

```

enquanto (RESP = "sim") faca
    leia(X)
    R <- X * 3
    escreval(R)
    escreval("Deseja continuar? ")
    leia(RESP)
fimenquanto
fimalgoritmo

```

Veja que o contador foi substituído pela variável *RESP*, que enquanto for igual a “sim” executará a rotina existente entre as instruções **enquanto** e **fimenquanto**. Neste caso, o número de vezes que a rotina irá se repetir será controlado pelo usuário e será encerrada somente quando alguma informação diferente de “sim” for fornecida para a variável *RESP*.

5.1.1 - Exercícios de Entrega Obrigatória (até ____/____/____) (Lista 03)

- 1) Desenvolva os algoritmos, seus respectivos diagramas de bloco e codificação em português estruturado. Usar na resolução dos problemas apenas estruturas de repetição do tipo **enquanto** (Você deve gravar o exercício “a” como L03A, o exercício “b” como L03B, e assim por diante).
- a) Apresentar os resultados de uma tabuada de multiplicar (de 1 até 10) de um número qualquer.
 - b) Apresentar o total da soma obtida dos cem primeiros números inteiros (1+2+3+4+...+98+99+100).
 - c) Elaborar um programa que apresente no final o somatório dos valores pares existentes na faixa de 1 até 500.
 - d) Apresentar todos os valores numéricos inteiros ímpares situados na faixa de 0 a 20. Para verificar se o número é ímpar, efetuar dentro da malha a verificação lógica desta condição com a instrução **se**, perguntando se o número é ímpar; sendo, mostre-o; não sendo, passe para o próximo passo.
 - e) Apresentar os resultados das potências de 3, variando do expoente 0 até o expoente 15. Deve ser considerado que qualquer número elevado a zero é 1, e elevado a 1 é ele próprio. Observe que neste exercício não pode ser utilizado o operador de exponenciação do portuguol (^).
 - f) Elaborar um programa que apresente como resultado o valor de uma potência de uma base qualquer elevada a um expoente qualquer, ou seja, de B^E , em que B é o valor da base e E o valor do expoente. Observe que neste exercício não pode ser utilizado o operador de exponenciação do portuguol (^).
 - g) Escreva um programa que apresente a série de Fibonacci até o décimo quinto termo. A série de Fibonacci é formada pela seqüência: 1, 1, 2, 3, 5, 8, 13, 21, 34, ..., etc. Esta série se caracteriza pela soma de um termo atual com o seu anterior subseqüente, para que seja formado o próximo valor da seqüência. Portanto começando com os números 1, 1 o próximo termo é $1+1=2$, o próximo é $1+2=3$, o próximo é $2+3=5$, o próximo $3+5=8$, etc.
 - h) Elaborar um programa que apresente os valores de conversão de graus Celsius em Fahrenheit, de 10 em 10 graus, iniciando a contagem em 10 graus Celsius e finalizando em 100 graus Celsius. O programa deve apresentar os valores das duas temperaturas. A fórmula de conversão é $F = \frac{9C + 160}{5}$, sendo F a temperatura em Fahrenheit e C a temperatura em Celsius.
 - i) Elaborar um programa que efetue a leitura de 10 valores numéricos e apresente no final o total do somatório e a média aritmética dos valores lidos.
 - j) Elaborar um programa que apresente os resultados da soma e da média aritmética dos valores pares situados na faixa numérica de 50 a 70.
 - k) Elaborar um programa que possibilite calcular a área total de uma residência (sala, cozinha, banheiro, quartos, área de serviço, quintal, garagem, etc.). O programa deve solicitar a entrada do nome, a largura e o comprimento de um determinado cômodo. Em seguida, deve apresentar a área

do cômodo lido e também uma mensagem solicitando do usuário a confirmação de continuar calculando novos cômodos. Caso o usuário responda “NAO”, o programa deve apresentar o valor total acumulado da área residencial.

- I) Elaborar um programa que efetue a leitura de valores positivos inteiros até que um valor negativo seja informado. Ao final devem ser apresentados o maior e o menor valores informados pelo usuário.

5.2 – Repetição do Tipo: Teste Lógico no Fim do Looping.

Caracteriza-se por uma estrutura que efetua um teste lógico no fim de um looping. Esta estrutura é parecida com a **enquanto**. A estrutura em questão é denominada de **repita**, sendo conseguida com a utilização do conjunto de instruções **repita...ate**.

A estrutura **repita...ate** tem o seu funcionamento controlado por decisão. Porém, irá efetuar a execução de um conjunto de instruções *pelo menos uma vez* antes de verificar a validade da condição estabelecida. Diferente da estrutura **enquanto** que executa somente um conjunto de instruções, enquanto a condição é verdadeira.

Desta forma **repita** tem seu funcionamento em sentido contrário a **enquanto**, pois sempre irá processar um conjunto de instruções no mínimo uma vez até que a condição se torne **Verdadeira**. Para a estrutura **repita** um conjunto de instruções é executado enquanto a condição se mantém **Falsa** e até que ela seja **Verdadeira**.

Para exemplificar a utilização de **repita**, será utilizado o mesmo exemplo anterior: “Pedir a leitura de um valor para a variável X, multiplicar esse valor por 3, implicando-o à variável R, e apresentar o valor obtido, repetindo esta seqüência por cinco vezes”.

Algoritmo

1. Criar uma variável para servir como contador com valor inicial 1;
2. Ler um valor para a variável X ;
3. Efetuar a multiplicação do valor de X por 3, implicando o resultado em R;
4. Apresentar o valor calculado contido na variável R;
5. Acrescentar +1 a variável contador, definida no passo 1;
6. Repetir os passos 2, 3, 4 e 5 até que o contador seja maior que 5.

Diagrama de Blocos

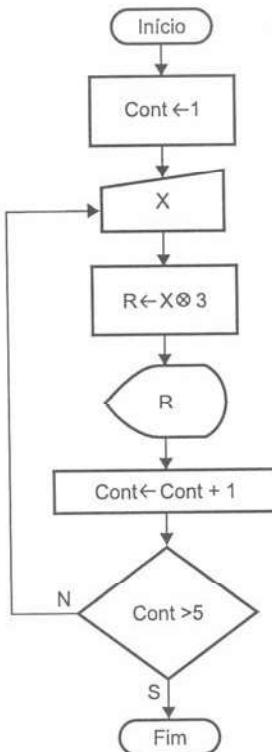


Figura 5.4 - Exemplo da utilização da instrução repita...até que.

Português Estruturado

```

algoritmo "Looping_2A"
var
  X, R: inteiro
  CONT: inteiro

  inicio
    CONT <- 1
    repita
      leia (X)
      R <- X * 3
      escreval (R)
      CONT <- CONT + 1
    ate (CONT > 5)
  fimalgoritmo
  
```

Logo após o início do programa, a variável contador é atribuída com o valor 1 (`CONT <- 1`). Em seguida, a instrução **repita** indica que todo trecho de instruções situado até a instrução **ate** (`CONT > 5`) deverá ter seu processamento repetido até que a condição (`CONT > 5`) seja satisfeita.

Sendo assim, tem início a execução da rotina de instruções contidas entre as instruções **repita...ate**. Depois de efetuar a primeira execução das instruções o programa encontra a linha `CONT <- CONT + 1`; neste momento a variável `CONT` é somada a mais 1, passando a ter o valor 2. Em seguida é encontrada a linha com a instrução **ate** (`CONT > 5`), que efetuará o retorno à instrução **repita** para que seja reprocessada a seqüência de comandos, até que o valor da variável `CONT` seja maior que 5 e encerre o looping.

A seguir, é apresentado o exemplo em que não se utiliza o contador como forma de controle de execução de vezes de uma rotina em uma estrutura de repetição. Considere que será o usuário que encerrará o processamento segundo a sua vontade.

Algoritmo

1. Iniciar o programa e o modo de laço repita;
2. Ler um valor para a variável X ;

3. Efetuar a multiplicação do valor de X por 3, implicando o resultado em R;
4. Apresentar o valor calculado contido na variável R;
5. Solicitar do usuário se este deseja ou não continuar o programa;
6. Repetir os passos 2, 3, 4 e 5 até que a resposta do usuário seja diferente de **sim**.

Diagrama de Blocos

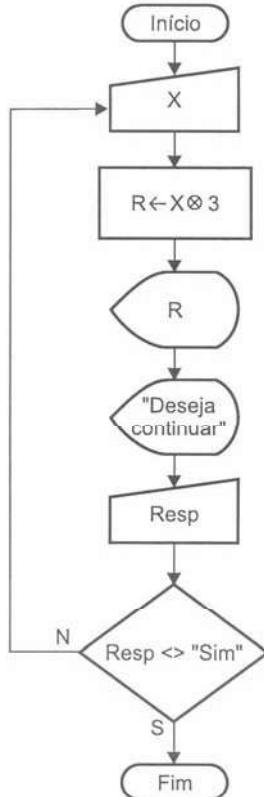


Figura 5.5 - Exemplo de um looping controlado pelo usuário.

Português Estruturado

```

algoritmo "Looping_2B"
var
  X, R: inteiro
  RESP: caracter

inicio
  RESP <- "sim"
  repita
    leia(X)
    R <- X * 3
    escreval(R)
    escreva("Deseja continuar? ")
    leia(RESP)
  ate (RESP <> "sim")
fimalgoritmo
  
```

Veja que a variável *RESP* não precisou ser inicializada com algum valor, pois mesmo que o tivesse feito, não exerteria influência direta sobre a estrutura **repita**, uma vez que no mínimo são sempre executadas as instruções constantes no looping, pra somente no final ser verificada a condição, no caso *RESP <> "sim"*.

5.2.1 - Exercícios de Entrega Obrigatória (até ____/____/____) (Lista 04)

1) Desenvolva os algoritmos, seus respectivos diagramas de bloco e codificação em português estruturado. Usar na resolução dos problemas apenas estruturas de repetição do tipo **repita** (Você deve gravar o exercício “a” como L04A, o exercício “b” como L04B, e assim por diante).

- a) Apresentar os quadrados dos números inteiros de 15 a 200.
- b) Elaborar um programa que apresente no final o somatório dos valores pares existentes na faixa de 1 até 500.
- c) Apresentar todos os números divisíveis por 4 que sejam menores que 200. Para verificar se o número é divisível por 4, efetuar dentro da malha a verificação lógica desta condição com a instrução **se**, perguntando se o número é divisível; sendo, mostre-o; não sendo, passe para o próximo passo. A variável que controlará o contador deve ser iniciada com o valor 1.
- d) Elaborar um programa que efetue o cálculo e no final apresente o somatório do número de grãos de trigo que se pode obter num tabuleiro de xadrez, obedecendo à seguinte regra: colocar um grão de trigo no primeiro quadro e nos quadros seguintes o dobro do quadro anterior. Ou seja, no primeiro quadro coloca-se 1 grão, no segundo quadro colocam-se 2 grãos (neste momento têm-se 3 grãos), no terceiro quadro colocam-se 4 grãos (tendo neste momento 7 grãos), no quarto colocam-se 8 grãos (tendo-se então 15 grãos) até atingir o sexagésimo quarto (64°) quadro. Utilize variáveis do tipo **real** como acumuladores.
- e) Elaborar um programa que efetue a leitura de 15 valores numéricos inteiros e no final apresente o total do somatório da fatorial de cada valor lido.
- f) Elaborar um programa que efetue a leitura sucessiva de valores numéricos e apresente no final o total do somatório, a média aritmética e o total de valores lidos. O programa deve fazer as leituras dos valores enquanto o usuário estiver fornecendo valores positivos. Ou seja, o programa deve parar quando o usuário fornecer um valor negativo. Não se esqueça que o usuário pode entrar como primeiro número um número negativo, portanto, cuidado com a divisão por zero no cálculo da média.
- g) Elaborar um programa que apresente como resultado o valor do fatorial dos valores ímpares situados na faixa numérica de 1 a 10.
- h) Elaborar um programa que possibilite calcular a área total de uma residência (sala, cozinha, banheiro, quartos, área de serviço, quintal, garagem, etc.). O programa deve solicitar a entrada do nome, a largura e o comprimento de um determinado cômodo. Em seguida, deve apresentar a área do cômodo lido e também uma mensagem solicitando do usuário a confirmação de continuar calculando novos cômodos. Caso o usuário responda “NAO”, o programa deve apresentar o valor total acumulado da área residencial.
- i) Elaborar um programa que efetue a leitura de valores positivos inteiros até que um valor negativo seja informado. Ao final devem ser apresentados o maior e o menor valores informados pelo usuário.
- j) Elaborar um programa que apresente o resultado inteiro da divisão de dois números quaisquer. Para a elaboração do programa, não utilizar em hipótese alguma o conceito do operador aritmético DIV. A solução deve ser alcançada com a utilização de looping. Ou seja, o programa deve apresentar como resultado (quociente) quantas vezes o divisor cabe no dividendo.

5.3 – Repetição do Tipo: Variável de Controle

Anteriormente, foram vistas duas formas de elaborar looping. Uma usando o conceito **enquanto** e a outra usando o conceito **repita**. Foi visto também como elaborar rotinas que efetuaram a execução de um looping um determinado número de vezes com a utilização de um contador (por meio de uma variável de controle).

Porém, existe uma possibilidade de facilitar o uso de contadores finitos sem fazer uso das duas estruturas anteriores, deixando-as para utilização de loopings em que não se conhece de antemão o número de vezes que uma determinada seqüência de instruções deverá ser executada. Os loopings que possuem um número

finito de execuções poderão ser processados por meio de estrutura de laços contados do tipo **para**, sendo conseguida com a utilização do conjunto de instruções **para...de...ate...passo...faca...fimpara**.

A estrutura **para...de...ate...passo...faca...fimpara** tem o seu funcionamento controlado por uma variável denominada contador. Sendo assim, poderá executar um determinado conjunto de instruções um determinado número de vezes.

Com relação à apresentação gráfica em um diagrama de blocos, existem várias formas adotadas por diversos profissionais no mercado, sendo assim, não existe um padrão de montagem do diagrama.

Diagrama de Blocos

Com relação ao diagrama de blocos, será indicada a variável a ser controlada com a implicação dos valores de início, fim e incremento, separados por vírgula. Para representar esta operação em um diagrama de blocos, será utilizado o símbolo denominado *Processamento predefinido* ou *Preparação*, representado pela figura de um hexágono.

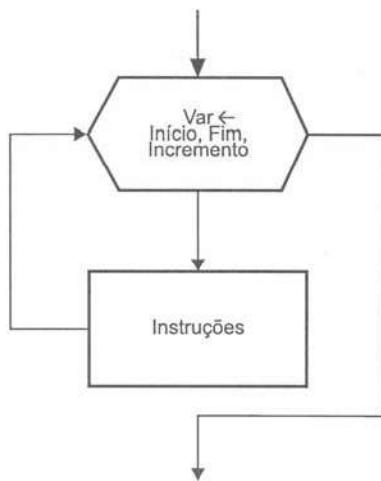


Figura 5.6 - Estrutura do símbolo da instrução *para...fim_for*.

Português Estruturado

```
para <variável> de <início> ate <fim> passo <incremento> faca  
    <instruções>  
fimpara
```

Para exemplificar, considere o problema anterior: “Pedir a leitura de um valor para a variável X, multiplicar esse valor por 3, implicando-o à variável de resposta R, e apresentar o valor obtido, repetindo esta seqüência por cinco vezes”.

Algoritmo

1. Definir um contador, variando de 1 a 5;
2. Ler um valor para a variável X;
3. Efetuar a multiplicação do valor de X por 3, implicando o resultado em R;
4. Apresentar o valor calculado, contido na variável R;
5. Repetir os passos 2, 3, 4 e 5 até que o contador seja encerrado.

Diagrama de Blocos

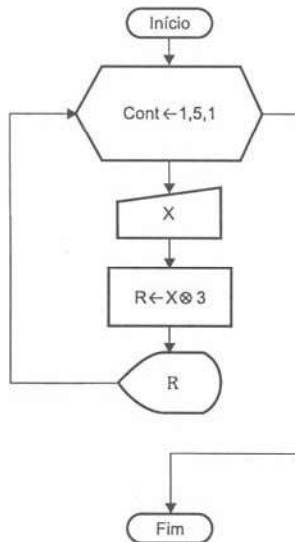


Figura 5.7 - Exemplo da utilização da estrutura para...de...até...passo...faça...fim para.

Português Estruturado

```

algoritmo "Looping_2A"
var
  X, R: inteiro
  CONT: inteiro

  inicio
    para CONT de 1 ate 5 passo 1 faca
      leia(X)
      R <- X * 3
      escreval(R)
    fimpara
  finalgoritmo
  
```

Será executado o conjunto de instruções entre **para** e a instrução **fimpara**, sendo a variável CONT (variável de controle) inicializada com valor 1 e incrementada de mais 1 por meio da instrução **passo** até o valor 5. Este tipo de estrutura de repetição poderá ser utilizado todas as vezes que houver a necessidade de repetir trechos finitos, em que se conhecem os valores inicial e final.

5.4 – Considerações entre os Tipos de Estrutura de Looping

No decorrer deste capítulo, foram apresentadas três estruturas de controle em nível de repetição: **enquanto**, **repita** e **para**, cada qual com sua característica de processamento. Dentro deste aspecto, deve-se notar que as estruturas mais versáteis são **enquanto** e **repita**, pois podem ser substituídas uma pela outra além de poderem substituir perfeitamente a estrutura **para**. Porém há de considerar-se que nem toda estrutura **enquanto** ou **repita** poderá ser substituída por uma estrutura **para**. Isto ocorre quando em uma estrutura utilizam-se condições que não envolvam o uso de variáveis de controle como contador.

5.5 – Estruturas de Controle Encadeadas

No capítulo anterior, foi discutido o fato de ocorrer o encadeamento das estruturas de decisão. Este fato pode também ocorrer com as estruturas de repetição. E neste ponto poderá ocorrer o encadeamento de um tipo de estrutura de repetição com outro tipo de estrutura de repetição. A existência destas ocorrências vai depender do problema a ser solucionado.

Devemos aqui salientar um pequeno detalhe. Nunca procure decorar estas regras, pois isto é impossível. Você precisa conhecer os comandos de entrada, processamento e saída, as estruturas de decisão e de repetição. Desta forma, conhecendo bem, você saberá utilizá-las no momento que for conveniente, pois na

resolução de um problema, ele “pedirá” a estrutura mais conveniente. E, você, conhecendo-as bem, saberá automaticamente o momento certo de utilizar-as.

Para exemplificar os tipos de aninhamento que poderão ser combinados, observe a seguir os exemplos em nível de diagrama de blocos e português estruturado. Ainda não serão apresentados exemplos da utilização prática destes aninhamentos. Mais adiante você terá contato com estes detalhes que ocorrerão de forma automática.

5.5.1 – Encadeamento de Estruturas Enquanto com Enquanto

Exemplo de encadeamento de estrutura **enquanto** com estrutura **enquanto**. A figura a seguir mostra a estrutura do desenho do diagrama de blocos.

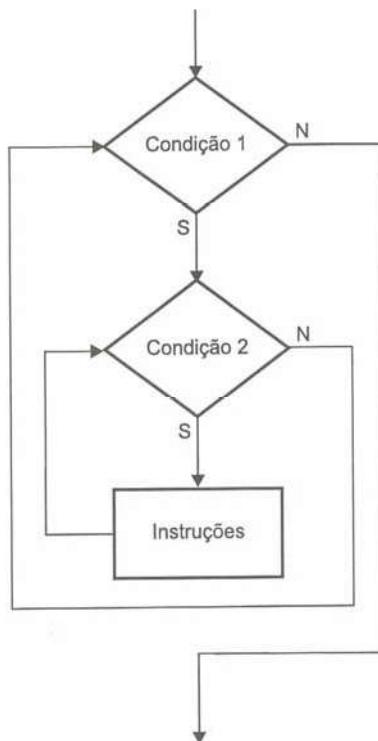
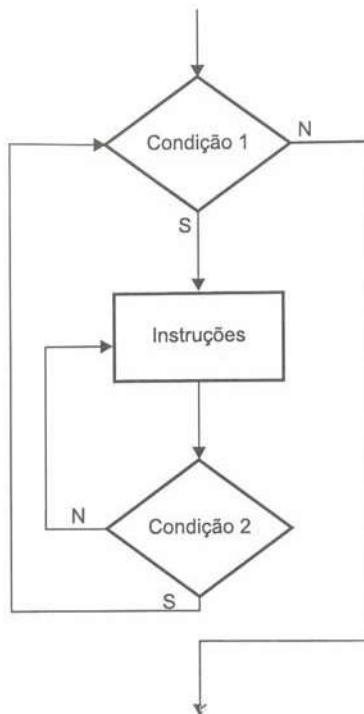


Figura 5.8 - Encadeamento de enquanto com enquanto.

Português Estruturado

```
enquanto (<condição1>) faça
  enquanto (<condição2>) faça
    <instruções>
  fimenquanto
fimenquanto
```

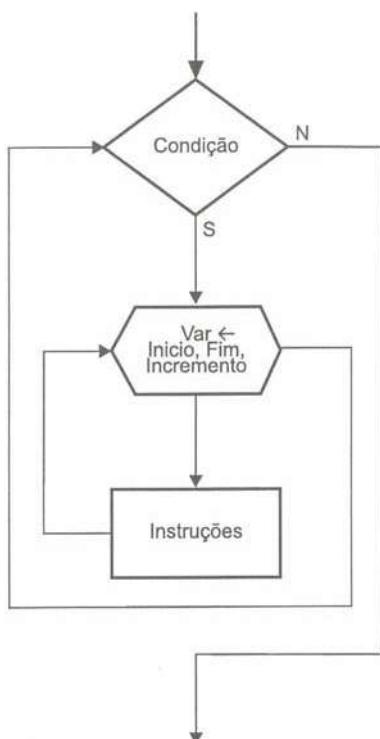
5.5.2 – Encadeamento de Estrutura Enquanto com Repita



Português Estruturado

```
enquanto (<condição1>) faça
    repita
        <instruções>
    ate (<condição2>)
fimenquanto
```

5.5.3 – Encadeamento de Estrutura Enquanto com Para

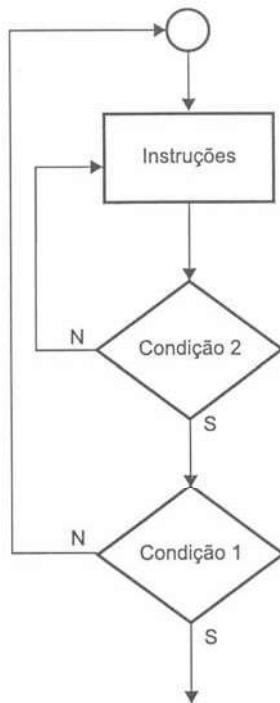


Português Estruturado

```
enquanto (<condição>) faça
```

```
para <var> de <início> ate <fim> passo <incremento> faça  
  <instruções>  
fimpara  
fimenquanto
```

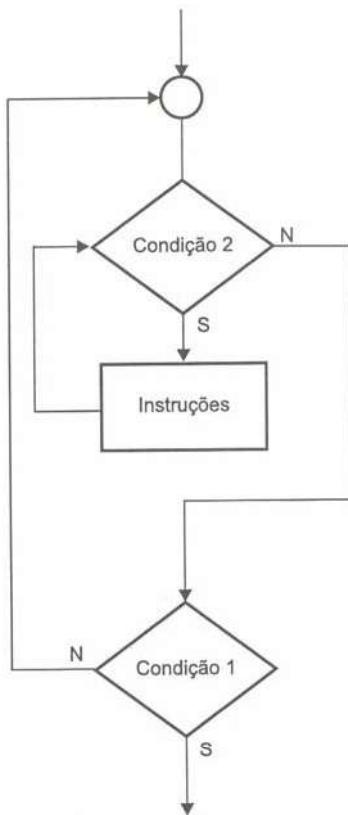
5.5.4 – Encadeamento de Estrutura Repita com Repita



Português Estruturado

```
repita  
  repita  
    <instruções>  
    ate (<condição>)  
  ate (<condição>)
```

5.5.5 – Encadeamento de Estrutura Repita com Enquanto



Português Estruturado

```
repita
    enquanto (<condição2>) faça
        <instruções>
    fimenquanto
ate (<condição1>)
```

5.5.6 – Encadeamento de Estrutura Repita com Para

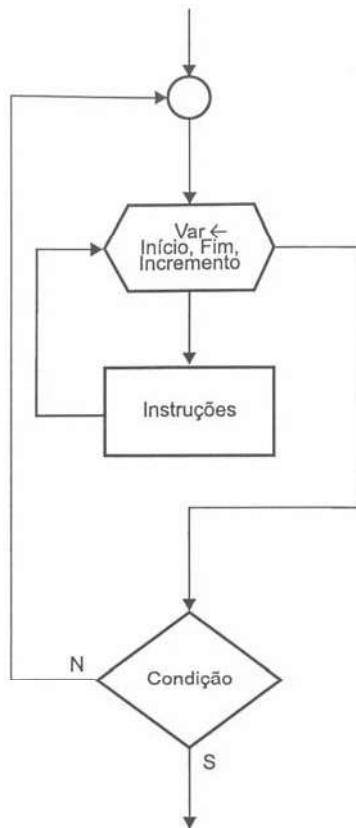


Figura 5.13 - Encadeamento de repita com para.

Português Estruturado

```
repita
  para <var> de <início> ate <fim> passo <incremento> faca
    <instruções>
  fimpara
  ate (<condição>)
```

5.5.7 – Encadeamento de Estrutura Para com Para

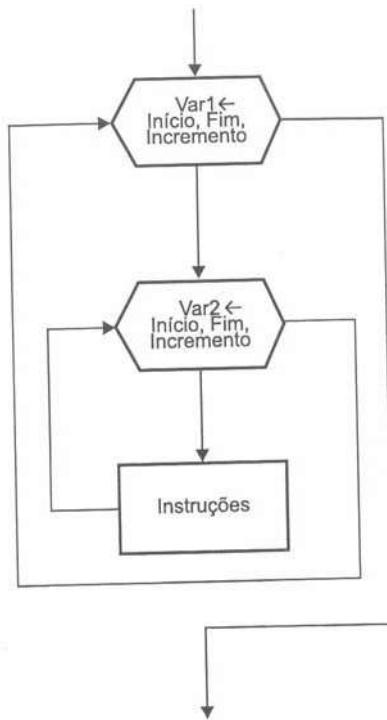


Figura 5.14 - Encadeamento de para com para.

Português Estruturado

```
para <var1> de <início> ate <fim> passo <incremento> faca
  para <var2> de <início> ate <fim> passo <incremento> faca
    <instruções>
  fimpara
fimpara
```

5.5.8 – Encadeamento de Estrutura Para com Enquanto

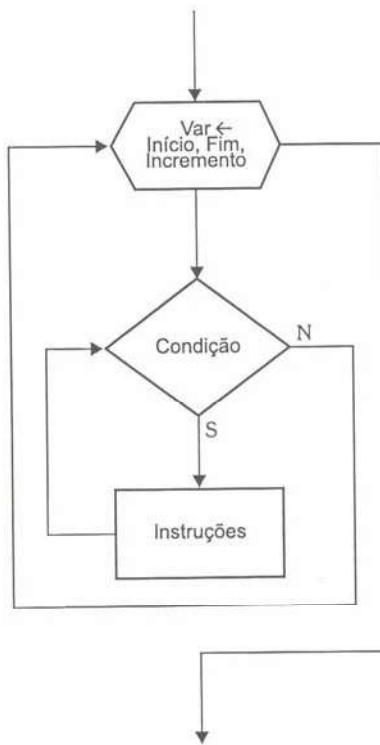


Figura 5.15 - Encadeamento de para com enquanto.

Português Estruturado

```
para <variavel> de <início> ate <fim> passo <incremento> faça
    enquanto (<condição>) faça
        <instruções>
    fimenquanto
fimpara
```

5.5.9 – Encadeamento de Estrutura Para com Repita

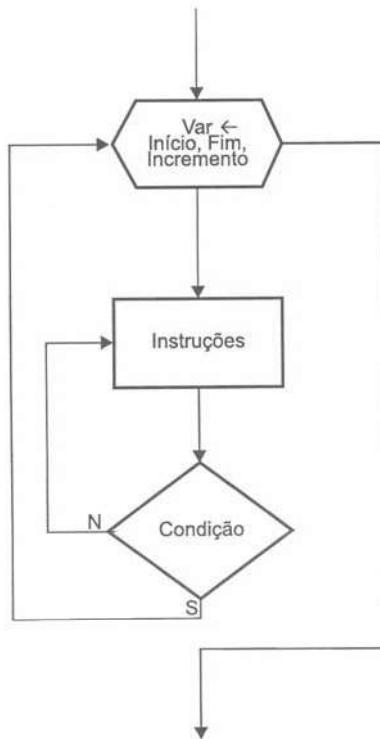


Figura 5.16 - Encadeamento de para com repita.

Português Estruturado

```

para <variavel> de <início> ate <fim> passo <incremento> faca
  repita
    <instruções>
    ate (condição)
fimpara
  
```

5.6 – Exercício de Aprendizagem

A seguir, são apresentados alguns exemplos da utilização das estruturas de repetição. Considere o seguinte problema: “Elaborar o algoritmo, diagrama de blocos e codificação em português estruturado de um programa que efetue o cálculo do fatorial do número 5, 5!”. Desta forma, temos que $5! = 5 \times 4 \times 3 \times 2 \times 1$ ou seja $5! = 120$.

Fatorial é o produto dos números naturais desde 1 até o inteiro n. Sendo assim, o cálculo de um fatorial é conseguido pela multiplicação sucessiva do número de termos. No caso do cálculo de uma fatorial de número 5, este é o número de termos a ser utilizado. Desta forma, o programa deverá executar as multiplicações sucessivamente e acumulá-las a fim de possuir o valor 120 após 5 passos. O número de passos deverá ser controlado por um contador.

Algoritmo

1. Inicializar as variáveis FATORIAL e CONTADOR com 1;
2. Multiplicar sucessivamente a variável FATORIAL pela variável CONTADOR;
3. Incrementar 1 à variável CONTADOR, efetuando o controle até 5;
4. Apresentar ao final o valor obtido.

Pelo fato de ter que efetuar o cálculo de uma factorial de 5 ($5!$), isto implica que o contador deverá variar de 1 a 5, e por este motivo deverá ser a variável CONTADOR inicializada com valor 1. Pelo fato de a variável FATORIAL possuir ao final o resultado do cálculo da factorial pretendida, esta deverá ser inicializada com valor 1. Se ela for incializada com zero, não existirá resultado final, pois qualquer valor multiplicado por zero resulta zero.

Multiplicar sucessivamente implica em efetuar a multiplicação da variável CONTADOR com o valor atual da variável FATORIAL a cada passo controlado pelo looping. No caso, por cinco vezes. Abaixo é indicado esta ocorrência na linha 2 do algoritmo.

1. Inicializar as variáveis FATORIAL e CONTADOR com 1;
2. Repetir a execução dos passos 3 e 4 por cinco vezes;
3. $FATORIAL \leftarrow FATORIAL * CONTADOR;$
4. Incrementar 1 à variável CONTADOR;
5. Apresentar ao final o valor obtido.

A seguir a resolução do problema do cálculo de factorial, utilizando as estruturas de repetição, sendo apresentados os diagramas de blocos e a codificação em português estruturado de cada estrutura.

Exemplo 1 – Estrutura de repetição: Enquanto

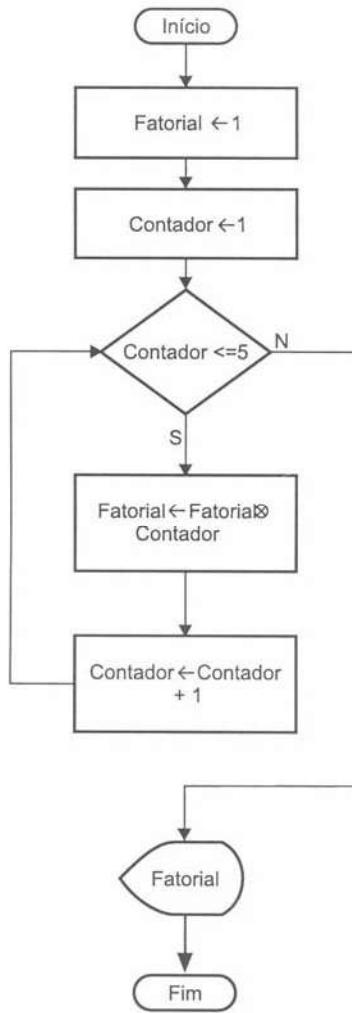


Figura 5.17 - Fatorial com estrutura enquanto.

Observe dentro do looping a indicação de dois contadores, sendo que o primeiro funciona como um acumulador, pois ele terá no final o valor do resultado do factorial, e o segundo sendo utilizado para controlar a execução do looping a ser a base para o cálculo do acumulador.

Logo no início do diagrama de blocos, as variáveis CONTADOR e FATORIAL são igualadas em 1. Na primeira passagem dentro do looping, a variável FATORIAL é implicada pelo seu valor atual, no caso 1, multiplicado pelo valor da variável CONTADOR também com valor 1 que resulta 1. Em seguida a variável

CONTADOR é incrementada por mais 1, tendo agora o valor 2. Como 2 é menor ou igual a cinco, ocorre um novo cálculo. Desta vez a variável FATORIAL que possui o valor 1 é multiplicada pela variável CONTADOR que possui o valor 2, resultando 2 para o FATORIAL. Daí a variável CONTADOR é incrementada de mais 1, tendo agora o valor 3. Desta forma, serão executados os outros cálculos até que a condição se torne falsa e seja então apresentado o valor 120. Veja abaixo, a tabela com os valores das variáveis antes e durante a execução do looping:

CONTADOR	FATORIAL	FATORIAL <- FATORIAL*CONTADOR	Comentários
1	1	1	Valor inicial das variáveis
2	1	2	Cálculo do fatorial com contador em 2
3	2	6	Cálculo do fatorial com contador em 3
4	6	24	Cálculo do fatorial com contador em 4
5	24	120	Cálculo do fatorial com contador em 5

Perceba que quando a variável CONTADOR está com valor 5, a variável FATORIAL está com o valor 120. Neste ponto, o looping é encerrado e é apresentado o valor da variável FATORIAL.

Português Estruturado

```

algoritmo "Fatorial_A"
var
    CONTADOR : inteiro
    FATORIAL : inteiro

    inicio
        FATORIAL <- 1
        CONTADOR <- 1
        enquanto (CONTADOR <=5) faca
            FATORIAL <- FATORIAL * CONTADOR
            CONTADOR <- CONTADOR + 1
        fimenquanto
        escreva("Fatorial de 5 é = ", FATORIAL)
    finalgoritmo

```

No código acima, estão sendo apresentadas pela instrução **escreva()** duas informações, sendo uma mensagem e uma variável. Note que isto é feito utilizando uma vírgula para separar as duas informações.

Exemplo 2 – Estrutura de repetição: Repita

Resolução do problema fazendo uso da estrutura de repetição **repita**.

Diagrama de Blocos

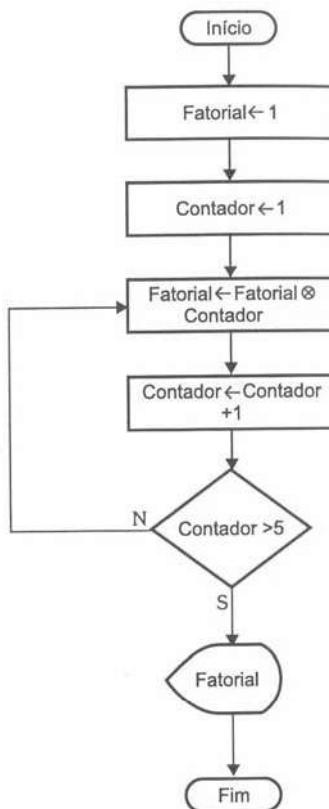


Figura 5.18 - Fatorial com estrutura **repita**.

Português Estruturado

```

algoritmo "Fatorial_B"
var
  CONTADOR : inteiro
  FATORIAL : inteiro

  inicio
    FATORIAL <- 1
    CONTADOR <- 1
    repita
      FATORIAL <- FATORIAL * CONTADOR
      CONTADOR <- CONTADOR + 1
    ate (CONTADOR > 5)
    escreva("Fatorial de 5 é = ", FATORIAL)
  fimalgoritmo
  
```

Exemplo 3 – Estrutura de repetição: Para

Resolução do problema fazendo uso da estrutura de repetição **para**.

Diagrama de Blocos

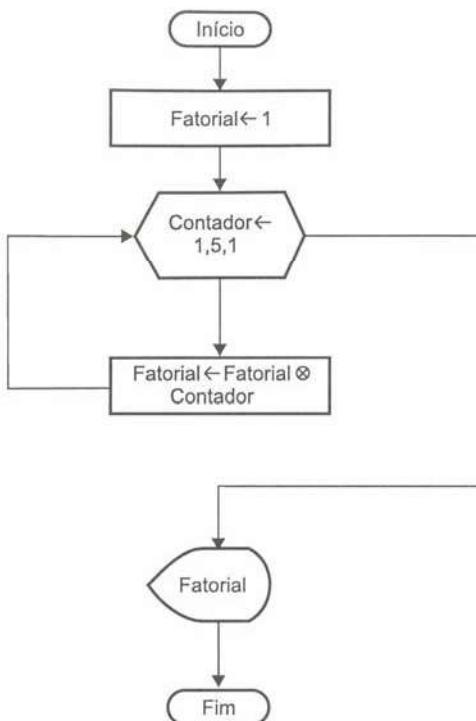


Figura 5.19 - Fatorial com estrutura para.

Português Estruturado

```

algoritmo "Fatorial_B"
var
  CONTADOR: inteiro
  FATORIAL: inteiro

  inicio
    FATORIAL <- 1
    para CONTADOR de 1 ate 5 passo 1 faca
      FATORIAL <- FATORIAL * CONTADOR
    fimpara
    escreva("Fatorial de 5 é igual a ", FATORIAL)
  finalgoritmo
  
```

Exemplo 4

Observe que o algoritmo apresentado anteriormente para a resolução do problema de fatorial soluciona o cálculo apenas para o fatorial de 5. Seria melhor possuir a solução aberta para que um programa calcule o fatorial de um número qualquer, e que pudesse calcular outros fatoriais até que o usuário não mais deseje utilizar o programa. Sendo assim, o programa deverá pedir ao usuário a sua continuidade ou não.

No código já existente, deverão ser criadas duas variáveis, sendo uma para a confirmação da continuidade da execução do programa e outra para determinar o cálculo do valor do fatorial. O exemplo seguinte faz uso do encadeamento de estruturas de repetição.

1. Inicializar as variáveis FATORIAL e CONTADOR com 1;
2. Declarar as variáveis RESP (resposta) para confirmação e N para receber o limite de valor para o cálculo do fatorial;
3. Enquanto RESP do usuário for sim, executar os passos 3, 4, 5, 6 e 7;
4. Repetir a execução dos passos 4 e 5 por N vezes;
5. FATORIAL <- FATORIAL * CONTADOR;
6. Incrementar 1 à variável CONTADOR;
7. Apresentar ao final o valor obtido.

Diagrama de Blocos

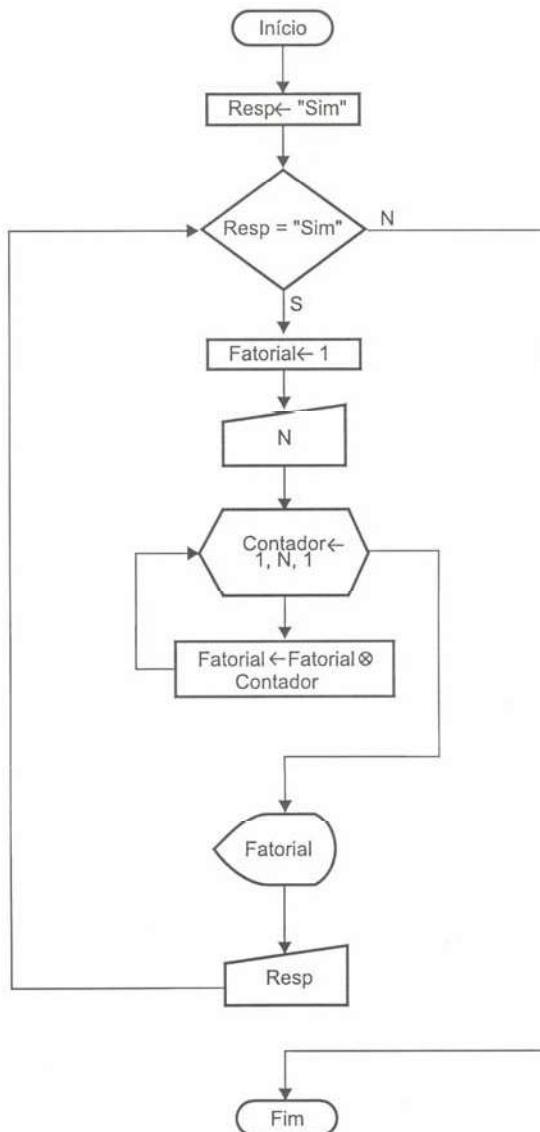


Figura 5.20 - Implementação do número de vezes e cálculo de qualquer fatorial.

Português Estruturado

```

algoritmo "Fatorial_D"
var
  CONTADOR: inteiro
  FATORIAL: inteiro
  RESP: literal
  N: inteiro

início
  RESP <- "sim"
  enquanto (RESP = "sim") faça
    FATORIAL <- 1
    escreva("Fatorial de que numero: ")
    leia(N)
    para CONTADOR de 1 ate N passo 1 faça
      FATORIAL <- FATORIAL * CONTADOR
    fimpara
    escreval("Fatorial de ", N, " é igual a ", FATORIAL)
    escreva("Deseja continua?")
    leia(RESP)
  fimenquanto
fimalgoritmo
  
```

5.6.1 - Exercícios de Entrega Obrigatória (até ____/____/____) (Lista 05)

1) Desenvolva os algoritmos, seus respectivos diagramas de bloco e codificação em português estruturado. Usar na resolução dos problemas apenas estruturas de repetição do tipo **para** (Você deve gravar o exercício “a” como L05A, o exercício “b” como L05B, e assim por diante).

- a) Apresentar os quadrados dos números inteiros de 15 a 200.
- b) Apresentar os resultados de uma tabuada de multiplicar (de 1 até 10) de um número qualquer.
- c) Apresentar o total da soma obtida dos cem primeiros números inteiros ($1+2+3+4+\dots+98+99+100$).
- d) Elaborar um programa que apresente no final o somatório dos valores pares existentes na faixa de 1 até 500.
- e) Apresentar todos os valores numéricos inteiros ímpares situados na faixa de 0 a 20. Para verificar se o número é ímpar, efetuar dentro da malha a verificação lógica desta condição com a instrução **se**, perguntando se o número é ímpar; sendo, mostre-o; não sendo, passe para o próximo passo.
- f) Apresentar todos os números divisíveis por 4 que sejam menores que 200. Para verificar se o número é divisível por 4, efetuar dentro da malha a verificação lógica desta condição com a instrução **se**, perguntando se o número é divisível; sendo, mostre-o; não sendo, passe para o próximo passo. A variável que controlará o contador deve ser iniciada com o valor 1.
- g) Apresentar os resultados das potências de 3, variando do expoente 0 até o expoente 15. Deve ser considerado que qualquer número elevado a zero é 1, e elevado a 1 é ele próprio. Observe que neste exercício não pode ser utilizado o operador de exponenciação do portuguol (^).
- h) Elaborar um programa que apresente como resultado o valor de uma potência de uma base qualquer elevada a um expoente qualquer, ou seja, de B^E , em que B é o valor da base e E o valor do expoente. Observe que neste exercício não pode ser utilizado o operador de exponenciação do portuguol (^).
- i) Escreva um programa que apresente a série de Fibonacci até o décimo quinto termo. A série de Fibonacci é formada pela seqüência: 1, 1, 2, 3, 5, 8, 13, 21, 34, ..., etc. Esta série se caracteriza pela soma de um termo atual com o seu anterior subsequente, para que seja formado o próximo valor da seqüência. Portanto começando com os números 1, 1 o próximo termo é $1+1=2$, o próximo é $1+2=3$, o próximo é $2+3=5$, o próximo $3+5=8$, etc.
- j) Elaborar um programa que apresente os valores de conversão de graus Celsius em Fahrenheit, de 10 em 10 graus, iniciando a contagem em 10 graus Celsius e finalizando em 100 graus Celsius. O programa deve apresentar os valores das duas temperaturas. A fórmula de conversão é $F = \frac{9C + 160}{5}$, sendo F a temperatura em Fahrenheit e C a temperatura em Celsius.
- k) Elaborar um programa que apresente como resultado o valor do fatorial dos valores ímpares situados na faixa numérica de 1 a 10.