

Algoritmos e Programação

Ricardo Argenton Ramos

**Baseado nos slides do
professor Jadsonlee
da Silva Sá**

**Universidade Federal do Vale do São Francisco - UNIVASF
Colegiado de Engenharia de Computação - CECOMP**

Criando um Algoritmo

- ✓ Os passos necessários para a construção de um algoritmo:
 - ler atentamente o enunciado do problema, compreendendo-o e destacando os pontos mais importantes;
 - definir os dados de entrada, ou seja, quais dados serão fornecidos;
 - definir os dados de saída, ou seja, quais dados serão gerados depois do processamento;
 - definir o processamento, ou seja, quais cálculos serão efetuados e quais as restrições para esses cálculos. O processamento é responsável pela obtenção dos dados de saída com base nos dados de entrada;
 - definir as variáveis necessárias para armazenar as entradas e efetuar o processamento;
 - elaborar o algoritmo;

Pseudocódigo

- ✓ **Método de representação de algoritmos** → Utilizaremos pseudocódigo.
- ✓ **Pseudocódigo** → Técnica textual de representação de um algoritmo - Também conhecida como Português Estruturado ou Portugol.
- ✓ Técnica é baseada em uma PDL (*Program Design Language*), que é uma linguagem genérica na qual é possível representar um algoritmo de forma semelhante à das linguagens de programação.

Estrutura de um Algoritmo

NOME DO ALGORITMO

VAR

declaração de variáveis

INICIO DO ALGORITMO

bloco de comandos

FIM DO ALGORITMO

```
algoritmo "soma dois numeros"
```

```
var
```

```
    n1, n2, d: inteiro
```

```
inicio
```

```
    escreval("Digite dois numeros")
```

```
    leia(n1)
```

```
    leia(n2)
```

```
    d<-n1+n2
```

```
    escreva("A soma eh igual a ", d)
```

```
fimalgoritmo
```

Comandos Visualg

✓ Declaração de variáveis.

Var

n1, n2: inteiro

d: real

k: caractere

sinal: logico

✓ Comandos de atribuição.

n1 <- 3

n2 <- n1+ 4

k <- "teste de atribuição"

sinal <- verdadeiro

Comandos Visualg

✓ Entrada de dados.

Sintaxe → leia (<lista de variáveis>)

Exemplos:

leia (n1) - O valor digitado será armazenado na variável n1.

leia (k) - Um ou vários caracteres digitados serão armazenados na variável k (Definida como caractere).

✓ No caso de utilizar leia (<lista de variáveis>), será respeitada a ordem da lista de variáveis, da esquerda para direita.

- **Exemplo:** leia (n1, k, n2)

Comandos Visualg

✓ Saída de dados.

Sintaxe → escreva (<lista-de-variáveis>)

escreval (<lista-de-variáveis>)

Obs: escreval → imprime a variável na tela e o cursor vai para uma nova linha.

Exemplos:

escreva (n1) - Será mostrado na tela o conteúdo da variável n1.

escreva ("O texto digitado foi ", k) - Será mostrado o texto entre "" e depois o conteúdo da variável k.

Comandos Visualg

✓ Saída de dados.

- É possível especificar o número de colunas da tela onde se deseja escrever um determinado valor.
- **Exemplo:** considerando uma variável inteira x , o comando escreva ($x:5$) escreve o valor da variável x em 5 colunas, alinhado-o à direita.
- **Exemplo:** considerando y como uma variável real, o comando escreva ($y:6:2$), escreve seu valor em 6 colunas colocando 2 casas decimais.

Exercícios

- 1) Escreva um programa que receba dois números e calcule a média deles.
- 2) Escreva um programa que receba três números e depois peça o valor dos pesos e depois calcule a média ponderada dos números.
- 3) Faça um programa que leia uma temperatura em graus Celsius e calcule o correspondente em Fahrenheit. Sabendo que: $F = (180 * (C + 32)) / 100$

Estruturas de Controle de Fluxo

- ✓ Os algoritmos desenvolvidos até o momento constituem uma seqüência de ações que sempre são executadas em sua totalidade indiferente do valor da entrada de dados.
- ✓ Para a resolução de determinados problemas ou para a execução de determinadas tarefas é necessária a realização de um conjunto distinto de ações e este conjunto é definido com base na análise da entrada de dados.

Estruturas de Controle de Fluxo

- ✓ **Exemplo:** um algoritmo capaz de efetuar o cálculo do imposto de renda devido por um determinado contribuinte. Neste caso dependendo da quantidade de dependentes, do valor de sua renda e outros fatores o cálculo será feito de formas distintas.

Instrução Condicional

- ✓ Considere um problema que exija uma decisão.
 - Tomemos como exemplo uma divisão, onde haja a necessidade de que o algoritmo verifique se o divisor é igual ou diferente de zero. Se for igual não é possível dividir. Se for diferente é possível dividir.

- ✓ Sintaxe:

```
se (<expressão-lógica>) então  
    <sequência de comandos>  
senão  
    <sequência de comandos>  
fimse
```

Instrução Condicional

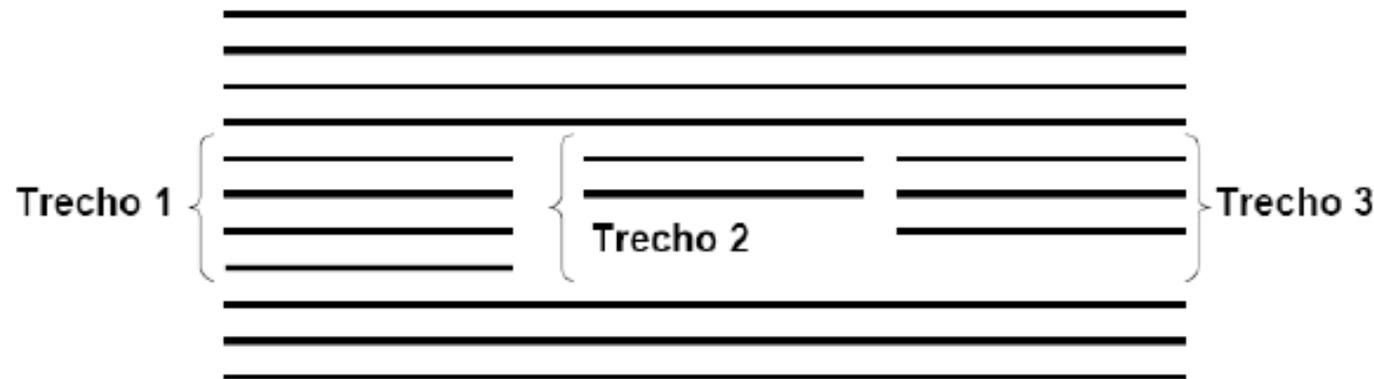
✓ Exemplo:

```
algoritmo "Divisão"

var
    n1, n2: inteiro
    resultado: real
inicio
    escreva ("Digite o dividendo ==> ")
    leia (n1)
    escreva ("Digite o divisor ==> ")
    leia (n2)
    se (n2=0) entao
        escreva ("impossivel dividir por 0")
    senao
        resultado <- n1/n2
        escreva ("O resultado eh ==>", resultado)
    fimse
fimalgoritmo
```

Comando de Seleção Múltipla

- ✓ Em algumas situações ao chegarmos a uma determinada instrução de um algoritmo devemos selecionar um dentre alguns trechos a seguir, tendo como base para esta escolha um conjunto de valores.



- ✓ Para lidar com casos deste tipo foi criado o comando de seleção múltipla.

Comando de Seleção Múltipla

✓ Sintaxe:

escolha (<variável>)

caso <valor11>, <valor12>, ..., <valor1n>

<seqüência-de-comandos-1>

caso <valor21>, <valor22>, ..., <valor2m>

<seqüência-de-comandos-2>

...

outrocaso

<seqüência-de-comandos-extra>

fimescolha

Comando de Seleção Múltipla

```
algoritmo " Exemplo Seleção Múltipla"  
var  
    time: caractere  
inicio  
    escreva ("Entre com o nome de um time de futebol: ")  
    leia (time)  
    escolha (time)  
        caso "Sport", "Santa Cruz", "Nautico", "Petrolina"  
            escreval ("É um time pernambucano.")  
        caso "Vitoria da Conquista", "Bahia de Feira",  
            "Camaçari", "Feirense"  
            escreval ("É um time baiano.")  
        outrocaso  
            escreval ("É de outro estado.")  
    fimescolha  
fimalgoritmo
```