



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. Removendo de qualquer posição

- © Inicialmente, **devemos verificar se a posição está ou não ocupada**. Se não estiver devemos retornar FALSO. Caso contrário, **devemos verificar se a remoção é do começo ou do fim da Lista** se for um destes casos simplesmente chamamos os métodos que já fizemos.

1.1. Removendo de qualquer posição

- ◎ Por fim, se a remoção é no interior da Lista devemos atualizar as referências dos nós relacionados ao nó que vamos remover (anterior e próximo).
- ◎ O próximo do anterior deve ser o próximo e o anterior do próximo deve ser o anterior.

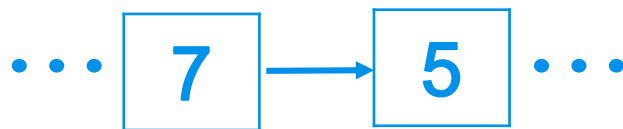
1.1. Removendo de qualquer posição

- Exemplo de remoção no meio da lista **com pelo menos 3 elementos**:

ANTES DA REMOÇÃO



DEPOIS DA REMOÇÃO



1.1. Removendo de qualquer posição

```
public boolean remove(int posicao) {  
    if(this.posicaoValida(posicao)) {  
        if(posicao==0) {  
            this.removeDoInicio();  
            return true;  
        }else if(posicao==this.totalDeElementos-1) {  
            this.removeDoFinal();  
            return true;  
        }else {  
            No anterior = this.pegarNo(posicao-1);  
            No atual = anterior.getProximo();  
            anterior.setProximo(atual.getProximo());  
            this.totalDeElementos--;  
            return true;  
        }  
    }else {  
        return false;  
    }  
}
```

1.2. Verificando se um elemento está na Lista

- © Esta operação deve percorrer a Lista e comparar com o método **equals(Object)** o elemento procurado contra todos os elementos da Lista.

1.2. Verificando se um elemento está na Lista

```
public boolean contem(Object elemento) {  
    No atual = this.inicio;  
    for(int i=0; i < this.totalDeElementos; i++) {  
        if(atual.getElemento().equals(elemento)) return true;  
        atual = atual.getProximo();  
    }  
    return false;  
}
```


1.3. Verificando o tamanho da lista

- © Esta operação não tem segredo, pois já temos um atributo que possui esta informação.

```
public int tamanho() {  
    return this.totalDeElementos;  
}
```

2. Listas Duplamente Encadeadas

1. Listas duplamente encadeadas

- © As listas duplamente encadeadas podem ser usadas quando várias operações de inserção e remoção de elementos são necessárias. Elas são estruturas de dados semelhantes às listas simplesmente encadeadas e sua alocação da memória é feita durante a execução. No entanto, em comparação com as listas simplesmente encadeadas, a conexão entre os elementos é feita através de dois ponteiros (**um que aponta para o elemento anterior e o outro para o seguinte**).

1.1 Adicionando no começo da Lista

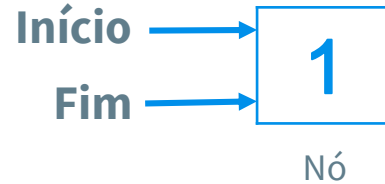
◎ Exemplo de inserção em lista **vazia**:

ANTES DA INSERÇÃO

Início = Null

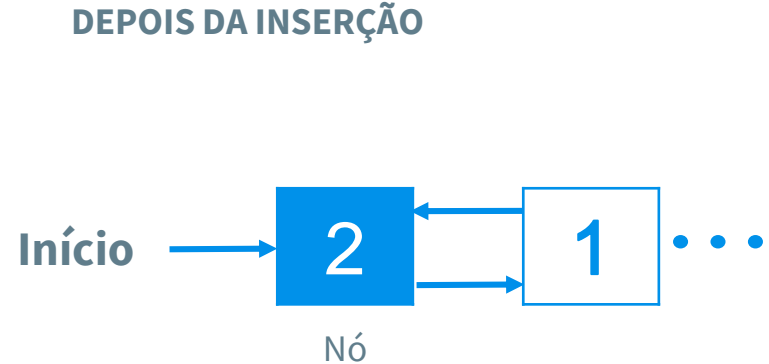
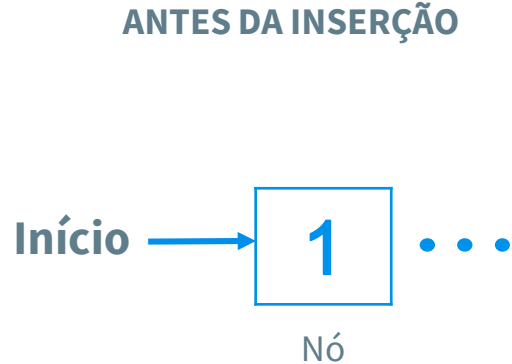
Fim = Null

DEPOIS DA INSERÇÃO



1.1 Adicionando no começo da Lista

☉ Exemplo de inserção em lista **não vazia**:



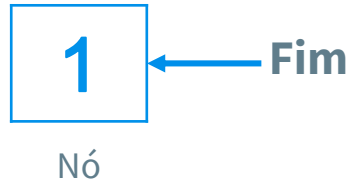
1.1 Adicionando no começo da Lista

```
public void adicionaNoComeco(Object elemento) {  
    if (this.totalDeElementos == 0) {  
        No novo = new No(elemento);  
        this.inicio = novo;  
        this.fim = novo;  
    }else {  
        No novo = new No(elemento, this.inicio);  
        this.inicio.setAnterior(novo);  
        this.inicio=novo;  
    }  
    this.totalDeElementos++;  
}
```

1.1 Adicionando no final da Lista

☉ Exemplo de inserção em lista **não vazia**:

ANTES DA INSERÇÃO



DEPOIS DA INSERÇÃO



1.1 Adicionando no final da Lista

```
public void adicionaNoFinal(Object elemento) {  
    if(this.totalDeElementos==0) {  
        this.adicionaNoInicio(elemento);  
    }else {  
        No novo = new No(elemento);  
        this.fim.setProximo(novo);  
        novo.setAnterior(this.fim);  
        this.fim = novo;  
        this.totalDeElementos++;  
    }  
}
```


**CALMA,
RESPIRA!**



Obrigado!

Perguntas?



heraldo.junior@ifsertao-pe.edu.br



heraldolimajr.com.br