

UNIVERSIDAD NACIONAL AUTÓNOMA DE
MÉXICO
Facultad de Ciencias



Estructuras de Datos

Práctica 6: Lista ligada parte II

Profesora:

Amparo López Gaona

Ayudante: Adrián Aguilera Moreno

Ayudante de Laboratorio: Kevin Jair Torres Valencia

Objetivo

El objetivo de esta práctica es que los estudiantes implementen una lista simplemente ligada en Java, partiendo de la interfaz `Coleccion<T>` desarrollada en la práctica anterior. Los estudiantes deberán completar los métodos definidos en la clase `ListaLigada<T>`, incluyendo el iterador para recorrer la lista.

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 ligada es una agrupación de objetos denominados nodos, no necesariamente almacenados en forma adyacente. Cada nodo contiene el elemento (o dato) y un enlace o liga a su sucesor. El último elemento tendrá un enlace con valor null para indicar que no tiene sucesor.

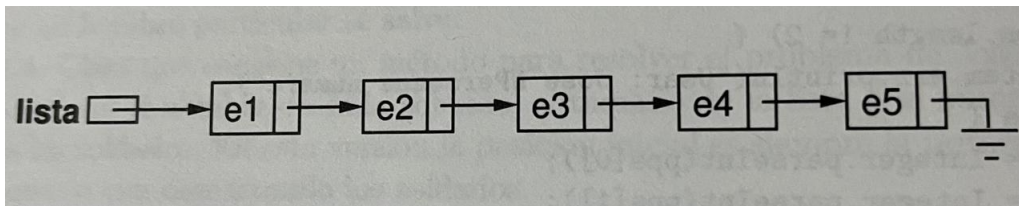


Figure 1: Lista ligada con cinco nodos.

Una lista ligada suele tener un principio y un fin, pero no se puede llegar en tiempo constante a cualquier elemento por su posición. El acceso a los elementos en la lista es secuencial, es decir, para acceder a cualquier elemento es necesario recorrer secuencialmente los elementos en la lista hasta llegar al elemento deseado.

Las listas ligadas, como se han descrito, pueden variar al implementarse, si se decide tener un apuntador al inicio de la lista, un apuntador al final de la lista, si la lista terminará con el valor null, como se mostro en la figura, o bien con un nodo especial, si se usará una sola liga o bien dos.

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 `ListaLigada<T>` con la estructura básica y la definición de la clase interna `Nodo`.
- La clase `IteradorListaSimple`, que servirá para implementar la funcionalidad de recorrido de la lista.

Por lo que la implementación queda de la siguiente forma:

- Completar la clase `IteradorListaSimple` implementando los métodos `hasNext()` y `next()`.
- El constructor de `ListaLigada<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.
- Finalmente, para facilitar la manipulación de la lista, también deberán implementar los siguientes métodos.
 - `darLongitud()`
 - `darCabeza()`
 - `darCabezaElemento()`
 - `getRabo()`
 - `getRaboElemento()`

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.