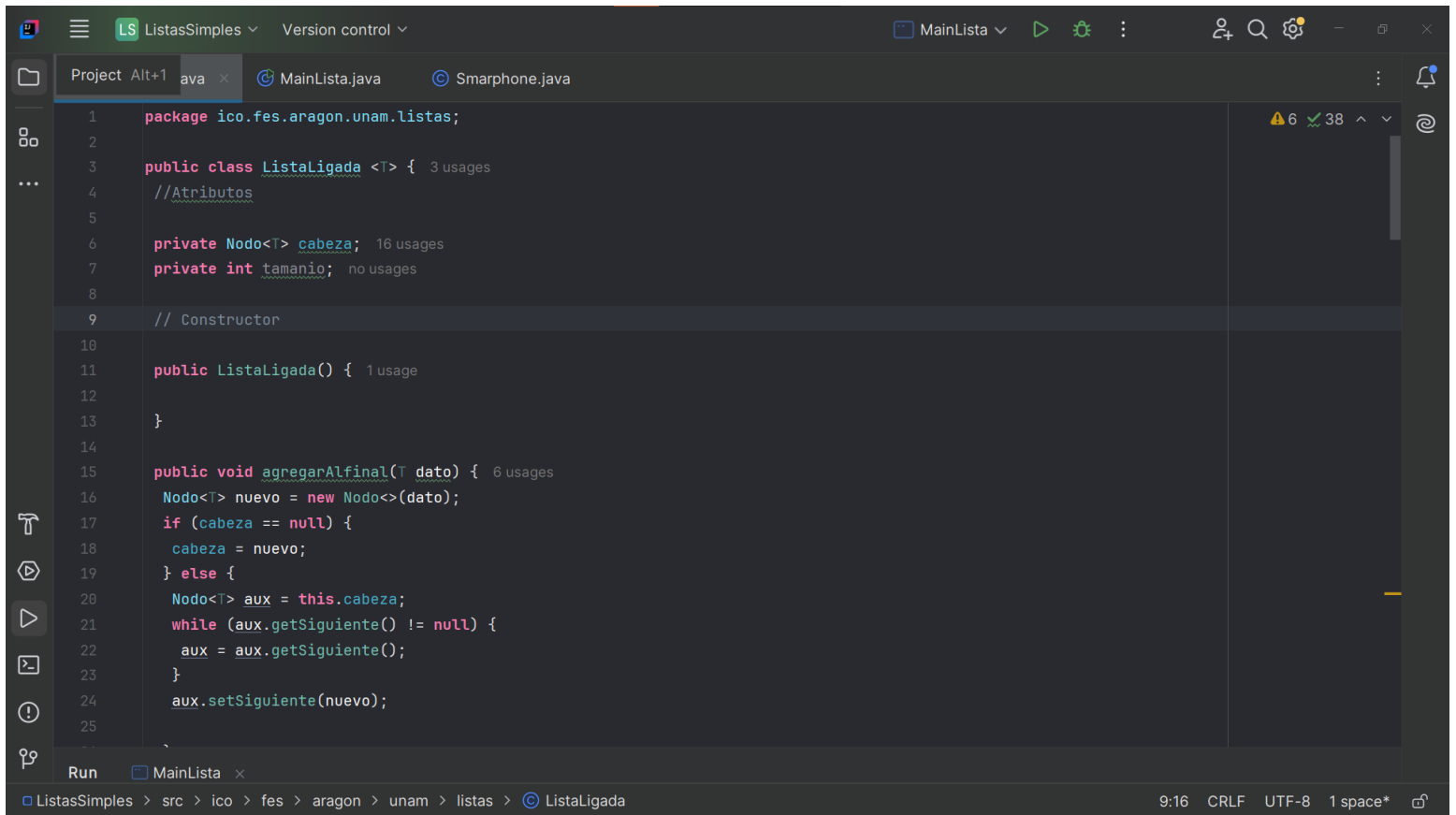


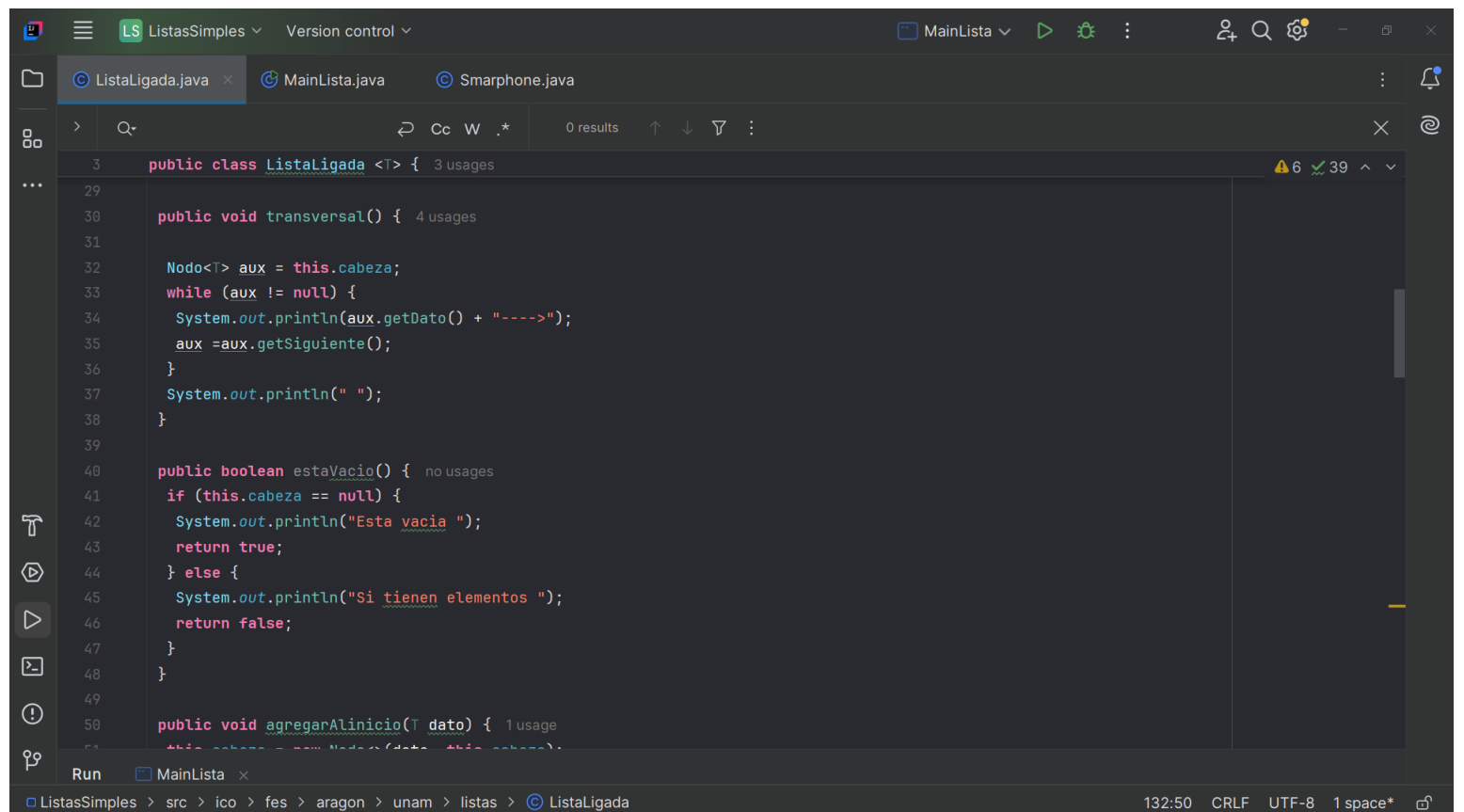
Código lista simple:



The screenshot shows an IDE window with the file `MainLista.java` open. The code defines a `Listaligada` class with a generic type `T`. It includes attributes `cabeza` and `tamanio`, a constructor, and a method `agregarAlfinal` for adding elements to the end of the list.

```
1 package ico.fes.aragon.unam.listas;
2
3 public class Listaligada <T> { 3 usages
4     //Atributos
5
6     private Nodo<T> cabeza; 16 usages
7     private int tamanio; no usages
8
9     // Constructor
10
11     public Listaligada() { 1 usage
12
13     }
14
15     public void agregarAlfinal(T dato) { 6 usages
16         Nodo<T> nuevo = new Nodo<>(dato);
17         if (cabeza == null) {
18             cabeza = nuevo;
19         } else {
20             Nodo<T> aux = this.cabeza;
21             while (aux.getSiguiete() != null) {
22                 aux = aux.getSiguiete();
23             }
24             aux.setSiguiete(nuevo);
25         }
26     }
27 }
```

The bottom status bar shows the file path: `ListasSimples > src > ico > fes > aragon > unam > listas > Listaligada`, the time `9:16`, and encoding settings `CRLF UTF-8 1 space*`.



The screenshot shows an IDE window with the file `ListaLigada.java` open. The code continues the implementation of the `Listaligada` class, adding methods `transversal` for traversing the list and `estaVacio` for checking if the list is empty. It also shows the beginning of the `agregarAlinicio` method.

```
29 public class Listaligada <T> { 3 usages
30
31     public void transversal() { 4 usages
32
33         Nodo<T> aux = this.cabeza;
34         while (aux != null) {
35             System.out.println(aux.getDato() + "---->");
36             aux = aux.getSiguiete();
37         }
38         System.out.println(" ");
39     }
40
41     public boolean estaVacio() { no usages
42         if (this.cabeza == null) {
43             System.out.println("Esta vacia ");
44             return true;
45         } else {
46             System.out.println("Si tienen elementos ");
47             return false;
48         }
49     }
50
51     public void agregarAlinicio(T dato) { 1 usage
52         this.cabeza = new Nodo<>(dato, this.cabeza);
53     }
54 }
```

The bottom status bar shows the file path: `ListasSimples > src > ico > fes > aragon > unam > listas > ListaLigada`, the time `132:50`, and encoding settings `CRLF UTF-8 1 space*`.

LS ListasSimplesVersion controlMainLista

ListaLigada.javaMainLista.javaSmarphone.java

>Q-0 results

```
3 public class ListaLigada <T> { 3 usages
50 public void agregarAlInicio(T dato) { 1 usage
51     this.cabeza = new Nodo<>(dato, this.cabeza);
52
53 }
54
55
56 public void agregarDespuesde(T datoExis, T datoNuevo){ 1 usage
57
58     Nodo<T> aux = this.cabeza;
59     Nodo<T> copia;
60     while (aux != null){
61         if(aux.getDato() == datoExis){
62             copia = aux.getSiguiente();
63             aux.setSiguiente(new Nodo<>(datoNuevo,copia));
64         }
65         aux = aux.getSiguiente();
66     }
67 }
68
69 public int buscar(T dato){ no usages
70     int indice = 0;
71     int contador = 0;
```

RunMainLista

ListasSimples > src > ico > fes > aragon > unam > listas > ListaLigada132:50CRLFUTF-81 space*

LS ListasSimplesVersion controlMainLista

ListaLigada.javaMainLista.javaSmarphone.java

>Q-0 results

```
3 public class ListaLigada <T> { 3 usages
68
69 public int buscar(T dato){ no usages
70     int indice = 0;
71     int contador = 0;
72     Nodo<T> aux = this.cabeza;
73     while(aux != null){
74         contador++;
75         if(aux.getDato() == dato){
76             indice = contador;
77             break;
78         }
79         aux = aux.getSiguiente();
80     }return indice;
81 }
82
83 public void eliminar(int posicion){ 1 usage
84     int contador = 0;
85     Nodo<T> aux = this.cabeza;
86     while (aux != null){
87         contador++;
88         if(contador == posicion-1 ){
89             aux.setSiguiente(aux.getSiguiente().getSiguiente());
90         }
```

RunMainLista

ListasSimples > src > ico > fes > aragon > unam > listas > ListaLigada132:50CRLFUTF-81 space*

LS ListasSimples Version control

MainLista MainLista Smartphone.java

Project Alt+1

0 results

6 39

```
3 public class ListaLigada <T> { 3 usages
83 public void eliminar(int posicion){ 1 usage
90 }
91     aux= aux.getSiguiente();
92 }
93
94 }
95
96 public void eliminarElfinal(){ no usages
97     int i = obtenerTamanio();
98     int contador = 0;
99     Nodo<T> aux = this.cabeza;
100     while(aux != null) {
101         contador++;
102         if(contador == i - 1){
103             aux.setSiguiente(null);
104         }
105         aux = aux.getSiguiente();
106     }
107 }
108
109 public void eliminarElprimero(){ 1 usage
110     Nodo<T> copia = new Nodo<>(this.cabeza.getSiguiente().getDato(),
111                               this.cabeza.getSiguiente().getSiguiente());
```

Run MainLista

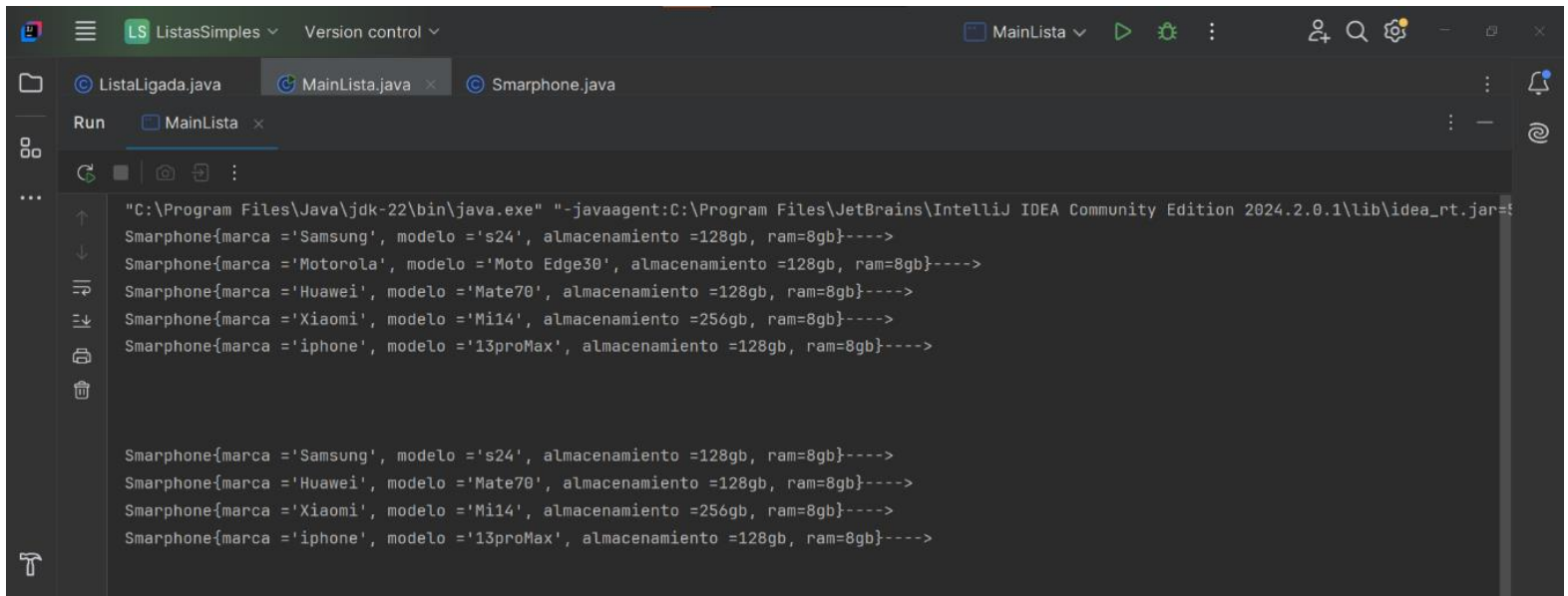
ListasSimples > src > ico > fes > aragon > unam > listas > ListaLigada 132:50 CRLF UTF-8 1 space*

Código main:

```
1 package ico.fes.aragon.unam.main;
2
3 import ico.fes.aragon.unam.clases.Smarphone;
4 import ico.fes.aragon.unam.listas.ListaLigada;
5
6 public class MainLista {
7
8     public static void main(String[] args) {
9         Smarphone sm1 = new Smarphone( marca: "Samsung", modelo: "s24", almacenamiento: 128, ram: 8);
10        Smarphone sm2 = new Smarphone( marca: "Motorola", modelo: "Moto Edge30", almacenamiento: 128, ram: 8);
11        Smarphone sm3 = new Smarphone( marca: "Huawei", modelo: "Mate70", almacenamiento: 128, ram: 8);
12        Smarphone sm4 = new Smarphone( marca: "Xiaomi", modelo: "Mi14", almacenamiento: 256, ram: 8);
13        Smarphone sm5 = new Smarphone( marca: "iphone", modelo: "13proMax", almacenamiento: 128, ram: 8);
14        // agregando los objetos a la lista
15        ListaLigada<Smarphone> ll = new ListaLigada<>();
16        ll.agregarAlfinal(sm1);
17        ll.agregarAlfinal(sm2);
18        ll.agregarAlfinal(sm3);
19        ll.agregarAlfinal(sm4);
20        ll.agregarAlfinal(sm5);
21        //imprimiendo los datos
22        ll.transversal();
23        //Eliminar el de la pocision 2
24        ll.eliminar( posicion: 2);
25        //Imprimiendo los datos despues de la eliminacion
```

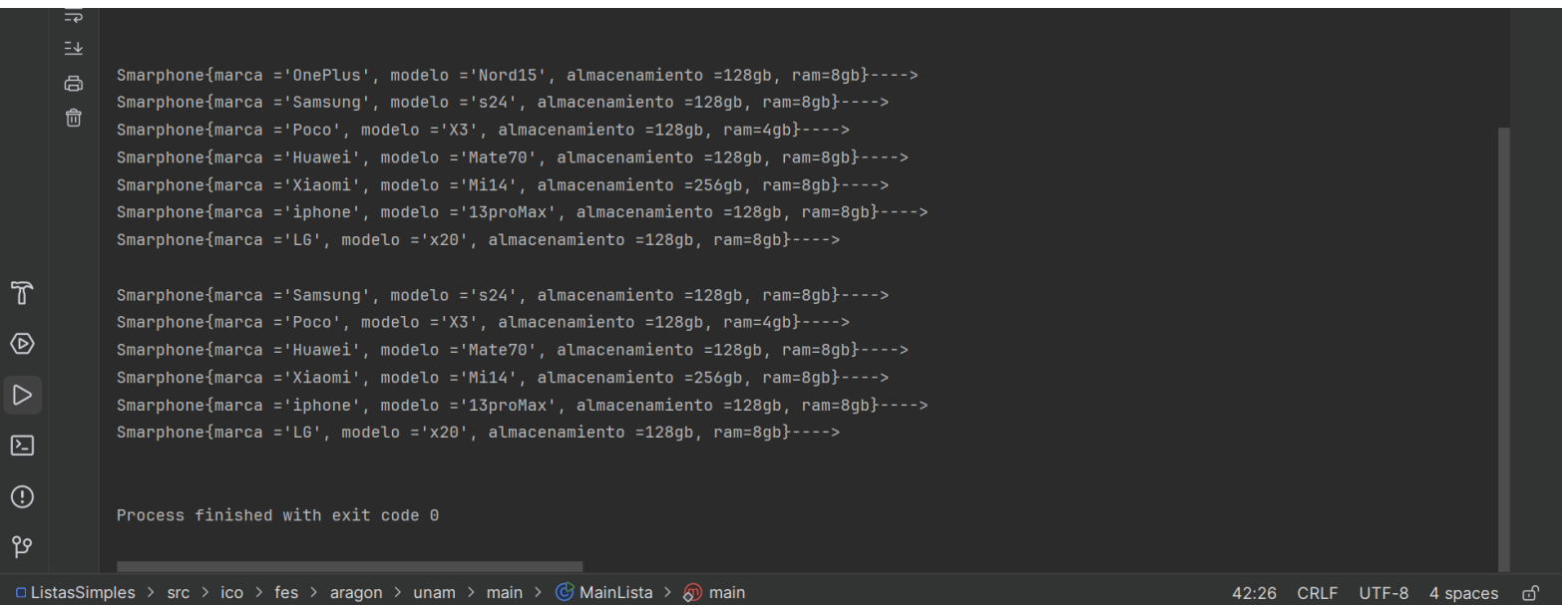
```
6     public class MainLista {
8         public static void main(String[] args) {
21            //imprimiendo los datos
22            ll.transversal();
23            //Eliminar el de la pocision 2
24            ll.eliminar( posicion: 2);
25            //Imprimiendo los datos despues de la eliminacion
26            System.out.println("\n");
27            ll.transversal();
28            //actualizar el elemento 2
29            ll.agregarDespuesde(sm1,new Smarphone( marca: "Poco", modelo: "X3", almacenamiento: 128, ram: 4));
30
31            //agregar un elemento y otro al final
32            Smarphone sm6 = new Smarphone( marca: "OnePlus", modelo: "Nord15", almacenamiento: 128, ram: 8);
33            Smarphone sm7 = new Smarphone( marca: "LG", modelo: "x20", almacenamiento: 128, ram: 8);
34            ll.agregarAlinicio(sm6);
35            ll.agregarAlfinal(sm7);
36            //volver a imprimir todo
37            System.out.println("\n");
38            ll.transversal();
39            // eliminar el primero en la lista
40            ll.eliminarElprimero();
41            // volver a imprimir
42            ll.transversal();
43        }
```

Pruebas



```
"C:\Program Files\Java\jdk-22\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.0.1\lib\idea_rt.jar=12880:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2024.2.0.1\bin" -Dfile.encoding=UTF-8
Smarphone{marca = 'Samsung', modelo = 's24', almacenamiento =128gb, ram=8gb}---->
Smarphone{marca = 'Motorola', modelo = 'Moto Edge30', almacenamiento =128gb, ram=8gb}---->
Smarphone{marca = 'Huawei', modelo = 'Mate70', almacenamiento =128gb, ram=8gb}---->
Smarphone{marca = 'Xiaomi', modelo = 'Mi14', almacenamiento =256gb, ram=8gb}---->
Smarphone{marca = 'iphone', modelo = '13proMax', almacenamiento =128gb, ram=8gb}---->

Smarphone{marca = 'Samsung', modelo = 's24', almacenamiento =128gb, ram=8gb}---->
Smarphone{marca = 'Huawei', modelo = 'Mate70', almacenamiento =128gb, ram=8gb}---->
Smarphone{marca = 'Xiaomi', modelo = 'Mi14', almacenamiento =256gb, ram=8gb}---->
Smarphone{marca = 'iphone', modelo = '13proMax', almacenamiento =128gb, ram=8gb}---->
```



```
Smarphone{marca = 'OnePlus', modelo = 'Nord15', almacenamiento =128gb, ram=8gb}---->
Smarphone{marca = 'Samsung', modelo = 's24', almacenamiento =128gb, ram=8gb}---->
Smarphone{marca = 'Poco', modelo = 'X3', almacenamiento =128gb, ram=4gb}---->
Smarphone{marca = 'Huawei', modelo = 'Mate70', almacenamiento =128gb, ram=8gb}---->
Smarphone{marca = 'Xiaomi', modelo = 'Mi14', almacenamiento =256gb, ram=8gb}---->
Smarphone{marca = 'iphone', modelo = '13proMax', almacenamiento =128gb, ram=8gb}---->
Smarphone{marca = 'LG', modelo = 'x20', almacenamiento =128gb, ram=8gb}---->

Smarphone{marca = 'Samsung', modelo = 's24', almacenamiento =128gb, ram=8gb}---->
Smarphone{marca = 'Poco', modelo = 'X3', almacenamiento =128gb, ram=4gb}---->
Smarphone{marca = 'Huawei', modelo = 'Mate70', almacenamiento =128gb, ram=8gb}---->
Smarphone{marca = 'Xiaomi', modelo = 'Mi14', almacenamiento =256gb, ram=8gb}---->
Smarphone{marca = 'iphone', modelo = '13proMax', almacenamiento =128gb, ram=8gb}---->
Smarphone{marca = 'LG', modelo = 'x20', almacenamiento =128gb, ram=8gb}---->

Process finished with exit code 0
```