

Spark and heroku

Realizado por Juan Guillermo Romero para Arquitecturas Empresariales

05 Febrero 2021

Resumen

En este informe se realiza el debido analisis y descripcion del proceso, dise~no y el manejo de diferentes herramientas para la construccion de un programa que tiene como finalidad realizar el calculo de la media y la desviacion estandar haciendo uso de LikendList pero con el valor agregado que esta sera una implementacion propia, y manejando el mini framework spark junto a heroku donde se realizo el despliegue.

1. Introducci3n

El proyecto Spark and Heroku tiene como fin realizar el calculo en cuanto a la media y la desviacion estandar de una serie de datos que seran leidos desde la pagina web realizada con el mini framework spark la cual esta corriendo en el servidor Heroku, estos datos despues de ser leidos se almacenaran de uno en uno en una implementacion PROPIA de la coleccion LikendList java, basada en esta se realizo una implementacion de una lista encadenada, y creando los diferentes metodos pertenecientes a esta. Seguido de esto el programa re dirigira a un path llamado resultados que como su nombre lo dice mostrara los resultados de la media y la desviacion.

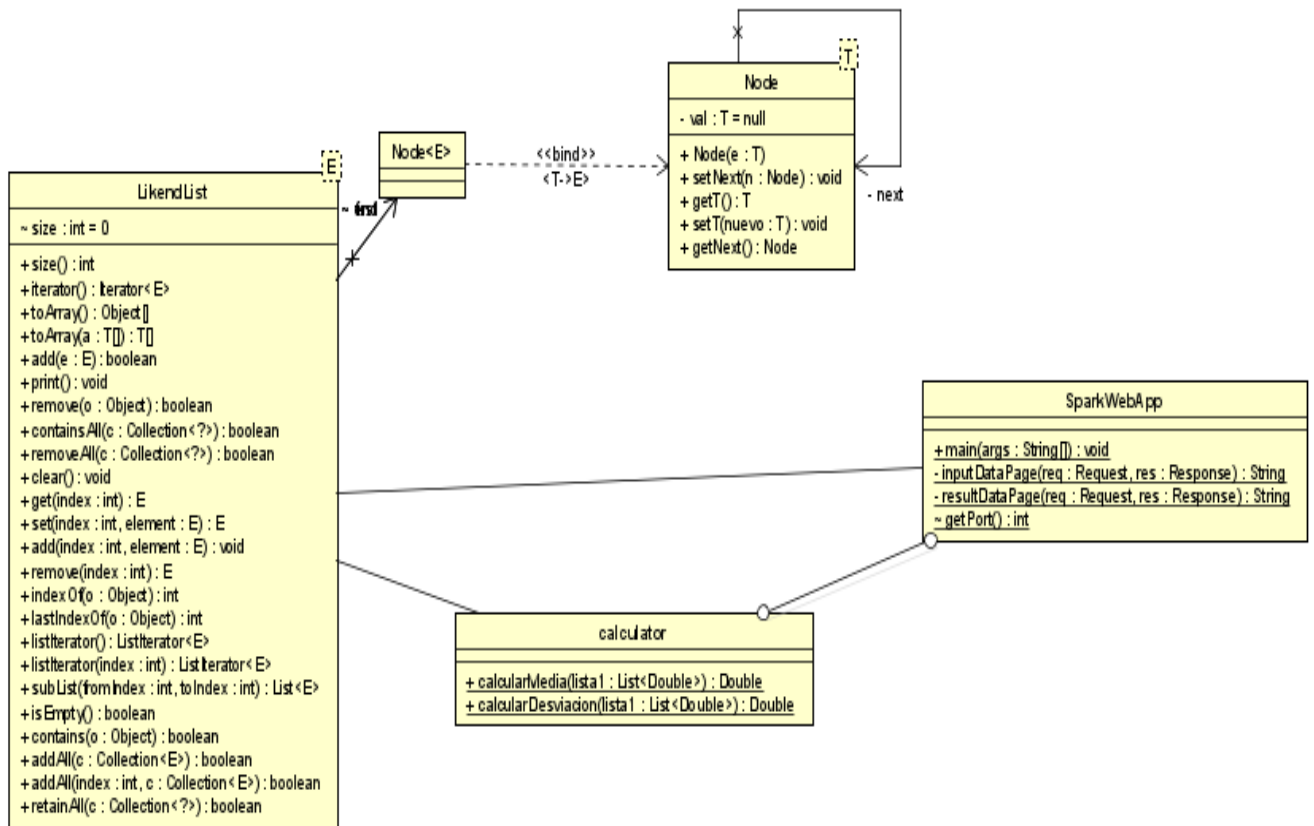
2. Dise~no

Se realizo el uso del mini framework Spark este tiene como finalidad desarrollar sus aplicaciones web de la manera m´as expresiva posible y con un m´ınimo de repetici3n. En este dise~no la clase SparkWeb es la encargada de generar el Frontend el cual obtendra los datos e imprimira los resultados para ello esta hace uso de la clase calculator que realiza el calculo de la media y la desviacion estandar y esta a la vez hace uso de nuestra LikendList y sus metodos.

La clase Calculator tiene dos metodos uno para calcular la media y otro para la desviacion estos recibiran como parametro una lista de tipo List, lo que se realizo fue pasar los datos obtenidos en el frontend a una lista tipo LikendList esta es una creacion propia que esta basada en una lista encadenada a partir de Nodos, para ello se crearon las dos clases LikendList y Node.

LikendList implementa List con todos sus respectivos metodos, esto para poder sobre escribirlos, y realizar una lista encadenada basada en nodos es decir creamos una clase Node la cual conocera un elemento en especifico de la lista y el nodo que esta antes que el en la lista, esto para que LikendList verifique si no han agregado ningun elemento este Node sera la cabeza de nuestra estructura y si ya tiene un elemento y se quiere ingresar otro, el nuevo sera la cabeza y se le dara a conocer quien iba antes y asi hasta formar la lista encadenada.

Se realizaron varias implementaciones de los metodos de LikedList como remove, iterator, get entre otros lo cuales se consideraron importantes para el manejo de los datos que tiene la lista LikendList.



3. Ejecucion de pruebas

A continuacion se muestra una muestra de que el programa fue exitoso gracias al dise~no y la estructura de codigo

```

C:\Windows\System32\cmd.exe
TESTS
-----
Running edu.escuelaing.arep.calculator.calculatorTest
calcularDesviacion2
62.26
calcularMedia2

entree60.32
calcularMedia
550.6
calcularDesviacion
572.03
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.227 sec
Running edu.escuelaing.arep.Linked.LikendListTest
print
size
add
iterator
Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.016 sec
Running edu.escuelaing.arep.sparkweb.SparkWebAppTest
main
getPort
Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.009 sec

Results :

Tests run: 10, Failures: 0, Errors: 0, Skipped: 0
  
```

Calcular media y desviacion estandar

ingrese los numeros separados por coma

160.0,591.0,114.0,229.0,230

Resultado media: 550.6

Resultado desviacion estandar: 572.026844746915

Calcular media y desviacion estandar

ingrese los numeros separados por coma

15.0,69.9,6.5,22.4,28.4,65.9,

Resultado media: 60.31999999999999

Resultado desviacion estandar: 62.25583060601186