Introducción a Maven y Git

Germán Andrés Ospina Quintero

Escuela Colombiana de Ingeniería Julio Garavito
Ingeniería de Sistemas
Arquitecturas Empresariales
Bogotá
2021

Germán Andrés Ospina Quintero

Informe

Luis Daniel Benavides Navarro

Escuela Colombia de Ingeniería Julio Garavito
Ingeniería de Sistemas
Arquitecturas Empresariales
Bogotá
2021

CONTENIDO

LISTA DE TABLAS

LISTA DE FIGURAS

GLOSARIO

Java: Java es un tipo de lenguaje de programación y una plataforma informática, creada y comercializada por Sun Microsystems en el año 1995. Se constituye como un lenguaje orientado a objetos, su intención es permitir que los desarrolladores de aplicaciones escriban el programa una sola vez y lo ejecuten en cualquier dispositivo. (Content, 2019)

Git: Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. (Wikipedia, 2021)

Linked List: en informática, una lista enlazada es una colección lineal de elementos de datos cuyo orden no viene dado por su ubicación física en la memoria. En cambio, cada elemento apunta al siguiente. Es una estructura de datos que consta de una colección de nodos que juntos representan una secuencia. (Wikipedia, 2020)

Maven: Maven es una herramienta de software para la gestión y construcción de proyectos Java creada por Jason van Zyl, de Sonatype, en 2002. Es similar en funcionalidad a Apache Ant, pero tiene un modelo de configuración de construcción más simple, basado en un formato XML. (Wikipedia, 2020)

Sistema Complejo: está compuesto por varias partes interconectadas o entrelazadas cuyos vínculos crean información adicional no visible ante el observador como resultado de las interacciones entre elementos. (Wikipedia, 2020)

RESUMEN

Para este taller, se hizo necesario realizar la implementación, personalizada, de una Linked List, a partir de la interfaz proporcionada por la librería de Java, en donde se pudieran almacenar una serie de datos con los cuales se pudiera calcular la media y la desviación estándar de estos. Adicionalmente, se realizaron pruebas unitarias para validar que los cálculos sean correctos, una documentación de cada método, viéndose plasmada en el archivo Javadoc, y la elaboración de un README en donde se puede apreciar el paso a paso para la instalación y ejecución del proyecto. El proyecto, en su totalidad, fue alojado en repositorio en la plataforma GitHub.

PALABRAS CLAVE: Sistema complejo, Linked List, Maven, Git, GitHub, media, desviación estándar.

1. INTRODUCCIÓN

Inicialmente, haciendo uso de las principales herramientas que fueron mencionadas en el glosario: Maven y GitHub, se implementa un software que logre calcular la media y la desviación estándar de un conjunto de n números reales, conjunto que se representa a partir de una linked list con una implementación propia que debe ser compatible con la API de colecciones de Java.

Los datos para evaluar fueron presentados en el documento que describe el taller, que describe la actividad a desarrollar como dos columnas de datos. A partir de estos datos, y una vez hecha la implementación, se deben hacer al menos dos pruebas con cada columna de datos.

Finalmente, todos los pasos para la instalación, ejecución y generación de la documentación del proyecto deben verse debidamente detallados en un archivo README cargado, junto con el código fuente de la aplicación, a un repositorio en GitHub.

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Introducir a Maven y Git a través de un ejercicio propuesto en clase de laboratorio.

2.2 OBJETIVOS ESPECÍFICOS

Desarrollar un ejercicio propuesto en clase para maximizar el entendimiento en las herramientas

Redactar un README con información importante sobre cómo utilizar comandos específicos que permitan ejecutar la aplicación.

3 PLANTEAMIENTO DEL PROBLEMA

3.1 DEFINICIÓN DEL PROBLEMA

Usando GitHub y Maven escriba un programa que calcule la media y la desviación estándar de un conjunto de números reales. Usando una linked list debe guardar los números a ser usados para hacer los cálculos. Finalmente, debe hacer al menos dos pruebas con los datos presentes en las columnas de la siguiente tabla:

Column 1	Column 2
Estimate Proxy Size	Development Hours
160	15.0
591	69.9
114	6.5
229	22.4
230	28.4
270	65.9
128	19.4
1657	198.7
624	38.8
1503	138.2

Tabla 1

3.2 JUSTIFICACIÓN

En el mundo académico y en el mundo laboral de hoy, es de suma importancia tener claros los conceptos, y saber aplicarlos, de herramientas que permitan gestionar y construir proyectos de software, como Maven y Gradle, y de herramientas que permitan el control de versiones, como GitHub o GitLab. No dominar estas herramientas representa un atraso técnico muy grande, puesto que son herramientas fundamentales en el mundo tecnológico de hoy en día

4 DESARROLLO DEL PROYECTO

El proyecto se dividió en tres paquetes:

1. El primero, con el nombre de “edu.escuelaing.arep.stat”, contiene una clase con el mismo nombre y que contiene dos métodos de clase (estáticos) que permiten calcular la media y la desviación estándar
2. El segundo, con el nombre de “edu.escuelaing.arep.stat.exceptions”, contiene una clase especial para las excepciones que se puedan presentar en la clase del paquete anterior
3. Y el ultimo, con el nombre de “edu.escuelaing.arep.util”, contiene tres clases que permiten definir la linked list; LinkedList, es la clase que implementa la interfaz List, LinkedListIterator, es la clase que implementa la interfaz Iterator, esto con el fin de implementar el método iterator en la anterior clase y, por último, la clase Node que implementa los nodos necesarios para la linked list.

Como forma de presentar los resultados y validar que estos sean correctos se realizaron pruebas unitarias:

1. Se calcula la media de los datos presentes en la primera columna:

```java
@Test
public void shouldGetMeanColumnOne() {
    Double expected = 550.6;
    LinkedList<Double> lista = new LinkedList<Double>();
    lista.add(160.0);
    lista.add(591.0);
    lista.add(114.0);
    lista.add(229.0);
    lista.add(230.0);
    lista.add(270.0);
    lista.add(128.0);
    lista.add(1657.0);
    lista.add(624.0);
    lista.add(1503.0);
    Double actual = 0.0;
    try {
        actual = Stat.mean(lista);
    } catch (StatException e) {
        e.printStackTrace();
    }
    assertEquals(expected, actual);
}
```

Figura 1

2. Se calcula la media de los datos presentes en la segunda columna

```java
@Test
public void shouldGetMeanColumnTwo() {
    Double expected = 60.32;
    LinkedList<Double> lista = new LinkedList<Double>();
    lista.add(15.0);
    lista.add(69.9);
    lista.add(6.5);
    lista.add(22.4);
    lista.add(28.4);
    lista.add(65.9);
    lista.add(19.4);
    lista.add(198.7);
    lista.add(38.8);
    lista.add(138.2);
    Double actual = 0.0;
    try {
        actual = Stat.mean(lista);
    } catch (StatException e) {
        e.printStackTrace();
    }
    assertEquals(expected, actual);
}
```

Figura 2

3. Se calcula la desviación estándar de los datos presentes en la primera columna:

```java
@Test
public void shouldGetDeviationColumnOne() {
    Double expected = 572.03;
    LinkedList<Double> lista = new LinkedList<Double>();
    lista.add(160.0);
    lista.add(591.0);
    lista.add(114.0);
    lista.add(229.0);
    lista.add(230.0);
    lista.add(270.0);
    lista.add(128.0);
    lista.add(1657.0);
    lista.add(624.0);
    lista.add(1503.0);
    Double actual = 0.0;
    try {
        actual = Stat.stddev(lista);
    } catch (StatException e) {
        e.printStackTrace();
    }
    assertEquals(expected, actual);
}
```

Figura 3

4. Se calcula la desviación estándar de los datos presentes en la segunda columna:

```java
@Test
public void shouldGetDeviationColumnTwo() {
    Double expected = 62.26;
    LinkedList<Double> lista = new LinkedList<Double>();
    lista.add(15.0);
    lista.add(69.9);
    lista.add(6.5);
    lista.add(22.4);
    lista.add(28.4);
    lista.add(65.9);
    lista.add(19.4);
    lista.add(198.7);
    lista.add(38.8);
    lista.add(138.2);
    Double actual = 0.0;
    try {
        actual = Stat.stddev(lista);
    } catch (StatException e) {
        e.printStackTrace();
    }
    assertEquals(expected, actual);
}
```

Figura 4

A través de Maven, se ejecuta el comando test (fase) para poder ejecutar las pruebas y validar que el programa se desempeñando de manera correcta y optima:

```
[INFO] -------------------------------------------------------
[INFO]  T E S T S
[INFO] -------------------------------------------------------
[INFO] Running edu.escuelaing.arep.app.AppTest
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.04 s - in edu.escuelaing.arep.app.AppTest
[INFO]
[INFO] Results:
[INFO]
[INFO] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0
[INFO]
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  12.643 s
[INFO] Finished at: 2021-01-27T11:52:09-05:00
[INFO] ------------------------------------------------------------------------
```

Figura 5

CONCLUSIONES

1. El mundo laboral de hoy y en día, al igual que en el académico, se hace indispensable saber el manejo de las herramientas básicas para el desarrollo de todo proyecto de software: Maven y GitHub.
2. Las interfaces, son una gran herramienta, en el mundo de desarrollo de software en Java, para poder diseñar un arquetipo o un esqueleto con el cual podamos no solo establecer un diseño, sino que también establecer una conexión entre clases que no tengan una relación.
3. Documentar el código permite 5 principales cosas: ayuda a entender nuestro código, ayuda a que los demás entiendan el código, ayuda a corregir errores más fácilmente, mantiene claro el objetivo del código y el código se vuelve reutilizable. (Rodríguez, 2014)

Bibliografía

Content, R. R. (5 de Junio de 2019). *¿Qué es Java? Conoce las particularidades de este lenguaje de programación*. Obtenido de ¿Qué es Java? Conoce las particularidades de este lenguaje de programación: https://rockcontent.com/es/blog/que-es-java/

Rodríguez, R. T. (5 de Julio de 2014). *5 razones para documentar nuestro código* . Obtenido de 5 razones para documentar nuestro código : https://desarrolladores.me/2014/07/5-razones-para-documentar-nuestro-codigo/

Wikipedia. (21 de Diciembre de 2020). *Linked list*. Obtenido de Linked list: https://en.wikipedia.org/wiki/Linked_list

Wikipedia. (10 de Octubre de 2020). *Maven*. Obtenido de Maven: https://es.wikipedia.org/wiki/Maven

Wikipedia. (23 de Noviembre de 2020). *Sistema complejo*. Obtenido de Sistema complejo: https://es.wikipedia.org/wiki/Sistema_complejo

Wikipedia. (26 de Enero de 2021). *Wikipedia*. Obtenido de Git: https://es.wikipedia.org/wiki/Git