

Listas Dinâmicas

LDE: Lista Dinâmica Encadeada
(simples)

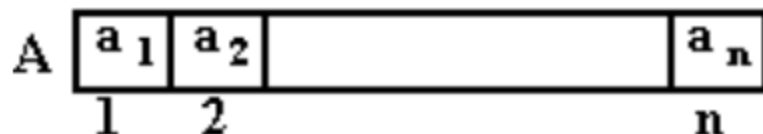
(Fonte: Material adaptado dos Slides do prof. Monael.)

Listas

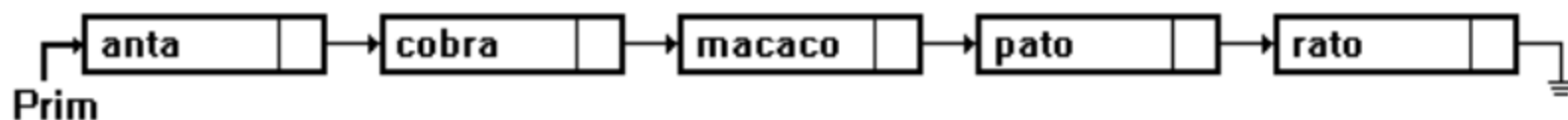
- Uma lista é uma estrutura de dados em que os elementos estão organizados em uma ordem linear. Uma lista pode ser:
 - **Simplesmente Ligada**: A partir de um elemento da lista não se alcança o elemento anterior.
 - **Duplamente Ligada**: A partir de um elemento da lista se alcança o elemento anterior.
 - **Ordenada**: A ordem linear da lista corresponde à ordem das chaves.
 - **Não ordenada**: Os elementos aparecem em qualquer ordem.
 - **Circular**: A partir do primeiro elemento da lista se alcança o último. E a partir do último elemento da lista se alcança o primeiro.
 - **Não circular**: A partir do primeiro não se alcança o último. E do último elemento não se alcança o primeiro.

Listas

- Quanto a implementação, as listas podem ser:
 - **Estáticas**: Os elementos são armazenados em um vetor.



- **Dinâmicas**: Os elementos são alocados dinamicamente conforme necessidade. Cada elemento armazena os dados e um ponteiro para o próximo elemento da lista.



Listas

- Estrutura de Dados do tipo Lista:
 - Lista Estática Seqüencial (LES)
 - Lista Estática Encadeada (LEE)
 - Lista Estática Duplamente Encadeada (LEDE)
 - Lista Dinâmica Encadeada (LDE)
 - Lista Dinâmica Duplamente Encadeada (LDDE)

Lista Dinâmica Encadeada

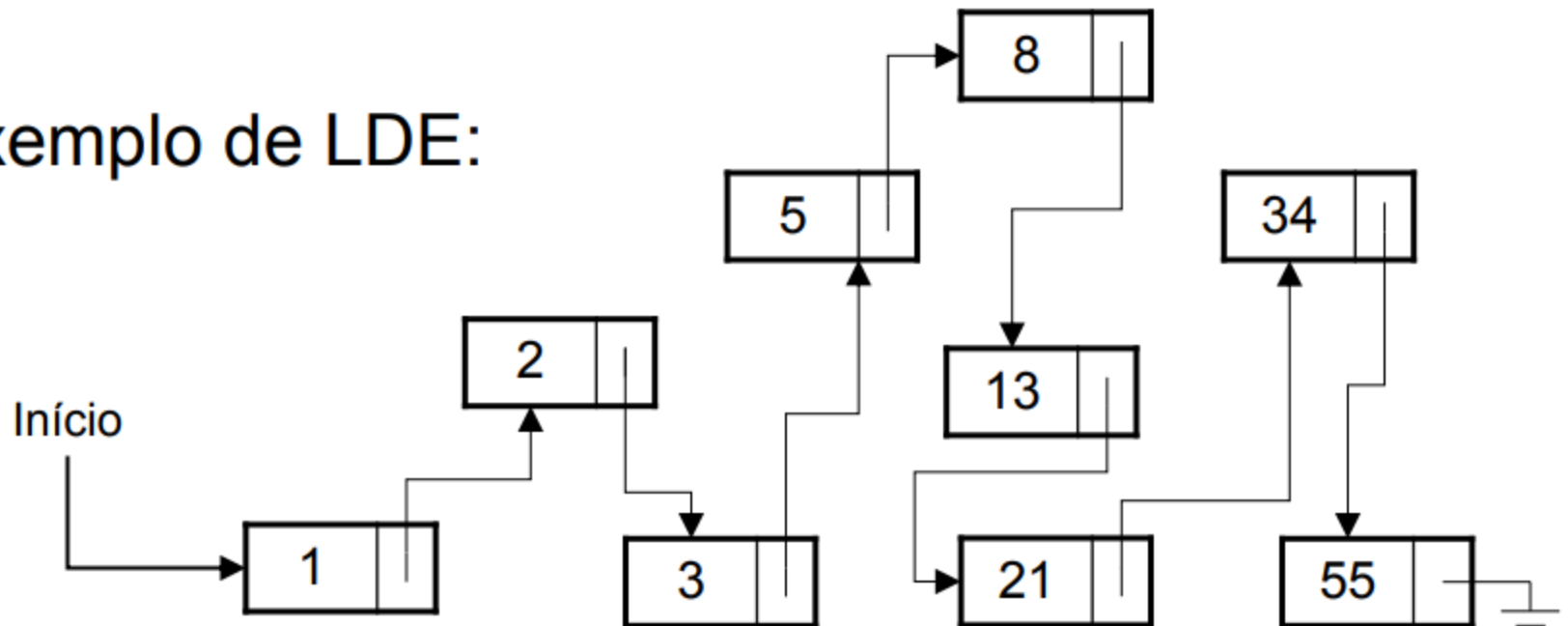
- Características LDE:
 - – Não há uma ordenação contígua.
 - Ou seja, na memória os elementos não estão ordenados de forma seqüencial como na LES.
 - A lista é ordenada através do encadeamento de seus ponteiros.
 - Embora não haja ordenação contígua, os dados devem estar ordenados se percorridos um a um através de seus apontadores.

Lista Dinâmica Encadeada

- Exemplo de LES:

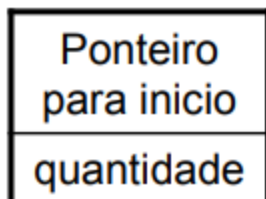
0	1	2	3	4	5	6	7	8
1	2	3	5	8	13	21	34	55

- Exemplo de LDE:

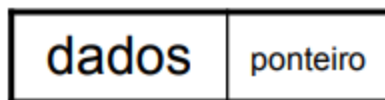


Lista Dinâmica Encadeada

- Em Ciência da Computação uma lista dinâmica encadeada é uma estrutura de dados que:
 - Consiste de uma seqüência de registros
 - Cada registro tem um campo que contém um ponteiro para o próximo item da lista



Lista



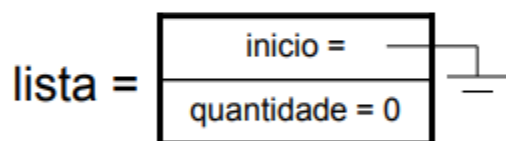
Item da lista

Lista Dinâmica Encadeada

- Operações Básicas
 - Inserção de um elemento na Lista
 - Eliminação de um elemento da Lista
 - Consulta da pertinência de um elemento na Lista

Lista Dinâmica Encadeada

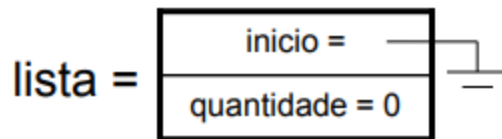
- Funcionamento (Inserção)



A priori a lista está vazia: início aponta para NULL e quantidade é 0.

Lista Dinâmica Encadeada

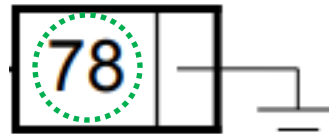
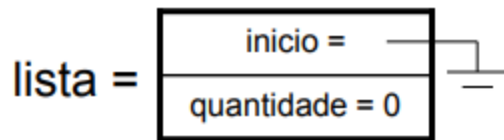
- Funcionamento (Inserção)



- • Inserir o elemento: 78
- • Alocar memória para um item.
 - Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
 - Caso a lista esteja vazia.
 - Faça lista no campo início apontar para o item
 - Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

- Funcionamento (Inserção)

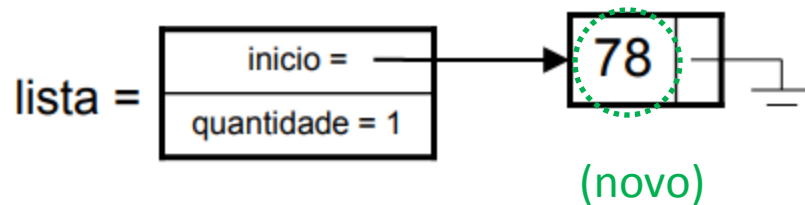


(novo)

- • Inserir o elemento: 78
- • Alocar memória para um item.
- • Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- • Caso a lista esteja vazia.
 - Faça lista no campo início apontar para o item
 - Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

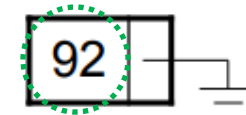
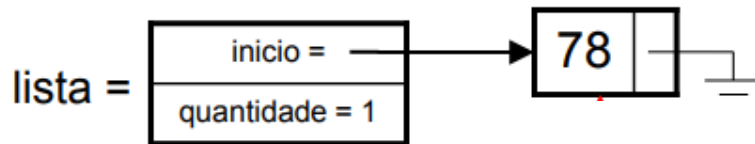
- Funcionamento (Inserção)



- Inserir o elemento: 78
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- • Caso a lista esteja vazia.
- • Faça lista no campo início apontar para o item
- • Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

- Funcionamento (Inserção)

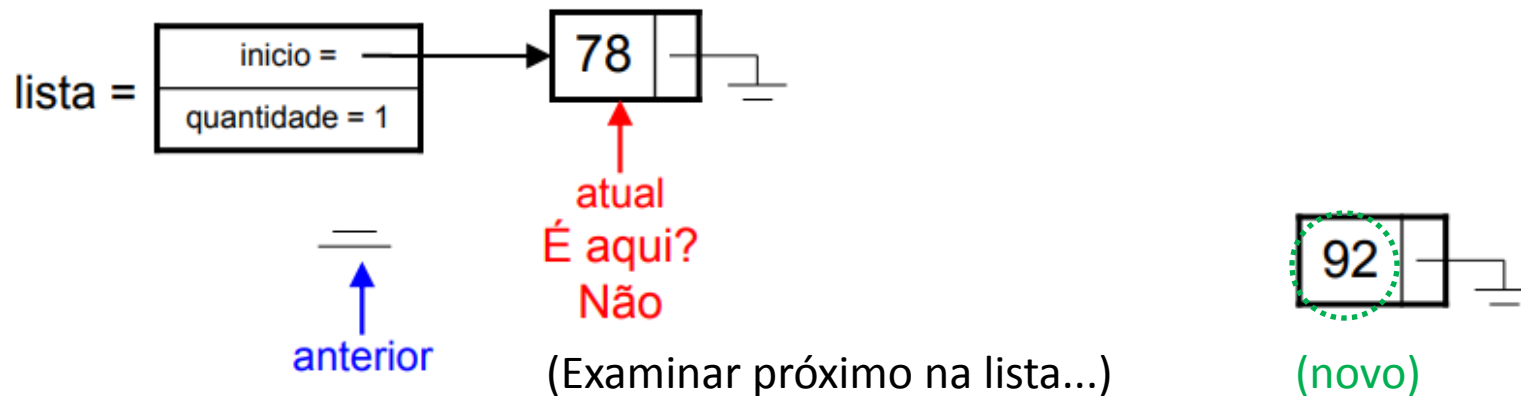


(novo)

- • Inserir o elemento: 92
- • Alocar memória para um item.
- • Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- • A lista não está vazia.
- • Percorra a lista desde o início a procura da posição correta do item ou o final da lista
 - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
 - Atribua ao campo próximo do elemento anterior o endereço do novo item
 - Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

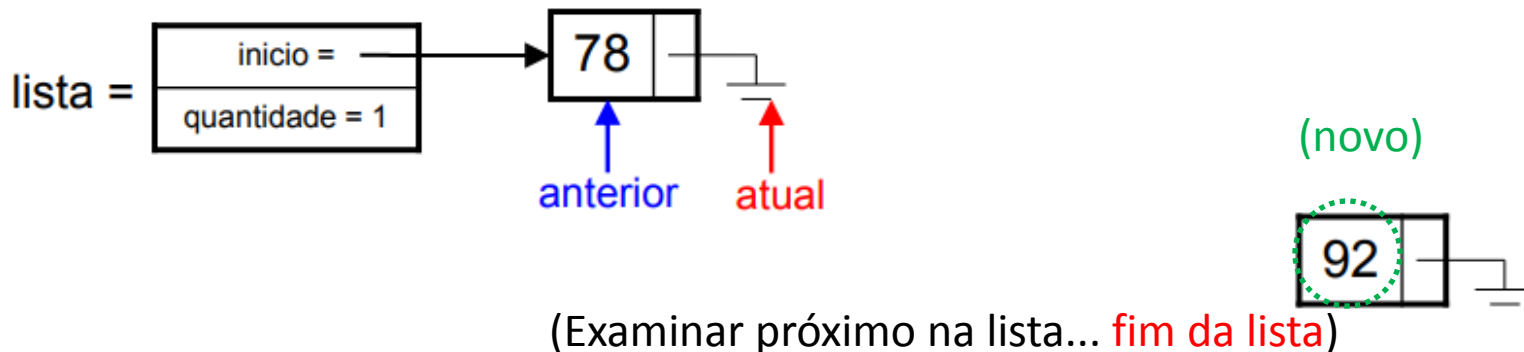
- Funcionamento (Inserção)



- Inserir o elemento: 92
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.
- - Percorra a lista desde o início a procura da posição correta do item ou o final da lista
 - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
 - Atribua ao campo próximo do elemento anterior o endereço do novo item
 - Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

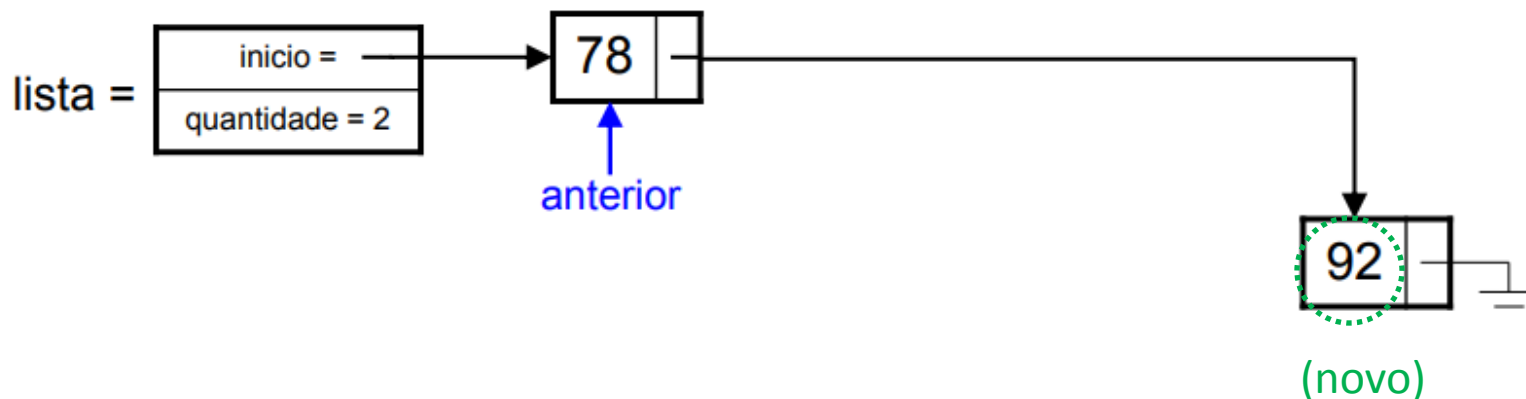
- Funcionamento (Inserção)



- Inserir o elemento: 92
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.
- • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- • Atribua ao campo próximo do elemento anterior o endereço do novo item
- Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

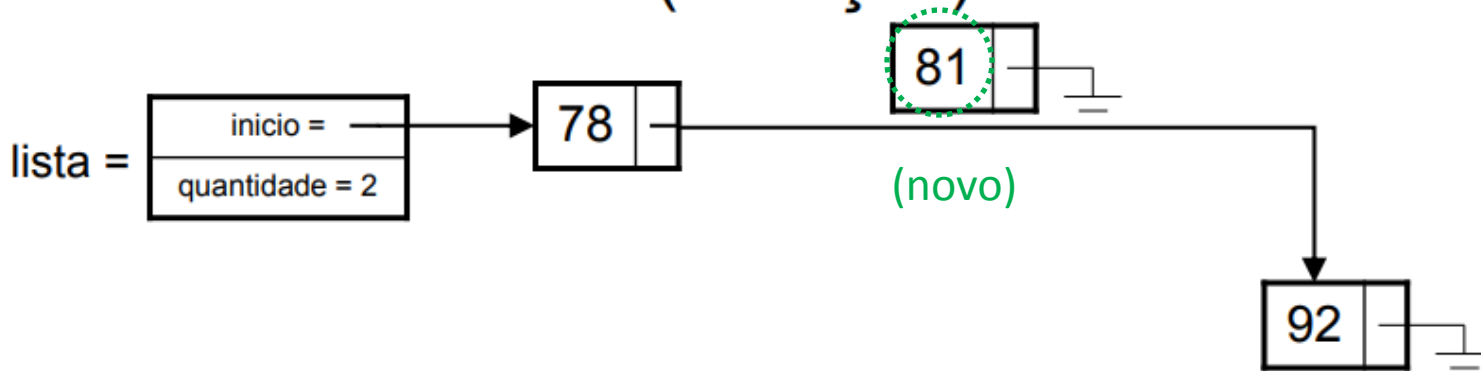
- Funcionamento (Inserção)



- Inserir o elemento: 92
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.
- • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- • Atribua ao campo próximo do elemento anterior o endereço do novo item
- • Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

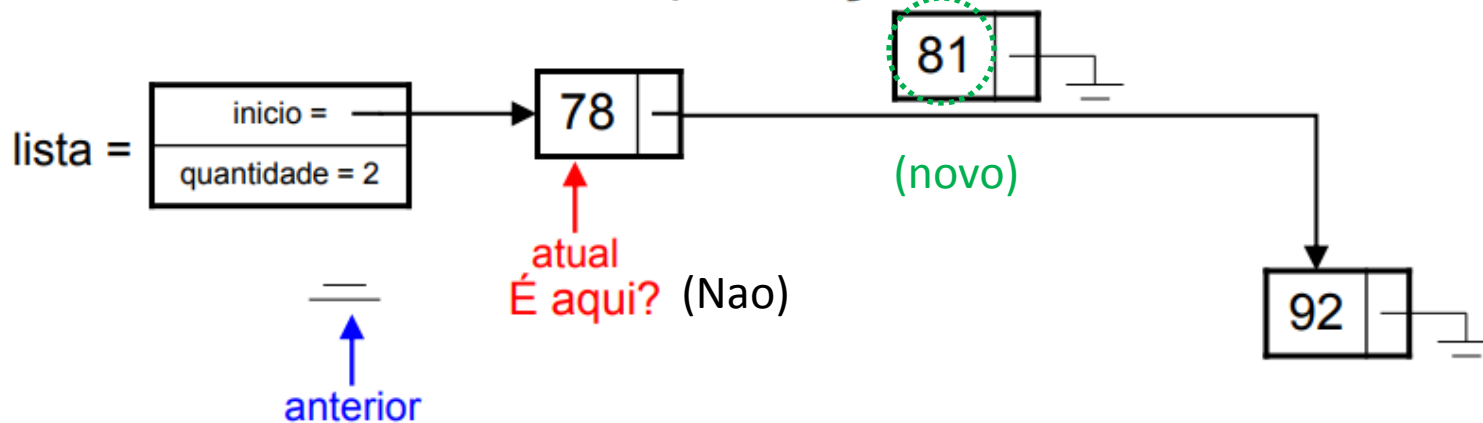
- Funcionamento (Inserção)



- • Inserir o elemento: 81
- • Alocar memória para um item.
- • Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.
 - Percorra a lista desde o início a procura da posição correta do item ou o final da lista
 - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
 - Atribua ao campo próximo do elemento anterior o endereço do novo item
 - Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

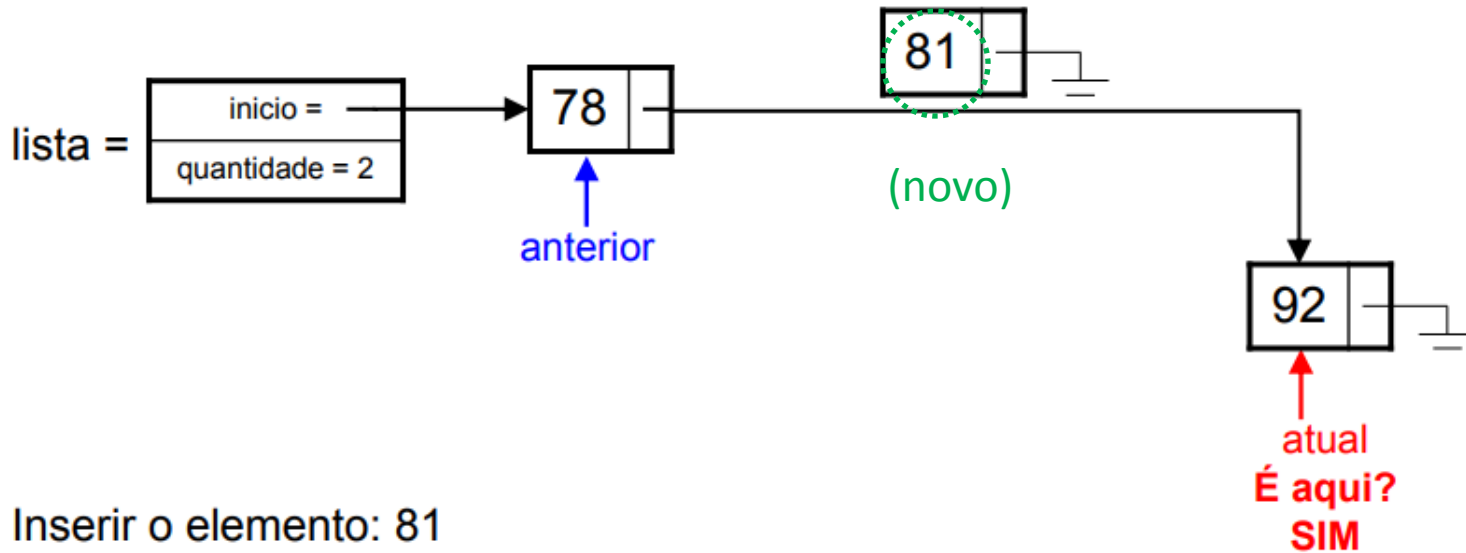
- Funcionamento (Inserção)



- Inserir o elemento: 81
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- • A lista não está vazia.
- • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
 - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
 - Atribua ao campo próximo do elemento anterior o endereço do novo item
 - Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

- Funcionamento (Inserção)



- Inserir o elemento: 81
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.

→ • A lista não está vazia.

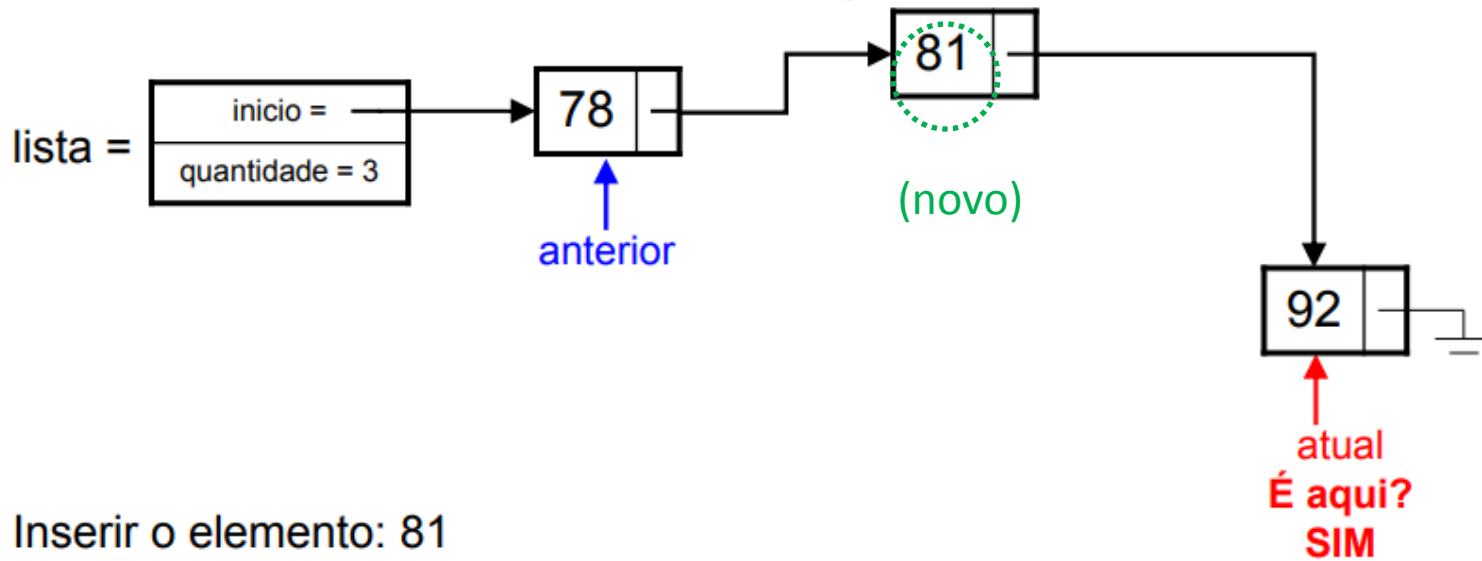
→ • Percorra a lista deste o início a procura da posição correta do item ou o final da lista

→ • Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.

- Atribua ao campo próximo do elemento anterior o endereço do novo item
- Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

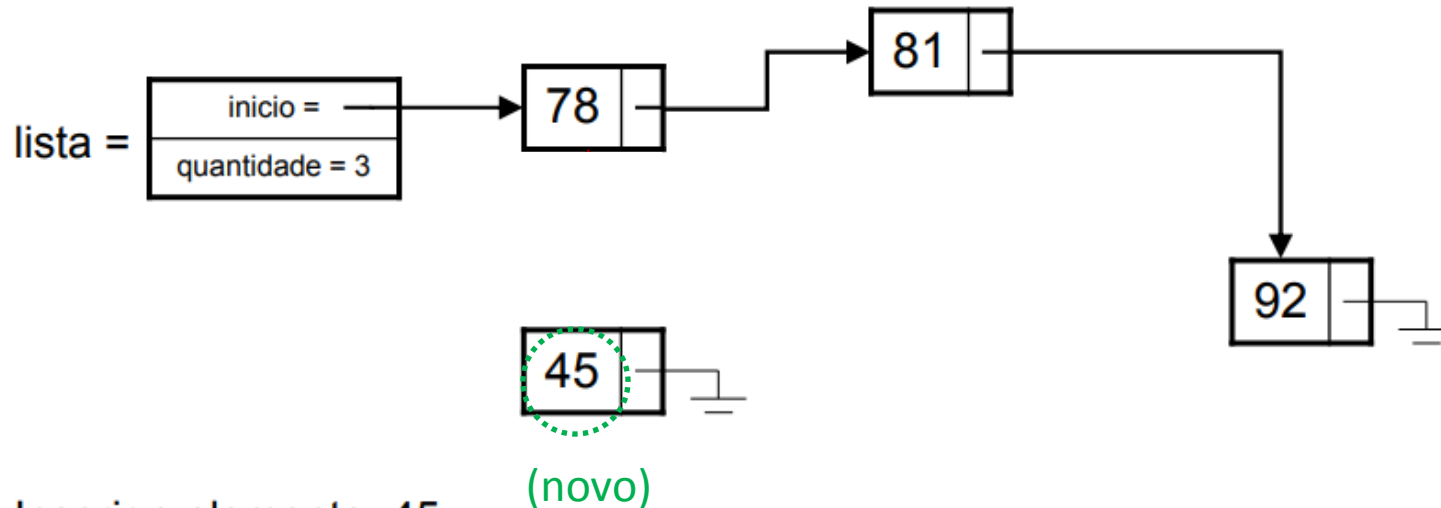
- Funcionamento (Inserção)



- Inserir o elemento: 81
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.
 - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
 - Ao encontrar atribua ao campo próximo de novo item o endereço elemento atual.
 - Atribua ao campo próximo do elemento anterior o endereço do novo item
 - Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

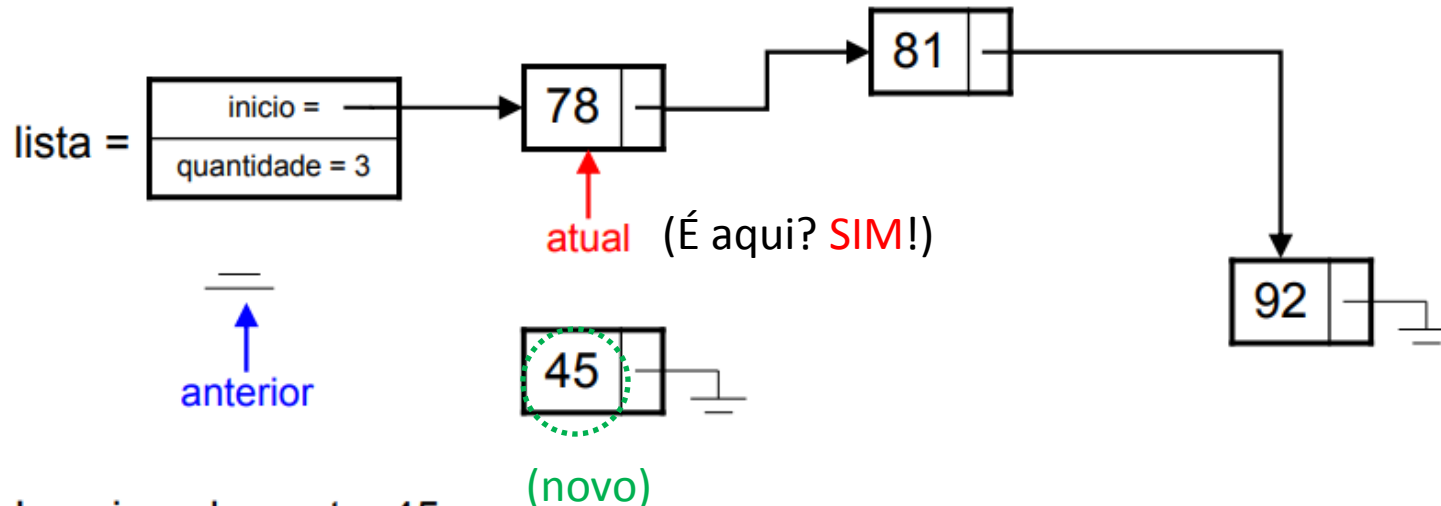
- Funcionamento (Inserção)



- • Inserir o elemento: 45
- • Alocar memória para um item.
- • Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.
 - Percorra a lista desde o início a procura da posição correta do item ou o final da lista
 - O anterior é nulo ? Se sim significa que é o primeiro da lista, então
 - Atribua ao campo próximo do novo item o valor de atual.
 - Atribua ao campo inicio da lista o endereço novo elemento.

Lista Dinâmica Encadeada

- Funcionamento (Inserção)



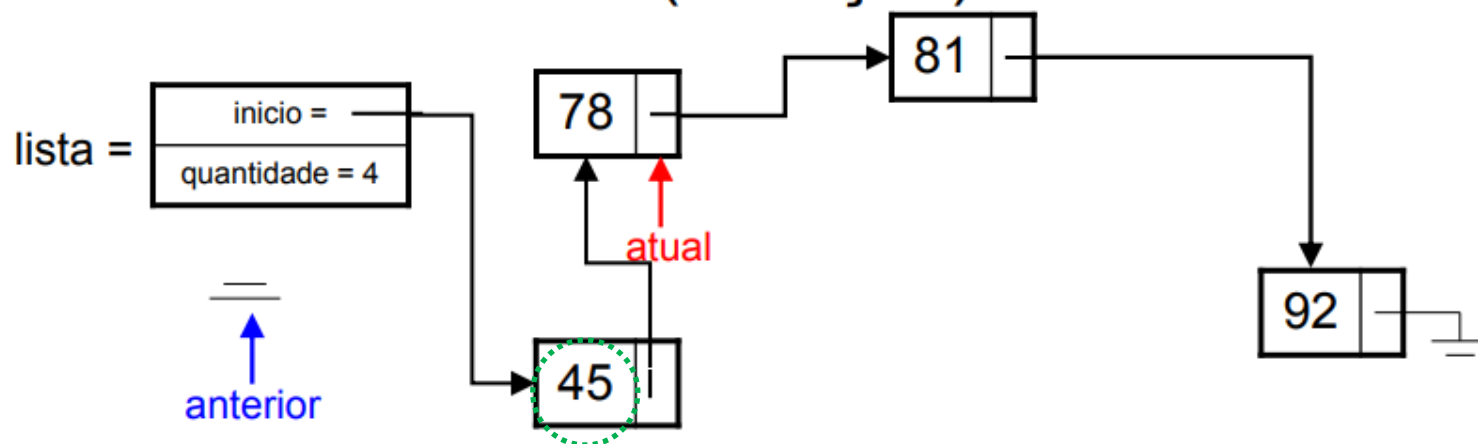
- Inserir o elemento: 45
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.

→ • A lista não está vazia.

- • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- • O anterior é nulo ? Se sim significa que é o primeiro da lista, então
 - Atribua ao campo próximo do novo item o valor de atual.
 - Atribua ao campo inicio da lista o endereço novo elemento.

Lista Dinâmica Encadeada

- Funcionamento (Inserção)



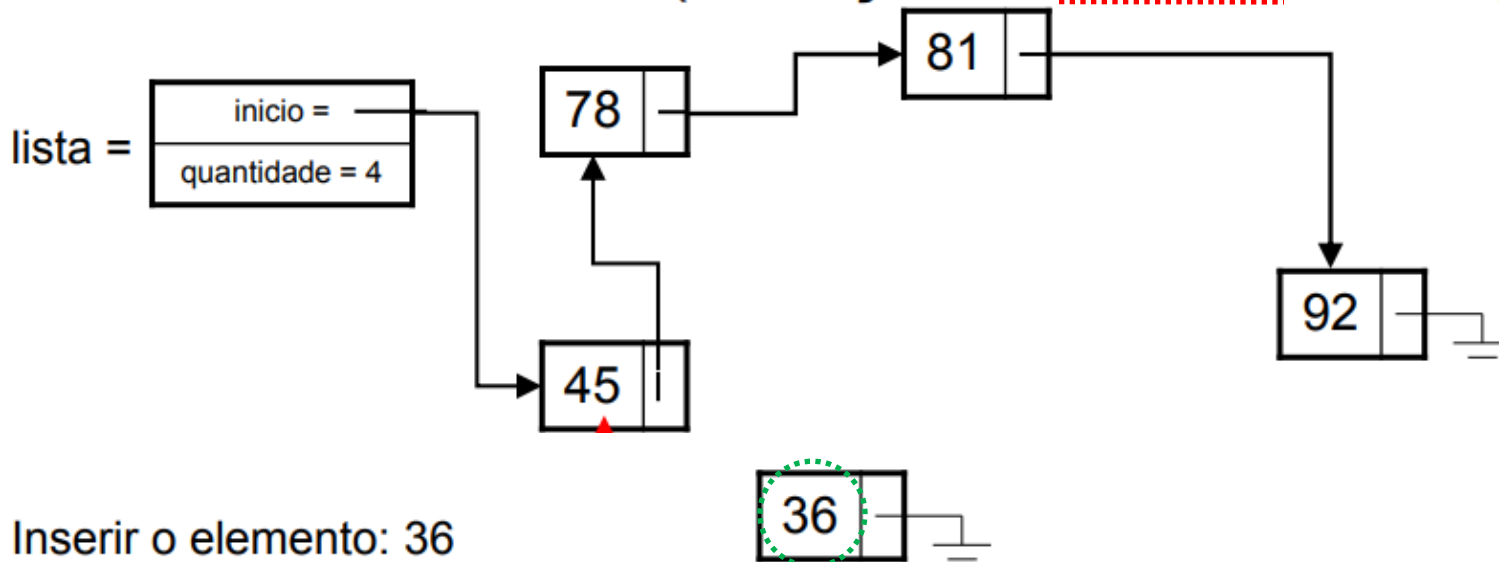
- Inserir o elemento: 45 (novo)
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.
 - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- O anterior é nulo ? Se sim significa que é o primeiro da lista, então
- Atribua ao campo próximo do novo item o valor de atual.
- Atribua ao campo inicio da lista o endereço novo elemento.

Lista Dinâmica Encadeada

- Funcionamento (Inserção)
 - Observe cada situação da inserção
 - Inserir no início da Lista
 - Vazia
 - Não Vazia
 - Inserir no meio da Lista
 - Inserir no final da Lista

Lista Dinâmica Encadeada

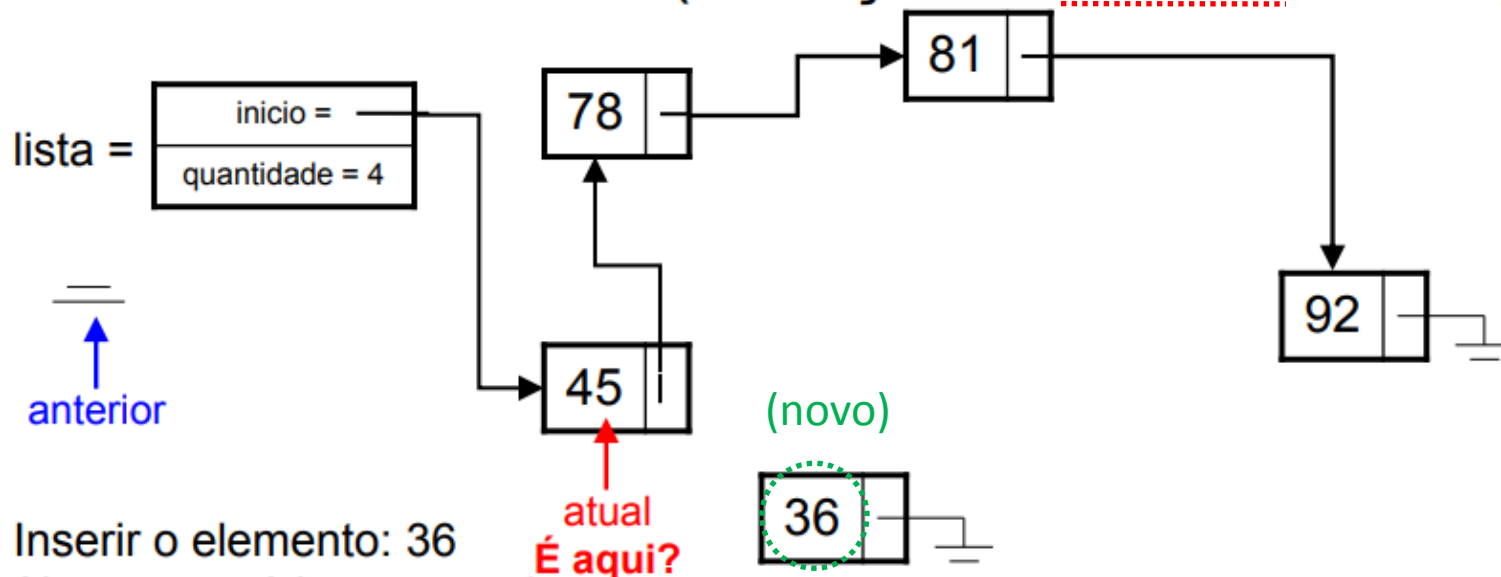
- Funcionamento (Inserção no início da lista)



- Inserir o elemento: 36
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.
 - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
 - O anterior é nulo ? Se sim significa que é o primeiro da lista, então
 - Atribua ao campo próximo do novo item o valor de atual.
 - Atribua ao campo inicio da lista o endereço novo elemento.

Lista Dinâmica Encadeada

- Funcionamento (Inserção no início da lista)



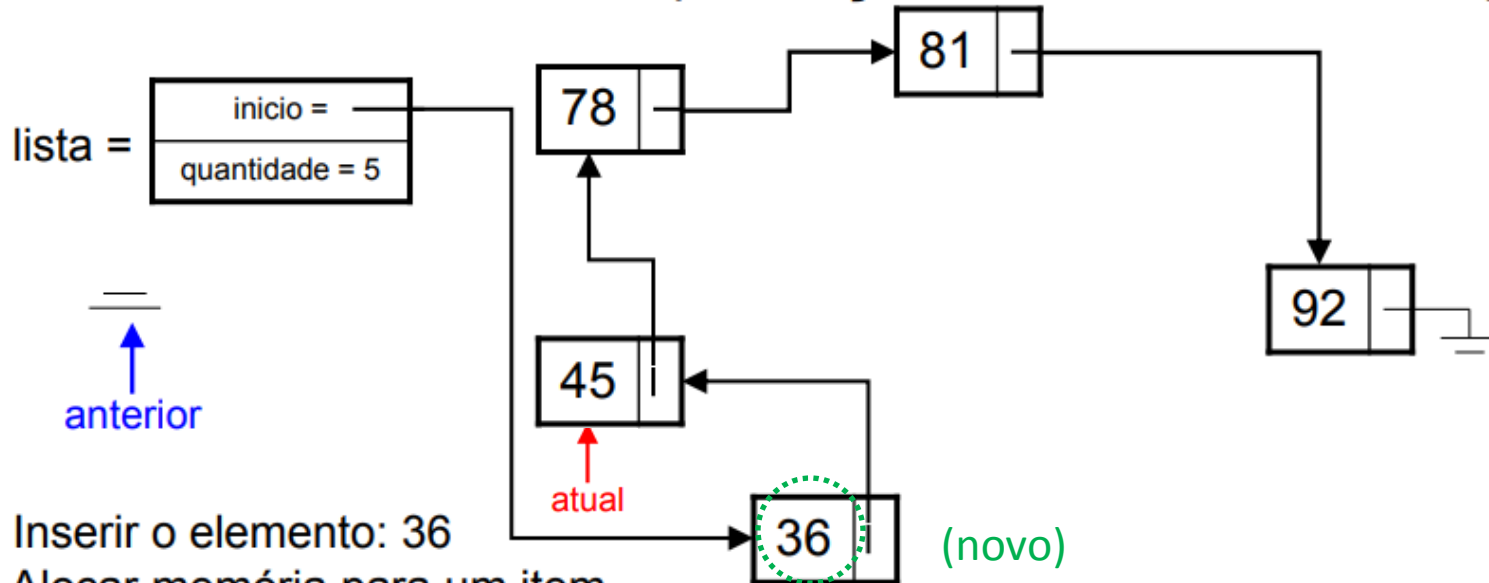
- Inserir o elemento: 36
- Alocar memória para um item. Sim!
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.

→ A lista não está vazia.

- • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- • O anterior é nulo ? Se sim significa que é o primeiro da lista, então
 - Atribua ao campo próximo do novo item o valor de atual.
 - Atribua ao campo inicio da lista o endereço novo elemento.

Lista Dinâmica Encadeada

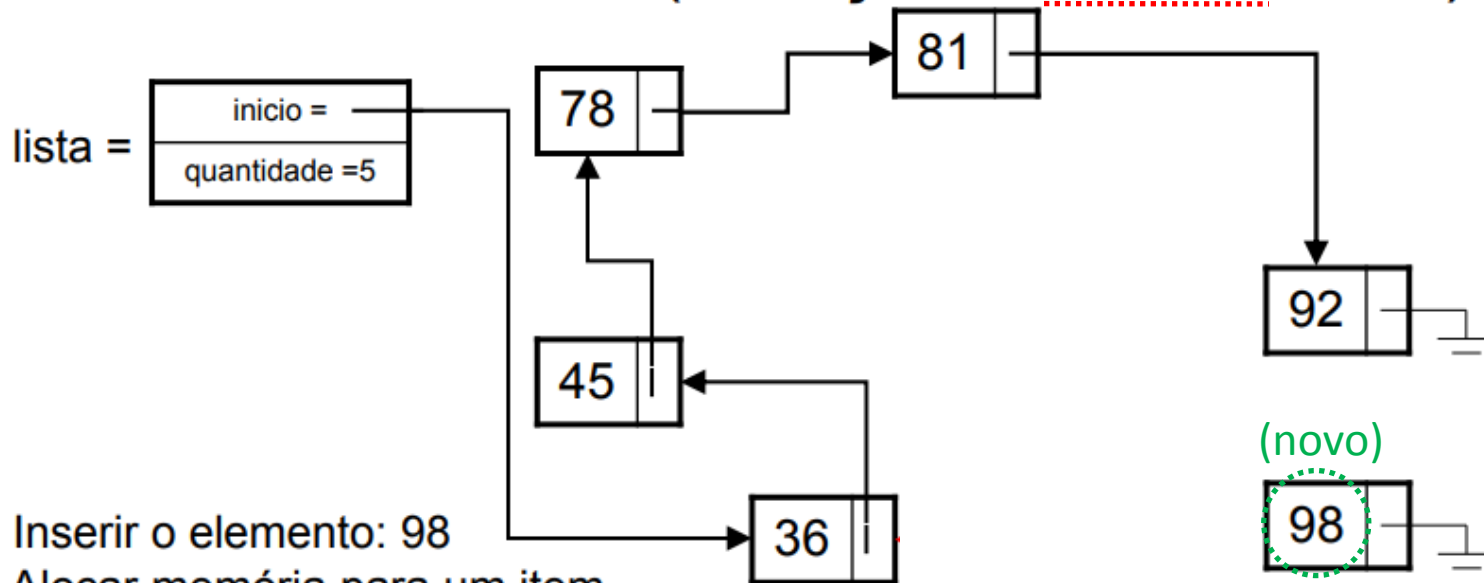
- Funcionamento (Inserção no início da lista)



- Inserir o elemento: 36
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.
 - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- • O anterior é nulo ? Se sim significa que é o primeiro da lista, então
- • Atribua ao campo próximo do novo item o valor de atual.
- • Atribua ao campo inicio da lista o endereço novo elemento.

Lista Dinâmica Encadeada

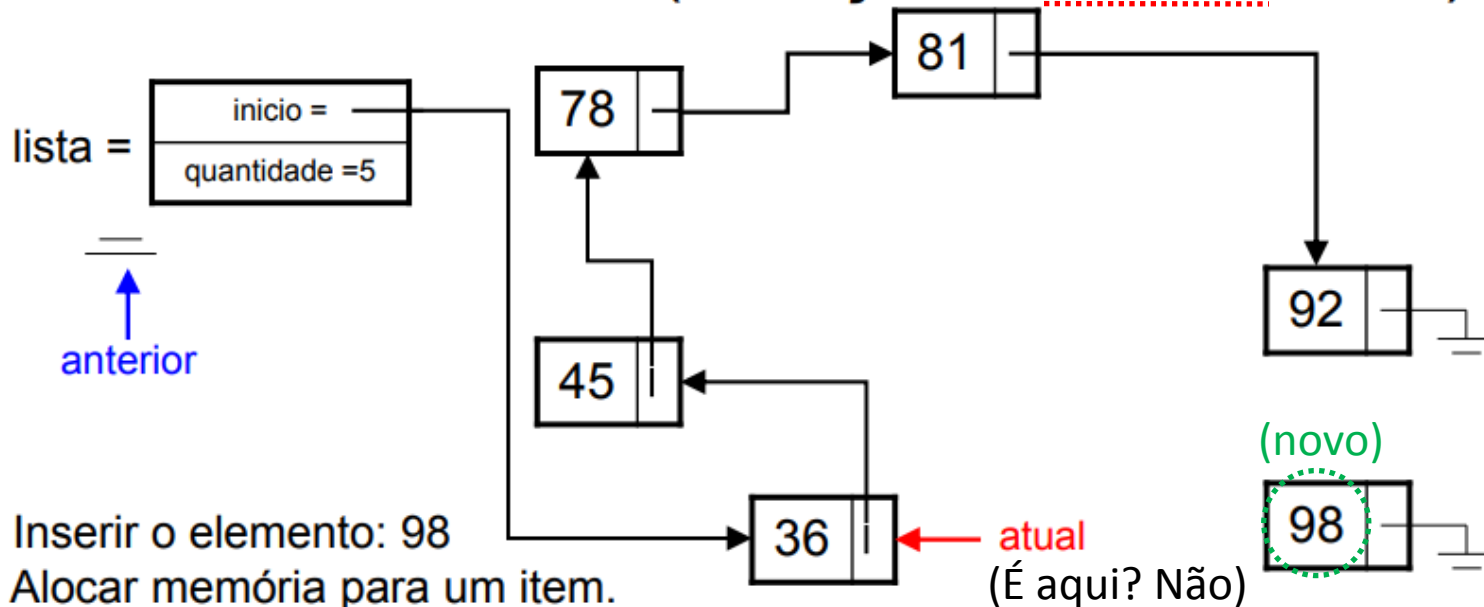
- Funcionamento (Inserção no final da lista)



- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.
 - Percorra a lista deste o início a procura da posição correta do item ou o final da lista
 - Ao terminar atribua ao campo próximo de novo item o ponteiro para elemento atual.
 - Atribua ao campo próximo do elemento anterior o endereço do novo item
 - Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

- Funcionamento (Inserção no final da lista)



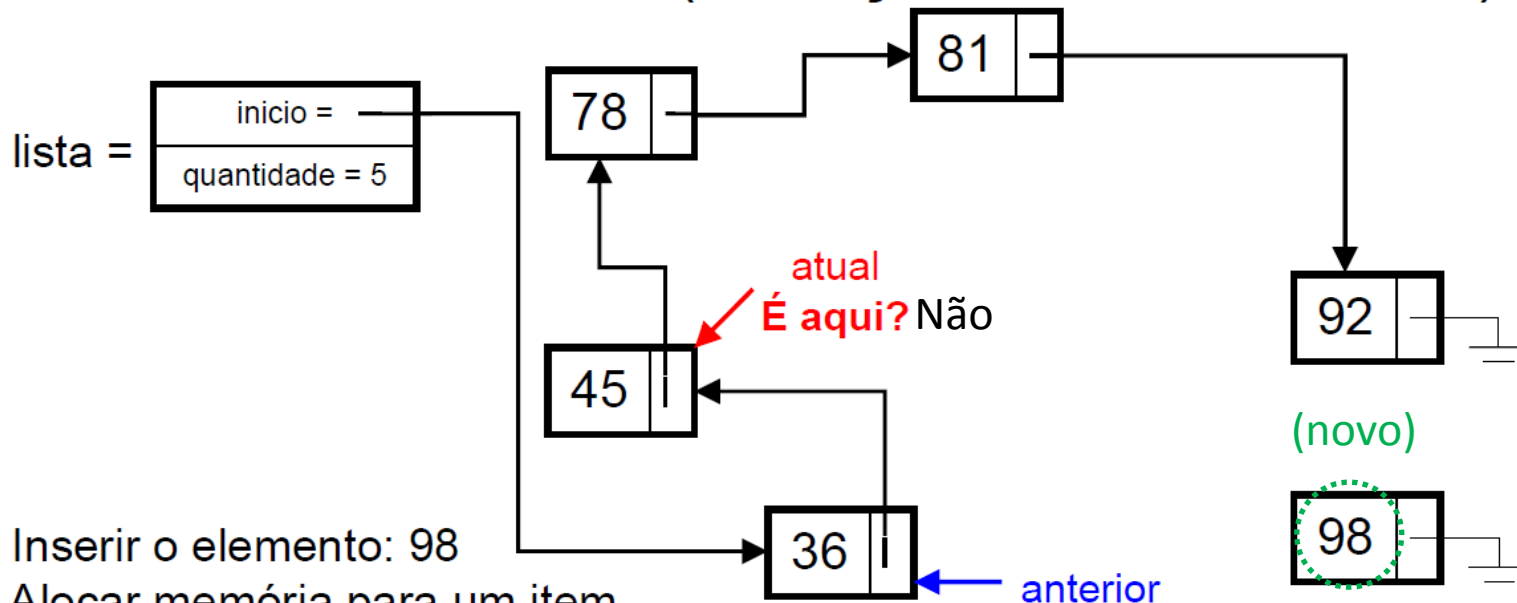
- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.

→ A lista não está vazia.

- • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- Ao terminar atribua ao campo próximo de novo item o ponteiro para elemento atual.
- Atribua ao campo próximo do elemento anterior o endereço do novo item
- Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

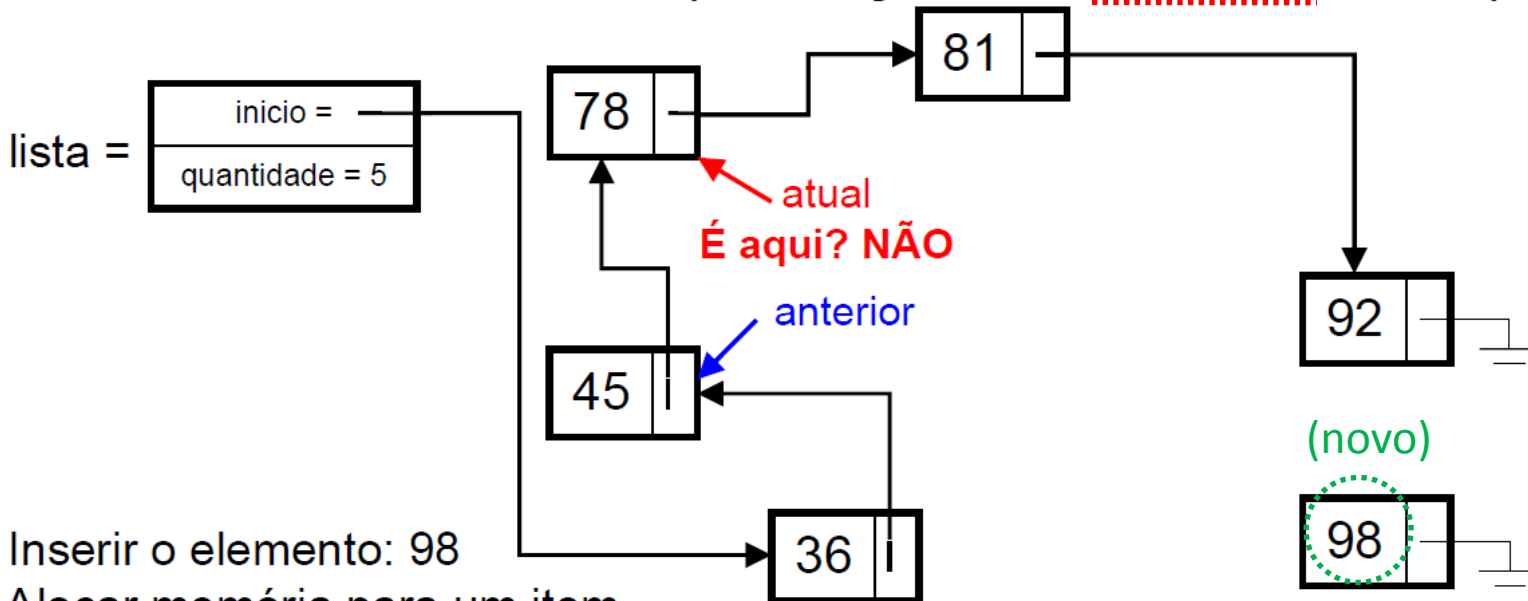
- Funcionamento (Inserção no final da lista)



- Inserir o elemento: 98
 - Alocar memória para um item.
 - Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
 - A lista não está vazia.
- • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- Ao terminar atribua ao campo próximo de novo item o ponteiro para elemento atual.
 - Atribua ao campo próximo do elemento anterior o endereço do novo item
 - Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

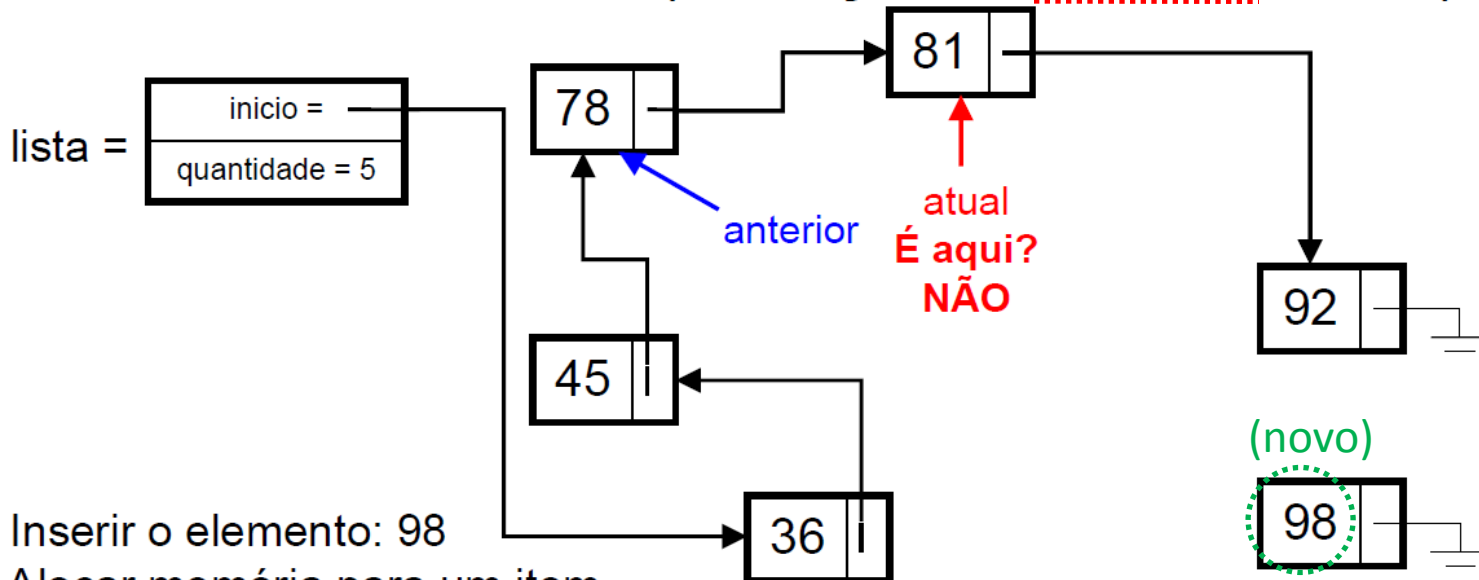
- Funcionamento (Inserção no final da lista)



- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.
- • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- Ao terminar atribua ao campo próximo de novo item o ponteiro para elemento atual.
- Atribua ao campo próximo do elemento anterior o endereço do novo item
- Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

- Funcionamento (Inserção no final da lista)

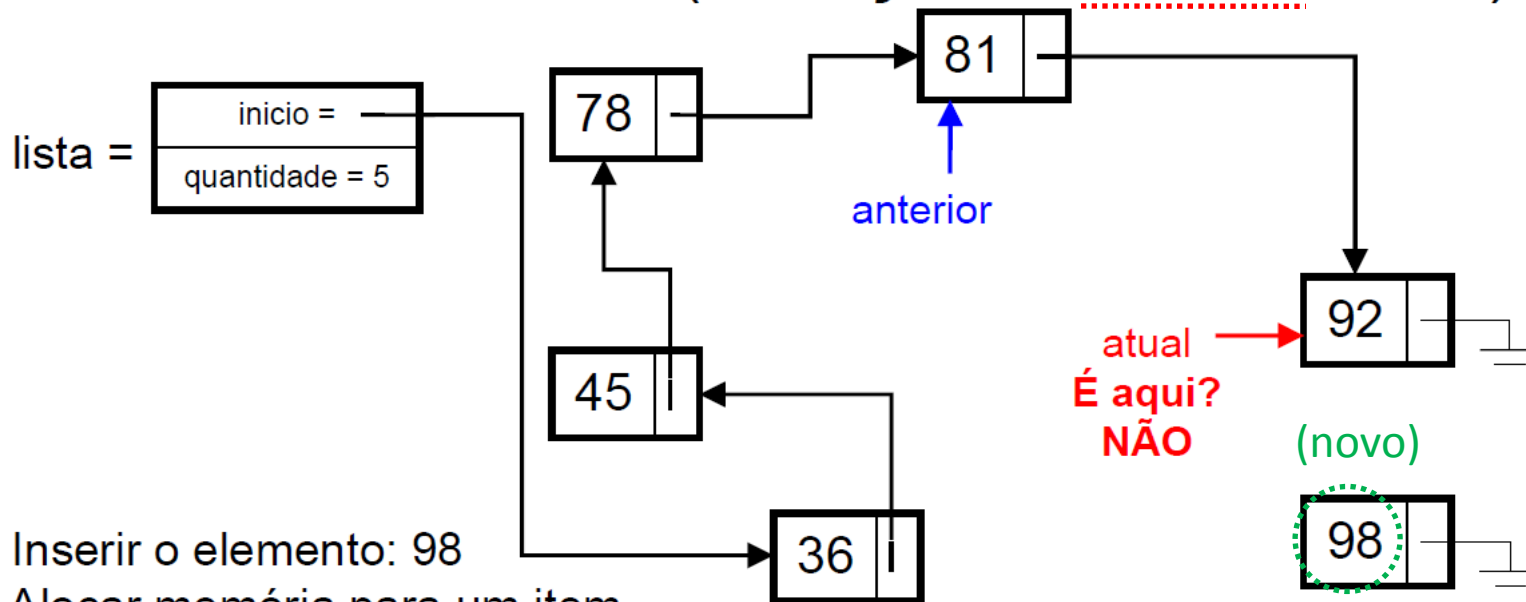


- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.

-
- Percorra a lista deste o início a procura da posição correta do item ou o final da lista
 - Ao terminar atribua ao campo próximo de novo item o ponteiro para elemento atual.
 - Atribua ao campo próximo do elemento anterior o endereço do novo item
 - Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

- Funcionamento (Inserção no final da lista)

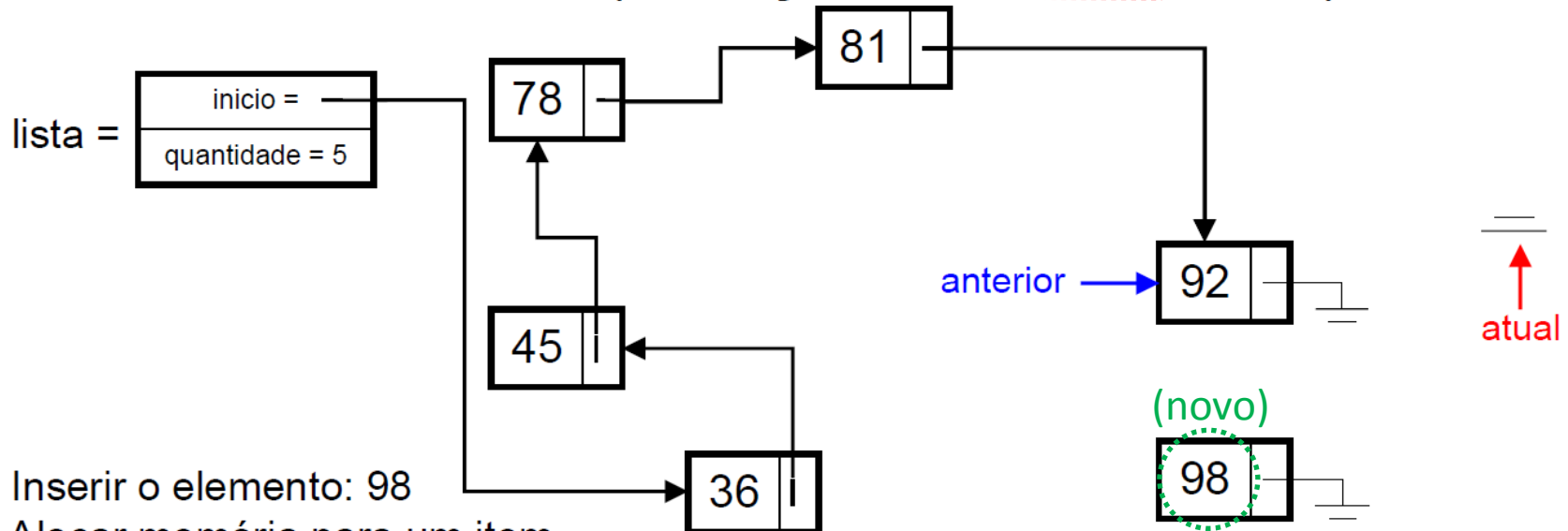


- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.

- • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- Ao terminar atribua ao campo próximo de novo item o ponteiro para elemento atual.
- Atribua ao campo próximo do elemento anterior o endereço do novo item
- Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

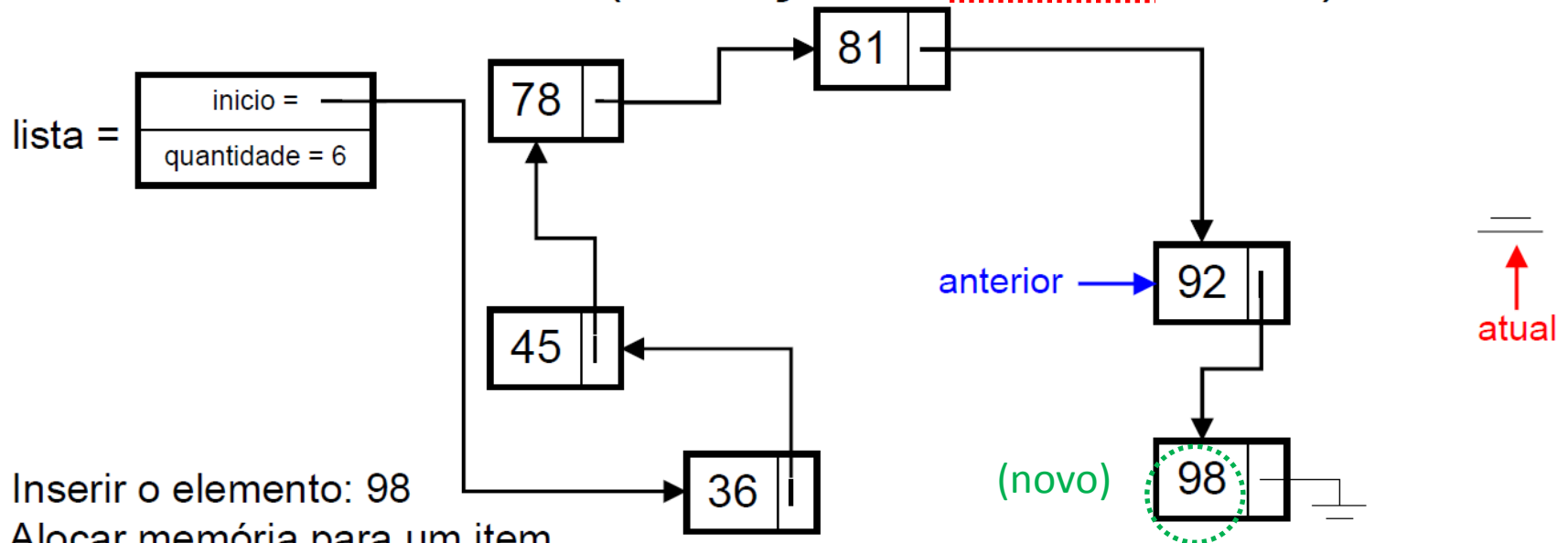
- Funcionamento (Inserção no final da lista)



- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.
- • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- • Atribua ao campo próximo do elemento anterior o endereço do novo item
- • Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

- Funcionamento (Inserção no final da lista)



- Inserir o elemento: 98
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.

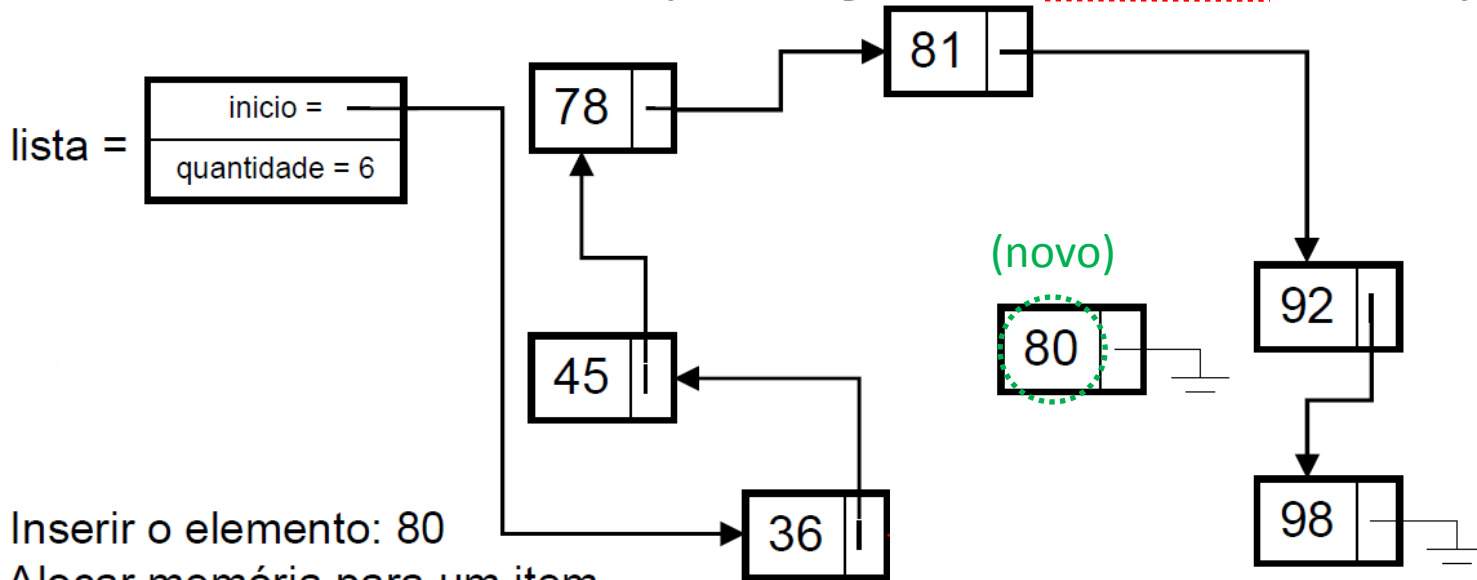
→ • Percorra a lista deste o início a procura da posição correta do item ou o final da lista

→ • Atribua ao campo próximo do elemento anterior o endereço do novo item

→ • Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

- Funcionamento (Inserção no meio da lista)

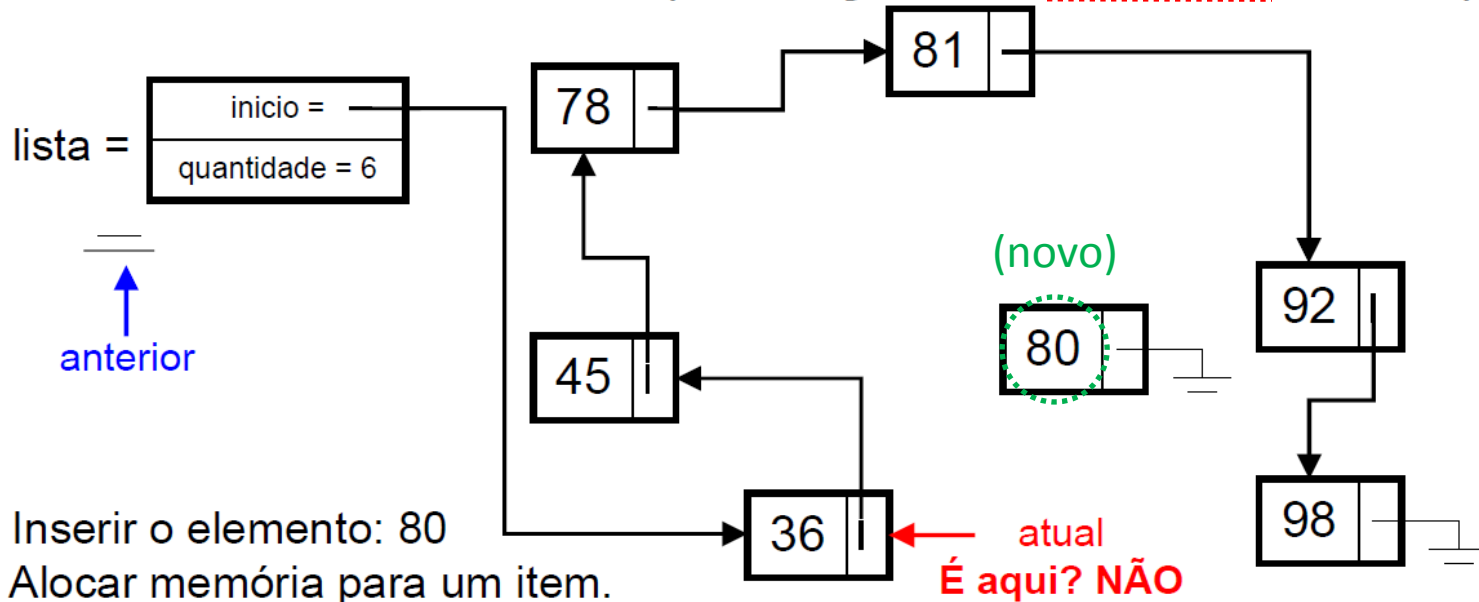


- Inserir o elemento: 80
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.

- Percorra a lista deste o início a procura da posição correta do item ou o final da lista
- Ao terminar atribua ao campo próximo de novo item o ponteiro para elemento atual.
- Atribua ao campo próximo do elemento anterior o endereço do novo item
- Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

- Funcionamento (Inserção no meio da lista)



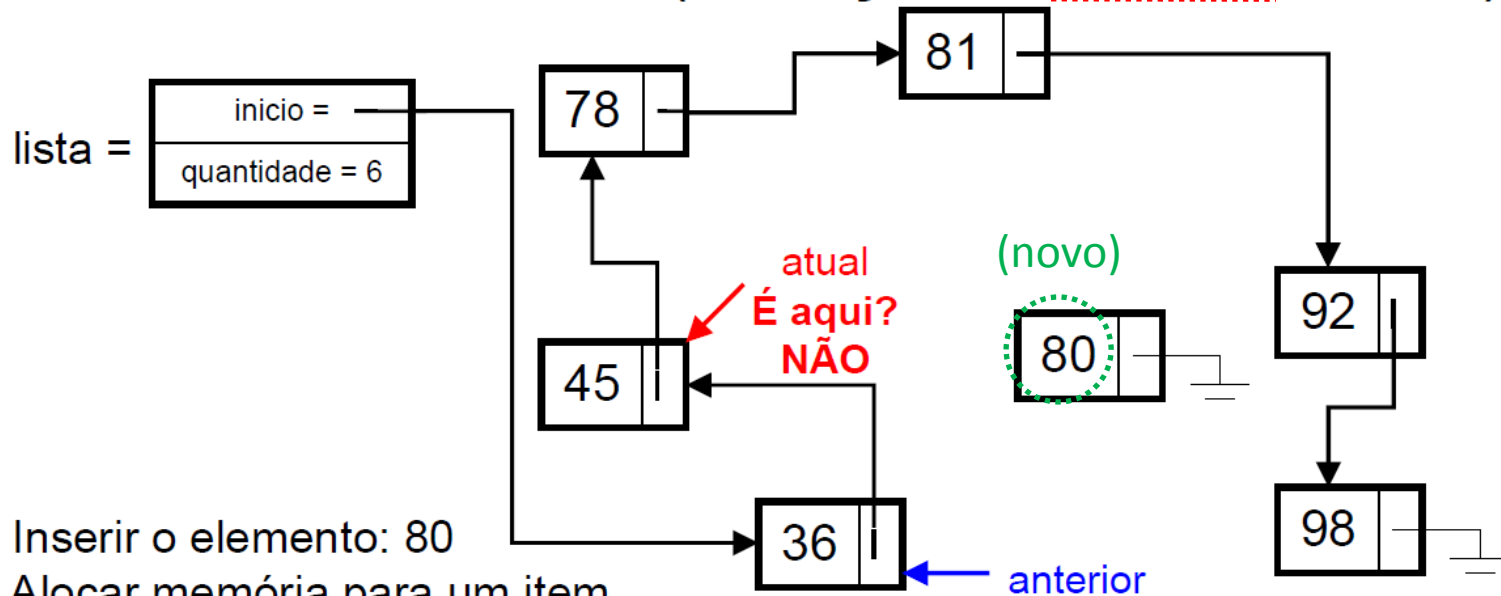
- Inserir o elemento: 80
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.

→ A lista não está vazia.

- • Percorra a lista deste o início a procura da posição correta do item ou o final da lista
 - Ao terminar atribua ao campo próximo de novo item o ponteiro para elemento atual.
 - Atribua ao campo próximo do elemento anterior o endereço do novo item
 - Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

- Funcionamento (Inserção no meio da lista)



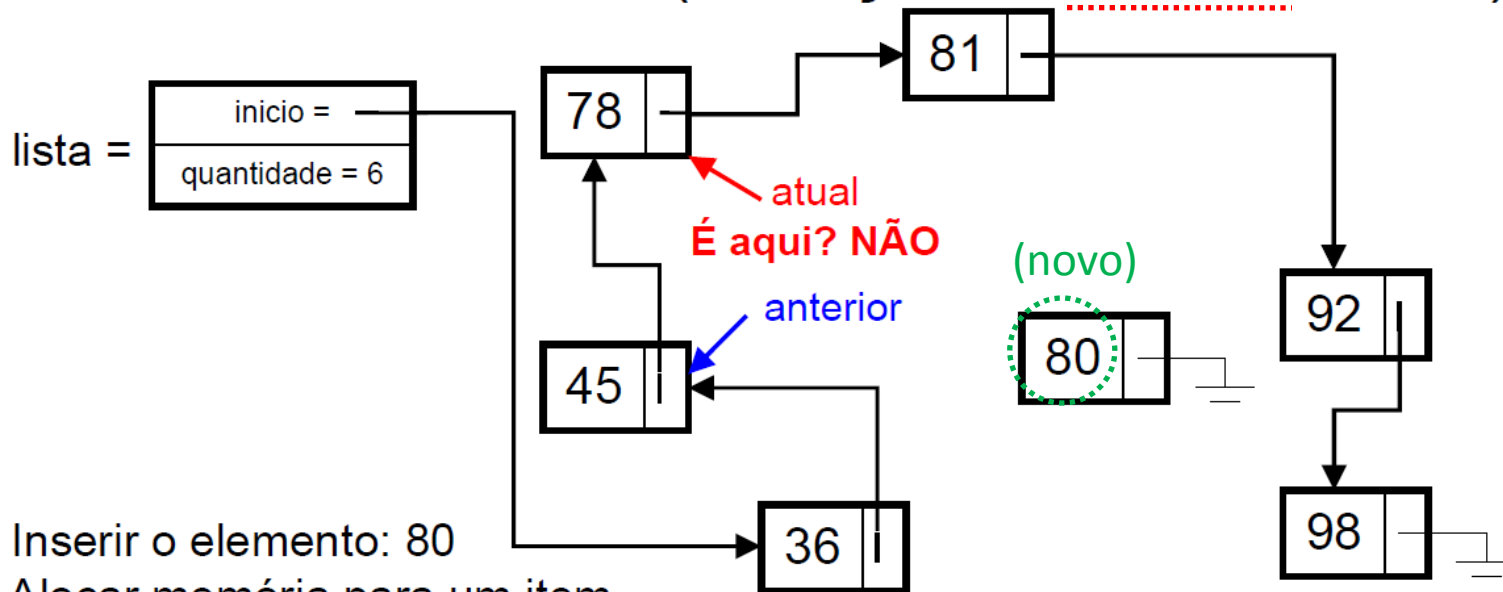
- Inserir o elemento: 80
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.

→ A lista não está vazia.

- Percorra a lista deste o início a procura da posição correta do item ou o final da lista
 - Ao terminar atribua ao campo próximo de novo item o ponteiro para elemento atual.
 - Atribua ao campo próximo do elemento anterior o endereço do novo item
 - Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

- Funcionamento (Inserção no meio da lista)



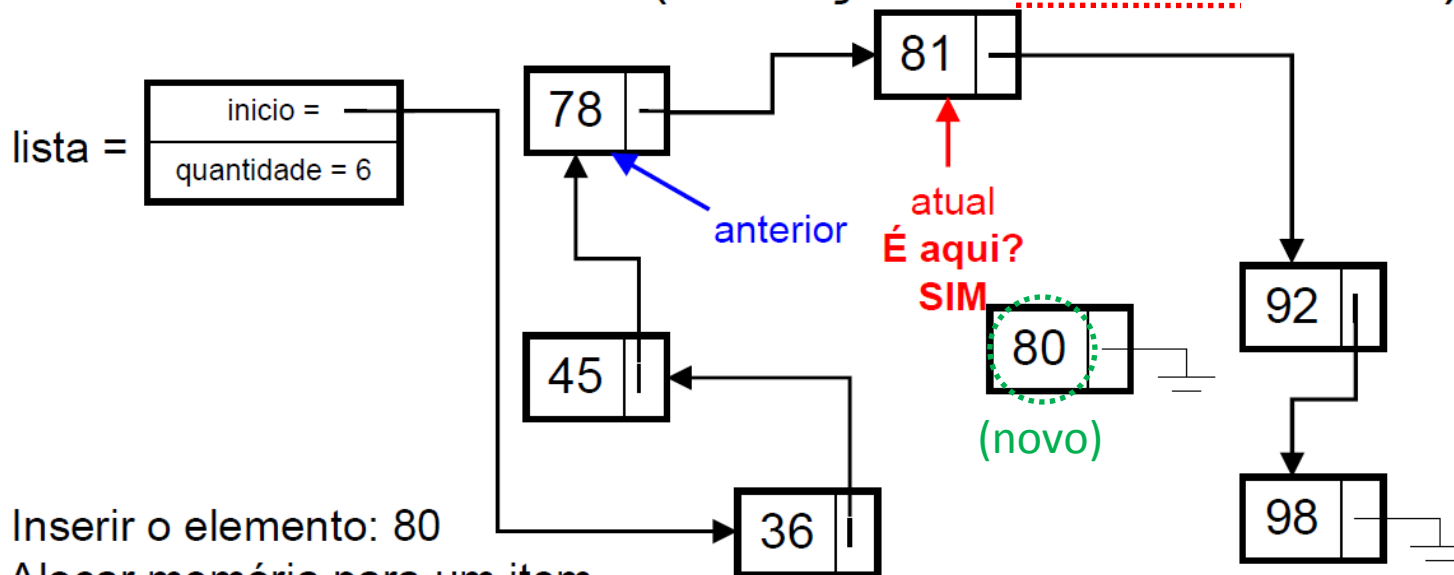
- Inserir o elemento: 80
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.

→ A lista não está vazia.

- Percorra a lista deste o início a procura da posição correta do item ou o final da lista
 - Ao terminar atribua ao campo próximo de novo item o ponteiro para elemento atual.
 - Atribua ao campo próximo do elemento anterior o endereço do novo item
 - Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

- Funcionamento (Inserção no meio da lista)

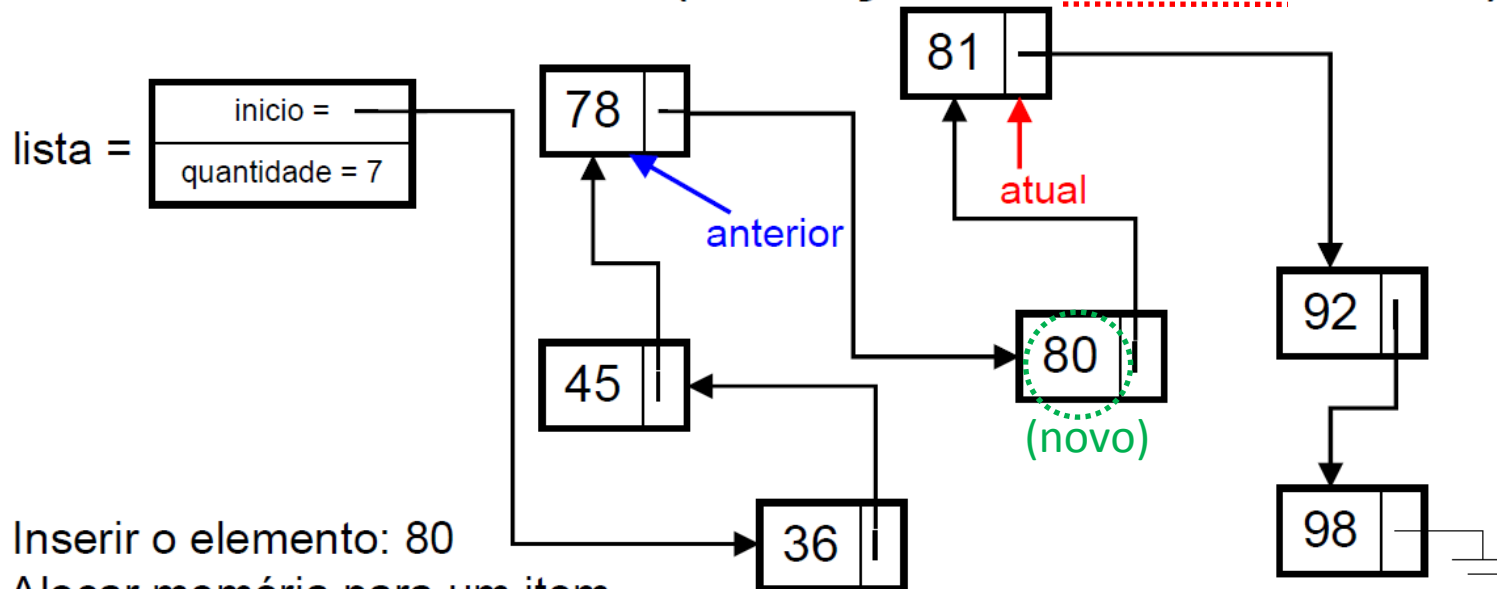


- Inserir o elemento: 80
- Alocar memória para um item.
- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.
- A lista não está vazia.

- • Percorra a lista desde o início a procura da posição correta do item ou o final da lista
- • Ao terminar atribua ao campo próximo de novo item o ponteiro para elemento atual.
- • Atribua ao campo próximo do elemento anterior o endereço do novo item
- • Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

- Funcionamento (Inserção no meio da lista)



- Inserir o elemento: 80

- Alocar memória para um item.

- Atribuir a chave a ele e atribuir ao campo próximo o valor nulo.

- A lista não está vazia.

- • Percorra a lista desde o início a procura da posição correta do item ou o final da lista

- • Ao terminar atribua ao campo próximo de novo item o ponteiro para elemento atual.

- • Atribua ao campo próximo do elemento anterior o endereço do novo item

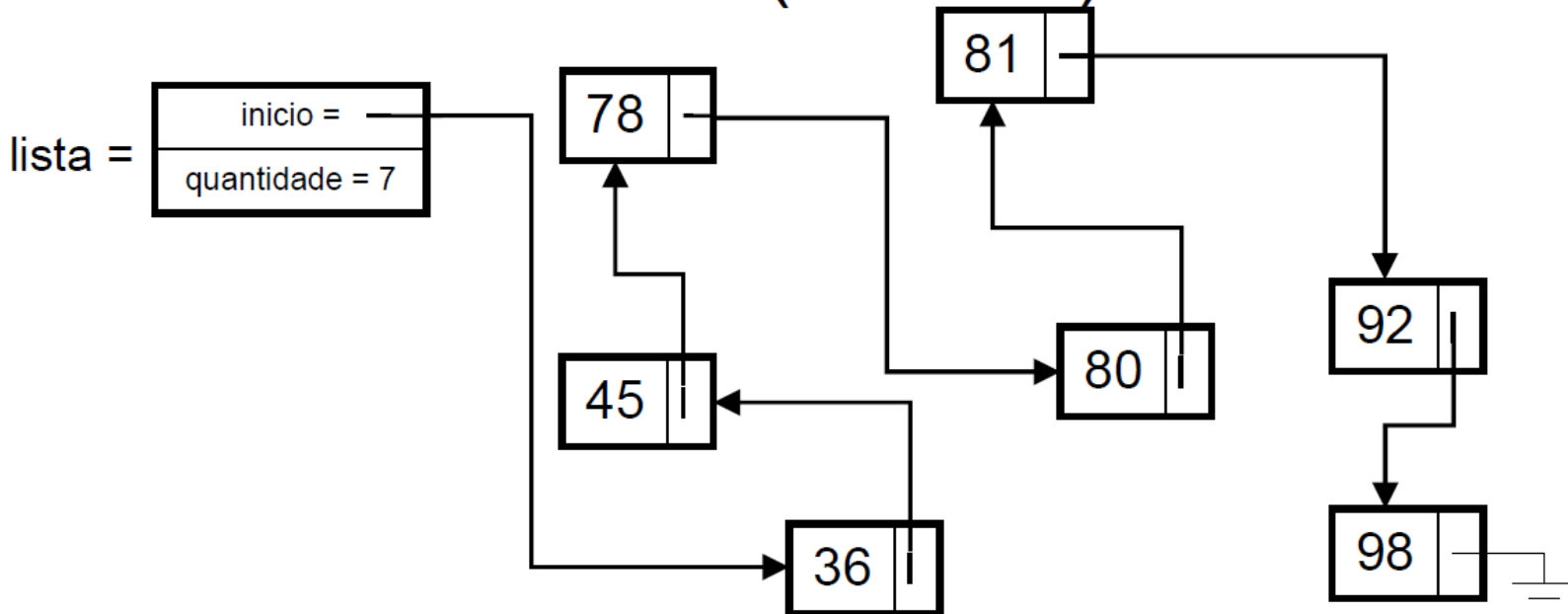
- • Incremente a quantidade em uma unidade.

Lista Dinâmica Encadeada

- Inserção: (Lembretes...)
 - Caso 1: Inserir primeiro elemento em uma lista vazia:
 - Checar ponteiro para início, se for NULL apontar para novo item.
 - Caso 2: Inserir primeiro elemento de uma lista não vazia:
 - Quando encontrar a posição de inserção, verificar se o ponteiro para anterior é NULL, caso seja verdade, próximo do item novo recebe o início e o início recebe o item novo.
 - Caso 3: Inserir o último elemento de uma lista:
 - Quando chegar ao final da lista (atual é NULL). Campo próximo do item novo recebe atual (NULL) e campo próximo do apontado por anterior recebe o endereço do novo item.
 - Caso 4: Inserir no meio da lista:
 - Quando encontra posição de inserção e anterior não é NULL. Campo próximo do item novo recebe atual e campo próximo do anterior recebe o endereço do item novo.

Lista Dinâmica Encadeada

- Funcionamento (Busca)

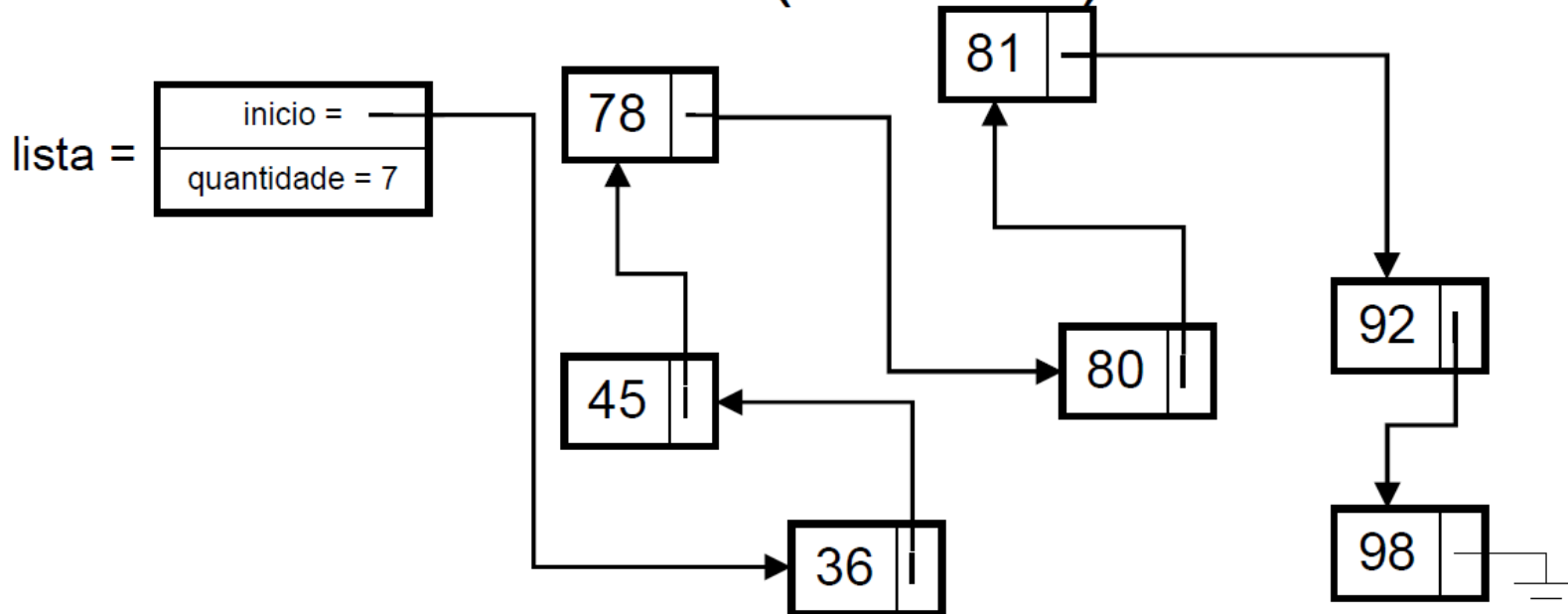


Como faremos a consulta de um item na lista ?

- A lista está ordenada. Busca Binária ? ✗
- Busca Seqüencial (Linear) ✓

Lista Dinâmica Encadeada

- Funcionamento (Busca)



Como faremos a consulta de um item na lista ?

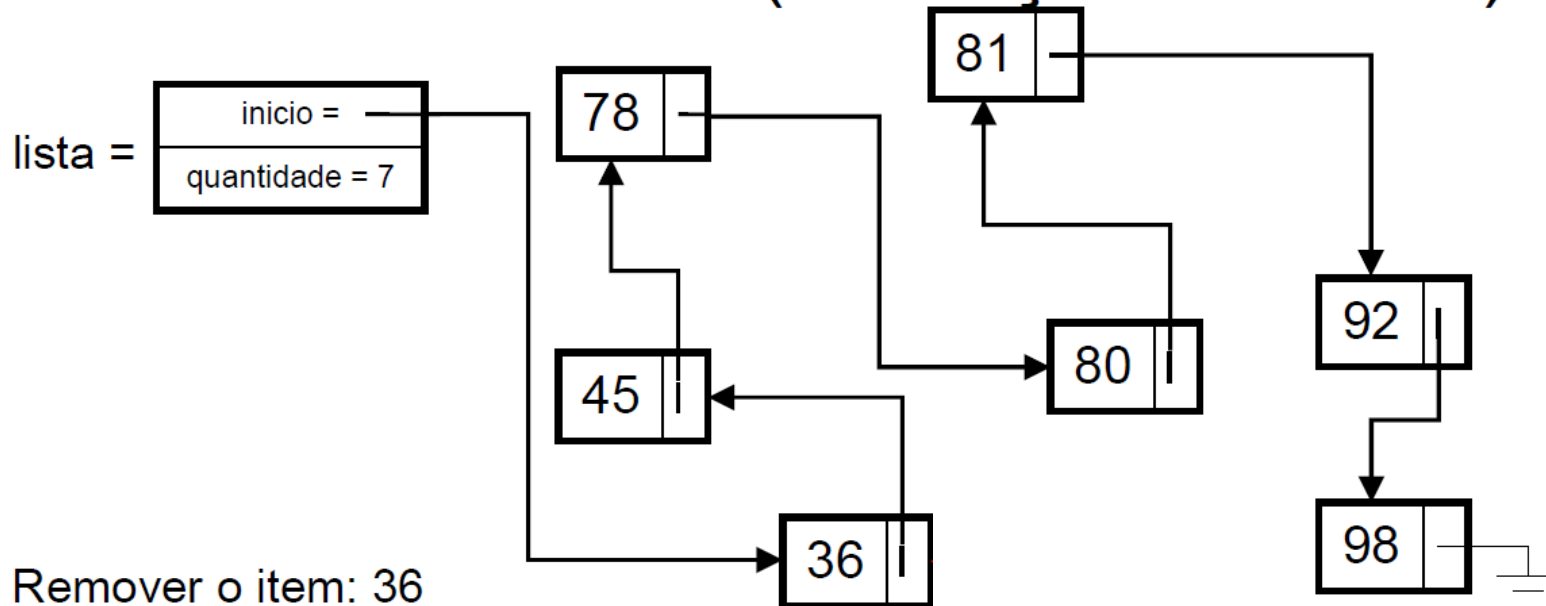
- Comece com o apontado por lista inicio.
- Exiba o conteúdo. Vá para o próximo.
- Repita até que encontre NULL

ListaDinâmica Encadeada

- Funcionamento (Remoção)
 - Observe cada situação da remoção
 - Remover do início da Lista
 - Remover do meio da Lista
 - Remover do final da Lista
 - Remover de uma lista sem elementos

Lista Dinâmica Encadeada

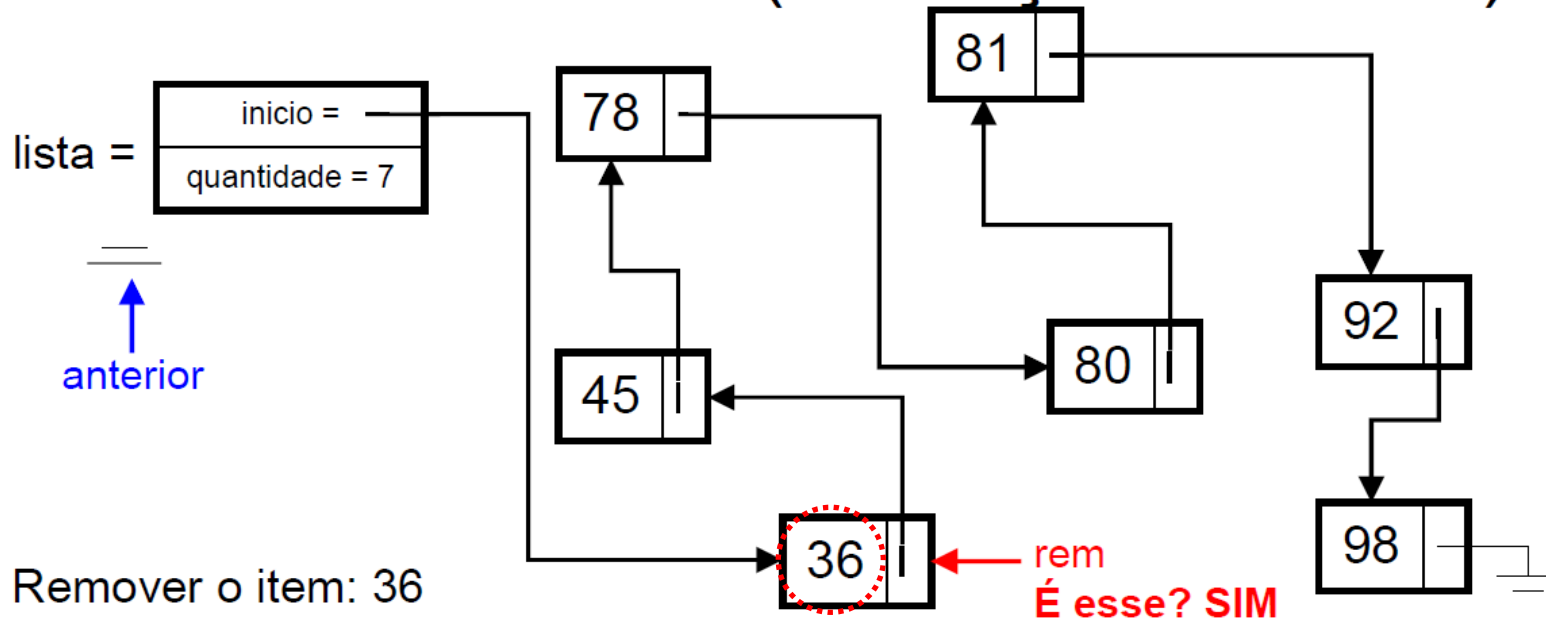
- Funcionamento (Remoção do início)



- Remover o item: 36
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for igual a NULL (excluindo o primeiro)
 - Campo início da lista recebe o campo próximo do item a ser removido.
 - Desaloca a memória do item a ser removido.
 - Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

- Funcionamento (Remoção do início)

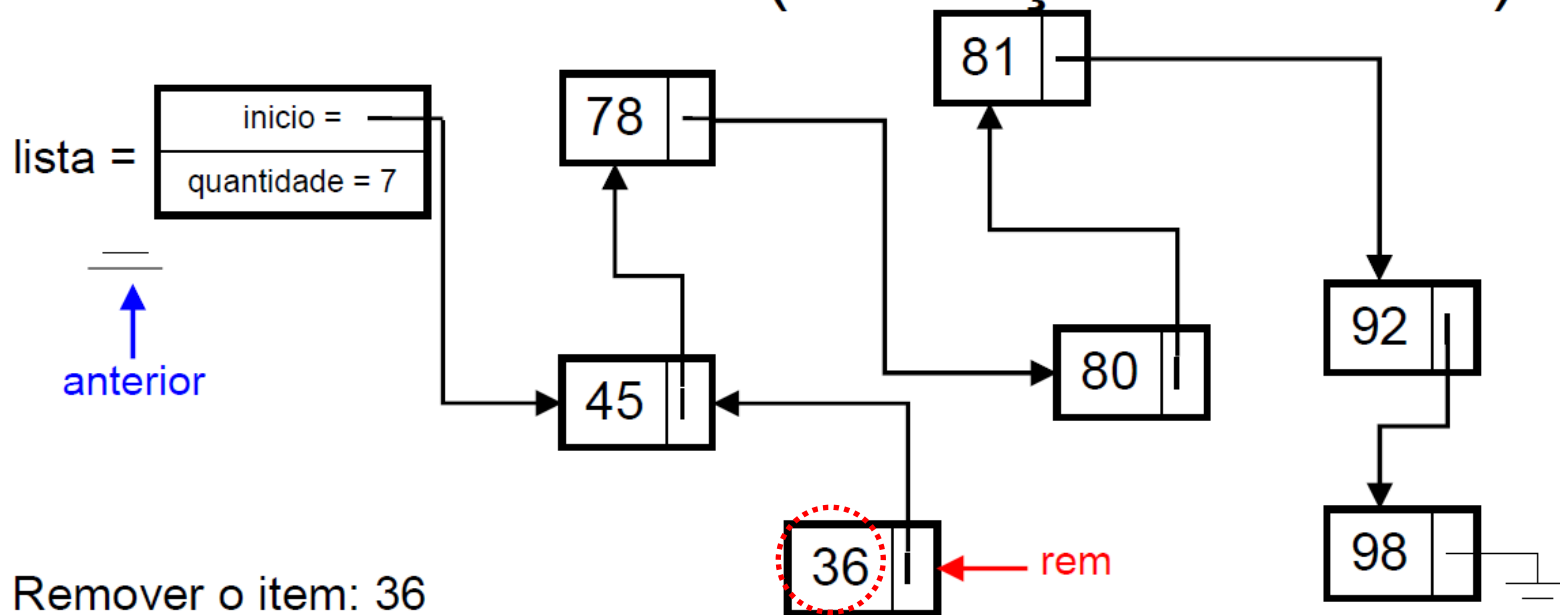


- Remover o item: 36

- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for igual a NULL (excluindo o primeiro)
 - Campo inicio da lista recebe o campo próximo do item a ser removido.
 - Desaloca a memória do item a ser removido.
 - Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

- Funcionamento (Remoção do início)

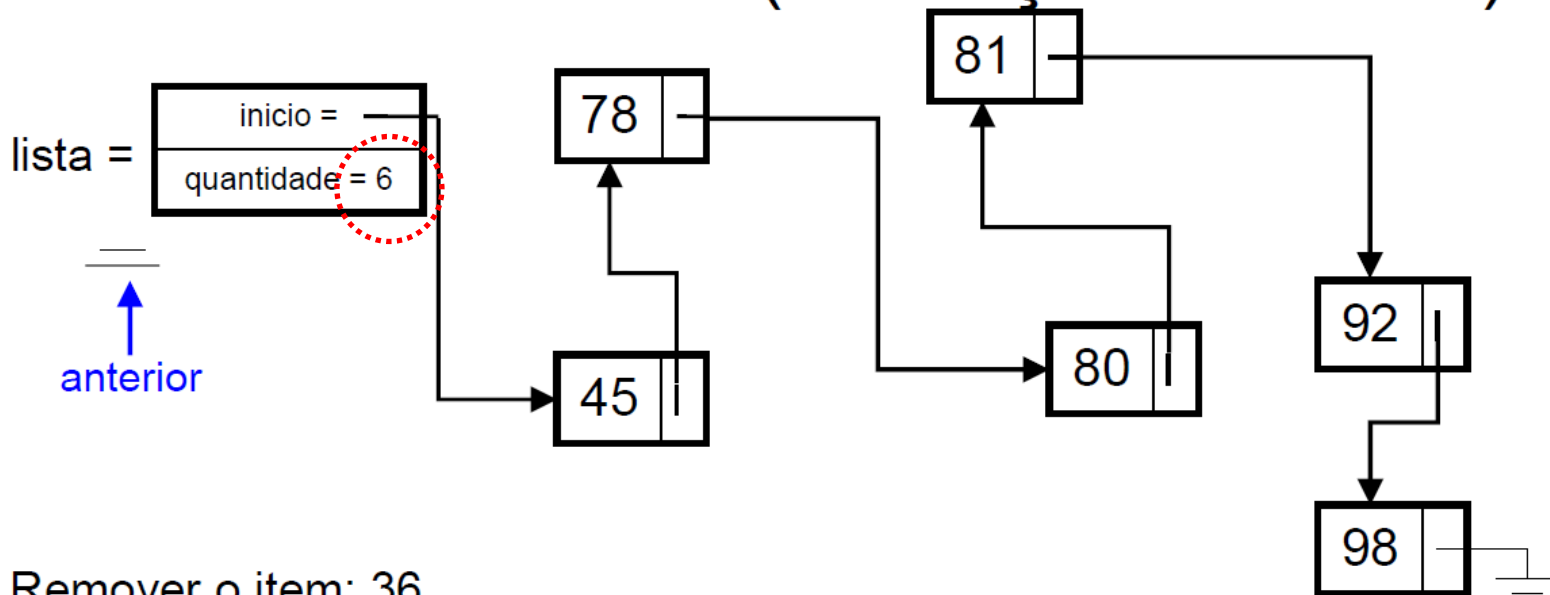


- Remover o item: 36

- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for igual a NULL (excluindo o primeiro)
 - Campo inicio da lista recebe o campo próximo do item a ser removido.
 - Desaloca a memória do item a ser removido.
 - Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

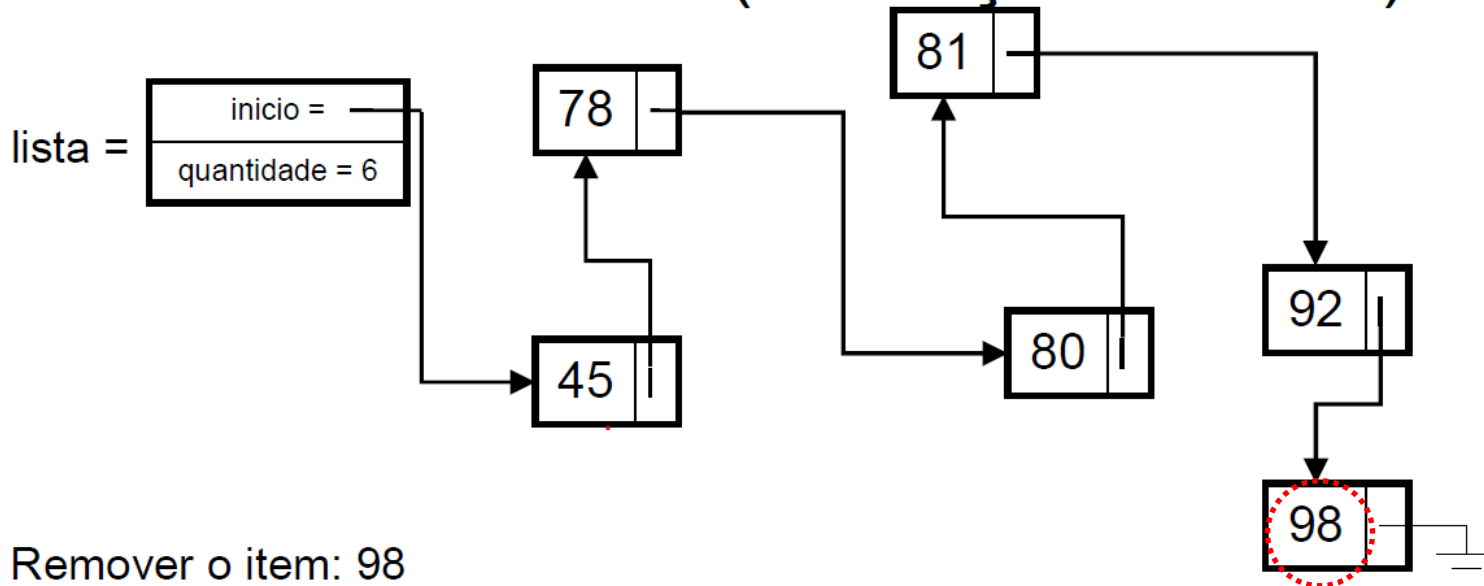
- Funcionamento (Remoção do início)



- Remover o item: 36
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for igual a NULL (excluindo o primeiro)
 - Campo inicio da lista recebe o campo próximo do item a ser removido.
- Desaloca a memória do item a ser removido.
- Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

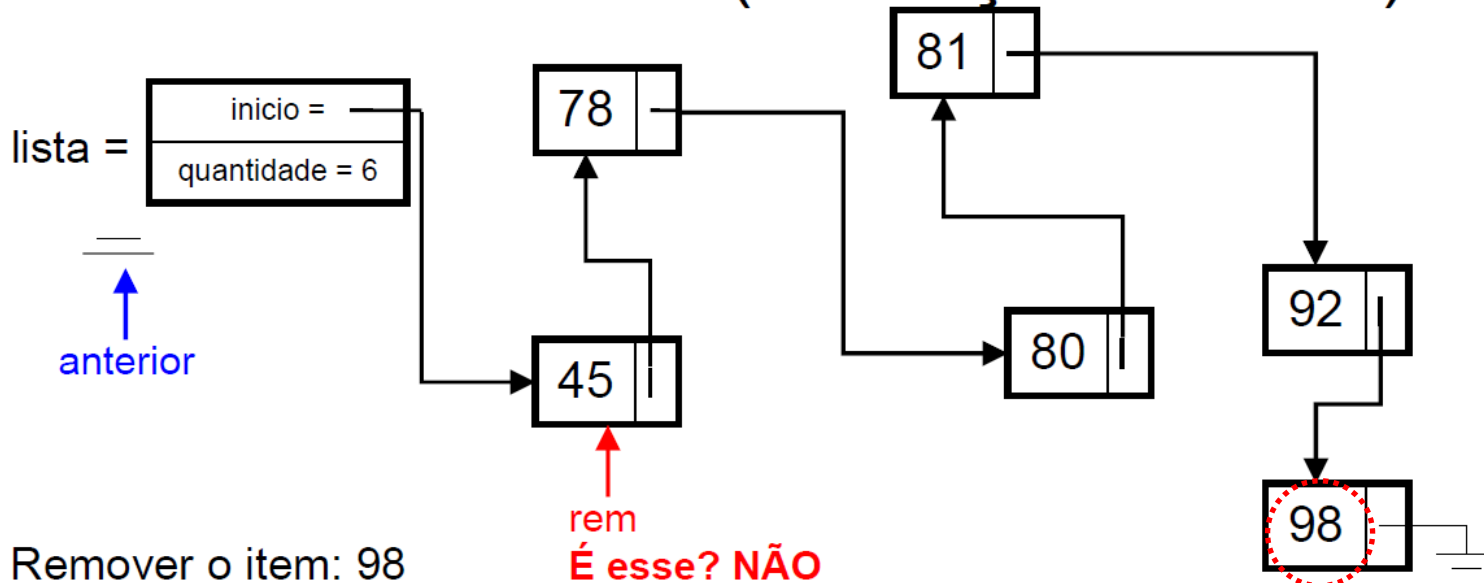
- Funcionamento (Remoção do final)



- • Remover o item: 98
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for diferente de NULL
 - Campo próximo do anterior recebe o campo próximo do item a ser removido.
 - Desaloca a memória do item a ser removido.
 - Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

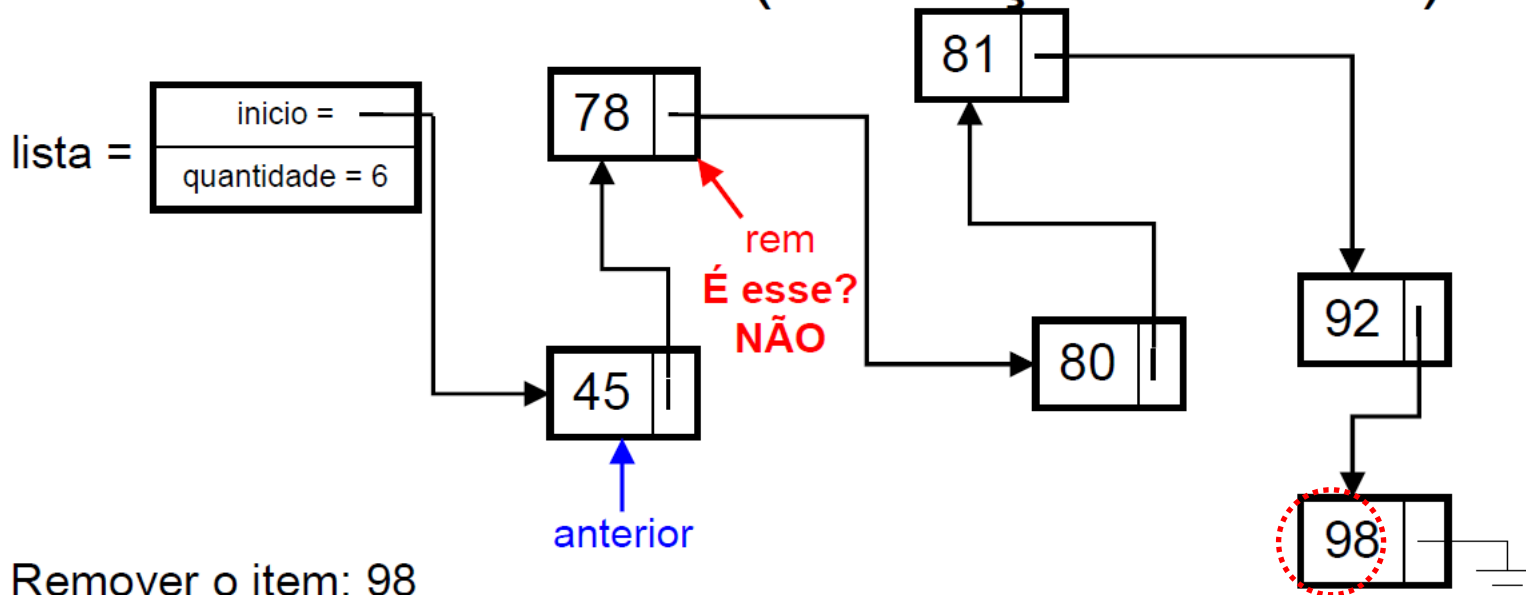
- Funcionamento (Remoção do final)



- Remover o item: 98
- • Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for diferente de NULL
 - Campo próximo do anterior recebe o campo próximo do item a ser removido.
 - Desaloca a memória do item a ser removido.
 - Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

- Funcionamento (Remoção do final)

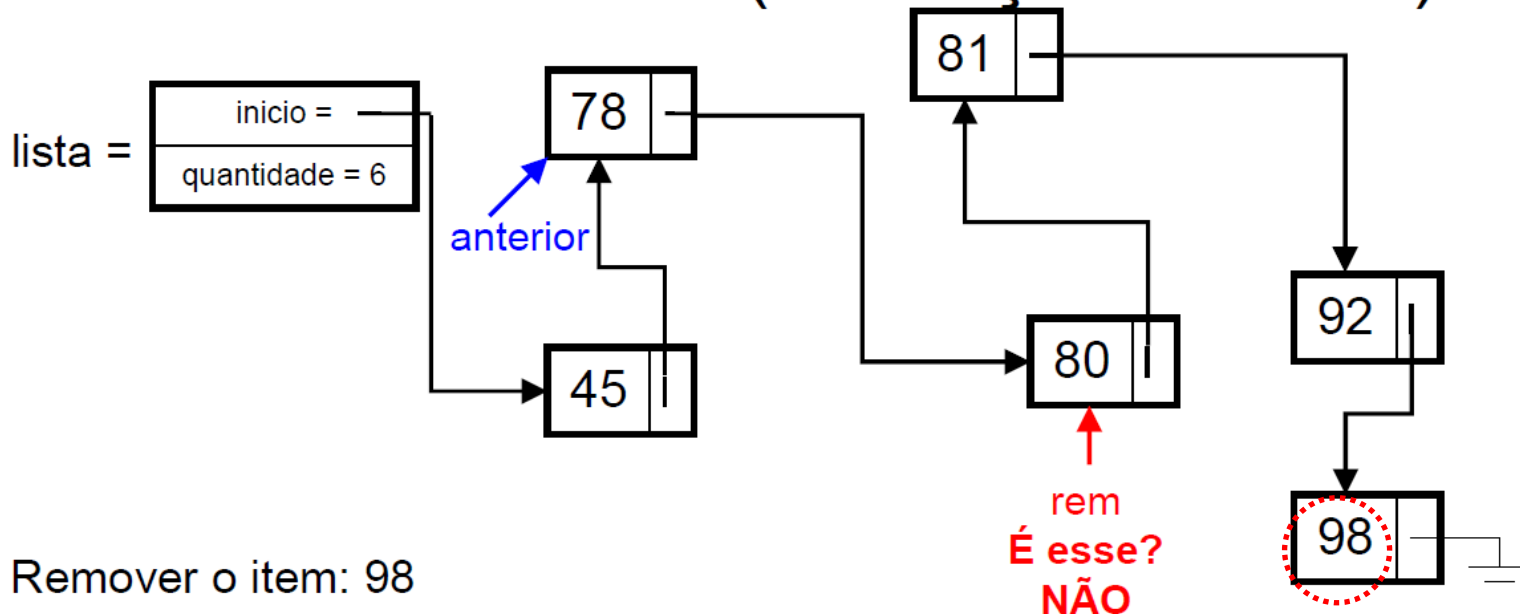


- Remover o item: 98

- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for diferente de NULL
 - Campo próximo do anterior recebe o campo próximo do item a ser removido.
 - Desaloca a memória do item a ser removido.
 - Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

- Funcionamento (Remoção do final)

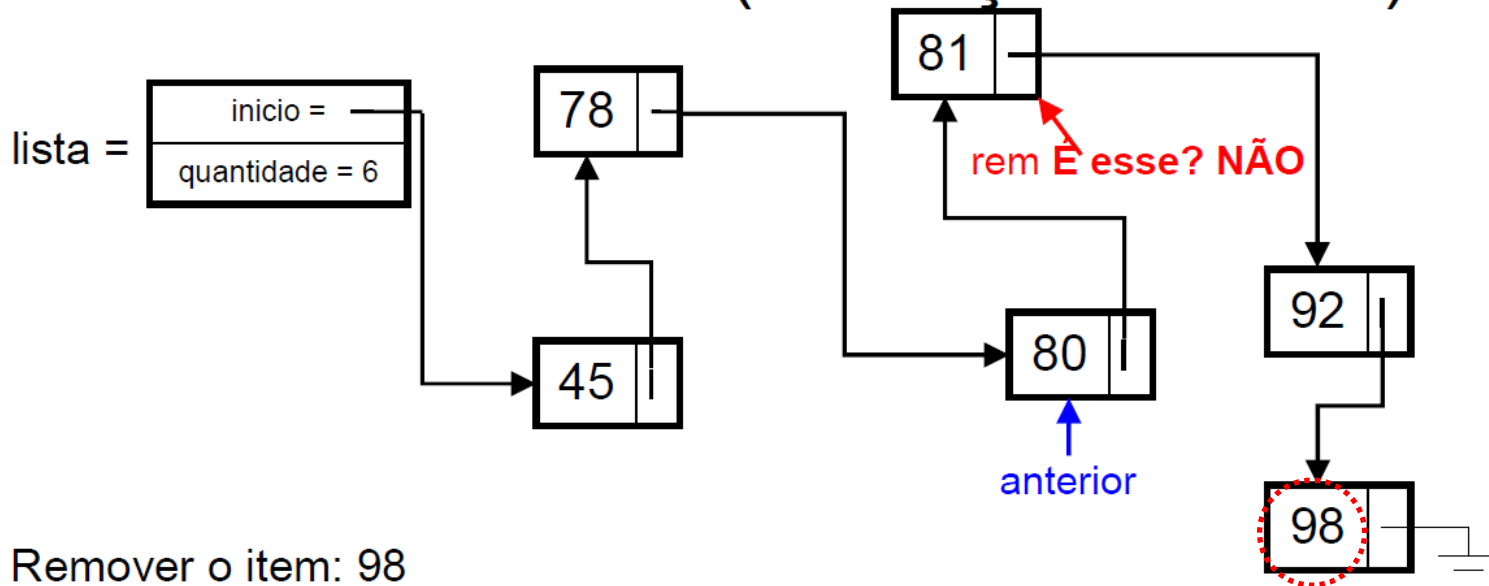


- Remover o item: 98

- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se `anterior` for diferente de NULL
 - Campo próximo do anterior recebe o campo próximo do item a ser removido.
 - Desaloca a memória do item a ser removido.
 - Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

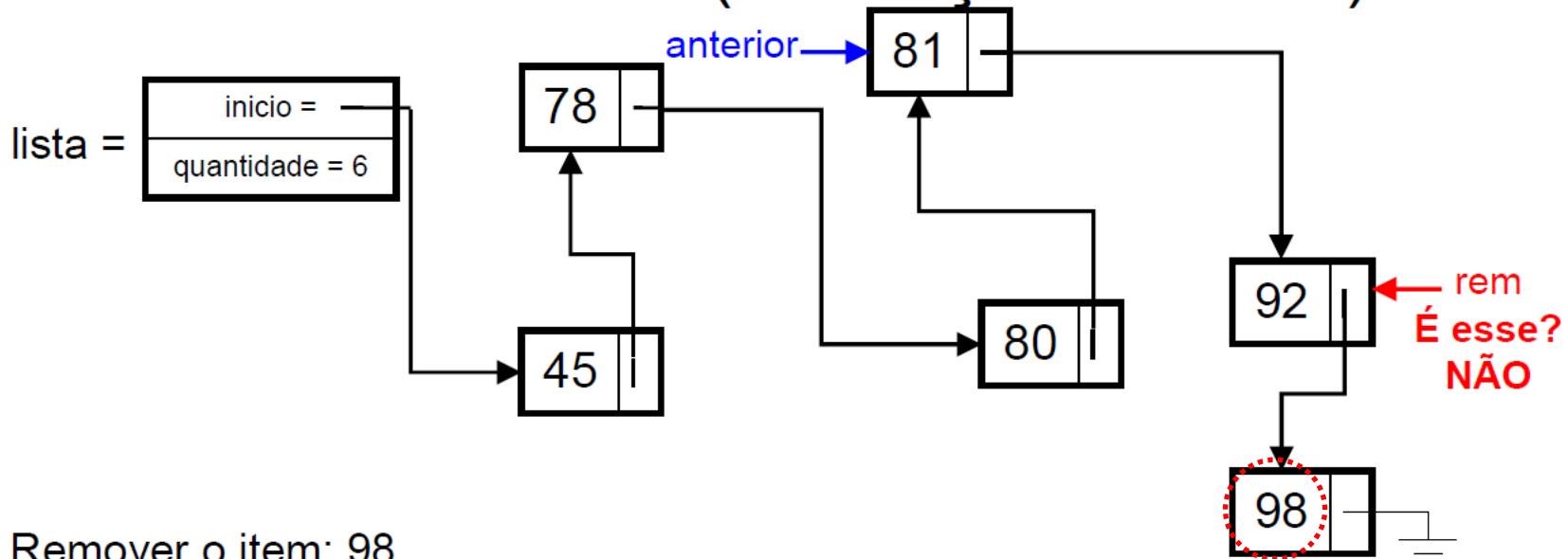
- Funcionamento (Remoção do final)



- Remover o item: 98
- • Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for diferente de NULL
 - Campo próximo do anterior recebe o campo próximo do item a ser removido.
 - Desaloca a memória do item a ser removido.
 - Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

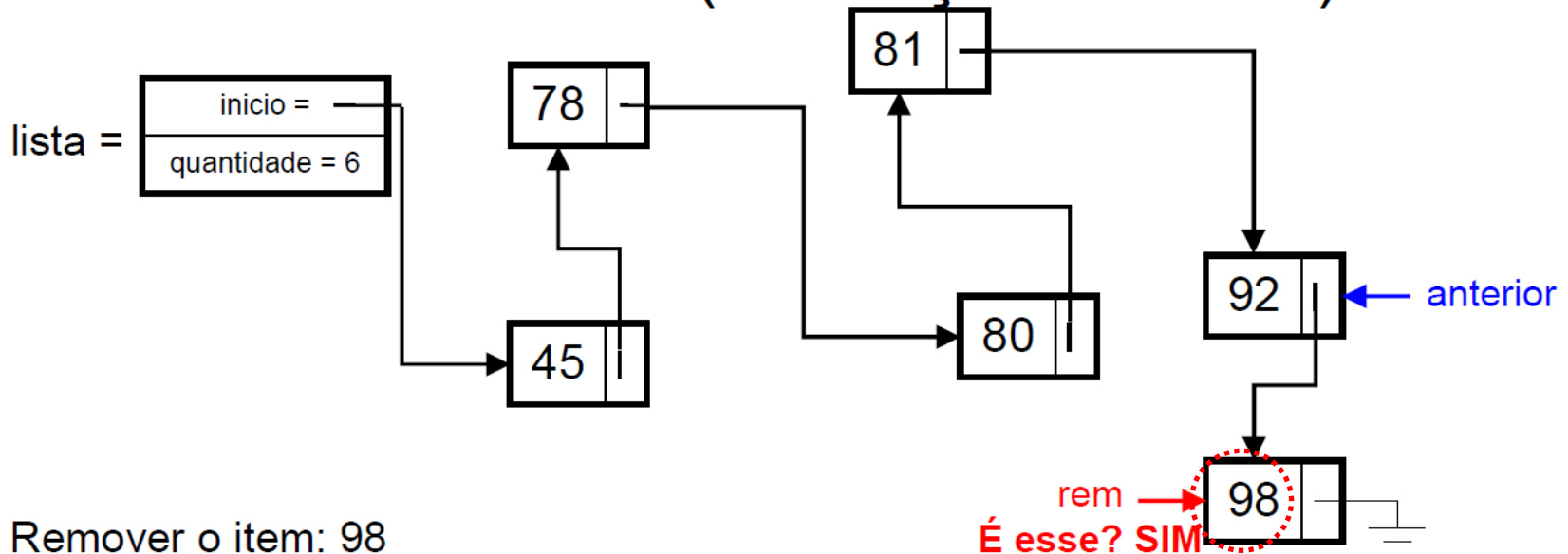
- Funcionamento (Remoção do final)



- Remover o item: 98
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for diferente de NULL
 - Campo próximo do anterior recebe o campo próximo do item a ser removido.
 - Desaloca a memória do item a ser removido.
 - Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

- Funcionamento (Remoção do final)

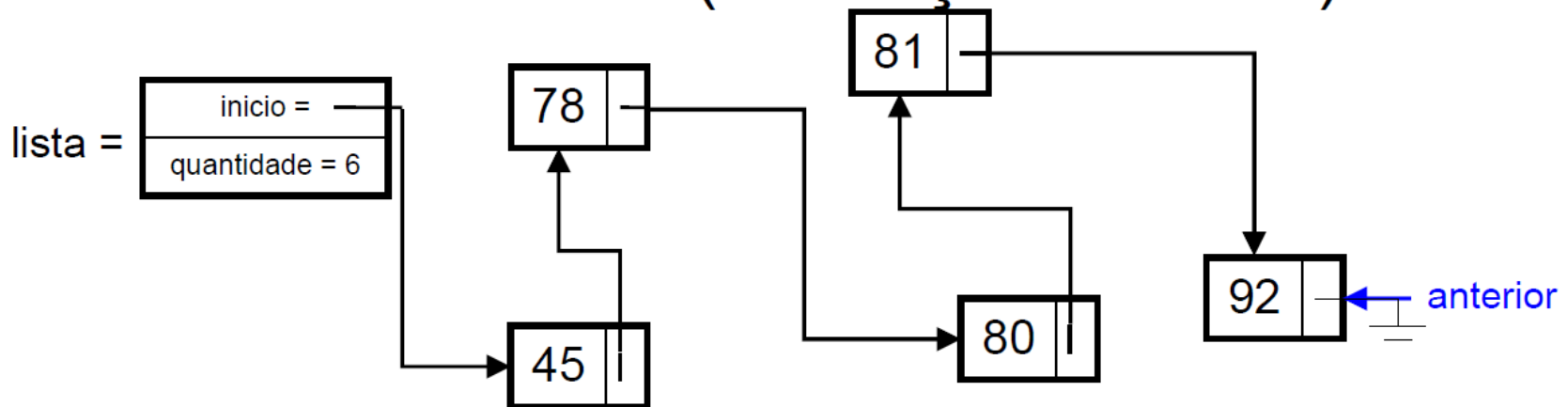


- Remover o item: 98

- • Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- • Se encontrar o item a ser removido.
 - • Se `anterior` for diferente de NULL
 - • Campo próximo do `anterior` recebe o campo próximo do item a ser removido.
 - Desaloca a memória do item a ser removido.
 - Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

- Funcionamento (Remoção do final)

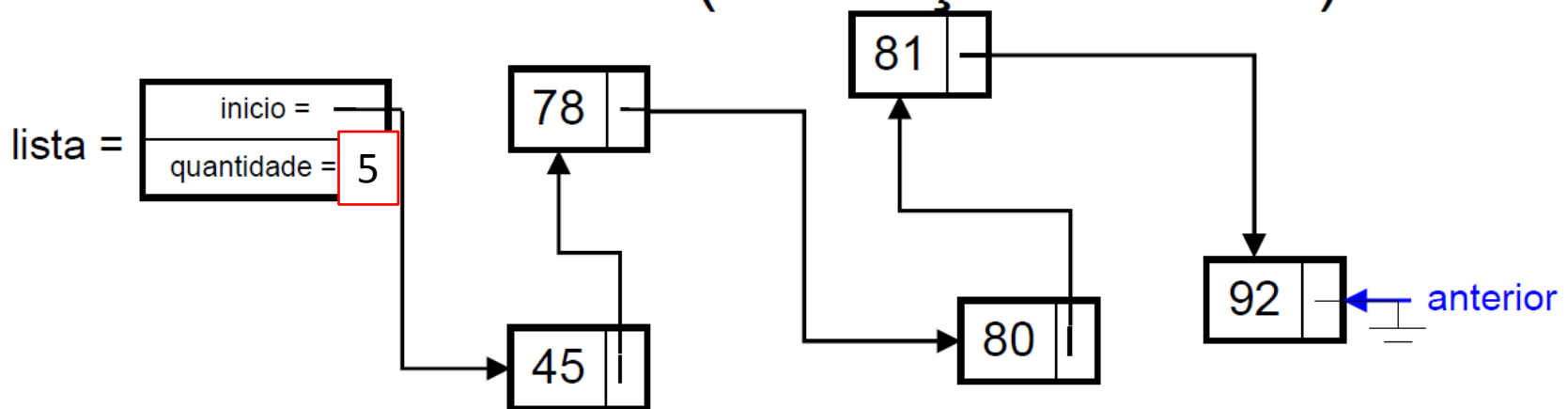


- Remover o item: 98

- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for diferente de NULL
 - Campo próximo do anterior recebe o campo próximo do item a ser removido.
- Desaloca a memória do item a ser removido.
- Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

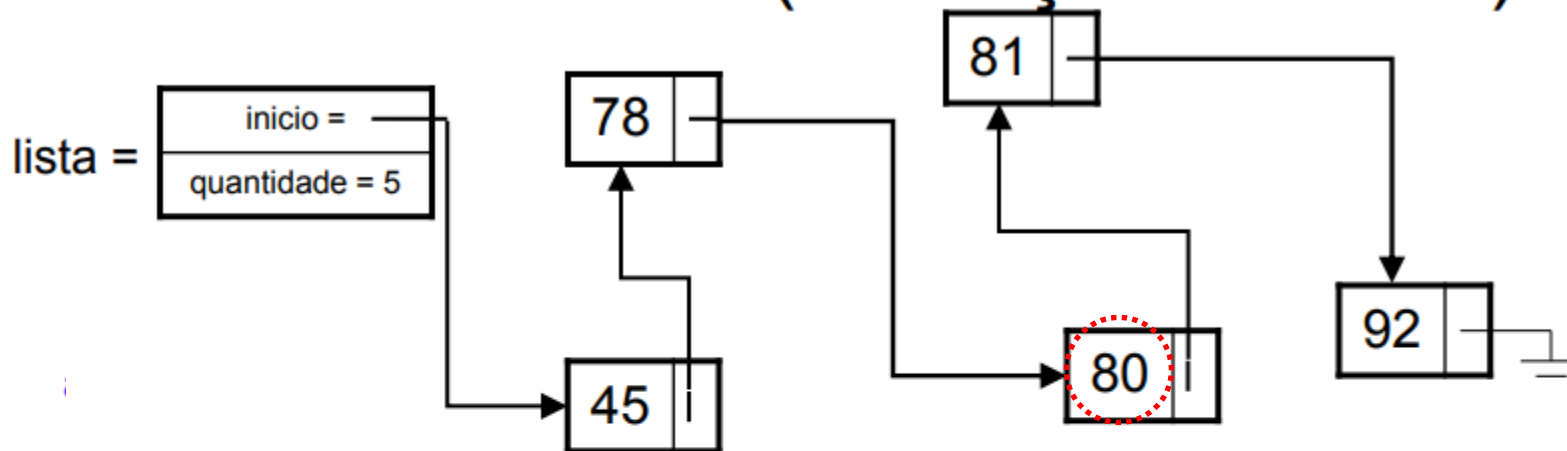
- Funcionamento (Remoção do final)



- Remover o item: 98
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for diferente de NULL
 - Campo próximo do anterior recebe o campo próximo do item a ser removido.
- Desaloca a memória do item a ser removido.
- Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

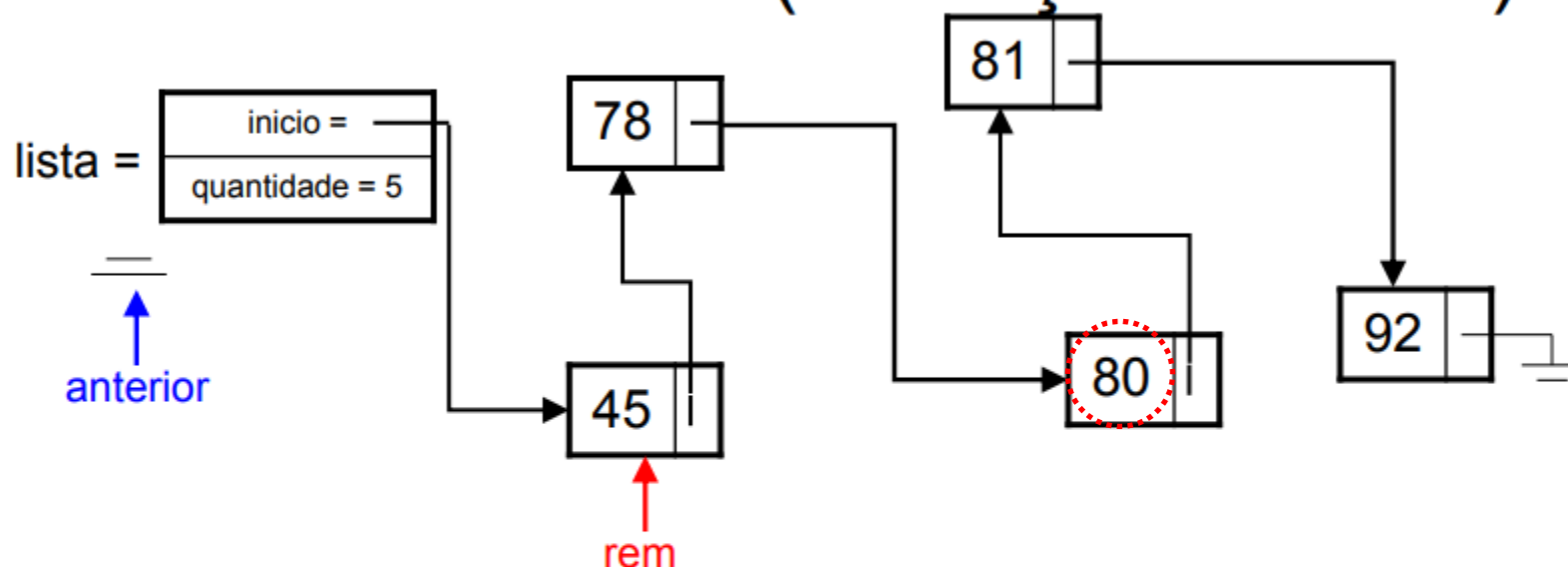
- Funcionamento (Remoção do meio)



- ➔ Remover o item: 80
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for diferente de NULL
 - Campo próximo do anterior recebe o campo próximo do item a ser removido.
 - Desaloca a memória do item a ser removido.
 - Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

- Funcionamento (Remoção do meio)



- Remover o item: 80

→ Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.

- Se encontrar o item a ser removido.

- Se anterior for diferente de NULL

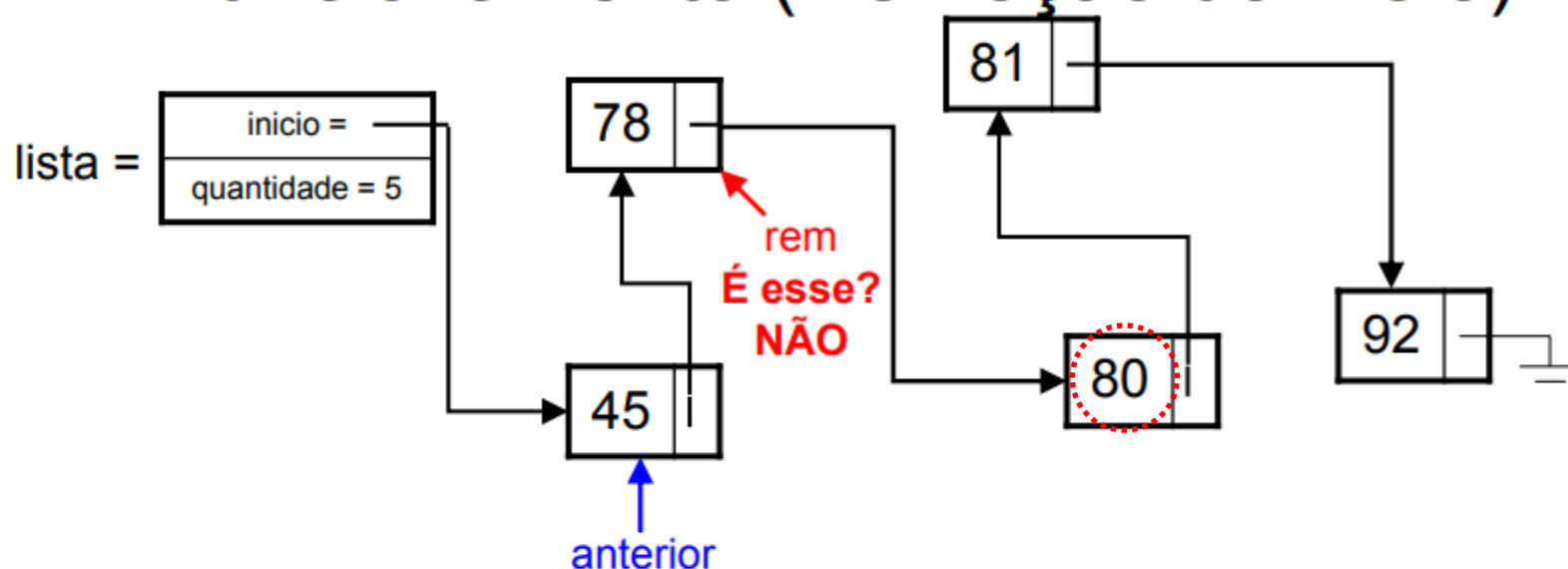
- Campo próximo do anterior recebe o campo próximo do item a ser removido.

- Desaloca a memória do item a ser removido.

- Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

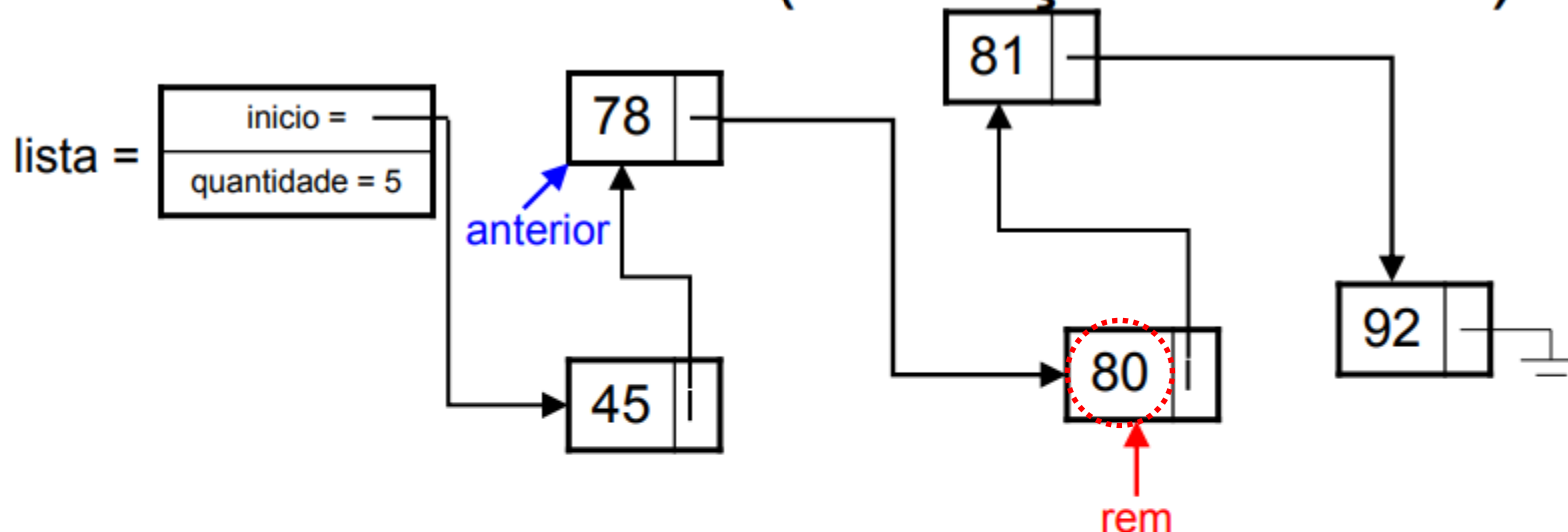
- Funcionamento (Remoção do meio)



- Remover o item: 80
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for diferente de NULL
 - Campo próximo do anterior recebe o campo próximo do item a ser removido.
 - Desaloca a memória do item a ser removido.
 - Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

- Funcionamento (Remoção do meio)

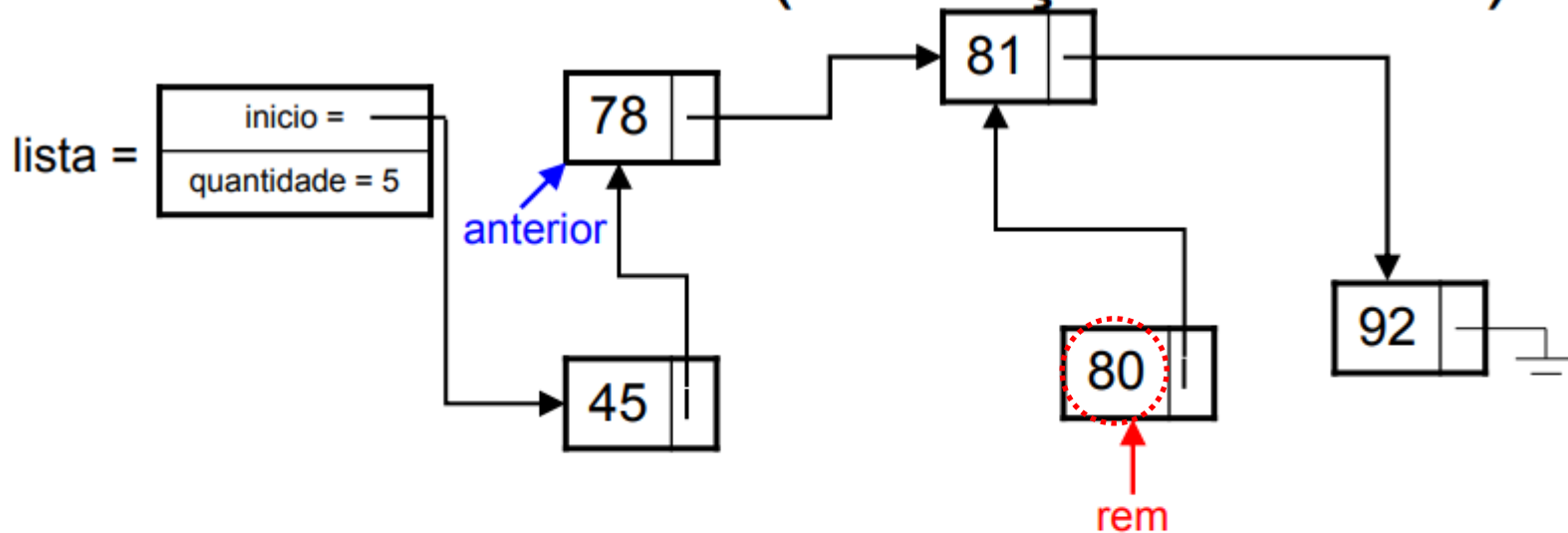


rem
É esse?
SIM

- Remover o item: 80
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for diferente de NULL
 - Campo próximo do anterior recebe o campo próximo do item a ser removido.
 - Desaloca a memória do item a ser removido.
 - Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

- Funcionamento (Remoção do meio)

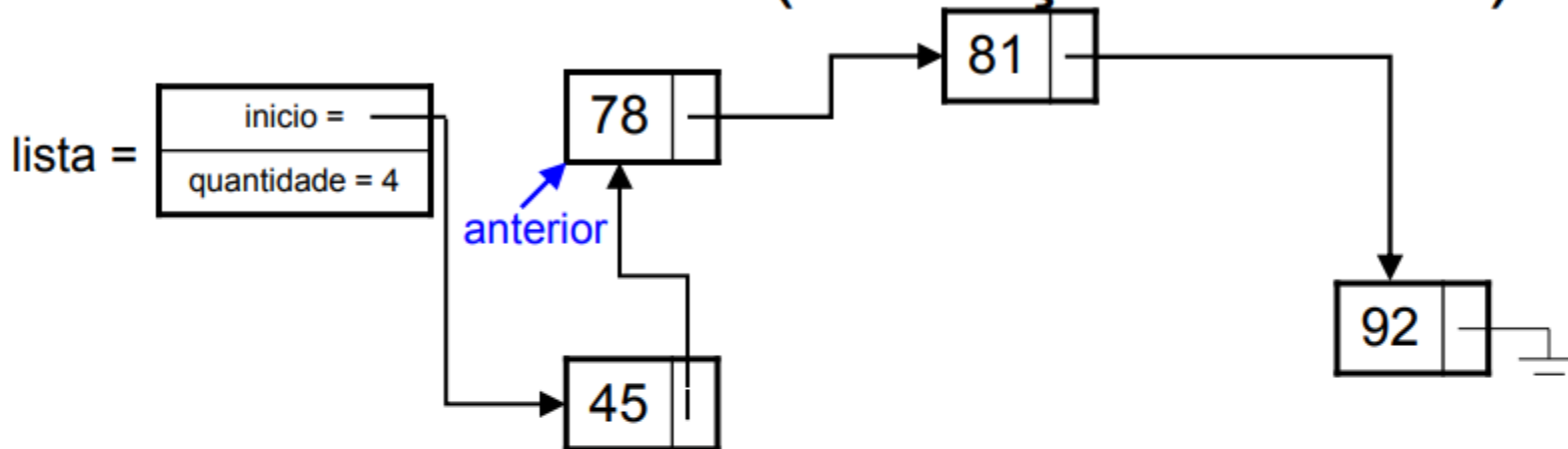


- Remover o item: 80

- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - • Se anterior for diferente de NULL
 - • Campo próximo do anterior recebe o campo próximo do item a ser removido.
 - • Desaloca a memória do item a ser removido.
 - • Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

- Funcionamento (Remoção do meio)



- Remover o item: 80
- Busque o elemento a ser removido, mantendo sempre um ponteiro para o anterior.
- Se encontrar o item a ser removido.
 - Se anterior for diferente de NULL
 - Campo próximo do anterior recebe o campo próximo do item a ser removido.
- • Desaloca a memória do item a ser removido.
- • Decrementa a quantidade em uma unidade

Lista Dinâmica Encadeada

- Estruturas:
 - tLista
 - tItem

Lista Dinâmica Encadeada

- Funções:

- `struct tLista * criaLista (void);`
- `struct tItem * criaItem (int);`
- `int busca (struct tLista *, int);`
- `void insere (struct tLista *, struct tItem *);`
- `struct tItem * remover(struct tLista *);`

ListaDinâmica Encadeada

- Implementação da Estrutura da Lista

```
1. struct tLista
2. {      int quantidade;
3.        struct tItem * inicio;
4. };
```

- Inicialização da Lista

```
1. struct tLista * criaLista(void)
2. {
3.     struct tLista * lista = (struct tLista*) malloc(sizeof(struct tLista));
4.     if (lista != NULL) {
5.         lista->inicio = NULL;    lista->quantidade = 0;
6.     }
7.     return lista;
8. }
```

Lista Dinâmica Encadeada

- Implementação da Estrutura dos itens

```
1. struct tItem
2. {
3.     int chave;
4.     struct tItem *proximo;
5. };
```

Por simplificação do exemplo nossos itens terão apenas um valor inteiro que será a chave, aqui pode-se acrescentar outros campos para a estrutura.

- Inicialização do item

```
1. struct tItem * criaItem(int ch)
2. {
3.     struct tItem * it = (struct tItem*) malloc(sizeof(struct tItem));
4.     if (it != NULL) {
5.         it->chave = ch;
6.         it->proximo = NULL;
7.     }
8.     return it;
9. }
```