



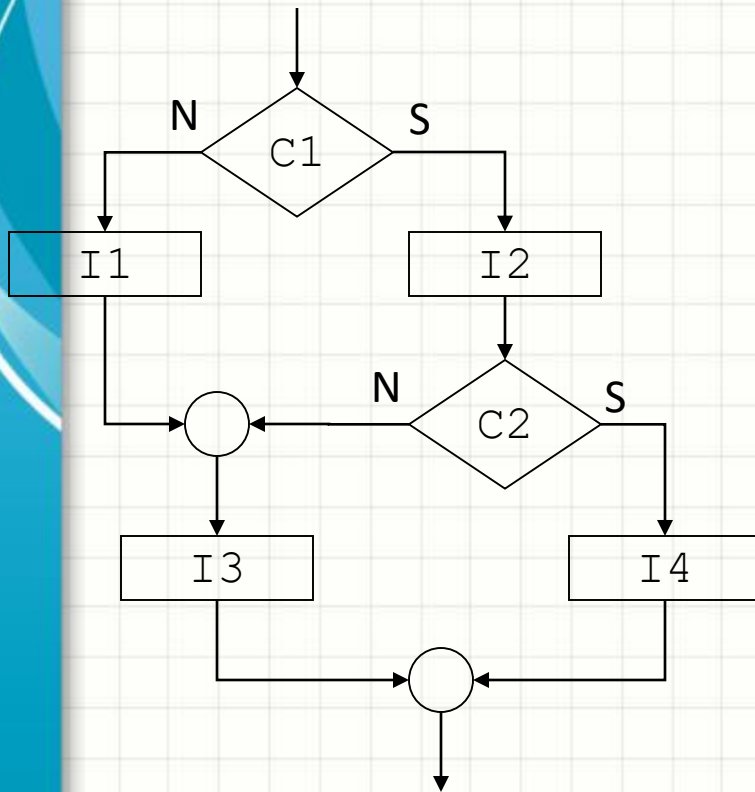
# LÓGICA DE PROGRAMAÇÃO

EAGS

Prof. Marcus Paulo Amorim

# Exercícios

3. Informe se é verdadeiro (V) ou falso (F) o que se afirma sobre o diagrama de blocos abaixo. A seguir, indique a opção com a sequência correta.



Considere:

C1 e C2 são condições.

I1, I2, I3 e I4 são instruções.

( ) I1 executa se C1 for falsa.

( ) I2 somente executa se C2 for verdadeira

( ) Se C1 e C2 forem falsas a I3 executa

( ) I4 sempre executa, independente de C1 e C2

a) V – F – V – V

b) F – F – F – V

~~c) V – F – V – F~~

d) F – V – V – F

# Exercícios

4. Verifique o resultado lógico das expressões de programação mencionadas abaixo, marque se é verdadeiro (V) ou falso (F). A seguir, assinale a opção com a sequência correta. Considere para as respostas os seguintes valores:  $X=1$ ;  $Y=5$ .

☐  $(X < 1) .e. (X > Y)$

☐  $(Y > 5) .ou. (X = 1)$

☐  $.não.(X \geq Y)$

~~a)~~ F – V – V

b) V – F – F

c) F – V – F

d) V – F – V

5. Marque a alternativa que identifica o tipo de operador lógico utilizado na tabela Abaixo.

Condição1	Condição2	Resultado
Falsa	Falsa	<i>Falso</i>
Verdadeira	Falsa	<i>Falso</i>
Falsa	Verdadeira	<i>Falso</i>
Verdadeira	Verdadeira	<i>Verdadeiro</i>

a) .ou.

b) .if.

~~c)~~ .e.

d) .não.

# Exercícios

6. Indique a opção que contém o valor final da variável X, após a execução do trecho de programa em português estruturado mostrado abaixo. Considere os seguintes valores para as variáveis: A=3; B=2; C=8; D=7.

```
se .não. (A > 3) .e. .não. (B < 5) então
```

```
    X ← 10
```

```
senão
```

```
    se (A >= 2) .ou. (C <= 1) então
```

```
        X ← (A + D) / 2
```

```
    senão
```

```
        se (A = 2) .ou. (B < 7) então
```

```
            X ← (A + 2) * (B - 2)
```

```
        senão
```

```
            X ← ((A + C) / B) * (C + D)
```

```
        fim_se
```

```
    fim_se
```

```
fim_se
```

a) 10

b) 0

c) 15

~~d) 5~~

# Exercício

Faça um algoritmo em português estruturado, que efetue o cálculo da área de uma circunferência, apresentando a medida da área calculada.

Sendo assim, basta estabelecer:

1. Ler um valor para o raio, no caso variável R;
2. Estabelecer que PI possui o valor 3.14159;
3. Efetuar o cálculo da área, elevando ao quadrado o valor de R e multiplicando por PI;
4. Apresentar o valor da variável A.

# Exercício

Faça um algoritmo em português estruturado, que efetue o cálculo da área de uma circunferência, apresentando a medida da área calculada.

Sendo assim, basta estabelecer:

1. Ler um valor para o raio, no caso variável R;
2. Estabelecer que PI possui o valor 3.14159;
3. Efetuar o cálculo da área, elevando ao quadrado o valor de R e multiplicando por PI;
4. Apresentar o valor da variável A.

```
programa area
var
    a : real
    r : real
inicio
    leia r
    a <- 3.14159 * r2
    escreva a
fim
```



Fazer um algoritmo em pseudocodigo para ler o ano de nascimento de uma pessoa, calcular e mostrar sua idade e, também, verificar e mostrar se ela já tem idade para votar (16 anos ou mais) e para conseguir a Carteira de Habilitação (18 anos ou mais).

**algoritmo "Maior Idade"**

**var**

**ano, idade: inteiro**

**inicio**

**escreval("digite seu ano de nascimento: ")**

**leia(ano)**

**idade <- 2022 - ano**

**escreval("Sua idade é: ", idade)**

**se(idade >= 18) entao**

**escreval("Ja tem idade para votar")**

**escreval("Já tem idade para ter habilitação")**

**senao**

**se(idade >= 16 ) entao**

**escreval("Já tem idade para votar")**

**senao**

**escreval("Não pode votar e nem ter habilitação")**

**fimse**

**fimse**

**fimalgoritmo**

# COMANDO CASO

É o comando que permite a opção entre várias alternativas de acordo com a verificação de uma condição lógico relacional.

No caso de nenhuma condição ser verdadeira, o algoritmo seguirá normalmente após a palavra FIM\_CASO sem executar nenhum comando associado ao comando CASO.

O comando Caso é Equivalente a uma sequência de comando SE SENÃO.



# Exemplo

Construa um algoritmo que leia um número de 1 a 3 e mostre a qual prêmio esse número se refere. Leve em consideração que existam 3 prêmios.

```
algoritmo "semnome"
var
    i : inteiro
    premio : caracter
inicio
    leia (i)
    escolha i
        CASO 1
            premio <- ("Bicicleta")
        CASO 2
            premio <- ("Relógio")
        CASO 3
            premio <- ("Casa")
        OUTROCASO
            premio <- ("Opcao inválida")
    FIMESCOLHA
    escreva (premio)
fimalgoritmo
```

# Exemplo

Algoritmo que seleciona o melhor time do Brasil.

```
algoritmo "Times"
var
    time: caractere
inicio
    escreva ("Entre com o nome do melhor time de futebol do Brasil: ")
    leia (time)
    escolha time
        caso "Flamengo"
            escreval ("Parabéns... Você acertou.")
        caso "Botafogo"
            escreval ("Mas esse time nem torcida tem.")
        caso "Fluminense"
            escreval ("Nossa... Você é doido.")
        caso "Vasco"
            escreval ("KKKK... Estamos falando de Times da primeira divisão.")
        outrocaso
            escreval ("Você não entendeu a pergunta.")
    fimescolha
finalgoritmo
```

# Estruturas de Controle

## Laços ou Malhas de Repetição

- Utilizado quando é necessário efetuar a repetição de um trecho de programa um determinado numero de vezes.
- Também chamado de **looping** ou **iteração**

# Estruturas de Controle

Supondo um programa que deva executar um determinado trecho de instruções por cinco vezes.

Por exemplo:

Imagine um programa que peça a leitura de um valor para a variável X 5 vezes, multiplique esse valor por 3, implicando-o à variável de resposta R, e apresente o valor obtido.

# Estruturas de Controle

Com o conhecimento que temos até hoje, faríamos:

Programa numero

Var

x, r : inteiro

Inicio

leia x

r <- x \* 3

escreva r

leia x

r <- x \* 3

escreva r

leia x

r <- x \* 3

escreva r

leia x

r <- x \* 3

escreva r

leia x

r <- x \* 3

escreva r

fim

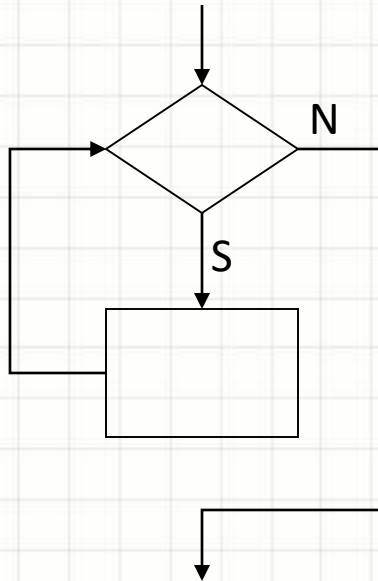
# Estruturas de Controle

Para estes casos existem comandos apropriados para efetuar a repetição de determinados trechos de programas o número de vezes que for necessário.



# Repetição com teste lógico no início do looping

- Um teste lógico é efetuado no início de um looping, verificando se é permitido executar um trecho de instruções subordinado a esse looping.
- **enquanto... faça... fim\_enquanto**



# Repetição com teste lógico no início do looping

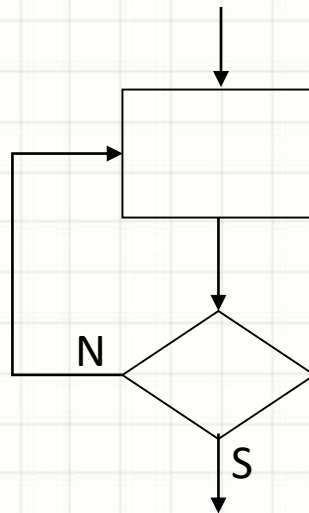
```
programa LOOPING_EXEMPLO1
var
    X, R : inteiro
    CONT : inteiro
início
    CONT ← 1
    enquanto (CONT <= 5) faça
        leia X
        R ← X * 3
        escreva R
        CONT ← CONT + 1
    fim_enquanto
fim
```

# Repetição com teste lógico no início do looping

```
programa LOOPING_EXEMPLO2
var
  X, R : inteiro
  RESP : caractere
início
  RESP ← "SIM"
  enquanto (RESP = "SIM") faça
    leia X
    R ← X * 3
    escreva R
    escreva "Deseja continuar?"
    leia RESP
  fim_enquanto
fim
```

# Repetição com **teste lógico no fim do looping**

- Efetua um teste lógico no fim de um looping.
- As instruções subordinadas a condição estabelecida serão executadas pelo menos uma vez antes ser feito o primeiro teste lógico.
- `repita... até_que...`



# Repetição com teste lógico no fim do looping

```
programa LOOPING_EXEMPLO3
var
    X, R : inteiro
    CONT : inteiro
início
    CONT ← 1
    repita
        leia X
        R ← X * 3
        escreva R
        CONT ← CONT + 1
    até_que (CONT > 5)
fim
```

# Repetição com teste lógico no fim do looping

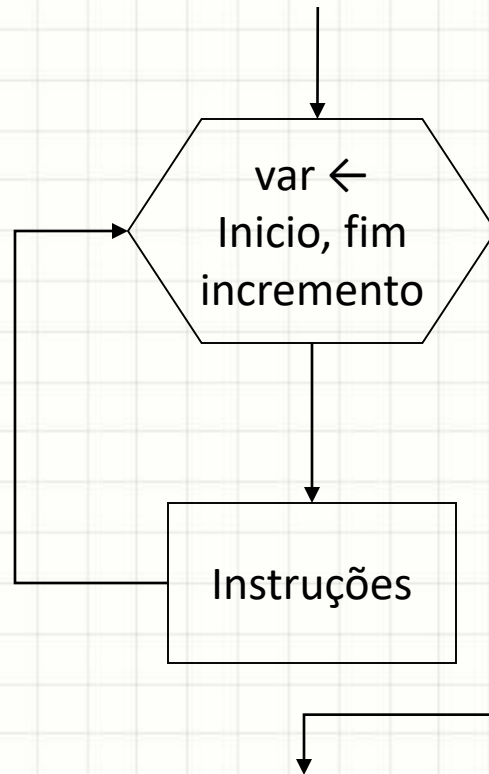
```
programa LOOPING_EXEMPLO4
var
  X, R : inteiro
  RESP : caractere
início
  repita
    leia X
    R ← X * 3
    escreva R
    escreva "Deseja continuar?"
    leia RESP
  até_que (RESP <> "SIM")
fim
```



# Repetição com Variável de Controle

- Normalmente usamos o **enquanto** e o **repita** quando não sabemos previamente a quantidade de vezes que queremos executar um looping.
- Quando conhecemos a quantidade de vezes que queremos executar o looping, isto é, quando o looping é **finito** utilizamos a instrução **para**.
- **para... de... até... passo... faça... fim\_para**

# Repetição com Variável de Controle



```
para <variável> de <início> até <fim> passo <incremento> faça  
    <instruções>  
fim_para
```

# Repetição com Variável de Controle (Exemplo)

- Pedir a leitura de um valor para a variável X, multiplicar esse valor por 3, implicando à variável de resposta R e apresentar o valor obtido, repetindo esta sequência por cinco vezes.

```
programa LOOPING_PARA
```

```
var
```

```
    X, R : inteiro
```

```
    CONT : inteiro
```

```
início
```

```
    para CONT de 1 até 5 passo 1 faça
```

```
        leia X
```

```
         $R \leftarrow X * 3$ 
```

```
        escreva R
```

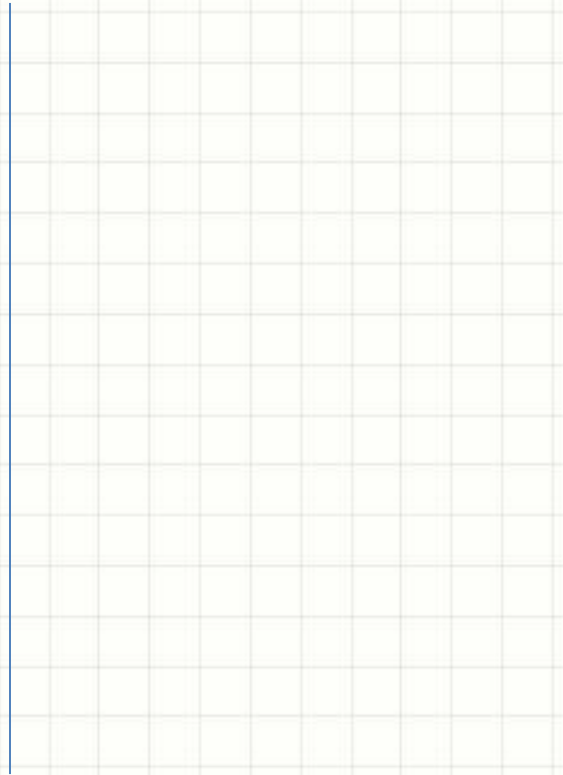
```
    fim_para
```

```
fim
```

# Exercício

Construa um algoritmo que exiba os números pares de 1 até 100.

Em português estruturado utilizando o *enquanto*, o *repita* e o *para*.

Two vertical blue lines are positioned on the grid background, intended for the student to write their algorithm. The lines are parallel and extend from the top of the text area down towards the bottom of the slide.