

Unidade IV

Você gosta de fazer exatamente a mesma coisa centenas de vezes? Em algum momento, essa operação vai ficar muito monótona, se não levar à loucura.

4 LAÇOS DE REPETIÇÃO

Uma das funções do computador, entre outras, é fazer operações repetitivas, que para o ser humano seriam muito enfadonhas. São muito comuns, durante o processamento de um programa, situações em que pode existir a necessidade de repetir um determinado conjunto de comandos por um certo número de vezes. Por exemplo: durante o processamento da folha de pagamento de uma empresa, o mesmo cálculo do salário conforme as horas trabalhadas é efetuado para cada um dos funcionários. Para evitar a repetição desse cálculo, utiliza-se a estrutura de repetição. As estruturas de repetição são também denominadas **laços** (*loops*).

4.1 Teoria

Existem três comandos que executam a estrutura dos laços de repetição: o **enquanto**, o **repita** e o **para**, que veremos detalhadamente. Apesar de todos terem a mesma funcionalidade, a de fazer repetições, conforme o problema a ser resolvido, cada um deles apresenta uma característica mais adequada. Conforme o conhecimento prévio e a quantidade de laços executados, ou se o número de vezes em que o conjunto de instruções será executado (iteração) for indeterminado, as estruturas de repetição poderão ser classificadas em:

- laços contados: quando se conhece previamente quantas vezes o comando composto no interior da construção será executado;
- laços condicionais: quando não se conhece o número de vezes em que o conjunto de instruções no interior do laço será repetido, pois a condição testada é modificada pelas instruções do interior do laço.

É importante ter em mente que os laços de repetição têm um funcionamento automatizado, muito particular em cada uma das formas, e o entendimento desse mecanismo é fundamental para montar corretamente os programas.

A nomenclatura técnica para a execução de um laço completo é **iteração**.

4.1.1 Laços condicionais

Quando existem blocos de programas que necessitam ser repetidos, porém não sabemos quantas vezes isso ocorrerá, utilizamos o laço condicional. No caso, são dois: o laço do **repita até que** e o do **enquanto faça**.

4.1.1.1 Repita até que

No laço **repita**, o fluxo do programa encontra o comando propriamente dito e reconhece que é o início do bloco de repetição, seguindo o fluxo normal até encontrar o comando **até que**, no qual uma condição é testada. Se o resultado do teste for Falso, o fluxo será devolvido ao início do bloco. Se o resultado do teste for Verdadeiro, o programa seguirá o fluxo, abandonando a repetição do bloco. O importante é saber que, nessa estrutura, o processo será repetido enquanto a condição testada for falsa.

Sintaxe:

```
repita
```

```
<Comandos>
```

```
Até que <Condição>
```

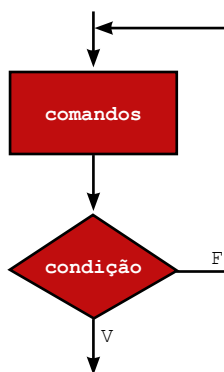


Figura 134 – Fluxograma do comando **repita até que**

Exemplo de aplicação

Faça um algoritmo que processe a seguinte enquete: **"Você tem computador em casa?"**, mostrando o número de pessoas que não possuem e o das pessoas que possuem computador. Para sair, dê a opção de escolha.

```
algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
```

```

se voto="s" entao
    cont_s<-cont_s+1
senao
    cont_n<-cont_n+1
fimse
escreva ("Deseja continuar? s/n")
leia (resp)
ate resp="n"
escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
    
```

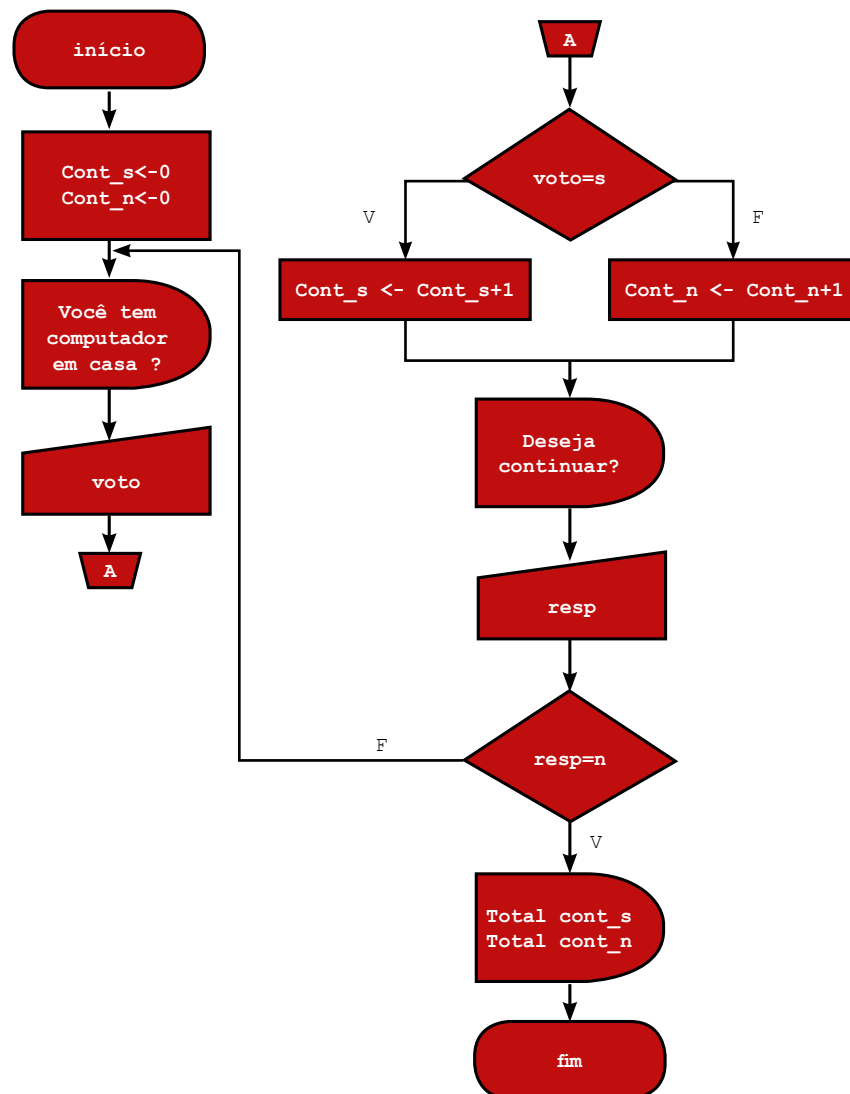


Figura 135 – Programa de pesquisa usando **repita**

Uma vez montado o programa, o passo seguinte é verificar se está funcionando adequadamente. Para isso, vamos fazer o teste de mesa, com duas entradas positivas para ter computador em casa e uma entrada negativa, e encerrar o processamento.

algoritmo "enquete"

```
var
    voto,resp: caractere
    cont_s,cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    ate resp="n"
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
```

Memoria: enquete

Figura 136 - Início com memória vazia

algoritmo "enquete"

Var

voto,resp: caractere
cont_s,cont_n: inteiro

Início

cont_s<-0
cont_n<-0

Repita

escreva ("Você tem computador em casa? s/n")
leia (voto)
se voto="s" entao
 cont_s<-cont_s+1
Senão
 cont_n<-cont_n+1

Fimse

escreva ("Deseja continuar? s/n")
leia (resp)
ate resp="n"

```

    escreva("total sim: ", cont_s, "total nao", cont_n)
finalgoritmo

```

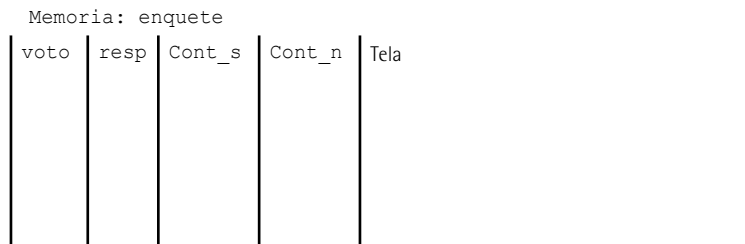


Figura 137 – O espaço para as variáveis é criado

```

algoritmo "enquete"
Var
    voto, resp: caractere
    cont_s, cont_n: inteiro
Inicio
    cont_s<-0
    cont_n<-0
    Repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        Senão
            cont_n<-cont_n+1
        Fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
        ate resp="n"
    escreva("total sim: ", cont_s, "total nao", cont_n)
finalgoritmo

```

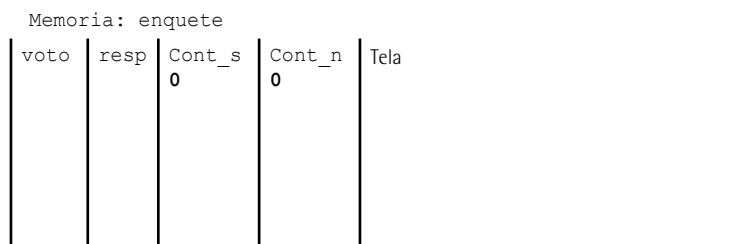


Figura 138 – As variáveis são inicializadas

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0

```

```

cont_n<-0
repita
  escreva ("Você tem computador em casa? s/n")
  leia (voto)
  se voto="s" então
    cont_s<-cont_s+1
  Senão
    cont_n<-cont_n+1
  Fimse
  escreva ("Deseja continuar? s/n")
  leia (resp)
  ate resp="n"
  escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo

```

Memoria: encuesta				
voto	resp	Cont_s	Cont_n	Tela
		0	0	Você tem computador em casa? s/n

Figura 139 – A mensagem é mostrada na tela

```

algoritmo "enquete"
var
  voto,resp: caractere
  cont_s,cont_n: inteiro
inicio
  cont_s<-0
  cont_n<-0
  repita
    escreva ("Você tem computador em casa? s/n")
    leia (voto)
    se voto="s" entao
      cont_s<-cont_s+1
    senao
      cont_n<-cont_n+1
    fimse
    escreva ("Deseja continuar? s/n")
    leia (resp)
    ate resp="n"
    escreva("total sim: ",cont_s, "total nao", cont_n)
  finalgoritmo

```

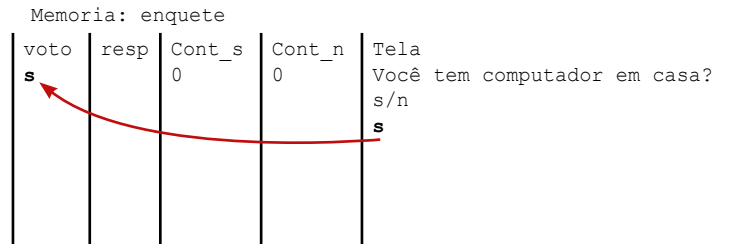



Figura 140 – Leitura do teclado

```

algoritmo "enquete"
Var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    ate resp="n"
    escreva("total sim: ", cont_s, "total nao", cont_n)
finalgoritmo
    
```

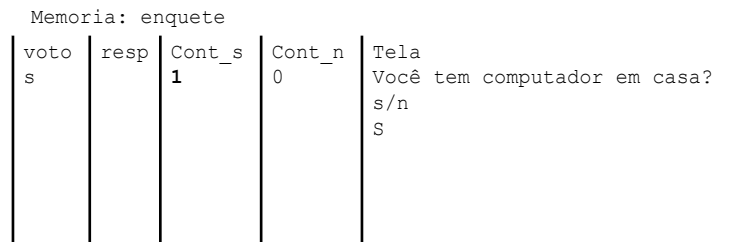


Figura 141 – Condicional para a contagem da opção

```

algoritmo "enquete"
var
    voto,resp: caractere
    cont_s,cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
        ate resp="n"
    escreva("total sim: ",cont_s, "total nao", cont_n)
fimalgoritmo

```

Memoria: enquete					
voto	resp	Cont_s	Cont_n	Tela	
s		1	0	Você tem computador em casa?	
				s/n	
				S	
				Deseja continuar? s/n	

Figura 142 – Exibição da pergunta

```

algoritmo "enquete"
Var
    voto,resp: caractere
    cont_s,cont_n: inteiro
Inicio
    cont_s<-0
    cont_n<-0
    Repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" então
            cont_s<-cont_s+1
        Senão
            cont_n<-cont_n+1
        Fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
        ate resp="n"
    escreva("total sim: ",cont_s, "total nao", cont_n)
fimalgoritmo

```

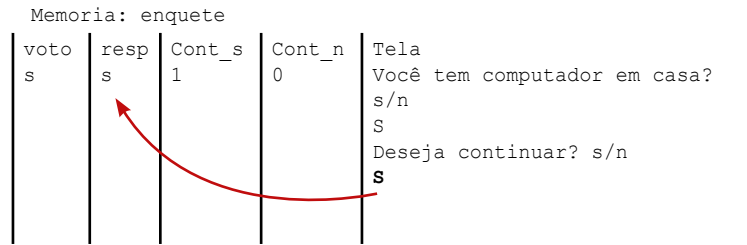



Figura 143 – Nova leitura do teclado

```

algoritmo "enquete"
Var
    voto,resp: caractere
    cont_s,cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    ate resp="n"
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
    
```

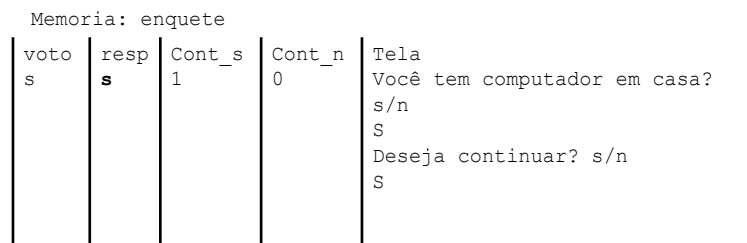


Figura 144 – Devolve ao início do laço

```

algoritmo "enquete"
Var
    voto,resp: caractere
    cont_s,cont_n: inteiro
Inicio
    cont_s<-0
    cont_n<-0
    Repita
        escreva ("Você tem computador em casa? s/n")
    
```

```

leia (voto)
se voto="s" entao
    cont_s<-cont_s+1
Senão
    cont_n<-cont_n+1
Fimse
escreva ("Deseja continuar? s/n")
leia (resp)
ate resp="n"
escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo

```

Memoria: encuesta				
voto	resp	Cont_s	Cont_n	Tela
s	s	1	0	Você tem computador em casa?
				s/n
				S
				Deseja continuar? s/n
				S
				Você tem computador em casa?
				s/n

Figura 145 - Mensagem na tela

```

algoritmo "enquete"
Var
    voto,resp: caractere
    cont_s,cont_n: inteiro
Inicio
    cont_s<-0
    cont_n<-0
    repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    ate resp="n"
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo

```

Memoria: encuesta

voto	resp	Cont_s	Cont_n	Tela
s	s	1	0	Você tem computador em casa?
n				s/n
				S
				Deseja continuar? s/n
				S
				Você tem computador em casa?
				s/n
				n

Figura 146 – Outra leitura do teclado

```

algoritmo "enquete"
var
    voto,resp: caractere
    cont_s,cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    ate resp="n"
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
    
```

Memoria: encuesta

voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
n				s/n
				S
				Deseja continuar? s/n
				S
				Você tem computador em casa?
				s/n
				n

Figura 147 – Condicional para escolher em que variável acumulará os não

```

algoritmo "enquete"
var
    voto,resp: caractere
    cont_s,cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    
```

```
repita
  escreva ("Você tem computador em casa? s/n")
  leia (voto)
  se voto="s" entao
    cont_s<-cont_s+1
  senao
    cont_n<-cont_n+1
fimse
escreva ("Deseja continuar? s/n")
leia (resp)
ate resp="n"
escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
```

Memoria: encuesta				
voto	resp	Cont_s	Cont_n	Tela
n	s	1	1	Você tem computador em casa? s/n S Deseja continuar? s/n S Você tem computador em casa? s/n N Deseja continuar? s/n

Figura 148 - Display na tela

```
algoritmo "enquete"
var
  voto,resp: caractere
  cont_s,cont_n: inteiro
inicio
  cont_s<-0
  cont_n<-0
  repita
    escreva ("Você tem computador em casa? s/n")
    leia (voto)
    se voto="s" entao
      cont_s<-cont_s+1
    senao
      cont_n<-cont_n+1
  fimse
  escreva ("Deseja continuar? s/n")
  leia (resp)
  ate resp="n"
  escreva("total sim: ",cont_s, "total nao", cont_n)
Fimalgoritmo
```

Memoria: encuesta

voto	resp	Cont_s	Cont_n	Tela
n	s	1	1	Você tem computador em casa?
	s			s/n
				S
				Deseja continuar? s/n
				S
				Você tem computador em casa?
				s/n
				N
				Deseja continuar? s/n
				s

Figura 149 – Mais uma leitura do teclado

```

algoritmo "enquete"
Var
    voto, resp: caractere
    cont_s, cont_n: inteiro
Inicio
    cont_s<-0
    cont_n<-0
    Repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        Senão
            cont_n<-cont_n+1
        Fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    ate resp="n"
    escreva("total sim: ", cont_s, "total nao", cont_n)
fimalgoritmo
    
```

Memoria: encuesta

voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
n	s			s/n
				S
				Deseja continuar? s/n
				S
				Você tem computador em casa?
				s/n
				N
				Deseja continuar? s/n
				s

Figura 150 – Retorno ao inicio do laço

```

algoritmo "enquete"
var
    voto,resp: caractere
    cont_s,cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    ate resp="n"
    escreva("total sim: ",cont_s, "total nao", cont_n)
Fimalgoritmo

```

Memoria: enquete				
voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
n	s			s/n
				S
				Deseja continuar? s/n
				S
				Você tem computador em casa?
				s/n
				N
				Deseja continuar? s/n
				s
				Você tem computador em casa?
				s/n

Figura 151 – Exibição da mensagem na tela

```

algoritmo "enquete"
var
    voto,resp: caractere
    cont_s,cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao

```

```

        cont_n<-cont_n+1
    fimse
    escreva ("Deseja continuar? s/n")
    leia (resp)
    ate resp="n"
    escreva("total sim: ",cont_s, "total nao", cont_n)
fim algoritmo

```

Memoria: encuesta

voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
n	s			s/n
s				S
				Deseja continuar? s/n
				S
				Você tem computador em casa?
				s/n
				N
				Deseja continuar? s/n
				s
				Você tem computador em casa?
				s/n
				s

Figura 152 – Leitura do teclado

```

algoritmo "enquete"
Var
    voto,resp: caractere
    cont_s,cont_n: inteiro
Inicio
    cont_s<-0
    cont_n<-0
    Repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        Senão
            cont_n<-cont_n+1
        Fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
        ate resp="n"
        escreva("total sim: ",cont_s, "total nao", cont_n)
fim algoritmo

```

Memoria: encuesta

voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
n	s	2		s/n
s				S
				Deseja continuar? s/n
				S
				Você tem computador em casa?
				s/n
				N
				Deseja continuar? s/n
				s
				Você tem computador em casa?
				s/n
				s

Figura 153 – Condicional para acumular na variável de contagem de sim

```

algoritmo "enquete"
Var
    voto,resp: caractere
    cont_s,cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    ate resp="n"
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo

```

Memoria: encuesta

voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
n	s	2		s/n
s				S
				Deseja continuar? s/n
				S
				Você tem computador em casa?
				s/n
				N
				Deseja continuar? s/n
				s
				Você tem computador em casa?
				s/n
				s
				Deseja continuar? s/n

Figura 154 – Mensagem exibida na tela


```

algoritmo "enquete"
var
    voto,resp: caractere
    cont_s,cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    ate resp="n"
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
    
```

Memoria: enquete

voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
n	s	2		s/n
s	n			S
				Deseja continuar? s/n
				S
				Você tem computador em casa?
				s/n
				N
				Deseja continuar? s/n
				s
				Você tem computador em casa?
				s/n
				s
				Deseja continuar? s/n
				n

Figura 155 - Leitura do teclado: desta vez, a resposta para continuar é não

```

algoritmo "enquete"
var
    voto,resp: caractere
    cont_s,cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    repita
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
    
```

```

    cont_n<-cont_n+1
fimse
escreva ("Deseja continuar? s/n")
leia (resp)
ate resp="n"
escreva("total sim: ",cont_s, "total nao", cont_n)
fimalgoritmo

```

Memoria: encuesta				
voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
n	s	2		s/n
s	n			S
				Deseja continuar? s/n
				S
				Você tem computador em casa?
				s/n
				N
				Deseja continuar? s/n
				S
				Você tem computador em casa?
				s/n
				S
				Deseja continuar? s/n
				N
				total sim: 2 total não 1"

Figura 156 – Como a condição do fim das iterações foi alcançada, sai do laço

4.1.1.2 Enquanto faça

O laço **enquanto** funciona da seguinte maneira: ao início do bloco de repetição, uma condição é testada. Se o resultado do teste for Falso, então o bloco será ignorado, prosseguindo normalmente após o **fimenquanto**. Caso o resultado do teste seja Verdadeiro, o bloco será executado e, ao encontrar a instrução **fimenquanto**, o fluxo será devolvido à instrução **enquanto**, e a condição, testada novamente, até que se torne falsa. Resumindo, o processo será repetido enquanto a condição testada for verdadeira.



Observação

A variável deverá estar inicializada, pois a condição resultará em erro.

Sintaxe:

```
enquanto <Condição> faça
```

```
<Comandos>
```

```
fim enquanto
```

O fluxograma da estrutura **enquanto** é:

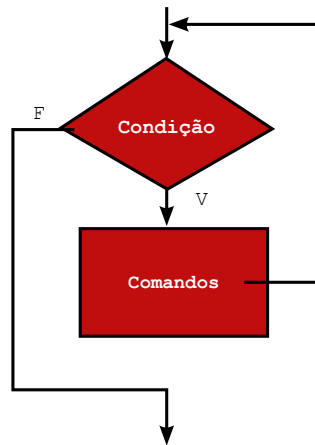


Figura 157 – Estrutura do **enquanto**

Exemplo de aplicação

Faça um algoritmo, usando o laço **enquanto**, que receba a seguinte enquete: **Você tem computador em casa?** Levante o número de pessoas que não possuem e o das que possuem. Para sair, dê a opção de escolha.

```
algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
fimalgoritmo
```

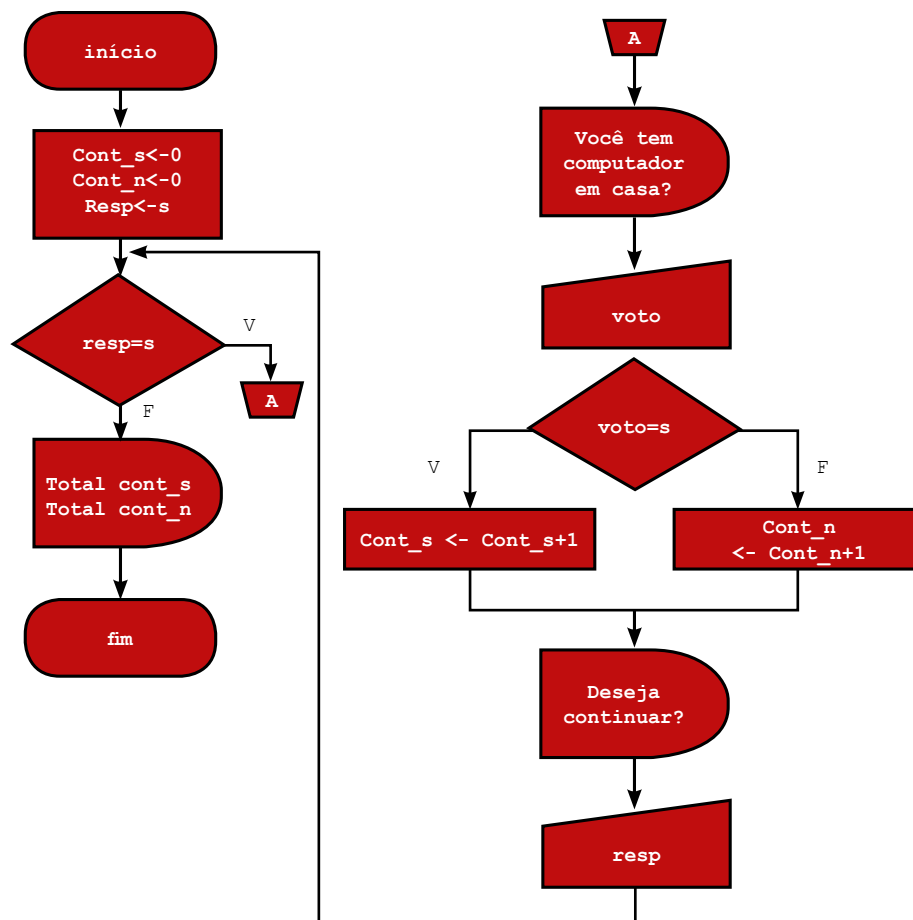


Figura 158 – Programa da pesquisa versão **enquanto**

Vamos fazer o teste de mesa com duas entradas positivas para ter computador em casa e uma entrada negativa, encerrar o processamento e verificar o funcionamento do laço **enquanto**.

algoritmo "enquete"

```

var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
    
```

```
    leia (resp)
  fimenquanto
    escreva("total sim: ", cont_s, "total nao", cont_n)
  fimalgoritmo
```

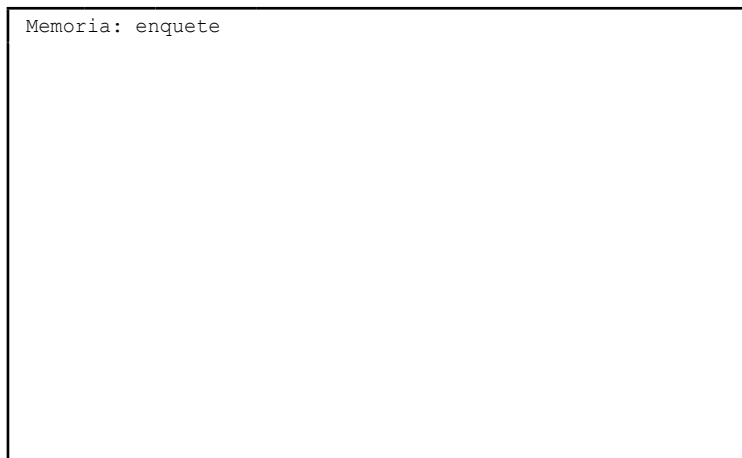


Figura 159 – Início com a memória vazia

```
algoritmo "enquete"
Var
  voto, resp: caractere
  cont_s, cont_n: inteiro
Inicio
  cont_s<-0
  cont_n<-0
  resp<-"s"
  enquanto resp="s" faca
    escreva ("Você tem computador em casa? s/n")
    leia (voto)
    se voto="s" entao
      cont_s<-cont_s+1
    Senão
      cont_n<-cont_n+1
    Fimse
    escreva ("Deseja continuar? s/n")
    leia (resp)
  fimenquanto
  escreva("total sim: ", cont_s, "total nao", cont_n)
fimalgoritmo
```

Memoria: encuesta				
voto	resp	Cont_s	Cont_n	Tela

Figura 160 – O espaço de memória das variáveis é criado

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
    
```

Memoria: encuesta				
voto	resp	Cont_s	Cont_n	Tela
	S	0	0	

Figura 161 – É preciso inicializar a variável resp

Nesse caso, a variável resp necessita ser inicializada, pois é uma condição necessária para a verificação do **enquanto** ser válida.

Memoria: encuesta				
voto	resp	Cont_s	Cont_n	Tela
	S	0	0	

```
algoritmo "enquete"
Var
    voto, resp: caractere
    cont_s, cont_n: inteiro
Inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
enquanto resp="s" faca
    escreva ("Você tem computador em casa? s/n")
    leia (voto)
    se voto="s" entao
        cont_s<-cont_s+1
```


Memoria: encuesta

voto	resp	Cont_s	Cont_n	Tela
s	S	0	0	Você tem computador em casa? s/n s

Figura 164 – Realização da leitura do teclado e atribuição à variável

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
fimalgoritmo
    
```

Memoria: encuesta

voto	resp	Cont_s	Cont_n	Tela
s	S	1	0	Você tem computador em casa? s/n s

Figura 165 – A condicional atualiza a variável acumuladora das respostas sim

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
    escreva ("Deseja continuar? s/n")
    leia (resp)
    fimenquanto
    escreva("total sim: ", cont_s, "total nao", cont_n)
finalgoritmo

```

Memoria: enquete				
voto	resp	Cont_s	Cont_n	Tela
s	S	1	0	Você tem computador em casa? s/n s Deseja continuar? s/n

Figura 166 – Ao chegar ao **senão**, passa para o *fimse*, mostrando a nova saída na tela

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse

```

```

    escreva ("Deseja continuar? s/n")
    leia (resp)
fimenquanto
    escreva("total sim: ", cont_s, "total nao", cont_n)
fimalgoritmo

```

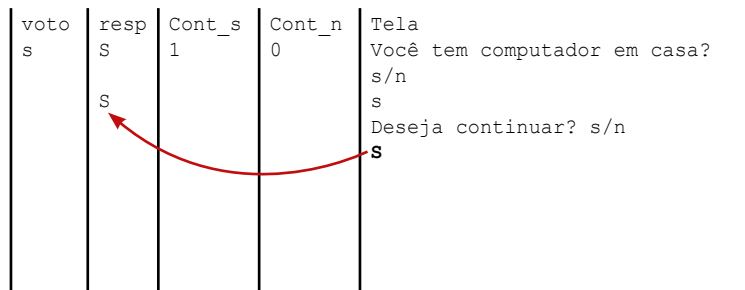


Figura 167 – Feita a leitura do teclado, o valor é armazenado na variável

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
▶ enquanto resp="s" faca
    escreva ("Você tem computador em casa? s/n")
    leia (voto)
    se voto="s" entao
        cont_s<-cont_s+1
    senao
        cont_n<-cont_n+1
    fimse
    escreva ("Deseja continuar? s/n")
    leia (resp)
    fimenquanto
    escreva("total sim: ", cont_s, "total nao", cont_n)
fimalgoritmo

```

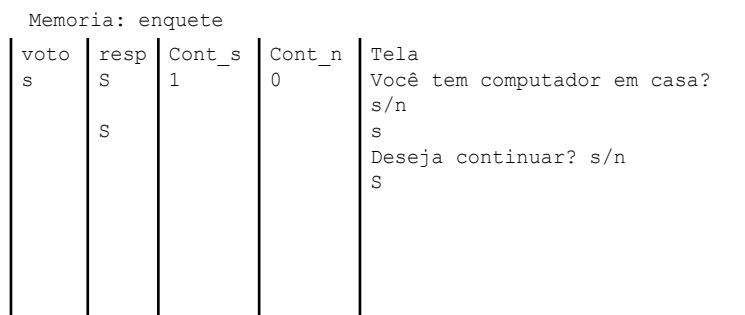


Figura 168 – O *fimenquanto* remete de volta para o início do laço

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
enquanto resp="s" faca
    escreva ("Você tem computador em casa? s/n")
    leia (voto)
    se voto="s" entao
        cont_s<-cont_s+1
    senao
        cont_n<-cont_n+1
    fimse
    escreva ("Deseja continuar? s/n")
    leia (resp)
fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
    
```

Memoria: enquete

voto	resp	Cont_s	Cont_n	Tela
s	S	1	0	Você tem computador em casa?
	S			s/n
				s
				Deseja continuar? s/n
				S

Figura 169 – Como o resultado é Verdadeiro, mais uma iteração é realizada

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
    
```

```

    escreva ("Deseja continuar? s/n")
    leia (resp)
fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
fimalgoritmo

```

Memoria: encuesta

voto	resp	Cont_s	Cont_n	Tela
s	S	1	0	Você tem computador em casa? s/n
	S			s Deseja continuar? s/n S Você tem computador em casa? s/n

Figura 170 – A mensagem é mostrada na tela

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
fimalgoritmo

```

voto	resp	Cont_s	Cont_n	Tela
s	S	1	0	Você tem computador em casa? s/n
n				s Deseja continuar? s/n S Você tem computador em casa? s/n

A red arrow points from the 'n' in the 'voto' column to the 'n' in the 'Tela' column, indicating a loop back.

Figura 171 – A leitura é feita

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo

```

Memoria: enquete				
voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
				s/n
n	s			s
				Deseja continuar? s/n
				s
				Você tem computador em casa?
				s/n
				n

Figura 172 – A condicional remete para a atualização da variável contador do não

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")

```

```

    leia (resp)
    fimenquanto
    escreva("total sim: ", cont_s, "total nao", cont_n)
fimalgoritmo

```

Memoria: encuesta				
voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
				s/n
n	s			s
				Deseja continuar? s/n
				s
				Você tem computador em casa?
				s/n
				n

Figura 173 – O contador é atualizado

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    fimenquanto
    escreva("total sim: ", cont_s, "total nao", cont_n)
fimalgoritmo

```

Memoria: encuesta				
voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
				s/n
n	s			s
				Deseja continuar? s/n
				s
				Você tem computador em casa?
				s/n
				n
				Deseja continuar? s/n

Figura 174 – A mensagem é exibida na tela

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
    
```

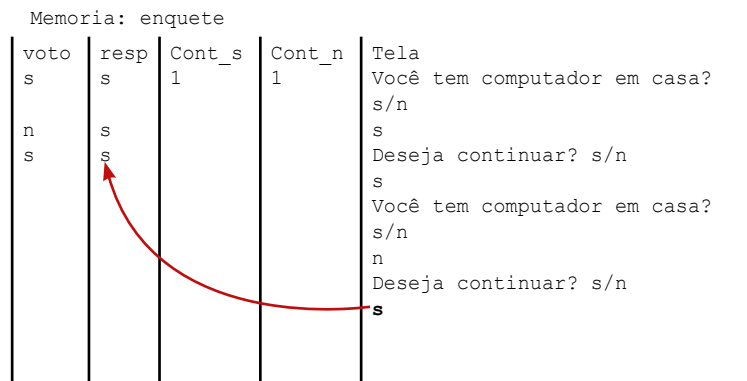


Figura 175 – A leitura é feita


```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo

```

Memoria: enquete				
voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
				s/n
n	s			s
s	s			Deseja continuar? s/n
				s
				Você tem computador em casa?
				s/n
				n
				Deseja continuar? s/n
			w	s

Figura 176 – O programa é remetido para o início do laço

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
    fimse

```

```
fimse
escreva ("Deseja continuar? s/n")
leia (resp)
fimenquanto
escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
```

Memoria: encuesta				
voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa? s/n
n	s			s
	s			Deseja continuar? s/n
				s
				Você tem computador em casa?
				s/n
				n
				Deseja continuar? s/n
			w	s

Figura 177 – Mais um laço será realizado

```
algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
```

Memoria: encuesta

voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
				s/n
n	s			s
	s			Deseja continuar? s/n
				s
				Você tem computador em casa?
				s/n
				n
				Deseja continuar? s/n
			w	s
				Você tem computador em casa?
				s/n

Figura 178 – Mensagem mostrada na tela

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
fimalgoritmo
    
```

Memoria: encuesta

voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
				s/n
n	s			s
	s			Deseja continuar? s/n
				s
				Você tem computador em casa?
				s/n
				n
				Deseja continuar? s/n
			w	s
				Você tem computador em casa?
				s/n
				S

Figura 179 – Realização da leitura

```
algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
```

Memoria: enquete				
voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa? s/n
n	s	2		s Deseja continuar? s/n
s	s			s Você tem computador em casa? s/n
				n Deseja continuar? s/n
			w	s Você tem computador em casa? s/n
				S

Figura 180 – A condicional atualiza o contador

```
algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
    fimse
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
```

```

fimse
escreva ("Deseja continuar? s/n")
leia (resp)
fimenquanto
escreva("total sim: ", cont_s, "total nao", cont_n)
finalgoritmo

```

Memoria: enquete

voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
				s/n
n	s	2		s
s	s			Deseja continuar? s/n
				s
				Você tem computador em casa?
				s/n
				n
				Deseja continuar? s/n
			w	s
				Você tem computador em casa?
				s/n
				S

Figura 181 – A contagem é feita

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
    fimse
    escreva ("Deseja continuar? s/n")
    leia (resp)
fimenquanto
escreva("total sim: ", cont_s, "total nao", cont_n)
finalgoritmo

```

Memoria: encuesta

voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
				s/n
n	s	2		s
s	s			Deseja continuar? s/n
				s
				Você tem computador em casa?
				s/n
				n
				Deseja continuar? s/n
			w	s
				Você tem computador em casa?
				s/n
				S
				Deseja continuar? s/n

Figura 182 – Mensagem para a continuidade

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        senao
            cont_n<-cont_n+1
        fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
fimalgoritmo
    
```

Memoria: enquete

voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa? s/n
n	s	2		s Deseja continuar? s/n
s	s			s Você tem computador em casa? s/n
	n			n Deseja continuar? s/n
			w	s Você tem computador em casa? s/n
				S Deseja continuar? s/n
				n

Figura 183 – É feita a leitura

```

algoritmo "enquete"
var
    voto, resp: caractere
    cont_s, cont_n: inteiro
inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
enquanto resp="s" faca
    escreva ("Você tem computador em casa? s/n")
    leia (voto)
    se voto="s" entao
        cont_s<-cont_s+1
    senao
        cont_n<-cont_n+1
    fimse
    escreva ("Deseja continuar? s/n")
    leia (resp)
fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
    
```

Memoria: encuesta

voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa?
				s/n
n	s	2		s
s	s			Deseja continuar? s/n
	n			s
				Você tem computador em casa?
				s/n
				n
				Deseja continuar? s/n
			w	s
				Você tem computador em casa?
				s/n
				S
				Deseja continuar? s/n
				n

Figura 184 - O fluxo é devolvido ao início do laço

```

algoritmo "enquete"
Var
    voto, resp: caractere
    cont_s, cont_n: inteiro
Inicio
    cont_s<-0
    cont_n<-0
    resp<-"s"
    enquanto resp="s" faca
        escreva ("Você tem computador em casa? s/n")
        leia (voto)
        se voto="s" entao
            cont_s<-cont_s+1
        Senão
            cont_n<-cont_n+1
        Fimse
        escreva ("Deseja continuar? s/n")
        leia (resp)
    fimenquanto
    escreva("total sim: ",cont_s, "total nao", cont_n)
finalgoritmo
    
```


Memoria: enquete

voto	resp	Cont_s	Cont_n	Tela
s	s	1	1	Você tem computador em casa? s/n
n	s	2		s Deseja continuar? s/n
s	s			s Você tem computador em casa? s/n
	n			n Deseja continuar? s/n
			w	s Você tem computador em casa? s/n
				S Deseja continuar? s/n
				n total sim: 2 total não: 1"

Figura 185 – Como a condição não é satisfeita, o laço é encerrado

No exemplo que veremos a seguir, como não sabemos de antemão quantas pessoas irão responder à enquete, utilizamos o comando *enquanto*, que somente interromperá as iterações quando o valor de resp for 's'.

Exemplo de aplicação

Faça um programa que calcule o fatorial de um número inteiro.

Dada a definição de fatorial:

"O fatorial é uma propriedade matemática [...] na determinação do produto dos antecessores de um número maior que 1, ou seja, fatorial de um número n (designado por $n!$) é igual a $n! = n(n - 1)(n - 2)(n - 3)...$ " (ANÁLISE..., 2014).

Assim, o fatorial de 5, por exemplo, é 120, pois:

$$5! = 5(5-1)(5-2)(5-3)(5-4)$$

$$5! = 5.4.3.2.1$$

$$5! = 120$$

Pela propriedade comutativa da multiplicação, a conhecida "A ordem dos fatores não altera o produto.", podemos escrever que:

$$5! = 1.2.3.4.5$$

$$5! = 120$$

Pela definição, também se aplica que $0! = 1$.

Criar uma variável chamada *fat* inicializada por 1 e acumular nela o produto da multiplicação por um número sequencial.

Assim:

- Para 1!

Início	fat
fat<-1	1
fat<-fat*1	1
fim	

Figura 186 – Cálculo de 1!

- Para 2!

Início	fat
fat<-1	1
fat<-fat*1	1
fat<-fat*2	2
fim	

Figura 187 – Cálculo para 2!

- Para 3!

Início	fat
fat<-1	1
fat<-fat*1	1
fat<-fat*2	2
fat<-fat*3	6
fim	

Figura 188 – Cálculo de 3!

- Para 4!

Início	fat
fat<-1	1
fat<-fat*1	1
fat<-fat*2	2
fat<-fat*3	6
fat<-fat*4	24
fim	

Figura 189 – Cálculo para 4!

- Para 5!

inicio	fat
fat<-1	1
fat<-fat*1	1
fat<-fat*2	2
fat<-fat*3	6
fat<-fat*4	24
fat<-fat*5	120
fim	

Figura 190 – Cálculo de 5!

Vamos então introduzir uma nova variável inteira i e utilizar o exemplo do 5!

inicio	fat	i
fat<-1	1	1
i<-1		
fat<-fat*i	1	1
i<-i+1		
fat<-fat*i	2	2
i<-i+1		
fat<-fat*i	6	3
i<-i+1		
fat<-fat*i	24	4
i<-i+1		
fat<-fat*i	120	5
fim		

Figura 191 – Uso do exemplo 5! incluindo uma nova variável

Nesse último programa, notamos que existem as linhas:

```
fat<-fat*i
i<-i+1
```

Estas se repetem algumas vezes; como os laços são estruturas de repetição, podemos escrever (no caso, usando o laço **enquanto**):

```
inicio
fat<-1
i<-1
enquanto i<=5 faca
fat<-fat*i
i<-i+1
fimenquanto
fimalgoritmo
```

Como a proposta do exercício é para qualquer número inteiro, e não apenas para 5, vamos colocar uma nova variável, em que será feita a leitura do número desejado pelo operador do programa, e também uma saída. Assim, o programa final será:

```
algoritmo "fatorial"
var
    fat, i, n: inteiro
inicio
    leia(n)
    fat<-1
    i<-1
    enquanto i<=n faça
        fat<-fat*i
        i<-i+1
    fimenquanto
    escreva(fat)
fimalgoritmo
```

Faça o teste de mesa para conferir a funcionalidade do código.



Observação

No Scratch, o laço **repita até que** e o **enquanto** são comandados pelo bloco:



Figura 192 – Bloco que comanda o laço **repita até que** e o **enquanto**

4.1.2 Laços contados

Os laços contados serão utilizados quando existir o conhecimento prévio do número de iterações a ser executado. Essa estrutura é conhecida como o laço do **para-até**. O laço **para** utiliza uma variável que é contador, e a cada iteração se atualiza esse valor, verificando se chegou ao seu limite.

Forma:

```
para <varcontrole> de <vl inicial> até <vl final> passo <passo> faça
    ...
fimpara
```

O fluxograma da estrutura **para** é:

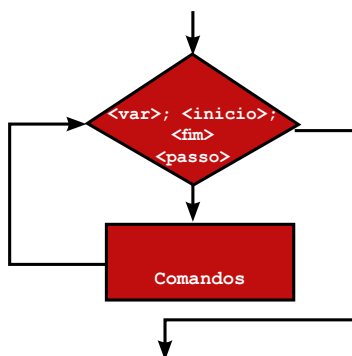


Figura 193 – Estrutura **para...até**

No caso de passo 1, pode-se omitir o comando *passo*.

Exemplo de aplicação

Para entender, passo a passo, um programa simples:

```
algoritmo "exemplo"
var
    i:inteiro
início
    para i de 1 até 7 passo 2 faça
        escreva(i)
    fimpara
fimalgoritmo
```

Fazendo o teste de mesa:

```
algoritmo "exemplo"
var
    i:inteiro
início
    para i de 1 até 7 passo 2 faça
        escreva(i)
    fimpara
fimalgoritmo
```

i	tela
---	------

Figura 194

```
algoritmo "exemplo"  
var  
    i:inteiro  
inicio  
    para i de 1 até 7 passo 2 faca  
        escreva(i)  
    fimpara  
fimalgoritmo
```



Figura 195 – Atribuição do valor inicial à variável de controle

Ao encontrar o comando para i de 1, o programa atribui o valor inicial à variável de controle, no caso, 1. Feito isso, o comando para i de 1 não será mais executado.

```
algoritmo "exemplo"  
var  
    i:inteiro  
inicio  
    para i de 1 até 7 passo 2 faca  
        escreva(i)  
    fimpara  
fimalgoritmo
```

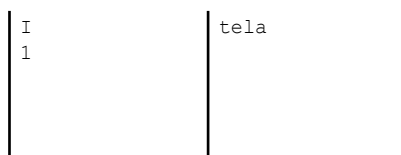


Figura 196– Teste da variável de controle

Feita a atribuição, o programa testa o conteúdo da variável de controle com o valor final <varcontrole> <= <vl final>, no caso, $i \leq 7$.

Como é verdadeiro, o programa entra no laço. Caso fosse falso, prosseguiria após o *fimpara*.

```
algoritmo "exemplo"  
var  
    i:inteiro  
inicio  
    para i de 1 ate 7 passo 2 faca  
        escreva(i)  
    fimpara  
fimalgoritmo
```



Figura 197- Execução dos comandos dentro do laço

```
algoritmo "exemplo"
var
    i:inteiro
inicio
    para i de 1 ate 7 passo 2 faca
        escreva(i)
    fimpara
finalgoritmo
```



Figura 198-Execução automática de uma nova instrução

Ao final da iteração, encontra o comando *fimpara*. Nesse momento, o programa faz, automaticamente, a seguinte instrução:

$\text{<varcontrole>} \leftarrow \text{<varcontrole>} + \text{passo}$

No caso:

$i \leftarrow i + 2$

```
algoritmo "exemplo"
var
    i:inteiro
inicio
    para i de 1 ate 7 passo 2 faca
        escreva(i)
    fimpara
finalgoritmo
```



Figura 199 – Retorno do programa à linha do **para**

```
algoritmo "exemplo"
var
    i:inteiro
inicio
```

```
para i de 1 ate 7 passo 2 faca
  escreva(i)
fimpara
finalgoritmo
```

i	tela
1	1
3	

Figura 200 – Realização de teste pelo programa

O programa testa se chegou ao final das iterações, fazendo $\langle \text{varcontrole} \rangle \leq \langle \text{vl final} \rangle$, no caso: $i \leq 7$. Como aqui o i contém 3, a expressão é verdadeira, e o programa volta para dentro do laço.

```
algoritmo "exemplo"
var
  i:inteiro
inicio
  para i de 1 ate 7 passo 2 faca
    escreva(i)
  fimpara
finalgoritmo
```

i	tela
1	1
3	3

Figura 201 – O programa executa os comandos dentro do laço

```
algoritmo "exemplo"
var
  i:inteiro
inicio
  para i de 1 ate 7 passo 2 faca
    escreva(i)
  fimpara
finalgoritmo
```

i	tela
1	1
3	3
5	

Figura 202 – Uma nova instrução é executada automaticamente pelo programa

Ao encontrar o comando *fimpara*, o programa faz, automaticamente, $\langle \text{varcontrole} \rangle \leftarrow \langle \text{varcontrole} \rangle + \text{passo}$. No caso, $i \leftarrow i + 2$.


```
algoritmo "exemplo"  
var  
    i:inteiro  
inicio  
    para i de 1 ate 7 passo 2 faca  
        escreva(i)  
    fimpara  
fimalgoritmo
```

i	tela
1	1
3	3
5	

Figura 203 – Retorno do programa para a linha do **para** e execução do teste

O programa retorna para a linha do **para** e testa se chegou ao final das iterações, $i \leq 7$. Como nesse caso o i contém 5, a expressão é verdadeira, e o programa volta para dentro do laço.

```
algoritmo "exemplo"  
var  
    i:inteiro  
inicio  
    para i de 1 ate 7 passo 2 faca  
        escreva(i)  
    fimpara  
fimalgoritmo
```

i	tela
1	1
3	3
5	5

Figura 204– Execução dos comandos dentro do laço

```
algoritmo "exemplo"  
var  
    i:inteiro  
inicio  
    para i de 1 ate 7 passo 2 faca  
        escreva(i)  
    fimpara  
fimalgoritmo
```

i	tela
1	1
3	3
5	5
7	

Figura 205– Nova execução automática de instrução pelo programa

No *fimpara*, o programa faz, automaticamente, $\langle \text{varcontrole} \rangle \leftarrow \langle \text{varcontrole} \rangle + \text{passo}$. No caso, $i \leftarrow i + 2$.

```

algoritmo "exemplo"
var
    i:inteiro
inicio
    para i de 1 ate 7 passo 2 faca
        escreva(i)
    fimpara
finalgoritmo

```

i	tela
1	1
3	3
5	5
7	

Figura 206– Volta do programa para a linha do **para** e realização do teste

O programa retorna para a linha do **para** e faz o teste se chegou ao final das iterações, $i \leq 7$. Como nesse caso o i contém 7, **devemos prestar atenção**, pois a expressão é verdadeira, já que a condição é menor ou igual; sete é igual a sete, e o programa volta para dentro do laço.

```

algoritmo "exemplo"
var
    i:inteiro
inicio
    para i de 1 ate 7 passo 2 faca
        escreva(i)
    fimpara
finalgoritmo

```

i	tela
1	1
3	3
5	5
7	7

Figura 207– O programa executa os comandos dentro do laço

```

algoritmo "exemplo"
var
    i:inteiro
inicio
    para i de 1 ate 7 passo 2 faca
        escreva(i)
    fimpara
finalgoritmo

```

I	tela
1	1
3	3
5	5
7	7
9	

Figura 208– Execução automática de uma instrução pelo programa

No *fimpara*, o programa faz, automaticamente, $\langle \text{varcontrole} \rangle \leftarrow \langle \text{varcontrole} \rangle + \text{passo}$. No caso, $i \leftarrow i + 2$, e o i passa a valer 9, voltando para a linha do **para**.

```

algoritmo "exemplo"
var
    i:inteiro
inicio
    para i de 1 ate 7 passo 2 faca
        escreva(i)
    fimpara
finalgoritmo
    
```

i	tela
1	1
3	3
5	5
7	7
9	

Figura 209– O programa desconsidera o laço e prossegue

Quanto ao teste para verificar se chegou ao final das iterações, $i \leq 7$, como no caso o i contém 9, **resulta em Falso**; assim, o programa pula o laço, passando a ser executado após a linha do *fimpara*.

Assim, vimos um exemplo do uso do passo no laço **para**. Abordaremos agora um exemplo mais prático, visto no tópico anterior desta unidade.

Observação

No Scratch, o laço **para... até que** é comandado pelo bloco:



Figura 210 – Laço **para... até que** no Scratch

No exemplo, o som do miado será repetido dez vezes.

Exemplo de aplicação

Faça um programa que calcule o fatorial de um número inteiro usando o laço **para**.

No caso do fatorial, pode não parecer, mas é um típico caso de laço contado, pois é claro quantas iterações vão acontecer. Esse número é determinado na leitura inicial, portanto não depende de o operador interromper ou não a execução do laço. Assim, o pseudocódigo será:

```
algoritmo "fatorial"  
var  
    fat, i, n:inteiro  
inicio  
    leia(n)  
    fat<-1  
    para i de 1 ate n faca  
        fat<-fat*i  
    fimpara  
    escreva(fat)  
fimalgoritmo
```

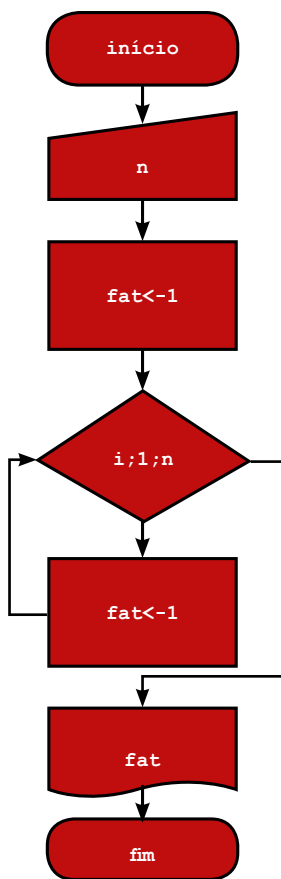


Figura 211 – Cálculo fatorial

Vamos fazer o teste de mesa para 4!

```

algoritmo "fatorial"
var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    fimpara
    escreva(fat)
fimalgoritmo

```

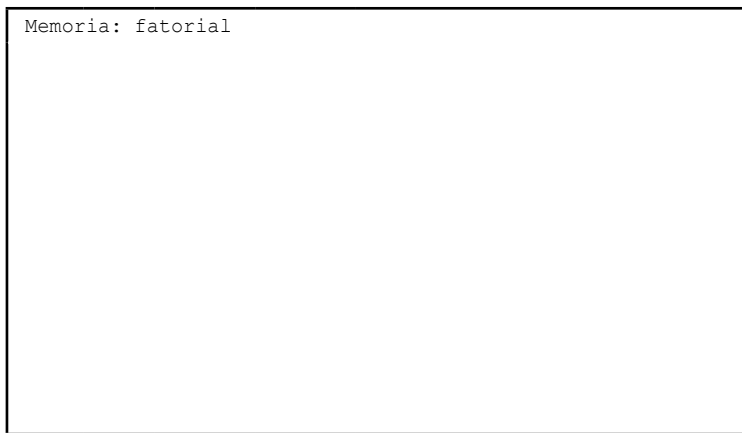


Figura 212- Reserva de um espaço de memória para o programa fatorial

```

algoritmo "fatorial"
var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    fimpara
    escreva(fat)
fimalgoritmo

```



Figura 213– São reservados espaços na memória para as variáveis

```

algoritmo "fatorial"
var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    fimpara
    escreva(fat)
fimalgoritmo
    
```

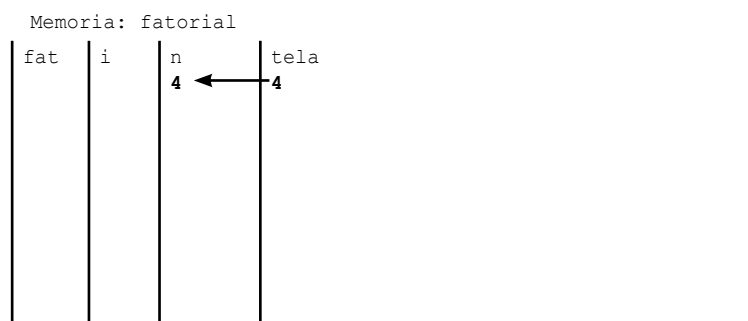


Figura 214– Leitura do valor de entrada

A leitura do valor de entrada é feita. Como a proposta é fazer o fatorial de 4, o valor será atribuído à variável *n*.

```

algoritmo "fatorial"
var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    fimpara
    escreva(fat)
fimalgoritmo
    
```



Figura 215– Inicialização de *fat*


```

algoritmo "fatorial"
var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    fimpara
    escreva(fat)
finalgoritmo

```

Como $i \leq n$ ($2 \leq 4$), o programa entra novamente no laço.

```
algoritmo "fatorial"
var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    fimpara
    escreva(fat)
fimalgoritmo
```

Memoria: fatorial

fat	i	n	tela
1	1	4	4
1	2		
2			

Figura 221– O conteúdo do laço é executado

```
algoritmo "fatorial"
var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    fimpara
    escreva(fat)
fimalgoritmo
```

Memoria: fatorial

fat	i	n	tela
1	1	4	4
1	2		
2	3		

Figura 222– Atualização da variável de controle

Ao chegar ao *fimpara*, o valor da variável de controle é atualizado.

```

algoritmo "fatorial"
var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    fimpara
    escreva(fat)
fimalgoritmo
    
```

Memoria: fatorial

fat	i	n	tela
1	1	4	4
1	2		
2	3		

Figura 223- O programa volta a entrar no laço

Como $i \leq n$ ($3 \leq 4$), o programa entra novamente no laço.

```

algoritmo "fatorial"
var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    fimpara
    escreva(fat)
fimalgoritmo
    
```

Memoria: fatorial

fat	i	n	tela
1	1	4	4
1	2		
2	3		
6			

Figura 224– O programa executa o conteúdo do laço

```

algoritmo "fatorial"
var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    fimpara
    escreva(fat)
fimalgoritmo
    
```

Memoria: fatorial

fat	i	n	tela
1	1	4	4
1	2		
2	3		
6	4		

Figura 225– Atualização do valor da variável de controle

Ao chegar ao *fimpara*, o valor da variável de controle é atualizado.

```

algoritmo "fatorial"
Var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    Fimpara
    escreva(fat)
fimalgoritmo
    
```

v

Memoria: fatorial

fat	i	n	tela
1	1	4	4
1	2		
2	3		
6	4		

Figura 226- O programa entra no laço mais uma vez

Como $i \leq n$ ($4=4$), o programa entra novamente no laço.

```

algoritmo "fatorial"
var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    fimpara
    escreva(fat)
fimalgoritmo
    
```

Memoria: fatorial

fat	i	n	tela
1	1	4	4
1	2		
2	3		
6	4		
24			

Figura 227- O programa executa o conteúdo do laço

```

algoritmo "fatorial"
var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    fimpara
    escreva(fat)
fimalgoritmo
    
```

Memoria: fatorial

fat	i	n	tela
1	1	4	4
1	2		
2	3		
6	4		
24	5		

Figura 228– O programa atualiza o valor da variável de controle

Ao chegar ao *fimpara*, o valor da variável de controle é atualizado.

```

algoritmo "fatorial"
var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    fimpara
    escreva(fat)
fimalgoritmo
    
```

Memoria: fatorial

fat	i	n	tela
1	1	4	4
1	2		
2	3		
6	4		
24	5		

Figura 229– Como $i \leq n$ ($5 > 4$) é Falso, sai do laço

```

algoritmo "fatorial"
var
    fat, i, n:inteiro
inicio
    leia(n)
    fat<-1
    para i de 1 ate n faca
        fat<-fat*i
    fimpara
    escreva(fat)
fimalgoritmo
    
```

fat	i	n	tela
1	1	4	4
1	2		24
2	3		
6	4		
24	5		

Figura 230– Exibição do resultado e encerramento do programa

Ao sair do laço, o programa mostra o resultado e é encerrado.

Como fizemos o cálculo do fatorial utilizando duas técnicas diferentes, podemos comparar os programas para compreendermos como funciona a automação do **para**.

<pre> algoritmo "fatorial" var fat, i, n:inteiro inicio leia(n) fat<-1 i<-1 enquanto i<=n faca fat<-fat*i i<-i+1 fimenquanto escreva(fat) fimalgoritmo </pre>	<pre> algoritmo "fatorial" var fat, i, n:inteiro inicio leia(n) fat<-1 para i de 1 ate n faca fat<-fat*i fimpara escreva(fat) fimalgoritmo </pre>
--	--

Assim, em verde, temos a inicialização da variável de controle; em azul, o teste do limite; e em marrom, o incremento e a devolução do fluxo.

4.2 Laboratório

Na linguagem C, temos os três tipos de estruturas de blocos, porém com algumas diferenças.

4.2.1 Repita até que

Como vimos no pseudocódigo no laço **repita**, o fluxo do programa encontra o comando propriamente dito, reconhece que é o início do bloco de repetição e segue o fluxo normal até encontrar o comando em que uma condição é testada. Na linguagem C, o comando que inicia o **repita** é o do segundo bloco demarcado pelas chaves.



Observação

A diferença entre o pseudocódigo e a linguagem C é que nesta, ao final, o teste é feito com o comando *while* (condição), executando novamente o laço se o resultado for Verdadeiro. É o contrário do pseudocódigo, em que, caso o resultado do teste seja Falso, o fluxo é devolvido ao início do bloco.

Sintaxe:

```
do {
    Bloco
}while (condição);
```

Assim, a tradução do programa da enquete usando **repita** será:

<pre> algoritmo "enquete" var voto, resp: caractere cont_s, cont_n: inteiro inicio cont_s<-0 cont_n<-0 repita escreva ("Você tem computador em casa? s/n") leia (voto) se voto="s" entao cont_s<-cont_s+1 senao cont_n<-cont_n+1 fimse escreva ("Deseja continuar? s/n") leia (resp) ate resp="n" escreva("total sim: ",cont_s, "to- tal nao", cont_n) finalgoritmo </pre>	<pre> #include <stdio.h> void main () { char voto,resp; int cont_s,cont_n; cont_s=0; cont_n=0; do{ printf("\nVocê tem computador em casa? s/n "); scanf ("%c", &voto); fflush(stdin); if (voto == 's') cont_s++; else cont_n++; printf("\nDeseja continuar? s/n "); scanf ("%c", &resp); fflush(stdin); }while (resp=='s'); printf("\ntotal sim: %d total nao: %d",cont_s, cont_n); } </pre>
--	--



Observação

O comando `fflush` serve apenas para limpar a área de leitura, pois, em alguns casos, quando ocorre a leitura de um caractere, podem acontecer erros. Existem outros comandos de leitura mais adequados, mas que ainda não fazem parte do conteúdo da matéria.

Ao executarmos, obtemos a seguinte tela:

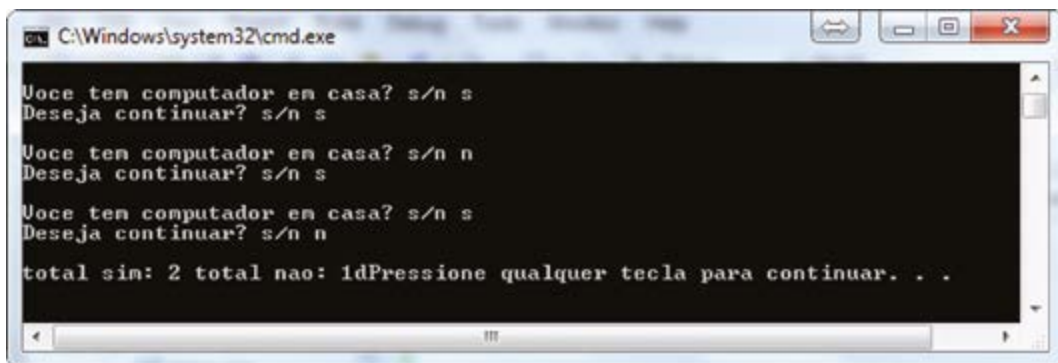


Figura 231 – Saída do programa da enquete usando *do...while*

4.2.2 Enquanto faça

Estruturalmente, o laço **enquanto faça** não difere do pseudocódigo.

```
while (expr) {  
    bloco de comandos  
}
```

O laço realizará as iterações do bloco de comandos enquanto o resultado de *expressao* for verdadeiro. Quando *expressao* se tornar falsa, o bloco de comando parará de ser executado, e o programa prosseguirá no comando seguinte ao bloco.

A tradução do programa "enquete" usando o laço **enquanto** será:

<pre> algoritmo "enquete" var voto, resp: caractere cont_s, cont_n: inteiro inicio cont_s<-0 cont_n<-0 resp<-"s" enquanto resp=="s" faca escreva ("Você tem computador em casa? s/n") leia (voto) se voto=="s" entao cont_s<-cont_s+1 senao cont_n<-cont_n+1 fimse escreva ("Deseja continuar? s/n") leia (resp) fimenquanto escreva("total sim: ",cont_s, "total nao", cont_n) fimalgoritmo </pre>	<pre> #include <stdio.h> void main () { char voto,resp; int cont_s,cont_n; cont_s=0; cont_n=0; resp='s'; while (resp=='s'){ printf("\nVocê tem computador em casa? s/n "); scanf ("%c", &voto); fflush(stdin); if (voto == 's') cont_s++; else cont_n++; printf("\Deseja continuar? s/n "); scanf ("%c", &resp); fflush(stdin); }; printf("\ntotal sim: %d total nao: %d",cont_s, cont_n); } </pre>
---	---

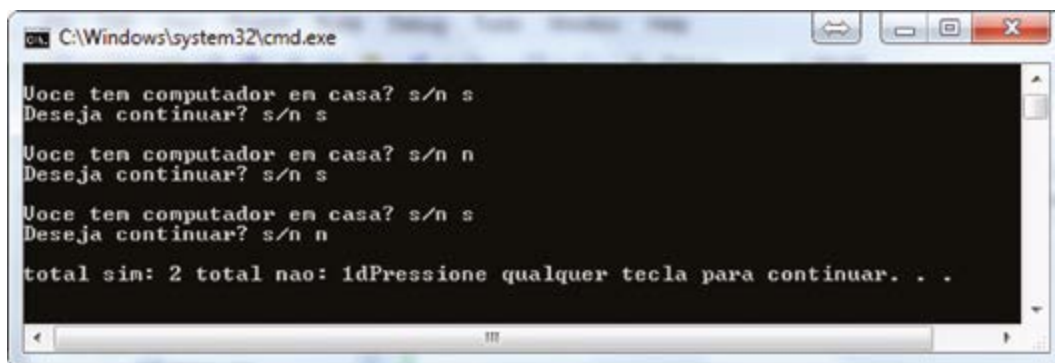


Figura 232 – Saída do programa usando *while*

Fazendo a tradução do programa do fatorial utilizando o **enquanto**.

<pre>algoritmo "fatorial" var fat,i,n:inteiro inicio leia(n) fat<-1 i<-1 enquanto i<=n faca fat<-fat*i i<-i+1 fimenquanto escreva(fat) fimalgoritmo</pre>	<pre>#include <stdio.h> void main () { ..int fat,i,n; ..scanf("%d",&n); ..fat=1; ..i=1; ..while (i<=n){ fat*=i; ..i++; } ..printf("%d\n",fat); }</pre>
--	--

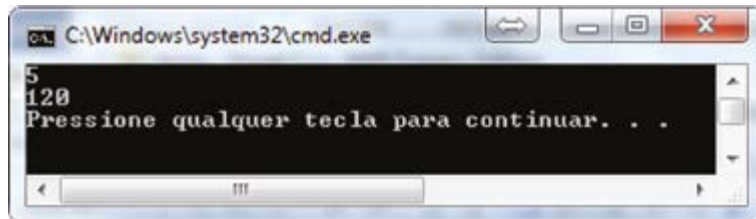


Figura 233 – Saída do programa fatorial usando *while* (enquanto)



Lembrete

Após a instrução do *while* (expressão), não se deve colocar o sinal de fim de comando (;) Um dos erros mais comuns em programação é colocar ponto e vírgula antes do bloco de comandos.

4.2.3 Para até que

Na linguagem C, a estrutura do funcionamento é semelhante ao **para** do pseudocódigo, mas indicada de forma explícita.

Sintaxe:

```
for (expr inicial; expr booleana; expr de incremento)
{
    bloco de comandos
}
```

A *expr inicial* inicializa a variável de controle. Caso a *expr booleana* seja verdadeira, o laço será executado; *expr de incremento* é o cálculo feito quando o fluxo do programa chega ao *fimpara*.

Para exemplificar e deixar claro o seu funcionamento, serão apresentados alguns casos de tradução.

- Caso 1

Pseudocódigo:

para i de 1 até 15 passo 2 faça

Linguagem C:

```
for( i = 1; i<=15; i=i+2)
```

ou

```
for( i = 1; i<=15; i+=2)
```

- Caso 2

Pseudocódigo:

para i de 3 até 15 faça

Linguagem C:

```
for( i = 3; i<=15; i++)
```

ou

```
for( i = 3; i<=15; i=i+1)
```

- Caso 3

Pseudocódigo:

para j de 20 até 10 passo -1 faça

Linguagem C:

```
for( j = 20; j>=10; i--)
```

ou

```
for( j = 20; j>=10; i=i-1)
```

A tradução do programa utilizado na explicação fica:

<pre>algoritmo "exemplo" var i:inteiro inicio para i de 1 ate 7 passo 2 faca escreva(i) fimpara finalgoritmo</pre>	<pre>#include <stdio.h> void main () { ..int i; ..for(i=1; i<=7; i+=2){ ..printf("%d\n",i); } }</pre>
--	---

A saída do programa fica:

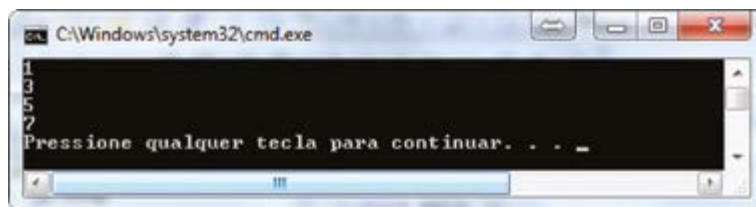


Figura 234 – Saída do programa usado como exemplo



Lembrete

Vimos anteriormente que `i++` é equivalente a `i=i+1`, e `i+=2` corresponde a `i=i+2`

A tradução do programa do fatorial utilizando **para** fica:

<pre>algoritmo "fatorial" var fat, i, n:inteiro inicio leia(n) fat<-1 para i de 1 ate n faca fat<-fat*i fimpara escreva(fat) finalgoritmo</pre>	<pre>#include <stdio.h> void main () { int fat,i,n; scanf("%d",&n); fat=1; for(i=1;i<=n;i++){ fat*=i; } printf("%d\n",fat); }</pre>
---	---

A saída desse exemplo é:

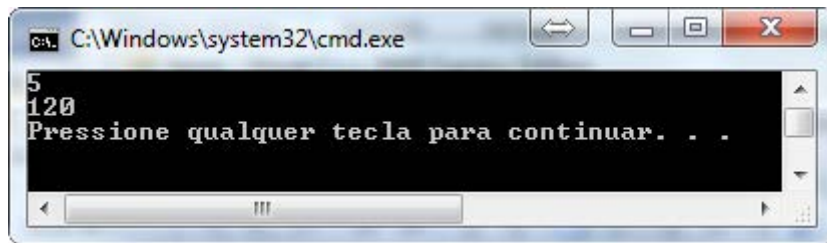


Figura 235 – Saída do programa fatorial



Resumo

Os laços de repetição foram o assunto da Unidade IV. Vimos inicialmente os laços condicionais, aqueles em que um bloco é repetido inúmeras vezes, porém não sabemos quando será interrompido. Nessa categoria se encaixam os laços **repita até que** e **enquanto faça**. A grande diferença entre ambos é que o teste da condição do primeiro acontece no final do bloco; assim, a primeira iteração (ou ciclo) é sempre realizada. No segundo, como a condição é testada no início, ao entrar no laço, se a condição não for satisfeita, a iteração sequer será realizada, e assim sucessivamente, sempre com o teste feito no início do bloco.

Depois, vimos o laço contado, aquele que é utilizado quando sabemos com antecedência quantas iterações serão necessárias. Esse laço se chama **para até** e, usando uma variável de controle, automaticamente, controla o incremento das execuções do bloco e testa a condição de término das iterações.

Compreendidos os conceitos de laços e vistos os comandos, passamos à tradução para a linguagem C. A estrutura do **repita até que**, na linguagem C, muda um pouco na verificação da condição de fim de laço. Em algoritmo, o laço acontece até que um certo evento ocorra, e na linguagem C o laço acontece enquanto um evento é executado. Em C, esse comando é: `do { } while (condição)`.

A estrutura do **enquanto** funciona exatamente da mesma forma, tanto no algoritmo quanto na linguagem C. O **enquanto** (condição *fimenquanto*) é traduzido em `while (condição) { };`.

O laço contado em C é escrito com três parâmetros: o primeiro inicializa a variável de controle, o segundo é a expressão lógica que determinará o encerramento do laço e o terceiro é a expressão do incremento a ser realizado. Assim, o comando que faz o laço contado fica: `for (<início de controle>; <condição de encerramento>; <incremento>)`.



Exercícios

Questão 1. Uma das funções do computador é realizar operações repetitivas, o que para o ser humano seria muito enfadonho. É muito comum durante o processamento a existência de situações em que há a necessidade de se repetir um determinado trecho de um programa certo número de vezes. Com base nessas informações, escolha a alternativa com a informação considerada correta:

- A) A estrutura dos laços de repetição é composta por três comandos: o enquanto, o repita e o para. Entretanto, o problema a ser resolvido não depende do laço de repetição, e cada um deles apresenta uma característica mais adequada ou não para a sua solução.
- B) As estruturas de repetição podem ser classificadas em laços contados e laços condicionais. Ambas possuem um funcionamento mecanizado, muito particular em cada uma das formas.
- C) Laços condicionais ocorrem quando existem blocos de programa que necessitam ser repetidos; porém, não se sabe quantas vezes ele será repetido.
- D) No laço repita, o fluxo do programa encontra o comando propriamente dito e reconhece que é o início do bloco de repetição; segue o fluxo normal até encontrar o comando até que, pelo qual uma condição é testada. Se o resultado do teste for falso, o fluxo não é devolvido ao início do bloco. Se o resultado do teste for positivo, o programa segue o fluxo abandonando a repetição do bloco.
- E) O laço enquanto testa uma condição no final do programa, e se o resultado do teste for falso, então o bloco será ignorado prosseguindo normalmente após o fim enquanto. Caso o resultado do teste seja positivo, o bloco será executado e ao encontrar a instrução fim enquanto o fluxo será devolvido à instrução enquanto e a condição testada novamente, até que a condição se torne falsa.

Resposta correta: alternativa C.

Análise das alternativas

A) Alternativa incorreta.

Justificativa: a estrutura dos laços de repetição é composta por três comandos: o enquanto, o repita e o para. Entretanto, dependendo do problema a ser resolvido, cada um deles apresenta uma característica mais adequada.

B) Alternativa incorreta.

Justificativa: as estruturas de repetição podem ser classificadas em laços contados e laços condicionais. Ambos possuem funcionamento automatizado, muito particular em cada uma das formas.

C) Alternativa correta.

Justificativa: os laços condicionais ocorrem no caso de existirem blocos de programa que necessitem ser repetidos; porém não se sabe quantas vezes será a repetição.

D) Alternativa incorreta.

Justificativa: no laço *repita*, o fluxo do programa encontra o comando propriamente dito e reconhece que é o início do bloco de repetição; segue o fluxo normal até encontrar o comando *até que*, onde uma condição é testada. Se o resultado do teste for falso, o fluxo é devolvido ao início do bloco. Se o resultado do teste for positivo, o programa segue o fluxo, abandonando a repetição do bloco

E) Alternativa incorreta.

Justificativa: o laço *enquanto* testa uma condição no início do programa e se o resultado do teste for falso, então o bloco será ignorado prosseguindo normalmente após o *fimenquanto*. Caso o resultado do teste seja positivo, o bloco será executado e, ao encontrar a instrução *fimenquanto*, o fluxo será devolvido à instrução *enquanto* e a condição testada novamente, até que a condição se torne falsa.

Questão 2. Na linguagem C existem três tipos de estruturas de blocos, porém com algumas diferenças: *repita até que*, *enquanto faça* e *para até que*. Com base nessas informações, escolha a alternativa com a informação correta:

I – Na linguagem C, a estrutura do funcionamento é semelhante à do *para* do pseudocódigo, mas não é indicada de forma explícita.

II – A estrutura do *repita até que* na linguagem C altera-se um pouco na verificação da condição de fim de laço.

III – Em algoritmo, o laço acontece até que um certo evento ocorra, e na linguagem C o laço acontece enquanto um evento é executado.

IV – O laço contado em C é escrito com três parâmetros: o primeiro inicializa a variável de controle, o segundo parâmetro é a expressão lógica que determinará o encerramento do laço e o terceiro é a expressão do decremento a ser realizado.

É correto apenas o que se afirma em:

A) I.

B) IV.

C) I e III.

D) II e III.

E) II e IV.

Resolução desta questão na plataforma.