

## Unidade 5

# Filas, Alocação Seqüencial e Estática

Vetor circular existe ou é fruto de nossa imaginação?

### Nossos Objetivos nesta Unidade:

- Entender o que é e para que serve uma estrutura do tipo Fila;
- Desenvolver habilidade para implementar uma estrutura de armazenamento do tipo Fila, como um Tipo Abstrato de Dados, com alocação seqüencial e estática de memória;
- Desenvolver habilidade para manipular Filas através dos operadores definidos para o Tipo Abstrato de Dados Fila.

### O Que É uma Fila?

Quando temos que esperar para sermos atendidos, estamos em uma fila. Fila para comprar passagem, fila para ser atendido pelo caixa do banco, pelo caixa do supermercado. Fila é isso mesmo.



**Figura 5.1 Ilustração do conceito de Fila**

Em uma fila séria, quem chega primeiro é atendido primeiro. Ou seja, novos elementos entram em uma fila sempre no final da fila. E o elemento que é atendido chamado o primeiro da fila.

### Definição: Fila

Fila é uma estrutura para armazenar um conjunto de elementos, que funciona da seguinte forma:

- Novos elementos entram no conjunto, exclusivamente, no fim da fila;
- O único elemento que posso retirar da fila em um dado momento, é o primeiro elemento da fila.




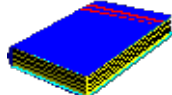

### Do Inglês: *Queue*, *FIFO*

Uma Fila (em Inglês: *Queue*) é uma estrutura que obedece o critério F.I.F.O.: *First In, First Out*. Ou seja, o primeiro elemento que entrou no conjunto será o primeiro a sair.

### Aplicação para Contextualização: Automação de Biblioteca

Considere que você, e um grupo de projetistas, precisam desenvolver um sistema para automação de uma biblioteca, com as seguintes características:

- Todos os livros devem ser cadastrados;
- O sistema deve informar se um determinado livro está ou não disponível nas estantes;
- Caso o livro não estiver disponível, o usuário poderá aguardar pela liberação do livro se cadastrando em uma fila de espera;
- Quando o livro for devolvido e liberado, o primeiro da fila deve ser contatado para vir buscá-lo.

Fila de Espera para o Livro	Livros do Acervo	Livro Disponível?
	 Trigonometria	não
	 Química Inorgânica	não
Fila vazia!	 Estruturas de Dados	sim

**Tabela 5.1 Filas de Espera por Livros em uma Biblioteca**

### Exercício 5.1: Detalhar o Tipo Abstrato de Dados Fila

Estamos na fase de Projeto do sistema de automação de uma biblioteca. O grupo de projetistas identificou a necessidade de um módulo para implementar a fila de espera, para cada livro. E coube a você definir melhor, e depois implementar, esse módulo. O que você faria a partir de então? Pare um pouco aqui, antes de prosseguir a leitura, e defina quais seriam seus próximos passos.

**Passo 1: Que Dados Devem Ser Armazenados?** Em algum momento você precisaria identificar os elementos do mundo real que são relevantes. Seriam esses os elementos que a fila teria que armazenar. Podemos considerar, por exemplo, que teríamos uma fila de espera para cada livro da biblioteca, e que cada fila de espera deveria armazenar o nome e o telefone de cada usuário que estiver na fila, ou seja, que estiver esperando pelo livro em questão.

**Passo 2: Que Operações Podem Ser Aplicadas aos Dados Armazenados?** Em algum momento também seria preciso definir a funcionalidade do Tipo Abstrato de Dados Fila. Ou seja, quais operações podem ser aplicadas aos dados armazenados na fila.

### Exercício 5.2: Definir a Funcionalidade do Tipo Abstrato de Dados Fila

Na Unidade 3 definimos um conjunto de operações para o Tipo Abstrato de Dados Pilha. As operações eram: Empilha, Desempilha, e outras. Defina operações para o Tipo Abstrato de Dados Fila. Pense no sistema de automação da biblioteca. Que operações seriam necessárias para o bom funcionamento do sistema?

### Funcionalidade de uma Fila

A Tabela 5.2 apresenta um possível conjunto de operações, para o Tipo Abstrato de Dados Fila.

Operações e Parâmetros	Funcionamento
Fila.Entra( QualFila, QualElemento, DeuCerto? )	Insere o elemento passado como parâmetro QualElemento, na fila passada no parâmetro

	QualFila. O elemento deve ser inserido no final da fila, ou seja, passando a ser o último elemento da fila. O parâmetro DeuCerto? indica se a operação foi bem sucedida ou não.
Fila.Sai( QualFila, QualElemento, DeuCerto? )	Retira o primeiro elemento da fila passada no parâmetro QualFila, retornando o valor do elemento que foi retirado no parâmetro QualElemento. O parâmetro DeuCerto? indica se a operação foi bem sucedida ou não.
Fila.Vazia?( QualFila )	Verifica se a fila passada como parâmetro (QualFila) está ou não vazia (vazia = sem nenhum elemento).
Fila.Cheia?( QualFila )	Verifica se a fila passada como parâmetro (QualFila) está ou não cheia (cheia = situação em que é impossível inserir mais elementos na fila).
Fila.Cria( QualFila )	Cria uma fila, iniciando sua situação como vazia.

**Tabela 5.2 Funcionalidade de uma Fila**

Uma Fila é um conjunto ordenado de elementos, ou seja, a ordem dos elementos no conjunto é importante. E a ordem dos elementos no conjunto é definida pela ordem em que os elementos entram no conjunto. Por exemplo, se em uma Fila F foram inseridos 4 elementos: A, B, C e D, nessa ordem, podemos representar a Fila F com um diagrama como o da Figura 5.2.

primeiro à      **A**            **B**            **C**            **D**            β último

**Figura 5.2 Fila F com quatro elementos A, B, C e D, inseridos nessa ordem**

Se eu quiser retirar um elemento da Fila F nesse momento, o único elemento que poderei retirar da fila será o elemento A, que é o primeiro elemento da fila. E se eu quiser inserir um elemento E na fila F, devo inseri-lo como o último elemento da fila. A Figura 5.3 apresenta a mesma Fila F da Figura 5.2, após a retirada de um elemento (elemento A) e a inserção de um novo elemento (elemento E).

primeiro à      **B**            **C**            **D**            **E**            β último

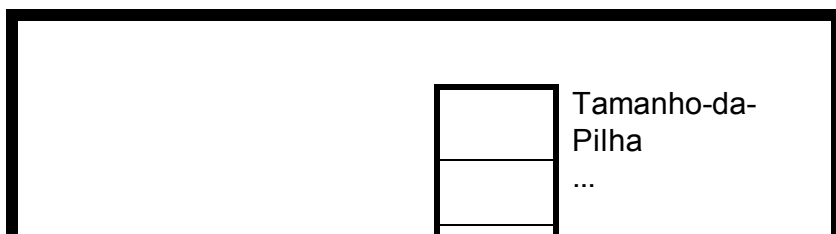
**Figura 5.3 Fila após a retirada de um elemento, e da inserção de novo elemento**

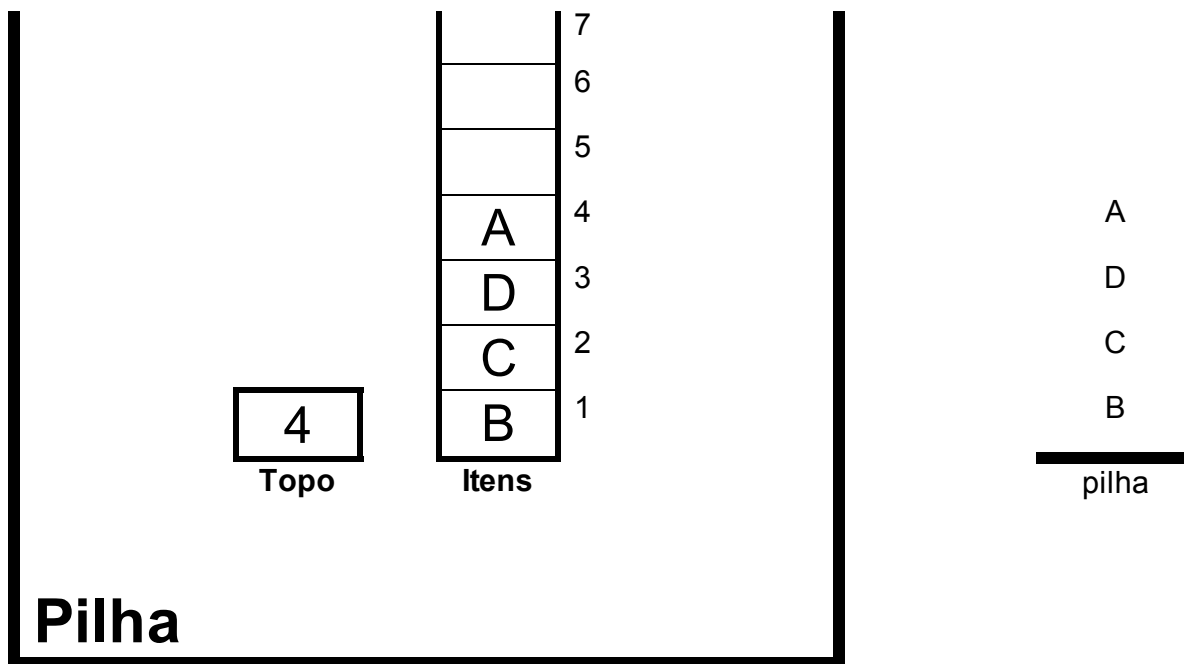
### **Simulação 5.1: Operações de uma Fila**

Teste o funcionamento das operações Entra e Sai, de uma Fila, interagindo com a [Simulação 5.1: Operações de uma Fila](#).

### **Implementação de uma Fila com Alocação Sequencial e Estática**

Na Unidade 3 implementamos uma Pilha com alocação sequencial e estática de memória. Utilizamos um vetor e um apontador para o Topo da Pilha, conforme mostra a Figura 5.4.





**Figura 5.4 Pilha com alocação seqüencial e estática de memória.**

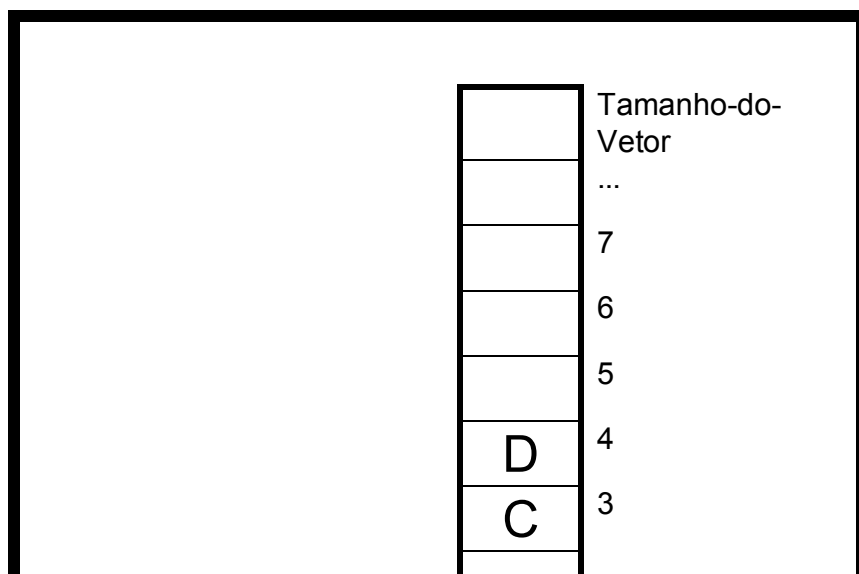
Pilha é pilha, e fila é fila, cada estrutura funciona de sua própria maneira. Mas considerando que a alocação de memória será semelhante, não dá para aproveitar algo de nossa implementação de Pilha, feita na Unidade 3?

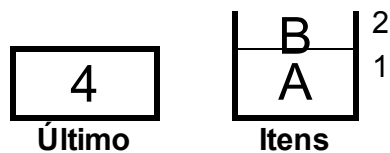
### **Exercício 5.3 Esquema da Implementação de uma Fila**

Procure adaptar a implementação de Pilha com alocação seqüencial e estática, da Unidade 3, e implemente uma Fila. Primeiramente faça um esquema (desenho) de como ficaria sua Fila. Depois implemente as operações Entra, Sai, e as demais definidas na Tabela 5.2.

#### **Primeira Solução: Com Realocação de Elementos**

Uma possível implementação de Fila, com alocação seqüencial e estática, é representada na Figura 5.5. A é o primeiro da fila, B é o segundo, C é o terceiro e D é o quarto e último elemento da fila. Temos uma variável, Último, que indica a posição do vetor em que está o último elemento da fila. Não precisamos de um indicador da posição do primeiro elemento da fila, pois, nesta implementação, vamos manter o primeiro elemento da fila sempre na primeira posição do vetor.

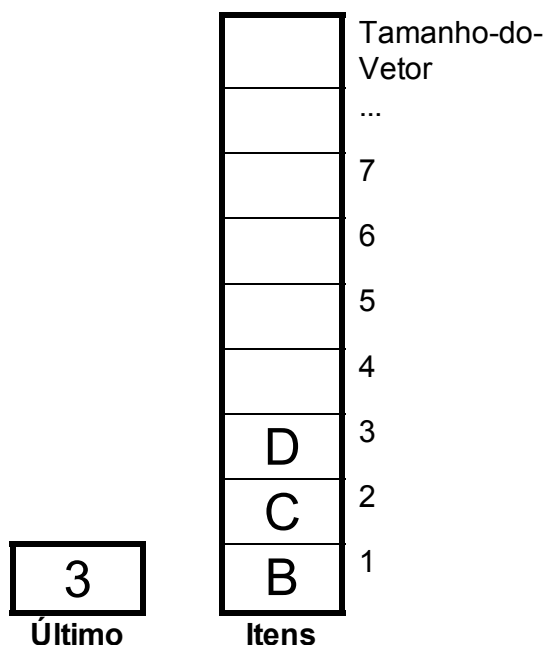




## Fila

Figura 5.5 Fila com realocação de elementos.

Se quisermos retirar um elemento da fila, como nossa fila é séria ( J ) temos que retirar o primeiro elemento, que está na posição 1 do vetor. E para manter o primeiro elemento da fila sempre na posição 1, temos que re-alocar todos os demais elementos, uma posição para baixo, no vetor: o elemento B sai da posição 2 e vai para a posição 1, C sai da 3 para a 2, e assim por diante. Assim, a retirada de um elemento da fila, resultaria na situação esquematizada na Figura 5.6. Note que o valor do indicador Último foi atualizado para 3. Isso significa que o último elemento da fila está na posição 3 do vetor.



## Fila

Figura 5.6 Fila após a retirada de um elemento – o elemento A.

Se quisermos inserir um novo elemento na fila, ele irá entrar na posição 4 do vetor, agora vazia, e o indicador Último seria atualizado (para o valor 4). Veja a Figura 5.7.



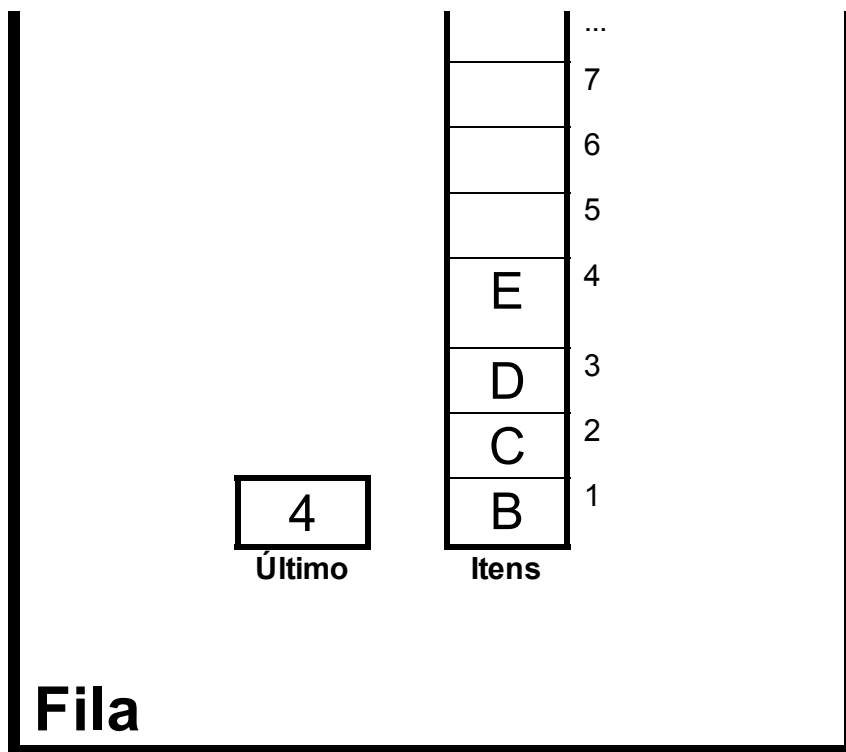


Figura 5.7 Fila após a inserção do elemento E.

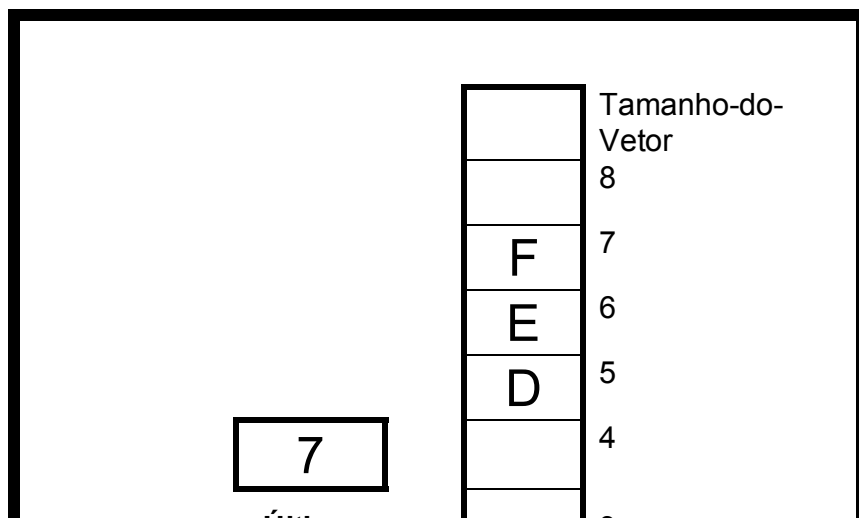
#### Exercício 5.4 Algoritmos para Fila, Com Realocação de Elementos

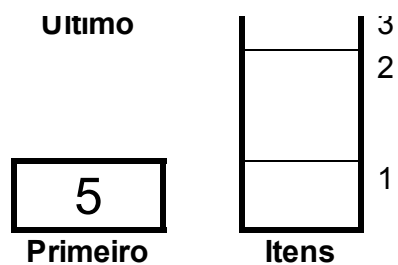
Implemente as operações definidas na Tabela 5.2 (entra na fila, sai da fila, vazia, cheia e cria), utilizando essa estratégia de implementação, com realocação de elementos.

#### Tentativa de Solução Sem Realocação de Elementos

Podemos ajustar a implementação da fila de modo a não termos a necessidade de re-alocar os elementos a cada operação de retirar um elemento da fila? Sim, podemos. Como podemos fazer isso? Pense um pouco antes de prosseguir.

Na solução com realocação, mantemos o primeiro elemento da fila sempre na primeira posição do vetor. Se deixarmos o primeiro elemento avançar para a posição, 2, 3 (e assim por diante) do vetor, eliminamos a re-alocação dos elementos. Mas para isso precisamos de um novo apontador, ou ponteiro, denominado Primeiro. Esse apontador irá indicar a posição do vetor em que se encontra o primeiro elemento da fila. Veja a Figura 5.8.





## Fila

**Figura 5.8 Fila sem realocação de elementos.**

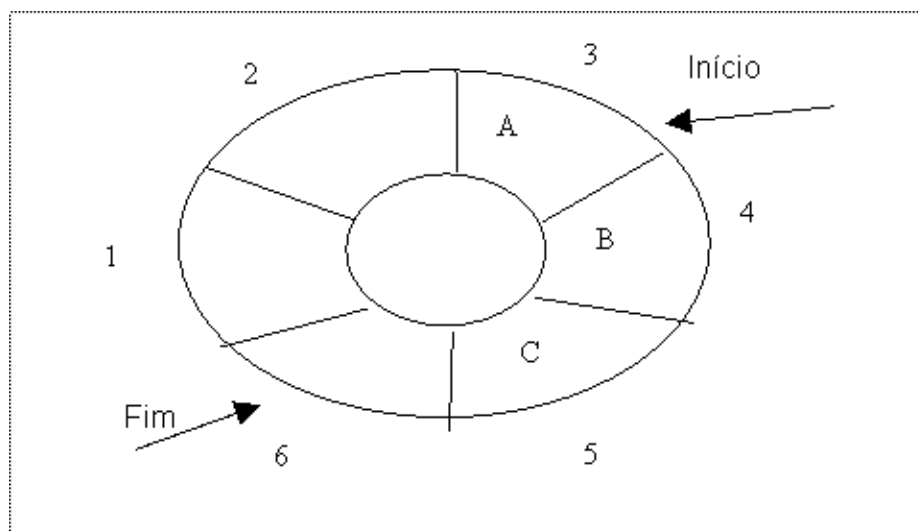
Essa implementação nos apresenta um novo problema a resolver. Se avançarmos tanto o apontador Último quanto o apontador Primeiro, eles vão passar para a posição 6, 7, e assim por diante, e uma hora o vetor acaba! Mas note que as posições iniciais (na Figura 5.8, as posições 1, 2, 3 e 4 do vetor) estão vazias. O problema a resolver no momento passa a ser: como reaproveitar as posições do vetor que forem ficando vazias? Reflita sobre essa questão, antes de prosseguir. Tente desenvolver sua própria solução.

### Tentativa de Solução com Vetor Circular

Podemos imaginar que o nosso vetor, ao invés de reto, é circular, conforme representado na Figura 5.9.

#### Vetor circular existe ou é fruto de nossa imaginação?

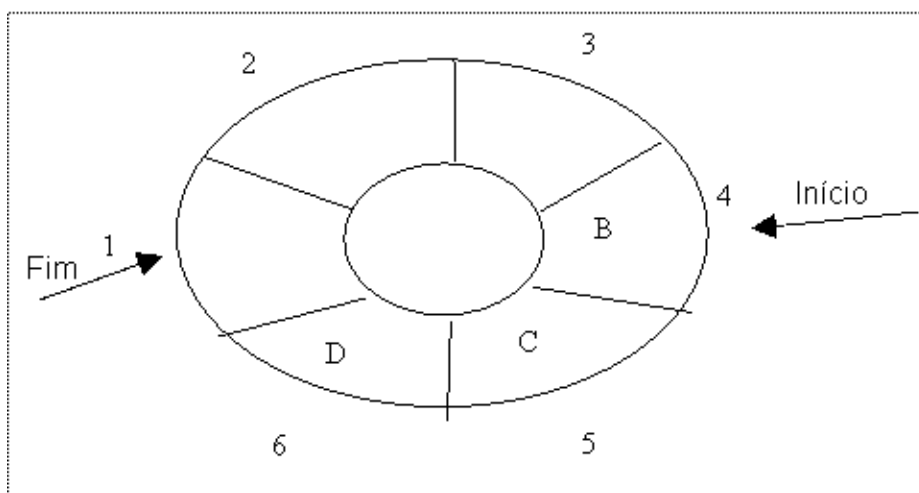
É fruto de nossa imaginação mesmo. Todas essas representações, ou seja, todos esses desenhos que estamos fazendo, são apenas representações abstratas. Só existem em nossa imaginação. Você nunca conseguirá ver, tocar ou cheirar um vetor, nem reto nem circular. Logo, podemos “imaginar” um vetor reto, e podemos “imaginar” também um vetor torto, ou circular. Na computação, trabalhamos com elementos abstratos. Se estiver confuso, leia novamente na Unidade 2, ou em um dicionário, os significados do termo “abstrato”.



**Figura 5.9 Fila implementada com um vetor circular.**

Note que na fila em um vetor circular, representada na Figura 5.9, fizemos algumas mudanças. Mudamos o nome do apontador do primeiro elemento da fila para *Início*. Esse apontador está indicando o início da fila. E ao invés de apontar o último elemento da fila, estamos indicando, com o apontador *Fim*, o final da fila, ou seja, a posição do vetor em que entrará o próximo elemento da fila.

A partir da situação representada na Figura 5.9, se retirarmos um elemento, retiraremos o elemento A, que está na posição 3 do vetor. O apontador *Início* deverá avançar para a posição 4. Se inserirmos um novo elemento D na fila, ele deverá entrar na primeira posição vaga, apontada pelo apontador *Fim*, ou seja, a posição 6. Veja o resultado na Figura 5.10.



**Figura 5.10 Fila com um vetor circular, após a retirada de A e a entrada de D**

Note na Figura 5.10 que o ponteiro *Fim* avançou da posição 6 para a posição 1 do vetor. Ou seja, em um “vetor circular”, quando chegamos ao final do vetor, retornamos para o início do vetor.

### **Exercício 5.5 Definir Condições para Fila cheia e para Fila Vazia**

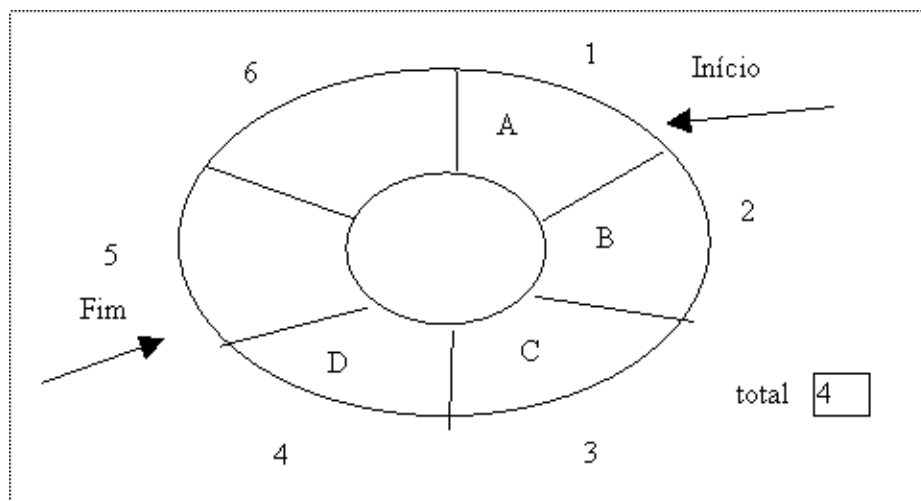
Nessa implementação com vetor circular, esquematizada na Figura 5.10, temos um problema a resolver. Precisamos implementar a operação Cheia, que verifica se a fila está ou não cheia (nesse caso, se o vetor está com todas as posições ocupadas), e a operação vazia, que verifica se a fila está ou não vazia (vetor com todas as posições disponíveis). Defina condições para fila cheia e fila vazia, lembrando que a condição não pode ser a mesma. Pense em uma solução antes de prosseguir.

### **Solução com Indicador de Total de Elementos**

Na solução com vetor circular, podemos diferenciar a condição de fila vazia da condição de fila cheia



Na solução com vetor circular, podemos diferenciar a condição de fila vazia da condição de fila cheia criando uma nova variável, que armazena o total de elementos, conforme a Figura 5.11.



**Figura 5.11 Fila em vetor circular, com indicador do total de elementos.**

Quando a fila está vazia, o total de elementos é igual a zero. Quando a fila está cheia, o total de elementos é igual ao tamanho do vetor.

### Exercício 5.6 Estrutura + Operações Cria, Cheia e Vazia

Constante      TamanhoDoVetor = 100      { ou outro valor }  
 Tipo            Elemento        = char            { ou outro tipo }  
                   Fila                = registro  
    Início, fim, total do tipo inteiro  
    Itens : vetor de 1 a TamanhoDoVetor de elementos do tipo Elemento

Cria(parâmetro por referência F do tipo Fila)  
 /\* Cria uma fila F, iniciando sua situação como vazia. \*/  
     F.Total = 0  
     F.Início = 1  
     F.Fim = 1

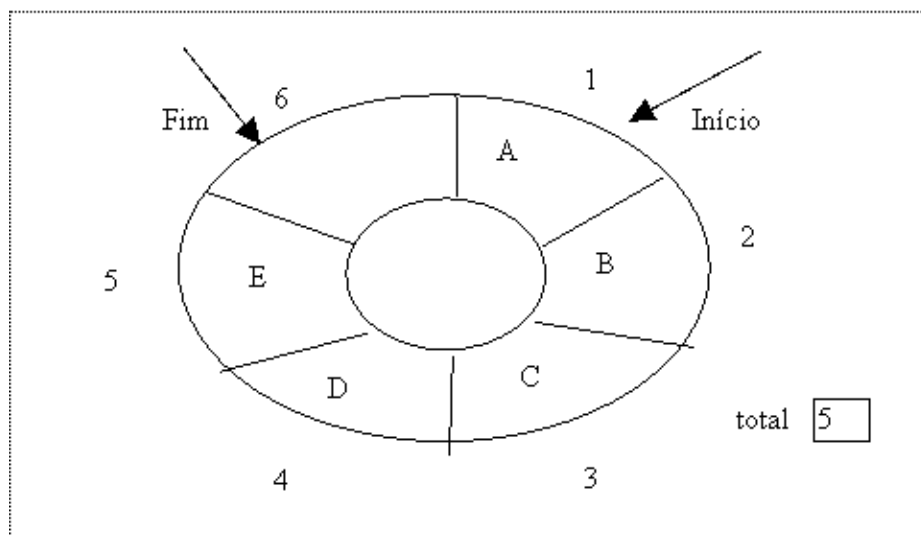
Boolean Vazia?( parâmetro por referência F do tipo Fila )  
 /\* Verifica se a Fila F está ou não vazia. \*/  
     Se F.total == 0  
     Então Retorne Verdadeiro /\* a fila está vazia \*/  
     Senão Retorne Falso

Boolean Cheia?( parâmetro por referência F do tipo Fila )  
 /\* Verifica se a Fila F está ou não cheia. \*/  
     Se F.total == TamanhoDoVetor  
     Então Retorne Verdadeiro /\* a fila está cheia \*/  
     Senão Retorne Falso

### Operação Entra na Fila

Para inserir um novo elemento na fila é preciso primeiramente verificar se a fila não está cheia, ou seja, verificar se há espaço no vetor para inserir o novo elemento. Se a fila não estiver cheia, inserimos o novo elemento na posição indicada pelo apontador Fim. Considerando como situação inicial a Figura 5.11, o

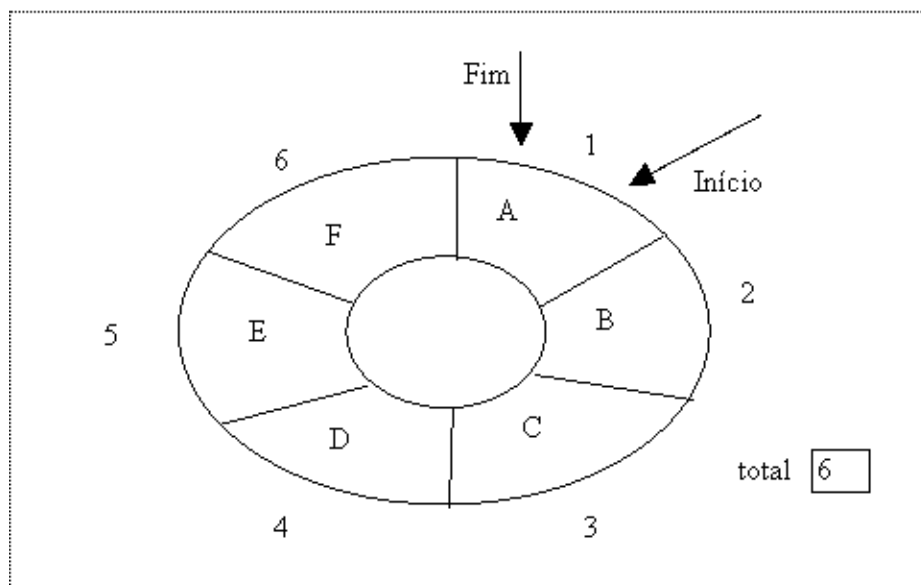
elemento na posição indicada pelo apontador Fim. Considerando como situação inicial a Figura 5.11, o novo elemento seria inserido na posição 5 do vetor. Após inserir, temos que avançar o apontador Fim. Na situação da Figura 5.11, o apontador Fim avançaria da posição 5 para a posição 6. E o contador Total precisaria ser aumentado, de 4 para 5, resultando na situação da Figura 5.12.



**Figura 5.12 Resultado da operação entra na fila.**

Avançar os apontadores Fim ou Início, em um vetor circular, significa aumentar o valor do apontador em 1 (de 1 para 2, ou de 2 para 3, e assim por diante), salvo na situação em que o apontador está na última posição no vetor. Nesse caso, avançar o apontador significa passar da última posição para a primeira. Na situação da Figura 5.12, se quisermos inserir mais um elemento, teríamos que avançar o apontador Fim, tirando-o da posição 6 e colocando-o na posição 1. Veja a situação resultante na Figura 5.13.

Note que, na situação da Figura 5.13 a fila está cheia. Tanto o apontador Início quanto o apontador Fim estão na posição 1 do vetor. E o contador Total é igual à constante TamanhoDoVetor.



### Figura 5.13 Fila cheia.

#### Exercício 5.7 Implementação da Operação Entra na Fila

Tente propor uma solução a esse exercício. Você encontrará uma solução no final da unidade, mas não olhe a solução. Procure realmente fazer, e depois olhar a solução. É assim que se aprende.

Entra (parâmetro por referência F do tipo Fila, parâmetro Elemento do tipo Char, parâmetro por referência Erro do tipo Boolean)

/\* Insere o parâmetro Elemento no final da fila F. O parâmetro Erro indica se a ocorreu ou não algum erro na operação. O único erro verificado é se houve tentativa de inserir um elemento em uma fila cheia \*/

#### Exercício 5.8 Implementação da Operação Sai da Fila

Tente propor uma solução a esse exercício. Você encontrará uma solução no final da unidade, mas não olhe a solução. Procure realmente fazer, e depois olhar a solução. É assim que se aprende.

Sai (parâmetro por referência F do tipo Fila, parâmetro por referência Elemento do tipo Char, parâmetro por referência Erro do tipo Boolean)

/\* Retira o primeiro elemento da fila F, retornando o valor do primeiro elemento no parâmetro Elemento. O parâmetro Erro indica se a ocorreu ou não algum erro na operação. O único erro verificado é se houve tentativa de retirar um elemento em uma fila vazia \*/

#### Exercícios de fixação

1. [O que é \(e como funciona\) uma fila?](#)
2. [Cite aplicações](#) de uma estrutura do tipo fila.
3. Faça um [esquema da implementação estática e seqüencial de uma fila](#) e explique resumidamente o funcionamento

3. Faça um [sequência de implementações estática e dinâmica de uma fila](#), e explique recursivamente e iterativamente.
4. Compare as implementações com realocação de elementos ([Exercício 5.4](#)) e em vetor circular ([exercícios 5.5 a 5.8](#)). Na sua opinião, qual é mais interessante? Por que?
5. Implemente, em uma linguagem de programação, o Tipo Abstrato de Dados Fila. Faça um programa simples para testar sua implementação de fila.
6. Desenvolva uma função para testar se uma fila F1 tem mais elementos do que uma fila F2. Pense em desenvolver da melhor maneira possível. Lembre do conceito de Tipos Abstratos de Dados (Unidade 2).

## Solução de exercícios de aula – operações Entra e Sai da fila

Entra (parâmetro por referência F do tipo Fila, parâmetro Elemento do tipo Char, parâmetro por referência Erro do tipo Boolean)

/\* Insere o parâmetro Elemento no final da fila F. O parâmetro Erro indica se a ocorreu ou não algum erro na operação. O único erro verificado é se houve tentativa de inserir um elemento em uma fila cheia \*/

```
Se Cheia ( F )
Então Erro = verdadeiro    /* não é possível inserir elemento em uma fila cheia */
Senão Erro = falso
      F.Total = F.Total + 1
      F.Itens[ F.Fim ] = Elemento
      Se F.Fim == TamanhoDoVetor
      Então F.Fim = 1
      Senão F.Fim = F.Fim + 1
```

## Simulação 5.2: Execução da Operação Insere

Veja a simulação da execução da operação insere, interagindo com a [Simulação 5.2: Execução da Operação Insere](#).

Sai (parâmetro por referência F do tipo Fila, parâmetro por referência Elemento do tipo Char, parâmetro por referência Erro do tipo Boolean)

/\* Retira o primeiro elemento da fila F, retornando o valor do primeiro elemento no parâmetro Elemento. O parâmetro Erro indica se a ocorreu ou não algum erro na operação. O único erro verificado é se houve tentativa de retirar um elemento em uma fila vazia \*/

```
Se Vazia( F )
Então Erro = verdadeiro    /* não é possível retirar um elemento de uma fila vazia */
Senão Erro = falso
      F.Total = F.Total - 1
      Elemento = F.Itens[ F.Início ]
      Se F.Início == TamanhoDoVetor
      Então F.Início = 1
      Senão F.Início = F.Início + 1
```