

# Estruturas lineares: pilhas e filas

Prof. Paulo Henrique Pisani

março/2022

# Tópicos

- Pilha:
  - Pilha estática;
  - Pilha dinâmica (com lista ligada).
- Fila:
  - Fila estática;
  - Fila dinâmica (com lista ligada).

Pilha

# Pilha

- **Pilha (*stack*)**: estrutura de dados que adota a estratégia **LIFO** (*Last In First Out*): último a entrar é o primeiro a sair;
- Operações básicas:
  - Empilhar / *Push* (inserção);
  - Desempilhar / *Pop* (remoção).

Pilha estática

# Pilha estática

- **Pilha estática**: implementa a estrutura de dados utilizando um **arranjo**;
- Portanto, os itens são armazenados em posições consecutivas na memória.

# Pilha estática

- Estrutura básica:

Pilha:

**itens** (referência para um  
vetor/arranjo)

**tamanho** (inteiro)

**topo** (inteiro)

Referência para  
o vetor alocado

Tamanho do  
vetor alocado

Índice do elemento no  
topo da pilha

# Pilha estática

- Inicialização

(topo == -1) ==> Pilha vazia!

itens=alocar vetor com tamanho 4

tamanho=4

topo=-1



0

1

2

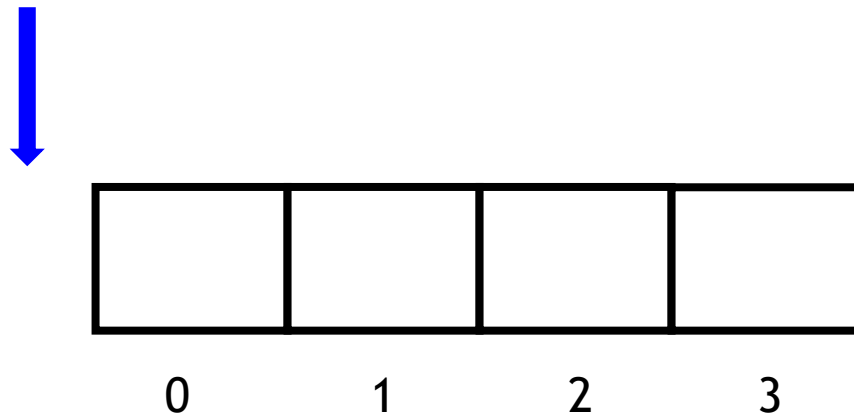
3



# Pilha estática

- Como empilhar?
  - **topo==tamanho-1 ?**
    - Sim: pilha cheia!
    - Não: então podemos empilhar.

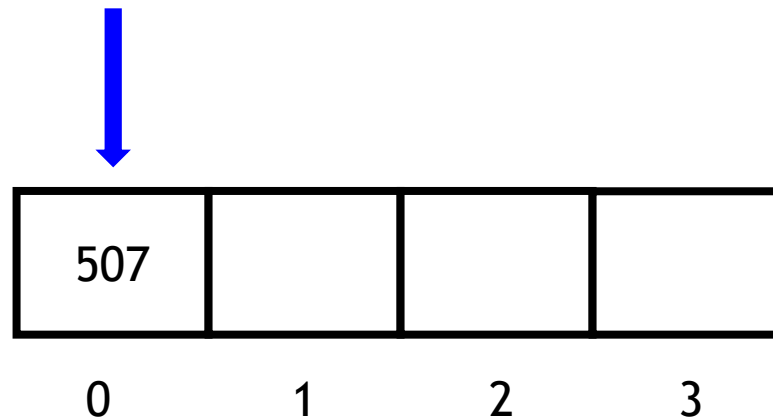
tamanho=4  
topo=-1



# Pilha estática

- Empilhar 507:
  - Verificar se pilha está cheia
  - `topo++`
  - `itens[topo]=507`

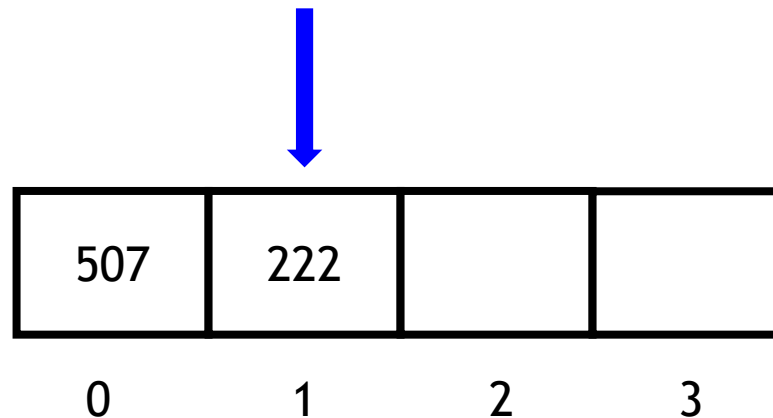
tamanho=4  
topo=0



# Pilha estática

- Empilhar 222:
  - Verificar se pilha está cheia
  - `topo++`
  - `itens[topo]=222`

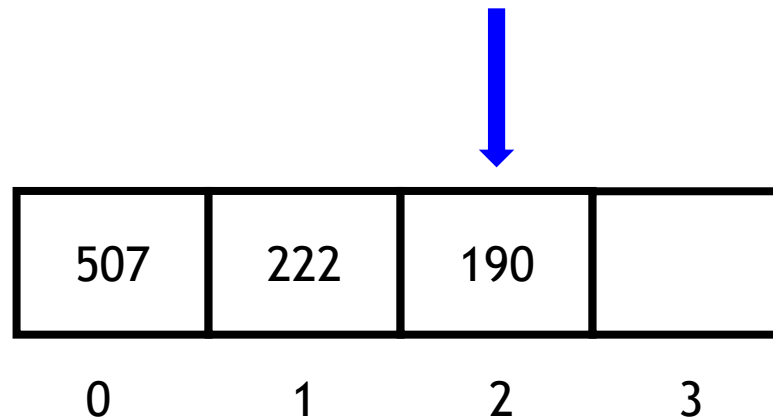
tamanho=4  
topo=1



# Pilha estática

- Empilhar 190:
  - Verificar se pilha está cheia
  - `topo++`
  - `itens[topo]=190`

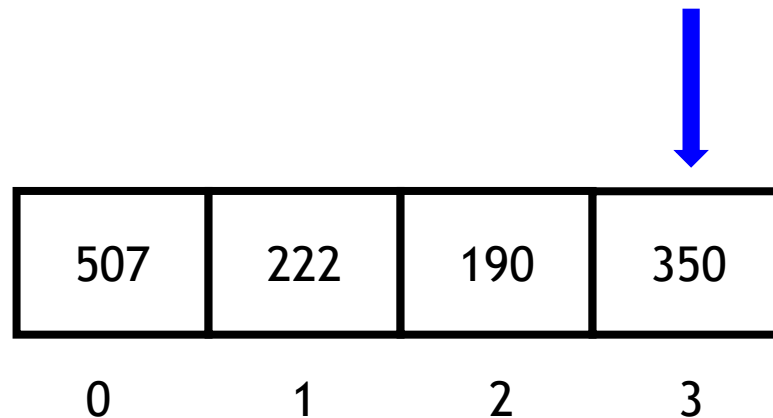
tamanho=4  
topo=2



# Pilha estática

- Empilhar 350:
  - Verificar se pilha está cheia
  - `topo++`
  - `itens[topo]=350`

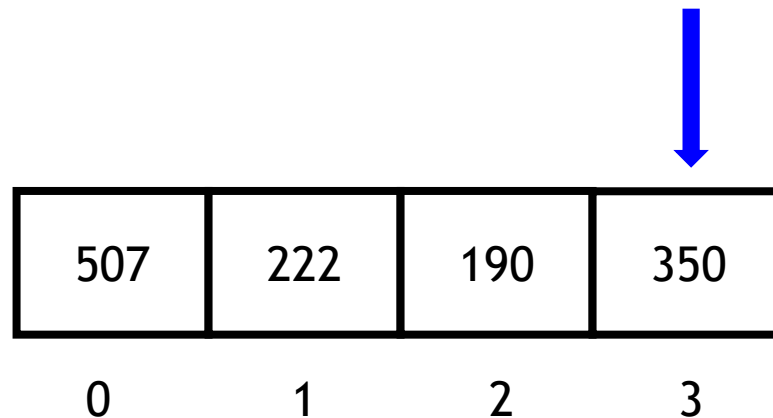
tamanho=4  
topo=3



# Pilha estática

- Empilhar 500:
  - **Pilha está cheia!**
  - **topo==tamanho-1**

tamanho=4  
topo=3



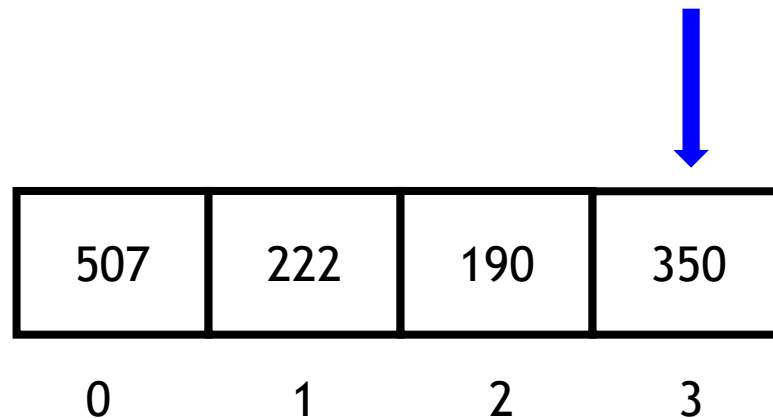
# Pilha estática

- Como desempilhar?

- **topo != -1?**

- Sim: podemos desempilhar
    - Não: pilha vazia!

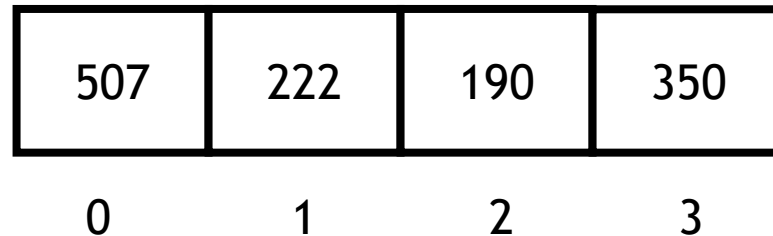
tamanho=4  
topo=3



# Pilha estática

- Desempilhar
  - Verificar se pilha está vazia
  - Salvar itens[topo]
  - topo--
  - Retonar item salvo

tamanho=4  
topo=3





# Pilha estática

- Desempilhar

Não

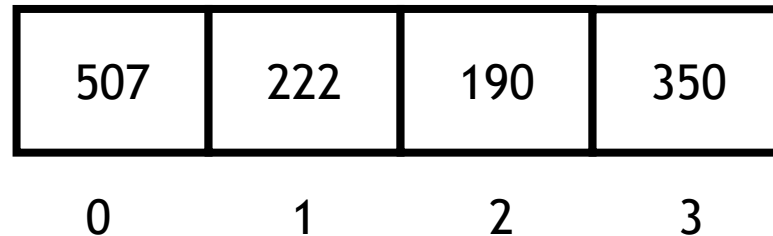
- Verificar se pilha está vazia

350

- Salvar itens[topo]
- topo--
- Retonar item salvo

tamanho=4

topo=3



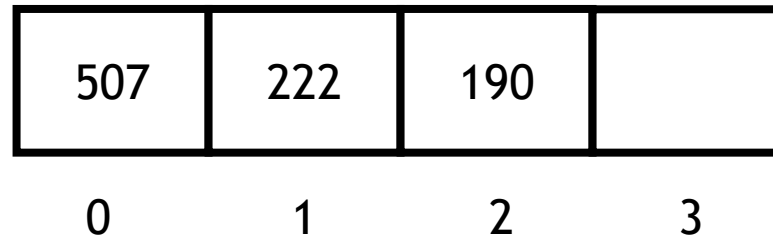
# Pilha estática

- Desempilhar
  - Verificar se pilha está vazia
  - Salvar itens[topo]
  - topo--
  - Retonar item salvo

2

350

tamanho=4  
topo=2



# Pilha estática

- Desempilhar

Não

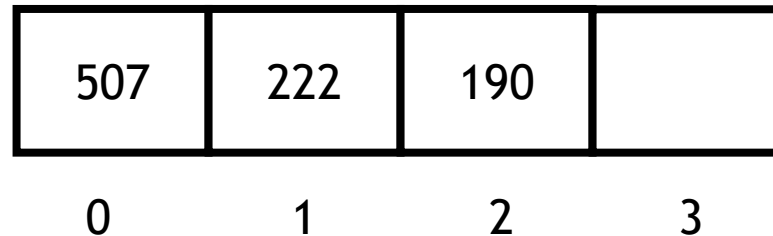
- Verificar se pilha está vazia

190

- Salvar itens[topo]
- topo--
- Retonar item salvo

tamanho=4

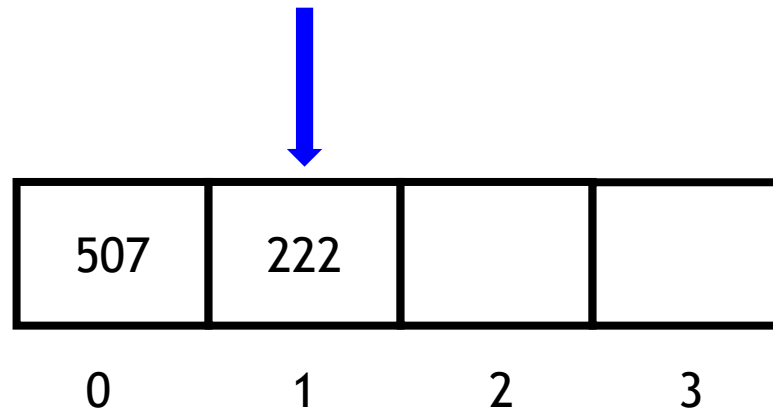
topo=2



# Pilha estática

- Desempilhar
  - Verificar se pilha está vazia
  - Salvar itens[topo]
  - topo--
  - Retonar item salvo

tamanho=4  
topo=1



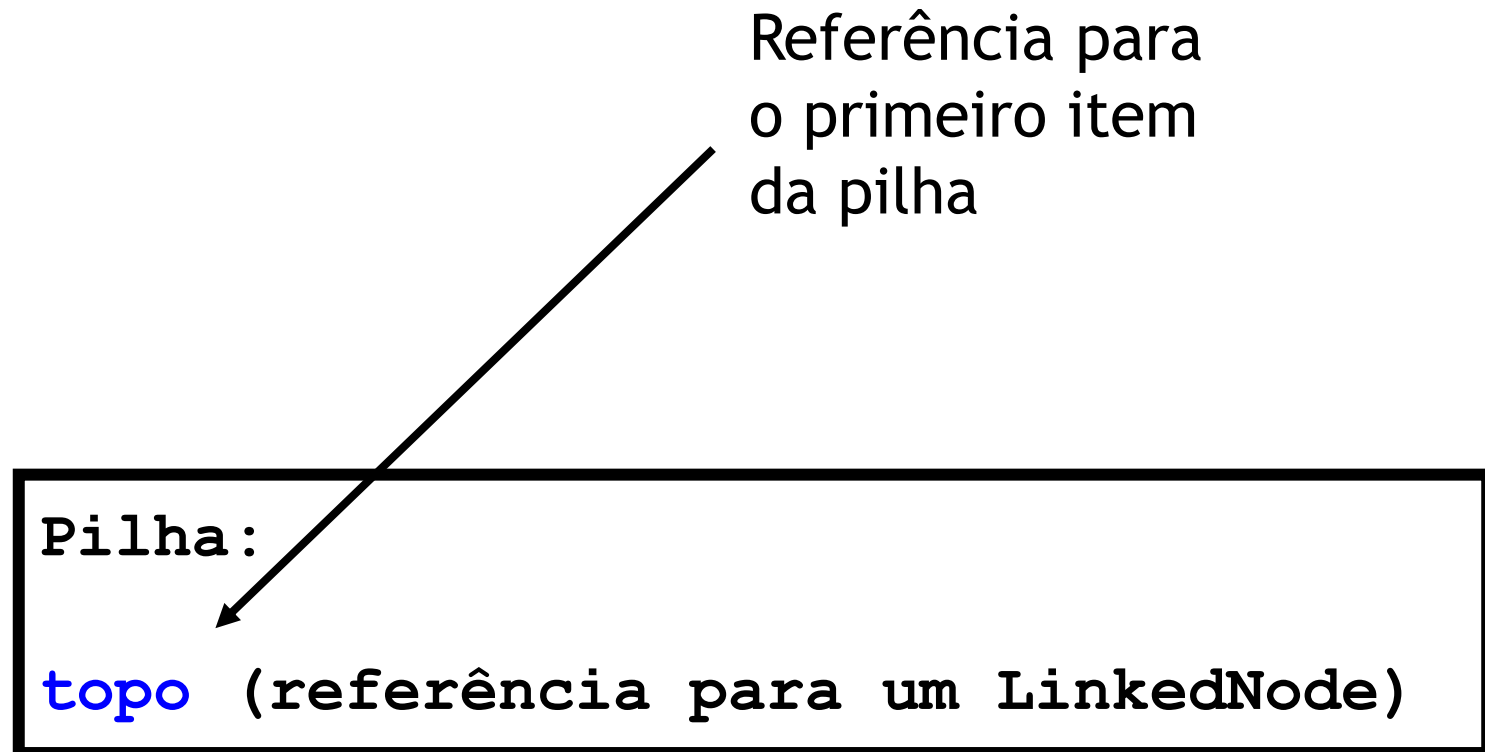
Pilha dinâmica

# Pilha dinâmica

- **Pilha dinâmica**: implementa a estrutura de dados utilizando uma **lista ligada**;
- Portanto, os itens são alocados em memória de acordo com a necessidade.

# Pilha dinâmica

- Estrutura básica:



# Pilha dinâmica

- Inicialização

```
topo = NULL
```

(topo == NULL) ==> Pilha vazia!



# Pilha dinâmica

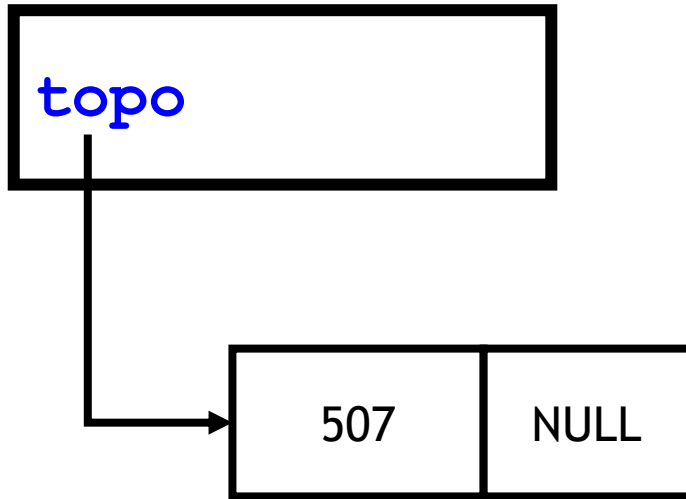
- Como empilhar?

`topo = NULL`

1. Alocar novo `LinkedListNode`
2. Adicioná-lo **ANTES** do item apontado por `topo`
3. Atualizar ponteiro `topo`

# Pilha dinâmica

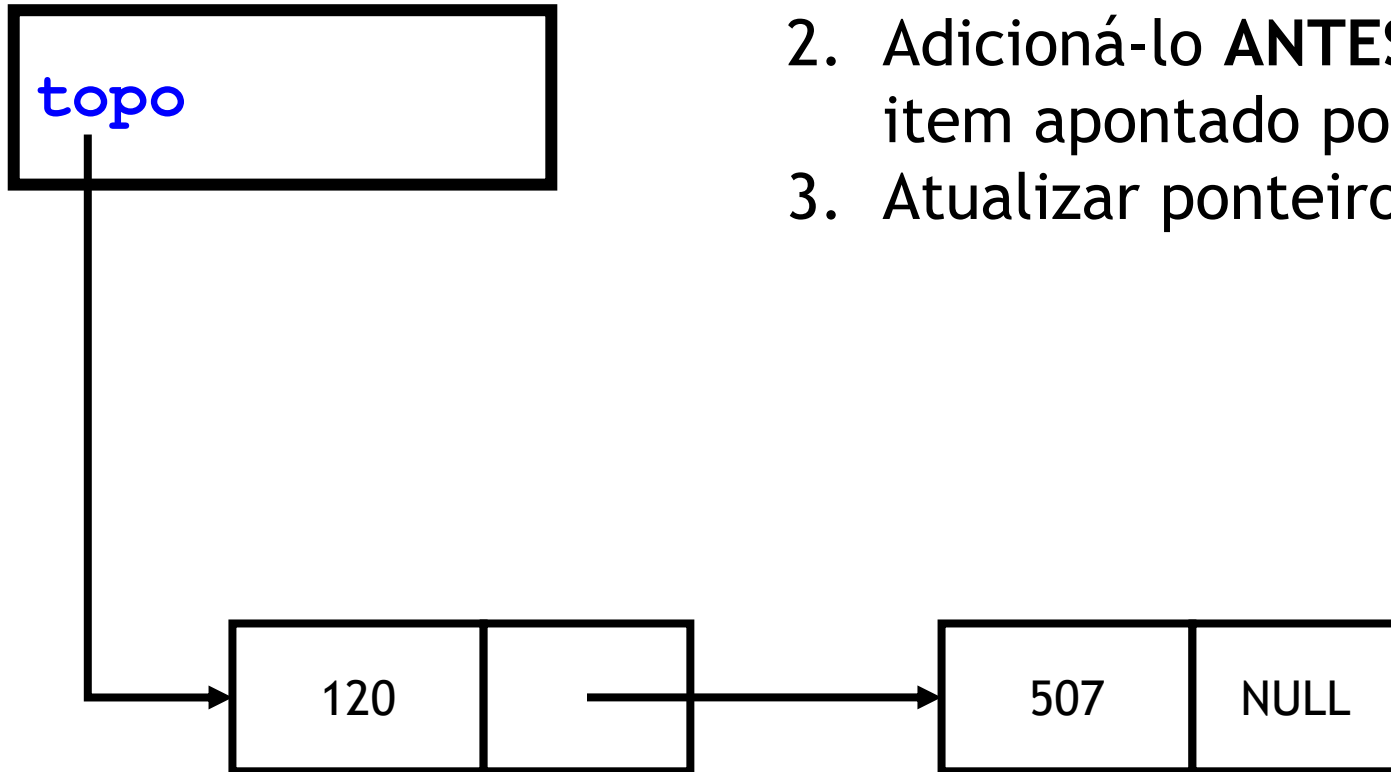
- Como empilhar?



1. Alocar novo ListNode
2. Adicioná-lo **ANTES** do item apontado por **topo**
3. Atualizar ponteiro **topo**

# Pilha dinâmica

- Como empilhar?



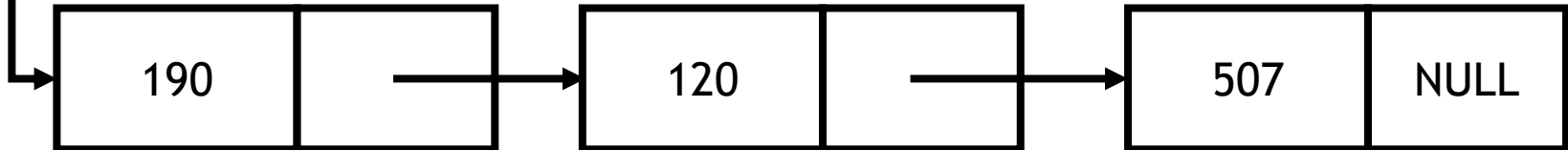
1. Alocar novo ListNode
2. Adicioná-lo **ANTES** do item apontado por **topo**
3. Atualizar ponteiro **topo**

# Pilha dinâmica

- Como empilhar?



1. Alocar novo ListNode
2. Adicioná-lo **ANTES** do item apontado por **topo**
3. Atualizar ponteiro **topo**

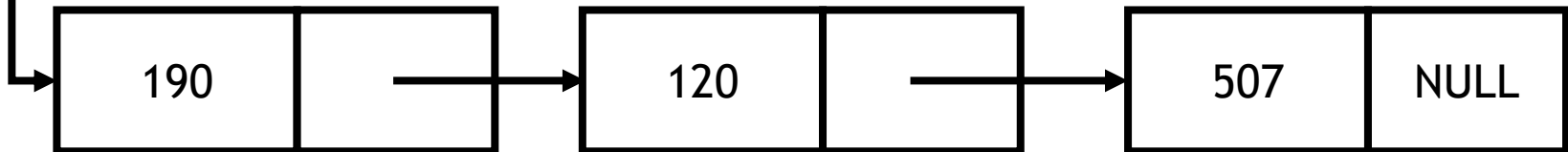


# Pilha dinâmica

- Como desempilhar?



1. Verificar se a pilha está vazia
2. Salvar item apontado por `topo`
3. `topo = topo->next`
4. Salvar valor do item
5. Liberar memória ocupada pelo item
6. Retornar valor

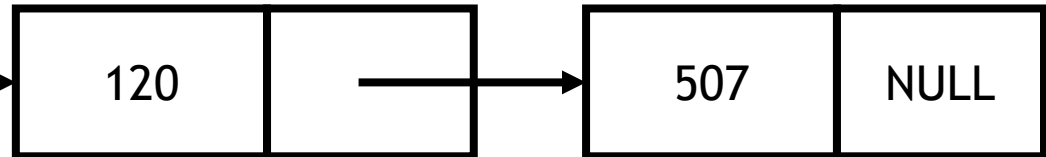


# Pilha dinâmica

- Como desempilhar?



1. Verificar se a pilha está vazia
2. Salvar item apontado por `topo`
3. `topo = topo->next`
4. Salvar valor do item
5. Liberar memória ocupada pelo item
6. Retornar valor

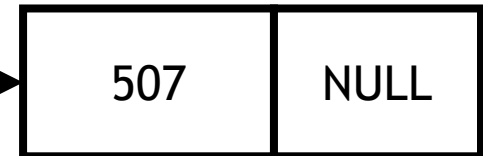


# Pilha dinâmica

- Como desempilhar?



1. Verificar se a pilha está vazia
2. Salvar item apontado por `topo`
3. `topo = topo->next`
4. Salvar valor do item
5. Liberar memória ocupada pelo item
6. Retornar valor



# Pilha dinâmica

- Como desempilhar?

`topo=NULL`

1. Verificar se a pilha está vazia
2. Salvar item apontado por `topo`
3. `topo = topo->next`
4. Salvar valor do item
5. Liberar memória ocupada pelo item
6. Retornar valor



Fila

# Fila

- **Fila (*queue*)**: estrutura de dados que adota a estratégia **FIFO** (*First In First Out*): primeiro a entrar é o primeiro a sair;
- Operações básicas:
  - Enfileirar/Enqueue (inserção);
  - Desenfileirar/Dequeue (remoção).

Fila estática

# Fila estática

- **Fila estática**: implementa a estrutura de dados utilizando um **arranjo**;
- Portanto, os itens são armazenados em posições consecutivas na memória.

# Fila estática

- Estrutura básica:

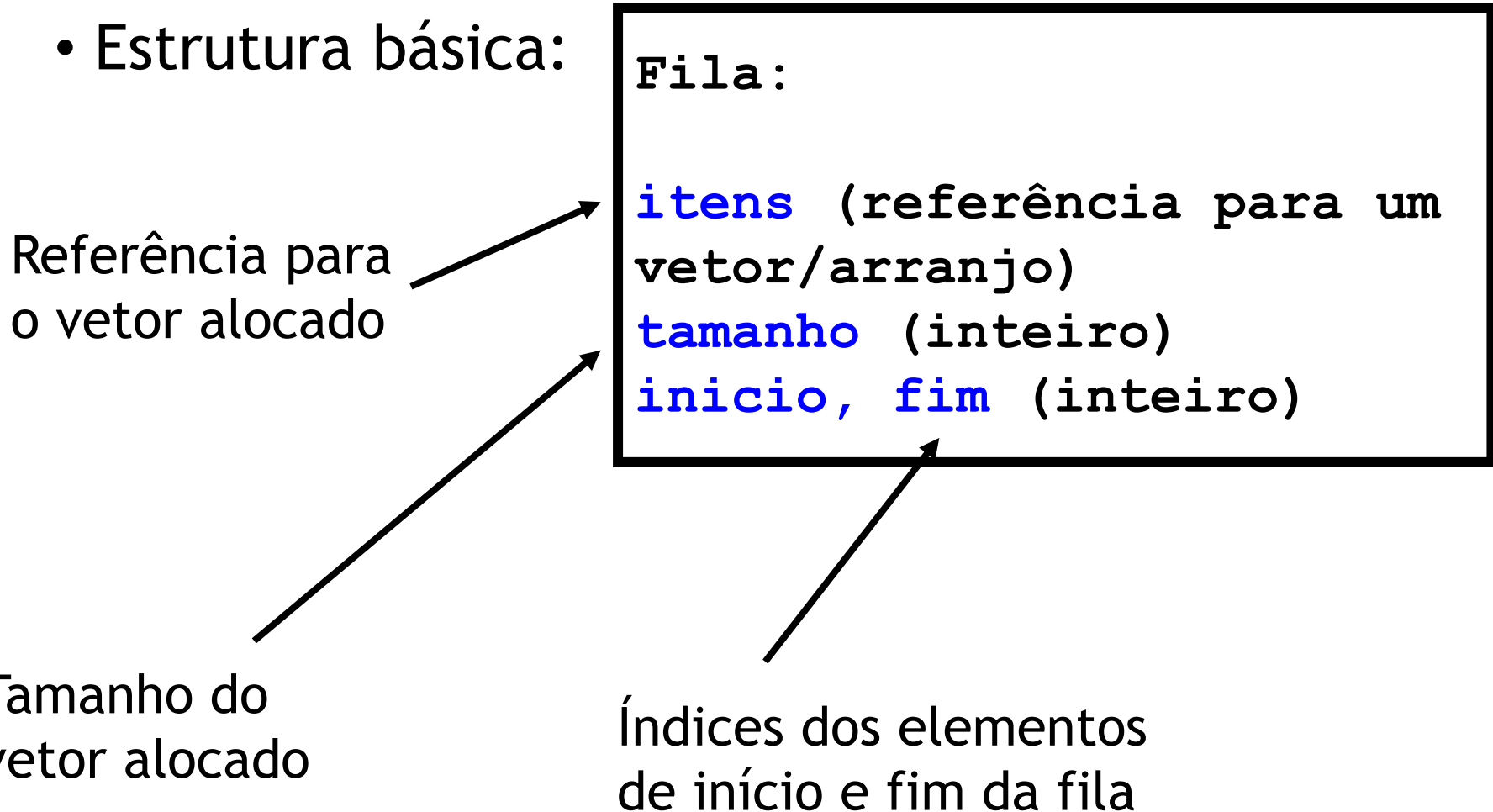
Fila:

`itens` (referência para um  
vetor/arranjo)

`tamanho` (inteiro)

`inicio, fim` (inteiro)

Referência para  
o vetor alocado



Tamanho do  
vetor alocado

Índices dos elementos  
de início e fim da fila

# Fila estática

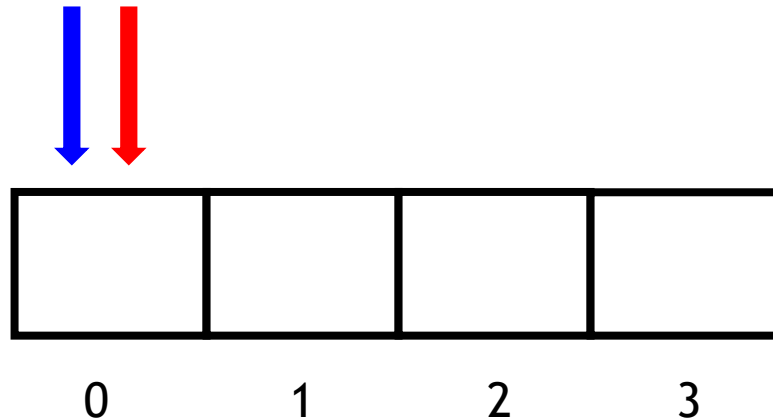
- Inicialização

`itens=alocar vetor com tamanho 4`

`tamanho=4`

`inicio=0`

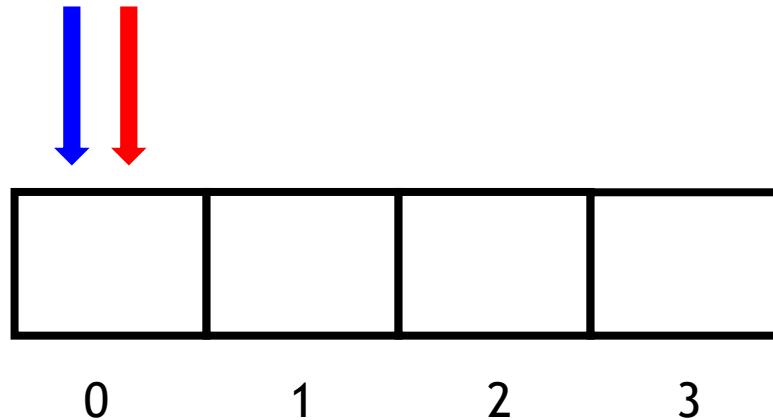
`fim=0`



# Fila estática

- Enfileirar o número 507

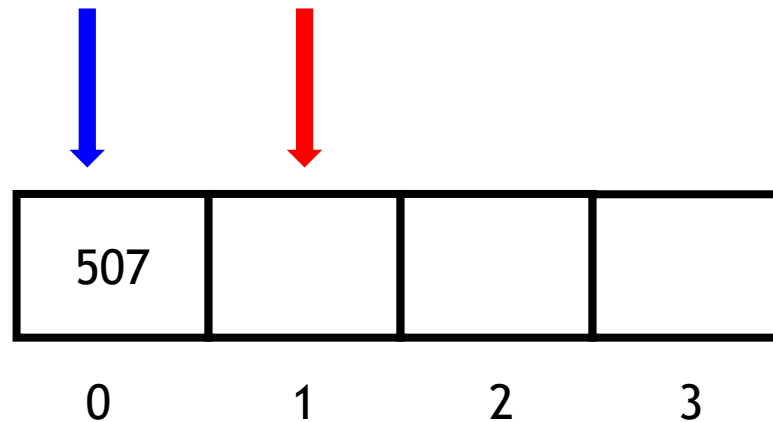
```
tamanho=4  
inicio=0  
fim=0
```



# Fila estática

- Enfileirar o número 507

```
tamanho=4  
inicio=0  
fim=1
```

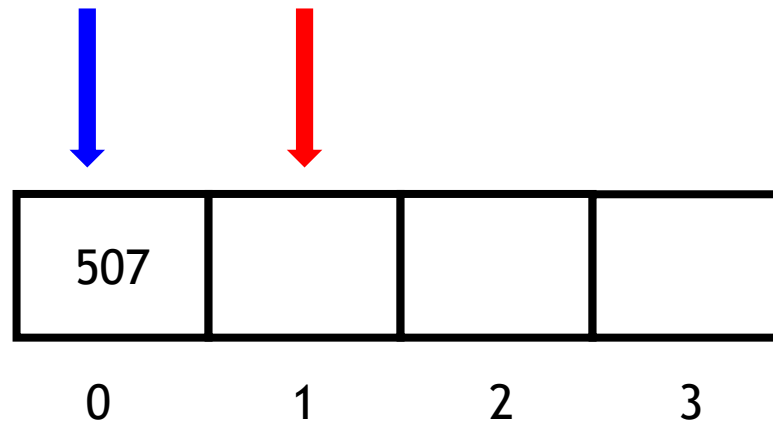




# Fila estática

- Enfileirar o número 222

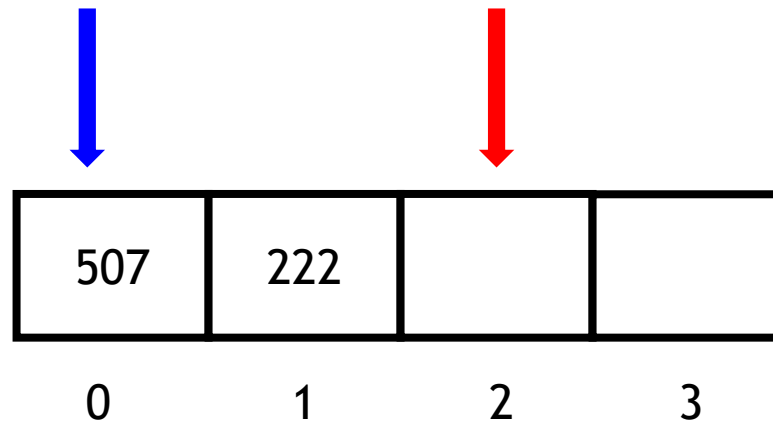
```
tamanho=4  
inicio=0  
fim=1
```



# Fila estática

- Enfileirar o número 222

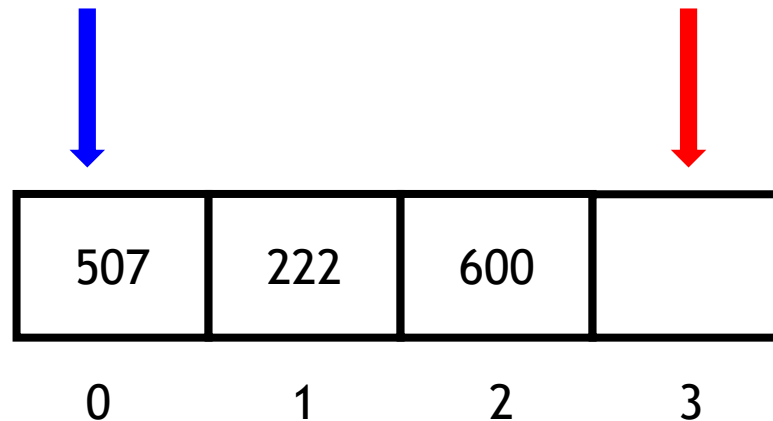
```
tamanho=4  
inicio=0  
fim=2
```



# Fila estática

- Enfileirar o número 600

```
tamanho=4  
inicio=0  
fim=3
```



# Fila estática

- Enfileirar o número 120



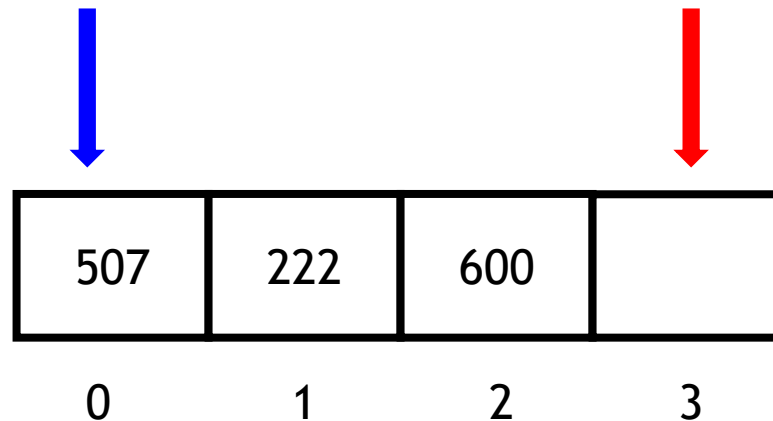
Erro! Fila está cheia!

$\text{fim} == \text{tamanho} - 1$

tamanho=4

inicio=0

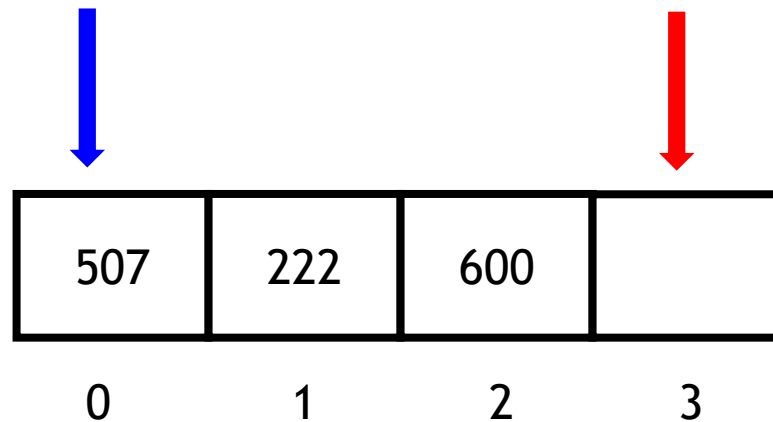
fim=3



# Fila estática

- Como desenfileirar?

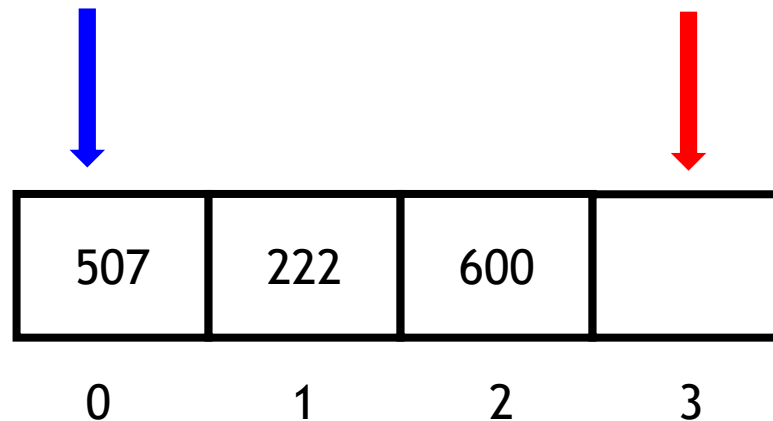
```
tamanho=4  
inicio=0  
fim=3
```



# Fila estática

- Como desenfilear?
  1. Salva número indicado por inicio (507);
  2. inicio++
  3. Retorna o número 507

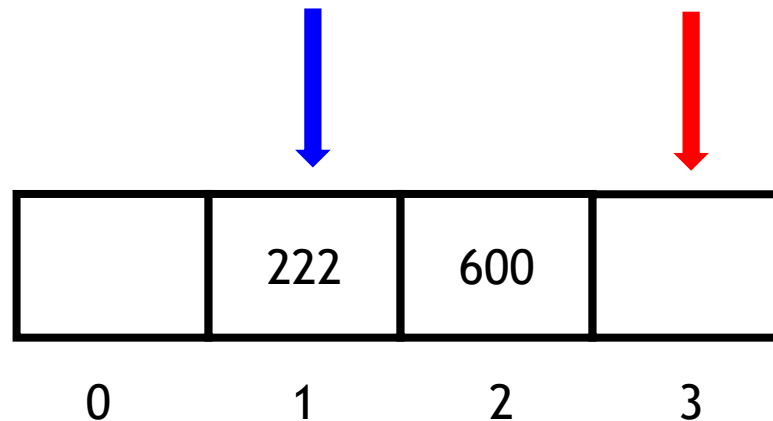
```
tamanho=4  
inicio=0  
fim=3
```



# Fila estática

- Como desenfileirar?
  1. Salva número indicado por inicio (507);
  2. inicio++
  3. Retorna o número 507

```
tamanho=4  
inicio=1  
fim=3
```



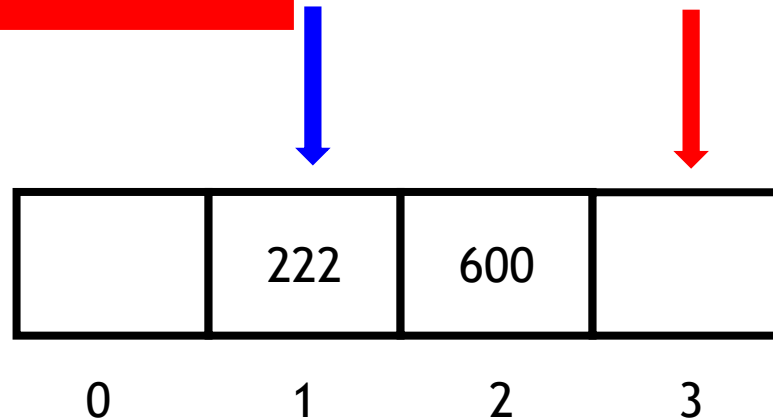
# Fila estática

- Como desenfileirar?

Outra estratégia seria mover todos os itens para o início do arranjo (assim como a lista estática faz).

Qual o problema disso?

```
tamanho=4  
inicio=1  
fim=3
```





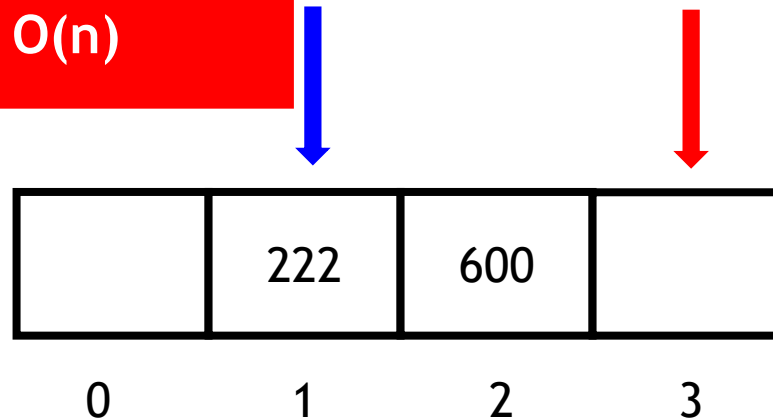
# Fila estática

- Como desenfileirar?

Outra estratégia seria mover todos os itens para o início do arranjo (assim como a lista estática faz).

Qual o problema disso?  
A operação se tornaria  $O(n)$

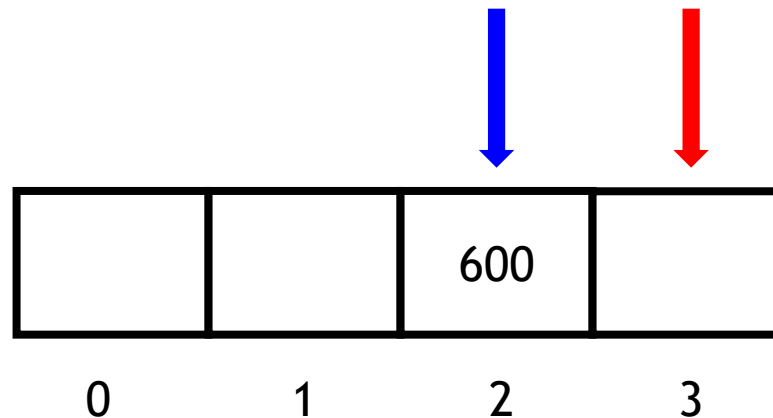
```
tamanho=4  
inicio=1  
fim=3
```



# Fila estática

- Como desenfileirar?
  1. Salva número indicado por inicio (222);
  2. inicio++
  3. Retorna o número 222

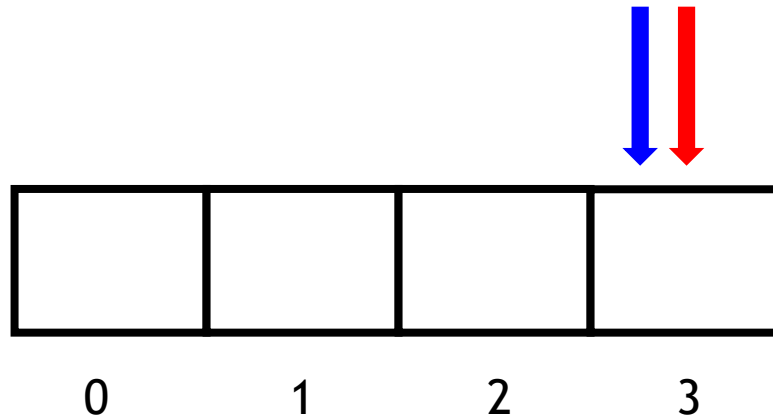
```
tamanho=4  
inicio=2  
fim=3
```



# Fila estática

- Como desenfileirar?
  1. Salva número indicado por início (600);
  2. início++
  3. Retorna o número 600

```
tamanho=4  
início=3  
fim=3
```

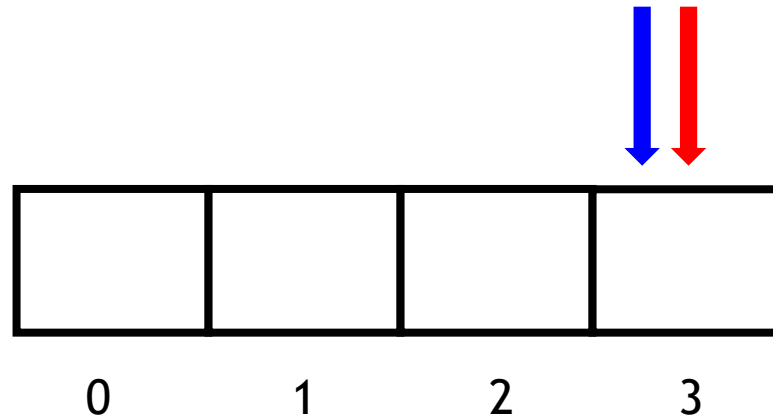


# Fila estática

- Fila ficou vazia!

1. E se enfileirar novo item?
  - Overflow (retorna fila cheia)
  - $\text{fim} == \text{tamanho} - 1$
2. Como solucionar esse problema?

```
tamanho=4  
inicio=3  
fim=3
```



# Fila estática

- Fila ficou vazia!

1. E se enfileirar novo item?
  - Overflow (retorna fila cheia)

- $fim == tamanho - 1$

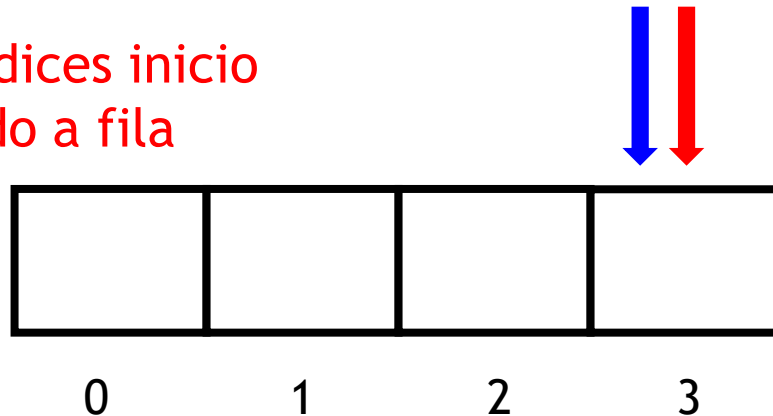
2. Como solucionar esse problema?

- Reiniciar índices *inicio* e *fim* quando a fila ficar vazia.

`tamanho=4`

`inicio=3`

`fim=3`

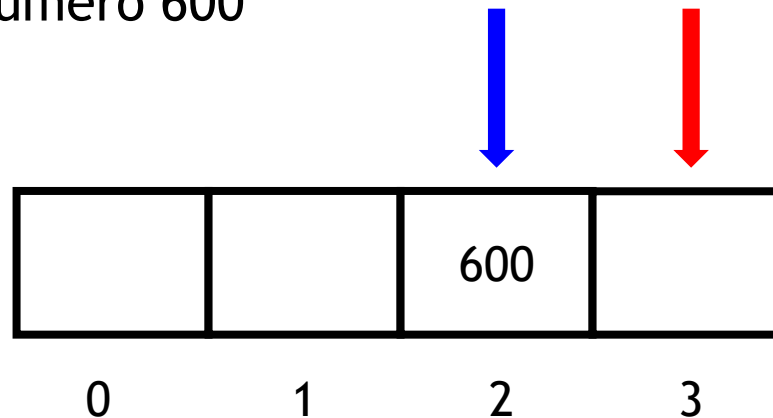


# Fila estática

- Retomando a situação anterior...
  1. Salva número indicado por início (600);
  2. **início == fim-1**
    1. início=0
    2. fim=0
  3. Retorna o número 600

tamanho=4  
início=2  
fim=3

Fila vai  
ficar  
vazia!



# Fila estática

- Retomando a situação anterior...

1. Salva número indicado por início (600);

2. **início == fim-1**

1. início=0

2. fim=0

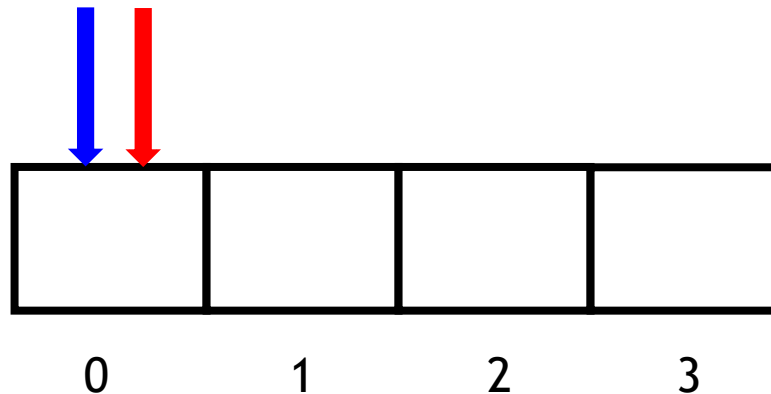
3. Retorna o número 600

tamanho=4

início=0

fim=0

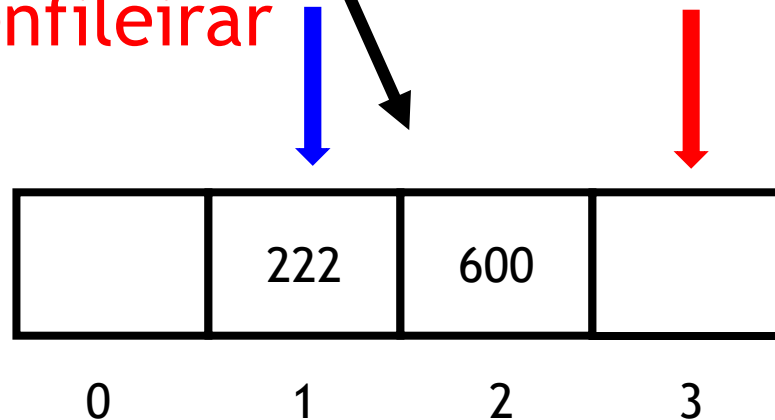
Fila vai  
ficar  
vazia!



# Problema

- Neste caso, a fila não ficará vazia após desenfileirar 222
- Mas ainda assim ocorrerá overflow se tentarmos enfileirar outro item!

tamanho=4  
inicio=1  
fim=3





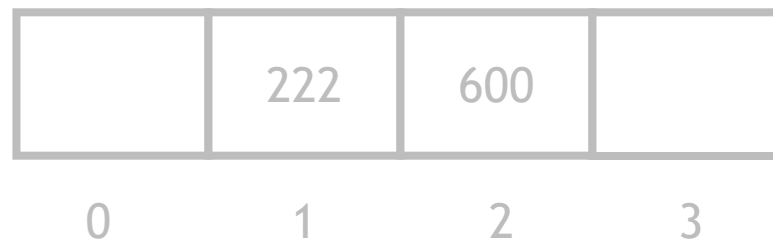
# Problema

- Neste caso, a fila não ficará vazia após desenfileirar 222

```
tamanho=4  
inicio=1  
fim=3
```

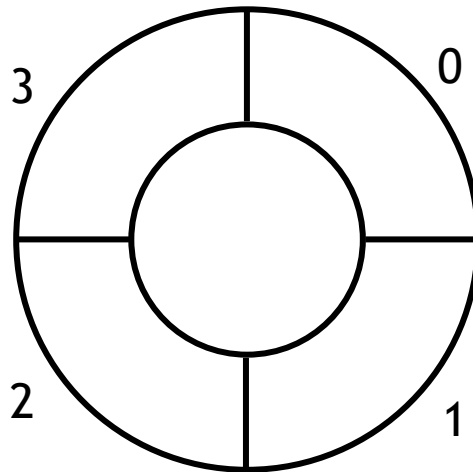
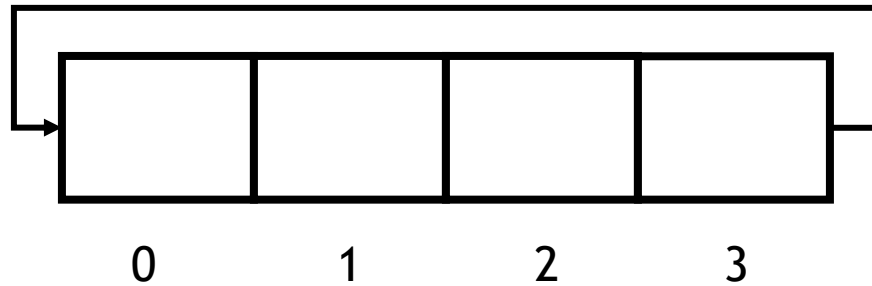
- Mas ocorre outro item.

Podemos melhorar essa solução?  
Sim, com a fila estática circular!



# Fila estática circular

- Trata o vetor como se estivesse em um círculo:



# Fila estática circular

- Inicialização:
  - Estrutura igual à fila estática não circular;
  - O que muda é a forma de tratar os índices **inicio** e **fim**.

Fila:

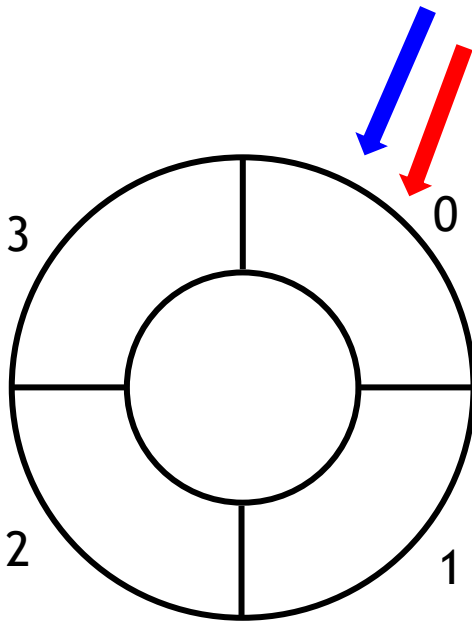
**itens** (referência para um vetor/arranjo)  
**tamanho** (inteiro)  
**inicio**, **fim** (inteiro)

# Fila estática circular

- Inicialização:

```
tamanho=4  
início=0  
fim=0
```

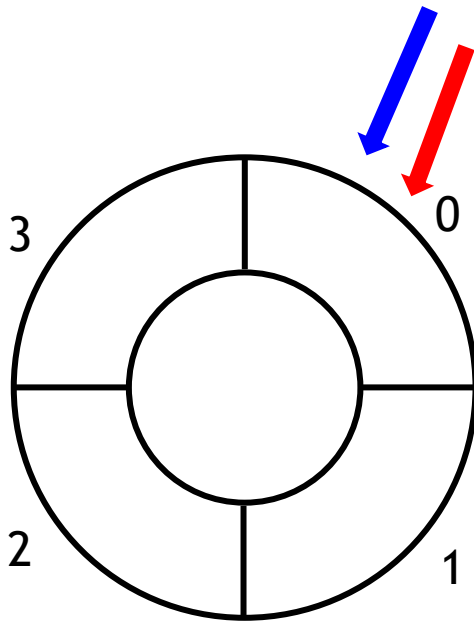
(início == fim) ==> Fila vazia!



```
itens=alocar vetor com tamanho 4
```

# Fila estática circular

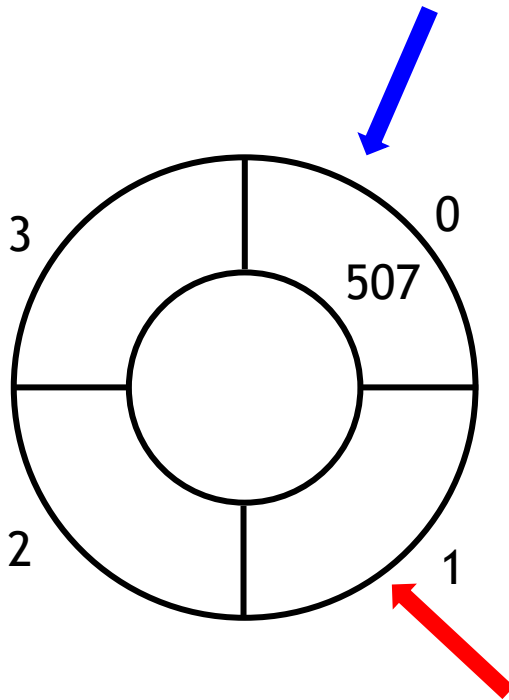
```
tamanho=4  
inicio=0  
fim=0
```



- Como enfileirar?
  - $\text{novo\_fim} = (\text{fim} + 1) \% \text{tamanho}$
  - **$\text{novo\_fim} == \text{inicio} ?$** 
    - **Sim: fila cheia!**
    - Não: podemos enfileirar.

# Fila estática circular

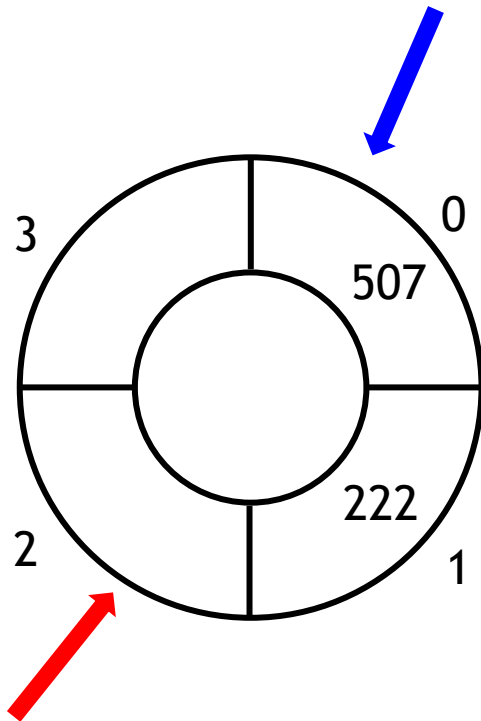
```
tamanho=4  
inicio=0  
fim=1
```



- Como enfileirar?
  - $\text{novo\_fim} = (\text{fim} + 1) \% \text{tamanho}$
  - **$\text{novo\_fim} == \text{inicio} ?$** 
    - **Sim: fila cheia!**
    - Não: podemos enfileirar.
- Enfileirar 507:
  - Verifica se fila está cheia
  - $\text{itens}[\text{fim}] = 507$
  - $\text{fim} = \text{novo\_fim}$

# Fila estática circular

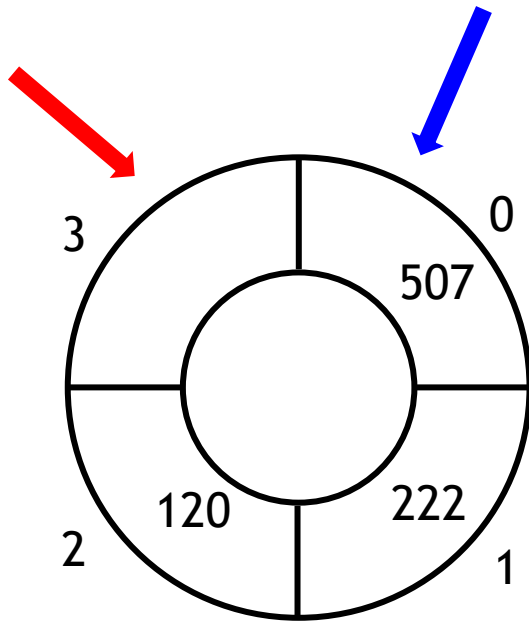
```
tamanho=4  
inicio=0  
fim=2
```



- Como enfileirar?
  - $\text{novo\_fim} = (\text{fim} + 1) \% \text{tamanho}$
  - **$\text{novo\_fim} == \text{inicio}$  ?**
    - **Sim: fila cheia!**
    - Não: podemos enfileirar.
- Enfileirar 222:
  - Verifica se fila está cheia
  - $\text{itens}[\text{fim}] = 222$
  - $\text{fim} = \text{novo\_fim}$

# Fila estática circular

```
tamanho=4  
inicio=0  
fim=3
```

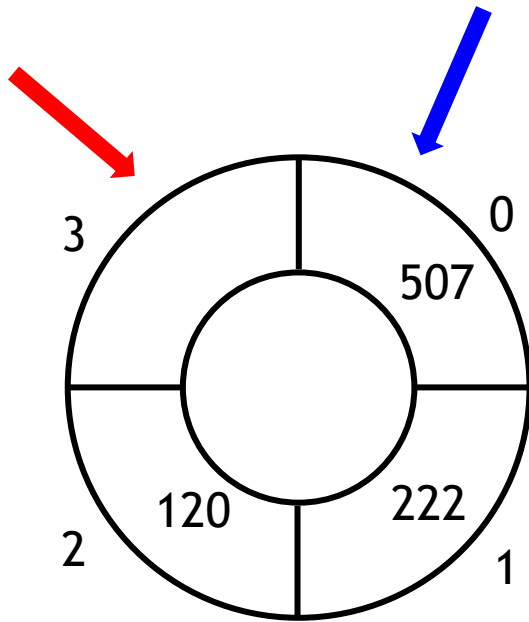


- Como enfileirar?
  - $\text{novo\_fim} = (\text{fim} + 1) \% \text{tamanho}$
  - **$\text{novo\_fim} == \text{inicio}$  ?**
    - **Sim: fila cheia!**
    - Não: podemos enfileirar.
- Enfileirar 120:
  - Verifica se fila está cheia
  - $\text{itens}[\text{fim}] = 120$
  - $\text{fim} = \text{novo\_fim}$



# Fila estática circular

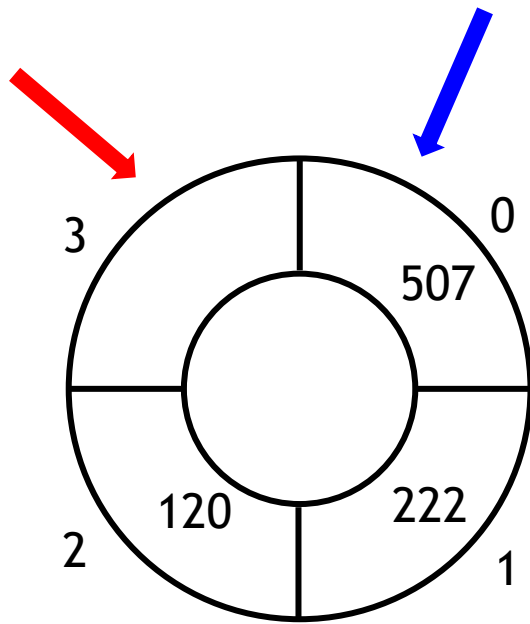
```
tamanho=4  
inicio=0  
fim=3
```



- Como enfileirar?
  - $\text{novo\_fim} = (\text{fim} + 1) \% \text{tamanho}$
  - **$\text{novo\_fim} == \text{inicio}$  ?**
    - Sim: fila cheia!
    - Não: podemos enfileirar.
- Enfileirar 300:
  - Verifica se fila está cheia
  - **Fila está cheia!**
    - $(3 + 1) \% 4 == \text{inicio}$

# Fila estática circular

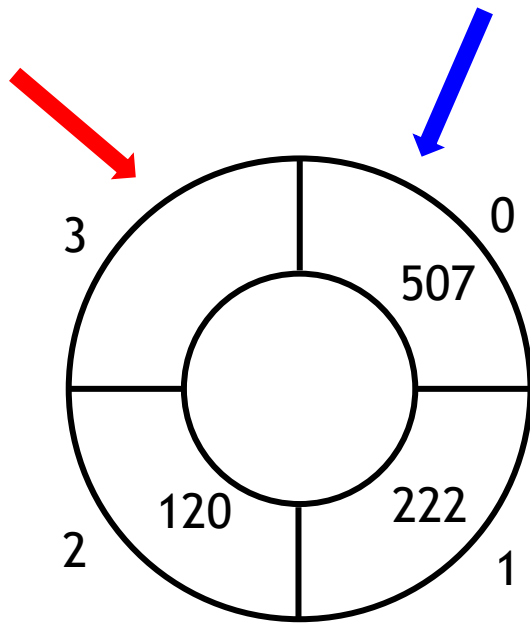
```
tamanho=4  
inicio=0  
fim=3
```



- Como desenfileirar?
  - **inicio==fim?**
    - **Sim: fila vazia!**
    - Não: Podemos desenfileirar.
  - Salva itens[inicio]
  - $\text{novo\_inicio} = (\text{inicio} + 1) \% \text{tamanho}$
  - **inicio = novo\_inicio**
  - **Retorna item salvo**

# Fila estática circular

```
tamanho=4  
inicio=0  
fim=3
```



## • Como desenfileirar?

Não

### • **inicio==fim?**

- **Sim: fila vazia!**

- Não: Podemos desenfileirar.

507

- Salva itens[inicio]

1

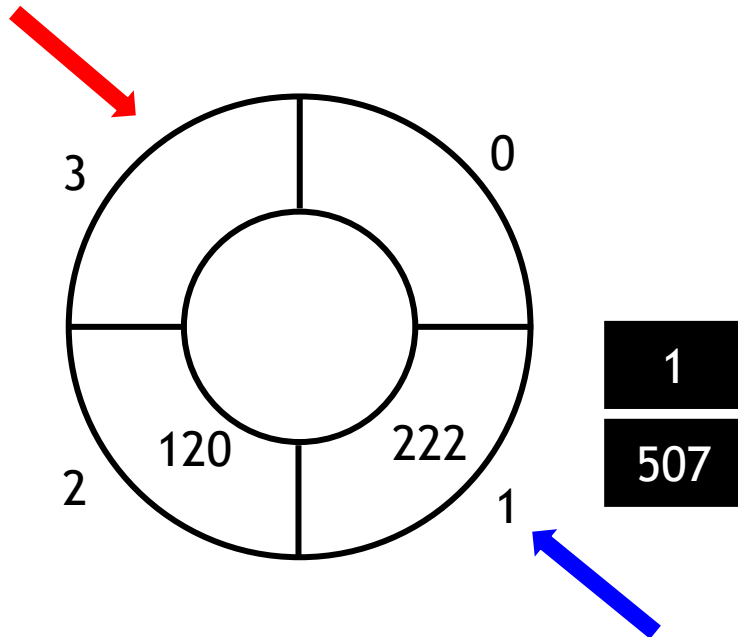
- $\text{novo\_inicio} = (\text{inicio} + 1) \% \text{tamanho}$

- **inicio = novo\_inicio**

- **Retorna item salvo**

# Fila estática circular

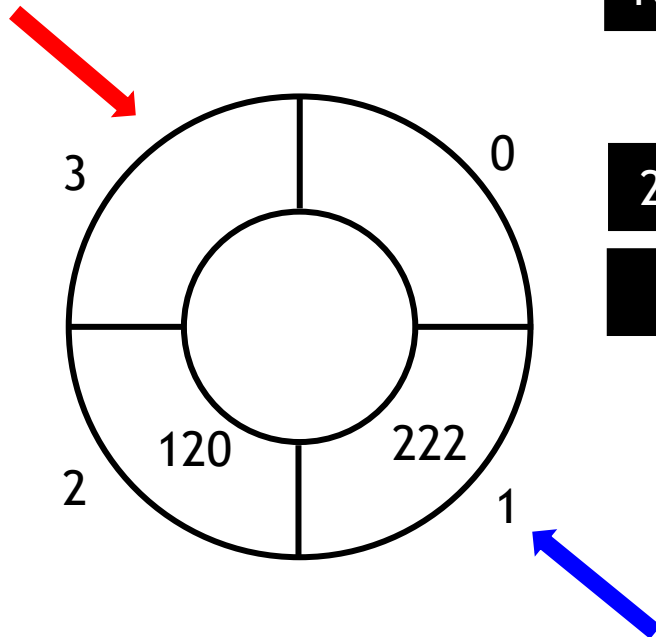
```
tamanho=4  
inicio=1  
fim=3
```



- Como desenfileirar?
  - **inicio==fim?**
    - **Sim: fila vazia!**
    - Não: Podemos desenfileirar.
  - Salva itens[inicio]
  - $\text{novo\_inicio} = (\text{inicio} + 1) \% \text{tamanho}$
  - **inicio = novo\_inicio**
  - **Retorna item salvo**

# Fila estática circular

```
tamanho=4  
inicio=1  
fim=3
```



## • Como desenfileirar?

Não

### • **inicio==fim?**

- **Sim: fila vazia!**

- Não: Podemos desenfileirar.

222

- Salva itens[inicio]

2

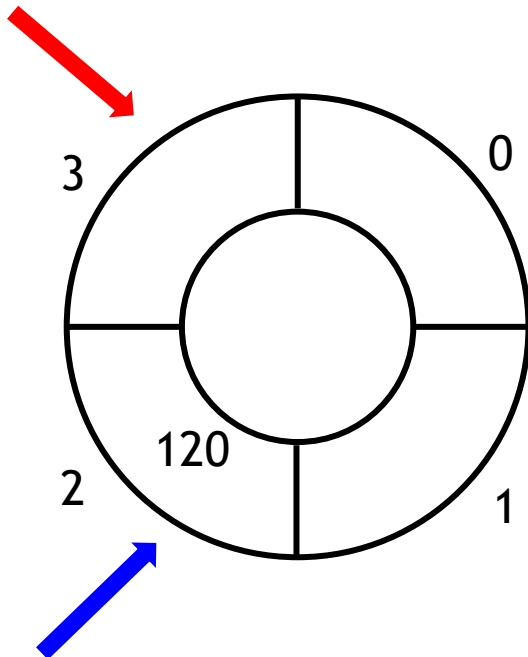
- $\text{novo\_inicio} = (\text{inicio} + 1) \% \text{tamanho}$

- **inicio = novo\_inicio**

- **Retorna item salvo**

# Fila estática circular

```
tamanho=4  
inicio=2  
fim=3
```



- Como desenfileirar?

- **inicio==fim?**

- **Sim: fila vazia!**

- Não: Podemos desenfileirar.

- Salva itens[inicio]

- $\text{novo\_inicio} = (\text{inicio} + 1) \% \text{tamanho}$

- **inicio = novo\_inicio**

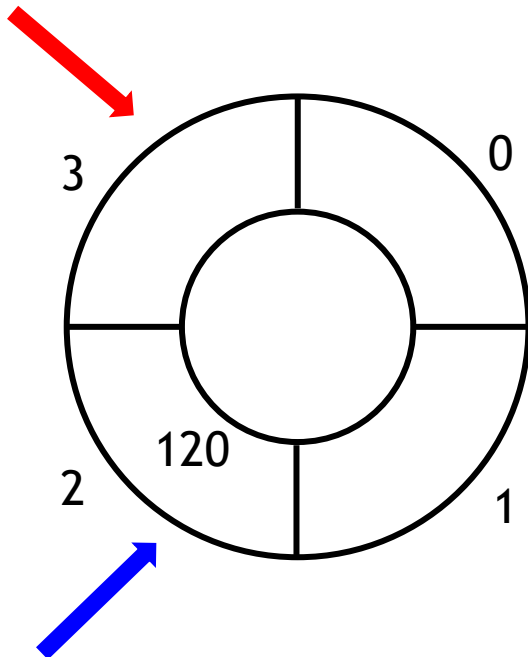
- Retorna item salvo

2

222

# Fila estática circular

```
tamanho=4  
inicio=2  
fim=3
```



## • Como desenfileirar?

Não

### • **inicio==fim?**

- **Sim: fila vazia!**

- Não: Podemos desenfileirar.

120

- Salva itens[inicio]

3

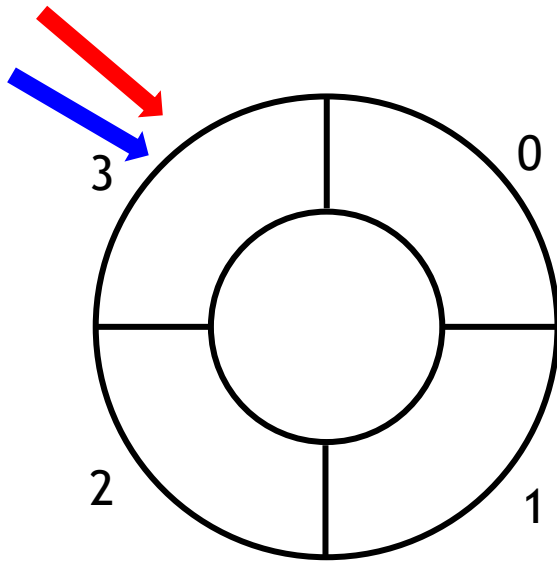
- $\text{novo\_inicio} = (\text{inicio} + 1) \% \text{tamanho}$

- **inicio = novo\_inicio**

- **Retorna item salvo**

# Fila estática circular

```
tamanho=4  
inicio=3  
fim=3
```



- Como desenfileirar?

- **inicio==fim?**

- **Sim: fila vazia!**

- Não: Podemos desenfileirar.

- Salva itens[inicio]

- $\text{novo\_inicio} = (\text{inicio} + 1) \% \text{tamanho}$

- **inicio = novo\_inicio**

- Retorna item salvo

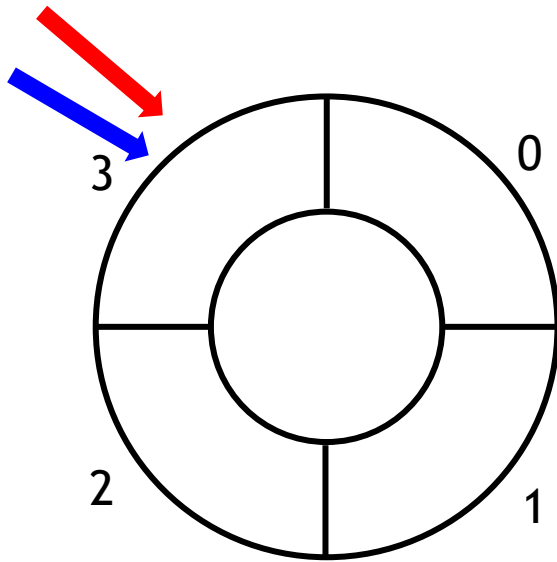
3

120



# Fila estática circular

```
tamanho=4  
inicio=3  
fim=3
```

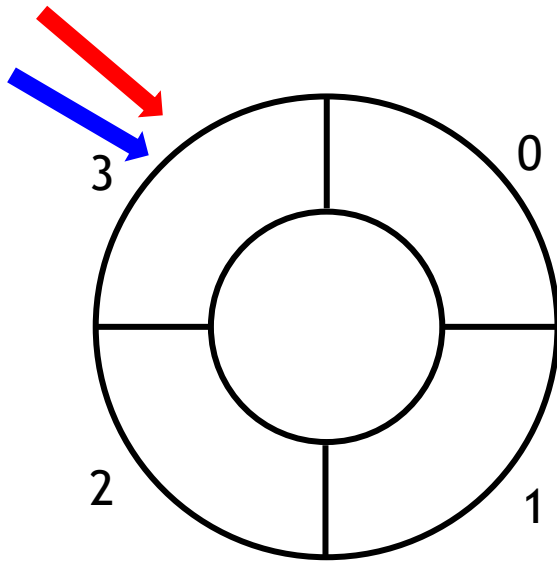


- Como desenfileirar?
  - **inicio==fim?**
    - **Sim: fila vazia!**
    - Não: Podemos desenfileirar.

**(inicio == fim) ==> Fila vazia!**

# Fila estática circular

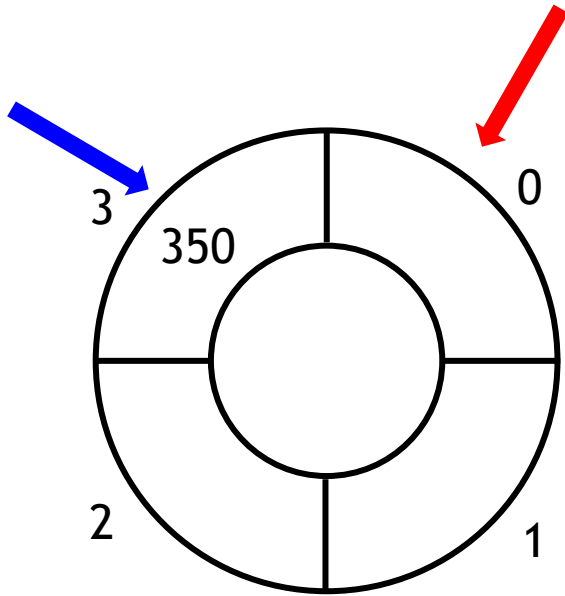
```
tamanho=4  
inicio=3  
fim=3
```



- Enfileirar 350:
  - $\text{novo\_fim} = (\text{fim} + 1) \% \text{tamanho}$
  - $\text{novo\_fim} = (3 + 1) \% 4 = 0$
  - **$\text{novo\_fim} == \text{inicio} ?$** 
    - Não: então podemos enfileirar!

# Fila estática circular

```
tamanho=4  
inicio=3  
fim=0
```



- Enfileirar 350:
  - $\text{novo\_fim} = (\text{fim} + 1) \% \text{tamanho}$
  - $\text{novo\_fim} = (3 + 1) \% 4 = 0$
  - **$\text{novo\_fim} == \text{inicio} ?$** 
    - Não: então podemos enfileirar!
  - $\text{itens}[\text{fim}] = 350$
  - $\text{fim} = \text{novo\_fim}$

# Fila dinâmica

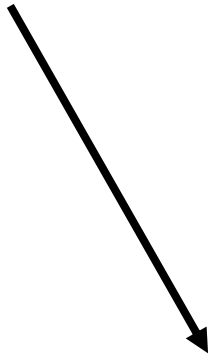
# Fila dinâmica

- **Fila dinâmica**: implementa a estrutura de dados utilizando uma **lista ligada**;
- Portanto, os itens são alocados em memória de acordo com a necessidade.

# Fila dinâmica

- Estrutura básica:

Referências para  
o início e para o  
fim da fila



**Fila:**

**inicio** (referência para um `LinkedListNode`)  
**fim** (referência para um `LinkedListNode`)

# Fila dinâmica

- Inicialização

```
início=NULL  
fim=NULL
```

**início==NULL ==> Fila vazia!**

# Fila dinâmica

- Como enfileirar?

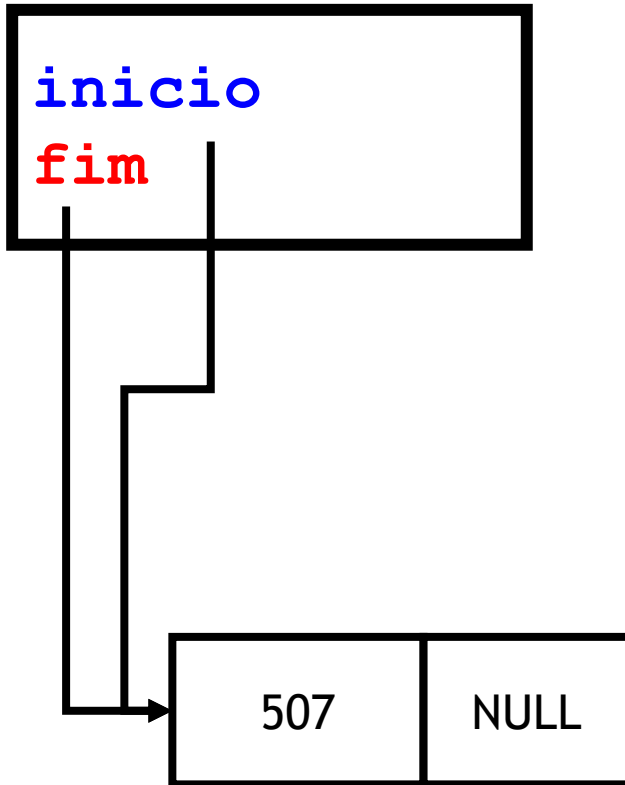
```
início=NULL  
fim=NULL
```

1. Alocar novo LinkedNode
2. Adicioná-lo logo após o item apontado por **fim**
3. Atualizar ponteiro **fim**
4. Quando a fila está vazia, atualizar ponteiro **início** também



# Fila dinâmica

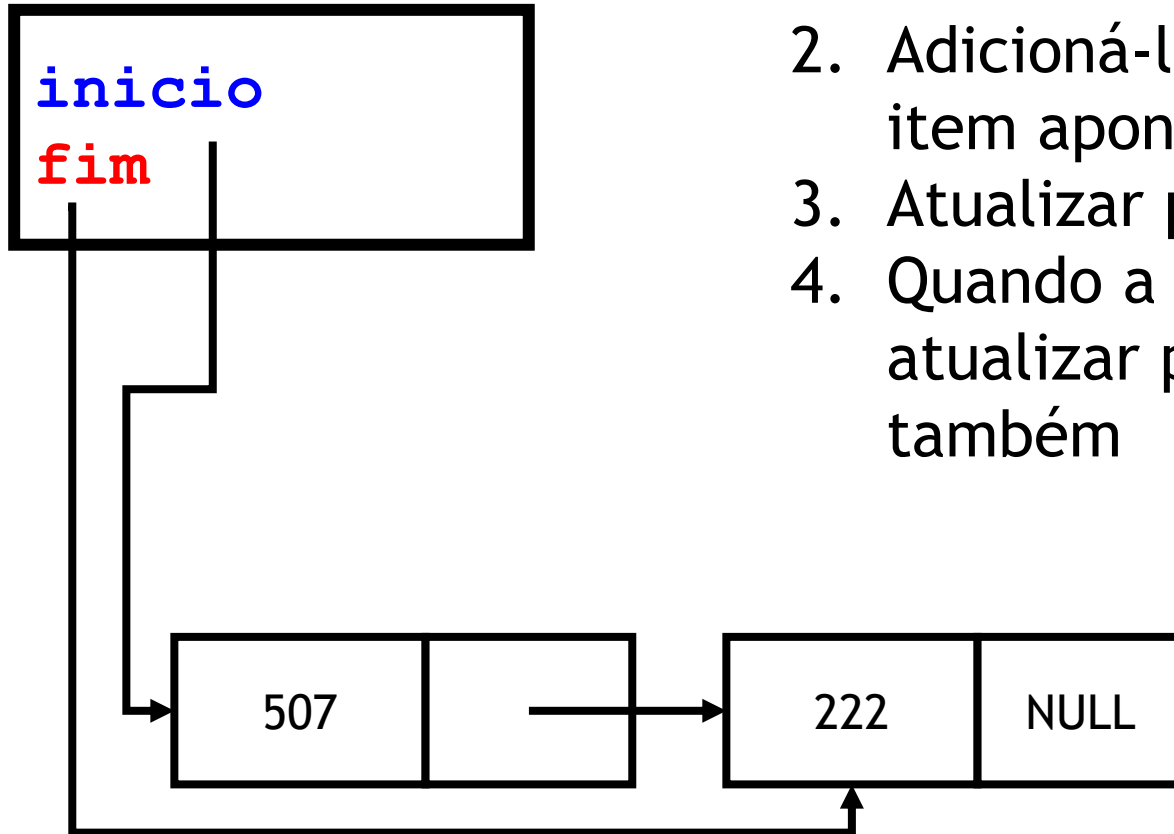
- Como enfileirar?



1. Alocar novo ListNode
2. Adicioná-lo logo após o item apontado por **fim**
3. Atualizar ponteiro **fim**
4. Quando a fila está vazia, atualizar ponteiro **inicio** também

# Fila dinâmica

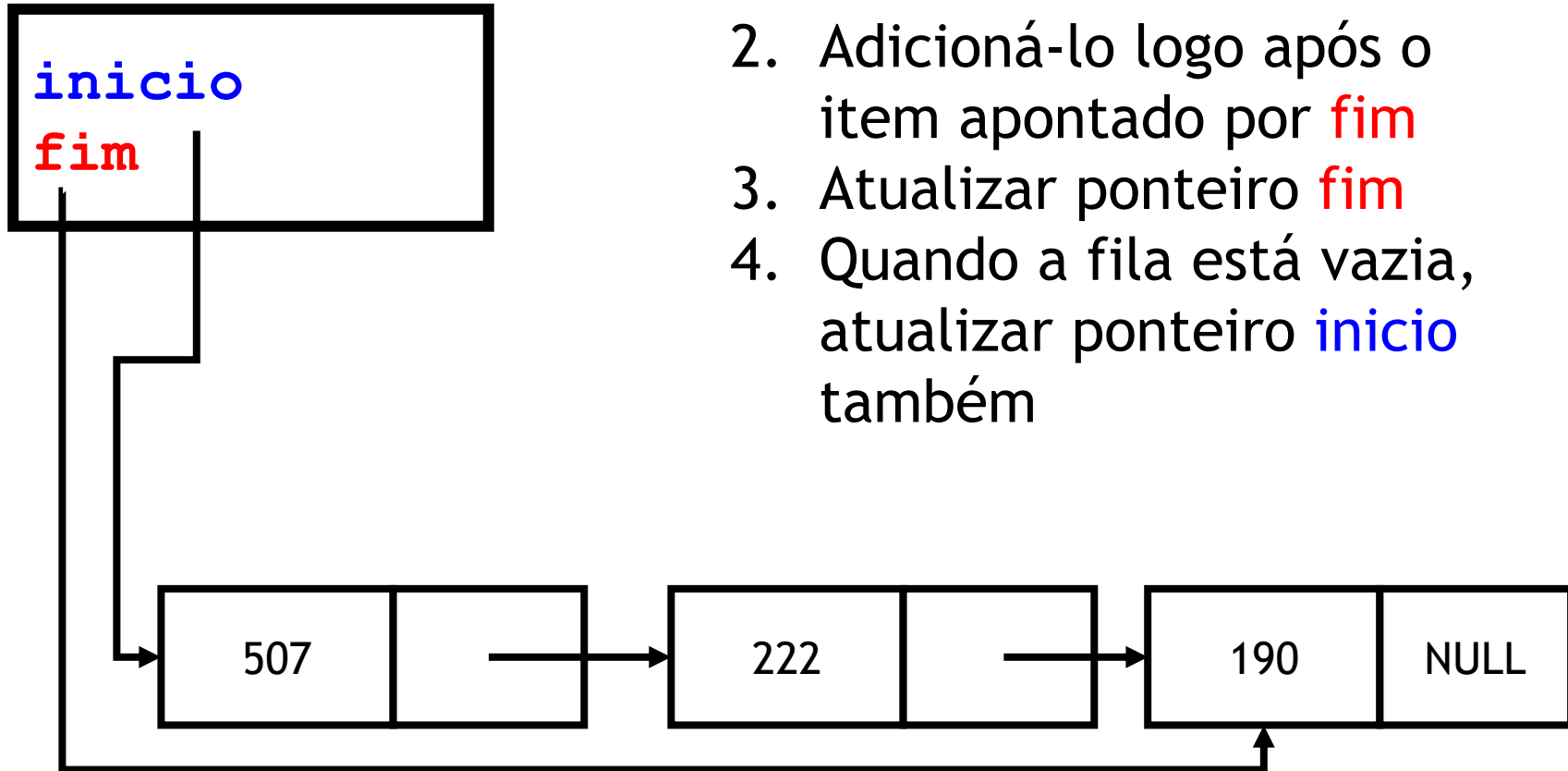
- Como enfileirar?



1. Alocar novo `LinkedList`
2. Adicioná-lo logo após o item apontado por `fim`
3. Atualizar ponteiro `fim`
4. Quando a fila está vazia, atualizar ponteiro `inicio` também

# Fila dinâmica

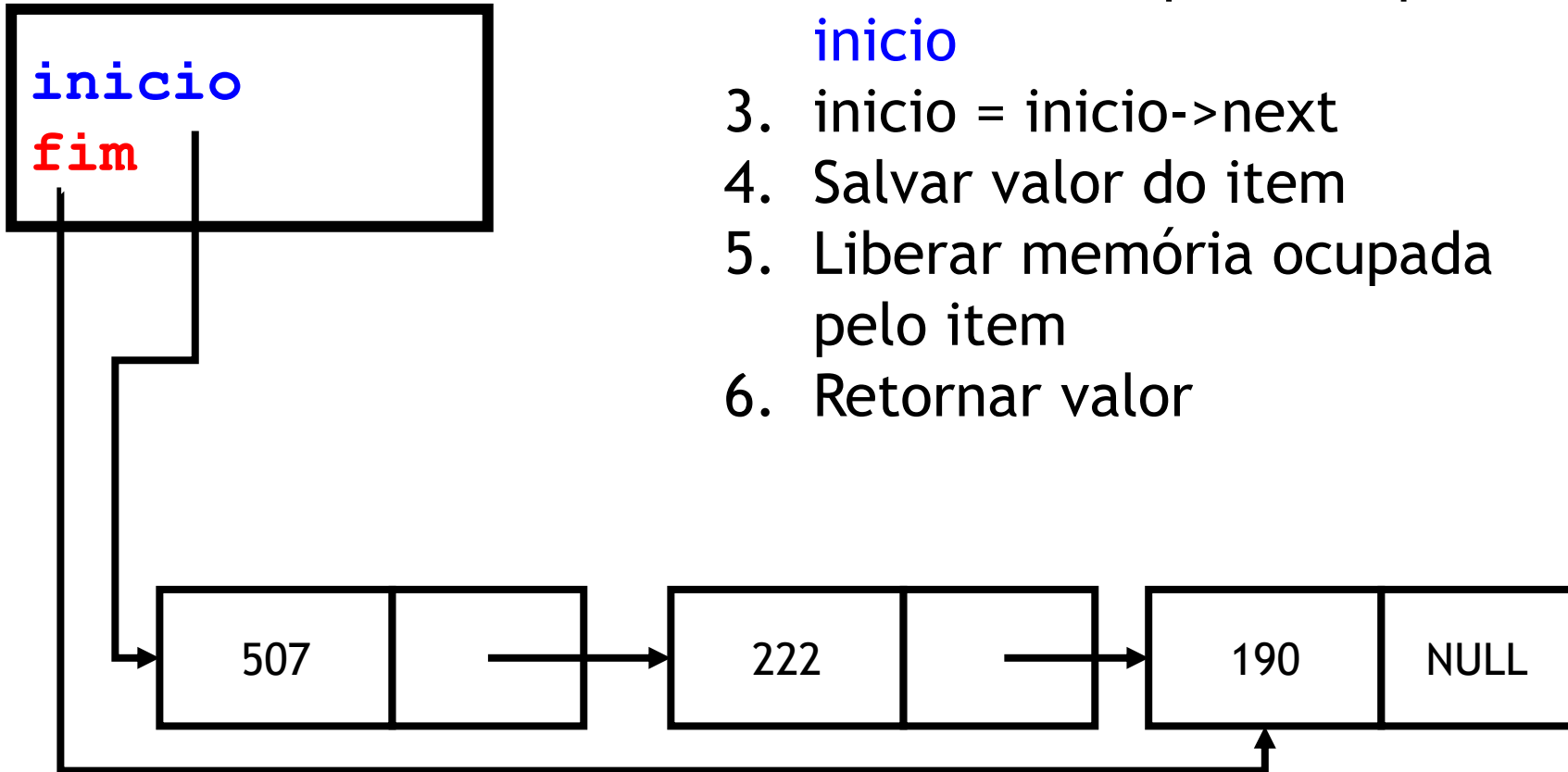
- Como enfileirar?



1. Alocar novo ListNode
2. Adicioná-lo logo após o item apontado por **fim**
3. Atualizar ponteiro **fim**
4. Quando a fila está vazia, atualizar ponteiro **inicio** também

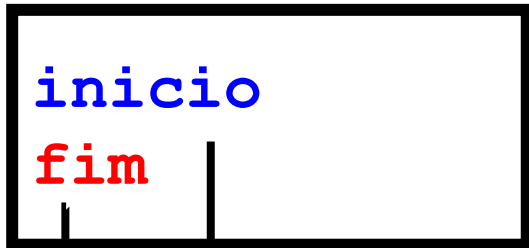
# Fila dinâmica

- Como desenfileirar?

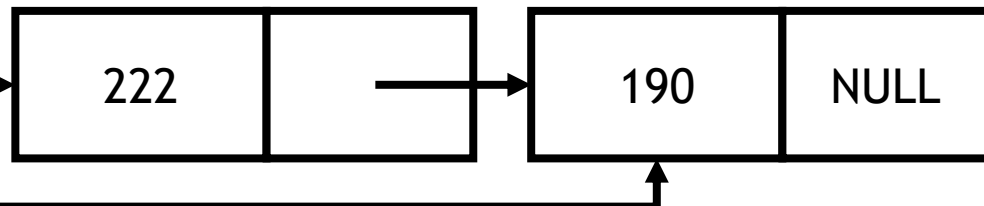


# Fila dinâmica

- Como desenfileirar?

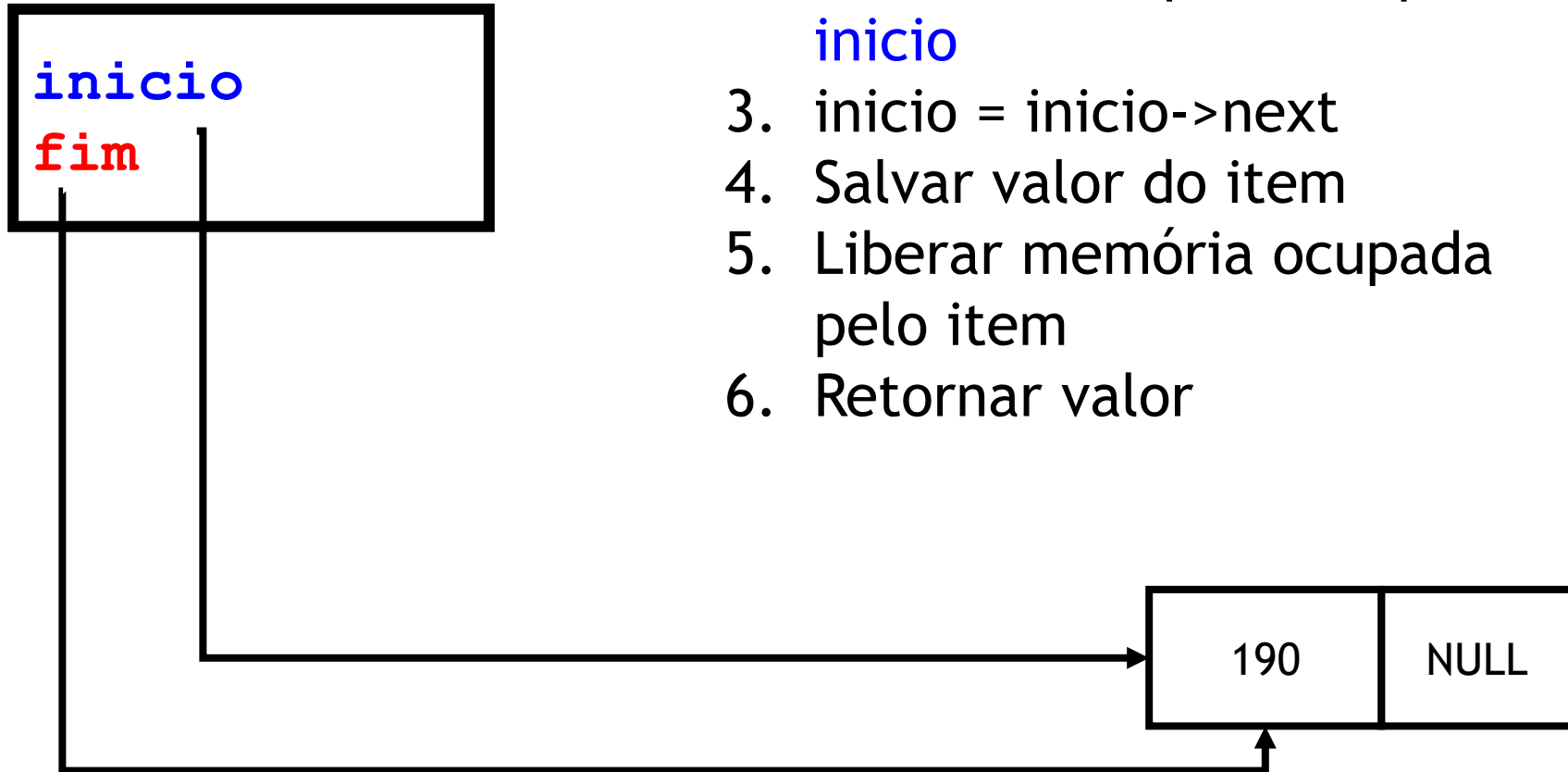


1. Verificar se fila está vazia
2. Salvar item apontado por **inicio**
3.  $\text{inicio} = \text{inicio} \rightarrow \text{next}$
4. Salvar valor do item
5. Liberar memória ocupada pelo item
6. Retornar valor



# Fila dinâmica

- Como desenfileirar?



# Fila dinâmica

- Como desenfileirar?

```
início=NULL  
fim=NULL
```

1. Verificar se fila está vazia
2. Salvar item apontado por **início**
3. `início = início->next`
4. Salvar valor do item
5. Liberar memória ocupada pelo item
6. Retornar valor

Quando desenfileirar o último elemento, volte para  
`início=fim=NULL`

# Resumo



# Resumo

**Pilha**

**Operações:**

- Empilhar/push
- Desempilhar/pop

**LIFO**

**Fila**

**Operações:**

- Enfileirar/enqueue
- Desenfileirar/dequeue

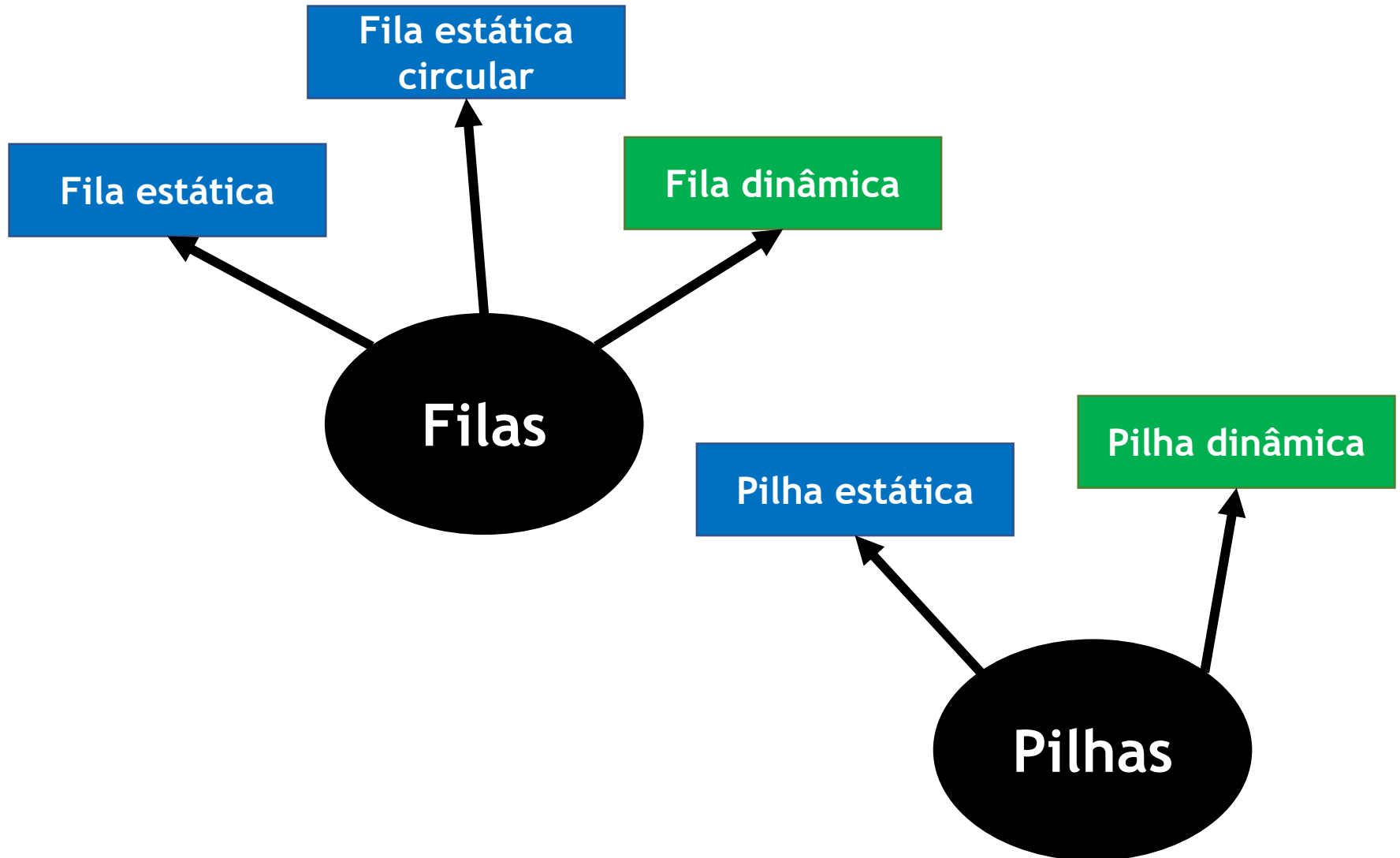
**FIFO**

# Outros tipos

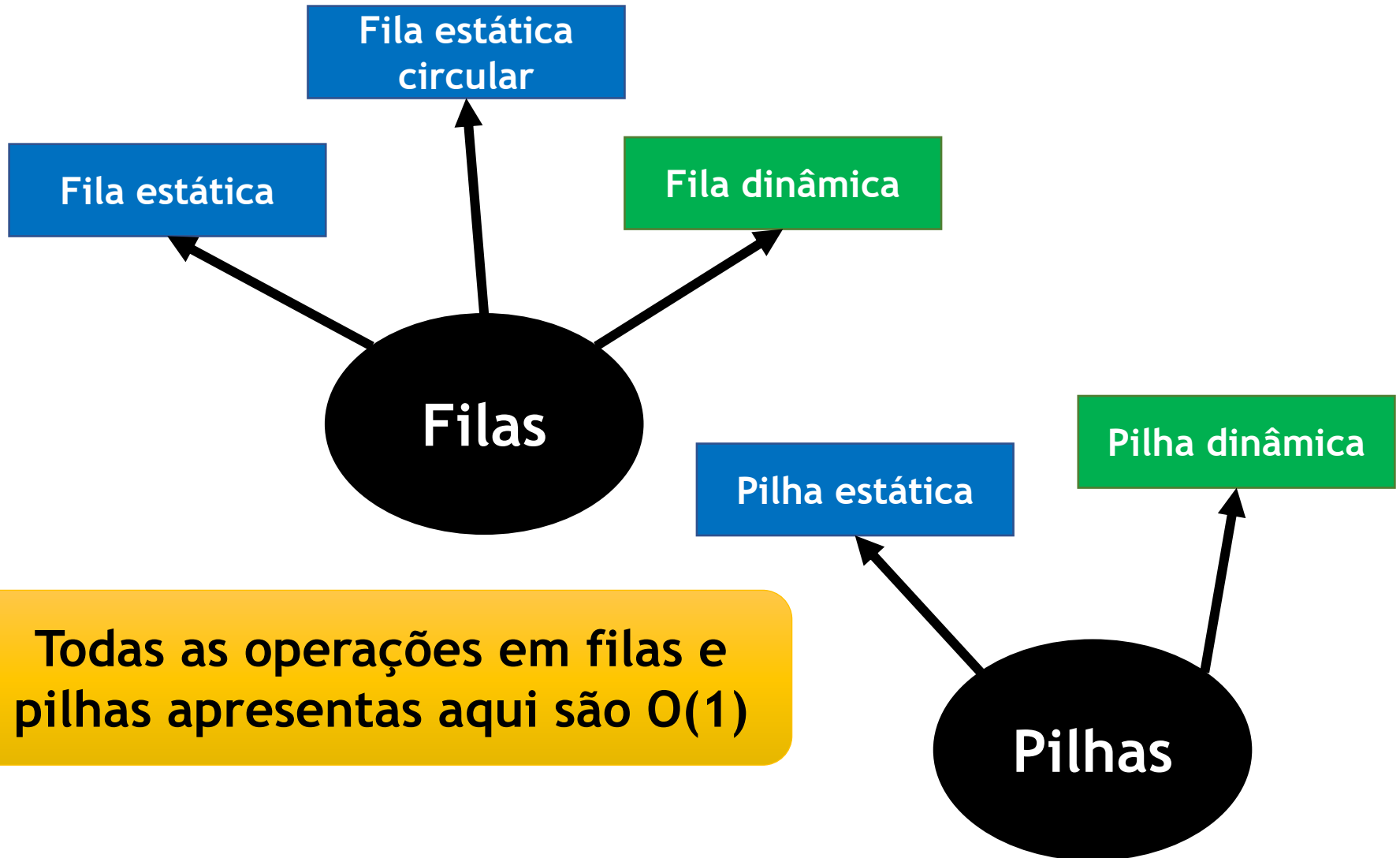
- **Deque:** permite remover elementos do início ou do fim (combinação de fila e pilha);
- **Pilha auto-ajustável:** pilha estática que aloca/desaloca memória de acordo com a carga;
- **Pilhas múltiplas:** permite que mais de uma pilha compartilhe o mesmo arranjo.



# Resumo



# Resumo



# Referências

- Slides sobre Listas, Pilhas e Filas do Prof. Monael Pinheiro Ribeiro - Algoritmos e Estruturas de Dados I
- Nivio Ziviani. Projeto de Algoritmos: com implementações em Pascal e C. Cengage Learning, 2015.
- Robert Sedgewick. Algorithms in C/C++/Java, Parts 1-4 (Fundamental Algorithms, Data Structures, Sorting, Searching). Addison-Wesley Professional.
- Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Algoritmos: Teoria e Prática. Elsevier, 2012.

Este material recebeu contribuições da  
Profa. Mirtha Lina Fernández Venero.