



Universidad Nacional Autónoma
de México

Facultad de Estudios Superiores
Aragón

Tarea 4: Listas Ligadas

Estructura de Datos

Prof. Hernández Cabrera Jesús

Grupo: 1306

Angeles Mermejo Octavio Emiliano

Código Fuente Listas Ligadas:

```
1 package fes.aragon.clases;
2
3 4 usages
4 public class ListaLigada<T> {
5     18 usages
6     private Nodo<T> head;
7     no usages
8     private int tamaño;
9
10    1 usage
11    public ListaLigada() {
12    }
13
14    5 usages
15    public void transversal(){
16        Nodo<T> aux = this.head;
17        while (aux != null){
18            System.out.print(aux.getData() + " | ");
19            aux = aux.getSiguiente();
20        }
21        System.out.println("");
22    }
23
24    no usages
25    public boolean estaVacía(){
26        if (this.head == null) {
27            System.out.println("Esta Vacía");
28            return true;
29        }
30        System.out.println("No Esta Vacía");
31        return false;
32    }
33
34    no usages
35    public int getTamaño(){
36        Nodo aux = this.head;
37        int contador = 0;
38        while (aux!=null){
39            contador++;
40            aux = aux.getSiguiente();
41        }
42        return contador;
43    }
44
45    5 usages
46    public void agregarAlFinal(T dato){
47        Nodo<T> nuevo = new Nodo<>(dato);
48        if(head == null){
49            head = nuevo;
50        }else {
51            Nodo<T> aux = this.head;
52            while (aux.getSiguiente() != null){
53                aux = aux.getSiguiente();
54            }
55            aux.setSiguiente(nuevo);
56        }
57    }
58 }
```

```
45         aux = aux.getSiguiente();
46     }
47     aux.setSiguiente(nuevo);
48 }
49 }
50
```

2 usages

```
51 public void agregarAlInicio(T dato){
52     Nodo<T> nuevo = new Nodo<>(dato);
53     this.head = new Nodo<>(dato, this.head);
54 }
```

⚠ 11 ✖ 11 ^ v

no usages

```
56 public void agregarDespuesDe (Nodo<T> referencia, T dato){
57     Nodo<T> nuevo = new Nodo<>(dato);
58     if (referencia != null){
59         nuevo.setSiguiente(referencia.getSiguiente());
60         referencia.setSiguiente(nuevo);
61     }
62 }
63
```

1 usage

```
64 public void eliminar (int posicion){
65     if(this.head == null || posicion < 0){
66         return;
67     }
68     if (posicion == 0){
69         this.head = this.head.getSiguiente();
70         return;
71     }
72
73     Nodo<T> actual = this.head;
74     for(int i = 0 ; i < posicion -1 && actual != null; i++){
75         actual = actual.getSiguiente();
76     }
77
78     if (actual == null || actual.getSiguiente() == null){
79         return;
80     }
81     actual.setSiguiente(actual.getSiguiente().getSiguiente());
82 }
83
```

1 usage

```
84 public void eliminarElPrimer(){
85     if(this.head != null){
86         this.head = this.head.getSiguiente();
87     }
88 }
89
```

no usages

```
90 public void eliminarElFinal() {
```

```
90 public void eliminarElFinal() {
91     Nodo<T> aux = this.head;
92     while(aux.getSiguiente().getSiguiente() != null){
93         aux.getSiguiente();
94     }
95     aux.setSiguiente(null);
96 }
97
98 no usages
99 public int buscar (T dato) {
100     Nodo<T> aux = this.head;
101     int posicion = 0 ;
102     while (aux != null){
103         if(aux.getData().equals(dato)){
104             return posicion;
105         }
106         aux = aux.getSiguiente();
107         posicion++;
108     }
109     return -1;
110 }
```

```
110
111 1 usage
112 public void actualizar (int posicion , T dato){
113     Nodo<T> aux = this.head;
114     int posicion1 = 0 ;
115     while (aux != null){
116         if(posicion1 == posicion){
117             aux.setData(dato);
118             return;
119         }
120         aux = aux.getSiguiente();
121         posicion1++;
122     }
123 }
```

Código Fuente Nodo:

```
1 package fes.aragon.clases;
2
3 public class Nodo<T> {
4     private T dato;
5     private Nodo<T> siguiente;
6
7     public Nodo() {
8     }
9
10    public Nodo(T dato) { this.dato = dato; }
11
12    public Nodo(T dato, Nodo<T> siguiente) {
13        this.dato = dato;
14        this.siguiente = siguiente;
15    }
16
17    public T getData() { return dato; }
18
19    public void setData(T dato) { this.dato = dato; }
20
21    public Nodo<T> getSiguiente() { return siguiente; }
22
23    public void setSiguiente(Nodo<T> siguiente) {
24        this.siguiente = siguiente;
25    }
26
27    @Override
28    public String toString() { return dato + " " + siguiente; }
29 }
```

Código Fuente SmartPhone:

```
1 package fes.aragon.clases;
2
3 11 usages
4 public class SmartPhone {
5     4 usages
6     private String marca;
7     4 usages
8     private int almacenamiento;
9     4 usages
10    private String color;
11    4 usages
12    private int AnioDeFabricacion;
13    4 usages
14    private int ram;
15    2 usages
16    private int posicion;
17
18    8 usages
19    public SmartPhone(int posicion ,String marca, int anioDeFabricacion, int ram, int almacenamiento, String color){
20        this.marca = marca;
21        AnioDeFabricacion = anioDeFabricacion;
22        this.ram = ram;
23        this.almacenamiento = almacenamiento;
24        this.color = color;
25        this.posicion =posicion;
26    }
27
28    no usages
29    public String getMarca() { return marca; }
30
31    no usages
32    public void setMarca(String marca) { this.marca = marca; }
33
34    no usages
35    public int getAnioDeFabricacion() { return AnioDeFabricacion; }
36
37    no usages
38    public void setAnioDeFabricacion(int anioDeFabricacion) { AnioDeFabricacion = anioDeFabricacion; }
39
40    no usages
41    public int getRam() { return ram; }
42
43    no usages
44    public void setRam(int ram) { this.ram = ram; }
45
46    no usages
47    public int getAlmacenamiento() { return almacenamiento; }
48
49    no usages
50    public void setAlmacenamiento(int almacenamiento) { this.almacenamiento = almacenamiento; }
51
52    no usages
53    public String getColor() { return color; }
54
55    no usages
56    public void setColor(String color) { this.color = color; }
57
58
59
60
```

```
61     @Override
62     public String toString() {
63         return
64             "\n\tSmartPhone '" + posicion + '\'' +
65             "\n\tMarca: '" + marca + '\'' +
66             ",\n\tAño de Fabricación: " + AnioDeFabricacion +
67             ",\n\tRAM: " + ram + " GB" +
68             ",\n\tAlmacenamiento: " + almacenamiento + " GB" +
69             ",\n\tColor: '" + color + '\'';
70     }
71
72 }
73
```

Código Fuente Main:

```
1 package fes.aragon.inicio;
2
3 import fes.aragon.clases.ListaLigada;
4 import fes.aragon.clases.SmartPhone;
5
6 public class Main {
7     public static void main(String[] args) {
8         ListaLigada<SmartPhone> celulares = new ListaLigada<>();
9
10        celulares.agregarAlInicio(new SmartPhone( posicion: 1, marca: "Iphone" , anioDeFabricacion: 2015, ram: 8 , almacenamiento: 128 , color: "Azul"));
11        celulares.agregarAlFinal(new SmartPhone( posicion: 2, marca: "Samsung" , anioDeFabricacion: 2022, ram: 12 , almacenamiento: 64 , color: "Verde"));
12        celulares.agregarAlFinal(new SmartPhone( posicion: 3, marca: "Motorola" , anioDeFabricacion: 2024, ram: 12 , almacenamiento: 256 , color: "Negro"));
13        celulares.agregarAlFinal(new SmartPhone( posicion: 4, marca: "Huawei" , anioDeFabricacion: 2021, ram: 12 , almacenamiento: 256 , color: "Blanco"));
14        celulares.agregarAlFinal(new SmartPhone( posicion: 5, marca: "Xiaomi" , anioDeFabricacion: 2019, ram: 8 , almacenamiento: 64 , color: "Gris"));
15        ListaLigada<SmartPhone> respaldo = celulares;
16
17        System.out.println("Lista Inicial: ");
18        celulares.transversal();
19
20        celulares.eliminar( posicion: 1);
21        System.out.println("Lista Actualizada: Se Elimino el dispositivo 2");
22        celulares.transversal();
23
24        respaldo.actualizar( posicion: 1,new SmartPhone( posicion: 2, marca: "OPPO" , anioDeFabricacion: 2022 , ram: 16, almacenamiento: 256, color: "Rosa"));
25        System.out.println("Lista actualizada: Actualizamos los datos del SmartPhone 2");
26        respaldo.transversal();
27
28        celulares.agregarAlInicio(new SmartPhone( posicion: 1, marca: "Google" , anioDeFabricacion: 2024, ram: 16, almacenamiento: 128 , color: "Blanco"));
29        celulares.agregarAlFinal(new SmartPhone( posicion: 6, marca: "Honor" , anioDeFabricacion: 2021, ram: 8, almacenamiento: 128, color: "Negro"));
30        System.out.println("Lista Actualizada: Se agregaron dos SmartPhone nuevos en el inicio y al final de la lista");
31        celulares.transversal();
32
33        celulares.eliminarElPrimer();
34        System.out.println("Lista Actualizada : Se Elimino el primer SmartPhone de la lista");
35        celulares.transversal();
36    }
37 }
```


Ejecución del Programa:

```
/opt/java/jdk-17.0.2/bin/java -javaagent:/opt/java/idea-IC-233.14015.106/lib/idea_rt.jar=38511:/opt/java/idea-IC-233.14015.106/bin -Dfile.encoding=UTF-8
Lista Inicial:
```

```
SmartPhone '1'
Marca: 'Iphone',
Año de Fabricación: 2015,
RAM: 8 GB,
Almacenamiento: 128 GB,
Color: 'Azul'
SmartPhone '2'
Marca: 'Samsung',
Año de Fabricación: 2022,
RAM: 12 GB,
Almacenamiento: 64 GB,
Color: 'Verde'
SmartPhone '3'
```

```
Marca: 'Motorola',
Año de Fabricación: 2024,
RAM: 12 GB,
Almacenamiento: 256 GB,
Color: 'Negro'
SmartPhone '4'
Marca: 'Huawei',
Año de Fabricación: 2021,
RAM: 12 GB,
Almacenamiento: 256 GB,
Color: 'Blanco'
SmartPhone '5'
Marca: 'Xiaomi',
Año de Fabricación: 2019,
RAM: 8 GB,
Almacenamiento: 64 GB,
Color: 'Gris'
```

```
Lista Actualizada: Se Elimino el dispositivo 2
```

```
SmartPhone '1'
Marca: 'Iphone',
Año de Fabricación: 2015,
RAM: 8 GB,
Almacenamiento: 128 GB,
Color: 'Azul'
SmartPhone '3'
Marca: 'Motorola',
Año de Fabricación: 2024,
RAM: 12 GB,
Almacenamiento: 256 GB,
Color: 'Negro'
SmartPhone '4'
Marca: 'Huawei',
Año de Fabricación: 2021,
RAM: 12 GB,
Almacenamiento: 256 GB,
Color: 'Blanco'
SmartPhone '5'
Marca: 'Xiaomi',
Año de Fabricación: 2019,
RAM: 8 GB,
Almacenamiento: 64 GB,
Color: 'Gris'
```

```
Lista actualizada: Actualizamos los datos del SmartPhone 2
```

```
SmartPhone '1'
Marca: 'Iphone',
Año de Fabricación: 2015,
RAM: 8 GB,
Almacenamiento: 128 GB,
Color: 'Azul'
SmartPhone '2'
Marca: 'OPPO',
```

```
Año de Fabricación: 2022,  
RAM: 16 GB,  
Almacenamiento: 256 GB,  
Color: 'Rosa'  
SmartPhone '4'  
Marca: 'Huawei',  
Año de Fabricación: 2021,  
RAM: 12 GB,  
Almacenamiento: 256 GB,  
Color: 'Blanco'  
SmartPhone '5'  
Marca: 'Xiaomi',  
Año de Fabricación: 2019,  
RAM: 8 GB,  
Almacenamiento: 64 GB,  
Color: 'Gris'
```

Lista Actualizada: Se agregaron dos SmartPhone nuevos en el inicio y al final de la lista

```
SmartPhone '1'  
Marca: 'Google',  
Año de Fabricación: 2024,  
RAM: 16 GB,  
Almacenamiento: 128 GB,  
Color: 'Blanco'  
SmartPhone '1'  
Marca: 'Iphone',  
Año de Fabricación: 2015,  
RAM: 8 GB,  
Almacenamiento: 128 GB,  
Color: 'Azul'  
SmartPhone '2'  
Marca: 'OPPO',  
Año de Fabricación: 2022,  
RAM: 16 GB,  
Almacenamiento: 256 GB,  
Color: 'Rosa'  
SmartPhone '4'  
Marca: 'Huawei',  
Año de Fabricación: 2021,  
RAM: 12 GB,  
Almacenamiento: 256 GB,  
Color: 'Blanco'  
SmartPhone '5'  
Marca: 'Xiaomi',  
Año de Fabricación: 2019,  
RAM: 8 GB,  
Almacenamiento: 64 GB,  
Color: 'Gris'  
SmartPhone '6'  
Marca: 'Honor',  
Año de Fabricación: 2021,  
RAM: 8 GB,  
Almacenamiento: 128 GB,  
Color: 'Negro'
```

Lista Actualizada : Se Elimino el primer SmartPhone de la lista

```
SmartPhone '1'  
Marca: 'Iphone',  
Año de Fabricación: 2015,  
RAM: 8 GB,  
Almacenamiento: 128 GB,  
Color: 'Azul'  
SmartPhone '2'  
Marca: 'OPPO',  
Año de Fabricación: 2022,  
RAM: 16 GB,  
Almacenamiento: 256 GB,  
Color: 'Rosa'  
SmartPhone '4'
```

```
SmartPhone '4'  
Marca: 'Huawei',  
Año de Fabricación: 2021,  
RAM: 12 GB,  
Almacenamiento: 256 GB,  
Color: 'Blanco'  
SmartPhone '5'  
Marca: 'Xiaomi',  
Año de Fabricación: 2019,  
RAM: 8 GB,  
Almacenamiento: 64 GB,  
Color: 'Gris'  
SmartPhone '6'  
Marca: 'Honor',  
Año de Fabricación: 2021,  
RAM: 8 GB,  
Almacenamiento: 128 GB,  
Color: 'Negro'
```

```
Process finished with exit code 0
```