Estruturas de Dados: **Pilhas, Filas e Deques**

Helena Graziottin Ribeiro
hgrib@ucs.br

UCS
UNIVERSIDADE
DE CAXIAS DO SUL

Pilhas, filas e deques: listas com restrições

Definição:

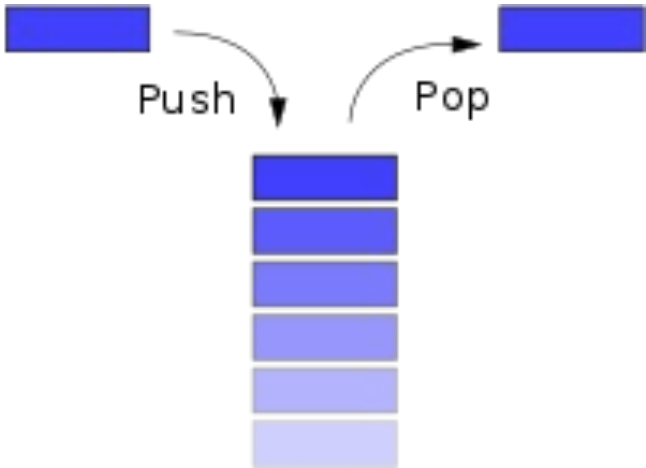
- formas restritas de **listas,** nas quais elementos só podem ser inseridos e removidos em posições pré-determinadas
- **listas** com políticas de acesso específicas

Pilhas

- inserções e remoções sempre no mesmo lado
- LIFO (*last in first out*): os elementos saem da pilha na ordem reversa de chegada

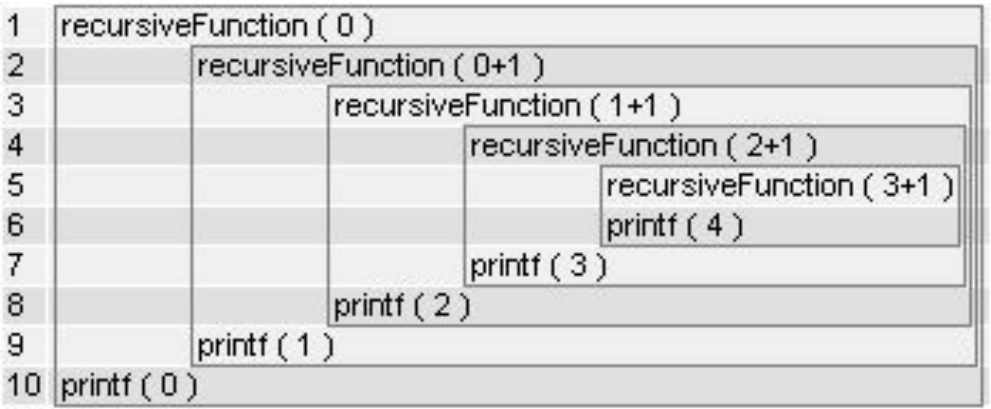
Pilhas

- Inserções e remoções de elementos sempre no **topo** da pilha.
- Operações:
  - Empilhar (***push***): inserir elemento no topo
  - Desempilhar (***pop***): retirar elemento do topo



Pilhas: aplicações

- Chamadas de funções: guardar endereço de retorno



Pilhas: aplicações

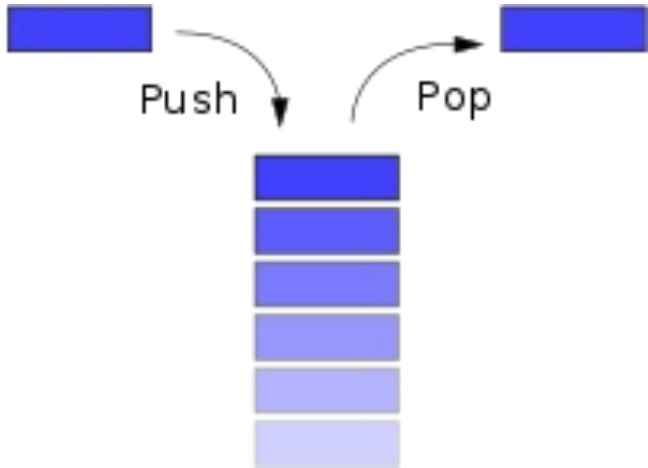
- Jogos: guardar caminho de volta

Pilhas: aplicações

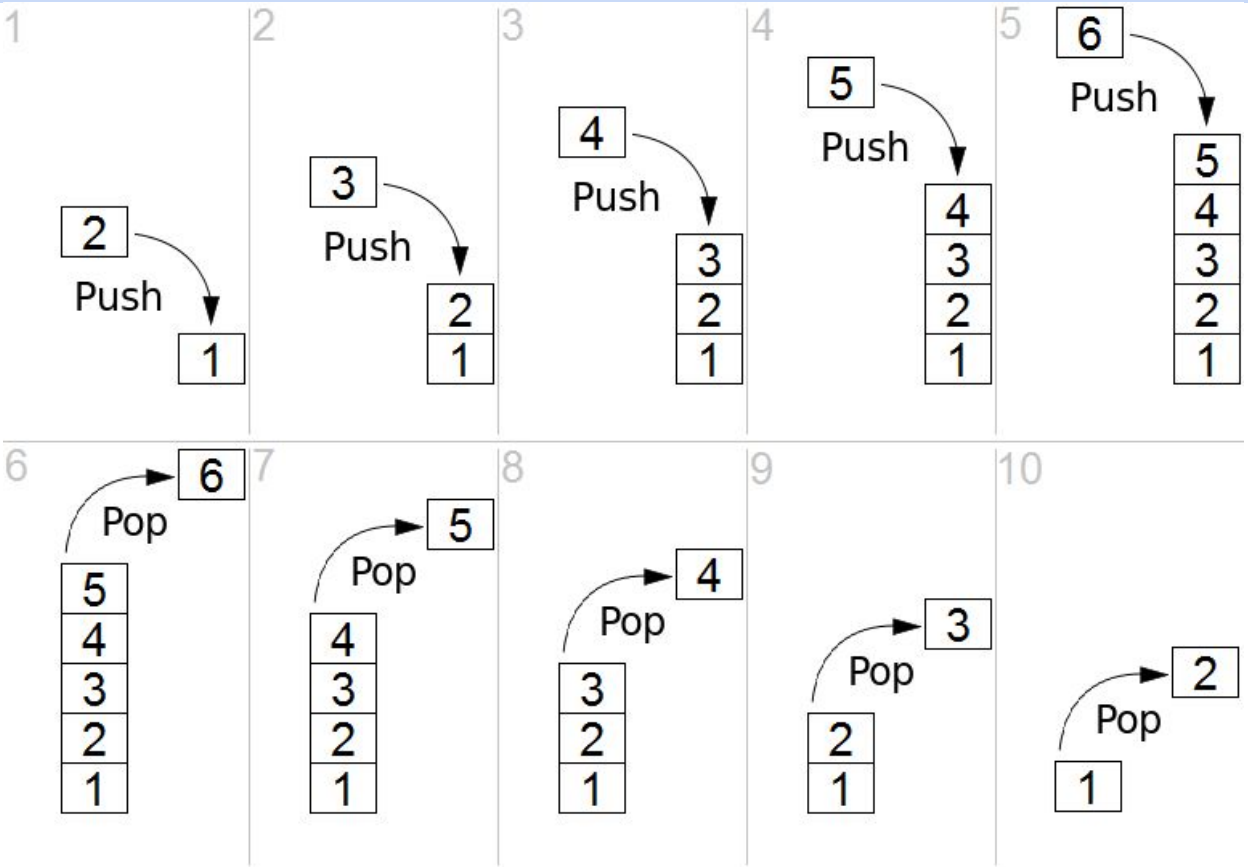
- aplicativos: desfazer comandos

Pilhas

```
TAD Pilha {
Dados: topo
Operações:   novaPilha ()
        empilha(E: elemento)
        desempilha(S: elemento)
        vazia(S: lógico)
}
```

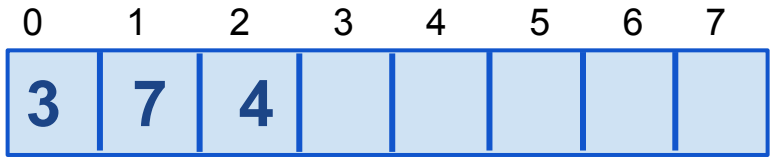


Pilhas: implementação



Pilhas: implementação

Por vetores:

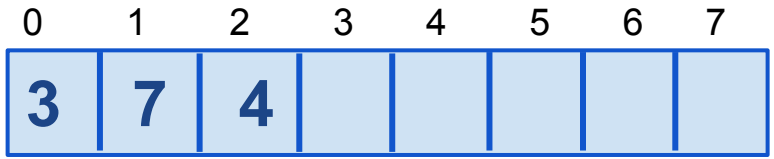


topo = 2

Pilhas: implementação

Por vetores:

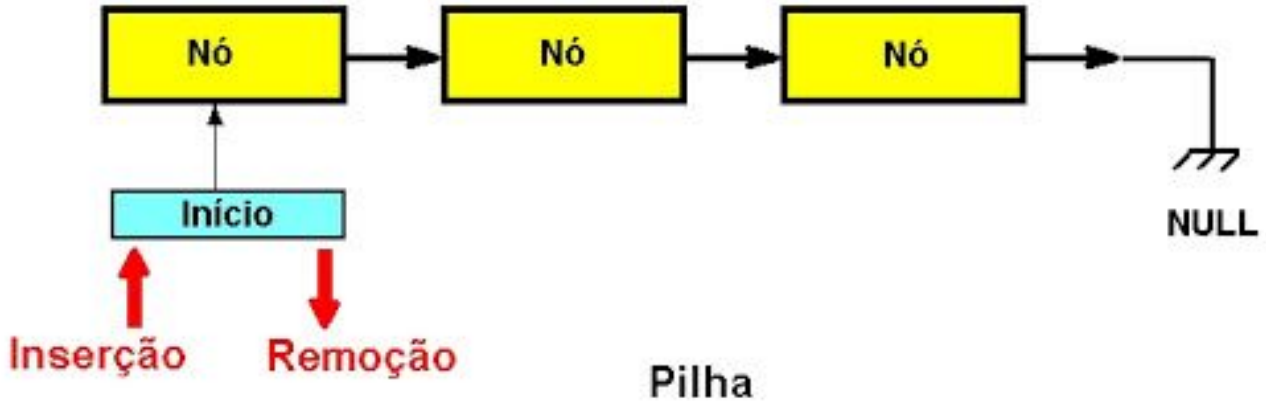
- simplificação de listas baseadas em vetores
  - insere no fim, retira do fim
- topo = primeira posição livre (ou, última ocupada)



topo = 2

Pilhas: implementação

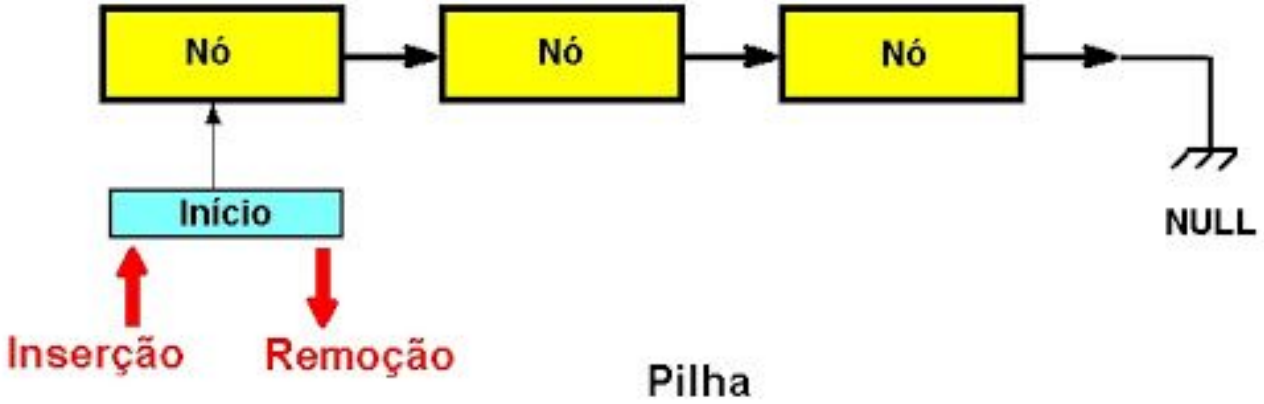
Por listas encadeadas:



Pilhas: implementação

Por listas encadeadas:

- simplificação de listas baseadas em vetores
  - insere sempre no início, retira do início
- topo = inicio

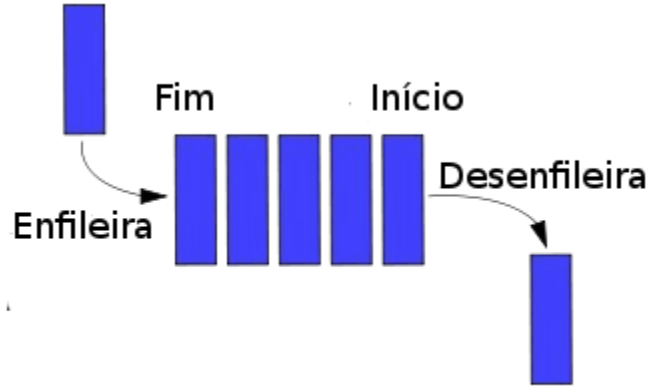


Filas

- inserções no fim e remoções no início
- FIFO (*first in first out*): os elementos saem da fila na ordem de chegada

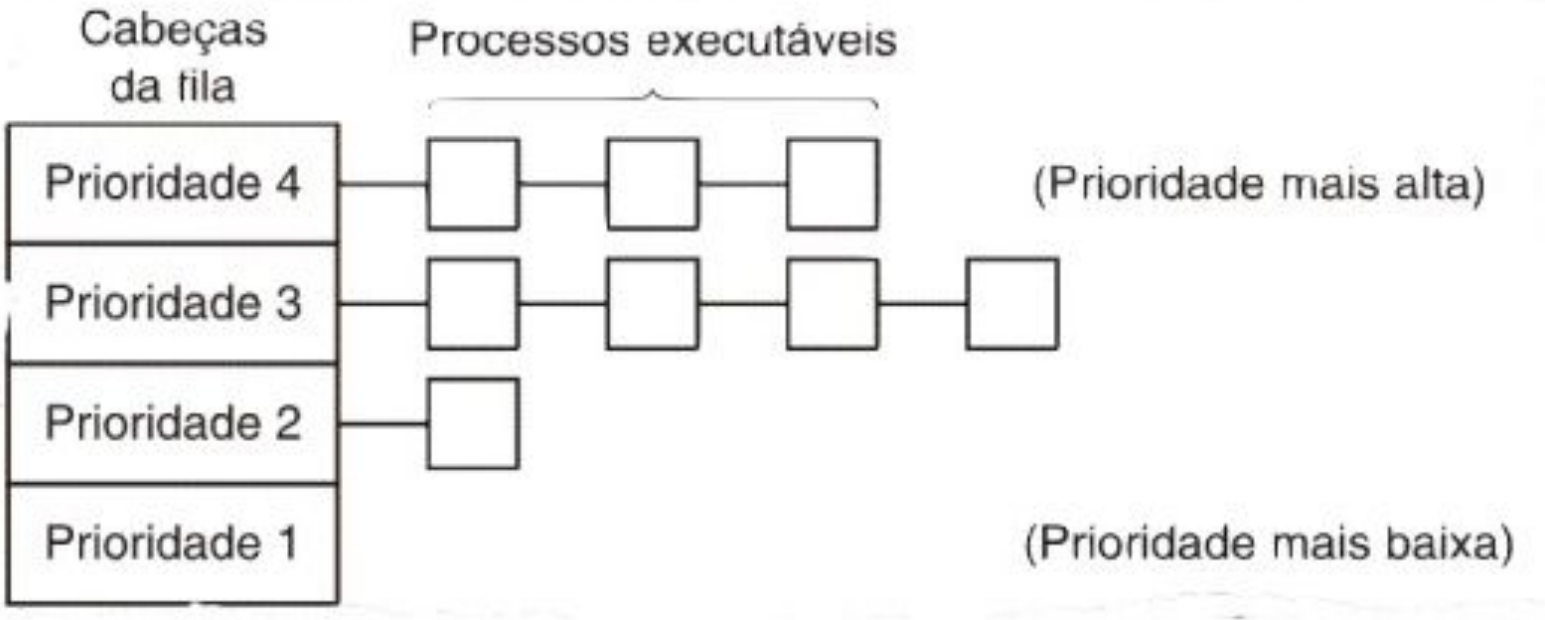
Filas

- **Inserções no final**, e **remoções no início** da fila.
- Operações:
  - **Enfileirar**: inserir elemento no final
  - **Desenfileirar**: retirar elemento do início



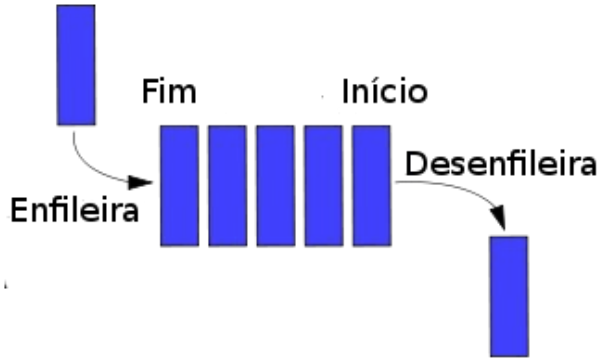
Filas: aplicações

- ordem de execução de processos

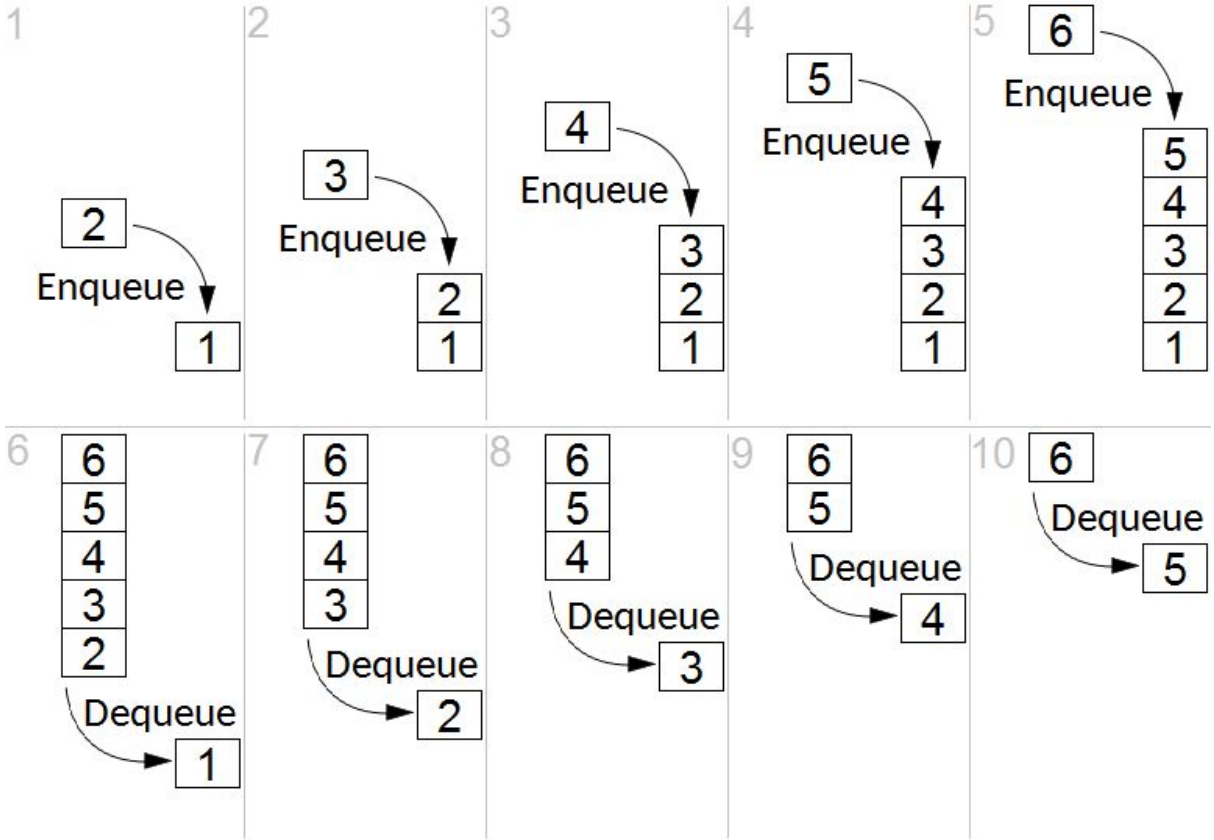


Filas

```
TAD Fila {
Dados: primeiro, último
Operações:   novaFila ()
        enfileira(E: elemento)
        desenfileira(S: elemento)
        vazia(S: lógico)
}
```



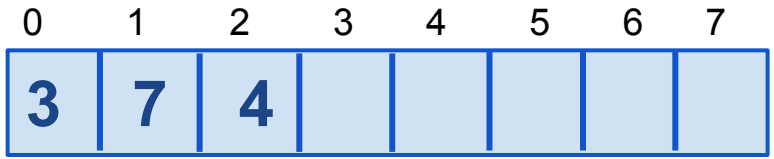
Filas


1
2
Enqueue
1
2
3
Enqueue
2
1
3
4
Enqueue
3
2
1
4
5
Enqueue
4
3
2
1
5
6
Enqueue
5
4
3
2
1
6
6
5
4
3
2
Dequeue
1
7
6
5
4
3
Dequeue
2
8
6
5
4
Dequeue
3
9
6
5
Dequeue
4
10
6
Dequeue
5

Filas: implementação

Por vetores:

- simplificação de listas baseadas em vetores
  - insere no fim, retira do início
- acesso sempre no inicio e no fim

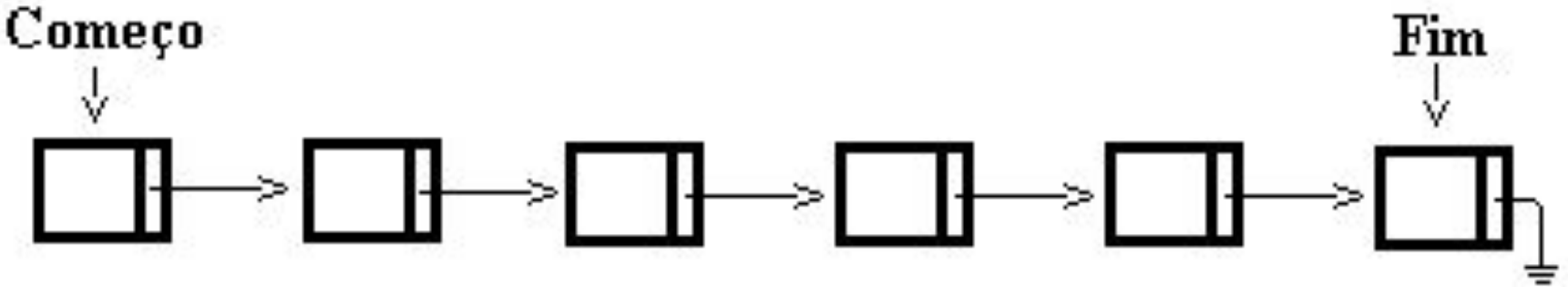


primeiro = 0 **último = 2**

Filas: implementação

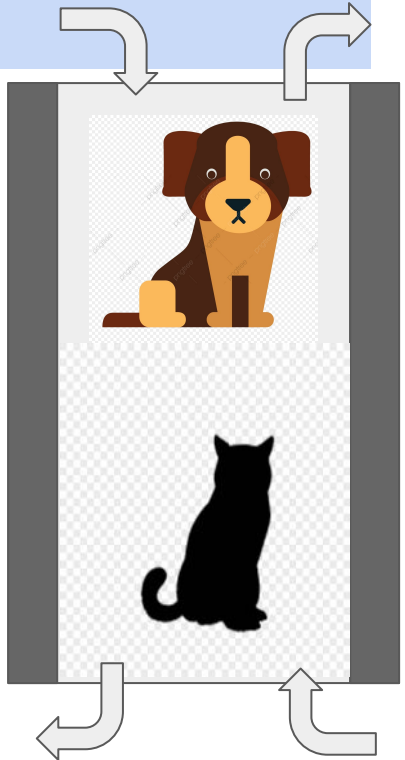
Por listas encadeadas:

- simplificação de listas encadeadas simples
  - insere no fim, retira do início
- primeiro=inicio, último=fim



Deques (ou dequeue)

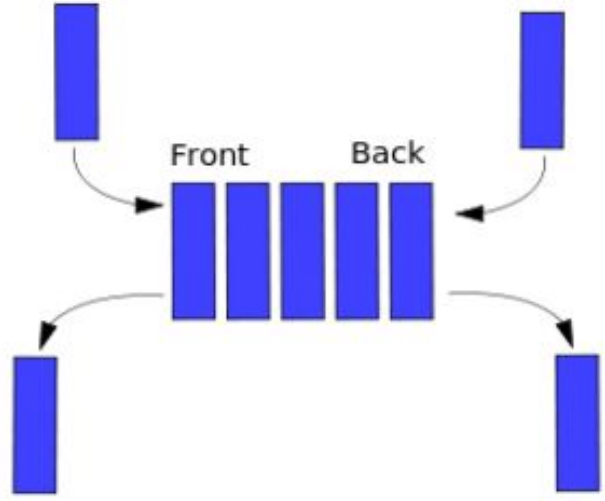
- ***D****ouble* ***E****nded* ***QUE****ue*
- inserções e remoções nas extremidades
- deque geral, deque de entrada restrita, deque de saída restrita



Deques

Definição:

- formas restritas de **listas,** nas quais elementos só podem ser acessados, inseridos e retirados no início e/ou no final da lista

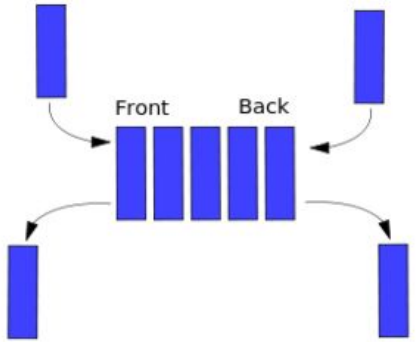


Deque geral

```
TAD Deque{

Dados: ladoA, ladoB

Operações:    novaDeque ()

        incluiInicio(E: elemento)

        incluiFinal(E: elemento)

        excluiInicio(S: elemento)

        excluiFinal(S: elemento)

        vazia(S: lógico)

}
```

Deques com restrições de acesso

- **deque com entrada restrita:** inserções só podem ser realizadas em uma das extremidades e remoções nas duas
- **deque com saída restrita:** inserções podem ser realizadas nas duas extremidades e remoções apenas em uma



Deque entrada restrita

```
TAD DequeEntradaRestrita{

Dados: ladoA, ladoB

Operações:    novaDequeEntradaRestrita ()

        inclui(E: elemento)

        excluiInicio(S: elemento)

        excluiFinal(S: elemento)

        vazia(S: lógico)

}
```

Deque saída restrita

```
TAD DequeSaidaRestrita{

Dados: ladoA, ladoB

Operações:    novaDequeSaida Restrita ()

        incluiInicio(E: elemento)

        incluiFinal(E: elemento)

        exclui(S: elemento)

        vazia(S: lógico)

}
```

Deques: implementações

- Por vetores
- Por listas encadeadas:
  - listas encadeadas simples ou duplas?