

Mean and Standad Desviation

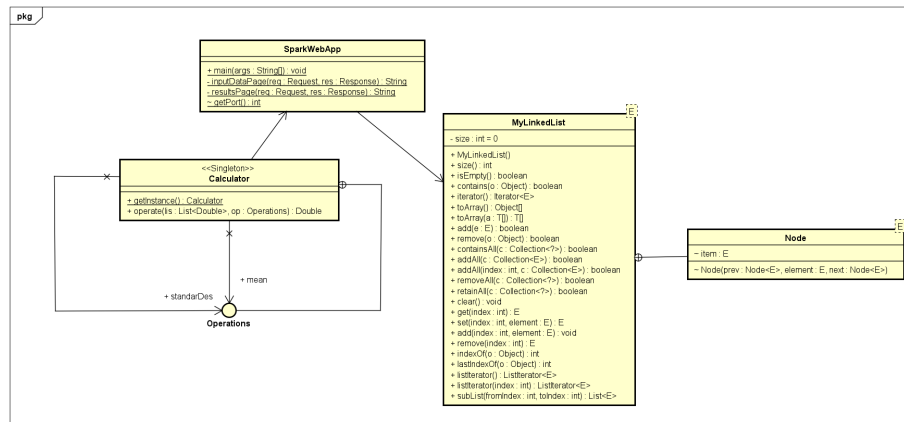
Andrés Felipe González Cárdenas

August 2020

1 Introducción

Este programa ofrece el servicio de calcular la **media** y la **desviación estándar** usando una interfaz web, en la cual el usuario ingresará manualmente los datos, en este documento podrá encontrar la explicación del diseño realizado, algunos test que se realizaron y al final las conclusiones que pudimos sacar después de realizar el programa.

2 Diseño



La clase **SparkWebApp** es la clase principal, la cual fue realizada con **SparkJava**, esta clase tiene dos métodos, uno con el que se le da al navegador el formulario **html** por el cual se van a recibir los datos de entrada y otro método el cual le enviará el resultado al navegador para que el usuario pueda ver el calculo de la media y desviación estándar. **SparkWebApp** al recibir los datos los almacena en la colección **MyLinkedList**, esta clase es nuestra propia implementación de una **LinkedList**, **SparkWebApp** además usa la clase **Calculator**, que es una clase **Singleton**, esta clase contiene la interfaz funcional **Operate** con el método **operation** el cual se usa para realizar la

implementación de las operaciones que queremos que tenga nuestra clase **Calculator**. La clase **Calculator** lleva como atributos **mean** y **standardDes**, las cuales son implementaciones usando funciones **lambda**.

3 Test

Para simular las entradas del usuario usaremos las siguientes listas con los datos de entrada.

```
String[] lis1 = {"160", "591", "114", "229", "230", "270", "128", "1657", "624", "1503"};  
String[] lis2 = {"15.0", "69.9", "6.5", "22.4", "28.4", "65.9", "19.4", "198.7", "38.8", "138.2"};
```

Luego haremos el mismo proceso que la clase **SparkWebApp**, el cual es usar la clase **Calculator** para realizar los calculos.

```
@Test  
public void meanPrueba1(){  
    Double res = cal.operate(l1, cal.mean);  
    assertEquals(550.6, res, 2);  
}
```

4 Conclusiones

- Al realizar el programa vimos el potencial que tiene el framework **Spark-Java**, el cual nos facilita la creación de apps web ahorrando muchas invocaciones que tendríamos que hacer.
- Al implementar la clase **Calculator** con una interfaz funcional para cada una de las operaciones, nos da la posibilidad de un futuro realizar una fácil extensión de nuestro proyecto, permitiéndonos adicionar otra operación usando solo una función **lambda** con la operación.
- Al implementar nuestra propia **LinkedList** podemos realizar facilmente una extensión de ella agregando cualquier operación que necesitemos, además nuestra lista está hecha para almacenar cualquier tipo de objeto.