

UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO
Facultad de Ciencias



Estructuras de Datos

Práctica 7: Lista doblemente ligada

Profesora:

Amparo López Gaona

Ayudante: Adrián Aguilera Moreno

Ayudante de Laboratorio: Kevin Jair Torres Valencia

Objetivo

Desarrollar una lista doblemente ligada en Java, implementando los métodos de inserción, acceso, búsqueda y eliminación de elementos.

Introducción

Las estructuras de datos son fundamentales en la programación, ya que permiten organizar y manipular la información de manera eficiente. Una de las estructuras más utilizadas es la lista simplemente ligada, la cual ofrece una forma dinámica de almacenar elementos sin la necesidad de definir un tamaño fijo desde el inicio.

Una lista doblemente ligada es una estructura de datos en la que cada elemento, o nodo, tiene dos punteros (o referencias): uno hacia el siguiente nodo y otro hacia el nodo anterior. Esta característica permite que la lista pueda ser recorrida en ambas direcciones: de adelante hacia atrás y de atrás hacia adelante.

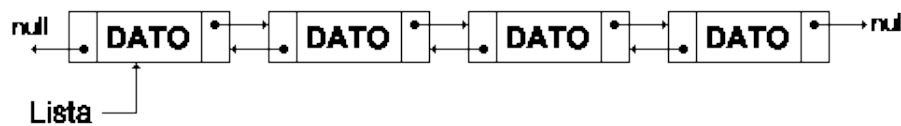


Figure 1: Lista doblemente ligada con cuatro nodos.

Cada nodo de una lista doblemente ligada contiene tres partes:

- Dato: El valor o la información que almacena el nodo.
- Puntero al siguiente nodo: Una referencia al siguiente nodo en la lista (si existe).
- Puntero al nodo anterior: Una referencia al nodo anterior en la lista (si existe).

Esta estructura de datos es útil cuando se necesita realizar inserciones y eliminaciones en cualquier parte de la lista de manera eficiente, ya que permite moverse fácilmente entre nodos adyacentes.

Algunas de la ventajas de una lista doblemente ligada son:

- Se pueden recorrer en ambas direcciones (de principio a fin y de fin a principio).
- Se pueden realizar operaciones de inserción y eliminación más eficientes, ya que puedes acceder fácilmente al nodo anterior o siguiente.

Desarrollo

En esta práctica, se les proporcionará lo siguiente:

- La interfaz `Coleccion<T>`, que define los métodos básicos para manejar una colección genérica.
- La clase `ListaDobleLigada<T>` con la estructura básica y la definición de la clase interna `Nodo`.
- La clase `IteradorListaDoble`, que servirá para implementar la funcionalidad de recorrido de la lista.

Por lo que la implementación queda de la siguiente forma:

- El constructor de `ListaDobleLigada<T>` para inicializar los atributos `cabeza`, `rabo` y `longitud`.
 - La implementación de los métodos de la interfaz `Coleccion<T>`:
 - `insertar(T elemento)`: Agregar un nuevo nodo al final de la lista.
 - `eliminar(int indice)`: Eliminar el nodo en la posición indicada.
 - `acceder(int indice)`: Retornar el elemento almacenado en la posición indicada.
 - `buscar(T elemento)`: Retornar `true` si el elemento está en la lista, `false` en caso contrario.
 - `vaciar()`: Eliminar todos los elementos de la lista.
 - Para observar que si funciona correctamente la lista, implementen el método `imprimirInverso()` para imprimir la lista en orden inverso.
 - Asegúrense de manejar excepciones cuando sea necesario, por ejemplo, si el índice está fuera de rango.
 - Finalmente ejecuta el archivo `Main.java` y verifica que los resultados sean correctos.
- Extra:** Desarrollen un método para ordenar las películas por el año dentro de la clase `ListaDobleLigada` así como mostrar su funcionamiento en la clase principal.

Formato de Entrega

1. Las prácticas se entregarán en parejas.
2. NO incluir los archivos .class dentro de la carpeta.
3. Los archivos de código fuente deben estar documentados.
4. Se pueden discutir y resolver dudas entre los integrantes del grupo. Pero cualquier práctica plagiada total o parcialmente será penalizada con cero para los involucrados.
5. La práctica se debe subir al Github Classroom correspondiente.
6. La entrega en classroom debe contener el link HTTPS y SSH de su repositorio y es lo único que se debe entregar.
7. El horario y día de entrega se acordará en la clase de laboratorio y no deberá sobrepasar 2 clases de laboratorio.