

# ESTRUTURA DE DADOS FILA

**Prof. Joaquim Uchôa**  
**Profa. Juliana Gregghi**  
**Prof. Renato Ramos**



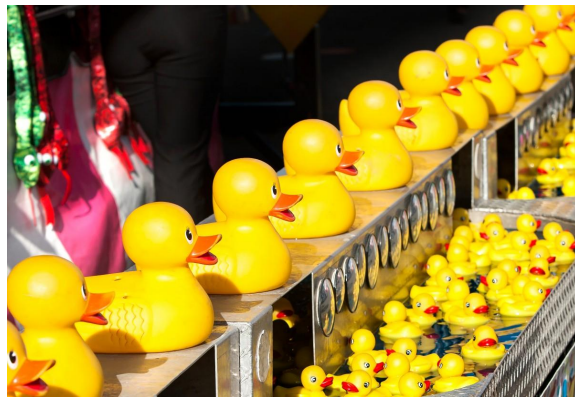
- Definição e Visão Geral
- Operações Básicas sobre Filas
- Informação Importante
- Implementação Utilizando Vetores

# DEFINIÇÃO E VISÃO GERAL

A word cloud of various terms in different shades of blue, arranged in a roughly triangular shape. The words include: outra, Conjunto, coisas, diferentes, soldados, alinhadas, etc, altura, geralmente, Fila, Sequência, fileira, organizada, fins, chegada, cronológica, criterios, diversos, ordem, idade.

# DEFINIÇÃO

- De acordo com o dicionário  
o que é uma fila?



# DEFINIÇÃO

- De acordo com o dicionário o que é uma fila?<sup>1</sup>
  - Sequência de pessoas ou coisas alinhadas uma atrás da outra, organizada geralmente por ordem cronológica de chegada ou por diferentes critérios (de altura, de idade etc.) e para diversos fins
  - Conjunto de soldados em fileira

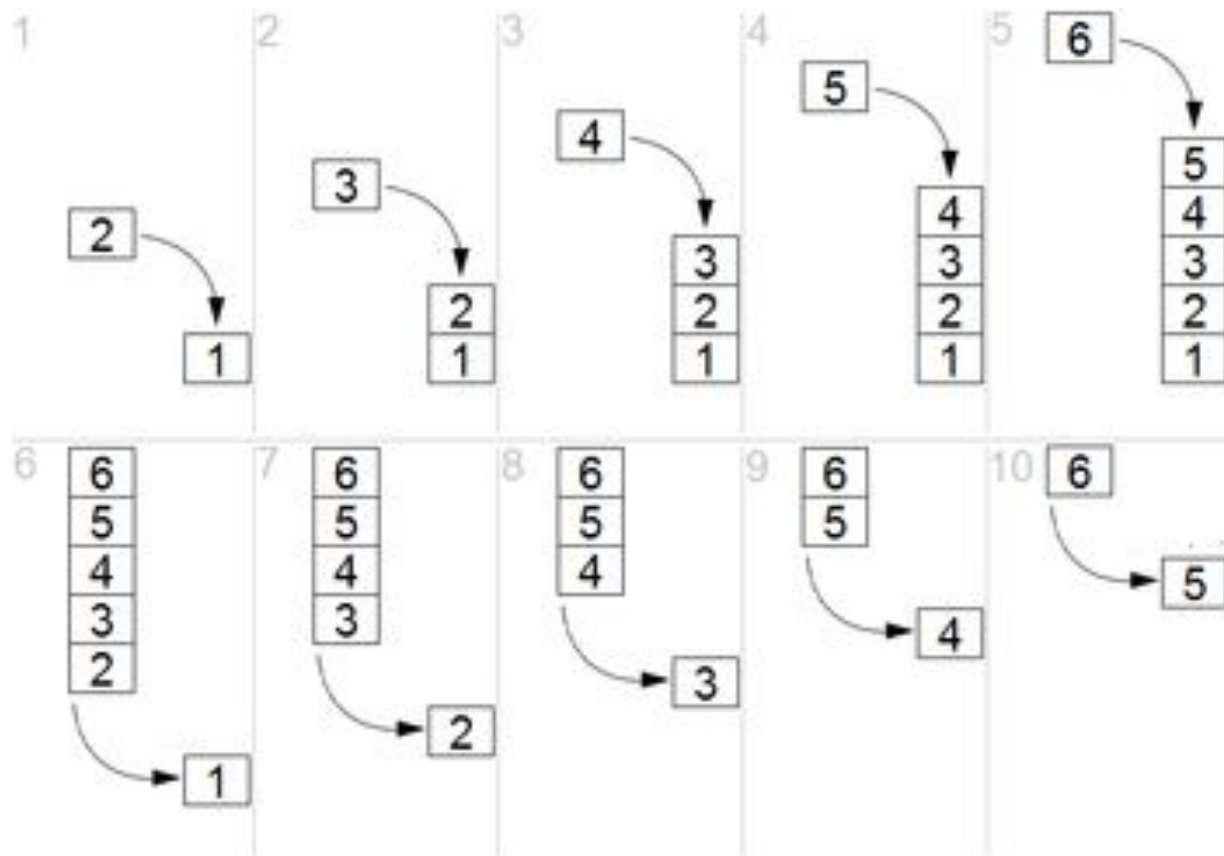
<sup>1</sup>Segundo dicionário michaelis on-line <http://michaelis.uol.com.br/busca?id=OWQE> 5

# DEFINIÇÃO

- E em Computação?
  - É uma estrutura de dados que assume a ideia de sequência de objetos para armazenar as informações.
  - É baseada no princípio "*First In First Out*" (FIFO)
    - O primeiro inserido será o primeiro a ser retirado.
    - **A manipulação é feita pelas extremidades por função pré-definida. Por uma se insere e na outra se retira.**
    - **Não é possível o acesso direto aos demais dados.**

# DEFINIÇÃO

## Ideia Básica



# APLICAÇÕES TRADICIONAIS DE FILAS

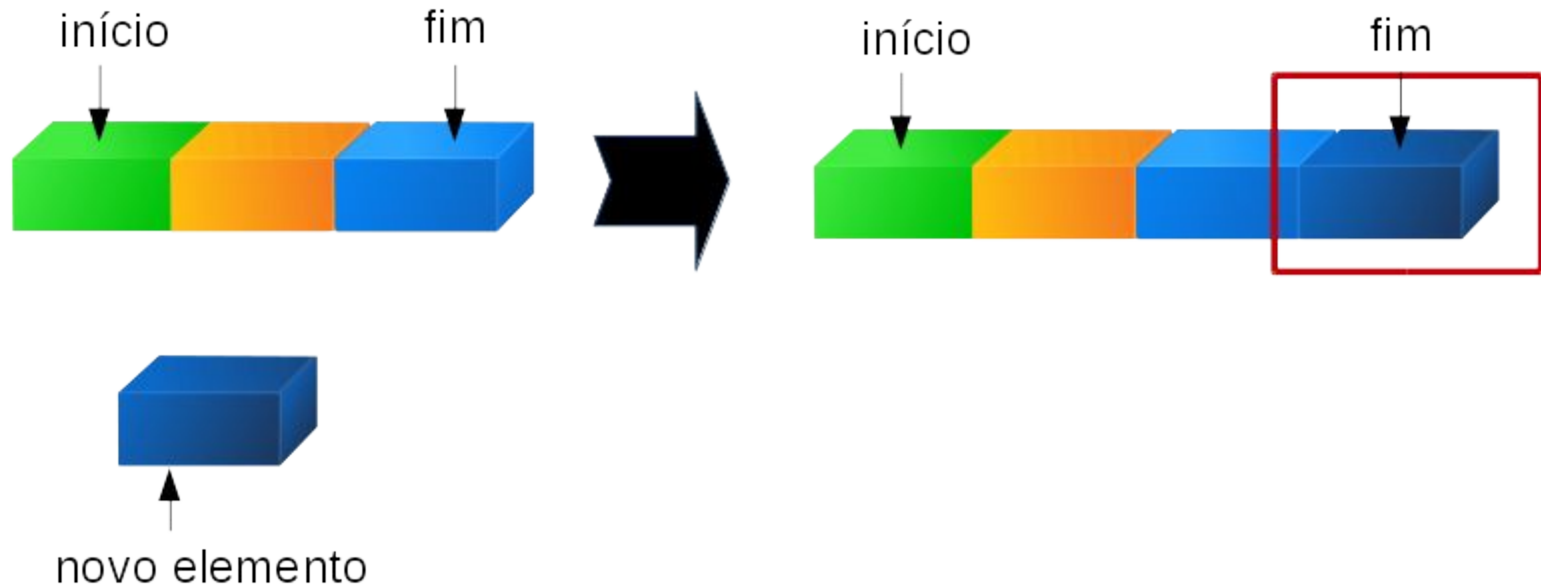
Filas são utilizadas em diversas situações na computação. Por exemplo, são utilizadas para implementação de buffers e nas seguintes aplicações:

- Escalonamento de processos na CPU;
- Escalonamento de acesso à memória secundária;
- Sincronização de processos assíncronos;
- Gerenciamento de fila de impressão;
- Controle de interrupções de hardware;
- Controle de chamadas em call centers;
- Busca em largura em grafos.



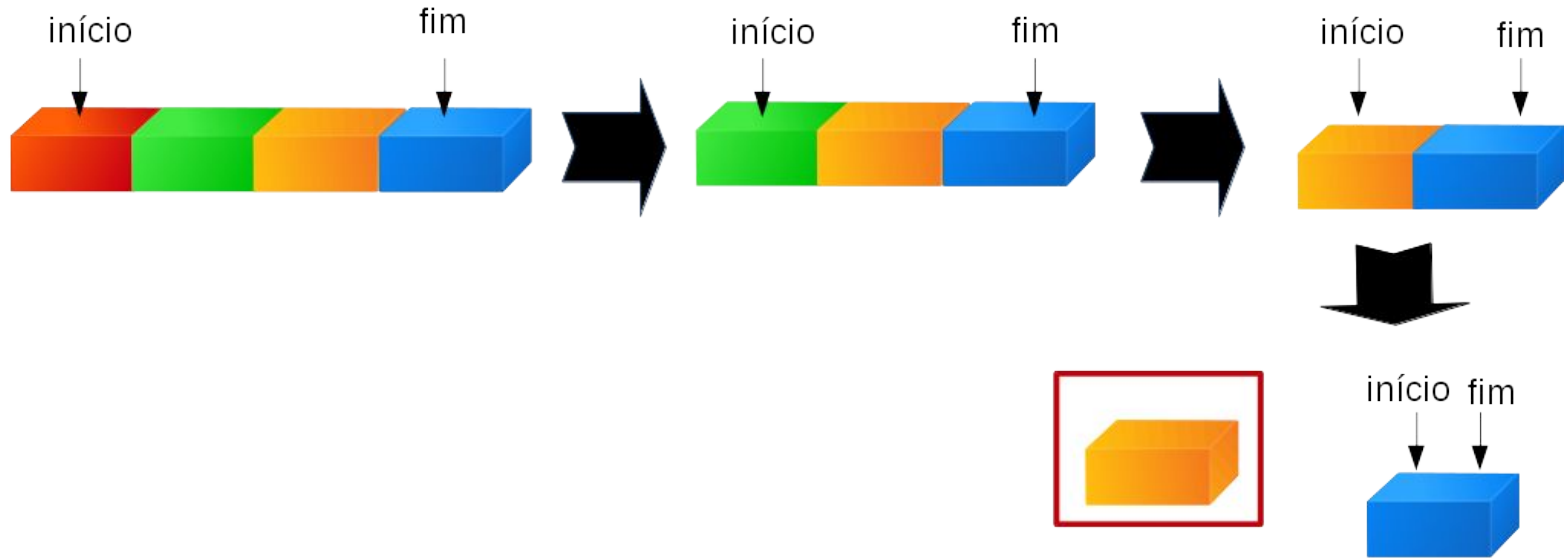
# INSERÇÃO NA FILA

A inserção deve ser sempre após o último elemento inserido.



# RETIRADA DE ELEMENTOS NA FILA

A remoção deve ser sempre daquele que foi inserido primeiro. Caso se deseje acessar, por exemplo, o conteúdo do nó laranja, todos os demais devem ser desenfileirados antes.



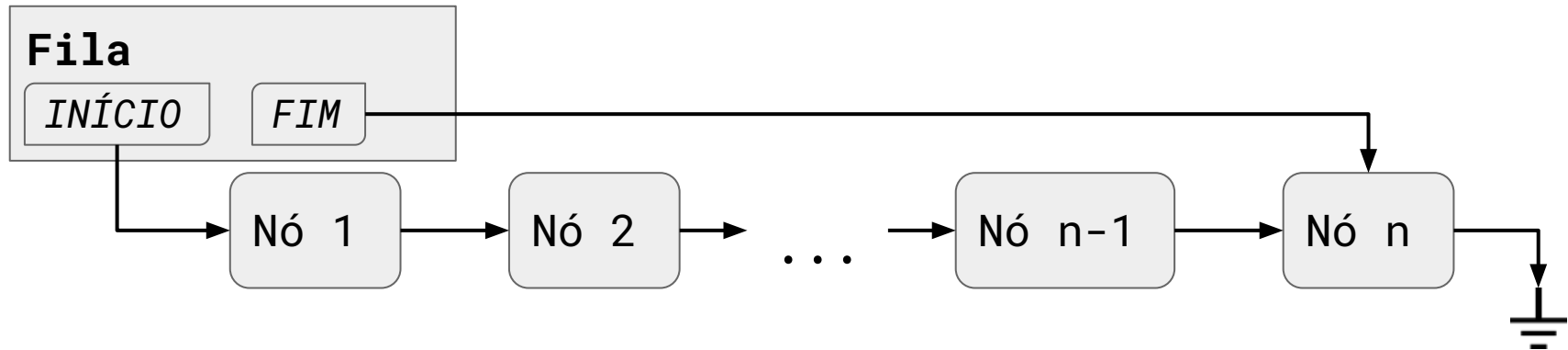
# FORMAS DE IMPLEMENTAÇÃO

Uma fila pode ser implementada de diferentes maneiras, as mais comuns são:

- Implementação usando encadeamento de nós (elementos) da fila;
- Implementação usando arranjo para armazenamento dos elementos.

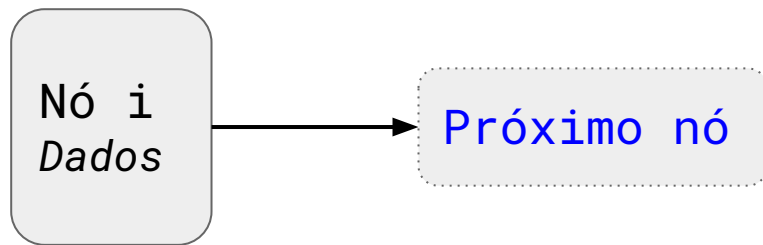
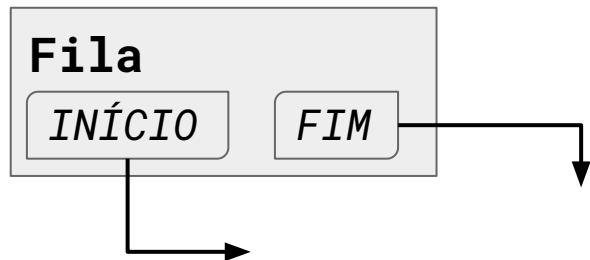
# IMPLEMENTAÇÃO USANDO ENCADEAMENTO - I

A implementação mais usual de fila é a usando encadeamento. Cada nó contém informações e aponta para o próximo nó. O último nó da fila aponta para NULO, indicando seu término. A fila sabe apenas onde está o início e o fim.



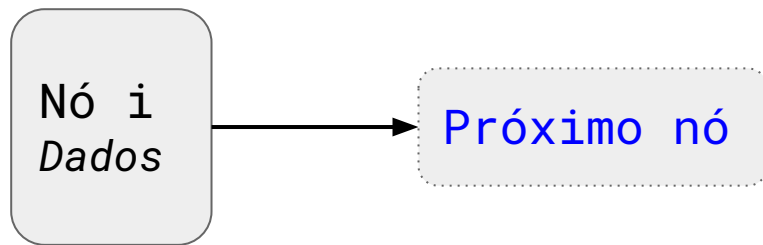
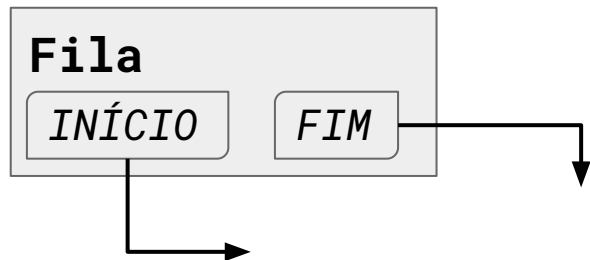
# IMPLEMENTAÇÃO USANDO ENCADEAMENTO - II

Em uma implementação por encadeamento, uma fila contém um apontador para o início e outro para o fim. Cada nó contém o dado armazenado e um apontador para o próximo nó na estrutura.



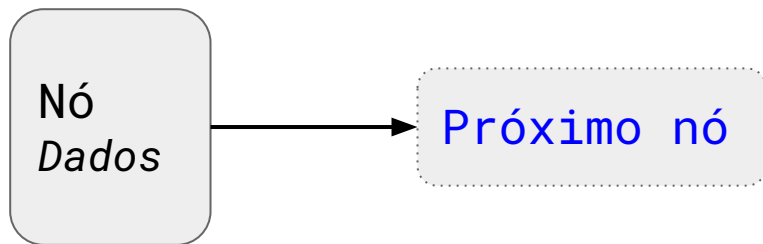
# IMPLEMENTAÇÃO USANDO ENCADEAMENTO - III

A fila e seus nós são implementados tradicionalmente como estruturas ou classes. Usaremos a segunda abordagem, usando uma classe para fila e outra para os nós.



# CLASSE NOH

Cada nó da fila possui atributos com os dados armazenados, que pode ser desde um simples inteiro até uma estrutura ou outra classe. Além disso, possui um apontador para o próximo nó.

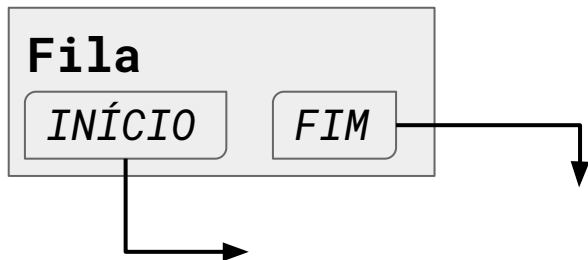


**class**  
**noh**

```
int dado;  
noh* proximo;
```

# CLASSE FILA

A classe fila contém geralmente, além dos apontadores para seu início e fim, um atributo para informar a quantidade de elementos (nós) que ela possui. Outros atributos podem ser necessários, dependendo do problema resolvido.



**class**  
**fila**

```
int tamanho;  
noh* inicio;  
noh* fim;
```



# OPERAÇÕES BÁSICAS SOBRE FILAS



# OPERAÇÕES BÁSICAS SOBRE FILAS

Uma fila possui um conjunto de operações básicas, para seu correto uso:

- Criação / destruição
- Inserção (ou enfileiramento)
- Remoção (ou desenfileiramento)

# criação da fila

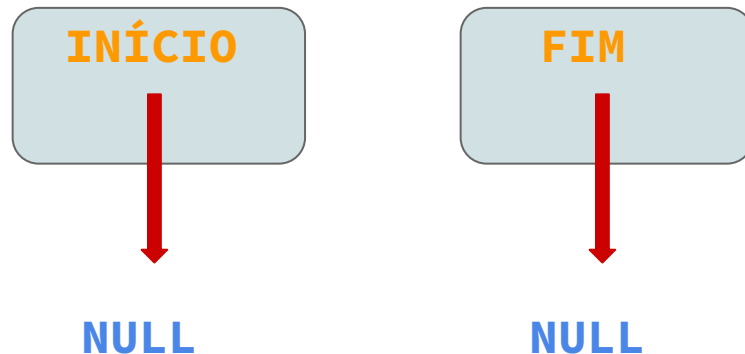
*criarFila():*

```
inicio ← NULL;
```

```
fim ← NULL;
```

```
// informação opcional
```

```
tamanho ← 0;
```



A criação da fila consiste basicamente em garantir que seus apontadores não referenciem uma região válida de memória.

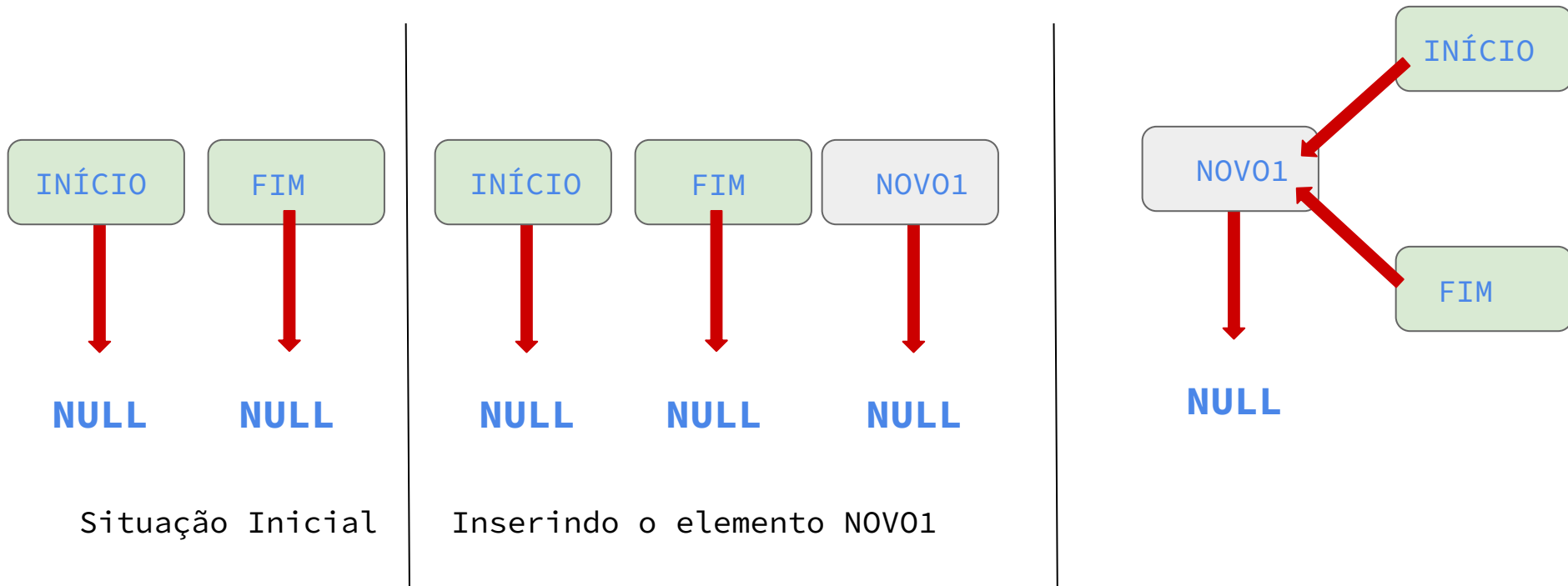
# INSERÇÃO DE ELEMENTO

## enqueue(valor):

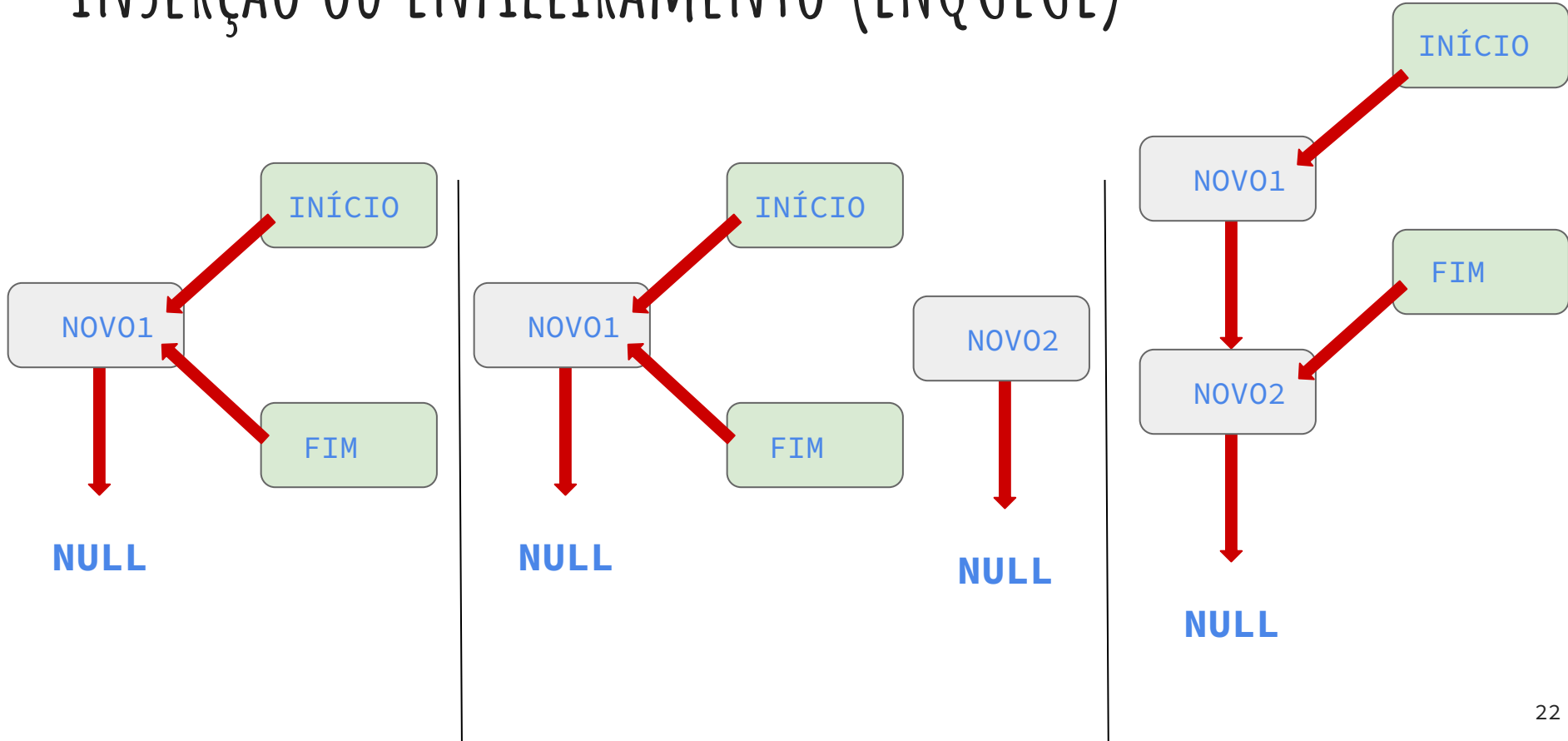
```
novο ← criar_noh(valor);  
se filaVazia()  
    inicio ← novo;  
senão  
    fim.proximo ← novo;  
fim ← novo;  
tamanho++;
```

A inserção em uma fila, consiste em criar um novo nó com o dado a ser armazenado e posteriormente colocá-lo no fim da fila. Assim, ele passa a ser o próximo elemento do fim antigo e o fim passa a ser o novo elemento inserido. Checagem adicional é necessária para verificar se fila estava ou não vazia antes da inserção.

# INSERÇÃO OU ENFILEIRAMENTO (ENQUEUE)



# INSERÇÃO OU ENFILEIRAMENTO (ENQUEUE)



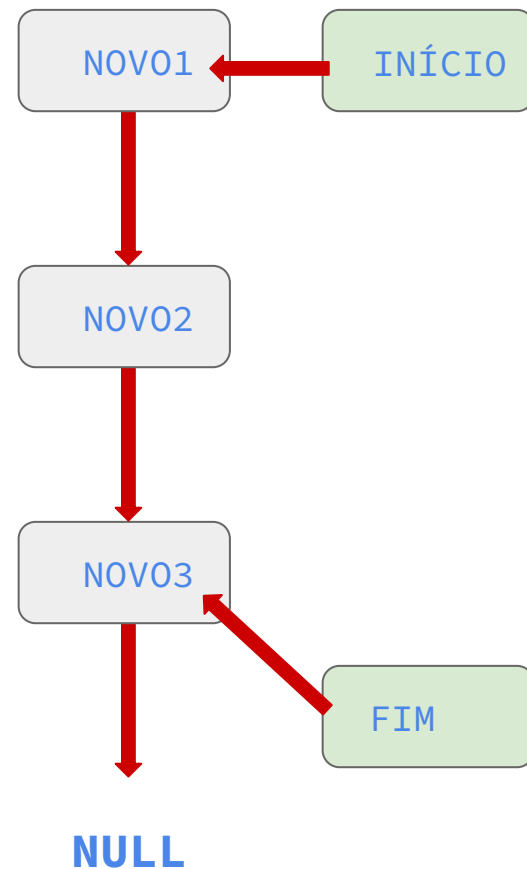
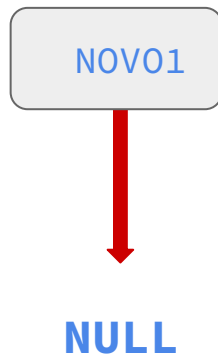
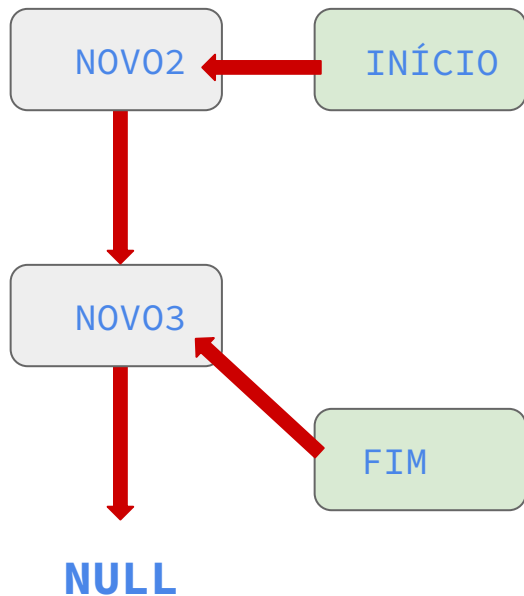
# RETIRADA DE ELEMENTO

A retirada de um elemento da fila consiste em acessar o seu início, retornando-o para a ação desejada. Nesse caso, o novo início passa a ser o próximo nó do início anterior, que será removido do sistema.

## dequeue():

```
aux ← inicio;  
dado ← aux.valor;  
inicio ← aux.proximo;  
apagar(aux);  
tamanho--;  
se filaVazia()  
    fim ← nulo;  
// faz ação desejada  
// (e.g.: retorno)  
efetuaAcao(dado);
```

# DESENFILERAMENTO(DEQUEUE)



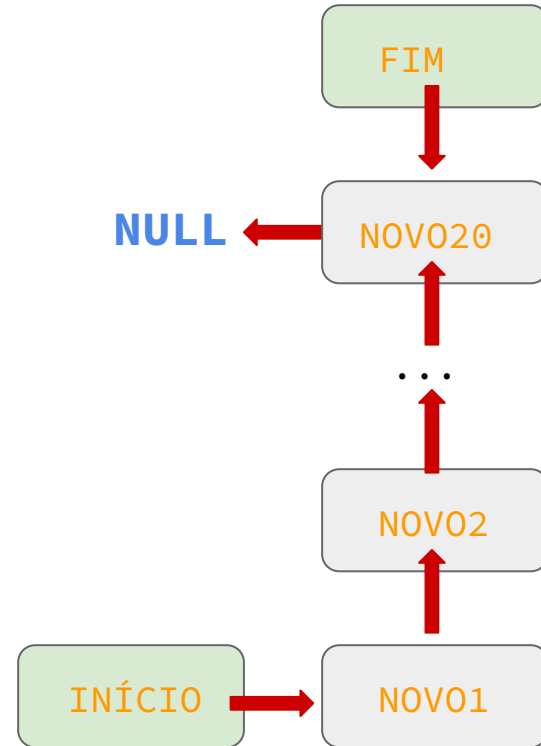


INFORMAÇÃO  
IMPORTANTE



# INFORMAÇÃO IMPORTANTE

- O acesso à Fila deve ser sempre feito pelo FIM, para inserção, e INÍCIO, para retirada.
- As ligações entre os nós é para que a estrutura permaneça integrada.



# INFORMAÇÃO IMPORTANTE

Mas professor, dessa forma eu só posso retirar o elemento que o INÍCIO aponta? E se, por exemplo, na estrutura anterior, eu quiseser utilizar o elemento NOV012, o que fazer?



# INFORMAÇÃO IMPORTANTE

Pela definição, não é possível acessar os outros elementos que não estão no início. Assim, não é possível desenvolver uma função da fila que resolva isso.

Mas você pode retirar todos os elementos da fila até encontrar o elemento desejado. Vamos deixar claro, que essa situação deve ser feita via retirada de elementos, **não** pode ser um procedimento da fila.

# OUTRAS OPERAÇÕES EM FILAS - I

Além das operações básicas apresentadas, as filas podem implementar:

- verificar o número de elementos da fila;
- verificar se a fila está ou não vazia;
- retirar todos os elementos da fila;
- opcionalmente acessar o início da fila;
- opcionalmente depurar/imprimir a fila.

## OUTRAS OPERAÇÕES EM FILAS - II

- A retirada de todos os elementos da fila é por meio de desenfileiramento, um por um.
- O destrutor é implementado geralmente usando o método para retirar todos os elementos da fila.

## OUTRAS OPERAÇÕES EM FILAS - III

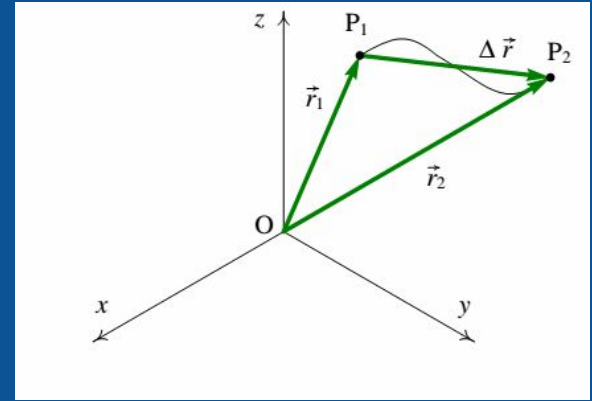
- O acesso ao início (espia) costuma ser disponibilizado em várias implementações. Nesse caso, há o acesso ao dado, mas sem sua retirada, que continua no início da fila.

## OUTRAS OPERAÇÕES EM FILAS - IV

- Implementações iniciais às vezes fazem uso de mecanismos de percorrimto para impressão/depuração dos dados. Esse método é considerado uma **quebra de estrutura**, pois acessa os elementos intermediários sem desenfileiramento. Assim, esse recurso só pode ser usado, para fins didáticos ou de depuração.

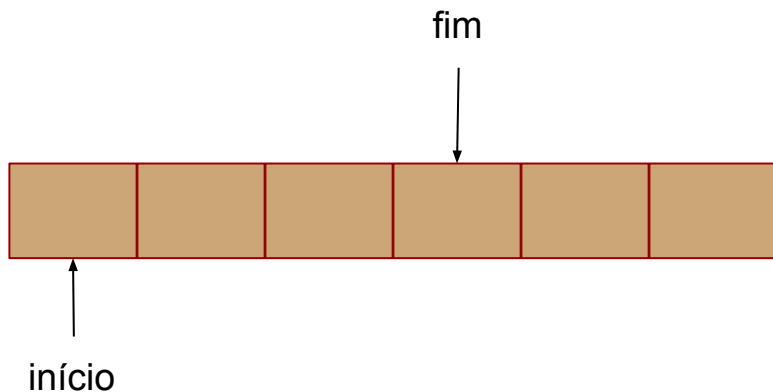


# IMPLEMENTAÇÃO UTILIZANDO VETOR



# FILAS EM ARRANJOS

Assim como as pilhas, as filas também podem ser implementadas por meio de arranjos.

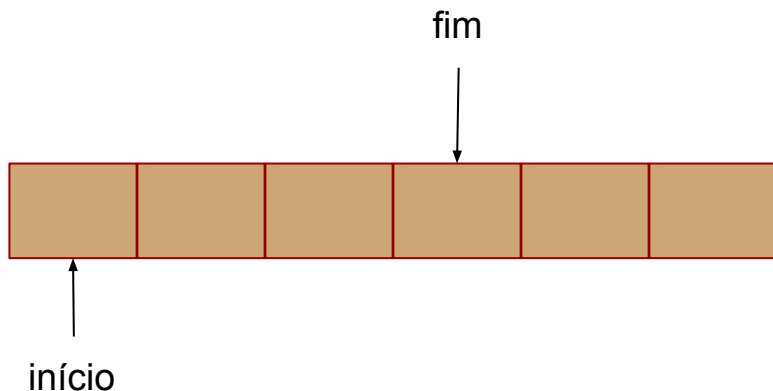


## **Vantagem**

- Operações rápidas (acesso, enfileiramento, desenfileiramento).

# FILAS EM ARRANJOS

Assim como as pilhas, as filas também podem ser implementadas por meio de arranjos.



## Desvantagem

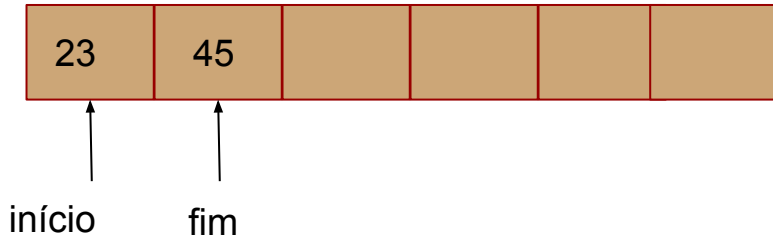
- Espaço alocado sem necessidade;
- Impossibilidade de aumentar a capacidade da fila, sem redimensionamento do vetor.

# Considere os elementos 23, 45, 11 e 89.

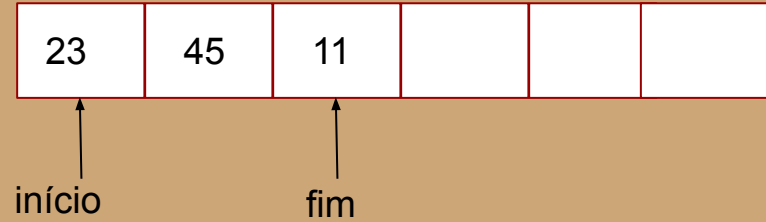
Inserir elemento na fila



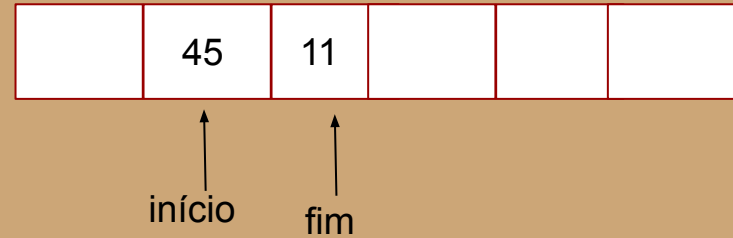
Inserir elemento na fila



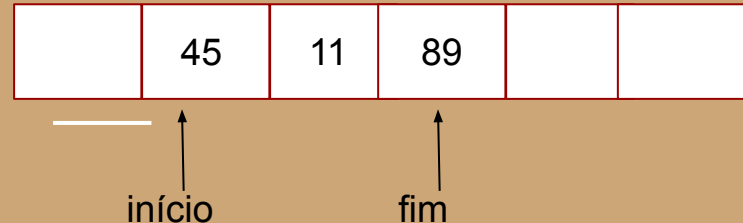
Inserir elemento na fila



Remover elemento da fila

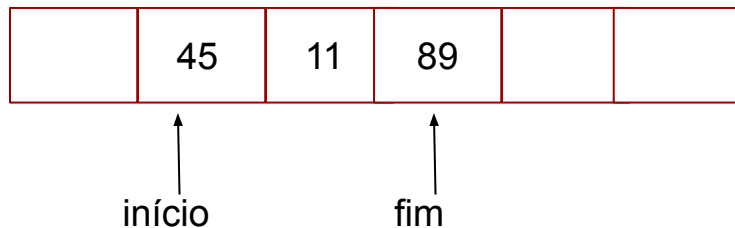


Inserir elemento na fila



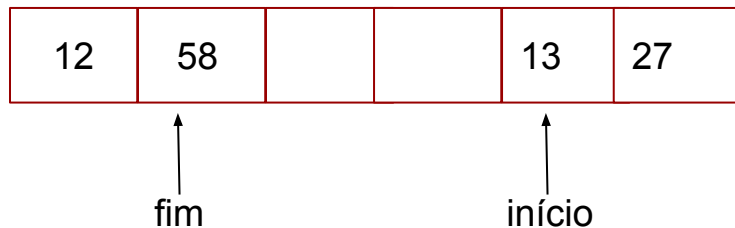
# FILAS CIRCULARES - I

Geralmente implementações de filas em arranjos são chamadas de filas circulares. O motivo é que sem isso, o arranjo não é reorganizado à medida que o início avança:



# FILAS CIRCULARES - II

À medida que as operações são executadas, é possível que a posição do fim fique antes da posição de início em uma fila circular:



Isso é conseguido em implementação utilizando o operador de módulo.

# INSERÇÃO EM FILA CIRCULAR

**enqueue(valor):**

```
se (tamanho = capacidade)
    gerarErro("fila cheia");
se (posInicio = -1) { // fila vazia
    posInicio++;
}
posFim ← (posFim + 1) % capacidade;
dados[posFim] ← valor;
tamanho++;
```

**tamanho** ⇒  
número de  
posições  
ocupadas no  
arranjo

**capacidade** ⇒  
quantidade de  
posições  
alocadas no  
arranjo

# REMOÇÃO EM FILA CIRCULAR

## dequeue():

```
se ( tamanho == 0 )  
    gerarErro("fila vazia");  
aux ← dados[posInicio];  
se (posInicio = posFim) { //removeu último elemento da fila  
    posFim = -1;  
    posInicio = -1;  
} senão  
    posInicio ← (posInicio + 1) % capacidade;  
tamanho--;  
efetuaAção(aux); // geralmente retorno
```



SOBRE O MATERIAL



# SOBRE ESTE MATERIAL

Material produzido coletivamente, principalmente pelos seguintes professores do DCC/UFLA:

- Joaquim Quinteiro Uchôa
- Juliana Galvani Greggi
- Renato Ramos da Silva

Inclui contribuições de outros professores do setor de Fundamentos de Programação do DCC/UFLA.

Esta obra está licenciado com uma Licença [Creative Commons Atribuição 4.0 Internacional](https://creativecommons.org/licenses/by/4.0/).