

Estrutura de Dados I

Filas

Prof. Dr. Marcelo Otone Aguiar

Universidade Federal do Espírito Santo - UFES

26 de janeiro de 2026

Conteúdo

- Filas
- Fila estática
- Fila dinâmica

Filas

- Estrutura de dados linear **restrita**.
- Baseada na política **FIFO**.
- Muito usada em programação e sistemas.

FIFO = *First In, First Out*

Equivale à = Primeiro a entrar, primeiro a sair

Filas

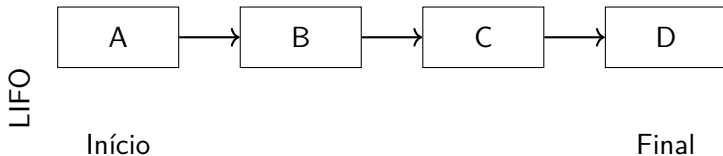
Uma Fila é uma estrutura de dados em que:

- A inserção ocorre **somente no final**
- A remoção ocorre **somente no início**
- **Não é permitido** acesso direto a elementos intermediários

Analogia:

- Fila de banco
- Fila de impressão
- Fila de processos

Filas



Operações Básicas da Fila

- **Enqueue:** insere no final da fila
- **Dequeue:** remove do início da fila
- **Front (ou Peek):** acessa o primeiro elemento

Exemplos:

- `Enqueue(f, x)` //Insere x no final
- `Dequeue(f)` //Remove o início
- `Front(f)` //Retorna o início sem remover o elemento

Implementações de Filas

Uma Fila pode ser implementada de duas formas:

Estática:

- Uso de vetor
- Tamanho fixo
- Pode causar desperdício ou *overflow*

Dinâmica:

- Lista encadeada
- Tamanho variável
- Uso mais eficiente da memória

Fila Estática - Implementação

- **FilaEstatica.h:** definir
 - Os protótipos das funções
 - O tipo de dado armazenado na Fila
 - O ponteiro **Fila**
 - Tamanho do vetor usado na Fila
- **FilaEstatica.c:** definir
 - O tipo de dados **Fila**
 - Implementar as suas funções.

Código em C - Estrutura da Fila

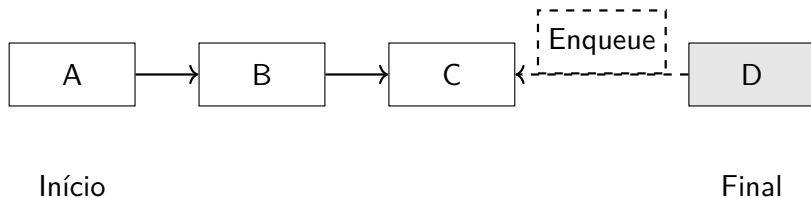
```
1 //Arquivo FilaEstatica.h
2 #define MAX 100
3 struct aluno {
4     int matricula;
5     char nome[30];
6     float n1, n2, n3;
7 };
8 typedef struct fila Fila;
9
10 //Arquivo FilaEstatica.c
11 struct fila {
12     int inicio, final, qtd;
13     struct aluno dados[MAX];
14 };
15
16 //Programa principal
17 Fila *fi;
```

Fila Estática - Implementação

As funções listadas a seguir para uma **Fila** podem ser implementadas utilizando as funções que já conhecemos de **lista estática**:

- Criação da Fila
- Liberar a Fila da memória
- Obter o tamanho da Fila
- Verificar se a Fila está cheia
- Verificar se a Fila está vazia

Inserção no final (*Enqueue*)



Inserção no final (*Enqueue*)

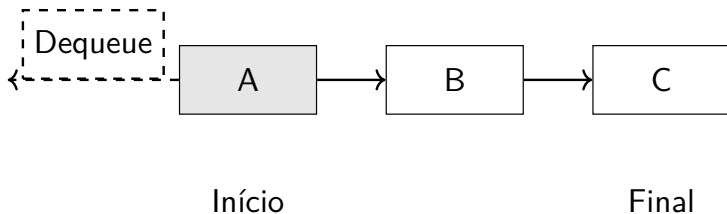
- Em **Filas estáticas** a inserção é sempre no *final*
- Inserir no **final da Fila**, equivale à função de **inserir no final** da lista sequencial estática
- Também existe o caso em que a inserção é feita em uma Fila vazia
- Não se pode inserir em uma Fila cheia.

Código em C - Inserção no final (*Enqueue*)

```
1 //Arquivo FilaEstatica.c
2 int enqueue(Fila* fi, struct aluno al){
3     if (fi == NULL)
4         return 0;
5     if (fila_cheia(fi))
6         return 0;
7     fi->dados[fi->final] = al;
8     fi->final = (fi->final + 1) % MAX;
9     fi->qtd++;
10    return 1;
11 }
```

```
1 //Programa principal
2 int x = enqueue(li, &dados_aluno);
```

Remoção no início (*Dequeue*)



Remoção no início (*Dequeue*)

- Em **Filas estáticas** a remoção é sempre no *início*
- Remover no **início da Fila**, equivale à função de **remover no início** da lista sequencial estática
- Não se pode remover em uma Fila vazia.

Código em C - Remoção no início (*Dequeue*)

```
1 int dequeue(Fila* fi){  
2     if (fi == NULL || fila_vazia(fi))  
3         return 0;  
4     fi->inicio = (fi->inicio + 1) % MAX;  
5     fi->qtd--;  
6     return 1;  
7 }
```

```
1 //Programa principal  
2 int x = dequeue( fi );
```


Consulta no início (*Front*)

- Em **Filas estáticas** a consulta é sempre no *início*
- Consultar no **início da Fila**, equivale à consultar o primeiro elemento da lista sequencial estática.

Código em C - Consulta no início (*Front*)

```
1 int front(Fila* fi, struct aluno al){  
2     if (fi == NULL || fila_vazia(fi))  
3         return 0;  
4     *al = fi->dados[fi->inicio];  
5     return 1;  
6 }
```

```
1 //Programa principal  
2 int x = front(fi, &dados_aluno);
```

Fila Dinâmica - Implementação

- **FilaDin.h:** definir
 - Os protótipos das funções
 - O tipo de dado armazenado na Fila
 - O ponteiro **Fila**
- **FilaDin.c:** definir
 - O tipo de dados **Fila**
 - Implementar as suas funções.

Código em C - Estrutura da Fila

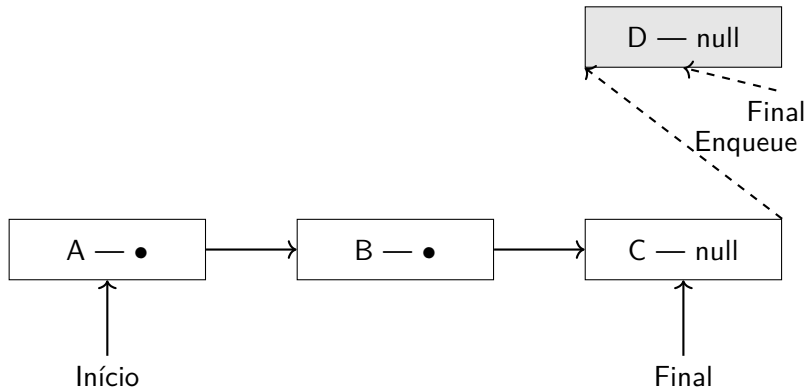
```
1 //Arquivo FilaDin.h
2 struct aluno {
3     int matricula;
4     char nome[30];
5     float n1, n2, n3;
6 };
7 typedef struct fila Fila;
8 //Arquivo FilaDin.c
9 struct fila { //estrutura que controla a fila
10     struct elemento *inicio;
11     struct elemento *final;
12 };
13 struct elemento {
14     struct aluno dados;
15     struct elemento *prox;
16 };
17 typedef struct elemento Elem;
```

Fila Dinâmica - Implementação

As funções listadas a seguir para uma **Fila** podem ser implementadas utilizando as funções que já conhecemos de **lista simplesmente encadeada**:

- Criação da Fila
- Liberar a Fila da memória
- Obter o tamanho da Fila
- Verificar se a Fila está vazia

Inserção no final (*Enqueue*)



Inserção no final (*Enqueue*)

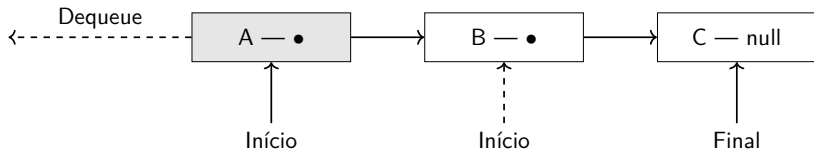
- Em **Filas dinâmicas** a inserção é sempre no *final*
- Inserir no **final da Fila**, equivale à função de **inserir no final** da lista simplesmente encadeada
- Também existe o caso em que a inserção é feita em uma Fila vazia.

Código em C - Inserção no final (*Enqueue*)

```
1 int enqueue(Fila* fi, struct aluno al){
2     if (fi == NULL) return 0;
3     Elem* no = (Elem*) malloc(sizeof(Elem));
4     if (no == NULL) return 0;
5     no->dados = al;
6     no->prox = NULL;
7     if (fi->final == NULL) //fila vazia
8         fi->inicio = no;
9     else
10         fi->final->prox = no;
11     fi->final = no;
12     return 1;
13 }
```

```
1 //Programa principal
2 int x = enqueue(fi, dados_aluno);
```


Remoção no início (*Dequeue*)



Remoção no início (*Dequeue*)

- Em **Filas dinâmicas** a remoção é sempre no *início*
- Remover no **início da Fila**, equivale à função de **remover no início** da lista simplesmente encadeada
- Não se pode remover em uma Fila vazia.

Código em C - Remoção no início (*Dequeue*)

```
1 int dequeue(Fila* fi){
2     if (fi == NULL)
3         return 0;
4     if (fi->inicio == NULL) //fila vazia
5         return 0;
6     Elem *no = fi->inicio;
7     fi->inicio = fi->inicio->prox;
8     if (fi->inicio == NULL) //fila ficou vazia
9         fi->final = NULL;
10    free(no);
11    return 1;
12 }
```

```
1 //Programa principal
2 int x = dequeue(fi);
```

Consulta no início (*Front*)

- Em **Filas dinâmicas** a consulta é sempre no *início*
- Consultar no **início da Fila**, equivale à consultar o primeiro elemento da lista simplesmente encadeada.

Código em C - Consulta no início (*Front*)

```
1 int front(Fila* fi, struct aluno al){
2     if (fi == NULL)
3         return 0;
4     if (fi->inicio == NULL) //fila vazia
5         return 0;
6     *al = fi->inicio->dados;
7     return 1;
8 }
```

```
1 //Programa principal
2 int x = front(fi, &dados_aluno);
```