

INTRODUCCIÓN A SISTEMAS COMPLEJOS: JAVA, MAVEN Y GIT

Federico Barrios Meneses

January 2021

1 Introducción

Haciendo uso de Maven y GIT, se desarrollo una aplicación Java cuyo propósito es calcular la media y la desviación estándar de un conjunto de n números reales y que tiene como requerimiento realizar una implementación propia de una lista enlazada (Linked List). El objetivo de utilizar éstas herramientas es estandarizar el ciclo de vida de la construcción de software y tener un control de versiones cada vez que se realiza un cambio en el archivo que sirva de respaldo en caso de cualquier eventualidad.

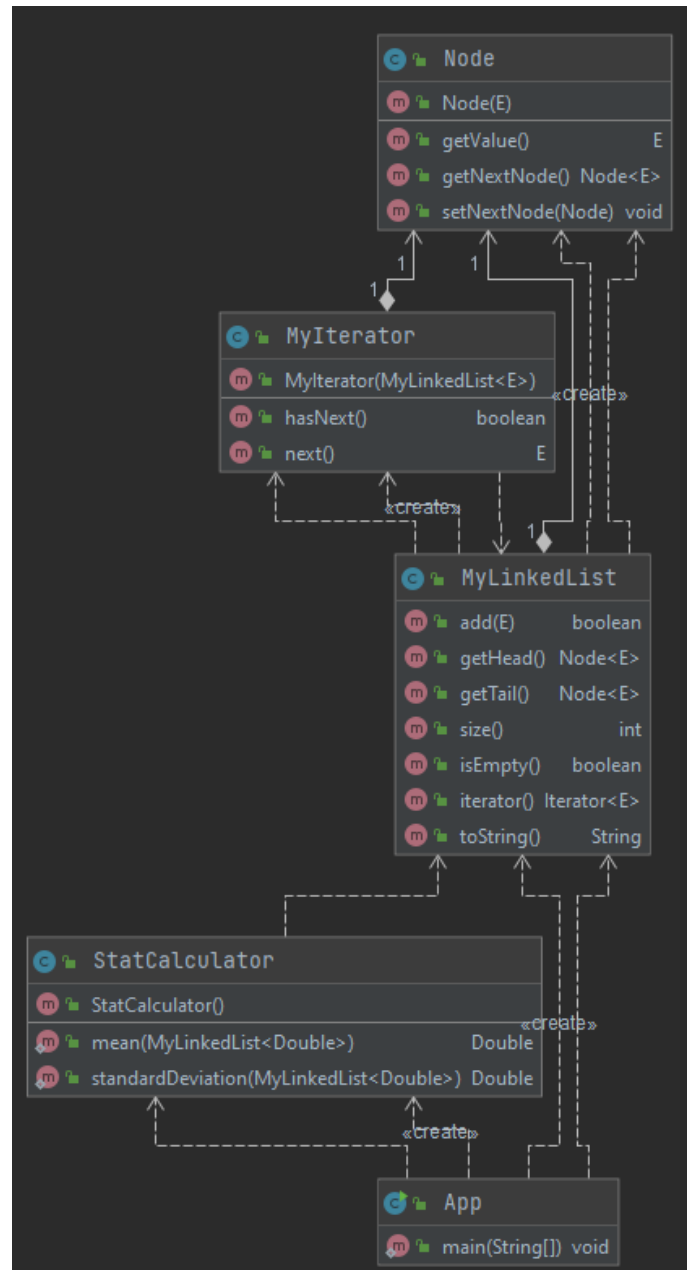
2 Diseño

Para desarrollar esta aplicación, teniendo en cuenta sus requerimientos, se dividió el programa en cinco clases principales:

- **App** es la clase que ejecuta la aplicación, también escanea el archivo de entrada.
- **StatCalculator** es la clase que realizará el calculo de la media y la desviación estándar del conjunto de números.
- **MyLinkedList** es la clase que crea el contenedor dónde se almacenarán los n números reales.
- **MyIteratr**, ya que MyLinkedList es una implementación propia, no existe un iterador por defecto en Java que entienda e itere sobre el tipo de arreglo que se crea con dicha clase, por lo cual MyIterator es un iterador personalizado para la clase MyLinkedList.
- **Node** es la clase que crea contenedores individuales para cada uno de los números dentro de la lista.

2.1 Diagrama de clases

Teniendo en cuenta lo anterior, se realizó el siguiente diagrama de clases:



2.2 Descripción del diseño

- **¿Cómo se calculó la media?** Para calcular la media se recorrió el arreglo de números con un ciclo for y se realizó la suma total de todos ellos, para así dividir el resultado entre la cantidad de números dentro del arreglo, para así retornar la media.
- **¿Cómo se calculó la desviación estándar?** Para calcular la desviación estándar se recorrió el arreglo de números con un ciclo for para hacer la sumatoria de el cuadrado de cada uno de ellos menos la media. Una vez hecha esta sumatoria, se dividió el resultado entre la cantidad de números dentro del arreglo menos uno, para luego calcular la raíz cuadrada del valor resultante, para así retornar la desviación estándar.
- **Implementación de los nodos:** Para que la linked list funcione, era necesario crear un tipo de objeto que almacene el valor de un solo número y una referencia al número siguiente dentro de la lista. Por esto se creó la clase Node. Cuyas únicas funcionalidades son guardar un valor, asignar y guardar una referencia a otro nodo.
- **Implementación de la LinkedList** Para realizar una linked list desde cero, se creó una referencia al primer nodo y otra al último nodo, para que esta lista funcione es indispensable el método add e iterator. Para el método add se agrega un nodo al final del arreglo cambiando la referencia a la cola, es decir, al último nodo y para el método iterator se necesitaba un iterador personalizado para nuestro tipo de arreglo, por lo cual se creó una clase nueva.
- **Implementación del iterador:** Para que se iterara sobre un objeto de tipo MyLinkedList, se realizó un método que retorne el valor siguiente y un método que indique si existe un valor en el nodo siguiente.