Revisão – COM140 – Introdução a Conceitos de Computação

1. Lista de tópicos

- História da computação
- Representação de dados no computador
- Operações lógicas e tabela-verdade
- Portas lógicas e circuitos
- Arquitetura de von Neumann, hardware básico e funcionalidades
- Ciclo de busca-execução
- Barramentos
- Algoritmos e linguagens de programação
- Redes
- Computação em nuvem

2. Considerações iniciais

O objetivo deste documento é fornecer uma visão geral dos tópicos centrais abordados na disciplina Introdução a Conceitos de Computação. É importante salientar que a ausência de certos tópicos nesta revisão não significa que sua relevância seja menor no contexto da disciplina. Em relação aos tópicos incluídos, são apresentados conceitos básicos sobre computadores, representação de dados, hardware, software, sistema operacional e redes de computadores. Espera-se que a leitura dessas notas não apenas ajude a compreender os conceitos discutidos, mas também a promover a compreensão de tópicos relacionados.

3. Revisão

3.1. História da computação

A computação tem como objetivo coletar, produzir e analisar dados para gerar conhecimento. Registros históricos mostram que a humanidade tem utilizado máquinas de calcular há muito tempo, como o ábaco, que foi usado antes de 3000 a.C. A partir do século XVII surgiram os primeiros equipamentos mais sofisticados para auxiliar em cálculos matemáticos, como as calculadoras mecânicas construídas por Wilhelm Schickard em 1623 e Blaise Pascal em 1642. Houve ainda projetos de máquinas de calcular, como as desenvolvidas por Charles Babbage entre 1821 e 1834. No entanto, foi a partir de 1937 que começou o desenvolvimento dos computadores eletrônicos digitais. O primeiro deles, conhecido por ABC (*Atanasoff-Berry Computer*), foi concebido em 1937 e testado em 1942. Posteriormente, em 1945, surgiu o ENIAC, o primeiro computador eletrônico de propósito geral. Outro marco importante ocorreu em 1951, com o desenvolvimento do EDVAC, o primeiro computador eletrônico que utilizava programas previamente armazenados. O EDVAC, desenvolvido por John von Neumann, possui uma arquitetura ainda utilizada atualmente, conhecida por "arquitetura de von Neumann", na qual o computador é subdividido em três partes principais: a unidade central de processamento, a unidade de memória e os dispositivos de entrada e saída de dados.

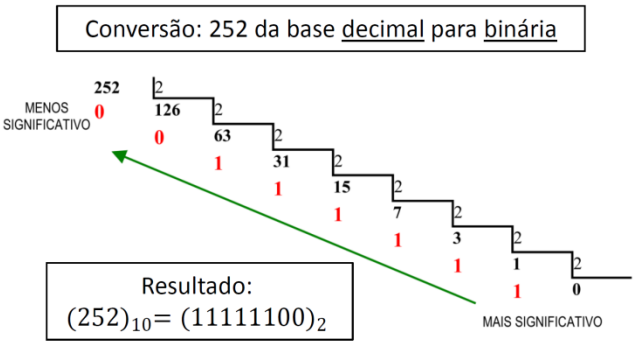
A evolução dos computadores pode ser dividida em cinco gerações: primeira geração – uso de circuitos eletrônicos valvulados, como o ENIAC, EDVAC e ABC; segunda geração – as válvulas foram substituídas por transistores, tornando os computadores menores, mais baratos e confiáveis; terceira geração – uso de circuitos integrados em pequena e média escala; quarta geração – uso de tecnologia firmware e circuitos integrados em escalas grande, muito grande e ultra grande; quinta geração – ainda teórica e envolve integração com DNA e computação quântica.

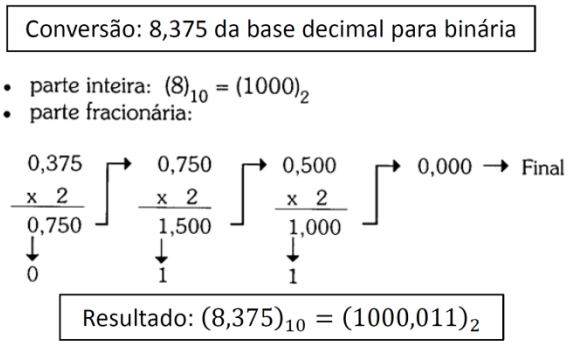
3.2. Representação de dados no computador

Os dados são essenciais para o funcionamento de um sistema computacional, representando tanto entrada quanto saída. Eles são compostos por dois estados - presença ou ausência de energia - que são convertidos para a representação binária, utilizada em todos os tipos de dados, desde números até imagens e sons. A hierarquia "dados-informação-conhecimento" destaca que os dados são apenas observações, enquanto a informação é derivada das relações entre esses dados e o conhecimento é gerado a partir da síntese dessas informações. Embora os dados observados na natureza sejam frequentemente contínuos, sua conversão para representação binária permite que eles sejam facilmente manipulados pelos computadores.

Embora os dados observados na natureza possam representar fenômenos contínuos, como valores numéricos no conjunto dos reais, eles precisam ser convertidos para a representação binária antes de serem processados pelo computador. Na representação binária, existem apenas dois símbolos/dígitos, 0 e 1, que podem ser combinados para representar qualquer número inteiro, positivo ou negativo, bem como números fracionários com quantidade finita de dígitos decimais.

A conversão de números inteiros da base decimal para a base binária é realizada através de uma sequência de divisões por 2, em que os restos das divisões são utilizados para representar o número na base binária. Já para a conversão de números com "parte fracionária", o processo é feito em duas etapas: a primeira converte a parte inteira, e a segunda converte a "parte fracionária" por meio de multiplicações sucessivas. Os diagramas abaixo ilustram os processos de conversão mencionados.





A representação de números inteiros negativos pode ser realizada segundo as notações "complemento de dois" e "excesso". Em ambas as notações, o primeiro bit (mais significativo – à esquerda) é destinado à representação do sinal. Enquanto em "excesso" o bit do sinal igual 1 indica que o número é positivo, em "complemento de dois" o número será positivo se o bit do sinal for zero.

A fim de exemplificar o funcionamento dessas notações, supondo um sistema que usa 6 bits para representar números inteiros, temos:

- Em "excesso", as sequências binárias são colocadas em ordem crescente (000000, 000001, …, 111111), em que 000000 equivale a -2^5; 100000 representa o zero; e 111111 é igual a 2^5-1.
- Em "complemento de dois", todas as possíveis sequências binárias são ordenadas de forma ascendente, começando em 000000 até 111111, de modo que: (i) 000000 é a representação decimal para o zero; (ii) o intervalo 000001 até 011111 representam os valores decimais 1 até 2^5-1; (iii) a sequência 100000, 100001, …, 111111 que representa os valores -2^5 , -2^5-1, …, -1 (ou seja, compreende os valores -32, -31, …, -1).

Além da base binária, as bases octal (i.e., base 8 – com conjunto de símbolos {0,1,…,7}) e hexadecimal (i.e., base 16 – com conjunto de símbolos {0,1,…,9,A,B,C,D,E,F}) também são utilizadas. A conversão de um número inteiro na base decimal para octal/hexadecimal segue processo similar à conversão para base binária, através de sucessivas divisões. Ainda, a representação binária pode ser convertida para as bases octal/hexadecimal através do agrupamento da representação binária

em grupos de três/quatro bits e interpretação da respectiva quantidade nas bases octal/hexadecimal. Por exemplo:
100101010 = (100)(101)(010) = 452 na base 8.
000100101010 = (0001)(0010)(1010) = 12A na base 16.

O processo de conversão de um número, expresso em uma base β qualquer para a base decimal, é feita de forma direta segundo a expressão:

$$(a_n a_{n-1} \dots a_1 a_0 a_{-1} \dots a_{-m})_\beta$$
$$= a_n \cdot \beta^n + a_{n-1} \cdot \beta^{n-1} + \cdots + a_1 \cdot \beta^1 + a_0 \cdot \beta^0 + a_{-1} \cdot \beta^{-1} + \cdots + a_{-m} \cdot \beta^{-m}$$

Em que $a_n, \dots, a_0$ representam os algarismos da arte inteira do número e $a_{-1}, \dots, a_{-m}$ são os algarismos na parte "fracionária" do número.

Nesse mesmo contexto, a representação de números com "parte fracionária" no computador é usualmente feita com emprego da notação por ponto flutuante. Segundo essa notação, para uma dada base β (geralmente binária), um número x é composto por sinal (s), mantissa (F) e expoente (E), proporcionando, assim, a representação:

$$x = sF \cdot \beta^E$$

Em que F compreende um número da forma "$0, d_0 d_1 \dots d_n$" e E é o expoente do número, definido por um número inteiro.

Além da representação de números, é comum a necessidade de representar símbolos e caracteres diversos. No entanto, surge a questão: como tratar esses caracteres uma vez que eles não são números? Nesse contexto são inseridas as codificações de texto, que podem ser entendidas como tabelas que relacionam cada caractere a uma sequência binária. Existem diversos métodos de codificação de texto, como o ASCII, EBCDIC e Unicode. O ASCII e o EBCDIC utilizam 1 byte (ou seja, uma sequência de oito zeros e uns) para representar cada caractere, o que permite expressar até 2^8=256 códigos, incluindo caracteres imprimíveis e não imprimíveis. Já o Unicode utiliza 2 bytes, possibilitando a representação de uma gama ainda maior de caracteres (até 65536), incluindo os latinos, gregos, orientais, dentre outros.

3.3. Operações lógicas e tabela-verdade

As operações lógicas são fundamentais para a realização de cálculos matemáticos no computador, já que as operações aritméticas básicas são realizadas através de operações lógicas. A Unidade Lógica Aritmética (ULA) é responsável por executar esse processo, como será discutido posteriormente. Para entender melhor como as operações aritméticas são realizadas no computador, é sugerida a leitura dos textos-base.

As operações lógicas elementares/básicas são: conjunção ($\wedge$), disjunção ($\vee$) e negação ($\neg$). As operações de conjunção e disjunção operam sobre duas entradas e resulta em uma saída. Tais entradas e saídas são valores lógicos "Verdadeiro" (V) ou "Falso" (F). Em relação à operação de negação, esta é realizada sobre um único valor lógico.

Uma ferramenta importante no entendimento das operações lógicas elementares, ou obtidas pela combinação destas, são as tabelas-verdade. Abaixo são exibidas tais tabelas para a conjunção, disjunção e negação. A e B representam entradas genéricas e as linhas da tabela expressam as possíveis combinações (exaustivamente).

A	$\neg A$
V	F
V	V

A	B	$A \wedge B$
F	F	F
F	V	F
V	F	F
V	V	V

A	B	$A \vee B$
F	F	F
F	V	V
V	F	V
V	V	V

Uma vez conhecido o comportamento da disjunção, conjunção e negação, é possível avaliar o comportamento de uma expressão qualquer, por exemplo $\neg\big(A \wedge (B \vee C)\big)$, em que A, B e C são entradas. Com uso de uma tabela-verdade é possível expressar todos os comportamentos possíveis para esta expressão. Por exemplo:

A	B	C	$B \vee C$	$A \wedge (B \vee C)$	$\neg\big(A \wedge (B \vee C)\big)$
F	F	F	F	F	V
F	F	V	V	F	V
F	V	F	V	F	V
F	V	V	V	F	V
V	F	F	F	F	V
V	F	V	V	V	F
V	V	F	V	V	F
V	V	V	V	V	F

A construção da tabela-verdade acima compreendeu em: (i) alocar as entradas A, B e C nas colunas mais à esquerda e preencher todas as possíveis combinações sem repetição; (ii) respeitando a ordem de prioridade das operações e dos parênteses, foram resolvidas as "partes" $B \vee C$, $A \wedge (B \vee C)$ e $\neg\big(A \wedge (B \vee C)\big)$, sendo esta última parte referente à coluna que contém o resultado da expressão para cada entrada A, B e C.

Na esteira dessas discussões, torna-se conveniente definir elementos da álgebra de Boole. Trata-se de uma "reinterpretação" da conjunção ($\cdot$), disjunção ($+$) e negação ($'$), além de utilizar 1 e 0 em substituição aos símbolos V e F para indicação dos valores lógicos Verdadeiro e Falso.

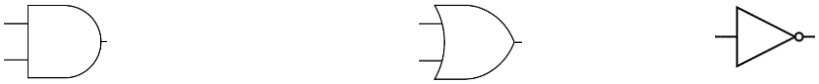
Com esta reinterpretação, as operações lógicas se aproximam ainda mais da Álgebra Elementar e do funcionamento dos circuitos eletrônicos (abordados a seguir). Algumas propriedades da álgebra de Boole que merecem destaque são:

$A + 0 = A$	$A + B = B + A$	$A \cdot (B + C) = A \cdot B + A \cdot C$ [distributiva]
$A + 1 = 1$	$A \cdot B = B \cdot A$	$(A + B)' = A' \cdot B'$ [DeMorgan "+"]
$A \cdot 0 = 0$	$(A + B) + C = A + (B + C)$	$(A \cdot B)' = A' + B'$ [DeMorgan "·"]
$A \cdot 1 = A$	$(A \cdot B) \cdot C = A \cdot (B \cdot C)$	

Com apoio das propriedades apresentadas, é possível analisar expressões lógicas e verificar possíveis equivalências mais simplificadas. Por exemplo, supondo que A e B são entradas na expressão $(A + B)' \cdot B$. A aplicação da lei de DeMorgan ao lado direto da expressão, e em seguida a propriedade comutativa, temos $A' \cdot B' \cdot B = A' \cdot (B' \cdot B) = A' \cdot 0 = 0$. Como conclusão, temos que $(A + B)' \cdot B$ é uma expressão que, independente dos valores de A e B, sempre resulta no valor 0 (Falso).

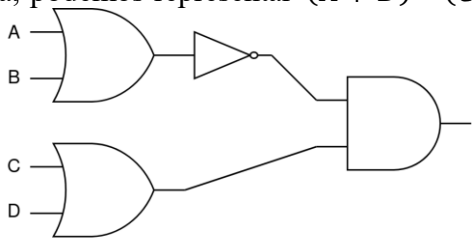
3.4. Portas lógicas e circuitos

As operações de conjunção, disjunção e negação são representadas por meio de "portas lógicas", que aproximam a álgebra de Boole dos circuitos eletrônicos. Essas operações elementares são denominadas e ilustradas graficamente da seguinte forma:



Porta AND (conjunção) Porta OR (disjunção) Porta NOT (negação)

Segundo essa abordagem gráfica, podemos representar $(A + B)' \cdot (C + D)$ por:



Uma vez que a expressão lógica é extraída/interpretada com base na representação gráfica do circuito, tabelas-verdade e as propriedades da álgebra de Boole podem ser utilizadas para analisar o comportamento do circuito. Além das portas AND, OR e NOT, existem outras portas lógicas que podem ser replicadas por meio da combinação dessas portas. Algumas delas são a XOR (disjunção exclusiva), NAND (negação da conjunção) e NOR (negação da disjunção).

3.5. Arquitetura de von Neumann, hardware básico e funcionalidades

Segundo a arquitetura de (Jhon von) Neumann, o hardware é dividido em três subsistemas: memória, a unidade central de processamento (CPU – *Central Processing Unit*), e entrada e saída (E/S).

A memória compreende a área onde são armazenados os programas e os dados durante o processo, a ULA efetua as operações lógicas de cálculo, a UC rege as operações sobre a memória, ULA e E/S. O subsistema E/S atua como meio que permite entradas de programas e dados vindos de fora do computador e gera saídas de dados para o exterior do computador.

A CPU é composta por dois módulos principais, a unidade de controle (UC) e a unidade lógica e aritmética (ULA). A UC controla o funcionamento do computador, uma vez que é responsável por determinar o fluxo como os processos são executados e os dados interpretados. Por outro lado, a ULA é responsável por efetuar as operações aritméticas e lógicas sobre os dados (já codificados como cadeias binárias).

A memória é essencial para armazenar dados no computador. Para um tratamento mais eficiente, os dados são organizados em grupos de bits, chamados de "palavras", que são acessados por meio de um "endereço de memória". É importante ressaltar que existem diferentes tipos de memória no computador, cada um com uma "hierarquia de velocidade". Esses tipos são: registradores, cache, memória principal e memória auxiliar (ou secundária). Os registradores são pequenas porções de memória de altíssima velocidade que auxiliam principalmente nas operações lógicas e aritméticas. A memória cache é responsável por armazenar dados que provavelmente serão utilizados em breve, e fica fisicamente entre a CPU e a memória principal. Isso faz com que o computador ganhe velocidade ao buscar dados mais rapidamente. A memória principal armazena os dados usados pelo computador em tempo de funcionamento, e é a maior porção de memória do computador. No entanto, seu acesso

é mais lento em comparação à memória cache. Por fim, as memórias secundárias são responsáveis por armazenar informações quando o computador não está em funcionamento, e não são voláteis. Exemplos de memórias secundárias são pendrive e CD/DVD.

Os dispositivos de entrada e saída são responsáveis pela comunicação do meio externo com o computador e vice-versa. Como "dispositivos de entrada", podemos citar o teclado, mouse, scanner, joystick etc. Exemplos de "dispositivos de saída" são o monitor e a impressora.

3.6. Ciclo de busca-execução

Em complemento às discussões anteriores, em uma "máquina de von Neumann", os dados e as instruções são armazenados na memória (logo, são endereçáveis) e tratados de forma análoga. As instruções que serem executadas são dispostas na forma de uma "lista sequencial de processos". Diante dessas condições, o funcionamento do computador efetua o seguinte ciclo (de busca-decodificação-execução): (i) na "busca", a UC ordena que seja copiada a próxima instrução no "Registrador de Instrução" (RI – que é uma parte da UC), e o endereço dessa instrução é mantida no "Registrador de Controle" (RC – outra parte da UC), e em seguida um contador é incrementado para se referir à próxima instrução na memória; (ii) na "decodificação", a instrução no RI é decodificada pela UC, que resulta em um código binário para alguma operação no sistema computacional; (iii) na etapa de "execução", após a decodificação, a UC envia a ordem de execução à alguma componente da CPU (e.g., "carregar/ler um item da memória", "somar o conteúdo de dois registradores e colocar o resultado em um registrador de saída" etc.).

Após essa sequência de passos, o processo retorna para o primeiro passo a fim de executar a próxima instrução contida na lista de processos. Cabe observar que ao retornar ao primeiro passo, o endereço da instrução a ser executada já está armazenada no contador de programa.

3.7. Barramentos

Os três subsistemas discutidos anteriormente (CPU, memória e E/S) estão conectados diante da necessidade de troca de informação. Em especial, a CPU e a memória estão conectados por três grupos de conexão, denominados barramentos de dados, de endereço e de controle.

- O barramento de dados é composto por diversos "fios de conexão", sendo responsável por transportar um bit por vez. O número de "fios" equivalente ao tamanho da palavra usada pelo computador (e.g., 32 bits, 64 bits etc.).

- O barramento de endereço permite acessar determinada palavra da memória, logo, o número de "fios" depende do espaço de memória. Supondo que a memória total do computador é fragmentada em 2^n palavras, seriam necessários *n* fios para acessar cada palavra (ex.: 2048 bits de memória $\rightarrow$ 64$\times$palavras de 32 bits = 2^6 palavras, permitindo gerar as combinações 000000, 000001, …, 111111). Essa relação entre as combinações (sequências binárias) e o endereço das palavras é também conhecido como mapeamento do espaço de memória.

- O barramento de controle permite a comunicação entre a CPU e a memória (e.g., a CPU envia um código para a memória ordenando uma leitura/escrita de dados). A quantidade de "fios" depende do número de comando existentes. Se existem 2^m comandos, são necessários *m* fios para gerar 2^m combinações diferentes.

3.8. Software e Sistemas Operacionais

Software é um conjunto de algoritmos e estruturas de dados desenvolvidos para executar uma determinada tarefa em um ambiente computacional. Esses algoritmos são implementados usando linguagens de programação. É comum se referir a um software como um "programa".

Os softwares podem ser categorizados de acordo com a tarefa para a qual foram desenvolvidos. Existem softwares básicos ou de sistema, utilitários, aplicativos, plug-ins e embarcados. Os softwares básicos auxiliam a execução de outros softwares ou interagem com dispositivos de hardware. Os softwares utilitários garantem o bom funcionamento do sistema (e.g., antivírus, firewall e monitor de dispositivos/rede). Os softwares de aplicação são desenvolvidos para atender a uma demanda específica (e.g., edição de texto, manipulação de imagens, desenvolvimento de programas e comunicação). Os plug-ins são extensões de softwares maiores que fornecem funções adicionais. Por fim, um software embarcado é usado para funções específicas em um dispositivo projetado para realizar uma tarefa particular.

Dentre os softwares básicos está o sistema operacional, responsável pelo funcionamento e gerenciamento de todo o ambiente computacional, incluindo outros softwares e dispositivos de hardware. O sistema operacional oferece interface com o usuário, além de controlar o uso da CPU, memória, dados (arquivos) e dispositivos de entrada/saída.

Os sistemas operacionais evoluíram ao longo dos anos. Nas décadas de 1970 e início de 1980, era comum o uso de sistemas operacionais em lote, em que os programas eram executados de acordo com uma fila organizada por uma pessoa que gerenciava o computador. Nesses sistemas, era comum que os programas fossem armazenados em fichas perfuradas. Com a evolução, surgiram os sistemas operacionais de tempo compartilhado, em que vários programas de diferentes usuários podem ser executados simultaneamente. Os sistemas operacionais de tempo real têm como característica responder instantaneamente às solicitações dos usuários. Além disso, existem os sistemas híbridos, em que vários usuários realizam o processamento em um sistema computacional maior, acessando-o via um computador terminal.

3.8. Algoritmos e linguagens de programação

Um algoritmo é uma sequência de passos específicos que deve ser executada para resolver um problema. Sua característica fundamental é que ele deve produzir uma resposta para uma entrada ou interação específica. É importante ressaltar que existem diferentes maneiras de construir um algoritmo, e essas diferentes abordagens podem ser comparadas e avaliadas por sua corretude e eficiência.

A corretude de um algoritmo é a capacidade de produzir a saída correta e esperada em relação às entradas apresentadas. Por outro lado, a eficiência refere-se à capacidade do algoritmo de utilizar recursos computacionais e tempo de forma adequada. Quanto mais eficiente for um algoritmo, menos recursos e tempo serão necessários para sua execução.

Fora do contexto computacional, é possível expressar um algoritmo de diferentes maneiras, como por meio da fala, escrita ou desenhos. No entanto, quando o objetivo é construir um software, é necessário que a sequência de passos ou instruções seja compreendida pelo computador. Para isso, são utilizadas linguagens de programação, que são compostas por um conjunto de símbolos que seguem uma sintaxe específica e permitem a definição de instruções não ambíguas, que podem ser entendidas pelo computador.

As linguagens de programação e os computadores evoluíram ao longo dos anos. As linguagens de máquina (ou primeira geração) exigiam a manipulação de códigos binários para a definição de instruções, tornando o processo muito complexo. As linguagens de montagem (segunda geração)

foram desenvolvidas para simplificar a programação, mas ainda exigiam uma alta interação do programador com o hardware. As linguagens de alto nível (terceira geração) eliminaram a necessidade de interação do programador com o hardware e forneceram um ambiente mais natural para a construção de algoritmos.

Existem também as linguagens de altíssimo nível (quarta geração), que utilizam sintaxe ainda mais próxima da linguagem natural. Além disso, as linguagens de alto nível são organizadas de acordo com diferentes paradigmas de programação. Existem quatro paradigmas principais: imperativo (ou estruturado), funcional, lógico e orientado a objetos. Cada um desses paradigmas fornece um modelo conceitual para a implementação de algoritmos. Algumas linguagens de programação podem atender a mais de um paradigma, sendo caracterizadas como "multiparadigma" (ou "de múltiplos paradigmas").

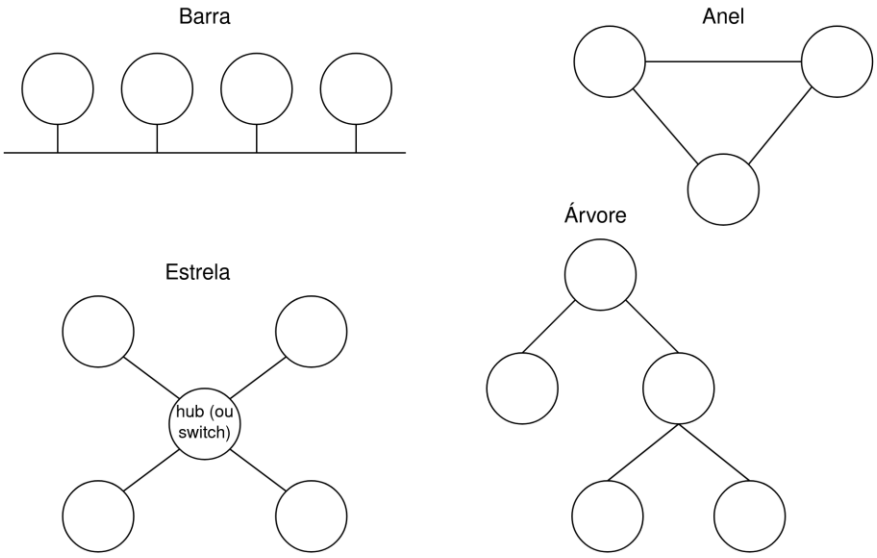
No paradigma imperativo, que foi o primeiro a surgir e ainda é o mais utilizado, os algoritmos são construídos por meio de uma sequência de comandos elementares que manipulam dados e operam sobre eles. No paradigma funcional, todos os processos são efetuados por meio de funções que recebem dados de entrada e retornam valores ou ações como saída. O paradigma lógico é baseado em conceitos da lógica clássica, que permite modelar a lógica que rege um dado problema diante de fatos e axiomas. Já o paradigma orientado a objetos permite organizar o problema em termos de objetos, que possuem características, funções próprias e interagem entre si.

3.9. Redes

Uma rede de computadores é definida quando um conjunto de computadores interligados entre si a fim de permitir o compartilhamento de recursos e informações. O exemplo mais evidente de rede de computadores é a Internet, a qual revolucionou a forma como trabalhamos e interagimos em sociedade. As redes de computadores podem ser caracterizadas segundo diferentes características, a citar: o meio de transmissão; sua topologia (forma); e extensão (tamanho/abrangência); e protocolos de comunicação.

Em relação ao meio de transmissão, a conexão entre computadores pode ser concretizada através de cabos ou transmissão nas frequências de rádio ou mesmo infravermelho. Cabos de pares trançados (similar aos utilizados em linhas telefônicas), coaxiais (similar aos utilizados em TV a cabo) ou mesmo cabos de fibra óptica (que utiliza luz para representar a transmissão de bits) são diferentes tipos de cabos que podem ser empregados para interligar computadores. A taxa de transmissão de informação é diretamente influenciada pelo tipo de cabo utilizado, o uso dos cabos de fibra óptica e os pares de cabos trançados garantem a maior e menor velocidade de transmissão, respectivamente. Como já mencionado, é possível utilizar as frequências de rádio ou outros comprimentos de onda como meio físico para estabelecer conexão entre computadores sem o uso de cabos/fios, proporcionando, assim, uma conexão wireless (tradução para "sem fio").

A respeito da topologia, esta defina o formato como os computadores estão conectados entre si. O formato da rede influencia diretamente como as informações são transitadas pela rede a fim de comunicar dois computadores ou mesmo qual será o efeito sobre toda a rede quando uma (ou mais) ligação é interrompida. As principais topologias são: barra; estrela; anel; e árvore.



Conforme ilustrado, a topologia em barra é caracterizada por uma mesma conexão, que, por sua vez, pode causar tempo de espera por alguns computadores da rede quando outro computador faz uso da conexão. A topologia estrela utiliza hub ou switch como ponto de distribuição de conexão para os demais computadores da rede, compreendendo uma configuração muito comum nas redes wireless. A topologia anel dispensa o uso de hub/switch e é capaz de contornar um possível rompimento de conexão entre dois pontos, já que cada computador nessa configuração possui duas conexões. A topologia em árvore também dispensa o uso de hub/switch, porém, em situações em que ocorre interrupção de conexão, são formadas sub-redes isoladas. Além desses exemplos de topologias, é possível ainda combinar tais exemplos a fim definir novas topologias.

A distância física entre os computadores que compõem a rede define a sua extensão, as quais podem ser classificadas em LAN (*Local Area Networks*), MAN (*Metropolitan Area Networks*) e WAN (*Wide Area Networks*). Uma LAN é característica de áreas pequenas, como laboratórios, escritórios ou residências, comumente estabelecidas por conexões wireless ou Ethernet (i.e., utilizando cabos que se conectam a um hub/switch) e com altas taxas de transmissão. Uma MAN é caracterizada pela situação em que os computadores da rede estão localizados em posições distintas de uma cidade ou região metropolitana, separados por quilômetros de distância. É comum que a taxa de transmissão em uma MAN seja inferior quando comparada a uma LAN. Por fim, uma WAN geralmente conecta redes LANs e MANs separadas por grandes distâncias, bem como inclui trechos de acesso público. A Internet é uma WAN que cobre todo o planeta!

Os protocolos de comunicação em uma rede definem com os computadores realizam a troca de informação entre si. Os protocolos incluem a forma de codificação dos sinais, o formato das mensagens, a identificação e endereço dos nós interligados, e regras para troca de informação. O envio contínuo ou particionado em partes/pacotes são duas formas básicas de transmitir dados pela rede. No caso contínuo, os dados são enviados em um fluxo sequencial. Por outro lado, quando os dados são enviados em pacotes, o computador receptor é responsável por remontar a informação ao fim com base no conjunto de pacotes recebidos.

Os protocolos também estabelecem a capacidade e responsabilidade de cada computador na rede. Nesse sentido, existem as arquiteturas peer-to-peer e cliente-servidor. Na arquitetura peer-to-peer cada computador possui capacidades e responsabilidades iguais. Ao contrário, a arquitetura cliente-servidor estabelece que alguns computadores são responsáveis por servir outros. Exemplos de finalidade de "computadores servidores" são: armazenar dados e disponibilizar aos usuários da rede (servidor de arquivos); gerenciar impressões (servidor de impressão); executar programas e serviços

solicitados (servidor de aplicação); hospedar recursos que podem ser acessados pela internet (servidor web).

3.10. Computação em nuvem

A computação em nuvem é uma tecnologia relativamente recente que tem revolucionado a forma como a computação é feita. Em essência, essa tecnologia permite que programas sejam executados e dados sejam armazenados em servidores remotos, eliminando a necessidade de hardware local. Isso significa que recursos computacionais podem ser acessados por meio de uma plataforma de serviços na nuvem, que é acessada pela internet. No entanto, é importante destacar que é necessária uma conexão confiável à internet para acessar esses recursos.

Em relação ao software, não há distinção significativa entre programas que são executados localmente ou na nuvem, exceto pelo fato de que as "versões na nuvem" geralmente possuem recursos que se beneficiam da natureza da computação em nuvem, como o acesso e o compartilhamento de dados na nuvem. No que diz respeito ao hardware, é necessário o uso de servidores escaláveis capazes de lidar com o processamento e a distribuição de dados. Esses servidores possuem semelhanças com computadores convencionais, mas são equipados com tecnologia que permite o processamento e acesso remoto em grande escala.

4. Considerações finais

Este texto versou sobre uma breve revisão da disciplina Introdução a Conceitos de Computação. De modo geral, foi possível notar uma segmentação entre tópicos gerais sobre computadores, que incluem conceitos elementares para o entendimento da organização e funcionamento de um computador, seguido por discussões sobre hardware, software e redes de computadores.

Espera-se que a leitura deste texto ajude a aprimorar e solidificar seu conhecimento sobre conceitos fundamentais da computação. Como sugestão final, é indicado que leia novamente os textos-base propostos durante a disciplina, focando nos itens em que ainda não domina. Isso garantirá uma compreensão mais ampla dos conceitos apresentados na disciplina e habilitará você no enfrentamento de desafios mais complexos na área da computação.