

INSTITUTO MAUÁ DE TECNOLOGIA



14

Filas



Filas

Fila (*queue*)

Armazena elementos que estão em **espera** para serem processados (exemplo: fila de banco);

em computação, uma fila serve para:

- armazenar eventos do sistema a serem processados;
- controlar o acesso simultâneo a recursos compartilhados (disco, impressora, rede, . . .);
- controlar o processo de recebimento de mensagens em um programa (exemplo: servidor WEB);
- simulações de situações do mundo real (exemplo: aeroporto, caixas de um supermercado);
- aplicações específicas em alguns algoritmos (busca em largura, compressão Huffman, etc.).

Fila (*queue*)

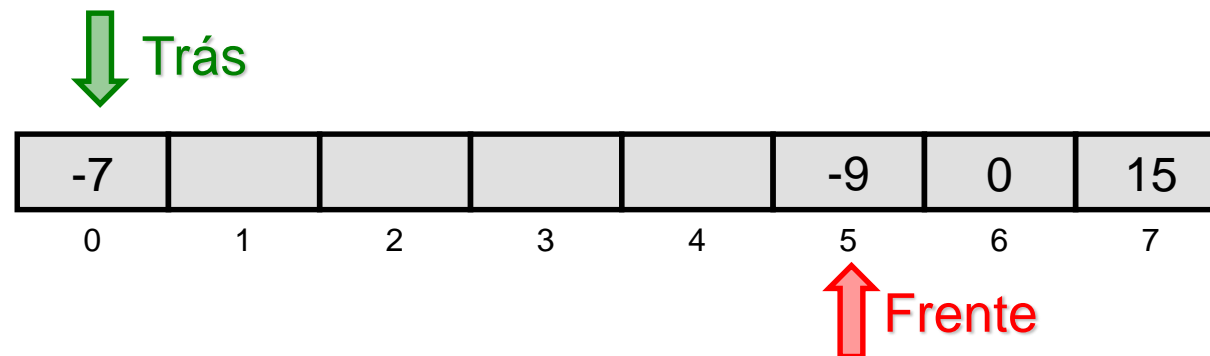
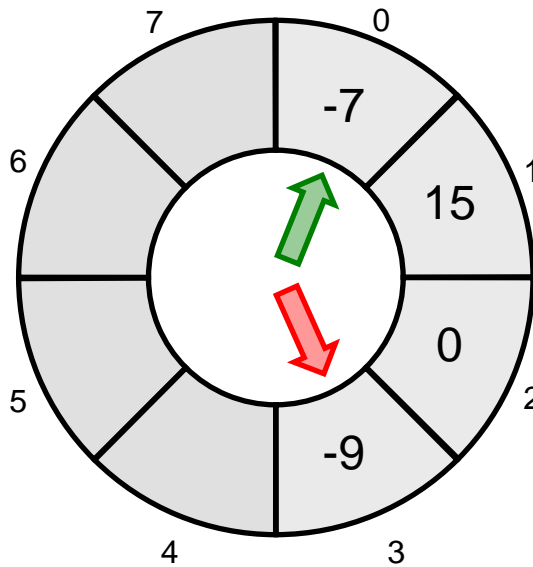
FIFO: First In First Out;

a operação de inserção coloca um elemento no **final** da fila enquanto que a operação de retirada retira o **primeiro** elemento da fila;

estudaremos uma fila implementada com um vetor e acessada de forma circular:

- itens são inseridos na posição “trás”;
- itens são retirados na posição “frente”;
- a fila está vazia quando a contagem do número de elementos é 0;
- a fila está cheia quando a contagem do número de elementos é o tamanho total da fila.

Fila Circular (*Buffer Circular*)



fila.h

```
#ifndef FILA_H_INCLUDED
#define FILA_H_INCLUDED

#define MAX 20
#define MAX_FILA 10

typedef struct {
    float valor;
    char texto[MAX];
} Dados;

typedef struct {
    Dados elemento[MAX_FILA];
    int quantidade, frente, tras;
} TFila;
```

fila.h

```
//cria uma fila nova  
void CriarFila(TFila *f);  
//insere elemento  
int InserirNaFila(TFila *f, Dados dados);  
//retira elemento  
int RetirarDaFila(TFila *f, Dados *dados);  
//indica se a fila está vazia  
int FilaVazia(TFila f);  
//indica se a fila está cheia  
int FilaCheia(TFila f);  
//retorna o tamanho da fila  
int QuantidadeNaFila(TFila f);  
  
#endif // FILA_H_INCLUDED
```

fila.c

```
#include <stdio.h>
#include <stdlib.h>
#include "fila.h"

void CriarFila(TFila* f) {
    /* define valores iniciais das propriedades da fila */
    f->quantidade = 0;
    f->frente = 0;
    f->tras = -1;
}
```



```
int InserirNaFila(TFila *f, Dados dados) {  
    int fc = FilaCheia(*f);  
    if (!fc) {  
        /* incrementa a posição "trás" no buffer circular*/  
        f->tras = (f->tras + 1) % MAX_FILA;  
        /* insere o elemento no fim da fila */  
        f->elemento[f->tras] = dados;  
        /* incrementa a quantidade */  
        f->quantidade++;  
    }  
    return !fc; /* retorna se conseguiu inserir o dado */  
}
```

```
int RetirarDaFila(TFila *f, Dados *dados) {  
    int fv = FilaVazia(*f);  
    if (!fv) {  
        /* recupera o dado do início da fila */  
        *dados = f->elemento[f->frente];  
        /* incrementa a posição "frente" no buffer circular*/  
        f->frente = (f->frente + 1) % MAX_FILA;  
        /* decrementa a quantidade */  
        f->quantidade--;  
    }  
    return !fv; /* retorna se conseguiu remover o dado */  
}
```

fila.c

```
int FilaVazia(TFila f) {  
    return f.quantidade == 0;  
}  
  
int FilaCheia(TFila f) {  
    return f.quantidade == MAX_FILA;  
}  
  
int QuantidadeNaFila(TFila f) {  
    return f.quantidade;  
}
```