

Taller Heroku

Alejandro Toro Daza

Profesor:
Luis Daniel Benavides Navarro

Arquitecturas Empresariales

Escuela Colombiana de Ingeniería Julio Garavito
20 de Enero del 2021
Bogotá D.C.

TABLA DE CONTENIDO

1. INTRODUCCIÓN	2
2. OBJETIVOS	2
3. MARCO TEÓRICO	2
4. DISEÑO Y CONSTRUCCIÓN	4
4.1. CÁLCULOS	4
4.2. LISTAS ENLAZADAS	4
4.3. PRUEBAS	5
5. ARQUITECTURA	9
6. CONCLUSIONES	10
7. REFERENCIAS	11

1. INTRODUCCIÓN

En el Taller Ejercicio Maven y GIT se escribirá un programa para calcular la desviación estándar y media de un conjunto de n números reales. El programa debe realizar la lectura de los n números reales de un archivo de texto, el cual se debe utilizar una lista enlazada o *Linked List* para almacenar los n números para los respectivos cálculos. Para esto, se debe escribir una implementación propia de una lista enlazada y esta debe ser compatible con la API de colecciones de Java. Luego, se implementará con Spark una página web la cual tendrá una interfaz de usuario, y el usuario podrá en esta interfaz ingresar los valores de los cuales quiere hallar la Media y la Desviación Estándar. Esta página web será desplegada en la herramienta Heroku, brindando así la oportunidad de que cualquier usuario pueda acceder a ella y utilizarla.

2. OBJETIVOS

- Elaborar un programa en Java que permita calcular la media y la desviación estándar realizando la respectiva lectura de los n números reales de un archivo de texto.
- Implementar pruebas de unidad para verificar que los resultados obtenidos en dicho programa son correctos.
- Realizar la respectiva documentación, diseño y construcción del código fuente para entregar el taller con todos sus requerimientos previos.
- Desplegar la aplicación en Heroku para poder ejecutarla desde cualquier ordenador, junto con la Integración Continua para su debida revisión y verificación de que el código se encuentre en total funcionamiento.

3. MARCO TEÓRICO

- **Media:** Valor promedio de un conjunto de datos numéricos, calculada como la suma del conjunto de valores dividida entre el número total de valores. La media, a diferencia de la esperanza matemática, es un término matemático. Por su parte, la esperanza matemática es un término estadístico, relacionado con las probabilidades. El cálculo de ambas variables viene, muchas veces, a ser el mismo. No obstante, no siempre se utilizan en el mismo contexto.

La fórmula para calcular la Media es la siguiente:

\bar{x} = Media.

n = Número de valores.

x_i = Valores del conjunto de datos.

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

- **Varianza:** La varianza es una medida de dispersión que representa la variabilidad de una serie de datos respecto a su media. Formalmente se calcula como la suma de los residuos al cuadrado divididos entre el total de observaciones.

También se puede calcular como la desviación típica al cuadrado. Dicho sea de paso, entendemos como residuo a la diferencia entre el valor de una variable en un momento y el valor medio de toda la variable.

La fórmula para calcular la Varianza es la siguiente:

$$S^2 = \text{Varianza.}$$

x_i = Valores del conjunto de datos.

\bar{x} = Media.

n = Número de valores.

$$S^2 = \frac{\sum (x_i - \bar{x})^2}{n - 1}$$

- **Desviación Estándar:** La desviación estándar o desviación típica es una medida que ofrece información sobre la dispersión media de una variable. La desviación estándar es siempre mayor o igual que cero.

La fórmula para calcular la Desviación Estándar es la siguiente:

σ = Desviación Estándar.

n = Número de valores.

x_i = Valores del conjunto de datos.

\bar{x} = Media.

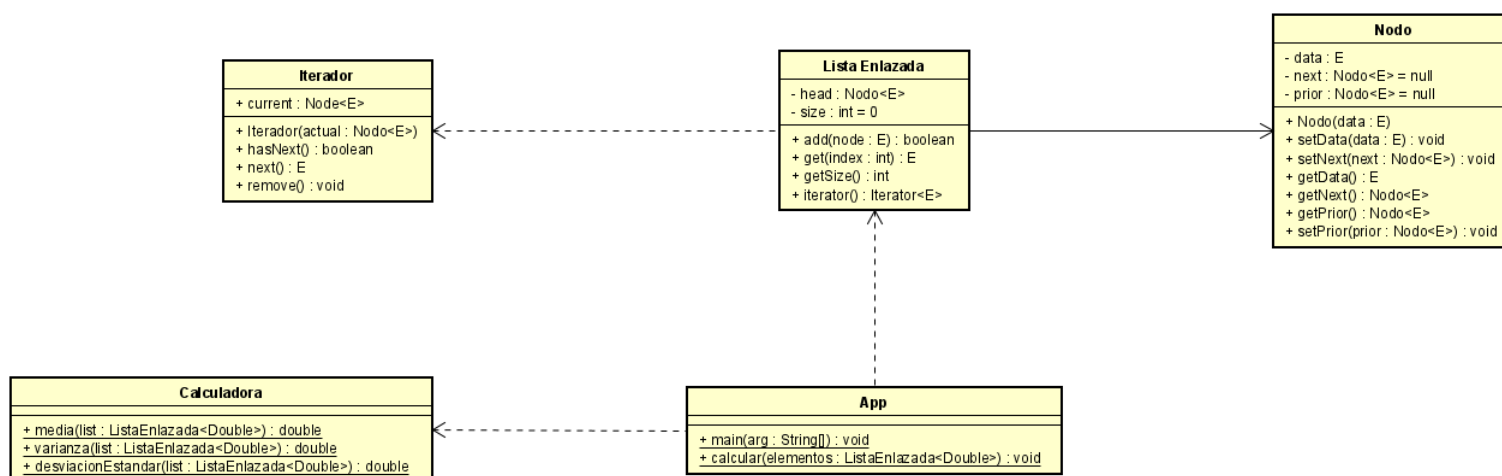
$$\sigma = \sqrt{\frac{\sum (x_i - \bar{x})^2}{n}}$$

- **Lista Enlazada:** Una lista enlazada es una estructura de datos dinámica. La cantidad de nodos en una lista no es fija y puede crecer y contraerse a demanda. Cualquier aplicación que tenga que tratar con un número desconocido de objetos necesitará usar una lista vinculada. Las listas enlazadas se encuentran entre las estructuras de datos más simples y comunes. Se pueden usar para implementar muchos otros tipos de datos abstractos comunes, incluyendo listas, pilas, colas, matrices asociativas y expresiones S, aunque no es raro implementar esas estructuras de datos directamente sin utilizar una lista vinculada como base.

4. DISEÑO Y CONSTRUCCIÓN

Para la creación del proyecto, se dispuso de cinco clases en total. La clase **Main** que se encarga de inicializar el programa, la cual hace un llamado directo a la clase **Calculadora**, la cual se encarga única y exclusivamente de calcular la media, varianza y desviación estándar. Asimismo, llama a la clase **Lista Enlazada** que se encarga de realizar la lista enlazada de todos los valores ingresados en el archivo de texto llamado **Valores.txt**, la cual, mediante dos clases, **Nodo** e **Iterador**, se encarga de realizar las operaciones pertinentes para realizar los cálculos empleando la lista enlazada.

Diagrama de Clases:



4.1. CÁLCULOS

Para calcular la Media y la Desviación Estándar, se creó una clase llamada **Calculadora**, la cual tiene tres métodos que son, **media**, **varianza** y **desviacionEstandar**. El método llamado **varianza**, fue necesario implementarlo para facilitar los cálculos, ya que, para calcular la Desviación Estándar, bastaba solo con obtener la raíz cuadrada de la Varianza obtenida anteriormente, para así entregar un código fuente más ordenado y completo.

4.2. LISTAS ENLAZADAS

Para la realización de la Lista Enlazada que llevaba a cabo todos los datos ingresados en el archivo de texto, se utilizaron cuatro métodos, que son: **add**, **get**, **getSize** e **iterator**. Estos métodos estaban encargados de agregar un nodo a la lista enlazada, obtener un elemento dentro de la lista enlazada, obtener el tamaño de la lista enlazada y crear un iterador para desplazarse entre los elementos de la lista enlazada respectivamente. Para esto se crearon tres clases. **ListaEnlazada**, **Nodo** e **Iterador**. La clase **ListaEnlazada** se encarga de, como su nombre lo dice, enlazar los valores, utilizando los métodos anunciados anteriormente. La clase **Nodo** se encarga de almacenar la información de los nodos de cada valor, y la clase

Iterador actúa como un iterador cuya función principal es desplazarse entre los nodos de la lista enlazada.

4.3. PRUEBAS

Pruebas en Maven

Para verificar que todo el código está funcionando con total normalidad, realizando los cálculos apropiadamente y obteniéndolos de una lista enlazada, se crearon varias pruebas unitarias utilizando **Junit**, herramienta que nos provee el IDE que se utilizó para la realización de todo el código fuente, Eclipse. Para estas pruebas unitarias, también se utilizó Maven, en el cual ejecutando los comandos como **mvn test** también se probó que el código funcionara con total normalidad, demostrando así total funcionamiento del código fuente retornando los resultados esperados. Los valores fueron primero ingresados en una hoja de cálculo, en la cual se calcularon tanto la media como la desviación estándar de los datos que se usaron para las pruebas. Dichos valores fueron registrados en las tablas a continuación.

Valores de las Pruebas 1 y 2:

	Prueba 1
	10.0
	12.0
	13.0
	16.0
	9.0
	8.0
	12.0
	8.0
	6.0
	16.0
Media	11
Desviación Estándar	3.22490309932

	Prueba 2
	320.69
	401.45
	395.12
	390.63
	290.65
	299.65
	345.67
	356.78
	389.92
	406.82
	410.34
	421.02
	362.46
	395.33
	402.56
Media	372.606
Desviación Estándar	40.1074162385628

Valores de las Pruebas 3 y 4:

	Prueba 3
	4.0
	3.0
	2.0
	7.0
	8.0
	4.0
	5.0
	3.0
	5.0
	7.0
	6.0
	Media
Desviación Estándar	1.831767425

	Prueba 4
	11203.34
	12023.64
	10463.81
	13001.95
	9956.63
Media	11329.874
Desviación Estándar	1088.94684914554

Valores de la Prueba 5:

	Prueba 5
	899.65
	905.32
	799.73
	850.65
	870.49
	889.78
	905.61
	895.79
	800.01
	834.95
	Media
Desviación Estándar	39.5728638842326

Luego de haber realizado las pruebas en Junit utilizando *assertEquals* con los mismos valores de las tablas mostradas con anterioridad, al ejecutar el comando de Maven para correr las pruebas **mvn test**, vemos que las pruebas fueron ejecutadas exitosamente sin ningún error de compilación.

```
-----
T E S T S
-----
Running edu.escuelaing.arep.app.AppTest
Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.055 sec

Results :

Tests run: 5, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 2.578 s
[INFO] Finished at: 2021-01-21T20:07:32-05:00
[INFO] -----
```

Pruebas en Heroku

Luego de realizar la respectiva configuración del código, con las dependencias añadidas en el archivo **pom.xml** y el **Procfile**, al acceder a la URL de la aplicación desplegada en Heroku (<https://areplab2.herokuapp.com/>) vemos que la aplicación muestra la interfaz de usuario de la siguiente forma, demostrando que la aplicación dispone de una interfaz desplegada en Heroku, donde ingresamos los valores **77,65,54,96,85,63,45,76,68,59**.



Al realizar clic en el botón **Calcular Media y Desviación Estándar**, se despliega otra interfaz de usuario, en la cual muestra los resultados tanto para la Media como para la Desviación Estándar por aparte, demostrando que la ejecución de la aplicación en Heroku funciona correctamente. Además, para comodidad del usuario se añadió un botón llamado **Regresar a la Pagina Principal**, en el que el usuario puede regresar a la primera interfaz y registrar otros datos para hallar la Media y la Desviación Estándar de estos.



5. ARQUITECTURA

Al realizar el respectivo despliegue en Heroku de la aplicación, y añadiendo la integración continua con la herramienta CircleCI, vemos que la arquitectura del programa es cliente-servidor, principalmente por lo que la aplicación se encuentra en un servidor de Heroku, la cual puede ser accedida desde cualquier navegador, enviando peticiones HTTP al servidor, el cual las recibe y las procesa para posteriormente por medio de un controlador ofrecido por el framework Spark realiza los respectivos **get** y **post** para su respectivo procesamiento. La estructura de la arquitectura luego de realizar el despliegue y la integración continua puede ser descrita de la siguiente forma:

- La aplicación primero utiliza una estructura de datos que funciona como memoria, que es la lista enlazada, que se encarga de guardar los datos ingresados por el cliente, mediante el método **add** implementado en el código fuente, junto con el **read** que se encarga de leer los datos y realizar los cálculos.
- Luego el servidor maneja los request, también implementados en el código para manejar abstractamente la comunicación entre el cliente y el servidor, mediante el protocolo HTTP para el envío y el recibimiento de información.
- Después como el código está implementado en Java, procesa los datos utilizando este lenguaje, para así poder retornar el resultado correcto pedido por el cliente, para así poder realizar los cálculos, y mediante el mismo protocolo HTTP devolverle el resultado al cliente por medio de una interfaz creada en HTML (otro lenguaje).
- Finalmente, la integración continua se maneja junto con Heroku, ya que dentro de la configuración de Heroku se optó por desplegar la aplicación si y solo si el código pasa las pruebas en CircleCI (herramienta utilizada para realizar la integración continua) para así llevar un mejor control de calidad del código, para que se compile apropiadamente para que el cliente o usuario no presente inconvenientes desplegando la aplicación.

6. CONCLUSIONES

- La realización de los cálculos de la Media, Varianza y Desviación Estándar dejó un claro aprendizaje de cómo se puede llevar a cabo estadísticas utilizando lenguajes de programación como Java, los cuales permitieron desarrollar un programa capaz de cumplir dichas funciones como lo fueron tomar estadísticas de datos ingresados por usuarios, y así entregar ordenadamente por medio de este estas estadísticas de forma ordenada y completa.
- De la misma forma, utilizando la estructura de datos de Lista Enlazada, también dejó un claro aprendizaje de cómo se puede mediante listas con nodos realizar cálculos estadísticos para poder obtener los respectivos resultados, demostrando que, mediante diferentes estructuras de datos, también se puede desarrollar un programa capaz de adjuntar los datos ingresados por un usuario en una lista enlazada y así obtener los resultados esperados.
- Asimismo, el despliegue en Heroku permitió poder ejecutar la aplicación que calcula la Media y la Desviación Estándar para poder probar que el funcionamiento en nube sirva para poder compilar el código en cualquier ordenador, utilizando el protocolo HTTP para la comunicación cliente-servidor para asegurar una total ejecución del código de manera remota.
- Por otro lado, se implementó integración continua en todo el código fuente para llevar un control de calidad del código, para asegurar que el código esté funcionando totalmente sin ningún problema tanto de compilación como de los resultados retornados después de realizar las respectivas operaciones, y así poder desplegar la aplicación sin presentar ningún tipo de errores.

7. REFERENCIAS

- López, José Francisco. “Media.” *Economipedia*, 13 Ene. 2021, economipedia.com/definiciones/media.html.
- López, José Francisco. “Varianza.” *Economipedia*, 12 Jan. 2021, economipedia.com/definiciones/varianza.html.
- López, José Francisco. “Desviación Típica - Definición, Qué Es y Concepto.” *Economipedia*, 12 Ene. 2021, economipedia.com/definiciones/desviacion-tipica.html.
- Salcedo, Luis. “Linked List: Listas Enlazadas - Implementación En Python.” *Mi Diario Python*, 1 Julio 2018, pythondiario.com/2018/07/linked-list-listas-enlazadas.html.