

# UNIVERSIDAD TECMILENIO

Mauricio Estrada De la Garza - AL02976904  
Jaob Antonio Sandoval Ramirez (3065611)  
Ana Patricia Diosdado Trejo (2980620)  
26/01/2024  
Estructura de Datos  
Actividad 2

## Actividad 2 : Implementación Básica de Listas Ligadas

### Objetivo de la actividad:

El objetivo de esta tarea es que los estudiantes practiquen la implementación de operaciones básicas en listas ligadas en el lenguaje de programación Java.

Se espera que los alumnos refuercen sus habilidades en la manipulación de estructuras de datos mediante la implementación de funciones que añaden, buscan y eliminan elementos en una lista ligada.

### Requerimientos para la actividad:

Uso Java jdk a través de un IDE, preferentemente Netbeans.

### Instrucciones para el alumno:

Basado en el código generado durante la clase:

<https://utmedu.sharepoint.com/:f/s/EstructuradeDatos2024-1/Eq9uwPUvHHxJrwpX-BkyjN8BHc7ixGdAaQZcuavnIPA5VQ?e=PSmNaI>

[Links to an external site.](#)

### Crear las siguientes funciones:

- insertAtFirst
- insertAtEnd
- insertbyPosition
- removeFirstPosition
- removeLastPosition
- removeByPosition
- removeByData

- Find/Search
- FindByPosition
- IsExist (Si existe el dato)
- getPositionByData
- getDataByPosition

Conceptos a calificar:

**Importante:** En el caso de hacer uso de las siguiente librerías, en automático se considerará un NE:

`import java.util.ArrayList;`

`import java.util.List;`

`import java.util.LinkedList;`

`import java.util.Arrays;`

O alguna similar ya programada.

1. Definición de funciones: La correcta definición de funciones.
  1. La correcta naturaleza: void, Integer, Boolean, etc
  1. Los parámetros correctos de la función.
  1. La demostración en pantalla del correcto funcionamiento de las funciones implementadas.
  1. Funcionalidad implementada.
- 
1. Reflexión: Discutir sobre cómo desarrollaron el programa en usando lista ligadas. 🍪🍪 *Brownie points, por explicación de eficiencia.*  
🍪🍪😊😊

Entregables:

- Código Fuente, archivos con extensión .java (Recuerda, cualquier otro tipo de archivo no es permitido, i.e. .class, .zip, .jar, etc)

- **Compilación sin errores.**
- **Cumplimiento de Especificaciones.**
- **Reporte.**
- **!!!! Reflexión individual por integrante del equipo. !!!!!**
- **Recordar que solo se revisa la última entrega en canvas.**

**\* Sin reporte y/o código fuente, ningún porcentaje es considerado ⇒  
Calificación NE**

**Restricción: Equipos de no más de cinco personas, no menos de cuatro. 😊**

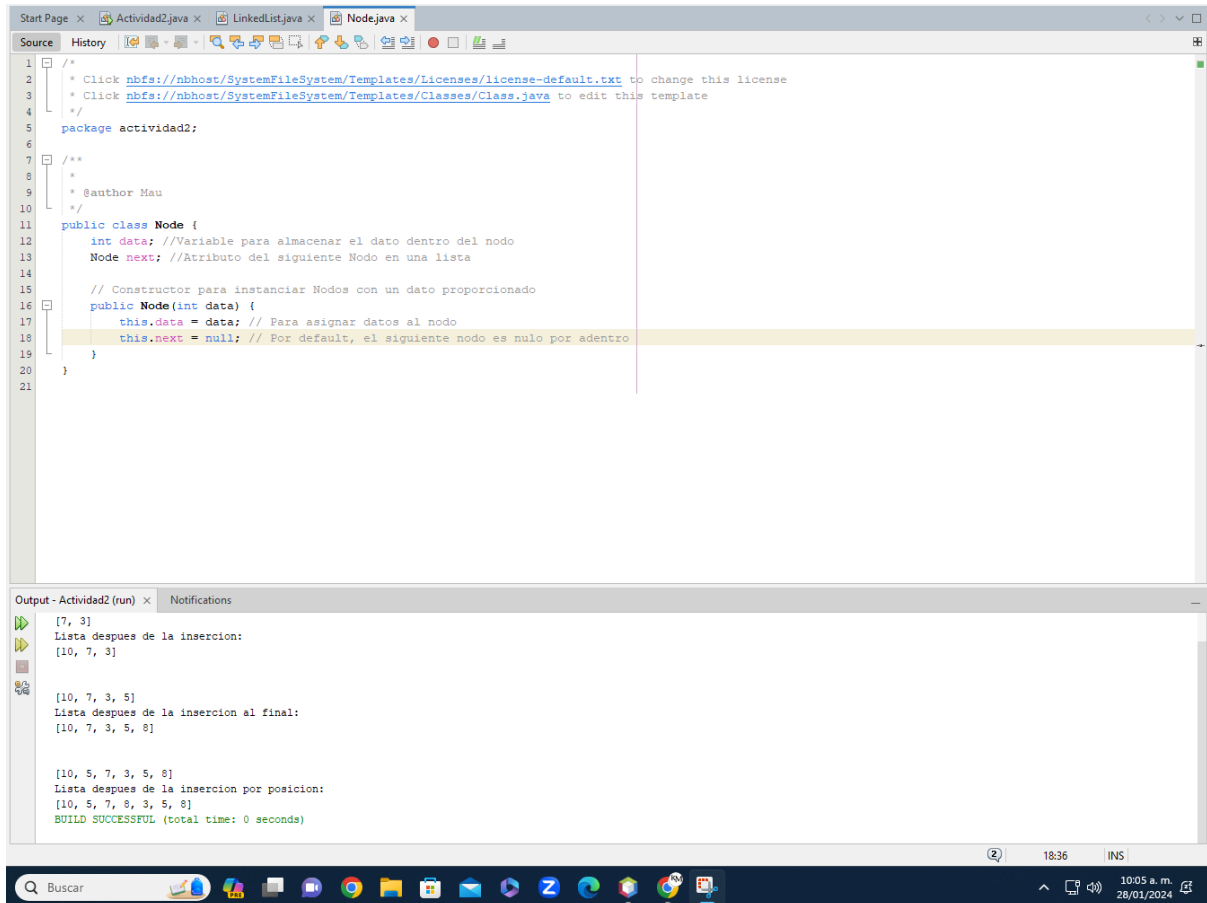
## **REPORTE:**

**Mau:**

Empecé primero por crear la clase Nodo, le implemente 2 atributos, uno de tipo int llamado data para almacenar elementos de enteros únicamente y el otro es una composición recursiva llamada "next" para hacer referencia al siguiente nodo de una lista, después está el constructor de la clase, y el next es igual a nulo ya que por default el nodo es nulo si no tiene dato contenido. Después hice una clase llamada "LinkedList" donde le implemente un atributo privado llamado "Node head" y en el constructor, la cabeza es igual a nulo. Hay 4 métodos en esta clase, el primero es el de "insertAtFirst" que tiene como parámetro el dato que tendrá el nodo adentro, lo primero que hice en el método es instanciar un objeto de la clase nodo y le puse como argumento el parámetro asignado al método. Después declare que el siguiente nuevo nodo será igual a la cabeza, y luego la cabeza es igual al nuevo Nodo. El segundo método es el de "insertAtEnd", igual es el mismo parámetro, también instancie un objeto de la clase Nodo con el mismo argumento, luego hice una condicional donde si la cabeza estaba nulo, entonces que me retornara que la cabeza es igual al nuevo nodo. Y luego declare que el nodo actual es igual a la cabeza, luego implemente un ciclo while donde si el siguiente nodo actual no es nulo, entonces que el nodo actual sea igual al siguiente nodo actual. y más abajo declare que el siguiente nodo actual es igual al nuevo nodo. El tercer método es el de "insertByPosition" los parámetros son de tipo int, uno es para el elemento de la posición en la lista y el otro es para indicar la posición del elemento en la lista, luego puse una condición donde si la posición proporcionada por el usuario es menor a cero, entonces no se podrá imprimir la instrucción indicada por el usuario, luego instancie un objeto de la clase Nodo al igual que los métodos anteriores, luego puse una condicional donde si la posición es igual a 0, entonces el siguiente nuevo nodo es igual a la cabeza y luego declare que la cabeza es igual al nuevo nodo, por lo tanto el siguiente nuevo nodo es igual al nuevo nodo. Y el nodo actual lo declare que fuera igual a la cabeza. Después implemente un ciclo For donde si el índice es menor a la posición menos Y que el nodo actual no sea nulo, entonces se imprime el índice y además el nodo actual es igual al siguiente nodo actual, después hice una condicional donde si el nodo actual es igual a nulo, entonces el dato de salida será que la posición en la lista no existe. más adelante declare que el siguiente nuevo nodo fuera igual al siguiente nodo actual y que además el siguiente nodo actual fuera igual a un nuevo nodo. El último método que implemente fue para imprimir una lista, declare que el nodo actual es igual a la cabeza, luego implemente en el algoritmo que se imprima el corchete izquierdo "[" para darle inicio a la lista, después hice un ciclo while donde si el nodo actual no está nulo, entonces se imprimen todos los elementos del nodo actual y que se igual al siguiente nuevo nodo. Después puse una condicional donde si el dónde actual no está nulo entonces se imprima la coma, y al final puse que se imprima el corchete derecho para darle final a la lista "]"

## CÓDIGO:

Mau:



The screenshot shows an IDE with three tabs: 'Start Page', 'Actividad2.java', and 'LinkedList.java'. The 'Actividad2.java' tab is active, displaying the following Java code:

```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/license-default.txt to change this license
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/Class.java to edit this template
4  */
5  package actividad2;
6
7  /**
8   *
9   * @author Mau
10  */
11  public class Node {
12      int data; //Variable para almacenar el dato dentro del nodo
13      Node next; //Atributo del siguiente Nodo en una lista
14
15      // Constructor para instanciar Nodos con un dato proporcionado
16      public Node(int data) {
17          this.data = data; // Para asignar datos al nodo
18          this.next = null; // Por default, el siguiente nodo es nulo por adentro
19      }
20  }
21
```

The 'Output - Actividad2 (run)' tab shows the following output:

```
[7, 3]
Lista despues de la insercion:
[10, 7, 3]

[10, 7, 3, 5]
Lista despues de la insercion al final:
[10, 7, 3, 5, 8]

[10, 5, 7, 3, 5, 8]
Lista despues de la insercion por posicion:
[10, 5, 7, 8, 3, 5, 8]
BUILD SUCCESSFUL (total time: 0 seconds)
```

The bottom of the screen shows a Windows taskbar with the search bar, taskbar icons, and system tray showing the time as 10:05 a.m. on 28/01/2024.

```
Start Page x Actividad2.java x LinkedList.java x Node.java x
Source History
115 public void display() {
116
117     //Nodo actual = Cabeza del Nodo
118     Node current = head;
119
120     System.out.print(": ");
121
122     //Ciclo:
123     //Mientras el nodo actual no este nulo...
124     while (current != null) {
125
126         //Se imprimira los datos del nodo actual
127         System.out.print(": " + current.data);
128
129         //Y el nodo actual sera igual al siguiente nodo actual
130         current = current.next;
131
132         //Condional mientras el nodo actual no sea nulo
133         if (current != null) {
134             //Se imprimira una coma despues del elemento insertado
135             System.out.print(", ");
136         }
137     }
138     System.out.println(": ");
139 }
140
141
142 }
```

Output - Actividad2 (run) x Notifications

```
[7, 3]
Lista despues de la insercion:
[10, 7, 3]

[10, 7, 3, 5]
Lista despues de la insercion al final:
[10, 7, 3, 5, 8]

[10, 5, 7, 3, 5, 8]
Lista despues de la insercion por posicion:
[10, 5, 7, 8, 3, 5, 8]
BUILD SUCCESSFUL (total time: 0 seconds)
```

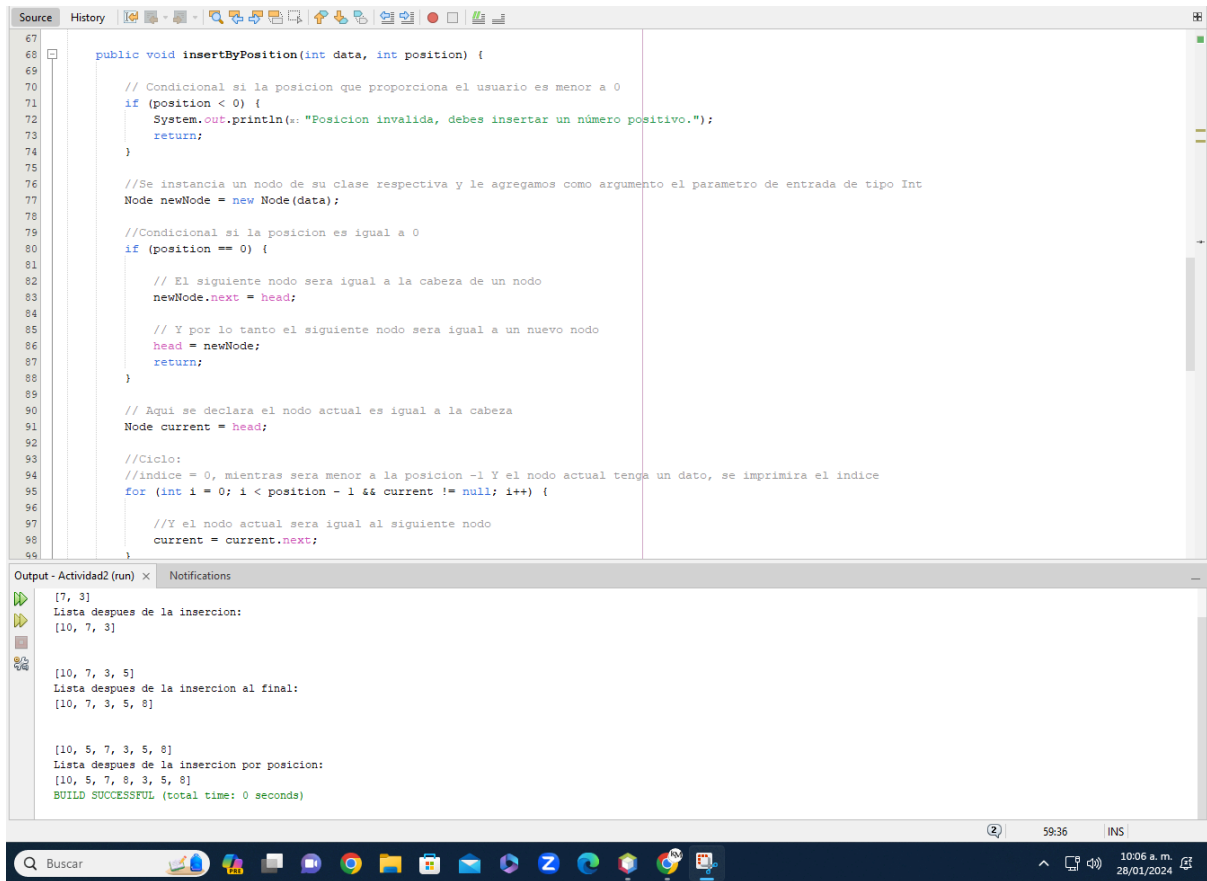
```
Start Page x Actividad2.java x LinkedList.java x Node.java x
Source History
94 //indice = 0, mientras sera menor a la posicion -1 Y el nodo actual tenga un dato, se imprimira el indice
95 for (int i = 0; i < position - 1 && current != null; i++) {
96
97     //Y el nodo actual sera igual al siguiente nodo
98     current = current.next;
99 }
100
101 //Condional si el nodo actual es igual a nulo
102 if (current == null) {
103     System.out.println(": Posición fuera del rango, No se puede insertar en la posición especificada.");
104     return;
105 }
106
107 //El siguiente nuevo nodo es igual al siguiente nodo actual
108 newNode.next = current.next;
109
110 //Y por lo tanto el siguiente nodo actual es igual a un nuevo nodo
111 current.next = newNode;
112 }
113
114 //Metodo para imprimir una lista
115 public void display() {
116
117     //Nodo actual = Cabeza del Nodo
118     Node current = head;
119
120     System.out.print(": ");
121
122     //Ciclo:
123     //Mientras el nodo actual no este nulo...
124     while (current != null) {
125
126         //Se imprimira los datos del nodo actual
127         System.out.print(": " + current.data);
128
129         //Y el nodo actual sera igual al siguiente nodo actual
130         current = current.next;
131
132         //Condional mientras el nodo actual no sea nulo
133         if (current != null) {
134             //Se imprimira una coma despues del elemento insertado
135             System.out.print(", ");
136         }
137     }
138     System.out.println(": ");
139 }
140
141
142 }
```

Output - Actividad2 (run) x Notifications

```
[7, 3]
Lista despues de la insercion:
[10, 7, 3]

[10, 7, 3, 5]
Lista despues de la insercion al final:
[10, 7, 3, 5, 8]

[10, 5, 7, 3, 5, 8]
Lista despues de la insercion por posicion:
[10, 5, 7, 8, 3, 5, 8]
BUILD SUCCESSFUL (total time: 0 seconds)
```



Start Page x Actividad2.java x LinkedList.java x Node.java x

Source History

```
37
38 //Metodo para insertar un elemento al final de una lista
39 public void insertAtEnd(int data) {
40
41     //Objeto instanciado de la clase Nodo que tiene como argumento
42     //el parametro de tipo int declarado en el metodo
43     Node newNode = new Node(data);
44
45     //Condicional:
46     //Si la cabeza no tiene ningun dato, entonces regresa un nuevo nodo
47     if (head == null) {
48         head = newNode;
49         return;
50     }
51
52     //El nodo actual sera igual a la cabeza del nodo
53     Node current = head;
54
55     //Ciclo:
56     //Mientras el nodo siguiente no sea nulo...
57     //Entonces el nodo actual sera igual al nodo siguiente
58     while (current.next != null) {
59         current = current.next;
60     }
61
62     //Y por lo tanto, el nodo actual sera igual a un nuevo nodo
63     current.next = newNode;
64 }
65
66 //Metodo para agregar un elemento en cualquier posicion de una lista
67
68 public void insertByPosition(int data, int position) {
69
```

Output - Actividad2 (run) x Notifications

```
[7, 3]
Lista despues de la insercion:
[10, 7, 3]

[10, 7, 3, 5]
Lista despues de la insercion al final:
[10, 7, 3, 5, 8]

[10, 5, 7, 3, 5, 8]
Lista despues de la insercion por posicion:
[10, 5, 7, 8, 3, 5, 8]
BUILD SUCCESSFUL (total time: 0 seconds)
```

25:42 INS

Start Page x Actividad2.java x LinkedList.java x Node.java x

Source History

```
4
5 package actividad2;
6
7 /**
8  *
9  * @author Mau
10 */
11
12 public class LinkedList {
13     private Node head; // Atributo para la cabeza del nodo
14
15     // La cabeza del nodo es el perimetro de su circulo
16
17
18
19     //Constructor de la clase
20     public LinkedList() {
21         this.head = null; // Se indica que por default la cabeza esta en nulo
22     }
23
24     //Metodo para insertar un elemento desde la posicion 0 de una lista
25     public void insertAtFirst(int data) {
26
27         //Objeto instanciado de la clase Nodo que tiene como argumento
28         //el parametro de tipo int declarado en el metodo
29         Node newNode = new Node(data);
30
31         //El siguiente nodo declarado sera igual a la cabeza de un nodo
32         newNode.next = head;
33
34         //La cabeza sera igual a un nuevo nodo, por lo tanto un nuevo nodo es un nodo
35         head = newNode;
36     }
37
```

Output - Actividad2 (run) x Notifications

```
[7, 3]
Lista despues de la insercion:
[10, 7, 3]

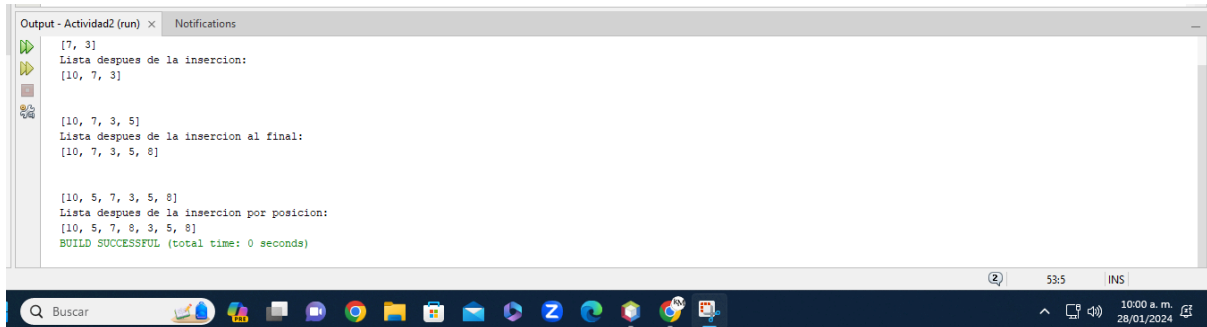
[10, 7, 3, 5]
Lista despues de la insercion al final:
[10, 7, 3, 5, 8]

[10, 5, 7, 3, 5, 8]
Lista despues de la insercion por posicion:
[10, 5, 7, 8, 3, 5, 8]
BUILD SUCCESSFUL (total time: 0 seconds)
```

122:17 INS

## PRUEBAS:

**Mau:**



```
Output - Actividad2 (run) x Notifications
[7, 3]
Lista despues de la insercion:
[10, 7, 3]

[10, 7, 3, 5]
Lista despues de la insercion al final:
[10, 7, 3, 5, 8]

[10, 5, 7, 3, 5, 8]
Lista despues de la insercion por posicion:
[10, 5, 7, 8, 3, 5, 8]
BUILD SUCCESSFUL (total time: 0 seconds)
```

## CONCLUSIÓN:

- **Mau:** La verdad si me confundí bastante al principio para poder entender bien cómo funcionaba el algoritmo y cómo podía implementar lo que vi de teoría sobre los nodos el semestre anterior a realmente aplicarlo en Programación.



## CÓDIGO:

```
13 List personas = new List();
14
15 Persona persona1 = new Persona( name: "Paco", edad: 50);
16 Node newNode = new Node(persona1);
17 Node newNode2 = new Node( i: 2);
18 Node newNode3 = new Node( i: "hola mundo!!");
19
20 personas.insert(persona1);
21 personas.insert(newNode2);
22 personas.insert( i: "AAA");
23 personas.insert( i: true);
24 personas.insert( i: " Hola mundo ");
25 personas.insert( i: " a ");
26 personas.insert( i: 234.88);
27 personas.insertToEnd( i: " END ");
28 personas.insertToStart( i: " START");
29 personas.insertByPosition( n: 2, i: " POSITION");
30
31 System.out.println(personas.getPositionByData( i: " a "));
32
```

ListaGenerica x

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\
7
La lista mide: 10
Los valores de la lista son: ->[ STARTPersona@6e8cf4c6Node@12edcd21AAA PO
Process finished with exit code 0
```

En esta parte del código se pueden ver el uso correcto e implementación de las primeras funciones insert y también el cómo funciona el getPositionBydata.

```
19
20     personas.insert(persona1);
21     personas.insert(newNodo2);
22     personas.insert(i: "AAA");
23     personas.insert(i: true);
24     personas.insert(i: "Hola mundo ");
25     personas.insert(i: " a ");
26     personas.insert(i: 234.88);
27     personas.insertToEnd(i: " END ");
28     personas.insertToStart(i: " START");
29     personas.insertByPosition(n: 2, i: " POSITION");
30     personas.removeFirstPosition();
31     personas.removeLastPosition();
32     personas.removeByPosition(n: 4);
33     personas.removeByData(i: "Hola mundo ");
34
35     System.out.println(personas.getPositionByData(i: " a "));
36
```

ListaGenerica x

⌂ ⋮

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\Intel
4
La lista mide: 6
Los valores de la lista son: ->[Persona@6e8cf4c6Node@12edcd21AAA POSITION a 234.88]

Process finished with exit code 0
```

Aquí ya se puede observar como se cambia la lista después de usar las funciones remove para quitar elementos, también así actualizando el tamaño de la lista.

```
30     personas.removeFirstPosition();
31     personas.removeLastPosition();
32     personas.removeByPosition( n: 4);
33     personas.removeByData( i: " Hola mundo ");
34     Node nodoEncontrado = (Node) personas.find( i: " POSITION");
35
36
37     if (nodoEncontrado != null) {
38         System.out.println("Nodo encontrado: " + nodoEncontrado.data);
39     } else {
40         System.out.println("El valor no se encuentra en la lista.");
41     }
42
43
44     System.out.println(personas.getPositionByData( i: " a "));
45
46
47
48
49
```

ListaGenerica x

"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea\_rt.jar=12738:C:\Program Files\JetBrains\IntelliJ IDEA\bin" -Didea.config.path=C:\Program Files\JetBrains\IntelliJ IDEA\conf -Didea.home.path=C:\Program Files\JetBrains\IntelliJ IDEA\bin -Didea.platform.prefix=Java21 -jar C:\Program Files\JetBrains\IntelliJ IDEA\bin\idea.jar

Nodo encontrado: POSITION

4

La lista mide: 6

Los valores de la lista son: ->[Persona@6e8cf4c6Node@12edcd21AAA POSITION a 234.88]

Process finished with exit code 0

Acá podemos ver el uso de find encontrando de manera correcta el Nodo deseado.

```
34 Node nodoEncontrado = (Node) personas.find( i: " hola");
35
36
37 if (nodoEncontrado != null) {
38     System.out.println("Nodo encontrado: " + nodoEncontrado.data);
39 } else {
40     System.out.println("El valor no se encuentra en la lista.");
41 }
42
43
44 System.out.println(personas.getPositionByData( i: " a "));
45
46
47
48
49
```

ListaGenerica x

"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA\lib\idea\_rt.jar=12739:C:\Program Files\Java\jdk-21\bin" -Dfile.encoding=UTF-8

El valor no se encuentra en la lista.

4

La lista mide: 6

Los valores de la lista son: ->[Persona@6e8cf4c6Node@12edcd21AAA POSITION a 234.88]

Process finished with exit code 0

Y acá de manera donde el método fallo al no encontrar el nodo buscado.

```
41     }
42
43     Node nodoFindByPosition = (Node) personas.findByPosition( n: 0);
44     if (nodoFindByPosition != null) {
45         System.out.println("Nodo encontrado: " + nodoFindByPosition.data);
46     } else {
47         System.out.println("El valor no se encuentra en la lista.");
48     }
49
50
51
52     System.out.println(personas.getPositionByData( i: " a "));
53

```

ListaGenerica x

```
"C:\Program Files\Java\jdk-21\bin\java.exe" "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA
El valor no se encuentra en la lista.
Nodo encontrado: Persona@5b480cf9
4
La lista mide: 6
Los valores de la lista son: ->[Persona@5b480cf9Node@12edcd21AAA POSITION a 234.88]

Process finished with exit code 0
```

Ya en esta parte del código podemos ver el uso también del `findByPosition` en este caso funcionando de manera correcta al lograr encontrar el nodo y en dado caso donde se proporcione un valor mayor a dado por la lista dira que el valor no se encuentra en la lista.

```
49         if(personas.isExist( i: "AAA"))
50             System.out.println("sTrue");
51         else{
52             System.out.println("False");
53         }
54
55
56
57         System.out.println(personas.getPositionByData( i: "
58
59
60
61
62
63
64
```

ListaGenerica x

El valor no se encuentra en la lista.  
 Nodo encontrado: Persona@5b480cf9  
 True  
 4  
 La lista mide: 6  
 Los valores de la lista son: ->[Persona@5b480cf9Node@12edcd21AAA  
 Process finished with exit code 0

Ya en esta parte se ve el correcto funcionamiento de la función isExist ya que se encontró que el valor dado se encuentra en la lista.

```
48 }
49 if(personas.isExist( i: "AA"))
50     System.out.println("sTrue");
51 else{
52     System.out.println("False");
53 }
54
55
56
57 System.out.println(personas.getPositionByDat
58
59
60
61
62
63
64
```

ListaGenerica x

El valor no se encuentra en la lista.  
Nodo encontrado: Persona@5b480cf9  
False  
4  
La lista mide: 6  
Los valores de la lista son: ->[Persona@5b480cf9Node@12edc

Y aquí un caso donde no se encontró.

```
53     }
54     System.out.println(personas.getDataByPosition(n: 3));
55
56
57
58     System.out.println(personas.getPositionByData(i: " a "));
59
60
61
```

ListaGenerica x

Nodo encontrado: Persona@5b480cf9  
False  
POSITION  
4  
La lista mide: 6  
Los valores de la lista son: ->[Persona@5b480cf9Node@12edcd21AAA POSITION a 234.88]

Y ya por último un caso donde se encuentran las ultimas funciones get con su correcto funcionamiento.

#### CONCLUSIÓN:

- Jaob: En esta actividad tuve la oportunidad de entender el funcionamiento interno de las listas en Java dándome una visión más amplia de como se acomoda todo el programa internamente, y como se logra optimizar el uso de memoria para guardar objetos u otros datos necesarios.
- Ana: Por mi parte encontré bastante complicado comprender más que nada el funcionamiento correcto de cada función. Entendía lo que trataba de hacer, pero al momento de tratar de ejecutarlo siempre salía algo mal. Fue bastante complicado, pero con la práctica espero ganar más experiencia en cuanto a la implementación de funciones para las listas.