

## ÍNDICE

### I – ALGORITMO

#### 1- INTRODUÇÃO

#### 2- DEFINIÇÕES DE ALGORITMO

#### 3- CONDIÇÕES DE ENTRADA E DE SAÍDA

#### 4- LINGUAGEM DE PROGRAMAÇÃO

### II - PARTES DE UM PROGRAMA PASCAL

#### 1- ORGANIZAÇÃO

#### 2- IDENTIFICADORES

#### 3- TIPOS DE DADOS

#### 4- COMANDO DE ATRIBUIÇÃO

#### 5- EXPRESSÕES

#### 6- FUNÇÕES PRÉ-DEFINIDAS

#### 7- OPERAÇÕES DE ENTRADA E SAÍDA

#### 8- ESTRUTURA SEQUENCIAL

#### 9- ESTRUTURAS DE SELEÇÃO

#### 10- ESTRUTURAS DE REPETIÇÃO

#### 11- VARIÁVEIS COMPOSTAS

#### 12- MODULARIZAÇÃO DO ALGORITMO

#### 13- FUNÇÕES

### I - ALGORITMO

#### 1 – INTRODUÇÃO

O computador é capaz de coletar informações, processar e nos fornecer um resultado com extrema rapidez e eficiência, mas não passam de máquinas de inteligência restrita, para isto é necessário que o homem desenvolva os programas, softwares capazes de nos fornecer estes resultados. Uma das etapas fundamentais para o desenvolvimento dos softwares é a construção de Algoritmo.

#### 2 – DEFINIÇÕES DE ALGORITMO

- Processo de cálculo, ou de resolução de um grupo de problemas semelhantes, em que se estipulam, com generalidade e sem restrições, regras formais para a obtenção do resultado, ou da solução do problema (Aurélio).

- Conjunto predeterminado e bem definido de regras e processos destinados à solução de um problema, com um número finito de etapas (Aurélio).

- Uma seqüência de operações que devem ser executadas ordenadamente de forma não-ambígua com o objetivo de resolver um determinado problema.

Por definição verificamos que algoritmo não se aplica apenas a informática, pode ser caracterizado tanto para uma receita de bolo quanto para montagens Industriais. Usamos algoritmos em procedimentos humanos diariamente.

#### 3- CONDIÇÕES DE ENTRADA E DE SAÍDA

Nenhum algoritmo será completo, normalmente algumas condições serão esquecidas, assim ao se construir um algoritmo o seu responsável deve especificar as condições de entrada e de saída.

#### 4- LINGUAGEM DE PROGRAMAÇÃO

É o conjunto de símbolos e regras que associamos semânticas utilizadas para programar computadores. Ao traduzir-se um Algoritmo para uma linguagem de programação obtêm-se um programa, que passará a ser compreendido pelo computador. Existem várias linguagens de Programação, podemos citar algumas delas: PASCAL, COBOL, DELPHI, VISUAL BASIC.. Usaremos em nosso curso a linguagem PASCAL, sendo esta a mais indicada no ensino acadêmico.

### II - PARTES DE UM PROGRAMA PASCAL

#### 1- ORGANIZAÇÃO

- Cabeçalho: Nome do Programa;

- Dicionário de dados: Define os dados (variáveis e constantes) a serem usados no programa;

- Corpo: Define o tratamento a ser dado aos dados de programa;

- Comentários: Auxilia nas outras partes do programa, esclarecendo aspectos, e outras peculiaridades do programa.

## 2 – IDENTIFICADORES

É uma seqüência de caracteres alfanuméricos começados obrigatoriamente por uma letra e por caracteres seguintes, que podem ser letras ou dígitos, não sendo permitido o uso de caracteres especiais, podendo ser:

### 2.1- Palavras Reservadas. São elementos definidos pela linguagem. Exemplos:

And – Operador lógico e ;	Array – Marca o início de um vetor;
Begin – marca o início de um bloco de código;	Case - Comando de seleção;
Const – Define constantes;	Div- Operação de divisão;
Do – Marca o início de comandos das operações: For e While;	Downto – decrementa uma variável;
Else – Usado juntamente com o comando if. (senão);	End – Marca o final de um bloco de código;
File – Marca o início da definição de um tipo arquivo;	Not – Operador lógico de negação;
For – Tipo de iteração que contém iniciação e testes automáticos;	If – comando de seleção simples;
Goto – executa comando forçado de desvio por um outro ponto “label”;	Mod - Operador de resto de divisão entre inteiros;
Label – permite declarar nomes para comandos;	Procedure- Chave de identificação de procedimento;
Or – operador lógico (ou);	Repeat – comando de iteração (repita);
Program – Chave para declaração de programas;	Strig – tipo cadeia de caracteres;
Set – usado para criar variáveis enumeráveis;	To – incremento usado no comando em for;
Then – marca o início de comando do teste if;	Until – Usado no comando Repeat (até);
Type – chave para declarar tipo dados;	While – Comando de iteração (enquanto);
Var – chave para declaração de Variáveis;	

### 2.1- Constante (CONST)

É um determinado valor fixo que não se modifica, durante a execução de um programa. Conforme o seu tipo, podemos classifica-la como sendo numérica, lógica e literal.

Exemplo:	PROGRAM Constante;	“Cabeçalho”
	CONST	“Dicionário de Dados”
	Pi:= 3.1415926;	
	Maxnum:= 50;	
	i:=10;	
	j := 100;	
	BEGIN	“Início do corpo do programa”
	. . . .	

As constantes são definidas no dicionário de dados dos programas, possibilitando a sua utilização no corpo do programa.

### 2.3- Variável (VAR)

São elementos que recebem valores. Cada variável corresponde a um endereço de memória, cujo o conteúdo pode variar durante a execução de um programa. Embora a variável possa assumir diferentes valores, ela só armazena um valor a cada instante.

Exemplo:	PROGRAM Variáveis;	“Cabeçalho”
	VAR	“Dicionário de Dados”
	Total, Maior, Menor := INTEGER;	
	Nota, Media, altura := REAL;	
	Resposta := BOOLEAN;	
	BEGIN	“Início do corpo do programa”
	. . . .	

As variáveis também são definidas no dicionário de dados do programa, possibilitando a sua utilização no corpo deste.

## 3. TIPOS DE DADOS

A natureza dos valores (dados) que podemos armazenar em uma variável é denominado tipo de dado.. Os tipos de dados são: Numeral (Inteiro ou Real); Literal (caracter); Lógico (verdadeiro ou falso).

### 3.1-Integer (Numeral Inteiro)

É designado por um intervalo de números inteiros (Exemplo:1, -1, -5000, 9999), associado as seguintes operações: Soma: + Subtração: - Multiplicação: \* Divisão: DIV Resto da divisão inteira: MOD.

Obs.: A operação divisão (/) pode ser aplicada a valores Inteiros, porém o resultado será um valor Real.

### 3.2-Real (Numeral Real)

É designado por um intervalo de números reais (Exemplo: 1.5, -2.48), possuindo as seguintes operações associadas:  
Soma: +                      Subtração: -                      Multiplicação: \*                      Divisão real: /

### 3.3-Char (Literal Character)

Designa qualquer carácter (letras, números e caracteres especiais).

### 3.4-String (Conjunto Literal Character)

Designa um conjunto de carácter

### 3.5-BOOLEAN (Lógico Verdadeiro ou Falso)

Designa apenas dois valores, TRUE e FALSE, possuindo as seguintes operações associadas:  
AND                      NOT                      OR

## 4- COMANDO DE ATRIBUIÇÃO

É a maneira de se atribuir um valor para uma variável, sendo adotado pelo PASCAL um símbolo de atribuição “:=”.

Exemplo: Menorvalor:=50; j := 100; Nota:=10.50;

Deve se tomar cuidado, pois os tipos de dados devem ser compatíveis com os predefinidos na declaração das variáveis.

## 5-EXPRESSÕES

### 5.1 - Expressões Aritméticas

São operadores aritméticos cujos operandos são constantes e/ou variáveis do tipo numérico (adição, subtração, multiplicação, divisão, potenciação e radiciação).

Existe sempre uma relação de prioridade:

Prioridade 1 = multiplicação, divisão;

Prioridade 2 = adição, subtração.

Podemos utilizar parênteses para alterar as prioridades, não se pode usar colchetes ou chaves.

### 5.2 – Expressões Relacionais

É uma comparação realizada entre valores do mesmo tipo básico (=, <, >, <>, ), fornecendo um resultado do tipo BOOLEAN (TRUE ou FALSE).

Sendo formada por três partes: <expressão da esquerda> <operador> <expressão da direita>

Os operadores podem ser: = Igual; <> Diferente; < Menor que; > Maior que;

<= Menor ou igual; >= Maior ou igual.

### 5.3- Expressões Lógicas

#### 5.3.1- Definição

Denomina-se expressão lógica a expressão cujos operadores são lógicos e cujos os operandos são relações constantes e/ou variáveis do tipo lógico, fornecendo um resultado do tipo BOOLEAN (TRUE ou FALSE).

Os operadores lógicos são: AND Conjunção; OR Disjunção; NOT Negação.

#### 5.3.2- Proposição

É todo o conjunto de palavras ou símbolos que exprimem um pensamento de sentido completo, isto é, afirmam fatos ou exprimem juízos que formamos a respeito.

#### 5.3.3- Princípios Adotados como Regras nas expressões Lógicas

Princípio da não contradição - uma proposição não pode ser verdadeira e falsa ao mesmo tempo.

Princípio do terceiro excluído - toda proposição ou é verdadeira ou é falsa, isto é, verifica-se sempre um destes casos e nunca um terceiro.

Valores Lógicos das Proposições - chama-se valor lógico de uma proposição a verdade se a proposição é verdadeira e a falsidade se a proposição é falsa.

VALOR LÓGICO	SÍMBOLO
Verdade (TRUE)	V
Falsidade (FALSE)	F

#### 5.3.4- Operadores Lógicos

- AND - A conjunção de duas proposições é verdadeira se e somente se ambas são verdadeiras.

Variável	Variável	Resultado
----------	----------	-----------

A	B	A and B
V	V	V
V	F	F
F	V	F
F	F	F

- OR- A disjunção de duas proposições é verdadeira se e somente se, pelo menos, uma delas for verdadeira.

Variável	Variável	Resultado
A	B	A or B
V	V	V
V	F	V
F	V	V
F	F	F

- NOT - A negação de uma variável é alterar o seu resultado BOOLEAN, ou seja, o que é TRUE passa a ser FALSE e vice-versa.

Variável	NEGAÇÃO
A	DE A
V	F
F	V

### 5.3.5- Prioridades das Expressões

PRIORIDADE	OPERADOR
1	NOT
2	*, / , DIV, MOD, AND
3	+, -, OR
4	=, <, >, <=, >=

## 6- FUNÇÕES PRÉ-DEFINIDAS

- ABS - É o valor absoluto(módulo) de um número, usados nos tipos de dados: INTEGER ou REAL.  
Exemplo: ABS(5) = 5    ABS(-5) = 5
- PRED e SUCC - São predecessor e sucessor, usados nos tipos de dados: INTEGER, CHAR e BOOLEAN.  
Exemplo: PRED(5) = 4    PRED(FALSE) = TRUE    SUCC("f")= "g"
- SQR e SQRT - São quadrado e raiz quadrada respectivamente, usados no tipos de dados: INTEGER E REAL.  
Exemplo: SQR(4) = 16    SQRT(16) = 4
- TRUNC - Transforma um valor real em Inteiro, abandonando sua parte fracionada, tipos de dados: REAL.  
Exemplo: TRUNC(5.1) = 5    TRUNC(5.8) = 5
- ROUND- Arredonda o número Real para o Inteiro mais próximo, usado no tipo de dados: REAL.  
Exemplo: ROUND(5.1) = 5    ROUND(5.8) = 6
- ODD- Retorna verdadeiro (TRUE) se o número for ímpar, usado nos tipo de dados: INTEGER.  
Exemplo: ODD(6) = FALSE    ODD(5) =TRUE
- SIN, COS, ARQ- São as funções trigonométricas: Seno, Coseno e Arco-Tangente, para ângulos em radianos, usados nos tipos de dados: INTEGER e REAL.
- LN e EXP- São o logaritmo neperiano e exponencial respectivamente, usados nos tipos de dados: INTEGER e REAL.
- UPCASE - Transforma um caracter minúsculo em maiúsculo, usado nos tipos de dados: CHAR  
Exemplo: UPCASE ('a') = 'A'
- MODE- É o resto de uma divisão Inteira- Exemplo: 6 MOD 3 = 0

## 7- OPERAÇÕES DE ENTRADA E SAÍDA

O computador armazena os valores das variáveis em sua memória.

Para determinarmos estes valores através de um programa, usamos os operadores de entrada de dados, e se quisermos mostrá-los usamos os operadores de saída de dados:

**OPERADORES DE ENTRADA DE DADOS:**

- READ(<,.....,>), transfere um ou mais dados para uma ou mais variáveis;
- READLN(<,.....,>) transfere um ou mais dados para uma ou mais variáveis e posiciona no início da próxima linha.
- OPERADORES DE SAÍDA
- WRITE(<,.....,>), Mostra texto, e/ou dados na tela;
- WRITELN(<,.....,>), Mostra texto, e/ou dados na tela e posiciona o cursor no início da próxima linha

Obs.: Todo dado a ser lido deve ser do mesmo tipo das variáveis especificadas no dicionário de dados.

## 8- ESTRUTURA SEQUENCIAL

É o conjunto de ações que serão executadas numa seqüência linear de cima para baixo e da esquerda para a direita, isto é, da mesma ordem que foram escritas.

Toda ação é seguida por um ponto-e-virgula (;), que tem o objetivo de separar uma ação de outra e auxiliar a organização seqüencial das ações.

### Estrutura:

Início {começo do algoritmo}

Comando A;

Comando B;

.....

Comando n;

Fim. {fim do algoritmo}

Exemplo: Calcular a média aritmética de tres notas de um Aluno:

### Algoritmo:

Real : N1, N2, N3, MA;

Leia (N1, N2, N3) ;

MA <= (N1+ N2 + N3) / 3;

Escreva (MA)

Fim Algoritmo.

### Algoritmo traduzido para o PASCAL

PROGRAM Media3numeros;

{Este programa calcula a media aritmetica das 3 notas de um aluno}

USES

CRT;

VAR

N1,N2,N3,MA: REAL;

BEGIN

CLRSCR;

WRITELN('Entre com as tres notas do aluno para calcular a media');

READLN (N1,N2,N3);

MA:= (N1+N2+N3) / 3;

WRITELN('A media do Aluno e= ',MA:2:2);

READLN;

END.

## 9- ESTRUTURAS DE SELEÇÃO

É a seleção que permite a escolha de um grupo de ações e estruturas a ser executado por determinadas condições, representada por expressões lógicas, são ou não satisfeitas.

### 9.1- Seleção Simples

É uma expressão lógica que quando inspecionada, pode gerar um resultado falso ou verdadeiro.

### Estrutura:

Se (condição)

Então

Comando único

Fim Se

Exemplo:

Calcular o resultado da média aritmética das tres notas do aluno, e só mostra o resultado se a média for maior que 7.

Algoritmo:

Real : N1, N2, N3, MA;

Leia (N1, N2, N3) ;

$MA \leftarrow (N1 + N2 + N3) / 3$ ;

Se  $MA > 7$

Escreva (MA)

Fim Algoritmo.

Algoritmo traduzido para o PASCAL

```
PROGRAM Ifmedia7
```

```
USES
```

```
  CRT;
```

```
VAR
```

```
  N1,N2,N3,MA: REAL;
```

```
BEGIN
```

```
  CLRSCR;
```

```
  WRITELN('Entre com as tres notas do aluno para calcular a sua media');
```

```
  READLN (N1,N2,N3);
```

```
  MA:= (N1+N2+N3) / 3;
```

```
  IF MA > 7.0 THEN
```

```
    BEGIN
```

```
      WRITELN('A media do Aluno e=',MA:2:2);
```

```
    END;
```

```
  READLN;
```

```
END.
```

## 9.2- Seleção Composta

É uma expressão lógica que quando inspecionada, pode gerar um resultado falso ou verdadeiro, executando uma nova seleção.

Estrutura:

Se (condição)

Então

Início

Comando 1;

Comando 2;

.....

Comando n;

Fim Então;

Senão

Início

Comando 3;

Comando 4;

.....

Comando m;

Fim Senão;

Fim Se;

Exemplo:

Calcular o resultado da média aritmética das tres notas do aluno, caso a média for maior que 7, mostrar o resultado “Aprovado” e o valor, caso contrário, “Reprovado” e o seu valor.

Algoritmo:

Real : N1, N2, N3, MA;

Leia (N1, N2, N3) ;

$MA \leftarrow (N1 + N2 + N3) / 3$ ;

```

Se MA > 7;
Escreva (“Aprovado”);
Escreva (MA);
Senão;
Escreva (“Reprovado”);
Escreva (MA);
Fim Se
Fim Algoritmo.

```

Algoritmo traduzido para o PASCAL

```

PROGRAM Ifmedia3;
USES
  CRT;
VAR
  N1,N2,N3,MA: REAL;
BEGIN
  CLRSCR;
  WRITELN('Entre com as tres notas do aluno para calcular a media');
  READLN (N1,N2,N3);
  MA:=( N1+N2+N3) / 3;
  IF MA > 7.0 THEN
    BEGIN
      WRITELN('O Aluno foi Aprovado');
      WRITELN('A sua media e= ',MA:2:2);
    END
  ELSE
    BEGIN
      WRITELN('O Aluno foi Reprovado');
      WRITELN('A sua media e= ',MA:2:2);
    END;
  READLN;
END.

```

### 9.3- Comando Case

É a seleção que permite a escolha de um grupo de seqüência de operações, dependendo do valor seletor.

O valor seletor é comparado com cada valor de uma seqüência de valores, e se algum valor comparado for igual ao seletor, a seqüência de operações relacionadas a ele será executada.

Caso nenhuma igualdade for obtida, nenhuma operação é executada por ele, passando então para o comando seguinte ao CASE.

#### Estrutura:

```

CASE <seletor> of
  <Valor 1> : <seqüência 1>;
  <Valor 2> : <seqüência 2>;
  .....
  <Valor n> : <seqüência n>;
END.

```

Obs. Se existir mais de um comando  
por linha usar BEGIN e END para  
cada grupo de comandos

Exemplo:

Calcular o resultado da média aritmética das três notas do aluno, usando o comando CASE

Escolher entre 3 Opções:

- 1 - Se a media for maior que sete mostrar Aprovado;
- 2 - Se a média estiver entre 5 e 7, mostrar Em Recuperação e a nota que faltou;
- 3 - Se a média for menor que 5 mostrar Reprovado e a nota que faltou.

#### Algoritmo:

```

Real : N1, N2, N3, MA, Difer;
Leia (N1, N2, N3) ;
MA <= (N1+ N2 + N3) / 3;
Escreva ( “ 1-nota maior que 7;    2- nota entre 5 e 7;    3 –Nota menor que 5”);

```

```

Leia "Opção";
Escolha "Opção"
  1 : Escreva ("Aprovado");
  2 : Escreva ("Em Recuperação");
      Difer <= 7-MA;
      Escreva ('O Aluno necessita de Difer pontos);
  3: Escreva ("Reprovado");
      Difer <= 7-MA;
      Escreva ('O Aluno perdeu por Difer pontos);
Fim Algoritmo.

```

### Algoritmo traduzido para o PASCAL

```

PROGRAM CaseMedia3;
USES
  CRT;
VAR
  N1,N2,N3,MA,Difer: REAL;
  OPCA0: INTEGER;
BEGIN
  CLRSCR;
  WRITELN('Entre com as tres notas do aluno para calcular a media');
  READLN (N1,N2,N3);
  MA:= (N1+N2+N3) / 3;
  WRITELN ('A media do Aluno e = ',MA:2:2);
  WRITELN("");
  WRITELN ('Escolha entre as opcoes:');
  WRITELN ('1 : somente se a media for maior que 7');
  WRITELN ('2 : somente se a media for menor que 7 e maior que 5');
  WRITELN ('3 : somente se a media for menor que 5');
  READLN (OPCA0);
  WRITELN("");
  CASE OPCA0 OF
    1: BEGIN
      WRITELN ('Aprovado');
    END;
    2: BEGIN
      WRITELN ('Em Recuperação');
      Difer:= 7.0-MA;
      WRITELN ('O Aluno necessita de= ', Difer:2:2,' pontos');
    END;
    3: BEGIN
      WRITELN ('Reprovado');
      Difer:= 7.0-MA;
      WRITELN ('O Aluno perdeu por= ', Difer:2:2,' pontos');
    END;
  END;
  READLN;
END.

```

## 10- ESTRUTURAS DE REPETIÇÃO

As estruturas de repetição são usadas para conjunto de ações a serem executadas repetidas vezes, existem três tipos básicos de repetição no PASCAL:

“WHILE”, “FOR” e “REPEAT”

### 10.1 - Comando While

O Comando While é usado para repetir “n” vezes uma seqüência de instruções, sendo este conhecido previamente .

Antes de executar a seqüência a condição é verificada. Se a condição não for atendida mesmo na primeira vez, a seqüência não é acessada.



A seqüência será executada até a condição se tornar falsa, logo a sua condição deverá ser alterada, em um determinado instante, dentro da seqüência.

-  
**Estrutura:**  
WHILE <condição> DO  
BEGIN  
    <seqüência de operações>  
END.

Exemplo:  
Calcular a média de um grupo de números, enquanto não entrar com um número de valor zero.

-  
**Algoritmo:**  
Real : N1, Soma, Media;  
Inteiro: Contador;  
Soma = Zero  
Contador <= -1  
N1 <= 1  
Enquanto N1 <> de Zero  
Faça  
Leia (N1);  
Soma <= N1+ Soma  
Contador <= Contador +1  
Fim Enquanto;  
Escreva ( A media dos Contador números é igual a Media);  
Fim do Programa.

#### **Algoritmo traduzido para o PASCAL**

```
PROGRAM WhileMedia;
USES
  CRT;
VAR
  N1,Soma,Media: REAL;
  Contador: INTEGER;
BEGIN
  CLRSCR;
  Soma:=0;
  Media:=0;
  Contador:=-1;
  N1:=1;
  WHILE ( N1 <> 0) DO
    BEGIN
      WRITELN('Entre o valor para se calcular a media dos numeros');
      READLN (N1);
      Soma:= Soma + N1;
      Contador:= Contador + 1;
    END;
  Media:= Soma / Contador;
  WRITELN('A media dos ',Contador,' numeros e= ',Media:2:2);
  READLN;
END.
```

#### **10.2- Comando FOR**

Este comando é utilizado quando queremos realizar uma operação durante um número fixo e conhecido de vezes.

-  
**Estrutura:**  
FOR <var>:= <limite inferior> TO <limite superior> DO  
BEGIN  
    <seqüência de operações>  
END

Ou  
FOR <var>:= <limite superior > DOWTO <limite inferior > DO  
BEGIN  
    <seqüência de operações>  
END

Se fizermos a seguinte variação:

```
For I = 1 to 5 Do
  Begin
    <seqüência de operações>;
  End.
```

As operações serão feitas em um intervalo de 1 até 5 sendo incrementada de 1 em 1.

Se fizermos a seguinte variação:

```
For I = 5 DOWTO 1 Do
  Begin
    <seqüência de operações>;
  End.
```

As operações serão feitas em um intervalo de 5 até 1 sendo decrementada de 1 em 1.

Exemplo:

Calcular a média de um grupo de números, sendo informado a quantidade de números para ser tirado a média, e os seus valores, usando o comando FOR.

#### Algoritmo:

```
Real : Numero, Soma, Media;
Inteiro: I, Quantidade;
Soma <= Zero;
Leia (Quantidade);
Para I = 1 ate Quantidade
Faça;
Leia (Numero);
Soma <= Numero + Soma
Fim Para;
Media <= Soma / Quantidade
Escreva ( A media dos Quantidade números é igual a Media);
Fim do Programa.
```

#### Algoritmo traduzido para o PASCAL

```
PROGRAM ForMedia;
USES
  CRT;
VAR
  Numero,Soma,Media: REAL;
  I, Quantidade: INTEGER;
BEGIN
  CLRSCR;
  Soma:=0;
  WRITELN('Entre com a quantidade de numeros para ser tirado a media');
  READLN(Quantidade);
  FOR I:= 1 TO Quantidade DO
    BEGIN
      WRITELN ('Entre com o valor do ',I,'o numero');
      READLN(Numero);
      Soma:=Soma+Numero;
    END;
  Media:= Soma / Quantidade;
  WRITELN('A media dos ',Quantidade,' numeros e= ',Media:2:2);
  READLN;
END.
```

### 10.3-Comando REPEAT / UNTIL

É usado quando não conhecemos previamente o número de vezes que uma seqüência deve ser repetida. Neste comando a condição é executada pelo menos uma vez, pois a condição fica após a seqüência de operações.

#### Estrutura:

```
REPEAT
  <seqüência de operações>
UNTIL <condição>
```

Exemplo:

Calcular a soma dos números de uma seqüência linear determinada pelo usuário.

#### Algoritmo:

```
Inteiro: Numero, Soma;
Soma <= 0;
Leia (Numero);
Repita
Soma <= Numero + Soma
Numero <= Numero -1
Até o Numero ficar igual a zero
Escreva Soma
Fim do Programa
```

#### Algoritmo traduzido para o PASCAL

```
PROGRAM Re soma;
USES
  CRT;
VAR
  Numero,Soma:INTEGER;

BEGIN
  CLRSCR;
  Soma:=0;
  WRITELN('Entre com o ultimo numero da sequencia');
  READLN (Numero);
  REPEAT
    Soma:= Soma+Numero;
    Numero:= Numero-1;
  UNTIL (Numero=0);
  WRITELN ('A Soma desta sequencia e ', Soma);
  READLN;
END.
```

## 11- VARIÁVEIS COMPOSTAS

### 11.1 – Variável Composta Homogênea

A variável composta homogênea corresponde a um conjunto de dados de um mesmo tipo, onde seus elementos são identificados por um índice, podendo ser unidimensionais (vetor) ou multidimensionais (matriz).

a)Vetor:

#### Estrutura:

```
ARRAY[i..j] OF tipo;
```

Obs: Podemos trabalhar no Pascal definindo variáveis de um mesmo tipo, usando o comando Type em um programa, devemos declarar primeiramente o seu tipo. Esta declaração deve ser feita no dicionário de dados antes de declarar as variáveis.

Exemplo da declaração de tipo:

Type

```
Tipo_Vetor= Array [1..10] of Integer;
```

Var

Vet1,Vet2,Vet3: Tipo\_Vetor

Exemplo de programa usando Vetor:

Calcular a média de uma turma de N alunos, limitando a 50 alunos, (usando vetor) , onde o índice representa o número do aluno, e o elemento a nota de cada aluno respectivamente. Entre com os dados do vetor do exemplo a.

Algoritmo:

Real: Soma, Média;

Inteiro: I, Nalunos;

Vetor: Vetalunos de 1 a 50 elementos reais;

Leia(Nalunos);

Soma <= 0;

Para I = 1 ate Nalunos;

Faça;

Leia (Vetalunos[I]);

Soma <= Soma + Vatalunos[I];

Fim Para;

Media <= Soma/Nalunos;

Escreva ('Média das notas: ',Media);

Fim do Programa.

Algoritmo traduzido para o PASCAL

PROGRAM VetMedia;

{Este programa calcula a media de N alunos usando vetor}

USES

CRT;

VAR

Soma,Media: REAL;

Nalunos,I: INTEGER;

Vetalunos: Array [1..50] of Real;

BEGIN

CLRSCR;

Soma:=0;

WRITELN('Entre com a quantidade de alunos');

READLN(Nalunos);

FOR I:= 1 TO Nalunos DO

BEGIN

WRITELN ('Entre com a nota do ',I,' aluno');

READLN (Vetalunos[I]);

Soma:=Soma+Vetalunos[I];

END;

Media:= Soma / Nalunos;

WRITELN('A media dos ',Nalunos,' Alunos e= ',Media:2:2);

READLN;

END.

b)Matrizes

Estrutura:

ARRAY[i..j, k..y] OF tipo;

Exemplo:

Faça um programa que leia as Notas de N alunos da matéria A, leia também as Notas dos mesmos alunos da matéria B e C, e mostre a média entre as três matérias de cada aluno, usando matriz para as alunos e suas notas e vetor para calcular a média das notas de um mesmo aluno.

Algoritmo:

Real: Nota;

Inteiro: I,J, Nalunos;

Matriz: Matnota de 1 a 50 e 1 a 3 elementos reais;

Vetor: Vetmedia de 1 a 50 elementos reais;

Leia(Nalunos)

Para I = 1 ate Nalunos;

Faça;

```

VetMedia[I] <= 0;
Para J= 1 ate 3;
Faça
Leia (MatNota[I,J];
VetMedia<= Vetmedia[I]+Matnota[I,J];
Fim Para;
Fim Para;
Para I = 1 ate Nalunos;
Faça
Escreva ('Média das notas: ',Vetmedia[I]/3);
Fim Para
Fim do Programa.

```

### Algoritmo traduzido para o PASCAL

```

PROGRAM Matrizno;
USES
  CRT;
VAR
  Nota: REAL;
  Nalunos,I,J: INTEGER;
  MatNota: Array [1..50, 1..3] of Real;
  VetMedia: Array [1..50] of Real;
BEGIN
  CLRSCR;
  WRITELN('Entre com a quantidade de alunos');
  READLN(Nalunos);
  FOR I:= 1 TO Nalunos DO
    BEGIN
      VETMedia[I]:=0;
      FOR J:= 1 TO 3 DO
        BEGIN
          WRITELN ('Entre com a nota do ',I,' aluno da materia ',J);
          READLN (MatNota[I,J]);
          VetMedia[I]:= VetMedia[I]+MatNota[I,J];
        END;
      END;
    WRITELN('As medias dos ',Nalunos,' Alunos sao:');
    FOR I:= 1 TO Nalunos DO
      BEGIN
        Writeln('A Media do ',I,' Aluno e ',VetMedia[I]/3:2:2);
      END;
    READLN;
  END.

```

## 12- MODULARIZAÇÃO DO ALGORITMO

Para o ser humano torna-se difícil descrevermos as instruções mais detalhas de forma que consigamos memorizar várias instruções complexas ao mesmo tempo, mas para o computador as instruções devem ser detalhas a um nível que o computador seja capaz de interpreta-las e resolve-las de maneira adequada.

A solução para o homem resolver este tipo de problema é dividir um problema complexo em vários outros subproblemas elementares, de tal forma que o homem consiga simila-los com maior facilidade.

### Estrutura:

```

Procedure nome;
declaração dos objetos locais à procedure
BEGIN
comandos da procedure
END;

```

As variáveis de um programa podem ser transferidas a um sub-programa por duas maneiras:

- Por Referência



```
FUNCTION nome: tipo
declararão dos objetos locais à function
BEGIN
comandos da function
. . .
nome := X;
END;
```

**Exemplo: Este programa calcula o fatorial de um numero, usando função**

```
PROGRAM FuncFat;
USES
  CRT;
VAR
  Numero : Integer;
FUNCTION Fatorial:Integer;
VAR
  I,Acumula:Integer;
BEGIN
  Acumula := 1;
  I := 1;
  WHILE (I <= Numero) DO
    BEGIN
      Acumula := Acumula * I;
      I := I + 1;
    END;
  Fatorial := Acumula;
END;

BEGIN
  CLRSCR;
  WRITELN ('Digite o valor do Numero: ');
  READLN(Numero);
  WRITELN ('O Fatorial de ',NUMERO , ' e = ', Fatorial);
  READLN;
END.
```

## REFERENCIA BIBLIOGRÁFICA

- GIMARÃES/LAJES, ALGORITMOS E ESTRUTURAS DE DADOS – 16ª TIRAGEM, LIVROS TÉCNICOS E CIENTÍFICOS EDITORA;
- APOSTILA CONSTRUÇÃO DE ALGORITMOS, CURSO DE FORMAÇÃO DE PROGRAMADOR DE MICROCOMPUTADOR, SENAC /ES – 1994.
- APOSTILA DE EXERCÍCIOS DE ALGORITMO, CURSO DE FORMAÇÃO DE PROGRAMADOR DE MICROCOMPUTADOR, SENAC /ES – 1994.
- APOSTILA ALGORITMOS, PROFª. TÂNIA, CURSO DE PÓS-GRADUAÇÃO EM ANÁLISE DE SISTEMAS UVV/ES – 1999.
- EXERCÍCIOS ALGORITMOS, PROFª. TÂNIA CURSO DE PÓS-GRADUAÇÃO EM ANÁLISE DE SISTEMAS UVV/ES – 1999;
- APOSTILA DE ALGORITMO DISPONÍVEL NA INTERNET, <http://200.241.216.16/reenge/pdi/sumario.html>, CONSULTADA EM 17/04/199;
- APOSTILA DE ALGORITMO DISPONÍVEL NA INTERNET, <http://www.angelfire.com/bc/fontini/algoritm.html>, CONSULTADA EM 20/06/1999.