



INSTITUTO FEDERAL
Sertão Pernambucano
Campus Salgueiro

Estrutura de Dados e Algoritmos com Java

Prof. Heraldo Gonçalves Lima Junior
heraldo.junior@ifsertao-pe.edu.br

1. Listas Encadeadas (continuação...)

1.1. Pegando um elemento da lista

- ◎ Para pegar um elemento é muito fácil: **basta pegarmos o nó** em que aquele elemento se encontra e **acessar o elemento de dentro dele**.
- ◎ Podemos utilizar o **pegaNo(int)** previamente criado.

1.1. Pegando um elemento da lista

```
public Object pega(int posicao) {  
    return this.pegaNo(posicao).getElemento();  
}
```

- ◎ Perceba que este método consome tempo linear. Esta é uma grande desvantagem da Lista Ligada em relação aos Vetores. Vetores possuem o chamado acesso aleatório aos elementos: qualquer posição pode ser acessada em tempo constante.

1.2. Removendo um elemento do começo

- ⦿ Antes de tentar remover **devemos verificar se a posição está ocupada.**
- ⦿ Depois, basta “**avançar**” a referência que aponta para o **primeiro** nó.
- ⦿ Por fim, é importante perceber que a Lista pode ficar vazia. Neste caso, devemos colocar **null na referência que aponta para o último nó.**

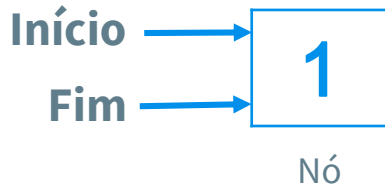
1.2. Removendo um elemento do começo

- © Se não fizermos isso ficaríamos em um estado inconsistente, em que o atributo **primeira** é **null** e o **última não**, ou seja, tem uma última mas não tem uma primeira. Isso não faria sentido.

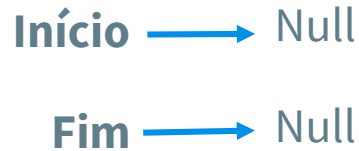
1.2. Removendo um elemento do começo

- Exemplo de remoção no início em lista **com apenas um elemento**:

ANTES DA INSERÇÃO

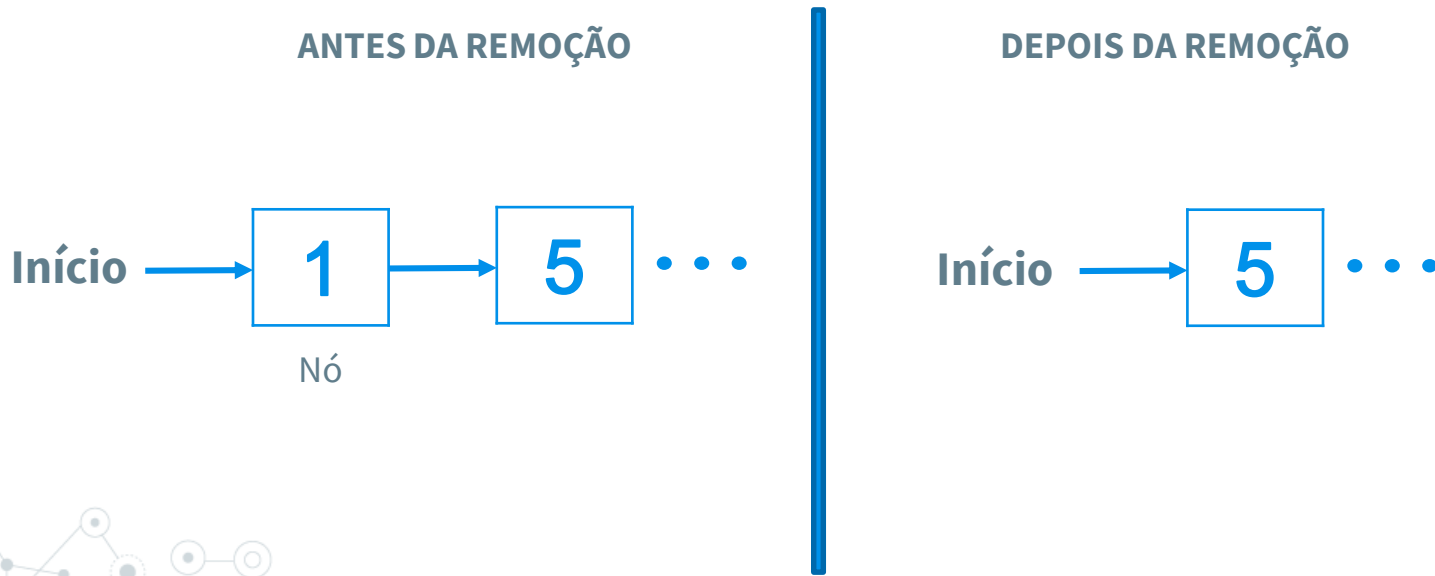


DEPOIS DA INSERÇÃO



1.2. Removendo um elemento do começo

- Exemplo de remoção no início em lista **com pelo menos 2 elementos**:



1.2. Removendo um elemento do começo

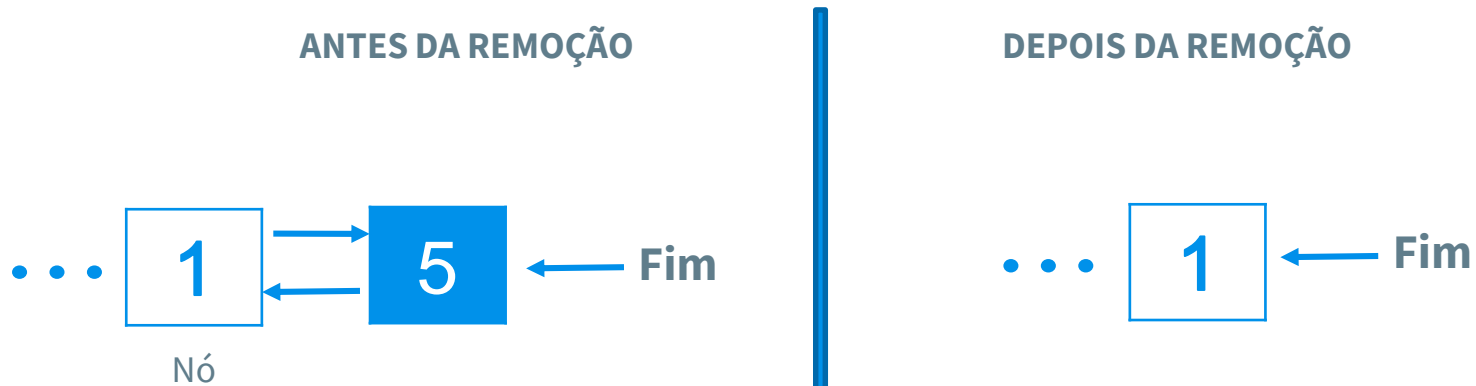
```
public void removeDoInicio() {  
    this.inicio = this.inicio.getProximo();  
    this.totalDeElementos--;  
  
    if(this.totalDeElementos==0) {  
        this.fim = null;  
    }  
}
```

1.3. Removendo um elemento do final

- ⦿ A primeira verificação a ser feita é se a última posição existe. Podemos fazer isso através do método já criado **posicaoValida(int)**.
- ⦿ Se a Lista estiver com apenas um elemento então remover do fim é a mesma coisa que remover do começo. Logo, podemos reutilizar o método **removeDoInicio()** para este caso.

1.2. Removendo um elemento do final

- Exemplo de remoção no final em lista **com pelo menos 2 elementos**:



1.3. Removendo um elemento do final

```
public void removeDoFinal() {  
    if(this.totalDeElementos==0) {  
        System.out.println("Lista vazia! Impossível remover!");  
    }else if(this.totalDeElementos==1){  
        this.removeDoInicio();  
    }else {  
        No anterior = this.pegarNo(this.totalDeElementos-2);  
        anterior.setProximo(null);  
        this.fim = anterior;  
        this.totalDeElementos--;  
    }  
}
```

1.3. Removendo um elemento do final

- ⦿ O problema aqui é como pegar o penúltimo nó. Podemos fazer isso usando o **pegaNo(int)** mas isso consumiria tempo linear. Como queremos consumo constante teremos que achar outra solução.
- ⦿ Resolveremos isso utilizando as listas duplamente encadeadas, mais a frente.

**CALMA,
RESPIRA!**



Obrigado!

Perguntas?



heraldo.junior@ifsertao-pe.edu.br



heraldolimajr.com.br