

.....

# Filas Estáticas

.....

## Estrutura de Dados

Prof. MSc. [igor.borges@anhembi.br](mailto:igor.borges@anhembi.br)



Universidade  
Anhembi Morumbi

LAUREATE INTERNATIONAL UNIVERSITIES

# Agenda

- Filas
  - Conceito
  - Aplicações
- Filas Estáticas
  - Operações
  - Código
  - Filas Circulares Estáticas

.....

# Filas

.....



Universidade  
Anhembi Morumbi  
LAUREATE INTERNATIONAL UNIVERSITIES

# Filas



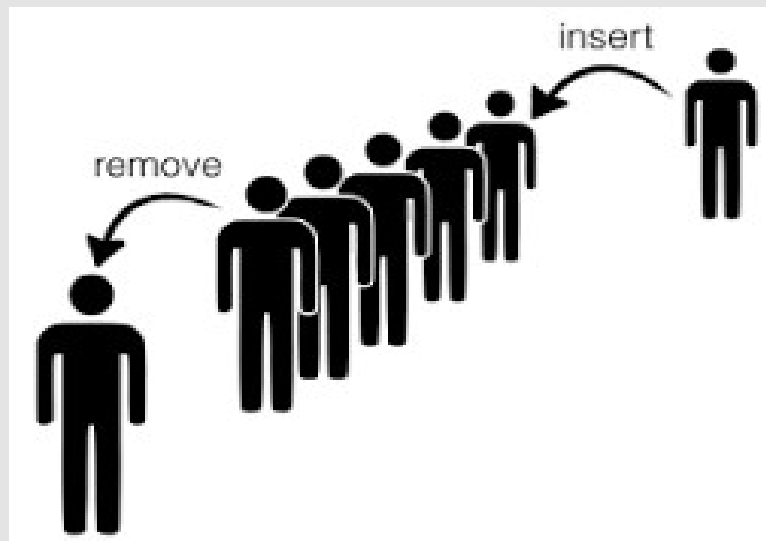
# Filas

- Conjunto ordenado de itens no qual somente em uma das extremidades novos itens podem ser inseridos, E somente na OUTRA extremidade itens podem ser removidos



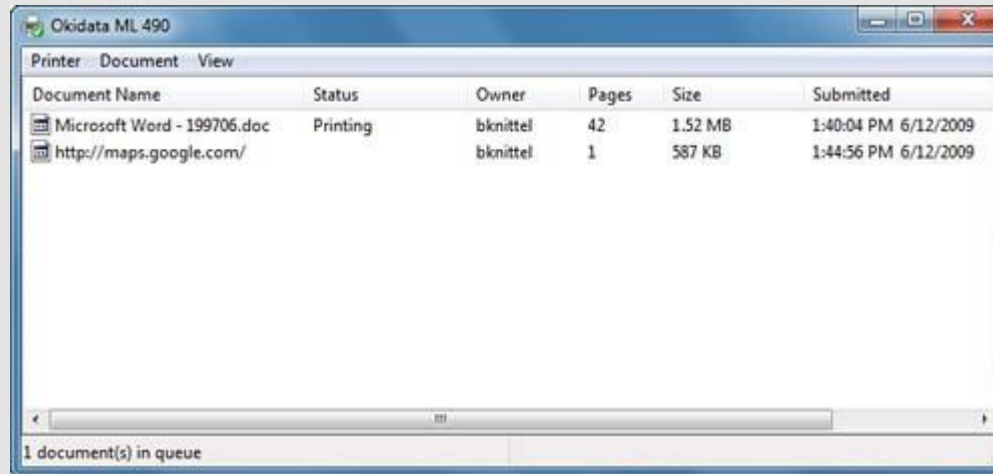
# Filas

- FIFO – First In, First Out → O primeiro a entrar será o primeiro a sair
- A extremidade onde os itens são removidos é o **começo** da fila e a extremidade onde são inseridos é o **final** da fila.



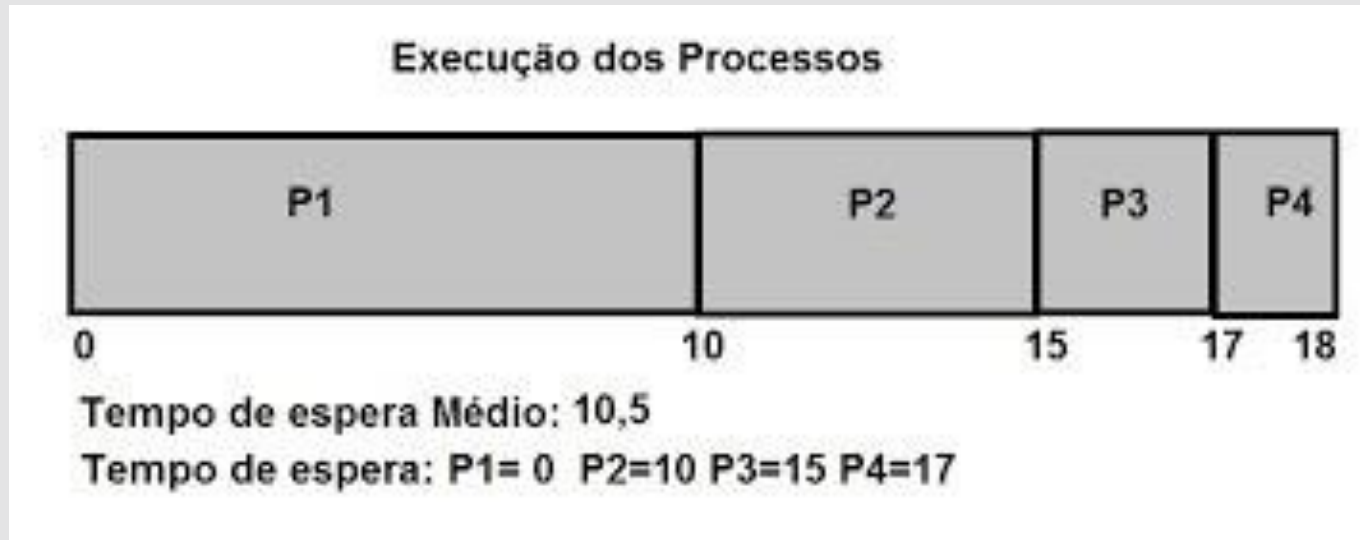
# Aplicação - Filas

- Fila de Impressão



# Aplicação - Filas

- Processos





.....

# Filas Estáticas

.....



Universidade  
Anhembi Morumbi  
LAUREATE INTERNATIONAL UNIVERSITIES

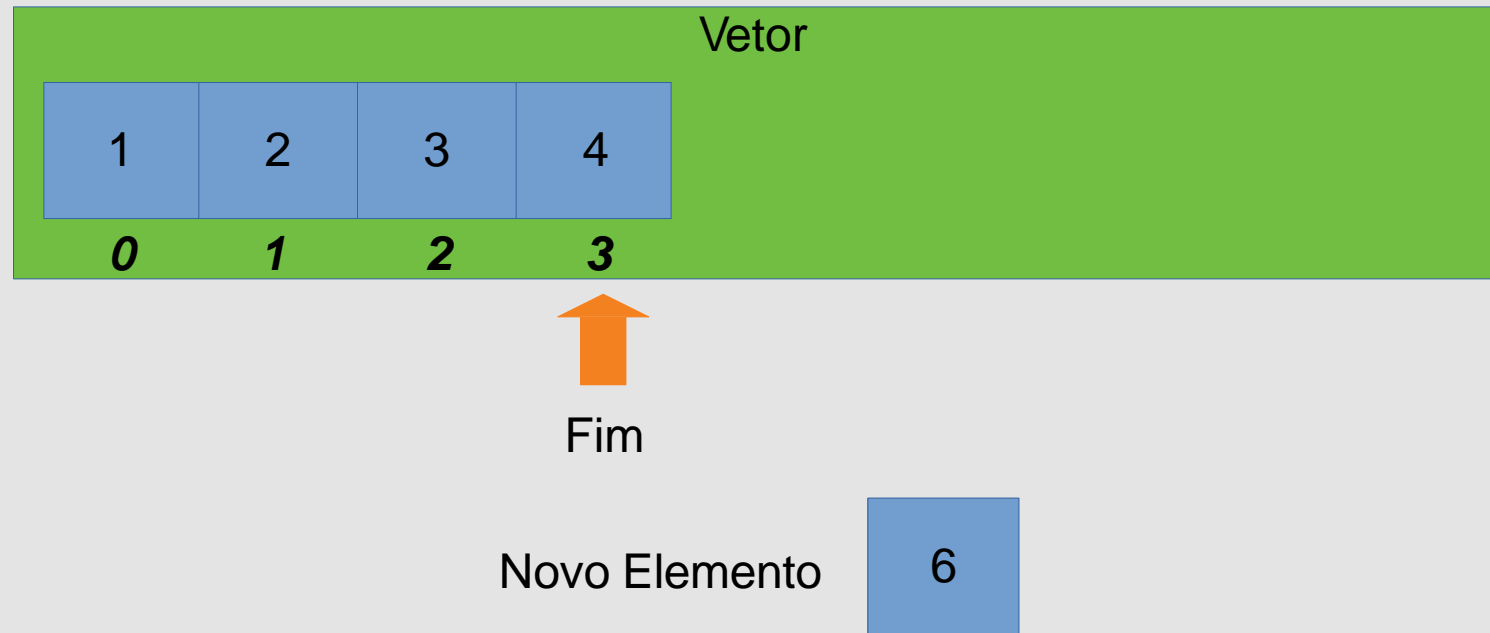
# Operações – Fila Estática

- Fila
  - Inglês UK - *Queue*
- Inserir um elemento na Fila → `enqueue`
- Remover um elemento da Fila → `dequeue`
- Verificar se a Fila está Vazia → `isEmpty`
- Verificar se a Fila está cheia → `isFull`
- Exibe o próximo da Fila → `peek`
- Exibir a Fila → `display`

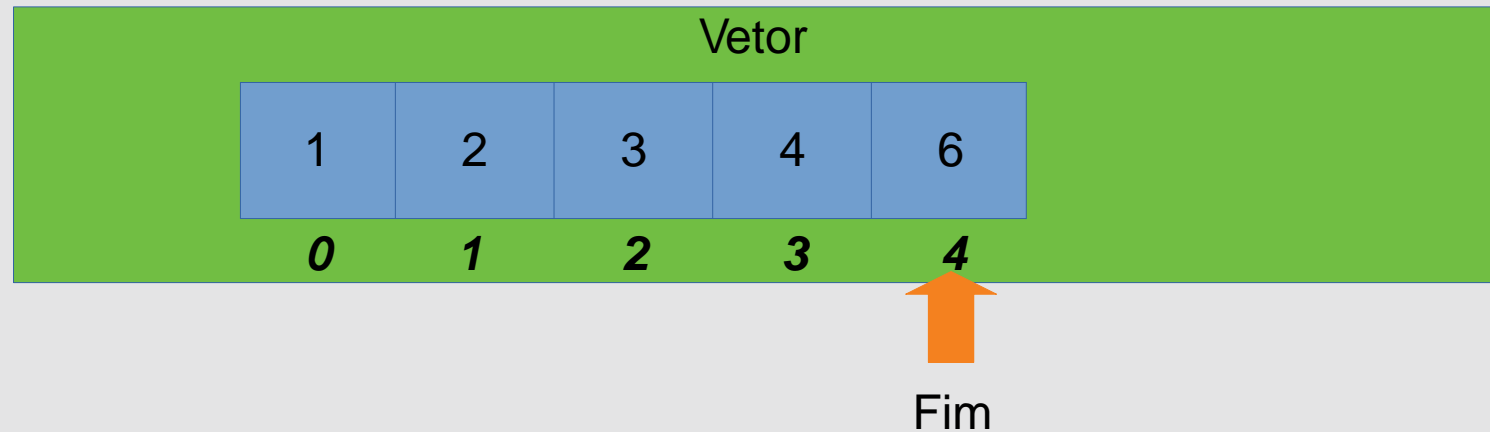
# Código - Fila Estática

```
public class Fila {  
  
    private int vetorFila[];  
    private int fim;  
  
    public Fila(int maxSize) {  
        vetorFila = new int[ maxSize ];  
        fim = -1;  
    }  
    // Métodos das operações da fila  
    // ...  
}
```

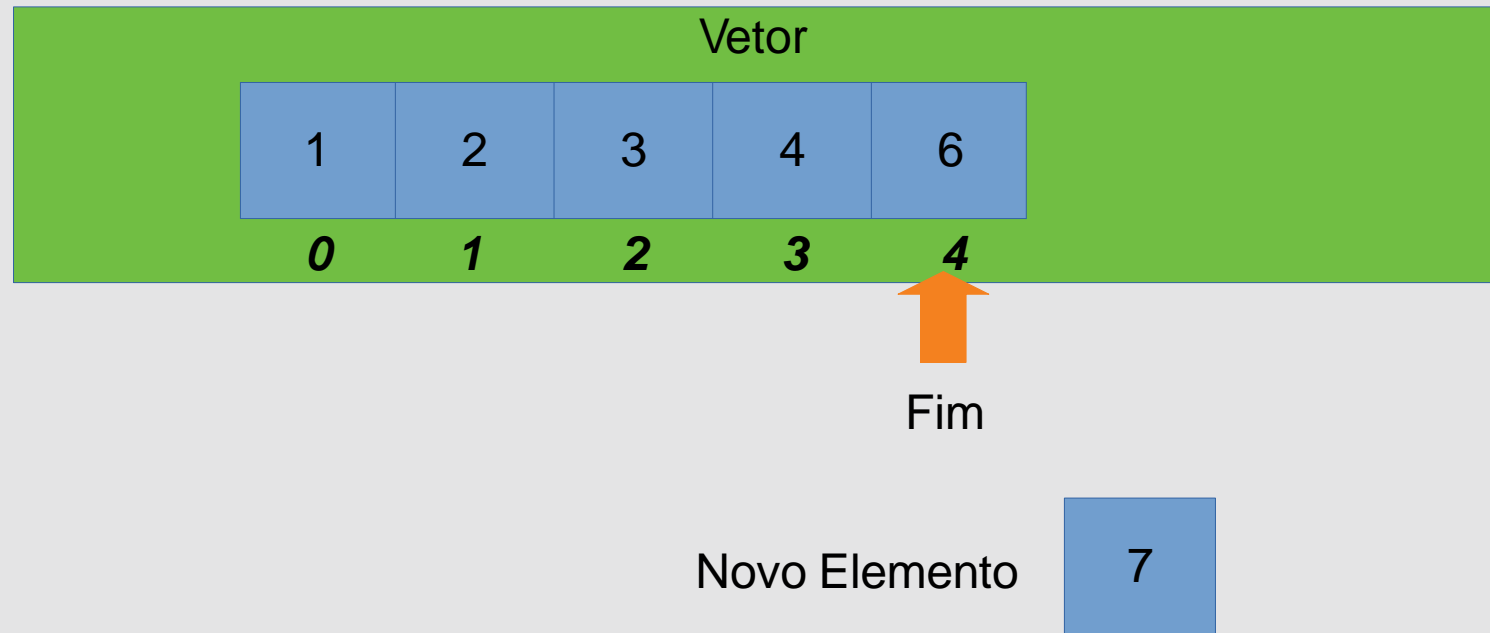
# enqueue – Fila Estática



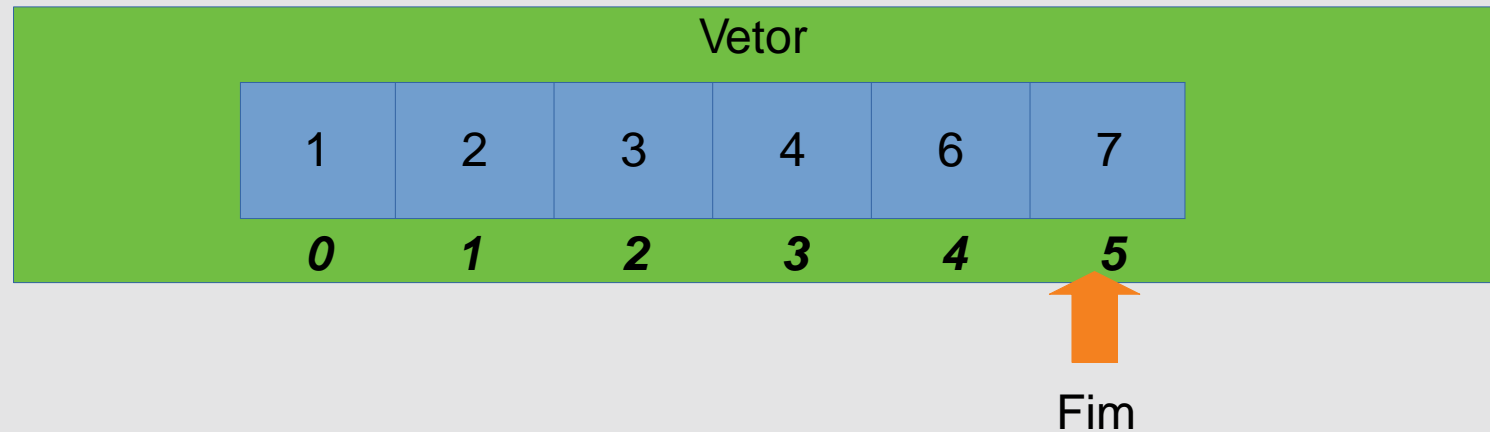
# enqueue – Fila Estática



# enqueue – Fila Estática



# enqueue – Fila Estática



# Código - isFull()

```
// verifica se a fila está cheia  
public boolean isFull() {  
    return ( fim == vetorFila.length-1 );  
}
```



# Código – enqueue

## Fila Estática

```
// insere na fila estática
public void enqueue(int j) {
    if ( !isFull() ){
        fim++;
        vetorFila[fim] = j;
    }
}
```

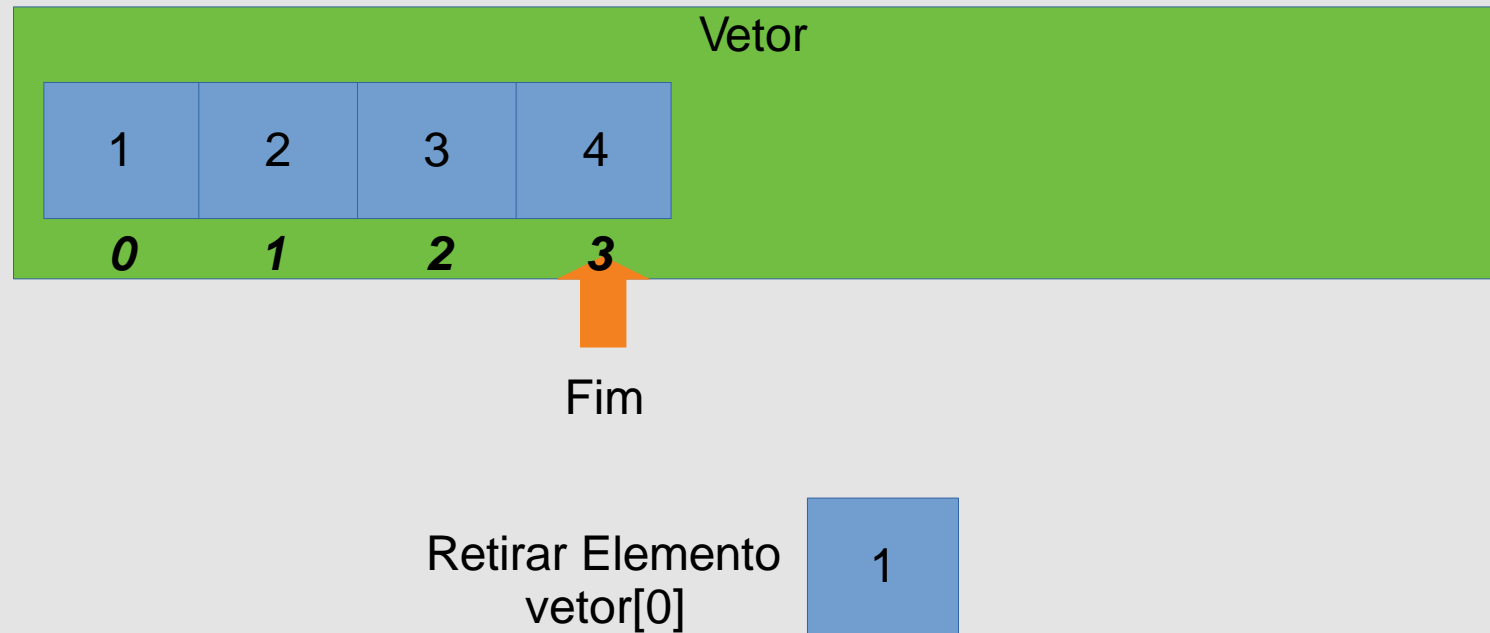
# Código display()

```
// imprime os elementos da fila
public void display() {
    System.out.println("*** Imprimindo a fila ***");

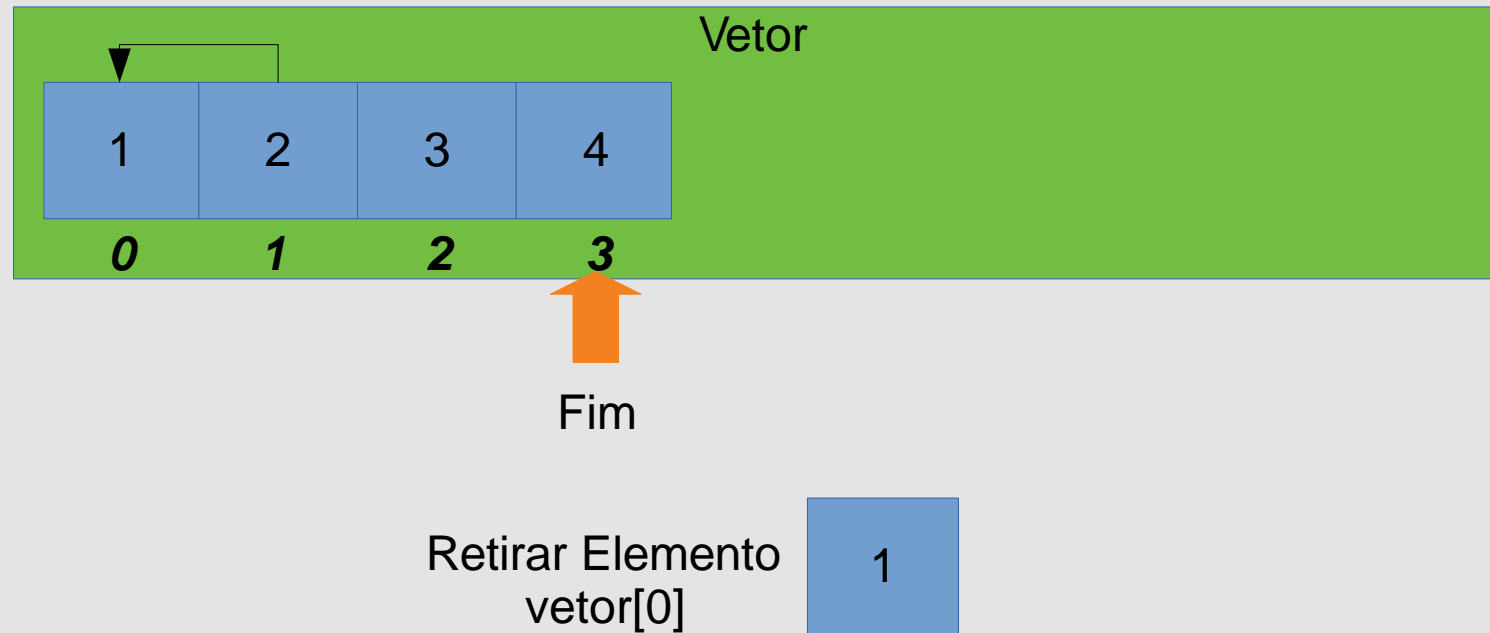
    for(int i=0; i <= fim; i++){
        System.out.print(vetorFila[i] + "
        ");
    }

    System.out.println("\n*****");
}
```

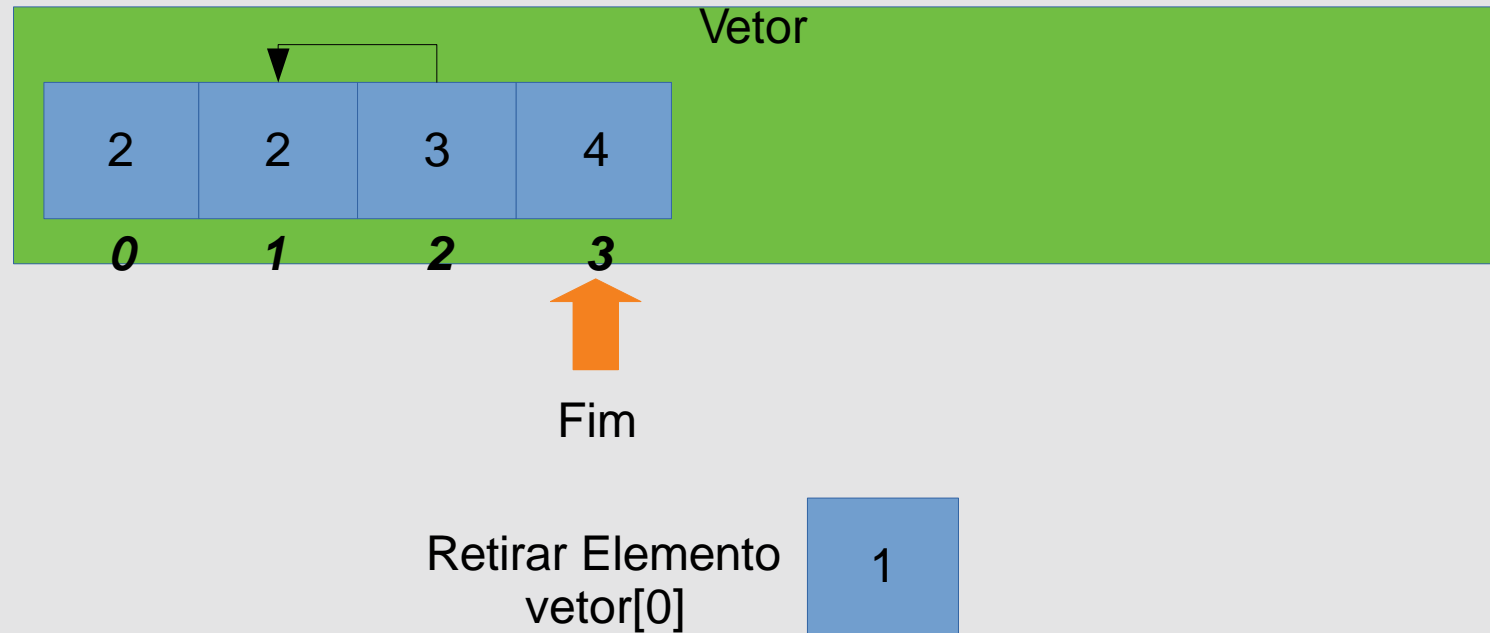
# dequeue – Fila Estática



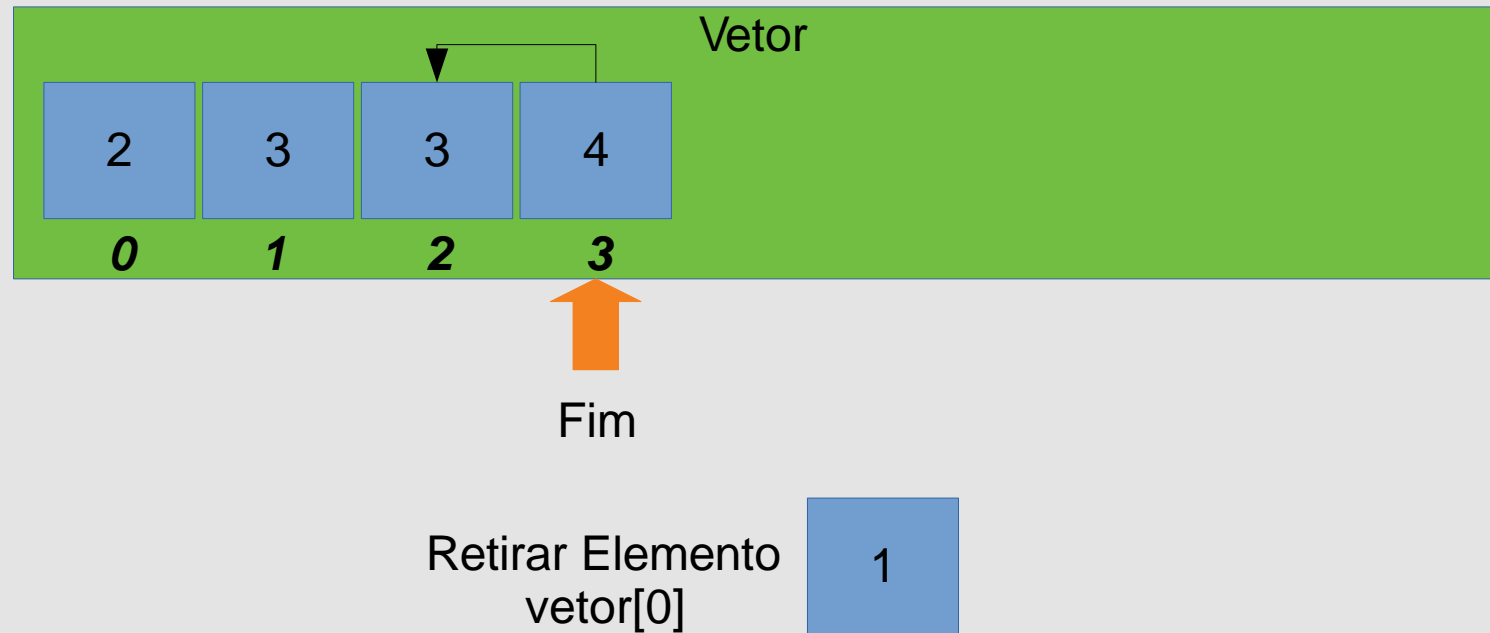
# dequeue – Fila Estática



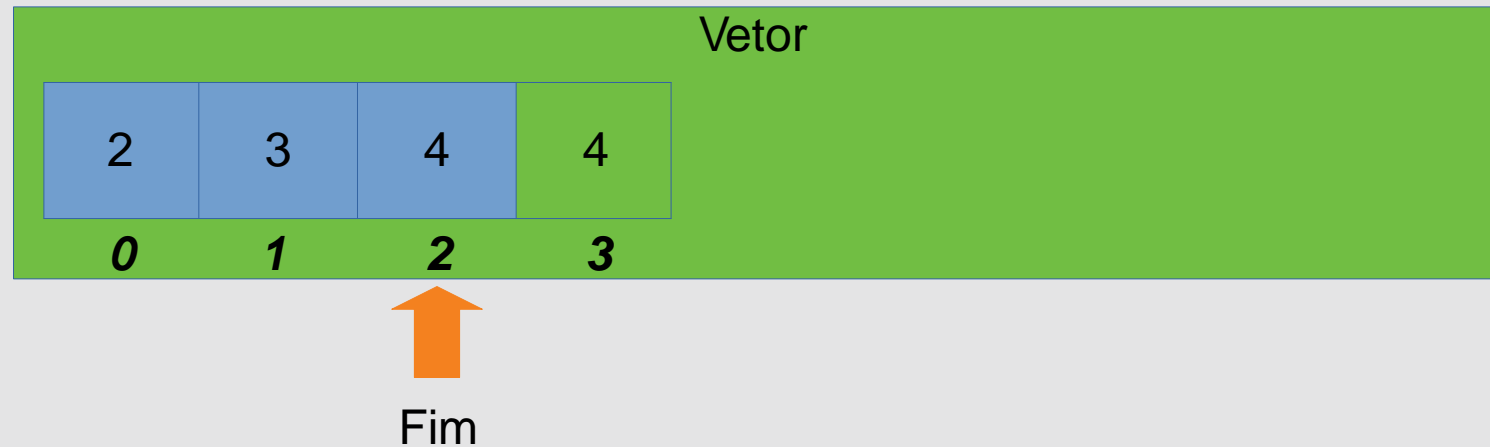
# dequeue – Fila Estática



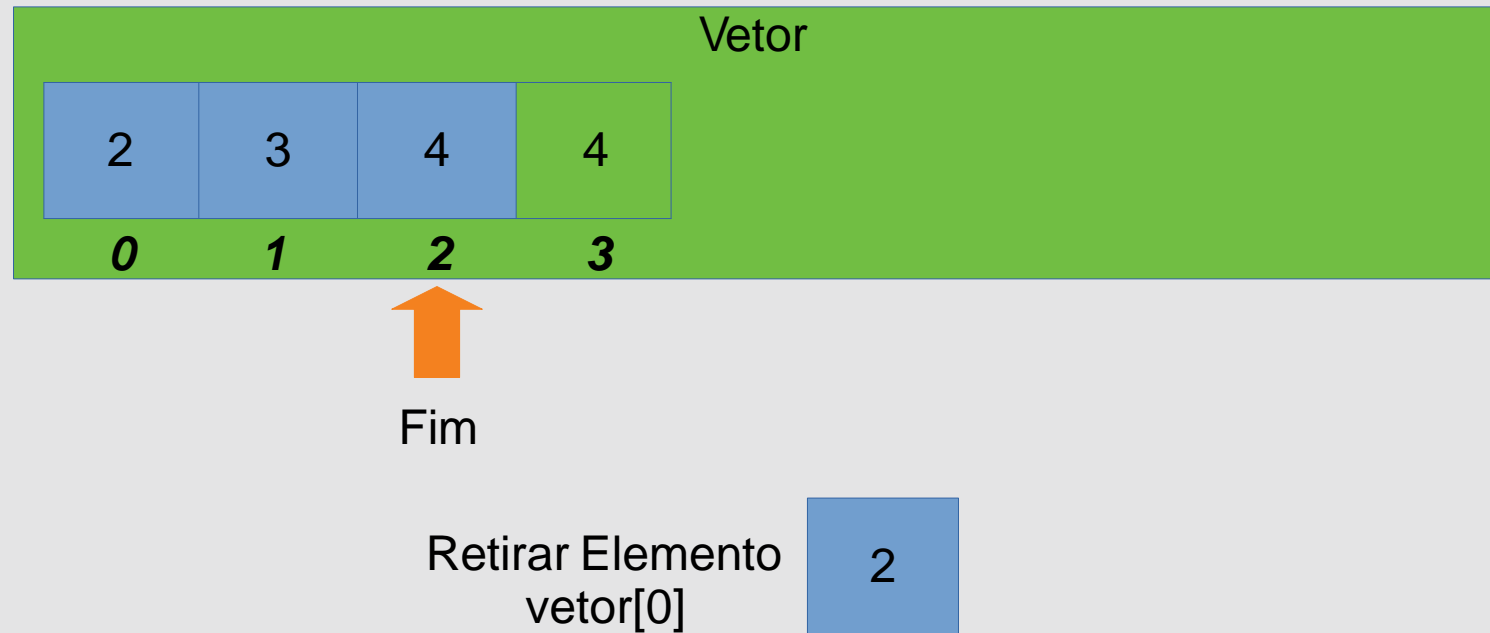
# dequeue – Fila Estática



# dequeue – Fila Estática

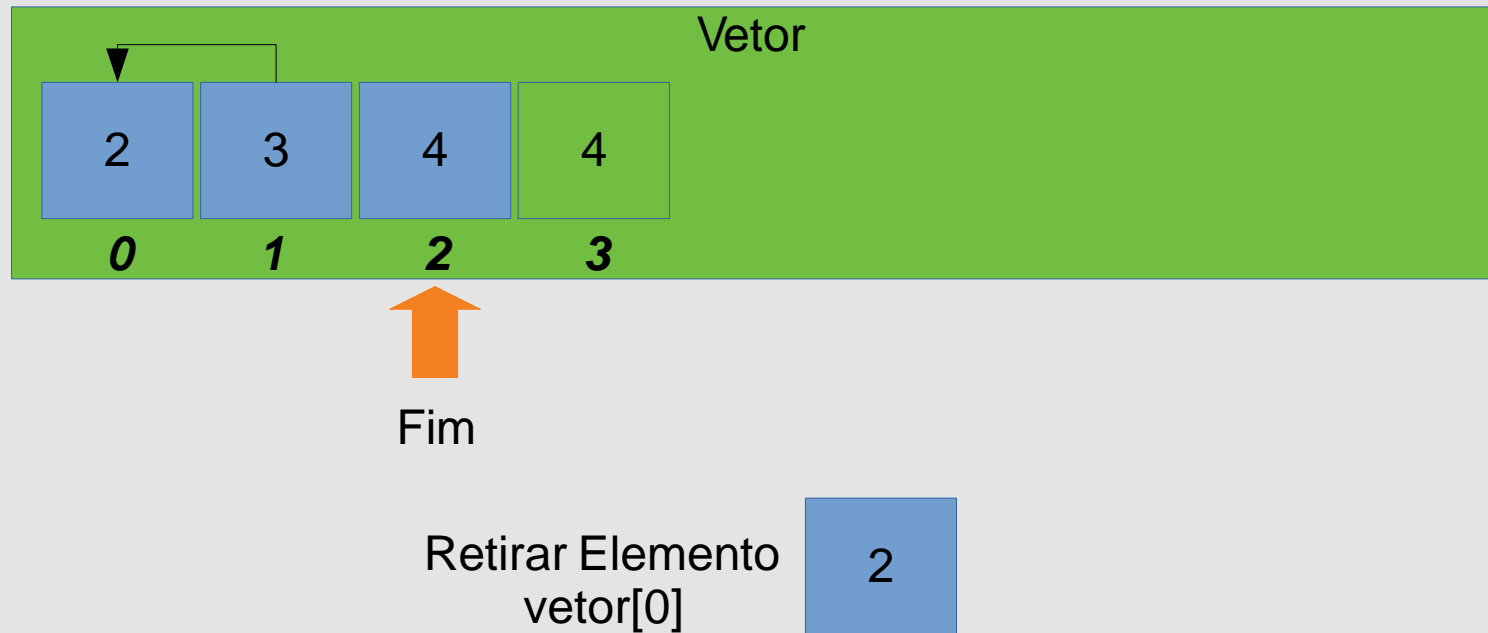


# dequeue – Fila Estática

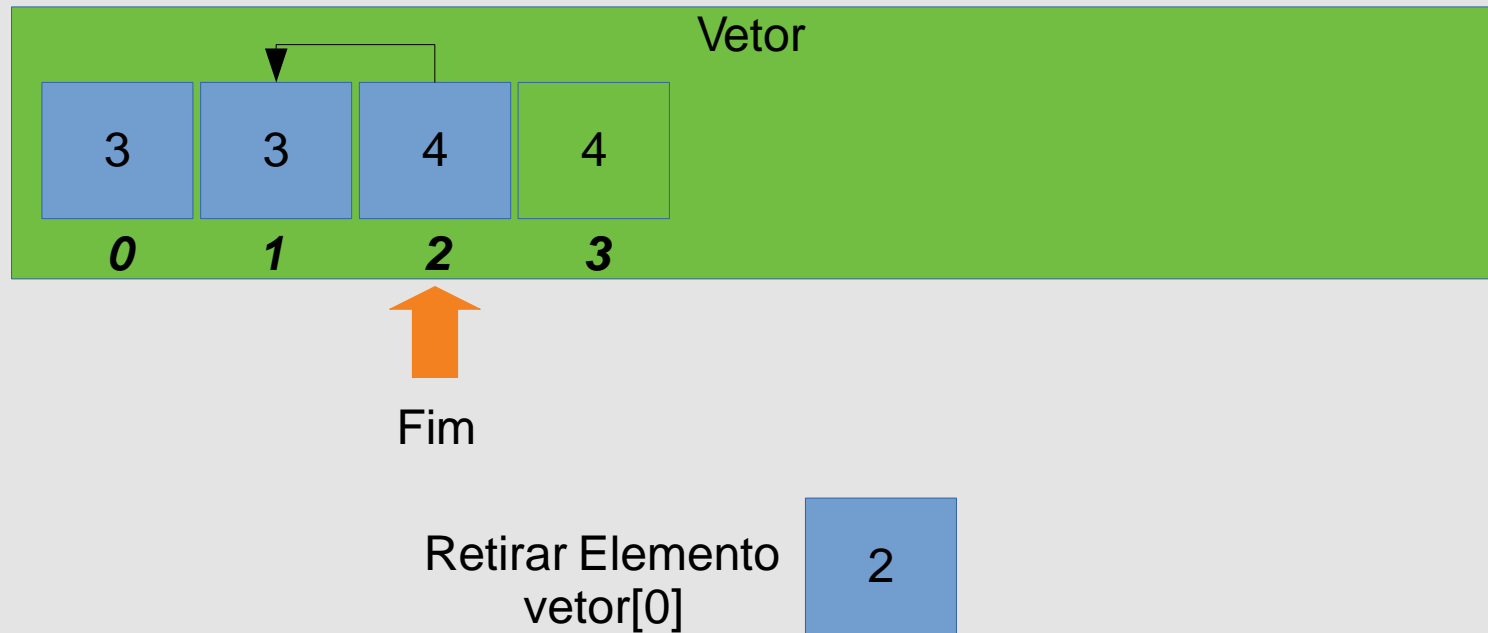




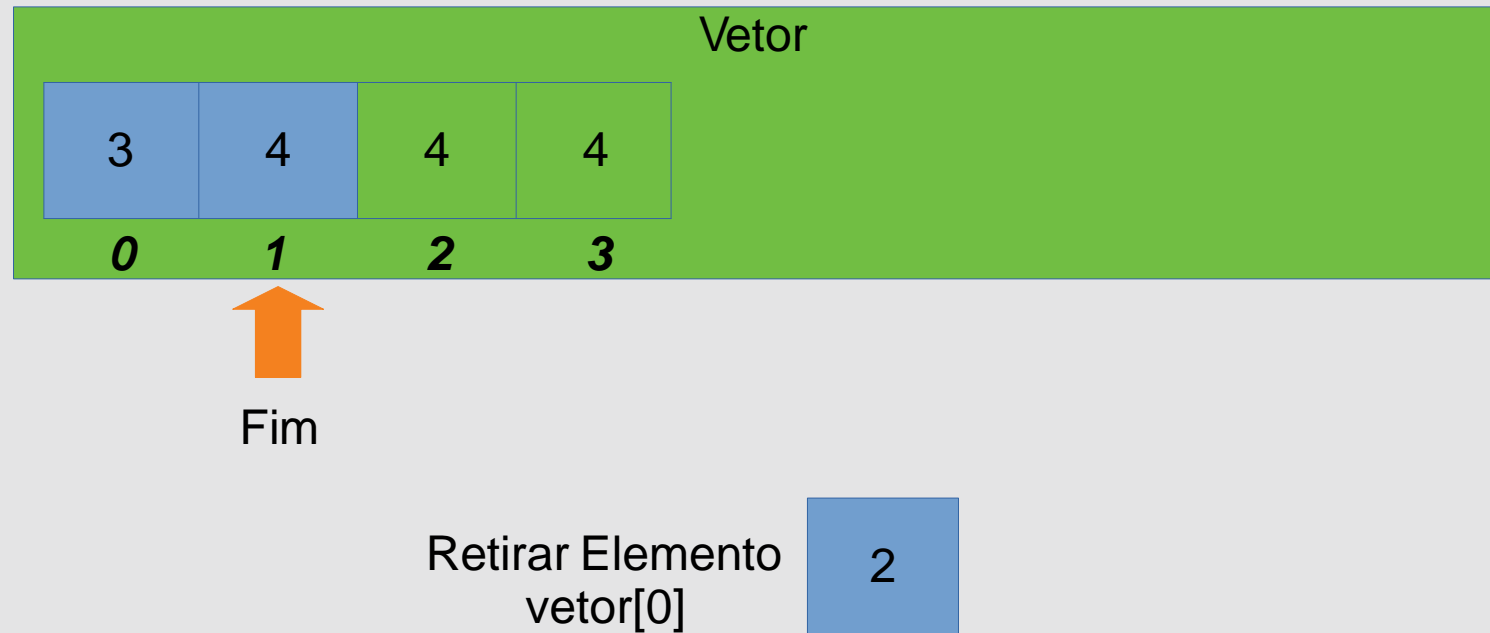
# dequeue – Fila Estática



# dequeue – Fila Estática



# dequeue – Fila Estática



# Código isEmpty()

```
// verifica se a fila está vazia  
public boolean isEmpty() {  
    return ( fim == -1 );  
}
```

# Código – dequeue

## Fila Estática

```
//Remove da fila Estática
public int dequeue() {
    int elemento;
    if (!isEmpty()) {
        elemento = vetorFila[0];
        for (int i = 0; i < fim; i++) {
            vetorFila[i] = vetorFila[i + 1];
        }
        fim--;
        return elemento;
    }
    else{
        return -1;
    }
}
```

# Código – peek

## Fila Estática

```
// retorna o primeiro elemento da Fila
public int peek(){
    if (!isEmpty()){
        return vetorFila[0];
    }
    else{
        return -1;
    }
}
```

.....

# Filas Circulares Estáticas

.....

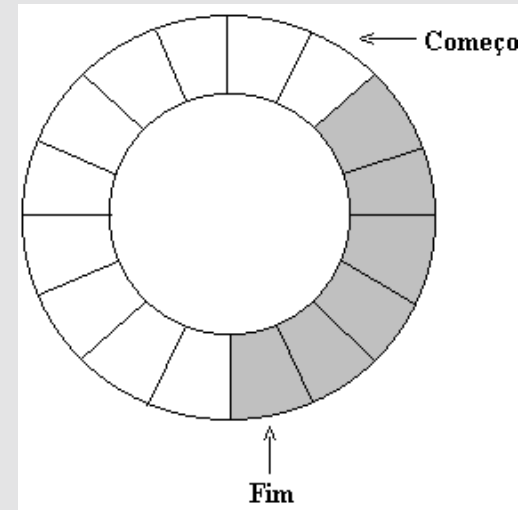


Universidade  
Anhembi Morumbi

LAUREATE INTERNATIONAL UNIVERSITIES

# Fila Circular Estática

- Note que a operação de remoção da implementação apresentada, **necessita deslocar os elementos uma posição à esquerda**, para manter sempre o primeiro elemento da fila na posição zero.
  - Uma alternativa de implementação, para evitar esse deslocamento, é assumir que a fila é circular
  - Por que?
    - Na fila circular temos os ponteiros de começo e fim





# Fila Circular Estática

- Na fila circular com capacidade de no máximo `maxElementos` elementos, precisamos **diferenciar** quando a fila está **vazia** e quando a fila está **cheia**.
- Para isso, assumimos que exista uma variável, `qtdElementos`, que é incrementada ou decrementada sempre que se insere ou se retira elementos da fila respectivamente.

# Código - Fila Circular Estática

```
public class FilaCircular {  
  
    private int vetorFila[];  
    private int fim;  
    private int inicio;  
    private int qtdElementos;  
  
    public FilaCircular(int maxSize) {  
        vetorFila = new int[ maxSize ];  
        fim = 0;  
        inicio = 0;  
        qtdElementos = 0;  
    }  
    // Métodos das operações da fila  
    // ...  
}
```

# Código – isFull

## Fila Circular Estática

```
//Verifica se a fila circular estática está cheia  
public boolean isFull() {  
  
    return (qtdElementos == this.vetorFila.length);  
  
}
```

# Código – isEmpty

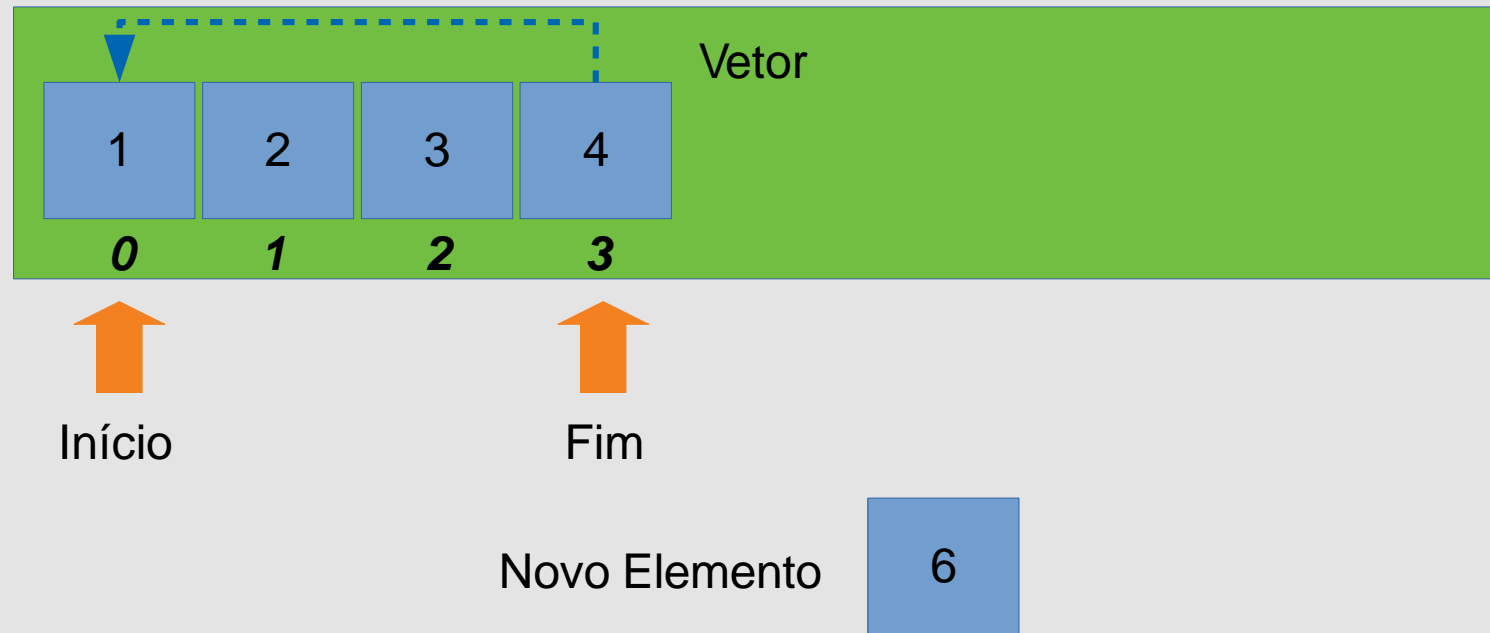
## Fila Circular Estática

```
//Verifica se a fila circular estática está vazia
public boolean isEmpty() {

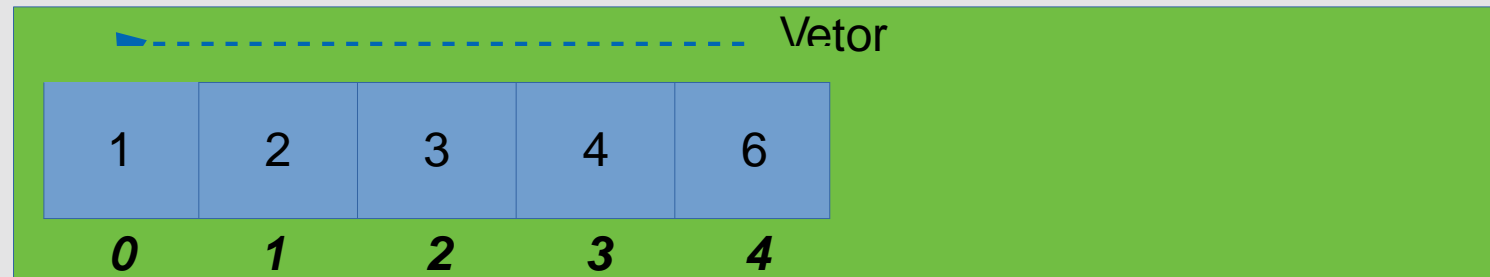
    return (qtdElementos == 0);

}
```

# enqueue – Fila Circular Estática



# enqueue – Fila Circular Estática



↑  
Início

↑  
Fim

QtdElementos ++  
Antes: 4  
Depois: 5

# enqueue – Fila Circular Estática

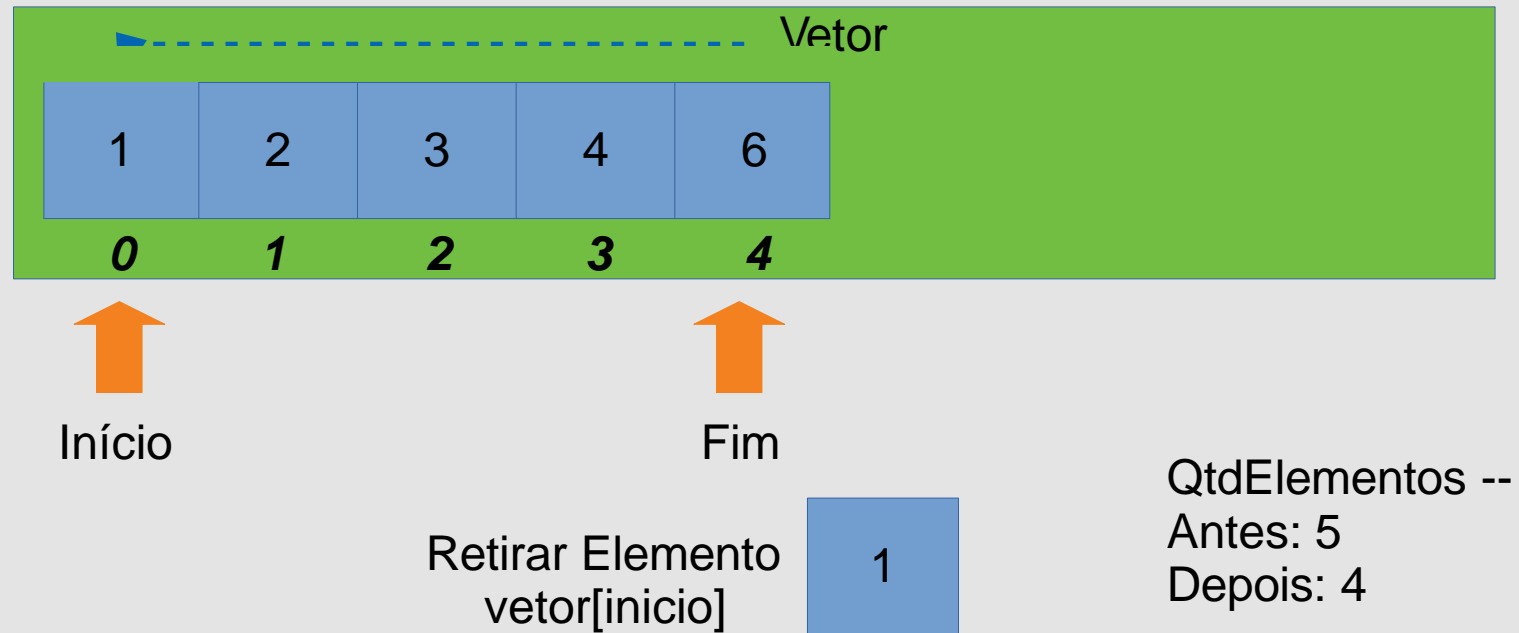
```
//Insere um elemento na fila
public void enqueue(int dado) {
    if (!isFull()) {
        vetorFila[fim] = dado;
        fim++;
        qtdeElementos++;
        if(fim == vetorFila.length) {
            fim = 0;
        }
    }
}
```

# display – Fila Circular Estática

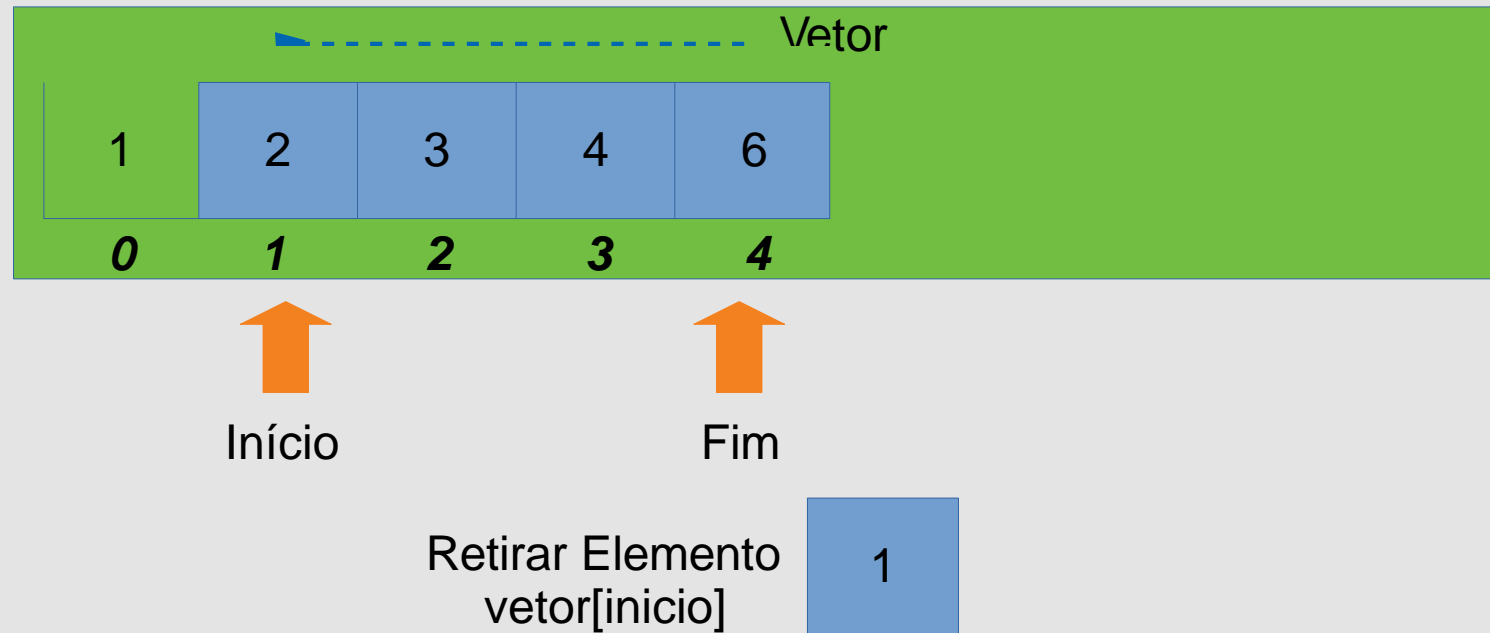
```
//Imprime os elementos da fila
public void display() {
    System.out.println("*** imprimindo a fila ***");
    int max = qtdeElementos + inicio;
    for( int i = inicio; i < max; i++){
        System.out.print(vetorFila[i % vetorFila.length] + " ");
    }
    System.out.println("\n*****");
}
```



# dequeue – Fila Estática



# dequeue – Fila Estática



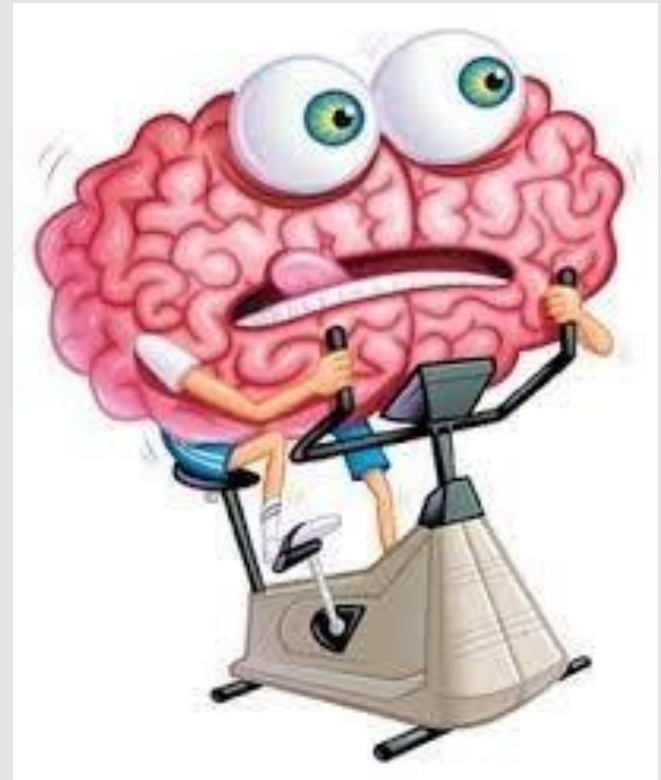
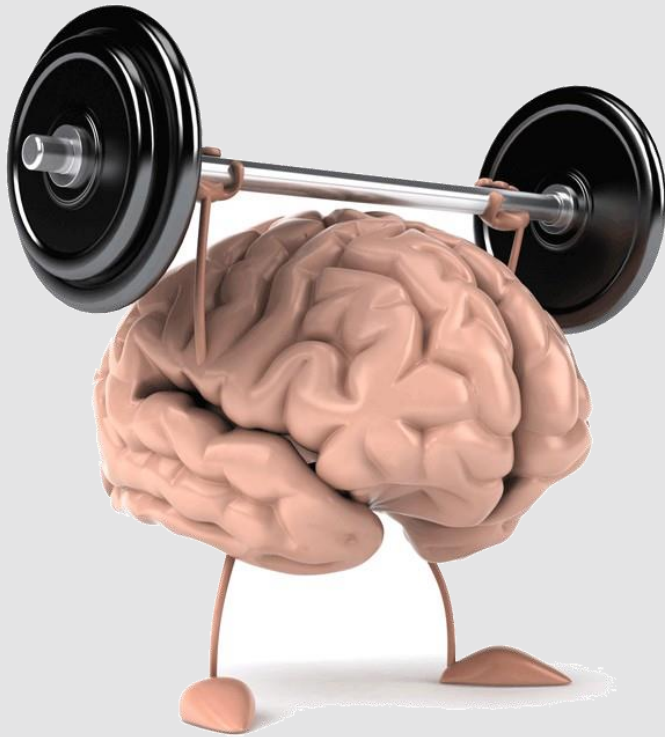
# dequeue – Fila Circular Estática

```
//remove um elemento da fila
public int dequeue() {
    if (!isEmpty()) {
        int elemento = vetorFila[inicio];
        qtdeElementos--;
        inicio++;
        if(inicio == vetorFila.length) {
            inicio = 0;
        }
        return elemento;
    }
    else{
        return -1;
    }
}
```

# Dúvidas



# Exercícios



# Exercícios

- 1) Modifique o TAD Fila para armazenar caracteres ao invés de números e teste.
- 2) Escreva o TAD Fila Circular conforme as explicações anteriores e teste.

# Referências

- <http://www.ic.unicamp.br/~ra069320/PED/MC102/1s2008/Apostilas/Cap12.pdf>

.....

# Obrigado

.....

Adaptado das aulas do professor Bruno Moritani  
[igor.borges@anhembi.br](mailto:igor.borges@anhembi.br)



Universidade  
Anhembi Morumbi

LAUREATE INTERNATIONAL UNIVERSITIES