



ESCUELA
COLOMBIANA
DE INGENIERÍA
JULIO GARAVITO

LABORATORIO SPARK - CALCULADORA ESTADÍSTICA

LUIS DANIEL BENAVIDES NAVARRO

Arquitecturas Empresariales

AUTOR:

JOHANN SEBASTIAN PÁEZ CAMPOS

AGOSTO 12 DEL 2020

Tabla de Contenidos

1. Introducción	2
2. Diseño y Arquitectura	2
3. Pruebas	3
4. Conclusiones	5
5. Referencias	5

1. Introducción

En este escrito se mostrará y se detallará la continuación del laboratorio anterior en el cual se realizó un programa que lee números de un archivo y calcula la media y desviación estándar. Para esta continuación se hará, una página web que permita utilizar la aplicación y realizar los mismos calculos utilizando Spark.

2. Diseño y Arquitectura

En el aplicativo tiene una arquitectura muy sencilla en cuanto a la parte web, si quiere ver la arquitectura de la lógica, puede consultar el diseño en la entrega anterior.

La arquitectura de la aplicación es cliente-servidor, normalmente interactuamos mucho con esa arquitectura con aplicativos web (como lo es el caso). En la figura 1, podemos observar un cliente (pc), que al momento de realizar una petición (en este caso post), envia los datos (números a ser calculados) y el servidor de heroku donde está desplegada la aplicación, responde y muestra la información para el usuario.

Post /operation: Esta petición se encarga de recibir la información enviada por el usuario y realizar los respectivos calculos (media y desviación estándar) para posteriormente ser mostrados. [1]

Get /: Al momento de recibir la información por el post, el servidor se encarga de actualizar los datos y el cliente (pc) tiene que interpretarlos para así ver los resultados finales. [2]

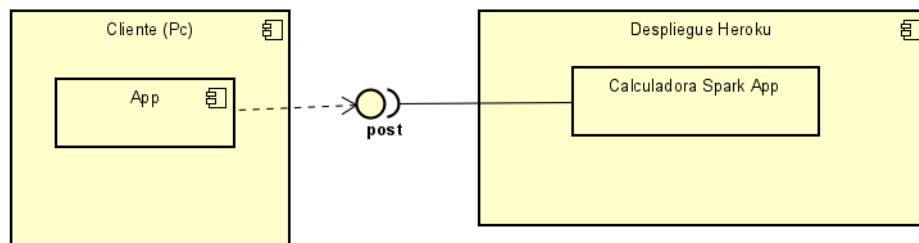


Figura 1: Representación básica de la arquitectura descrita anteriormente.

3. Pruebas

Verificando nuevamente que el aplicativo web cumple con los requisitos, se utilizarán los casos dados en la guía y vamos a comparar los resultados obtenidos con los valores dados.

En la figura 2, se puede observar dos tablas. La tabla 1 tiene dos columnas con una lista de números y la tabla 2 tiene el valor esperado de la media y desviación estándar para estos valores dados.

Column 1 Estimate Proxy Size	Column 2 Development Hours
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

Table 1

Test	Expected Value		Actual Value	
	Mean	Std. Dev	Mean	Std. Dev
Table 1: Column 1	550.6	572.03		
Table 1: Column 2	60.32	62.26		

Table 2

Figura 2: Tablas con la información de la media y desviación estándar de dos listas de números.

Como podemos observar en la figura 2, la media y desviación estándar para la columna 1 es 550.6 y 572.03 respectivamente y para la columna 2 es 60.32 y 62.26.

Ahora procedemos a calcular la media y desviación estándar de estas dos listas de números con el aplicativo web desplegado en heroku.

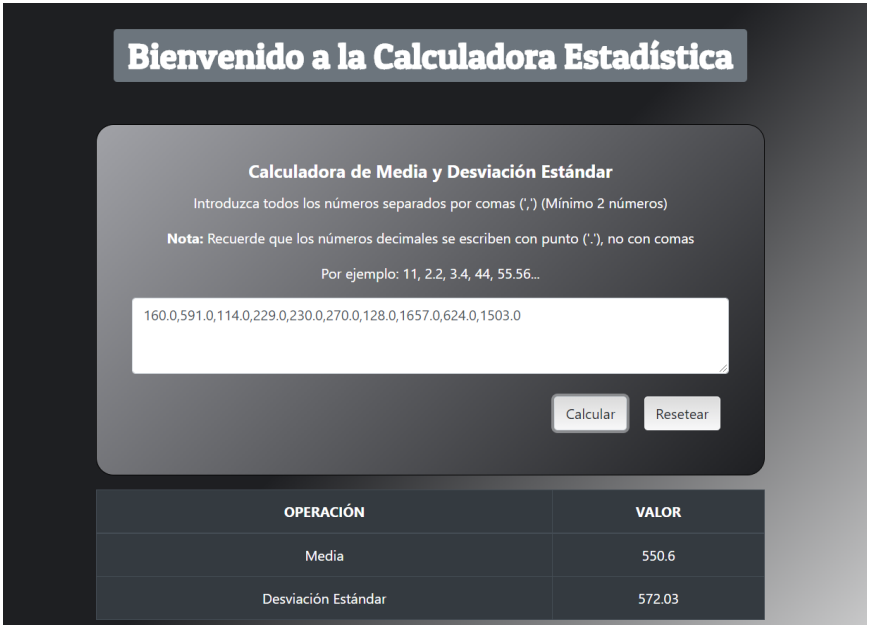


Figura 3: Resultados de los dos casos de la figura 3.



Figura 4: Resultados de los dos casos de la figura 3.

En la figura 3, 4 se exponen los resultados del programa creado y vemos que son iguales a los valores dados en la figura 2.

4. Conclusiones

- El aplicativo web funciona igualmente que por consola y cumple con los requisitos establecidos.
- Una manera fácil de crear una aplicación web es utilizar SPARK y desplegar el aplicativo con Heroku.

5. Referencias

- [1] *POST*, <https://developer.mozilla.org/es/docs/Web/HTTP/Methods/POST>, Accessed on 2020-08-20.
- [2] *HTTP Request / Response*, <https://sites.google.com/site/conceptoprogramacion/request-response>, Accessed on 2020-08-20.