



**TECNOLOGIA E INOVAÇÃO
EM PROL DA INDÚSTRIA**



Técnico em Informática

Programação de aplicativos – 140h

Prof^a: Francisleide Almeida

Pseudocódigo

- Esta forma de representação de algoritmos é rica em detalhes, como a definição dos tipos das variáveis usadas no algoritmo e, por assemelhar-se bastante à forma em que os programas são escritos, encontra muita aceitação.

Pseudocódigo

- A forma geral da representação de um algoritmo na forma de pseudocódigo é a seguinte:

Bibliotecas

```
<declaração_de_variáveis> //Variáveis globais  
<subalgoritmos>
```

Nome-do-algoritmo()

Início

```
<declaração_de_variáveis> //Variáveis Locais  
<corpo_do_algoritmo>
```

Fim.

Pseudocódigo

Bibliotecas

```
<declaração_de_variáveis> //Variáveis globais  
<subalgoritmos>
```

Nome-do-algoritmo()

Início

```
<declaração_de_variáveis> //Variáveis Locais  
<corpo_do_algoritmo>
```

Fim.

- **Bibliotecas** são funções escritas pelos fabricantes do compilador e já estão pré-compiladas.
- **<Nome_do_algoritmo>** é um nome simbólico dado ao algoritmo com a finalidade de distingui-lo dos demais.
- **<declaração-de-variáveis>** consiste em uma porção opcional onde são declaradas as variáveis globais usadas no algoritmo principal e, eventualmente, nos subalgoritmos com variáveis locais.
- **<subalgoritmos>** consiste de uma porção opcional do pseudocódigo onde são definidos os subalgoritmos (veremos adiante).
- **Início** e **Fim** são respectivamente as palavras que delimitam o início e o término do conjunto de instruções do corpo do algoritmo.

Ferramentas

- Editor:
 - Um programa de computador nada mais é do que um grupo de comandos logicamente dispostos para executar em uma determinada tarefa. Esses comandos são gravados em um *arquivo-texto* que é processado ao comando do usuário e passa então a executar cada um dos comandos que lá estão gravados.



Visual Studio Express 2015 for Windows
Desktop

visuALG



Texmaker 4.4.1

Sublime Text

Definição de variáveis

- Variáveis devem ser declaradas previamente para reserva de espaço na memória;

<nome_da_variável>:<tipo_da_variável>

- Pode se declaradas mais de uma variável na mesma linha, se forem do mesmo tipo, separando por vírgula.

Definição de variáveis

- Regras:
 - Máximo 30 caracteres para o nome da variável;
 - Todo nome deve ser composto apenas por letras, números e sublinhado
 - Deve começar com uma letra
 - Não pode ser igual a nenhuma palavra reservada, ou seja, palavras que já possuam um significado próprio para a linguagem como por exemplo, true, var ou if

Atividade

1. Na lista seguinte, assinale com **V** os nomes de variáveis válidos e com **I** os inválidos.

- | | | |
|---------------------------------|---------------------------------|---------------------------------------|
| <input type="checkbox"/> abc | <input type="checkbox"/> 3abc | <input type="checkbox"/> a |
| <input type="checkbox"/> 123a | <input type="checkbox"/> a?B | <input type="checkbox"/> acdl |
| <input type="checkbox"/> _ | <input type="checkbox"/> Aa | <input type="checkbox"/> 1 |
| <input type="checkbox"/> Al23 | <input type="checkbox"/> _1 | <input type="checkbox"/> AO123 |
| <input type="checkbox"/> al23 | <input type="checkbox"/> _al23 | <input type="checkbox"/> b312 |
| <input type="checkbox"/> AB CDE | <input type="checkbox"/> etc... | <input type="checkbox"/> guarda-chuva |

Atribuição

- Principal maneira de armazenar informação em uma variável.

Sua sintaxe é: `<nome_de_variável> ← <expressão>`

```
algoritmo "Exemplo01"  
  
var  
    preco_unit, preco_tot : real  
    quant : inteiro  
inicio  
    preco_unit <- 5.0  
    quant <- 10  
    preco_tot <- preco_unit * quant  
fimalgoritmo
```

Entrada/Saída de dados

- A entrada de dados foi criada para receber dados de um dispositivo, geralmente o teclado.

Ex:

leia x

- A saída de dados é o meio pelo qual, informações contidas na memória do computador são colocadas no dispositivo de saída.

Ex:

Escreva x

Ex:

```

algoritmo "Exemplo02"

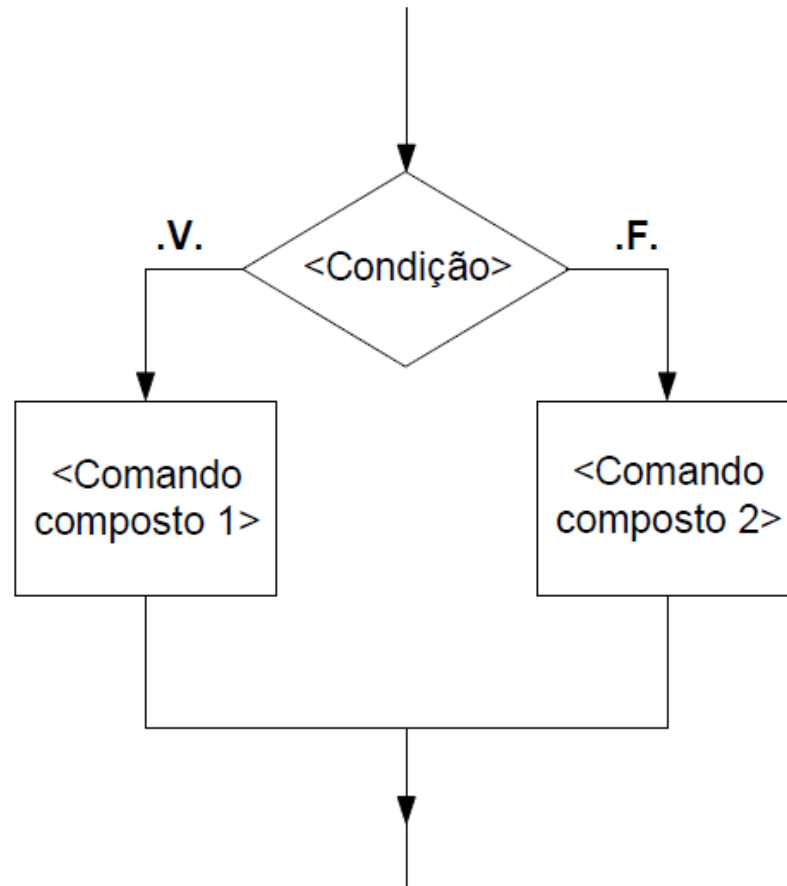
var
    preco_unit, preco_tot : real
    quant : inteiro
inicio
    Leia(preco_unit)
    Leia(quant)
    preco_tot <- preco_unit * quant
    Escreva("O preço total é: ", preco_tot)
fimalgoritmo
    
```


Estrutura de decisão

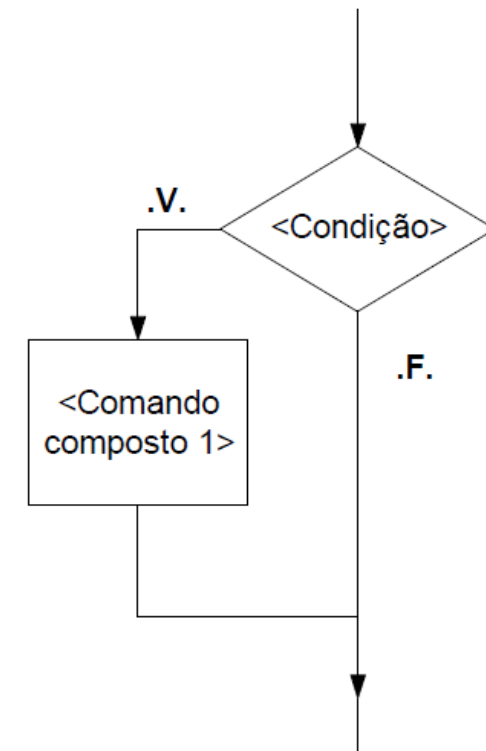
- Estrutura tipo Se:
 - Uma única condição (expressão lógica) é avaliada.

```
Se (<condição>) {  
    <Comando_composto_1>;  
} Senão {  
    <Comando_composto_2>;  
}
```

Estrutura “Se”

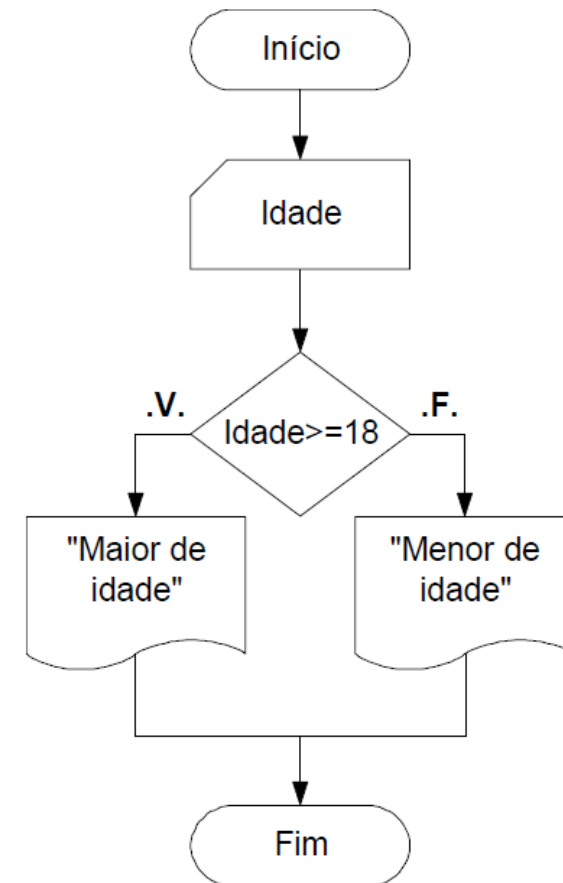


```
Se (<condição>)  
{  
    <Comando_composto_1>;  
}
```



Exemplo “Se”

```
algoritmo "Idade"  
  
var  
    idade : inteiro  
inicio  
    Leia(idade)  
    Se (idade >= 18) entao  
        Escreva("Maior de idade")  
    senao  
        Escreva("Menor de idade")  
    fimse  
fimalgoritmo
```



Exemplo 2

- Crie um algoritmo no visualg para verificar, dentre 2 números qual o maior deles.

Exemplo 2 - Solução

```
algoritmo "maiornumero"  
  
var  
num1, num2: inteiro  
  
inicio  
// Seção de Comandos  
escreva("Digite o primeiro número: ")  
leia(num1)  
escreva("Digite o segundo número: ")  
leia(num2)  
  
se num1 > num2 entao  
    escreva("O primeiro número ",num1," é maior que o segundo",num2)  
senao  
    escreva("O segundo número",num2," é maior que o primeiro",num1)  
fimse  
  
fimalgoritmo
```


Se aninhando

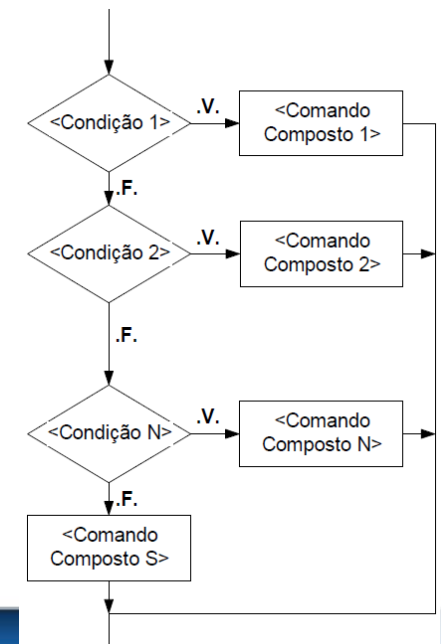
- Verificam vários casos inserindo umas estruturas **Se/Senão** em outras.

```
Se (grau >= 90)
    Escreva ("A");
Senão
    Se (grau >= 80)
        Escreva ("B");
    Senão
        Se (grau >= 70)
            Escreva ("C");
        Senão
            Se (grau >= 60)
                Escreva ("D");
            Senão
                Escreva ("F");
```

Decisão do Tipo Escolha

- Este tipo de estrutura é uma generalização da estrutura **Se**, onde somente uma condição era avaliada e dois caminhos podiam ser seguidos.

```
Escolha (<variável>) {  
  Caso <Valor_1> : <Comando_composto_1>; Pare;  
  Caso <Valor_2> : <Comando_composto_2>; Pare;  
  .....  
  Caso <Valor_N> : <Comando_composto_N>; Pare;  
  Senão <Comando_composto_S>;  
}
```



Exemplo - Caso

- Escrever um algoritmo que solicite ao usuário um número inteiro. Então, o algoritmo deve verificar se o valor informado foi 0, 1 ou 2. Caso não seja nenhum dos valores 0, 1 ou 2, o algoritmo deve retornar a mensagem: “Opção Inválida”.

Solução

```
algoritmo "Escolhe"  
var  
    numero:inteiro  
inicio  
    escreva("digite um número inteiro: ")  
    leia(numero)  
    escolha numero  
        caso 0  
            escreval("Opção escolhida 0")  
        caso 1  
            escreval("Opção escolhida 1")  
        caso 2  
            escreval("Opção escolhida 2")  
        outrocaso  
            escreval("Opção inválida")  
    fimescolha  
fimalgoritmo
```

Atividades

1 - Escreva um algoritmo em Fluxograma e em Pseudocódigo que leia 3 números informados pelo usuário e apresente-os ordenados.

2 - Faça um algoritmo (pseudocódigo e fluxograma) que lê o nome de um produto, o preço e a quantidade comprada. Escreva o nome do produto comprado e o valor total a ser pago, considerando que são oferecidos descontos pelo número de unidades compradas, segundo a tabela abaixo:

- a) Até 10 unidades: valor total
- b) De 11 a 20 unidades: 10% de desconto
- c) De 21 a 50 unidades: 20% de desconto
- d) Acima de 50 unidades: 25% de desconto