

Aula 12: Fila (Queue)



DCC302-Estrutura de Dados I
Prof. Me. Acauan C. Ribeiro

Roteiro – Estrutura de Dados Fila

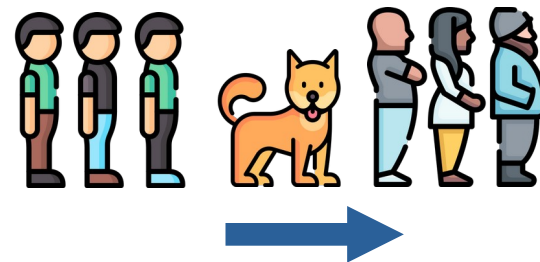
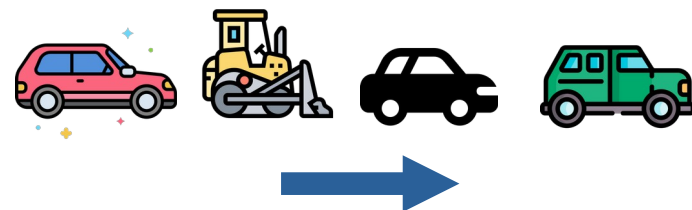
- O que é uma Fila?
- Funcionamento da Fila
- Operações
- Aplicações



O que é a estrutura de dados Fila?

Definição: Fila (Queue) é uma **estrutura de dados linear**, que opera seguindo o critério de Primeiro a Entrar é o Primeiro a Sair (**FIFO**), **First In First Out**.

- Seu nome como fica evidente faz referencia ao comportamento que esta palavra representa no mundo real. Como uma fila de carros em uma faixa, uma fila de pessoas esperando para realizar o pedido no caixa, etc;
- A Fila é um **Tipo Abstrato de Dados** que tem um limite (predefinido) de capacidade;
- É uma estrutura de dados simples que permite adicionar e remover elementos seguindo uma ordem específica (particular);
- Essa ordem é a **FIFO** (Primeiro a Entrar é o Primeiro a Sair);

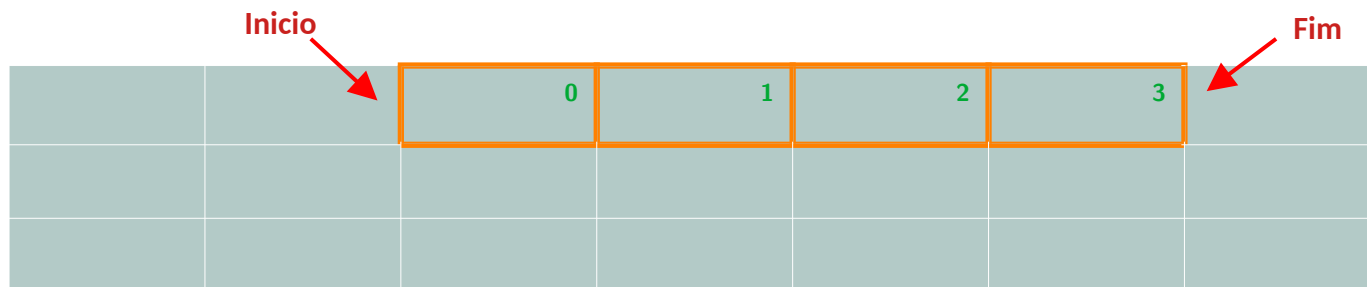


Funcionamento da Fila

Fila (Queue) é uma **estrutura de dados linear**, que opera seguindo o critério de **Primeiro a Entrar é o Primeiro a Sair (FIFO)**, **First In First Out**.

→ Elementos são **adicionados no final** da fila, cauda (**Fim**). Essa ação é chamada de **Enfileirar (Enqueue)**;

→ Elementos são **removidos do início** da fila, ou seja da cabeça da estrutura (**Início**). Essa ação é chamada de **Desenfileirar (Dequeue)**



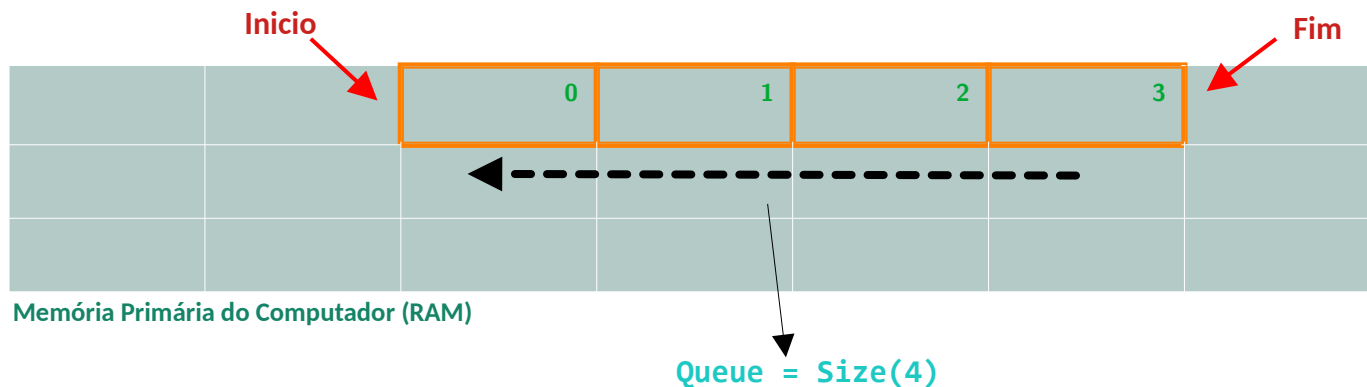
Memória Primária do Computador (RAM)

Funcionamento da Fila

Fila (Queue) é uma **estrutura de dados linear**, que opera seguindo o critério de Primeiro a Entrar é o Primeiro a Sair (**FIFO**), **First In First Out**.

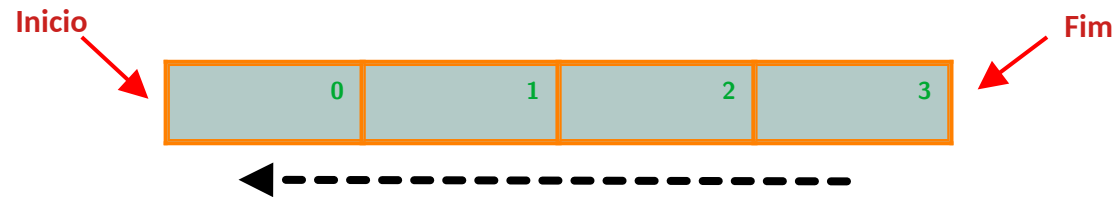
→ Elementos são **adicionados no final** da fila, cauda (**Fim**). Essa ação é chamada de **Enfileirar (Enqueue)**;

→ Elementos são **removidos do início** da fila, ou seja da cabeça da estrutura (**Início**). Essa ação é chamada de **Desenfileirar (Dequeue)**



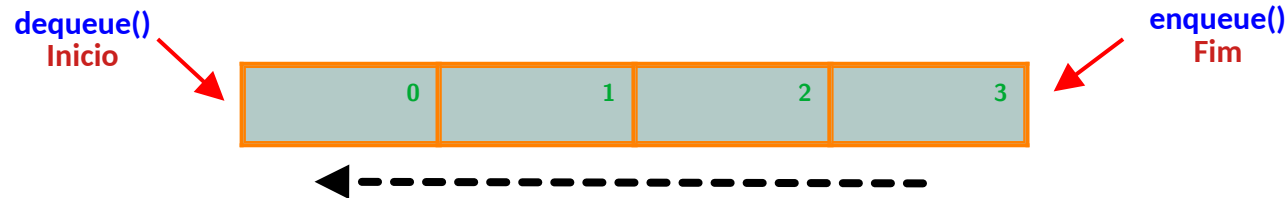
Operações Padrões - Fila

- 1) enqueue() - Elementos são adicionados no final da fila;
- 2) dequeue() - Elementos são removidos do início (cabeça) da fila;
- 3) isFull() - Verifica se a Fila está cheia;
- 4) isEmpty() - Verifica se a Fila está vazia;
- 5) count() - Retorna o n° total de elementos existentes na Fila naquele momento;



Operações Padrões - Fila

- 1) `enqueue()` - Elementos são adicionados no final da fila;
- 2) `dequeue()` - Elementos são removidos do início (cabeça) da fila;
- 3) `isFull()` - Verifica se a Fila está cheia;
- 4) `isEmpty()` - Verifica se a Fila está vazia;
- 5) `count()` - Retorna o n° total de elementos existentes na Fila naquele momento;

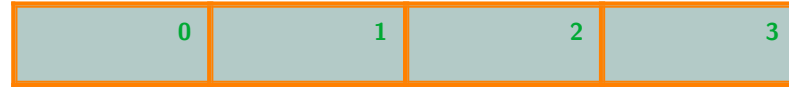


Inicialização

```
int arr[4];
```

```
int fim = -1;
```

```
int incio = -1;
```



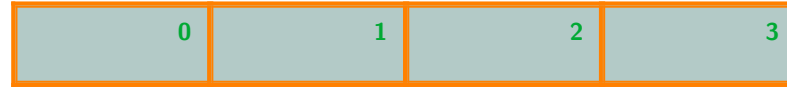
--	--	--

Inicialização

```
int arr[4];
```

```
int fim = -1;
```

```
int inicio = -1;
```



```
isEmpty() {  
    if(inicio == -1 && fim == -1)  
        return true;  
    else  
        return false;  
}
```

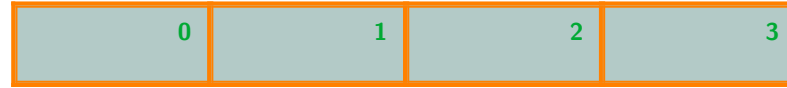
```
isFull() {  
    if(fim == size(arr)-1)  
        return true;  
    else  
        return false;  
}
```

Inicialização

```
int arr[4];
```

```
int fim = -1;
```

```
int inicio = -1;
```



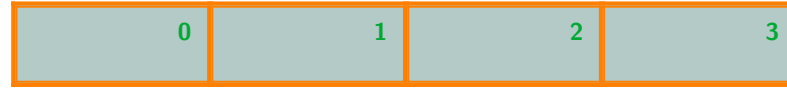
```
isEmpty() {  
    if(inicio == -1 && fim == -1)  
        return true;  
    else  
        return false;  
}
```

```
isFull() {  
    if(fim == size(arr)-1)  
        return true;  
    else  
        return false;  
}
```

```
enqueue() {  
    if(isFull()) return;  
    else if(isEmpty()) {  
        fim = inicio = 0;  
    }  
    else {  
        fim++;  
    }  
    arr[fim] = value;  
}
```

Inicialização

```
int arr[4];  
int fim = -1;  
int inicio = -1;
```



```
isEmpty() {  
    if(inicio == -1 && fim == -1)  
        return true;  
    else  
        return false;  
}  
  
isFull() {  
    if(fim == size(arr)-1)  
        return true;  
    else  
        return false;  
}
```

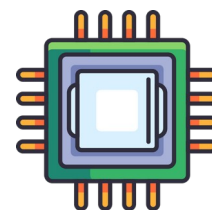
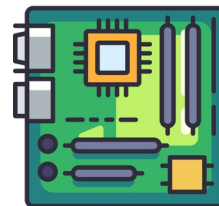
```
enqueue() {  
    if(isFull()) return;  
    else if(isEmpty()) {  
        fim = inicio = 0;  
    }  
    else {  
        fim++;  
    }  
    arr[fim] = value;  
}
```

```
dequeue() {  
    int x = 0;  
    if(isEmpty()) return;  
    else if(inicio == fim) {  
        x = arr[inicio];  
        inicio = fim = -1;  
    }  
    else {  
        x = arr[inicio];  
        inicio++;  
    }  
    return x;  
}
```

Algumas aplicações – Fila

Fila (Queue) é usada quando coisas precisam ser processadas seguindo a ordem de **First In First Out**, como por exemplo:

- Escalonamento de tarefas da **CPU** e **DISCO**;
- Resolver as interrupções de **sistemas em tempo real**. Essas interrupções são tratadas na ordem que elas chegam, “first come first served”;
- Na vida real, Sistemas de **Call Centers** utilizam Queues, para atender as ligações na ordem que elas estão na espera.
- Quando dados são transferidos de maneira **assíncrona** entre dois processos, um Fila é utilizada para sincronização.



Implementação

Referências

- Queue Data Structure in C++ Programming - <https://simplesnippets.tech/>