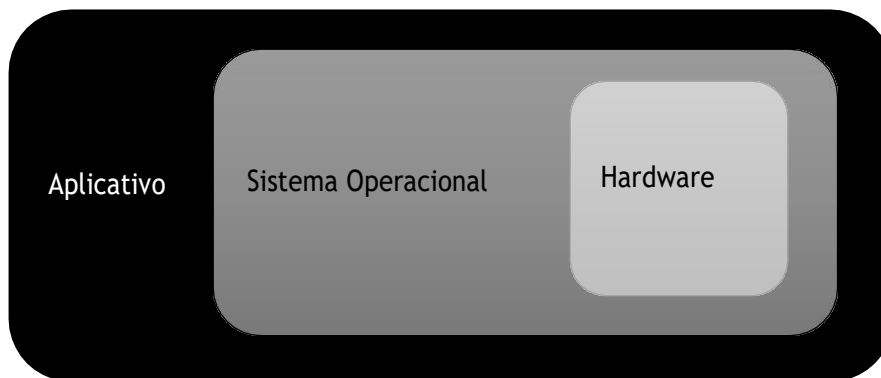
**domínio público** → sem copyright, pode ser distribuído gratuitamente

## SISTEMA OPERACIONAL

É formado por duas partes:

**Kernel** → Núcleo, realiza funções básicas

**Shell** → responsável pela comunicação com o usuário



Uma das funções do SO é fazer a carga da memória dos programas a serem executados. SO também é um programa

A carga do SO se dá através de um pequeno programa denominado bootstrap que está armazenado numa memória tipo ROM

Quando o computador é ligado o contador de programa da CPU é apontado para a primeira instrução do bootstrap e a executa, após a carga do SO (bootstrap), o contador aponta para a primeira instrução do SO e este é executado

- Fornece interface de comunicação
- Possibilita o uso eficiente e controlado dos componentes de hardware
- Da suporte aos programas
- Gerencia memória principal
- Gerencia o sistema de armazenamento secundário
- Gerencia os serviços de rede
- Estabelece a proteção do sistema

**PROCESSOS** → Pode ser definido como o estado de um programa em um determinado momento do processamento. é um programa em execução, ou seja, um programa que iniciou, mas não foi concluído.

Um processo pode estar em execução ou esperando tempo da CPU

**processo dinâmico** → É dinâmico enquanto o programa é um conjunto finito e estático de instruções.

O processo é a atividade de executar programas, é uma instância do programa. Pode-se ter vários processos de um mesmo programa

**THREADS** → É uma forma de um processo dividir a si mesmo em duas ou mais tarefas que podem ser executadas concorrentemente. São múltiplos fluxos de execução de um processo

É a realização de mais de uma tarefa ao mesmo tempo

## **SISTEMA TIME SHARING**

para utilizar com eficiência os recursos de um sistema computacional, foi criada a multiprogramação. A ideia é manter vários processos na memória, ao mesmo tempo, e somente atribuir um

- recurso a um processo que esteja precisando
- o usuário interage com o sistema
- execução de vários processos simultaneamente em time sharing
- escalonar e alocar recurso para diferentes programas, decidindo qual programa utilizará que recurso em que momento

**SISTEMAS PESSOAIS** → criação de uma SO para um único usuário

**SISTEMAS PARALELOS** → passa a ser possível realizar múltiplas tarefas no mesmo computador.

**MULTIPROCESSAMENTO SIMETRICO** → Possuem mais de um processador, compartilhando todos os recursos de HW (barramento, memória, etc) e sob o controle de um único SO. Mesmo que uma CPU falhe,

o sistema não para (win, linux, etc)

**MULTIPROCESSAMENTO ASSIMETRICO** → cada CPU tem uma tarefa específica. Possui CPU mestre, onde o SO é executado, e escravas que são subordinadas ao mestre. Falha do mestre interrompe o processamento. Quase não existe mais

## **MONOTAREFA X MULTITAREFA X MULTIUSUARIO**

**monotarefa** → uma tarefa realizada a cada vez

**multitarefa** → realiza mais de uma tarefa ao mesmo tempo

**multiusuário** → SOs que possibilitam sua utilização por vários usuários.

Não tem como ser multiusuário e monotarefa ao mesmo tempo

**SISTEMAS DISTRIBUIDOS →** Os computadores podem estar longes um do outro. o programa pode ser parte executado em um computador, e parte executado em outro

**SISTEMAS DE TEMPO REAL →** O processamento tem de ser realizado dentro dos limites de tempo, senão haverá falhas.

- Sistema de tempo real critico - garante que tarefas críticas sejam executadas a seu tempo de maneira precisa
- Sistema de tempo real nao crítico - tarefa crítica pode receber prioridade, até ser concluída, não é interrompida

## **SO PARA MOBILE**

### **TIPOS DE SO PARA PC**

- Microsoft
- Linux
- Mac
- IBM OS/2 WARP

**SO DE REDE →** são SOs para servidores, diferentes dos para desktop e executam tarefas adicionais com controle de acesso compartilhado, a recursos do servidos

- Microsoft
  - Novel
  - UNIX
  - Derivados do UNIX
  - Linux
- 

## **LINGUAGENS DE PROGRAMAÇÃO**

Conjunto de palavras e regras que servem para construir programas que instruirão o sistema de computação a realizar tarefas. cada linguagem tem seu conjunto de regras

### **COMO CRIAR UM PROGRAMA SIMPLES**

1. Entender o problema
2. Criar a logica
3. Escrever a lógica, de forma organizada em linguagem humana
4. Codificar em linguagem de programação
5. O codigo deve ser traduzido para linguagem de máquina pelo compilador/interpretador, gerando um codigo binário

**Compilador →** Transforma o codigo em um executavel. nao e necessário ler o codigo novamente se quiser executar de novo. Se tiver erro de sintaxe, sintática ou semântica, não compila.

codigo fonte → compilação → código-objeto → ligação → codigo-executavel

**Interpretador** → Não gera executável, precisa ler toda vez que executa. se tiver erro de sintaxe ou sintático, não executa. Se for erro semântico, ainda assim executa linha por linha, mas quando chega no erro, ele para

### COMO ESCOLHER UMA LP:

- Cada LP é voltada para criação de determinados tipos de aplicação, então varia de acordo com o que você quiser fazer
- Complexidade do programa a ser criado
- Tipo de sistema em que o programa será executado
- Desempenho
- Conhecimento da equipe
- Disponibilidade da LP

### São quatro gerações de LP:

**1ª** De mais baixo nível, o praticamente no nível da máquina (*assembly*)

**2ª** Avanço em relação ao assembly. comandos são através de palavras utilizadas no dia a dia (read, write, if, while, etc) (*algol*)

**3ª** Programação estruturada (tem ênfase no uso de sub-rotinas, laços de repetição, condicionais e estruturas em bloco) (*Pascal*)

**3.1** De alto nível de uso geral (nem todos concordam)

**3.2** Orientada a objeto

**3.3** Especializadas

**4ª** Mais alto nível (*python*)

**4.1** De consulta

**4.2** Geradoras de programas

**4.3** De prototipação

---

## REDE DE COMPUTADORES

### O QUE É REDE DE COMPUTADORES?

- Combinação de hardware e software, possibilitando a comunicação de dados de um local para outro;
- Uma coleção de computadores autônomos, interconectados por um meio qqr. Dois computadores são ditos em redes quando são capazes de trocar informações;
- Interconexão de computadores e equipamentos de computações que usa fios ou ondas de rádio e pode compartilhar dados e recursos computacionais.

Para que uma rede seja realmente útil e produtiva, ela deve ter: desempenho, confiabilidade e segurança

**Desempenho** → Depende do número de usuários e capacidade do hardware, meio de transmissão e eficiência do software.

É medido através do:

- Tempo de trânsito;
- Tempo de resposta;

**Confiabilidade** → Medida pela frequência de falhas e pelo tempo necessário para recuperar uma falha

**Segurança** → É necessária uma política de segurança para atingir os objetivos

- **Confidencialidade** → acesso limitado a quem for autorizado pelo proprietário da informação.
- **Integridade** → garantia de que a informação mantenha todas as características originais.
- **Disponibilidade** → garantia de que a informação estará sempre disponível para o uso

## CLASSIFICACAO DE REDES:

**PAN** → Personal Area Network

**LAN** → Local Area Network (site único)

**WLAN** → Wireless Local Area Network

**SAN** → Storage Area Network

**MAN** → Metropolitan Area Network

**WMAN** → Wireless Metropolitan Area Network

**WAN** → Wide Area Network (Vários sites interligados remotamente)

## TOPOLOGIAS DE REDES:

- **TOPOLOGIA FISICA** (Como estão dispostos e como se conectam fisicamente os nós de uma rede)
  - Barramento
  - Estrela
  - Anel
  - Arvore (mais indicada e utilizada ultimamente)
- **TOPOLOGIA LÓGICA** (Como os nós se comunicam logicamente)
  - **Redes ponto-a-ponto** → Computadores ligados um nos outros através de um sistema e topologia qualquer. Cada usuario e responsável pelos seus recursos, podendo compartilha-los. Não há controle centralizado dos recursos. Geralmente utilizado com poucos computadores.
  - **Rede cliente Servidor** → Acesso e controlado pelo servidor. Melhor organização, segurança e performance.



## PARA UMA REDE SER ÚTIL E PRODUTIVA A MESMA DEVE CONTER:

- **Desempenho** → medido através do tempo de resposta
- **Confiabilidade** → medida pela frequência de falhas
- **Segurança**
  - Confidencialidade → acesso limitado a uma ou mais pessoas
  - Integridade → manter o arquivo intacto, sem alteração nenhuma
  - Disponibilidade → garantia de que vai estar sempre disponível

**SERVIDOR** → Executa tarefas para outros computadores. Podem ser dedicados ou não

- **Dedicado** → exclusivamente para tarefas de rede
- **Não dedicado** → além de ser servidor, também é estação de trabalho

**CLIENTES** → usam e abusam do servidor

**ETHERNET** → protocolo para redes locais (LAN) baseada na troca de quadros (frames). Define cabeamento e sinais elétricos para a camada física.

---

## SEGURANÇA DA INFORMAÇÃO

Criptografia (Baseada nos números primos)

**Hash** → Resumindo, a função Hash é qualquer algoritmo que mapeie dados grandes e de tamanho variável para pequenos dados de tamanho fixo. Por esse motivo, as funções Hash são conhecidas por resumirem o dado. A principal aplicação dessas funções é a comparação de dados grandes ou secretos.

hash é um algoritmo que mapeia dados de comprimento variável para dados de comprimento fixo.

### Três pilares da segurança da informação:

**1. Confidencialidade** → Criptografia; controle de acesso; acordo de confidencialidade; treinamentos / conscientização; profissionalmente está ligado a princípios éticos

Perde a confidencialidade quando o sigilo é quebrado

**2. Integridade** → Criptografia; controle de acesso; acordo de confidencialidade; treinamentos / conscientização; profissionalmente está ligado a princípios éticos  
Perde a integridade quando a informação pode ser manuseada por uma pessoa não autorizada.

**3. disponibilidade** → Controle em segurança física; monitoramento; controle de ativos (qualquer coisa que tenha valor); planos de contingência; rotinas de backup e de restauração; uso de soluções em nuvem.  
Perde a disponibilidade quando a informação deixa de estar acessível

**cloud computing** → uso sob demanda da computação, sem preocupação com o gerenciamento físico

**SaaS - Software as a Service** → 'aluguel' do software

**IaaS - Infrastructure as a Service** → 'aluguel' de máquinas virtuais

**PaaS - Platform as a Service** → 'aluguel' da plataforma completa

**Nuvem pública** → aberta para uso público

**Privada** → exclusiva para um único usuário (uma empresa, por exemplo)

**Híbrida** → uma nuvem privada possa ter seus recursos ampliados a partir de uma reserva de recursos em uma nuvem pública

**VIRTUALIZAÇÃO DE APLICAÇÃO** → Uma camada de software instalado entre o sistema operacional e a aplicação virtualizada fica responsável pela abstração do SO, bibliotecas e drivers. Evita a necessidade de validar todas as bibliotecas necessárias para a execução do mesmo

**VIRTUALIZAÇÃO DO DESKTOP** → Executa diversos SO's simultaneamente na mesma máquina (*máquinas virtuais*)

**VIRTUALIZAÇÃO DE APRESENTAÇÃO** → Mantém o armazenamento das aplicações em servidores centralizados

**VIRTUALIZAÇÃO DE PERFIL** → Os usuários têm documentos e perfis centralizados no servidor

**ISOLAMENTO** → as aplicações e os serviços de uma máquina virtual não interferem no SO original nem em outras máquinas virtuais

**ENCAPSULAMENTO** → encapsula um sistema operacional e todos os aplicativos dentro de um pacote de softwares. pode passar uma máquina virtual pra outro computador sem problemas

**PROVISIONAMENTO** → A máquina virtual consome somente a quantidade do espaço físico em uso

- Aumenta o uso do armazenamento
- Elimina necessidade de provisionar discos virtuais em excesso
- Reduz custo de armazenamento
- Capacidade sob demanda



#### CAMADA DE VIRTUALIZACAO

