Despliegue en Heroku y Spark

Johan Nicolas Cortes Torres
Escuela Colombiana de Ingenieria Julio Garivito
AREP

August 2020

Abstract

En el presente documento se encontrará información acerca de la elaboración de un laboratorio que se basaba en implementar una aplicacion web que se pueda desplegar en heroku usando el FrameWork de Spark, asi mismo esta laboratorio presente como se realizo el cálculo de la media y la desviación estándar. También se encontrará una explicación de una lista doblemente enlazada que se implementó por completo y como esta se usó para el cálculo de la media y la desviación estándar de diferentes datos. Adicionalmente, se presentan unas pruebas de dichos cálculos.

1 Introduction

El laboratorio se divide en tres partes fundamentales. La primera parte trata de la implementacion de una aplicacion web con el framework de spark que finalmente se va a desplegar en heroku. La segunda parte trata de implementar una lista doblemente enlazada y la tercera parte trata de implementar el cálculo de la media y la desviación estándar, de datos específicos.

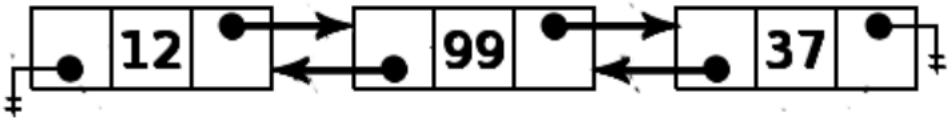
Spark es un simple y expresivo marco web de Java/Kotlin DSL construido para un desarrollo rápido. La intención de Sparks es proporcionar una alternativa para los desarrolladores de Java/Kotlin que quieran desarrollar sus aplicaciones web de la forma más expresiva posible y con un mínimo de complejidad.

Heroku es una empresa que se especializa en ofrecer servicios de plataforma administrada, por sus siglas en inglés PasS, en otras palabras ofrece servicios de servidores y redes administrados por Heroku en donde se pueden alojar aplicaciones de diferentes lenguajes de programación como Python, Java, PHP y más.

Una lista doblemente enlazada es una estructura de datos que se utiliza para mantener colecciones de datos, esto por medio de nodos los cuales albergan en si tres valores:

- el nodo anterior

- el nodo siguiente
- su valor



La media es el promedio de un conjunto de datos, esta se calcula mediante la suma de todos los datos otorgados divido entre a cantidad de datos que se sumaron, la fórmula de esta es:

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

La Desviación estándar es una medida de la extensión o dispersión de un conjunto de datos, la cual se calcula mediante una sumatoria de la resta de cada valor y la media, al cuadrado, divido la cantidad de datos menos 1 y la raíz cuadrada de esta división. La fórmula de esta es:

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

2 Implementacion

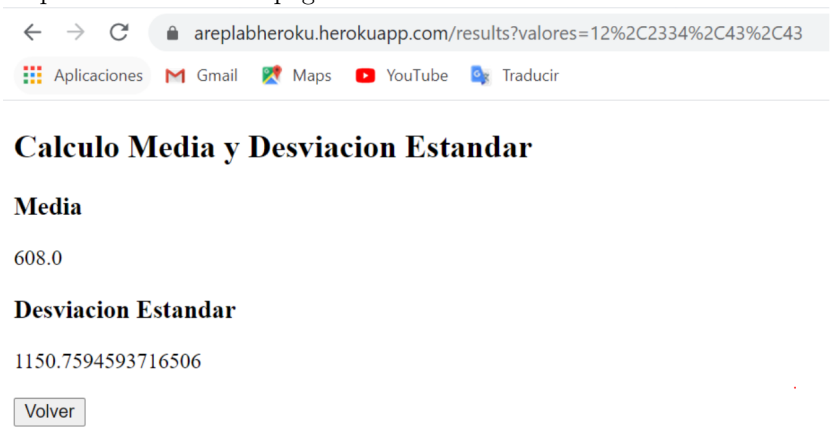
La clase principal se llama App, esta clase crea un objeto de tipo linkedList y un objeto de tipo Calculos al cual le manda la LinkedList anteriormente creada. Para la adición de los datos se lee un archivo de texto el cual se encuentra ubicado con las clases. En el proyecto existen tres archivos (datos1, datos2 y

datos3) los cuales contienen tres ejemplos diferentes de datos para su lectura y la realización de los cálculos de la media y desviación estándar.

2.1 Spark Framework

Para la implementacion de la aplicacion web se implemento la clase SparkWebApp, la cual cuenta con los siguientes metodos:
- inputDataPage: Este metodo nos permite crear un formulario en html el cual recibe los datos para hacer el envio de estos para los calculos porteriores.

- resultsPage: Este metodo recibe los datos del metodo inputDataPage y calcula la media y desviacion estandar de los datos que son ingresados por la pagina y asi presentarlos en una pagina web.



- getPort: Este metodo retorna el puereto por el que esta corriendo la aplicacion.

2.2 Lista doblemente enlazada

Para la implementación de la lista doblemente enlazada, se utilizaron dos clases:

- LinkedList: esta es la implementación de la lista doblemente enlazada, que se basa en nodos. Esta implementación tiene como referencia dos nodos la cabeza, que se refiere al primer elemento ingresado en la lista y la cola que se refiere al último elemento ingresado en ella. En esta clase, ademas de los getters y setter de los atributos, podemos encontrar los métodos:
    - addNode: este agrega un nodo al final de la lista.
    - removeNode: remueve el ultimo nodo de la lista.
    - nextNode: retorna el siguiente nodo en la lista.
    - priorNode: retorna el anterior nodo de la lista.

- Node: esta es la implementación de cada nodo que contiene la lista doblemente enlazada, que se basa en tres atributos, el nodo siguiente, el nodo anterior y valor que alberga dicho nodo. Esta clase solo contiene como métodos los getters y setters de cada atributo.

2.3 Calculos

Para la implementación de los cálculos de la media y la desviación estándar, se usó la clase Cálculos la necesita una lista doblemente enlazada para su creación. Esta clase cuenta con dos métodos:

- mean: este método calcula, a partir de la lista dada, la media de los datos albergados en ella.
- deviation:este método calcula, a partir de la lista dada, la desviación estándar de los datos.

2.4 Pruebas

Para la implementación de las pruebas se usó la clase appTest, la cual se encuentra en el paquete Test, implementándola en base de JUNIT. Para esto se realizaron dos pruebas con distintos conjuntos de datos los cuales se tomaron del enunciado de laboratorio.

Column 1 Estimate Proxy Size	Column 2 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

Table 1

Test	**Expected Value**	
	Mean	*Std. Dev*
Table 1: Column 1	550.6	572.03
Table 1: Column 2	60.32	62.26

Estos datos fueron ingresados manualmente donde en la clase, la cual al ser ejecutada mediante Maven, por el comando mvn package nos arroja el siguiente resultado:

```
-------------------------------------------------------
 T E S T S
-------------------------------------------------------
Running edu.escuelaing.arep.AREPLAB1.AppTest
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.194 sec

Results :

Tests run: 2, Failures: 0, Errors: 0, Skipped: 0

[INFO]
[INFO] --- maven-jar-plugin:2.4:jar (default-jar) @ AREPLAB1 ---
[INFO] Building jar: C:\Users\johan\Desktop\U\10 Semestre\AREP\AREPLAB1\target\AREPLAB1-1.0-SNAPSHOT.jar
[INFO] ------------------------------------------------------------------------
[INFO] BUILD SUCCESS
[INFO] ------------------------------------------------------------------------
[INFO] Total time:  2.690 s
[INFO] Finished at: 2020-08-12T19:22:51-05:00
[INFO] ------------------------------------------------------------------------
```

3 Conclusion

Concluyendo se realizó correctamente la implementación completa del laboratorio propuesto por la clase de AREP, se logró recordar el funcionamiento de Maven y de GitHub para la implementación de este. Se logro realizar una implementación correcta de una lista doblemente enlazada y sus respectivos nodos y finalmente se logró realizar los cálculos correctamente, de la media y desviación estándar, de los diferentes datos otorgados.

4 Referencias

[1] Laboratorio
`http://campusvirtual.escuelaing.edu.co/moodle/pluginfile.php/181210/mod_resource/content/0/EnunciadoTallerEjercicioMVNGit.pdf`
Repositorio
`https://github.com/jnicolasct/AREP-LAB1`