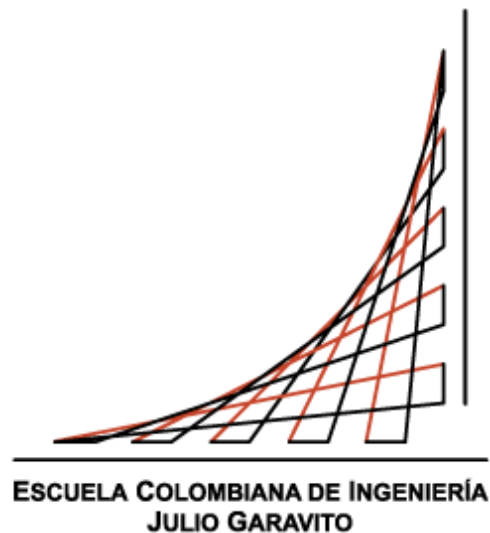


Taller 2 Arem

Fernando Barrera Barrera

Luis Daniel Benavides Navarro

Arquitecturas Empresariales



Índice

1. Introduccion	2
2. Conceptos Basicos	2
3. Diseño	2
4. Pre-Requisitos	3
5. Intalacion	3
6. Test	4
7. Ejecucion	4
8. Conclusion	5
9. Bibliografía	6

1. Introduccion

Este taller fue hecho para desarrollar una aplicacion web haciendo el uso del framework sparkweb que captura los datos de entrada por una vista de la pagina web que sera usados para el calculo de algunas operaciones estadisticas tales como la media aritmetica y la desviacion estandar haciendo uso de las estructuras de datos de listas encadenadas y devuelve la respuesta de esas operaciones por una vista de respuesta de la pagina web.

2. Conceptos Basicos

- **Maven:** Herramienta de software dedicada a la estructuturacion y construccion de proyectos java. [7]
- **Git:** software de control de versiones de proyectos [1]
- **Java :** es un lenguaje de programacion orientado a objetos que se desarrollo en los años 90 [5]
- **Heroku :** plataforma de despliegue de software que funciona de forma similar a un servicio de almacenamiento en la nube [3]
- **Spark :** es un framework que nos permite crear aplicaciones de forma sencilla y ligera [2]
- **LinkedList :** una estructura de datos que almacena nodos donde cada nodo tiene su respectivo valor y puede tener como referencia otros nodos [6]
- **Media :** es el valor caracteristicos en un conjunto de datos cuantitativos [8]
- **Desviacion Estandar :** la desvicion estandar es la medida que se utiliza para cuantificar la variacion de un conjunto de datos cuantitativos [8]
- **Funcion Lambda :** Es una funcion que funciona como una subrutina para construir una respuesta de una fucion de orden superior [4]

3. Diseño

Acontinuacion prodra observar el diagrama de clases de la aplicacion .

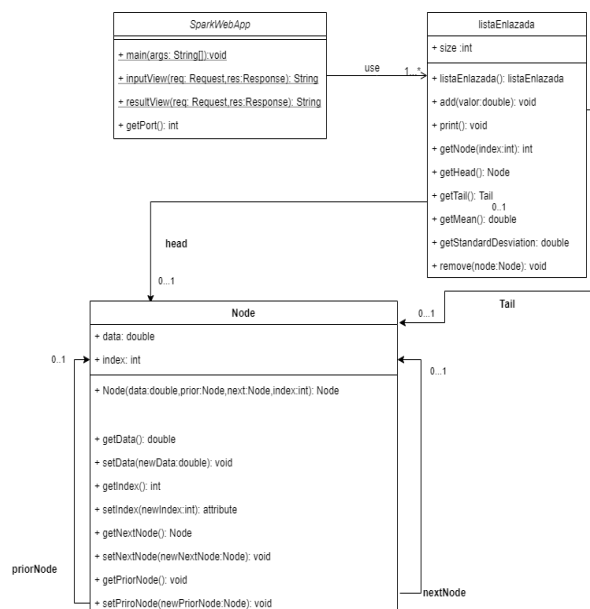


Figura 1:

La implementacion de esta aplicacion esta compuesta por tres clases ,la clase sparkwebapp que es la clase inicializadora que se encarg de iniciar el servidor web spark, la clase lista enlazada es una implementacion de una estructura de datos la cual almacena nodos y clase Nodo que tiene un valor,un indice y dos nodos nextNode y PriorNode enlazados.

la clase sparkwebapp es la encargada de levantar el servidor web definiendo unas peticiones y respuestas a apartir de algunas funciones lambda como porejemplo el metodo inuputView(req,res) es un metodo que recibe como parametros una request y un response y al final retorna un string html como response :

```
private static String inputView(Request req, Response res) {
    String view = "<DOCTYPE html>"
    + "<html>"
    + "<body style='background-color:#CCCC00;'>"
    + "<center>"
    + "<h2>Calculadora de Media y Desviacion estandar</h2>"
    + "<br><br>"
    + "<h2>Instrucciones : Porfavor ingrese todos los datos que desee separados por una coma ,para los valores flotantes use</h2>"
    + "<h2>Ejemplo : {2,3,5,6,7,8,9,3.5,4,1.8}</h2>"
    + "<form action='/'results'>"
    + "  Datos = {"
    + "    <input type='text' name='datos'>"
    + "  }"
    + "  <br><br>"
    + "  <input type='submit' value='Calcular'>"
    + "</form>"
    + "</center>"
    + "</body>"
    + "</html>";
    return view;
}
```

Figura 2:

el otro metodo que recibe como parametros una request y un response es el metodo resultsView que es llamado por inputView ya que la vista input al capturar los datos que el usuario ingreso manda una peticion al resultsView el cual se encarga de realizar los calculos y visualizarlos en pantalla :

```
private static String resultsView(Request req, Response res) {
    String datos[] = req.queryParams("datos").split(",");
    listaEnlazada linkedlist = new listaEnlazada();
    for(String d:datos) {
        linkedlist.add(Double.parseDouble(d));
    }
    double media = linkedlist.getMean();
    double desviacionEstandar = linkedlist.getStandardDesviation();
    String view = "<DOCTYPE html>"
    + "<html>"
    + "<body style='background-color:#32CD32;'>"
    + "<style>"
    + "table, th, td {"
    + "border: 1px solid black;"
    + "border-collapse: collapse;"
    + "}"
    + "</style>"
    + "<center>"
    + "<h2>Calculadora de Media y Desviacion estandar</h2>"
    + "  <br><br>"
    + "<h2>Acontinucion encontrara una tabla con sus respectivos resultados"
```

Figura 3:

4. Pre-Requisitos

- Git
- Java
- Maven

5. Intalacion

Para iniciar la instalacion del programa primero clone el repositorio donde se encuentra alojado el programa desde la consola de comandos

```
C:\Users\jm_14\Downloads>git clone https://github.com/fernando-b15/Arem-Taller2
Cloning into 'Arem-Taller2'...
remote: Enumerating objects: 88, done.
remote: Counting objects: 100% (88/88), done.
remote: Compressing objects: 100% (50/50), done.

Unpacking objects: 100% (88/88), done.
C:\Users\jm_14\Downloads>
```

Figura 4:

Despues se procede a entrar al directorio donde se encuentra el programa y se procede a empaquetarlo con el comando mvn package:

```
INFO] Copying jetty-webapp-9.4.30.v20200611.jar to C:\Users\jm_14\Downloads\Arem-Taller2\target\dependency\jetty-webapp-9.4.30.v20200611.jar
[INFO] Copying jetty-xml-9.4.30.v20200611.jar to C:\Users\jm_14\Downloads\Arem-Taller2\target\dependency\jetty-xml-9.4.30.v20200611.jar
[INFO] Copying jetty-servlet-9.4.30.v20200611.jar to C:\Users\jm_14\Downloads\Arem-Taller2\target\dependency\jetty-servlet-9.4.30.v20200611.jar
[INFO] Copying jetty-security-9.4.30.v20200611.jar to C:\Users\jm_14\Downloads\Arem-Taller2\target\dependency\jetty-security-9.4.30.v20200611.jar
[INFO] Copying websocket-server-9.4.30.v20200611.jar to C:\Users\jm_14\Downloads\Arem-Taller2\target\dependency\websocket-server-9.4.30.v20200611.jar
[INFO] Copying websocket-common-9.4.30.v20200611.jar to C:\Users\jm_14\Downloads\Arem-Taller2\target\dependency\websocket-common-9.4.30.v20200611.jar
[INFO] Copying websocket-client-9.4.30.v20200611.jar to C:\Users\jm_14\Downloads\Arem-Taller2\target\dependency\websocket-client-9.4.30.v20200611.jar
[INFO] Copying jetty-client-9.4.30.v20200611.jar to C:\Users\jm_14\Downloads\Arem-Taller2\target\dependency\jetty-client-9.4.30.v20200611.jar
[INFO] Copying websocket-servlet-9.4.30.v20200611.jar to C:\Users\jm_14\Downloads\Arem-Taller2\target\dependency\websocket-servlet-9.4.30.v20200611.jar
[INFO] Copying websocket-api-9.4.30.v20200611.jar to C:\Users\jm_14\Downloads\Arem-Taller2\target\dependency\websocket-api-9.4.30.v20200611.jar
[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 15.563 s
[INFO] Finished at: 2020-08-20T18:55:26-05:00
[INFO] -----
C:\Users\jm_14\Downloads\Arem-Taller2>
```

Figura 5:

6. Test

Ahora procedemos a ejecutar las pruebas de la aplicacion desde la consola de comandos como se vera acontinuacion

```
[INFO] --- maven-surefire-plugin:2.12.4:test (default-test) @ Arem-Taller2 ---
[INFO] Surefire report directory: C:\Users\jm_14\Downloads\Arem-Taller2\target\surefire-reports

-----
T E S T S
-----
Running edu.escuelaing.aren.AppTest
Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.092 sec

Results :

Tests run: 6, Failures: 0, Errors: 0, Skipped: 0

[INFO] -----
[INFO] BUILD SUCCESS
[INFO] -----
[INFO] Total time: 3.105 s
[INFO] Finished at: 2020-08-20T18:55:58-05:00
[INFO] -----
C:\Users\jm_14\Downloads\Arem-Taller2>
```

Figura 6:

7. Ejecucion

Para ejecutar el programa basta con acceder a linea de comandos windows y digitar el siguiente comando que encendera el servidor web spark

```
C:\Users\jm_14\Downloads\Arem-Taller2>java -cp target\classes;target\dependency\* edu.escuelaing.aren.SparkWebApp
SLF4J: Failed to load class "org.slf4j.impl.StaticLoggerBinder".
SLF4J: Defaulting to no-operation (NOP) logger implementation
SLF4J: See http://www.slf4j.org/codes.html#StaticLoggerBinder for further details.
```

Figura 7:

Ahora solo basta con abrir el buscador y digitar localhost:4567 como se vera acontinuacion:

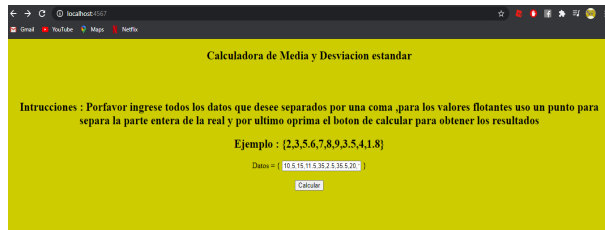


Figura 8:

Una vez usuario digite los datos necesarios con el boton calcular se enviara una peticion a resultsView la cual se encargara de recibir los datos digitados por el usuario y almacenarlos en la lista enlazada para realizar los calculos y visualizarlos en pantalla como se vera acontinuacion:



Figura 9:

8. Conclusion

Este taller fue util para explorar las funcionalidades del framework spark para el desarrollo de servidores web de forma sencilla y ligera definiendo vistas atraves de funciones que son definidas a su vez por funciones lambda y ademas tambien nos ayudo a comprender de una mejor manera como funciona un microservicio

9. Bibliografía

- [1] Código Facilito. *Git*. URL: <https://codigofacilito.com/articulos/que-es-git>. (entered: 16-08-2015).
- [2] Spark Framework. *Spark Framework*. URL: https://es.wikipedia.org/wiki/Spark_Framework. (entered: 20-09-2019).
- [3] Heroku. *Heroku*. URL: <https://es.wikipedia.org/wiki/Heroku>. (entered: 20-12-2019).
- [4] Función Lambda. *Función Lambda*. URL: <https://www.tokioschool.com/noticias/expresiones-lambda-uso-programacion-aplicaciones/>. (entered: 2020).
- [5] Wikipedia. *Java*. URL: [https://es.wikipedia.org/wiki/Java_\(https://es.wikipedia.org/wiki/Java\)](https://es.wikipedia.org/wiki/Java_(https://es.wikipedia.org/wiki/Java)). (entered: 23-06-2020).
- [6] Wikipedia. *LinkedList*. URL: https://es.wikipedia.org/wiki/Lista_enlazada. (entered: 13-06-2020).
- [7] Wikipedia. *Maven*. URL: <https://es.wikipedia.org/wiki/Maven>. (entered: 31-03-2020).
- [8] Wikipedia. *Media*. URL: https://es.wikipedia.org/wiki/Media_aritmetica. (entered: 28-07-2020).