# Algoritmo

- **Algoritmo** = *é uma seqüência de instruções para resolver um poblema.*
- **Propriedades:**
  - **Garantia de término:** o algoritmo resolve o problema em uma quantidade finita de tempo.
  - **Exatidão:** cada instrução deve ter um significado claro.
  - **Efetividade:** cada instrução deve ser básica o suficiente para ser executada por qualquer agente usando lápis e papel.

## Exemplo.

Algoritmo para deteminar Máximo Divisor Comum (MDC) de dois números reais.

1. Chame o maior número de $a$ e o menor de $b$.
2. Divida $a$ por $b$ e chame o resto de $r$.
3. Se $r$ é igual a zero então o MDC é igual a $b$ e a execução das instruções termina aqui. Caso contrário, siga para a próximo instrução.
4. Atribua o valor $b$ a $a$ e o valor $r$ a $b$.
5. Volte para a instrução 2.

**Problema algorítmico:** *é um problema para o qual existe uma solução na forma de um algoritmo.*

Problemas algorítmicos geralmente possuem muitas **instâncias**.

Um algoritmo é dito **correto** quando ele sempre termina e produz a resposta correta para *todas* as instâncias de um dado problema.

# Processamento de dados

A execução de um algoritmo é composto de 3 partes:

- **entrada:** é um conjunto de informações que é requisitada para que as instruções do algoritmo possam ser executadas.
- **processo:** é a seqüência de instruções que compõe o algoritmo.
- **saída:** é o resultado obtido com a execução do processo para a entrada fornecida.

# Construção de Algoritmos

Construir um algoritmo para resolver um problema significa encontrar uma solução para o problema e descrevê-la como uma seqüência finita de ações.

A *capacidade* de resolver problemas pode ser vista como uma *habilidade* a ser adquirida e pode ser obtida pela combinação de duas partes:

- Conhecimento: adquirido pelo estudo $\Longrightarrow$ quais táticas, estratégias e planos usar e quando usar;
- Destreza: adquirida pela prática $\Longrightarrow$ agilidade na resolução de problemas.

Passos a serem seguidos:

- **Análise preliminar:** entender o problema com a maior precisão possível, identificando os dados e os resultados desejados.
- **Solução:** desenvolver um algoritmo para o problema.
- **Teste de qualidade:** executar o algoritmo desenvolvido com uma entrada para a qual o resultado seja conhecido.
- **Alteração:** se o resultado do teste de qualidade não for satisfatório, altere o algoritmo e submeta-o a um novo teste de qualidade.
- **Produto final:** algoritmo concluído e testado, pronto para ser aplicado.

# Resolução de Problemas e Abstração

*Maria tinha cinco maçãs e João tinha três. Quantas maçãs eles tinham juntos?*

O *processo de abstração* pode ser visto como constando de **níveis** $\Longrightarrow$ grau de simplificação de uma abstração.

Algoritmos bem projetados são organizados em níveis de abstração.

Instruções estão organizadas de uma forma em que é possível entender o algoritmo sem, contudo, ter de entender os detalhes de todas as instruções de uma só vez.

Construção do algoritmo utilizam ferramentas, como os *módulos*, que agrupam instruções que realizam uma determinada tarefa.

# Computadores

Computador $\Longrightarrow$ é uma máquina que manipula informação (**dados**) na forma de números e caracteres.

- *"Como informaremos ao computador o algoritmo que deve ser executado para obtermos o resultado desejado?"*
- *"Como informaremos a entrada do algoritmo?"*
- *"Como receberemos o resultado do algoritmo?"*

**Programação:** ato de instruir o computador para que ele resolva um determinado problema.

**Linguagem de programação:** linguagem apropriada para descrever algoritmos computacionais.

**Programa:** é o algoritmo escrito em uma linguagem de programação.

**Linguagem de máquina:** é a linguagem de programação própria do computador.

**Compilador:** traduz automaticamente um programa em uma linguagem de programação para a linguagem de máquina.

Tanto os programas quanto os seus dados são inseridos no computador através dos **periféricos de entrada**.

As instruções e os dados inseridos são armazenados na **memória**.

Os dados resultantes do processamento são devolvidos nos **periféricos de saída**.

Um programa é executado no computador através de um dispositivo interno denominado **Unidade Central de Processamento (UCP ou CPU)**.

**Hardware:** é a máquina propriamente dita e todos os periféricos ligados a ela.

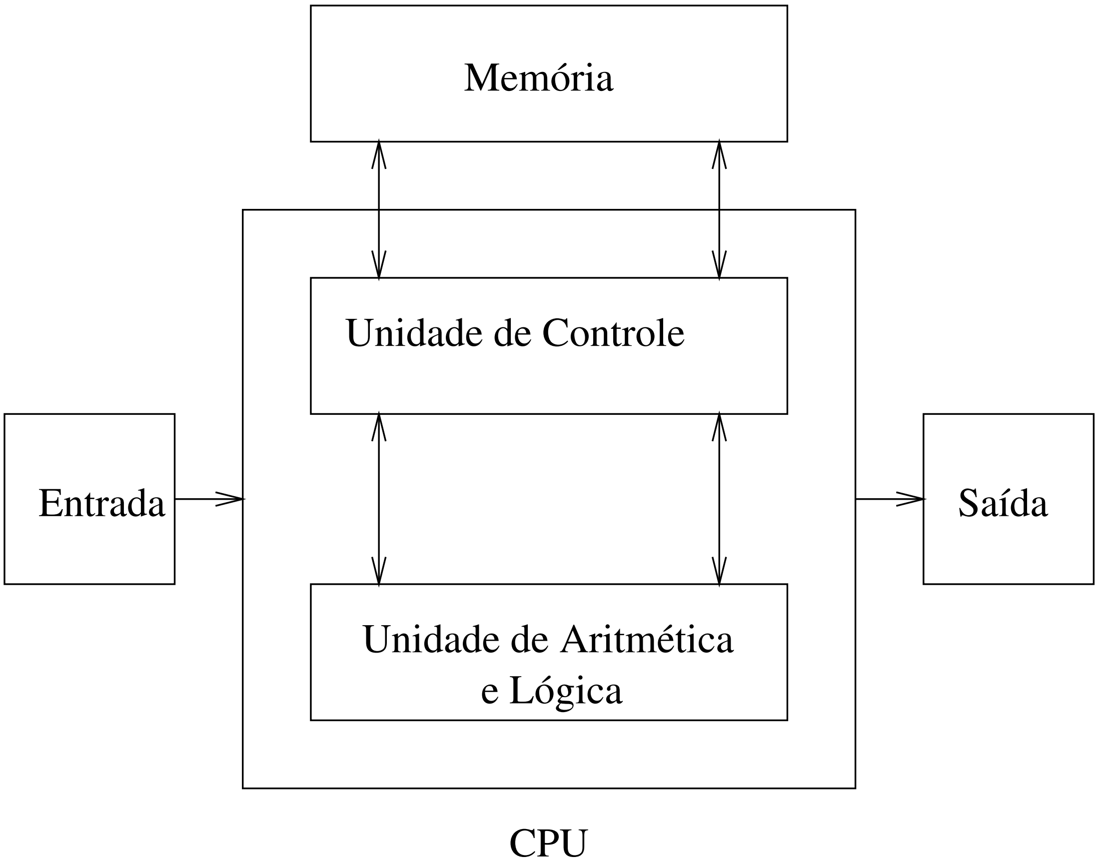
**Software:** se refere aos programas que fazem a máquina realizar alguma tarefa.

Para podermos criar *softwares* devemos adquirir capacidades para:

- Desenvolver algoritmos para solucionar problemas envolvendo transfomação de informação;
- Usar uma linguagem de progamação.

**A coisa mais importante que você pode fazer como estudante de Computação é desenvolver sua habilidade para resolver problemas.**

# Hardware




- Toda transmissão de dados, manipulação e recuperação é realizada através de pulsos elétricos e magnéticos representando seqüências de dígitos binários (**bits**).

- **bit:** é a menor unidade de informação (*binary digit*).

- Estas seqüências de 0's e 1's são organizadas em 1 ou mais bytes.

- **byte:** é um grupo de 8 bits.
- As instruções e dados são armazenados na memória do computador. Existem dois tipos de memória:
  - Memória primária
  - Memória secundária
- **Memória primária (principal ou temporária):** armazena as instruções e os dados de um programa em execução. É composta por uma seqüência de posições de tamanho fixo, cada qual possuindo um identificador distinto denominado **endereço**.
- Cada posição da memória armazena uma instrução ou parte dela ou um dado.
- A CPU comunica-se constantemente com a memória para obter a próxima instrução a ser executada ou um dado.
- A localização de instruções e dados é feita pelos seus endereços na memória.

- Tamanho da memória é dado em *bytes*.
  - 1 *quilobyte* (1 kbyte ou 1 k) = 1024 bytes
  - 1 *megabyte* (1 Mbyte ou 1 M) = 1024 kbytes
  - 1 *gigabyte* (1 Gbyte ou 1 G) = 1024 Mbytes
  - 1 *terabyte* (1 Tbyte ou 1 T) = 1024 Gbytes
- A memória principal é volátil, perdendo seu conteúdo quando o computador é desligado.
- Sua principal característica é a rapidez em que as informações nela contidas são obtidas.

- **Memória secundária:** armazena programas e dados que computador pode executar e utilizar em um dado instante.

- Esta memória é permanente não perdendo suas informações quando o computador é desligado.

- Quando um programa contido em uma memória secundária precisa ser executado, ele primeiramente é transferido para a memória principal, juntamente com os dados necessários à sua execução, e daí sua execução é iniciada.

- Exemplos de memória secundária: disco rígido, disco flexível, CD-ROM.

- Quanto à permissão de acesso, as memórias podem ser classificadas em:

  - Memória de Acesso Aleatório (RAM – *Random Access Memory*). Permite a leitura e a escrita em qualquer posição de memória.

  - Memória Somente para Leitura (ROM – *Read Only Memory*). Permite apenas a leitura dos dados.

- Unidade Central de Processamento (UCP ou CPU) $\Longrightarrow$ responsável por buscar as instruções e os dados do programa que estão armazenados na memória, decodificar as instruções e executar a tarefa descrita por elas com os respectivos dados.
  - Unidade de Controle (UC): é onde as seqüências de códigos binários, que representam as instruções a serem executadas, são identificadas e através da qual os dados são obtidos da memória.
  - Unidade de Aritmética e Lógica (UAL): é onde as instruções são efetivamente executadas.
  - Registradores: são os elementos de memória da CPU. Armazenam os dados a serem imediatamente processados.
- Atualmente, toda a CPU é construída em uma única pastilha (*chip*) de silício que é denominada **microprocessador**.
- Estes microprocessadores são classificados pelo tamanho (comprimento em bits) da **palavra** que é a unidade de informação que eles conseguem processar de uma só vez.

**Unidades de entrada**

– servem para introduzir programas ou dados no computador.

– são denominados periféricos de entrada.

– Exemplos: teclado, mouse, unidade de disco, *scanner*, leitora ótica.

**Unidades de saída**

– servem para receber programas ou dados do computador.

– são denominados periféricos de saída.

– Exemplos: vídeo, impressora e unidade de disco.

# Linguagens de Programação

– Linguagem de máquina

$\Longrightarrow$ instruções descritas como seqüências de bytes.

01000011 00111010 00111011 01000001 00101011 01000100

– Linguagem montadora ou *assembly*

$\Longrightarrow$ instruções compostas por letras, denominadas mnemônicos.

ADD CL,BH

$\Longrightarrow$ utiliza-se um compilador denominado **montador** ou **assembler**, que traduz o código escrito em *assembly* para linguagem de máquina.

– Linguagens de alto nível

$\Longrightarrow$ instruções em língua inglesa.

$\Longrightarrow$ utilização de compiladores mais complexos.

# Software

- *Softwares* básicos: são programas que administram o funcionamento do computador e nos auxiliam a usá-lo.
- *Softwares* aplicativos: são programas que executam com o auxílo dos *softwares* básicos e realizam tarefas tipicamente resolvidas pelo computador.

Principais *softwares* básicos:

- **Sistema operacional:** conjunto de programas que gerencia o computador e serve como *interface* entre os programas do usuário e a máquina. Pode ser:
  - Monotarefa: permite a execução de apenas um programa de cada vez. Ex.: DOS.
  - Multitarefa: permite mais de um programa em execução simultaneamente. Ex.: UNIX, Windows NT.

- **Utilitários:** programas de uso genérico que funcionam em conjunto com o sistema operacional e que têm como objetivo executar funções comuns em um computador. Ex.: formatadores de disco, programas para transferência de dados, etc.

- **Compiladores:** programas que traduzem um programa em uma linguagem de programação específica para seu equivalente em uma linguagem de máquina específica.

- **Interpretadores:** programas que interpretam o significado e executam cada instrução de um programa.

- **Depuradores:** programas que auxiliam o programador a encontrar erros em seus programas.

# A Computação como Disciplina

**Computação:** o estudo de processos sistemáticos que descrevem e transformam informação: suas teorias, análise, projeto, eficiência, implementação e a aplicação.

algoritmos = "processos sistemáticos que descrevem e transformam informação"

*"O que pode ou não ser automatizado?"*

Principais áreas:

- **Arquitetura:** estuda as várias formas de fabricação e organização de máquinas.
- **Linguagens de Programação:** estuda os métodos para projeto e tradução de linguagens de programação.
- **Teoria da Computação:** são feitas e respondidas perguntas tais como: "Uma determinada tarefa pode ser realizada por computador?", "Qual é o número mínimo de operações necessárias para qualquer algoritmo que execute determinada tarefa?"

- **Análise de Algoritmos:** estudo da medida do tempo e do espaço que os algoritmos necessitam para realizar determinadas tarefas.

- **Projetos de Algoritmos:** estuda os métodos para desenvolver algoritmos de forma rápida, eficiente e confiável.

O estudo da Computação também compreende o estudo de métodos para representar e armazenar dados a serem utilizados pelos algoritmos durante suas execuções.

Assim, Computação é o estudo de algoritmos e suas estruturas de dados.

Neste curso estudaremos métodos para construir algoritmos e também algumas estruturas de dados simples para representar, no computador, os dados utilizados pelos algoritmos que construiremos.