

Universidade Federal da Paraíba

Centro de Informática

Departamento de Informática

Estrutura de Dados

Filas

▶ Tiago Maritan

▶ tiago@ci.ufpb.br

Conteúdos Abordados

- ▶ O Conceito de Fila
- ▶ Filas com Representação Sequencial
- ▶ Filas com Representação Dinâmica

Filas

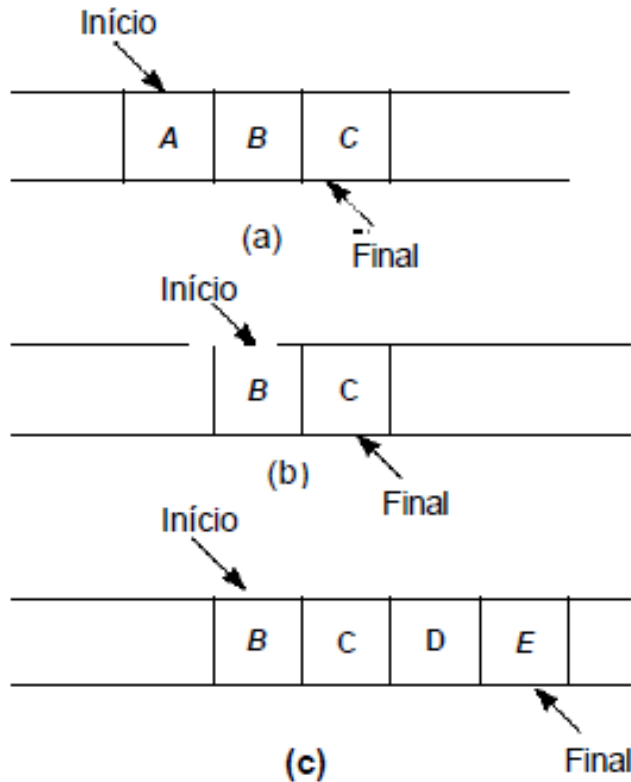
- ▶ Listas que possuem a seguinte característica:
 - ▶ **Inserção** é sempre feita no **final da fila** (lista)
 - ▶ **Remoção** é feita no **início da fila**.
- ▶ Analogia: filas do nosso dia-a-dia
 - ▶ Fila de banco, fila de carros num pedágio, etc.
- ▶ **FIFO: First In First Out**



Aplicações de Filas

- ▶ Escalonamento de "Jobs":
 - ▶ Fila de processos aguardando os recursos do SO;
- ▶ Fila de pacotes a serem transmitidos numa rede de comutação de pacotes.
- ▶ Fila de impressão

Operações em uma fila



- ▶ `insere(q, A);`
- ▶ `insere(q, B);`
- ▶ `insere(q, C);`

- ▶ `x = retira (q);`

- ▶ `insere(q, D);`
- ▶ `insere(q, E)`

Operações com Filas

▶ Interface:

- ▶ Criar uma fila vazia;
- ▶ Testar se a fila está vazia;
- ▶ Verificar se a fila está cheia;
- ▶ Obter o tamanho da fila;
- ▶ Consultar o elemento da frente da fila;
- ▶ Inserir um novo elemento no fundo da fila;
- ▶ Remover o elemento da frente da fila.

Filas com Representação Sequencial

Implementação de Filas Sequenciais

- ▶ Fila sequencial é uma estrutura com:
 - ▶ **Um array (vetor)** de elementos;
 - ▶ **Dois índices** que representam o **início** e **fim** da fila;

```
public class FilaSeq {  
    private int dados[];  
    private int inicio;  
    private int fim;  
  
    private int nElementos;  
  
    ...  
}
```


Implementação de Filas Sequenciais

- ▶ Problemas na implementação sequencial:
 - ▶ Quando índice **fim** possui o valor $MAX - 1$ não implica, necessariamente, que haja $MAX - 1$ elementos na fila.
 - ▶ Isto só é verdade, se índice **início** possuir o valor zero;
 - ▶ **início** da fila move-se para a direita a cada retirada de elemento.
- ▶ Solução:
 - ▶ Implementar como uma **fila circular**
 - Reutilizando as posições já ocupadas pela fila;

Implementação de Filas Sequenciais

```
public FilaSeq() {
    inicio = 0;
    fim = -1;
    nElementos = 0;

    tamMax = 100;
    dados = new int[tamMax];
}

/** Verifica se a Fila está vazia */
public boolean vazia () {
    if (nElementos == 0)
        return true;
    else
        return false;
}
```

Implementação de Filas Sequenciais

```
/**Verifica se a Fila está cheia */  
public boolean cheia () {  
    if (nElementos == tamMax)  
        return true;  
    else  
        return false;  
}  
  
/** Obtém o tamanho da Fila */  
public int tamanho() {  
    return nElementos;  
}
```

Implementação de Filas Sequenciais

```
/** Consulta o elemento do início da fila.  
    Retorna -1 se a fila estiver vazia. */  
public int primeiro() {  
    if (vazia())  
        return -1; // Erro: Fila vazia  
  
    return dados[inicio];  
}  
  
// continua...
```

Implementação de Filas Sequenciais

```
/**Insere um elemento no fim de uma fila
  Retorna false se a fila estiver cheia,
  true caso contrário. */
public boolean insere(int valor) {
    if (cheia()) {
        return false;
    }

    fim = (fim + 1) % tamMax; // Circularidade
    dados[fim] = valor;

    nElementos++;
    return true;
}
```

Implementação de Filas Sequenciais

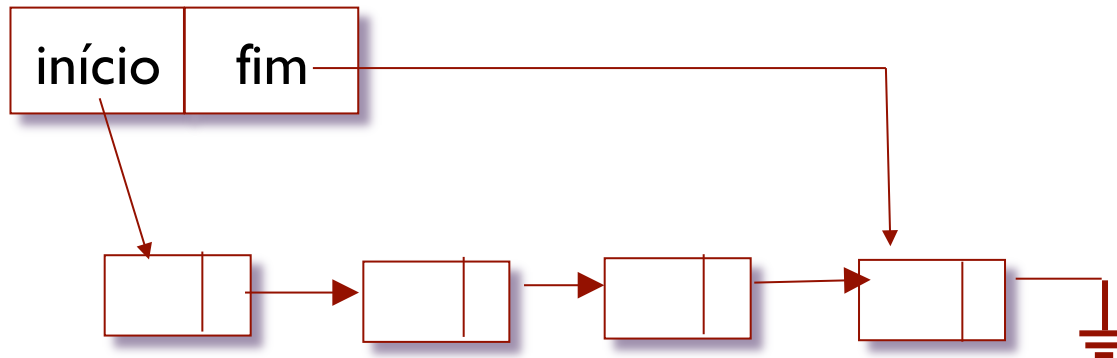
```
/** Remove o elemento do início da fila e
    Retorna o valor removido.
    Retorna -1 se a fila estiver vazia.*/
public int remove() {
    if (vazia())
        return -1;

    int res = primeiro();
    inicio = (inicio + 1) % tamMax; //Circularidade
    nElementos--;
    return res;
}
```

Filas com Representação Dinâmica

Implementação de Filas com Representação Dinâmica

- ▶ Implementa como uma **lista com duas cabeças**
 - ▶ Uma cabeça aponta para o início da fila
 - ▶ Outra aponta para o fim;



Implementação de Filas Encadeadas

```
public class No{
    private int conteudo;
    private No prox;

    public No(){
        prox = null;
    }

    // Métodos get e set
}
```

```
public class Fila {
    private No inicio;
    private No fim;
    private int nElementos;

    public Fila(){
        inicio = null;
        fim = null;
        nElementos = 0;
    }
}
```

Implementação de Filas Encadeadas

```
/**Verifica se a Fila está vazia */  
public boolean vazia () {  
    if (nElementos == 0)  
        return true;  
    else  
        return false;  
}  
  
/** Obtém o tamanho da Fila */  
public int tamanho () {  
    return nElementos;  
}
```

Implementação de Filas Encadeadas

```
/** Consulta o elemento do início da fila  
Retorna -1 se a fila estiver vazia */  
public int primeiro () {  
    if (vazia())  
        return -1; // Erro: Fila vazia  
  
    return inicio.getConteudo();  
}
```

```
/** Insere um elemento no fim de uma fila
    Retorna true se a insercao funciona e
    false se e true caso contrario*/
public boolean insere (int valor) {
    No novoNo = new No();
    novoNo.setConteudo(valor);
    novoNo.setProx(null);

    if (vazia()) {    //Inserção em fila vazia
        inicio = novoNo;
        fim = novoNo;
    }
    else {
        fim.setProx(novoNo); // liga com a fila
        fim = novoNo; // atualiza o novo fim
    }
    nElementos++;
    return true;
}
```

Implementação de Filas Encadeadas

```
/**Retira o elemento do início da fila.  
Retorna -1 se a fila estiver vazia. */  
public int remove() {  
    if (vazia())  
        return -1; // Erro: Fila vazia  
  
    int valor = primeiro();  
    if (inicio == fim) // Fila com 1 elemento  
        fim = null;  
  
    No p = inicio;  
    inicio = p.getProx();  
    nElementos--;  
  
    p= null;  
    return valor;  
  
}
```

Universidade Federal da Paraíba

Centro de Informática

Departamento de Informática

Estrutura de Dados

Filas

▶ Tiago Maritan

▶ tiago@ci.ufpb.br

