

## Estruturas de Dados (Cap. 12)

### Sumário:

- Estruturas e Alocação Dinâmica de Memória;
- Filas;
- Pilhas;
- Listas Encadeadas; e
- Árvore Binária.

## Alocação Dinâmica de Memória

- Criação da variável durante a execução do programa;
  - Tamanho Variável durante o programa;
  - Uso do operador malloc (ex.: *Memory Allocation*);
- Oposto a Alocação em tempo de compilação
  - Tamanho fixo durante todo programa;
  - Ex.: `int x;`  
`char nome[50];`

## Alocação Dinâmica de Memória

```
//Alocação da Memória
void malloc (número de bytes);

//retorna tamanho de um tipo
int sizeof (tipo);

//Liberação da Memória
void free (void *p);
```

## Estruturas de Dados (Cap. 12)

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main()
{
    char*   nome = NULL;

    nome = malloc(sizeof(char)*50); //1
    strcpy(nome,"carlos fernando crispim"); //2
    printf("%s\n",nome);
    free(nome); //3

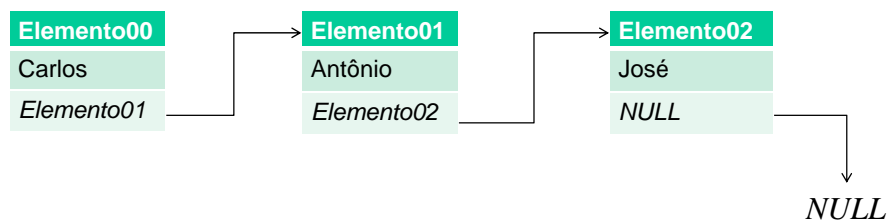
    system("PAUSE");

    return 0;
}
```

## Estruturas de Dados (Cap. 12)

Lista de estruturas encadeadas:

```
struct TipoEstrutura{  
    char* nome;  
    TipoEstrutura* proxElemento;  
};
```



## Tipos de Estrutura de Dados

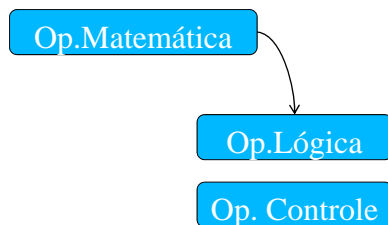
- Pilhas
- Filas
- Listas Encadeadas
- Árvore Binária

## Pilha

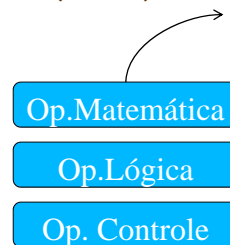
- Lista linear de informações.
- Acesso: último a entrar é o primeiro a sair
  - Last-in first-Out(LIFO)
- Duas operações:
  - push → empilha
  - pop → desempilha
- Cuidado: overflow, underflow (estouro de pilha)
- Exemplo de uso: Operações Recursivas, Torres de Hanoi.

## Pilha.Operações

push → empilha  
(Inserir)



pop → desempilha  
(Excluir)

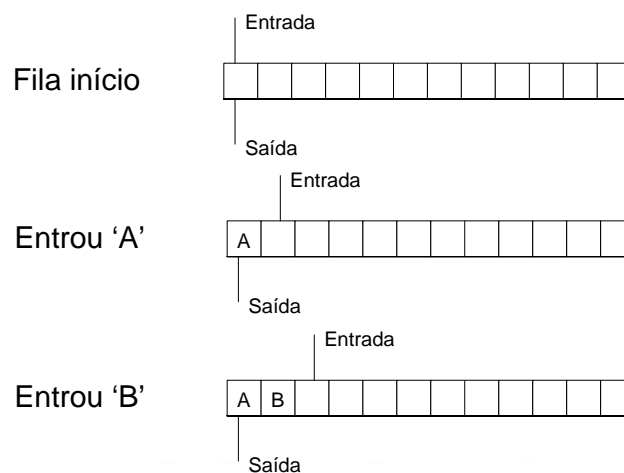


Próximo execução:  
**Op. Matemática**

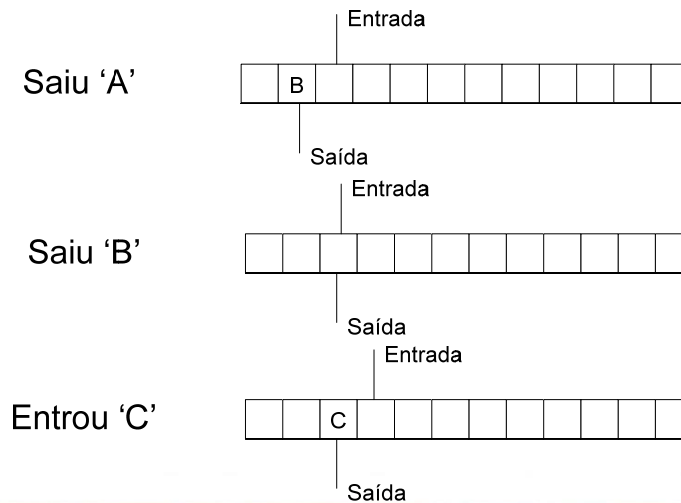
## Fila

- Lista linear de informações;
- Acesso: Primeiro a entrar, Primeiro a sair
  - First-in first-out(FIFO);
- Exemplos: *spool* da impressora (vários usuários, uma impressora);
- Duas operações: armazenamento e recuperação;
- Recuperação: destrói /perde a informação armazenada.

## Fila.Inserção&Exclusão



## Fila. Inserção & Exclusão



## Listas Encadeadas

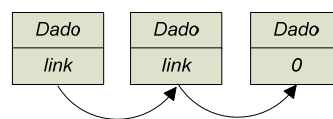
- Filas e pilhas estão sujeitas a regras rigorosas.
- Acessar um item em uma pilha requer sua remoção.
- Lista encadeada: permite acesso aleatório (sem remover os demais).
- Cada informação carrega um ou mais elos de ligação.
- Possui uma estrutura complexa de dados.
- Listas encadeadas: → singularmente encadeado  
→ duplamente encadeada

## Listas Singularmente Encadeadas

Requer que cada item de informação contenha um elo com o próximo elemento da lista

### - Estrutura:

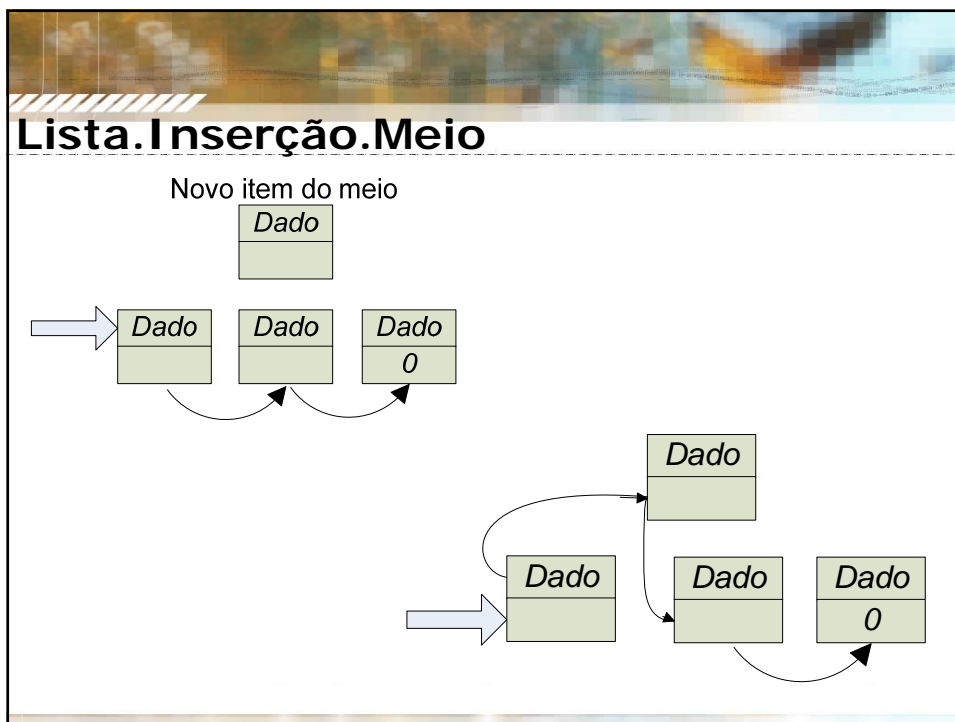
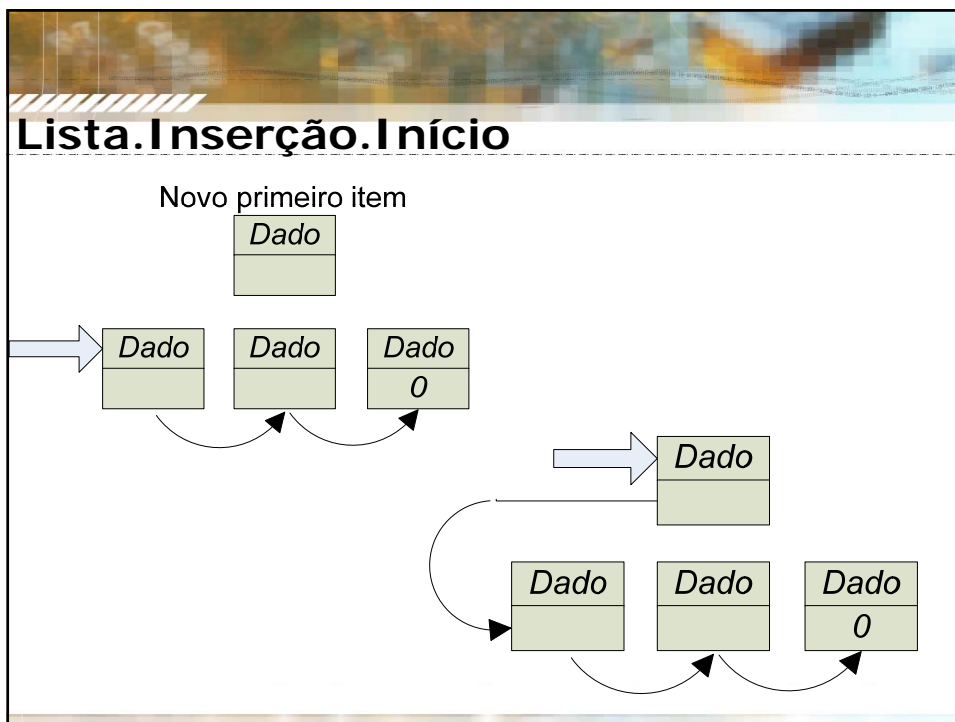
```
struct endereço endereço {  
    char nome [40];  
    char rua [40];  
    char cidade [20];  
    char estado [3];  
    char cep [11];  
    struct endereço * prox;  
} info;
```



## Lista.Inserção

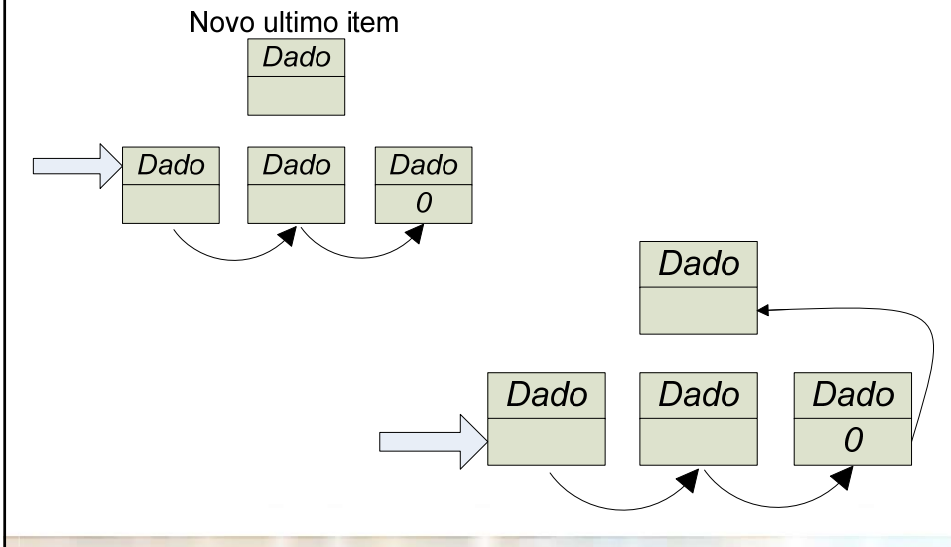
### 1) Passos:

- aloca memória para a nova informação (nova estrutura);
- Carrega informação na estrutura;
- Decide o local/posição de inserção (início, meio e fim)

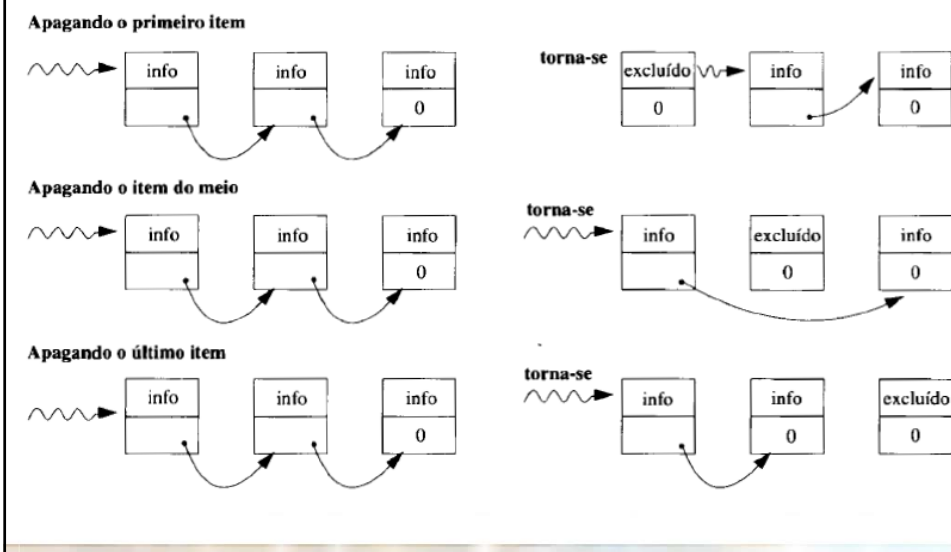




## Lista.Inserção.Fim

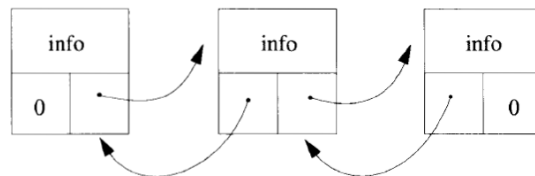


## Lista.Exclusão

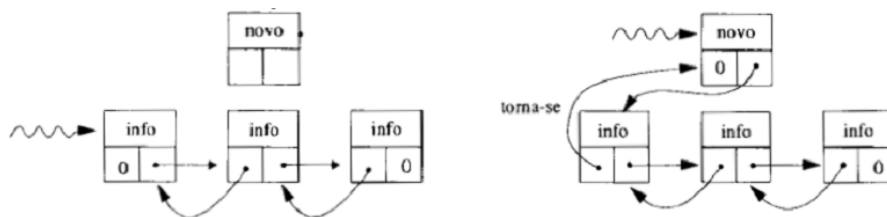


## Lista Duplamente Encadeada (LDE)

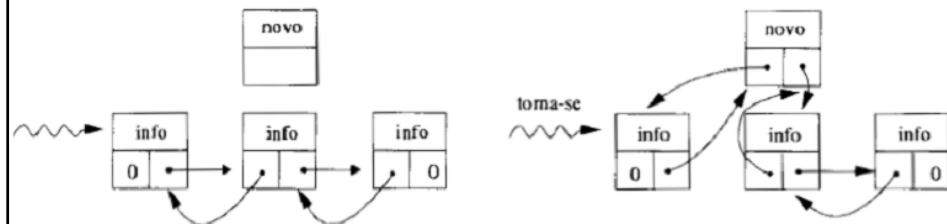
- dois elos de ligação:
  - para o elemento posterior
  - para o elemento anterior
- vantagem: a lista pode ser percorrida em ambas as direções
- Inserção e exclusão similares a lista singularmente encadeada, porém os dois elos devem ser alteradas



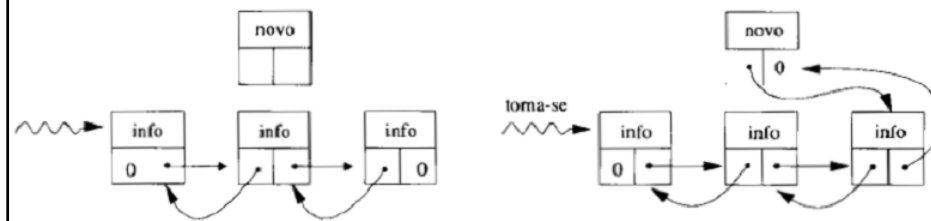
## Lista LDE. Inserção. Início



## Lista LDE.Inserção.Meio



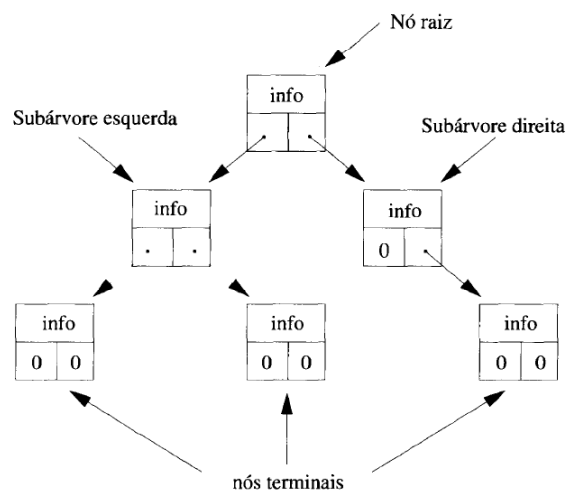
## Lista LDE.Inserção.Fim



# Árvores Binárias

- Estrutura de dados não-linear bidimensional.
- Existem vários tipos de organização.
- Quando ordenadas permitem (busca, inserção e exclusão) acesso muito rápido.
- Forma especial de lista encadeada.
- Cada item consiste em uma informação, um elo esquerdo e um direito.
- Operação de recuperação não destrutiva.
- Altura: número de camadas que as raízes atingem.
- Nó: cada item
- Nó terminal: nó que não tem subárvores (subitens).

## Exemplo de Árvore



## Exercícios

1. Desenvolva um programa utilizando-se de fila circular que devolve o valor de um sinal senoidal de entrada defasado (avanço ou atraso de fase) de um valor de ângulo definido na estrada da função. O sinal deverá ser gerado com um número de pontos estabelecidos pelo usuário



1. Desenvolva um programa (função) que calcula o valor RMS e valor médio de uma forma de onda senoidal distorcida. O sinal deverá ser gerado com um número de pontos estabelecidos pelo usuário.