

INTRODUCTION TO COMPLEX SYSTEMS, JAVA, MVN, AND GIT

Andrés Felipe Quintero Duque

Ingenieria de Sistemas, Escuela Colombiana de Ingeniería Julio Garavito, Bogotá,
Colombia.

Email: andres.quintero-d@mail.escuelaing.edu.co

13, Agosto 2020

Abstract

Se realizó una pequeña aplicación, en la cual se utilizaron conceptos de estructuras de datos, java, maven y git. A lo largo de este artículo se profundizará acerca de su diseño, implementación y funcionamiento.

1 Introducción

La implementación surge gracias al problema que se desea solucionar, para ello diseñe un pequeño modelo de clases, de tal forma que al momento de realizar la implementación se facilitara el proceso, adicionalmente se necesitaba conocimientos básicos en el área de estadística.

El problema al que se necesitaba dar solución, parte de un conjunto de datos numéricos dentro de un archivo de texto plano, en donde se necesitaba aplicar una serie de operaciones para calcular la media y la desviación estándar de los datos ya mencionados.

Estos datos fueron almacenados en una estructura de datos llamada LinkedList, fue necesario implementar esta estructura, debido a que no se permitía utilizar el API de Java, para esta implementación fue importante recordar el funcionamiento de la estructura de datos mencionada. Esta funciona por medio de nodos almacenados en una lista, en este caso cada nodo tiene dos elementos: El valor(Números del archivo) y una referencia al nodo siguiente, estos dos elementos nos van a permitir recorrer la estructura y de esta manera realizar las operaciones para calcular la media y la desviación estándar de este conjunto de datos.

Ecuaciones

Las ecuaciones que se necesitan para obtener la media y la desviación estándar son respectivamente:

La media es una medida de tendencia central. A continuacion su formula:

$$X_{avg} = \sum_{i=1}^n \frac{X_i}{n} \quad (1)$$

La desviación estándar es una medida que se utiliza para cuantificar la variación o la dispersión de un conjunto de datos numéricos. A continuacion su formula:

$$desv = \sqrt{\frac{(X_i - X_{avg})^2}{n - 1}} \quad (2)$$

2 Modelo

En la imagen se muestra el modelo de clases que diseñe para la implementación de la aplicación

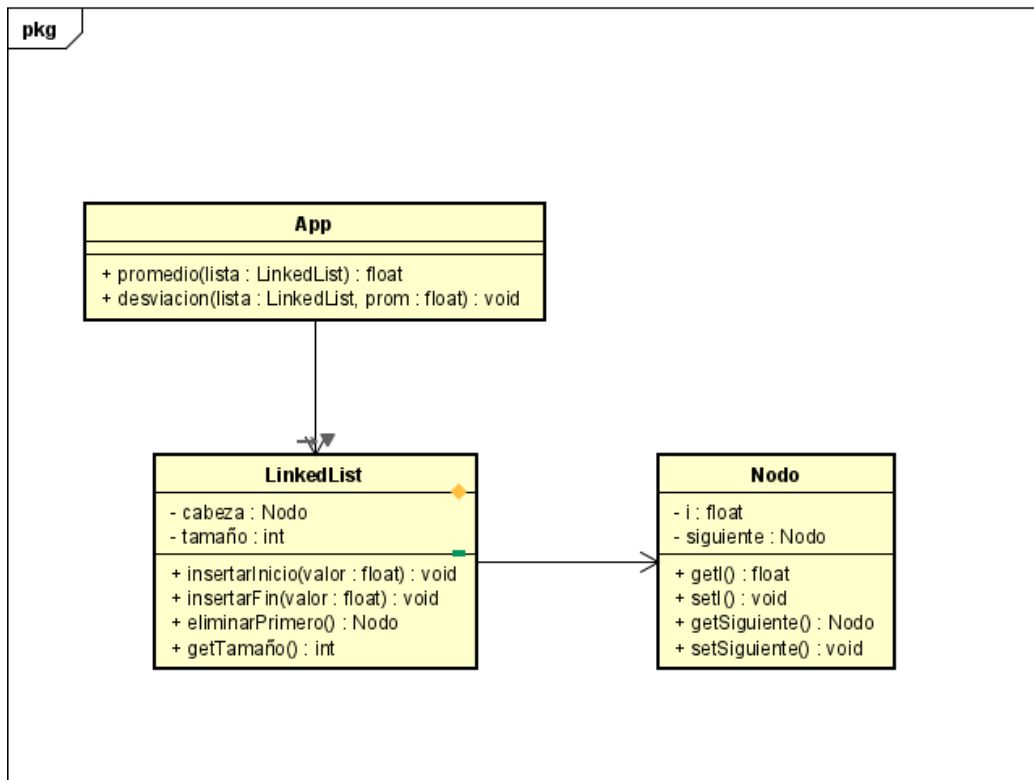


Figure 1: Modelo de clases.

Como se muestra en la imagen fue necesario realizar la implementación de tres clases, las cuales permiten realizar todas las operaciones para el procesamiento de los datos contenido en el archivo mencionado en una sección anterior.

3 Implementacion

Seguinedo el modelo presentado anteriormente 1, la implementación se realizo de la siguiente manera:

3.1 LinkedList

Para implementar esta estructura de datos es importante tener su definición clara. Esta consiste en una secuencia de nodos, en los que se guardan campos de datos y una o dos referencias, enlaces o punteros al nodo anterior o posterior. En este caso específico los datos almacenados fueron un conjunto de números flotantes, la referencia se hizo respecto al siguiente nodo, esto junto con unos métodos que permiten adicionar y eliminar elementos dentro de la LinkedList.

3.2 Nodo

Esta clase lo que permite es referenciar los nodos de la LinkedList, tiene como atributos el valor y la referencia al siguiente nodo.

3.3 App

Esta es la clase principal que se encarga de leer y procesar los datos obtenidos, e.g realiza el promedio y la desviación estándar de los datos recolectados.

3.4 Spark

Este framework se utilizó ya que permite el desarrollo de aplicaciones web en Java. Spark incluye por defecto el servidor de aplicaciones Jetty de manera que las aplicaciones web creadas con Spark pueden lanzarse como cualquier programa Java.

Con lo dicho anteriormente se realiza una app web que permite realizar todas las operaciones como se muestra en la imagen:

Calculadora de media y desviación estándar de un conjunto de números reales

Ingrese el conjunto de números separado por un espacio

Ej. 1;2;3;4;9	Calcular
---------------	----------

Figure 2: Servicio Web 1

4 Pruebas

La implementación de las pruebas se encuentra en la clase AppTest.java dentro del paquete de pruebas, adicionalmente se usó Junit y de acuerdo a los datos utilizados arrojó los siguientes resultados:

```
cd D:\backup\Documentos\ECI\AREP\Tarea Introduccion MVN-GIT\Introduccion; "JAVA_HOME=C:\\Program Files\\Java\\jdk1.8.0_241"
Running NetBeans Compile On Save execution. Phase execution is skipped and output directories of dependency projects (with
Scanning for projects...

-----
Building Introduccion 1.0-SNAPSHOT
-----

--- maven-surefire-plugin:2.12.4:test (default-cli) @ Introduccion ---
Surefire report directory: D:\backup\Documentos\ECI\AREP\Tarea Introduccion MVN-GIT\Introduccion\target\surefire-reports

-----
T E S T S
-----
Running edu.escuelaing.arep.app.AppTest
Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.058 sec

Results :

Tests run: 3, Failures: 0, Errors: 0, Skipped: 0

-----
BUILD SUCCESS
-----

Total time: 1.735 s
Finished at: 2020-08-13T14:21:48-05:00
Final Memory: 11M/305M
-----
```

Figure 3: Pruebas con Junit.

5 Conclusiones

De acuerdo al trabajo realizado se puede concluir lo siguiente:

- El uso de listas encadenadas permite ser más flexible con el orden de los elementos almacenados dentro de ella.
- Se logra comprender el funcionamiento de maven, git y java
- Se logra dar solución al problema inicial, calculando la media y la desviación estándar por medio de la implementación realizada.

6 Bibliografia

- <https://docs.oracle.com/javase/7/docs/api/java/util/LinkedList.html>
- repositorio: <https://github.com/andresQD/ArepIntroduccion>