

Estruturas de Dados

Prof. Rodrigo Martins
rodrimartins2005@gmail.com

Cronograma da Aula

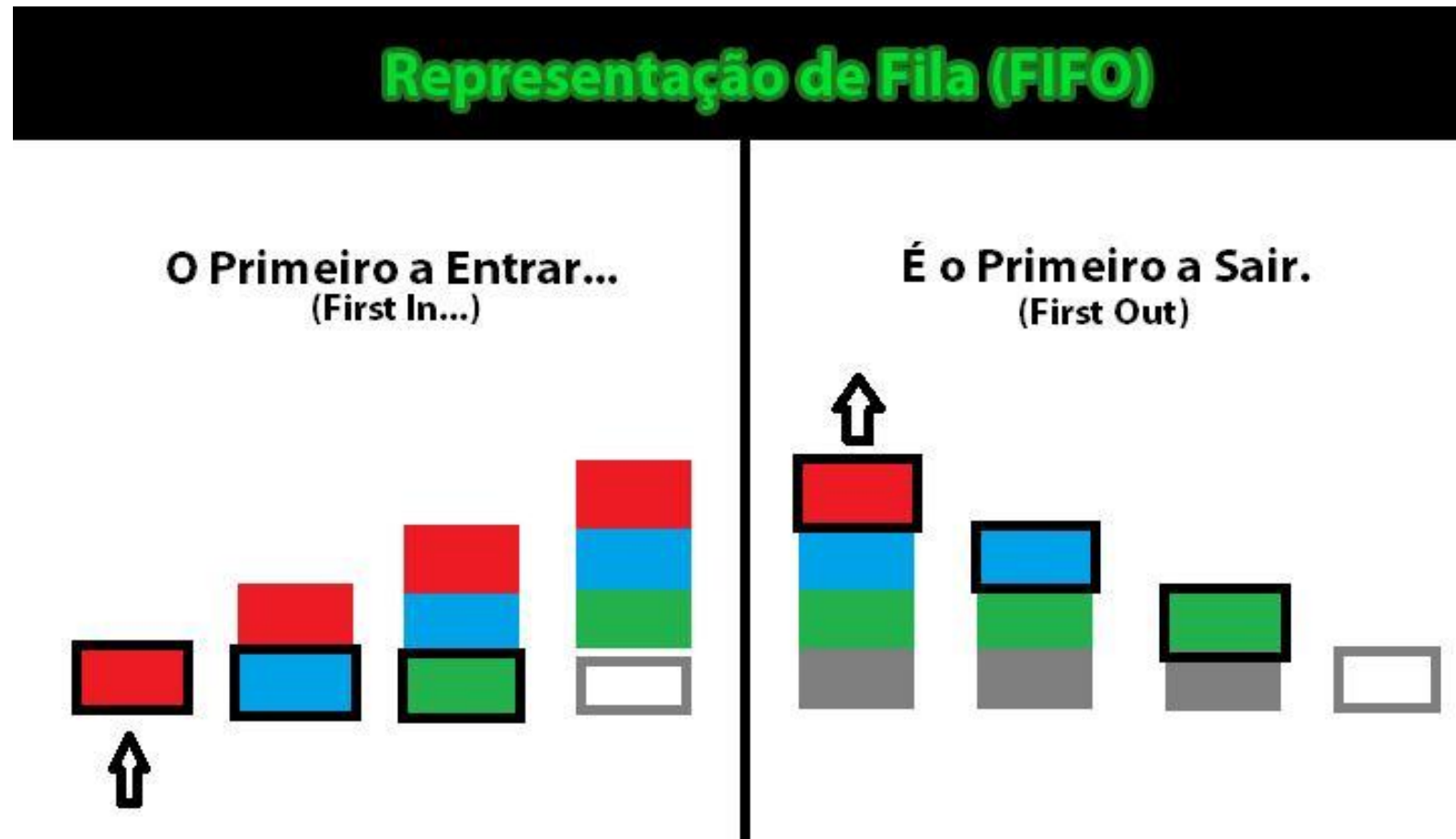
- Filas
- Exemplos
- Exercícios

Fila

- Uma fila é uma estrutura na qual os dados são inseridos em um extremo e retirados no outro extremo.
- São também chamadas de FIFO (“First In First Out” = “Primeiro a Entrar e o Primeiro a Sair”) ou LIFO (“Last In Last Out” = “Último a Entrar e o Último a Sair”).

Fila

- Sua representação pode ser vista abaixo:



Filas - Exemplos

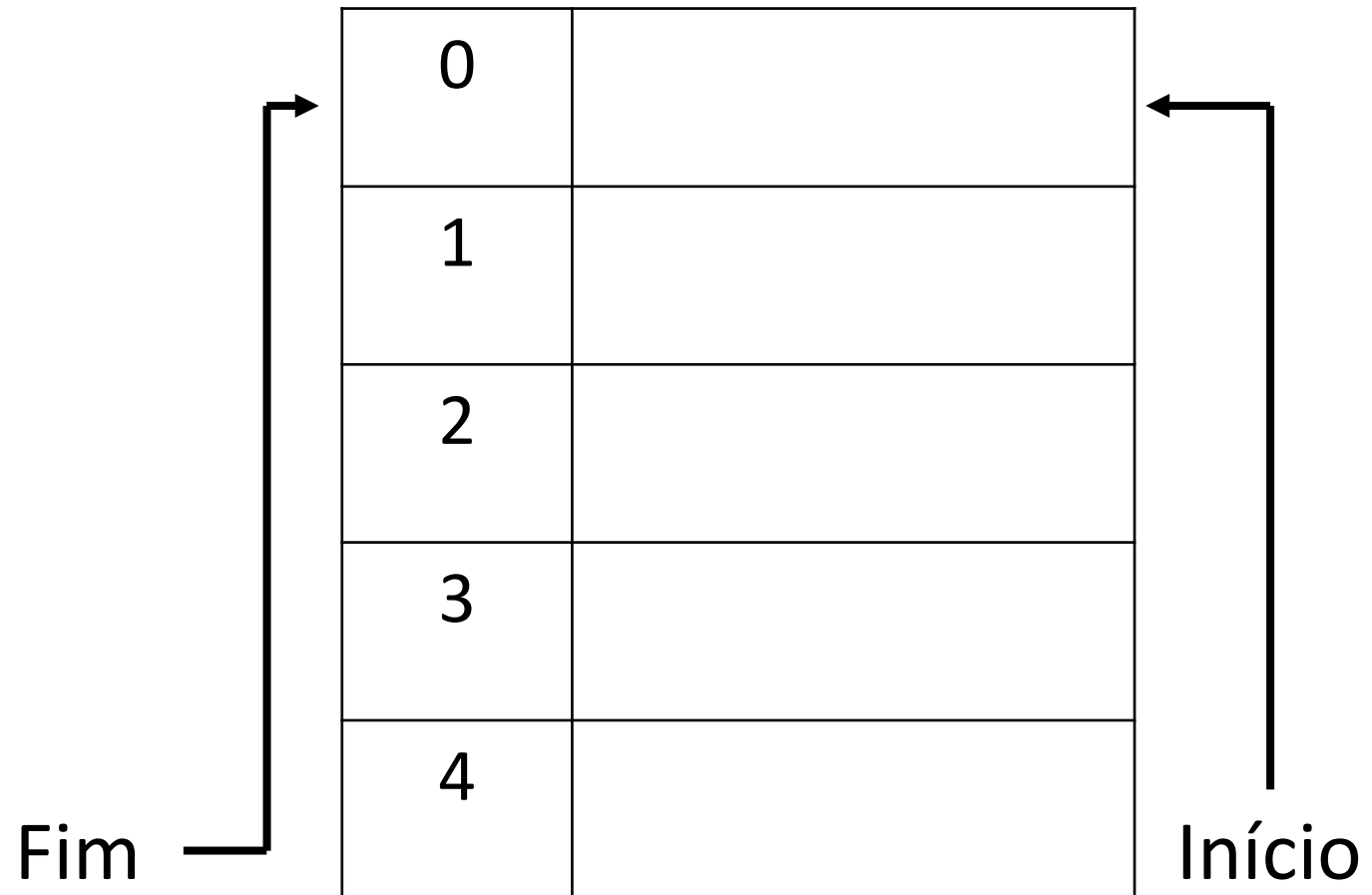
- São exemplos de uso de fila em um sistema:
 - Controle de documentos para impressão;
 - Troca de mensagens entre computadores numa rede;
 - Entre outros.

Fila

- A implementação desta estrutura requer a utilização de um vetor, onde um elemento é inserido no final da fila (a direita) e retirado do início da fila (a esquerda).
- Esta implementação é dificultada devido ao fato da fila se locomover dentro do vetor. Se nada for feito para controlar esta locomoção, a fila atingirá o final do vetor (lado direito) e a próxima inserção não poderá ser feita, mesmo que haja espaço do lado esquerdo do vetor.
- Uma forma de resolver este problema é: sempre que um elemento for removido (lado esquerdo) deslocar os demais elementos da fila uma casa a esquerda.

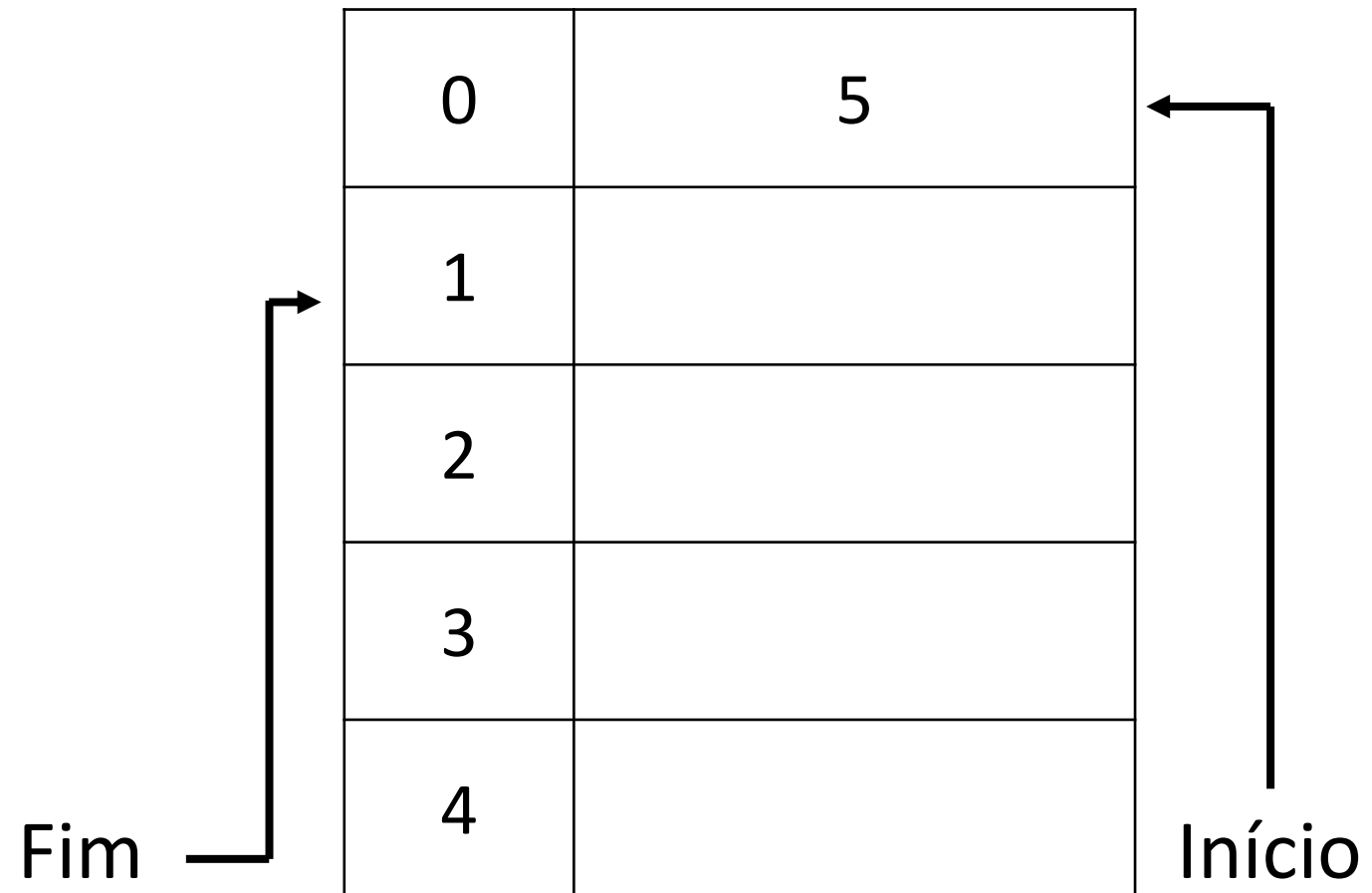
Fila Simples

- Novamente precisamos de um vetor de inteiros.
- Precisamos também de uma variável indicando o início e o fim da fila.
- Tanto o início quanto o final da fila apontam para o começo do vetor.



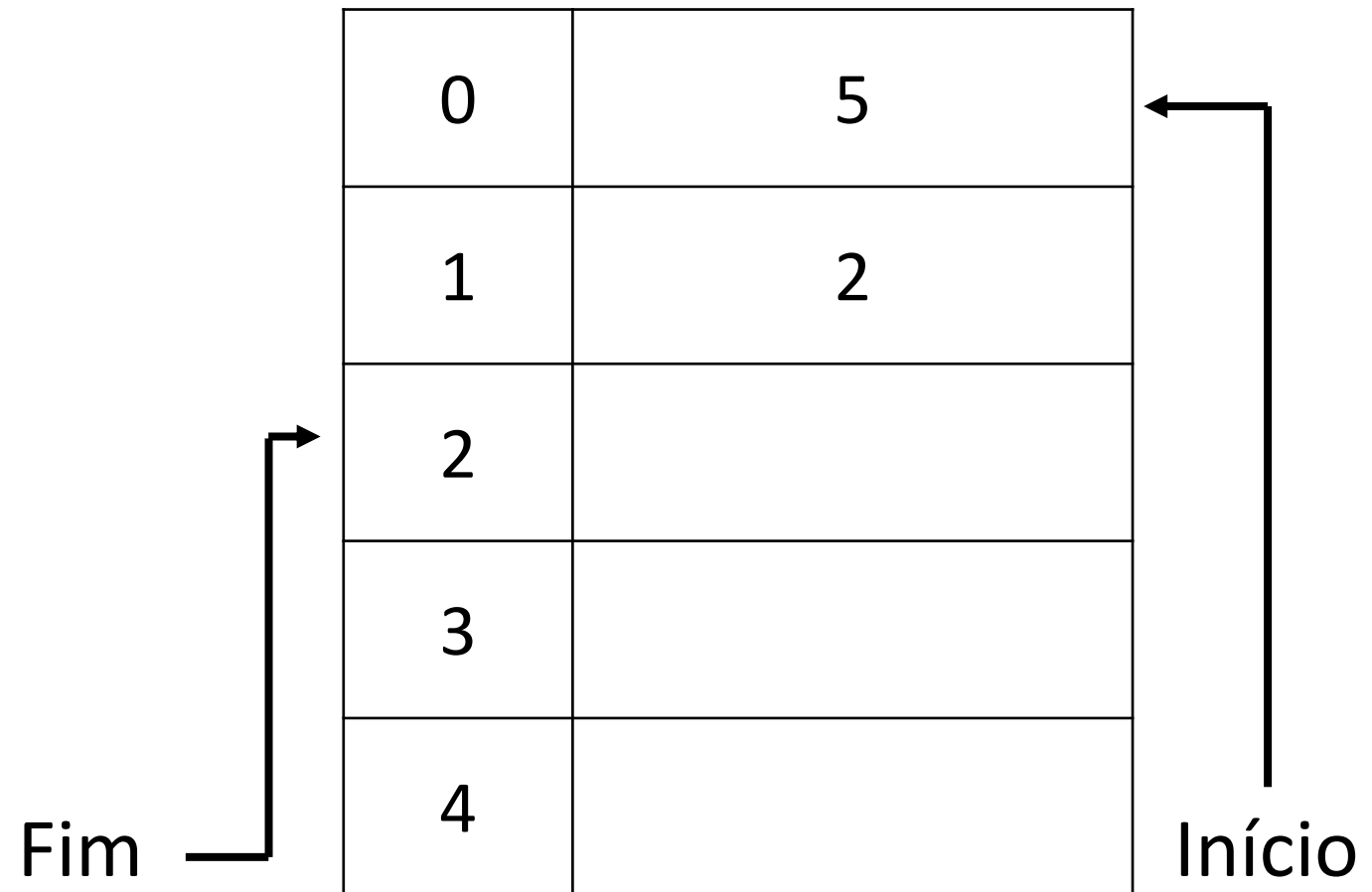
Fila Simples

- Para adicionarmos um elemento na fila devemos colocá-lo na posição apontada por fim e depois incrementarmos o fim.



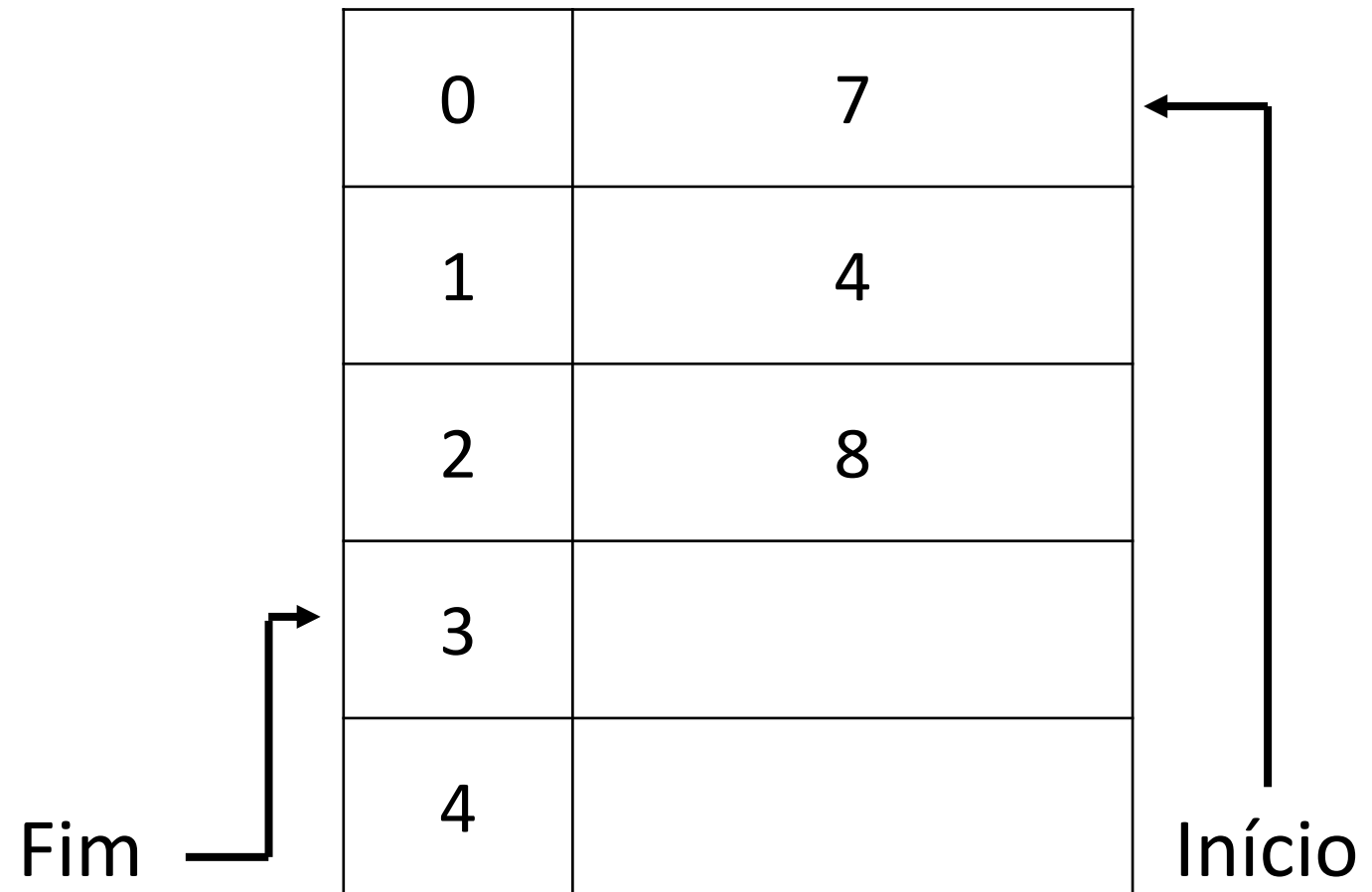
Fila Simples

- Completando a fila....



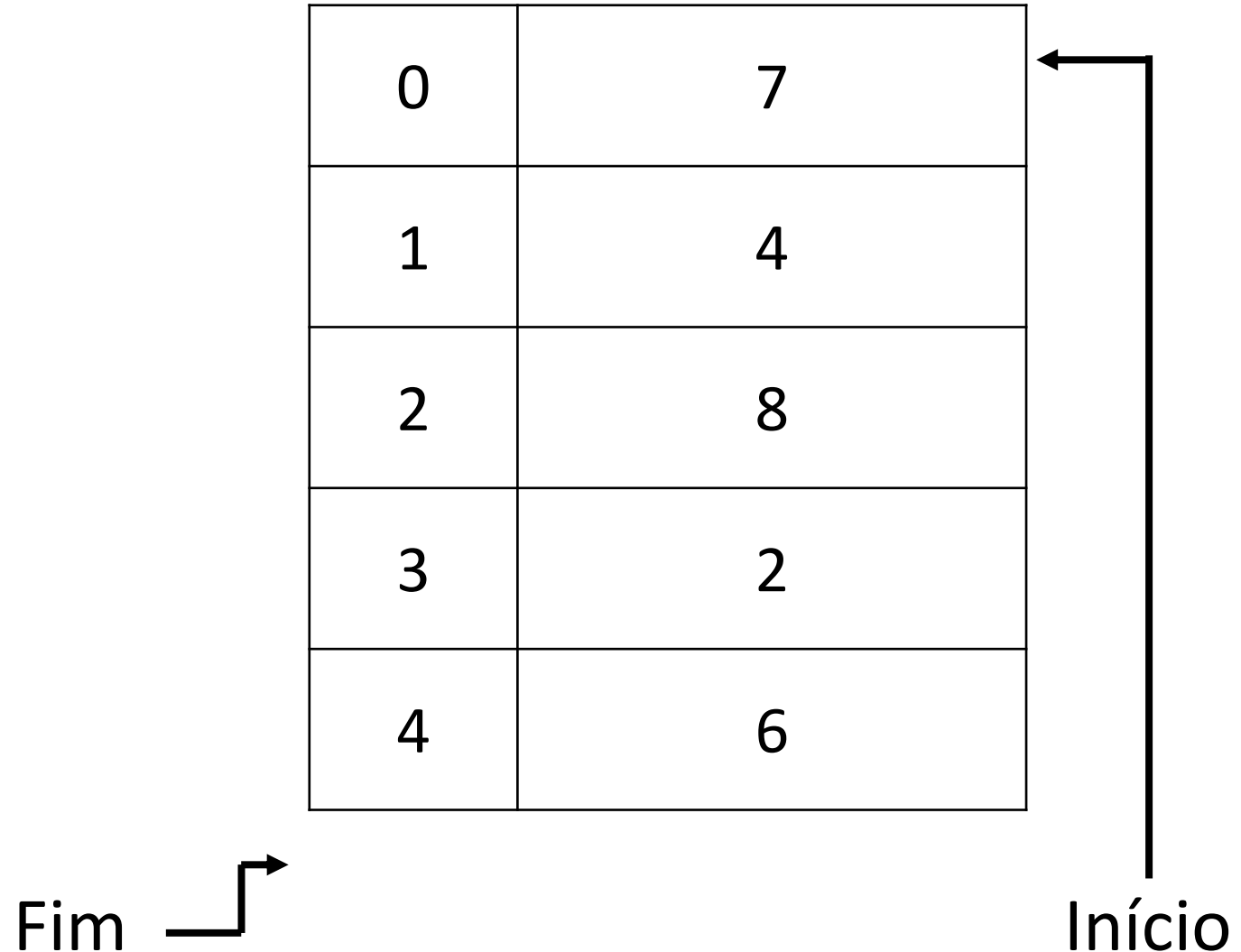
Fila Simples

- Completando a fila....



Fila Simples


- Ao chegar no fim a fila está cheia.
- Não é possível mais entrar outros elementos.
- Na fila simples, nem mesmo se alguém sair da fila, outros elementos não podem mais entrar.



Fila Simples

- A retirada dos elementos na fila se faz copiando o elemento para uma outra variável.
- Incrementamos o início e devolvemos o conteúdo da variável.

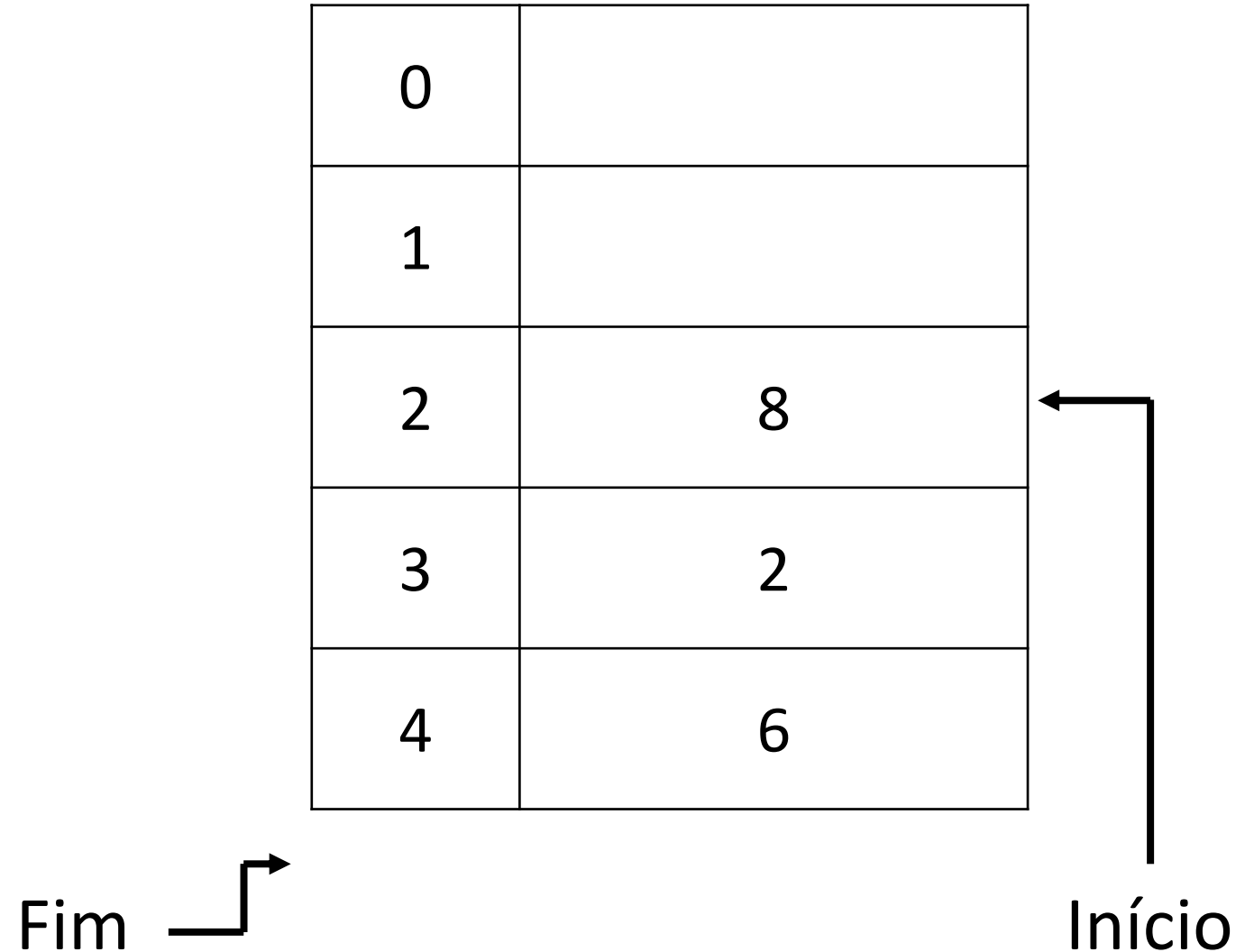
0	7
1	4
2	8
3	2
4	6

Fim 

 Início

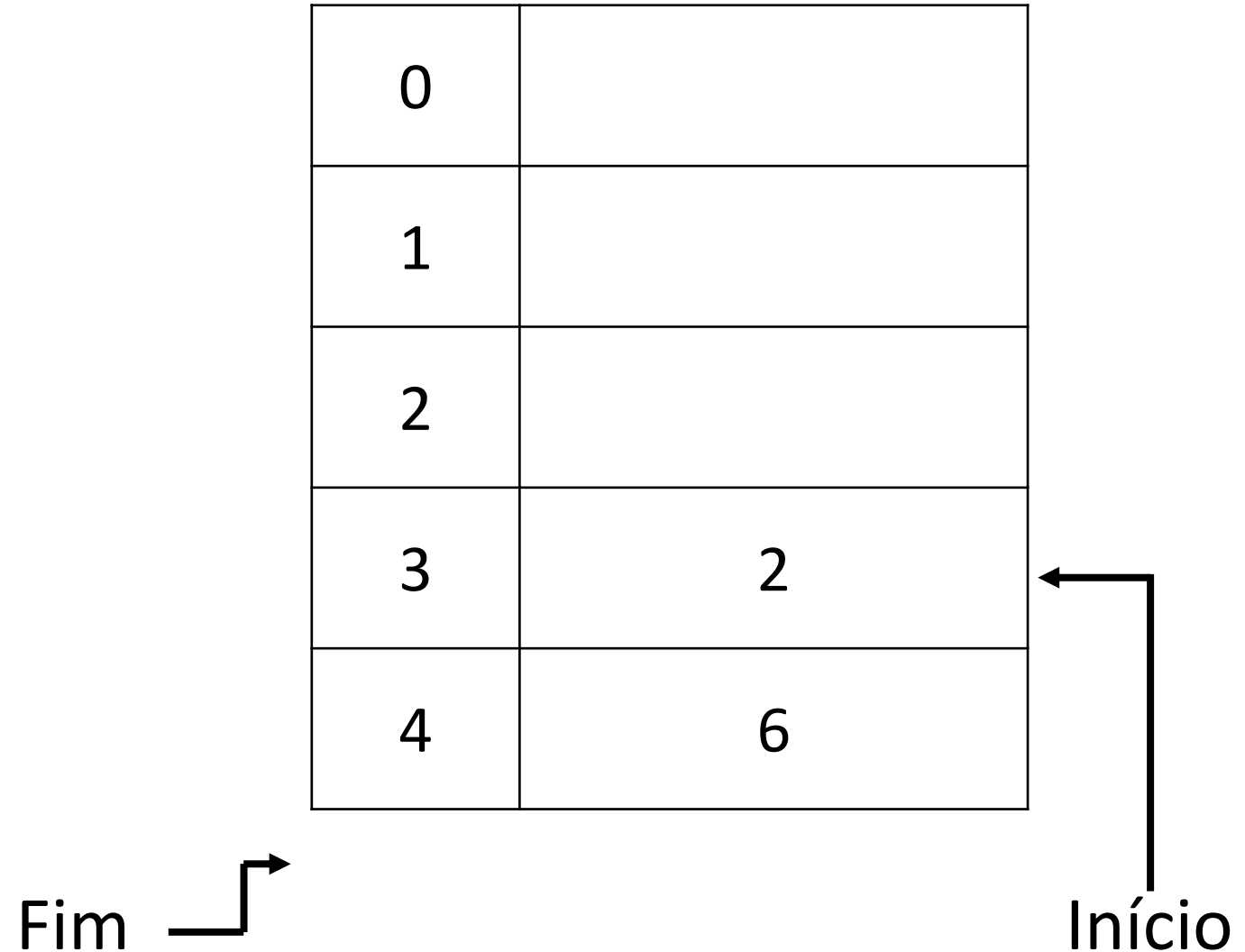
Fila Simples

- Assim retiramos todos...



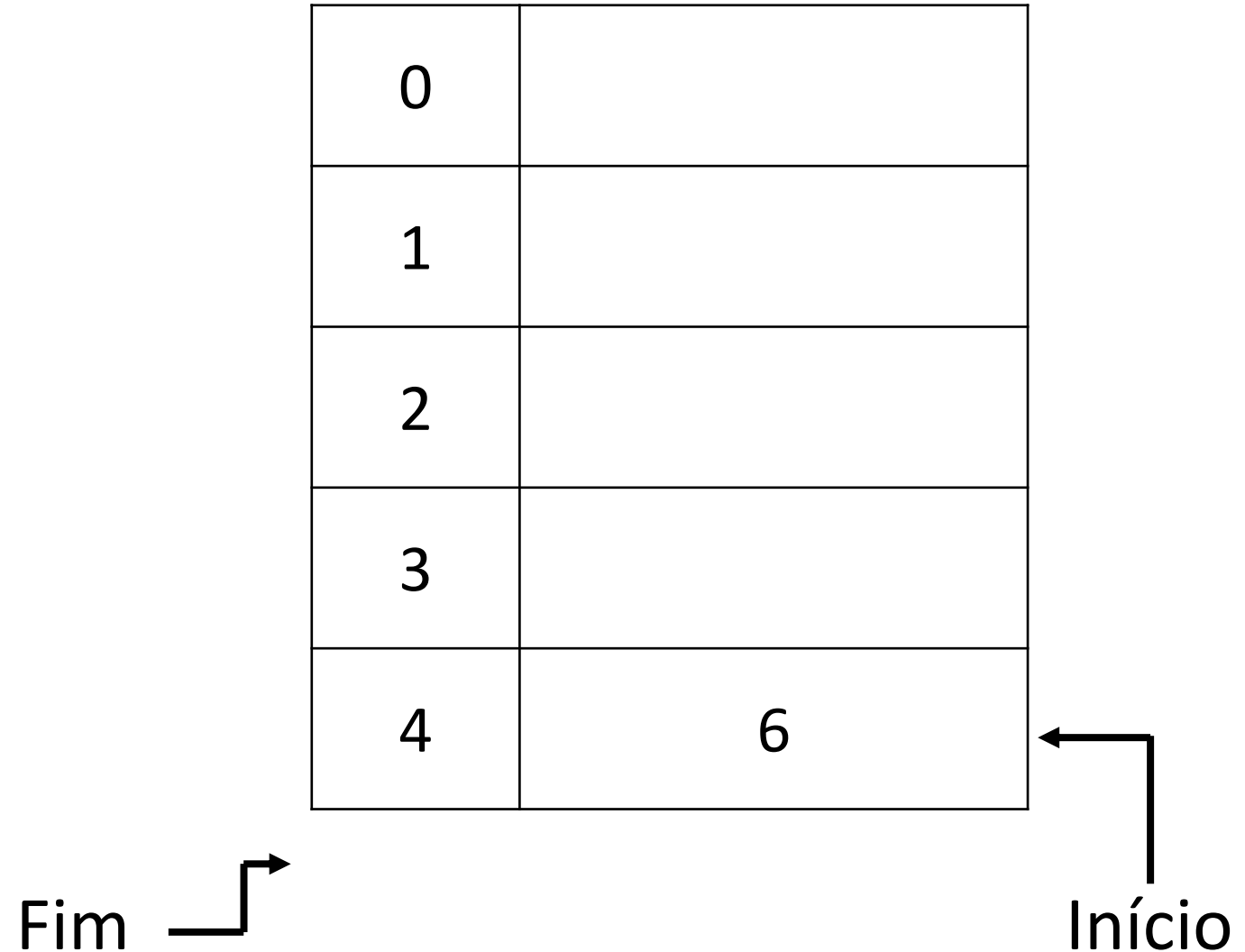
Fila Simples

- Assim retiramos todos...



Fila Simples

- Assim retiramos todos...



Fila Simples

- Assim retiramos todos...

0	
1	
2	
3	
4	

Fim ↗

↖ Início

Exemplo de Fila Estática

*filaSimples.cpp X

```
1  #include <iostream>
2  #include <string>
3
4  #define TAM 10
5
6  using namespace std;
7
8  void iniciaPosFila(int *inicio, int *fim)
9  {
10     *inicio = 0;
11     *fim = -1;
12 }
13
14
15 void imprimeVetor(int vetor[TAM], int tamanho)
16 {
17
18     cout << endl;
19
20     for (int cont = 0; cont < TAM; cont++)
21     {
22         cout << vetor[cont] << endl;
23     }
24 }
25
```

Exemplo de Fila Estática

```
26  int retornaTamanho(int fim, int inicio)
27  {
28      return (fim - inicio) + 1;
29  }
30
31  bool filaCheia(int fim)
32  {
33      if (fim == TAM - 1)
34      {
35          return true;
36      }
37      else
38      {
39          return false;
40      }
41  }
42
43  bool filaVazia(int inicio, int fim)
44  {
45      if (inicio > fim)
46      {
47          return true;
48      }
49      else
50      {
51          return false;
52      }
53  }
```

Exemplo de Fila Estática

```
54
55 void enfileirar(int fila[TAM], int valor, int *fim)
56 {
57     if (filaCheia(*fim))
58     {
59         cout << "FILA CHEIA";
60     }
61     else
62     {
63         *fim = *fim + 1;
64         fila[*fim] = valor;
65     }
66 }
67
```

Exemplo de Fila Estática

```
68 void desenfileirar(int fila[TAM], int *inicio, int fim)
69 {
70
71     if (filaVazia(*inicio, fim))
72     {
73         cout << "IMPOSSIVEL DESINFELEIRAR FILA VAZIA";
74     }
75     else
76     {
77         cout << "O valor " << fila[*inicio] <<
78         " foi removido" << endl;;
79         fila[*inicio] = 0;
80         *inicio = *inicio + 1;
81     }
82 }
83
```

Exemplo de Fila Estática

```
84 int main(int argc, char** argv)
85 {
86     int fila[TAM] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0};
87     int inicio, fim;
88
89     iniciaPosFila(&inicio, &fim);
90
91     enfileirar(fila, 10, &fim);
92     enfileirar(fila, 9, &fim);
93     enfileirar(fila, 8, &fim);
94     enfileirar(fila, 7, &fim);
95     enfileirar(fila, 6, &fim);
96     enfileirar(fila, 5, &fim);
97     enfileirar(fila, 4, &fim);
98     enfileirar(fila, 3, &fim);
99     enfileirar(fila, 2, &fim);
100    enfileirar(fila, 1, &fim);
101
102    imprimeVetor(fila, retornaTamanho(fim, inicio));
103
104    desenfileirar(fila, &inicio, fim);
105    desenfileirar(fila, &inicio, fim);
106    desenfileirar(fila, &inicio, fim);
107
108    imprimeVetor(fila, retornaTamanho(fim, inicio));
109
110    return 0;
111 }
112
```

Exemplo de Fila Dinâmica

*filaDinamica.cpp X

```
4     using namespace std;
5
6     int main(int argc, char** argv)
7     {
8         /*
9         empty = vazio
10        size = tamanho
11        front =
12        back =
13        push =
14        pop =
15        */
16
17        queue <string> pessoas;
18
19        pessoas.push("RODRIGO"); //push adiciona elementos na fila
20        pessoas.push("ANDRE");
21        pessoas.push("MARIA");
22        pessoas.push("ANA");
23        pessoas.push("JOSE");
24        pessoas.push("RITA");
25
```

Exemplo de Fila Dinâmica

```
26 // size mostra o tamanho da fila
27 cout << "Tamanho da Fila: " << pessoas.size() << endl;
28 cout << "Primeira pessoa " << pessoas.front() << endl;
29 cout << "Ultima pessoa " << pessoas.back() << endl << endl;
30
31
32 while(!pessoas.empty()){
33     cout << "Primeira pessoa " << pessoas.front() << endl;
34     pessoas.pop(); // retira elementos da pilha
35 }
36
37
38 return 0;
39 }
```

Exercício

1. Escreva um programa em C++ que simule um sistema de atendimento em uma loja. O programa deve utilizar uma fila para gerenciar a ordem de atendimento dos clientes.

- O programa deve oferecer as seguintes funcionalidades:
- Inserir cliente na fila:
 - O programa deve solicitar ao usuário o nome do cliente que deseja ser atendido e inserir o cliente na fila de atendimento.
- Atender próximo cliente:
 - O programa deve remover o próximo cliente da fila e exibir uma mensagem com o nome do cliente que está sendo atendido.
- Exibir fila de espera:
 - O programa deve exibir a lista de clientes que estão aguardando na fila, na ordem em que foram inseridos.
- Encerrar o programa:
 - O programa deve permitir ao usuário encerrar o programa.

Referência desta aula

- Notas de Aula do Prof. Prof. Armando Luiz N. Delgado baseado em revisão sobre material de Prof.a Carmem Hara e Prof. Wagner Zola
- <http://www.cplusplus.com/reference/>

Obrigado

Rodrigo