

LISTAS DINÂMICAS ENCADEADAS

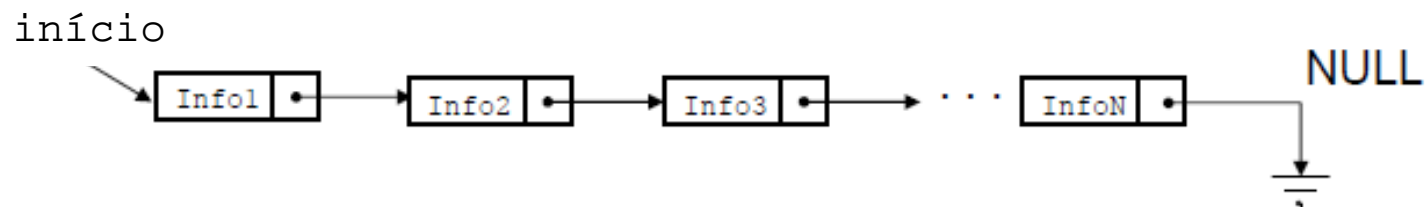
Profa.: Mirlem R. R. Pereira



INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
AMAZONAS

www.ifam.edu.br

Listas Dinâmicas Encadeadas

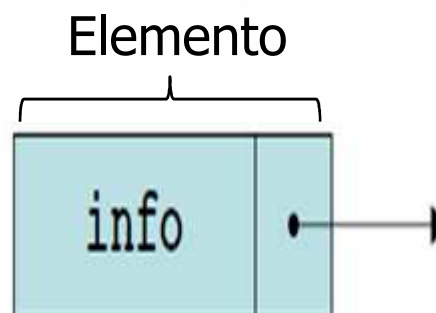


- Características:
 - seqüência encadeada de elementos, chamados de *nós da lista*
 - nó da lista é representado por dois campos:
 - a informação armazenada e
 - o ponteiro para o próximo elemento da lista
 - a lista é representada por um ponteiro para o primeiro nó
 - o ponteiro do último elemento é NULL

Listas Dinâmicas Encadeadas

Estrutura com ponteiro para ela mesma

```
struct elemento {  
    int info;  
    struct elemento* prox;  
};  
typedef struct elemento Elemento;
```



Listas Dinâmicas Encadeadas

- Os nós da lista estão dispostos de maneira aleatória na memória (posições não obrigatoriamente contíguas) e são interligados por ponteiros: por isso é também chamada de lista não-sequencial.
- Crescem (ou decrescem) à medida que elementos são inseridos (ou removidos).
- Alocação/liberação desses endereços gerenciada pelo S.O., por meio de comandos da linguagem de programação. Em Linguagem C faz a gerência de memória através malloc e free.
- Existem outras implementações: simplesmente ou duplamente encadeadas e listas encadeadas circulares; com ou sem nós sentinela

Listas Dinâmicas Encadeadas

VANTAGENS

- A inserção e remoção de elementos podem ser feitas sem deslocar os itens seguintes da lista
- Não há necessidade de previsão do número de elementos da lista; o espaço necessário é alocado em tempo de execução
- Facilita o gerenciamento de várias listas (fusão, divisão,...)

DESVANTAGENS

- Acesso indireto aos elementos
- Tempo variável para acessar os elementos (depende da posição do elemento)
- Gasto de memória maior pela necessidade de um novo campo para o ponteiro

Listas Dinâmicas Encadeadas

VANTAGENS

- A inserção e remoção de elementos podem ser feitas sem deslocar os itens seguintes da lista
- Não há necessidade de previsão do número de elementos da lista; o espaço necessário é alocado em tempo de execução
- Facilita o gerenciamento de várias listas (fusão, divisão,...)

DESVANTAGENS

- Acesso indireto aos elementos
- Tempo variável para acessar os elementos (depende da posição do elemento)
- Gasto de memória pela necessidade de um novo campo para o ponteiro

Implementar

LISTA DE INTEIROS

1. Criação da Lista
2. Inserção do primeiro elemento
3. Inserção no início da lista
4. Inserção entre outros elementos
5. Inserção no fim da lista
6. Remoção no início da lista
7. Remoção entre outros elementos
8. Remoção no fim da lista
9. Achar um elemento na lista
10. Imprimir a Lista
11. Tamanho da lista