

LÓGICA DE PROGRAMAÇÃO

INSTRUTOR: TIAGO DA ROSA VALÉRIO

E-MAIL: tiagodev.valerio@gmail.com

ALGORITMOS

Algoritmo é uma sequência lógica de instruções que podem ser executadas. É importante ressaltar que, qualquer tarefa que siga determinado padrão, pode ser descrita por um algoritmo, como por exemplo: CRIAR UMA CONTA NO INSTAGRAM ou CALCULAR O SALDO FINANCEIRO DE UM ESTOQUE.

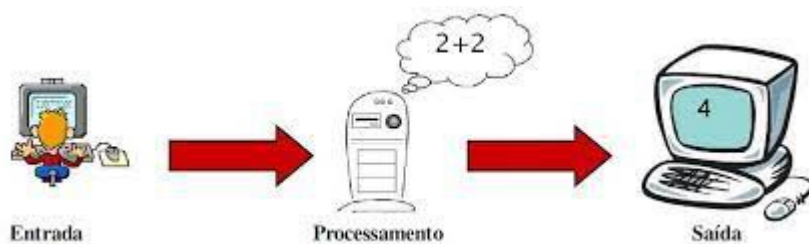
Entretanto ao montar um algoritmo, precisamos primeiro dividir o problema apresentado em três fases fundamentais.

Onde temos:

ENTRADA: São os dados de entrada do algoritmo.

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

SAÍDA: São os dados já processados.



EXEMPLO DE ALGORITMO

Imagine o seguinte problema: calcular a média final dos alunos da 3ª série. Os alunos realizaram quatro provas: P1, P2, P3 e P4. Onde:

$$\text{MÉDIA} = (P1 + P2 + P3 + P4) / 4$$

Para montar o algoritmo proposto, faremos três perguntas:

- 1) Qual são os dados de Entrada ?
 - 2) Qual será o processamento a ser utilizado ?
 - 3) Quais serão os dados de saída ?
- 1) Os dados de entrada são P1, P2, P3 e P4.

- 2) O procedimento será somar os dados de entrada e dividir por 4.
- 3) Os dados de saída serão a média final.

EXERCÍCIOS

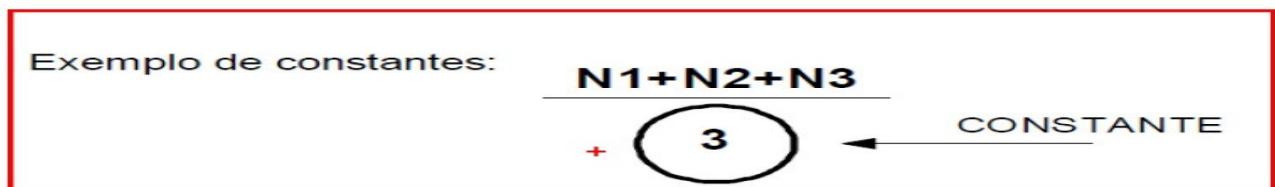
- 1) Defina os dados de entrada, processamento, e saída para os seguintes problemas:
 - a) Manter a temperatura da sala entre 18 e 22 graus.
 - b) Calcular frete de uma viagem.

CONSTANTES e VARIÁVEIS

Variáveis e constantes são os elementos básicos que um programa manipula. Uma variável é um espaço reservado na memória do computador para armazenar um tipo de dado determinado.

Variáveis devem receber nomes para poderem ser referenciadas e modificadas quando necessário. Um programa deve conter declarações que especificam de que tipo são as variáveis que ele utilizará e as vezes um valor inicial. Estes tipos podem ser por exemplo: inteiros, decimais, caracteres e booleanos. As expressões combinam variáveis e constantes para calcular novos valores.

Constante é um determinado valor fixo que não se modifica ao longo do tempo, durante a execução de um programa. Conforme o seu tipo, a constante é classificada como sendo decimal, inteira, booleana e caracter.



TIPOS DE DADOS

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

Exemplos:

10 número inteiro positivo

-10 número inteiro negativo

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

Exemplos:

25.03 número real positivo com duas casas decimais

-10.5 número real negativo com uma casa decimal

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.

Exemplos:

“Maria” String de comprimento 5

“123” String de comprimento 3

“A” String de comprimento 1

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.

EXERCÍCIOS

1) Identifique os tipos de dados, sendo (I) Inteiro, (D) Decimal, (A) Alfanumérico e (B) Boleana:

- | | | | |
|-------------|---------------|----------------|-----------------|
| () 1.23 | () ‘tomates’ | () Verdadeiro | () $1 + 2.1$ |
| () ‘Falso’ | () ‘1’ | () $3 - 2$ | () ‘ $1 + 1$ ’ |

2) Identifique o tipo de dados para cada uma das situações abaixo:

- a) Nome do aluno.
- b) Saldo da conta bancária.
- c) Idade do cliente do mercado.
- d) Nota do aluno.

OPERADORES

Os operadores são meios pelo qual incrementamos, decrementamos, comparamos e avaliamos dados dentro do computador. Temos três tipos de operadores:

- Operadores Aritméticos
- Operadores Relacionais
- Operadores Lógicos

OPERADORES ARITMÉTICOS

Os operadores aritméticos são os utilizados para obter resultados numéricos. Além da adição, subtração, multiplicação e divisão, podem utilizar também o operador para exponenciação. Os símbolos para os operadores aritméticos são:

OPERAÇÃO	SIMBOLO
Adição	+
Subtração	-
Multiplicação	*
Divisão	/
Exponenciação	**

Hierarquia das Operações Aritméticas

1 ° () Parênteses

2 ° Exponenciação

3 ° Multiplicação, divisão (o que aparecer primeiro)

4 ° + ou – (o que aparecer primeiro)

Exemplo:

$$\text{TOTAL} = \text{PRECO} * \text{QUANTIDADE}$$

$$1 + 7 * 2 ** 2 - 1 = 28$$

$$3 * (1 - 2) + 4 * 2 = 5$$

EXERCÍCIOS

1 – Resolva as expressões matemáticas:

- a) $1 + 9 * 3$
- b) $2 / 1 * (3 + 1)$
- c) $2 ** 3 + (2 + (6 / 3))$
- d) $3 + 4 ** 2$

VISUAL G

O VISUAL G é uma ferramenta criada para quem está iniciando seus aprendizados em Lógica Computacional ou Construção de Algoritmos Computacionais. Nele você poderá digitar seus comandos em português e ver seu programa funcionando.

Supondo que temos a seguinte problemática.

João tem um aviário e precisa do valor de suas vendas pela quantidade de ovos vendidos para cada cliente. Neste caso ele vai informar ao sistema a quantidade de ovos e o valor da unidade, sendo demonstrado o valor de venda, desta forma teremos o seguinte algoritmo:

```

1 Algoritmo "vendaOvos"
2 // Disciplina   : [Linguagem e Lógica de Programação]
3 // Professor    : Tiago da Rosa Valério
4 // Descrição    : Venda de Ovos
5 // Autor(a)     : Tiago da Rosa Valério
6 // Data atual   : 20/05/2021
7 Var
8 // Seção de Declarações das variáveis
9 qtdOvos : inteiro
10 valorUnid, valorVenda : real
11
12 Inicio
13 // Seção de Comandos, procedimento, funções, operadores, etc...
14 Escreva("Informe a quantidade de ovos vendidos:")
15 Leia(qtdOvos)
16 Escreva("Informe o valor da unidade R$:")
17 Leia(valorUnid)
18 valorVenda <- qtdOvos * valorUnid;
19 Escreva("Valor da Venda R$:", valorVenda)
20
21 Fimalgoritmo

```

Para resolução deste problema inicialmente foi verificado quais as entradas que seriam informadas pelo usuário, no caso a quantidade de ovos e o valor da unidade. Depois foi verificado o processamento a ser realizado (quantidade de ovos * preço da unidade) e por último demonstrado saída para o usuário final, que seria o preço da venda.

EXERCÍCIOS

1) Construa um algoritmo para solucionar a seguinte situação :

- Leia a cotação do dólar
- Leia um valor em dólares
- Converta esse valor para Real
- Mostre o resultado

OPERADORES RELACIONAIS

Os operadores relacionais são utilizados para comparar String de caracteres e números. Os valores a serem comparados podem ser constantes ou variáveis.

Estes operadores sempre retornam valores lógicos (verdadeiro ou falso). Para estabelecer prioridades no que diz respeito a qual operação executar primeiro, utilize os parênteses.

Os operadores relacionais são:

DESCRIÇÃO	SIMBOLO
Igual a	=
Diferente de	<> ou #
Maior que	>
Menor que	<
Maior ou Igual a	>=
Menor ou Igual	<=

Exemplo:

Tendo duas variáveis A = 5 e B = 3. Os resultados das expressões seriam:

OPERAÇÃO	SIMBOLO
A = B	False
A <> B	True
A > B	True
A < B	False
A >= B	True
A <= B	False

OPERADORES LÓGICOS

Os operadores lógicos servem para combinar resultados de expressões, retornando se o resultado final é verdadeiro ou falso.

Os operadores lógicos são:

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

A tabela abaixo mostra todos os valores possíveis criados pelos três operadores lógicos (AND, OR e NOT).

1º Valor	Operador	2º Valor	Resultado
T	AND	T	T
T	AND	F	F
F	AND	T	F
F	AND	F	F
T	OR	T	T
T	OR	F	T
F	OR	T	T
F	OR	F	F
T	NOT		F
F	NOT		T

Exemplos:

Suponha que temos três variáveis $A = 5$, $B = 8$ e $C = 1$. Os resultados das expressões seriam:

Expressões			Resultado
$A = B$	AND	$B > C$	False
$A < > B$	OR	$B < C$	True
$A > B$	NOT		True
$A < B$	AND	$B > C$	True
$A \geq B$	OR	$B = C$	False
$A \leq B$	NOT		False

EXERCÍCIOS

- 1) Tendo as variáveis SALARIO, IR e SALLIQ, e considerando os valores abaixo. Informe se as expressões são verdadeiras ou falsas.

SALARIO	IR	SALLIQ	EXPRESSÃO	V ou F
100	0	100	$(SALLIQ \geq 100)$	
200	10	190	$(SALLIQ < 190)$	
300	15	285	$SALLIQ = SALARIO - IR$	

- 2) Sabendo que $A=3$, $B=7$, $C=4$ e $D=10$, informe se as expressões abaixo são verdadeiras ou falsas.

- a) $(A+C) > B$ ()
- b) $B \geq (A + 2)$ ()
- c) $C = (B - A)$ ()
- d) $(C+A) > B$ ()
- e) $(A > C) \text{ AND } (C \leq D)$ ()
- f) $(A \geq C) \text{ OR } (D \geq C)$ ()

COMANDOS DE DECISÃO

Os comandos de decisão ou desvio fazem parte das técnicas de programação que conduzem as estruturas de programas que não são totalmente sequenciais. Com as instruções de SALTO ou DESVIO pode-se fazer com que o programa proceda de uma ou outra maneira, de acordo com as decisões lógicas tomadas em função dos dados ou resultados anteriores. As principais estruturas de decisão são: “Se Então”, “Se então Senão” e “Caso Selecione”.

SE ENTÃO / IF ... THEN

A estrutura de decisão “SE/IF” normalmente vem acompanhada de um comando, ou seja, se determinada condição for satisfeita pelo comando SE/IF então execute determinado comando.

Imagine um algoritmo que determinado aluno somente estará aprovado se sua média for maior ou igual a 5.0, veja no exemplo de algoritmo como ficaria.

SE MEDIA >= 5.0 ENTÃO

ESCREVA(ALUNO APROVADO)

FIM SE

Com estas estruturas poderemos executar determinadas ações conforme a necessidade. Podemos imaginar o que ocorreu no Japão. Vamos construir uma algoritmo que seja informado o índice de abalos sísmicos por um técnico e se chegar a determinada escala que seja gerado alerta máximo.

```
1 Algoritmo "ondaSismica"
2 // Disciplina   : [Linguagem e Lógica de Programação]
3 // Professor    : Tiago da Rosa Valério
4 // Descrição    : Verifica perigo de terremotos
5 // Autor(a)     : Tiago da Rosa Valério
6 // Data atual   : 20/05/2021
7 Var
8 // Seção de Declarações das variáveis
9 indiceSismico : real
10
11 Inicio
12 // Seção de Comandos, procedimento, funções, operadores, etc...
13 Escreva("Informe o índice de atividades sísmicas: ")
14 Leia(indiceSismico)
15 se indiceSismico >= 10 então
16     Escreva("Alerta Máximo! Perigo de tsunami!!!")
17 fimse
18
19 Fimalgoritmo
```

SE ENTÃO SENÃO / IF ... THEN ... ELSE

A estrutura de decisão “SE/ENTÃO/SENÃO”, funciona exatamente como a estrutura “SE”, com apenas uma diferença, em “SE” somente podemos executar comandos caso a condição seja verdadeira, diferente de “SE/SENÃO” pois sempre um comando será executado independente da condição, ou seja, caso a condição seja “verdadeira” o comando da condição será executado, caso contrário o comando da condição “falsa” será executado.

Em algoritmo ficaria assim:

```
SE MÉDIA >= 5.0 ENTÃO
    ESCREVA(ALUNO APROVADO)
SENÃO
    ESCREVA(ALUNO REPROVADO)
FIM SE
```

No exemplo acima está sendo executada uma condição que, se for verdadeira, executa o comando “APROVADO”, caso contrário executa o segundo comando “REPROVADO”. Podemos também dentro de uma mesma condição testar outras condições.

Vamos melhorar nosso exemplo anterior, agora vamos adicionar uma mensagem, caso o índice sísmico não representa problemas:

```
1 Algoritmo "ondaSismica"
2 // Disciplina   : [Linguagem e Lógica de Programação]
3 // Professor    : Tiago da Rosa Valério
4 // Descrição    : Verifica perigo de terremotos
5 // Autor(a)     : Tiago da Rosa Valério
6 // Data atual   : 20/05/2021
7 Var
8 // Seção de Declarações das variáveis
9 indiceSismico : real
10
11 Inicio
12 // Seção de Comandos, procedimento, funções, operadores, etc...
13 Escreva("Informe o índice de atividades sísmicas: ")
14 Leia(indiceSismico)
15 se indiceSismico >= 10 então
16     Escreva("Alerta Máximo! Perigo de tsunami!!!")
17 senao
18     Escreva("Sem riscos! Podem estudar programação!!!")
19 fimse
20
21 Fimalgoritmo
```

EXERCÍCIOS

1) Implemente um algoritmo para calcular o imposto de renda de um trabalhador. No caso o sistema deverá solicitar a informação do salário do funcionário. Se o salário for superior a R\$: 3000,00 deve pagar 15% de imposto para os demais valores deve ser 2%.

2) Tendo como dados de entrada a altura e o sexo de uma pessoa, construa um algoritmo que calcule seu peso ideal, utilizando as seguintes fórmulas:

Para homens: $(72.7 * h) - 58$

Para mulheres: $(62.1 * h) - 44.7$ (h = altura)

COMANDOS DE REPETIÇÃO

Utilizamos os comandos de repetição quando desejamos que um determinado conjunto de instruções ou comandos sejam executados um número definido ou indefinido de vezes, ou enquanto um determinado estado de coisas prevalecer ou até que seja alcançado.

ENQUANTO X, PROCESSAR (Do While ... Loop)

Neste caso, o bloco de operações será executado enquanto a condição x for verdadeira. O teste da condição será sempre realizado antes de qualquer operação.

Enquanto a condição for verdadeira o processo se repete. Podemos solicitar que sejam informados 5 usuários com a seguinte estrutura.

```
1 Algoritmo "exemplo"
2
3 Var
4 x : inteiro
5 usuario : caracter
6 Inicio
7 x <- 0
8 enquanto x < 5 faça
9     Escreva("Informe o usuário:")
10    Leia(usuario)
11    x <- x + 1
12 fimenquanto
13
14 Fimalgoritmo
```

Esta estrutura de repetição, executa os comandos enquanto a condição for verdadeira, desta forma será executado até a variável x ter o valor 5.

EXERCÍCIOS

- 1) Faça um algoritmo que receba 10 numeros e demonstre se cada número é positivo ou negativo.

PROCESSAR ... ATÉ QUE X (Do ... Loop Until)

Neste caso, executa-se primeiro o bloco de operações e somente depois é realizado o teste de condição. Se a condição for verdadeira, o fluxo do programa continua normalmente. Caso contrário é processado novamente os comandos antes do teste da condição. Desta forma podemos criar um algoritmo para receber números até que seja informado 0(zero), demonstrada se cada número informado é par ou impar.

```
1 Algoritmo "exemplo"
2
3 Var
4 x : inteiro
5 Inicio
6 repita
7     Escreva("Informe um número: ")
8     Leia(x)
9     se x MOD 2 = 0 entao
10        Escreval(x, " é par")
11    senao
12        Escreval(x, " é impar")
13    fimse
14 ate x = 0
15
16 Fimalgoritmo
```

Nesta estrutura os comandos são executados e apenas no final temos a condição, sendo que a condição de parada é a variável x com o valor 0(zero).

Exercícios

1) Construa um algoritmo que recebe números, sendo que a sua condição de para é valor 0. Depois deverá ser demonstrada a soma dos números informados pelos usuários.

PARA X ATE Y FAÇA (FOR ... NEXT)

Esta estrutura de repetição é a mais utilizada em linguagens de programação, a principal diferença é na sua declaração que indicamos o valor inicial da variável, sua condição de parada e como a mesma será implementada. No exemplo abaixo, iremos solicitar o nome e idade de 4(pessoas) e iremos demonstrar se são maiores de idade.

```
1 Algoritmo "exemplo"
2
3 Var
4 x, idade : inteiro
5 nome : caracter
6 Inicio
7 para x de 1 ate 4 passo 1 faca
8     Escreva("Informe o seu nome: ")
9     Leia(nome)
10    Escreva("Informe a sua idade: ")
11    Leia(idade)
12    se idade >= 18 entao
13        Escreval(nome, " é maior de idade!")
14    senao
15        Escreval(nome, " é menor de idade!")
16    fimse
17 fimpara
18
19 Fimalgoritmo |
```

Esta estrutura é utilizada pela facilidade de leitura, pois a estrutura será executando com o valor inicial 1 para a variável x, para cada iteração será somado 1 na variável x e quando esta variável chegar ao valor 4 será concluída a estrutura de repetição.

EXERCÍCIOS

- 1) Escrever um algoritmo que leia um número qualquer e demonstre a sua tabuada.