Listas encadeadas
Lista de exercícios 1

Disciplina: Estrutura de Dados I
Faculdade de Sistemas de Informação
Universidade Federal do Sul e Sudeste do Pará

Enunciado geral

Considere a implementação inicial disponível em:

- https://github.com/Dreyton/ed1-linked-list
- Implemente e teste os métodos solicitados nos próximos slides.
- A submissão dos exercícios no SIGAA deve ser um arquivo compactado com os códigos implementados ou um link para um repositório público no seu github com os códigos implementados.

Q1 - Inserções

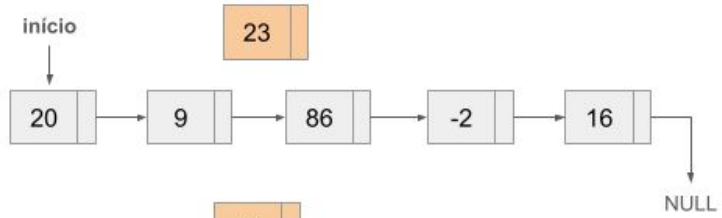
Implementar inserção em uma posição qualquer:

- Importante -> Considerar que os nós são indexados (0, 1, 2, …, n).
- Ver os dois próximos slides para entender o encadeamento necessário.
- Notar que se a inserção for no início (posição 0), já existe código para tal.
- Notar que se a inserção for no fim, já existe código para tal.

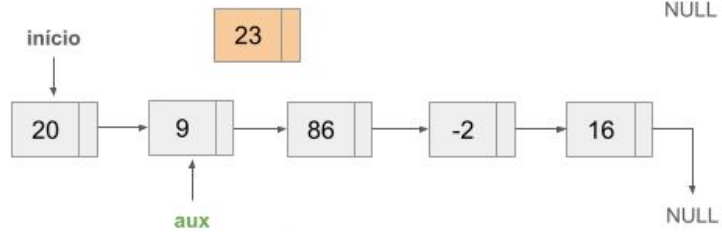
Q1 - Inserções

addAtPosition: Adicionar em qualquer posição da lista

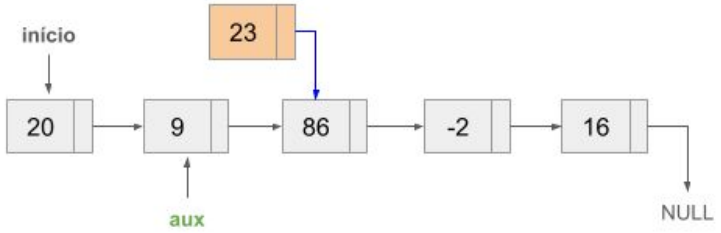
Adicionar o **elemento 23** na **posição 2**



índice aux deve apontar para o elemento que aponta para a posição desejada.



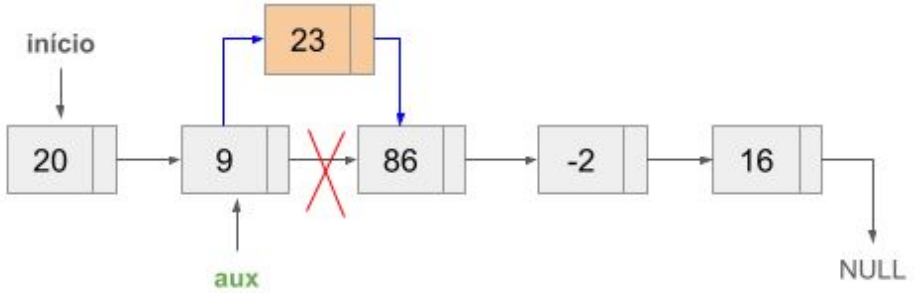
Novo elemento aponta para o próximo elemento de aux



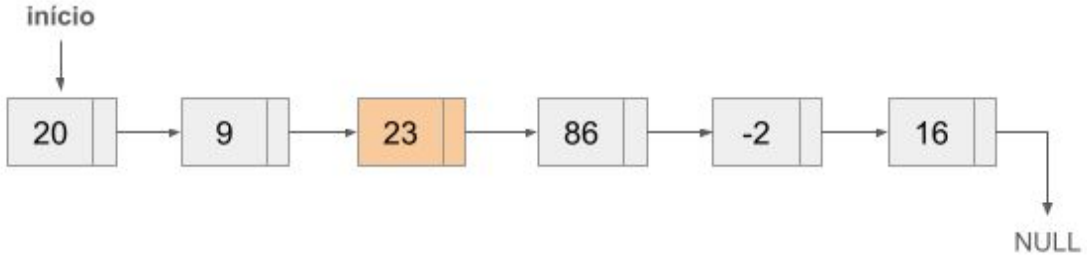
Q1 - Inserções

2/2

aux aponta para o novo elemento 23



Lista final



Q2 - Posição e existência de um elemento

Obter a posição de um elemento:

- Procura um elemento na lista encadeada e retorna sua posição. Se não existir retorna -1.

Verificar se um elemento existe na lista:

- Retorna um *true* se o elemento existir e *false* caso o elemento não esteja na lista.

Q3 - Obter um elemento em uma posição específica

Retornar o elemento na posição especificada da lista encadeada:

- Se o índice não existir, retorna *null*.
- **Dica**: Criar dois métodos, um para localizar o nó e retorná-lo, se ele existir; e outro método que a partir desse retorno também retorna o valor armazenado pelo nó.

Q4 - Remover elemento de uma posição específica

Remove o elemento em uma posição específica da lista encadeada e retorna o elemento. Se o índice não existir, retorna null:

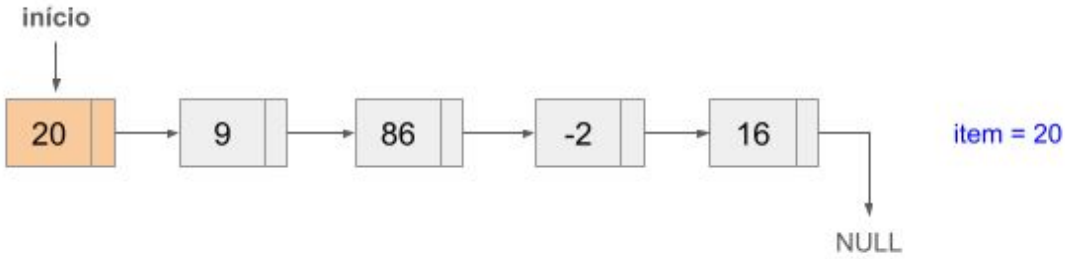
- No mesmo método (ou em métodos separados) implementar remoção no início da lista, remoção no final da lista e remoção em alguma posição no meio da lista.
- **Dica**: Utilizar as funções auxiliares já criadas anteriormente.
- Ver os três próximos slides para entender o encadeamento necessário.

Q4 - Remover elemento de uma posição específica

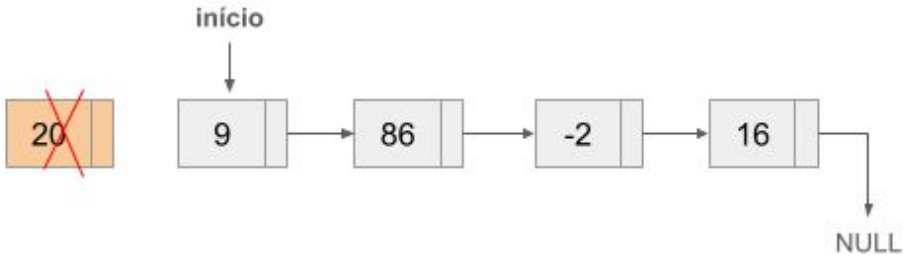
removeAtPosition: Remove em uma posição específica

1) Remover do início da lista

Cria uma variável item e obtém o valor do nó inicial da lista



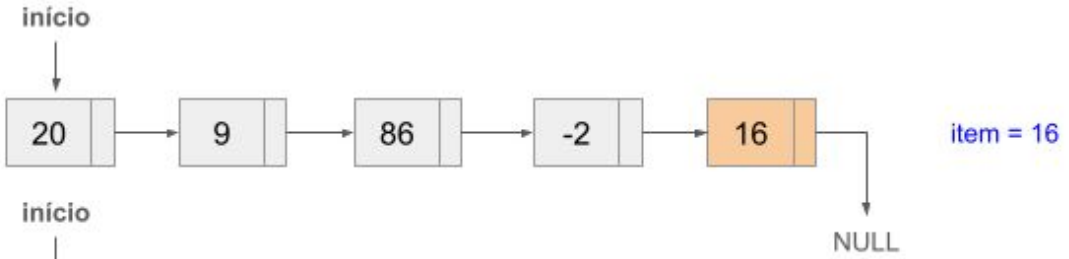
início aponta para o próximo nó da lista encadeada.



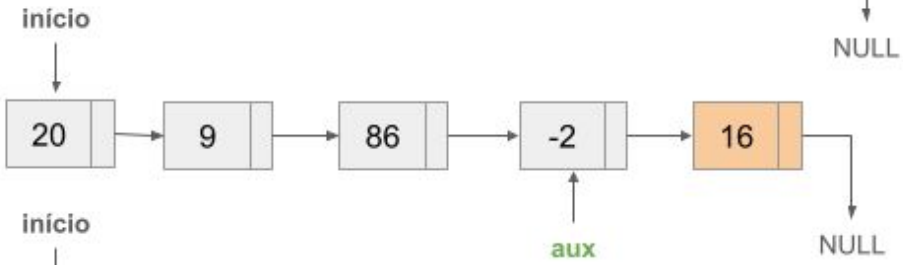
Q4 - Remover elemento de uma posição específica

2) Remover do final da lista

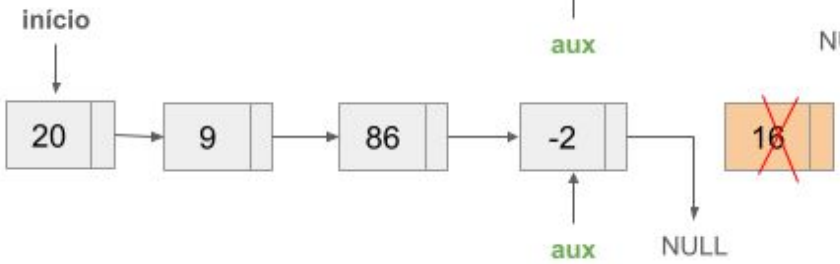
Cria uma variável item e obtém o valor do último nó da lista



Utiliza-se uma variável auxiliar que deve referenciar o elemento que aponta para o elemento a ser removido.



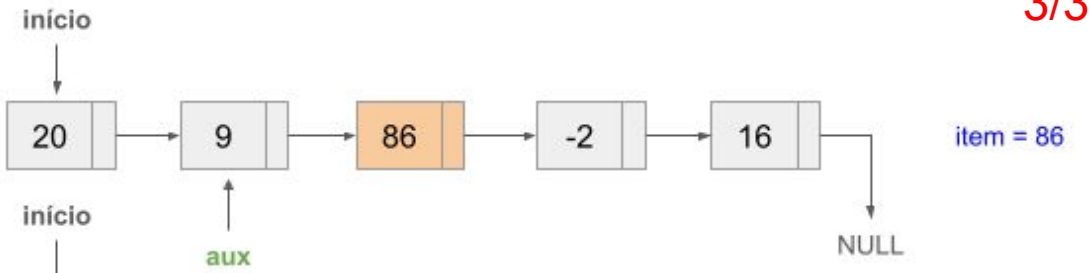
aux passa a apontar para nulo



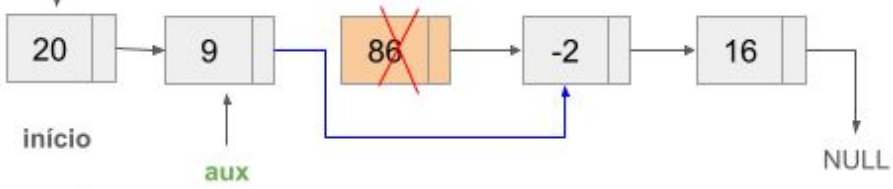
Q4 - Remover elemento de uma posição específica

3) Remover de alguma posição no meio da lista

variável aux aponta para o nó cujo próximo elemento da lista é a posição a ser removida.



Utiliza-se uma variável auxiliar que deve referenciar o elemento que aponta para o elemento a ser removido.



Lista final