

Algoritmos e Programação I

Módulo 7 - Fila: Estática e Encadeada

Prof^a. Elisa de Cássia Silva Rodrigues

Fila

Definição:

- ▶ Estrutura de dados do tipo **lista**, com **restrições** para **inserção** e **remoção** de elementos, utilizada para armazenar e organizar uma sequência de elementos do mesmo tipo.
- ▶ As **inserções** ocorrem **apenas no final** e **remoções** ocorrem **apenas no início** da fila (extremidades opostas).
- ▶ Estrutura do tipo **FIFO (*First In First Out*)**, onde o primeiro elemento a entrar é o primeiro a sair.



A implementação das operações de uma fila depende do tipo de alocação de memória usada (**estática** ou **dinâmica**).

- Operações básicas:

- ▶ Criação da fila.
- ▶ Inserção de um elemento no final da fila.
- ▶ Remoção de um elemento do início da fila.
- ▶ Consulta ao primeiro elemento da fila.
- ▶ Destruição da fila.
- ▶ Informação sobre tamanho da fila.
- ▶ Informação sobre a fila estar vazia ou cheia.

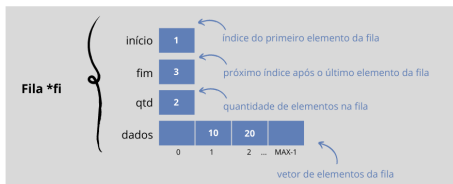
Fila Estática

Definição:

- Estrutura de dados do tipo **fila** que é definida utilizando **alocação estática** e **acesso sequencial** dos elementos.

Características:

- Definida por um **vetor** com elementos sequenciais na memória.
- Exige a definição prévia do número máximo de elementos (**MAX**).
- Campos adicionais para armazenar o **início**, o **final** e a **quantidade** de elementos na fila.



Fila Estática

- Definição do TAD Fila Estática:

- ▶ Definir os arquivos `filaEstatica.h` e `filaEstatica.c`.
- ▶ Declarar o tipo de dado que irá representar a fila no `arquivo .h`:
- ▶ Definir o tipo de dado que será armazenado dentro da fila (`int`).
- ▶ Declarar a estrutura para representar a fila estática no `arquivo .c`:

```
typedef struct fila Fila;  
  
struct fila  
{  
    int inicio;  
    int fim;  
    int qtd;  
    int dados[MAX]; // MAX representa o tamanho da fila  
};
```

- Declarar um ponteiro do tipo `Fila` para acessar o TAD (`main.c`):

```
Fila *fi;
```

Fila Estática

• Criação da fila:

- ▶ Antes de usar uma fila é preciso criar uma **fila vazia**.
- ▶ Isto é, alocar um espaço na memória para a estrutura:
 - ★ Alocação dinâmica da estrutura **Fila** usando **malloc()**.
- ▶ A fila está vazia, quando **qtd = 0**.



Note que o vetor **dados[]** que armazena os elementos da fila é alocado estaticamente durante a alocação da estrutura **Fila**.

- Destruição da fila:

- ▶ Deve-se liberar a memória alocada para a estrutura:
 - ★ Liberação da estrutura **Fila** usando **free()**.

Fila *fi = NULL

Fila Estática

- Informações básicas sobre a fila:

- ▶ Tamanho da fila (valor do campo **qtd**).
- ▶ Fila vazia (**qtd** = 0).
- ▶ Fila cheia (**qtd** = **MAX**).

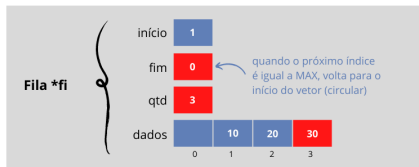
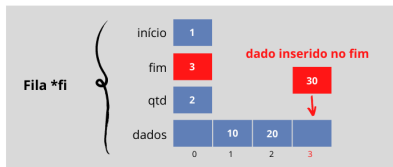


- Inserção (enfileirar):

- ▶ Ato de guardar elementos dentro da fila.
- ▶ Apenas inserção no final da fila.
- ▶ Operação de inserção envolve o teste de estouro da fila:
 - ★ Necessário verificar se é possível inserir um novo elemento na fila.
 - ★ Ou seja, se a fila não está cheia.

● Inserção no final da fila:

- ▶ Não envolve o deslocamento de elementos do vetor.
- ▶ O novo elemento é inserido logo após a última posição ocupada do vetor, ou seja, na primeira posição livre que é indicada pelo índice armazenado na variável **fim** da estrutura **Fila**.
- ▶ Ao fim da operação deve-se incrementar o campo **qtd**.



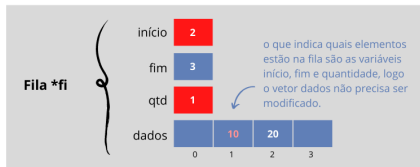
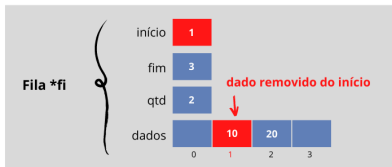
- Remoção (desenfileirar):

- ▶ Existindo uma fila, e ela possuindo elementos, é possível excluí-los.
- ▶ Apenas remoção no início da fila.
- ▶ Operação de remoção envolve o teste de fila vazia.
 - ★ Necessário verificar se existem elementos dentro da fila.
 - ★ Ou seja, se a fila não está vazia.

Fila Estática

• Remoção do início da fila:

- ▶ Não envolve o deslocamento de elementos do vetor.
- ▶ O elemento a ser removido ocupa a primeira posição da fila, ou seja, a posição indicada pelo índice armazenado na variável **início** da estrutura **Fila**.
- ▶ Ao fim da operação deve-se decrementar o campo **qtd**.



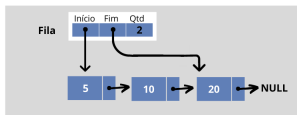
Fila Encadeada

- Definição:

- ▶ Estrutura de dados do tipo **fila** que é definida utilizando **alocação dinâmica** e **acesso encadeado** dos elementos.

- Características:

- ▶ É necessário armazenar um ponteiro para o nó descritor da fila (armazena **início**, **fim** e **quantidade**).
- ▶ Cada elemento possui um dado e um ponteiro para o próximo da fila.
- ▶ Cada elemento é alocado dinamicamente quando é inserido na fila.
- ▶ Se um elemento é removido, a memória alocada para ele é liberada.



Fila Encadeada

- Definição do TAD Fila Encadeada:

- ▶ Definir os arquivos `filaEncadeada.h` e `filaEncadeada.c`.
- ▶ Declarar o tipo de dado que irá representar a fila no `arquivo .h`:
- ▶ Definir o tipo de dado que será armazenado dentro da fila (`int`).
- ▶ Declarar a estrutura para representar a fila encadeada no `arquivo .c`:

```
typedef struct descritor Fila;  
  
struct descritor  
{  
    struct elemento * inicio;  
    struct elemento * fim;  
    int qtd;  
};
```

- Definição do TAD Fila Encadeada:

- ▶ Declarar a estrutura para representar o elemento no arquivo `.c`:

```
struct elemento{  
    int dado;  
    struct elemento * prox;  
};  
  
typedef struct elemento Elemento;
```

- Declarar um ponteiro do tipo Fila para acessar o TAD (`main.c`):

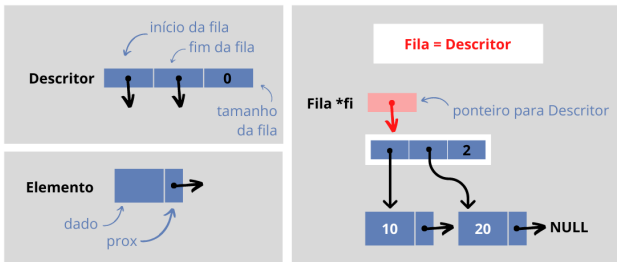
```
Fila *fi;
```

Fila Encadeada

- Ilustração dos tipos de dados **Fila** e **Elemento**:

```
typedef struct elemento Elemento;
```

```
typedef struct descriptor Fila;
```

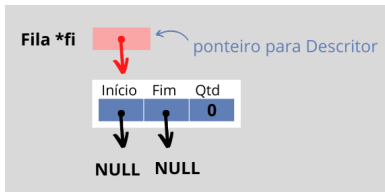


Note que, nesta implementação, a estrutura **Fila** é definida pela **struct descriptor** (nó descriptor).

Fila Encadeada

- Criação da fila:

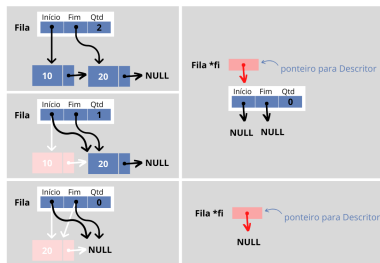
- ▶ Antes de usar uma fila é preciso criar uma **fila vazia**.
- ▶ Isto é, alocar um espaço na memória para o descritor da fila:
 - ★ Alocação dinâmica de um ponteiro do tipo **Fila** usando **malloc()**.
- ▶ A fila está vazia, quando **fi != NULL** e **fi->qtd == 0**.



Fila Encadeada

• Destruição da fila:

- ▶ Inicialmente, deve-se liberar a memória alocada para todos os elementos da fila:
 - ★ Liberação da estrutura **Elemento** usando **free()**.
- ▶ Deve-se liberar a memória alocada para o nó descritor da fila:
 - ★ Liberação do ponteiro do tipo **Fila** usando **free()**.



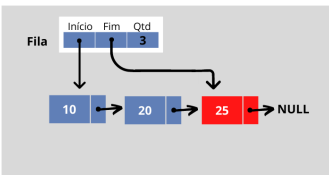
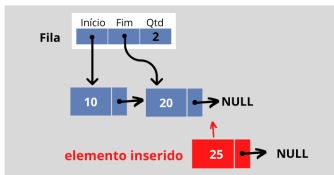
- Inserção (enfileirar):

- ▶ Ato de guardar elementos dentro da fila.
- ▶ Apenas inserção no final da fila.
- ▶ Operação de inserção envolve alocação dinâmica de memória:
 - ★ Necessário verificar se a fila existe (`fi != NULL`).
 - ★ Se existir, deve-se verificar se o novo elemento foi alocado corretamente.

Fila Encadeada

● Inserção no final da fila:

- ▶ Envolve a criação de um novo elemento (alocação de memória).
- ▶ Atribui-se o valor do novo elemento ao campo **dado**.
- ▶ O ponteiro **prox** do novo elemento aponta para **NULL**.
- ▶ O ponteiro **prox** do elemento **fi->fim** aponta para o novo elemento.
- ▶ Se a fila estiver vazia, o ponteiro **fi->inicio** aponta para o novo elemento.
- ▶ O ponteiro **fi->fim** aponta para o novo elemento.



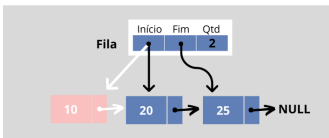
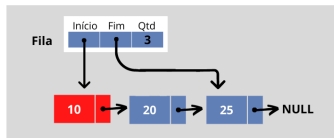
- Remoção (desenfileirar):

- ▶ Existindo uma fila, e ela possuindo elementos, é possível excluí-los.
- ▶ Apenas remoção no início da fila.
- ▶ Operação de remoção envolve o teste de fila vazia.
 - ★ Necessário verificar se a fila existe (`fi != NULL`).
 - ★ Se existir, deve-se verificar se existem elementos dentro da fila.
 - ★ Ou seja, se a fila não está vazia (`fi->qtd != 0`).

Fila Encadeada

• Remoção do início da fila:

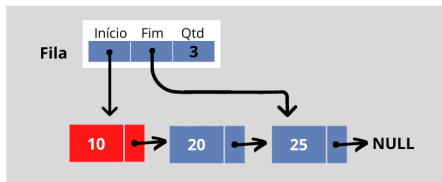
- ▶ Envolve a liberação da memória alocada para o 1º elemento da fila.
- ▶ O ponteiro que indica o 1º elemento (**fi->inicio**) aponta para o 2º elemento.
- ▶ Por fim, libera-se a memória do 1º elemento usando a função **free()**.
- ▶ Se a fila ficar vazia, então **fi->fim = fi->inicio = NULL**.



Fila Encadeada

- Consulta:

- ▶ A fila permite acesso apenas ao primeiro elemento (**fi->inicio**).



Fila Estática x Fila Encadeada

Características	Fila Estática	Fila Encadeada
Vantagem	Facilidade de criar e destruir a lista	Melhor uso da memória
	Complexidade dos algoritmos não depende do tamanho da fila: $O(1)$	Não precisa definir tamanho máximo
Desvantagem	Necessidade de definir o tamanho máximo	Necessidade de percorrer a fila toda para destruí-la.
Utilização	Filas pequenas	Quando o tamanho da fila não é definido
	Quando o tamanho pode ser definido	

- Implementação TAD Fila Encadeada:

https://repl.it/@elisa_rodrigues/Modulo7-FilaEncadeada

- Exercício para fixação:

Implemente a TAD Fila Estática.

① BACKES, A. *Estrutura de dados descomplicada em linguagem C*. 2016.

-> **Capítulo 6: Filas**

-> **Material Complementar - Vídeo aulas (31ª a 37ª):**

<https://programacaodescomplicada.wordpress.com/indice/estrutura-de-dados/>