

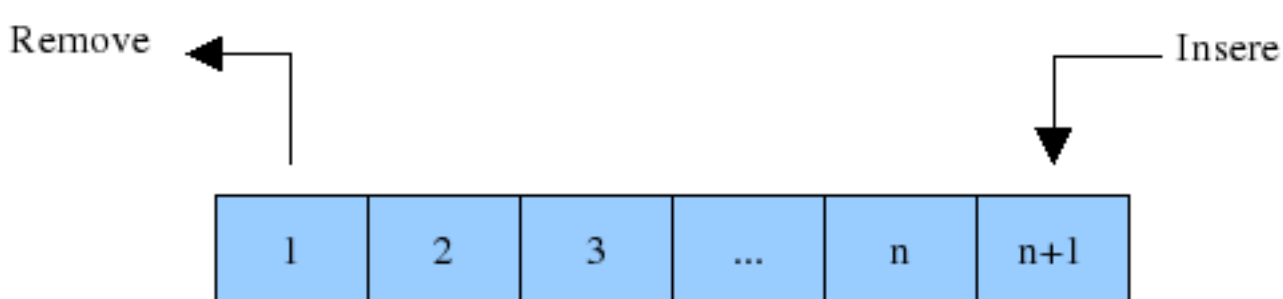
Estrutura de Dados e Algoritmos

2º Semestre de 2009

Professor Ricardo Farias

Filas - Queue

São estruturas de dados do tipo FIFO (first-in first-out), onde o primeiro elemento a ser inserido, será o primeiro a ser retirado, ou seja, adiciona-se itens no fim e remove-se do início.



São exemplos de uso de fila em um sistema:

- Controle de documentos para impressão;
- Troca de mensagem entre computadores numa rede;
- etc.

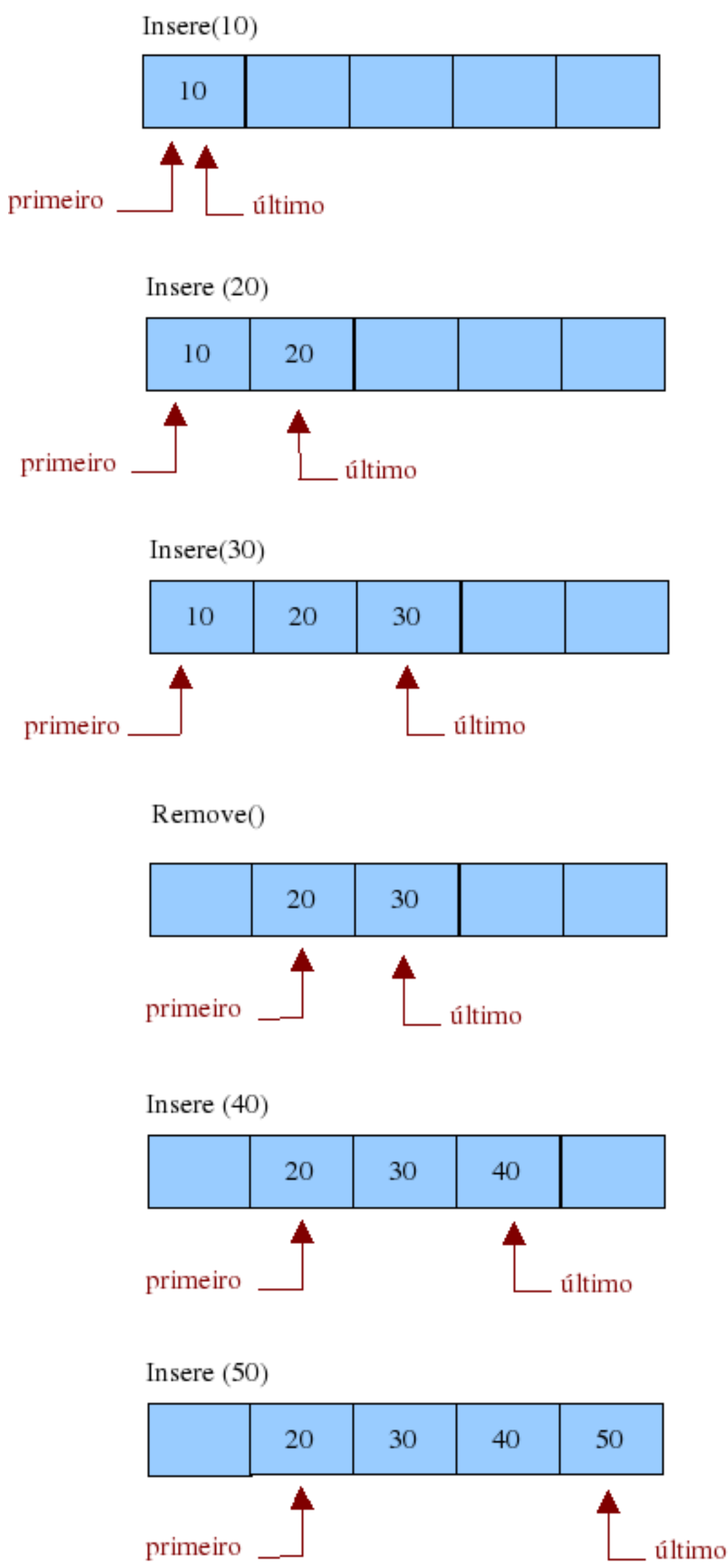
A implementação de filas pode ser realizada através de vetor (alocação do espaço de memória para os elementos é contígua) ou através de listas encadeadas (próxima aula).

Operações com Fila:

Todas as operações em uma fila podem ser imaginadas como as que ocorre numa fila de pessoas num banco, exceto que o elementos não se movem na fila, conforme o primeiro elemento é retirado. Isto seria muito custoso para o computador. O que se faz na realidade é indicar quem é o primeiro.

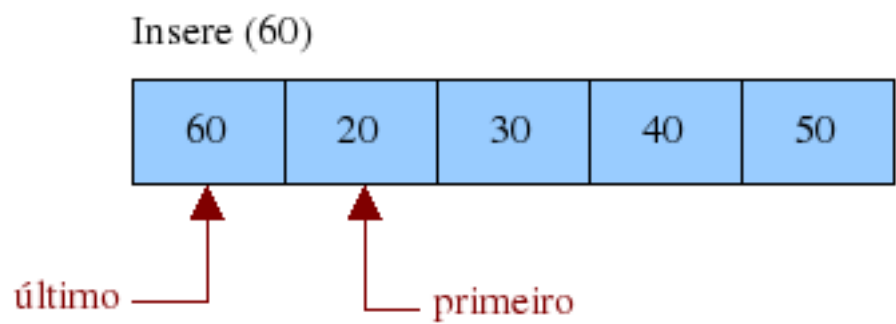
- criação da fila (informar a capacidade no caso de implementação sequencial - vetor);
- enfileirar (enqueue) - o elemento é o parâmetro nesta operação;
- desenfileirar (dequeue);
- mostrar a fila (todos os elementos);
- verificar se a fila está vazia (isEmpty);
- verificar se a fila está cheia (isFull - implementação sequencial - vetor).

Supondo uma fila com capacidade para 5 elementos (5 nós).

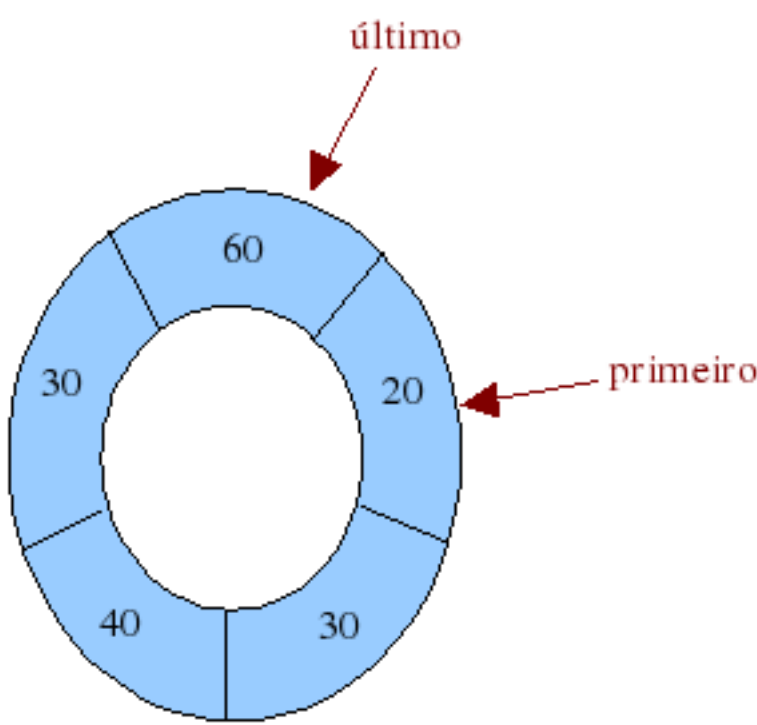


Na realidade a remoção de um elemento da fila é realizada apenas alterando-se a informação da posição do último.

Para evitar problemas de não ser capaz de inserir mais elementos na fila, mesmo quando ela não está cheia, as referências primeiro e último circundam até o início do vetor, resultando numa fila circular.



Desta forma a fila simula uma representação circular:



Veja o algoritmo a seguir para uma fila de números reais:

```
#include

struct Fila {
    int capacidade;
    float *dados;
    int primeiro;
    int ultimo;
    int nItens;
};

void criarFila( struct Fila *f, int c ) {
    f->capacidade = c;
    f->dados = (float*) malloc (f->capacidade * sizeof(float));
    f->primeiro = 0;
    f->ultimo = -1;
    f->nItens = 0;
}

void inserir(struct Fila *f, int v) {
    if(f->ultimo == f->capacidade-1)
        f->ultimo = -1;

    f->ultimo++;
    f->dados[f->ultimo] = v; // incrementa ultimo e insere
    f->nItens++; // mais um item inserido
}

int remover( struct Fila *f ) { // pega o item do começo da fila
    int temp = f->dados[f->primeiro++]; // pega o valor e incrementa o primeiro
    if(f->primeiro == f->capacidade)
        f->primeiro = 0;

    f->nItens--; // um item retirado
    return temp;
}

int estaVazia( struct Fila *f ) { // retorna verdadeiro se a fila está vazia
    return (f->nItens==0);
}

int estaCheia( struct Fila *f ) { // retorna verdadeiro se a fila está cheia
    return (f->nItens == f->capacidade);
}

void mostrarFila(struct Fila *f){
    int cont, i;
    for ( cont=0, i= f->primeiro; cont < f->nItens; cont++){
        printf("%.2f\t",f->dados[i++]);
        if (i == f->capacidade)
            i=0;
    }
    printf("\n\n");
}

void main () {
    int opcao, capa;
    float valor;
    struct Fila umaFila;

    // cria a fila
    printf("\nCapacidade da fila? ");
    scanf("%d",&capa);
    criarFila(&umaFila, capa);

    // apresenta menu
    while( 1 ){
        printf("\n1 - Inserir elemento\n2 - Remover elemento\n3 - Mostrar Fila\n0 - Sair\n\nOpcao? ");
        scanf("%d", &opcao);

        switch(opcao){
            case 0: exit(0);
            case 1: // insere elemento
                if (estaCheia(&umaFila)){
                    printf("\nFila Cheia!!!\n\n");
                }
                else {
                    printf("\nValor do elemento a ser inserido? ");
                    scanf("%f", &valor);
                    inserir(&umaFila,valor);
                }
                break;
            case 2: // remove elemento
                if (estaVazia(&umaFila)){
                    printf("\nFila vazia!!!\n\n");
                }
                else {
                    valor = remover(&umaFila);
                    printf("\n%lf removido com sucesso\n\n", valor) ;
                }
                break;
            case 3: // mostrar fila
                if (estaVazia(&umaFila)){
                    printf("\nFila vazia!!!\n\n");
                }
                else {
                    printf("\nConteudo da fila => ");
                    mostrarFila(&umaFila);
                }
                break;
            default:
                printf("\nOpcao Invalida\n\n");
        }
    }
}
```