**PRÁCTICA 17 | LISTAS DOBLEMENTE LIGADAS**

# Introducción

Ya se hablaba en prácticas anteriores que las listas simples ligadas eran de gran utilidad, siento que las listas doblemente ligadas son una optimización de las anteriores, para poder realizar operaciones sobre esta estructura de datos de manera más ágil e incluso aumentar su funcionalidad. Una de las grandes ventajas será el poder retroceder entre los nodos y, por ende, la posibilidad de hacer una operación partiendo del final hacia el inicio.

# Desarrollo

En una práctica anterior, donde se creó la clase **ListaOrdenada**, para ser concreto, ya se habían implementado los métodos que posee la interfaz **Lista.** El código de los métodos heredados de esta misma interfaz en la clase principal de esta práctica **(ListaDoblementeLigada)** es un tanto parecido, pero no en su totalidad. A diferencia de ListaOrdenada la nueva clase sí permite elementos repetidos. Además, recordemos que ya no se está trabajando con una Lista simple, ahora se tiene que considerar también la liga izquierda de cada nodo.

El proceso de programación de los métodos de **ListaDoblementeLigada** no me causó tantos problemas, a excepción de unos pocos métodos como lo fueron: agregar y eliminar. El problema principal radicó en que no estaba considerando las ligas izquierdas de los Nodos, solamente las derechas. Aparte, en un método interno que hice para la clase, no estaba trasladando la iteración de la lista hacia la siguiente liga, sino que se quedaba en una misma posición todo el tiempo, causando que se ciclara el programa cuando hacía uso de esa función.

Como se mencionó anteriormente, la clase **ListaOrdenada** fue la anfitriona de la práctica. Sobre ella, se hizo la mayoría del código, haciendo las correspondientes implementaciones provenientes de los métodos de la interfaz **Lista**. En cambio, el resto de líneas de código quedó para las típicas pruebas del funcionamiento de lo que se pide en la práctica **(PruebaListaDobleLigada)**.

En esta documentación, a diferencia de las anteriores, se mencionarán solamente los archivos que fueron creados para el desarrollo de la práctica junto con el paquete donde están situados. Para efectos de comodidad de visión incluyo en esta ocasión un “diagrama de clases”, donde se aprecian los métodos y atributos que están en la clase y un poco de cómo está estructurada esta sección del proyecto. De ahora en adelante, si el tiempo lo permite, estaré incluyendo el diagrama en cada práctica.

A continuación, se indican los archivos creados y el paquete donde se encuentran:

## ListaDobleLigada (paquete edlineal).

## PruebaCentroComputo.java (paquete interfaces).

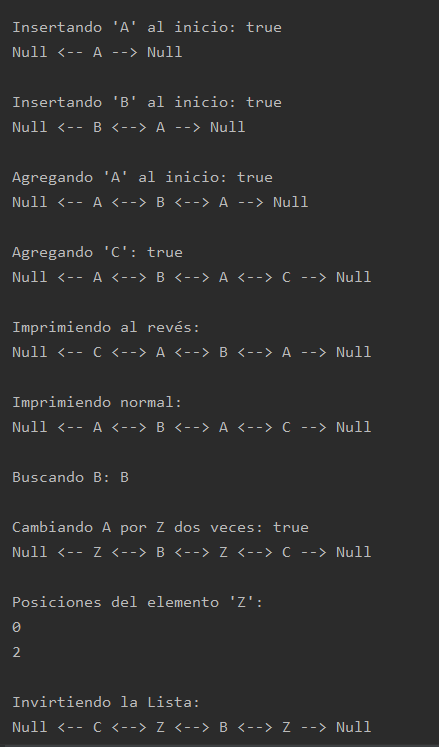
La clase anterior es donde se encuentran las pruebas de la práctica.

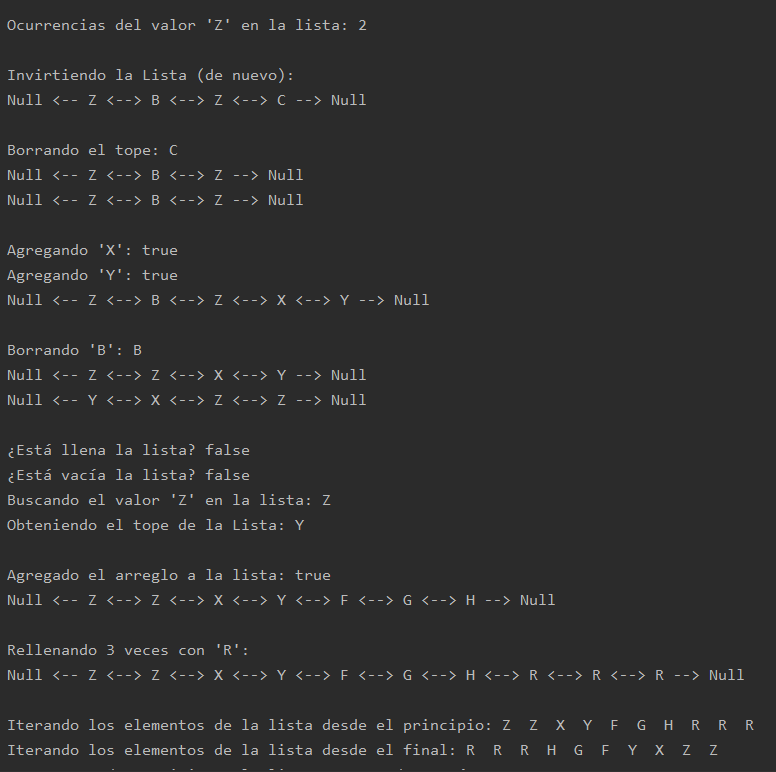
# D:\Users\karlo\Downloads\Blank Diagram.pngDiagrama de clases de la estructura de la práctica

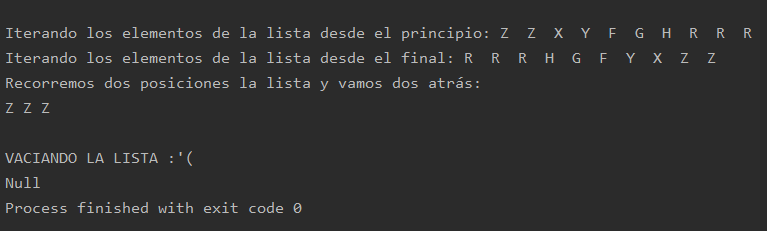
**Para mayor información sobre los métodos en la clase creada vaya a las siguiente ruta dentro del proyecto:**

* **./docs/edlineal/ListaDobleLigada.html**

# Capturas del programa funcionando







# Conclusiones

Las listas simples ligadas ya habían sido de gran utilidad en la práctica anterior para poder almacenar los datos y recuperarlos. Ahora con este nuevo tipo de lista, además de que fue más fácil programar algunos métodos, también hay la posibilidad de optimizar un poco más las operaciones que se hacen a los datos. También, es más fácil iterar los elementos en la Lista existiendo la posibilidad de ir hacia adelante o hacia atrás en los valores contenidos según sea el caso.

Definitivamente, una Lista doblemente ligada es una gran mejora respecto de una simplemente ligada.