



# Filas Circulares

Tipo Abstrato de Datos Fila Circular

Visto  
anteriormente

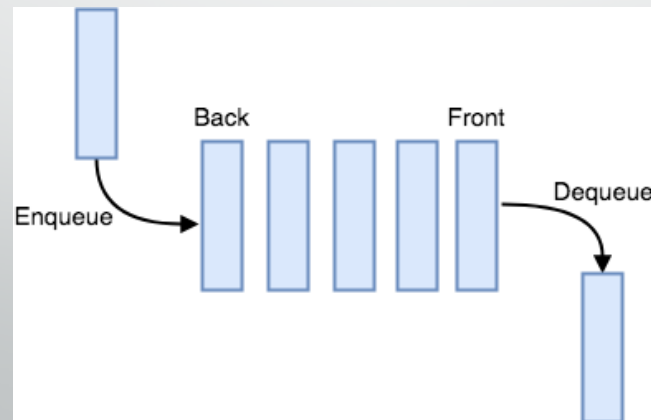
- Foram apresentados os TAD Arranjo, Listas e Filas.
- Foi apresentada a implementação de uma Fila usando arranjos 1D em Java.
  - Implementação apresentada funciona sob os conceitos de uma Fila: *First In First Out*.
- Implementação apresentada é pouco eficiente na operação de saída (pop), pois demanda a movimentação de todos os elementos da fila uma posição à frente.

## Objetivos deste Tutorial

- Apresentar o TAD Fila Circular, que é uma melhoria na implementação da Fila tradicional.

# TAD Filas

- Filas são listas lineares que apresentam as seguintes restrições:
  - As inserções são feitas no fim da fila
  - As remoções são feitas no início da fila.



FIRST IN – FIRST OUT (FIFO)

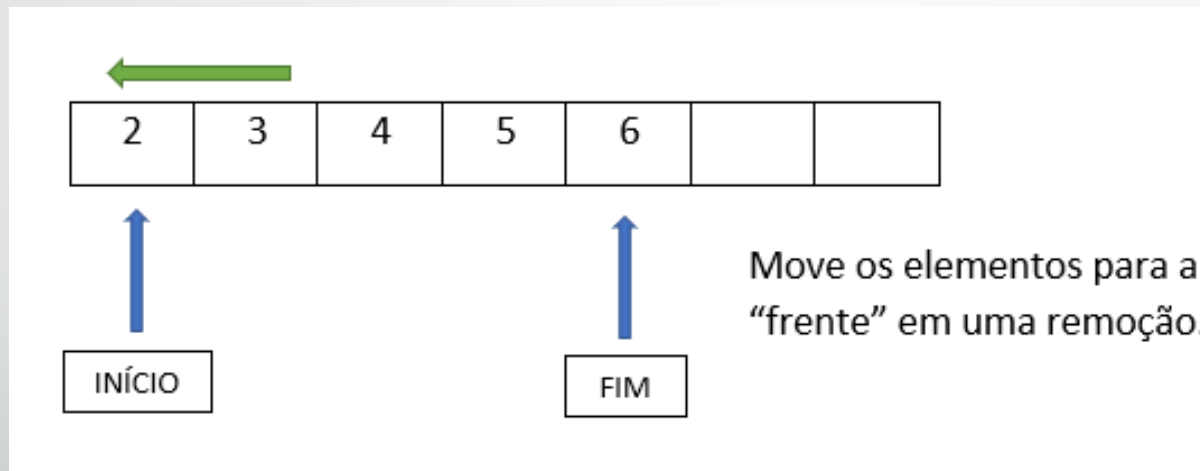
# Operações

- **Push(valor) / enqueue(valor)** – insere o “valor” no fim da fila, se ela não estiver cheia, e atualiza a fila.
- **Pop() / dequeue( )** – retorna o valor que está no início da fila, se ela não estiver vazia, e atualiza a fila.
- **IsEmpty()** – retorna se a fila está vazia.
- **IsFull()** – retorna se a fila está cheia.
- **Initialize()** – cria ou inicializa a fila.
- **Destroy()** – apaga o conteúdo da fila.

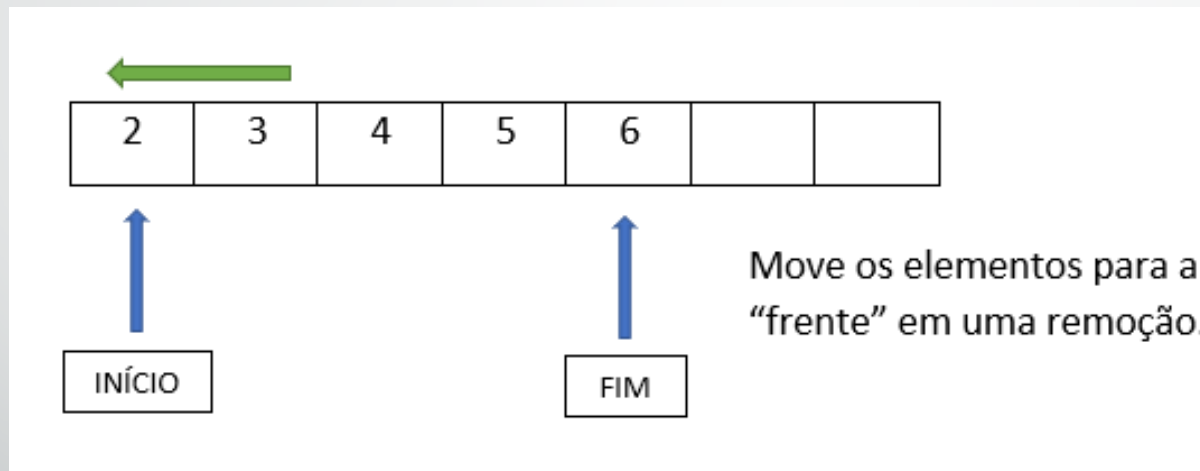
# Operações

- **Push(valor) / enqueue(valor)** – insere o “valor” no fim da fila, se ela não estiver cheia, e atualiza a fila.
- **Pop() / dequeue( )** – retorna o valor que está no início da fila, se ela não estiver vazia, e atualiza a fila. (BAIXA EFICIÊNCIA)
- **IsEmpty()** – retorna se a fila está vazia.
- **IsFull()** – retorna se a fila está cheia.
- **Initialize()** – cria ou inicializa a fila.
- **Destroy()** – apaga o conteúdo da fila.

## Remoção de um elemento da fila



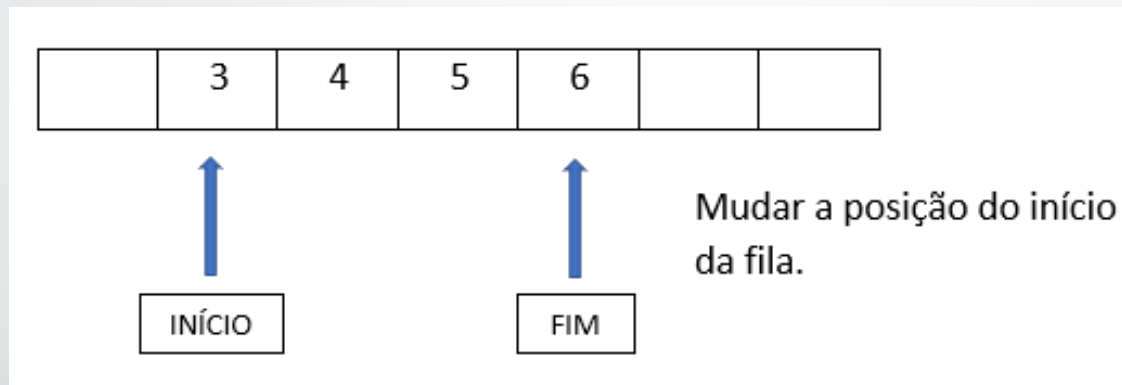
## Remoção de um elemento da fila



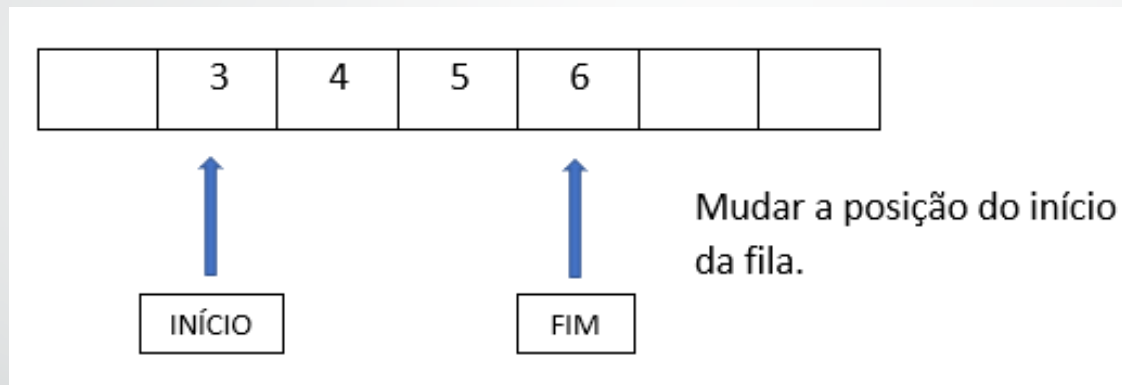
Muitas instruções computacionais.



## Proposta de melhoria na remoção (pop)

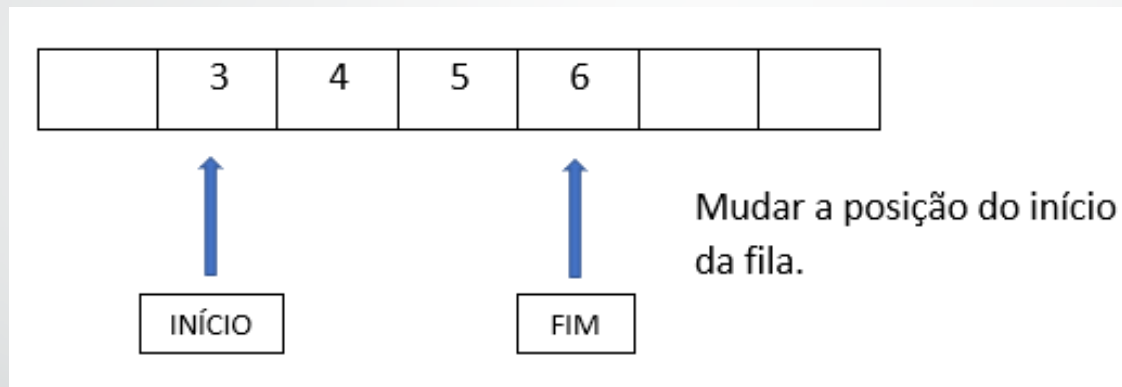


## Proposta de melhoria na remoção (pop)



Evita a movimentação dos elementos no arranjo.

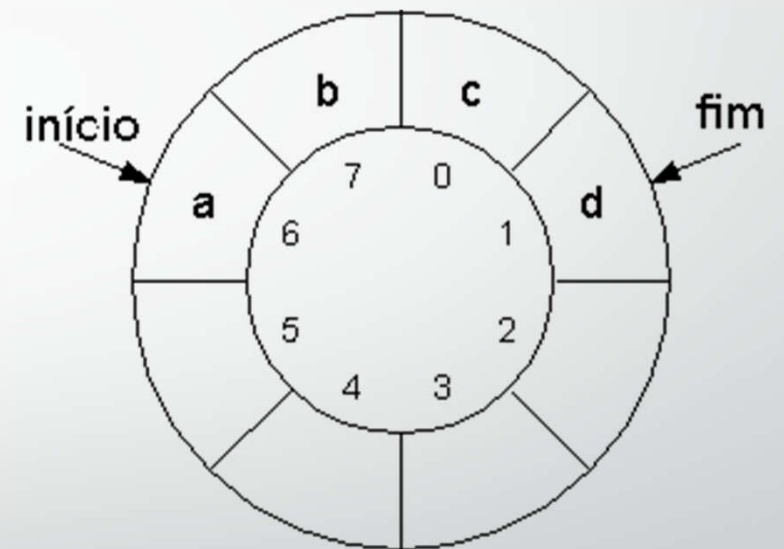
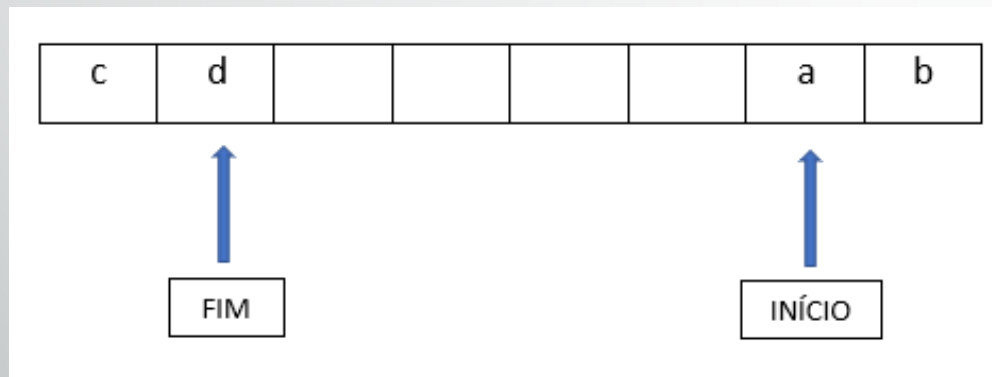
## Proposta de melhoria na remoção (pop)



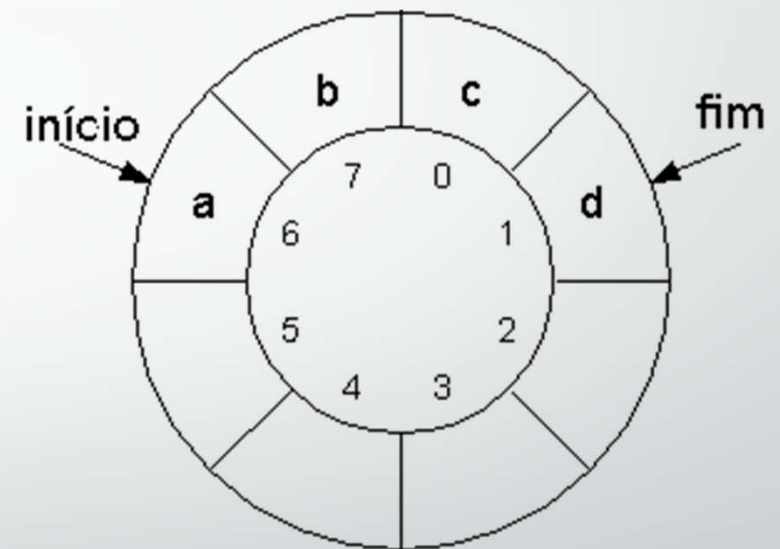
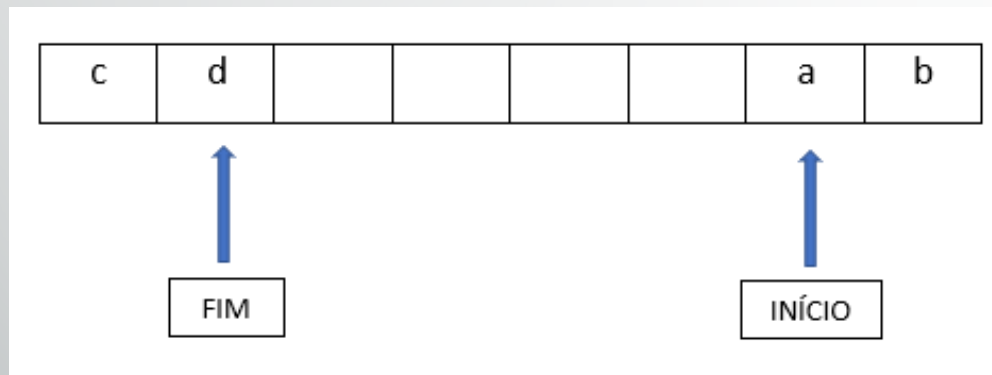
Evita a movimentação dos elementos no arranjo.

Libera posições para serem ocupadas pela fila.

# Fila Circular



# TAD Fila Circular



O final do Arranjo está ligada (logicamente) com o começo do Arranjo.

## Atributos de uma fila circular

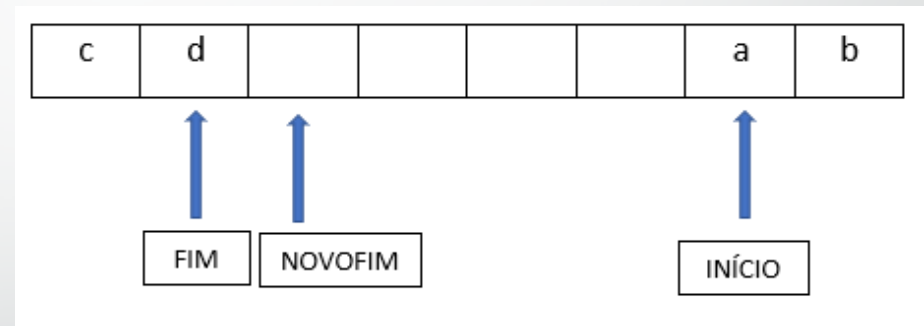
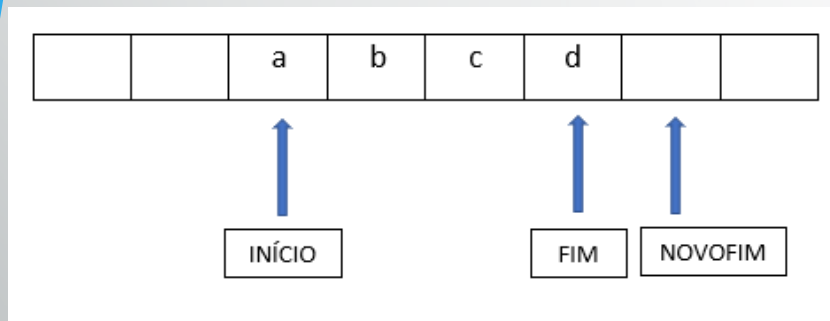
- ARRANJO 1D para armazenar os elementos da fila
- inicio  $\Rightarrow$  guarda a posição do início da fila
- fim  $\Rightarrow$  guarda a posição do fim da fila

## Método push(valor) - inserção

- Inserção em fila vazia  $\Rightarrow$  inicio = fim = 0
- Inserção será feita sempre na posição posterior ao fim da fila, se a fila não estiver cheia.
  - 1º) definir a nova posição de fim da fila.
  - 2º) verificar se a nova posição está vaga.
    - Se estiver vaga, fazer a inserção
    - Se não estiver vaga  $\Rightarrow$  Erro de fila cheia!!!

# Método push(valor) - inserção

1ª Situação: novoFim indica posição válida e vaga

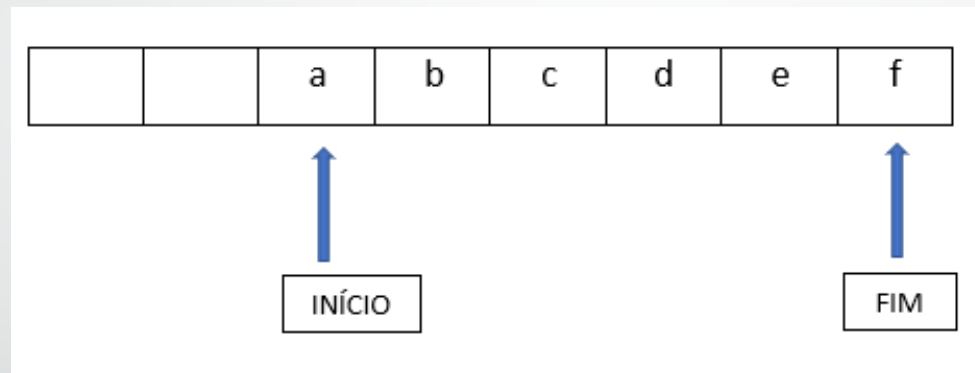


$$\text{novoFim} = \text{fim} + 1$$



# Método push(valor) - inserção

2ª Situação: novoFim indica posição fora do vetor

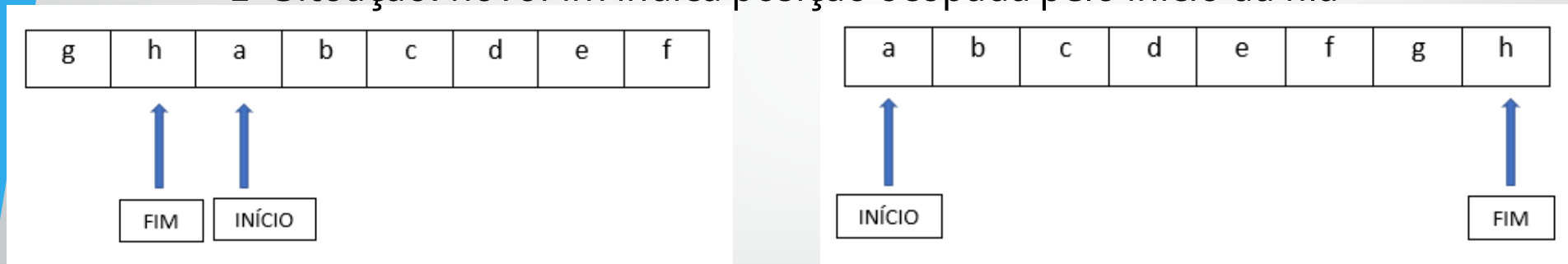


$\text{novoFim} = \text{fim} + 1$

se  $(\text{novoFim} == \text{vetor.length}) \Rightarrow \text{novoFim} = 0$

# Método push(valor) - inserção

1ª Situação: novoFim indica posição ocupada pelo inicio da fila



se novoFim == inicio (fila cheia!)

## Método push(valor) - inserção

- Como definir a nova posição do fim?
    - $\text{novoSfim} = \text{fim} + 1$
    - $\text{se } (\text{novoSfim} == \text{vetor.length}) \Rightarrow \text{novoSfim} = 0$

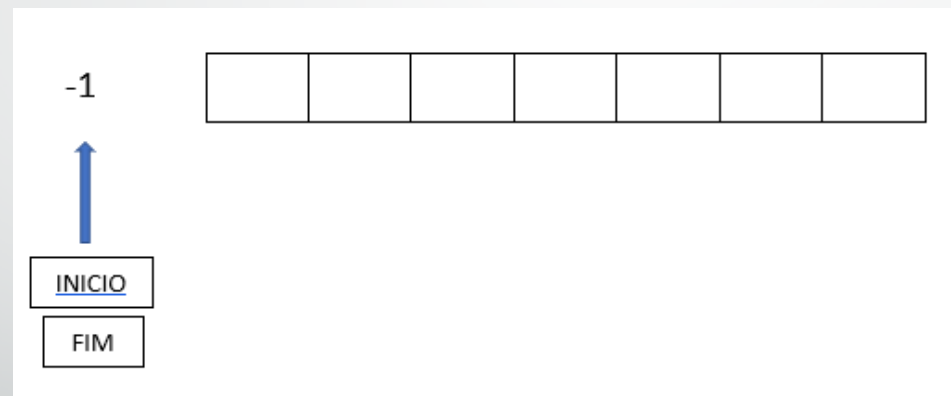
} Define novoFim
  - $\text{se } (\text{novoSfim} != \text{inicio}) \Rightarrow \text{fim} = \text{novoSfim}$
  - $\text{se } (\text{novoSfim} == \text{inicio}) \Rightarrow \text{Fila Cheia!}$
- } Atualiza o fim, se fila não estiver cheia

## Método pop() - remoção

- Fila vazia  $\Rightarrow$   $\text{inicio} = \text{fim} = -1$
- Remoção será feita sempre do início da fila, se ela não estiver vazia
  - Se  $(\text{inicio} \neq \text{fim}) \Rightarrow \text{novoInicio} = \text{inicio} + 1$ 
    - Se  $(\text{novoInicio} == \text{vetor.length}) \text{novoInicio} = 0$
  - Se  $(\text{inicio} == \text{fim}) \Rightarrow \text{inicio} = \text{fim} = -1$

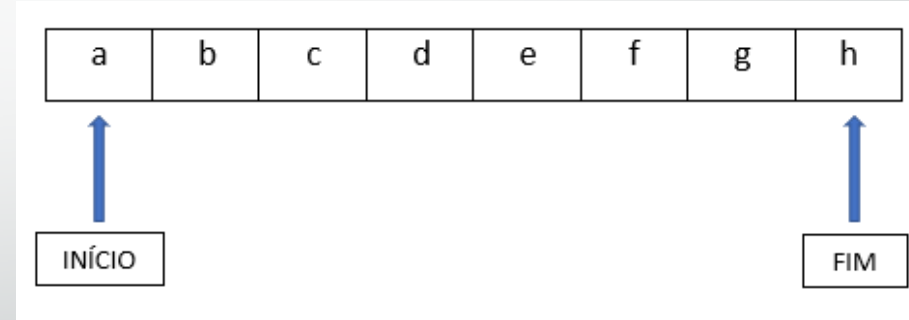
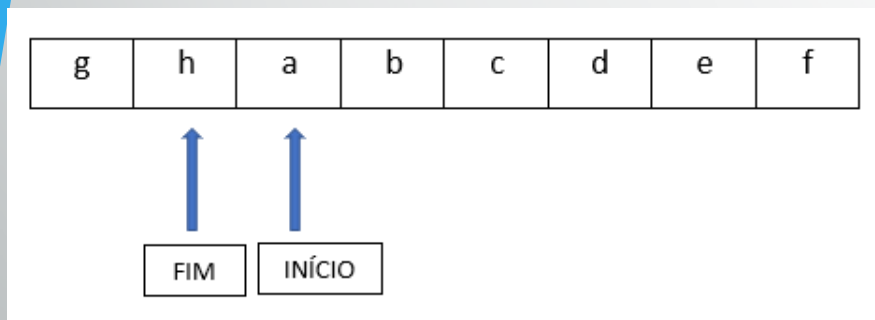
## Fila Vazia – isEmpty()

- Critério para fila vazia:  $\text{inicio} == \text{fim} == -1$



## Fila Cheia – isFull( )

- Critério para fila cheia:  $\text{novoFim} == \text{inicio}$



## TAD Filas Circulares

### Conclusões:

- Movimentação do início torna a implementação de filas mais eficiente.
- Discutimos algoritmos que tratam a junção do final do arranjo com o começo do arranjo na definição das posições início e fim da fila.
- Discutimos como identificar filas vazia e cheia.