

Alessandro Calin
Fila

ESTRUTURA DE DADOS
alessandrocalin@gmail.com
981524482



FILA → É uma estrutura de dados que inicialmente foi feita de forma sequencial (vetor). Hoje temos lista encadeada.

Segue o critério FIFO (First in First out) (Primeiro a entrar é o primeiro a sair).

Exemplo: Fila de impressão



- **Processamento de tarefas em um sistema operacional (gerenciamento de processos):** Fila é usada para agendar processos ou tarefas. Por exemplo, o escalonador de processos utiliza uma fila para gerenciar quais processos devem ser executados em qual ordem.
- **Impressoras:** Quando várias pessoas enviam documentos para uma impressora, eles são colocados em uma fila de impressão, sendo processados na ordem em que chegaram.
- **Buffer de dados:** Em comunicação de rede, filas são usadas para armazenar pacotes de dados que chegam mais rápido do que podem ser processados. Um buffer de recepção age como uma fila, permitindo que os pacotes sejam processados conforme a capacidade.



- **Atendimento ao cliente:** Fila é usada em sistemas de call center, onde as chamadas dos clientes são enfileiradas e atendidas conforme a ordem de chegada.
- **Transmissão de mensagens em sistemas distribuídos:** Filas são usadas em sistemas de mensageria, como RabbitMQ ou Apache Kafka, onde as mensagens enviadas entre componentes são colocadas em fila para serem processadas sequencialmente.
- **Simulação de filas em teoria das filas:** Na modelagem de sistemas de espera, como filas em bancos ou supermercados, usa-se a estrutura de dados fila para simular o comportamento de pessoas ou objetos que esperam pelo serviço.



Como implementar uma pilha?

Pelo menos 3 funções:

- Função de enfileira

- Função de desenfileira

- Função principal (main)

Passo a passo para implementação da Fila:

“Foi colocado no caderno para os alunos copiarem”



Passo a passo para implementar uma fila.

FIFO - PRIMEIRO A ENTRAR, PRIMEIRO A SAIR.

EXEMPLOS:

Gerenciamento de fila de impressão



Passo a passo para implementar uma fila.

Toda fila é formada por três variáveis:

Um vetor

Uma variável como nome de frente (F) → desenfileirar

Uma variável com o nome de retaguarda (R) → enfileirar



Passo a passo para implementar uma pilha.

Todo Fila é uma Struct, pois possui três variáveis que podem ser de tipos diferentes.



Passo a passo para implementar uma fila.

- 1) Importar as Bibliotecas;
- 2) Definir a Struct;
- 3) Criar a função enfileirar;
 - 3.1) Definir a função enfileirar;
 - 3.2) Na função enfileirar temos que passar como parâmetro o valor a ser enfileirado e um ponteiro para manipular a Struct;
 - 3.3) Verificar se a fila esta cheia. A fila está cheia quando o R (retaguarda) chegar ao limite (tamanho do vetor)
 - 3.4) Incrementar a variável R
 - 3.5) Armazenar o valor dentro da fila.



Passo a passo para implementar uma desenfileirar.

4) Criar a função desenfileirar

4.1) É retornável e possui como parâmetro apenas um ponteiro;

4.2) Declarar uma variável local para retirar o valor da fila e retornar pra main;

4.3) Verificar se a fila está vazia. A fila está vazia quando a variável F (frente) chegar ao limite;

4.4) Desenfileirar o valor do vetor, jogando para a variável auxiliar (local);

4.5) Incrementar a variável F;

4.6) Retornar o valor pra main.



Passo a passo para implementar a main.

5) Criar a função main

5.1) Declarar a Struct na main

5.2) Declarar uma variável para receber o valor digitado pelo usuário (a ser enfileirado)

5.3) Iniciar o R (retaguarda) com -1

5.4) Iniciar o F (frente) com 0

5.5) Chamar a função enfileira o número de vezes do tamanho do vetor (terá que ficar dentro de uma estrutura de repetição)

5.4) Fazer uma estrutura de repetição para desenfileirar o valor



```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
#include<conio.h>
```

```
#define TAM 10
```

```
typedef struct{
```

```
    int R, F;
```

```
    int vet[TAM];
```

```
}filavet;
```

```
void enfileirar(int valor, filavet *p){  
    if (p -> R == TAM-1){  
        printf("\n Fila Cheia!");  
        exit(0);  
    }  
    p -> R++;  
    p -> vet[p -> R] = valor;  
}
```

```
int desenfileirar (filavet *p){  
    int aux;  
    if (p -> F == TAM){  
        printf("\nFila Vazia!");  
        exit(1);  
    }  
    aux = p -> vet[p -> F];  
    p -> F++;  
    return aux;  
}
```

```
int main(void){  
    filavet fila;  
    int valor;  
    fila.R = -1;  
    fila.F = 0;  
    for(int i=0; i< TAM; i++){  
        printf("Digite o valor a ser enfileirado: \n");  
        scanf("%d", &valor);  
        enfileirar(valor, &fila);  
    }  
    for(int i=0; i < TAM; i++){  
        printf("\n%d", desenfileirar(&fila));  
    }  
    return(0);  
}
```