

# Algoritmos

## Laços de Repetições – *Loops*

Prof. Dr. Osmar Betazzi Dordal

8

# Laços de Repetições – *Loops*

# Laços de Repetição – *Loops*

- A técnica denominada **laços ou ciclos**, que pode ser referenciada como malhas, ou pelos termos análogos em inglês *loopings* ou *loops*.
- É uma estrutura de programação que **facilita repetir** determinados trechos de código.
- Essa técnica reduz o trabalho de programação, principalmente quando é preciso repetir várias vezes alguma ação importante do programa.

# Laços de Repetição – *Loops*

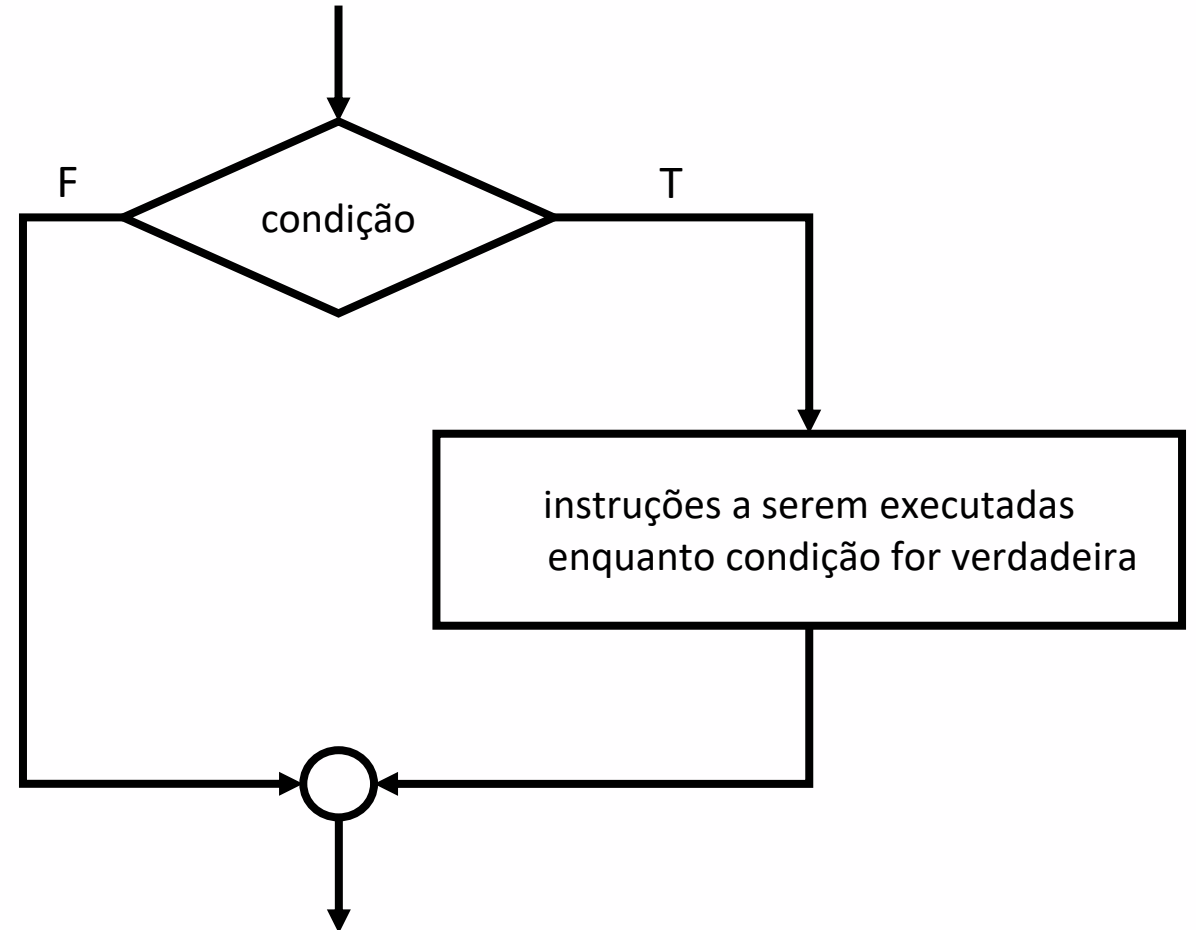
- Os laços de repetições, temos os laços:
  - Condicionais Pré-teste;
  - Condicionais Pós-teste; e
  - Incondicionais

# Laços de Repetição – *Loops*

- Condicionais Pré-teste:

```
enquanto (<condição>) faça  
    <instruções a serem executadas  
    enquanto condição for verdadeira>  
fim_enquanto
```

```
while (<condição>) do  
begin  
    <instruções a serem executadas  
    enquanto condição for verdadeira>  
end
```



# Laços de Repetição – *Loops*

- Condicionais Pré-teste:

```
while (<condição>) do
begin
    <instruções a serem executadas
    enquanto condição for verdadeira>
end
```

```
1 Program laco_condicional_pre_teste;
2 var
3     x: integer;
4 Begin
5     x := 0;
6
7     while (x <= 10) do
8     begin
9
10        writeln('imprimindo o valor de x: ', x);
11        inc(x);
12
13    end
14
15 End.
```

# Laços de Repetição – *Loops*

- Condicionais Pré-teste:

```
while (<condição>) do
begin
    <instruções a serem executadas
    enquanto condição for verdadeira>
end
```

```
imprimindo o valor de x: 0
imprimindo o valor de x: 1
imprimindo o valor de x: 2
imprimindo o valor de x: 3
imprimindo o valor de x: 4
imprimindo o valor de x: 5
imprimindo o valor de x: 6
imprimindo o valor de x: 7
imprimindo o valor de x: 8
imprimindo o valor de x: 9
imprimindo o valor de x: 10
```

```
1 Program laco_condicional_pre_teste;
2 var
3     x: integer;
4 Begin
5     x := 0;
6
7     while (x <= 10) do
8     begin
9
10        writeln('imprimindo o valor de x: ', x);
11        inc(x);
12
13    end
14
15 End.
```

Saída



# Laços de Repetição – *Loops*

- Condicionais Pré-teste:
  - Desenvolver um programa de computador que **leia um valor inteiro qualquer**;
  - Divida-o **sucessivas vezes por 2, até que o seu valor seja menor que 5**; e
  - Apresente como **resultado a quantidade de sucessivas divisões realizadas**.



# Laços de Repetição – *Loops*

- Condicionais Pré-teste:
  - Desenvolver um programa de computador que **leia um valor inteiro qualquer**;
  - Divida-o **sucessivas vezes por 2**, até que o seu valor **seja menor que 5**; e
  - Apresente como **resultado a quantidade de sucessivas divisões realizadas**.

```
1 Program divisoes_sucessivas;  
2 var  
3   n, r: integer;  
4  
5 Begin  
6  
7   r := 0;  
8   write('informe um número inteiro: ');  
9   readln(n);  
10  
11  while (n >= 5) do  
12  begin  
13    n := n div 2;  
14    r := r + 1;  
15  end;  
16  
17  writeln('O número de divisões foi: ', r);  
18  
19 End.
```

# Laços de Repetição – *Loops*

```
1 Program divisoes_sucessivas;  
2 var  
3   n, r: integer;  
4  
5 Begin  
6  
7   r := 0;  
8   write('informe um número inteiro: ');  
9   readln(n);  
10  
11  while (n >= 5) do  
12  begin  
13    n := n div 2;  
14    r := r + 1;  
15  end;  
16  
17  writeln('O número de divisões foi: ', r);  
18  
19 End.
```

r	n

# Laços de Repetição – *Loops*

```
1 Program divisoes_sucessivas;  
2 var  
3   n, r: integer;  
4  
5 Begin  
6  
7   r := 0;  
8   write('informe um número inteiro: ');  
9   readln(n);  
10  
11  while (n >= 5) do  
12  begin  
13    n := n div 2;  
14    r := r + 1;  
15  end;  
16  
17  writeln('O número de divisões foi: ', r);  
18  
19 End.
```

n	r
0	0

# Laços de Repetição – *Loops*

```
1 Program divisoes_sucessivas;  
2 var  
3   n, r: integer;  
4  
5 Begin  
6  
7   r := 0;  
8   write('informe um número inteiro: ');  
9   readln(n);  
10  
11  while (n >= 5) do  
12  begin  
13    n := n div 2;  
14    r := r + 1;  
15  end;  
16  
17  writeln('O número de divisões foi: ', r);  
18  
19 End.
```

n	r
0	0

# Laços de Repetição – *Loops*

```
1 Program divisoes_sucessivas;  
2 var  
3   n, r: integer;  
4  
5 Begin  
6  
7   r := 0;  
8   write('informe um número inteiro: ');  
9   readln(n);  
10  
11  while (n >= 5) do  
12  begin  
13    n := n div 2;  
14    r := r + 1;  
15  end;  
16  
17  writeln('O número de divisões foi: ', r);  
18  
19 End.
```

n	r
0	0

Tela

Informe um número inteiro:

# Laços de Repetição – *Loops*

```
1 Program divisoes_sucessivas;
2 var
3     n, r: integer;
4
5 Begin
6
7     r := 0;
8     write('informe um número inteiro: ');
9     readln(n);
10
11     while (n >= 5) do
12     begin
13         n := n div 2;
14         r := r + 1;
15     end;
16
17     writeln('O número de divisões foi: ', r);
18
19 End.
```

n	r
10	0

Tela

```
Informe um número inteiro: 10
```

# Laços de Repetição – *Loops*

```
1 Program divisoes_sucessivas;
2 var
3     n, r: integer;
4
5 Begin
6
7     r := 0;
8     write('informe um número inteiro: ');
9     readln(n);
10
11     while (n >= 5) do
12     begin
13         n := n div 2;
14         r := r + 1;
15     end;
16
17     writeln('O número de divisões foi: ', r);
18
19 End.
```

n	r
10	0

(10 >= 5) **T**

Tela

```
Informe um número inteiro: 10
```

# Laços de Repetição – *Loops*

```
1 Program divisoes_sucessivas;
2 var
3   n, r: integer;
4
5 Begin
6
7   r := 0;
8   write('informe um número inteiro: ');
9   readln(n);
10
11  while (n >= 5) do
12  begin
13    n := n div 2;
14    r := r + 1;
15  end;
16
17  writeln('O número de divisões foi: ', r);
18
19 End.
```

n	r
5	0

$10 \text{ div } 2 = 5$

Tela

Informe um número inteiro: 10



# Laços de Repetição – *Loops*

```
1 Program divisoes_sucessivas;  
2 var  
3   n, r: integer;  
4  
5 Begin  
6  
7   r := 0;  
8   write('informe um número inteiro: ');  
9   readln(n);  
10  
11  while (n >= 5) do  
12  begin  
13    n := n div 2;  
14    r := r + 1;  
15  end;  
16  
17  writeln('O número de divisões foi: ', r);  
18  
19 End.
```

n	r
5	1

Tela

```
Informe um número inteiro: 10
```

# Laços de Repetição – *Loops*

```
1 Program divisoes_sucessivas;  
2 var  
3   n, r: integer;  
4  
5 Begin  
6  
7   r := 0;  
8   write('informe um número inteiro: ');  
9   readln(n);  
10  
11  while (n >= 5) do  
12  begin  
13    n := n div 2;  
14    r := r + 1;  
15  end;  
16  
17  writeln('O número de divisões foi: ', r);  
18  
19 End.
```

n	r
5	1

(5 >= 5) **T**

Tela

```
Informe um número inteiro: 10
```

# Laços de Repetição – *Loops*

```
1 Program divisoes_sucessivas;
2 var
3   n, r: integer;
4
5 Begin
6
7   r := 0;
8   write('informe um número inteiro: ');
9   readln(n);
10
11  while (n >= 5) do
12  begin
13    n := n div 2;
14    r := r + 1;
15  end;
16
17  writeln('O número de divisões foi: ', r);
18
19 End.
```

n	r
2	1

 5 div 2 = 2

Tela

Informe um número inteiro: 10

# Laços de Repetição – *Loops*

```
1 Program divisoes_sucessivas;
2 var
3     n, r: integer;
4
5 Begin
6
7     r := 0;
8     write('informe um número inteiro: ');
9     readln(n);
10
11     while (n >= 5) do
12     begin
13         n := n div 2;
14         r := r + 1;
15     end;
16
17     writeln('O número de divisões foi: ', r);
18
19 End.
```

n	r
2	2

$5 \text{ div } 2 = 2$

Tela

Informe um número inteiro: 10

# Laços de Repetição – *Loops*

```
1 Program divisoes_sucessivas;
2 var
3     n, r: integer;
4
5 Begin
6
7     r := 0;
8     write('informe um número inteiro: ');
9     readln(n);
10
11     while (n >= 5) do
12     begin
13         n := n div 2;
14         r := r + 1;
15     end;
16
17     writeln('O número de divisões foi: ', r);
18
19 End.
```

n	r
2	2

(2 >= 5) **F**

Tela

```
Informe um número inteiro: 10
```

# Laços de Repetição – *Loops*

```
1 Program divisoes_sucessivas;  
2 var  
3   n, r: integer;  
4  
5 Begin  
6  
7   r := 0;  
8   write('informe um número inteiro: ');  
9   readln(n);  
10  
11  while (n >= 5) do  
12  begin  
13    n := n div 2;  
14    r := r + 1;  
15  end;  
16  
17  writeln('O número de divisões foi: ', r);  
18  
19 End.
```

n	r
2	2

Tela

```
Informe um número inteiro: 10  
O número de divisões foi: 2
```

# Laços de Repetição – *Loops*

- Condicionais Pós-teste:

**repita**

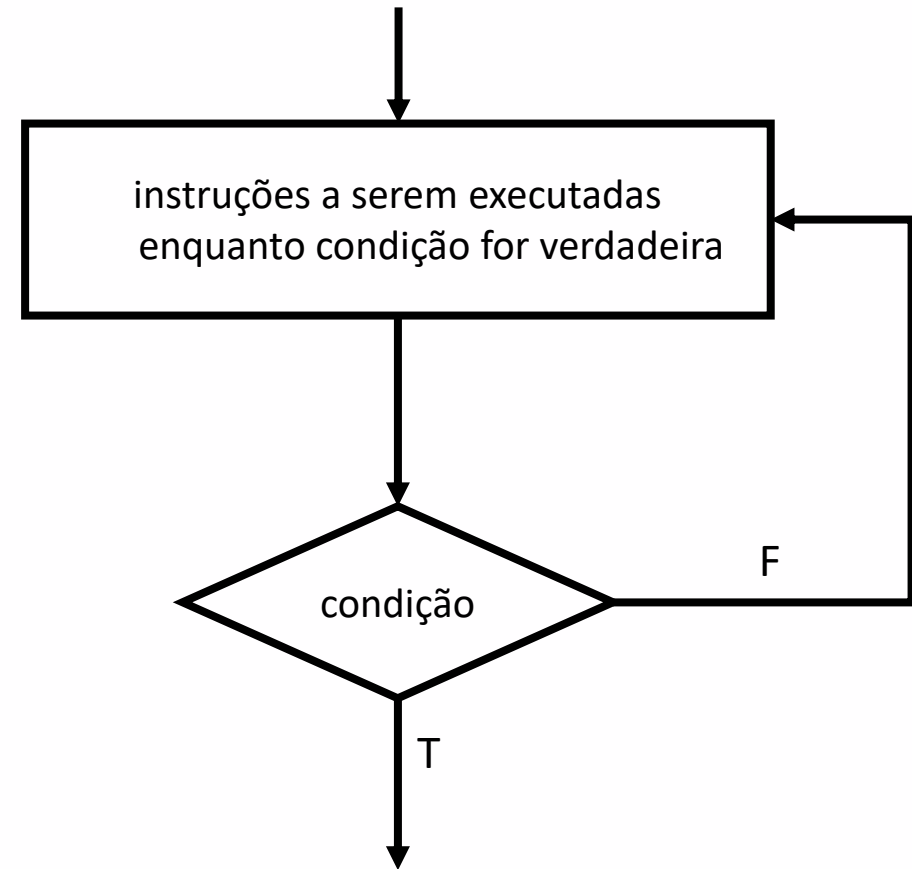
<instruções a serem executadas  
enquanto condição for **verdadeira**>

**até\_que** (<condição – para quando verdadeiro>)

**repeat**

<instruções a serem executadas  
enquanto condição for **verdadeira**>

**until** (<condição – para quando verdadeiro>)



# Laços de Repetição – *Loops*

- Condicionais Pós-teste:
  - Desenvolver um programa de computador que apresente a soma dos cinco primeiros números inteiros.



# Laços de Repetição – *Loops*

- Condicionais Pós-teste:
  - Desenvolver um programa de computador que apresente a soma dos cinco primeiros números inteiros.

```
1 Program laco_posteste;  
2 var  
3   r, i: integer;  
4  
5 Begin  
6  
7   i := 1;  
8   r := 0;  
9  
10  repeat  
11    r := r + i;  
12    i := i + 1;  
13  until (i > 5);  
14  
15  writeln(r);  
16  
17  
18  
19 End.
```

# Laços de Repetição – *Loops*

- Condicionais Pós-teste:
  - Desenvolver um programa de computador que leia um número inteiro qualquer, eleve-o ao quadrado e apresente o resultado do cálculo.
  - Essa operação deve ocorrer até que o usuário responda S de Sim à pergunta “Deseja encerrar(S/N)?”

# Laços de Repetição – *Loops*

- Condicionais Pós-teste:
  - Desenvolver um programa de computador que leia um número inteiro qualquer, eleve-o ao quadrado e apresente o resultado do cálculo.
  - Essa operação deve ocorrer até que o usuário responda S de Sim à pergunta “Deseja encerrar(S/N)?”

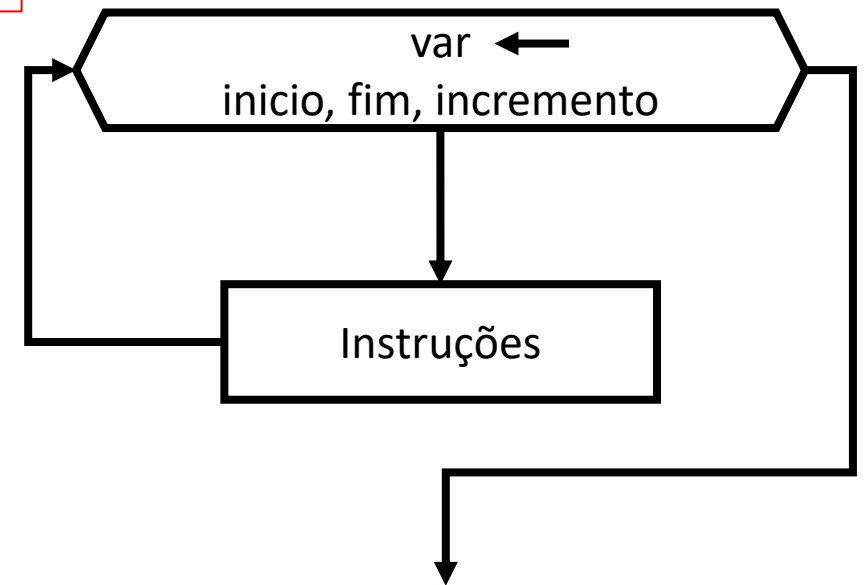
```
1 Program Calculo_interativo;
2 var
3     n, r: integer;
4     decisao: char;
5 Begin
6     decisao := 'N';
7     repeat
8         write('Informe um número inteiro: ');
9         readln(n);
10        r := n * n;
11        writeln('o valor de ', n, ' ao quadrado é: ', r);
12        writeln('Deseja encerrar?(S/N) ');
13        read(decisao);
14    until(decisao = 'S');
15    writeln('FIM');
16
17 End.
```

# Laços de Repetição – *Loops*

- Incondicional:

```
para <variável> de < valor inicial > até < valor final > passo <incremento> faça  
    < comando a repetir >  
fim_para
```

```
for <variável> := <valor inicial> to/downto <valor final> do  
    < comando a repetir >
```



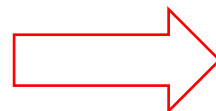
# Laços de Repetição – *Loops*

- Incondicional:

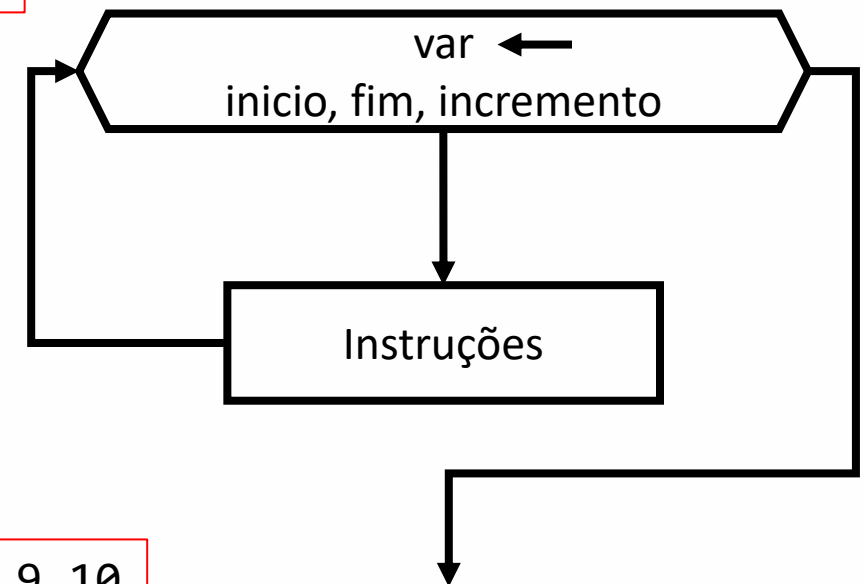
```
para <variável> de < valor inicial > até < valor final > passo <incremento> faça  
    < comando a repetir >  
fim_para
```

```
for <variável> := <valor inicial> to/downto <valor final> do  
    < comando a repetir >
```

```
1 Program Lacos_incondicionais;  
2 var  
3   i: integer;  
4 Begin  
5  
6   for i := 0 to 10 do  
7     write(i, ' ');  
8  
9   writeln();  
10  
11  for i := 10 downto 0 do  
12    write(i, ' ');  
13  
14 End.
```



0	1	2	3	4	5	6	7	8	9	10
10	9	8	7	6	5	4	3	2	1	0

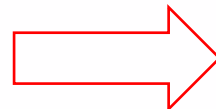


# Laços de Repetição – *Loops* – *break*

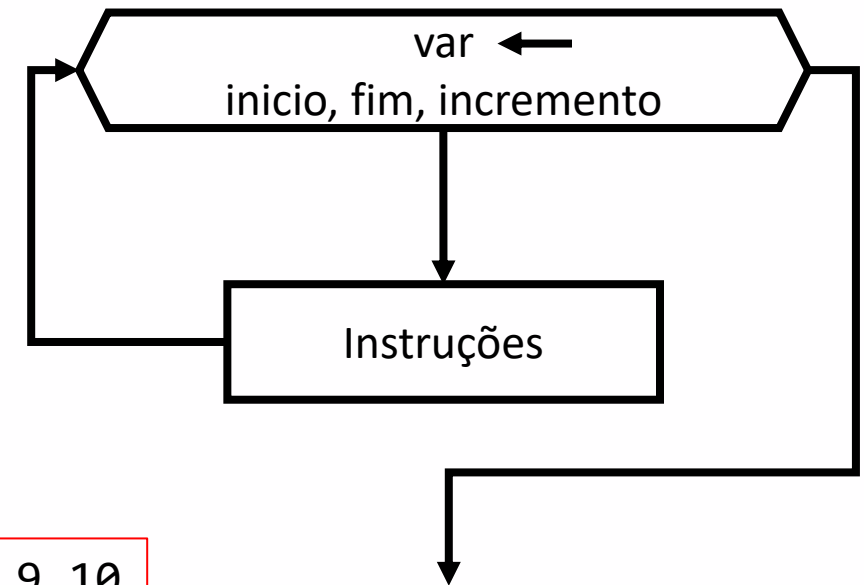
- Incondicional:

```
for <variável> := <valor inicial> to/downto <valor final> do  
    < comando a repetir >
```

```
1 Program Lacos_incondicionais;  
2 var  
3   i: integer;  
4 Begin  
5  
6   for i := 0 to 10 do  
7     write(i, ' ');  
8  
9     writeln();  
10  
11   for i := 10 downto 0 do  
12     begin  
13       write(i, ' ');  
14       break;  
15     end;  
16  
17 End.
```



0 1 2 3 4 5 6 7 8 9 10  
10



# Laços de Repetição – *Loops* – *break*

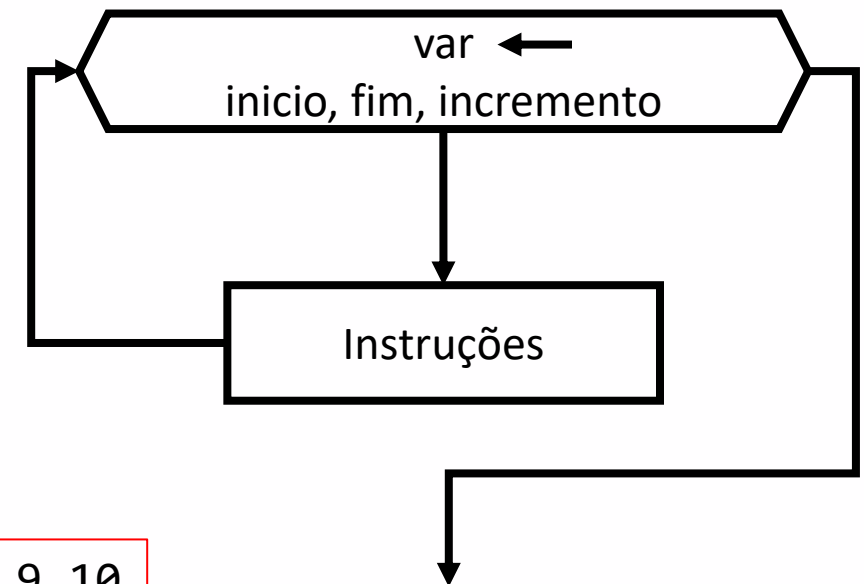
- Incondicional:

```
for <variável> := <valor inicial> to/downto <valor final> do  
    < comando a repetir >
```

```
1 Program Lacos_incondicionais;  
2 var  
3   i: integer;  
4 Begin  
5  
6   for i := 0 to 10 do  
7     write(i, ' ');  
8  
9     writeln();  
10  
11   for i := 10 downto 0 do  
12     begin  
13       write(i, ' ');  
14       break;  
15     end;  
16  
17 End.
```

*begin e end?*

0 1 2 3 4 5 6 7 8 9 10  
10



# 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.