



INSTITUTO FEDERAL DE  
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA  
SUL-RIO-GRANDENSE

# Estrutura de Dados

## Filas estáticas

Prof. Silvana Teodoro

[silvanateodoro@charqueadas.ifsul.edu.br](mailto:silvanateodoro@charqueadas.ifsul.edu.br)

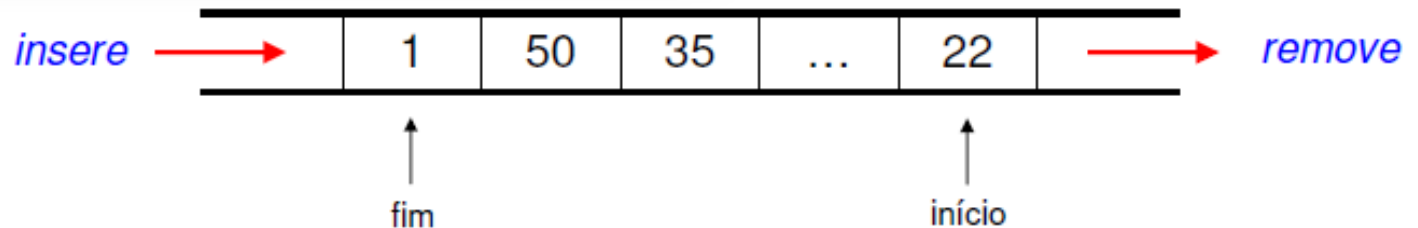
# Sumário

- Filas
- Funções
- Atividade
- Referências
- Leitura recomendada



# Filas

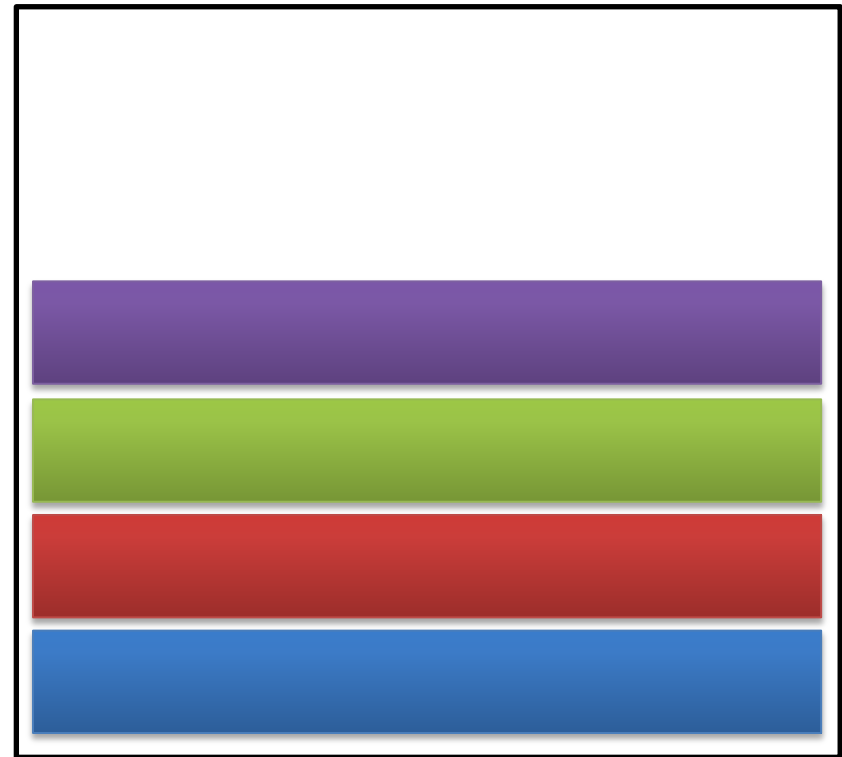
- Um **novo elemento** é **inserido no final** da fila e um elemento é **retirado do início** da fila.



- Diferença:
  - **Fila** = “o primeiro que entra é o primeiro que sai” (**FIFO**)
  - **Pilha** = “o último que entra é o primeiro que sai” (**LIFO**)

# Ordem de Entrada x Saída

- O que ocorre na fila de um banco?

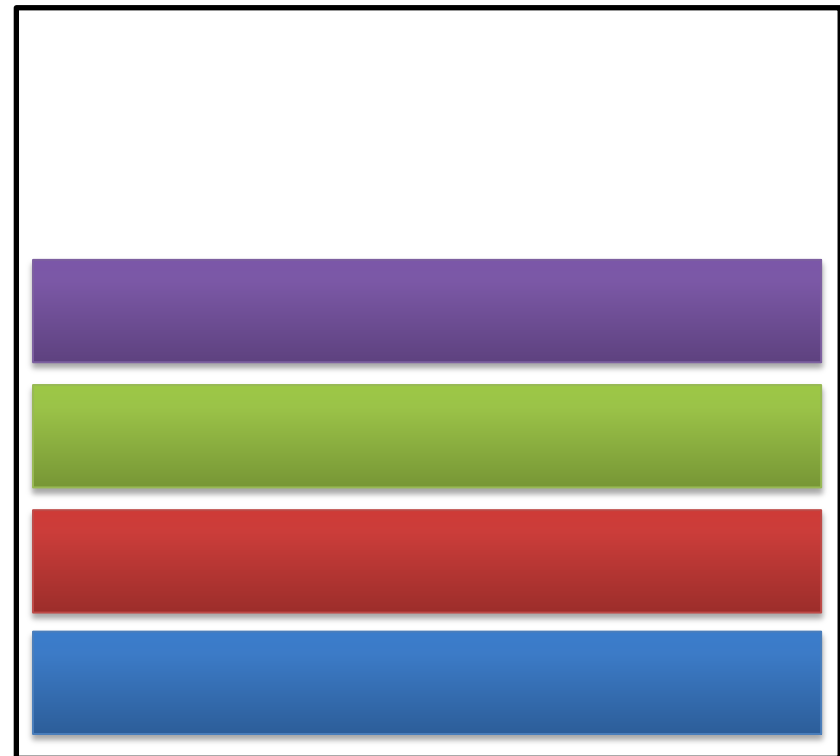


Atendimento

# Ordem de Entrada x Saída

- O que ocorre na fila de um banco?

A PRIMEIRA  
PESSOA A  
ENTRAR...  
SERÁ A  
PRIMEIRA A  
SAIR...



Atendimento

# Funções para filas

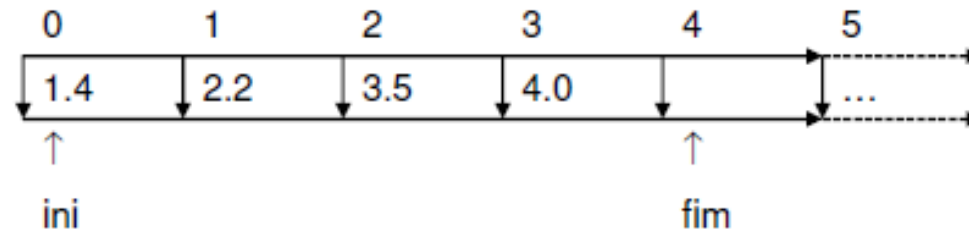
- Função **fila\_cria**
  - **Aloca** dinamicamente a estrutura da fila
  - **Inicializa** seus campos e **retorna seu ponteiro**
- Função **fila\_insere** e função **fila\_retira**
  - **Insere no final** e **retira do início**, respectivamente, um valor real da fila
- Função **fila\_vazia**
  - Informa se a fila **está ou não vazia**
- Função **fila\_libera**
  - **Destrói a fila**, liberando toda a memória usada pela estrutura



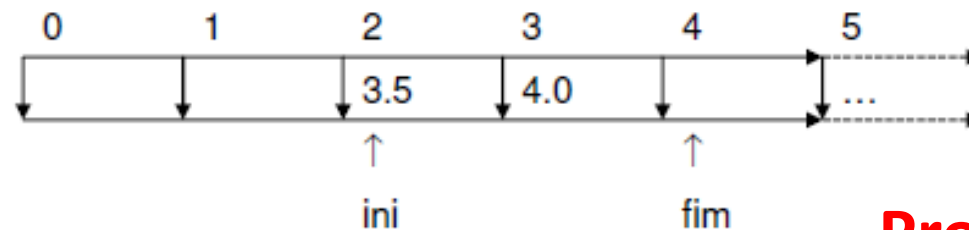
# Implementação de fila com vetor

- Processo de **inserção e remoção em extremidades opostas** da fila faz com que a fila “**ande**” no vetor

inserção dos elementos 1.4, 2.2, 3.5, 4.0



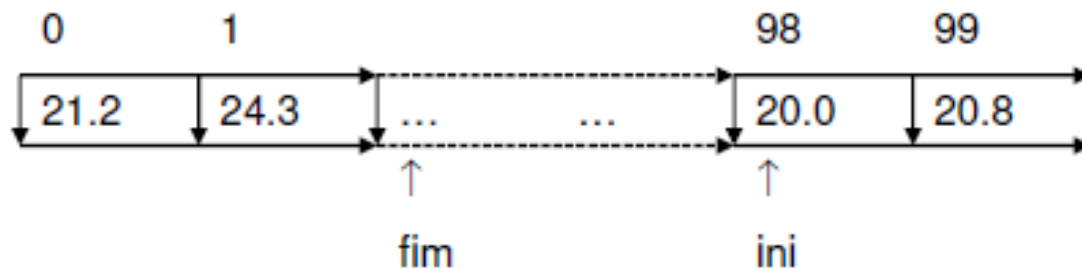
remoção de dois elementos



**Problema???**

# Implementação de fila com vetor

- Incremento das posições do vetor de forma “circular”:
  - Se o último elemento da fila ocupa a última posição do vetor, os novos elementos são inseridos a partir do início do vetor.
  - Exemplo:
    - Quatro elementos, 20.0, 20.8, 21.2 e 24.3
    - Distribuídos dois no fim do vetor e dois no início





# Implementação de fila com vetor

- Incremento das posições do vetor de forma “**circular**”:
  - usa o operador módulo “%” (ou “resto da divisão”)
- Parâmetros da fila:
  - n** = número de **elementos** na fila
  - ini** = posição do **próximo elemento** a ser **retirado** da fila
  - fim** = posição onde será **inserido o próximo elemento**
  - N** = número máximo de elementos

```
...  
fim = (ini+n)%N  
...
```



# Implementação de fila com vetor

- $FIM = (2 + 4) \% 8$
- $FIM = 6$
- $Ini = 2 / Fim = 6$

```
...  
fim = (ini+n)%N  
...
```



# Implementação de fila com vetor

- Estrutura da fila:

```
#define N 100      /* número máximo de elementos */

struct fila {
    int n;          /* número de elementos na fila */
    int ini;        /* posição do próximo elemento a ser retirado da fila */
    float vet[N];
};
```

# Interfaces para filas

```
typedef struct fila Fila;
```

```
Fila* fila_cria (void);
```

```
void fila_inserir (Fila* f, float v);
```

```
float fila_retira (Fila* f);
```

```
int fila_vazia (Fila* f);
```

```
void fila_libera (Fila* f);
```

tipo Fila:

- definido na interface
- depende da implementação do struct fila

# Implementação de fila com vetor

- Função fila\_cria
  - Aloca dinamicamente um vetor
  - Inicializa a fila como sendo vazia (número de elementos = 0)

```
Fila* fila_cria (void)
{
    Fila* f = (Fila*) malloc(sizeof(Fila));
    f->n = 0;           /* inicializa fila como vazia      */
    f->ini = 0;         /* escolhe uma posição inicial    */
    return f;
}
```

# Implementação de fila com vetor

- Função fila\_inserere
  - Insere (enqueue) um elemento no final da fila
  - Usa a próxima posição livre do vetor, se houver

```
void fila_inserere (Fila* f, float v)
{ int fim;
  if (f->n == N) { /* fila cheia: capacidade esgotada */
    printf("Capacidade da fila estourou.\n");
    exit(1);      /* aborta programa */
  }
  /* insere elemento na próxima posição livre */
  fim = (f->ini + f->n) % N;
  f->vet[fim] = v;
  f->n++;
}
```



# Implementação de fila com vetor

- Função fila\_retira
  - Retira (dequeue) o elemento do início da fila, retornando o seu valor
  - Verifica se a fila está ou não vazia

```
float fila_retira (Fila* f)
{ float v;
  if (fila_vazia(f)) {
    printf("Fila vazia.\n");
    exit(1);      /* aborta programa */
  }
  /* retira elemento do início */
  v = f->vet[f->ini];
  f->ini = (f->ini + 1) % N;
  f->n--;
  return v;
}
```

# Atividade 1

- Implemente as funções básicas de operação de filas vistas anteriormente, adicionando também uma função de impressão (do início para o final da fila).
- Simule uma sequência de inserções e remoções proposta, verificando a adequação através de chamadas à função de impressão criada.





# Atividade 2

- Combinação de filas
- Implemente uma função que receba três filas, `f_res`, `f1` e `f2`, e transfira alternadamente os elementos de `f1` e `f2` para `f_res`.
- Ao final de função, as filas `f1` e `f2` devem estar vazias.
- A fila `f_res` pode ou não estar vazia inicialmente



# Referências

- Material cedido pelo Prof. Dr. Lourenço Basso
- CELES, Waldemar; CERQUEIRA, Renato; RANGEL, José Lucas. **Introdução a Estrutura de Dados**. Editora Campus, 2004.
- Material didático do Departamento de Informática da PUC-Rio (2014).

