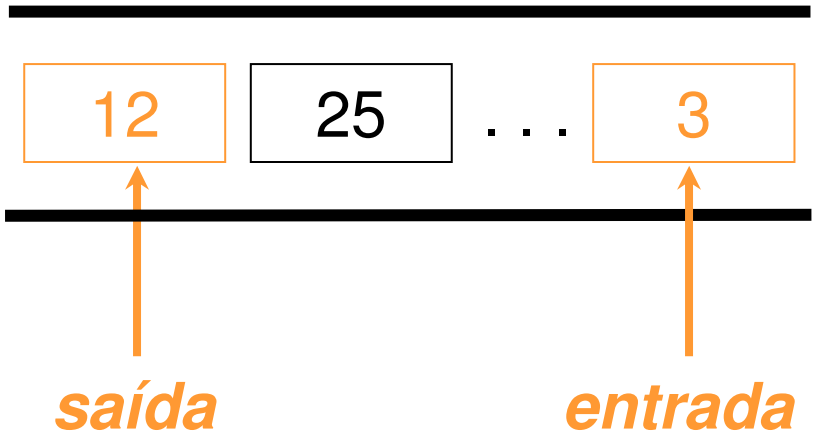
Conteúdo

1. Introdução
2. Listas
3. Pilhas e Filas
4. Árvores
5. Árvores de Pesquisa
   - Árvore Binária e Árvore AVL
   - Árvore N-ária e Árvore B
6. Tabelas de Dispersão (Hashing)
7. Métodos de Acesso a Arquivos
8. Métodos de Ordenação de Dados

Filas

Fila

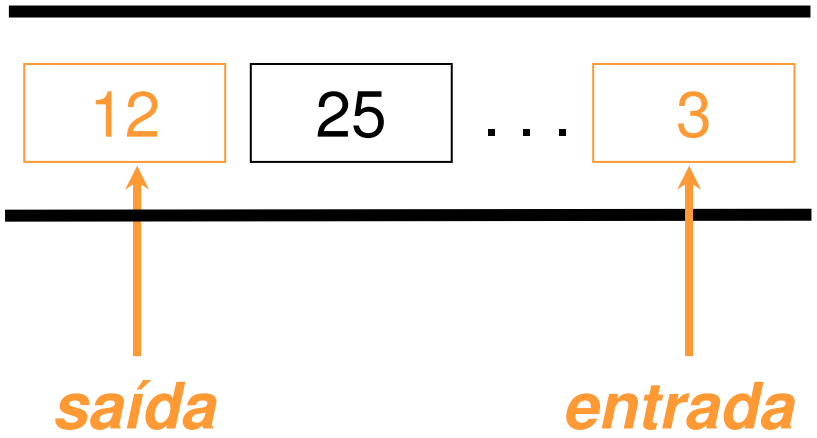
FILA: lista linear cujas inserções se realizam em uma extremidade (fim da lista) e exclusões ocorrem na extremidade oposta (início da lista).



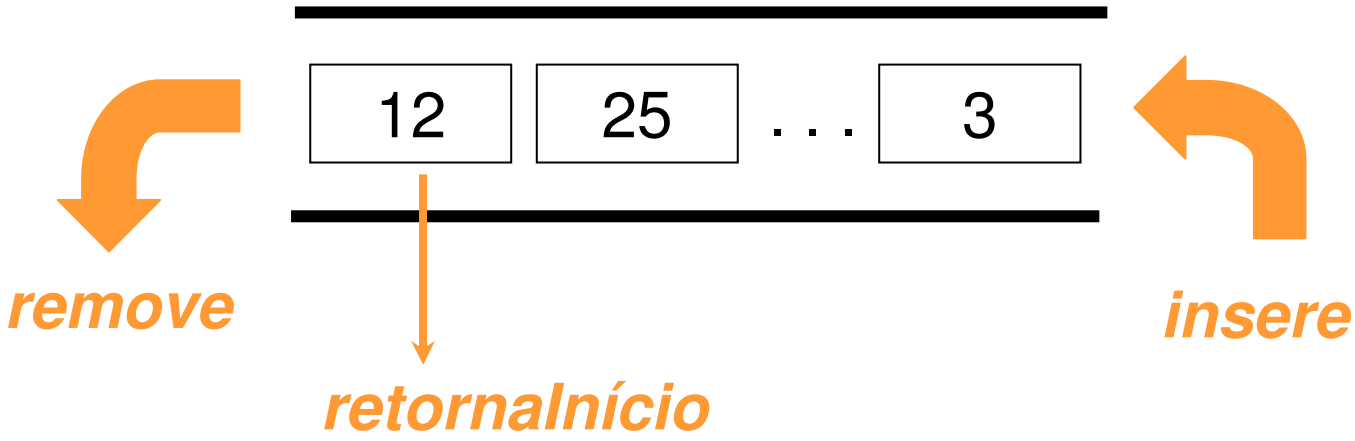
➡ As filas são conhecidas como listas FIFO (First In First Out) - primeiro que entra, primeiro que sai.

Exemplos de Filas

- Fila de pessoas na caixa do banco
- Fila de automóveis no pedágio
- Fila de “objetos” quaisquer, ...



Operações sobre uma Fila



FIFO: “first-in; first-out”

Operações sobre uma Fila

⇨ Uma fila contém:

- um método para inserir elementos na fila (insere)
- um método para retirar elementos da fila (remove)
- um método para acessar o início da fila (retornaInicio)

Interface da Fila

```
public interface Fila<E> extends EstruturaDeDados {

    ???

}
```

Interface da Fila

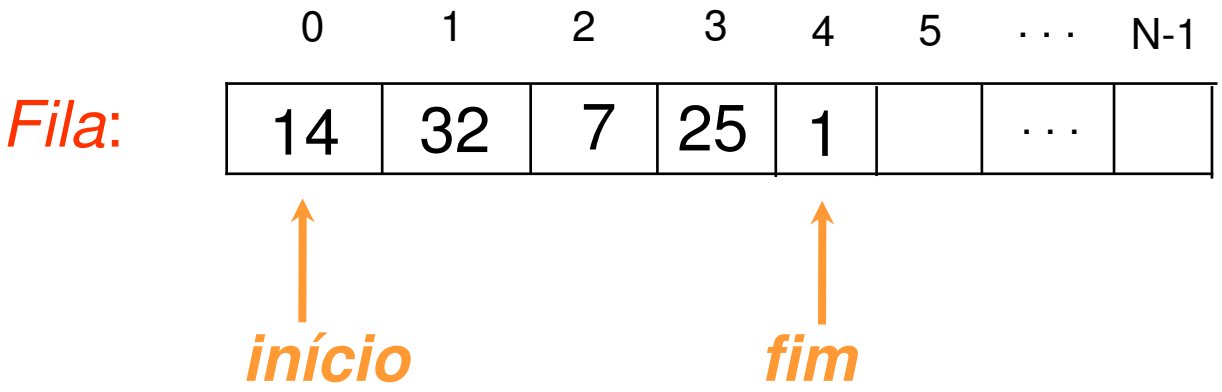
```
public interface Fila <E> extends EstruturaDeDados {
    public void insere (E elemento);
    public E remove () throws ExcecaoEstruturaVazia;
    public E retornaInicio () throws ExcecaoEstruturaVazia;
    public boolean contem (E elemento);
    public int retornaPosicao (E elemento);
}
```

Alternativas de Implementação

- Fila como Array
  - Opção 1: primeiro elemento está sempre na primeira posição do array
  - Opção 2: primeiro elemento é indicado pela posição do início
- Fila como Lista Encadeada

Fila como Array

Opção 1: Seja *fila* uma estrutura linear representada através de um array com N posições, onde o primeiro elemento da fila está sempre na primeira posição do array.



Numero de Elementos: 5

Fila como Array - Construtor

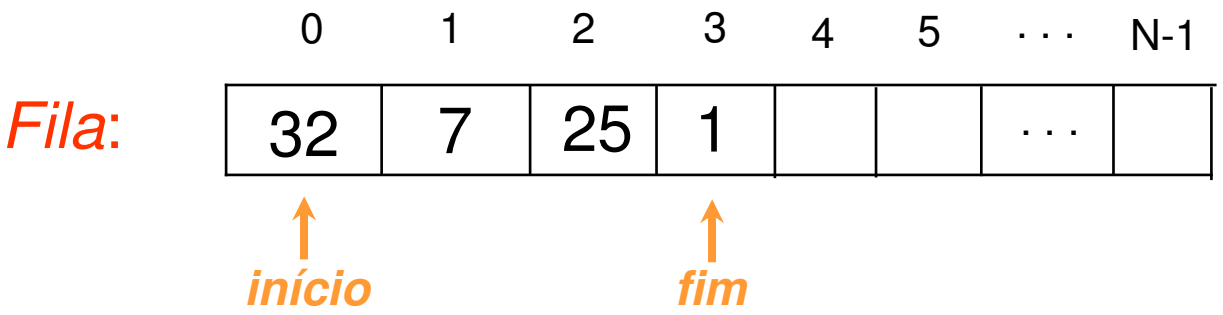
```
public class FilaArray<E> implements Fila<E> {


    private E[] elementos;
    private int inicio;
    private int fim;
    private int numElementos;
```

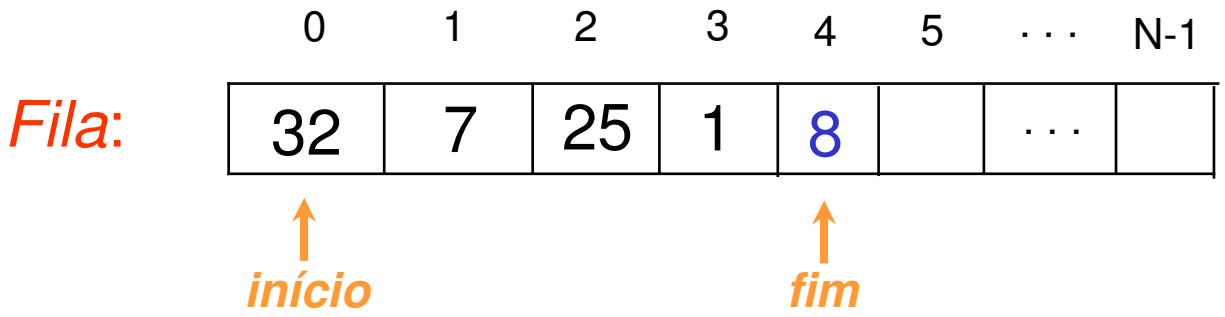
Fila como Array - Construtor

```
public class FilaArray<E> implements Fila<E> {

   private E[] elementos;
   private int numElementos;

public FilaArray (int tamanho) {
   this.elementos = (E[]) new Object[tamanho];
   this.numElementos = 0;  }

public FilaArray () {
   this.elementos = (E[]) new Object[10];
   this.numElementos = 0;  }
```

Complexidade: O(?)

Fila como Array - Insere





Numero de Elementos: 5

Fila como Array - Insere

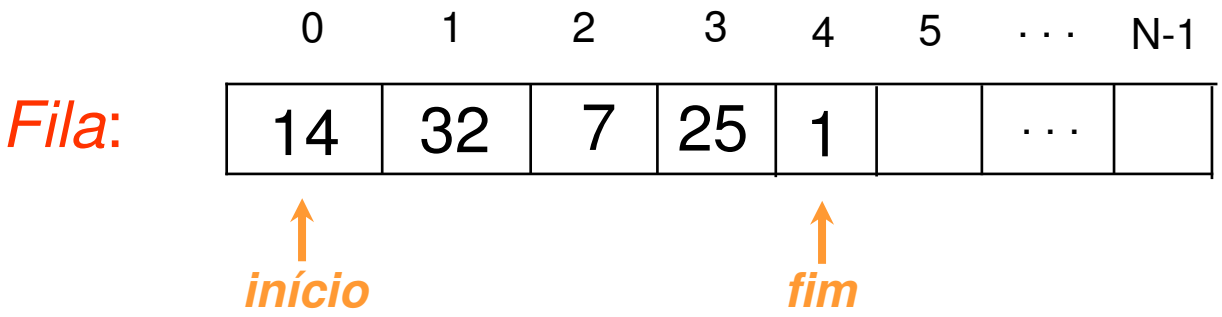
```
public void insere (E elemento) {

???

}
```

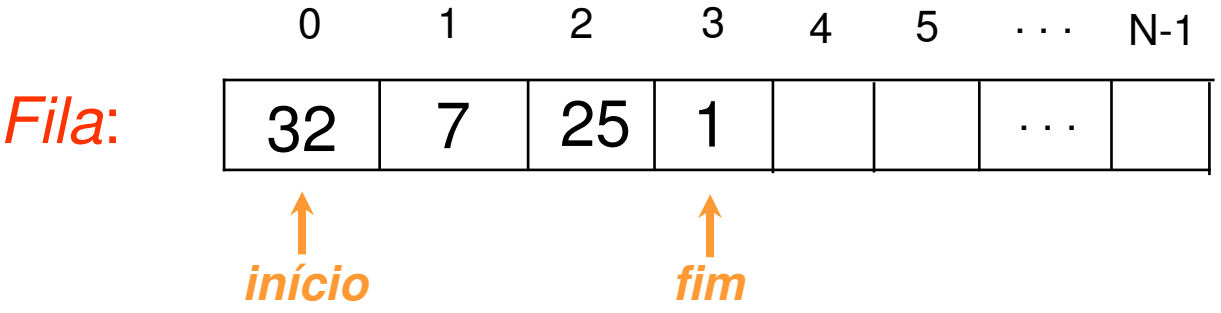
Fila como Array - Insere

```
public void insere (E elemento) {
   if (this.numElementos == this.elementos.length){
        E[] novoArray = (E[]) new Object[this.elementos.length * 2];
        System.arraycopy(elementos,0,novoArray,0,this.elementos.length);
        this.elementos = novoArray;
   }
   this.elementos[this.numElementos] = elemento;
   this.numElementos++;
}
```

Complexidade: O(?)

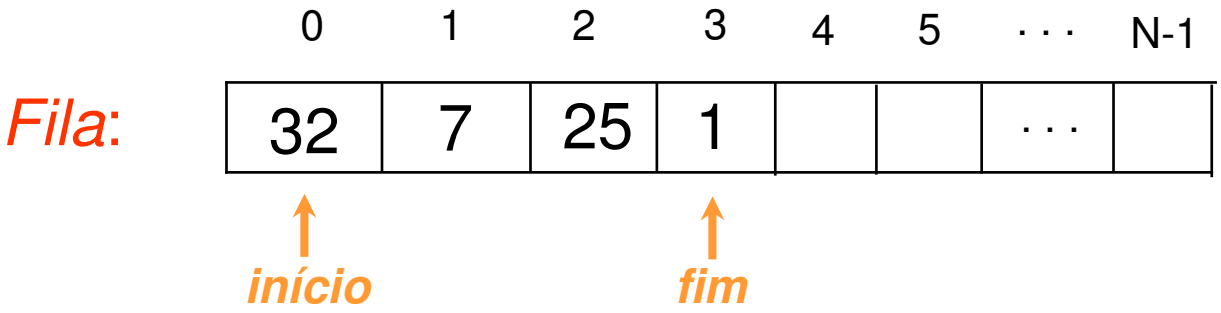
Fila como Array - Remove





Numero de Elementos: 4

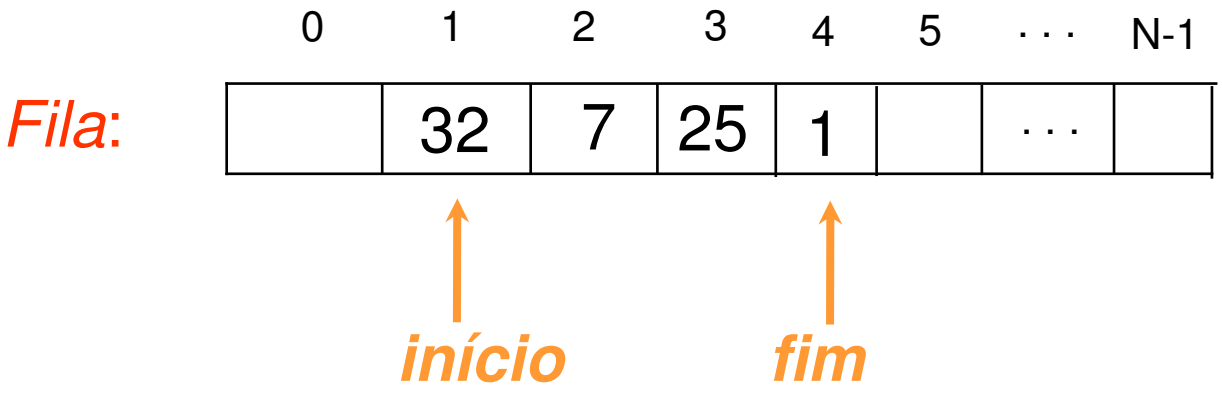
Fila como Array - Retorna Início



➡ ***Início = 32***

Fila como Array

Opção 2: Seja *fila* uma estrutura linear representada através de um array com N posições, onde o primeiro e o último elemento da fila são indicados através dos atributos *início* e *fim*.



Numero de Elementos: 4

Exclusão de Elementos da Fila é mais eficiente!

Fila como Array - Construtor

```
public class FilaArray2<E> implements Fila<E> {
   private E[] elementos;
   private int inicio;
   private int fim;
   private int numElementos;

public FilaArray2 () {


???


}
```

Fila como Array - Construtor

```
public class FilaArray2<E> implements Fila<E> {
   private E[] elementos;
   private int inicio;
   private int fim;
   private int numElementos;

public FilaArray2 (int tamanho) {
   this.elementos = (E[]) new Object[tamanho];
   this.inicio = 0;
   this.fim = -1;
   this.numElementos = 0;  }
```

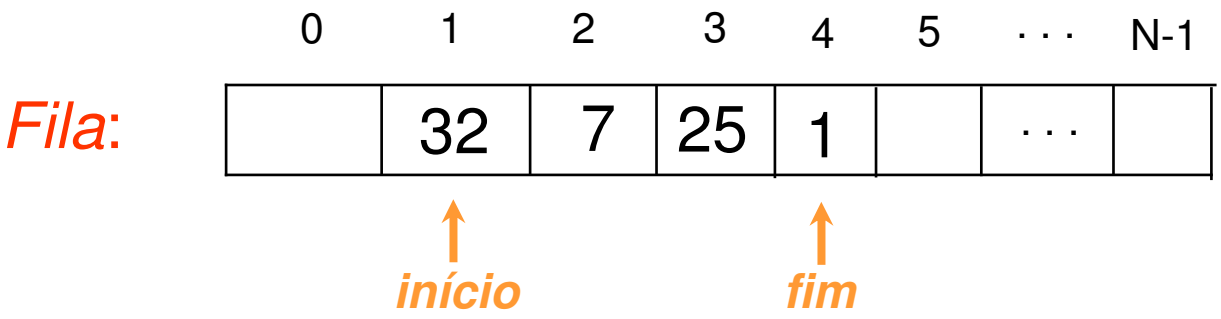
Complexidade: O(?)

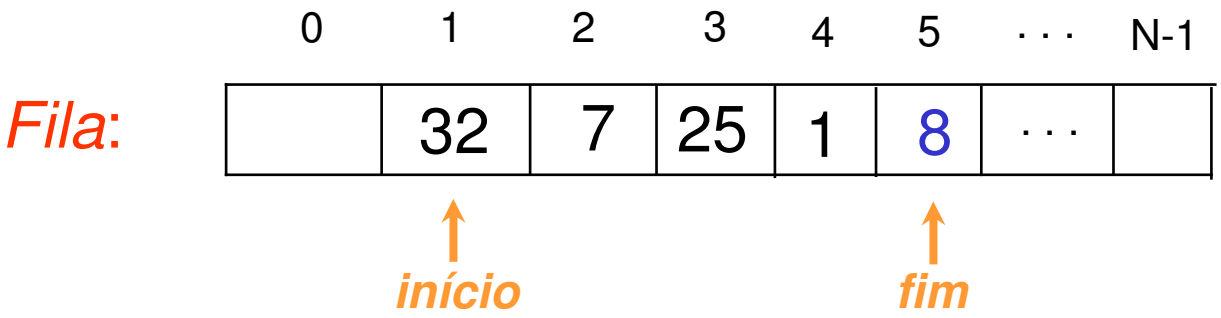
Fila como Array - Construtor

```
public FilaArray2 () {
   this.elementos = (E[]) new Object[100];
   this.inicio = 0;
   this.fim = -1;
   this.numElementos = 0;  }
```

Complexidade: O(?)

Fila como Array - Insere





Numero de Elementos: 5

Fila como Array - Insere

```
public void insere (E elemento) {

???

}
```

Fila como Array - Insere

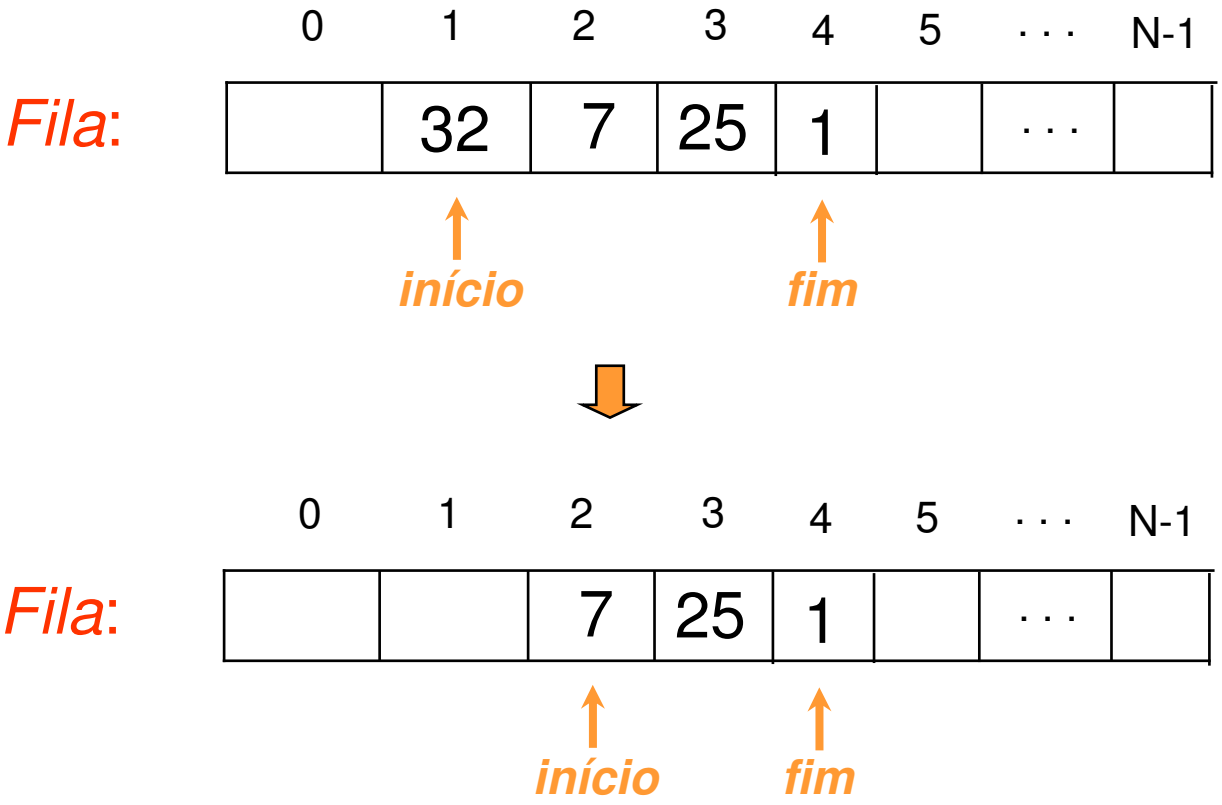
```
public void insere (E elemento) {
   if (this.numElementos == this.elementos.length) {
      E[] novoArray = (E[]) new Object[this.elementos.length * 2];
      System.arraycopy (elementos,this.inicio,novoArray,0,this.elementos.length-
      this.inicio);
      if (this.fim < this.inicio)
            System.arraycopy (elementos,0,novoArray,this.elementos.length-
            this.inicio,this.fim+1);
      this.elementos = novoArray;
      this.inicio = 0;
      this.fim = this.numElementos-1;
   }
   ...
```

Fila como Array - Insere

```
...
    if (this.fim == this.elementos.length-1)
        this.fim = 0;
    else
        this.fim++;
    this.elementos[this.fim] = elemento;
    this.numElementos++;
}
```

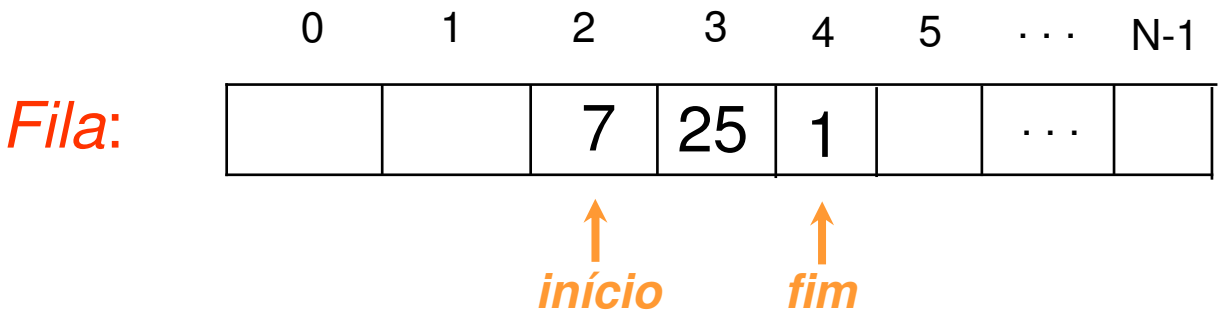
Complexidade: O(?)

Fila como Array - Remove



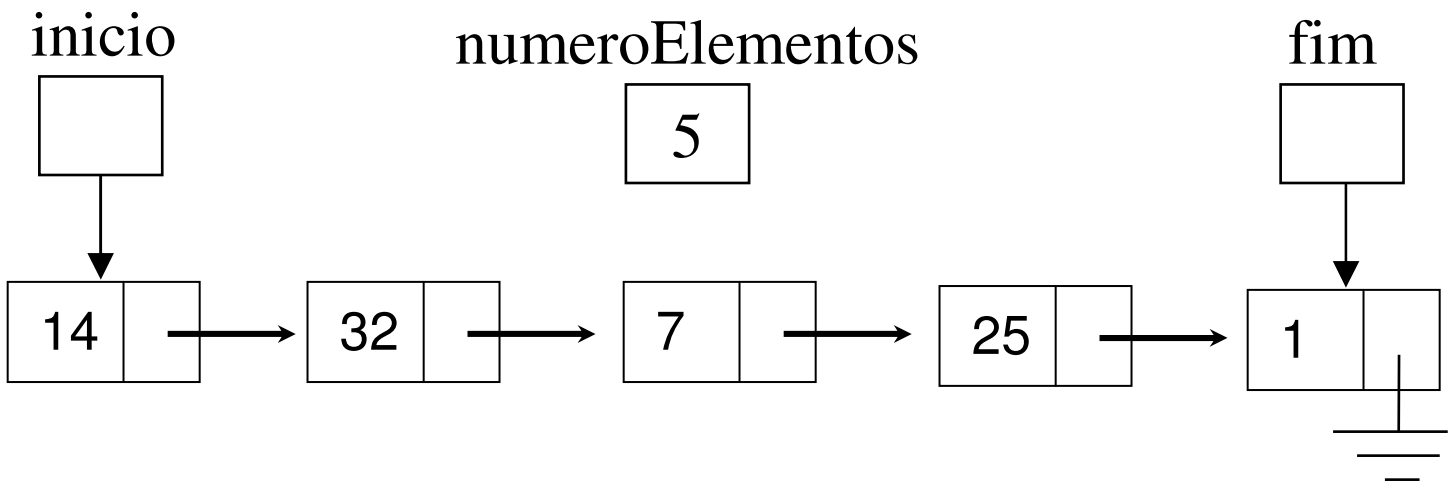
Numero de Elementos: 3

Fila como Array - Retorna Início



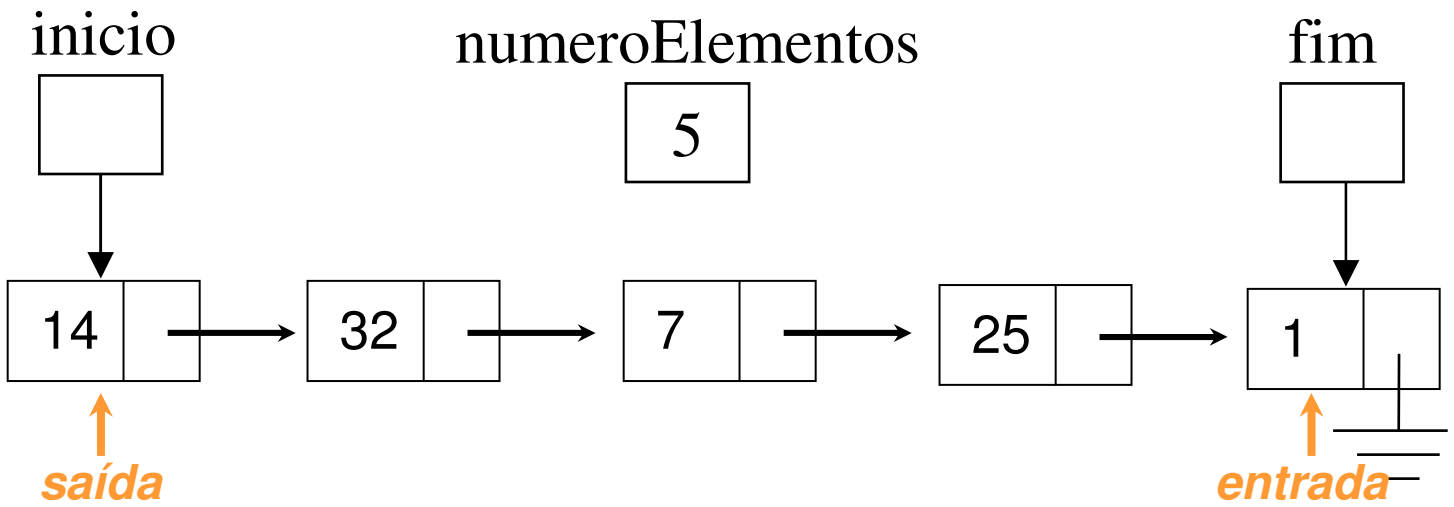
➡ ***Início = 7***

Fila Encadeada



Onde deve ficar a entrada e a saída da fila?

Fila Encadeada



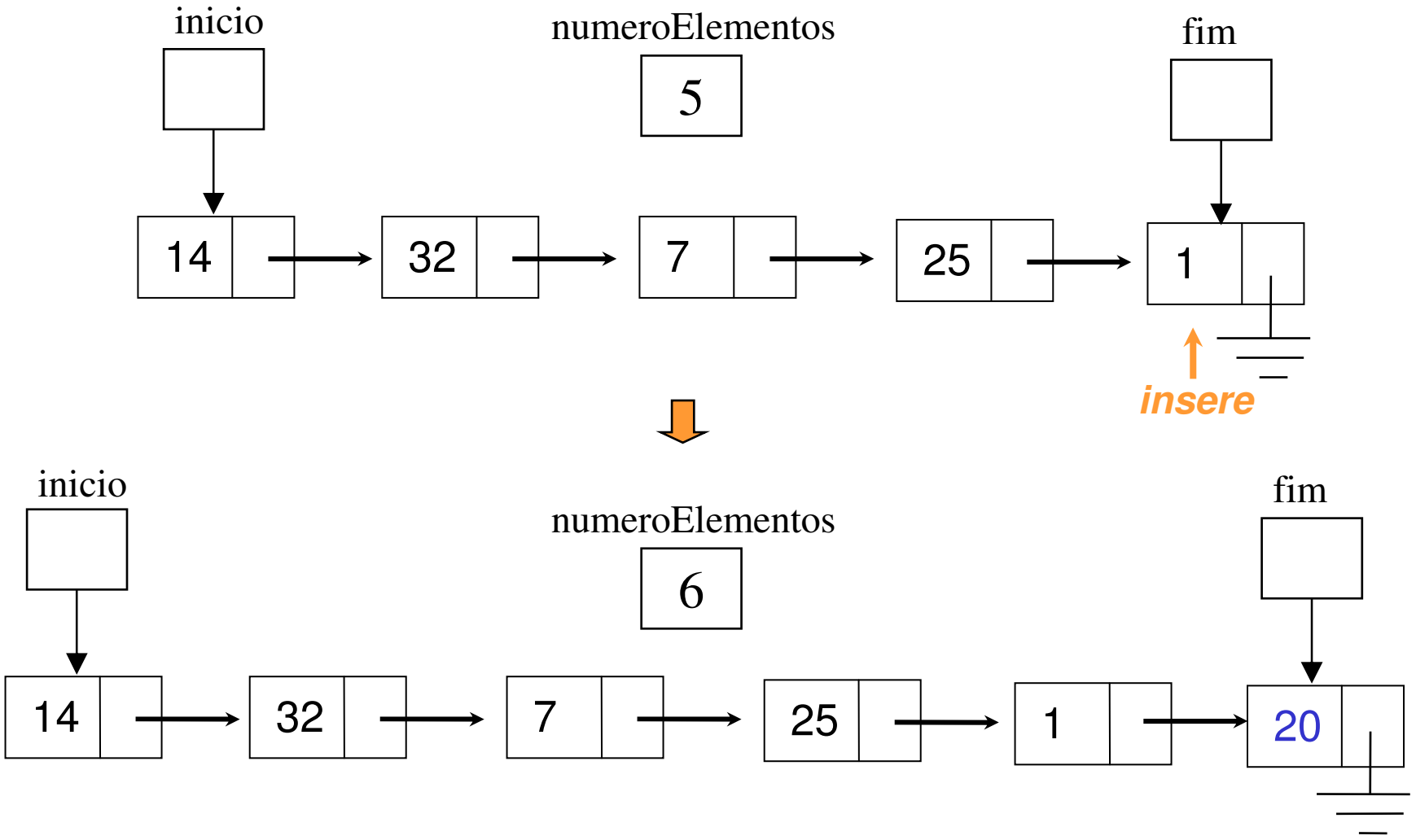
Onde deve ficar a entrada e a saída da fila?

Fila Encadeada - Construtor

```
public class FilaEncadeada<E> implements Fila<E> {
   private Nodo inicio;   // o inicio é a saída
   private Nodo fim;      // o fim é a entrada
   private int numElementos;

public FilaEncadeada() {
   this.inicio = null;
   this.fim = null;
   this.numElementos = 0;
}

…
}
```

Complexidade: O(1)

Fila Encadeada - Insere



Fila Encadeada - Insere

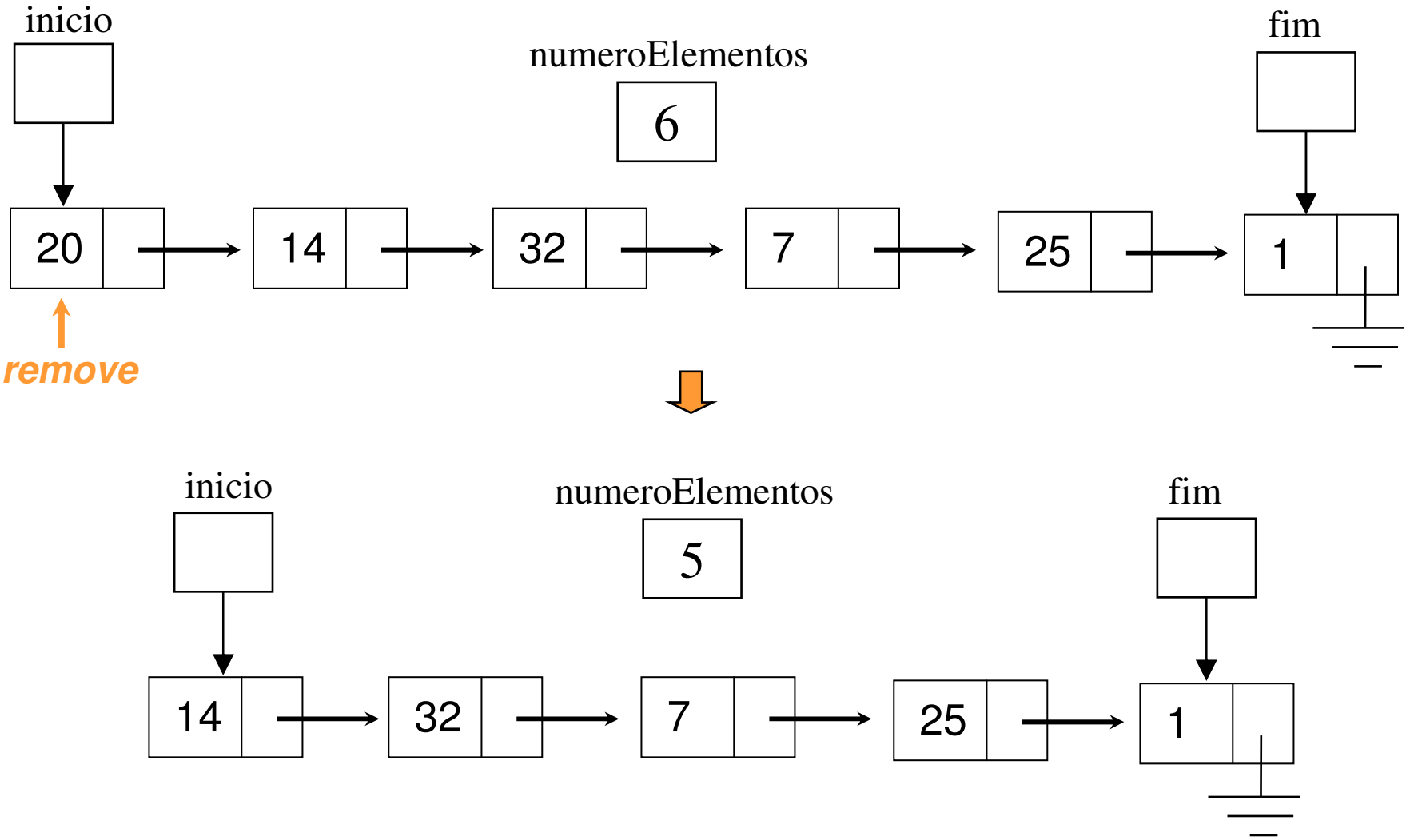
```
public void insere (E elemento) {

???

}
```

Fila Encadeada - Insere

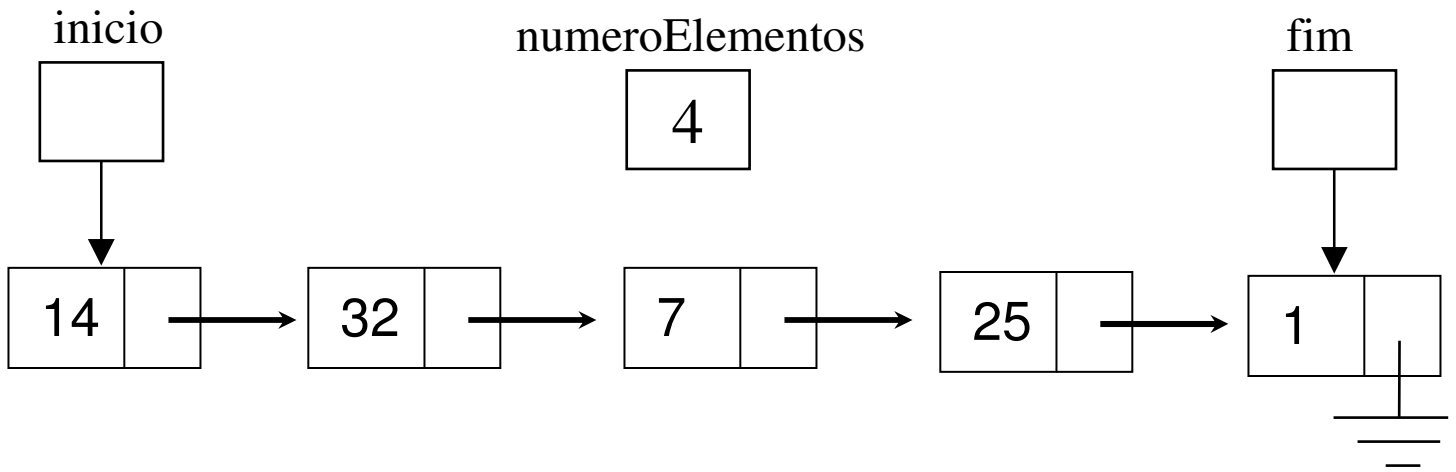
```
public void insere (E elemento) {
   Nodo nodo = new Nodo(elemento);
   if (this.fim == null) {
        this.inicio = nodo;
        this.fim = nodo; }
   else {
        this.fim.atribuiProximo(nodo);
        this.fim = nodo;  }
   numElementos++;
}
```

Complexidade: O(?)

Fila Encadeada - Remove



Fila Encadeada - Retorna Início



Início = 14