



Universidad Nacional Autónoma de
México

Facultad de Estudios Superiores
Aragón

Alumno: Leal Sanchez Bryan Uriel

Materia: Estructura de Datos

Grupo: 1360

Profesor: Hernández Cabrera Jesús

Listas Ligadas

© Nodo.java × main.java © smartPhone.java © Object.java

```
1 package listaLigada;
2
3 class Nodo 15 usages
4 {
5     Object valor; 6 usages
6     Nodo siguiente; 22 usages
7
8     public Nodo(Object valor) 3 usages
9     {
10         this.valor = valor;
11         this.siguiente = null;
12     }
13 }
14
15 class Listaligada 2 usages
16 {
17     private Nodo head; 18 usages
18     private int tamaño; 10 usages
19
20     public Listaligada() 1 usage
21     {
22         this.head = null;
23         this.tamaño = 0;
24     }
25
26     public boolean estaVacía() 4 usages
27     {
28         return head == null;
29     }
30
31     public int getTamaño() no usages
```

```

15  class Listaligada 2 usages
31      public int getTamanio() no usages
32      {
33          return tamanio;
34      }
35
36      public void agregarAlFinal(Object valor) 6 usages
37      {
38          Nodo nuevoNodo = new Nodo(valor);
39          if (estaVacía())
40          {
41              head = nuevoNodo;
42          } else
43          {
44              Nodo temp = head;
45              while (temp.siguiete != null)
46              {
47                  temp = temp.siguiete;
48              }
49              temp.siguiete = nuevoNodo;
50          }
51          tamanio++;
52      }
53
54      public void agregarAlInicio(Object valor) 1 usage
55      {
56          Nodo nuevoNodo = new Nodo(valor);
57          if (estaVacía())
58          {
59              head = nuevoNodo;
60          }
    
```

```
© Nodo.java × main.java smartPhone.java Object.java
15  class Listaligada 2 usages
54  public void agregarAlInicio(Object valor) 1 usage
60      } else
61      {
62          nuevoNodo.siguiente = head;
63          head = nuevoNodo;
64      }
65      tamaño++;
66  }
67
68  public void agregarDespuesDe(Object referencia, Object valor) no usages
69  {
70      Nodo temp = head;
71      while (temp != null && !temp.valor.equals(referencia))
72      {
73          temp = temp.siguiente;
74      }
75      if (temp != null)
76      {
77          Nodo nuevoNodo = new Nodo(valor);
78          nuevoNodo.siguiente = temp.siguiente;
79          temp.siguiente = nuevoNodo;
80          tamaño++;
81      }
82  }
83
84  public void eliminar(int posicion) 1 usage
85  {
86      if (posicion < 0 || posicion >= tamaño)
87      {
88          throw new IndexOutOfBoundsException("Posición inválida");
```

```
© Nodo.java × main.java smartPhone.java Object.java
15 class Listaligada 2 usages
84 public void eliminar(int posicion) 1 usage
89     }
90     if (posicion == 0)
91     {
92         head = head.siguiente;
93     } else
94     {
95         Nodo temp = head;
96         for (int i = 0; i < posicion - 1; i++)
97         {
98             temp = temp.siguiente;
99         }
100         temp.siguiente = temp.siguiente.siguiente;
101     }
102     tamano--;
103 }
104
105 public void eliminarElPrimero() 1 usage
106 {
107     if (!estaVacia())
108     {
109         head = head.siguiente;
110         tamano--;
111     }
112 }
113
114 public void eliminarElFinal() no usages
115 {
116     if (estaVacia()) return;
117 }
```

```
© Nodo.java × main.java © smartPhone.java © Object.java
15  class Listaligada 2 usages
114  public void eliminarElFinal() no usages
118      if (tamano == 1)
119      {
120          head = null;
121      } else
122      {
123          Nodo temp = head;
124          while (temp.siguiete.siguiete != null)
125          {
126              temp = temp.siguiete;
127          }
128          temp.siguiete = null;
129      }
130      tamano--;
131  }
132
133  public int buscar(Object valor) no usages
134  {
135      Nodo temp = head;
136      int posicion = 0;
137      while (temp != null)
138      {
139          if (temp.valor.equals(valor))
140          {
141              return posicion;
142          }
143          temp = temp.siguiete;
144          posicion++;
145      }
```

© Nodo.java × main.java © smartPhone.java © Object.java

```
15      class Listaligada 2 usages
133      public int buscar(Object valor) no usages
146          return -1; // No encontrado
147      }
148
149      public void actualizar(Object aBuscar, Object valor) 1 usage
150      {
151          Nodo temp = head;
152          while (temp != null)
153          {
154              if (temp.valor.equals(aBuscar))
155              {
156                  temp.valor = valor;
157                  return;
158              }
159              temp = temp.siguiente;
160          }
161      }
162
163      public void transversal() 5 usages
164      {
165          Nodo temp = head;
166          while (temp != null)
167          {
168              System.out.println(temp.valor);
169              temp = temp.siguiente;
170          }
171      }
172  }
173
```

```
© Nodo.java  main.java ×  smartPhone.java  Object.java
1 package listaLigada;
2
3 public class main
4 {
5     public static void main(String[] args)
6     {
7         ListaLigada lista = new ListaLigada();
8
9         // Agregar 5 SmartPhones
10        lista.agregarAlFinal(new SmartPhone( marca: "Apple", modelo: "iPhone 15 proMax", precio: 999.99));
11        lista.agregarAlFinal(new SmartPhone( marca: "Samsung", modelo: "Galaxy S23", precio: 799.99));
12        lista.agregarAlFinal(new SmartPhone( marca: "Google", modelo: "Pixel 6", precio: 599.99));
13        lista.agregarAlFinal(new SmartPhone( marca: "OnePlus", modelo: "9 Pro", precio: 729.99));
14        lista.agregarAlFinal(new SmartPhone( marca: "Xiaomi", modelo: "Mi 11", precio: 699.99));
15
16        // Imprimir contenido
17        System.out.println("Contenido inicial:");
18        lista.transversal();
19
20        // Eliminar el de la posición 2
21        lista.eliminar( posición: 2);
22        System.out.println("\nDespués de eliminar el elemento en la posición 2:");
23        lista.transversal();
24
25        // Actualizar el segundo elemento
26        lista.actualizar(new SmartPhone( marca: "Samsung", modelo: "Galaxy S23", precio: 799.99), new SmartPhone( marca: "Samsung", modelo: "Galaxy S23", precio: 899.99));
27        System.out.println("\nDespués de actualizar el segundo elemento:");
28        lista.transversal();
29
30        // Agregar un elemento al inicio y otro al final
31        lista.agregarAlInicio(new SmartPhone( marca: "Sony", modelo: "Xperia 5", precio: 649.99));
```



```
© Nod...  main.java ×  smartPhone.java  Object.java
3      public class main
5          public static void main(String[] args)
20          // Eliminar el de la posición 2
21          lista.eliminar( posición: 2);
22          System.out.println("\nDespués de eliminar el elemento en la posición 2:");
23          lista.transversal();
24
25          // Actualizar el segundo elemento
26          lista.actualizar(new SmartPhone( marca: "Samsung", modelo: "Galaxy S23", precio: 799.99), new SmartPhone( marc
27          System.out.println("\nDespués de actualizar el segundo elemento:");
28          lista.transversal();
29
30          // Agregar un elemento al inicio y otro al final
31          lista.agregarAlInicio(new SmartPhone( marca: "Sony", modelo: "Xperia 5", precio: 649.99));
32          lista.agregarAlFinal(new SmartPhone( marca: "Huawei", modelo: "P50", precio: 599.99));
33          System.out.println("\nDespués de agregar al inicio y al final:");
34          lista.transversal();
35
36          // Eliminar el primero
37          lista.eliminarElPrimero();
38          System.out.println("\nDespués de eliminar el primero:");
39          lista.transversal();
40      }
41  }
42
```

```
© Nod...  main.java  smartPhone.java ×  Object.java
1      package listaLigada;
2
3      class SmartPhone { 9 usages
4          private String marca; 2 usages
5          private String modelo; 2 usages
6          private double precio; 2 usages
7
8          public SmartPhone(String marca, String modelo, double precio) { 9 usages
9              this.marca = marca;
10             this.modelo = modelo;
11             this.precio = precio;
12         }
13
14         @Override
15         public String toString() {
16             return "SmartPhone{" +
17                 "marca='" + marca + '\'' +
18                 ", modelo='" + modelo + '\'' +
19                 ", precio=" + precio +
20                 '}';
21         }
22     }
23
```

```
ject ▾
  © Nod...
  © smartPhone.java
  © Object.java

main ×

SmartPhone{marca='Sony', modelo='Xperia 5', precio=649.99}
SmartPhone{marca='Xiaomi', modelo='Mi 11', precio=699.99}

Después de eliminar el elemento en la posición 2:
SmartPhone{marca='Apple', modelo='iPhone 15 proMax', precio=999.99}
SmartPhone{marca='Samsung', modelo='Galaxy S23', precio=799.99}
SmartPhone{marca='OnePlus', modelo='9 Pro', precio=729.99}
SmartPhone{marca='Xiaomi', modelo='Mi 11', precio=699.99}

Después de actualizar el segundo elemento:
SmartPhone{marca='Apple', modelo='iPhone 15 proMax', precio=999.99}
SmartPhone{marca='Samsung', modelo='Galaxy S23', precio=799.99}
SmartPhone{marca='OnePlus', modelo='9 Pro', precio=729.99}
SmartPhone{marca='Xiaomi', modelo='Mi 11', precio=699.99}

Después de agregar al inicio y al final:
SmartPhone{marca='Sony', modelo='Xperia 5', precio=649.99}
SmartPhone{marca='Apple', modelo='iPhone 15 proMax', precio=999.99}
SmartPhone{marca='Samsung', modelo='Galaxy S23', precio=799.99}
SmartPhone{marca='OnePlus', modelo='9 Pro', precio=729.99}
SmartPhone{marca='Xiaomi', modelo='Mi 11', precio=699.99}
SmartPhone{marca='Huawei', modelo='P50', precio=599.99}

Después de eliminar el primero:
SmartPhone{marca='Apple', modelo='iPhone 15 proMax', precio=999.99}
SmartPhone{marca='Samsung', modelo='Galaxy S23', precio=799.99}
SmartPhone{marca='OnePlus', modelo='9 Pro', precio=729.99}
SmartPhone{marca='Xiaomi', modelo='Mi 11', precio=699.99}
SmartPhone{marca='Huawei', modelo='P50', precio=599.99}
```