



**INSTITUTO  
FEDERAL**

Sudeste de  
Minas Gerais

Campus  
Manhuaçu

# Estruturas de Dados I

## Filas

Prof. Leonardo C. R. Soares - [leonardo.soares@ifsudestemg.edu.br](mailto:leonardo.soares@ifsudestemg.edu.br)

Instituto Federal do Sudeste de Minas Gerais

1 de novembro de 2023





# Filas

## Descrição

- Filas são um tipo abstrato de dados com a característica de que **o primeiro elemento a ser inserido é o primeiro a ser removido** (política FIFO – First in First Out).





# Filas

## Descrição

- ▶ Filas são um tipo abstrato de dados com a característica de que **o primeiro elemento a ser inserido é o primeiro a ser removido** (política FIFO – First in First Out).
- ▶ Considera-se uma analogia com filas de elementos, como pessoas, processos etc.





# Filas

## Descrição

- ▶ Filas são um tipo abstrato de dados com a característica de que **o primeiro elemento a ser inserido é o primeiro a ser removido** (política FIFO – First in First Out).
- ▶ Considera-se uma analogia com filas de elementos, como pessoas, processos etc.
- ▶ Os usos de filas incluem filas de impressão e filas de processamento em sistemas operacionais, entre outros.





# Operações

O tipo abstrato de dados (TAD) fila deve, obrigatoriamente, suportar os métodos:

- ▶ enfileirar(o): Insere o objeto no fim da fila.





# Operações

O tipo abstrato de dados (TAD) fila deve, obrigatoriamente, suportar os métodos:

- ▶ enfileirar(o): Insere o objeto no fim da fila.
- ▶ desenfileirar(): Retira o objeto no início da fila e o retorna; se a fila estiver vazia, ocorre um erro.





# Operações

O tipo abstrato de dados (TAD) fila deve, obrigatoriamente, suportar os métodos:

- ▶ `enfileirar(o)`: Insere o objeto no fim da fila.
- ▶ `desenfileirar()`: Retira o objeto no início da fila e o retorna; se a fila estiver vazia, ocorre um erro.

Adicionalmente, podemos definir os seguintes métodos auxiliares:

- ▶ `tamanho()`: Retorna o número de objetos na fila.





# Operações

O tipo abstrato de dados (TAD) fila deve, obrigatoriamente, suportar os métodos:

- ▶ `enfileirar(o)`: Insere o objeto no fim da fila.
- ▶ `desenfileirar()`: Retira o objeto no início da fila e o retorna; se a fila estiver vazia, ocorre um erro.

Adicionalmente, podemos definir os seguintes métodos auxiliares:

- ▶ `tamanho()`: Retorna o número de objetos na fila.
- ▶ `vazia()`: Retorna um *boolean* indicando se a fila está vazia.







# Implementação por referência

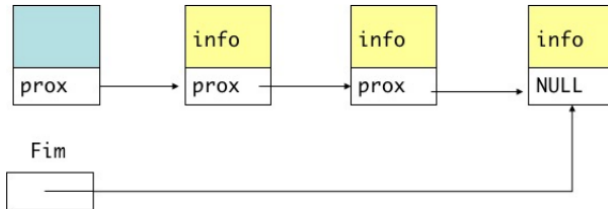
- A fila é implementada por meio de células, tal que cada célula contém um item da fila e um apontador para a próxima célula.





# Implementação por referência

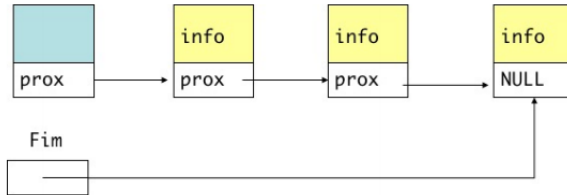
- ▶ A fila é implementada por meio de células, tal que cada célula contém um item da fila e um apontador para a próxima célula.
- ▶ A estrutura contém um apontador para a frente da fila (célula cabeça) e um apontador para a parte de trás da fila (fim).





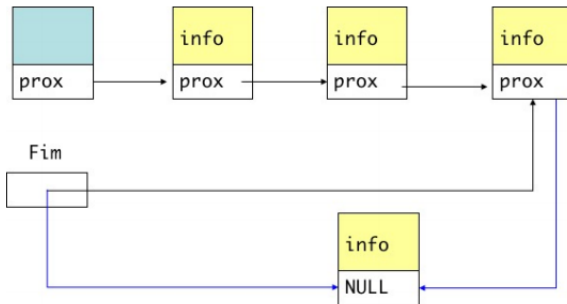
## Implementação por referência - Inserção

De acordo com a política **FIFO**, há apenas uma opção de posição onde podemos inserir elementos: o fim da fila (ou seja, a última posição).





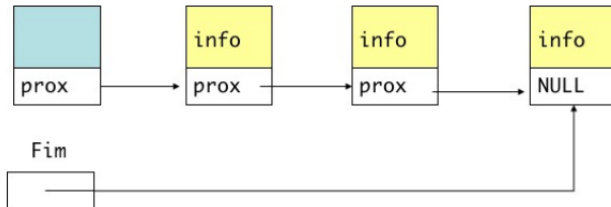
## Implementação por referência - Inserção





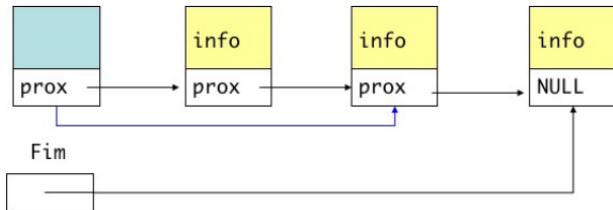
## Implementação por referência - Remoção

De acordo com a política **FIFO**, há apenas uma opção de posição onde podemos remover elementos: o início da fila (ou seja, primeira posição).





## Implementação por referência - Remoção





# Complexidade

A complexidade de todas as operações é mantida da implementação de Lista:

- ▶ Enfileirar:  $\theta(1)$
- ▶ Desenfileirar:  $\theta(1)$





# Perguntas?







Baixe o exemplo aqui





# Exercícios

Implemente o exemplo sem o uso de *generics*. Você terá que fazer uma pilha para pacientes e outra para professores.





# Exercícios

Considere uma pilha  $P$  vazia e uma fila  $F$  não vazia. Utilizando apenas os testes de fila e pilha vazias, as operações Enfileira, Desenfileira, Empilha (push), Desempilha (pop), e uma variável  $aux$  do tipo  $\langle \text{elemento} \rangle$ , escreva uma função que inverta a ordem dos elementos da fila.





# Exercícios

Escreva um programa em Java que permita cadastrar Strings em uma fila  $q$  (utilize Referência). Os cadastros (enfileiramento) acontecerão de dois em dois. O usuário, para preencher esta fila inicial, irá informar um nome e uma operação (A, B ou X) a cada operação de enfileiramento. Quando o usuário terminar o cadastro, os elementos da fila original serão separados em duas filas,  $a$  e  $b$ , de acordo com a operação (GIT).

- ▶ Nome A - Adiciona nome à fila  $a$
- ▶ Nome B - Adiciona nome à fila  $b$
- ▶ Nome X - Adiciona nome à fila que tenha menos elementos ( $a$  ou  $b$ ). Se ambas as filas tiverem o mesmo número de elementos, o nome é descartado e não é adicionada a nenhuma.





## Exercícios

Por exemplo, se a fila  $q$  for preenchida com [Luis, B, Pedro, A, Luisa, A, Joao, X, Jose, X, Miguel, B] quando for processada acontecerá:

1. Luis, B - Luis é adicionado à fila b;
2. Pedro, A - Pedro é adicionado à fila a;
3. Luisa, A - Luisa é adicionada à fila a;
4. Joao, X - Joao é adicionado à fila b, que tem apenas 1 elemento (Luis) contra os dois da fila b (Pedro e Luisa);
5. Jose, X - José é descartado (ambas as filas têm 2 elementos);
6. Miguel, B - Miguel é adicionado à fila b;

No final, a fila  $a$  ficaria com [Pedro, Luisa] e a fila  $b$  ficaria com [Luis, Joao, Miguel]. A fila  $q$  deve ficar vazia.







# Referências

- ▶ CARVALHO, Marco Antonio Moreira de. **Projeto e análise de algoritmos**. 01 mar. 2018, 15 jun. 2018. Notas de Aula. PPGCC. UFOP
- ▶ GOODRICH, Michael T.; TAMASSIA, Roberto. **Estruturas de Dados & Algoritmos em Java**. Bookman Editora, 2013.
- ▶ ZIVIANI, Nivio. **Projeto de Algoritmos com implementações em Java e C++**, 2007.

