



INSTITUTO FEDERAL
Catarinense

Professor :

Diego Ricardo Krohl

diego.krohl@ifc-videira.edu.br

Algoritmos

Introdução a Informática

- O objetivo da informática é auxiliar o homem em trabalhos rotineiros, repetitivos ou que necessitem de grandes volumes de dados para gerar informação;
- Segundo o dicionário (Aurélio, 2004):
 - Informática: "Ciência que visa ao tratamento da informação através do uso de equipamentos da área de processamento de dados."
 - Dado: "Elemento ou quantidade conhecida, que serve de base a resolução de um problema."
 - Informação: "Tudo aquilo que, por ter alguma característica distinta, pode ser ou é apreendido, assimilado ou armazenado pela percepção e pela mente humana."

Definição de algoritmos

- Uma sequência bem definida de passos que devem ser realizados em uma determinada ordem para atingir um objetivo desejado;
- Algoritmo 1 - Fazer um sanduíche segundo [ASCENCIO e CAMPOS 2002];
 - Passo 1 - Pegar o pão.
 - Passo 2 - Cortar o pão ao meio.
 - Passo 3 - Pegar a maionese.
 - Passo 4 - Passar a maionese no pão.
 - Passo 5 - Pegar e cortar alface e tomate.
 - Passo 6 - Colocar alface e tomate no pão.
 - Passo 7 - Pegar o hambúrguer.
 - Passo 8 - Fritar o hambúrguer.
 - Passo 9 - Colocar o hambúrguer no pão.

Definição de algoritmos

- Questões importantes a respeito de um algoritmo:
- Formalização e falta de detalhes;
- Para quem o algoritmo foi escrito?
- Não existe uma única solução.
 - Exemplo para a preparação de um miojo.

- Passo 1 – Colocar água na chaleira.
- Passo 2 – Ferver a água.
- Passo 3 – Pegar uma panela média.
- Passo 4 – Colocar o miojo dentro da panela.
- Passo 5 – Colocar água fervente sobre o miojo.
- Passo 6 – Colocar em fogo médio.
- Passo 7 – Aguardar 3 minutos.
- Passo 8 – Desligar o fogão.
- Passo 9 – Misturar o pacote de tempero.
- Passo 10 – Comer.

Programas de computadores

- Programas de computadores são algoritmos escritos em uma linguagem de computador, como Pascal ou C, e são interpretados e executados pelo computador;
- Os algoritmos interpretados por computador são específicos e não podem executar instruções que não foram definidas dentro do seu escopo;

Formas de representação de algoritmos

- **Descrição narrativa:** os algoritmos são expressos em linguagem natural;
- **Fluxograma convencional:** uma representação gráfica que emprega formas geométricas padronizadas para indicar as ações e decisões que devem ser executadas para resolver o problema.

Formas de representação de algoritmos

- **Pseudolinguagem:** emprega uma linguagem intermediária entre a linguagem natural e uma linguagem de programação para descrever os algoritmos.

Descrição narrativa

- Exemplo do Miojo!

Fluxograma Convencional

- A representação de um fluxograma convencional utiliza diversas figuras geométricas que descrevem cada uma das ações do código.

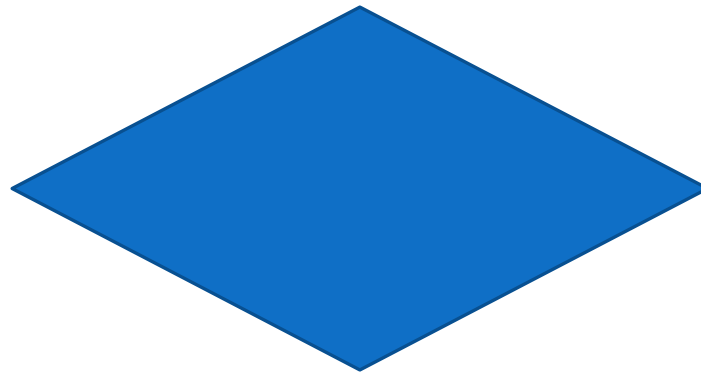
Fluxograma Convencional

- O Retângulo representa “Processamento”;
- Exemplo: colocar água para ferver.



Fluxograma Convencional

- O Losango representa “Decisão”, quando uma ou mais ações podem ser tomadas;
- Exemplo: colocar água para ferver.



Fluxograma Convencional

- O quadrado com um canto chanfrado representa “Entrada de dados”;
- Exemplo: usuário informa um dado.



Fluxograma Convencional

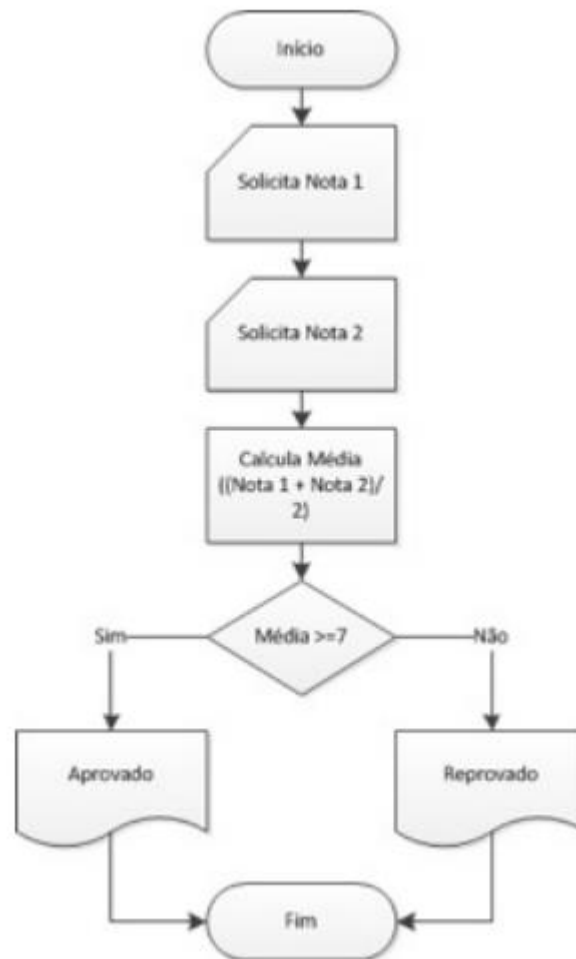
- O quadrado com cantos arredondados representa início ou fim de um algoritmo.



Fluxograma Convencional

- Vamos fazer o fluxograma da média de um aluno com duas notas:

Fluxograma Convencional



Fluxograma Convencional

- A representação gráfica é uma das ferramentas mais utilizadas para representação de algoritmos, porém, sua utilização não traz muita atenção aos dados, ficando difícil descrevê-los e o algoritmo torna-se complicado conforme o número de instruções aumenta.

Pseudolinguagem

- A pseudolinguagem procura utilizar uma linguagem que esteja o mais próxima possível de uma linguagem de programação de computadores, mas evitando definir regras de construção gramatical;

Pseudolinguagem

- Utiliza-se a linguagem natural, restringindo o escopo da linguagem para algumas palavras;
- Cria-se uma versão reduzida de um trecho de código que seria escrito em uma linguagem C, Pascal, Visualg ou outras.

Exemplo de código em Pseudolinguagem

```
algoritmo "CALCULA_DOBRO"  
var  
    varNum, varDobro : inteiro  
inicio  
    leia(varNum)  
    varDobro  $\leftarrow$  2*varNum  
    Escreva (varDobro)  
fimalgoritmo
```

Vantagens da Pseudolinguagem

- Utilizando o português para definir o código fica mais fácil para escrever, ao invés de utilizar comandos próprios de uma linguagem;
- A passagem do algoritmo escrito em uma pseudolinguagem para uma linguagem de programação de computador é mais rápida do que em outros formatos.

Teste de mesa

- Após desenvolver um algoritmo é preciso testá-lo!
- Esse teste é chamado de TESTE DE MESA, que significa seguir as instruções do algoritmo de maneira precisa para verificar se o procedimento utilizado está correto ou não;

Teste de mesa

- Exemplo:

varNum	varDobro
2	4
6	12

- Nesse exemplo deve-se atribuir valores a “varNum”, executar o cálculo descrito no algoritmo e avaliar se “varDobro” vai resultar como na tabela anterior.

Ambiente de programação - Computador

- Os computadores podem ser usados de forma eficiente na solução de certos tipos de problemas.
- Os problemas que suportam tratamento por computador em geral envolvem grandes quantidades de dados ou são problemas de natureza complexa, exigindo a execução de um grande número de passos para alcançar a solução.
- Basicamente, são problemas na área de processamento de dados e na área científica.

Processamento de Dados

- Define-se por processamento de dados qualquer atividade que, utilizando informações (ou dados), efetua transformações para obter novas informações (ou dados) como resultado.



Solução de um problema

- Entendimento do problema;
- Criação de uma sequência de operações (ou ações) que, quando executadas, produzem a solução;
- Execução dessa sequência de operações;
- Verificação da adequação da solução.

Programação

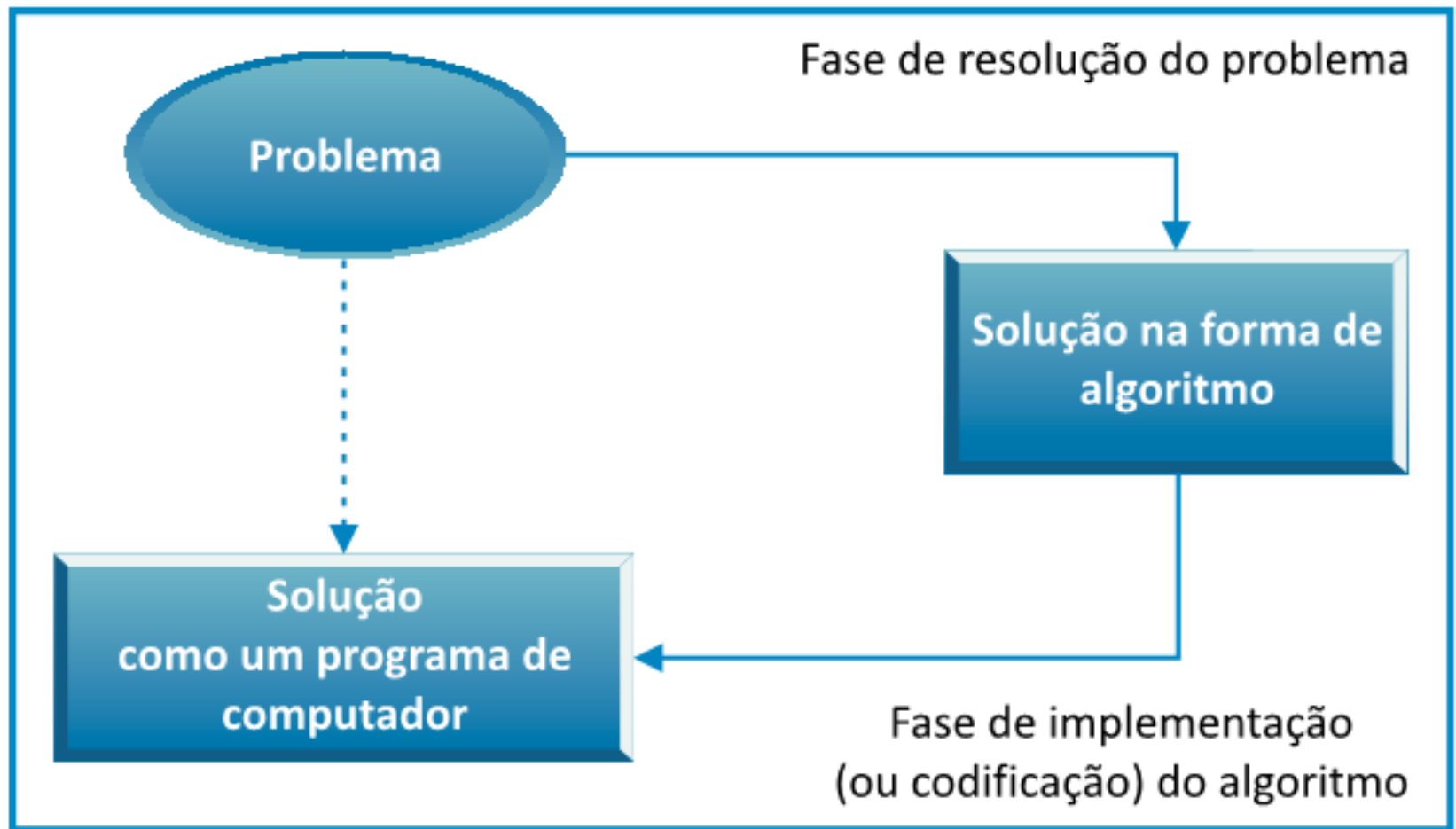
- É a sequência de planejamento, projeto, escrita e testes de instruções desempenhados pelo computador.
- É uma arte e uma ciência.
 - Arte porque existem muitas maneiras de se realizar o trabalho de programação. Existe espaço para uma considerável dose de criatividade.
 - É também uma ciência, porque existem algumas regras que devem ser seguidas, porque é necessário o uso de lógica e porque existem alguns métodos rigorosos de programação que asseguram a eficiência, economia e a utilidade dos programas gerados.

Programação

- Para tornar o trabalho de programação mais fácil, divida sistematicamente em partes menos complexas (essa técnica denomina-se “dividir para conquistar”).
- Um programa é considerado confiável quando consegue fazer com que o computador cumpra com o objetivo proposto.

Programação

- Os programas construídos devem ser eficazes, realizando a tarefa definida de forma eficiente, utilizando os melhores meios para realizá-la.
- Ao se construir um programa, o objetivo principal é vencer a complexidade do problema a ser solucionado.



Fases da programação – Etapas

- Modelização (ou resolução) do problema: determinação do modelo de solução para o problema proposto na forma de um algoritmo computacional. (Independente da linguagem de programação)
- Implementação: é a transformação (ou codificação) do algoritmo em alguma linguagem de programação adequada ao modelo elaborado.

Linguagens de programação

- Linguagem de programação é um conjunto finito de palavras, comandos e instruções, escritos com o objetivo de orientar a realização de uma tarefa pelo computador.

Linguagens de programação

- O computador trabalha somente com códigos numéricos (linguagem de máquina), baseados nos números 0 e 1 (sistema binário), que representam impulsos elétricos, ausente e presente.

Linguagens de programação

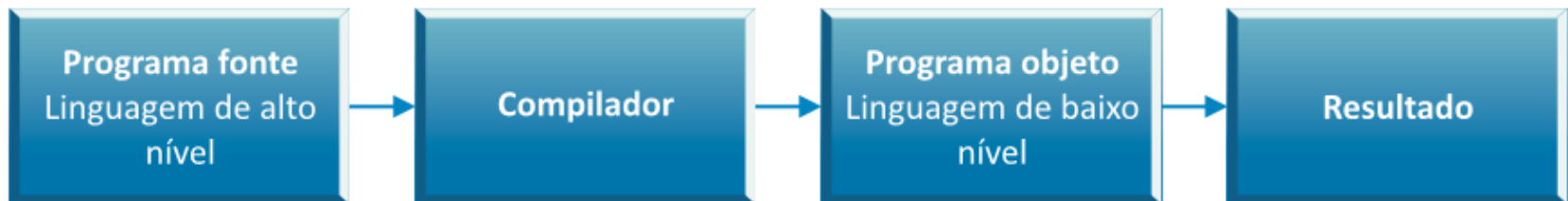
- Qualquer linguagem de programação deve estar situada entre dois extremos: o da linguagem natural do homem (muito clara, porém lenta) e o da linguagem de máquina (muito rápida, porém complexa).
- **Nível de linguagem:** alto nível para as mais próximas da linguagem humana; baixo nível para as mais semelhantes à linguagem de máquina.

Tradutores

- Para que um computador possa “entender” um programa escrito em uma linguagem de alto nível, torna-se necessário um meio de tradução entre a linguagem utilizada no programa e a linguagem de máquina. Esse meio pode ser de dois tipos: **compilador** e **interpretador**.

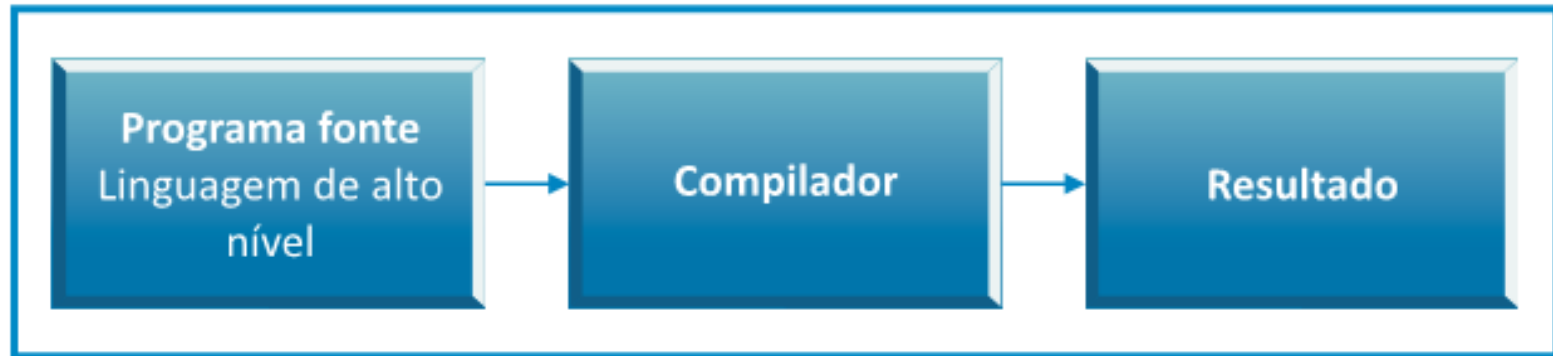
Compilador

- Traduz o programa escrito em linguagem de alto nível (programa-fonte) para um programa equivalente escrito em linguagem de máquina (programa-objeto).



Interpretador

- Traduz e envia para execução instrução por instrução e o programa permanece na forma de fonte.



Tipos de dados

- Um algoritmo é constituído de uma sequência de instruções a serem seguidas para a obtenção de soluções a algum problema;
- Nessa solução, são manipulados diversos dados;
- Esses dados podem ser os apresentados pelo problema – os dados intermediários – que são gerados durante a execução do algoritmo, ou os dados que representam a solução do problema.

Tipos de dados

- Todo dado a ser tratado em um algoritmo deve pertencer a algum tipo, que irá determinar o domínio de seu conteúdo;
- Os tipos mais comuns de dados são conhecidos como tipos primitivos de dados;
- São eles: inteiro, real, caractere e lógico.

Tipos de dados

- Inteiro: Toda e qualquer informação numérica que pertença ao conjunto dos números inteiros (negativo, nulo ou positivo);
- Exemplos: 39; 0; -56 entre outros;

Tipos de dados

- Real: Toda e qualquer informação numérica que pertença ao conjunto dos números reais (negativo, nulo ou positivo, inteiro ou fracionário);
- Exemplos: -4, 3, 0, 35; 1,23;

Tipos de dados

- Caractere: São as sequências contendo letras, números e símbolos especiais. Uma sequência de caracteres deve ser indicada entre aspas (“ ”). Esse tipo de dado é também conhecido como alfanumérico, string, literal ou texto.
- Exemplos:
 - “Rua Alfa, 52 Apto 1”;
 - “Fone: 3276-9988”;
 - “04387-456”; “ “;
 - “7”.

Tipos de dados

- Lógico: São caracterizados como tipos lógicos os dados com valores **verdadeiros** e **falsos**, sendo que podem representar **apenas** um desses dois valores.
- Ele é chamado por alguns de tipo booleano, devido à contribuição do filósofo e matemático inglês George Boole na área da lógica matemática.

Tipos de dados

Tipo	Exemplo	VisuAlg	Java
Inteiro	2 45 100	inteiro	int
Real	2,456 101,19	real	float
Caractere	"A" "4" "Teste"	caractere	char / string
Lógico	Verdadeiro Falso	logico	true / false

Constantes

- Constante: É um determinado valor fixo que não se modifica ao longo do tempo durante a execução de um programa. Conforme o seu tipo, a constante é classificada como sendo inteira, real, caractere e lógica.
- Exemplo: `constPI = 3.14`

Variáveis

- Variável: É um objeto (uma posição frequentemente localizada na memória), capaz de armazenar um valor ou expressão.
- As variáveis só “existem” em tempo de execução, sendo associadas a “nomes”, chamadas identificadores, durante a criação dos algoritmos.
- Em relação à variável, estamos tratando de uma região de memória (do computador) previamente identificada, cuja finalidade é armazenar os dados ou informações de um programa por um determinado espaço de tempo.

Variáveis

- Pense que a memória do computador se organiza tal qual um armário com várias divisões. Cada divisão é identificada por um endereço diferente em uma linguagem que o computador entende.
- O computador armazena os dados nessas divisões, sendo que em cada divisão só é possível armazenar um dado e, toda vez que o computador armazenar um dado em uma dessas divisões, o dado que antes estava armazenado é eliminado. Mas lembre-se de que o conteúdo pode ser alterado, mas somente um dado pode ser armazenado por vez naquela divisão.

Variáveis

- Uma variável é composta por dois elementos básicos:
 - Conteúdo: valor da variável e identificador;
 - Nome: dado à variável para possibilitar sua utilização e localização.
- Uma vez definidos o nome e o tipo de uma variável, não será possível alterá-los no decorrer do algoritmo. Por outro lado, o conteúdo de uma variável é um objeto de constante modificação no decorrer do programa;
- Em algoritmos, as variáveis serão definidas no início, por meio de um comando definido conforme o exemplo seguinte.

Variáveis

- Exemplo:

```
algoritmo "CALCULA_DOBRO"  
var  
    varNum, varDobro : inteiro  
inicio  
  
finalgoritmo
```

- Como pode ser visto, “varNum” e “varDobro” são os nomes das variáveis, e é por meio desses nomes que é possível alterar e resgatar o valor presente nela.

Variáveis

- **Variáveis Globais:**
- Uma variável global é uma variável acessível em todos os escopos de um algoritmo. O mecanismo de interação com variáveis globais é chamado ambiente global.
- O uso de variáveis globais é geralmente considerado inadequado, pois seu conteúdo pode ser potencialmente modificado de qualquer local e qualquer parte de um código pode depender dele.

Variáveis

- **Variáveis Locais:**
- São aquelas declaradas no início de um subalgoritmo. São visíveis, ou seja, podem ser utilizadas somente pelo subalgoritmo onde foram declaradas. Outros subalgoritmos ou mesmo o algoritmo principal não pode utilizá-la.

Constantes X Variáveis

- A diferença primordial entre esses dois conceitos é que constantes são valores inalterados e variável é uma entidade capaz de manifestar diferenças em valor.
- A variável possui qualquer valor dentro de um campo determinado. Atua como uma “gaveta”, onde se pode guardar qualquer valor, desde que o tipo seja respeitado.

Expressões

- É possível ligar o conceito de expressão ao conceito de expressão (ou fórmula) matemática, onde um conjunto de variáveis e constantes numéricas relaciona-se por meio de operadores, compondo uma fórmula que, uma vez avaliada, resulta em um valor.
- As expressões dividem-se em:
 - Aritméticas;
 - Relacionais;
 - Lógicas.

Expressões - Aritméticas

- Expressões aritméticas são aquelas que apresentam como resultado um valor numérico que pode ser um número inteiro ou real, dependendo dos operandos e operadores. Observe os operadores aritméticos na tabela seguinte.

Expressões - Aritméticas

Operadores	Simbologia	Prioridade
Soma	+	3
Subtração	-	3
Multiplicação	*	2
Divisão	/	2
Módulo	%	2
Exponenciação	^	1
Radiciação	//	1

Expressões - Aritméticas

- O operador de multiplicação é o caractere asterisco;
- Expressões aritméticas podem manipular operandos de dois tipos: reais e inteiros.
 - Se todos os operandos de uma expressão são do tipo inteiro, então a expressão fornece como resultado um número inteiro. Caso pelo menos um dos operandos seja real, o resultado será real.

Exemplos:

- $A + B - C$
- A / B
- $3.14 * (A + B)$

Expressões - Aritméticas

- Regras:
- Primeiro observa-se a prioridade dos operadores.
- Caso haja empate na ordem de prioridade, deve-se resolver a expressão da esquerda para a direita.
- Parênteses servem para mudar a ordem de prioridade de execução das operações. Quando houver parênteses dentro de parênteses, as expressões dentro dos mais internos são **avaliadas primeiro**.

Expressões - Aritméticas

- Exemplos:
- $A=2$
- $B=4$
- $C=1$
- $A*B-C = ?$
 - $A*(B-C) = ?$
 - $B + A / C + 5 = ?$
 - $(B + A) / (C + 5) = ?$

Expressões - Relacionais

- Comparar caracteres e números;
- Os valores a serem comparados podem ser literais ou variáveis;
- Retornam valores lógicos (Verdadeiro ou falso);
- Prioridade é igual às expressões aritméticas;

Descrição	Simbologia
Igual a	=
Diferente de	<>
Maior que	>
Menor que	<
Maior ou igual a	>=
Menor ou igual a	<=

Expressões - Relacionais

- Exemplos:
- $A=5$;
- $B=3$;

Expressão	Resultado
$A = B$	Falso
$A \neq B$	Verdadeiro
$A > B$	Verdadeiro
$A < B$	Falso
$A \geq B$	Verdadeiro
$A \leq B$	Falso

Expressões - Lógicas

- Servem para combinar resultados de expressões, retornando se o resultado final é verdadeiro ou falso;
 - **E:** uma expressão “e” é verdadeira se todas as condições forem verdadeiras;
 - **Ou:** uma expressão “ou” é verdadeira se pelo menos uma condição for verdadeira;
 - **Não:** uma expressão “não» inverte o valor da expressão ou condição; se verdadeira, inverte para falsa e vice-versa;
 - **Xou:** Uma expressão “xou” tem resultado verdadeiro se seus dois operandos lógicos forem diferentes e falso, se forem iguais. Os resultados das operações lógicas são sempre valores lógicos (verdadeiro ou falso);

Expressões - Lógicas

- Tabela verdade: São todas as combinações possíveis entre os operadores:

A	B	A e B
F	F	F
F	V	F
V	F	F
V	V	V

A	B	A ou B
F	F	F
F	V	V
V	F	V
V	V	V

A	B	A xou B
F	F	F
F	V	V
V	F	V
V	V	F

A	nao A
F	V
V	F

Expressões - Lógicas

- Abaixo está uma tabela com todos os valores possíveis criados pelos três operadores lógicos.

Primeiro valor	Operador	Segundo valor	Resultado
Verdadeiro	e	Verdadeiro	Verdadeiro
Verdadeiro	e	Falso	Falso
Falso	e	Verdadeiro	Falso
Falso	e	Falso	Falso
Verdadeiro	ou	Verdadeiro	Verdadeiro
Verdadeiro	ou	Falso	Verdadeiro
Falso	ou	Verdadeiro	Verdadeiro
Falso	ou	Falso	Falso
Verdadeiro	nao		Falso
Falso	nao		Verdadeiro

Funções

- O conceito de função está intimamente ligado ao conceito de função (fórmula) matemática, onde um conjunto de variáveis e constantes numéricas relaciona-se por meio de operadores, compondo uma fórmula que, uma vez avaliada, resulta em um valor.
- As funções ajudam a reduzir a quantidade de expressões dentro de um algoritmo, já que em uma única linha é possível obter o resultado de uma fórmula matemática que poderia resultar em muitas linhas.

Tipos de Funções

- Numéricas:
- São aquelas cujo resultado da avaliação é do tipo numérico, seja ele inteiro ou real.
Somente podem ser efetuadas entre números propriamente apresentados ou variáveis numéricas.

Tipos de Funções

- Caractere:
- São aquelas cujo resultado da avaliação é do tipo caractere. Somente podem ser efetuadas entre caracteres propriamente apresentados ou variáveis literais do tipo caractere.

Tipos de Funções

- Conversão de tipos:
- São aquelas que convertem valores numéricos de um determinado tipo para outro valor numérico de outro tipo;
- Entre esses tipos é possível encontrar diversas funções matemáticas, descritas conforme a próxima tabela.

Tipos de Funções

- As funções matemáticas podem variar de acordo com cada linguagem de programação.

Função	Resultado
sen (x)	Seno de x
cos (x)	Cosseno de x
tg (x)	Tangente de x
arcsen (x)	Arco cujo seno é x
arccos (x)	Arco cujo cosseno é x
arctg (x)	Arco cuja tangente é x
abs (x)	Valor absoluto (módulo) de x
int (x)	A parte inteira de x
frac (x)	A parte fracionária de x
ard (x)	Arredondamento de x
rnd (x)	Valor randômico de x

Atribuição

- Como o próprio nome diz, um comando de atribuição permite fornecer um valor a uma variável (guardar um objeto em uma gaveta), em que o tipo do dado deve ser compatível com o tipo da variável;
- Somente podemos atribuir um valor lógico a uma variável capaz de comportá-lo, ou seja, uma variável declarada como sendo do tipo lógico.

Atribuição

- Exemplo:

```
algoritmo "CALCULA_DOBRO"  
var  
    varNum, varDobro : inteiro  
inicio  
    varNum  $\leftarrow$  3 + 3  
    varDobro  $\leftarrow$  2 * varNum  
finalgoritmo
```

Comandos de entrada e saída

- Um comando de entrada (leitura) tem por função transferir dados do meio externo para a memória de trabalho do computador.
- Os dados do meio externo, na maioria das vezes, entram em nosso computador por meio do teclado;

Comandos de entrada e saída

- Exemplo:

```
algoritmo "CALCULA_DOBRO"  
var  
    varNum, varDobro : inteiro  
inicio  
    leia (varNum)  
finalgoritmo
```

- Como você pode ver nesse exemplo, é possível fazer com que o computador fique aguardando que um valor seja digitado. O valor digitado será armazenado na variável "varNum".

Comandos de entrada e saída

- Ao contrário de um comando de entrada, um comando de saída é utilizado quando necessitamos apresentar valores ao meio externo. Os valores são comumente apresentados na tela do computador.
- Exemplo:

```
algoritmo "CALCULA_DOBRO"  
var  
    varNum, varDobro : inteiro  
inicio  
    leia (varNum)  
    varDobro ← 2 * varNum  
    escreva (varDobro)  
finalgoritmo
```