

Introducción a Heroku

Germán Andrés Ospina Quintero

5 de Febrero del 2021

Contents

1	Glosario	2
2	Resumen	4
3	Introducción	5
4	Estructura	6
5	Diseño	7
6	Arquitectura	8
7	Pruebas	9

1 Glosario

Java: Java es un tipo de lenguaje de programación y una plataforma informática, creada y comercializada por Sun Microsystems en el año 1995. Se constituye como un lenguaje orientado a objetos, su intención es permitir que los desarrolladores de aplicaciones escriban el programa una sola vez y lo ejecuten en cualquier dispositivo. (Content, 2019)

Git: Git es un software de control de versiones diseñado por Linus Torvalds, pensando en la eficiencia, la confiabilidad y compatibilidad del mantenimiento de versiones de aplicaciones cuando éstas tienen un gran número de archivos de código fuente. (Wikipedia, 2021)

Linked List: en informática, una lista enlazada es una colección lineal de elementos de datos cuyo orden no viene dado por su ubicación física en la memoria. En cambio, cada elemento apunta al siguiente. Es una estructura de datos que consta de una colección de nodos que juntos representan una secuencia. (Wikipedia, 2020)

Maven: Maven es una herramienta de software para la gestión y construcción de proyectos Java creada por Jason van Zyl, de Sonatype, en 2002. Es similar en funcionalidad a Apache Ant, pero tiene un modelo de configuración de construcción más simple, basado en un formato XML. (Wikipedia, 2020)

Spark: es un conjunto de librerías para el desarrollo de aplicaciones web en Java inspirado en el framework Sinatra para Ruby. Spark incluye por defecto el servidor de aplicaciones Jetty de manera que las aplicaciones web creadas con Spark pueden lanzarse como cualquier

programa Java. (Wikipedia, 2020)

2 Resumen

Para este taller, se hizo necesario realizar la implementación, personalizada, de una Linked List, a partir de la interfaz proporcionada por la librería de Java, en donde se pudieran almacenar una serie de datos con los cuales se pudiera calcular la media y la desviación estándar de estos; estos datos son obtenidos através de un cliente desarrollado en JavaScript y procesados, en el Backend, con la creación de una API desarrollada con el Framework Spark. Adicionalmente, se realizaron pruebas unitarias para validar que los cálculos sean correctos, una documentación de cada método, viéndose plasmada en el archivo Javadoc, y la elaboración de un README en donde se puede apreciar el paso a paso para la instalación y ejecución del proyecto. El proyecto, en su totalidad, fue alojado en repositorio en la plataforma GitHub.

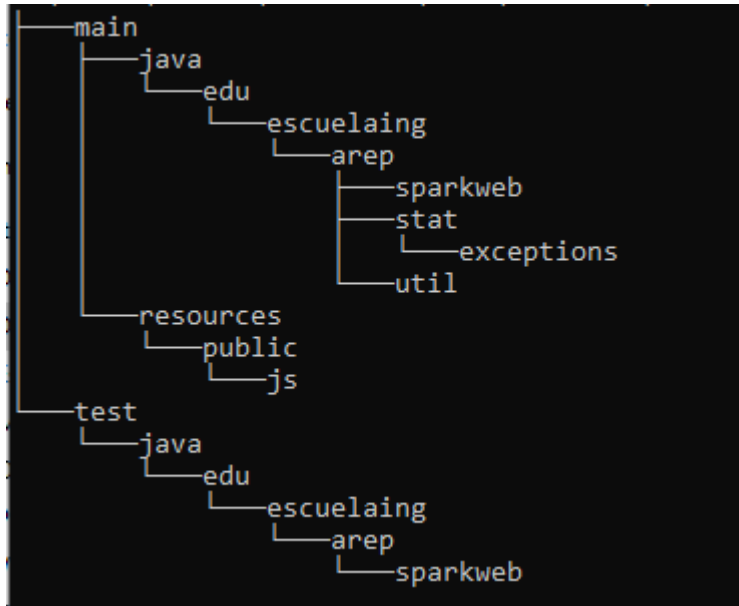
3 Introducción

Inicialmente, haciendo uso de las herramientas para diseñar el código fuente, incluyendo el framework para desarrollo de aplicaciones web: Spark, desplegarlo y hacer integración continua, maven, Heroku y Circle CI respectivamente, se desarrollo una aplicación web que permite calcular la media y desviación estandar de un conjunto de números reales. Estos numeros, son ingresados a través del cliente desarrollado con JavaScript y JQuery.

En este documento, se presenta y se describe la estructura, el diseño y arquitectura de la aplicación, a su vez, se identifica claramente las abstracciones de memoria, canales de comunicaciones e interpretes

Por último, se hace un análisis de las pruebas hechas para validar la correctitud de la aplicación

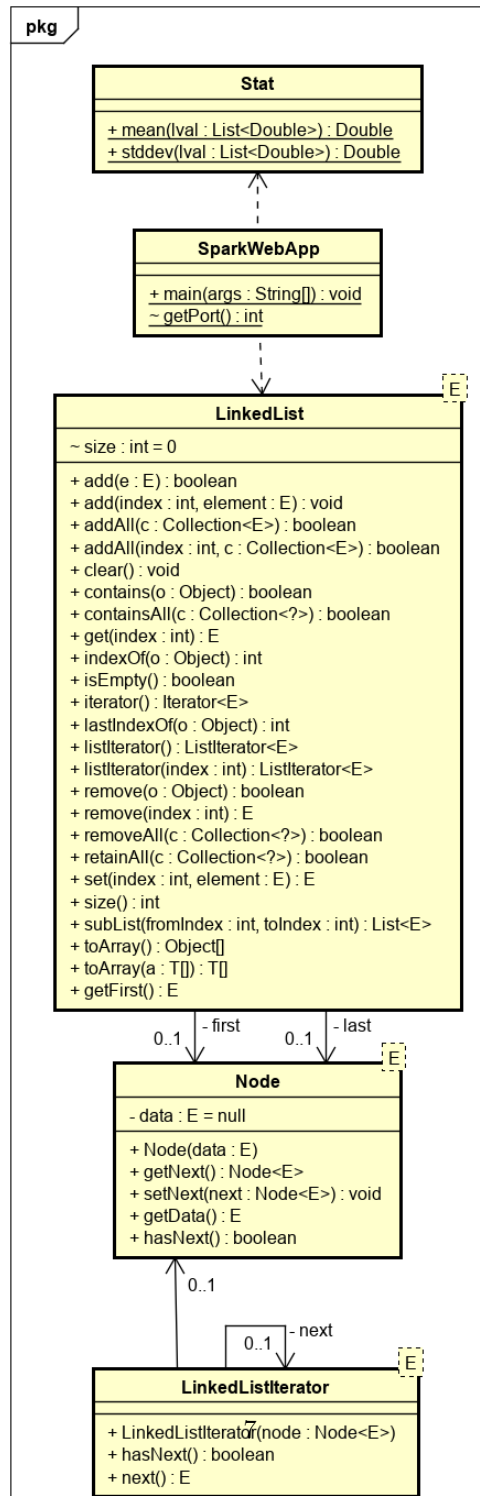
4 Estructura



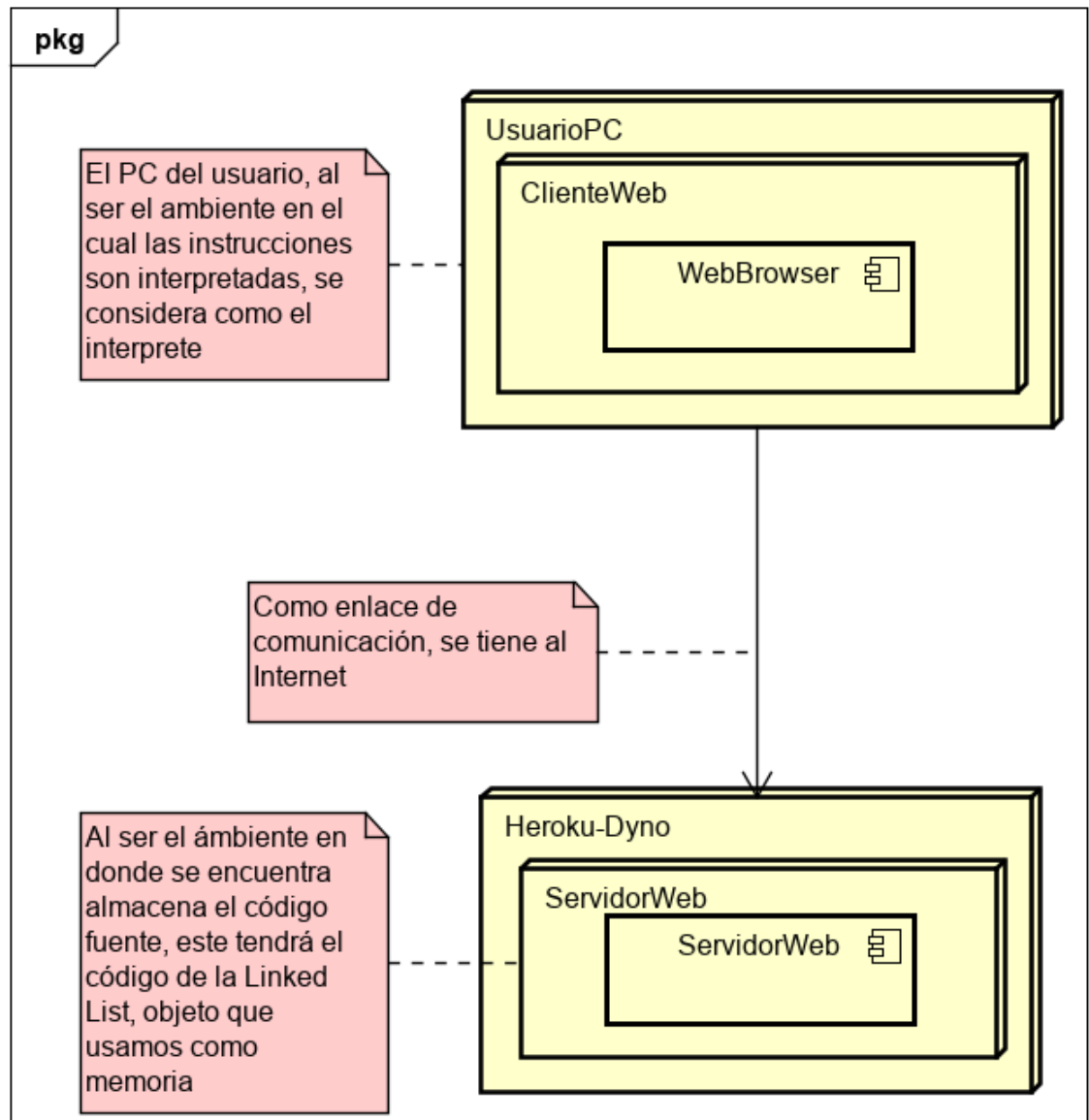
La aplicación presenta dos carpetas principales: main y test. La primera, presenta todo el código fuente de la aplicación, estructurado en la carpeta arep donde se encuentran los paquetes: sparkweb, en donde se encuentra la implementación del servidor web, stat, con la implementación de las operaciones requeridas, media y desviación estándar, y util, en donde se desarrolla la implementación de la Linked List a partir de la interface. También, tiene un directorio llamado resources que contiene toda la implementación del cliente web.

La segunda, contiene las cuatro pruebas hechas al código fuente de la aplicación.

5 Diseño



6 Arquitectura



7 Pruebas

Como se mencionó anteriormente, se realizaron 4 pruebas para validar la correctitud del funcionamiento de la aplicación

La primera prueba valida que se calcule correctamente la media de los datos de la primera columna

```
@Test
public void shouldCalculateMeanFirstColumn() throws IOException{
    String data = "160%20591%20114%20229%20230%20270%20128%201657%20624%201503";
    URL url = new URL("https://sparkwebapp.herokuapp.com/mean?list=" + data);
    URLConnection urlConnection = url.openConnection();
    BufferedReader read = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()))
    String inputLine = read.readLine();
    assertTrue(inputLine.equals("550.6"));
    read.close();
}
```

La segunda prueba valida que se calcule correctamente la media de los datos de la segunda columna

```
@Test
public void shouldCalculateMeanSecondColumn() throws IOException{
    String data = "15.0%2069.9%206.5%2022.4%2028.4%2065.9%2019.4%20198.7%2038.8%20138.2";
    URL url = new URL("https://sparkwebapp.herokuapp.com/mean?list=" + data);
    URLConnection urlConnection = url.openConnection();
    BufferedReader read = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()))
    String inputLine = read.readLine();
    assertTrue(inputLine.equals("60.32"));
    read.close();
}
```

La tercera prueba valida que se calcule correctamente la desviación estándar de los datos de la primera columna

```

@Test
public void shouldCalculateStddevFirstColumn() throws IOException{
    String data = "160%20591%20114%20229%20230%20270%20128%201657%20624%201503";
    URL url = new URL("https://sparkwebapp.herokuapp.com/stddev?list=" + data);
    URLConnection urlConnection = url.openConnection();
    BufferedReader read = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()))
    String inputLine = read.readLine();
    assertTrue(inputLine.equals("572.03"));
    read.close();
}

```

La cuarta prueba valida que se calcule correctamente la desviación estándar de los datos de la segunda columna

```

@Test
public void shouldCalculateStddevSecondColumn() throws IOException{
    String data = "15.0%2069.9%206.5%2022.4%2028.4%2065.9%2019.4%20198.7%2038.8%20138.2";
    URL url = new URL("https://sparkwebapp.herokuapp.com/stddev?list=" + data);
    URLConnection urlConnection = url.openConnection();
    BufferedReader read = new BufferedReader(new InputStreamReader(urlConnection.getInputStream()))
    String inputLine = read.readLine();
    assertTrue(inputLine.equals("62.26"));
    read.close();
}

```