



# Entendendo as Aplicações da Lista Ligada

**S**erão apresentadas a seguir as operações de Lista Ligada para inserir no meio, remover e buscar, bem como as implementações na linguagem de programação Java. Para as operações em pseudocódigo, utilizaremos as seguintes definições de registro:

```
// definição do tipo registro No com os campos abaixo
```

```
tipo No = registro
```

```
    elemento ← 0 numérico_inteiro; // campo que armazena o elemento
```

```
    prox ← nulo No; // campo que armazena o endereço do próximo nó
```

```
fimregistro;
```

Onde o campo “elemento” armazena o elemento e o campo “prox” é o ponteiro que armazena o endereço do próximo nó que contém o elemento seguinte da lista ligada.

```
// definição do tipo registro ListaLigada com os campos abaixo
```

```
tipo ListaLigada = registro // o tipo registro chama-se ListaLigada
```

primeiro  $\leftarrow$  nulo No; // campo primeiro que armazena o primeiro elemento da  
lista



ultimo  $\leftarrow$  nulo No; // campo ultimo que armazena o último elemento da lista  
ligada

fimregistro;

Onde o campo “primeiro” é um ponteiro que armazena o endereço do primeiro elemento da lista ligada e o campo “último” é um ponteiro que armazena o endereço do último elemento da lista.

## INSERIR DA LISTA LIGADA

“InserirMeio” é um módulo-procedimento da operação Inserir no Meio que recebe como parâmetro um elemento do tipo **No** a ser inserido e a posição onde ele deve estar e o insere no meio da lista ligada.

InserirMeio(NovoNo  $\uparrow$  **No**, posicao **numérico\_inteiro**)

**início\_módulo**

**Declarar**

NoTemp  $\leftarrow$  primeiro **No**;

NroNos, posAux  $\leftarrow$  1 **numérico\_inteiro**;



NroNos  $\leftarrow$  ContarNos( );

**se** (posicao  $\leq$  1)

**então**

InserirInicio(NovoNo);

**senão**

**se** (posicao  $>$  NroNos)

**então**

InserirFinal(NovoNo);

**senão**

**enquanto** (posAux  $>$  (posicao  $-1$ ))

NoTemp  $\leftarrow$  NoTemp.prox;

posAux  $\leftarrow$  posAux + 1;

**fimenquanto;**

NovoNo.prox  $\leftarrow$  NoTemp.prox;

NoTemp.prox  $\leftarrow$  NovoNo;

**fimse;**

**fimse;**

**fim\_módulo;**



Em Java, o método InserirMeio.

```
public void InserirMeio(No NovoNo, int posicao){
    No NoTemp = primeiro;
    int NroNos, posAux = 1;

    NroNos = ContarNos( );
    if (posicao <= 1){
        InserirInicio(NovoNo);
    }
    else{
        if (posicao > NroNos){
            InserirFinal(NovoNo);
        }
        else{
            while (posAux < (posicao - 1)){
                NoTemp = NoTemp.prox;
                posAux = posAux + 1;
            }
            NovoNo.prox = NoTemp.prox;
            NoTemp.prox = NovoNo;
        }
    }
}
```

## REMOVER DA LISTA LIGADA

“Remover” é um módulo-procedimento que recebe o elemento a ser removido da operação e, se o elemento não existir ou a lista ligada estiver vazia, ele não fará nada. Caso contrário, removerá a primeira ocorrência desse elemento da lista ligada.

Remover (elemento **numérico\_inteiro**)



**início\_módulo**

**Declarar**

NoTemp  $\leftarrow$  primeiro **No**;

NoAnt  $\leftarrow$  **nulo** No;

**se** (primeiro.elemento = elemento)

**então**

primeiro  $\leftarrow$  primeiro.prox;

**senão**

**enquanto** (NoTemp  $\neq$  **nulo** e NoTemp.elemento  $\neq$  elemento)

NoAnt  $\leftarrow$  NoTemp;

NoTemp  $\leftarrow$  NoTemp.prox;

**fimenquanto;**

**se** (NoTemp  $\neq$  **nulo**)

**então**

NoAnt.prox  $\leftarrow$  NoTemp.prox;

**fimse;**

**se** (NoTemp = ultimo)

**então**



ultimo  $\leftarrow$  NoAnt;

**fimse;**

**fimse;**

**fim\_módulo;**

O método remover no Java.

```
public void Remover (int elemento) {  
    No NoTemp = primeiro;  
    No NoAnt = null;  
  
    if (primeiro.elemento == elemento) {  
        primeiro = primeiro.prox;  
    }  
    else {  
        while (NoTemp != null && NoTemp.elemento != elemento) {  
            NoAnt = NoTemp;  
            NoTemp = NoTemp.prox;  
        }  
        if (NoTemp != null) {  
            NoAnt.prox = NoTemp.prox;  
        }  
        if (NoTemp == ultimo) {  
            ultimo = NoAnt;  
        }  
    }  
}
```

**BUSCAR DA LISTA LIGADA**



“BuscarNo” é um módulo-função que recebe o elemento a ser pesquisado da operação Buscar e retorna o nó cujo elemento foi pesquisado.

**No** BuscarNo (elemento **numérico\_inteiro**)

**início\_módulo**

**Declarar**

$i \leftarrow 1$  **numérico\_inteiro**;

NoTemp  $\leftarrow$  primeiro **No**;

**enquanto** (NoTemp  $\neq$  **nulo**) **faça**

**se** (NoTemp.elemento = elemento)

**então**

escrever ("No ", NoTemp.elemento , " posição " , i);

**retornar** NoTemp;

**fimse**;

$i \leftarrow i + 1$ ;

NoTemp  $\leftarrow$  NoTemp.prox;

**fimenquanto**;

retornar nulo;



fim\_módulo;

Método BuscarNo no Java.

```
public No BuscarNo (int elemento)
{
    int i = 1;
    No NoTemp = primeiro;

    while (NoTemp != null)
    {
        if (NoTemp.elemento == elemento)
        {
            System.out.println("No " + NoTemp.elemento + " posição "
                               + i);
            return NoTemp;
        }
        i = i + 1;
        NoTemp = NoTemp.prox;
    }
    return null;
}
```

## APLICANDO LISTAS LIGADAS

Vamos desenvolver parte de um programa em Java que recebe do usuário dez números inteiros numa lista ligada, inserindo apenas os divisíveis por três e mostrando esses números.

### Algoritmo No

início\_algoritmo



**tipo** No = **registro**



elemento  $\leftarrow$  0 **numérico\_inteiro**;

**prox**  $\leftarrow$  nulo No;

**fimregistro**;

**fimalgoritmo**.

## Algoritmo ListaLigada

**início\_algoritmo**

// definição do tipo registro ListaLigada com os campos abaixo

**tipo** ListaLigada = **registro** // o tipo registro chama-se ListaLigada

primeiro  $\leftarrow$  nulo **No**;

ultimo  $\leftarrow$  nulo **No**;

**fimregistro**;

**lógico** ListaVazia( )

**início\_módulo**

**se** (primeiro = **nulo** e ultimo = **nulo**)

**então**

**retornar** verdadeiro;

**senão**



**retornar** falso;

**fimse;**

**fim\_módulo;**

InserirInicio (novoNo  $\uparrow$  **No**)

**início\_módulo**

**se** (ListaVazia())

**então**

ultimo  $\leftarrow$  novoNo;

**senão**

novoNo.prox  $\leftarrow$  primeiro;

**fimse;**

primeiro  $\leftarrow$  novoNo;

**fim\_módulo;**

InserirFinal (novoNo  $\uparrow$  **No**)

**início\_módulo**

**se** (ListaVazia())

**então**



primeiro  $\leftarrow$  novoNo;

**senão**

ultimo.prox  $\leftarrow$  novoNo;

**fimse;**

ultimo  $\leftarrow$  novoNo;

**fim\_módulo;**

**numérico\_inteiro** ContarNos()

**início\_módulo**

**Declarar**

tamanho  $\leftarrow$  0 **numérico\_inteiro;**

NoTemp  $\leftarrow$  primeiro **No;**

**enquanto** (NoTemp  $\neq$  nulo) **faça**

tamanho  $\leftarrow$  tamanho + 1;

NoTemp  $\leftarrow$  NoTemp.prox;

**fimenquanto;**

**retornar** tamanho;

**fim\_módulo;**



InserirMeio(NovoNo  $\uparrow$  **No**, posicao **numérico\_inteiro**)

**início\_módulo**

**Declarar**

NoTemp  $\leftarrow$  primeiro **No**;

NroNos, posAux  $\leftarrow$  1 **numérico\_inteiro**;

NroNos  $\leftarrow$  ContarNos( );

**se** (posicao  $\leq$  1)

**então**

InserirInicio(NovoNo);

**senão**

**se** (posicao  $>$  NroNos)

**então**

InserirFinal(NovoNo);

**senão**

**enquanto** (posAux  $>$  (posicao  $-1$ ))

NoTemp  $\leftarrow$  NoTemp.prox;

posAux  $\leftarrow$  posAux + 1;



**fimenquanto;**

NovoNo.prox  $\leftarrow$  NoTemp.prox;

NoTemp.prox  $\leftarrow$  NovoNo;

**fimse;**

**fimse;**

**fim\_módulo;**

Remover (elemento **numérico\_inteiro**)

**início\_módulo**

**Declarar**

NoTemp  $\leftarrow$  primeiro **No**;

NoAnt  $\leftarrow$  **nulo** No;

**se** (primeiro.elemento = elemento)

**então**

primeiro  $\leftarrow$  primeiro.prox;

**senão**

**enquanto** (NoTemp  $\neq$  **nulo** e NoTemp.elemento  $\neq$  elemento)

NoAnt  $\leftarrow$  NoTemp;



NoTemp  $\leftarrow$  NoTemp.prox;

**fimenquanto;**

**se (NoTemp  $\neq$  nulo)**

**então**

NoAnt.prox  $\leftarrow$  NoTemp.prox;

**fimse;**

**se (NoTemp = ultimo)**

**então**

ultimo  $\leftarrow$  NoAnt;

**fimse;**

**fimse;**

**fim\_módulo;**

ElementoInicio()

**início\_módulo**

**se (não ListaVazia())**

**então**

escrever ("O primeiro elemento da lista ligada é ", primeiro.elemento);

**senão**



escrever (“Lista Ligada vazia”);

**fimse;**

**fim\_módulo;**

ElementoFinal()

**início\_módulo**

**se (não ListaVazia())**

**então**

escrever ("O último elemento da lista ligada é " , ultimo.elemento);

**senão**

escrever (“Lista Ligada vazia”);

**fimse;**

**fim\_módulo;**

**No** BuscarNo (elemento **numérico\_inteiro**)

**início\_módulo**

**Declarar**

$i \leftarrow 1$  **numérico\_inteiro;**

NoTemp  $\leftarrow$  primeiro **No**;



**enquanto** (NoTemp  $\neq$  nulo) **faça**

**se** (NoTemp.elemento = elemento)

**então**

            escrever ("No ", NoTemp.elemento , " posição " , i);

**retornar** NoTemp;

**fimse**;

    i  $\leftarrow$  i + 1;

    NoTemp  $\leftarrow$  NoTemp.prox;

**fimenquanto**;

**retornar** nulo;

**fim\_módulo**;

MostrarLista()

**início\_módulo**

**Declarar**

NoTemp  $\leftarrow$  primeiro **No**;

i  $\leftarrow$  1 **numérico\_inteiro**;





**enquanto** (NoTemp <> nulo) **faça**

escrever ("Elemento ", NoTemp.elemento , " posição " , i);

NoTemp  $\leftarrow$  NoTemp.prox;

i  $\leftarrow$  i + 1;

**fimenquanto;**

**fim\_módulo;**

**fimalgoritmo.**

### **Algoritmo** Exemplo3

**início\_algoritmo**

**Declarar**

intLista = **novo** ListaLigada();

i, num **numérico\_inteiro;**

**para** i **de** 1 **até** 10 **passo** + 1 **faça**

ler(num);

**se** (num **mod** 3 = 0)

**então**

```
intLista.InserirFinal(novo No(num));
```



```
fimse;
```

```
fimpara;
```

```
intLista.MostrarLista();
```

```
fim_algoritmo.
```

## APLICANDO LISTAS LIGADAS EM JAVA

Vamos desenvolver parte de um programa em Java que recebe do usuário dez números inteiros numa lista ligada, inserindo apenas os divisíveis por três e mostrando esses números.

Precisamos definir a estrutura de nó, a classe com as funções e procedimentos de manipulação da Lista Ligada e a classe que vai executar o algoritmo.

```
import javax.swing.;
```

```
class No {
```

*int elemento;*



*No prox;*

*No (int elem) {*

*elemento = elem;*

*prox = null;*

*}*

*}*

*class ListaLigada*

*{*

*No primeiro, ultimo;*

*ListaLigada () {*

*primeiro = null;*

*ultimo = null;*

*}*

*public boolean ListaVazia()*

*public void InserirInicio (No novoNo)*

*public void InserirFinal (No novoNo)*



*public int ContarNos ( ) {...}*

*public void InserirMeio(No NovoNo, int posicao) {...}*

*public void Remover (int elemento) {...}*

*public void ElementoInicio( ) {...}*

*public void ElementoFinal( ) {...}*

*public No BuscaNo (int elemento) {...}*

*public void MostrarLista( ) {...}*

*class Exemplo3 {*

*public static void main(String arg[])*

*{*

*ListaLigada intLista = new ListaLigada();*

*int i, num;*

*for (i = 1; i <= 10; i++)*

*{*

*num = Integer.parseInt(JOptionPane.showInputDialog(*

*“Digite um número inteiro”));*

```
if (num % 3 == 0)
```



```
{
```

```
    intLista.InserirFinal(new No(num));
```

```
}
```

```
}
```

```
intLista.MostrarLista();
```

```
System.exit(0);
```

```
}
```

```
}}
```

### Atividade extra

### **Indicação de leitura:**

*Você pode utilizar o livro Estrutura de Dados: algoritmos, análise da complexidade e implementações em Java e C/C++, da Ana Fernanda Gomes Ascencio e Graziela Santos de Araújo, no capítulo 3 de aplicações da Lista Ligada.*

**\*\***



### Referência Bibliográfica

CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Algoritmos Teoria e Prática**. Editora Cammpus. 3a Edição. 2012.

GOODRICH, M. T.; TAMASSIA, R. **Estruturas de Dados & Algoritmos em Java**. Editora Grupo A: Bookman, 5a Edição. 2013.


### **Atividade Prática 12 – Entendendo as aplicações de Lista Ligada**

*Título da Prática: Aplicações com Lista Ligada em Java*

*Objetivos: Entender como utilizar o netbeans para desenvolver programas em Java para manipular e aplicar as operações com Lista Ligada*

*Materiais, Métodos e Ferramentas: Computador, netbeans, Java.*

### **Atividade Prática**

O Algoritmo de manipulação das operações de Lista Ligada para desenvolver um algoritmo que recebe do usuário cinco números reais numa lista ligada e mo.   
esses números, pode ser escrito como segue.

Desenvolva o programa em Java deste algoritmo no NetBeans.

### **Algoritmo No**

***início\_algoritmo***

***tipo*** No = ***registro***

*elemento*  $\leftarrow$  0 ***numérico\_real***;

***prox***  $\leftarrow$  nulo No;

***fimregistro***;

***fimalgoritmo***.

### **Algoritmo ListaLigada**

***início\_algoritmo***

// definição do tipo registro ListaLigada com os campos abaixo

***tipo*** ListaLigada = ***registro*** // o tipo registro chama-se ListaLigada

*primeiro*  $\leftarrow$  nulo **No**;

*ultimo*  $\leftarrow$  nulo **No**;

***fimregistro;***



***lógico*** *ListaVazia()*

***início\_módulo***

***se*** (*primeiro* = ***nulo*** e *ultimo* = ***nulo***)

***então***

***retornar*** verdadeiro;

***senão***

***retornar*** falso;

***fimse;***

***fim\_módulo;***

*InserirInicio* (*novoNo*  $\uparrow$  ***No***)

***início\_módulo***

***se*** (*ListaVazia()*)

***então***

*ultimo*  $\leftarrow$  *novoNo*;

***senão***

*novoNo.prox*  $\leftarrow$  *primeiro*;



***fimse;***



*primeiro*  $\leftarrow$  *novNo*;

***fim\_módulo;***

*InserirFinal* (*novNo*  $\uparrow$  **No**)

***início\_módulo***

***se*** (*ListaVazia()*)

***então***

*primeiro*  $\leftarrow$  *novNo*;

***senão***

*ultimo.prox*  $\leftarrow$  *novNo*;

***fimse;***

*ultimo*  $\leftarrow$  *novNo*;

***fim\_módulo;***

***numérico\_inteiro*** *ContarNos*( )

***início\_módulo***

***Declarar***

*tamanho*  $\leftarrow$  0 ***numérico\_inteiro;***

*NoTemp* ← primeiro **No**;



**enquanto** (*NoTemp* <> nulo) **faça**

*tamanho* ← *tamanho* + 1;

*NoTemp* ← *NoTemp.prox*;

**fimenquanto**;

**retornar** *tamanho*;

**fim\_módulo**;

*InserirMeio*(*NovoNo* ↑ **No**, *posicao* **numérico\_inteiro**)

**início\_módulo**

**Declarar**

*NoTemp* ← primeiro **No**;

*NroNos*, *posAux* ← 1 **numérico\_inteiro**;

*NroNos* ← *ContarNos*( );

**se** (*posicao* ≤ 1)

**então**

*InserirInicio*(*NovoNo*);

**senão**



**se** (*posicao* > *NroNos*)

**então**

*InserirFinal*(*NovoNo*);

**senão**

**enquanto** (*posAux* > (*posicao* - 1))

*NoTemp* ← *NoTemp.prox*;

*posAux* ← *posAux* + 1;

**fimenquanto;**

*NovoNo.prox* ← *NoTemp.prox*;

*NoTemp.prox* ← *NovoNo*;

**fimse;**

**fimse;**

**fim\_módulo;**

*Remover* (elemento **numérico\_real**)

**início\_módulo**

**Declarar**

*NoTemp* ← primeiro **No**;

$NoAnt \leftarrow \textit{nulo } No;$



*se* ( $primeiro.elemento = elemento$ )

*então*

$primeiro \leftarrow primeiro.prox;$

*senão*

*enquanto* ( $NoTemp \neq \textit{nulo e } NoTemp.elemento \neq elemento$ )

$NoAnt \leftarrow NoTemp;$

$NoTemp \leftarrow NoTemp.prox;$

*fimenquanto;*

*se* ( $NoTemp \neq \textit{nulo}$ )

*então*

$NoAnt.prox \leftarrow NoTemp.prox;$

*fimse;*

*se* ( $NoTemp = ultimo$ )

*então*

$ultimo \leftarrow NoAnt;$

*fimse;*

*fimse;*

***fim\_módulo;***



*ElementoInicio()*

***início\_módulo***

***se (não ListaVazia())***

***então***

*escrever ("O primeiro elemento da lista ligada é ", primeiro.elemento);*

***senão***

*escrever ("Lista Ligada vazia");*

***fimse;***

***fim\_módulo;***

*ElementoFinal()*

***início\_módulo***

***se (não ListaVazia())***

***então***

*escrever ("O último elemento da lista ligada é ", ultimo.elemento);*

***senão***

*escrever ("Lista Ligada vazia");*

***fimse;***



***fim\_módulo;***

↑ **No** BuscarNo (elemento **numérico\_real**)

***início\_módulo***

***Declarar***

***$i \leftarrow 1$  numérico\_inteiro;***

***NoTemp  $\leftarrow$  primeiro No;***

***enquanto* (NoTemp  $\neq$  nulo) ***faça*****

***se* (NoTemp.elemento = elemento)**

***então***

***escrever* ("No ", NoTemp.elemento , " posição ",  $i$ );**

***retornar* NoTemp;**

***fimse;***

***$i \leftarrow i + 1$ ;***

***NoTemp  $\leftarrow$  NoTemp.prox;***

***fimenquanto;***

***retornar nulo;***

***fim\_módulo;***



***MostrarLista()***

***início\_módulo***

***Declarar***

***NoTemp***  $\leftarrow$  primeiro **No**;

***i***  $\leftarrow$  1 **numérico\_inteiro**;

***enquanto*** (***NoTemp***  $\neq$  nulo) ***faça***

***escrever*** ("Elemento ", ***NoTemp.elemento*** , "posição " , ***i***);

***NoTemp***  $\leftarrow$  ***NoTemp.prox***;

***i***  $\leftarrow$  ***i*** + 1;

***fimenquanto***;

***fim\_módulo***;

***fimalgoritmo***.

***Algoritmo Exemplo1***

***início\_algoritmo***

***Declarar***

*realLista* = **novo** *ListaLigada*();



*i* **numérico\_inteiro**;

*num* **numérico\_real**;

**para** *i* **de** 1 **até** 5 **passo** + 1 **faça**

*ler*(*num*);

*realLista*.*InserirFinal*(**novo** *No*(*num*));

**fimpara**;

*realLista*.*MostrarLista*();

**fim\_algoritmo.**

### **Gabarito Atividade Prática**

**import** *javax.swing*;

**class** *No*





```
{
```

```
double elemento;
```

```
No prox;
```

```
No (double elem)
```

```
{
```

```
elemento = elem;
```

```
prox = null;
```

```
}
```

```
}
```

```
class ListaLigada
```

```
{
```

```
No primeiro, ultimo;
```

```
ListaLigada ()
```

```
{
```

```
primeiro = null;
```

```
ultimo = null;
```

}



**public boolean** ListaVazia( )

{

**if** (primeiro == **null** && ultimo == **null**)

{

**return true;**

}

**else**

{

**return false;**

}

}

**public void** InserirInicio (No novoNo)

{

**if** (ListaVazia())

{

ultimo = novoNo;

```
}
```



```
else
```

```
{
```

```
    novoNo.prox = primeiro;
```

```
}
```

```
primeiro = novoNo;
```

```
}
```

```
public void InserirFinal (No novoNo)
```

```
{
```

```
    if (ListaVazia())
```

```
    {
```

```
        primeiro = novoNo;
```

```
    }
```

```
    else
```

```
    {
```

```
        ultimo.prox = novoNo;
```

```
    }
```

```
    ultimo = novoNo;
```

}



**public int** ContarNos ( )

{

**int** tamanho = 0;

No NoTemp = primeiro;

**while** (NoTemp != null)

{

tamanho = tamanho + 1;

NoTemp = NoTemp.prox;

}

**return** tamanho;

}

**public void** InserirMeio(No NovoNo, int posicao)

{

No NoTemp = primeiro;

**int** NroNos, posAux = 1;



```
NroNos = ContarNos( );
```

```
if (posicao <= 1)
```

```
{
```

```
    InserirInicio(NovoNo);
```

```
}
```

```
else
```

```
{
```

```
    if (posicao > NroNos)
```

```
    {
```

```
        InserirFinal(NovoNo);
```

```
    }
```

```
else
```

```
{
```

```
    while (posAux < (posicao - 1))
```

```
    {
```

```
        NoTemp = NoTemp.prox;
```

```
        posAux = posAux + 1;
```

```
    }
```

```
NovoNo.prox = NoTemp.prox;
```



```
NoTemp.prox = NovoNo;
```

```
}
```

```
}
```

```
}
```

```
public void Remover (double elemento)
```

```
{
```

```
No NoTemp = primeiro;
```

```
No NoAnt = null;
```

```
if (primeiro.elemento == elemento)
```

```
{
```

```
    primeiro = primeiro.prox;
```

```
}
```

```
else
```

```
{
```

```
    while (NoTemp != null && NoTemp.elemento != elemento)
```

```
{
```

```
NoAnt = NoTemp;
```



```
NoTemp = NoTemp.prox;
```

```
}
```

```
if(NoTemp != null)
```

```
{
```

```
NoAnt.prox = NoTemp.prox;
```

```
}
```

```
if (NoTemp == ultimo)
```

```
{
```

```
ultimo = NoAnt;
```

```
}
```

```
}
```

```
}
```

```
public void ElementoInicio()
```

```
{
```

```
if (! ListaVazia())
```

```
{
```

```
System.out.println("O primeiro elemento é " +  
    primeiro.elemento);
```



```
}
```

```
else
```

```
{
```

```
System.out.println("Lista Ligada Vazia");
```

```
}
```

```
}
```

```
public void ElementoFinal( )
```

```
{
```

```
if (! ListaVazia())
```

```
{
```

```
System.out.println("O último elemento é " +  
    ultimo.elemento);
```

```
}
```

```
else
```

```
{
```

```
System.out.println("Lista Ligada Vazia");
```

```
}
```



}



**public** No BuscarNo (**double** elemento)

{

**int** i = 1;

No NoTemp = primeiro;

**while** (NoTemp **!=** null)

{

**if** (NoTemp.elemento == elemento)

{

System.out.println("No " + NoTemp.elemento + " posição "  
+ i);

**return** NoTemp;

}

i = i + 1;

NoTemp = NoTemp.prox;

}

**return** null;

}



**public void** MostrarLista( )

{

**int** i = 1;

No NoTemp = primeiro;

**while** (NoTemp != null)

{

System.out.println("Elemento " + NoTemp.elemento + " posição  
" + i);

NoTemp = NoTemp.prox;

i = i + 1;

}

}

}

**class** Exemplo1

{

**public static void main (String arg [])**



**{**

ListaLigada realLista = **new** ListaLigada();

**int** i;

**double** num;

**for** (i = 1 ; i <= 5 ; i++)

**{**

num = Double.parseDouble(JOptionPane.showInputDialog(  
    “digite um número real”));

realLista.InserirFinal(**new** No(num));

**}**

realLista.MostrarLista();

System.exit(0);

**}**

**}**

**Ir para exercício**

