

Capítulo I

Conceitos Iniciais

Lógica de programação é a técnica de encadear pensamentos para atingir determinado objetivo. O aprendizado desta técnica é necessário, para quem quer trabalhar com desenvolvimento de sistemas e programas.

Algoritmo é uma seqüência de passos finitos com o objetivo de solucionar um problema.

O estudo da lógica é o estudo dos métodos e princípios usados para distinguir o raciocínio correto do incorreto. Naturalmente, essa definição não pretende afirmar que só é possível argumentar corretamente com uma pessoa que já tenha estudado lógica. Afirmá-lo seria tão errôneo quanto pretender que só é possível correr bem, se estudou física e fisiologia, necessárias para a descrição dessa atividade. Alguns excelentes atletas ignoram completamente os processos complexos que se desenrolam dentro deles próprios quando praticam o esporte.

Assim também acontece no nosso dia-a-dia. Quantas vezes já vimos um algoritmo e não sabíamos que aquela seqüência de passos chamava-se algoritmo.

Um exemplo bem freqüente é quando queremos falar em algum telefone público.

Algoritmo para falar em um telefone público

- 1 – Retirar o telefone do gancho;
- 2 – Esperar o sinal;
- 3 – Colocar o cartão;
- 4 – Discar o número;
- 5 – Falar no telefone;
- 6 – Colocar o telefone no gancho.

O algoritmo é exatamente esse conjunto de passos que resolveu o problema de uma pessoa falar no telefone. É como se fôssemos ensinar uma máquina a fazer alguma tarefa específica.

Outro exemplo clássico é um algoritmo para resolver o problema de fritar um ovo que poderia estar escrito em forma de uma receita. A receita é um algoritmo, pois é formada de ações que devem ser tomadas para fritar um ovo.

Algoritmo para fritar um ovo

- 1 – pegar frigideira, ovo, óleo e sal;
- 2 – colocar óleo na frigideira;
- 3 – acender o fogo;
- 4 – colocar a frigideira no fogo;
- 5 – esperar o óleo esquentar;
- 6 – colocar o ovo;
- 7 – colocar o sal;
- 8 – retirar quando estiver pronto.

Cada linha do algoritmo podemos chamar de uma instrução, logo, podemos dizer que um algoritmo é um conjunto de instruções.

Assim como fritar um ovo, nos algoritmos computacionais não podemos trocar ou omitir certas instruções, caso contrário não obteremos o resultado esperado. Por exemplo, se omitirmos a instrução acender o fogo, não teremos ao final do algoritmo um ovo frito.

Instrução indica a um computador uma ação elementar a ser executada.

Até as coisas mais simples podem ser descritas por um algoritmo. Por exemplo:

Algoritmo para o fim de semana

- 1 – vejo a previsão do tempo;

Professora Lucélia Alves de Oliveira

2 – se fizer sol

vou à praia;

senão

vou estudar;

3 – almoçar;

4 – ver televisão;

5 – dormir.

EXERCÍCIOS

1 – Fazer um algoritmo para tomar banho:

2 – Crie um algoritmo para fazer uma prova:

3 – Faça um algoritmo para somar dois números:

Programa de Computador nada mais é do que um algoritmo escrito numa linguagem de computador (C, Pascal, Fortran, Delphi, Cobol, Java e outras). É a tradução para o inglês do algoritmo feito em português. O mais importante de um programa é a sua lógica, o raciocínio utilizado para resolver o problema, que é exatamente o algoritmo.

A forma de escrever um algoritmo em pseudocódigo (algoritmo que não usa nenhuma linguagem de programação) vai variar de autor para autor, pois, um traduz ao pé da letra a linguagem C, outro, o Pascal, outro, mistura as duas linguagens e assim por diante.

É importante lembrar que estas variações vão sempre ocorrer, podemos dizer que é uma variação de autores adotados.

FASES

A principal finalidade de um computador é realizar a tarefa de processamento de dados, isto é, receber dados através de dispositivos de entrada que podem ser, por exemplo, teclado, mouse, scanner, entre outros; realizar operações com estes dados e gerar uma resposta que será expressa em um dispositivo de saída que pode ser, por exemplo, uma impressora, um monitor de vídeo, entre outros.

Entretanto ao montar um algoritmo, precisamos primeiro dividir o problema apresentado em três fases fundamentais:



Onde:

ENTRADA: São os dados de entrada do algoritmo.

PROCESSAMENTO: São os procedimentos utilizados para chegar ao resultado final.

SAÍDA: São os dados já processados, os resultados, mostrados na tela do computador (monitor de vídeo) ou impressora.


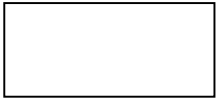
Diagrama de Blocos



Diagrama de bloco: é uma forma padronizada para representar os passos lógicos de um determinado processamento.

Com o diagrama, também conhecido como fluxograma, podemos definir uma seqüência de símbolos, com significado bem definido, portanto, sua principal função é a de facilitar a visualização dos passos de um processamento.

Simbologia do Diagrama de Bloco

Existem diversos símbolos em um diagrama de bloco. Veja no quadro abaixo alguns dos símbolos que iremos utilizar:

Símbolo	Função
 TERMINAL	Indica o início ou fim de um processamento Exemplo: Início do algoritmo
 PROCESSAMENTO	Processamento em geral Exemplo: $x \leftarrow 2+3$

 ENTRADA MANUAL DE DADO	Indica entrada de dados pelo usuário via teclado Exemplo: Digite a nota da prova 1
 EXIBIR	Mostra informações ou resultados Exemplo: Mostre o resultado do cálculo

EXERCÍCIOS

1. Construir um diagrama de blocos que:
 - leia quatro números
 - calcule a média
 - mostre o resultado.
2. Construa fluxograma que:
 - leia o salário de um empregado
 - calcule o novo salário sabendo que lê teve um aumento de 15%
 - mostre o resultado na tela
3. Agora faça um algoritmo para o exercício 2.

Capítulo II

Variáveis

É um local na memória principal, isto é, um endereço que armazena um conteúdo.

O conteúdo de uma variável pode ser de vários tipos: inteiro, real, caractere (literal), String (cadeia), lógico, entre outros.

Uma vez definidos o nome e o tipo de uma variável, não podemos alterá-los no decorrer de um algoritmo. Por outro lado, o conteúdo da variável pode ser modificado no decorrer do programa, de acordo com a necessidade.

Em algoritmos, as variáveis são definidas no início por meio do comando definido:

Nome da variável: tipo da variável;

Exemplos:

A: inteiro;

X: real;

Nome: string (ou cadeia);

Regras para formação do nome de uma variável:

- 1 – O primeiro caractere é uma **letra**;
- 2 – Se houver mais de um caractere, poderemos usar letras ou dígitos;
- 3 – Nomes de variáveis escritas com letras maiúsculas serão diferentes de letras minúsculas em muitas linguagens de programação. Lembre-se: peso é **diferente** de PESO;
- 4 – Nenhuma palavra reservada poderá ser nome de uma variável.

Nomes Válidos	Nomes Não-Válidos
Nome, profissao,n, n1,PESO, A	2X -> Começa por algarismo peso do aluno -> espaço não é permitido

Tipos de Variáveis

Numéricas

São aquelas que armazenam dados numéricos, podendo ser divididos em duas classes:

Inteiro: são aqueles que **não** possuem componentes decimais ou fracionários, podendo ser positivos ou negativos.

*Normalmente uma variável do tipo inteira poderá ocupar **1, 2 ou 4 bytes** na MP.*

Exemplos:

10 número inteiro positivo
-10 número inteiro negativo

Real: são aqueles que podem possuir componentes decimais ou fracionários, podendo também ser positivos ou negativos.

*Normalmente uma variável do tipo real poderá ocupar **4 ou 8 bytes** na MP.*

Exemplos:

25.03 número real positivo com duas casas decimais
235. número real positivo com zero casas decimais
-10.5 número real negativo com uma casa decimal

Cadeia ou String ou Alfanuméricas: são aqueles que possuem letras e/ou números. Pode em determinados momentos conter somente dados numéricos ou somente letras. Se usado somente para armazenamento de números, não poderá ser utilizado para operações matemáticas.

O número de bytes possíveis para armazenamento de uma variável desse tipo dependerá da quantidade de caracteres.

Exemplos:

"Maria"	String de comprimento 5
"123"	String de comprimento 3
"A"	String de comprimento 1

Caractere: é aquele que armazena apenas uma letra ou dígito, é uma string de comprimento 1. Se usado para armazenar número não poderá ser utilizado para operações matemáticas.

*Normalmente uma variável do tipo caractere ocupará **1 byte** na **MP**.*

Exemplos:

- "A" caractere que armazena uma letra
- "5" caractere que armazena um número

Lógico

Também conhecido como booleano. É representado no algoritmo pelos dois únicos valores lógicos possíveis: verdadeiro ou falso. Porém é comum encontrar em outras referências outros pares de valores lógicos como: sim/não, 1/0, true/false, verdadeiro/falso.

Constantes

Constante é um determinado valor fixo que não se modifica ao longo do tempo, durante a execução de um programa. Conforme seu tipo, a constante é classificada como sendo numérica, lógica ou literal.

As declarações das constantes em um algoritmo são feitas antes das declarações das variáveis:

Sintaxe:

Nome da constante = valor da constante;

Expressões

O conceito de expressão em termos computacionais está intimamente ligado ao conceito de expressão (ou fórmula) matemática, onde um conjunto de variáveis e constantes numéricas relaciona-se por meio de operadores compondo uma fórmula que, uma vez avaliada, resulta num valor.

Os operadores matemáticos são:

Operador	Função
+	Somar
-	Subtrair
*	Multiplicar
/	Dividir
Div	Quociente inteiro
Mod	Resto da divisão

Os operadores Div e Mod só podem ser aplicados com operandos inteiros.

O operador / sempre conduz a um resultado real.

Com os operadores +, - e * , se pelo menos um dos operandos for real, então o resultado será real.

Exemplos de expressões aritméticas:

$$3 * 3 = 9$$

$$9 / 2 = 4.5$$

$$9 \text{ DIV } 2 = 4$$

$$9 \text{ MOD } 2 = 1$$

$$2 / 9 = 0.22$$

$$2 + 4.6 = 6.6$$

Expressão Relacional

Uma expressão relacional é uma comparação realizada entre dois valores de mesmo tipo, tendo como resposta sempre um valor booleano (verdadeiro ou falso). Estes valores são representados na relação através de constantes, variáveis ou expressões aritméticas.

Os operadores matemáticos são:

Símbolo	Descrição
=	Igual
< >	Diferente
< =	Menor ou igual
> =	Maior ou igual
>	Maior que
<	Menor que

Exemplo de relações:

$X = 1$ $y = 2$ $z = 5$

$X * X + Y > Z$

$1 * 1 + 2 > 5$

$1 + 2 > 5$

$3 > 5$

Resultado desta expressão: FALSO

Expressão Lógica

Uma expressão lógica serve para combinar resultados de expressões aritméticas e/ou relacionais, variáveis e/ou constantes, retornando verdadeiro ou falso.

Como exemplo de operadores lógicos, matematicamente conhecidos temos:

E	And
Ou	Or
Não	Not

E / AND Uma expressão AND (E) é verdadeira se todas as condições forem verdadeiras.

OR / OU Uma expressão OR (OU) é verdadeira se pelo menos uma condição for verdadeira.

NOT / NÃO Uma expressão NOT (NÃO) inverte o valor da expressão ou condição, se verdadeira inverte para falsa e vice-versa.

Tabela E (AND)	Tabela OU (OR)	Tabela NÃO (NOT)
V e V = V	V ou V = V	Não V = F
V e F = F	V ou F = V	Não F = V
F e V = F	F ou V = V	
F e F = F	F ou F = F	

As prioridades entre os operadores são:

1º - ()

2º - funções

3º - Não

4º - *, /, Div, Mod, E

5º - +, -, Ou

6º - =, < >, <, <=, >, >=.

Observação: em caso de empate entre as prioridades, resolver da esquerda para a direita.

EXERCÍCIOS

1. Identifique o tipo dos dados:

- (a) inteiro;
- (b) real;
- (c) lógico;
- (d) cadeia ou literal

- | | | | |
|-------------------------------------|--------------------------------|--------------------------------|------------------------------------|
| <input type="checkbox"/> verdadeiro | <input type="checkbox"/> 'c*d' | <input type="checkbox"/> falso | <input type="checkbox"/> '1 2 3 4' |
| <input type="checkbox"/> 'aula' | <input type="checkbox"/> 897 | <input type="checkbox"/> '345' | <input type="checkbox"/> -18.589 |
| <input type="checkbox"/> -0.342 | <input type="checkbox"/> 35.23 | <input type="checkbox"/> -23 | <input type="checkbox"/> 'Maria' |

2. Indique os identificadores como válidos ou inválidos:

- (a) identificador válido
- (b) identificador inválido

- | | | |
|--|---------------------------------|---|
| <input type="checkbox"/> ano | <input type="checkbox"/> ai! | <input type="checkbox"/> 3/1 |
| <input type="checkbox"/> media_salario | <input type="checkbox"/> A15B34 | <input type="checkbox"/> nome-aluno |
| <input type="checkbox"/> média | <input type="checkbox"/> 'aula' | <input type="checkbox"/> 5 ^a |

3. Faça a declaração de 2 variáveis do tipo inteira, 2 do tipo real, 2 de um único caractere e 2 do tipo cadeia de caracteres.

4. Indique qual o resultado das expressões aritméticas abaixo:

Sendo: $x = 6.0$ $y = 2$ $z = 4.0$ $a = 8$ $b = 7.5$ $c = 7.7$
 $d = 12$ $p = 4$ $q = 3$ $r = 10$ $s = 2.5$

- | | |
|-------------------------|-------------------------------------|
| a) $x + y - z * a =$ | f) $((z / a) + b * a) - d =$ |
| b) $d \text{ div } y =$ | g) $100 * (q \text{ div } p) + r =$ |
| c) $d \text{ mod } y =$ | h) $p * (r \text{ mod } q) - q/2 =$ |
| d) $y \text{ div } d =$ | i) $\text{raiz}(r - (q * q)) =$ |
| e) $y \text{ mod } d =$ | j) $(a + r) * r =$ |

5. Dadas as informações abaixo, informe qual o valor das relações (verdadeiro ou falso):

- a) $a = 2.0$, $b = 9.0$, $\text{nome} = \text{'ana'}$, $\text{profissao} = \text{'advogado'}$
 $a + 1 \geq \text{raiz}(b)$ ()
 $\text{nome} < > \text{'ana'}$ ()
 $\text{profissao} = \text{'médico'}$ ()
- b) $a = 6.0$, $b = 121.0$, $\text{nome} = \text{'pedro'}$, $\text{profissao} = \text{'médico'}$
 $a + 1 \geq \text{raiz}(b)$ ()
 $\text{nome} < > \text{'ana'}$ ()
 $\text{profissao} = \text{'médico'}$ ()
- c) $x = 3$, $y = 4$, $z = 16$, $\text{nome} = \text{'maria'}$, $\text{resultado} = \text{verdadeiro}$
 $(x + y > z) \text{ and } (\text{nome} = \text{'maria'})$ ()
 $(\text{resultado}) \text{ or } (y \geq x)$ ()
 $(\text{not resultado}) \text{ and } (z \text{ div } y + 1 = x)$ ()
 $(\text{nome} = \text{'josé'}) \text{ and } (x + y + z < (y * y))$ ()

CAPÍTULO III

Comandos Básicos

Comando de Atribuição

Este comando é utilizado para atribuir valores a variáveis e, em português (português estruturado), é representado por <- (seta da direita para esquerda).

Exemplos de comandos de atribuição:

```
Cor <- 'verde';
```

```
Teste <- falso;
```

```
Media <- (n1 + n2) / 2;
```

```
X <- raiz (16);
```

Comando de Entrada

O comando de entrada **LEIA** é utilizado para receber dados do teclado.

É o comando que permite que o usuário digite dados, possibilitando um “diálogo com o computador”. O dado digitado é armazenado na variável indicada no comando.

Lembre-se de que o nome de uma variável representa uma posição de memória.

Sintaxe:

```
Leia (nome de uma variável);
```

Exemplo:

```
Leia (n);
```

Comando de Saída

É o comando responsável por enviar um resultado, uma informação ao usuário. O valor de cada variável é buscado na memória e mostrado em um dispositivo de saída. Através

desse comando o computador pode emitir os resultados e outras mensagens para o usuário através da tela do computador (monitor) ou uma impressora.

Sintaxe:

Escreva (expressão ou variável ou constantes);

Exemplos:

Escreva (' Aprender lógica com esta apostila ficou bem mais fácil!');

Escreva (' Digite o seu nome: ');

Escreva (A + B);

Escreva ('A média das notas é = ', media);

Exemplo: Algoritmo que lê um nome digitado pelo usuário e mostra uma mensagem.

Português Estruturado

Programa exemplo1;

Var

nome: literal;

Início

Escreva ('Digite um nome: ');

Leia (nome);

Escreva(nome, ' é um nome muito bonito! ');

Fim.

Pascal

Program exemplo1;

Var

nome: string;

Begin

Write ('Digite um nome: ');

Read (nome);

Write (nome, ' é um nome muito bonito! ');

End.

EXERCÍCIOS - Estrutura Seqüencial

1. Faça um algoritmo que leia dois números, calcule e imprima a soma desses dois números.
2. Faça um programa que receba dois números reais, calcule e mostre a subtração do primeiro número pelo segundo.

3. Faça um programa que receba dois números inteiros, calcule e imprima a divisão do primeiro número pelo segundo.
4. Faça um programa que leia um número e informe a metade e o dobro desse número.
5. Escreva um programa que receba um número via teclado e informe em seguida a metade e o quadrado desse número.
6. Escrever um programa que permita receber o nome e a idade de uma pessoa e em seguida, informar o nome digitado e a idade da pessoa daqui a 30 anos.
7. Faça um programa que leia três notas de um aluno, calcule e imprima a média aritmética entre essas notas.
8. Faça um programa que receba dois números inteiros, calcule e imprima:
 - soma do dois números;
 - subtração do primeiro pelo segundo;
 - subtração do segundo pelo primeiro;
 - produto dos dois números;
 - divisão do primeiro pelo segundo;
 - quociente inteiro da divisão do primeiro pelo segundo;
 - resto da divisão do primeiro pelo segundo.
9. Faça um programa que receba quatro números inteiros, calcule e mostre a soma desses números.
10. Faça um programa que receba três notas e seus respectivos pesos. Calcule e mostre a média ponderada dessas notas.
11. Faça um programa que receba o valor do salário de um funcionário. Calcule e mostre o novo salário, sabendo-se que este sofreu aumento de 25%.

12. Faça um programa que receba o valor do salário de um funcionário e o percentual de aumento, calcule e mostre o valor do aumento e o novo salário.
13. Faça um programa que receba o valor do salário-base de um funcionário, calcule e mostre o salário a receber, sabendo-se que este funcionário tem gratificação de 5% sobre o salário-base e paga imposto de 7% sobre o salário-base.
14. Faça um programa que receba o valor do salário-base de um funcionário, calcule e mostre o salário a receber, sabendo-se que este funcionário tem gratificação de R\$ 1.000,00 e paga imposto de 10% sobre o salário-base.

CAPÍTULO IV

Estruturas de Seleção

CONCEITOS

Os algoritmos até agora seguiram um mesmo padrão: entrava-se com dados, estes eram processados e alguma informação era mostrada na tela.

Dessa forma, o computador mais parecia uma máquina de calcular. O aprendizado de novos conceitos, como a estrutura de seleção, nos dará uma visão maior da complexidade de tarefas que ele poderá executar.

Um exemplo do nosso dia-a-dia: imagine-se diante de um caixa eletrônico e suponha que sua senha seja 1234:

Na tela aparece a mensagem:

- Digite sua senha : ■

E o cursor (■ ou |) fica piscando:

Você digita os algarismos da sua senha.

Neste momento, a Unidade Lógica e Aritmética (um dos componentes da CPU) verifica se os números que você digitou são iguais a **1234**. Caso tenha sido, a transação bancária continua; mas, se você digitou algo diferente, aparece na tela a mensagem: **SENHA INVÁLIDA**.

SINTAXES:

Seleção Simples

Se condição então

Início

Comando (s);

Fim;

Seleção Composta

Se condição então

Início

Comando (s);

Fim

Senão

Início

Comando (s);

Fim;

Podemos constatar que esta estrutura faz parte do nosso cotidiano:

- Se eu não tiver prova, vou ao clube; senão vou estudar.
- Se eu tiver aumento, troco de carro; senão espero o 13º salário.
- Se minha média for maior ou igual a sete, passo direto; senão faço exame final.

A única coisa diferente é a forma como iremos escrevê-la, vamos adaptar os algoritmos para uma linguagem específica de um modo formal, obedecendo as regras da linguagem.

Exemplo: Algoritmo que lê um número e mostra uma mensagem dizendo se o número digitado é ou não múltiplo de 5.

Português Estruturado:

Programa ExemploSe;

Var

num, resto: inteiro;

Início

Escreva ('Digite um número: ');

Leia (num);

resto <- num mod 5;

Se resto = 0 então

Escreva (num, ' é múltiplo de 5')

```
Senão
    Escreva (num, ' não é
múltiplo de 5');
Fim.
```

Pascal:

```
Program ExemploSe;
Var
num, resto: integer;
Begin
```

```
Write ('Digite um número: ');
Read (num);
resto := num mod 5;
If resto = 0 then
    Write (num, ' é múltiplo de 5')
Else
    Write (num, ' não é múltiplo
de 5');
End.
```

Observações:

- Antes de *Else* não se usa ponto e vírgula;
- O *Begin* e o *End* são obrigatórios somente quando se tem mais de um comando dentro do *If* e do *Else*.

EXERCÍCIOS – Estrutura de Seleção

1. Faça um programa que leia um número e informe se o dobro do número é maior que 35.
2. Faça um programa que leia um número e informe se a metade do número é menor que 12.
3. Faça um programa que leia dois números e informe apenas se o primeiro é maior que o segundo.
4. Faça um programa que leia dois números e informe se o primeiro é igual ao segundo.
5. Faça um programa que leia dois números e informe se o primeiro é maior, menor ou igual ao segundo.

6. Faça um programa que leia três números e informe qual dos três é o maior.
7. Fazer um programa que receba um número e mostre se ele é positivo, negativo ou nulo (0) .
8. Elabore um algoritmo que leia um número e mostre o número com a mensagem 'é par' ou 'é ímpar'.
9. Faça um programa que leia a altura e o sexo de uma pessoa, calcule e imprima seu peso ideal, utilizando as seguintes fórmulas:
Para homens: $(72.7 * h) - 58$
Para mulheres: $(62.1 * h) - 44.7$ (h = altura)
10. Faça um programa que receba quatro notas de um aluno, calcule e imprima a média aritmética das notas e a mensagem de aprovado para média superior ou igual a 7.0 ou a mensagem de reprovado para média inferior a 7.0.
11. Faça um programa que calcule e imprima o salário reajustado de um funcionário de acordo com a seguinte regra:
 - salários até R\$ 300,00, reajuste de 50%
 - salários maiores que R\$ 300,00, reajuste de 30%
- 12.A prefeitura do Rio de Janeiro abriu uma linha de crédito para os funcionários municipais. O valor máximo da prestação não poderá ultrapassar 30% do salário bruto. Fazer um algoritmo que permita entrar com o salário bruto e o valor da prestação e informar se o empréstimo pode ou não ser concedido.
13. Ler um número qualquer e exibir na tela uma mensagem indicando se ele é positivo, negativo ou nulo (zero). Se ele for positivo, exibir também a raiz quadrada deste número. Se ele for negativo você deve escrever uma mensagem dizendo 'Não é possível calcular a raiz deste número'.

14. Ler um número inteiro e exibir na tela a mensagem 'Par' se ele for um número par, ou 'Ímpar' se ele for um número ímpar.
15. Faça um programa que receba três notas de um aluno, calcule e mostre a média aritmética e as mensagens de acordo com a tabela abaixo. Para os alunos de exame, calcule e mostre a nota que deverá ser tirada no exame para aprovação, considerando que a média no exame é de 6,0.

MÉDIA	MENSAGEM
[0,0 a] 3,0	Reprovado
[3,0 a] 7,0	Exame
[7,0 a 10,0]	Aprovado

16. Faça um programa que receba três números e os mostre em ordem crescente.
17. Dados três valores X, Y e Z, verificar se eles podem ser os lados de um triângulo e, se forem, verificar se é um triângulo equilátero, isósceles ou escaleno. Se eles não formarem um triângulo informar ao usuário tal situação. Considerar que:

Condição para ser triângulo: O comprimento de cada lado de um triângulo é menor que a soma dos outros dois lados.

Chama-se triângulo equilátero o triângulo que tem os três lados iguais;
Chama-se triângulo isósceles o que tem o comprimento de dois lados iguais;
Chama-se triângulo escaleno o triângulo que têm os três lados diferentes.

18. Faça um programa que receba o código de origem de um produto e informe a sua procedência. A procedência obedece a seguinte tabela:

Código de origem	Procedência
1	Sul
2	Norte
3	Leste
4	Oeste
5 ou 6	Nordeste
7, 8 ou 9	Sudeste
10 até 20	Centro-oeste
21 até 30	Noroeste

Alternativa de Múltiplas Escolhas

É uma alternativa para quando o algoritmo tiver muitos **SEs**, deixando o programa com uma estrutura melhor.

Sintaxe:

Português Estruturado

Caso <nome da variável> **seja**

Alvo 1: comando 1;

Alvo 2: comando 2;

Alvo n: comando n;

Fim;

Ou

Caso <nome da variável> **seja**

Alvo 1: comando 1;

Alvo 2: comando 2;

Alvo 3: comando 4;

Senão

comando 5;

Fim;

Pascal

Case <nome da variável> **of**

Alvo 1: comando 1;

Alvo 2: comando 2;

Alvo n: comando n;

End;

Ou

Case <nome da variável> **of**

Alvo 1: comando 1;

Alvo 2: comando 2;

Alvo 3: comando 4;

Else

comando 5;

End;

Observações:

- No *Case* também pode fazer uso do *Else*;
- Antes do *Else* se usa ; (ponto e vírgula);
- O *Case* **não** tem *Begin*, mas, tem *End*;

- A variável que fará a verificação deverá ser declarada do tipo *Integer*, *Char* ou *Boolean*.

Exemplo: Este programa lê a idade de um nadador e informa a sua categoria seguindo as seguintes regras: Infantil A (5 - 7 anos), Infantil B (8 – 10 anos), Juvenil A (11 – 13 anos), Juvenil B (14 – 17 anos), Sênior (maiores de 18 anos).

Português Estruturado

```
Programa Exemplo;  
var  
    idade: integer;
```

```
Início  
Escreva('Idade do nadador: ');  
Leia (idade);
```

Caso idade seja

```
5..7: Escreva ('Categoria Infantil A');  
8..10:Escreva ('Categoria Infantil B');  
11..13:Escreva('Categoria JuvenilA');  
14..17:Escreva('Categoria JuvenilB');  
18..99:Escreva ('Categoria Sênior');
```

```
Senão  
Escreva('Idade inválida!');  
Fim;          //Fim do Caso  
Fim.
```

Pascal

```
Program Exemplo;  
var  
    idade: integer;
```

```
Begin  
Write ('Digite a idade do nadador: ');  
Read(idade);
```

Case idade of

```
5..7: Write ('Categoria Infantil A');  
8..10:Write ('Categoria Infantil B');  
11..13:Write('Categoria Juvenil A');  
14..17:Write ('Categoria Juvenil B');  
18..99:Write ('Categoria Sênior');
```

```
Else  
Write('Idade inválida!');  
End;          //End do Case  
End.
```


EXERCÍCIOS – Alternativa de Múltiplas Escolhas

1. Faça um programa que receba o código de origem de um produto e informe a sua procedência. (**Obs.:** este é o exercício 14 da estrutura de seleção **se**, mas, agora ele será feito utilizando o escolha caso.) A procedência obedece a seguinte tabela:

Código de origem	Procedência
1	Sul
2	Norte
3	Leste
4	Oeste
5 ou 6	Nordeste
7, 8 ou 9	Sudeste
10 até 20	Centro-oeste
21 até 30	Noroeste

2. Faça um programa que leia um número digitado pelo usuário e o escreva por extenso se ele estiver entre 1 e 5 inclusive, caso contrário mostrar uma mensagem: 'Opção de número inválida!'

3. Escrever um algoritmo que leia um peso na Terra e o número de um planeta e imprima o valor do seu peso neste planeta. A relação de planetas é dada a seguir juntamente com o valor das gravidade relativas à Terra:

<i>Nº</i>	<i>gravidade relativa</i>	<i>Planeta</i>
1	0.37	Mercúrio
2	0.88	Vênus
3	0.38	Marte
4	2.64	Júpiter
5	1.15	Saturno
6	1.17	Urano

Para calcular o peso no planeta escolhido use a seguinte fórmula:

Peso no Planeta = Peso / 10 * gravidade.

CAPÍTULO V

Estruturas de Repetição

Vamos supor que nossos algoritmos precisarão ser executados mais de uma vez e, para que não tenhamos que reescrever trechos idênticos que aumentariam consideravelmente o tamanho do programa, utilizaremos as estruturas de repetição.

Comando Enquanto (While - Do)

Esta estrutura é recomendada quando o número de repetições for desconhecido. Para que funcione corretamente é necessário observar:

- É necessário um teste (uma condição) para interromper a repetição, esta estrutura testa a condição e só executa o que está dentro dela se for verdadeira.
- A variável que testa a condição deverá ter seu valor atribuído no início do programa através de um comando de leitura ou de atribuição, antes da estrutura do enquanto e dentro da estrutura, como último comando.
- O Enquanto começa com Início (Begin) e termina com Fim (End;)
- Pode-se usar outras estruturas dentro do Enquanto (While), como: If, Case, etc.

Sintaxe:

Português Estruturado:

ENQUANTO condição FAÇA

INÍCIO

Comando 1;

Comando 2;

Comando n;

FIM;

Pascal:

WHILE condição DO

BEGIN

Comando 1;

Comando 2;

Comando n;

END;

Exemplo:

Faça um programa que leia vários números e imprima a metade de cada número, o programa termina quando o 0 (zero) for digitado:

PORTUGUÊS ESTRUTURADO:

```
PROGRAMA Enquanto1;
VAR
num: INTEIRO;
metade:REAL;
INÍCIO
ESCREVA ('Digite um número inteiro ');
LEIA (num);
ENQUANTO num <> 0 FAÇA
INÍCIO
metade <- num/2;
ESCREVA ('A metade de ', num, ' é ',
metade);
ESCREVA ('Digite outro número ou zero
para sair do programa: ');
LEIA (num);
FIM;
FIM.
```

PASCAL:

```
PROGRAM Enquanto1;
VAR
num: INTEGER;
metade:REAL;
BEGIN
WRITE ('Digite um número inteiro ');
READ(num);
WHILE num <> 0 DO
BEGIN
metade := num/2;
WRITELN ('A metade de ', num, ' é ',
metade:0:1);
WRITELN ('Digite outro número ou zero
para sair do programa: ');
READ(num);
END;
END.
```

Comando Repita (Repeat - Until)

Estrutura recomendada quando o número de repetições for desconhecido, sendo necessária uma chave (um teste) para interromper a repetição.

Sua diferença em relação ao enquanto é que ela testa a condição ao final, significando que ela executa o trecho pelo menos uma vez.

Muitas pessoas não entendem por que existe esta estrutura se a estrutura **Enquanto (While)** pode fazer o que ela faz. Na verdade tem muito programador que não faz uso dela.

Sintaxe:

Português Estruturado:

Repita

comandos;

Até condição;

Pascal

Repeat

comandos;

Until condição;

EXERCÍCIOS – Comando Enquanto (WHILE)

1. Entrar com vários números e imprimir quantos números foram digitados.
2. O IBGE fez uma pesquisa, ele quer saber quantos dentistas foram entrevistados. Fazer um programa que pergunte aos entrevistados qual a sua profissão e ao final diga quantas pessoas foram entrevistadas e quantas são dentistas (considerar dentista, Dentista e DENTISTA).
3. Dado um país A, com 5.000.000 de habitantes e uma taxa de natalidade de 3% ao ano, e um país B com 7.000.000 de habitantes e uma taxa de natalidade de 2% ao ano, calcular e imprimir o tempo necessário para que a população do país A ultrapasse a população do país B.

4. Uma pousada estipulou o preço da diária em R\$ 40,00 e mais uma taxa de serviços diários de:
 - R\$ 15,00, se o número de dias for menor que 10;
 - R\$ 8,00, se o número de dias for maior ou igual a 10.Criar um algoritmo que imprima o nome, o valor da conta de cada cliente e ao final o total arrecadado pela pousada.
5. Criar um algoritmo que entre com vários números inteiros e positivos e informe a quantidade de números múltiplos de 3 (três). O programa será encerrado quando o usuário digitar 0 (zero) ou menos.
6. Criar um algoritmo que entre com vários números inteiros e positivos e informe a média dos números múltiplos de 3 (três). O programa será encerrado quando o usuário digitar 0 (zero) ou menos.
7. Uma fábrica produz e vende vários produtos e para cada um deles tem-se o nome, quantidade produzida e quantidade vendida. Criar um algoritmo que mostre:
 - Para cada produto, nome, quantidade no estoque e uma mensagem se o produto tiver menos de 50 itens no estoque.
8. Faça um programa que leia a idade e a altura de várias pessoas. Calcule e informe a média das alturas das pessoas com mais de 50 anos. Para encerrar o programa digite zero para idade.
9. Faça um programa que leia vários números, quando o zero for digitado o programa será finalizado. Mostre no final desse programa a soma dos números positivos, a soma dos negativos e a soma total dos positivos e negativos juntos.
10. O GDF realizou uma pesquisa entre vários habitantes do DF, coletando dados sobre o rendimento familiar e o número de filhos de cada família. O GDF quer saber:
 - A média dos rendimentos da população;
 - Média do número de filhos.

11. Uma empresa decidiu fazer um levantamento em relação aos candidatos que se apresentarem para preenchimento de vagas no seu quadro de funcionários. Suponha que você seja o programador dessa empresa, criar um programa que leia para cada candidato a idade, o sexo e se tem experiência no serviço (S ou N). Para encerrar o programa, digite zero para idade. Calcule e escreva:
 - O número de candidatos do sexo feminino;
 - O número de candidatos do sexo masculino;
 - A idade média dos homens que já tem experiência no serviço.

12. Uma empresa classifica seus funcionários em três níveis de acordo com um índice de produtividade. São eles (1) Excelente, (2) Bom e (3) Regular. Cada nível acrescenta um abono ao salário base do funcionário, de acordo com a seguinte tabela:
 - Excelente 80% do salário base;
 - Bom 50% do salário base;
 - Regular 30% do salário base.O programa deve ler a matrícula do funcionário, seu salário base e seu nível de abono. Calcular e imprimir o salário a ser pago (salário a ser pago é = salário base + abono). O programa será encerrado quando for digitado 0 para matrícula.

13. Faça um programa que receba vários números positivos ou negativos, terminada por zero. O programa deve fornecer como saída, a soma dos números positivos e a soma dos números negativos.

14. Faça um programa que leia os dados de vários alunos, contendo o número da matrícula, as três notas e a frequência. Calcule e mostre: para cada aluno o número de matrícula, a nota final e a mensagem (aprovado ou reprovado); a maior e a menor nota da turma; o total de alunos reprovados;

15. Faça um programa que receba a idade, a altura e o peso de várias pessoas. Calcule e mostre:
 - A quantidade de pessoas com idade superior a 50 anos;
 - A média das alturas das pessoas com idade entre 10 e 20 anos;
 - A percentagem de pessoas com peso inferior a 40 quilos entre todas as pessoas analisadas.

16. Uma empresa deseja aumentar seus preços em 20%. Faça um programa que leia o código, o preço de custo de vários produtos e que calcule o novo preço de cada um deles. Calcule também a média de preços com e sem aumento. Mostre o código e o novo preço de cada produto e, no final, as médias. A entrada de dados deve terminar quando for recebido um código de produto menor ou igual a zero.
17. Faça um programa que apresente um menu de opções para cálculo das seguintes operações entre dois números: adição, subtração, multiplicação e divisão. O programa deve permitir a escolha da operação desejada, a entrada dos números, a exibição do resultado e a volta ao menu de opções. O programa só termina quando for escolhida a opção de saída.

Comando Para (For)

Esta estrutura de repetição é utilizada quando se sabe o número de vezes em que um trecho do programa deve ser repetido.

Sintaxe:

Português Estruturado

```
PARA a<-valor inicial ATÉ valor final FAÇA
  INÍCIO
      Comando 1;
      Comando 2;
  FIM;
```

PASCAL

```
FOR a := valor inicial ATÉ valor final DO
  INÍCIO
      Comando 1;
      Comando 2;
  END;
```

Observações:

1. O identificador (a variável “a” no exemplo anterior) tem que ser declarada do tipo inteiro (Integer).
2. A variável que controla a repetição poderá ser impressa se precisarmos dela para numerar uma lista, posicionar, etc.
3. A variável que controla a repetição **jamais** aparecerá num comando de **leitura** dentro do bloco de repetição.

Exemplo:

Algoritmo para escrever os números de 1 até 10:

Português Estruturado

```
Programa para1;  
Var  
    I : Inteiro;  
Início  
    Para i <- 1 até 10 faça  
        Início  
            Escreva (' - ', i);  
        fim;  
Fim.
```

PASCAL

```
Program para1;  
Var  
    I : Integer;  
Begin  
    For i := 1 to 10 do  
        Begin  
            Write (' - ', i);  
        End;  
End.
```

EXERCÍCIOS – Comando Para (For)

1. Imprimir todos os números de 100 até 1.
2. Criar um algoritmo que imprima todos os números pares no intervalo de 1 a 100.
3. Criar um algoritmo que entre com cinco números e imprima o quadrado de cada número.

4. Entrar com 10 números e imprimir a metade de cada número.
5. Criar um algoritmo que imprima todos os números de 1 até 100 e a soma deles.
6. Escreva um programa que receba a idade de 10 pessoas, calcule e imprima a quantidade de pessoas maiores de idade (idade \geq 18 anos).
7. Entrar com nome, idade e sexo de 10 pessoas. Imprimir o nome se a pessoa for do sexo masculino e tiver mais de 21 anos.
8. Em uma eleição presidencial, existem três candidatos. Os votos são informados através de código. Os códigos utilizados são:
1, 2, 3 \longrightarrow votos para os respectivos candidatos;
0 \longrightarrow votos em branco
outros códigos \longrightarrow votos nulos
Escreva um programa que calcule e imprima:
- total de votos para cada candidato;
- total de votos nulos;
- total de votos em branco.
9. Criar um algoritmo que entre com quatro notas de cada aluno de uma turma de 20 alunos e mostre:
- A média de cada aluno;
- A média da turma;
- Quantos alunos foram aprovados (média 7.0)
10. Imprimir as tabuadas de multiplicar de 1 até 10.
11. Criar um algoritmo que deixe escolher qual a tabuada de multiplicar que se deseja imprimir

12. Um empresa está fazendo a estatística de seus funcionários, ela precisa saber quantas funcionárias têm com mais de 40 anos para encaminhá-las para exames de mamografia. Fazer um programa que leia o nome, a idade e o sexo de seus 100 funcionários e imprima o nome se for do sexo feminino e tiver mais de 40 anos.
13. Faça um programa que receba a idade de 10 pessoas. O programa deve calcular e mostrar a quantidade de pessoas com idade maior que 18 anos.
14. Faça um programa que mostre a tabuada de multiplicação (de 1 a 10) para os 6 primeiros números primos. Ao mudar de uma base para outra o programa deve mostrar uma mensagem ao usuário e aguardar que alguma tecla seja pressionada para então montar a tabuada para a próxima base.
15. Faça um programa que receba a idade de 15 pessoas, calcule e mostre a quantidade de pessoas em cada faixa etária.

Faixa etária	Idade
1	Até 15 anos
2	De 16 a 30 anos
3	De 31 a 45 anos
4	De 46 a 60 anos
5	Acima de 61 anos

16. Faça um programa que leia uma série de números positivos inteiros. Calcule e mostre o maior número digitado. A entrada de dados deve terminar quando um número negativo foi digitado.
17. Faça um programa que receba um número, calcule e mostre o fatorial desse número. Sabe-se que: $n! = 1*2*3*...*(n-1)*n$;
 $0! = 1$, por definição.

ANEXO I

Questões de Raciocínio Lógico:

1. Preencha o quadro a seguir de tal maneira que a soma dos números que ficam sobre uma linha, ou sobre uma coluna, ou sobre uma diagonal, dê sempre 15 e todos os números têm de ser diferentes:

2. Entram num restaurante para jantar três mulheres, cada uma com duas filhas. Só existiam 7 lugares. Nenhuma ficou de pé. Como isso é possível?
3. Tenho 3 camisas: A, B e C. Uma é VERDE, uma é BRANCA e outra é AZUL, não necessariamente nesta ordem. Sabe-se somanete que a a camisa A é VERDE, B não É VERDE e C Não é AZUL. Quais as cores de A, B, e C, nessa ordem?
4. Dentro de uma caixa fechada, há uma bola branca e uma bola preta. Numa segunda caixa fechada, há duas bolas brancas e, numa terceira caixa fechada, há

duas bolas pretas. Cada caixa possui uma etiqueta indicando o conteúdo das caixas, mas alguém misturou as três etiquetas de modo que **todas** as etiquetas estão erradas. Você seria capaz de escolher apenas uma das seis bolas de modo tal que, olhando a sua cor, você possa dizer o conteúdo de cada uma das caixas?

ANEXO II

Transferência de Comandos: Português Estruturado para Pascal

Veja como seria a transferência de alguns comandos de Português Estruturado para Pascal:

OBS: Os dados da tabela estão em ordem alfabética.

Português Estruturado	Pascal
← (atribuição)	:=
Algoritmo ou Programa	Program
Cadeia (de caracteres) ou Literal	String
Caracter	Char
E	And
Enquanto n <> 0 faça	While n <> 0 do
Escreva ('.....');	Write ('.....');
Fim	End
Início	Begin
Inteiro	Integer
Leia (nome da variável);	Read (nome da variável);
Nome:arranjo [1..10] de cadeia (ou string);	Nome: array [1..10] of string;
Ou	Or
Para i←1 até 10 faça	For i:= 1 to 10 do
Se x > 10 então	If x > 10 then

Professora Lucélia Alves de Oliveira

Senão	Else
Var ou Variáveis	Var