“Automação é um processo em que uma tarefa deixa de ser executada pelo homem e passa a ser realizada por máquinas, não importando se estas máquinas são mecânicas ou eletrônicas.” (Walter Luiz Caram Saliba. Técnicas de Programação: Uma abordagem estruturada. São Paulo. Makron Books, 1993)

Ocorre automação quando todas as etapas de uma tarefa são conhecidas, e a máquina ou o processo que executarão estas etapas possui condições de fazê-lo, com o objetivo de obter, sempre e no mesmo intervalo de tempo, resultados iguais.

A forma como especificamos ou descrevemos as etapas realizadas durante a automação de uma tarefa é chamada de ALGORITMO.

* Uma sequência lógica de instruções que, a partir dos dados de entrada, realiza um processamento, fornecendo dados de saída.
* Instruções são as regras fornecidas para a realização de uma tarefa.
* Em informática, instrução é cada uma das ações básicas executadas por um computador.

**Variáveis**

* Uma variável é um local na memória do computador que armazena um valor.
* O valor que é armazenado é de um determinado tipo de dado, e deve possuir um nome, através do qual será referenciado.
* No decorrer da execução de um programa uma variável pode assumir valores diferentes.
* Para compor o nome de uma variável existem algumas regras:
  + O primeiro caractere é uma letra.
  + Se houver mais de um caractere, só poderemos usar: letra ou algarismo.
  + Nenhuma palavra reservada à ferramenta (linguagem de programação) poderá ser usada como nome de uma variável.
* As variáveis são definidas no início de um programa, quando o espaço necessário para seu armazenamento é reservado na memória.
* ***Lembre-se: No decorrer da execução de um programa uma variável pode assumir valores diferentes, a cada instante.***

**Declaração de variáveis**

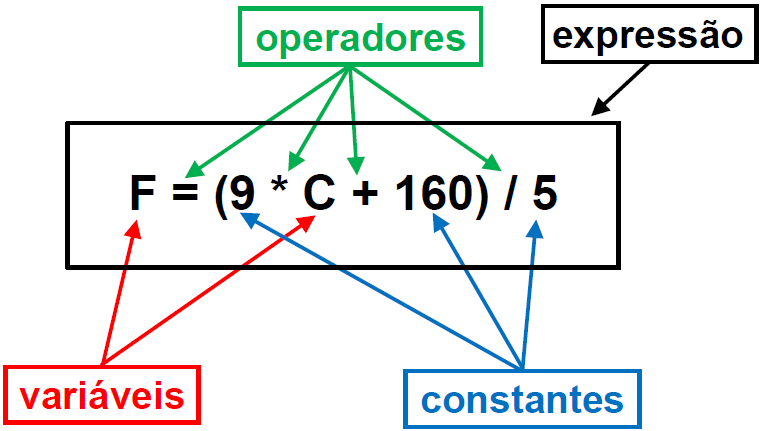
* Declarar uma variável significa defini-la, no início de um programa, ou seja, relacionar formalmente um nome a um tipo de dados, respeitando as regras existentes.
* É muito importante que um nome de uma variável tenho significado relacionado ao conteúdo que irá armazenar.
* Exemplos de declaração de variáveis:
  + literal nome
  + real altura, base,raio
  + inteiro idade

**Tipos de dados e sua utilização**

* Chamamos de tipo de dado o “conjunto de valores, operações e relações já implementadas (disponíveis para uso)” em uma linguagem de programação.
* Quando escrevemos um programa em uma linguagem de programação, surge o conceito de domínio, ou seja, da limitação do conjunto de valores dos elementos representados.
* Os tipos de dados que utilizamos em Algoritmos (ou pseudocódigo) são: inteiro, real, lógico e literal.
* “O tipo de dados INTEIRO é aquele que não possui componentes decimais ou fracionários, podendo ser positivo ou negativo.” Exemplos:
  + 45
  + 0
  + -13
* “O tipo de dado Real é aquele que pode possuir componentes decimais ou fracionários, podendo também ser positivo ou negativo. Exemplos:
  + 45.5
  + 0.1
  + -13.45
* Tipo de dado LITERAL é definido como “constituído por uma sequência de caracteres com letras, dígitos e/ou símbolos especiais. Este tipo de dado é também muitas vezes chamado de alfanumérico, cadeia de caracteres ou, ainda, String.” (Walter Luiz Caram Saliba. Técnicas de Programação: Uma abordagem estruturada. São Paulo. Makron Books, 1993). Exemplos:
  + “Hoje é dia 05 de março”
  + “R$45.02”
  + “ ”
  + “XyZ731%&#”
* Tipos de Dados – LÓGICO. O tipo LÓGICO representa dados com valor verdadeiro e falso. Importante: este tipo de dado poderá representar apenas um dos dois valores.
* É importante perceber a diferença entre dados LITERAIS e NUMÉRICOS (INTEIRO ou REAL):
  + “R$45.02” - literal
  + 45.02 - real
  + 45 - inteiro

**Expressões**

* “Uma expressão é uma combinação de variáveis, constantes e operadores, que, uma vez avaliada, resulta em um valor.” (Walter Luiz Caram Saliba. Técnicas de Programação: Uma abordagem estruturada. São Paulo. Makron Books, 1993)
* Quais são os conceitos envolvidos na definição de expressão?
  + Variáveis – tratam valores que podem sofrer alterações ao longo da execução do programa.
  + Constantes – são valores que, ao longo da execução de um programa, não são alterados.
  + Operadores – atuam sobre os operandos produzindo resultados.

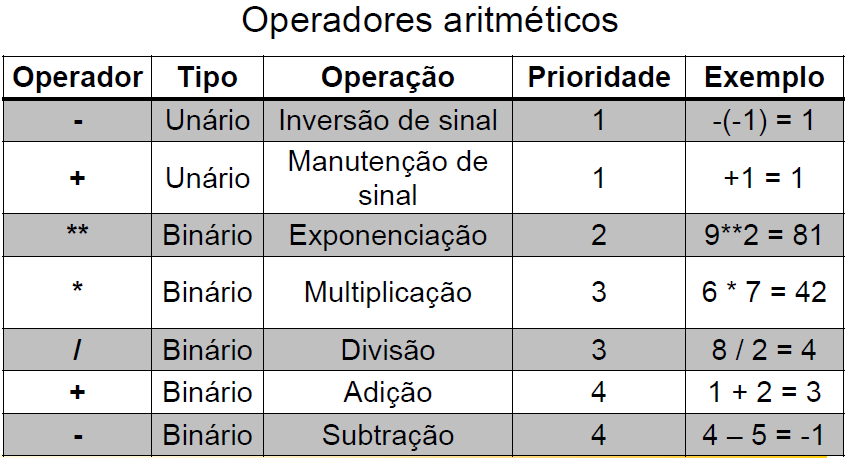


**Operadores lógicos, aritméticos e relacionais e prioridade de operadores**

* Os operadores podem ser:
  + Binários: atuam sobre dois operandos, realizando uma operação:
    - 45 \* 2
  + Unários: atua sobre um operando, cuja operação é a inversão ou manutenção de sinal:
    - 1

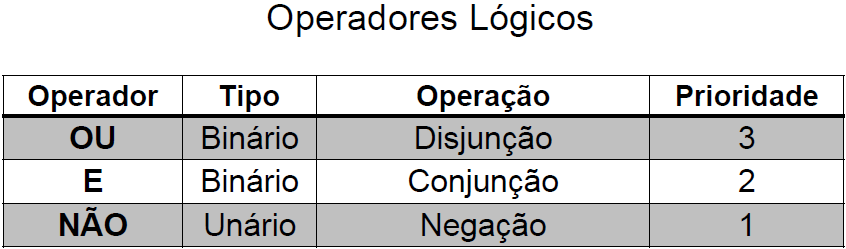
**As expressões cujo resultado é do tipo numérico são chamadas de expressões aritméticas.**

* Estas expressões utilizam variáveis numéricas e operadores aritméticos.

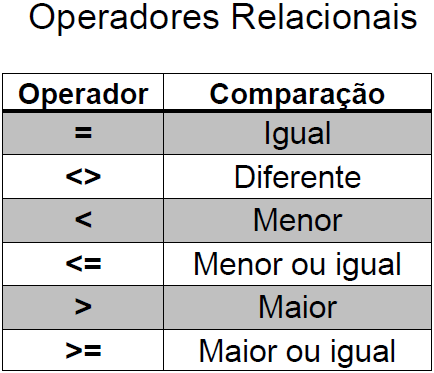


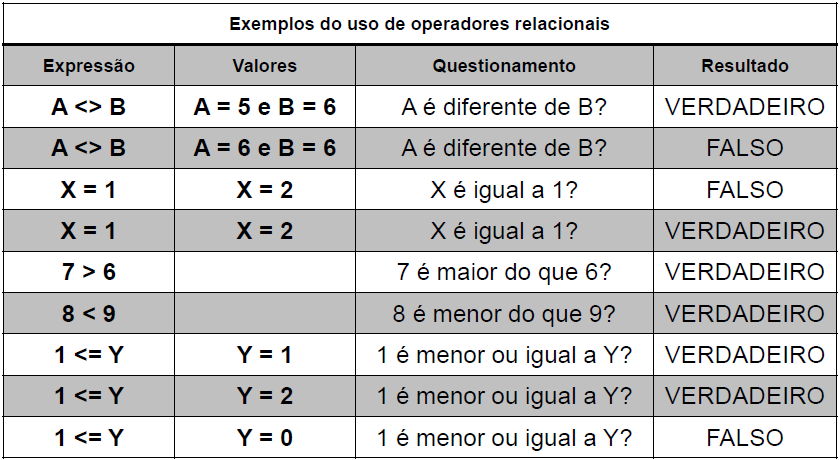
**As expressões cujo resultado é do tipo lógico (verdadeiro ou falso) são chamadas de expressões lógicas.**

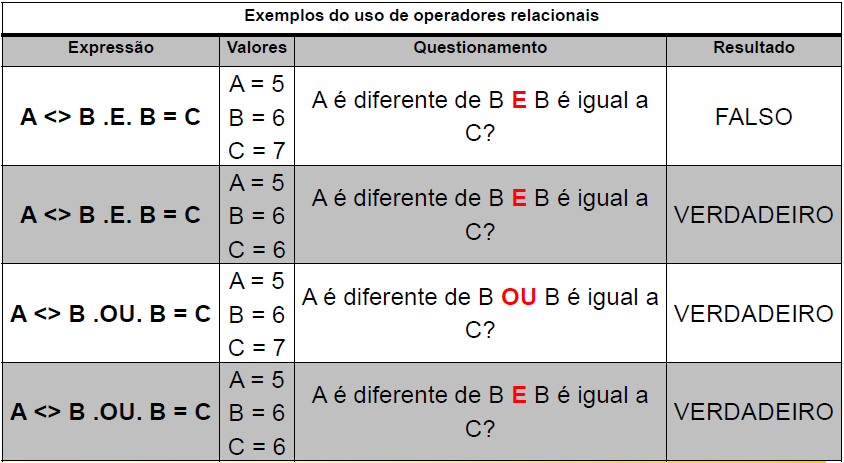
* Estas expressões utilizam operadores lógicos e operadores relacionais.
* Os operadores lógicos atuam sobre proposições e os operadores relacionais são utilizados quando realizamos comparações.



* Uma proposição é uma sentença que exprime um pensamento de sentido completo.
* Podem ser simples (possuem apenas uma proposição) ou compostas (possuem mais de uma proposição ligadas por operadores lógicos).
* Proposições simples:
  + •Hoje é terça-feira.
  + •O número 13 é ímpar.
* Cada uma destas proposições pode nos devolver dois resultados: FALSO ou VERDADEIRO
* Proposições compostas:
  + Hoje é terça-feira **e** choveu.
  + O número 13 é ímpar **e** primo.
  + Amanhã assisto ao jogo **ou** vou à aula.
* Cada uma destas proposições pode nos devolver dois resultados: FALSO ou VERDADEIRO
* As operações realizadas sobre proposições são:
  + Negação;
  + Conjunção;
  + Disjunção.
* Negação (NÃO)
  + Dada uma proposição chamada de p, a negação de p (NÃO p) significa a inversão do valor de p. Se o valor de p for VERDADEIRO, ele passa a ser FALSO. Se for FALSO passa a ser VERDADEIRO.
* Conjunção (E)
  + Dada a conjunção de duas proposições a e b (a E b), seu valor será VERDADEIRO quando ambas forem verdadeiras, e FALSO nos demais casos.
* Disjunção (OU)
  + Dada a disjunção de duas proposições a ou b (a ou b), seu valor será VERDADEIRO quando ao menos uma das duas proposições for verdadeira, e FALSO quando as duas forem falsas.

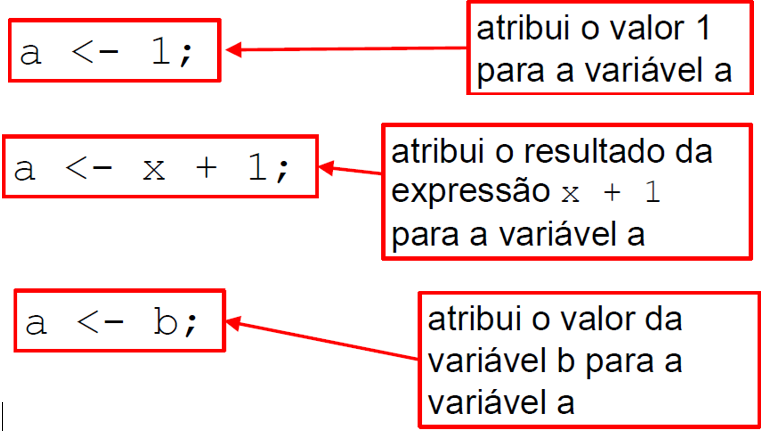






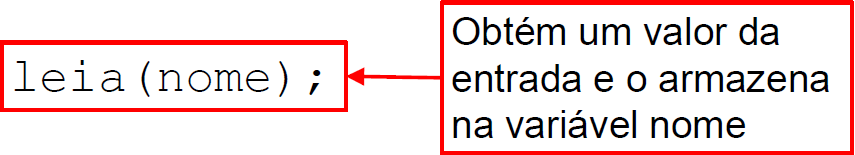
**Comando de atribuição**

* É representado por uma senta que aponta para a esquerda (<-), e atribui um valor a uma variável (armazena um valor, ou o conteúdo de uma variável, ou o resultado de uma expressão, em uma variável).



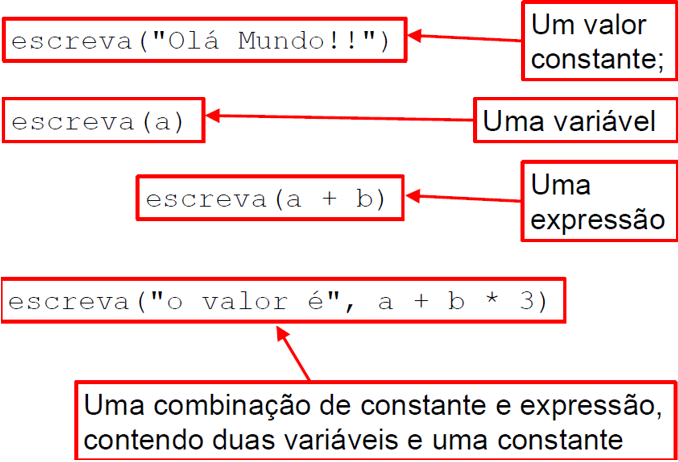
**Comando de Entrada**

* Os dados de entrada para um algoritmo são obtidos através de um Comando de Entrada, e armazenados em uma variável, que é definido como leia(variável).



**Comando de Saída**

* Os dados de saída, ou informações, ou orientações para o usuário do algoritmo são exibidos através de um Comando de Saída.
* A execução padrão de um comando de saída mostra uma informação no monitor.
* Definimos um comando de saída como: escreva(<instruções>) Por <instruções> entendemos:
  + Um valor constante;
  + Uma variável;
  + Uma expressão;
  + Uma combinação de constantes, variáveis e expressões.



**Estrutura sequencial – estrutura básica de um algoritmo/pseudocódigo**

* Muitas vezes, quando falamos de Algoritmos e de Português Estruturado (ou até mesmo de Lógica de Programação), estamos nos referindo a pseudocódigo.
* Pseudocógico é uma forma de representação de um algoritmo, muito próxima de uma linguagem de programação.
* As regras para a escrita da solução de um problema, quando utilizamos Descrição Narrativa são bem flexíveis.
* Quando utilizamos pseudocódigo as regras passam a ser rígidas, já que vamos representar as ações que um algoritmo deverá tomar.
* Estrutura básica para um pseudocódigo

**programa** <nome>

**declaracoes**

<lista de variáveis>

**inicio**

<comandos>

**fim**

**programa** – representa o início do pseudocódigo ou algoritmo que estamos escrevendo. <nome> - é o nome que será dado ao programa (não utilizar caracteres especiais).

* **declarações** – marca o início da área onde as variáveis serão declaradas.
* <lista de variáveis> – é a área onde escrevemos a lista de variáveis que serão utilizadas em nosso algoritmo.
* **inicio** – marca o início do algoritmo. A partir deste ponto escrevemos a solução do problema proposta começa.
* <comandos> – nesta área vamos escrever os comandos e instruções que executarão as entradas de dados, o processamento e as saídas de dado de nosso algoritmo.
* **fim** – marca o fim do algoritmo. Como o **inicio**, é obrigatoriamente informado.

**Exemplo 01**

Tendo como dados de entrada dois pontos quaisquer de um plano cartesiano P(X1,Y1)e Q(X2,Y2) calcule e mostre a distância entre eles. A distância é dada pela fórmula: raiz((X2 - X1)^2 + (Y2 - Y1)^2)

**primeiro passo:** relacionar os dados de entrada:

* coordenadas de P (x1 e y1);
* coordenadas de Q (x2 e y2).

**segundo passo:** relacionar os dados de saída:

* distância entre P e Q

**terceiro passo:** relacionar as etapas para a solução do problema:

* obter o valor da coordenada X para o ponto P;
* obter o valor da coordenada Y para o ponto P;
* obter o valor da coordenada X para o ponto Q;
* obter o valor da coordenada Y para o ponto Q;
* calcular a distância entre os pontos;
* mostrar a distância calculada.

**quarto passo:** como o processamento será executado:

* de posse dos valores das coordenadas dos pontos P e Q, resolvemos a fórmula:

distância = raiz((X2 - X1)^2 + (Y2 - Y1)^2)

**d <- ((x2-x1)^2 + (y2-y1)^2)^0.5**

**quinto passo:** escrever o algoritmo(pseudocódigo):

**programa ex08**

**declaracoes**

**real x1, x2, y1, y2, d**

**inicio**

**escreva("digite o valor de x para o ponto P: ")**

**leia(x1)**

**escreva("digite o valor de y para o ponto P: ")**

**leia(y1)**

**escreva("digite o valor de x para o ponto Q: ")**

**leia(x2)**

**escreva("digite o valor de y para o ponto Q: ")**

**leia(y2)**

**d <- ((x2-x1)^2 + (y2-y1)^2)^0.5**

**escreva("a distância entre os dois pontos é de: ", d)**

**fim**

**Exemplo 02**

Calcular o preço de venda de um carro. O preço de venda é formado pelo **preço da montadora**, mais 15% de **lucro**, mais 11% de **IPI**, mais 17% de **ICM**. As porcentagens são sobre o preço da montadora, que é lido. Informe o **preço final** e o **valor dos impostos**.

**primeiro passo:** relacionar os dados de entrada:

* preço da montadora

**segundo passo:** relacionar os dados de saída

* preço final (preço de venda)
* valor dos impostos

**terceiro passo:**

relacionar as etapas para a solução do problema:

* obter o preço da montadora
* calcular o valor do lucro
* calcular o valor do ipi
* calcular o valor do icm
* calcular o valor final do veículo (preço de venda)
* calcular o valor total dos impostos

**quarto passo:**como o processamento será executado:

* a partir do preço da montadora:
  + calcular o lucro: lucro = preço da montadora \* 0,15
  + calcular o ipi: ipi = preço da montadora \* 0,11
  + calcular o icm: icm = preço da montadora \* 0,17
  + calcular o preço de venda: preço de venda = preço da montadora+lucro+ipi+icm
  + calcular o valor total dos impostos: valor total dos impostos = ipi + icm

**quinto passo:** escrever o algoritmo (pseudocódigo)

**programa ex05**

**declaracoes**

**real pm, lucro, icm, ipi, pv, vt**

**inicio**

**escreva("Digite o preço da montadora :")**

**leia(pm)**

**lucro <- pm \* 0.15**

**ipi <- pm \* 0.11**

**icm <- pm \* 0.17**

**pv <- pm + lucro + ipi + icm**

**vt <- ipi + icm**

**escreva("O valor total dos impostos é de :", vt)**

**escreva("O preço de venda é :", pv)**

**fim**