

## Portugol ou Pseudo-Código (Linguagem Estruturada)

### Conceito

A Linguagem Estruturada é a forma que tem sido mais utilizada para a elaboração de algoritmos. É a forma que mais se assemelha com a forma em que os programas são escritos nas linguagens de programação.

A Linguagem Estruturada, ou algorítmica, é suficientemente geral para que a passagem do algoritmo ao programa deva ser quase que uma operação direta, não importando a linguagem de programação a ser utilizada.

### Operadores

**Operadores Aritméticos:** +; -; \* (multiplicação); / (divisão real); **DIV** (divisão de inteiros); **MOD** (resto da divisão inteira);  $\uparrow$  (potenciação).

Exemplos:  $1.0/0.5 = 2.0$ ;  $7/2 = 3.5$ ;  $7DIV2 = 3$ ;  $7MOD2 = 1$ ;  $2^2 = 2 \uparrow 2$ .

**Operadores Relacionais:** < (menor);  $\leq$  (menor ou igual); = (igual);  $\neq$  (diferente);  $\geq$  (maior ou igual); > (maior).

### Operadores Lógicos: E; OU; NÃO

- Operador Lógico E -> só resultará em VERDADEIRO, se todas as condições forem verdadeiras.
- Operador Lógico OU -> só resultará em FALSO, se todas as condições forem falsas.
- Operador Lógico NÃO -> nega uma condição. Então, se vier antes de uma condição verdadeira, o resultado será FALSO. E, se vier antes de uma condição falsa, o resultado será VERDADEIRO.

Veja o quadro a seguir:

X	Y	X E Y	X OU Y	NÃO X	NÃO Y
V	V	V	V	F	F
F	V	F	V	V	F
V	F	F	V	F	V

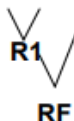
F	F	F	F	V	V
---	---	---	---	---	---

## Hierarquia de Operadores:

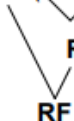
1. Parênteses (com vários níveis)
2. Funções
3. Potenciação ( $\uparrow$ )
4.  $*$ ;  $/$ ; **DIV**; **MOD**
5.  $+$ ;  $-$
6. Operadores Relacionais

### Exemplos:

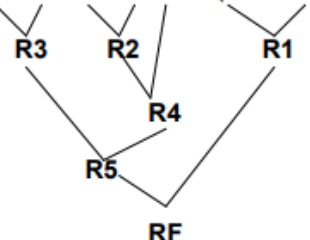
1)  $X + Y - Z$



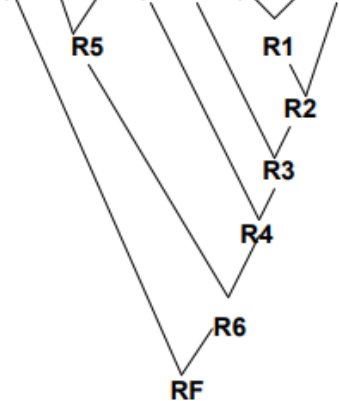
2)  $X * (Y - Z)$



3)  $A = B * C - E \uparrow F / G + (A \text{ DIV } B) \Rightarrow A \leftarrow RF$



4)  $X = (A - B / C * (E + A * (C \uparrow B - E))) \Rightarrow X \leftarrow RF$



**Observação:** Com prioridades iguais, como no exemplo ao lado os sinais  $+$  e  $-$ , resolve-se primeiro a operação mais à esquerda.

## Regras para Construção dos Algoritmos

### Declaração de Constantes

- Constantes são dados que durante a execução do programa, permanecem com os seus valores inalterados;
- É importante tratá-las pelo nome.

## **Declaração de Variáveis**

- Variáveis são dados cujos valores variam durante a execução do programa;
- São entidades que representam dados do programa;
- Possuem Nome e Valor;
- Representa uma posição de memória do computador. Ex: A=3; X=0; Y=10; K=0.01  
 { *Num dado momento, pois depois todos os valores podem estar alterados.*
- Tipos de Variáveis:
  - Numéricas:
    - Inteiras. Ex.: MÍNIMO = 0
    - Reais. Ex.: PI = 3.1416...
  - Literais. Ex: CIDADE = "Volta Redonda".; ESTADO = "R.J.";
  - Caracter. Ex: RESP = "S"; SEXO = "M"; LETRA = "A".
  - Lógicas. Ex: ACHEI = Verdadeiro; CHAVE = Falso.

## **Formação do Identificador (NOME das Variáveis e Constantes)**

- Começa sempre com uma letra que pode ser seguida de mais letras e/ou números;
- Deve ter no máximo 8 caracteres;
- Não são permitidos caracteres especiais (só letras e números);
- O nome deve refletir o significado da variável ou constante. Ex: X=16; Y=02; Z=1998 (*não são bons nomes*), mas DIA=10; MÊS=02; ANO=2012 (*são bons nomes*).

## **Comandos Básicos**

### **Comando de Atribuição (←)**

A instrução de ATRIBUIÇÃO permite que o conteúdo de uma variável seja alterado.

**Sintaxe:**

***Nome ← valor***

Um valor ou o resultado de uma *expressão* será armazenado sob um *nome* simbólico que está a esquerda do sinal de atribuição “ ← ”.

Tal expressão poderá ser um simples valor atribuído à uma constante ou variável, ou ainda poderá ser uma expressão aritmética, envolvendo outras variáveis previamente definidas. Contudo, o tipo do valor do resultado obtido através do cálculo da expressão, deve ser do mesmo tipo da variável que irá receber este valor.

**Exemplos:**  $X \leftarrow 0$ ;  $Y \leftarrow X$ ;  $A \leftarrow X+Y$  (expressão aritmética);  $DIA \leftarrow 30$ .

**Comando de Leitura**

Transporta informações de um periférico de entrada para a memória principal do computador.

As informações são lidas de um dispositivo de entrada, geralmente do teclado.

**Sintaxe:**

***leia (lista-de-variáveis)***

**Onde:** A execução da instrução de leitura pressupõe que os dados serão fornecidos do meio externo (dispositivo de entrada - teclado) e serão armazenados na memória sob os nomes simbólicos explicitados na *lista-de-variáveis*, na ordem dada.

**Exemplos:**

leia (NOME, TELEFONE)

leia (DIA, MES, ANO)

leia (NUMERO)

leia (PRODUTO, PRECO)

**Comando de Impressão (escreva)**

Transporta informações da memória principal do computador para um periférico de saída.

As informações são exibidas em um dispositivo de saída, geralmente em impressora ou vídeo.

**Sintaxe:**

### **escreva (*lista-de-variáveis/constantes* ou “*texto*”)**

A execução da instrução de impressão pressupõe que os dados estejam armazenados na memória e serão colocados disponíveis no meio externo (dispositivo de saída – impressora ou vídeo) através dos nomes simbólicos atribuídos às variáveis ou às constantes.

A opção “*texto*” prevista no formato da instrução permite também que sejam explicitados textos para a exibição de mensagens.

#### **Exemplos:**

escreva (“UniFoa – Centro Universitário de Volta Redonda”) → imprime texto

escreva (DIA, MES, ANO) → imprime os valores (conteúdo) das variáveis DIA, MES e ANO

escreva (“Dia = “, DIA, “Mês = “, MES, “Ano = “, ANO) → imprime textos e os valores (conteúdos) das variáveis.

escreva (“O resultado da expressão  $5 * 3 - 2 + 4 / 2$  é = “,  $5*3-2+4/2$ )  
↳ imprime texto e o resultado da expressão

### **Detalhamento das Regras**

- 1) ALGORITMO é sempre a primeira palavra, seguida do título (minúsculo);
- 2) Os passos são numerados;
- 3) Início e Fim claros;
- 4) Os passos (comentários) são escritos entre colchetes [ ];
- 5) Estruturas serão grifadas;
- 6) Usar “identação”, que é um deslocamento para direita das instruções subordinadas.

### **Estruturas Básicas**

#### **Sequência Simples**

As instruções do algoritmo são executadas uma após a outra, sem que haja desvios na sequência das instruções. Cada instrução é executada uma e somente uma vez.

**Exemplo:** Algoritmo para ler um valor numérico qualquer e imprimir o valor lido.

#### **ALGORITMO exemplo da Estrutura Sequencial**

- 1 [Início]
- 2 [Declaração de Variáveis]  
    NUMERO: real
- 3 [Leitura do Valor Numérico]

escreva ("Entre com um valor numérico: ")

leia (NUMERO)

4 [Impressão do número lido]

escreva ("O número lido foi: ", NUMERO)

5 [Fim]