



INGENIERÍA DE SISTEMAS

ARQUITECTURAS EMPRESARIALES
LABORATORIO No.2
MEDIA Y DESVIACIÓN ESTANDAR EN LA NUBE

Diego Alejandro Puerto Gómez
diego.puerto@mail.escuelaing.edu.co

Bogotá
Agosto 2020

1 Introducción

En este reporte se quiere integrar herramientas como Maven y GitHub las cuales brindan una estructura más clara y robusta a los proyectos desarrollados a una plataforma en la nube como lo es Heroku la cual lo almacenará y proveerá constantemente. Se plantea la construcción de un software que calcule la media y la desviación estándar de una serie de números reales tomando los datos directamente de un formulario web basado en una aplicación SparkWeb.

2 Resumen

La media y la desviación estándar son dos medidas estadísticas que sirven para medir la dispersión de los datos, dada una serie de números reales que se ingresan por medio de un campo ubicado en el formulario web, se quiere hacer un programa que procese los datos y calcule dichas dos medidas.

3 Diseño

Inicialmente se quiere almacenar dicha información en una estructura de datos llamada *LinkedList*, la cual consiste en una lista de datos de forma encadenada, cada elemento se llama nodo y cada uno de ellos almacena su información y la referencia del nodo siguiente. Es por esto por lo que la información sobre cuál es el nodo de cabecera, es sumamente importante. (1)

A pesar de que Java ya cuenta con una implementación de esta estructura, para el ejercicio se plantea hacer una implementación propia. Para ello se creó una clase llamada *LinkedList* la cual implementa *List*, lo que quiere decir que varios de los funcionamientos de sus métodos provienen de ella. Para cada uno de sus elementos se creó una clase llamada *Node* la cual tiene la característica de conocer la referencia al objeto siguiente.

Como método principal del programa se creó una clase *MeanAndDesviation* la cual recibe una lista de números tomados directamente de la página web, posteriormente almacenará nodo a nodo en la *LinkedList* y determinará su media y su desviación estándar. Es gracias al uso de *sparkWeb* que se permite la construcción del frontEnd y la lectura de los datos en los formularios web, que luego serán procesados en el backEnd de la aplicación.

El funcionamiento del frontEnd se da gracias a que se crea código HTML en los métodos Java, esto permite una transición de la información muy directa entre lo que se ve y el procesamiento en el backEnd, los métodos correspondientes son llamados según el path en el buscador, y es por ello que inicialmente se debe cargar la dirección en donde está la aplicación en la nube <https://boiling-cove-24254.herokuapp.com> y agregar */inputdata*, esto invocará al método *inputDataPage* el cual mostrará los formularios correspondientes de ingreso de datos. (Figura 1)

Luego de ingresar los números a ser evaluados separados por comas, el botón *Submit* que almacena una acción, redirige la página a */results*, esta llama al



← → ↻ boiling-cove-24254.herokuapp.com/inputdata

Aplicaciones ★ Bookmarks

Calculo de la media y la desviacion estandar

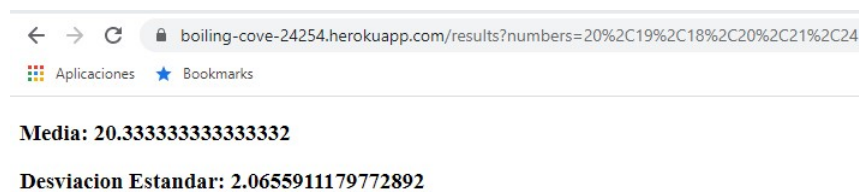
Ingrese los numeros que desee que sean procesados, separados por comas

Submit

Si da click en el boton "Submit", se enviarian los datos a un pagina llamada "/results".

Figure 1: Ingreso de Datos

método *resultsPage* y es este método el que transforma el valor de campo de entrada en una lista de números listos para ser procesados por el backEnd de la aplicación y posteriormente mostrar los resultados. (*Figura 2*)



← → ↻ boiling-cove-24254.herokuapp.com/results?numbers=20%2C19%2C18%2C20%2C21%2C24

Aplicaciones ★ Bookmarks

Media: 20.33333333333332

Desviacion Estandar: 2.0655911179772892

Figure 2: Resultados

Finalmente la clase *Calculadora* contiene dos métodos, uno para calcular la media *getMean* y otro para calcular la desviación estándar *getDeviation* los cuales asignan valores a dos variables que son las mostradas en la figura 2. La organización es como se muestra en la figura 3.

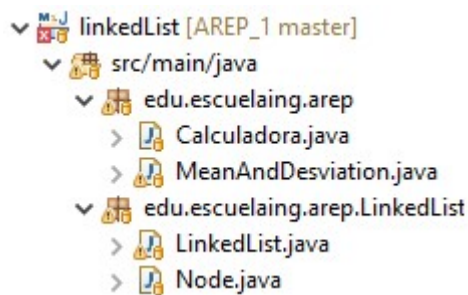


Figure 3: Estructura

Table 1: Resultados

Test	Expected Value		Actual Value	
	Mean	Std. Dev	Mean	Std. Dev
Column 1	550.6	572.03	550.6	572.0268
Column 2	60.32	62.26	60.32	62.2558

3.1 Memória

Se ha identificado en la estructura de datos LinkedList y como tal en cada uno de sus nodos, en las variables que almacenan la información de los formularios HTML.

3.2 Intérprete

JavaScript, Java y el buscador que se use, para efectos de estas pruebas, Google Chrome

3.3 Enlaces de comunicación

Internet para el correcto funcionamiento en Heroku y SparkWeb que posibilita la comunicación entre el frontEnd y el backEnd

4 Conclusiones

- Para la generación propia de la LinkedList no era necesario implementarla desde cero, usando la implementación de List se simplificaba en gran medida el trabajo
- Aunque existen varios tipos de LinkedList, la implementación que solo guarda la referencia de su cabeza y el cada nodo la referencia del siguiente, es una buena solución para el ejercicio propuesto
- SparkWeb facilita y modulariza el trabajo de una aplicación web brindando la posibilidad de hacer conexiones simples entre el frontEnd y el backEnd
- Heroku posibilita el almacenamiento en la nube de las aplicaciones con tan solo configurar el archivo Procfile y unas breves instrucciones para almacenarlo allí.

References

- [1] Personal Software Process for Engineers. Carnegie Mellon University
http://campusvirtual.escuelaing.edu.co/moodle/pluginfile.php/181210/mod_resource/content/0/EnunciadoTallerEjercicioMVNGit.pdf

- [2] Ejemplo formulario en SparkFile. Daniel Benavides
http://campusvirtual.escuelaing.edu.co/moodle/pluginfile.php/129029/mod_resource/content/1/SparkWebApp.java