



# **25070 - Construção de Algoritmos e Programação**

## **Aula 4 - 27/03/2019**

**Joice Otsuka**

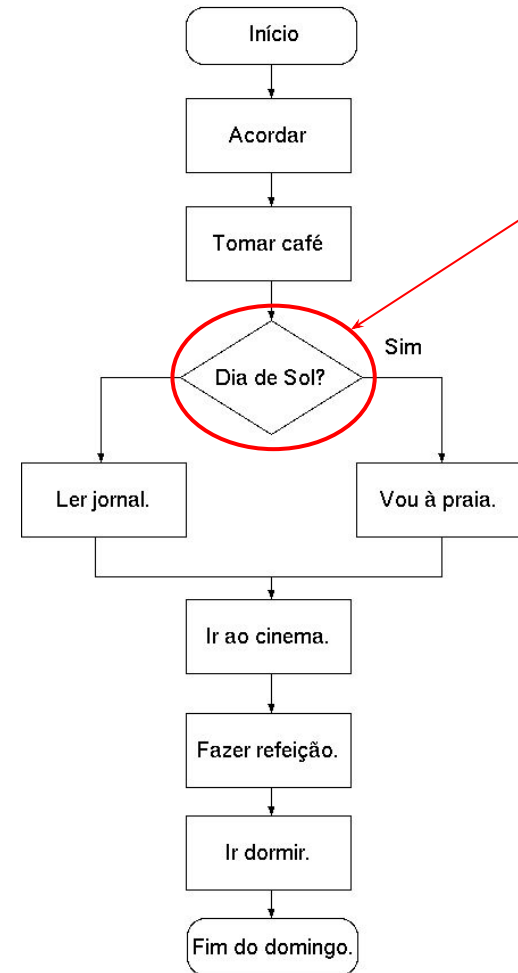
*joice@ufscar.br*

**Algoritmos - Estruturas condicionais**

---

# Controle de fluxo: Estruturas condicionais

## Fluxograma para um domingo





## Estruturas condicionais

- Usado quando temos que executar passos que dependem da verificação de alguma condição
- Condição é dada por uma **expressão booleana (avaliada como Verdadeiro ou Falso)**

# Comando if

```
if (condicao) then
```

```
Comando1  
Comando2  
...
```

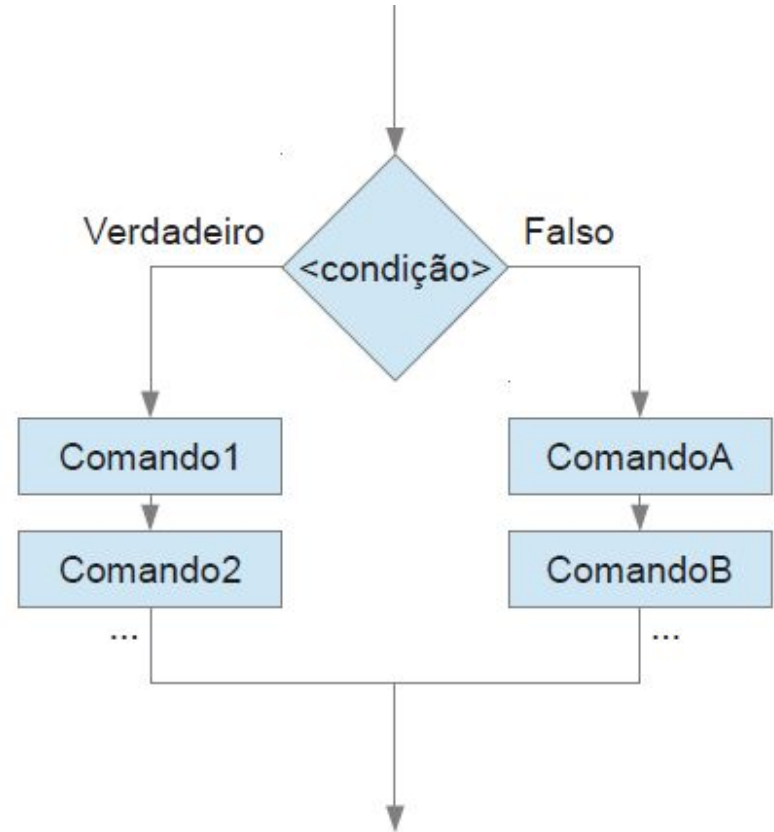
Bloco de comandos executados se condição for verdadeira

```
else
```

```
ComandoA  
ComandoB  
...
```

Bloco de comandos executados se condição for falsa

```
end if
```





## Exemplo

- Solicite a entrada de dois números inteiros  $a$  e  $b$ .  
Imprima o maior.



## Algoritmo Maior

{algoritmo que lê dois números inteiros e imprime o maior }

Input: dois números  $a, b \in \mathbb{Z}$

Output: o maior

begin

    read a,b

    if (a > b) then

        print(a)

    else

        print(b)

    end if

end



## Exercício

- Faça um algoritmo que leia as notas de duas provas de um aluno e calcule a média. O valor máximo de cada prova é 10.0. Considerando a média de aprovação 6.0, imprima uma mensagem informando a média do aluno e se ele foi aprovado ou não.





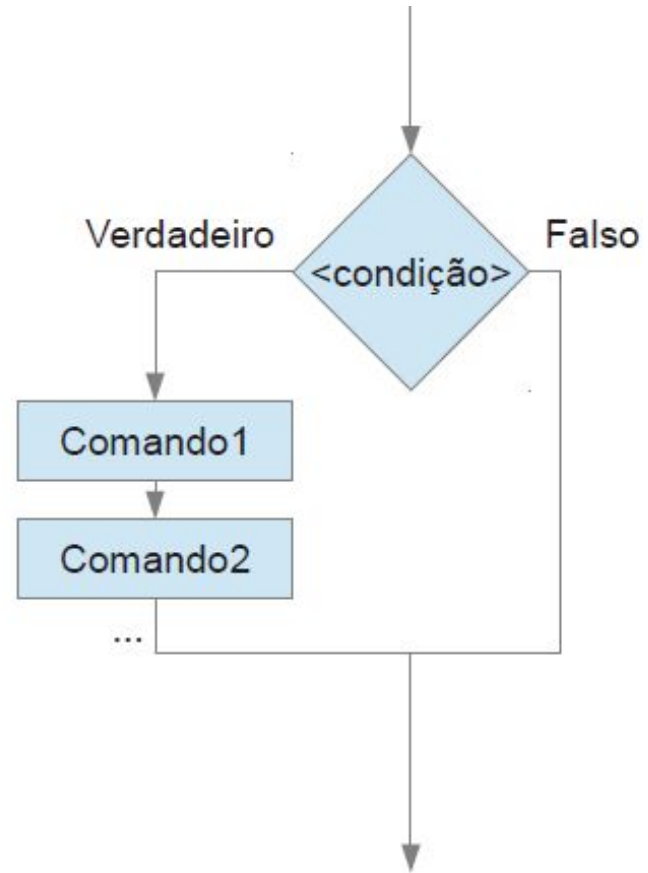
## Exercício

- Crie um algoritmo em pseudocódigo que solicite que o usuário informe o valor do seu salário e o tempo de trabalho em anos. Calcule o valor do bônus que o usuário deverá receber considerando que:
  - Se o tempo de trabalho for maior ou igual a 5 anos, o bônus será de 20% do salário
  - Caso contrário, o bônus será de 10% do salário
- Imprima o valor do bônus e o novo salário.

## Comando if

A cláusula **else** é opcional

```
if (condicao) then
    Comando1
    Comando2
    ...
end if
```





## Exemplo

- Solicite a entrada de um número inteiro e o imprima, caso seja par.



Algoritmo imprimePar

{ lê um inteiro e o imprime, caso seja par}

Input: inteiro  $a \in \mathbb{Z}$

Output: impressão dos números pares

begin

    read a

    if  $(a \% 2 = 0)$  then

        print(a)

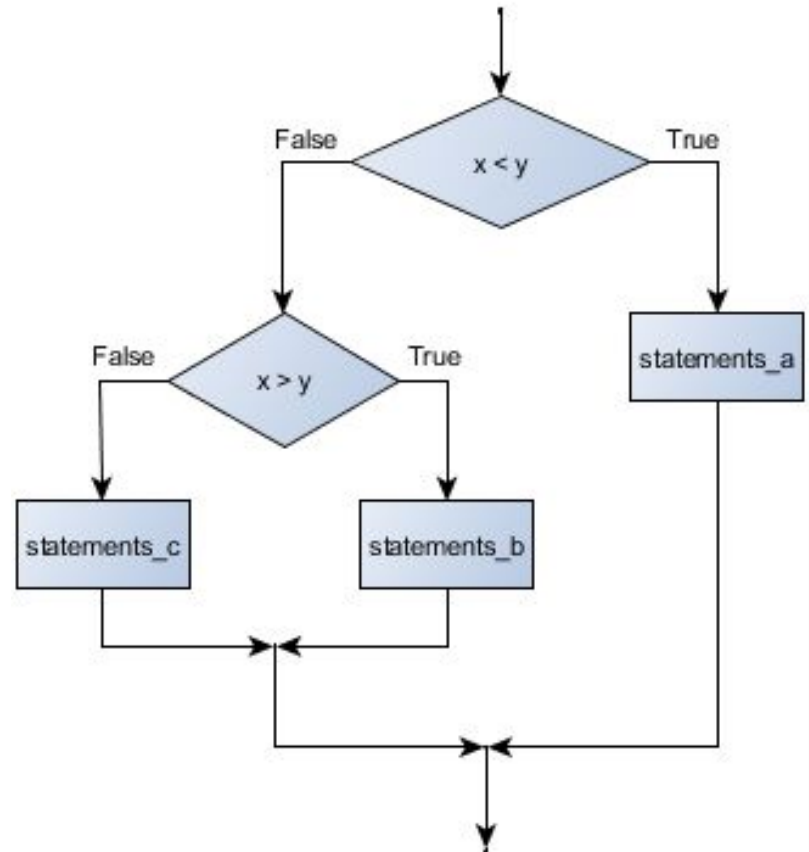
    end if

end

## Comandos if aninhados

Exemplo:

```
if (cond1) then
:   statements_a
else
:   if (cond2) then
:   :   statements_b
:   :   else
:   :   :   Statements_c
:   :   end if
end if
```





## Comandos if aninhados

Outro exemplo...

```
if (cond1) then
.   if (cond2) then
.       statements_b
.   else
.       Statements_c
.   end if
. else
.     if (cond3) then
.         statements_d
.     else
.         Statements_e
.     end if
. end if
```

```

if (cond1) then
    statements_a
else
    if (cond2) then
        statements_b
    else
        if (cond 3) then
            statements_c
        else
            statements_d
        end if
    end if
end if

```

=

```

if (cond1) then
    statements_a
else if (cond2) then
    statements_b
else if (cond3) then
    statements_c
else
    statements_d
end if

```

Pode haver vários else if, mas apenas um else.  
A primeira condição verdadeira será executada



## Exercício

- Construa um algoritmo que solicite o consumo de telefone no mês em minutos e calcule a conta de telefone de acordo com o consumo do mês e as seguintes taxas praticadas pela empresa, dependendo da faixa de consumo:
  - Abaixo de 200 minutos: R\$ 0.20 por minuto
  - De 200 a 400 minutos: R\$ 0.18 por minuto
  - Acima de 400 minutos: R\$ 0.15 por minuto
- Imprima o valor da conta telefônica



Algoritmo contaTelefonica

{Recebe minutos utilizados e calcula o total da conta a partir de  
tabela de tarifas dada}

Input: minutos  $\in \mathbb{Z}$

Output: total  $\in \mathbb{R}$

Begin

    read minutos

    if (minutos < 200) then

        valor  $\leftarrow$  0.2

    else if (minutos <= 400) then

        valor  $\leftarrow$  0.18

    else

        valor  $\leftarrow$  0.15

        end if

    total  $\leftarrow$  valor\*minutos

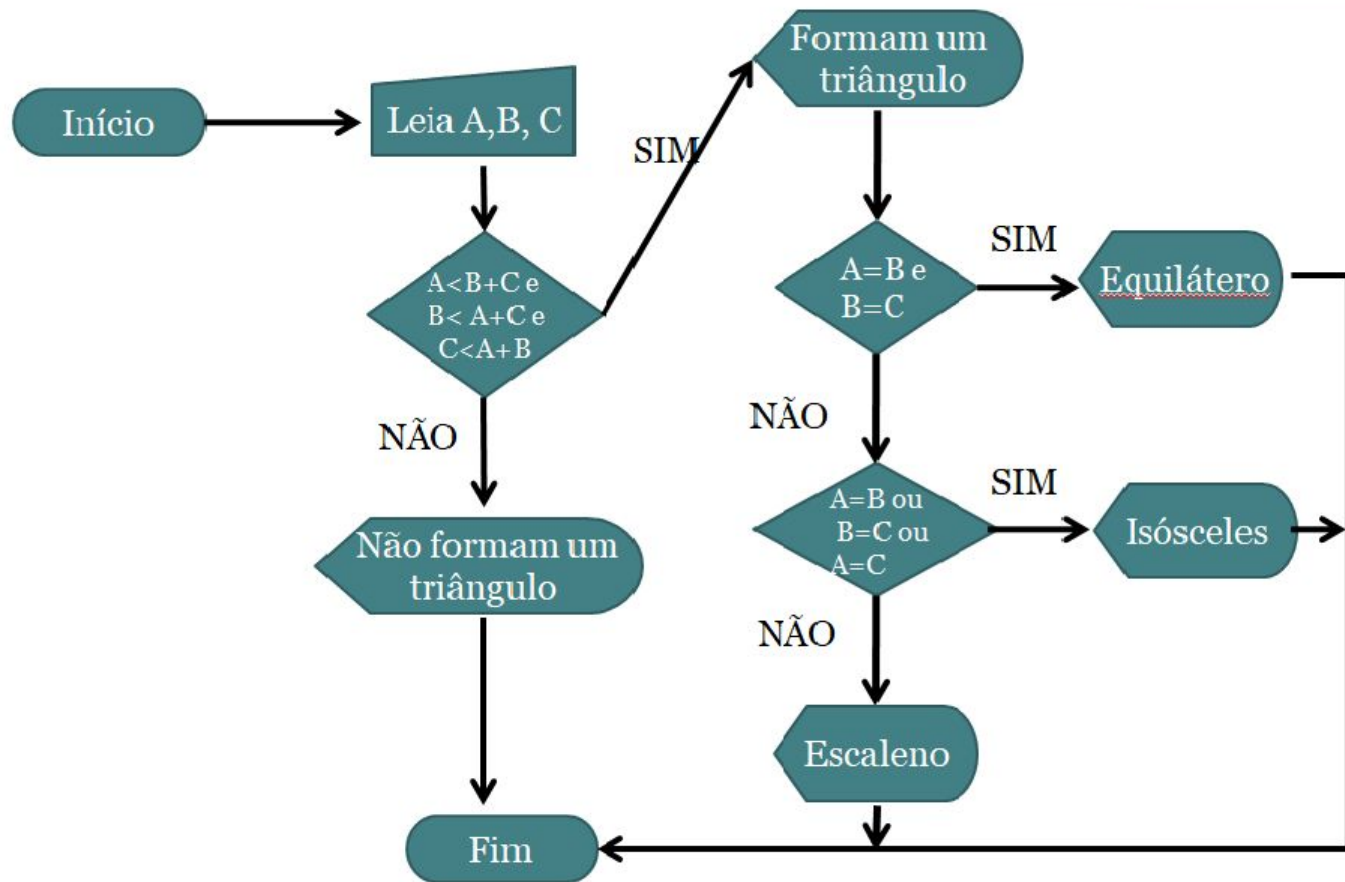
    print total

End



## Exercício

- Construa um algoritmo que solicite que o usuário informe os comprimentos de 3 segmentos e determine se um triângulo pode ser formado com os 3 segmentos, ou seja, se cada lado é menor do que a soma dos outros dois. Caso forme um triângulo, verifique o tipo formado, sendo:
  - Equilátero (todos os lados iguais)
  - Isósceles (dois lados iguais)
  - Escaleno (todos os lados diferentes)
- Elabore uma solução usando fluxograma e, posteriormente, em pseudocódigo





## Teste de mesa

- Numere cada instrução do algoritmo
- Criar tabela com colunas
  - nro da instrução
  - cada variável
- Simulação manual passo a passo do algoritmo. A cada passo
  - Atualização dos valores das variáveis a cada instrução
  - Identificação de falhas nas instruções, no fluxo de execução..



# Teste de mesa

- Definir casos de teste com valores de entrada e saída
  - Cobrir casos possíveis
  - Testar valores limítrofes (extremos de intervalo de validade)
  - Exceções (valores inválidos)



# Teste de mesa

Teste o seu algoritmo anterior com os valores:

- 3,4,5
- 5,4,4
- 4, 4 ,4
- 2,2,8

# Comando Switch

```
switch valor of
  case lista de valores do
    Bloco de comandos
  end case
  case lista de valores do
    Bloco de comandos
  end case
  default
    Bloco de comandos
  end default
end switch
```

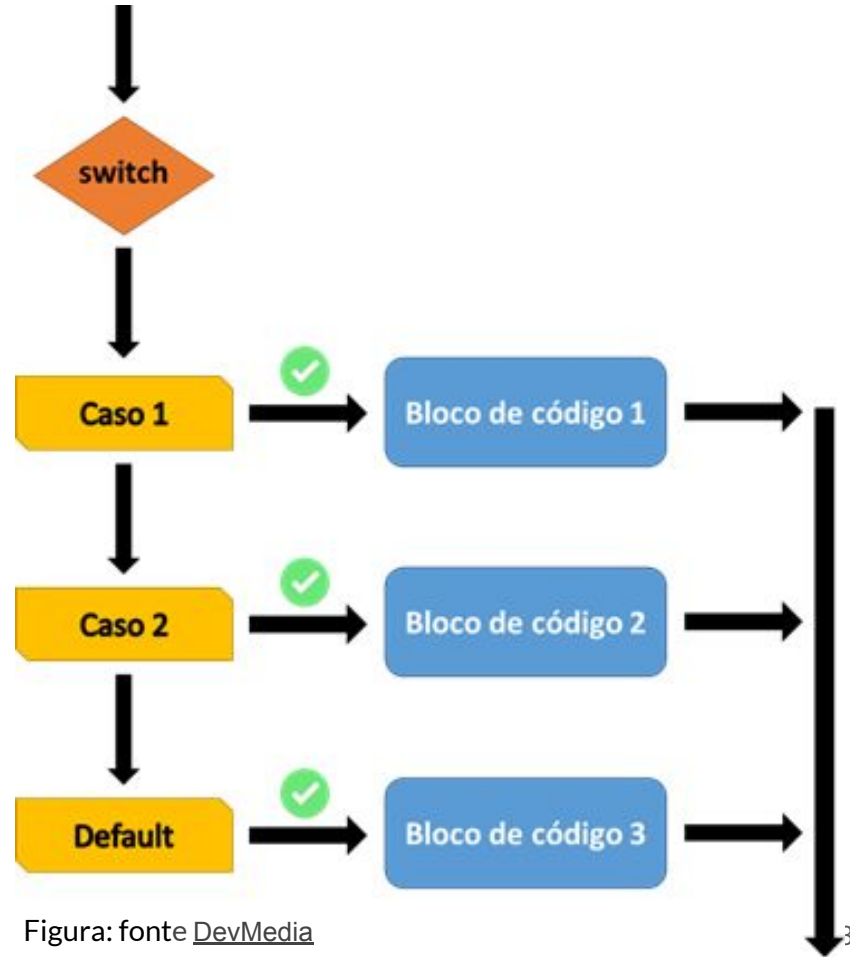


Figura: fonte [DevMedia](#)



## Comando switch

- Os valores em cada **case** devem ser inteiros ou caracteres (pode ser mapeado para inteiro)
- Não permite verificação de intervalos de valores e nem condições
- A cláusula default é opcional





## Exemplo

Determinação do número de dias  
de um mês

```
switch mês of
    case 1,3,5,7,8,10,12 do
        nDias←31
    end case
    case 4,6,9,11 do
        nDias←30
    end case
    case 2 do
        nDias←28
    end case
end switch
```



## Exemplo

Estado civil

```
switch estado_civil of
  case "S" do
    print("Solteiro(a)")
  end case
  case "C" do
    print("Casado(a)")
  end case
  case "V" do
    print("Viúvo(a)")
  end case
  case "D" do
    print("Desquitado(a)")
  end case
  default
    print("Desconhecido")
  end default
end switch
```



## Exercício

Leia um caracter e verifique se o caracter lido é vogal, imprimindo uma mensagem “é vogal” ou “não é vogal”. Desconsidere caracteres acentuados.

Proponha um algoritmo usando o comando switch



## Algoritmos - Organização visual

1. Instruções de mesmo nível devem iniciar na minha coluna
2. Quando houver quebra de linha (linhas longas), deve haver recuo à direita a partir da segunda linha
3. Blocos de comandos internos a uma estrutura (p. ex . if, while, repeat) devem ter recuo à direita para identificação do bloco
4. Linhas em branco devem ser utilizadas para separar blocos lógicos de instruções
5. Incluir comentários, para documentar e para facilitar o entendimento do código



Vídeo no youtube: Como fazer teste de mesa?

<https://youtu.be/Atcfaafvs4M>



# Exercícios

Lista de exercícios CAP (Prof. Jander): <https://goo.gl/mm8fut>

- Seções 3.1 a 3.5