

# Heroku: Intérpretes, canales de comunicación y memoria

## Calculadora Web

Davor J. Cortés  
dept. Ingeniería de Sistemas  
Escuela colombiana de Ingeniería Julio Garavito  
Bogotá, Colombia  
davor.cortes@mail.escuelaing.edu.co  
2020

**Resumen**—Este artículo presenta la implementación de una calculadora estadística desplegada en la web, con dos operaciones fundamentales como lo son la media de un conjunto de datos y su desviación estándar. Además, como parte del reto se implementó una lista encadenada simple y algunas de sus funcionalidades básicas. Los resultados obtenidos apuntan a que la calculadora cumple con la funcionalidad requerida y de allí también podemos concluir que la estructura de datos implementada cumple bien su papel.

### I. INTRODUCCIÓN

Los cálculos estadísticos nos ayudan a proveer información importante acerca de un conjunto de datos que por si solos no dicen mucho. La mayor parte del tiempo resultan en operaciones básicas repetitivas y en las que es fácil cometer un error por este motivo podemos delegarle la tarea de hacer estos cálculos a las máquinas.

Se debe implementar una calculadora estadística con dos funcionalidades, calcular la media y la desviación estándar de un conjunto de  $n$  números reales, haciendo uso de herramientas tecnológicas como Maven y Git. Se desplegará una aplicación web para mostrar los resultados y leer los números reales, almacenarlos en una lista enlazada que cuente con las operaciones básicas correspondientes a la colección de la API de Java.

Los resultados obtenidos serán evaluados por medio de pruebas unitarias, realizadas a la calculadora con datos como el estimado de horas de desarrollo. También se presenta la cobertura de las pruebas en cada uno de los

paquetes, clases y métodos usados para la implementación.

### II. DESCRIPCIÓN DEL PROBLEMA

El problema de la implementación se ocasiona algunas veces por una mala abstracción de los elementos que componen el sistema es por esto que debemos definir las operaciones y estructuras de datos a utilizar.

#### II-A. Lista Enlazada

Consiste en una secuencia de nodos, en los que se guardan campos de datos arbitrarios y una o dos referencias, enlaces o punteros al nodo anterior o posterior [?].

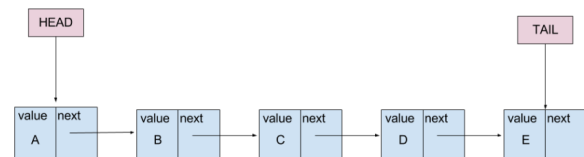


Figura 1. Lista enlazada simple

## II-B. Operaciones

**Media:** Es una medida de tendencia central, también conocida como promedio, está ubicada en el centro de los datos [?].

$$x_{avg} = \sum_{i=1}^n \frac{x_i}{2} \quad (1)$$

**Desviación Estándar:** Es una medida que se utiliza para cuantificar la variación o la dispersión de un conjunto de datos numéricos [?].

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

## III. PROPUESTA DE SOLUCIÓN

### III-A. Diseño

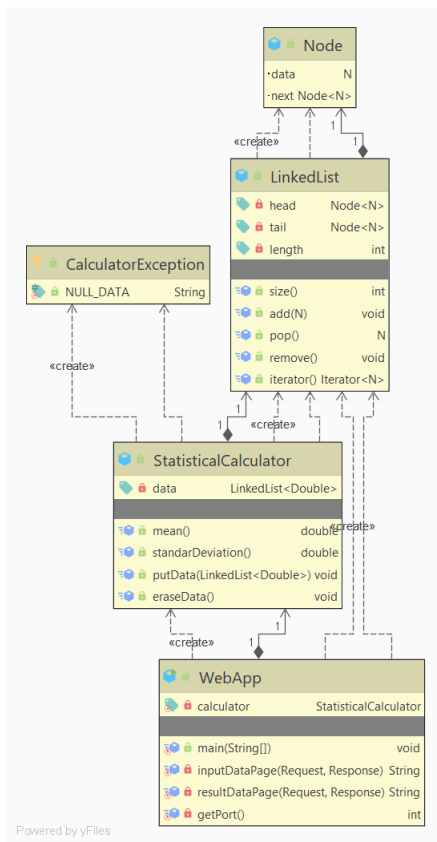


Figura 2. Diagrama de clases

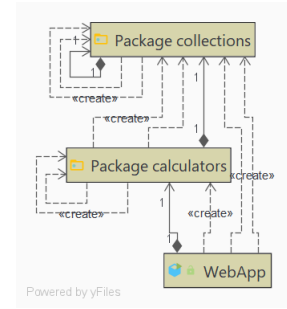


Figura 3. Diagrama de Paquetes

### III-B. Descripción

- **Node** : Es una clase genérica encargada de encapsular un dato de tipo N y de tener una referencia a un nodo siguiente. Está es la unidad básica con la que se formará la lista enlazada.
- **LinkedList** : Es una clase genérica que representa una abstracción de la lista enlazada simple, esta clase está compuesta de nodos y permite coleccionar datos de tipo N.
  - size( ) : Devuelve un entero que representa la cantidad de elementos coleccionados.
  - add(N): Colecciona un nuevo elemento.
  - pop( ) : Remueve y retorna el primer elemento de la colección.
  - remove( ) : Remueve el primer elemento de la colección
  - iterator( ) : Devuelve un iterable de la colección para que sea posible iterar sobre ella.
- **StatisticalCalculator** : Es una clase que nos permite instanciar una calculadora estadística para realizar el cálculo correspondiente sobre un conjunto de datos dado.
  - putData(datos): Permite ingresar el conjunto de datos en la calculadora.
  - eraseData( ) : Permite borrar los datos de la calculadora.
  - mean( ) : Devuelve el cálculo de la media de los datos.
  - standarDeviation( ) : Devuelve la desviación estándar de los datos.
- **WebApp** : Esta clase es la encargada de responder a las peticiones HTTP y de comunicar la lógica de la calculadora con la WEB.

## IV. EVALUACIÓN

### IV-A. Pruebas

Las pruebas fueron automatizadas gracias al framework JUnit y integradas gracias al servicio de CircleCi.

| Column 1            | Column 2    |
|---------------------|-------------|
| Estimate Proxy Size | Development |
| 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       |

| Test     | Expected Value |          | Actual Value |          |
|----------|----------------|----------|--------------|----------|
|          | Mean           | Std. Dev | Mean         | Std. Dev |
| Column 1 | 550.6          | 572.03   | 550.6        | 572.03   |
| Column 2 | 60.32          | 62.26    | 60.32        | 62.26    |

Figura 4. Tabla de Pruebas

### IV-B. cobertura

| Element     | Class, %   | Method, %   | Line, %      |
|-------------|------------|-------------|--------------|
| calculators | 100% (2/2) | 100% (6/6)  | 100% (25/25) |
| collections | 100% (3/3) | 80% (12/15) | 75% (27/36)  |

Figura 5. Tabla de Cobertura de Paquetes

| Element               | Class, %   | Method, %  | Line, %      |
|-----------------------|------------|------------|--------------|
| CalculatorException   | 100% (1/1) | 100% (1/1) | 100% (2/2)   |
| StatisticalCalculator | 100% (1/1) | 100% (5/5) | 100% (23/23) |

Figura 6. Tabla de Cobertura del Paquete Calculators

| Element    | Class, %   | Method, %  | Line, %     |
|------------|------------|------------|-------------|
| LinkedList | 100% (2/2) | 80% (8/10) | 74% (20/27) |
| Node       | 100% (1/1) | 80% (4/5)  | 77% (7/9)   |

Figura 7. Tabla de Cobertura del Paquete Collections

## V. CONCLUSIÓN

La calculadora estadística cumple con las funcionalidades especificadas, el uso de las herramientas tecnológicas como Maven y Git, aceleran el proceso de desarrollo, estructurar bien los directorios y fuentes del proyecto, manejo conciso de las versiones y cambios, además de facilitar el acceso a librerías y frameworks externos. Las clases genéricas son una gran opción a la hora de desarrollar colecciones de datos. Las clases anónimas pueden ser una gran solución para evitar clases innecesarias y sin mucho sentido. Spark es un framework muy útil y intuitivo que nos permite dar respuesta a las peticiones HTTP muy fácilmente sin agregar abstracciones innecesarias a nuestro proyecto.

## REFERENCIAS

- [1] M. Frazier, *Singly Linked Lists*, <http://www.mikadoescoding.com/Singly-Linked-Lists/>, Accessed on 2020-08-19.
- [2] *Standard Deviation*, [https://en.wikipedia.org/wiki/Standard\\_deviation](https://en.wikipedia.org/wiki/Standard_deviation), Accessed on 2020-08-19.
- [3] *Mean*, <https://en.wikipedia.org/wiki/Mean>, Accessed on 2020-03-19.