

# AED 1 - Simulação Fila de Banco

Bruno Tomé<sup>1</sup>, Cláudio Menezes<sup>1</sup>

<sup>1</sup>Instituto Federal de Minas Gerais (IFMG)  
São Luiz Gonzaga, s/nº - Formiga / MG - Brasil

ibrunotome@gmail.com, claudiomenezio@gmail.com

**Abstract.** *This report is about a simulator queue of a bank.*

**Resumo.** *Relatório sobre o trabalho 2 de AED 1, simulando uma fila de banco, fila única ou várias.*

## 1. Introdução

A proposta de realização deste trabalho é a simulação de uma fila de banco, usando a estrutura de dados fila, para representar a situação fila única ou várias filas.

## 2. Implementação

### 2.1. Descrição sobre as decisões de projeto e implementação do programa

A implementação foi feita em dupla, nenhum dos integrantes ficou responsável por uma parte específica, ambos usaram a combinação collabedit.com (ferramenta de codificação online, com opção de múltiplos editores em tempo real na nuvem) e Skype para codificarmos o algoritmo.

Também achamos conveniente setar a variável qtdCliente e tempoChegada como randômicas. Para a qtdCliente achamos plausível o máximo de 20 e tempoChegada 10.

### 2.2. Descrição das estruturas de dados usadas no programa

Utilizamos a biblioteca de fila oferecida pelo professor, e implementamos as funções código principal. Com o auxílio da biblioteca já implementada, a inserção e remoção na fila foram feitas facilmente. Acrescentamos ao TAD libfila no record fila a variável tamanho tipo integer, que guarda o tamanho da fila. Pois quando criamos uma fila o tamanho é nulo, quando inserimos o tamanho é aumentado em 1 e quando removemos é diminuído em uma unidade.

### 2.3. Funcionamento das principais funções e procedimentos utilizados

Comentado no código.

### 2.4. Formato de entrada e saída dos dados

O usuário não precisa entrar com dado algum, pois eles são gerados aleatoriamente. A saída de dados é impressa no console.

### 2.5. Como executar o programa

No terminal com o arquivo do código fonte no Desktop digite o seguinte código:

```
cd Desktop
```

```
fpc simulacaoBanco.pas -osimulacaoBanco.bin
```

```
./simulacaoBanco.bin
```

### **3. Análise das medidas realizadas**

Linguagem utilizada: Pascal, compilado no terminal do Mac, compilador fpc e pelo lazarus no Windows

Sistema Operacional Utilizado:

Bruno Tomé: OS X 10.9.4

Cláudio Menezes: Windows 8

#### **Respondendo o questionamento: É melhor uma fila única ou uma fila para cada atendente?**

1. De acordo com os testes realizados, a fila única com 5 atendentes, pois o tempo médio de espera para o cliente ser atendido é na maioria das vezes menor ou igual ao tempo de espera média para 5 filas.
2. Na maioria das vezes a fila única consegue atender um maior número de clientes durante o expediente do que as 5 filas.
3. A quantidade de transações realizadas é a mesma que a quantidade de clientes atendidos em um dia. Como as variáveis: tempo para chegar novos clientes e números de clientes a chegar, são randômicas, e o tempo do expediente é de 300 minutos, ou seja, 5 horas, a quantidade de transações em um dia no banco é praticamente a mesma.
4. Cada transação é selecionada randômicamente para cada cliente, por tanto, a dispersão na porcentagem de escolha de cada transação é bem pequena.
5. Na maioria dos testes realizados a quantidade de clientes após o expediente bancário na simulação de fila única é bem menor do que nas 5 filas.

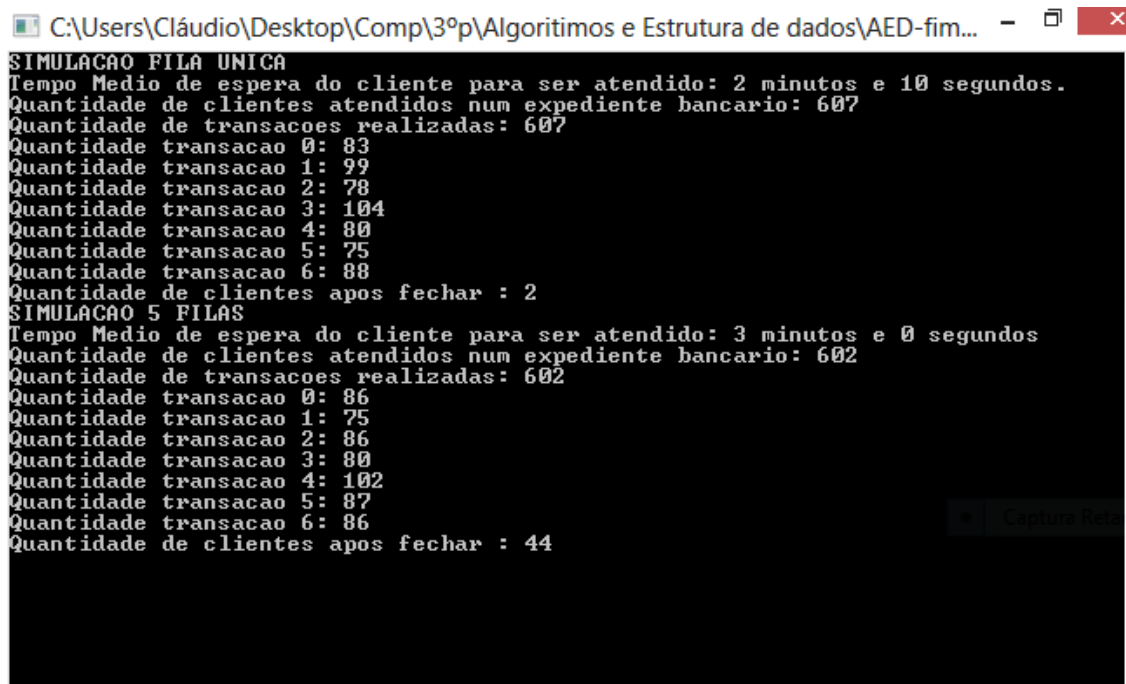
### **4. Bibliografia**

Biblioteca filadinamica.pas oferecida pelo professor.

Enunciado do trabalho.

### **5. Conclusão**

De acordo com os itens 1 e 5, podemos notar claramente que a fila única leva bastante vantagem sobre 5 filas.



```
SIMULACAO FILA UNICA
Tempo Medio de espera do cliente para ser atendido: 2 minutos e 10 segundos.
Quantidade de clientes atendidos num expediente bancario: 607
Quantidade de transacoes realizadas: 607
Quantidade transacao 0: 83
Quantidade transacao 1: 99
Quantidade transacao 2: 78
Quantidade transacao 3: 104
Quantidade transacao 4: 80
Quantidade transacao 5: 75
Quantidade transacao 6: 88
Quantidade de clientes apos fechar : 2
SIMULACAO 5 FILAS
Tempo Medio de espera do cliente para ser atendido: 3 minutos e 0 segundos
Quantidade de clientes atendidos num expediente bancario: 602
Quantidade de transacoes realizadas: 602
Quantidade transacao 0: 86
Quantidade transacao 1: 75
Quantidade transacao 2: 86
Quantidade transacao 3: 80
Quantidade transacao 4: 102
Quantidade transacao 5: 87
Quantidade transacao 6: 86
Quantidade de clientes apos fechar : 44
```

Figura 1. Exibição no console

## 6. Listagem do código fonte

### 6.1. Corpo

```
1 program simulacaoBanco;
2
3 uses libfila, crt;
4
5 var qtdT0, qtdT1, qtdT2, qtdT3, qtdT4, qtdT5, qtdT6 : word;
6
7 // FUNCAO CALCULA MEDIA DO TEMPO DE ATENDIMENTO PARA 5 FILAS E RETORNA
  O VALOR
8
9 function tempoMediaAtendimento5Filas(espera1,espera2,espera3,espera4,
  espera5,clientes1,clientes2,clientes3,clientes4,clientes5:word) :
  real;
10 var media1,media2,media3,media4,media5 :real;
11 begin
12     media1 := (espera1/clientes1);
13     media2 := (espera2/clientes2);
14     media3 := (espera3/clientes3);
15     media4 := (espera4/clientes4);
16     media5 := (espera5/clientes5);
17     tempoMediaAtendimento5Filas := (media1+ media2 + media3 + media4
        + media5)/5;
18 end;
19
20 function menorFila (fila1, fila2, fila3, fila4, fila5 : fila) : integer
  ;
21 var menor, i : integer;
22     vetor : array [1..5] of integer;
23 begin
```

```

24     menor := fila1.tamanho;
25
26     //vetor[1] := fila1.tamanho;
27     vetor[2] := fila2.tamanho;
28     vetor[3] := fila3.tamanho;
29     vetor[4] := fila4.tamanho;
30     vetor[5] := fila5.tamanho;
31     for i:=2 to 5 do
32     begin
33         if (vetor[i] < menor) then
34             menor := vetor[i];
35     end;
36     menorFila := menor;
37 end;
38 //PROCEDIMENTOS INCREMENTA CADA TRANSACAO
39
40 procedure qtdTransacao0 ();
41 begin
42     inc(qtdT0);
43 end;
44
45 procedure qtdTransacao1 ();
46 begin
47     inc(qtdT1);
48 end;
49
50 procedure qtdTransacao2 ();
51 begin
52     inc(qtdT2);
53 end;
54
55 procedure qtdTransacao3 ();
56 begin
57     inc(qtdT3);
58 end;
59
60 procedure qtdTransacao4 ();
61 begin
62     inc(qtdT4);
63 end;
64
65 procedure qtdTransacao5 ();
66 begin
67     inc(qtdT5);
68 end;
69
70 procedure qtdTransacao6 ();
71 begin
72     inc(qtdT6);
73 end;
74
75 // FUNCAO SELECIONA RANDOMICAMENTE UMA TRANSACAO E A RETORNA O VALOR DA
76     TRANSACAO SELECIONADA
77
77 function transacao (): word;
78 var opcao : byte;

```

```

79
80 begin
81     opcao := random(7);
82     case opcao of
83         0 : begin
84             transacao := 1;
85             qtdTransacao0;
86         end;
87         1 : begin
88             transacao := 2;
89             qtdTransacao1;
90         end;
91         2 : begin
92             transacao := 2;
93             qtdTransacao2;
94         end;
95         3 : begin
96             transacao := 1;
97             qtdTransacao3;
98         end;
99         4 : begin
100             transacao := 2;
101             qtdTransacao4;
102         end;
103         5 : begin
104             transacao := 3;
105             qtdTransacao5;
106         end;
107         6 : begin
108             transacao := 10;
109             qtdTransacao6;
110         end;
111     end;
112 end;
113
114 //PROCEDIMENTO SIMULACAO FILA UNICA
115 procedure simulacaoFilaBanco(filaUnica:fila);
116 var i, qtdCliente, totalClientesAtendidos, esperaTotalMedia : word;
117     tempoChegada, tempoTotal : integer;
118     parada :boolean;
119     caixa : array [1..5] of integer;
120
121 begin
122     esperaTotalMedia := 0;
123     totalClientesAtendidos := 0;
124     tempoTotal := 0;
125     tempoChegada := 0;
126     qtdCliente := 0;
127     parada:= false;
128     caixa[1]:=0;
129     caixa[2]:=0;
130     caixa[3]:=0;
131     caixa[4]:=0;
132     caixa[5]:=0;
133
134 // INICIO DO EXPEDIENTE VAI ATE 300 MIN = 5 HORAS

```

```

135 while (tempoTotal <= 300) do
136 begin
137     qtdCliente := random(20); // Quantidade de clientes
138
139     // INICIO INSERIR CLIENTE
140     for i:=1 to qtdCliente do
141     begin
142         inserir(filaUnica,transacao());
143         esperaTotalMedia:= filaUnica.fim^.item + esperaTotalMedia;
144         // TEMPO DE ESPERA TOTAL
145     end;
146     // FIM INSERIR CLIENTE
147     for i:=1 to 5 do
148     begin
149         caixa[i]:= tempoChegada;
150     end;
151     //INICIO REMOVER CLIENTE ATENDIDO
152     while ((parada = false) and (not(vazia(filaUnica)))) do
153     begin
154         // DECREMENTA O TEMPO
155
156         for i:=1 to 5 do
157         begin
158
159             if((caixa[i] > 0) and not(vazia(filaUnica))) then
160             begin
161                 caixa[i] := caixa[i] - filaUnica.inicio^.item;
162                 remover(filaUnica);
163             end;
164         end;
165
166         if ((caixa[1] <= 0) and (caixa[2] <= 0) and (caixa[3] <= 0)
167             and (caixa[4] <= 0) and (caixa[5] <=0)) then
168         begin
169             parada := true;
170         end;
171     end;
172
173     //FIM REMOVER CLIENTE ATENDIDO
174
175     totalClientesAtendidos := qtdCliente + totalClientesAtendidos;
176     tempoChegada := random(10); // Tempo para chegar novos clientes
177     tempoTotal := tempoTotal + tempoChegada;
178     parada:= false;
179 end; // FIM DO WHILE
180 // FIM DO EXPEDIENTE DE 300 MIN = 5 HORAS
181
182 //IMPRESSAO DOS RESULTADOS APOS SIMULACAO
183 writeln('Tempo Medio de espera do cliente para ser atendido: ',
        (esperaTotalMedia div totalClientesAtendidos), ' minutos e
        ', (((esperaTotalMedia mod totalClientesAtendidos)/60))
        :0:0, ' segundos.');
```

```

182 writeln('Quantidade de clientes atendidos num expediente
    bancario: ',totalClientesAtendidos);
183 writeln('Quantidade de transacoes realizadas: ',
    totalClientesAtendidos);

```

```

184     writeln('Quantidade transacao 0: ',qtdT0);
185     writeln('Quantidade transacao 1: ',qtdT1);
186     writeln('Quantidade transacao 2: ',qtdT2);
187     writeln('Quantidade transacao 3: ',qtdT3);
188     writeln('Quantidade transacao 4: ',qtdT4);
189     writeln('Quantidade transacao 5: ',qtdT5);
190     writeln('Quantidade transacao 6: ',qtdT6);
191     writeln('Quantidade de clientes apos fechar : ', filaUnica.
        tamanho);
192 end;
193
194 // PROCEDIMENTO SIMULACAO COM 5 FILAS
195 procedure simulacao5FilaBanco(fila1, fila2, fila3, fila4, fila5 : fila)
196 ;
197 var qtdCliente, totalClientesAtendidos, tempoTotal,tempoEsperaFila1,
198     tempoEsperaFila2,tempoEsperaFila3,tempoEsperaFila4,tempoEsperaFila5
199     ,
200     totalClienteFila1,totalClienteFila2,totalClienteFila3,
201     totalClienteFila4,
202     totalClienteFila5,auxTempoChegada, somaFilas : word;
203     tempoChegada : integer;
204     tempoMedioFinal : real;
205
206 begin
207
208     totalClientesAtendidos := 0;
209     tempoTotal := 0;
210     tempoChegada := 0;
211     qtdCliente := 0;
212     tempoEsperaFila1:=0;
213     tempoEsperaFila2:=0;
214     tempoEsperaFila3:=0;
215     tempoEsperaFila4:=0;
216     tempoEsperaFila5:=0;
217     totalClienteFila1:=0;
218     totalClienteFila2:=0;
219     totalClienteFila3:=0;
220     totalClienteFila4:=0;
221     totalClienteFila5:=0;
222
223     // INICIO DO EXPEDIENTE VAI ATE 300 MIN = 5 HORAS
224     while (tempoTotal <= 300) do
225     begin
226         qtdCliente := random(20); // Quantidade de clientes
227         totalClientesAtendidos := qtdCliente + totalClientesAtendidos;
228         // INICIO INSERIR CLIENTE
229         while (qtdCliente > 0) do
230         begin
231             // INSERIR FILA 1
232             if ((fila1.tamanho = menorFila(fila1, fila2, fila3, fila4,
233                 fila5)) and (qtdCliente > 0)) then
234             begin
235                 inserir(fila1,transacao());
236                 tempoEsperaFila1:=fila1.fim^.item + tempoEsperaFila1;
237                 // TEMPO DE ESPERA PARA CADA FILA
238                 totalClienteFila1:= totalClienteFila1 +1;

```

```

234         qtdCliente:= qtdCliente - 1;
235     end;
236
237     // INSERIR FILA 2
238     if ((fila2.tamanho = menorFila(fila1, fila2, fila3, fila4,
239         fila5)) and (qtdCliente > 0)) then
240     begin
241         inserir(fila2,transacao());
242         tempoEsperaFila2:=fila2.fim^.item + tempoEsperaFila2;
243         // TEMPO DE ESPERA PARA CADA FILA
244         inc(totalClienteFila2);
245         qtdCliente:= qtdCliente - 1;
246
247     end;
248
249     // INSERIR FILA 3
250     if ((fila3.tamanho = menorFila(fila1, fila2, fila3, fila4,
251         fila5)) and (qtdCliente > 0)) then
252     begin
253         inserir(fila3,transacao());
254         tempoEsperaFila3:=fila3.fim^.item + tempoEsperaFila3;//
255         TEMPO DE ESPERA PARA CADA FILA
256         inc(totalClienteFila3);
257         qtdCliente:= qtdCliente - 1;
258
259     end;
260
261     // INSERIR FILA 4
262     if ((fila4.tamanho = menorFila(fila1, fila2, fila3, fila4,
263         fila5)) and (qtdCliente > 0)) then
264     begin
265         inserir(fila4,transacao());
266         tempoEsperaFila4:=fila4.fim^.item + tempoEsperaFila4;//
267         TEMPO DE ESPERA PARA CADA FILA
268         inc(totalClienteFila4);
269         qtdCliente:= qtdCliente - 1;
270
271     end;
272
273     // INSERIR FILA 5
274     if ((fila5.tamanho = menorFila(fila1, fila2, fila3, fila4,
275         fila5)) and (qtdCliente > 0)) then
276     begin
277         inserir(fila5,transacao());
278         tempoEsperaFila5:=fila5.fim^.item + tempoEsperaFila5;//
279         TEMPO DE ESPERA PARA CADA FILA
280         inc(totalClienteFila5);
281         qtdCliente:= qtdCliente - 1;
282
283     end;
284 end; // FIM INSERIR CLIENTE
285
286 // INICIO REMOVER CLIENTE ATENDIDO DE CADA FILA
287
288 auxTempoChegada := tempoChegada; // TEMPO DE CHEGADA EH
289 IGUAL PARA TODAS FILAS
290 //FILA 1
291 while ((tempoChegada > 0) and not(vazia(fila1))) do
292 begin

```



```

281         tempoChegada := tempoChegada - fila1.inicio^.item;
282         remover(fila1);
283     end;
284
285     //FILA 2
286     tempoChegada := auxTempoChegada;
287
288     while ((tempoChegada > 0) and not(vazia(fila2))) do
289     begin
290         tempoChegada := tempoChegada - fila2.inicio^.item;
291         remover(fila2);
292     end;
293
294     //FILA 3
295     tempoChegada := auxTempoChegada;
296     while ((tempoChegada > 0) and not(vazia(fila3))) do
297     begin
298         tempoChegada := tempoChegada - fila3.inicio^.item;
299         remover(fila3);
300     end;
301
302     //FILA 4
303     tempoChegada := auxTempoChegada;
304
305     while ((tempoChegada > 0) and not(vazia(fila4))) do
306     begin
307         tempoChegada := tempoChegada - fila4.inicio^.item;
308         remover(fila4);
309     end;
310
311     //FILA 5
312     tempoChegada := auxTempoChegada;
313
314     while ((tempoChegada > 0) and not(vazia(fila5))) do
315     begin
316         tempoChegada := tempoChegada - fila5.inicio^.item;
317         remover(fila5);
318     end;
319
320     //FIM REMOVER CLIENTE ATENDIDO DE CADA FILA
321
322     tempoChegada := random(10); // Tempo para chegar novos clientes
323     tempoTotal := tempoTotal + tempoChegada;
324 end; // FIM DO WHILE
325 // FIM DO EXPEDIENTE DE 300 MIN = 5 HORAS
326
327 //IMPRESSAO DOS RESULTADOS APOS SIMULACAO
328 tempoMedioFinal := tempoMediaAtendimento5Filas(tempoEsperaFila1
    ,tempoEsperaFila2,tempoEsperaFila3,tempoEsperaFila4,
    tempoEsperaFila5,totalClienteFila1,totalClienteFila2,
    totalClienteFila3,totalClienteFila4,totalClienteFila5);
329 writeln('Tempo Medio de espera do cliente para ser atendido: ',
    (tempoMedioFinal):0:2, ' minutos. ');
330 writeln('Quantidade de clientes atendidos num expediente
    bancario: ', totalClientesAtendidos);

```

```

331         writeln('Quantidade de transacoes realizadas: ',
                 totalClientesAtendidos);
332         writeln('Quantidade transacao 0: ', qtdT0);
333         writeln('Quantidade transacao 1: ', qtdT1);
334         writeln('Quantidade transacao 2: ', qtdT2);
335         writeln('Quantidade transacao 3: ', qtdT3);
336         writeln('Quantidade transacao 4: ', qtdT4);
337         writeln('Quantidade transacao 5: ', qtdT5);
338         writeln('Quantidade transacao 6: ', qtdT6);
339         somaFilas := fila1.tamanho + fila2.tamanho + fila3.tamanho +
                 fila4.tamanho + fila5.tamanho;
340         writeln('Quantidade de clientes apos fechar : ', somaFilas);
341     end;
342
343 // PROGRAMA PRINCIPAL
344
345 var
346     filaUnica, fila1, fila2, fila3, fila4, fila5: fila;
347
348 begin
349     clrscr;
350     randomize;
351     qtdT0 := 0;
352     qtdT1 := 0;
353     qtdT2 := 0;
354     qtdT3 := 0;
355     qtdT4 := 0;
356     qtdT5 := 0;
357     qtdT6 := 0;
358     criar(filaUnica);
359
360     writeln('SIMULACAO FILA UNICA');
361     simulacaoFilaBanco(filaUnica); // CHAMADA DA SIMULACAO FILA UNICA
362     qtdT0 := 0;
363     qtdT1 := 0;
364     qtdT2 := 0;
365     qtdT3 := 0;
366     qtdT4 := 0;
367     qtdT5 := 0;
368     qtdT6 := 0;
369     writeln('SIMULACAO 5 FILAS');
370     criar(fila1);
371     criar(fila2);
372     criar(fila3);
373     criar(fila4);
374     criar(fila5);
375
376     simulacao5FilaBanco(fila1, fila2, fila3, fila4, fila5); // CHAMADA
        DA SIMULACAO COM 5 FILAS
377     readln;
378 end.

```

**simulacaoBanco.pas**

## 6.2. Biblioteca

```
1 | unit libfila;
```

```

2
3 interface
4
5 type
6     elem=word;
7
8     noh=record
9         item: elem;
10        proximo: ^noh;
11    end;
12
13    fila = record
14        inicio: ^noh;
15        fim: ^noh;
16        tamanho: integer;
17    end;
18
19 procedure criar(var f:fila);
20 function vazia (var f:fila):boolean;
21 function inserir (var f:fila; x:elem): boolean;
22 function remover (var f:fila): boolean;
23 function elemento_inicio(var f:fila): elem;
24 function elemento_fim(var f:fila): elem;
25
26 implementation
27
28 procedure criar(var f:fila);
29     begin
30         f.inicio:=nil;
31         f.fim:=nil;
32         f.tamanho:=0;
33     end;
34
35 function vazia (var f:fila):boolean;
36     begin
37         if f.inicio=nil then
38             vazia:=true
39         else
40             vazia:=false;
41     end;
42
43 function inserir (var f:fila; x:elem): boolean;
44     var
45         novo_noh: ^noh;
46     begin
47         new(novo_noh); { criar novo noh }
48         if (novo_noh <> nil) then
49             begin
50                 if (vazia (f)) then
51                     begin
52                         novo_noh^.item := x;
53                         novo_noh^.proximo := nil;
54                         f.fim := novo_noh;
55                         f.inicio := f.fim;
56                         f.tamanho:=f.tamanho+1;
57                         inserir:=true;

```

```

58         end
59     else
60     begin
61         novo_noh^.item := x;
62         novo_noh^.proximo := nil;
63         f.fim^.proximo := novo_noh;
64         f.fim := novo_noh;
65         f.tamanho:= f.tamanho +1;
66         inserir:=true;
67     end;
68 end
69 else
70     inserir:=false;
71 end;
72
73 function remover (var f:fila): boolean;
74 var
75     aux : ^noh; { armazena o noh a ser excluido }
76 begin
77     if (not vazia(f)) then
78     begin
79         aux := f.inicio;
80         f.inicio:=f.inicio^.proximo;
81         dispose (aux);
82         f.tamanho:=f.tamanho -1;
83         remover:=true;
84     end
85     else
86         remover:=false;
87     end;
88
89 function elemento_inicio(var f:fila): elem;
90 begin
91     elemento_inicio := f.inicio^.item;
92 end;
93
94 function elemento_fim (var f:fila): elem;
95 begin
96     elemento_fim := f.fim^.item;
97 end;
98 end.

```

**libfila.pas**