

## **Introdução**

Para ser um bom programador de computadores, independentemente da linguagem de programação, o estudante tem que adquirir uma boa base em lógica de programação, em especial, lógica algorítmica.

O conteúdo de lógica de programação é oferecido nos cursos de: tecnologia de com ênfase em informática, engenharias, ciência da computação, física, matemática, geologia, estatística, enfim a qualquer curso que vá utilizar o computador como uma ferramenta para otimizar seus processos.

Um dos objetivos deste material é permitir que o estudante desenvolva, gradual e progressivamente algoritmos eficientes para as suas necessidades acadêmicas. O outro objetivo é mostrar ao aluno que um bom entendimento da linguagem algorítmica propiciará um bom recurso para escolher uma linguagem de programação que poderá ser utilizada na sua vida profissional.

Neste material foi utilizada uma linguagem(sintaxe) simples, clara e objetiva para aproximar os termos (jargões) da informática à nossa linguagem coloquial. Para isso foram introduzidos alguns conceitos iniciais com intuito de preparar o aluno no uso simbologia e os termos adequados ao tema .

O material foi estruturado na seguinte ordem: uma abordagem simples e direta dos conceitos, exemplos, uma carga de exercícios resolvidos e uma outra carga de exercícios propostos.

Para um bom entendimento dos conceitos é importante que o aluno tenha paciência e perseverança, é importante que tenha também o compilador Pascal para implementar os algoritmos sugeridos. Os resultados só virão após algum tempo de dedicação ao assunto. É recomendável que o aluno só avance para o próximo item quando as dúvidas forem totalmente sanadas. Lembro, também que a consulta a outras fontes faz parte de um melhor aprendizado.

Aproveite bem o material.  
Prof. Msc. Carlos Alberto Bezerra e Silva.

"A arte de programar consiste na arte de organizar e dominar a complexidade. (Dijkstra)"

## **Fundamentos**

### ***Algoritmo***

-Conjunto pré-determinado de regras ou métodos bem definidos com o propósito de resolver problemas com um número finitos de procedimentos.

-Definição rigorosa de uma série de operações a serem efetuadas para obtermos um resultado desejável em um número definido de etapas.

-Uma seqüência lógica de instruções passo a passo destinados à soluções de um determinado problema.

### ***Programa***

-Conjunto de instruções codificadas, em linguagem específica para computadores(máquinas), que controlarão o computador(máquina) no percurso da execução de uma operação ou de uma série de operações desejáveis para a resolução de um problema definido.

#### **Exemplo:**

Uma I.E.S. deseja calcular a média dos seus alunos utilizando como critérios: peso 2 para a primeira avaliação e peso 3 para a segunda avaliação.

### ***Linguagem de programação***

-Linguagem utilizada para transcrever o algoritmo em programa, ou seja, um código inteligível e executável por máquina(computador).

#### **Exemplo:**

Pascal, Algol, Basic, C, C++, Fortran, Java, Delphi, Visual Basic, Cobol etc.

## ***Linguagem Algorítmica***

-É uma Pseudolinguagem que é utilizada para o ensino nas linguagens de programação.

### **Definição da linguagem algorítmica**

-A simbologia que será descrita nos próximos tópicos será utilizada nos algoritmos e em algumas linguagens de programação.

### **- Os dois pontos :**

-Utilizado para declararmos variáveis.

#### **Exemplo:**

```
Nota:Numérica.  
Salario:Numérica  
Nome:Caracteres.
```

### **- O ponto e vírgula ;**

-Utilizado para finalizarmos uma instrução(sentença).

#### **Exemplo:**

```
Escreva(x);  
Leia(y);
```

### **- A vírgula ,**

-Utilizada para separarmos variáveis e constantes.

#### **Exemplo**

```
Nome, Idade;  
Salario, HoraExtra;
```

## - O Ponto .

-É utilizado para finalizar o algoritmo.

### Exemplo

```
Início
...
Fim.
```

- Também é utilizado para separar a parte inteira da parte decimal de um número.

### Exemplo

```
7.5, 324.56
```

## As aspas simples(apóstrofo) ' '

-Utilizada para separarmos conteúdos literais.

### Exemplo

```
'Segunda', 'Azul', 'Josineura'
```

## Chaves { }

- Utilizado para aumentar a clareza do algoritmo, devem aparecer entre chaves { }. Os comentários não serão processados pelo algoritmo.

### Exemplo

```
{ Este programa foi criado no dia 10/03/2008 por Carlos Alberto}
Var  Cód,           {Código do curso}
      Nota,         {Nota do Aluno}
      Mat:Numérico; {Matrícula do aluno}
```

## Dois pontos igual :=

- Utilizado para atribuímos um valor a uma variável

### Exemplo:

Idade:=32; Sexo:='M'; Peso:=90.  
Media=(A1+A2)/5;

### Constantes =

- As constantes armazenam valores fixos, que não se alteram durante a execução do algoritmo. Os conjuntos podem assumir valores: numéricos, lógicos e literais, são representadas por identificadores e recebem seus valores por meio do símbolo da igualdade.

Identificador\_de\_constante=constante

### Exemplo:

Ano=2008; Filhos=VERDADEIRO; Bolsa\_Estudo=Falso.

### Identificadores

- É formado por uma letra ou uma letra seguida de outras ou dígitos. Não permite-se o uso de espaços ou qualquer outro caractere que não seja letra ou dígito.

### Exemplo

A, X15, Matricula.  
X-Y, A:B, 5B são **inválidos**.

### Palavras reservadas

- São identificadores que só podem ser usados com um sentido prefixado na linguagem. Também são conhecidos como palavras chaves. As palavras abaixo serão consideradas reservadas neste material.

E	VETOR	INÍCIO	CASO
CONST	DIV	FAÇA	ATÉ
SE NÃO	FIM	ARQUIVO	PARA
FUNÇÃO	IR PARA	SE	ALOQUE
ETIQUETA	MOD	NULO	NÃO
OU	PROCEDIEMNTO	PROGRAMA	DE
REGISTRO	REPITA	CONJUNTO	ENTÃO
TIPO	VAR	ENQUANTO	COM
FECHAR	ABRIR	ASSINALAR	NIL

### Constantes numéricas.

- São formadas por seqüências de dígitos que podem ou não ser precedidos por um sinal positivo(+) ou negativo(-). Eles também podem ser seguidos por um ponto(.) decimal e outras seqüência de dígitos

### Exemplo

```
25, +3254, 3.1415, 2.78, -0.5.
```

### Constantes lógicas

- São duas e representa-se pelas palavras VERDADEIRO e FALSO. Também são chamadas por booleans.

### Constantes literais

- As constantes literais são representadas pelos caracteres (letras, dígitos ou símbolos) colocados entre apóstrofos('')

### Exemplo

```
'Televisão', '*****', 'RA9620', '07/03/2008', 'H2O'.
```

### Variáveis

-São representadas por identificadores e podem ser numéricas, lógicas e literais. Ao contrário das constantes as variáveis armazenam valores que podem ser alteradas durante a execução do algoritmo.

- Associam um endereço de memória a um nome.

### Utilizando a linguagem algorítmica

Os algoritmos devem ser iniciados pela palavra reservadas **Algoritmo** seguida de um título, este de ponto e vírgula(;) para indicar o final da linha. Logo abaixo devemos adicionar as **Declarações** das **constantes** e das **variáveis** e na seqüência vem o **bloco** de **instruções** ou **comandos**.

### Exemplo

```
Algoritmo Título_do_algoritmo;  
Declarações  
Bloco.
```

O bloco de instruções é formado por comandos entre as palavras reservadas **Início** e **Fim**

## Exemplo

```
Algoritmo Título_do_algoritmo;  
Declarações  
Início  
    Comandos;  
Fim.
```

As variáveis utilizadas nos algoritmos são declaradas, isto é, são identificadas com um nome e indicam os tipos de dados que elas armazenam. As declarações são usadas para validar o uso de qualquer identificador que não seja pré-definido, indicando-se as características da variável que será representada. Todas as variáveis declaradas num bloco, devem ser incluídas numa única declaração da forma:

## Exemplo

```
Var Lista_de_identificadores: Tipo;  
    Lista_de_identificadores: Tipo;  
    ...
```

Onde:

Lista\_de\_identificadores Representa o Identificador.

**Tipo** Tipo de variavel.

Tipos pré-definidos da LINGUAGEM ALGORÍTMICA:

NUMÉRICO, LÓGICO, LITERAL

A utilização do tipo NUMÉRICO se dará quando quisermos associar o conteúdo da variável a números Inteiros ou Reais.

A utilização do tipo LÓGICO se dará quando quisermos associar o conteúdo da variável a uma situação VERDADEIRA OU FALSO.

A utilização do tipo LITERAL se dará quando quisermos associar o conteúdo da variável a uma cadeia de caracteres.

## Exemplo

```
Var SALARIO, X5: Numérico;  
    CONTADOR, X5: Numérico;  
    TESTE, FLAG: Lógico;  
    NOME, FONE: Literal;
```

## Declaração de tipos:

Na linguagem algorítmica utilizada neste material o aluno poderá criar novos tipos de dados além daqueles considerados pré-definidos. Um novo tipo é criado através de uma definição que determina um conjunto de valores e os associa a um identificador este conjunto. A declaração de tipo precede a declaração de variáveis.

```
TIPO Identificador_de_tipo= descrição;  
      Identificador_de_tipo= descrição;  
      Identificador_de_tipo= descrição;  
      ...
```

Onde TIPO é a palavra chave que inicia a declaração de tipos.

Neste material também será possível criar um tipo formado por elementos consecutivos de um tipo pré-definido.

```
TIPO Identificador_de_tipo= Intervalo;  
      Identificador_de_tipo= Intervalo;  
      Identificador_de_tipo= Intervalo;  
      ...
```

Onde TIPO é a palavra chave e Intervalo representa o tamanho do identificador

```
Var  Lista_de_identificadores: Tipo;  
      Lista_de_identificadores: Tipo;  
      ...
```

## Exemplo

```
TIPO Meses = 1..12;  
Var  Mês : Meses;
```

Neste caso a variável Mês é do tipo Meses assumindo valores na faixa de 1..12.

## Expressões Aritméticas:

- O sinal da multiplicação é um asterisco: \*
- O sinal da divisão é uma barra: /
- O quociente inteiro de uma divisão com operandos inteiros será calculado com o operador: **DIV**
- O resto de uma divisão inteira será calculado pelo operador **MOD**.
- A exponenciação será indicada pelo circunflexo: **^**
- A subtração é representada por -



## Exemplos

9 / 2 é igual a 4.5  
9 **DIV** 2 é igual a 4  
9 **MOD** 2 é igual a 1

Quanto a **prioridade** observe a tabela a seguinte.

Prioridade	Operadores
1ª	* / <b>DIV MOD</b>
2ª	+ -

## Exemplos

X + Y; TOTAL / N; A \* B + C; (NOTA1 \* 2 + NOTA2 \*3) / 5.

## Funções numéricas pré-definidas:

Nome	Resultado da Função	Tipo
SQRT(x)	Raiz quadrada de x	Real ou inteiro
SQR(x)	Quadrado de x	Real ou inteiro
ABS(x)	Valor absoluto de x	Real ou inteiro
INT(x)	Retorna a parte inteira de x	Inteiro

## Expressões lógicas:

- As expressões lógicas comparam valores utilizando um dos operadores

=	Igual	< >	Diferente
<=	Menor ou igual	<	Menor
>=	Maior ou igual	>	Maior

O resultado das expressões lógicas serão sempre um valor lógico VERDADEIRO ou FALSO.

Obs. A expressão A + B = C será verdadeira ou falsa dependendo do conteúdo do lado esquerdo da expressão A +B podendo ser igual ou diferente a C.

## Exemplo

X<>Y      Nome='Jeguinaldo'

$A * B - C < 0$  Ano=2008

**Exercício.** Com as declarações abaixo preencha a tabela com V de verdadeiro e F de falso.

**Var** X,Y:Numérico;  
Nome,Sexo:Literal;

X	Y	Nome	Sex	$X+Y \geq \text{SQRT}(Y)$	$\text{Nome} \neq \text{'Ordicléia'}$	$\text{Sexo} = \text{'F'}$
4,0	20	'joão'	'M'			
6,0	40	'Mario'	'M'			
1,5	7	'Ordicléia'	'F'			
6,0	2	'Ana'	'F'			
5,5	7	'Jeguinaldo'	'M'			

### Conjunção/Disjunção optativa OU

- Duas ou mais expressões unidas pelo operador OU formam uma sentença verdadeira se pelo menos uma delas for verdadeira.

#### Exemplo

$5 \geq 5$  OU  $3 < 8$  é VERDADEIRA  
 $7 = 7$  OU  $1 < 10$  é VERDADEIRA  
 $3 \leq 1$  OU  $5 = 8$  é FALSO

### Advérbio de negação NÃO

- Retorna o valor lógico oposto ao da expressão

#### Exemplo

$\text{Não}(10 > 8)$  é FALSO  
 $\text{Não}(3 = 5)$  é VERDADEIRA

### Tabela de prioridade das operações

<b>Prioridade</b>	<b>Operadores</b>
1	NOT
2	*, /, DIV, MOD, AND
3	+, -, OR
4	=, <, >, <=, >=

Pode-se usar diversos níveis de parentêses para dar uma ordem de execução às expressões distintas da tabela acima.

### **Comandos de entrada e saída.**

- O comando de **entrada(input)** que será utilizada na nossa linguagem algorítmica tem a seguinte sintaxe:

**Leia**(Lista\_de\_identificadores);

Onde,

Leia: Lê o valor da variável ou variáveis de uma unidade de entrada de dados como teclado por exemplo.

Lista\_de\_identificadores: Nome das variáveis.

O comando de **saída(output)** da nossa linguagem algorítmica tem a seguinte forma:

**Escreva**(Identificador e/ou constantes e/ou expressões)

Onde,

Escreva: Escreve o valor da variável ou variáveis de uma unidade de saída de dados como monitor por exemplo

Lista\_de\_identificadores: Nome das variáveis.

Constantes: Constantes literais entre apóstrofos(' ')

Expressões: Expressões entre vírgulas( , )

### **Estrutura seqüencial**

- Os algoritmos que escreveremos terão que iniciar com a palavra reservada ALGORITMO seguido de um título e um ponto e vírgula( ; ). Em seguida temos a área de declarações que compreende as instruções que se referem as variáveis e as constantes que serão utilizadas no algoritmo. É importante lembrar que as tais variáveis e as constantes não podem ser declaradas em outro lugar do algoritmo. Após as declarações vem o bloco de instruções que devem iniciar com as palavras INÍCIO seguido das instruções do algoritmo finalizados com ponto e vírgula( ; ) finalizando com a palavra FIM

## **Linguagem algorítmica Fluxograma**

**Algoritmo** título;

Declarações

**Var** Identificadores: Tipo;

**Início**

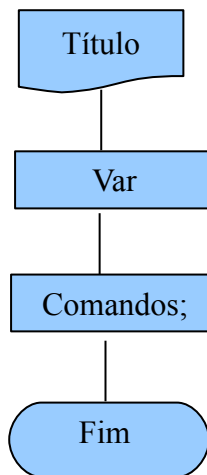
Comando 1;

Comando 2;

...

Comando n;

**Fim.**



Entre as palavras **Início** e **Fim** devem aparecer as instruções para que os dados sejam lidos através do dispositivo de entrada(teclado), processados com as devidas instruções, atribuições, expressões etc e em seguida exibidos no dispositivo de saída(monitor).

**Exemplo.** Algoritmo que lê dois números e calcula a média aritmética entre eles e a direita a codificação na linguagem Pascal

Algoritmo	Codificação na linguagem Pascal
<b>Algoritmo</b> Média;	<b>Program</b> Média;
<b>Var</b> A1,A2,Med:Numérica;	<b>Var</b> A1,A2,Med:Real;
<b>Início</b>	<b>Begin</b>
<b>Escreva</b> ('Digite o primeiro número');	<b>Write</b> ('Digite o primeiro número');
<b>Leia</b> (A1);	<b>Read</b> (A1);
<b>Escreva</b> ('Digite o segundo número');	<b>Write</b> ('Digite o segundo número');
<b>Leia</b> (A2);	<b>Read</b> (A2);
Med:=(A1+A2)/2;	Med:=(A1+A2)/2;
<b>Escreva</b> ('A média entre os números é',Med);	<b>Write</b> ('A média entre os números é',Med);
<b>Fim.</b>	<b>End.</b>

## Exercícios resolvidos

01 – Escreva um algoritmo para calcular e exibir a média de três números quaisquer.

```
Algoritmo Media_de_três_Num;  
  
Var X, Y, Z, Media:Numérico;  
  
Início  
    Escreva('Digite três números');  
    Leia(X, Y, Z);  
    Escreva('A média é:'+Media);  
Fim.
```

02 – Escreva um algoritmo que realize o cálculo do salário de um funcionário. O salário é baseado no número de salários mínimos, Leia o salário mínimo vigente e o número de salários recebido.

```
Algoritmo Salarios;  
  
Var Sal_Min, Num_Sal, Salario:Numérico;  
  
Início  
    Escreva('Digite o valor do salário mínimo □ R$ ');  
    Leia(Sal_Min);  
    Escreva('Digite o número de salários é □ ');  
    Leia(Num_Sal);  
    Salário:=Sal_Min*Num_Sal;  
    Escreva('O salário do funcionário é □ ', Salario);  
Fim.
```

03 – Escreva um algoritmo que calcule o preço de uma mercadoria dados o seu peso por Quilo.

```
Algoritmo Preço_Produto;  
  
Var Preço, Preço_Quilo, Peso:Numérico;  
  
Início  
    Escreva('Digite o preço por quilo □ R$ ');  
    Leia(Preço_Quilo);  
    Escreva('Digite o peso do produto □ ');  
    Leia(Peso); Preço:=Peso * Preço_Quilo;  
    Salário:=Sal_Min*Num_Sal;  
    Escreva('O preço do produto é □ ', Preço);  
Fim.
```

04 – Imagine um carro de tanque cheio. Escreva um algoritmo para calcular o consumo médio de combustível do carro. Leia a capacidade máxima do tanque quantos Km são percorridos usando todo o combustível.

```
Algoritmo consumo_médio;  
  
Var consumo, km, capacidade:Numérico;  
  
Início  
    Escreva('Qual a capacidade do tanque em Litros □ ');  
    Leia(capacidade);  
    Escreva('Quantidade de quilômetros percorridos □ ');  
    Leia(km);    consumo:=km / capacidade;  
    Escreva('O consumo foi de', consumo, 'km por litro !');  
Fim.
```

### **Estrutura condicional.**

- As estruturas condicionais alteram a execução do algoritmo impondo condições para que determinadas linhas sejam ou não executadas. A estrutura condicional mais comum é o “**Se**”, que pode aparecer sob duas formas.

#### **Sem desvio.**

```
Se Condição Então  
    Comando;
```

O comando só será executado se a condição for verdadeira. A condição é uma expressão lógica.

A comando pode ser simples ou composto. Considere composto aquele formado por diversos comandos delimitados pelas palavras Início e Fim.

```
Se Condição Então  
    Início  
        Comando;  
        Comando;  
        Comando;  
    Fim;
```

### **Estrutura com desvio.**

```
Se Condição Então  
    Comando 1  
Se não  
    Comando 2;
```

Nessa estrutura o algoritmo executará o comando 1 se a condição for verdadeira e executará o comando 2 se a mesma for falsa. Os comandos 1 e 2 também podem ser compostos.

### **Exercícios resolvidos**

5 – Faça um algoritmo que calcule a média das duas notas de um aluno e imprima a mensagem “Aprovado” se a média for maior ou igual a 6,0(seis) e se imprima “Retido” em caso contrário.

```
Algoritmo Média_Aluno;  
  
Var Média, nota1, Nota2:Numérico;  
  
Início  
    Escreva('Digite a nota 1 □ ');  
    Leia(Nota1);  
    Escreva('Digite a nota 2 □ ');  
    Leia(Nota2);  
    Média:=(Nota1+Nota2)/2;  
    Se (Média>=6) Então  
        Escreva('Aprovado !');  
    Se não  
        Escreva('Retido !');  
Fim.
```

6 – Faça um algoritmo que imprima uma das mensagens “Número par” ou “Número Ímpar”.

```
Algoritmo Par_Impar;  
  
Var Número:Numérico;  
  
Início  
    Escreva('Digite um número: ');  
    Leia(Número);  
    Se (Número MOD 2) Então
```

```

        Escreva('O número é par!');
Se não
        Escreva('o número é ímpar!');
Fim.

```

7 – faça um algoritmo que leia o salário de uma pessoa e calcule o imposto de renda a ser pago obedecendo a seguinte tabela.

Salário	Imposto
Até R\$ 1.372,81	<i>Isento</i>
Entre R\$ 1.372,82 e R\$ 2.743,25	0,15
Acima de R\$ 2.743,25	27,5

```

Algoritmo Imposto_Salário;

Var Salário:Numérico;

Início
    Escreva('Digite o valor do salário □ ');
    Leia(Salário);
    Se (Salário < 1372.81) Então
        Escreva('Isento de imposto!');
    Se não
        Se (Salário < 1372.82) e (Salário < 2743.25) Então
            Escreva('O imposto é de ',Salário * 0.15);
        Se não
            Escreva('O imposto é de ',Salário * 0.275);
Fim.

```

8 – Escreva um algoritmo que leia três valores numéricos e mostre o menor deles.

```

Algoritmo Menor_três;

Var N1,N2,N3, Menor:Numérico;

Início

```



```

Escreva('Digite o primeiro valor □ ');
Leia(N1);
Escreva('Digite o segundo valor □ ');
Leia(N2);
Escreva('Digite o terceiro valor □ ');
Leia(N3);
Se (N1 < N2) E (N1 < N3) Então
    Menor:=N1
Se não
    Se (N2 < N3) Então
        Menor:=N2;
    Se não
        Menor:=N3;
    Escreva('O menor é',Menor);

```

**Fim.**

9 – Escreva um algoritmo que receba três valores e informe se os mesmos representam os lados de um triângulo ou não. Caso seja, classifique-os quanto aos lados. Equilátero(três lados iguais), isósceles(dois lados iguais) e escaleno(todos os lados distintos)

**Algoritmo** triângulo;

**Var** x,y,z:Numérico;

**Início**

```

Escreva('Digite o primeiro valor □ ');
Leia(x);
Escreva('Digite o segundo valor □ ');
Leia(y);
Escreva('Digite o terceiro valor □ ');
Leia(z);
Se (x < y + z) e (y < x + z) E (z < x + y) Então
    Se (x = y) E (x = z) Então
        Escreva('O triângulo é equilátero')
    Se não
        Se (x = y) OU (x = z) OU (y = z) Então
            Escreva('O triângulo é isósceles')
    Se não
        Escreva('O triângulo é escaleno');
Se não
    Escreva('Os valores não representam os lados de um triângulo');

```

**Fim.**

10 – Faça um algoritmo que ordene três números fornecidos por uma usuário.

**Algoritmo** números\_crescentes;

**Var** x,y,z,aux:Numérico;

**Início**

**Escreva**('Digite o primeiro valor □ ');

**Leia**(x);

**Escreva**('Digite o segundo valor □ ');

**Leia**(y);

**Escreva**('Digite o terceiro valor □ ');

**Leia**(z);

**Se** (x > y) **OU** (x > z) **Então**

**Se** (y < z) **Então**

**Início**

aux:=x;

x:=y;

y:=aux;

**Fim**

**Se não**

**Início**

aux:=x;

x:=z;

z:=aux;

**Fim;**

**Se** ( y > z ) **Então**

**Início**

aux:=y;

y:=z;

z:=aux;

**Fim;**

**Escreva**(x , y , z );

**Fim.**

### **Estrutura de repetição (Segunda parte)**

- A estrutura de repetição em um algoritmo consiste na execução de um trecho do mesmo levando em consideração certas condições imposta pelo algoritmo.

Temos duas três formas de estruturas de repetição.

- A primeira forma repete uma seqüência de instruções indeterminada de vezes, até que uma condição seja satisfeita.

### Sintaxe

<b>Enquanto</b> condição <b>Faça</b> Comando 1
---

**Explicação:** Isso significa que a condição estabelecida executará o Comando 1 enquanto ela for verdadeira. Pode-se adicionar **Comando 2**, **Comando 3** etc para tornando esta estrutura composta.

**Exemplo.** Calcule a soma dos números pares de 50 a 100

<b>Algoritmo</b> Soma_Pares; <b>Var</b> soma,par:Numérico;  <b>Início</b> soma:=0; par:=50; <b>Enquanto</b> par<=100 <b>Faça</b> soma:=soma+par; par:=par+2; <b>Fim;</b> <b>Escreva</b> (soma); <b>Fim.</b>
--

- A segunda estrutura é utilizada para o mesmo fim. Fazemos uso dela quando queremos que o algoritmo execute uma quantidade de vezes pré-determinada de um trecho do algoritmo.

### Sintaxe

<b>Para</b> variável_que_controla_repetição:=valor_inicial <b>Até</b> valor_final <b>Faça</b> Comando 1;
---

**Explicação:** Essa estrutura controlará o fluxo do algoritmo mediante o conteúdo informado nas variáveis **variável\_que\_controla\_repetição:=valor\_inicial** e **valor\_final**. A informação deste conteúdos indicará quando o algoritmo será finalizado.

**Exemplo.** Calcule a soma dos números pares de 50 a 100

```

Algoritmo Soma_Pares;
Var soma,n:Numérico;
Início
    soma:=0;
    Para n:=25 Até 50 Faça
        soma:=soma+2*n;
    Escreva(soma);
Fim.

```

- A terceira estrutura é utilizada quando desejamos que o algoritmo execute um determinado trecho pelo menos uma vez.

### Sintaxe

```

Repita

    Comando 1
    Comando 2
    ...
Até (condição_da_variável_de_controle )

```

**Explicação:** Essa estrutura executará o trecho do algoritmo pelo menos uma vez. Quem fará este controle será a variável **condição da variável de controle** que estará associada a uma das comparações que se segue: maior, maior ou igual, menor, menor ou igual ou diferente.

**Exemplo.** Calcule a soma dos números pares de 50 a 100

```

Algoritmo Soma_Pares;
Var i:Numérico;
Início
    i:=26;
    Repita
        i:=i+2;
    Até que (i > 100);
    Escreva(i);
Fim.

```

### Exercícios resolvidos

13 – Faça um algoritmo que leia a nota de 2º alunos e depois calcule e mostre a média das notas.

```

Algoritmo média;

```

```

Var soma,média, nota,contador:Numérico;
Início
  Escreva('Digite as 20 notas dos alunos □ ');
  soma:=0;
  Para contador:=0 Até 20 Faça
  Início
    Leia(nota);
    soma:=soma+nota;
  Fim;
  média:=soma+nota;
  Escreva('A média é □ ');
Fim.

```

14 – Escreva um algoritmo que leia 100 números e informe quais são múltiplos de três.

```

Algoritmo múltiplo_de_três;

Var num,contador:Numérico;
Início
  Escreva('Digite os 100 números □ ');
  Para contador:=1 Até 100 Faça
  Início
    Leia(num);
    Se (num MOD 3 = 0 ) Então
      Escreva(num);
    Fim;
  Fim.

```

15 – Escreva um algoritmo que leia o salário, rg e imposto dos funcionários de uma empresa. Em seguida informe o salário líquido. Quando o rg for “0” o algoritmo deverá ser encerrado.

```

Algoritmo imposto_salario;

Var rg,sal_bruto,sal_liq,imposto:Numérico;

Início
  Escreva('Digite o rg');
  Leia(rg);
  Escreva('Digite o salário bruto');
  Leia(sal_bruto);
  Escreva('Digite o imposto');

```

```

Leia(imposto);
Início
    Leia(sal_bruto,imposto);
    sal_liq:=sal_bruto-imposto;
    Escreva('O salário líquido é ==>',sal_liq);
    Leia(ident);
Fim;
Fim.

```

16 – Escreva um algoritmo para resolver a somatória.

$$S = 1/1 - 2/4 + 3/9 - 4/16 + 5/25 - 6/36... - 10/100.$$

```

Algoritmo Somatória;

Var soma, contador:Numérico;

Início
    soma:=0;
    Escreva('calculando a somatória ==>');
    Para contador:=1 Até 10 Faça
        Início
            soma:=soma+contador*(-1 ^ (contador + 1)) / (contador ^ 2)
        Fim;
    Escreva('A somatória e:', soma);

Fim.

```

17 – Em um curso de programação um programador cometeu 1 erro em seu primeiro programa, 2 erros no segundo, 4 erros no terceiro e assim sucessivamente. Ele está cometendo por programa duas vezes a quantidade de erros do programa anterior. Se o curso dura treze semanas e o programador realiza dois programas por semana, calcular o número de erros que ele espera cometer em seu programa final.

```

Algoritmo Erro_Programador;

Var Erros,Programas, contador:Numérico;

Início
    Erros:=1;
    Programas:=2 * 13;
    Para contador:=2 Até Programas Faça
        Erros:=Erros * 2;
        Escreva('Número de erros ==>',Erros);

Fim.

```

## Estrutura de seleção

- Essa estrutura é utilizada quando desejamos executar uma das opções disponíveis pelo algoritmo. Na execução desta estrutura o algoritmo pára e permite que escolhamos uma delas. Contrariando a estrutura SE que não espera pelo escolha das opções na instante da execução do algoritmo

### Sintaxe 1

```
Caso variável_seletiva De  
    variável_seletiva 1: comando 1;  
    variável_seletiva 1: comando 2;  
    variável_seletiva 1: comando 3;  
    ...  
Fim;
```

**Explicação:** Nessa estrutura o algoritmo possui diversas opções de execução, tornando-o bem mais legível e estruturado, evitando-se desta maneira a utilização da estrutura condicional "SE" de forma repetida. Esta estrutura trás junto com ela uma restrição, onde, variável\_seletiva tem que ser uma variável do tipo char, integer ou boolean.

### Sintaxe 2

```
Caso variável_seletiva De  
    variável_seletiva 1: Início  
        Comando 1;  
        Comando 2;  
        ...  
        Fim;  
    variável_seletiva 2: Início  
        Comando 1;  
        Comando 2;  
        ...  
        Fim;  
    variável_seletiva 3: Início  
        Comando 1;  
        Comando 2;  
        ...  
        Fim;  
    ...  
Fim;
```

**Explicação:** A diferença nessa estrutura é que cada *variável\_seletiva* pode ter como execução um bloco de instruções iniciado por **Início** e finalizado por **Fim**.

**Exemplo 1.** Escreva um algoritmo que mostre as opções de um *menu* que simula um calculadora

```
Algoritmo Menu_Calculadora;  
{Um simples exemplo de como se trabalha com a Estrutura Case }  
Var escolha:Numérico;  
  
Início  
  Escreva('Escolha uma das opções');  
  Leia(escolha);  
  Caso escolha De  
    1: Escreva('Você escolheu a soma!')  
    2: Escreva('Você escolheu o produto!')  
    3: Escreva('Você escolheu a subtração!')  
    4: Escreva('Você escolheu a divisão!')  
  
  Fim;  
Fim.
```

**Exemplo 2.** Escreva um algoritmo que mostre as opções de um *menu* que simula uma calculadora

```
Algoritmo Menu_Calculadora;  
{Um simples exemplo de como se trabalha com a Estrutura Case }  
Var numero1,numero2:Numérico;  
Início  
  Escreva('Digite dois números');  
  Leia(numero1,numero2);  
  Caso numero1 De  
    1:  Início  
        numero2:=1;  
        Escreva('Entrou na primeira opção');  
      Fim;  
    2:  Início  
        numero2:=2;  
        Escreva('Entrou na segunda opção');  
      Fim;  
    3:  Início  
        numero2:=3;  
        Escreva('Entrou na terceira opção');  
      Fim;  
    4:  Início  
        numero2:=4;  
        Escreva('Entrou na quarta opção');  
      Fim;
```



**Fim.**

### Exercícios propostos

- 01 – Faça um algoritmo calcule e mostre a idade média de cem pessoas  
02 – Faça um algoritmo que escreva os números ímpares de 1 a 100  
03 – Escreva um algoritmo que leia o Nome, Idade e Sexo de 50 pessoas. Calcule e imprima:  
a) A média das idades das pessoas  
b) O total dos homens  
c) O total das mulheres com idade menor que 40 anos  
d) A maior e a menor idade e o nome de quem as possui.
- 04 – Qual o valor final das variáveis A,B e C no algoritmo abaixo?

**Algoritmo** Exercicio\_04;

**Var** A,B,C:Numérico;

**Início**

B:=10;

**Enquanto** B>5 **Faça**

**Início**

A:=(B ^ 2);

C:=A - 10;

**Enquanto** C>60 **Faça**

C:=C - 2 \* B;

B:=B - 1;

**Fim;**

**Fim.**

05 – A população A e B possuem 90 milhões de 200 milhões de pessoas respectivamente. A taxa de crescimento de A é 3% e a de B é 1,5% ao ano. Escreva um algoritmo que informe em quantos anos a população de A ultrapassará a população de B.

06 – Escreva um algoritmo que leia dados sobre um número indeterminado de pessoas de uma população. Os dados são Sexo(Masculino ou Feminino), cor dos olhos(Azuis, verdes ou castanhos), cor dos cabelos(louros, castanhos ou pretos) e a idade em anos.

A última leitura será indicada pelo valor da idade -1. Calcular e mostrar:

- a) A maior idade;  
b) A porcentagem dos indivíduos do sexo feminino cuja idade está entre 18 e 35 anos e que tenham cabelos louros.

07 – Escreva um algoritmo que calcule o fatorial de um número qualquer.

08 – Escreva um algoritmo para resolver cada um dos somatório abaixo.

a)  $S = 2^1/50 + 2^2/49 + 2^3/48 + \dots + 2^{50}/1$

b)  $S = 1/1 + 3/2 + 5/3 + 7/4 + \dots + 99/50$

09 – Supondo que numa tabela de 200 nomes possam haver nomes repetidos, informe quantas vezes apareceu determinado nome(dado) e caso não tenha aparecido nenhuma repetição mostre a mensagem “Nome não encontrado”.

10 – Considere o trinômio do segundo grau  $A * x^2 + B * x + C = 0$ . Informe os valores para os coeficientes A, B e C via teclado. Escreva um algoritmo que encontre as raízes reais desta equação. Caso não haja raízes reais o algoritmo deve mostrar a mensagem “Não existe raízes reais. O valor do delta é...”

11 – Criar um algoritmo que calcule e mostre o salário líquido de um funcionário nas condições vigentes quanto ao INSS e IR. (Pesquisar as referidas tabelas junto aos órgãos governamentais). O usuário deve informar o nome, salário mínimo, salário base.

### Estrutura de repetição

- A estrutura de repetição em um algoritmo consiste na execução de um trecho do mesmo levando em consideração certas condições imposta pelo algoritmo.

Temos duas três formas de estruturas de repetição.

- A primeira forma repete uma seqüência de instruções indeterminada de vezes, até que uma condição seja satisfeita.

### Sintaxe

<b>Enquanto</b> condição <b>Faça</b> Comando 1
---

**Explicação:** Isso significa que a condição estabelecida executará o Comando 1 enquanto ela for verdadeira. Pode-se adicionar **Comando 2**, **Comando 3** etc para tornando esta estrutura composta.

**Exemplo.** Calcule a soma dos números pares de 50 a 100

<b>Algoritmo</b> Soma_Pares; <b>Var</b> soma,par:Numérico;
---

```
Início
    soma:=0;
    par:=50;
    Enquanto par<=100 Faça
        soma:=soma+par;
        par:=par+2;
    Fim;
Escreva(soma);
Fim.
```

- A segunda estrutura é utilizada para o mesmo fim. Fazemos uso dela quando queremos que o algoritmo execute uma quantidade de vezes pré-determinada de um trecho do algoritmo.

### Sintaxe

```
Para variável_que_controla_repetição:=valor_inicial Até valor_final Faça
    Comando 1;
```

**Explicação:** Essa estrutura controlará o fluxo do algoritmo mediante o conteúdo informado nas variáveis **variável\_que\_controla\_repetição:=valor\_inicial** e **valor\_final**. A informação deste conteúdos indicará quando o algoritmo será finalizado.

**Exemplo.** Calcule a soma dos números pares de 50 a 100

```
Algoritmo Soma_Pares;
Var soma,n:Numérico;
Início
    soma:=0;
    Para n:=25 Até 50 Faça
        soma:=soma+2*n;
    Escreva(soma);
Fim.
```

- A terceira estrutura é utilizada quando desejamos que o algoritmo execute um determinado trecho pelo menos uma vez.

### Sintaxe

```
Repita
    Comando 1
```

Comando 2

...

**Até** (condição\_da\_variável\_de\_controle )

**Explicação:** Essa estrutura executará o trecho do algoritmo pelo menos uma vez. Quem fará este controle será a variável **condição\_da\_variável\_de\_controle** que estará associada a uma das comparações que se segue: maior, maior ou igual, menor, menor ou igual ou diferente.

**Exemplo.** Calcule a soma dos números pares de 50 a 100

**Algoritmo** Soma\_Pares;

**Var** i:Numérico;

**Início**

i:=26;

**Repita**

l:=i+2;

**Até que** (i > 100);

**Escreva**(i);

**Fim.**

### Exercícios resolvidos

13 – Faça um algoritmo que leia a nota de 2º alunos e depois calcule e mostre a média das notas.

**Algoritmo** média;

**Var** soma,média, nota,contador:Numérico;

**Início**

**Escreva**('Digite as 20 notas dos alunos □ ');

soma:=0;

**Para** contador:=0 **Até** 20 **Faça**

**Início**

Leia(nota);

soma:=soma+nota;

**Fim;**

média:=soma+nota;

**Escreva**('A média é □ ');

**Fim.**

14 – Escreva um algoritmo que leia 100 números e informe quais são múltiplos de três.

**Algoritmo** múltiplo\_de\_três;

**Var** num,contador:Numérico;

**Início**

**Escreva**('Digite os 100 números □ ');

**Para** contador:=1 **Até** 100 **Faça**

**Início**

            Leia(num);

**Se** (num MOD 3 = 0 ) **Então**

**Escreva**(num);

**Fim;**

**Fim.**

15 – Escreva um algoritmo que leia o salário, rg e imposto dos funcionários de uma empresa. Em seguida informe o salário líquido. Quando o rg for “0” o algoritmo deverá ser encerrado.

**Algoritmo** imposto\_salario;

**Var** rg,sal\_bruto,sal\_liq,imposto:Numérico;

**Início**

**Escreva**('Digite o rg');

**Leia**(rg);

**Escreva**('Digite o salário bruto');

**Leia**(sal\_bruto);

**Escreva**('Digite o imposto');

**Leia**(imposto);

**Início**

**Leia**(sal\_bruto,imposto);

        sal\_liq:=sal\_bruto-imposto;

**Escreva**('O salário líquido é ==>',sal\_liq);

**Leia**(ident);

**Fim;**

**Fim.**

16 – Escreva um algoritmo para resolver a somatória.

$S = 1/1 - 2/4 + 3/9 - 4/16 + 5/25 - 6/36 \dots - 10/100.$

**Algoritmo** Somatória;

**Var** soma, contador:Numérico;

**Início**

```

soma:=0;
Escreva('calculando a somatória ==>');
Para contador:=1 Até 10 Faça
    Início
        soma:=soma+contador(-1 ^ (contador + 1)) / (contador ^ 2)
    Fim;
Escreva('A somatória e:', soma);

```

**Fim.**

17 – Em um curso de programação um programador cometeu 1 erro em seu primeiro programa, 2 erros no segundo, 4 erros no terceiro e assim sucessivamente. Ele está cometendo por programa duas vezes a quantidade de erros do programa anterior. Se o curso dura treze semanas e o programador realiza dois programas por semana, calcular o número de erros que ele espera cometer em seu programa final.

```

Algoritmo Erro_Programador;

Var Erros,Programas, contador:Numérico;

Início
    Erros:=1;
    Programas:=2 * 13;
    Para contador:=2 Até Programas Faça
        Erros:=Erros * 2;
        Escreva('Número de erros ==>',Erros);

```

**Fim.**

### Estrutura de seleção

- Essa estrutura é utilizada quando desejamos executar uma das opções disponíveis pelo algoritmo. Na execução desta estrutura o algoritmo pára e permite que escolhamos uma delas. Contrariando a estrutura SE que não espera pelo escolha das opções na instante da execução do algoritmo

#### Sintaxe 1

```

Caso variável_seletiva De
    variável_seletiva 1: comando 1;
    variável_seletiva 1: comando 2;
    variável_seletiva 1: comando 3;
    ...
Fim;

```

**Explicação:** Nessa estrutura o algoritmo possui diversas opções de execução, tornando-o bem mais legível e estruturado, evitando-se desta maneira a utilização da estrutura condicional “SE” de forma repetida. Esta estrutura trás junto com ela uma restrição, onde, *variável\_seletiva* tem que ser uma variável do tipo char, integer ou boolean.

## Sintaxe 2

```

Caso variável_seletiva De
    variável_seletiva 1:      Início
                              Comando 1;
                              Comando 2;
                              ...
                              Fim;
    variável_seletiva 2: Início
                              Comando 1;
                              Comando 2;
                              ...
                              Fim;
    variável_seletiva 3: Início
                              Comando 1;
                              Comando 2;
                              ...
                              Fim;
    ...
Fim;

```

**Explicação:** A diferença nessa estrutura é que cada *variável\_seletiva* pode ter como execução um bloco de instruções iniciado por **Início** e finalizado por **Fim**.

**Exemplo 1.** Escreva um algoritmo que mostre as opções de um *menu* que simula um calculadora

```

Algoritmo Menu_Calculadora;
    {Um simples exemplo de como se trabalha com a Estrutura Case }
Var   escolha:Numérico;

Início
    Escreva('Escolha uma das opções');
    Leia(escolha);
    Caso escolha De
        1: Escreva('Você escolheu a soma!')
        2: Escreva('Você escolheu o produto!')
        3: Escreva('Você escolheu a subtração!')
        4: Escreva('Você escolheu a divisão!')
    Fim;

```

**Fim.**

**Exemplo 2.** Escreva um algoritmo que mostre as opções de um *menu* que simula uma calculadora

```
Algoritmo Menu_Calculadora;  
{Um simples exemplo de como se trabalha com a Estrutura Case }  
Var numero1,numero2:Numérico;  
Início  
  Escreva('Digite dois números');  
  Leia(numero1,numero2);  
  Caso numero1 De  
    1:  Início  
        numero2:=1;  
        Escreva('Entrou na primeira opção');  
        Fim;  
    2:  Início  
        numero2:=2;  
        Escreva('Entrou na segunda opção');  
        Fim;  
    3:  Início  
        numero2:=3;  
        Escreva('Entrou na terceira opção');  
        Fim;  
    4:  Início  
        numero2:=4;  
        Escreva('Entrou na quarta opção');  
        Fim;
```

**Fim.**

### Exercícios propostos

- 01 – Faça um algoritmo calcule e mostre a idade média de cem pessoas
- 02 – Faça um algoritmo que escreva os números ímpares de 1 a 100
- 03 – Escreva um algoritmo que leia o Nome, Idade e Sexo de 50 pessoas. Calcule e imprima:
- A média das idades das pessoas
  - O total dos homens
  - O total das mulheres com idade menor que 40 anos
  - A maior e a menor idade e o nome de quem as possui.
- 04 – Qual o valor final das variáveis A,B e C no algoritmo abaixo?



**Algoritmo** Exercicio\_04;

**Var** A,B,C:Numérico;

**Início**

B:=10;

**Enquanto** B>5 **Faça**

**Início**

A:=(B ^ 2);

C:=A - 10;

**Enquanto** C>60 **Faça**

C:=C - 2 \* B;

B:=B - 1;

**Fim;**

**Fim.**

05 – A população A e B possuem 90 milhões de 200 milhões de pessoas respectivamente. A taxa de crescimento de A é 3% e a de B é 1,5% ao ano. Escreva um algoritmo que informe em quantos anos a população de A ultrapassará a população de B.

06 – Escreva um algoritmo que leia dados sobre um número indeterminado de pessoas de uma população. Os dados são Sexo(Masculino ou Feminino), cor dos olhos(Azuis, verdes ou castanhos), cor dos cabelos(louros, castanhos ou pretos) e a idade em anos. A última leitura será indicada pelo valor da idade -1. Calcular e mostrar:

- A maior idade;
- A porcentagem dos indivíduos do sexo feminino cuja idade está entre 18 e 35 anos e que tenham cabelos louros.

07 – Escreva um algoritmo que calcule o fatorial de um número qualquer.

08 – Escreva um algoritmo para resolver cada um dos somatório abaixo.

a)  $S = 2^1/50 + 2^2/49 + 2^3/48 + \dots + 2^{50}/1$

b)  $S = 1/1 + 3/2 + 5/3 + 7/4 + \dots + 99/50$

09 – Supondo que numa tabela de 200 nomes possam haver nomes repetidos, informe quantas vezes apareceu determinado nome(dado) e caso não tenha aparecido nenhuma repetição mostre a mensagem "Nome não encontrado".

10 – Considere o trinômio do segundo grau  $A * x^2 + B * x + C = 0$ . Informe os valores para os coeficientes A, B e C via teclado. Escreva um algoritmo que encontre as raízes reais desta equação. Caso não haja raízes reais o algoritmo deve mostrar a mensagem "Não existe raízes reais. O valor do delta é..."

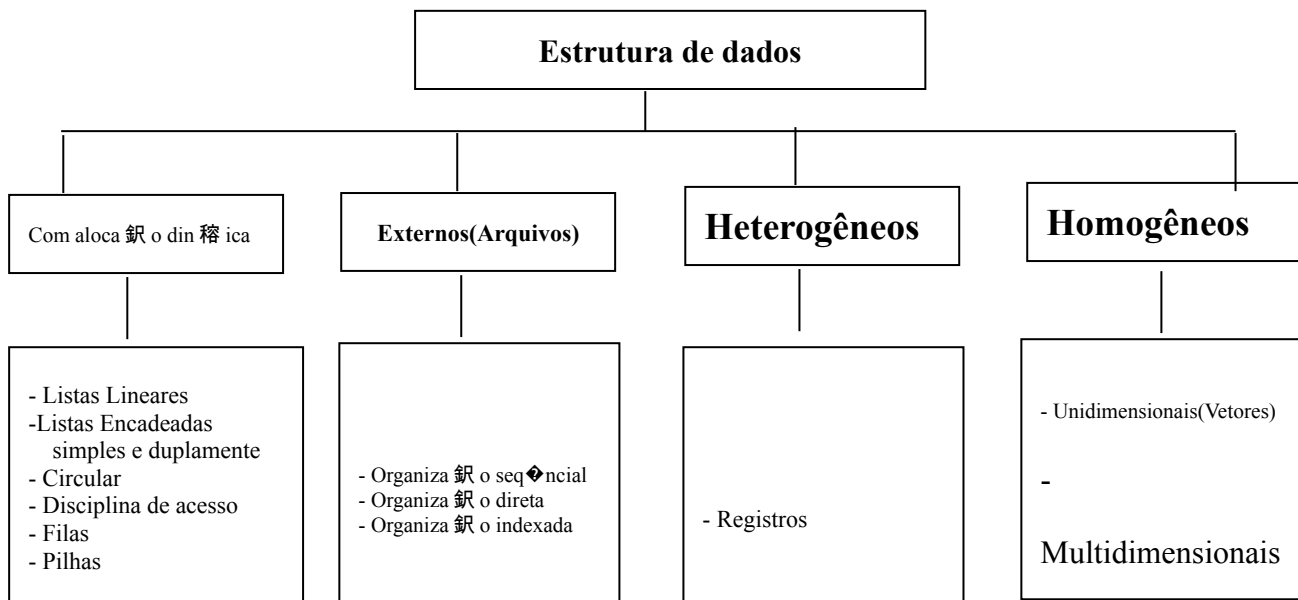
11 – Criar um algoritmo que calcule e mostre o salário líquido de um funcionário nas condições vigentes quanto ao INSS e IR. (Pesquisar as referidas tabelas junto aos órgãos governamentais). O usuário deve informar o nome, salário mínimo, salário base.

## Estrutura de dados

Cada um dos dados manipulados nos algoritmos deste material está associado a uma variável. As variáveis possuem uma faixa (escopo) de valores armazenáveis. Cada faixa de valor é chamada de tipo.

Os tipos aprendidos, até agora, nem sempre são suficientes para representar todos os dados de um algoritmo. Neste capítulo serão abordados novos tipos de dados chamados: tipos estruturados, a partir da composição dos dados primitivos. Os novos tipos de dados possuem uma estrutura que definem como os dados serão organizados e armazenados.

A maneira como os dados são estruturados, organizados e armazenados, ou seja, a estrutura de dados criada a partir dos tipos primitivos é tão importante quanto o próprio algoritmo na solução de problemas. Segundo Niklaus Wirth, criador da linguagem PASCAL, Um “Programa = Algoritmo + Estrutura de dados”



Tipo escalar

Seqüência de elementos que constituem um conjunto.

Exemplo

DECLARAÇÃO DO TIPO.

Dia = (Domingo, Segunda, Terça, Quarta, Quinta, Sexta, Sábado);

DECLARAÇÃO DE VARIÁVEIS.

Dia\_semana:Dia;

A variável dia\_da\_semana é do tipo Dia, ou seja, só pode armazenar apenas constantes contidas no conjunto "Dia".

Obs.: Os tipos já estudados Literal, Inteiro e Lógico se comportam como tipos escalares.

Subfaixas

- É um subconjunto do tipo escalar.

Exemplo.

DECLARAÇÃO DE TIPOS.

Meses=1..12;

Alfabeto=A ..Z;

Obs.: A declaração de tipos não declara variáveis, portanto estas deverão ser declaradas explicitamente como acima.

Estrutura de dados homogêneas

## Vetores

As variáveis simples armazenam apenas um dado de um tipo (inteira, literal, lógica etc), ou seja, o identificador da variável simples faz referência a apenas um elemento na memória. Esta limitação gerou a necessidade de novas estruturas para identificar e armazenar dados.

As variáveis do tipo vetor também armazenam dados de um só tipo primitivo de dados, porém o identificador da variável vetor pode referenciar mais de um elemento.

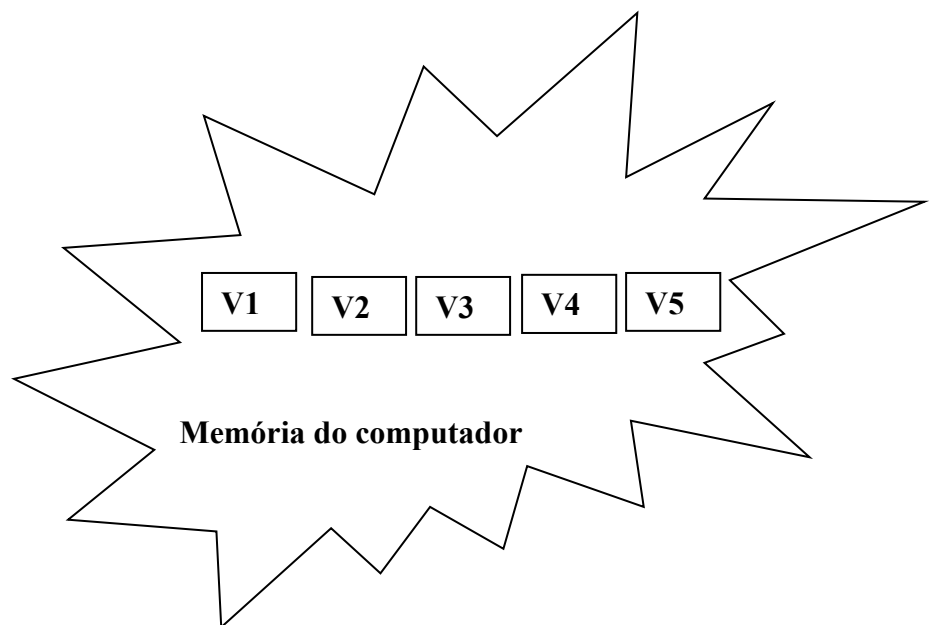
Os elementos de um vetor são identificados por um índice, que vai de um(1) até um limite pré-definido pelo programador para representar o conjunto de dados a serem processados.

O efeito na memória na utilização de variáveis simples e vetores para armazenar as cinco vogais.

Tipo Primitivo

Declaração de variáveis

V1,V2,V3,V4,V5:Literal;



Obs.:Foram necessários cinco variáveis(V1,V2,V3,V4 e V5) para representar as cinco vogais.

A atribuição das mesmas ficará

V1:='a'; V2:='e'; V2:='i'; V2:='o'; V2:='u';

Se a declaração deste tipo utilizasse vetores ficaria.

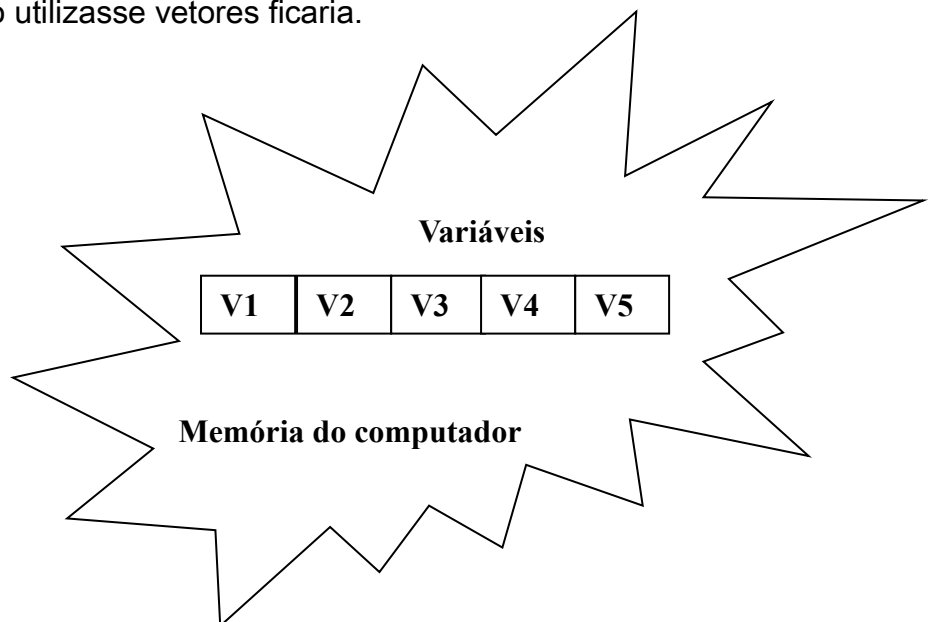
Tipo Estruturado

Declaração de tipo

Vogal: Vetor[5] de Literal;

Declaração de variáveis

V: Vogal;



Obs.: A variável “V” do tipo Vogal, que é um vetor de cinco elementos, pode armazenar até cinco elementos do tipo Literal. Cada um dos valores armazenados é referenciado através de um índice.

A atribuição dos valores fica

V[1]:=‘a’;      V[2]:=‘e’;      V[3]:=‘i’; V[4]:=‘o’;      V[5]:=‘u’;

Suponha um situação em que um algoritmo precise manipular 90.000 nomes dos alunos de uma faculdade. V1,V2,V3...V90000. Seriam 90000 variáveis. Usando uma variável do tipo vetor, teríamos apenas uma variável V: Vetor[90000] de Literal;

Exemplo. Qual o terceiro elemento do vetor abaixo?

#### Vetor N

Índice	1	2	3	4	5
Valor	7	2	6	9	6

Resposta N[3] vale 6

Exemplo. Como carregar um vetor N de 50 elementos numéricos inteiros. Crie uma variável inteira chamada “T” para se o índice.

Declaração  N: vetor[50] de inteiros; I: Numérico;  <b>Início</b> <b>Para</b> i:= 1 <b>Até</b> 50 <b>Faça</b> <b>Leia</b> (N[i]); <b>Fim.</b>
---

Obs.: Não somos obrigados a carregar todas as posições de um vetor. Podemos declarar um vetor capacidade para armazenar 100 elementos e precisar apenas de 90 temporariamente. Agora, você não poderá armazenar mais valores que o declarado.

**Exemplos.** Declarar um vetor “Nome” com capacidade para armazenar 200 nomes e carregar o vetor até que seja digitado o valor “Fim” . O valor “Fim” indicará a última posição do vetor a ser carregado.

Declaração

Nome: vetor[200] de Lietral;  
I:Numérico;

**Início**

**I:=0;**

**Leia(Nome[i]);**

**Enquanto** (Nome[i] <> “Fim”) **E** (i<200) **Faça**

**Início**

**I:=i+1;**

**Leia(N[i]);**

**Fim;**

**Fim.**

Obs.: O vetor continuará sendo carregado só enquanto as das condições forem verdadeiras