

# Algoritmos

## Desvios Condicionais

Prof. Dr. Osmar Betazzi Dordal

6

# Desvios Condicionais

# Desvios Condicionais

- As estruturas de desvios são utilizadas em algoritmos para efetuar uma tomada de decisão simples;
- Os símbolos de diagrama de blocos para a tomada de decisão são:

- Decisão: 

- Conexão: 

# Operadores Relacionais

- Um operador relacional é **utilizado para comparar a relação entre pares de valores**, uma condição definida pela relação de variáveis com constantes ou de variáveis com variáveis.
- Sempre deve existir um par de valores para determinar se a condição é verdadeira ou falsa.

- Operações **válidas**:  $A = B$ ;  $A > B$ ;  $A \geq B$
- Operações **inválidas**:  $A = B = C$ ;  $A > 10 > 8$

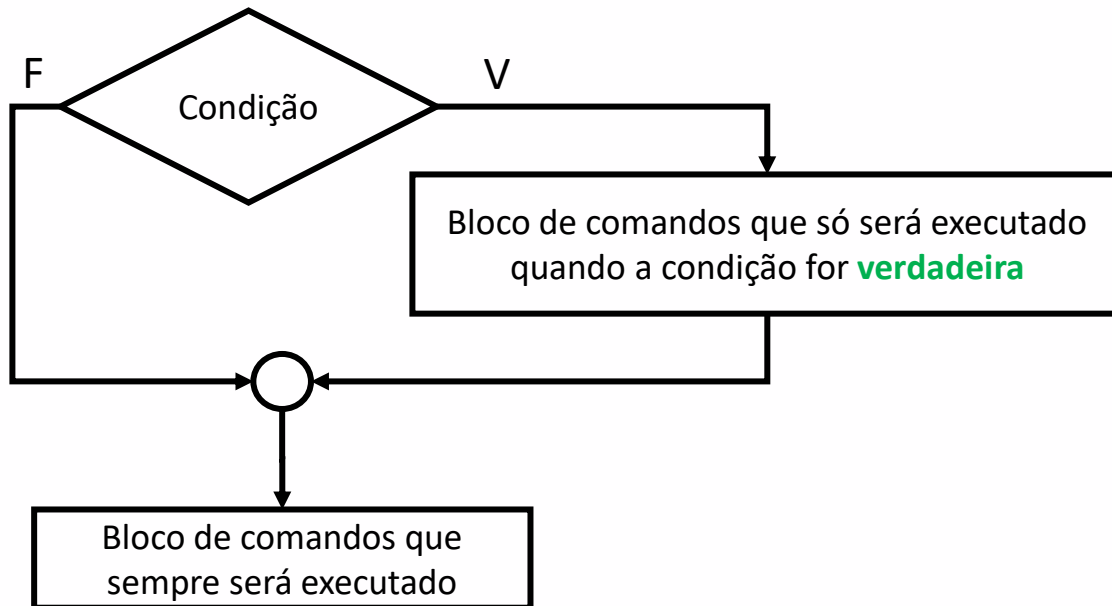
Símbolo	Operador	Descrição
=	=	Igual a
>	>	Maior que
<	<	Menor que
$\geq$	>=	Maior ou igual a
$\leq$	<=	Menor ou igual a
$\neq$	<>	Diferente de

# Desvios Condicionais

- Basicamente, são três modos de desvios condicionais:
  - Simples (se... então... fim\_se)
  - Composto (se... então... senão... fim\_se); e
  - Encadeado (se... então... senão... se... então... senão... fim\_se... fim\_se...)

# Desvio Condicional Simples

- Simples (se... então... fim\_se)



```
se (<condição>) então  
    <Bloco de comandos que só  
    será executado quando a  
    condição for verdadeira>  
fim_se
```

```
if (<condição>) then  
begin  
    Bloco de comandos que só  
    será executado quando a  
    condição for verdadeira;  
end
```

# Desvio Condicional Simples

- Simples (se... então... fim\_se)

```
1 Program Desvio_condicional_simples;  
2 var  
3   nota1, nota2, media: real;  
4  
5 Begin  
6   writeln('Entre com o valor da primeira nota: ');  
7   read(nota1);  
8   writeln('Entre com o valor da segunda nota: ');  
9   read(nota2);  
10  
11   media := ( nota1 + nota2 ) / 2;  
12  
13   if (media >= 7.0) then  
14   begin  
15     writeln('Aprovado');  
16   end  
17 End.
```

*se* (<condição>) *então*  
    <Bloco de comandos que só  
    será executado quando a  
    condição for **verdadeira**>  
*fim\_se*

*if* (<condição>) *then*  
*begin*  
    Bloco de comandos que só  
    será executado quando a  
    condição for **verdadeira**;  
*end*

# Desvio Condicional Simples

- Simples (se... então... fim\_se)

```
1 Program Desvio_condicional_simples;
2 var
3   nota1, nota2, media: real;
4
5 Begin
6   writeln('Entre com o valor da primeira nota: ');
7   read(nota1);
8   writeln('Entre com o valor da segunda nota: ');
9   read(nota2);
10
11   media := ( nota1 + nota2 ) / 2;
12
13   if (media >= 7.0) then
14   begin
15     writeln('Aprovado');
16   end
17 End.
```

```
6 Entre com o valor da primeira nota:
7 7.5
8 Entre com o valor da segunda nota:
9 8.5
10
11
12
13
14
15 Aprovado
```

**se** (<condição>) **então**  
    <Bloco de comandos que só  
    será executado quando a  
    condição for **verdadeira**>  
**fim\_se**

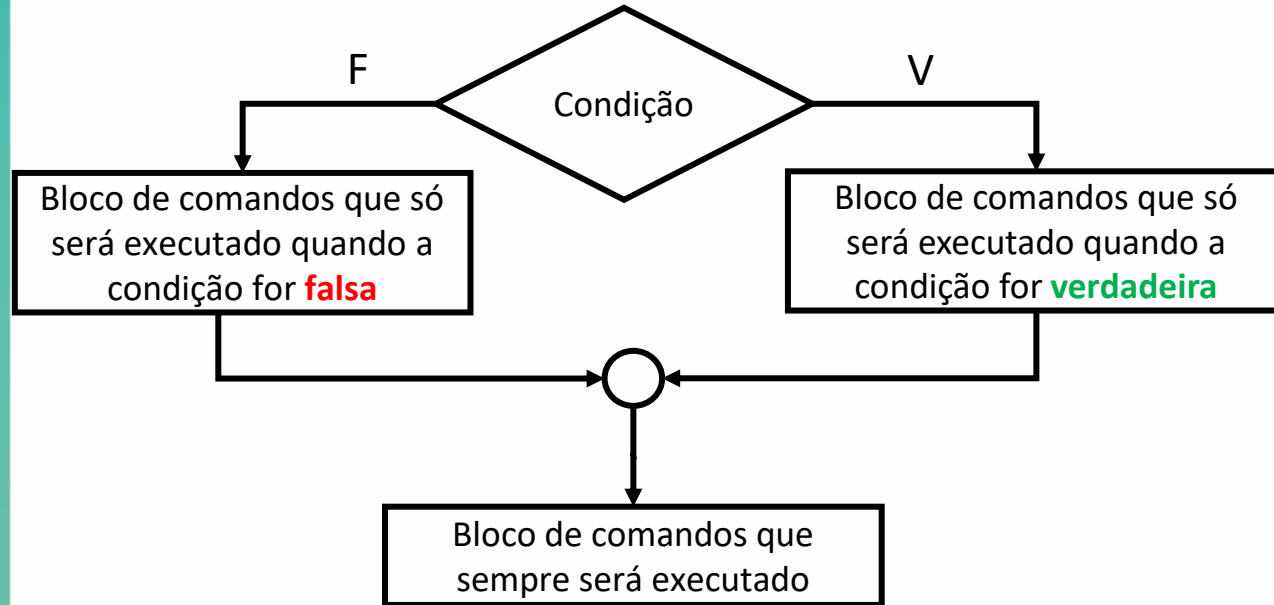
**if** (<condição>) **then**  
**begin**  
    Bloco de comandos que só  
    será executado quando a  
    condição for **verdadeira**;  
**end**

```
6 Entre com o valor da primeira nota:
7 5.8
8 Entre com o valor da segunda nota:
9 5.9
10
11
12
13
14
15 ?
```



# Desvio Condicional Composto

- Composto (se... então... senão... fim\_se);



```
se (<condição>) então  
    <Bloco de comandos que só  
    será executado quando a  
    condição for verdadeira>  
senão  
    <Bloco de comandos que só  
    será executado quando a  
    condição for falsa>  
fim_se
```

```
if (<condição>) then  
begin  
    Bloco de comandos que só  
    será executado quando a  
    condição for verdadeira;  
else  
    Bloco de comandos que só  
    será executado quando a  
    condição for falsa;  
end
```

# Desvio Condicional Composto

- Composto (se... então... senão... fim\_se);

```
1 Program Desvio_condicional_composto;
2 var
3     nota1, nota2, media: real;
4
5 Begin
6     writeln('Entre com o valor da primeira nota: ');
7     read(nota1);
8     writeln('Entre com o valor da segunda nota: ');
9     read(nota2);
10
11     media := ( nota1 + nota2 ) / 2;
12
13     if (media >= 7.0) then
14     begin
15         writeln('Aprovado');
16     end
17     else
18     begin
19         writeln('Reprovado');
20     end
21 End.
```

**if** (<condição>) **then**

**begin**

Bloco de comandos que só  
será executado quando a  
condição for **verdadeira**;

**else**

Bloco de comandos que só  
será executado quando a  
condição for **falsa**;

**end**

# Desvio Condicional Composto

- Composto (se... então... senão... fim\_se);

```
1 Program Desvio_condicional_composto;
2 var
3     nota1, nota2, media: real;
4
5 Begin
6     writeln('Entre com o valor da primeira nota: ');
7     read(nota1);
8     writeln('Entre com o valor da segunda nota: ');
9     read(nota2);
10
11     media := ( nota1 + nota2 ) / 2;
12
13     if (media >= 7.0) then
14     begin
15         writeln('Aprovado');
16     end
17     else
18     begin
19         writeln('Reprovado');
20     end
21 End.
```

```
6 Entre com o valor da primeira nota:
7 7.2
8 Entre com o valor da segunda nota:
9 7.8
10
11
12
13
14
15 Aprovado
```

**if** (<condição>) **then**  
**begin**

Bloco de comandos que só  
será executado quando a  
condição for **verdadeira**;

**else**

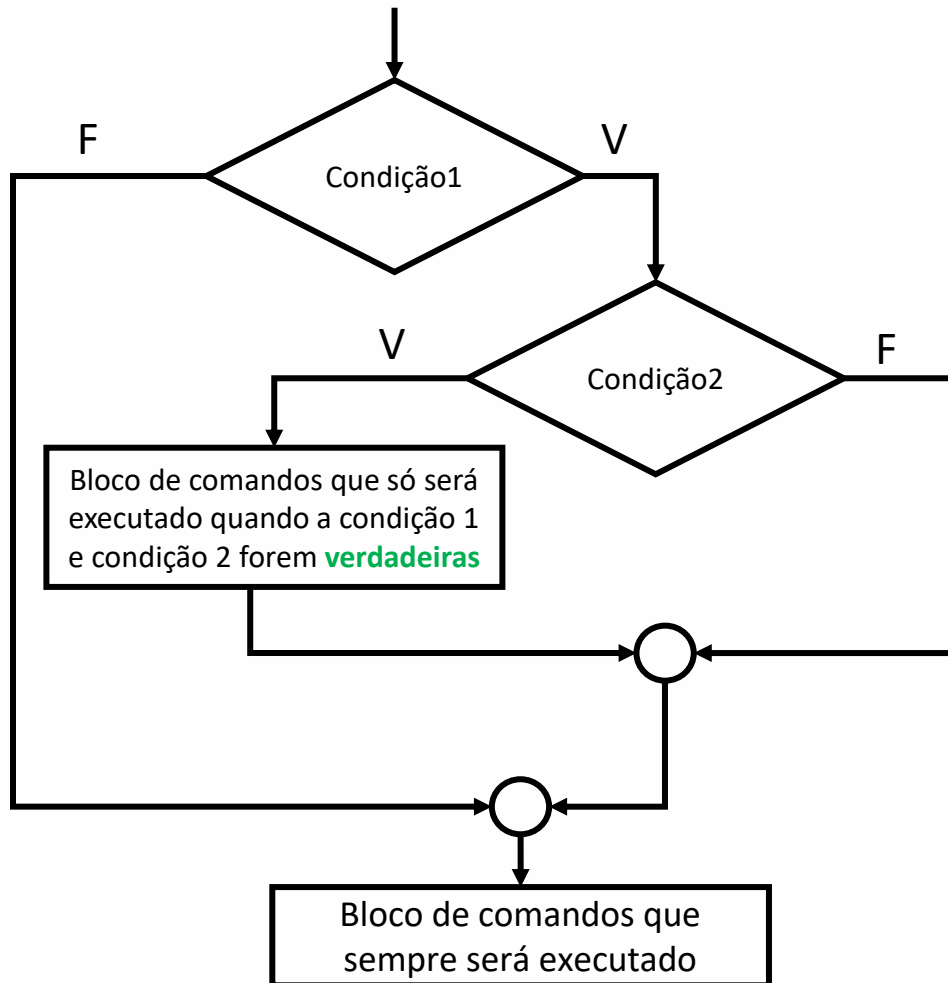
Bloco de comandos que só  
será executado quando a  
condição for **falsa**;

**end**

```
6 Entre com o valor da primeira nota:
7 6.0
8 Entre com o valor da segunda nota:
9 6.8
10
11
12
13
14
15
16
17
18
19 Reprovado
```

# Desvio Condicional Encadeado Simples

- Encadeado (se... então... senão... se... então... senão... fim\_se... fim\_se...)



```
se (<condição 1>) então  
    se (<condição 2>) então  
        <Bloco de comandos que só  
        será executado quando a  
        condição 1 e a condição 2  
        forem verdadeiras>  
    fim_se  
fim_se
```

```
if (<condição 1>) then  
    if (<condição 2>) then  
        Bloco de comandos que só  
        será executado quando a  
        condição 1 e a condição 2  
        forem verdadeiras;  
    end  
end
```

# Desvio Condicional Encadeado Simples

- Encadeado (se... então... senão... se... então... senão... fim\_se... fim\_se...)

```
1 Program Desvio_Condicional_Encadeado_Simples;  
2 var  
3   condicao1, condicao2: boolean;  
4  
5 Begin  
6  
7   condicao1 := true;  
8   condicao2 := true;  
9   if (condicao1) then  
10     if (condicao2) then  
11       writeln(condicao1, ' ', condicao2);  
12  
13   condicao1 := true;  
14   condicao2 := false;  
15   if (condicao1) then  
16     if (condicao2) then  
17       writeln(condicao1, ' ', condicao2);  
18  
19   condicao1 := false;  
20   condicao2 := true;  
21   if (condicao1) then  
22     if (condicao2) then  
23       writeln(condicao1, ' ', condicao2);  
24  
25 End.
```

```
if (<condição 1>) then  
  if (<condição 2>) then  
  
    Bloco de comandos que  
    só será executado  
    quando a condição 1 e a  
    condição 2 forem  
    verdadeiras;  
  
  end  
end
```

# Desvio Condicional Encadeado Simples

- Encadeado (se... então... senão... se... então... senão... fim\_se... fim\_se...)

```
1 Program Desvio_Condicional_Encadeado_Simples;  
2 var  
3   condicao1, condicao2: boolean;  
4  
5 Begin  
6  
7   condicao1 := true;  
8   condicao2 := true;  
9   if (condicao1) then  
10     if (condicao2) then  
11       writeln(condicao1, ' ', condicao2);  
12  
13   condicao1 := true;  
14   condicao2 := false;  
15   if (condicao1) then  
16     if (condicao2) then  
17       writeln(condicao1, ' ', condicao2);  
18  
19   condicao1 := false;  
20   condicao2 := true;  
21   if (condicao1) then  
22     if (condicao2) then  
23       writeln(condicao1, ' ', condicao2);  
24  
25 End.
```

TRUE TRUE

```
if (<condição 1>) then  
  if (<condição 2>) then  
  
    Bloco de comandos que  
    só será executado  
    quando a condição 1 e a  
    condição 2 forem  
    verdadeiras;  
  
  end  
end
```

# Desvio Condicional Encadeado Simples

- Encadeado (se... então... senão... se... então... senão... fim\_se... fim\_se...)

```
1 Program Desvio_Condicional_Encadeado_Simples;
2 var
3   condicao1, condicao2: boolean;
4
5 Begin
6
7   condicao1 := true;
8   condicao2 := true;
9   if (condicao1) then
10     if (condicao2) then
11       writeln(condicao1, ' ', condicao2);
12
13   condicao1 := true;
14   condicao2 := false;
15   if (condicao1) then
16     if (condicao2) then
17       writeln(condicao1, ' ', condicao2);
18
19   condicao1 := false;
20   condicao2 := true;
21   if (condicao1) then
22     if (condicao2) then
23       writeln(condicao1, ' ', condicao2);
24
25 End.
```

Apenas um comando  
para cada bloco sem o “;”

```
if (<condição 1>) then
  if (<condição 2>) then
```

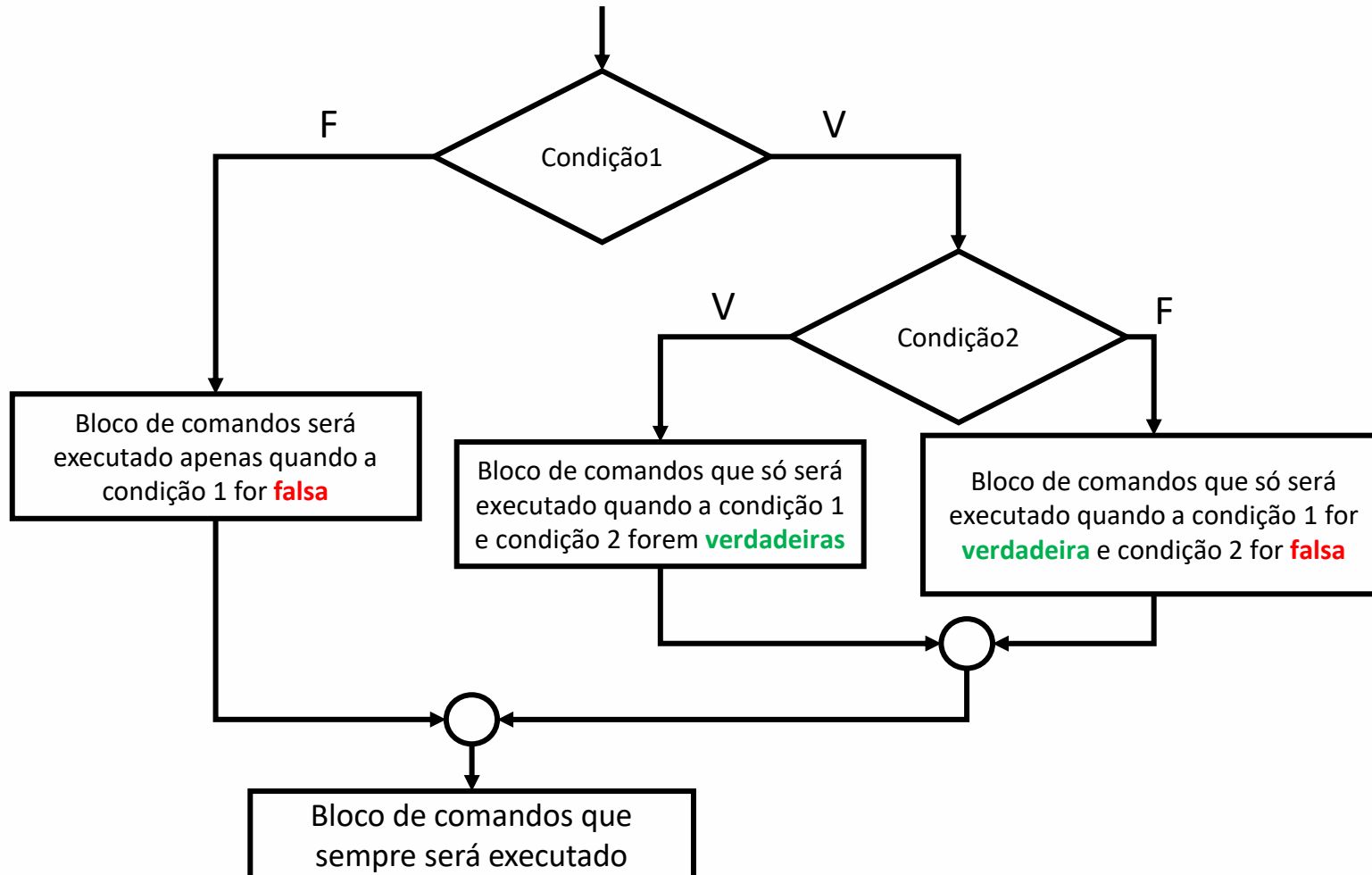
Bloco de comandos que  
só será executado  
quando a condição 1 e a  
condição 2 forem  
**verdadeiras**;

```
end
end
```

```
1 Program Desvio_Condicional_Encadeado_Simples;
2 var
3   condicao1, condicao2: boolean;
4
5 Begin
6
7   condicao1 := true;
8   condicao2 := true;
9   if (condicao1) then
10     begin
11       if (condicao2) then
12         begin
13           writeln(condicao1, ' ', condicao2);
14         end
15     end;
```

# Desvio Condicional Encadeado Composto

- Encadeado (se... então... senão... se... então... senão... fim\_se... fim\_se...)



```
se (<condição 1>) então
    se (<condição 2>) então
        <Ação executada quando a
        condição 1 e condição 2
        forem verdadeiras>
    senão
        <Ação executada quando a
        condição 1 for verdadeira
        e a condição 2 for falsa>
    fim_se
senão
    <Ação executada quando a
    condição 1 for falsa>
fim_se
```



# Desvio Condicional Encadeado Composto

- Encadeado (se... então... senão... se... então... senão... fim\_se... fim\_se...)

```
1 Program Desvio_Condicional_Encadeado_Composto;  
2 var  
3   condicao1, condicao2: boolean;  
4  
5 Begin  
6  
7   condicao1 := false;  
8   condicao2 := false;  
9  
10  if (condicao1) then  
11    if (condicao2) then  
12      writeln(condicao1, ' ', condicao2, ' linha 12')  
13    else  
14      writeln(condicao1, ' ', condicao2, ' linha 14')  
15  else  
16    writeln(condicao1, ' ', condicao2, ' linha 16');  
17  
18 End.
```

```
se (<condição 1>) então  
  se (<condição 2>) então  
    <Ação executada quando a  
      condição 1 e condição 2  
      forem verdadeiras>  
  senão  
    <Ação executada quando a  
      condição 1 for verdadeira  
      e a condição 2 for falsa>  
  fim_se  
senão  
  <Ação executada quando a  
    condição 1 for falsa>  
fim_se
```

FALSE FALSE linha 16

# Desvio Condicional Encadeado Composto

- Encadeado (se... então... senão... se... então... senão... fim\_se... fim\_se...)

```
1 Program Desvio_Condicional_Encadeado_Composto;
2 var
3     condicao1, condicao2: boolean;
4
5 Begin
6
7     condicao1 := true;
8     condicao2 := false;
9
10    if (condicao1) then
11        if (condicao2) then
12            writeln(condicao1, ' ', condicao2, ' linha 12')
13        else
14            writeln(condicao1, ' ', condicao2, ' linha 14')
15    else
16        writeln(condicao1, ' ', condicao2, ' linha 16');
17
18 End.
```

TRUE FALSE linha 14

```
se (<condição 1>) então
    se (<condição 2>) então
        <Ação executada quando a
        condição 1 e condição 2
        forem verdadeiras>
    senão
        <Ação executada quando a
        condição 1 for verdadeira
        e a condição 2 for falsa>
    fim_se
senão
    <Ação executada quando a
    condição 1 for falsa>
fim_se
```

# Desvio Condicional Encadeado Composto

- Encadeado (se... então... senão... se... então... senão... fim\_se... fim\_se...)

```
1 Program Desvio_Condicional_Encadeado_Composto;
2 var
3     condicao1, condicao2: boolean;
4
5 Begin
6
7     condicao1 := false;
8     condicao2 := true;
9
10    if (condicao1) then
11        if (condicao2) then
12            writeln(condicao1, ' ', condicao2, ' linha 12')
13        else
14            writeln(condicao1, ' ', condicao2, ' linha 14')
15    else
16        writeln(condicao1, ' ', condicao2, ' linha 16');
17
18 End.
```

FALSE TRUE linha 16

```
se (<condição 1>) então
    se (<condição 2>) então
        <Ação executada quando a
         condição 1 e condição 2
         forem verdadeiras>
    senão
        <Ação executada quando a
         condição 1 for verdadeira
         e a condição 2 for falsa>
    fim_se
senão
    <Ação executada quando a
     condição 1 for falsa>
fim_se
```

# Desvio Condicional Encadeado Composto

- Encadeado (se... então... senão... se... então... senão... fim\_se... fim\_se...)

```
1 Program Desvio_Condicional_Encadeado_Composto;  
2 var  
3   condicao1, condicao2: boolean;  
4  
5 Begin  
6  
7   condicao1 := true;  
8   condicao2 := true;  
9  
10  if (condicao1) then  
11    if (condicao2) then  
12      writeln(condicao1, ' ', condicao2, ' linha 12')  
13    else  
14      writeln(condicao1, ' ', condicao2, ' linha 14')  
15  else  
16    writeln(condicao1, ' ', condicao2, ' linha 16');  
17  
18 End.
```

TRUE TRUE linha 12

```
if (<condição 1>) then  
  if (<condição 2>) then  
    Ação executada quando a  
    condição 1 e condição 2  
    forem verdadeiras  
  else  
    Ação executada quando a  
    condição 1 for verdadeira  
    e a condição 2 for falsa  
else  
  Ação executada quando a  
  condição 1 for falsa;
```

Antes do *else* não há ponto e vírgula (;)

# Referências

- FORBELLONE, André L. **Lógica de Programação**. Prentice Hall Brasil, 3ª edição, 2005.
- VELOSO, Paulo; et al. **Estrutura de dados**. Rio de Janeiro: Campus, 4ª edição, 1996.
- LAGES & GUIMARAES. **Algoritmos e Estrutura de dados**. Ed. LTC, 1994.
- FARRER, H. **Algoritmos estruturados**. Rio de Janeiro: Guanabara Koogan, 3ª edição, 1989.
- LUIZ, Jaime. **Estrutura de dados e seus algoritmos**. Editora LTC.
- GUEDES, S. **Lógica de Programação Algorítmica**. Editora Pearson, 2014.
- MANZANO, José Augusto N. G. **Algoritmos lógica para desenvolvimento de programação de computadores**. Ed. 1, São Paulo, Erica 2016.
- MANZANO, José Augusto N. G. **Algoritmos técnicas de programação**. Ed 2, São Paulo, Erica, 2016.