



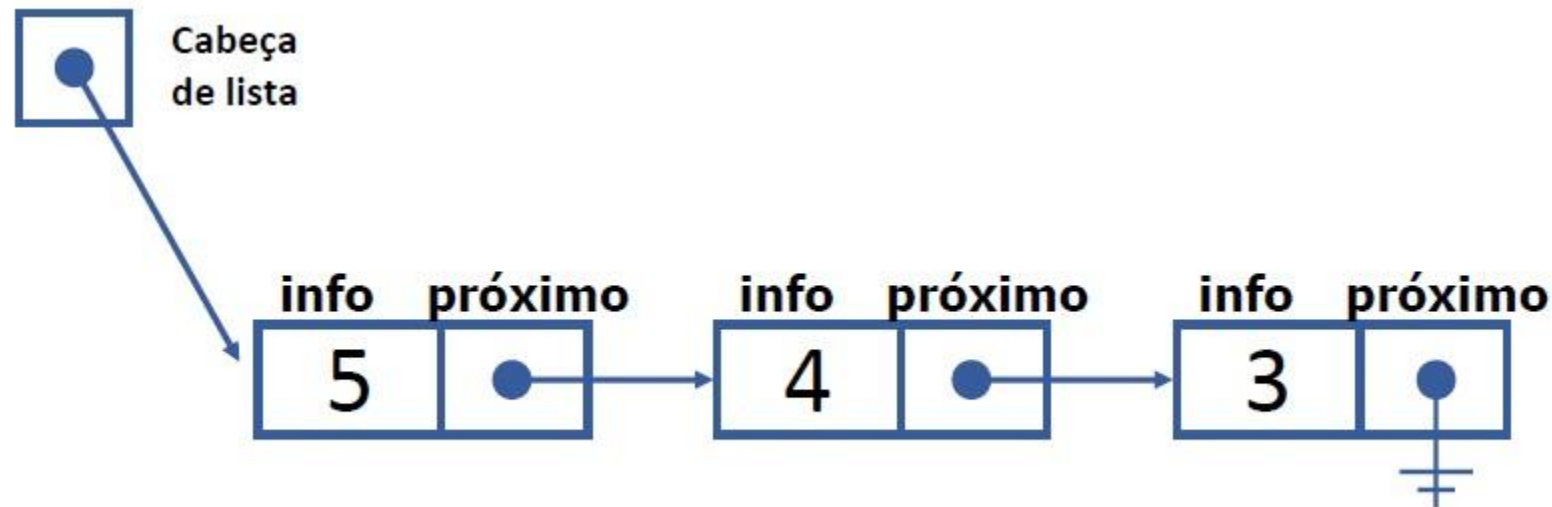
ESTRUTURA DE DADOS

Aula 7 – Listas Encadeadas

Prof. Rodrigo Maciel

LISTAS ENCADEADA SIMPLES - NÓS

- Cada item de dados é incorporado em um nó;
- Cada nó possui uma referência para o próximo nó da lista;
- Um campo da própria lista contém uma referência para o primeiro nó.



LISTAS ENCADEADAS - Posição X Relacionamento

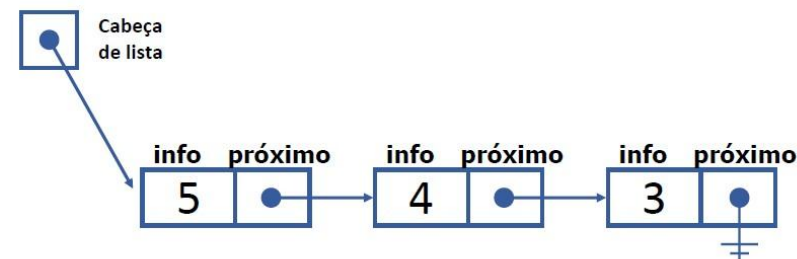
- **Vetor/Array (posição)**

- Cada item ocupa uma certa posição;
- Cada posição pode ser acessada usando um número de índice.



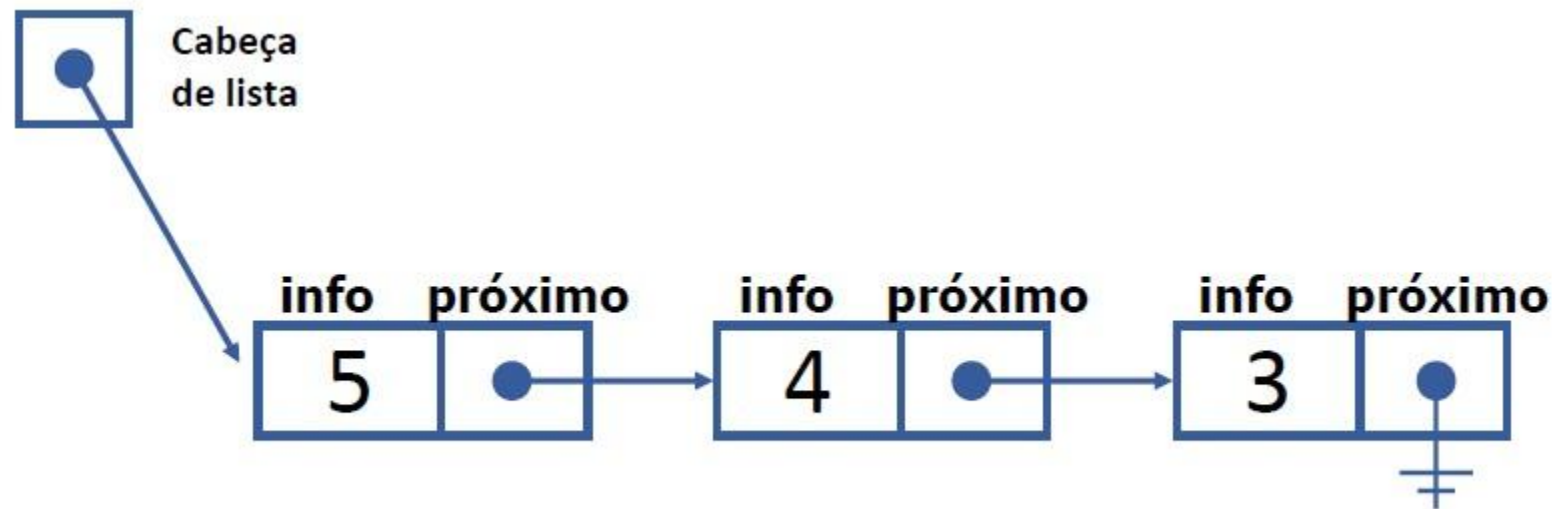
- **Lista (relacionamento)**

- A única maneira de encontrar um elemento é seguir a sequência de elementos;
- Um item de dados não pode ser acessado diretamente, ou seja, o relacionamento entre eles deve ser utilizado;
- Inicia com o primeiro item, vai para o segundo, então o terceiro, até encontrar o item pesquisado.



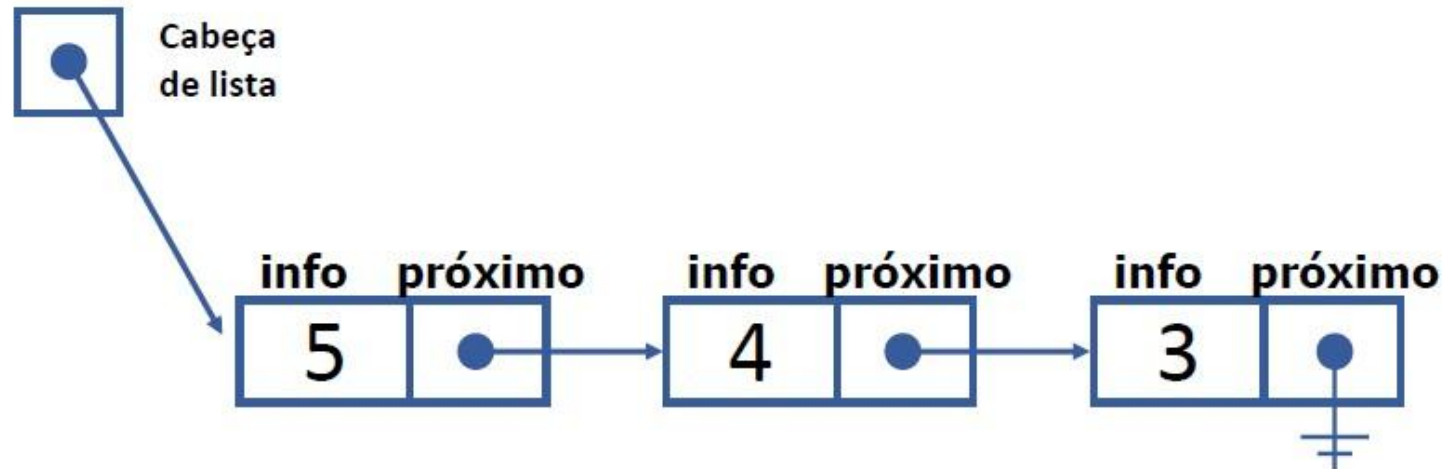
LISTAS ENCADEADAS

- Cada elemento da lista é armazenado em um objeto;
- Cada elemento da lista referencia o próximo e só é alocado dinamicamente quando é necessário;
- Para referenciar o primeiro elemento é utilizado um elemento chamado cabeça da lista.



LISTAS ENCADEADAS - OPERAÇÕES

- Inserir no início
- Excluir do início
- Mostrar lista
- Pesquisar
- Excluir da posição



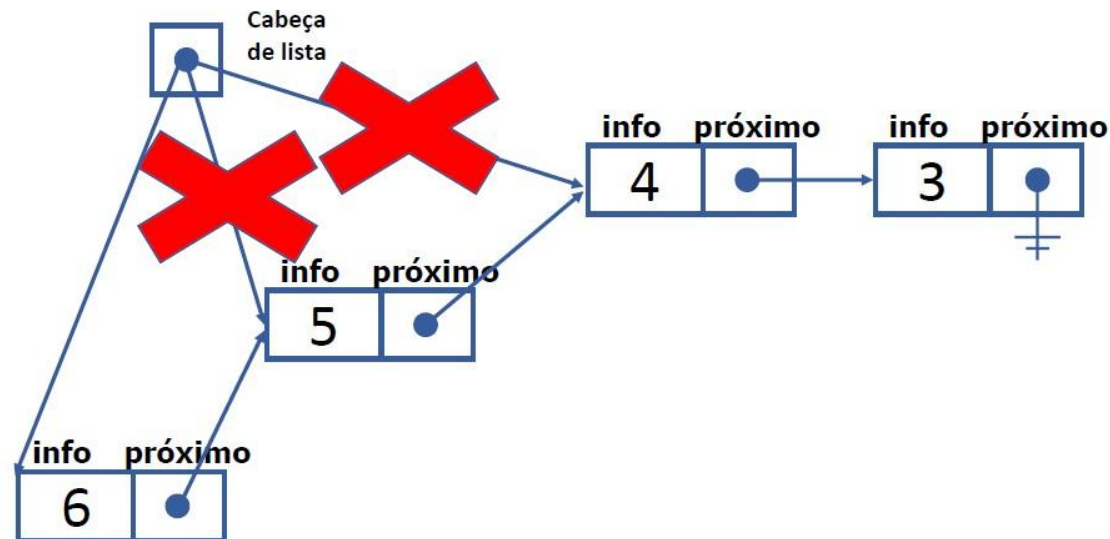
LISTAS ENCADEADAS - INSERE NO INÍCIO

- Insere um novo nó no início da lista
- É o local mais fácil para inserir um nó
- O próximo deste novo elemento deve ser o
- primeiro da lista
- A cabeça da lista aponta para o novo elemento

LISTAS ENCADEADAS - INSERE NO INÍCIO

Procedimentos:

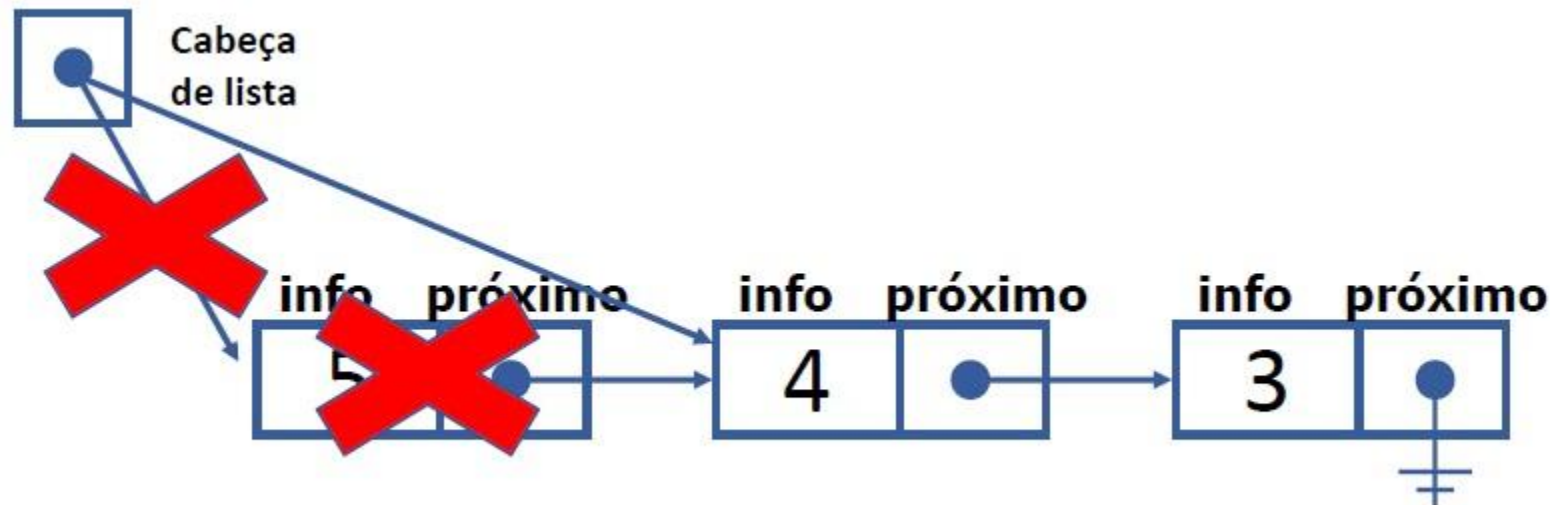
1. Criar um novo elemento (nó);
2. O próximo do novo nó criado vai apontar para o primeiro nó pré-existente da lista;
3. Destrói a conexão da cabeça da lista que aponta para o primeiro nó;
4. Faz o um novo apontamento da cabeça da lista para o nó recém criado.



LISTAS ENCADEADAS - EXCLUIR DO INÍCIO

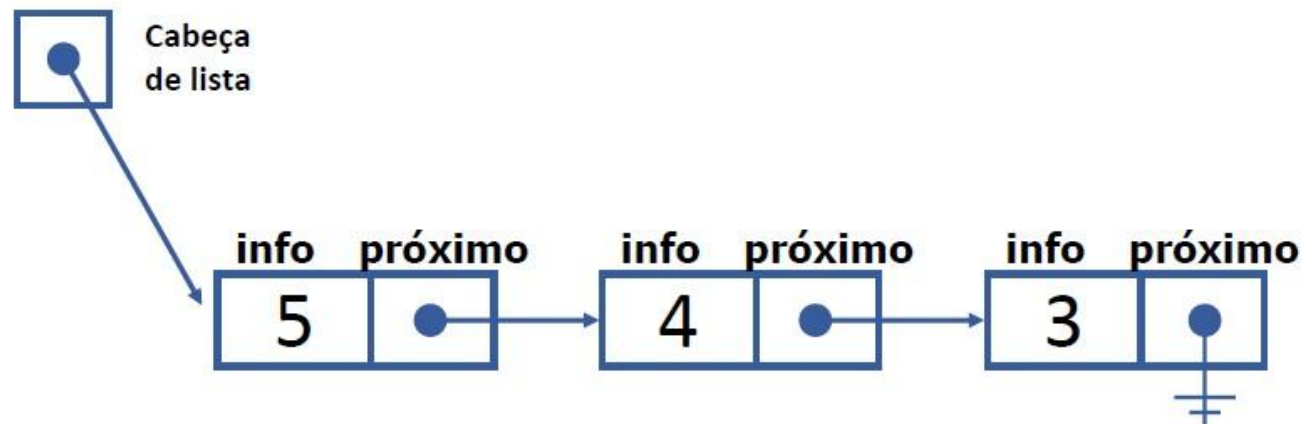
Procedimentos:

1. É o inverso do insere no início;
2. Desconecta o primeiro nó roteando de novo o primeiro para apontar para o segundo nó;
3. O segundo nó é encontrado por meio do campo próximo no primeiro nó.



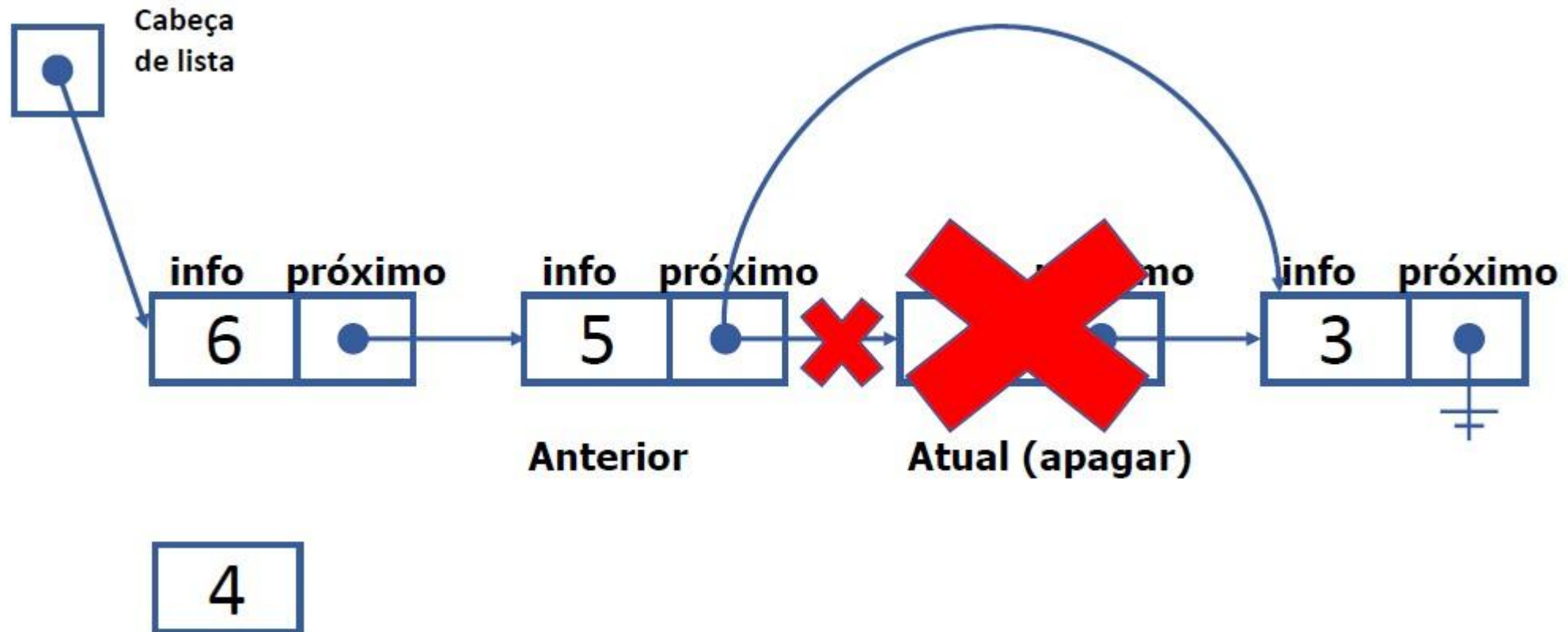
LISTAS ENCADEADAS - MOSTRAR LISTA E PESQUISAR

- Para exibir a lista, deverá ser iniciado no primeiro elemento, seguindo a sequência de referências de nó em nó;
- O final da lista é indicado pelo campo próximo no último nó apontando para null/none ao invés de outro nó;
- Os nós são percorridos e é verificado se o valor do elemento é aquele que está sendo procurado;
- Se atingir o final da lista sem encontrar o nó desejado, finaliza sem encontrar o elemento.



LISTAS ENCADEADAS - EXCLUIR DE POSIÇÃO QUALQUER

- O nó a ser eliminado deve ser localizado (pesquisa);
- Quando elimina o nó atual, terá que conectar o nó anterior ao nó seguinte.



LISTAS ENCADEADA SIMPLES - IMPLEMENTAÇÃO

Classes:

Nó
+ valor + proximo = Nulo
+ mostraNo()

Lista Encadeada
+ primeiro = Nulo
+ inserelInicio(valor) + mostrar() + pesquisar(valor) + excluirInicio() + excluirPosição(valor)

LISTAS ENCADEADA SIMPLES - IMPLEMENTAÇÃO

Algoritmo 1: Mostra Nó

mostra_nó():

inicio

 imprime(valor)

fim

LISTAS ENCADEADA SIMPLES - IMPLEMENTAÇÃO

Algoritmo 2: Inserir no início

inserir_inicio(valor):

inicio

 novo <- Nó(valor)

 novo.próximo <- primeiro

 primeiro <- novo

fim

LISTAS ENCADEADA SIMPLES - IMPLEMENTAÇÃO

Algoritmo 3: Mostrar Lista

mostar():

inicio

se primeiro = Nulo **então**

 Erro de lista vazia

return Nulo

 atual <- primeiro

enquanto atual \neq Nulo **então**

 atual.mostra_nó()

 atual <- atual.próximo

fim

LISTAS ENCADEADA SIMPLES - IMPLEMENTAÇÃO

Algoritmo 4: Excluir no Início da Lista

excluir_inicio():

inicio

se primeiro = Nulo **então**

 Erro de lista vazia

return Nulo

 temp <- primeiro

 primeiro <- primeiro.próximo

return temp

fim

LISTAS ENCADEADA SIMPLES - IMPLEMENTAÇÃO

Algoritmo 5: Pesquisar na Lista

pesquisar(valor):

inicio

se primeiro = Nulo **então**

 Erro de lista vazia

return Nulo

 atual <- primeiro

enquanto atual.valor \neq valor **então**

se atual.próximo = Nulo **então**

return Nulo

senão

 atual <- atual.proximo

return atual

fim

LISTAS ENCADEADA SIMPLES - IMPLEMENTAÇÃO

Algoritmo 6: Excluir em qualquer posição da Lista

excluir_qualquer(valor):

inicio

se primeiro = Nulo **então**

 Erro de lista vazia

return Nulo

 atual <- primeiro

 anterior <- primeiro

enquanto atual.valor ≠ valor **então**

se atual.próximo = Nulo **então**

return Nulo

senão

 anterior <- atual

 atual <- atual.próximo

se atual = primeiro **então**

 primeiro <- primeiro.próximo

senão

 anterior.próximo <- atual.próximo

return atual

fim