



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGON**



TAREA 1

P R E S E N T A

Alexis Hernández Zamudio

APROFESOR

Jesús Hernández Cabrera

Gpo:1158

URL del repositorio:

[**https://github.com/TyrBalder1439/Estructura-de-Datos-**](https://github.com/TyrBalder1439/Estructura-de-Datos-)



Ciudad Nezahualcóyotl, EDOMEX. 3 de Septiembre del 2024

```
1 public class ListaLigada<T> {
2     private class Nodo {
3         T dato;
4         Nodo siguiente;
5         public Nodo(T dato) {
6             this.dato = dato;
7             this.siguiente = null;
8         }
9     }
10    private Nodo head;
11    private int tamanio;
12    public ListaLigada() {
13        this.head = null;
14        this.tamanio = 0;
15    }
16    public boolean estaVacia() {
17        return head == null;
18    }
19    public int getTamanio() {
20        return tamanio;
21    }
22    public void agregarAlFinal(T valor) {
23        Nodo nuevo = new Nodo(valor);
24        if (estaVacia()) {
25            head = nuevo;
26        } else {
27            Nodo actual = head;
28            while (actual.siguiente != null) {
29                actual = actual.siguiente;
30            }
31            actual.siguiente = nuevo;
32        }
33        tamanio++;
34    }
```

```

33         tamaño++;
34     }
35     public void agregarAlInicio (T valor) {
36         Nodo nuevo = new Nodo(valor);
37         nuevo.siguiente = head;
38         head = nuevo;
39         tamaño++;
40     }
41     public void agregarDespuesDe(T referencia, T valor) {
42         Nodo actual = head;
43         while (actual != null && !actual.dato.equals(referencia)) {
44             actual = actual.siguiente;
45         }
46         if (actual != null) {
47             Nodo nuevo = new Nodo(valor);
48             nuevo.siguiente = actual.siguiente;
49             actual.siguiente = nuevo;
50             tamaño++;
51         }
52     }
53     public void eliminar(int posicion) {
54         if (posicion >= 0 && posicion < tamaño) {
55             if (posicion == 0) {
56                 eliminarElPrimero();
57             } else {
58                 Nodo actual = head;
59                 for (int i = 0; i < posicion - 1; i++) {
60                     actual = actual.siguiente;
61                 }
62                 actual.siguiente = actual.siguiente.siguiente;
63                 tamaño--;

```

```

63         tamaño--;
64     }
65 }
66 }
67 public void eliminarElPrimero() {
68     if (!estaVacia()) {
69         head = head.siguiete;
70         tamaño--;
71     }
72 }
73 public void eliminarElFinal() {
74     if (!estaVacia()) {
75         if (head.siguiete == null) {
76             head = null;
77         } else {
78             Nodo actual = head;
79             while (actual.siguiete.siguiete != null) {
80                 actual = actual.siguiete;
81             }
82             actual.siguiete = null;
83         }
84         tamaño--;
85     }
86 }
87 public int buscar(T valor) {
88     Nodo actual = head;
89     int posicion = 0;
90     while (actual != null) {
91         if (actual.dato.equals(valor)) {
92             return posicion;
93         }

```

```
93         }
94         actual = actual.siguiente;
95         posicion++;
96     }
97     return -1; // No encontrado
98 }
99 public void actualizar(T aBuscar, T valor) {
100     Nodo actual = head;
101     while (actual != null) {
102         if (actual.dato.equals(aBuscar)) {
103             actual.dato = valor;
104             return;
105         }
106         actual = actual.siguiente;
107     }
108 }
109 public void transversal() {
110     Nodo actual = head;
111     while (actual != null) {
112         System.out.println(actual.dato);
113         actual = actual.siguiente;
114     }
115 }
116 }
```

```
1 public class SmartPhone {
2     private String marca;
3     private String modelo;
4     private double precio;
5
6     public SmartPhone(String marca, String modelo, double precio) {
7         this.marca = marca;
8         this.modelo = modelo;
9         this.precio = precio;
10    }
11
12    public String getMarca() {
13        return marca;
14    }
15
16    public void setMarca(String marca) {
17        this.marca = marca;
18    }
19
20    public String getModelo() {
21        return modelo;
22    }
23
24    public void setModelo(String modelo) {
25        this.modelo = modelo;
26    }
27
28    public double getPrecio() {
29        return precio;
30    }
31
32    public void setPrecio(double precio) {
33        this.precio = precio;
34    }
35
36    @Override
37    public String toString() {
38        return "Telefono (Marca = " + marca + " Modelo= " + modelo + " Precio= " + precio + ")";
39    }
40 }
41
42
```

```
1 public class Main {
2     public static void main(String[] args) {
3         ListaLigada<SmartPhone> lista = new ListaLigada<>();
4         System.out.println("Contenido de la lista:");
5         lista.agregarAlFinal(new SmartPhone("Samsung", "Galaxy S23 plus", 16000));
6         lista.agregarAlFinal(new SmartPhone("Apple", "iPhone 15 pro", 24000));
7         lista.agregarAlFinal(new SmartPhone("Huawei", "p70 pro", 22000));
8         lista.agregarAlFinal(new SmartPhone("Google", "Pixel 7 Pro ", 14000));
9         lista.agregarAlFinal(new SmartPhone("Redmi", "Poco", 5697));
10        lista.transversal();
11
12        System.out.println("\neliminar posición 2:");
13        lista.eliminar(2);
14        lista.transversal();
15
16        System.out.println("\nactualizar el segundo elemento:");
17        lista.actualizar(new SmartPhone("Apple", "iPhone 15 pro", 24000), new SmartPhone("Nokia", "Experia x34", 9500));
18        lista.transversal();
19
20        System.out.println("\nagregar elementos al inicio y al final:");
21        lista.agregarAlInicio(new SmartPhone("Huawei", "P70 Ultra", 26500));
22        lista.agregarAlFinal(new SmartPhone("Sony", "Xperia plus", 11700));
23        lista.transversal();
24
25        System.out.println("\neliminar el primer elemento:");
26        lista.eliminarElPrimero();
27        lista.transversal();
28    }
```

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\
```

```
Contenido de la lista:
```

```
Telefono (Marca:Samsung Modelo:Galaxy S23 plus Precio= $16000.0)
```

```
Telefono (Marca:Apple Modelo:iPhone 15 pro Precio= $24000.0)
```

```
Telefono (Marca:Huawei Modelo:p70 pro Precio= $22000.0)
```

```
Telefono (Marca:Google Modelo:Pixel 7 Pro Precio= $14000.0)
```

```
Telefono (Marca:Redmi Modelo:Poco Precio= $5697.0)
```

```
eliminar posición 2:
```

```
Telefono (Marca:Samsung Modelo:Galaxy S23 plus Precio= $16000.0)
```

```
Telefono (Marca:Apple Modelo:iPhone 15 pro Precio= $24000.0)
```

```
Telefono (Marca:Google Modelo:Pixel 7 Pro Precio= $14000.0)
```

```
Telefono (Marca:Redmi Modelo:Poco Precio= $5697.0)
```

```
actualizar el segundo elemento:
```

```
Telefono (Marca:Samsung Modelo:Galaxy S23 plus Precio= $16000.0)
```

```
Telefono (Marca:Apple Modelo:iPhone 15 pro Precio= $24000.0)
```

```
Telefono (Marca:Google Modelo:Pixel 7 Pro Precio= $14000.0)
```

```
Telefono (Marca:Redmi Modelo:Poco Precio= $5697.0)
```

```
agregar elementos al inicio y al final:
```

```
Telefono (Marca:Huawei Modelo:P70 Ultra Precio= $26500.0)
```

```
Telefono (Marca:Samsung Modelo:Galaxy S23 plus Precio= $16000.0)
```

```
Telefono (Marca:Apple Modelo:iPhone 15 pro Precio= $24000.0)
```

```
Telefono (Marca:Google Modelo:Pixel 7 Pro Precio= $14000.0)
```

```
Telefono (Marca:Redmi Modelo:Poco Precio= $5697.0)
```

```
Telefono (Marca:Sony Modelo:Xperia plus Precio= $11700.0)
```

```
eliminar el primer elemento:
```

```
Telefono (Marca:Samsung Modelo:Galaxy S23 plus Precio= $16000.0)
```

```
Telefono (Marca:Apple Modelo:iPhone 15 pro Precio= $24000.0)
```

```
Telefono (Marca:Google Modelo:Pixel 7 Pro Precio= $14000.0)
```

```
Telefono (Marca:Redmi Modelo:Poco Precio= $5697.0)
```

```
Telefono (Marca:Sony Modelo:Xperia plus Precio= $11700.0)
```

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