

Introducción a Sistemas Complejos: Java, MVN y GIT

Sarah Camila Vieda Castro

Enero 2020

1 Introducción

Una estructura de datos es una colección de valores, la relación que existe entre estos valores y las operaciones que podemos hacer sobre ellos. Las estructuras de datos nos permiten resolver un problema de manera más sencilla gracias a que las reglas que las rigen nunca cambian. Una lista es aquella estructura que representa un número contable de elementos ordenados donde un mismo valor puede repetirse y considerarse un valor distinto a otro.

2 Arquitectura Ejercicio Propuesto

En la linked list encontramos que cada nodo que esta en la lista contiene un campo que contiene el valor que deseamos almacenar. Cada nodo contiene al menos un apuntador a otro nodo, uno que apunta a un nodo siguiente y otro que apunta a un nodo previo formando así una Linked List.

En el programa que se diseñó, elaboré tres clases, las cuales se dividen en las diferentes funciones que tiene este problema. La clase principal de este programa es ComplexSystem en donde se implementaron las operaciones media y desviación estándar para aplicarlas a la Linked List que se creará, los métodos creados para realizar estas operaciones son retornaMedia() y desviacionEstandar().

La segunda clase es Nodo en donde sus atributos son el valor que posee y su "siguiente" Nodo. Realicé la implementación del método equals el cual permite comparar dos Nodos.

Y tenemos la clase LinkedL la cual implementa la clase Lista, esta clase es una implementación propia y fue hecha con los nodos base, los cuales guardan el valor que contienen y el siguiente nodo a apuntar.

Los métodos que se implementaron en esta clase son:

- insertNodo() Método que inserta un nodo en la LinkedList.

- deleteNodo() Método que elimina un nodo en la LinkedList.
- getActual() Método que retorna el nodo actual sobre el cual estamos "parados".
- getPrimero() Método que retorna el nodo que se encuentra en la primera posición de la Linked List.
- getUltimo() Método que retorna el nodo que se encuentra en la última posición de la Linked List.
- getLen() Método que retorna la longitud de la Linked List.
- setNodoActual() Método que cambia el nodo actual por el nuevo nodo que entra en la Linked List.

Se realizaron las siguientes pruebas en la clase ComplexSystemTest:

- mediaLinkedList() Esta prueba calcula la media de los nodos que conforman la Linked List 'lis'.
- desviacionEstandarLinkedList() Esta prueba calcula la varianza de todos los datos de la Linked List 'lis' y luego la raíz cuadrada de esta.
- deberiaInsertar() Esta prueba busca insertar cada uno de los nodos en la Linked List 'lis'
- deberiaEliminar() Esta prueba busca eliminar cada uno de los nodos en la Linked List 'lis'