



Estruturas de Dados

Lista Encadeada



- Lista: É um conjunto de **zero** ou **mais itens** x_1, x_2, \dots, x_n na qual x_i é de um determinado **tipo** e n representa o tamanho da lista.
- Sua principal **propriedade** estrutural diz respeito às posições relativas dos itens.



- TADs da Lista Estática

Lista Encadeada



- Quais os problemas da lista estática?
 - Quantidade fixa de elementos
 - Memória alocada sem uso
 - Impossibilidade de alocar mais memória
- Como solucionar tais problemas?
 - Utilizar Estruturas de Dados que cresçam e diminuam na medida da necessidade, **Estruturas Dinâmicas**
 - **Malloc**, Alocação dinâmica de memória para armazenar os elementos

Lista Encadeada



- Características da **lista encadeada**
 - Cresce e diminui dinamicamente
 - Tamanho máximo não precisa ser definido
 - Provêem flexibilidade, permitindo que os itens sejam rearrumados eficientemente
 - Também chamada de **lista ligada**



- Definição?

Uma lista encadeada consiste de uma sequência linear de **nós** dinamicamente alocados, que são encadeados (ou conectados) através de ponteiros (ou apontadores)

- O que seria o **nó** da lista?



- Cada elemento da lista possui pelo menos **dois campos**: um para armazenar a **informação** e outro para o **endereço** do próximo (ponteiro)

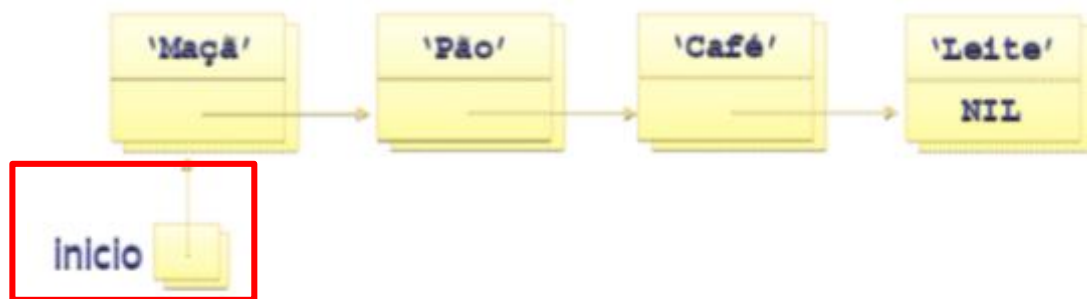


- Um **nó** possui uma seta apontando para fora. Essa seta representa um **ponteiro** que aponta para outro **nó**, formando uma **lista encadeada**





- Antes de começarmos, precisamos definir como a lista será representada
- Uma forma bastante comum é manter uma variável ponteiro para o **primeiro elemento** da lista encadeada





- Quando a lista estiver vazia, a variável ponteiro deve ter seu valor apontado para onde?
- Portanto, essa deve ser a iniciação da lista e também a forma de se verificar se ela se encontra vazia

NULL

Início





- Outro detalhe importante é quanto a posições!
 - Na implementação com vetores, uma posição é o valor inteiro entre **zero** e a variável **fim**
 - Na **lista encadeada** como é determinado a **posição**?

Com **listas encadeadas**, uma posição passa ser um **ponteiro** que aponta um determinado **nó** da lista



Criar Lista

- Pré-condição: Existir espaço na memória
- Pós-condição: Inicia a estrutura de dados

Limpar Lista

- Pré-condição: Nenhuma
- Pós-condição: Remove a estrutura de dados da memória

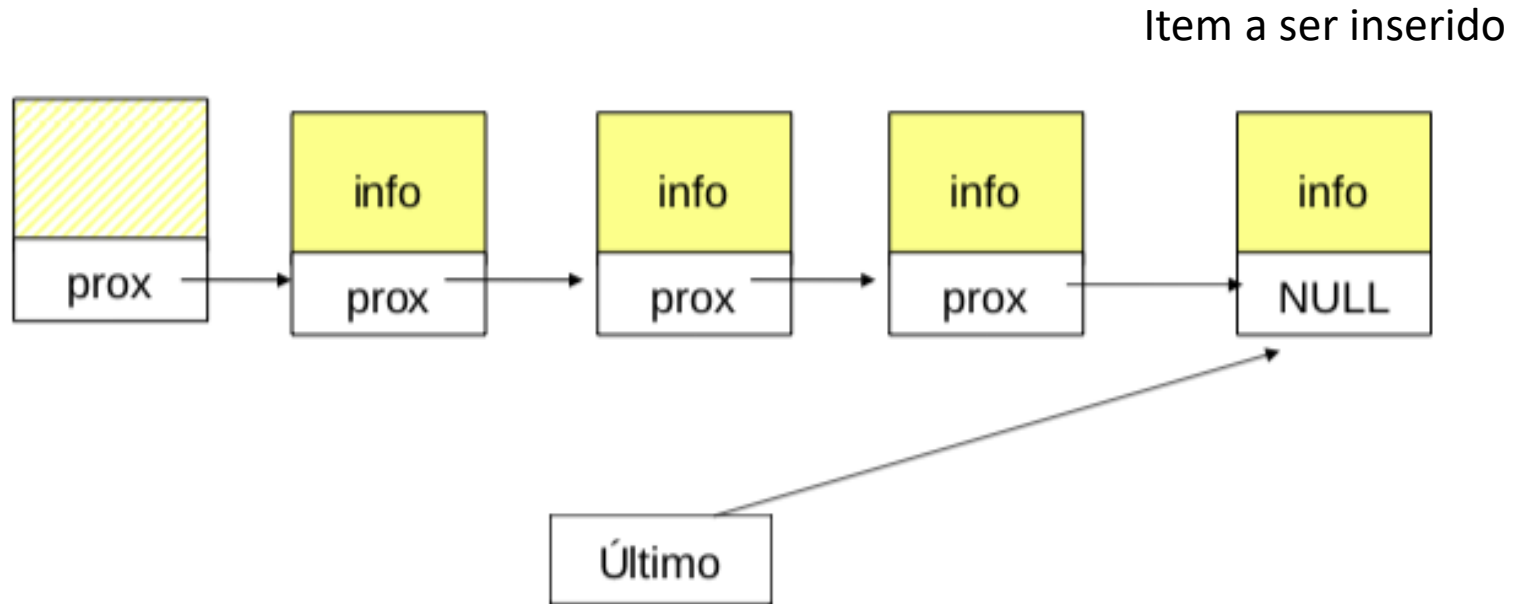


Inserir item – Última posição

- Pré-condição: Existir memória disponível
- Pós-condição: Insere um item na última posição, retorna **true** se a operação foi executada com sucesso, **false** caso contrário



- Inserir item – Última posição



- Como é inserir na **primeira** posição e **dado** uma chave?

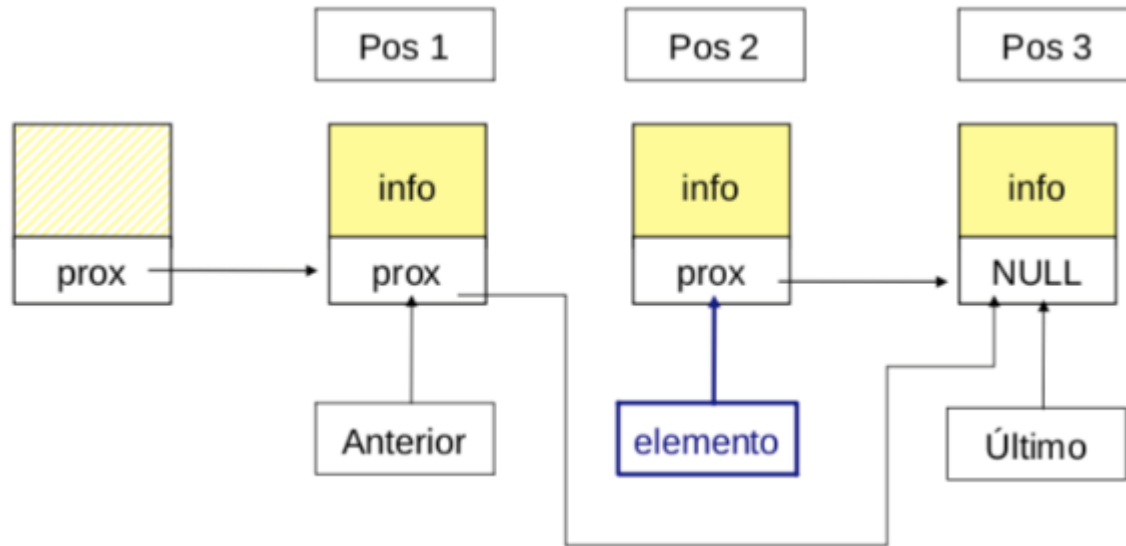


Remover item – Dado uma chave

- Pré-condição: Nenhuma
- Pós-condição: remove um determinado item da lista dado uma chave, retorna **true** se a operação foi executada com sucesso, **false** caso contrário



- Remover item – Dado uma chave



- Como é remover na **primeira** e **última** posição?



Recuperar item – Dado uma chave

- Pré-condição: Nenhuma
- Pós-condição: Recupera o item dado uma chave, retorna **true** se a operação foi executada com sucesso, **false** caso contrário

Contar número de itens

- Pré-condição: Nenhuma
- Pós-condição: Retorna o número de itens na lista



Verificar se a lista está vazia

- Pré-condição: Nenhuma
- Pós-condição: Retorna **true** se a lista estiver vazia e **falso** caso-contrário

Verificar se a lista está cheia

- Pré-condição:
- Pós-condição:

?



Imprimir lista

- Pré-condição: Nenhuma
- Pós-condição: Imprime na tela os itens da lista



Estruturas de Dados

Lista Encadeada