Taller 1 - Introducción Maven-Git

Calderón Ortega Andrés Mateo

Arquitecturas Empresariales
Ingeniería de sistemas, Escuela Colombiana de Ingeniería Julio Garavito, Bogota, Colombia

27 de enero de 2021

Resumen: Se realizará la explicación del funcionamiento de una aplicación que realiza 2 cálculos estadísticos como es la media y la desviación estándar; el desarrollo guiado a una programación Orientada a Objetos, su arquitectura básica y la estructura de datos utilizada para el almacenamiento de los datos durante el procesamiento de los resultados.

1. Introducción

EL problema tratado en este proyecto se basa en un problema estadístico; donde se debe calcular la media y la desviación estándar de un conjunto n de números; además de esto se debe utilizar una implementación de una lista enlazada que sea compatible con la API de Java, Para solucionar este problema en primera instancia se realizara una Arquitectura Orientada a Objetos para definir de manera ordenada cada una de las secciones que dividen el proyecto; para la implementación de la lista enlazada se decidió realizar la implementación de la interfaz ***List*** propia de Java y haciendo uso de la documentación de la clase Original ***LinkedList*** de Java se agregaron los métodos faltantes de la misma, la lectura de los conjuntos de números se realiza a partir de un archivo de texto plano con el nombre ”datos.txt”, el cual debe contener los datos de los conjuntos de la siguiente manera:

```
160 591 114 229 230 270 128 ... n
15.0 69.9 6.5 22.4 28.4 65.9 ... n
```

los datos decimales se utilizan con un ’.’ y la diferenciación de datos se entiende por un espacio en blanco entre ellos, para efectos prácticos se decidió que cada línea es un conjunto diferente de datos, pero con un pequeño cambio en la aplicación esta permitira leer un archivo completo como un conjunto único de datos, adicionalmente se es posible cambiar el nombre del archivo o la ruta de ubicación del mismo.

2. Arquitectura

2.1. Diseño de Clases

2.1.1. Implementacion Lista Enlazada

Para la realización de esta clase de tipo $List$, se decidió utilizar un nodo el cual contendrá el valor representativo de sí mismo y además un apuntador en memoria al siguiente nodo en la lista, esto para poder realizar un recorrido completo de la lista a partir de su cabeza, que será almacenada en la Clase $LinkedListImpl$.

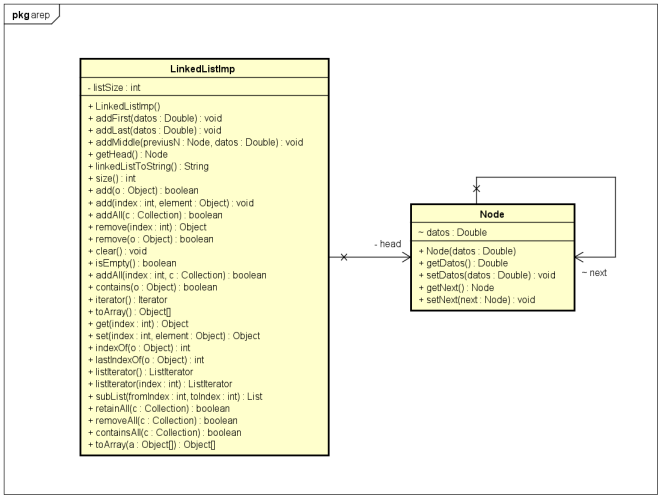

Figura 1: Modelo de clases 1

Podremos observar que la clase Node, contiene un atributo de tipo Double para almacenar en sí mismo el valor significativo que este tiene, adicionalmente cuenta con una referencia a un nodo siguiente que inicialmente es marcado como ”Null”significando esto que no hay más datos en

la lista, se tienen también los métodos para acceso a los mismos. En la clase LinkedListImp tendremos disponibles todos y cada uno de los métodos que posee una lista exceptuando algunos que no fueron considerados necesarios para la implementación, la lógica detrás de esta implementación se basa en el uso de nodos como referencias en la lista, para recorrer la lista para realizar cualquier tipo de acción (Búsqueda, Eliminación, Modificación, Adicionar) se realiza un ciclo de tipo while, que tiene fin cuando se llega a un nodo que no tiene otro como referencia, y esto querrá decir que es el final de la lista, puesto que todos y cada uno de los nodos deben tener una referencia empezando por la cabeza de la lista.

2.2. Diagrama General

Para la solución principal del problema estadístico decidimos realizar una separación entre la aplicación considerada main y la aplicación encargada de realizar la lógica de las ecuaciones estadísticas, teniendo esto en cuenta nuestra Aplicación principal es la clase .ªpp"que tiene una clase anidada llamada "LeectorArchivo"que es la encargada de realizar la lectura de los archivos, esta cuenta con dos funciones cada una para realizar la conversión de strings a un tipo lista, el método leerUniConjunto() se utiliza para leer el archivo de como si todo fuera un conjunto de número y el método leerMultiConjunto() es utilizado para considerar cada línea del archivo como un conjunto independiente.

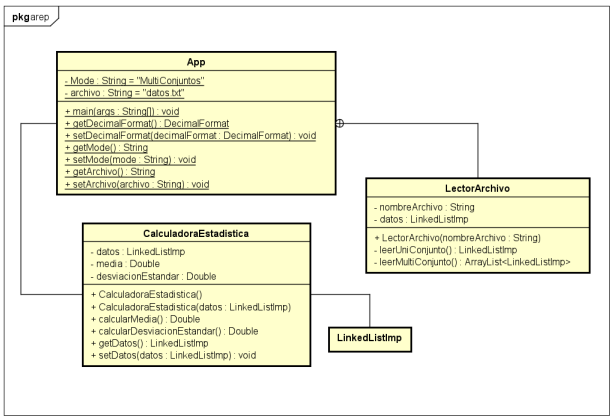

Figura 2: Modelo de clases 2

Luego de realizar la lectura correspondiente y pertinente se utiliza la clase CalculadoraEstadistica que por medio de operación y haciendo uso de la lista enlazada anteriormente mostrada resuelve los problemas de la media y la desviación estándar, para estos dos problemas se consideraron las siguientes ecuaciones:

Media (Promedio) :

$$x_{avg} = \frac{\sum_{i=1}^{n} x_i}{n} \tag{1}$$

Desviación Estándar:

$$\sigma = \sqrt{\frac{\sum_{i=1}^{n} (x_i - x_{avg})^2}{n-1}} \tag{2}$$

Finalmente los resultados son devueltos a la clase App que se encarga de mostrarlos en pantalla, se debe tener en cuenta que esta App fue diseñada para tener una precisión de 2 decimales, pero si se desea puede ser fácilmente modificable para que acceda a más decimales.

3. Pruebas

Para asegurar un buen funcionamiento se decidió hacer pruebas de unidad por medio del uso de JUnit, se realizaron un total de 6 pruebas, con las cuales se cubre el correcto funcionamiento de los cálculos estadísticos de la clase ÇalculadoraEstadistica", los datos fueron sacados de un documento de tipo .ªestudiantil"donde se realizaban estos cálculos.

Para correr las pruebas se realiza una ejecución por medio de consola, haciendo uso de maven, para esto utilizamos el siguiente comando:

```
$ mvn test
```

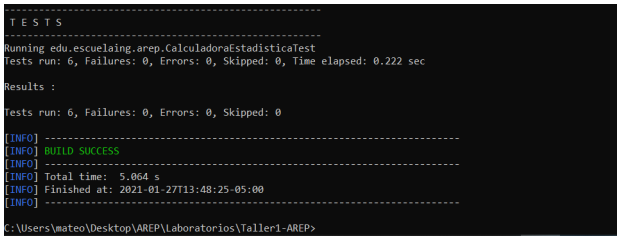

Figura 3: Resultados de pruebas Unitarias

Como podemos observar las pruebas pasan de manera satisfactoria, dando así un resultado positivo diciéndonos así que el funcionamiento del programa es correcto.

4. Conclusiones

Se pudo observar el gran beneficio que trae usar herramientas como maven al momento de la creacion de arquitecturas nuevas; gracias a su gran facilidad de creacion de ambientes de trabajo y en la realizacion de pruebas unitarias, ademas de su gran manejo de dependencias, adicionalmente se adquirio un mayor conocimiento en el uso de git y en la construccion de arquitecuturas.

Referencias

[1] Medium. 2021. Implementing Linked Lists with Java. [online] Available at: 'https://medium.com/swlh/implementing-linked-lists-with-java-25a4a708b5fc' [Accessed 27 January 2021].

[2] GeeksforGeeks. 2021. Implementing a Linked List in Java using Class - GeeksforGeeks. [online] Available at: 'https://www.geeksforgeeks.org/implementing-a-linked-list-in-java-using-class/' [Accessed 27 January 2021].

[3] Docs.oracle.com. 2021. LinkedList (Java Platform SE 7). [online] Available at: 'https://docs.oracle.com/javase/7/docs/api/java/util/LinkedList.html' [Accessed 27 January 2021].

[4] Daniel Benavides. 2021. Clase de laboratorio. [22 January 2021].

[5] David Vargas. 2021. Conversación Sobre Métodos de implementación. [24 January 2021]