



Programação I

</ Conceito de algoritmo />

- Definição;
- Programa;
- Estruturação;
- Operadores.



2021

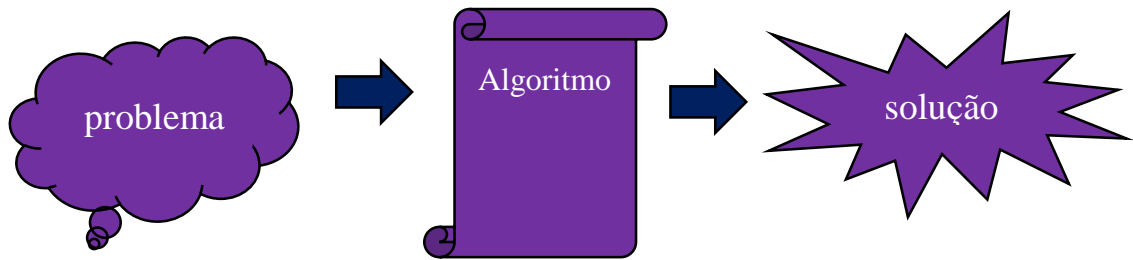
TekMind

Algoritmo

→ Definição

Algoritmo é um **conjunto finito** de regras que fornece uma **sequência** ordenada, e não ambígua de operações para **resolver um problema** específico passo a passo.

Para especificar uma sequência de passos, é necessário utilizar a ordem, portanto, precisamos utilizar a **lógica**.



Quando construímos um algoritmo, fixamos um **padrão de comportamento** para ser seguido passo a passo, que pode ser executado por outra pessoa ou máquina.

→ Por que precisamos dos algoritmos?

O algoritmo é a base inicial de toda programação de computadores. Pois:

- Especifica uma sequência de passos lógicos para o computador executar uma tarefa.
- Com uma ferramenta algorítmica, podemos compreender uma solução para um dado problema.

→ Características

- Existe fim;
- Não cria margem à dupla interpretação (não ambíguo);
- Receber dados de entrada do mundo exterior;
- Gera informações de saída para o mundo externo ao algoritmo;
- Afetivo (todas as etapas especificadas no algoritmo devem ser alcançáveis em um tempo finito).

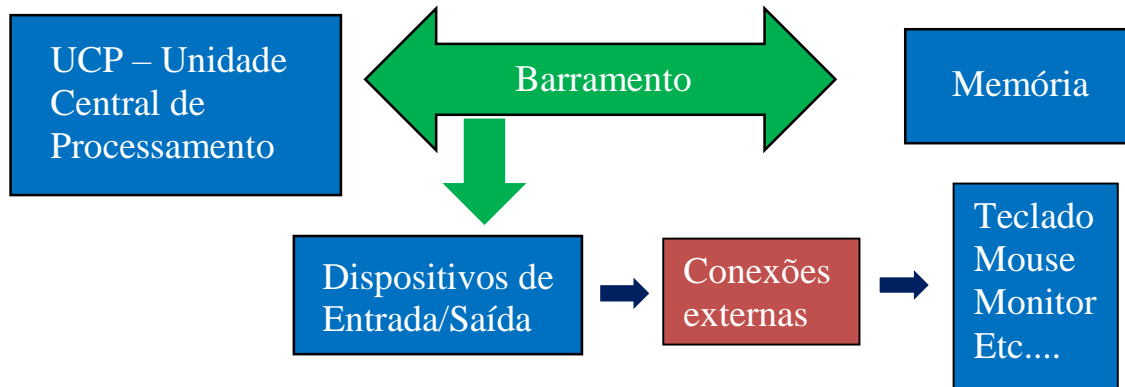


Programa de Computador

Programa é um tipo de algoritmo com operações específicas para o computador e exclusivos ao conjunto de instruções que o processador pode executar.

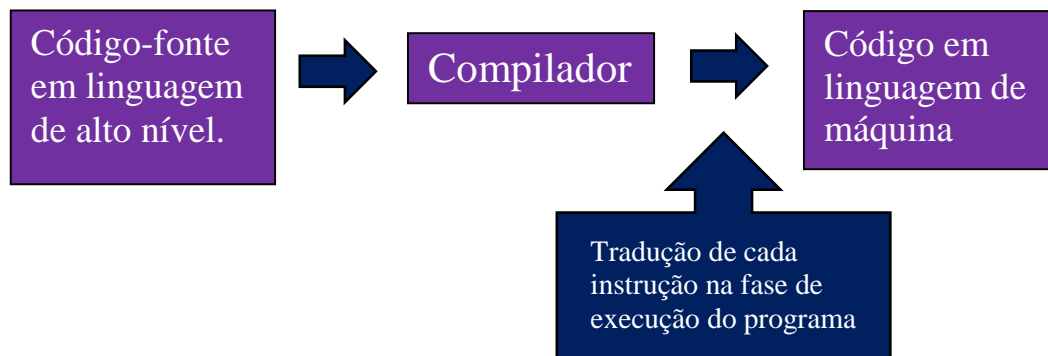
Classificamos as linguagens de programação segundo sua **proximidade** com a linguagem da máquina. Quanto maior a semelhança, mais baixo é o nível da linguagem. As mais "distantes" são conhecidas como **linguagens de alto nível**.

Os computadores das mais variadas arquiteturas têm funcionamento similar:



→ Executando um Programa

A partir do código em linguagem de alto nível, chamado como **código fonte**, o **compilador** gera um arquivo com o código em linguagem de máquina, conhecido como **código-objeto**. Esse código-objeto fica em disco e só é carregado em memória no momento da execução.



Estruturação de Algoritmos

→ Formas de Representação

– Linguagem natural

Faz-se o uso do português para descrever o algoritmo e é dirigida a orientar uma pessoa e não uma máquina, as quais não podem tomar decisões na base de escolher alternativas, mesmo que nos pareçam óbvias.

Vantagem:

- O português é bastante conhecido por nós.





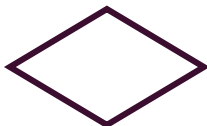




Desvantagens:

- Imprecisão;
- Pouca confiabilidade.

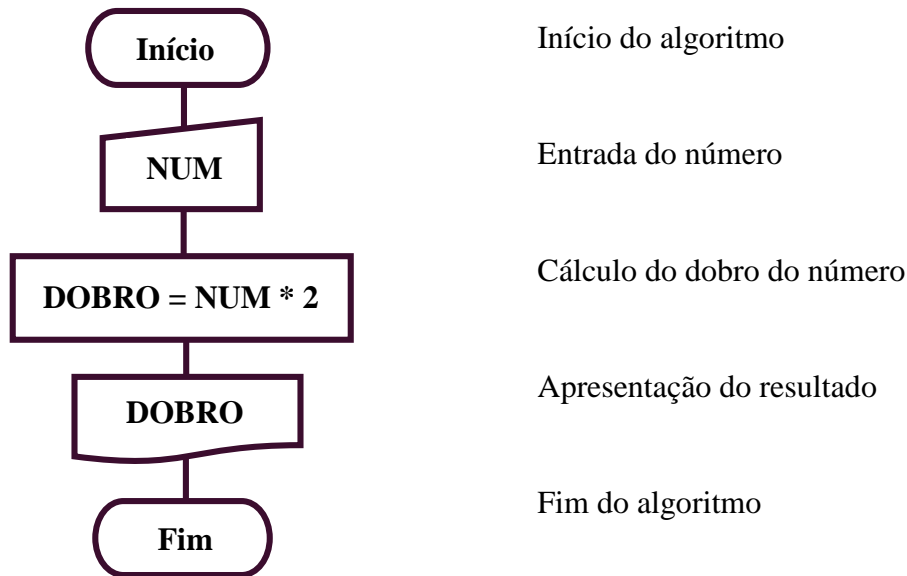
– Fluxograma ou Diagrama de blocos

Conjunto de **símbolos gráficos**, cada um representando uma **ação específica** e básica para ser executada por um computador. Ele mostra a linha de raciocínio usada pelo programador para resolver o problema proposto.

Na tabela a seguir, apresenta todos os símbolos e suas respectivas funções:

Símbolo	Significado	Descrição
	Terminal	Início e fim do fluxo lógico ou definição de sub-rotinas.
	Entrada Manual	Define entrada manual de dados por meio de um teclado conectado ao computador.
	Processamento	Executa ações de processamento relacionados a operações matemáticas.
	Exibição	Executa operações de saída de dados em um monitor de vídeo conectado ao computador.
	Decisão	Define desvios condicionais nas operações de tomadas de decisões e laços adicionais para repetição de trechos do programa.
	Preparação	Executa um laço incondicional que permite a modificação de instruções ou grupo de instruções limitadas no laço.
	Processo predefinido	Define um grupo de operações relacionadas a uma sub-rotina de processamento.
	Conector	Pontos de conexão entre trechos de programas apontados a outras partes do diagrama de blocos.
	Linha	Vínculos existentes entre os símbolos de um diagrama de blocos. Deve ter na ponta uma seta , indicando sua direção .

Exemplo:



Entretanto, a **representação gráfica** não é prática. A correção de uma linha de pensamento pode implicar a reconstrução de muitas instruções. Além disso, a construção de algoritmos mais complexos e longos pode se tornar extremamente trabalhoso, ocupando várias páginas.

Vantagens:

- Uma das ferramentas mais conhecidas;
- Padrão mundial.

Desvantagem:

- Complica à medida que o algoritmo cresce.

– Pseudocódigo

Consiste na definição de uma pseudolinguagem de programação, cujos comandos são em português, para representar algoritmos. Sua estrutura é inicializada deste jeito:

```
Algoritmo Nome_Do_Algoritmo
    variáveis
        Declaração das variáveis
    Início
        Corpo do Algoritmo
    Fim
```

Vantagens:

- Usa o português como base.
- Pode-se definir quais e como os dados vão estar estruturados.
- Passagem quase imediata do algoritmo para uma linguagem de programação qualquer.

Desvantagem:

- Não é padronizada.

Fundamentos

→ Manipulação de dados

Um **conjunto de dados** é um conjunto de objetos que tem em comum o mesmo comportamento diante de um conjunto definido de operações, podendo trabalhar com diferentes tipos de dados em um programa.

→ Identificadores, Constantes e Variáveis

– Identificadores

Nomes escolhidos para rotular as **variáveis, procedimentos e funções**, normalmente obedecendo as seguintes regras:

- ✓ Os primeiros caracteres devem ser uma **letra**;
- ✓ Ser formados por caracteres pertencentes ao seguinte conjunto {a,b,c...a,A,B,C...A,0,1,2...9};
- ✓ Devem explicar seu **conteúdo**.

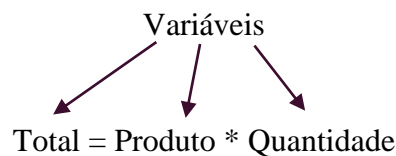
– Constantes

Representam valores que não variam no decorrer do algoritmo.

Exemplo: PI, “Falso”, “Opções”.

– Variáveis

Unidade básica de **armazenamento** das informações a nível de linguagem de programação.



Os tipos de dados e variáveis dependem da **finalidade** do algoritmo, sendo estes:

- **Inteiro:** qualquer número inteiro, negativo, nulo ou positivo.
Exemplo: 100,-4,0,98.....
- **Real:** qualquer número real, negativo, nulo ou positivo.
Exemplo: 1,3 816,97 3,0 -0,0055.....
- **Caracter:** qualquer conjunto de caracteres alfanuméricos (qualquer tipo numérico ou letra do alfabeto).
Exemplo: CASA, livro, 18, R\$ 55,36.....
- **Lógico:** armazena apenas os valores V (Verdadeiro) e F (Falso).

→ Declaração de Variáveis

Criação (ou definição) de locais na memória rotulados com o identificador da variável utilizada no algoritmo para **manipulação** de uma determinada informação.

Exemplo: NUM, X : inteiro; PI : constante;

Observação: Neste momento, as variáveis ainda não têm nenhum tipo de valor;

→ Operações básicas

É necessário que as variáveis tenham seus valores consultados ou alterados e, para isso, devemos definir um **conjunto de operadores**, sendo eles:

– Operador de Atribuição

Atribui um **valor** a uma variável ou constante. Sintaxe geral da linguagem:

IDENTIFICADOR → EXPRESSÃO

Exemplo: NOME ← ‘Fulano de tal’

Media ← (P1+P2) /2

Tipos de operadores matemáticos:

+ = Adição	Div = resultado inteiro de uma divisão.
* = Multiplicação	% (resto) = resto da divisão inteira de dois números inteiros.
- = Subtração	pow (base, expoente) = a^b exponenciação.
/ = Divisão	sqrt (valor numérico) = é a raiz quadrada de um número.

– Operadores Relacionais

Utilizados para relacionar variáveis ou expressões, resultando num **valor lógico** (verdadeiro ou falso), sendo eles:

== - igual	!= - diferente
< - menor	> - maior
<= - menor ou igual	>= - maior ou igual

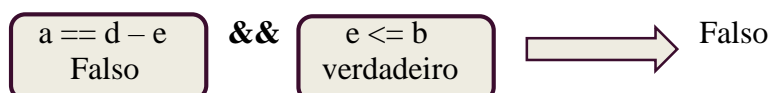
– Operadores Lógicos

Utilizados para avaliar expressões lógicas, sendo eles:

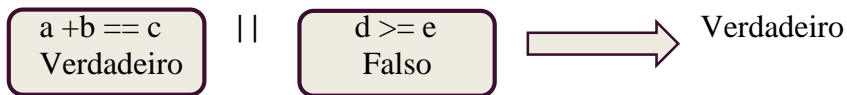
- **e (&&)** - lógico ou conjunção.
- **ou (|)** - ou lógico ou disjunção.
- **não (!)** - negação.

Exemplo: a = 5, b = 4, c = 9, d = -2, e = 3;

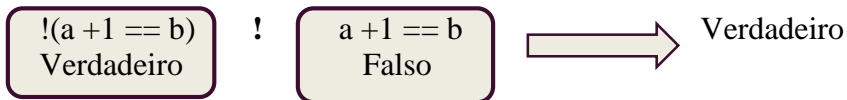
Uma expressão (**E**) é verdadeira se **todas as condições** forem verdadeiras.



Uma expressão (**OU**) é verdadeira se **pelo menos uma** condição for verdadeira.



Uma expressão (**NÃO**) **inverte o valor** da expressão ou condição, se verdadeira inverte para falsa e vice-versa.



→ **Prioridade de Operadores**

Durante a execução de uma expressão que envolve vários operadores, é necessário a existência de prioridades, caso contrário poderemos obter valores que não representam o resultado esperado.

- 1 - Efetuar operações embutidas em parênteses "mais internos";
- 2 - Efetuar Funções;
- 3 - Efetuar multiplicação e/ou divisão;
- 4 - Efetuar adição e/ou subtração;
- 5 - Operadores Relacionais;
- 6 - Operadores Lógicos.