

Pontificia Universidad Católica del Ecuador | SEDE ESMERALDAS

ESCUELA DE SISTEMAS Y COMPUTACIÓN

TESIS DE GRADO

USO DEL STACK MEAN PARA EL DESARROLLO DE UNA APLICACIÓN WEB ORIENTADA A LA TECNOLOGÍA BIG DATA

LÍNEA DE INVESTIGACIÓN

PROGRAMACIÓN Y DESARROLLO DE SOFTWARE

PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO DE SISTEMAS Y COMPUTACIÓN

AUTOR

SÁNCHEZ TARIRA MARCELO PATRICIO

ASESOR

MGT. JAIME SAYAGO

Esmeraldas – Enero, 2019

Tesis de grado aprobada luego de haber dado cumplimiento a los requisitos exigidos por el reglamento de grado de la PUCE Esmeraldas, previo a la obtención del título de INGENIERO DE SISTEMAS Y COMPUTACIÓN.

TRIBUNAL DE GRADUACIÓN

Título: "USO DEL STACK MEAN PARA EL DESARROLLO DE UNA APLICACIÓN WEB ORIENTADA A LA TECNOLOGÍA BIG DATA"

Autor: SÁNCHEZ TARIRA MARCELO PATRICIO

Mgt. Jaime Sayago Heredia f.-……
Asesor

Mgt. Wilson Chango Sailema f.-……
Lector N° 1

Mgt. Susana Patiño Rosado f.-……
Lector N° 2

Mgt. Xavier Quiñónez Kú f.-……
Director de Escuela

Ing. Maritza Demera Mejía f.-……...
Secretaria General PUCE Esmeraldas

Esmeraldas …. de ……. de 2019

DECLARACIÓN DE AUTORÍA

Yo, Sánchez Tarira Marcelo Patricio, portador de la cédula de identidad No. 0850015629, declaro que la presente investigación enmarcada en el actual trabajo de tesis es absolutamente original, auténtica y personal.

En tal virtud, manifiesto que el contenido, las conclusiones y resultados, y los efectos legales y académicos que se desprendan del trabajo propuesto de investigación y luego de la redacción de este documento son y serán de mi única y exclusiva responsabilidad legal y académica.

Sánchez Tarira Marcelo Patricio,
C.I.: 0850015629

CERTIFICACIÓN

Yo Mgt. Jaime Sayago Heredia, docente investigador de la PUCE Sede Esmeraldas, certifico que:

El trabajo de grado realizado por SÁNCHEZ TARIRA MARCELO PATRICIO bajo el título "USO DEL STACK MEAN PARA EL DESARROLLO DE UNA APLICACIÓN WEB ORIENTADA A LA TECNOLOGÍA BIG DATA" reúne los requisitos de calidad, originalidad y presentación exigibles a una investigación científica, y que además han sido incorporadas al documento final las sugerencias realizadas, en consecuencia, está en condiciones de ser sometido a la valoración del Tribunal encargado de juzgarlo.

Y para que conste a los efectos oportunos, firmo la presente en Esmeraldas, el …. de ……. de 2019.

Mgt. Jaime Sayago Heredia
Asesor

AGRADECIMIENTOS

A mis padres, hermanos y toda mi familia, por el apoyo constante y el cariño incondicional brindado a cada momento durante cada día.
Ustedes son los pilares de mi vida

A mis amigos, por alentarme cuando más lo necesito, por extender su mano en momentos difíciles y por estar siempre presentes dandome risas y alegrías.
Me faltarían páginas para describir todas las locuras y los buenos momentos a su lado.

A los docentes de la Escuela de Sistemas de la PUCE Sede Esmeraldas por haber compartido sus conocimientos a lo largo de mi formación como profesional, en especial a mi asesor y lectores, por servir de guía durante la realización de este trabajo.

¡Que nadie se quede afuera, les agradezco a todos!

Gracias desde lo más profundo de mi corazón.

DEDICATORIA

El presente trabajo de investigación, al igual que todos mis logros pasados y futuros estan completamente dedicados a mis padres, Wilson Eduardo Sánchez Lucas y Tatiana Esther Tarira España, quienes, con su comprensión, apoyo, sabiduría y sobre todo amor, han sabido formarme durante toda mi carrera estudiantil, y sobre todo, durante toda mi vida.

A ustedes les dedico este logro.
A ustedes les dedico mi vida entera.

ÍNDICE

RESUMEN 1
ABSTRACT 2
INTRODUCCIÓN 3
Presentación de la investigacion 3
Planteamiento del problema 3
Justificación 6
Objetivos 8
Objetivo General 8
Objetivos Específicos 8
CAPÍTULO I 9
MARCO DE REFERENCIA 9
1.1. ANTECEDENTES 9
1.2 BASES TEÓRICAS 12
1.2.1. STACKS DE DESARROLLO 12
1.2.2. STACK DE DESARROLLO MEAN 14
1.2.3. BIG DATA 16
1.2.4. TECNOLOGÍA BIG DATA Y APLICACIONES WEB 27
CAPÍTULO II 34
METODOLOGÍA 34
2.1. TIPO DE INVESTIGACIÓN 34
2.2. MÉTODOS Y TÉCNICAS 34
2.3 POBLACIÓN Y MUESTRA DE ESTUDIO 35
2.4. DESCRIPCIÓN Y VALIDACIÓN DEL INSTRUMENTO 35
2.5. TÉCNICAS DE PROCESAMIENTO Y ANÁLISIS 38
2.6. NORMAS ÉTICAS 41
CAPÍTULO III 42
RESULTADOS 42
3.1. APLICACIÓN WEB BIG DATA MEDIANTE EL STACK MEAN 42
3.2. HERRAMIENTAS USADAS EN EL ANÁLISIS BIG DATA 48
3.3. PARADIGMA BUSINESS INTELLIGENCE PARA EL ANÁLISIS DE INFORMACIÓN 50

3.4. EJECUCIÓN DE PRUEBAS....62
3.4.1. PRUEBAS DE CARGA Y ESTRÉS....62
3.4.2. PRUEBA DE TIEMPO DE RESPUESTA....67
3.5. ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS OBTENIDOS EN LAS PRUEBAS....70
3.5.1. ANÁLISIS DE RESULTADOS DE LA PRUEBA DE CARGA....71
3.5.2. ANÁLISIS DE RESULTADOS DE LA PRUEBA DE ESTRES....76
3.5.3. ANÁLISIS DE RESULTADOS DE LA PRUEBA DE TIEMPO DE RESPUESTA....79
CAPÍTULO IV....83
DISCUSIÓN....83
VENTAJAS Y DESVENTAJAS DEL STACK DE DESARROLLO MEAN COMO HERRAMIENTA PARA BIG DATA....83
CAPÍTULO V....87
5.1. CONCLUSIONES....87
5.2. RECOMENDACIONES....88
CAPÍTULO VI....90
6.1. REFERENCIAS BIBLIOGRÁFICAS....90
6.2. ANEXOS....94

ÍNDICE DE ILUSTRACIONES

Figura 1. Funcionamiento General del stack de desarrollo MEAN 16
Figura 2. Estadísticas mundiales del uso del internet y redes sociales. Agosto 2015....19
Figura 3. Estadísticas mundiales del uso del internet y redes sociales. Abril 2017.....20
Figura 4. Volumen de datos estimados a nivel mundial y aproximaciones futuras.....21
Figura 5. Ejemplo de un contador de palabras aplicando el paradigma MapReduce 25
Figura 6. Diagrama de Casos de uso de la aplicación desarrollada 36
Figura 7. Pantalla "Personas" de la aplicación web desarrollada 42
Figura 8. Pantalla "Timeline" de la aplicación web Desarrollada 43
Figura 9. Estructura principal de la aplicación 44
Figura 10. Principales colecciones de la base de datos 45
Figura 11. Ejemplo básico de un documento de la colección "users".....46
Figura 12. Ejemplo de script usado para generar datos de usuarios aleatorios 47
Figura 13. Ejemplo del JSON generado de forma aleatoria 48
Figura 14. Proceso básico Extract Transform Load aplicado a la base de datos.....51
Figura 15. Ejecución del proceso Extract Transform Load aplicado a los datos.....52
Figura 16. Ejemplo de la configuración realizada para el punto 1. 54
Figura 17. Ejemplo de la configuración realizada para el punto 1.1.2.....55
Figura 18. Ejemplo de la configuración realizada para el punto 4. 55
Figura 19. Principales colecciones de la base de datos 56
Figura 20. Configuración realizada para la conexión al DataWarehouse 56
Figura 21. Ejemplo de pantalla de diseño con un gráfico predeterminado 57
Figura 22. Ejemplo de la configuración necesaria para generar un reporte 58
Figura 23. Ejemplo de gráfico con estadísticas de varios gustos de usuarios 59
Figura 24. Ejemplo de gráfico con estadísticas de varios gustos clasificados.....60
Figura 25. Ejemplo de gráfico con estadísticas de usuarios clasificados 61
Figura 26. Interfaz Gráfica JMeter. Pantalla con script de una Consulta Simple 63
Figura 27. Interfaz Gráfica JMeter. Pantalla con resultados de ejecución 64
Figura 28. Interfaz Gráfica JMeter. Pantalla tabla de resumen de resultados 65
Figura 29. Interfaz online de la herramienta Blazemeter. Pantalla de configuración 66
Figura 30. Interfaz online de la herramienta Blazemeter. Ejemplo de ejecución 66
Figura 31. Interfaz Gráfica JMeter. Pantalla con ejemplo de petición login 67

Figura 32. Interfaz Gráfica JMeter. Pantalla con resultados de las peticiones 68
Figura 33. Interfaz Gráfica JMeter. Pantalla tabla de resumen de resultados 69
Figura 34. Interfaz Gráfica JMeter. Pantalla del gráfico de resumen 69
Figura 35. Interpretación Gráfica de los resultados obtenidos 74
Figura 36. Interpretación Gráfica de los resultados obtenidos 78

ÍNDICE DE TABLAS

Tabla 1. Comparativa de stack LAMP y derivados.....13
Tabla 2. Ejemplo de Requerimiento Funcional 37
Tabla 3. Ejemplo de No Requerimiento Funcional.....37
Tabla 4. Pruebas de Software No Funcionales a realizar 40
Tabla 5. Tiempos de respuesta promedio 70
Tabla 6. Tiempos de respuesta promedio 71
Tabla 7. Resultado de la Prueba de Carga 73
Tabla 8. Porcentajes de Tiempo de Respuesta.....75
Tabla 9. Resultado de la Prueba de Estrés.....77
Tabla 10. Resultado de la Prueba de Tiempo de Respuesta 80
Tabla 11. Porcentaje de Tiempo de Respuesta 81

RESUMEN

La presente investigación fue desarrollada con el propósito de analizar el funcionamiento del stack MEAN y cada uno de sus respectivos componentes, aplicándolo en la creación de una aplicación orientada a la tecnología Big Data y todo el proceso que involucra, en cuanto a esto, se llevó a cabo una investigación documental para determinar las definiciones de dicha tecnología, así como sus características, haciendo énfasis en sus posibles usos y la manera en la cual se puede generar conocimiento desde los datos adquiridos mediante el paradigma Business Intelligence y la aplicación del proceso Extract Transform Load sobre la información generada.

Por otro lado, la aplicación antes mencionada, fue desarrollada para ser capaz de soportar grandes cantidades de información, desde su recolección hasta su posterior análisis final, y la generación de un Data Warehouse con la información obtenida. Para determinar la factibilidad del stack MEAN como herramienta para el desarrollo de aplicaciones Big Data, fue necesario llevar a cabo un análisis investigativo de sus ventajas y desventajas, teniendo como base la aplicación de las respectivas pruebas de software y los resultados generados, emitiendo un criterio respecto al comportamiento de la aplicación en un estado de producción.

Palabras clave: stack, desarrollo, MEAN, big data, aplicación web, extract-transform-load, business intelligence, mongo, express, angular, node, javascript, data warehouse.

ABSTRACT

The present investigation was created with the purpose of analyzing the functioning of the MEAN stack and each one of its respective components, in order to develop an application oriented to Big Data technology and all the process that involves, in this regard, it was needed a documentary research to determinate the definitions of this technology, as well as its characteristics, emphasizing its possible uses and the way in which knowledge can be generated from the data acquired through the Business Intelligence paradigm and the application of the Extract Transform Load process over the information generated.

On the other hand, the aforementioned application was developed to be able to support large amounts of information, from its collection to its subsequent final analysis, and the generation of a Data Warehouse with the information obtained. To determine the feasibility of the MEAN stack as a tool for the development of Big Data applications, it was necessary to carry out an investigative analysis of its advantages and disadvantages, based on the application of the respective software tests and the results generated, issuing a criterion regarding the behavior of the application in a production state.

Keywords: stack, development, MEAN, big data, web application, extract-transform-load, business intelligence, mongo, express, angular, node, javascript, data warehouse.

INTRODUCCIÓN

Presentación de la investigación

La presente investigación, estudia primeramente los conceptos relacionados a las diferentes tecnologías que existen en el presente y que son consideradas como stacks de desarrollo, desde sus conceptos, hasta llegar a determinar el porqué de su importancia en la construcción de aplicaciones, partiendo de esto, se presenta una primera idea del conjunto de herramientas llamado stack MEAN, sobre el cual se definen sus características, ventajas y desventajas, el porqué de su creación y la forma en la cual funcionan cada una de sus partes, teniendo muy en cuenta que está desarrollado sobre un único lenguaje de programación

Luego, se encuentra también el estudio de una de las principales tecnologías en pleno apogeo en la actualidad como es el Big Data, se parte de sus definiciones, y como ha ido evolucionando con el pasar del tiempo, y sobre todo la importancia que ha llegado a tener para el tratamiento de grandes cantidades de información, y más aún los diferentes campos sobre los cuales se puede aplicar.

Finalmente, en este documento se trata el desarrollo de una aplicación construida sobre el stack MEAN, capaz de soportar el proceso completo que involucra el uso de la tecnología Big Data, desde la recolección de información, hasta su tratamiento y análisis, para concluir con la generación de conocimiento y utilidad en base a los datos obtenidos.

Planteamiento del problema

El Big Data como tal, es una tecnología relativamente nueva, brinda facilidades y diferentes aplicaciones según el campo en el que se lo aplique, sin embargo, desde su inicio se han planteado interrogantes en cuanto a sus límites, y más aún sobre la forma en la cual debería ser manejada.

Como su nombre lo indica, el hablar de Big Data involucra también hablar de grandes cantidades de datos, existen varias razones del origen de esta tecnología, entre las que resalta:

> El crecimiento en el volumen de datos generados por diferentes sistemas y actividades cotidianas en la sociedad ha forjado la necesidad de modificar, optimizar y generar métodos y modelos de almacenamiento y tratamiento de datos que suplan las falencias que presentan las bases de datos y los sistemas de gestión de datos tradicionales. (Hernández, Duque, & Moreno, 2017).

A pesar de no haber una medida exacta para representar cuando un conjunto de datos es considerado como Big Data, se debe tener en cuenta que este concepto que hace referencia al tamaño de los datos ha evolucionado con el paso del tiempo, y lo más seguro es que lo siga haciendo por el simple hecho que la tecnología avanza de forma exponencial, lo que antes era considerado como Big Data debido a las limitaciones del hardware en años pasados, ahora podría ser visto como un conjunto de datos con un tamaño sin importancia.

Por otro lado, según Hernández Leal, Duque Méndez & Moreno Cadavid (2017), Big Data no va dirigido únicamente a la definición de tamaño, sino que abarca tanto volumen como variedad de datos, sin mencionar velocidad de acceso y procesamiento, lo que esclarece que la información no solo es extensa en cantidad, si no también variada y diversa, con distintas especificaciones y diferentes criterios dependiendo del fin último para lo cual se recolectan los datos, cabe resaltar que si lo que se requiere es analizar información a manera de conseguir un beneficio a partir de esta, se debe tener en cuenta que se necesita una gran capacidad de procesamiento de datos. Además de estas consideraciones los datos almacenados por lo general son muy cambiantes, lo que indica una cierta velocidad con la que los datos son añadidos, creados o eliminados. Todas las características anteriores hacen que las tecnologías convencionales resulten obsoletas para el tratamiento de grandes cantidades de información.

A partir de la lógica anterior es posible notar que el tamaño de datos ideal que podría llegar a ser considerado Big Data va a ir variando según las capacidades tecnológicas, y más específicamente la capacidad del hardware dedicado al almacenamiento. Según expresan Camargo, Camargo, & Joyanes (2015) hoy las compañías no saben qué hacer con el gran volumen de datos e información almacenada en diferentes medios o bases de datos, los cuales pueden ser de gran importancia, principalmente en la toma de decisiones.

De esto surge una interrogante acerca de la forma adecuada de tratar aquello que en cierto momento podría llegar a ser considerado como Big Data.

Al referirse al uso de la tecnología Big Data, el problema radica simple y únicamente en el cómo, una de las principales ideas que saltan de manera general es el uso de tecnología web como principal motor para la recopilación y almacenamiento de la información, pero en el contexto definido anteriormente, es necesario tener muy en cuenta que la tecnología Big Data involucra también el tratamiento de dicha información recolectada. Dicho esto, son varios los frameworks, códigos y formas de crear aplicaciones web que existen, y que están dedicadas al análisis de grandes cantidades de datos.

Ante esta situación, una de las primeras complicaciones que aparecen acerca del tratamiento de datos es la forma en la que estos van a ser almacenados para luego ser consumidos. Una base de datos SQL tradicional sería totalmente obsoleta debido a la densidad y velocidad con la cual los datos son creados, ya que la forma estandarizada en la que el lenguaje SQL trata las operaciones que se quisieran realizar sobre un conjunto grande de datos conllevaría tiempos de espera relativamente más grandes, dependiendo de la densidad de los datos, esto si se lo compara con bases de datos NoSQL.

El término NoSQL, del acrónimo Not Only SQL, o no solo SQL en español, aparece como un cambio de paradigma frente al lenguaje de colas estructurado conocido normalmente como SQL, aunque parezca redundante, la principal diferencia entre estos dos radica en que por propia definición los sistemas NoSQL no utilizan SQL en sus consultas, sin contar que ambos difieren mucho en la manera en la que los datos son organizados, esto supone en determinados casos una mejora significativa en los tiempos de respuesta y rendimiento en las consultas realizadas sobre bases de datos de este tipo.

El stack de desarrollo MEAN se muestra como una posible solución capaz de manejar información suficientemente grande y variada para ser considerada Big Data. MEAN es el acrónimo para definir cuatro tecnologías unificadas con un solo propósito, incluye MongoDB, Express, Angular.JS y Node.JS. Según (Haviv, 2014), creador de MEAN, en su blog dedicado a dicho stack, la idea de este conjunto de componentes orientados a un mismo tipo de lenguaje nació al notar la sinergia que existía entre los mismos debido a que todos contaban con Javascript como base, a pesar que en primer lugar el término MEAN ya había sido acuñado por (Karpov, 2013) en un pequeño post realizado en la

página oficial de MongoDB, fueron Haviv junto con la compañía Israelí Linnovate quienes le dieron vida al stack como tal, y la comunidad dedicada al desarrollo open source le dio la fama que posee ahora.

Por lo antes expuesto, es necesario analizar el comportamiento del stack de desarrollo MEAN al ser usado junto con la tecnología Big Data, desde la recolección y almacenamiento de datos, hasta el consumo, tratamiento, y posterior análisis y uso de esos datos en situaciones relacionadas a la toma de decisiones, estadísticas o cualquier tipo de aplicación relacionadas a los datos recolectados.

Justificación

En la actualidad el crecimiento en volumen de los datos referente a cualquier tipo de información ha aumentado de forma exponencial, lejos de ser esto algo negativo, se han ido desarrollando cada vez más y mejores ideas y maneras de darle una utilidad a esta gran cantidad de información, es aquí donde inicia el Big Data como respuesta a la interrogante planteada de qué hacer con estos datos, y más allá de esto, el cómo analizarlos, cabe resaltar que a medida que ha ido evolucionando este concepto, han ido apareciendo aplicaciones orientadas a esta tecnología y sus posibles usos.

La presente investigación se justifica debido primeramente al gran auge que se está produciendo en los últimos tiempos en tecnologías relacionadas a la información, por su parte el Big Data está siendo explotado en casi todas las áreas posibles de la sociedad, del otro lado los stacks de desarrollo son cada vez más usados en proyectos grandes y el stack MEAN no es la excepción, ya que presenta un cambio del paradigma que normalmente se le da al desarrollo web tratando de mejorar ciertos criterios.

Luego se tiene que esta investigación se realiza también con la finalidad de demostrar si el mencionado stack de desarrollo MEAN, es capaz de soportar aplicaciones web que involucran grandes volúmenes de información con el propósito de ser analizados, a partir de esto, se pretenderá ser de utilidad en el desarrollo de maneras más eficientes de análisis de datos, ya que a pesar de que la tecnología Big Data ha tenido gran acogida en los últimos años, son pocas las herramientas libres dedicadas específicamente al tratamiento de información en aplicaciones web.

Otra de las razones de la utilidad de esta investigación, surge del beneficio que proveerá al desarrollo de futuras aplicaciones web orientadas al Big Data, ya que facilitará, por ejemplo, el determinar si el Stack de desarrollo MEAN es adecuado o no para los objetivos de una aplicación web que se desee desarrollar, debido a que se procurará dar pautas de los beneficios y ventajas, así como de las posibles falencias o déficits de dicho stack de desarrollo.

Finalmente, este trabajo será de beneficio al propio autor ya que permitirá aplicar los conocimientos adquiridos en el desarrollo y programación de aplicaciones, así como las habilidades de investigación y capacidad de desarrollo de soluciones a una problemática que involucra el tratamiento de grandes cantidades de información.

Objetivos

Objetivo General

Desarrollar una aplicación web orientada al uso de la tecnología Big Data para el análisis de grandes cantidades de información mediante la aplicación del stack de desarrollo MEAN.

Objetivos Específicos

- Identificar los diferentes tipos de técnicas, software y algoritmos utilizados para el análisis Big Data.
- Aplicar el paradigma Business Intelligence para el análisis de la información obtenida a partir de la aplicación web.
- Determinar el comportamiento de la aplicación desarrollada mediante el stack de desarrollo MEAN frente a grandes volúmenes de información.
- Determinar las ventajas y desventajas del uso del stack de desarrollo MEAN en aplicaciones web orientadas al Big Data.

CAPÍTULO I

MARCO DE REFERENCIA

1.1. ANTECEDENTES

Tomando como punto de partida que el problema sugiere una leve comparación entre el mencionado stack de desarrollo MEAN y distintas maneras de generar y analizar información que en algún momento comprendería Big Data, es posible identificar varios estudios anteriores en relación con la forma y herramientas para el desarrollo de este tipo de aplicaciones.

Poulter, Johnston, & Cox (2015) en su artículo científico analizan el uso del stack de desarrollo MEAN para implementarlo sobre una aplicación RESTful relacionada a la tecnología IOT, o internet de las cosas, los autores señalan que:

> "Los desarrolladores de aplicaciones web han estado utilizando el stack de herramientas de código abierto LAMP (que consiste en el sistema operativo Linux, el servidor web Apache, MySQL como base de datos y PHP como lenguaje de scripting) durante algún tiempo". (p. 1)

Por lo que podemos entender que el stack LAMP es considerado como un estándar en cuanto al desarrollo de aplicaciones web convencionales, sin embargo, los autores continúan señalando y coinciden con Jameel, Khanna, Rangappa, Patil, & Khalkar (2017) al expresar que "En los últimos años ha emergido un nuevo stack de herramientas conocido como MEAN que provee de cuatro herramientas que juntas proporcionan componentes tanto del lado del cliente como del servidor para aplicaciones web interactivas". En consonancia con esto, el artículo presentado por Jameel et al. (2017) compara MEAN con el conocido API de Java orientado a la construcción de aplicaciones RESTful de nombre JAX-RS, expresan que MEAN presenta las características suficiente para desenvolverse mejor en el desarrollo de aplicaciones web, haciendo énfasis en la escalabilidad horizontal que provee MEAN.

Por otro lado Stajcer, Stajcer, & Orescanin (2016) en su investigación aplican el stack MEAN en el desarrollo de una interfaz gráfica para una aplicación Big Data relacionada

a la identificación de diferentes tipos de fraude, los autores expresan que el stack MEAN provee una solución capaz de soportar el desarrollo de una interfaz gráfica de usuario con grandes cantidades de tablas, gráficos o cuadros según la cantidad de información, cabe señalar aquí que esta es una característica importante que sobresale, ya que para la presentación de resultados relacionados a información Big Data el presentar observaciones y comparaciones estadísticas llega a ser una herramienta muy útil para un posterior análisis de datos.

Los autores Ricci et al. (2014) presentan una investigación relacionada al uso de tecnologías web para el análisis de datos provenientes de los proyectos TAOS y TAOS 2 correspondientes a un sistema de telescopios administrados por la UNAM en México, durante su artículo comparan distintas maneras de presentar la información en el client-side, así como distintas maneras de procesarla en el server-side, argumentando como solución moderna el uso de Javascript en ambos casos y la posibilidad de la implementación de una base de datos no relacional como es MongoDB para la persistencia de los datos, expresando que posee un buen rendimiento en cuanto a datos estructurados en forma de árbol debido a su característico uso del formato JSON.

Duy Khue, Thanh Binh, Chang, Kim, & Chung (2017) en su artículo científico proponen el uso del Stack de desarrollo MEAN para el diseño e implementación de una aplicación de control sobre un sistema de señalización digital, como bien indican:

> "Los tipos convencionales de frontend están desarrollados usando HTML, CSS y Javascript, mientras que los manejadores de sistemas convencionales han sido desarrollados sobre LAMP y sus variantes. Actualmente el Stack MEAN ha ganado popularidad como principal aplicación de desarrollo web por diversas características que van desde mayor rapidez, hasta arquitecturas client-side y server-side más flexibles". (p. 1)

En concordancia con esto, los autores continúan describiendo los beneficios de MEAN presentándolo como una herramienta poderosa para el desarrollo de aplicaciones que conlleven un cierto nivel de análisis y manejo de datos.

A pesar de lo enfatizado en párrafos anteriores, existen autores como González, González, & López (2014) que en sus investigaciones analizan también posibles puntos débiles de

MEAN como stack de desarrollo web: “El uso de MEAN ha presentado debilidades esencialmente en el sistema gestor de bases de datos. Una alternativa al empleo de MongoDB pudiera ser el sustituirlo por SQLite, que ha demostrado estabilidad y eficiencia que validan su buen desempeño”. (p. 2), a partir de esto es evidente pensar en la necesidad de llevar acabo un buen análisis sobre el rendimiento de las diferentes partes de este stack de desarrollo.

1.2. BASES TEÓRICAS

1.2.1. STACKS DE DESARROLLO

Concepto

Un stack de desarrollo es comúnmente definido como un conjunto de herramientas dedicadas a solucionar los problemas y facilitar el desarrollo de aplicaciones de cualquier tipo y en cualquier ámbito, el término stack que es traducido al español como pila, es una analogía a la forma de apilar varias herramientas que en su conjunto ayudan a simplificar la tarea del desarrollo de aplicaciones en general. De la mano de esto, si nos referimos específicamente al desarrollo web, según la empresa española (1 & 1 Internet España S.L.U., 2016), "Los componentes de software de un web stack pueden combinarse de forma individual, por lo que existe un gran número de versiones, cada una para diferentes fines.", es decir, que un stack de desarrollo, no necesariamente nace como tal, sino que puede ser creado a partir de diferentes aplicaciones individuales que cumplan los requisitos necesarios para llevar a cabo la creación y mantenimiento de una aplicación según los requerimientos de la misma, a pesar de esto existen 4 componentes básicos y necesarios para comenzar el desarrollo de una aplicación, sobre todo el ámbito del desarrollo web.

1. Sistema operativo: Corresponde al conjunto de programas que conforman el sistema del servidor. Ejemplos de sistema operativo son Windows, Linux, Mac.
2. Servidor web: Es un software instalado en el sistema operativo encargado de responder las peticiones hechas por parte de un cliente. Ejemplos de servidores web son Apache, Nginx, IIS, Lighttpd.
3. Base de datos: Es un sistema alojado en el lado del servidor encargado de guardar información de manera permanente para que el servidor web puede acceder a esta de ser necesario. Ejemplos de bases de datos son MySQL, Oracle, MongoDB
4. Intérprete de órdenes: Es un lenguaje de programación que se encarga de interpretar tanto del lado del cliente como del servidor las páginas dinámicas que el servidor web no es capaz de procesar por sí solo. Ejemplos de intérpretes son PHP, Ruby, Perl, Python

Stacks de desarrollo web más comunes

LAMP y sus variaciones

El stack LAMP es en la actualidad el más común y conocido para el desarrollo de aplicaciones web, y a pesar de no haber sido desarrollado en un principio para trabajar en conjunto, sus componentes son una muy buena combinación para casi cualquier tipo de desarrollo.

STACK	S.O.	Servidor web	Base de datos	Intérprete
LAMP	Linux	Apache	MySQL	PHP
XAMPP	Soporta varios	Apache	MySQL	PHP, Perl
MAMP	Mac O.S.	Apache	MySQL	PHP
WAMP	Windows	Apache	MySQL	PHP
LAPP	Linux	Apache	PostgreSQL	PHP

Tabla 1. *Comparativa de stack LAMP y derivados.* ***Fuente: Autor.***

La ***Tabla 1*** muestra una comparación de los diferentes stacks de desarrollo derivados del modelo propuesto en el stack LAMP, cada uno con sus respectivos componentes.

WISA

Corresponde a la solución web brindada por Microsoft, sigue los mismos paradigmas de un stack de desarrollo normal, pero a diferencia del anterior este si fue pensado desde su creación para que sus componentes trabajen en conjunto. Las aplicaciones propuestas para el sistema operativo Windows son: Internet Information Services (IIS) como servidor web, SQL Server como base de datos y ASP.NET como intérprete.

J2EE

Es una plataforma o arquitectura con varias herramientas sobre las cuales se pueden desarrollar aplicaciones comunes de usuarios, aplicaciones empresariales o incluso aplicaciones orientadas a la web. En este último caso el stack de J2EE implementa páginas JSP y Servlets en el client-side y Enterprise JavaBeans en el server-side, además cuenta con controladores conocidos como JDBC para la conexión con diferentes tipos de bases de datos.

1.2.2. STACK DE DESARROLLO MEAN

Arquitectura

El stack de desarrollo MEAN fue sugerido por Amos Haviv, un desarrollador de software que, al trabajar de forma constante con las herramientas que lo conforman, notó la sinergia que existe entre las mismas. MEAN consiste en un conjunto de aplicaciones que proporcionan una solución tanto del lado del cliente como del lado del servidor incluyendo la base de datos, está conformado por MongoDB, Express, Angular.JS y Node.JS, una de las características que hace especial a MEAN es que el lenguaje de programación de todas sus aplicaciones es JavaScript, siendo este uno de sus puntos fuertes, ya que al tener como base un solo lenguaje, facilita la comunicación entre las herramientas del stack, y sobre todo con la base de datos.

Por otro lado, German, Salmeron, Ha, & Henderson (2016) definen MEAN como un stack de desarrollo de tipo framework, dogmático, y que simplifica y acelera el desarrollo de aplicaciones web, haciendo notar que el hecho de usar un único lenguaje de programación beneficia la manera en la cual se desarrollan las aplicaciones, tal y como resalta Karanjit (2016) al expresar que el uso de JavaScript tanto en el lado del cliente como del servidor hace de MEAN un stack de desarrollo más poderoso, facilitando la programación y reduciendo el tiempo dedicado al desarrollo de la aplicación.

MongoDB

MongoDB es una base de datos de tipo NoSQL y de código abierto, escrita en lenguaje C++ y con una interfaz que utiliza JavaScript para guardar datos y estructuras, por lo que se podría decir que está orientada a documentos.

De acuerdo con Poulter, Johnston, & Cox (2015) hay una creciente tendencia al uso de las bases de datos NoSQL debido a su flexibilidad y esquema dinámico, expresan que MongoDB es una base de datos de alto rendimiento construida sobre un formato JSON, a partir de esto es lógico deducir el por qué forma parte del stack de desarrollo MEAN, ya que al trabajar con lenguaje JavaScript facilitaría en gran medida cualquier tipo de operación y consulta puesto que estas pueden ser enviadas directamente a la base de datos para ser procesadas.

Express

Express es un framework para NodeJS, siendo aplicable del lado del servidor facilitando las operaciones relacionadas con peticiones HTTP. Como bien afirma Karanjit (2016) la importancia de este framework dentro del stack de desarrollo MEAN radica en la función que presta como un medio de transferencia de peticiones desde el cliente hacia la base de datos y de igual manera, las respuestas de la base de datos hacia el cliente.

AngularJS

AngularJS es un framework para JavaScript creado por Google y de código abierto, dedicado al FrontEnd, es decir el lado del cliente, y que se caracteriza por facilitar la creación de aplicaciones web sencillas siguiendo el paradigma Modelo Vista Controlador en páginas sobre todo de tipo SinglePage. "AngularJS contiene características asombrosas como data binding de doble vía y representa una solución completa para el desarrollo FrontEnd, siendo uno de los frameworks JavaScripts más usados para el desarrollo web." (Kadam, Chaudhari, Patil, & Chavhan, 2017, p. 16).

NodeJS

Tal y como sugiere González, González, & López (2014) "Node.js es un entorno de programación en JavaScript para el Backend basado en el motor V8 de JavaScript del navegador Google Chrome y orientado a eventos." (p. 6), continuando con esta definición, y contrario a la mayoría de frameworks para JavaScript NodeJS fue pensado para el despliegue de servidores web y la ejecución de funciones de manera asíncrona sobre estos, y en tiempo real, consta de un manejador de paquetes llamado npm que funciona en base a módulos que pueden ser añadidos a la ejecución.

Funcionamiento General

La ***Figura 1*** muestra en forma simple el funcionamiento de una página desarrollada mediante el stack de desarrollo MEAN y las funciones que cumplen cada una de sus aplicaciones, es posible identificar 6 pasos desde el momento en que el cliente realiza una petición en la aplicación web.

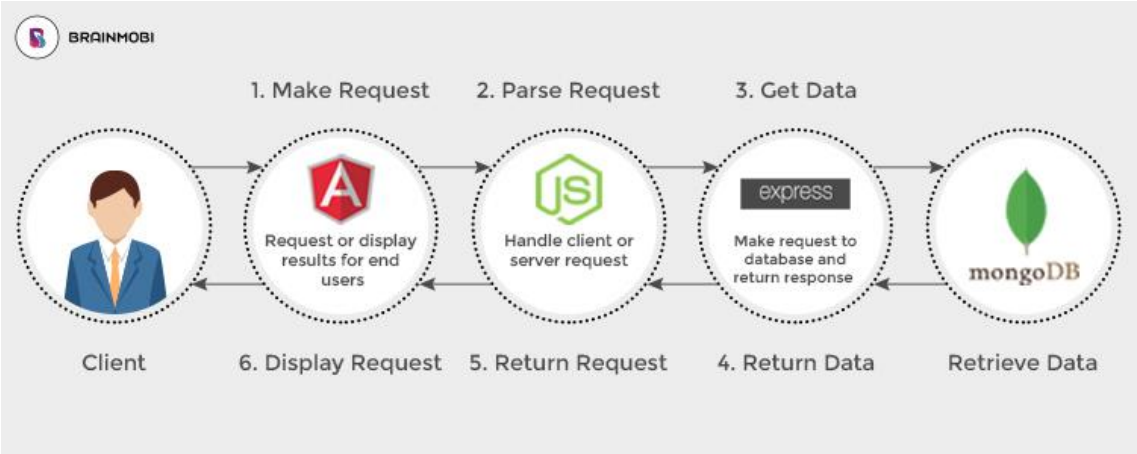


Figura 1. *Funcionamiento General del stack de desarrollo MEAN.* ***Fuente (Tripathi, 2017)***

1. El usuario realiza una petición desde la página web, es decir el FrontEnd diseñado mediante AngularJS
2. NodeJS se encarga de analizar gramaticalmente y manejar la petición del cliente.
3. Express se encarga de realizar la petición hacia la base de datos MongoDB.
4. De igual manera al obtener los datos Express se encarga de devolver la respuesta.
5. Luego es NodeJS otra vez el encargado de devolver la respuesta de la petición del lado del servidor.
6. Para que finalmente sea mostrada en el FrontEnd mediante AngularJS

1.2.3. BIG DATA

Conceptos

El concepto de Big Data como tecnología ha ido evolucionado conforme pasa el tiempo y a la par del aumento exponencial de la cantidad y volumen de información que se maneja en cualquier ámbito de nuestra sociedad, siendo justo ahora el internet, la mayor fuente de información acerca de todo, misma que sigue en crecimiento por lo que era solo cuestión de tiempo que se encontrara un método más flexible para el manejo de toda esta información, por otro lado, está planteada también la problemática que existe sobre empresas en auge, o de gran tamaño, debido a una creciente dificultad para el manejo adecuado de la información con la que cuenta, acerca de esto Camargo, Camargo, & Joyanes (2015) explican que “Hoy las compañías no saben qué hacer con el gran volumen

de datos e información almacenada en diferentes medios o bases de datos, los cuales pueden ser de gran importancia, principalmente en la toma de decisiones." (p. 4), es decir que todo tipo de empresa, instituciones y demás organismos en el mundo actual tarde o temprano se encuentran con el dilema de elegir la mejor manera en la que sus datos son almacenados ya que han quedado prácticamente obsoletos los viejos papeles, carpetas y cualquier forma que ahora podría llegar a ser considerada rudimentaria para llevar registros de cualquier tipo.

De forma general y muy simple Cotino (2017) da una de las primeras aproximaciones de lo que en un determinado momento fue el principal concepto de lo que representa el Big Data expresando que la parte que determina el "Big", dentro de la definición, se refiere al volumen masivo de los datos, de forma que es casi imposible llegar a medirlo a tal punto que lo compara con la infinidad del espacio, analizando esta pequeña definición es posible caracterizar al Big Data como una enorme colección de datos cuyo magnitud podría llegar a ser incomprensible para la capacidad de la mente humana.

A grandes rasgos Martínez (2015) afirma que "Big Data: se trata de aquel conjunto de datos que, por su tamaño ingente, sobrepasa la capacidad de ser gestionado por bases de datos de integración tradicionales." (p. 69), sea cual sea la definición o el autor que exprese un breve concepto sobre Big Data, y tal cual su nombre lo indica, este término se refiere principalmente a una enorme cantidad de datos e información, otro punto importante que se resalta en esta definición es que, por su condición compleja en cantidad, el análisis que conlleva la acumulación de tantos datos es imposible ser gestionado mediante bases de datos convencionales, por lo que son requeridos otros métodos como por ejemplo bases de datos NoSQL.

En este punto ya es sencillo el razonar y encontrar en casi cualquier definición de Big Data que involucre medida de datos, palabras que hacen referencia a información de tamaño grande, enorme, colosal, etc. A pesar de esto existen varias otras definiciones que no hacen referencia únicamente al volumen de los datos, sino también a su variedad o incluso al fin último de esa información, que podría llegar a ser el análisis de esta.

Debido a su naturaleza de extenso volumen y propiedades variables en cuanto al tipo de información, existen muchos otros conceptos para la tecnología Big Data, cada uno orientado a diferentes ramas según la manera en la que se desee aplicar o analizar los

datos almacenados. Por ejemplo, otra de las definiciones más comunes y generales para Big Data dice lo siguiente:

> El término hace referencia a sistemas que manipulan enormes cantidades de datos, sobre los que ejecutan diferentes tipos de análisis con técnicas propias de business analytics, data mining o text mining para buscar patrones. Podemos hablar de volúmenes de terabytes, petabytes (1.000.000.000.000.000 bytes), o más. (Serrano Cobos, 2013, p. 161)

Es posible notar de inmediato que este concepto no habla del Big Data describiendo la información como tal, sino que más bien utiliza el termino para referirse a sistemas dedicados al uso y manejo de tales cantidades de datos. En esta definición también es posible advertir que el volumen de información que puede llegar a ser valorada como Big Data es variable, pero siempre, considerablemente grande, lo que casi en cualquier caso implica un difícil tratamiento y análisis de estos.

Así mismo existen quienes le dan a esta tecnología un enfoque totalmente empresarial, autores como Chen, Chinag, & Storey (2012) asumen que el término Big Data es usado de forma correcta para describir no solo conjuntos de datos, si no también aplicaciones, métodos y técnicas de análisis de información compleja, dando a entender que Big Data involucra también la tecnología usada en el manejo y visualización de las mismas enormes cantidades de datos descritas anteriormente, pero esta vez con un fin específico orientado a un negocio, empresa o institución que necesite un manejo y análisis preciso de los datos con los que cuenta.

Dimensiones

Si se habla del tamaño de los datos relacionados al Big Data, resulta muy complicado determinar cantidades mínimamente exactas debido a la manera exponencial en la que ha ido creciendo no solo esta tecnología, sino también los datos e información a los que se puede tener acceso de manera sencilla. Y aparte de esto, las dimensiones son completamente variables y adaptables a las necesidades de quien use la información para un determinado propósito, por lo que muchas veces la dimensión como tal queda en segundo plano frente a las otras características relacionadas al Big Data.

Para poder definir de mejor manera las dimensiones, es necesario primero considerar la naturaleza variable del tipo de datos que se pueden llegar a tomar en cuenta como Big

Data, eso dependiendo del concepto y contexto en el cual se requiera aplicar esta tecnología. Esto se ve reflejado en la forma en la que empresas de distintos tamaños y visiones manejan sus datos, ya que es totalmente diferente la cantidad y tipos de información que manejan grandes compañías de redes sociales, por ejemplo, que la información que necesita una empresa dedicada a la contabilidad y en la cual también es posible aplicar esta tecnología. Dicho esto, es obvia la diferencia de magnitudes, por lo que llegar a medir datos que no están para nada relacionados llega a ser una tarea difícil de cumplir.

Luego, es también necesario tomar en cuenta cómo se ha ido desarrollando esta tecnología, y como lo sigue haciendo conforme pasan los años, sin mencionar su concepto en la actualidad, es decir, hablar de grandes volúmenes de datos hace 5 años, por poner una cifra, no es lo mismo que en la actualidad, existen muchos factores a considerar para llegar a una aproximación real del tamaño de datos.



Figura 2. *Estadísticas mundiales del uso del internet y redes sociales. Agosto 2015.* ***Fuente. (Kemp, We Are Social Singapore, 2017)***

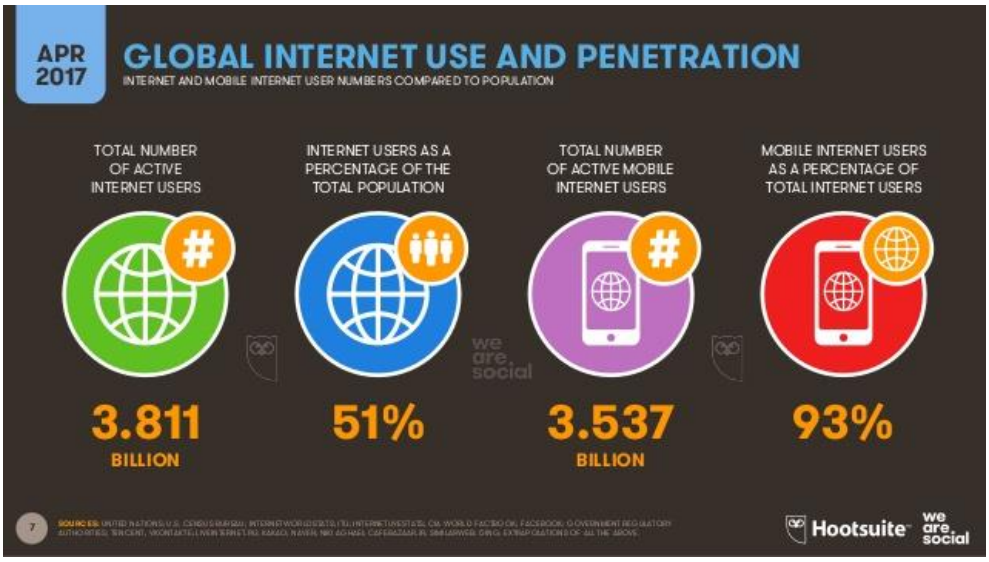


Figura 3. *Estadísticas mundiales del uso del internet y redes sociales. Abril 2017.* ***Fuente (Kemp, We Are Social Singapore, 2017)***

La ***Figura 2*** y ***Figura 3*** muestran una comparación estadística del porcentaje de usuarios activos de internet en el mundo en el año 2015 y 2017 respectivamente, pasando desde 3175 a 3811 billones de usuarios en la actualidad por parte del 51% de la población mundial total, en relación a esto John (2014), expresa que estamos en una era en la que cifras como las anteriores dejan de asombrar, poniendo ejemplos de casos como los de Google, que procesa 24 petabits de datos diariamente, o los 400 millones de tweets enviados por día. Si se considera la enorme cantidad de información que cada usuario provee en cualquier tipo de situación, se puede notar la creciente cantidad de datos que existen en el internet, siendo esta la principal fuente actual de información, dejando volúmenes de información difíciles de medir.



Figura 4. *Volumen de datos estimados a nivel mundial y aproximaciones futuras.* ***Fuente. (Guemes Seoane, 2016)***

La ***Figura 4*** muestra la evolución y aumento del volumen estimado de datos digitales a nivel mundial, y da una idea de la cantidad de información que va a existir en los años futuros llegando a los zettabits, siendo obvio que esta es una cifra que no se mantendrá estática, sino que por el contrario continuará aumentando.

A partir de las ideas planteadas anteriormente, es evidente deducir que las dimensiones del Big Data van a depender totalmente del ámbito en el que se lo aplique, pero siempre teniendo en cuenta que es un concepto que no va a disminuir, sino que está en constante aumento conforme avance la tecnología en general, conforme avanza la sociedad y conforme crece la cantidad de información a la que se podría llegar a tener acceso.

Características

Para Hernández, Duque, & Moreno (2017), "Big Data no va dirigido solo a gran tamaño, sino que abarca tanto volumen como variedad de datos y velocidad de acceso y procesamiento." (p. 2), en base a esto y de forma general es posible definir 3 características principales y comunes para que el Big Data quede definido completamente como tal.

Volumen: Se refiere básicamente al tamaño o peso de los datos objeto de análisis, McAfee & Brynjolfsson (2012) explican que actualmente al hablar del volumen como característica del Big Data se habla de muchos petabytes de información, siendo esto una aproximación que está en constante aumento.

Variedad: Según (Camargo et al., 2015), explican que "Se puede mencionar que va muy de la mano con el volumen, pues de acuerdo con éste y con el desarrollo de la tecnología, existen muchas formas de representar los datos." (p. 66), dando a entender que esta característica se refiere a la forma en la que los datos son adquiridos de diferentes fuentes que pueden variar dependiendo del contexto de los datos.

Velocidad: Esta característica se refiere a la velocidad con la cual los datos deben ser analizados de forma correcta para poder producir resultados en un tiempo determinado y obtener beneficios de estos.

> Además del gran volumen de información, esta existe en una gran variedad de datos que pueden ser representados de diversas maneras en todo el mundo, por ejemplo, de dispositivos móviles, audio, video, sistemas GPS, etc., de tal forma que las aplicaciones que analizan estos datos requieren que la velocidad de respuesta sea lo demasiado rápida para lograr obtener la información correcta en el momento preciso. (Barranco, 2012)

En base a lo anterior, es evidente que las 3 características principales sobre las cuales trabaja el Big Data están en completa sinergia para poder cumplir el fin último de análisis de los datos escogidos y darles un propósito, no obstante, otros autores definen también varias características más para que la definición de Big Data quede completa. Por ejemplo según el Massachusetts Institute of Technology, citado por Martínez Sesmero (2015) podría ser correcto añadir una cuarta "V", refiriéndose al valor aportado por estos grandes conjuntos de datos, viéndose esto reflejado en los resultados finales.

Problemática y Finalidad

La principal problemática del Big Data viene a ser su fin último, el qué hacer con los datos obtenidos y cómo hacerlo, es decir el análisis de toda la información recabada, este dilema trae consigo otros inconvenientes que van desde el recurso humano hasta el recurso de hardware y software necesario para llevar a cabo todas las operaciones relacionadas si se tratara de analizar, por ejemplo, petabytes de información. Es lo que expresa Serrano (2014) al sugerir que los principales problemas con esta tecnología tienen que ver en el cómo obtener los datos; donde almacenarlos; cual sería el costo relacionado a esto; la confiabilidad y veracidad de los mismos, puesto que para un correcto análisis

los datos tienen que ser reales y de actualidad; y luego cómo obtener el dato o conjunto de datos precisos que se necesita a partir de esta gran cantidad de información y de ser posible en tiempo real, sin olvidar que lo primordial es cómo generar el conocimiento deseado a partir del conocimiento disperso obtenido y más allá de esto como representarlo de manera entendible para hacer uso de este, dicho de otra manera, cómo segmentar y analizar datos, para llegar a una conclusión deseada, a partir de datos que muchas veces no se encuentran estructurados.

Aplicaciones

En la actualidad la cantidad de datos que se maneja y sobre todo que se encuentra en internet es suficiente para decir que las posibilidades son prácticamente infinitas, Big Data es un concepto que pude ser aplicado casi que, a cualquier ámbito en la sociedad, cualquier campo o especialidad que involucre un mínimo tratamiento de datos es susceptible a adoptar esta tecnología para mejorar su funcionamiento.

Citando a Cotino (2017), se refiere de forma acertada a este fenómeno como un territorio totalmente abierto a la imaginación, para finalizar diciendo que el Big Data se trata de una estadística del todo, es decir que la aplicación que se le dé a esta tecnología depende totalmente del contexto de la información y de quien la almacene o use.

Al día de hoy existen aplicaciones médicas, contables, sociales, de inteligencia artificial y cualquier campo tecnológico, pero donde particularmente se han hecho muchos avances en cuanto al Big Data es en el ámbito empresarial y de negocios, llegando a crear conceptos como el Business Intelligence para predecir comportamientos en las personas relacionados a productos y servicios comercializados, y todo esto sin hablar de los grandes en las redes sociales como Facebook, Google, Twitter, etc., que manejan y analizan cantidades de información inimaginables para propio beneficio.

Por otro lado, considerando el fin propio del Big Data, esta tecnología tiene aplicaciones muy específicas adaptables a cualquier ámbito como por ejemplo el análisis predictivo de datos, la cantidad de información que se puede llegar a obtener es suficiente como para identificar similitudes y llegar a conjeturar hacia donde van dirigidos determinados datos, lo cual culminaría en la predicción en base a la información actual. Algo similar ocurre con la detección de patrones, que al ser analizados es posible llegar a identificar tendencias en una cierta cantidad de datos que puede ser obtenida por ejemplo de una

población de personas, y de esta manera conocer la inclinación en el pensamiento de dicha población. Siendo estas dos aplicaciones, los principales ejemplos del uso general que se le puede dar a esta tecnología, es posible derivar de estos, usos tales como la optimización del rendimiento de empleados y procesos en un negocio, o la segmentación de clientes, apuntando siempre a la mejora en cualquier ámbito mediante la obtención de la información suficiente para esto.

Herramientas, softwares y soluciones Big Data

El origen de cualquier tipo de solución actual a la problemática del Big Data, se debe a que en determinado momento los ingenieros dedicados al análisis de este tipo de información se dieron cuenta que necesitaban modernizar sus herramientas puesto que la cantidad masiva de datos que trataban superaba la memoria de los ordenadores que utilizaban, y en ningún momento dejarían de aumentar, esto según explican Mayer-Schönberger & Cukier (2013), dio lugar a las herramientas y softwares tanto privados como de código abierto para el tratamiento de los datos masivos.

MapReduce

El primer concepto de MapReduce nace gracias a Google y su necesidad de querer implementar y mejorar la forma en la cual calculaba la distribución clasificación y ordenamiento del ranking de sus páginas de búsqueda, problemática conocida como PageRank, a partir de esto, esta tecnología fue creciendo como respuesta para el tratamiento de grandes volúmenes de información.

De acuerdo con Nuñez, Díaz de la Paz, & García (2016) "MapReduce es un modelo de programación para el procesamiento y generación de grandes volúmenes de datos en sistemas distribuidos y está especialmente pensado para tratar ficheros de gran tamaño." (p. 105), continuando con esta definición, es lógico pensar que al ser un paradigma dedicado a sistemas distribuidos fue pensado únicamente para ser aplicado en tecnologías específicas como la computación distribuida, o para infraestructuras organizadas en clústeres, pero el concepto de MapReduce puede ser aplicado en cualquier tipo de sistema al cual sea necesario implementar sus dos funciones principales gracias a las cuáles debe su nombre.

La definición de MapReduce como paradigma está basada en las funciones Map y Reduce del lenguaje de programación LISP, según afirman Dean & Ghemawat (2008),

investigadores de Google, este proceso toma como entrada pares de llave-valor dentro de un conjunto de grandes datos y de igual manera produce una salida de pares llave-valor reducido según determinado criterio o función. Por su parte, la función Map se encarga de tomar por separado y de manera paralela cada uno de los pares llave-valor indicados en un conjunto inicial de datos, luego, produce una salida intermedia del mismo tipo, que es tomada como entrada para la función Reduce, la cual se encarga de unir, ordenar y disminuir los valores que se producirán como salida a partir de un criterio dado.

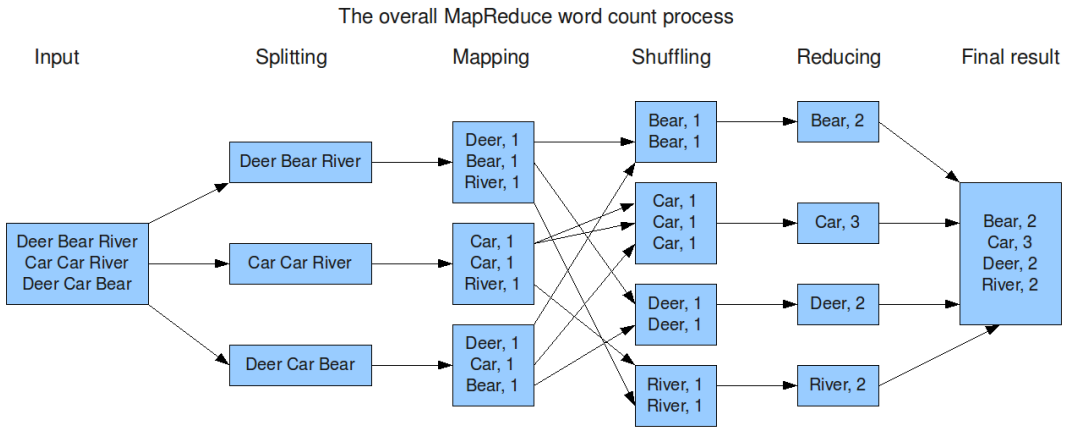


Figura 5. *Ejemplo de un contador de palabras aplicando el paradigma MapReduce.* ***Fuente. (Van Gronningen, 2009)***

Hadoop

Según Dominguez & Yeja (2015) Hadoop es una implementación de código abierto al paradigma y funciones presentes en MapReduce, dando a entender que su objetivo es simplemente facilitar el análisis de grandes cantidades de información sobre todo en ambientes distribuidos y sistemas basados en clústeres.

> Hadoop es una librería de Apache definida como un framework que permite hacer procesamiento de datos distribuido sobre volúmenes de datos de considerable tamaño sobre clúster. Está diseñado pensando en brindar poder de escalamiento desde un par de servidores hasta cientos de máquinas o nodos, las cuales manejan almacenamiento y procesamiento local. (Hernández et al., 2017, p. 14)

Cualquier definición acerca de Hadoop concuerda no solo en que esta herramienta es capaz de soportar volúmenes de información enormes, sino también en que es aplicable tanto a pequeña como a gran escala. Pero Hadoop como tal no es únicamente una herramienta, sino que llega a ser todo un abanico de posibilidades capaces de facilitar en gran medida todo lo referente al análisis Big Data, siendo las principales:

- Hadoop MapReduce
- Hadoop Common
- HDFS (Hadoop Distributed File System)
- Hadoop YARN (Yet Another Resource Negotiator)

BigQuery

Es una de las plataformas orientadas a la nube de Google, el cual lo define como un almacén de datos diseñado para el análisis Big Data, se presenta como una de las soluciones de software privativa creada para el tratamiento de petabytes de información.

Oracle Big Data Appliance

Es la solución Big Data proporcionada por Oracle, es una plataforma y software privativo que la empresa define como un sistema multi propósito diseñado para soportar la carga de trabajo, procesamiento y transmisión de grandes volúmenes de datos.

HDinsight

Es un software implementable y de código abierto diseñado por Microsoft, quien lo define como un servicio administrado de análisis de macrodatos, y que se encuentra optimizado para Hadoop y sus herramientas como Hive, Spark, HBase, etc.

Bases de datos NoSQL

El término NoSQL, acrónimo para Not Only SQL y que traduce No Solo SQL, es usado para definir aquellos tipos de bases de datos que no siguen el modelo relacional original de aquellas bases de datos SQL, desde un principio este modelo fue pensado para el tratamiento de grandes cantidades de datos por lo que asociarlo a la tecnología Big Data resulta algo normal en la actualidad.

A pesar de que, si es posible el análisis de grandes volúmenes de datos mediante el modelo SQL, resulta en una práctica totalmente ineficiente, dejando a las bases de datos NoSQL como la opción por excelencia para esta problemática.

Ejemplos de Bases de Datos NoSQL son:

- DynamoDB
- Cassandra
- CouchDB
- HBase (Parte de herramientas Hadoop)
- MongoDB

Lenguajes de programación para análisis de datos

Así como se han dado a conocer diferentes tipos de bases de datos especializadas en el tratamiento Big Data, existen también lenguajes de programación que por su estructura y forma de manejar los datos, facilitan el proceso de análisis de grandes cantidades de información, desde Data Mining hasta el Data Wharehousing y pasando por la formalización de las estructuras de datos.

Ejemplos de este tipo de lenguajes de programación son:

- Java
- R
- Octave
- Python
- Julia

1.2.4. TECNOLOGÍA BIG DATA Y APLICACIONES WEB

Big Data y sus aplicaciones orientadas a la web

"Las compañías pioneras en el uso de analíticas de grandes bases de datos han sido las que operan sobre internet, como son los motores de búsqueda, los sitios de redes sociales y los sitios de comercio en línea." (Hernández, Duque, & Moreno, 2017, p. 9)

Anteriormente en este documento se mencionaron aplicaciones para la tecnología Big Data dejando en claro que son muchos los campos en los cuales puede ser aprovechada, aún si se tiene en cuenta una perspectiva únicamente web las posibles aplicaciones siguen siendo enormes, aunque todas con un enfoque muy similar.

Una de las primeras ideas, y de las más fundamentales que concuerdan con la definición y uso del Big Data, y que está orientadas netamente a aplicaciones web, es simplemente el análisis de cualquier tipo de información proveniente de cualquier tipo de página web. Dicho esto, si se piensa en todas las ramificaciones que podría tener este único concepto de utilidad, y sin tener que razonar mucho es fácil regresar a las definiciones más básicas del Big Data, es decir, la aplicación de esta tecnología depende en gran medida del tipo de datos que se analicen, y de manera un poco más amplia del porqué y como de la obtención de dichos datos. Coincidiendo con la percepción que tienen Camargo, Camargo, & Joyanes (2015) acerca de los usos del Big Data, es lógico pensar en cuál es el origen de tanta información, duda que responden al expresar que todos esos datos provienen de la Web, es decir datos generados de la navegación en sitios web, rastros de clics, datos geoespaciales, documentos alojados en internet, bases de datos, redes sociales, etc.

Si se tiene en cuenta, por ejemplo, todos los datos que son generados mediante la navegación e historial de visitas a páginas web de todos los usuarios de internet, la cantidad de información posible llega a ser totalmente inconmensurable, y de igual manera todo lo que se podría llegar a conocer mediante esos datos. Un claro ejemplo de esto es la publicidad en internet, que ha llegado a tal punto de poder segmentar a los clientes, y mostrar la publicidad adecuada para cada uno, “La capacidad de recopilar los datos de geolocalización de los usuarios está convirtiéndose en algo extremadamente valioso. A escala individual permite la publicidad personalizada allí donde esté o se prevea vaya a estar una persona.” (Mayer-Schönberger & Cukier, 2013, p. 104), es decir de forma simple, algo ligera, y aunque suene absurdo, el análisis de datos mediante la aplicación de la tecnología Big Data permite convertir la información referente a la posición de una persona en una oportunidad de presentarle una publicidad que está de acuerdo a donde se encuentra o donde se encontrará en un futuro cercano, sin mencionar las preferencias del mismo.

Siguiendo esta misma línea relativa a los datos de navegación web de un usuario, es posible identificar otro ejemplo de uso para el Big Data orientado a la web, la idea de relacionar las tiendas online y el análisis de datos salta a la vista si se considera la capacidad que tienen estos portales para sugerir la siguiente compra, producto o servicio que podría ser de interés adquisitivo para un usuario.

En principio pensaremos que sólo empresas de internet de la envergadura de Amazon o eBay harán uso de estas tecnologías (y lo hacen) pero no tiene por qué ser así. Una pequeña tienda online utiliza hoy en día aplicaciones de analítica web para conocer qué hacen los usuarios que les visitan hasta que compran un producto, cuáles han sido los canales de publicidad y comunicación más eficaces y eficientes, o descubrir fallos de usabilidad o arquitectura de información, que impiden al usuario encontrar y utilizar lo que el responsable del sitio web desearía que encontrara y utilizara. (Serrano, 2014, p. 563)

En concordancia con lo anterior, todas las aplicaciones de tipo e-commerce en tiendas online, y cualquier tipo de sugerencia que estas puedan presentar están basadas en la información que los usuarios dejan al visitar estas páginas web, sin mencionar los productos o servicios similares que se hayan adquirido con anterioridad.

Dejando un poco de lado el sentido comercial en las aplicaciones para el Big Data, es evidente considerar otro uso por demás explotado en internet, se trata de la forma en la cual los buscadores web indexan y ordenan los resultados referentes a una búsqueda específica que un usuario este realizando, si se tienen en cuentan la cantidad de información y peticiones que tienen que resolver dichos buscadores web es lógico pensar que la manera más eficiente de realizar estas operaciones es mediante la aplicación de la tecnología Big Data.

Para que se tenga en cuenta la magnitud y volumen de la información que un buscador web debe procesar, se propone el siguiente ejemplo: Un artículo publicado por Taycher (2010) en la entonces página oficial de Google Books dio a conocer que para ese año la empresa reconocía un total de más de 129 millones de libros publicados en todo el mundo, siguiendo esta consideración Martínez (2016) afirma que de esta cantidad, únicamente son 15 millones de ejemplares son los que se encuentran digitalizados en la base de datos de Google, para culminar expresando que en cada libro es posible encontrar decenas de

miles de millones de palabras en cualquier idioma, las cuales pueden ser exploradas por medio de procedimientos computacionales, esto, haciendo una clara referencia al análisis Big Data, si bien los libros no son la única fuente de la cual un buscador web obtiene resultados, puesto que existen muchos otros orígenes de información que son considerados, como artículos científicos, documentos onlines, páginas web, etc., esta pequeña percepción del trabajo de búsqueda es suficiente para entender en parte la proporción e importancia del uso de esta aplicación para el Big Data. Como dato adicional, según la página oficial de la UNESCO, la cantidad de libros en el mundo crece a un ritmo de aproximadamente 2.2 millones, cifra que también va en aumento, por cada año, por lo que para finales del 2017 y teniendo en consideración lo anterior, se estima un total de más de 144 millones de libros publicadoes oficialmente en el mundo.

Redes sociales y el Big Data

Otra de las aplicaciones para la tecnología Big Data que salta naturalmente a la vista y que es casi obvia, es su uso en las redes sociales. Es lógico pensar que la cantidad de información que manejan este tipo de aplicaciones web se preste para ser considerada como Big Data, pero el uso de esta tecnología en las redes sociales va más allá de únicamente la cantidad y volumen de datos que se manejan normalmente ya que lo que verdaderamente importa en este ámbito es el uso que se le da a esta enorme cantidad de información que proviene de una enorme cantidad de usuarios activos.

Uno de los casos más básicos y comunes para este tipo de aplicación a la tecnología Big Data es el que expresan de forma simple Hernández, Duque, & Moreno (2017) al comentar que “Por ejemplo, en el caso de las redes sociales, se trataría de revisar las personas que posiblemente le interesarían seguir a otras dependiendo de sus amistades o seguidores.” (p. 19), únicamente con esta definición es casi intuitivo razonar la forma en la cual redes sociales como Facebook, por poner un ejemplo, se encargan de analizar los datos que tiene acerca de cada usuario, y en base a similitudes en gustos, intereses, amistades e incluso su localización es capaz de generar una sugerencia casi predictiva en los gustos de un usuario.

Continuando con la idea anterior este método para generar sugerencias no es la única forma en la que son explotados los datos masivos, si se piensa en las redes sociales como una especie de comunidad, conformada por una ingente cantidad de usuarios, resulta

evidente relacionarlas como una población sobre la cual es posible, con los suficientes datos, realizar cualquier tipo de operación estadística, y es en esta idea donde se origina el aplicar las grandes cantidades de información en redes sociales para descubrir, definir y predecir tendencias de una determinada población.

Business Intelligence

Una primera definición de este término y un poco más orientada a los negocios es la descrita por (Dresner, 2008), padre del Business Intelligence al definirlo como "El conocimiento ganado a través del acceso y análisis de la información empresarial", siendo este concepto la base de este paradigma.

Hablando en términos un poco más informáticos, el ya mencionado continuo y prácticamente imparable crecimiento de la cantidad de datos existente en internet fue una de las razones que en primera instancia, dio lugar a la aparición del término Big Data, y a partir de este los diferentes tipos de aplicaciones que se le da a esta tecnología, uno de los paradigmas más importantes que tomaron protagonismo debido a esto es lo que se conoce como Bussiness Intelligence, que en español traduce Inteligencia de Negocios, y se refiere a la forma en la cual las empresas logran recabar grandes cantidades de datos, que de por si se los obtiene de forma no estructurada, es decir desordenada y mediante la aplicación de diferentes métodos propios de la tecnología Big Data logran transformarla en información completamente relacionada y estructurada que será analizada para que sea útil en la empresa.

Coincidiendo con las definiciones anteriores, un concepto menos técnico es propuesto por la empresa (SINNEXUS, 2016) dedicada a ofrecer soluciones de Inteligencia de Negocios, al expresar que este término se refiere a "La habilidad para transformar los datos en información, y la información en conocimiento, de forma que se pueda optimizar el proceso de toma de decisiones en los negocios.", la idea principal en esto radica en que obteniendo la suficiente cantidad de datos acerca de su medio, como clientes, publicidad, estadísticas o informes anteriores, historiales, etc., una empresa sea capaz de tomar las decisiones correctas acerca de casi cualquier cosa, es decir con una conveniente cantidad de conocimientos sobre la mencionado anterior, es posible realizar las predicciones futuras correctas que permitan a la empresa salir adelante.

Era lógico pensar que, desde la concepción de este paradigma, siempre ha estado estrechamente relacionado a la cantidad de datos existentes para una empresa, y es ahora en la era informática cuando precisamente se evidenció que, debido a la enorme cantidad de datos existentes, el llegar a utilizarlos sería una tarea complicada, por lo que empezaron a aparecer soluciones de software y webs para la problemática que conlleva el análisis de datos masivos.

Data Warehousing Y Business Intelligence

Data Warehouse Y Data Mart

Según el argumento de Arredondo & Muñoz (2017), el término Data Warehouse o Almacenes de Datos en español, se refiere a "Un conjunto de datos orientados a temas específicos, integrados, que se acumulan en el tiempo, para apoyar la toma de decisiones, los cuales son integrados de distintas fuentes."(p. 74), tomando esto como punto de partida es posible notar la muy directa relación que tiene este concepto con el paradigma Business Intelligence, debido no únicamente a la aplicación general que va de la mano con la toma de decisiones en una empresa, sino más bien por el carácter analítico que conlleva el uso de un Data Warehouse en la tecnología Big Data, ya que representa una parte importante, y más que eso, una solución en el proceso de recolección, persistencia y análisis de datos. En concordancia con lo anterior, los Data Warehouse representan para cualquier tipo de organización un registro histórico que es analizado y convertido en información estratégica para el soporte en el proceso de toma de decisiones. (Cravero Leal, Sepulveda, Mate, Mazon, & Trujillo, 2014 ,p. 66)

Los Data Warehouse no deben ser comparados con las bases de datos transaccionales normales debido a que a pesar que cumplen funciones parecidas de almacenamiento de datos dentro de un contexto en una empresa, sus usos y orientaciones son totalmente distintos puesto que las bases de datos están diseñadas para la persistencia y operatividad de los datos, mientras que los Data Warehouse son usados específicamente como ayudas en la toma de decisiones y deben funcionar en conjunto con la base de datos general de la empresa.

Por otro lado, un Data Mart representa, de forma sencilla, un nivel inferior de un Data Warehouse, y está orientado por lo general a un cierto grupo de usuarios dentro de una empresa, los cuales consumen información de estos con un lenguaje determinado a partir

de modelos multidimensionales de base de datos, pueden ser implementados con las mismas herramientas que son aplicables a los Data Warehouse. Según Moody & Kortink (2000), los Data Marts son por lo general adaptaciones específicas de Almacenes de Datos diseñados para un cierto grupo de usuarios, pueden almacenarse de forma real, o creados virtualmente como vistas. (p. 7).

Desde las diferentes aristas sobre las que diversos autores se refieren a este tema hay una parte concurrente en todas las definiciones, y corresponde al fin último del uso de los Data Warehouses y Data Marts, en la cual explican que el punto central de la aplicación de esta tecnología radica en que deben servir como base y soporte para la toma de decisiones importantes en una empresa analizada desde el punto de vista de los datos relacionados a esta decisión que se posean actualmente.

OLAP y OLTP

El término OLAP, On-Line Analytical Processing, y Procesamiento Analítico On-Line en español, se refiere a una tecnología por lo general aplicada sobre Data Warehouses para el análisis estratégico de la información con el fin de segmentar datos, generando reportes de tipo estadístico, o encontrado tendencias y patrones que ayuden en la toma de decisiones.

"En general, estos sistemas deben soportar requerimientos complejos de análisis, analizar datos desde diferentes perspectivas y soportar análisis complejos contra un volumen ingente de datos. La funcionalidad de estos sistemas se caracteriza por ser un análisis multidimensional de datos corporativos, que soportan los análisis del usuario y posibilidades de navegación, seleccionando la información a obtener." (Hernández Lamillar, 2005, p. 4)

Por otro lado, el término OLTP, hace referencia al concepto de On-Line Transactional Processing, o Procesamiento Transaccional On-Line en español, y se refiere a la tecnología usada sobre todo en las bases de datos comunes operativas que utilizaría cualquier empresa, sobre las cuales se pueden llevar acabo las operaciones básicas de creación, lectura, actualización y eliminación. Moody & Kortink (2000), comparan ambos conceptos argumentando que el concepto OLAP es usado sobre un ambiente diseñado para ser del tipo Data Warehouse, y es profundamente diferente al ambiente operacional sobre el cual trabaja la tecnología OLTP.

CAPÍTULO II

METODOLOGÍA

2.1. TIPO DE INVESTIGACIÓN

La investigación presentada, primeramente, es una Investigación Documental, debido al carácter bibliográfico sobre el cual se desarrollaron sus fuentes teóricas que sirven como base para toda la parte práctica de la investigación misma.

Esta investigación también es considerada Experimental, debido a la forma en la cual se manejaron las variables necesarias para producir un resultado, lo que en este caso corresponde a la generación de una cantidad suficiente de información con una estructura definida, información que fue procesada y analizada por la aplicación desarrollada para determinar si es factible considerar MEAN como un stack de para el manejo de Big Data, y las aplicaciones que esta tecnología conlleva.

Por otro lado, la investigación es Descriptiva, debido al análisis presentado sobre las propiedades y características del stack de desarrollo MEAN a partir de su comportamiento concreto frente a grandes cantidades de datos, así como las posibilidades que presenta la tecnología Big Data aplicada a la web incluyendo el análisis estadístico de la información procesada.

Finalmente, la investigación es considerada Cuantitativa por la manera en la que se examinaron los datos, de una forma científica y matemática para comprender la incidencia de la tecnología Big Data y sus aplicaciones, en el desarrollo web, además de la capacidad del stack de desarrollo MEAN para procesar los datos y obtener resultados en base a estos.

2.2. MÉTODOS Y TÉCNICAS

El método usado para la presente investigación corresponde al Método Inductivo puesto que se han tomados en cuenta todos los elementos de la parte práctica en consonancia con

el estudio realizado sobre la parte teórica y el posterior análisis de los resultados obtenidos en base a los objetivos planteados, para llegar a una conclusión que demuestre o no la factibilidad del stack de desarrollo MEAN aplicado a la tecnología Big Data para el manejo y análisis de grandes cantidades de información.

De igual manera se usó el Método Experimental como medio para esclarecer las propiedades y características del stack MEAN frente a determinados casos y condiciones que se provocaron intencionalmente como parte del estudio y evaluación que se realizó sobre el stack de desarrollo.

La principal técnica que se usó para la recolección de datos es el manejo de la aplicación como tal, desarrollada mediante el stack MEAN, y sobre la cual se realizaron las respectivas pruebas de rendimiento, y de carga y estrés sobre la base de datos, puesto que esta fue estudiada con una cantidad tal de información, suficiente para el análisis de su comportamiento.

2.3. POBLACIÓN Y MUESTRA DE ESTUDIO

Debido a la naturaleza misma de la investigación realizada, los valores estadísticos correspondientes a la población y muestra sobre un universo de estudio determinado no aplican.

2.4. DESCRIPCIÓN Y VALIDACIÓN DEL INSTRUMENTO

Al ser una investigación orientada a determinar la factibilidad del stack de desarrollo MEAN para aplicaciones Big Data, el principal instrumento es la aplicación misma, la cual fue desarrollada con el propósito de soportar grandes cantidades de información y su correspondiente análisis, por lo que es descrita como una aplicación web de tipo red social que constará de registros e información relacionadas a usuarios y sus distintos tipos de preferencias propias de estos, datos sobre los cuales es posible llevar a cabo diferentes tipos de análisis con un objetivo.

MEAN Social Network es una aplicación web creada con la estructura y características básicas de una red social, con la particularidad que el entorno completo sobre el cual funciona fue diseñado únicamente con frameworks cuyo lenguaje de programación es JavaScript, y la base de datos con la cual funciona es MongoDB, herramientas que en su conjunto forman el denominado stack de desarrollo MEAN.

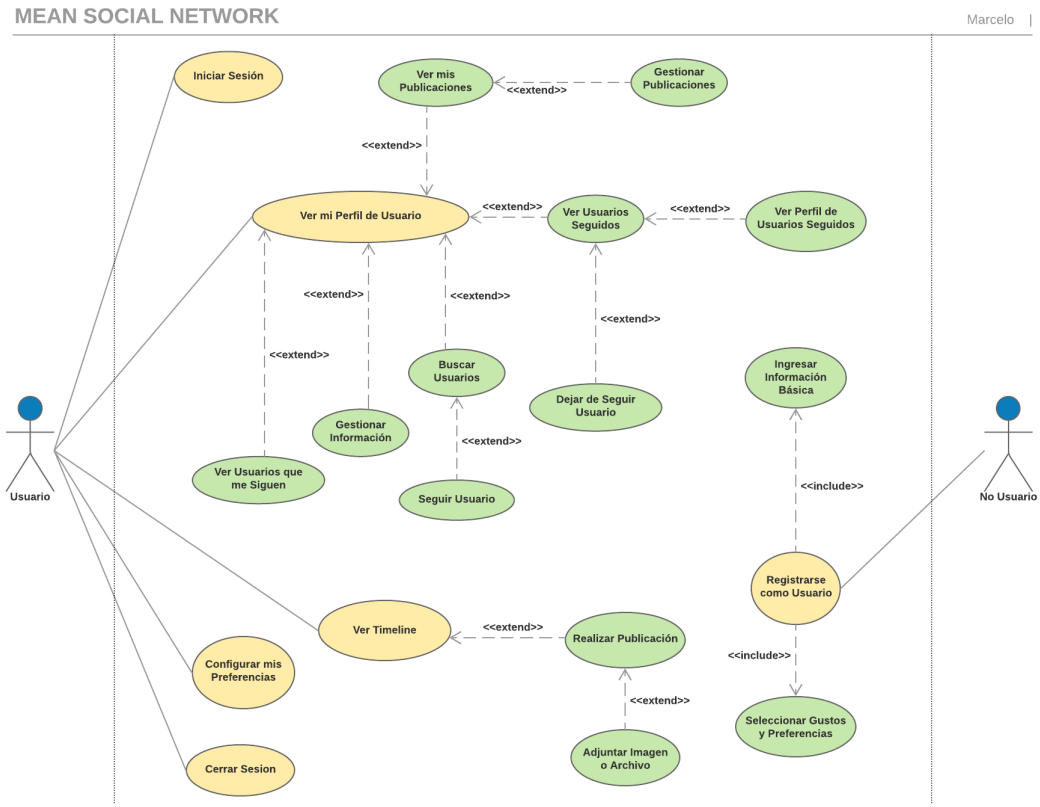


Figura 6. *Diagrama de Casos de uso de la aplicación desarrollada.* ***Fuente Autor. Ver ANEXO 1.***

Dentro de los requisitos funcionales en las especificaciones de requerimientos de software tenemos:

- Autentificación de Usuarios
- Registro de Usuarios
- Perfil de Usuario
- Timeline
- Configuración de Preferencias

Mientras que dentro de los requisitos no funcionales tenemos:

- Rendimiento
- Seguridad
- Fiabilidad
- Disponibilidad
- Portabilidad

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Autentificación de Usuario.
Características:	Los usuarios deberán identificarse para acceder a cualquier parte del sistema.
Descripción del requerimiento:	El sistema podrá ser accedido por cualquier usuario autentificado siempre y cuando este haya seguido los pasos de registro correspondientes.
Requerimiento NO funcional:	• RNF01 • RNF02 • RNF05
Prioridad del requerimiento: Alta	

Tabla 2. *Ejemplo de Requerimiento Funcional*. ***Fuente Autor. Ver ANEXO 1.***

La ***Tabla 2*** corresponde a una tabla explicativa de ejemplo de uno de los requerimientos funcionales de la aplicación presentado a manera de Historia de Usuario y según el formato IEEE 830 de Especificación de Requerimientos de Software.

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Interfaz del sistema.
Características:	El sistema presentara una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema.
Descripción del requerimiento:	El sistema debe tener una interfaz de uso intuitiva y sencilla.
Prioridad del requerimiento: Alta	

Tabla 3. *Ejemplo de No Requerimiento Funcional*. ***Fuente Autor. Ver ANEXO 1.***

La ***Tabla 3*** corresponde a una tabla explicativa de ejemplo de uno de los requerimientos no funcionales de la aplicación presentado a manera de Historia de Usuario y según el formato IEEE 830 de Especificación de Requerimientos de Software.

Para la especificación completa de requerimientos de software refiérase al ANEXO 1.

2.5. TÉCNICAS DE PROCESAMIENTO Y ANÁLISIS DE DATOS

Para el procesamiento y análisis de los resultados obtenidos durante la investigación, además de la propia aplicación desarrollada, fueron necesarias herramientas especializadas para llevar a cabo un correcto análisis de la toda la información generada, y así cumplir con el proceso completo del Big Data que no únicamente es la recolección de grandes cantidades de información, sino también el análisis de esta. De igual manera, fueron necesarias también herramientas dedicadas a las pruebas de software para obtener las estadísticas de funcionamiento de la aplicación deseadas y así realizar el análisis adecuado del funcionamiento de esta.

Al evaluar el funcionamiento general del stack MEAN como herramienta adecuada para aplicaciones Big Data, se aplicó uno de los conceptos del Data Wharehousing conocido como ETL, Extract – Transform – Load por sus siglas en inglés, lo cual corresponde a un análisis completo de la información obtenida desde una fuente, permitiendo filtrar datos, además de generar reportes y estadísticas necesarias para darle una utilidad a dicha información.

Con base en lo anterior, fueron evaluadas diversas herramientas para este fin, teniendo en cuenta sobre todo la capacidad de conexión e integración que presenten con la base de datos por excelencia de MEAN, MongoDB, y como objetivo principal generar utilidad para la información persistente de la aplicación:

- Stitch
- Pentaho
- Alooma
- Scriptella
- Apatar

Una vez analizadas las herramientas y conociendo la complejidad de los datos que pudieron haber sido generados, se consideró el uso de Pentaho en su versión de prueba de 30 días para llevar a cabo el proceso de análisis de la información y generar un Data Warehouse en base a esta, todo esto con el fin de determinar qué tan accesible y conveniente puede llegar a ser el análisis Big Data con información obtenida de una aplicación desarrollada con el stack MEAN.

Continuando con la descripción de las técnicas de procesamiento y análisis, para la correcta evaluación del funcionamiento global de la aplicación, y su desempeño en cuanto a análisis y carga de información, es necesario además realizar diferentes tipos de pruebas de software con el fin de analizar los puntos críticos de la aplicación misma.

Según Mera (2016), es posible dividir los tipos de pruebas de software en tres clases generales, Pruebas Funcionales, las cuales están basadas en la interoperabilidad de las operaciones que realiza un determinado sistema, Pruebas No Funcionales, las cuales corresponden a la medición de las características de una aplicacioón que puedan ser cuantificadas mediante una determinada escala, y Pruebas Estructurales, las cuales están centradas en los procedimientos del software, a partir de su propio código fuente, examinando los posibles flujos de ejecución en los procedimientos mediante diversas entradas que produzcan salidas controladas

Debido a la naturaleza y el tipo de la aplicación que se desarrolló, el conjunto de pruebas a considerar para su evaluación corresponde a las pruebas no funcionales, y en relación a los aspectos más importantes en aplicaciones orientadas a Big Data, las siguientes pruebas resaltan como principales métodos para analizar el funcionamiento y las características de la aplicación desarrollada con el stack MEAN.

PRUEBA	OBJETIVO
De Carga	Es un tipo de prueba cuyo objetivo es medir el rendimiento del software en diferentes escenarios. Las pruebas de carga consisten en simular demanda sobre una aplicación de software y medir el resultado. Estas pruebas se realizan bajo el concepto demanda esperada y también en condiciones que puedan producir una sobrecarga, con el fin de observar el cómo reacciona el software en diferentes situaciones.
De Estrés	Es un tipo de prueba que por lo general busca determinar la solidez de la aplicación al tratar de llegar al límite de la capacidad operativa tanto de la aplicación misma, como de la base de datos. Este tipo de prueba de software se utiliza para determinar la estabilidad de un sistema o aplicación, con especial atención en la disponibilidad y manejo de errores cuando se enfrenta a la sobrecarga.
De Tiempo de Respuesta	El objetivo de este tipo de prueba consiste en medir el tiempo de respuesta de la aplicación sobre los distintos tipos de operaciones y consultas a los que va a estar sometido, dependiendo así mismo de la cantidad de registros a los cuales tiene que acceder.

Tabla 4. *Pruebas de Software No Funcionales a realizar*. ***Fuente Mera (2016).***

La ***Tabla 4*** corresponde a una tabla explicativa con las definiciones de los tipos de pruebas no funcionales que serán llevadas a cabo para evaluar el funcionamiento de la aplicación.

Dicho lo anterior, para llevar a cabo las respectivas pruebas se consideraron las siguientes herramientas de evaluación, todas con el objetivo de realizar las pruebas descritas anteriormente y haciendo énfasis en el rendimiento general de la aplicación:

- Funkload
- SoapUI
- Jmeter
- WAPT
- WebLoad

Una vez analizadas las herramientas, y considerando los tipos necesarios de pruebas para analizar los puntos críticos en la aplicación se consideró adecuado el uso de JMeter como principal herramienta para llevar a cabo la ejecución de las pruebas, puesto que cuenta con las funcionalidades necesarias para obtener los datos que permitan realizar un análisis a nivel no funcional y tener una perspectiva más completa del rendimiento de la aplicación desarrollada. Dicho lo anterior, JMeter fue complementado con la herramienta online Blazemeter para simular concurrencia de usuarios durante la prueba de carga y la prueba de estrés. Luego de obtener los resultados requeridos mediante la aplicación de las pruebas, estos serán analizados con un enfoque de caja negra y dentro del marco de referencia correspondiente a las métricas relacionadas al Comportamiento Temporal y Cumplimiento de la Eficiencia del Software, pertenecientes a la característica de Eficiencia en el margen de la norma ISO/IEC 9126 para la Calidad del Software.

2.6. NORMAS ÉTICAS

Durante la realización de esta investigación y como futuro profesional me comprometo a no caer en el plagio, así como a utilizar los datos obtenidos mediante este medio de la mejor manera, con responsabilidad y ética profesional, a mantener la veracidad de la información aquí expuesta y preservar la privacidad de los datos que se puedan utilizar durante el desarrollo y prueba de la aplicación.

CAPÍTULO III

RESULTADOS

3.1. APLICACIÓN WEB BIG DATA MEDIANTE EL STACK MEAN

Para el cumplimiento de los objetivos planteados en esta investigación, se llevó a cabo el desarrollo de una aplicación web orientada al uso de la tecnología Big Data mediante el stack de desarrollo MEAN. La aplicación consiste en una página web de tipo red social, misma que contará con registros de usuarios con diferentes características, haciendo posible la segmentación y el análisis estadístico.

A nivel funcional la aplicación cuenta con las principales características de una red social, registro, login, y perfiles de usuarios, edición de los datos, sistema de seguimiento, sistema de publicación de contenido incluyendo ficheros adjuntos, sistema de mensajería privada, conteo de estadísticas principales, timeline dependiendo de los usuarios seguidos y un apartado donde se muestran los usuarios que usan la aplicación.

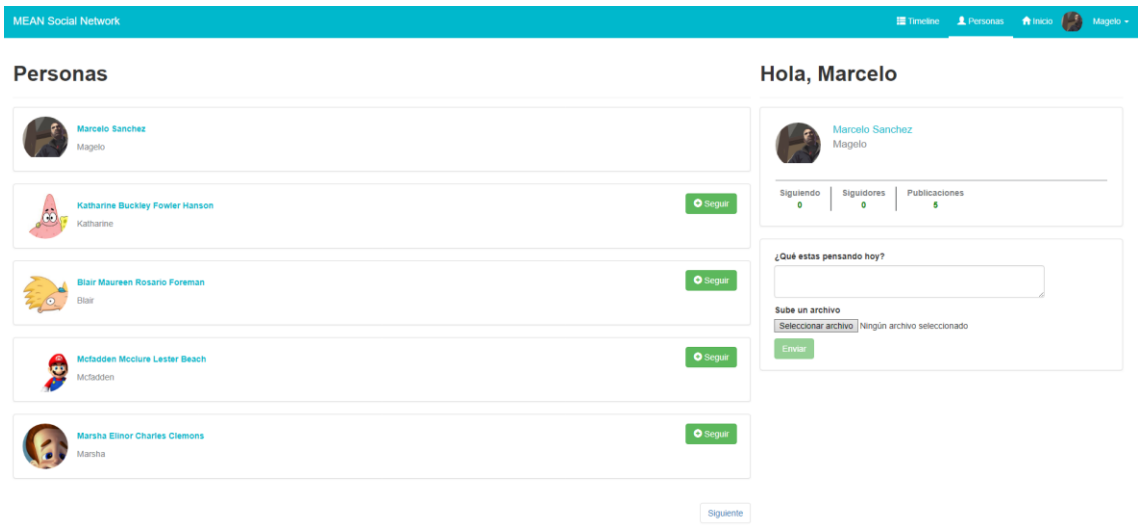


Figura 7**. Pantalla "Personas" de la aplicación web desarrollada.* ***Fuente Autor.

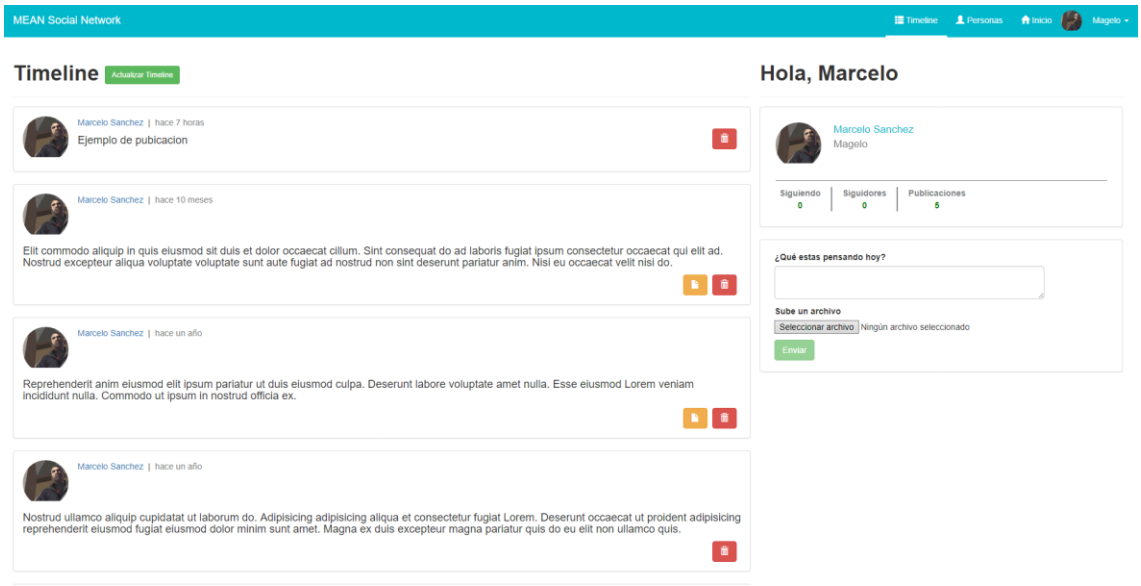


Figura 8. *Pantalla "Timeline" de la aplicación web Desarrollada.* ***Fuente Autor.***

A nivel estructural, la aplicación fue diseñada siguiendo el paradigma Modelo – Vista – Controlador tanto en el backend como en el frontend, por lo cual es posible identificar diferentes directorios sobre los cuales se encuentran los respectivos modelos para las partes críticas de la aplicación, siendo los principales el directorio “api/”, y el directorio “frontend-angular/”.



Figura 9. *Estructura principal de la aplicación para los directorios "api/", y "frontend-angular/".* ***Fuente Autor.***

Teniendo en cuenta la ***Figura 9***, es posible caracterizar el proceso de funcionamiento de la aplicación en base a sus directorios de la siguiente manera:

- El directorio "api/" contiene la estructura del backend sobre el cual el frontend puede realizar peticiones para realizar funcionalidades o acceder a información guardada en la base de datos.
- En los directorios "api/models/" y "api/controllers/", se encuentran definidos los modelos y controladores para dichos modelos respectivamente, siendo su funcionamiento igual al de cualquier otra aplicación que siga este paradigma, los

modelos definidos son: “follow”, “message”, “publication” y “user”, los cuales tienen un controlador asociado que determina las funcionalidades que puedan realizar, así como su lectura o escritura en la base de datos.

- Otro directorio importante es “api/routes/”, en el cual se definen las rutas a las cuales deberá acceder el frontend para las peticiones que necesite realizar.
- En el directorio “frontend-angular/” se encuentra todo el código Javascript necesaria para mostrar una interfaz agradable al usuario, así como las llamadas respectivas al backend para el acceso a la información.
- Los directorios “componets/”, “services/”, y “models/”, contienen los componetes, servicios y modelos respectivamente, necesarios para el funcionamiento del frontend y la correcta comunicación con el backend de la aplicación.

A nivel de base de datos, como es mencionado anteriormente en este documento, el stack de desarrollo MEAN trabaja con la base de datos NoSQL MongoDB, sobre la cual se guardan las colecciones que hacen referencia a los principales modelos de la aplicación definidos tanto en el backend como en el frontend.

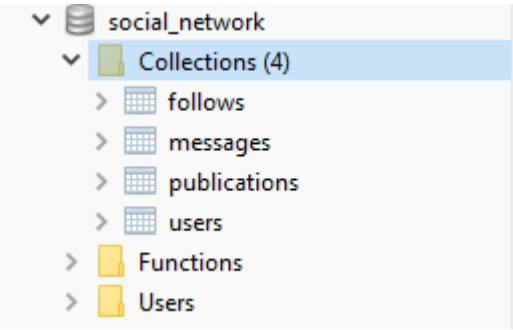


Figura 10. *Principales colecciones de la base de datos.* ***Fuente Autor.***

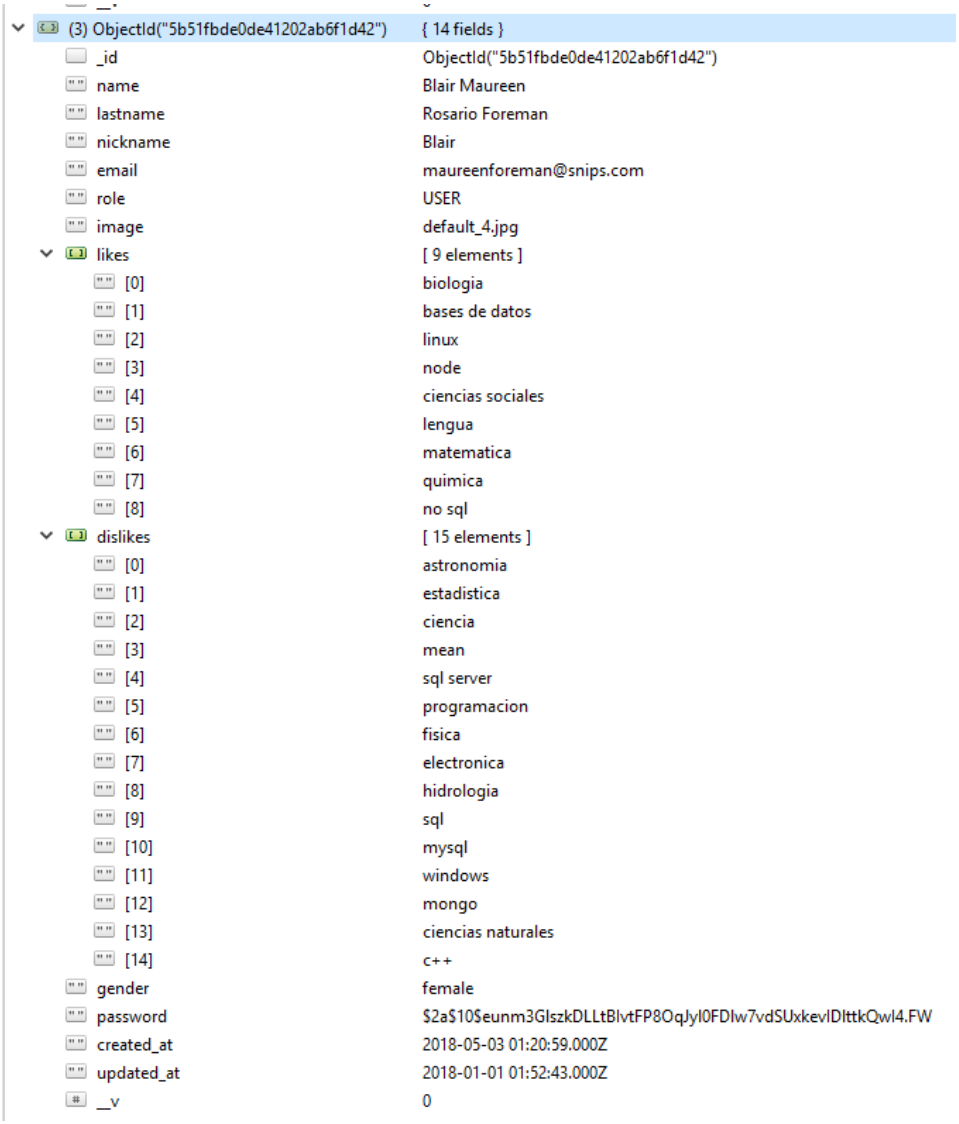


Figura 11. *Ejemplo básico de un documento de la colección "users" en la base de datos.*
Fuente Autor.

Para el despliegue de la aplicación se usó un servidor corriendo con el sistema operativo Linux Ubuntu en su versión 18.04, alojado en un droplet dedicado, contratado desde la empresa DigitalOcean, para uso específico de esta aplicación, mismo que cuenta con las siguientes características:

- 8 Gb en memoria RAM
- 4 vCpu’s
- Disco de estado sólido de 160GB

La información que se encuentra en la base de datos fue generada de forma aleatoria a partir de scripts creados en base a los requerimientos de modelos de la aplicación, y ejecutados en un generador de formatos JSON, disponible en: https://www.json-generator.com.

JSON GENERATOR

```
    name: '{{firstName()}} {{firstName()}}',
    lastname: '{{surname()}} {{surname()}}',
    nickname: function () {
      var nick = (this.name).split(" ", 1);
      return nick[0];
    },
    email: '{{email()}}',
    role: 'USER',
    image: function (tags) {
      var random = tags.integer(0, 10);
      return 'default_' + random + '.jpg';
    },
    likes: function (tags) {
      var likes = ['matematica', 'ciencia', 'astronomia', 'biologia', 'lengua', 'estadistica',
                   'fisica', 'quimica', 'psicologia', 'informatica', 'programacion', 'ciencias sociales',
                   'ciencias naturales', 'ecologia', 'hidrologia', 'electronica', 'mecanica',
                   'medio ambiente', 'bases de datos', 'desarrollo de software', 'sql', 'no sql',
                   'sql server', 'mysql', 'oracle', 'mongo', 'linux', 'windows', 'php', 'javascript',
                   'mean', 'c++', 'java', 'express', 'angular','node'];
      var array = [];
      var used = [];
      var size = likes.length;
      var random = tags.integer(3, size-1);
      var i;
      for (i=0; i<random; i++) {
        var random_aux = tags.integer(0, size-1);
        var like = likes[random_aux];
        var test = used.indexOf(like);
        while (test !== -1) {
          random_aux = tags.integer(0, size-1);
          like = likes[random_aux];
          test = used.indexOf(like);
        }
        array.push(like);
        used.push(like);
      }
      return array;
    },
    dislikes: function (tags) {
      var dislikes = ['matematica', 'ciencia', 'astronomia', 'biologia', 'lengua', 'estadistica',
                   'fisica', 'quimica', 'psicologia', 'informatica', 'programacion', 'ciencias sociales',
                   'ciencias naturales', 'ecologia', 'hidrologia', 'electronica', 'mecanica',
                   'medio ambiente', 'bases de datos', 'desarrollo de software', 'sql', 'no sql',
                   'sql server', 'mysql', 'oracle', 'mongo', 'linux', 'windows', 'php', 'javascript',
                   'mean', 'c++', 'java', 'express', 'angular','node'];
      var new_dislikes = [];
      var size_aux = dislikes.length;
      var used = this.likes;
      var j;
      for (j=0; j<size_aux; j++) {
        if (used.indexOf(dislikes[j]) === -1) {
          new_dislikes.push(dislikes[j]);
        }
      }
      var size = new_dislikes.length;
      var random = tags.integer(0, size-1);
      var array = [];
      var new_used = [];
```

Created by Vazha Omanashvili

Figura 12. *Ejemplo de script usado para generar datos de usuarios aleatorios.*
Fuente https://www.json-generator.com.

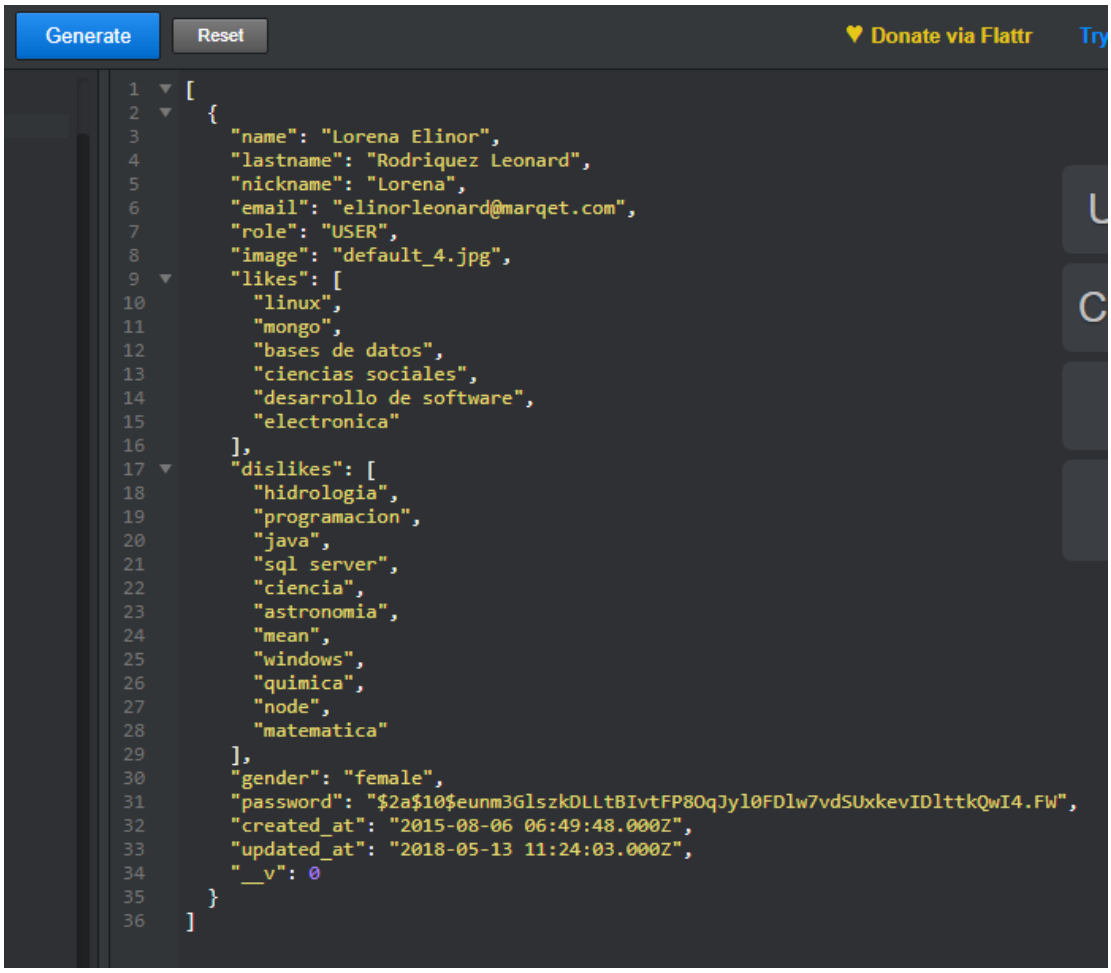


Figura 13. *Ejemplo del JSON generado de forma aleatoria según el modelo de Usuario.*
Fuente https://www.json-generator.com.

3.2. HERRAMIENTAS USADAS EN EL ANÁLISIS BIG DATA

Una de las partes más importantes del Big Data, por no decir la más crítica de todas, es la capacidad de análisis que se pueda obtener de la información procesada, por parte de una aplicación o software orientado a esta tecnología, dicho esto, son muchas las maneras que existen para llevar acabo el análisis Big Data, pero todas ellas coinciden en un objetivo principal, el cual corresponde a que los datos obtenidos y analizados tengan una utilidad determinada, es decir un fin específico por el cual se realice el análisis.

Tomando como punto de partida el análisis obtenido mediante la investigación realizada en las bases teóricas científicas de este documento acerca de la tecnología Big Data y sus aplicaciones, y haciendo énfasis en el apartado de Herramientas, software y soluciones

Big Data del punto 2.2.3, es posible identificar varios puntos clave que son aplicados actualmente al momento de llevar a cabo análisis de grandes cantidades de datos.

Empezando por el algoritmo MapReduce, por ejemplo, ya que representó una evolución en la forma de procesar información, y en base a este se han generado, y se siguen generando más y mejores herramientas y software capaces de evaluar conjuntos de datos que por sus características puedan llegar a ser una representación de Big Data.

De igual manera, cabe mencionar también que el nacimiento de las bases de datos no estructuradas figura como otro acontecimiento que provocó una mejora sustancial en la forma en la cual se analiza la información, no solo porque representa una manera más ligera de procesamiento, sino también que por el simple hecho de ser no estructuradas se ajusta perfectamente a la variedad de datos que pueden existir al momento de llevar acabo análisis Big Data con información que no ha pasado por ningún tipo de filtro.

Continuando con la lógica anterior, otro punto importante a favor del análisis Big Data es el creciente desarrollo de nuevos lenguajes de programación, así como la adaptabilidad de los antiguos, que por la forma en la que son interpretados permiten llevar a cabo operaciones de análisis de datos, y no siendo suficiente esto, también es posible generar nuevas funcionalidades en un lenguaje específico que facilitan el manejo de datos, ejemplo de esto último es el lenguaje de programación Python, el cual según Paruchuri (2016), es uno de los lenguajes mejores construidos para el desarrollo de herrameintas de analítica de datos, de igual manera el autor señala casi bajo el mismo propósito a R y lo caracteriza como uno de los lenguajes con la librería más completa para llevar acabo análisis estadísticos como parte de Big Data. No obstante, ambos son relativamente viejos, ya que en la actualidad cuentan con alrededor de 25 a 28 años de antigüedad, siguiendo esta premisa es posible resaltar de igual manera el lenguaje de programación base del stack de desarrollo objeto de la presente investigación, JavaScript, cuya primera aparición fue en el año 1995, y que se encuentra actualmente en auge debido a sus constante uso en el desarrollo de tecnologías web, sobresaliendo en el procesamiento de datos en aplicaciones de este tipo, de igual manera, cabe resaltar que también se han dado a conocer lenguajes de programación nuevos cuyas funcionalidades pueden ser orientadas al Big Data, ejemplo de esto, es el que propone (Pankaj, 2018) en una de sus publicaciones titulada “Golang for Big Data”, donde analiza el uso de Golang como herramienta dedicada al proceso completo de análisis de datos desde una perspectiva en la que refuta

la desconfianza en lenguajes de programación relativamente nuevos al ser aplicados en tecnologías en auge como es el Big Data.

3.3. PARADIGMA BUSINESS INTELLIGENCE PARA EL ANÁLISIS DE INFORMACIÓN

Como fue explicado anteriormente en este documento, la tecnología Big Data no involucra únicamente la recolección de información a enorme escala, sino también el respectivo análisis de esta, el qué hacer con la información que se ha obtenido, en pocas palabras darle un uso, y más allá de esto, también es necesario tener en claro el cómo hacerlo.

A la par de la evolución del Big Data, los conceptos relacionados con esta tecnología fueron desarrollándose cada vez más, de manera que facilitar el acceso y uso de la información se vuelva una tarea más sencilla y eficaz, ejemplo de esto es el Business Intelligence, ya que el uso de este paradigma se ha vuelto casi que obligatorio para el análisis de información Big Data, y de la mano de esto es necesario también mencionar uno de los procesos que involucra, como es el ETL o Extract Transform Load.

Para el respectivo análisis de la información generada mediante la aplicación MEAN SN, se utilizó la herramienta Pentaho Data Integration, capaz de realizar el proceso de ETL en base a los parámetros de búsqueda y clasificación dados desde las configuraciones respectivas hacia la base de datos de la cual obtendrá la información.

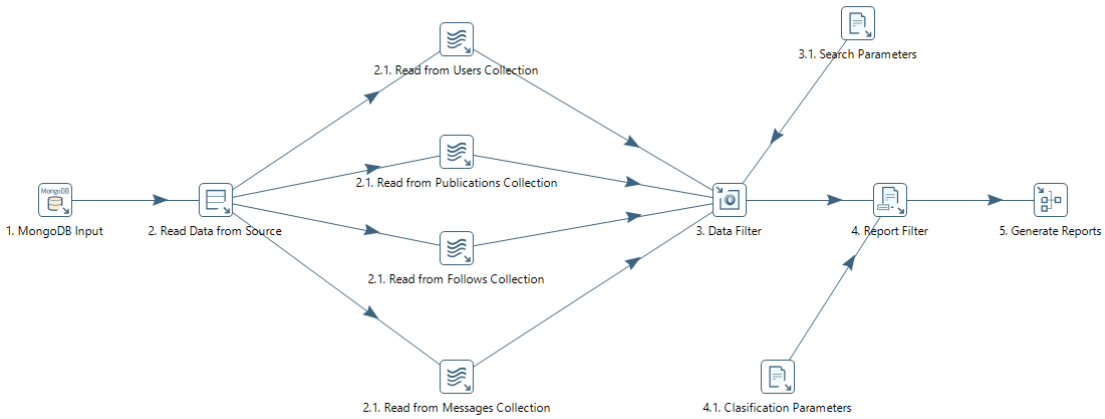


Figura 14. *Proceso básico Extract Transform Load aplicado a las colecciones de la base de datos de la aplicación MEAN SN.* ***Fuente Autor. Desde la herramienta Pentaho.***

En la ***Figura 14*** se aprecia de forma resumida el proceso ETL básico correspondiente a la creación de reportes mediante la herramienta Pentaho, descrito en los siguientes pasos:

1. **MongoDB Input:** Configuración de la conexión entre la herramienta y la base de datos de la aplicación.
2. **Read Data from Source:** Este paso corresponde al Extract dentro del proceso ETL, ya que, una vez realizada la conexión, la herramienta obtiene los datos sin ningún tipo de filtro desde la base de datos y cada una de sus colecciones.
    - **2.1. Read from Users Collection**
    - **2.2. Read form Publications Collection**
    - **2.3. Read from Follows Collection**
    - **2.4. Read from Messages Collection**
3. **Data Filter:** Una vez obtenidos los datos desde cada colección, se realiza el filtro correspondiente de búsqueda según los parámetros especificados, de esta manera la información obtenida al final estará ordenada según el uso que se le dará, este paso corresponde a la primera parte del Transform dentro del proceso ETL.
    - **3.1. Search Parameter:** Corresponde a la definición de los parámetros sobre los cuales la información será filtrada.

4. **Report Filter:** Cuando la información haya pasado por el primer filtro se procede a definir un parámetro que determinará el tipo de reporte que se generará al final, este paso corresponde a la segunda parte del Transform dentro del proceso ETL
    - **4.1. Clasification Filter:** Corresponde a la definición de los parámetros sobre los cuales la información será clasificada para ser mostrada en los reportes.
5. **Generating Reports:** Corresponde al paso final dentro del proceso ETL, Load, en el cual la información totalmente filtrada es "cargada" en un Data Warehouse que fue generado sobre la base de datos de la misma aplicación desarrollada con el satck Mean, para generar los reportes estadísticos correspondientes según el uso que se les dará a los datos.

Para esta investigación se usó como modelo base los datos correspondientes a la colección Users, misma que contiene toda la información guardada de los usuarios que utilizan la aplicación. Teniendo como base los datos de un usuario se realizaron los filtros correspondientes para tener información sobre la cual empezar a generar una base de conocimientos.

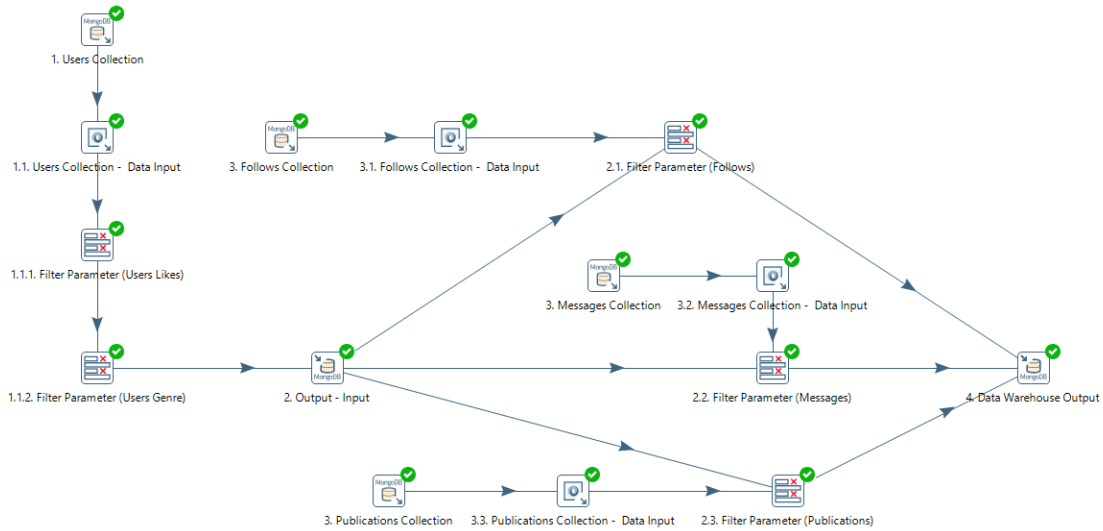


Figura 15. *Ejecución del esquema final del proceso Extract Transform Load aplicado a los datos de la aplicación MEAN SN.* ***Fuente Autor. Desde la herramienta Pentaho.***

El gráfico anterior correspondiente a la ***Figura 15*** muestra el esquema final del proceso seguido para la extracción, filtración y generación de la información necesaria para la

creación de reportes desde los datos persistentes en la aplicación MEAN SN, es posible apreciar los siguientes pasos:

1. **Users Collection:** Configuración de la conexión entre la herramienta y la base de datos de la aplicación MEAN SN, para la colección Users sobre la cual se obtienen los datos de los usuarios de la aplicación.
    - **1.1. Users Collection – Data Input:** Obtención de los datos de manera temporal para el tratamiento de la información dentro del contexto de la aplicación.
    - **1.2. Filter Parameter (Users Likes):** Parámetro que sirve como filtro para la información de entrada de la colección con la información de usuarios, filtrando los datos correspondientes a los gustos y preferencias de los usuarios, es decir los atributos *likes* y *dislikes* dentro de la colección.
    - **1.3. Filter Parameter (Users Gender):** Parámetro que sirve como filtro para la información de entrada de la colección con la información de usuarios, filtrando los datos correspondientes al género de los usuarios, es decir el atributo *genre* dentro de la colección.
2. **Output - Input:** Este paso corresponde a la creación de una colección auxiliar con los datos bases para los posteriores reportes, y de la cual se obtendrá los filtros para las búsquedas relacionadas en las siguientes entradas de datos, dichos filtros realizarán las respectivas búsquedas que coincidan con lo establecido en los gustos y preferencias de cada usuario, búsquedas que serán realizadas sobre sus Publicaciones. Información de seguimiento de usuarios y Mensajes.
    - **2.1. Filter Parameter (Follows):** Parámetro obtenido de la colección base que servirá como filtro para las búsquedas en la colección Follows.
    - **2.2. Filter Parameter (Messages):** Parámetro obtenido de la colección base que servirá como filtro para las búsquedas en la colección Messages.
    - **2.3. Filter Parameter (Publications):** Parámetro obtenido de la colección base que servirá como filtro para las búsquedas en la colección Publicationes.
3. **Entrada de datos desde las diferentes colecciones:** Configuración de la conexión entre la herramienta y la base de datos de la aplicación MEAN SN, para

las colecciones Follows, Messages y Publicationes sobre las cuales se obtienen los datos relacionados a las diferentes actividades de cada usuario de la aplicación, información en la cual se harán las respectivas búsquedas con los filtros establecidos anteriormente. Los puntos **3.1.**, **3.2.** y **3.3.** corresponden a la obtención de datos de cada colección desde la conexión establecida.

4. **DataWarehouse Output:** Corresponde a la salida final de la información con todos los filtros respectivos realizados, se generó una colección sobre la misma base de datos de la aplicación realizada con el stack Mean, misma colección que servirá de entrada para la realización de los reportes respectivos.

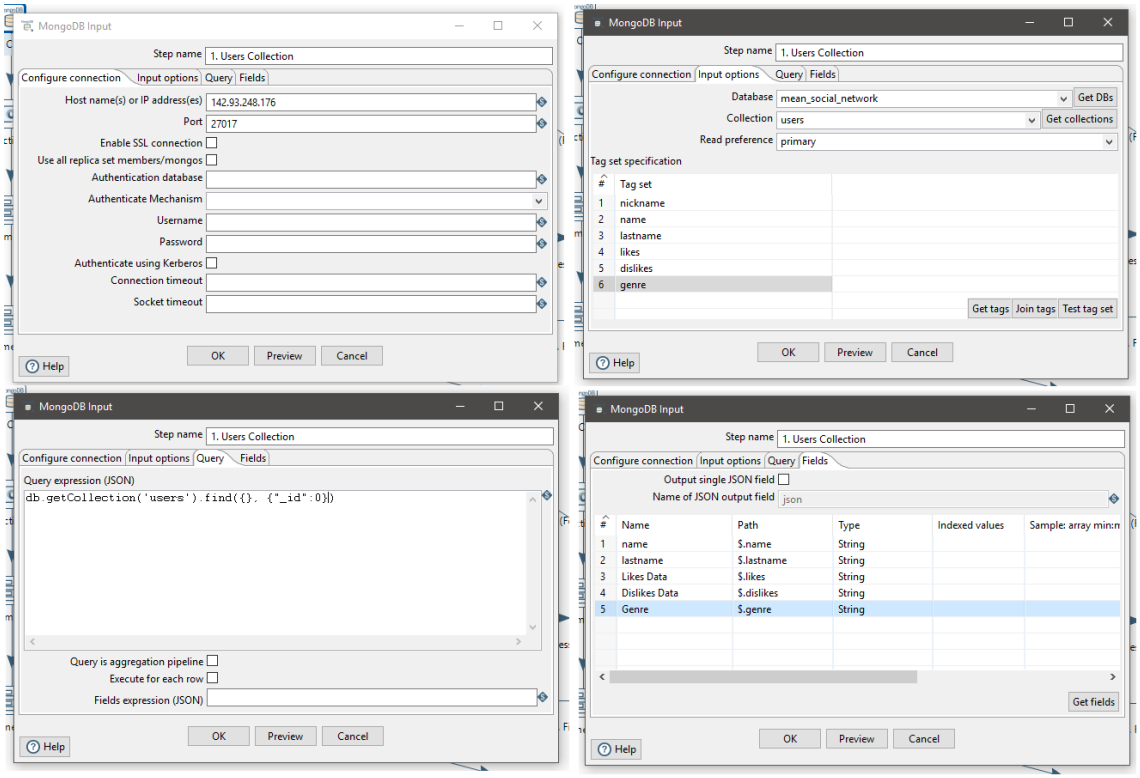


Figura 16. *Ejemplo de la configuración realizada para el punto 1. Users Collection.* ***Fuente Autor. Desde la herramienta Pentaho.***

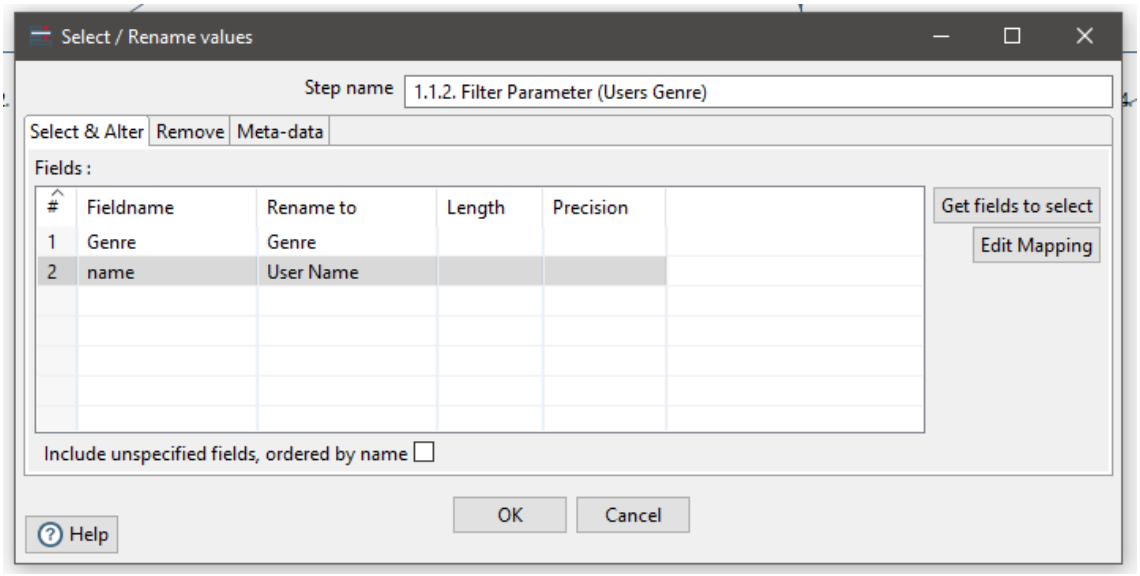


Figura 17. *Ejemplo de la configuración realizada para el punto 1.1.2. Filter Parameter (Users Genre).* ***Fuente Autor. Desde la herramienta Pentaho.***

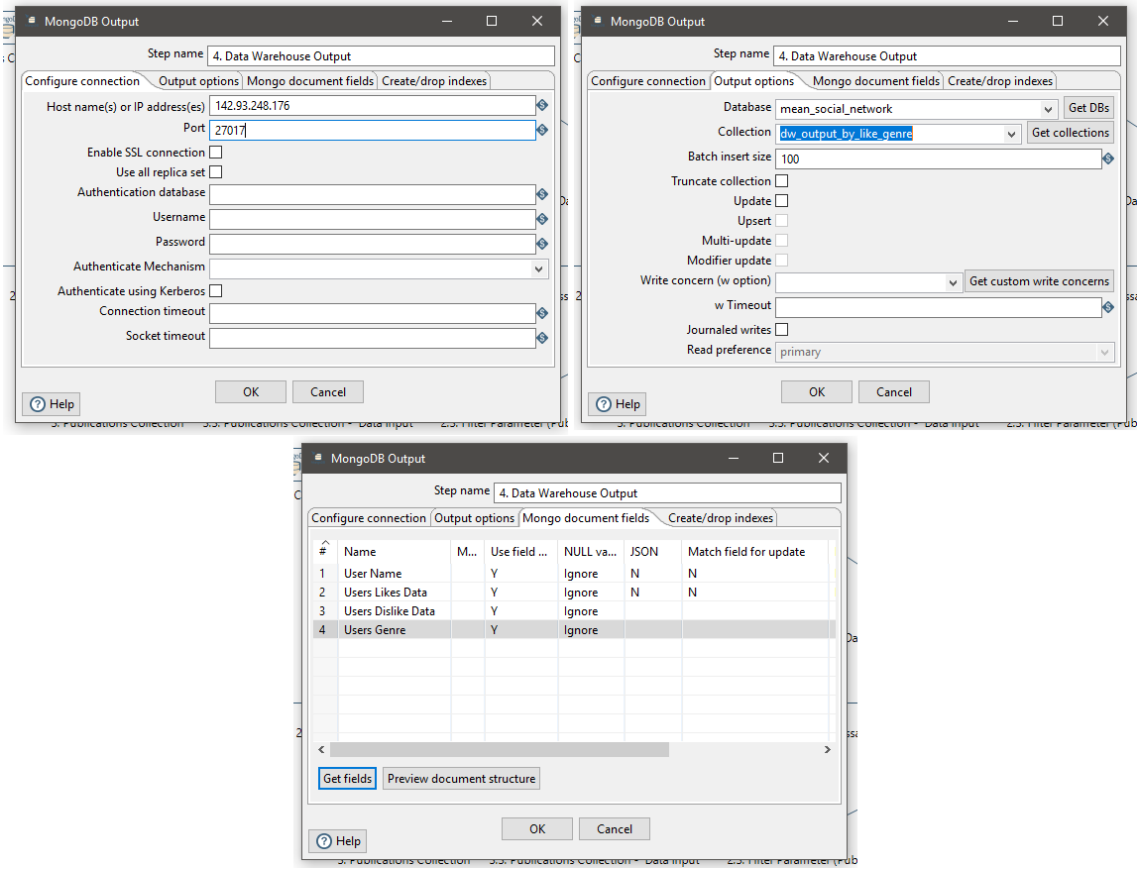


Figura 18. *Ejemplo de la configuración realizada para el punto 4. DataWarehouse Output.* ***Fuente Autor. Desde la herramienta Pentaho.***

Una vez filtrado los datos necesarios para obtener reportes significativos de la información, se procedió a utilizar el diseñador de reportes disponible desde la herramienta Pentaho. Cabe resaltar que para este punto, durante el proceso de filtración y generación de datos, fueron añadidas colecciones nuevas a la base de datos de la aplicación.

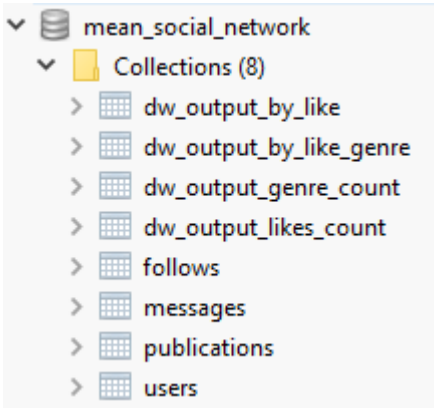


Figura 19. *Principales colecciones de la base de datos luego de la aplicación del Business Intelligence.* ***Fuente Autor.***

Para la correcta generación de los reportes es necesario realizar la configuración básica desde el diseñador de reportes para la obtención de la información dentro del DataWarehouse generado en la base de datos de la aplicación.

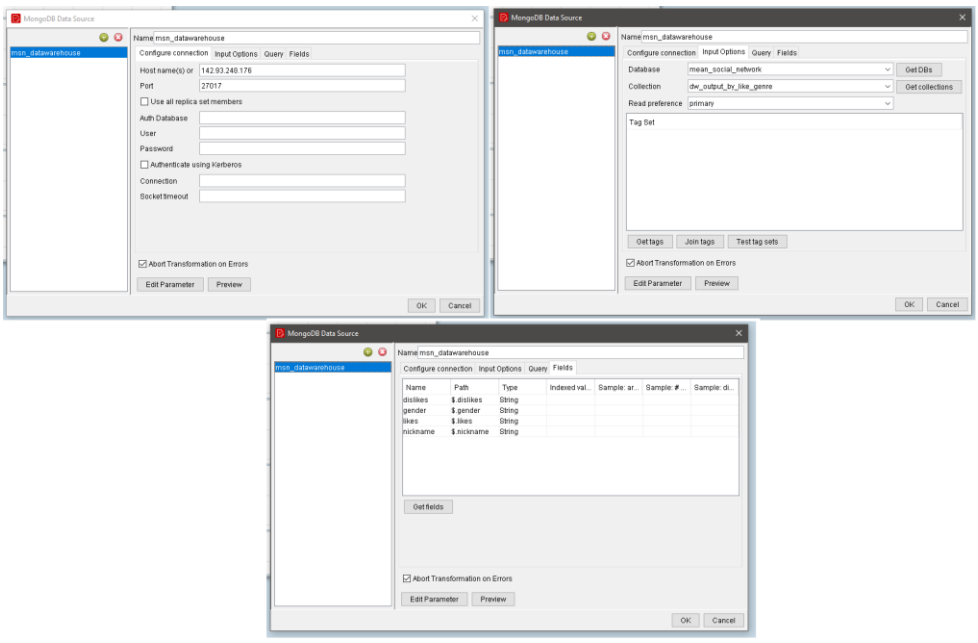


Figura 20. *Ejemplo de la configuración realizada para la conexión al DataWarehouse dentro de la base de datos de la aplicación MEAN SN.* ***Fuente Autor. Desde diseñador de reportes de la herramienta Pentaho.***

Una vez obtenida la fuente de los datos para generar reportes fue necesario configurar la plantilla básica que quedará como modelo para la presentación de la información obtenida a manera de gráficos estadísticos que puedan ser analizados.

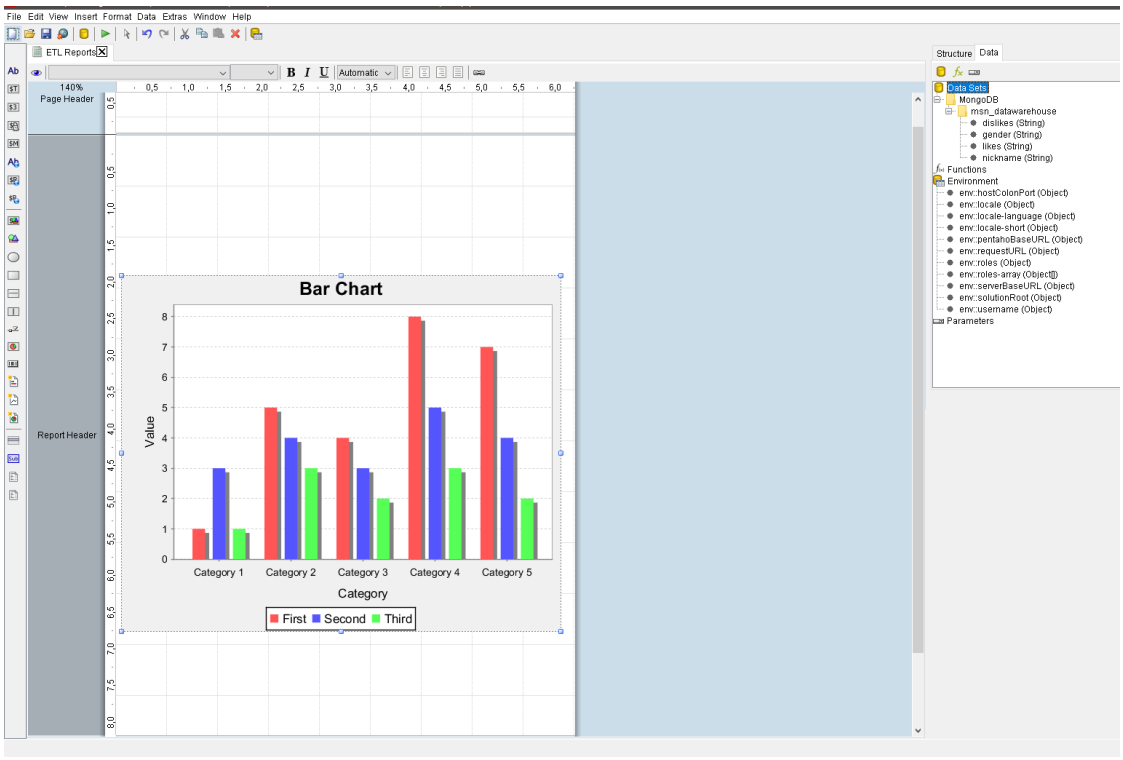


Figura 21. *Ejemplo de pantalla de diseño con un gráfico predeterminado.* ***Fuente Autor. Desde diseñador de reportes de la herramienta Pentaho***

Finalmente se generaron varios reportes en base a la información del DataWarehouse y diferentes opciones para filtrar la información deseada dependiendo de lo que se pretenda analizar.

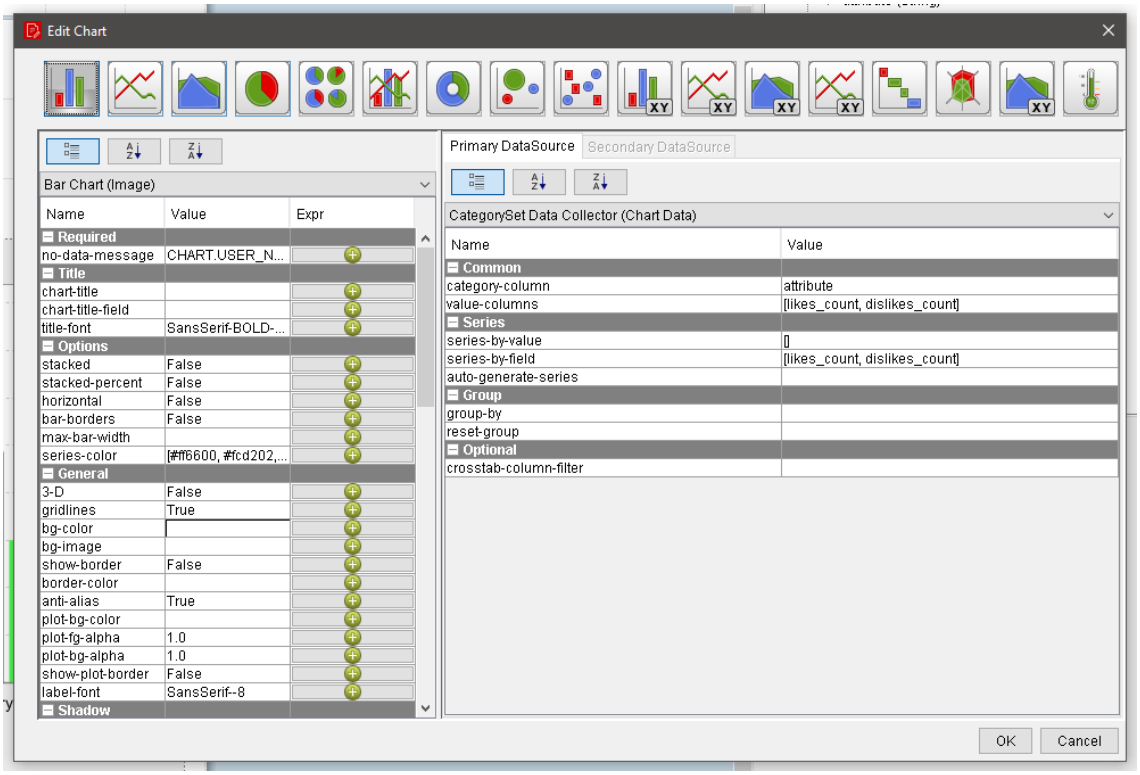


Figura 22. *Ejemplo de la configuración necesaria para generar un reporte de los Gustos y Preferencias de los uaurios de la aplicación.* ***Fuente Autor. Desde diseñador de reportes de la herramienta Pentaho***

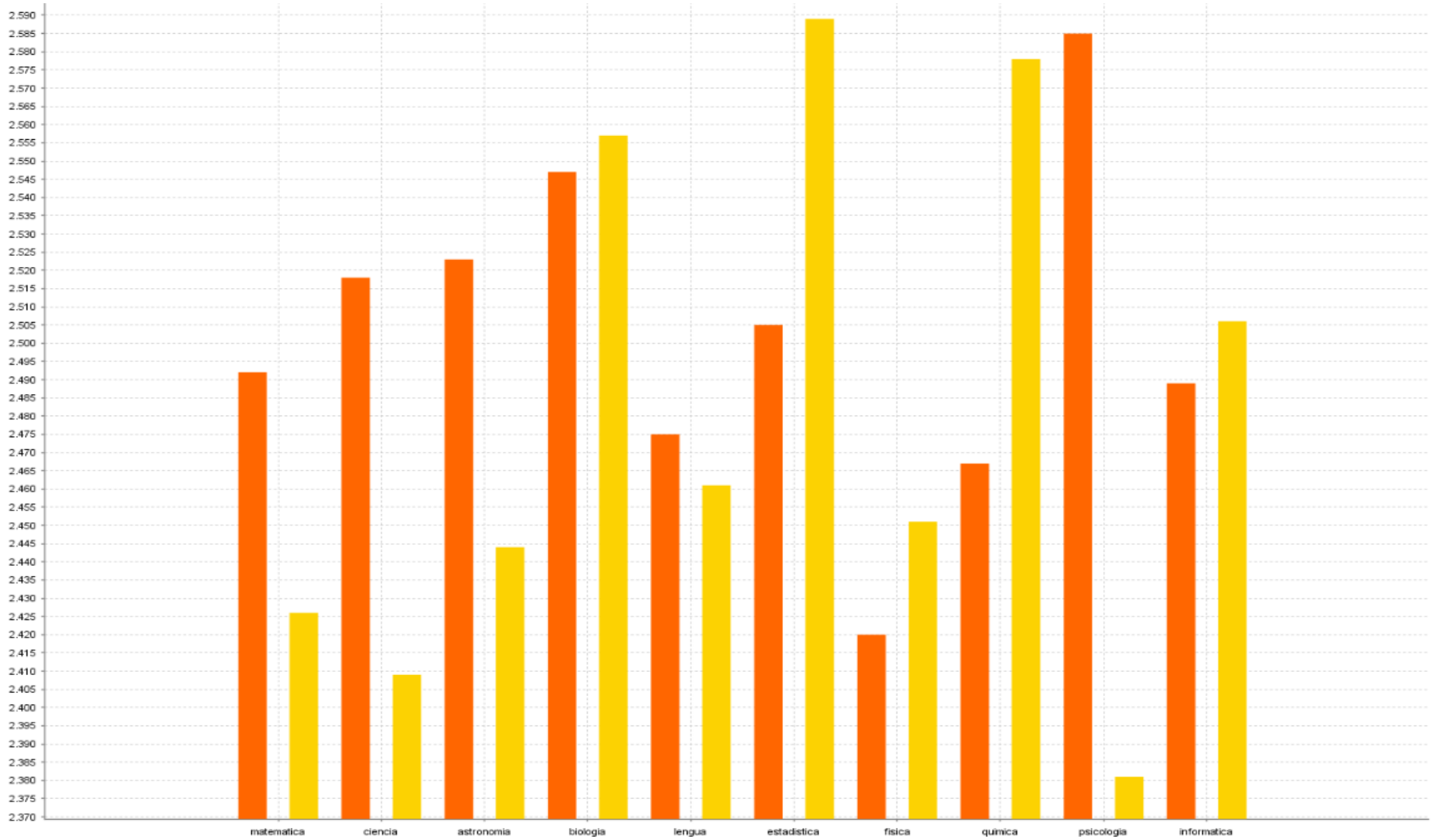


Figura 23. *Ejemplo de gráfico con estadísticas de varios gustos de usuarios.* ***Fuente Autor. Desde diseñador de reportes de la herramienta Pentaho.***

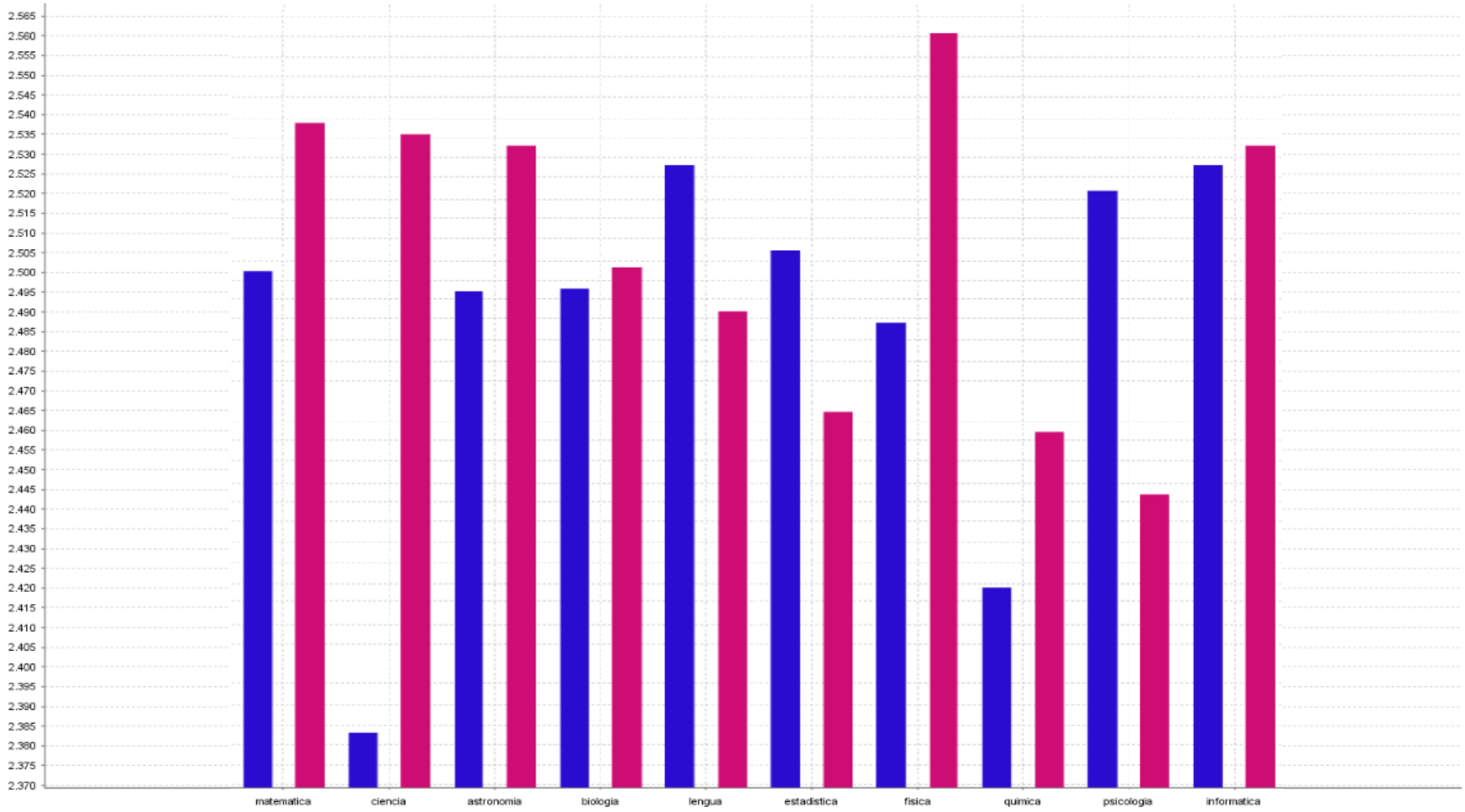


Figura 24. *Ejemplo de gráfico con estadísticas de varios gustos clasificados por género de usuarios.* ***Fuente Autor. Desde diseñador de reportes de la herramienta Pentaho.***

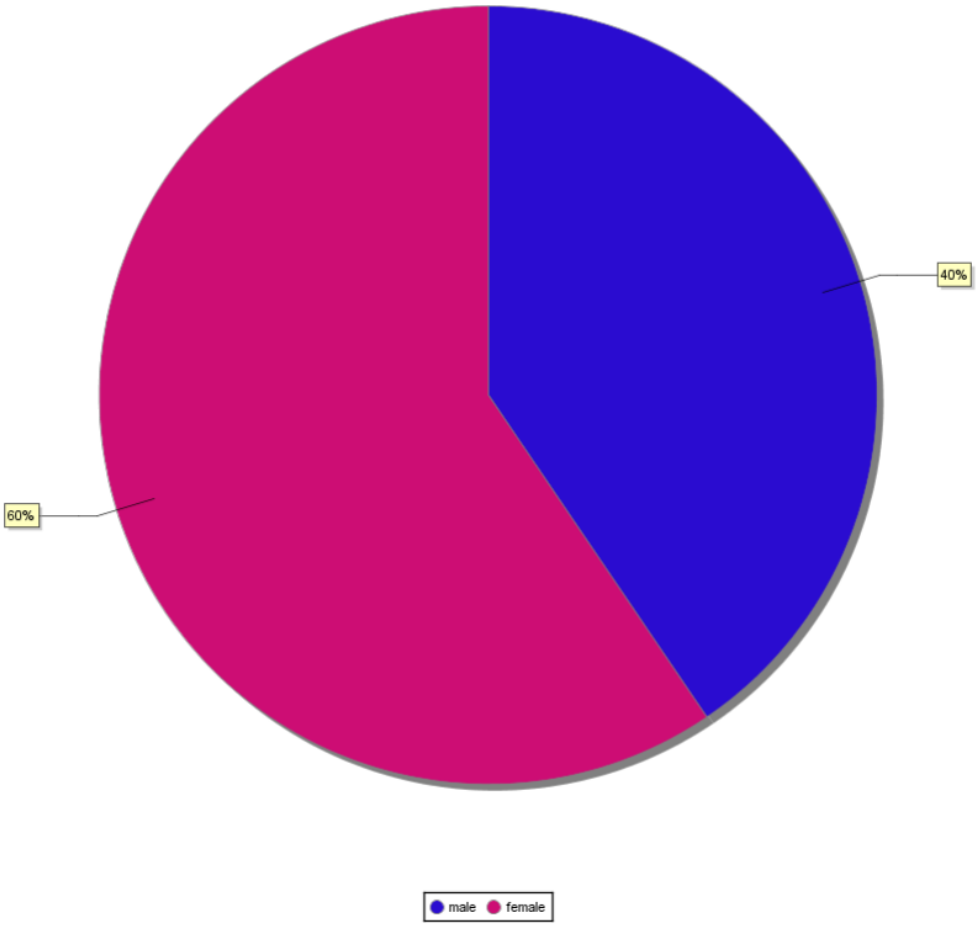


Figura 25. *Ejemplo de gráfico con estadísticas de usuarios clasificados por género.* ***Fuente Autor. Desde diseñador de reportes de la herramienta Pentaho.***

Los gráficos anteriores correspondientes a las ***Figuras 23, Figura 24*** **y** ***Figura 25*** muestan ejemplos de los reportes que se pueden obtener a partir del DataWarehouse generado anteriormente:

- La ***Figura 23*** muestra, por ejemplo, un reporte de la cantidad de usuarios a los cuales les gusta, o no, un determinado tema, el eje y representa la cantidad de usuarios, mientras que las barras naranjas y amarillas sobre el eje x, representan respectivamente si a los usuarios les gusta o no les gusta dicho tema.
- De manera similar la ***Figura 24*** muestra información un poco más "filtrada", ya que corresponde a un reporte de la cantidad de usuarios, clasificados por género, a los cuales les gusta un determinado tema, el eje y representa la cantidad de usuarios a los cuales les gusta un tema, mientras que las barras azules y fucsias sobre el eje x, representan respectivamente el genero del usuario.

- Finalmente, la ***Figura 25*** muestra un reporte general del porcentaje de usuarios clasificados por género.

En base a los reportes generados es posible obtener mucha información de utilidad sobre los usuarios que usan la aplicación, sobre todo estadísticas de gustos y preferencias con las cuales es posible, por ejemplo, filtrar un determinado grupo de usuarios que pueda llegar a ser objeto de estudio de algún tipo de mercado o investigación específica.

Así como se llevó a cabo los análisis y reportes descritos anteriormente es posible obtener muchos mas datos, aplicando los filtros adecuados y generando la información deseada el Business Intelligence como parte del proceso Big Data puede convertirse en una muy útil herramienta con todo tipo de aplicaciones.

3.4. EJECUCIÓN DE PRUEBAS

Para llevar a cabo de cada una de las pruebas se prepararon diferentes scripts que fueron ejecutados por la herramienta JMeter mencionada anteriormente, además de un ambiente de simulación de usuarios concurrentes en la aplicación online Blazemeter para las pruebas de Carga y Estrés.

Una vez obtenidos los resultados requeridos, estos serán analizados con un enfoque de caja negra y dentro del marco de referencia correspondiente a las métricas relacionadas al Comportamiento Temporal y Cumplimiento de la Eficiencia del Software, pertenecientes a la característica de Eficiencia dentro del margen de la norma ISO/IEC 9126 para la Calidad del Software.

3.4.1. PRUEBAS DE CARGA Y ESTRÉS

Tanto la prueba de carga como la prueba de estrés, realizadas ambas sobre la base de datos, van estrechamente relacionadas debido principalmente a su forma de ejecución, por lo que es posible desarrollar un mismo Test Plan en JMeter que sea utilizado para obtener resultados en ambas pruebas. Los scripts fueron desarrollados para simular

diferentes operaciones que serían comunes en el uso de una aplicación de este tipo, para dichos fines cuentan con las siguientes características:

- *Consulta Simple:* Este script pretende simular la consulta básica correspondiente a la obtención de la cantidad total de registros dentro de cualquier colección en la base de datos, siendo esta una de las operaciones más simples pero necesarias en casi cualquier aplicación web.
- *Operaciones de Inserción:* Los scripts siguientes corresponden a las operaciones de inserción en las colecciones Publications, Follows y Messages de la base de datos, pretendiendo simular la carga que tendría la base de datos en este tipo de situaciones y el "estrés" al cual estaría sometida de forma común por una gran cantidad de usuarios.

Dejando de lado los scripts mencionados, el Test Plan desarrollado también contiene indicadores de respuesta propios de la aplicación, conocidos como "Response Assertion", los cuales mostraran la respuesta obtenida a la petición realizada, así como el estado de la misma en caso de error.

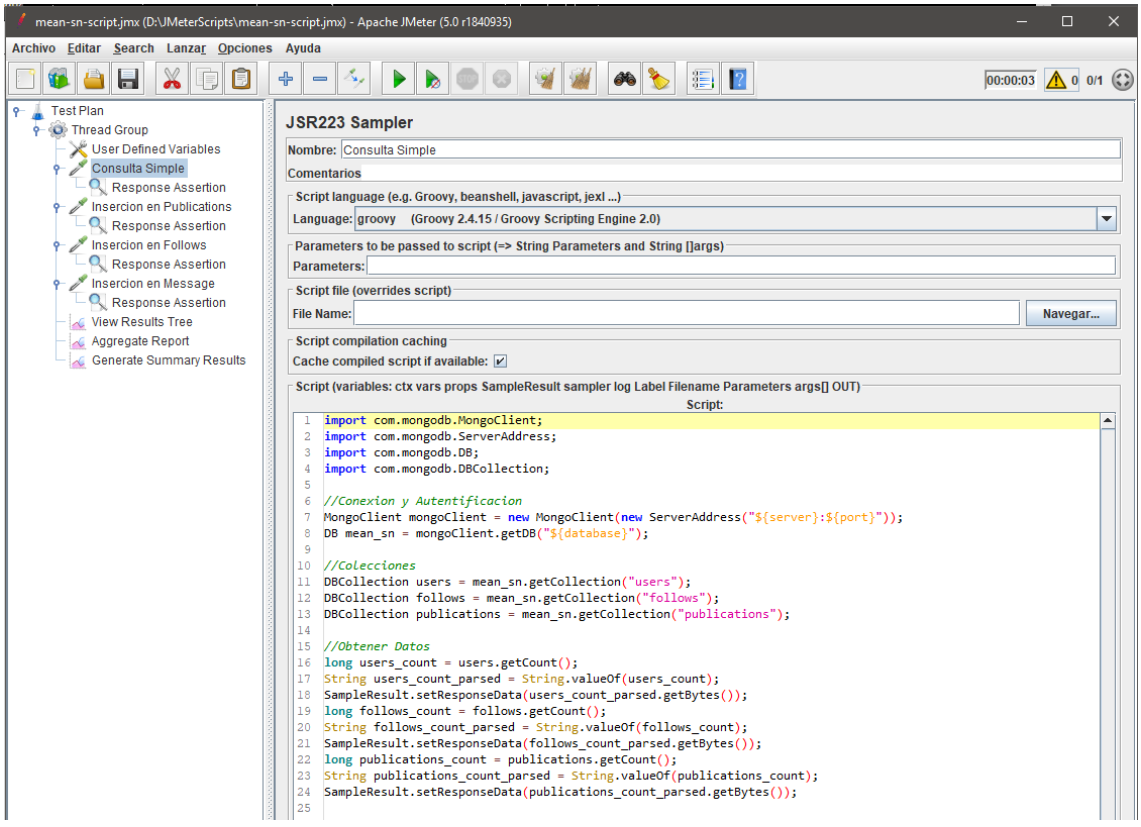


Figura 26. *Interfaz Gráfica JMeter. Pantalla con script de una Consulta Simple hacia la base de datos.* ***Fuente Autor. Ver ANEXO 2.***

Para revisar el status de cada prueba según se vaya ejecutando se agregó también un árbol general de resultados que muestra cada una de las peticiones ejecutadas y la respuesta devuelta por parte de la base de datos.

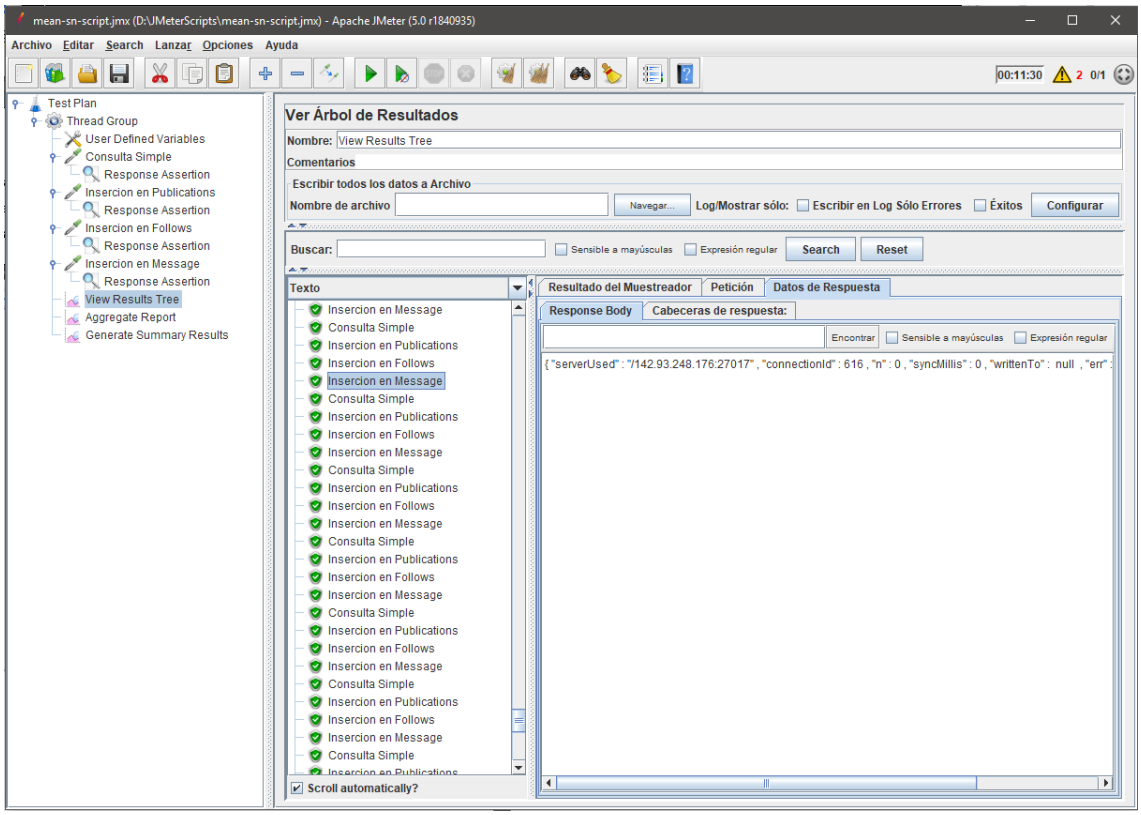


Figura 27. *Interfaz Gráfica JMeter. Pantalla con resultados de ejecución de todos los scripts.* ***Fuente Autor.***

Finalmente para la visualiación de los resultados se generó una tabla que mostrará las estadisticas principales sobre las cuales se llevará a cabo el análisis correspondiente, desde el tiempo de respuesta promedio obtenido por el total de peticiones realizadas, hasta el porcentaje de rendimiento calculado por la aplicación.

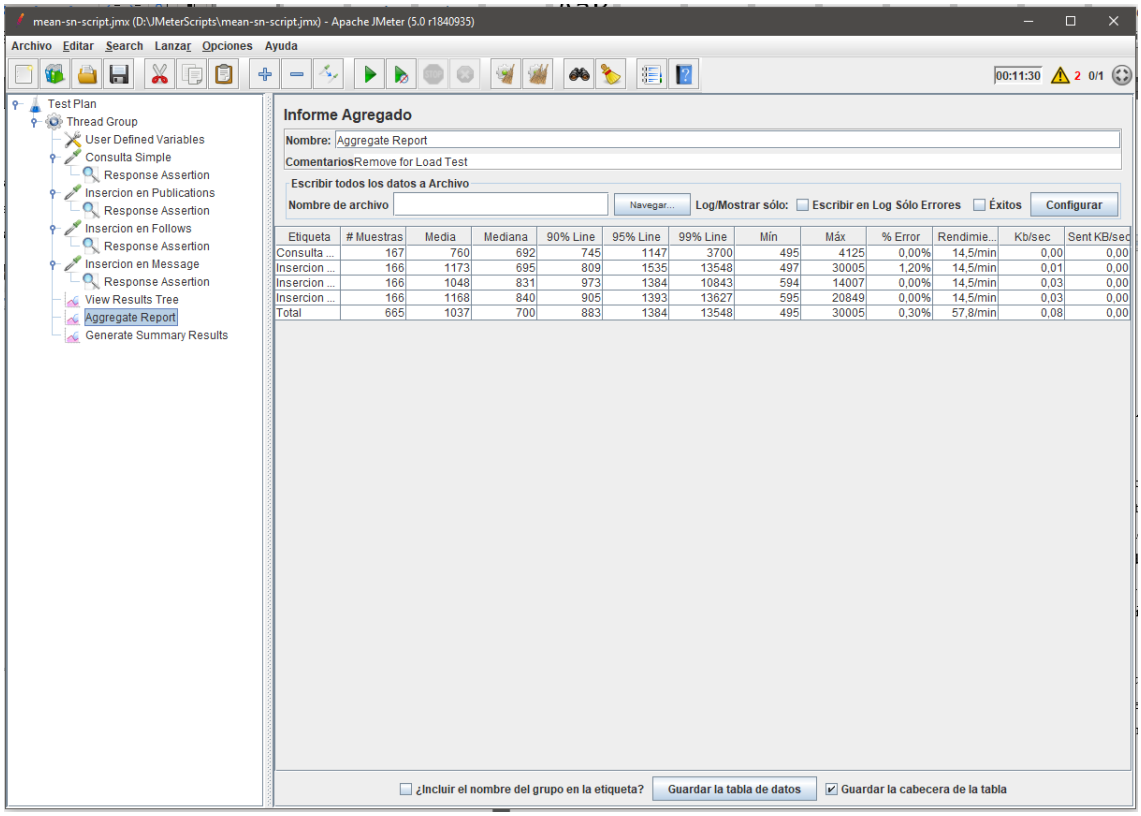


Figura 28. *Interfaz Gráfica JMeter. Pantalla tabla de resumen de resultados obtenidos.* ***Fuente Autor.***

Una vez preparado el respectivo plan de pruebas, fue subido a la herramienta online Blazemeter, además de esto se realizó la configuración necesaria para generar una gran cantidad de peticiones de usuarios concurrentes y obtener los respectivos resultados, dada la configuración máxima de la herramienta, esto corresponde a 5000 usuarios concurrentes, cada uno de los cuales realizará varias veces por segundo lo indicado en los scripts de Consulta Simple, e Inserciones en las respectivas colecciones de las bases de datos.

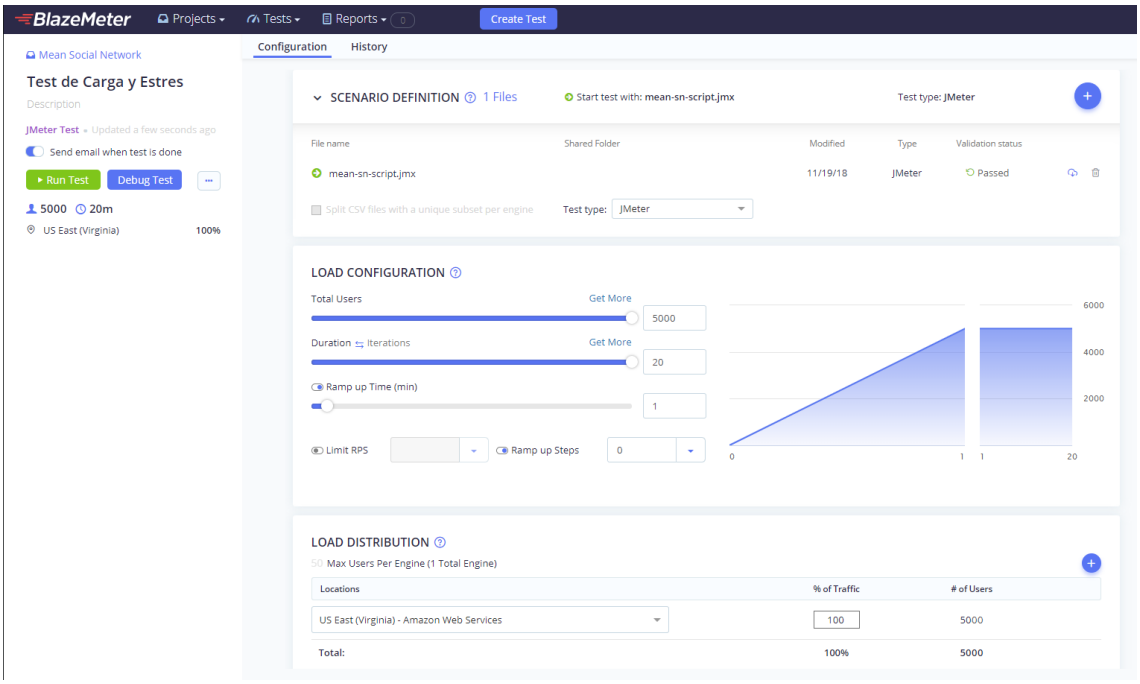


Figura 29. *Interfaz online de la herramienta Blazemeter. Pantalla de configuración de requisitos para la ejecución de las pruebas.* ***Fuente Autor.***

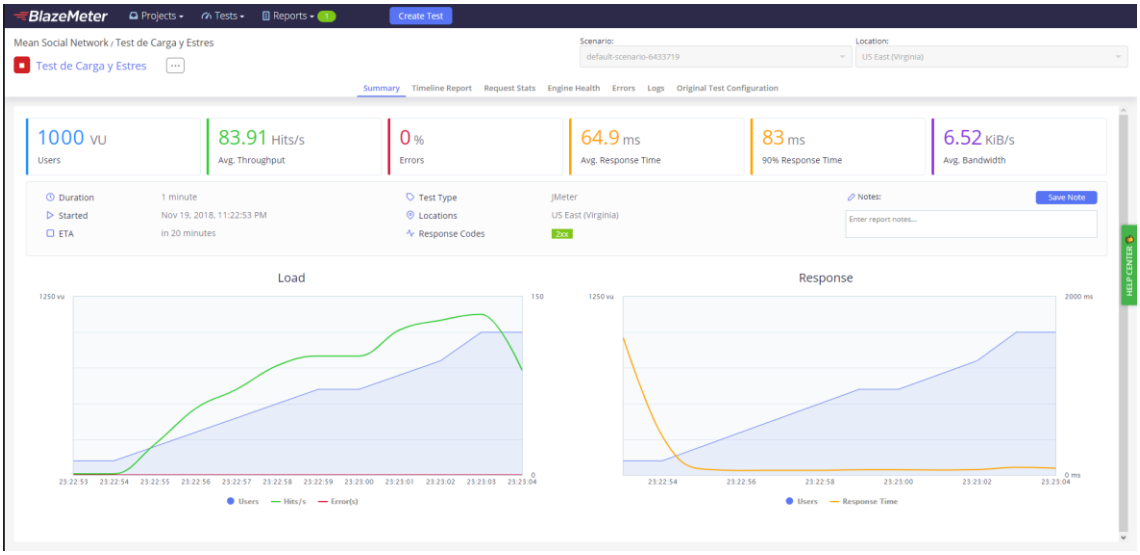


Figura 30. *Interfaz online de la herramienta Blazemeter. Ejemplo de pantalla durante la ejecución de las pruebas.* ***Fuente Autor.***

3.4.2. PRUEBA DE TIEMPO DE RESPUESTA

A diferencia de las pruebas anteriormente descritas, la prueba de tiempo de respuesta involucra no únicamente el tiempo que se demora la base de datos en responder a una operación solicitada, sino también las peticiones que realizan los componentes del propio stack MEAN entre sí, además de la adecuada respuesta que llegue por parte del mismo, permitiendo de esta manera tener una visión más completa del funcionamiento de la aplicación, y no únicamente someter a carga la base de datos, si no también el resto de componentes.

Una vez claro lo anterior, se preparó un script en JMeter con la navegación típica que podría presentar un usuario común al momento de utilizar la aplicación, tomando en consideración las siguientes acciones:

- Acceso a la aplicación
- Login de un usuario
- Publicaciones
- Revisión de mensajes
- Nueva Publicación
- Actualización de datos del usuario

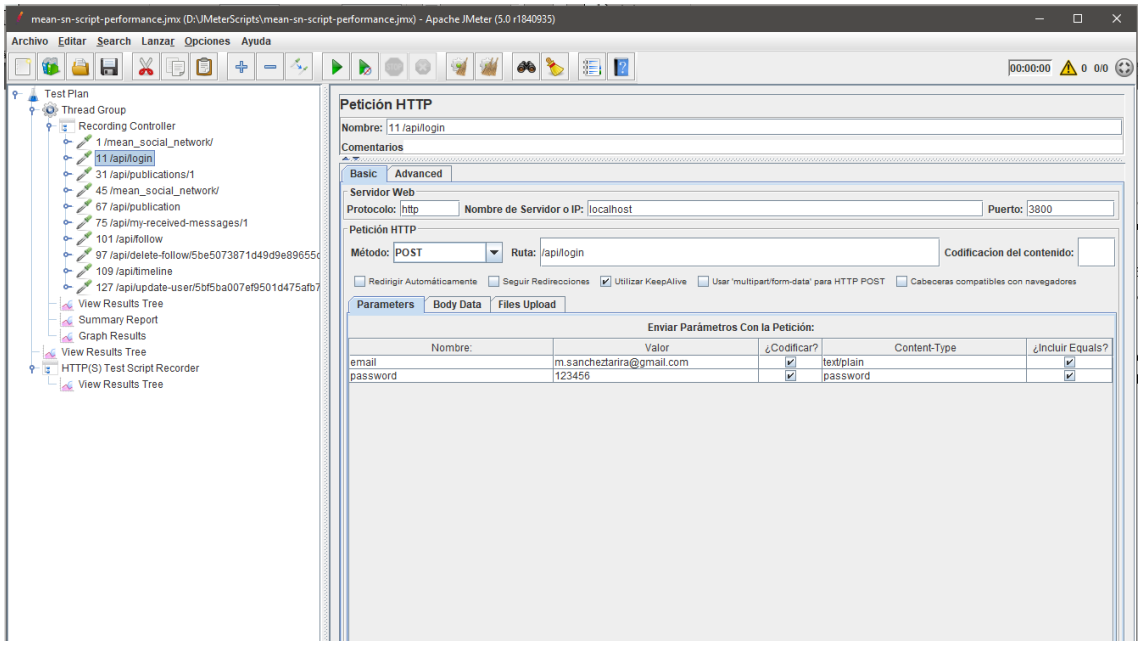


Figura 31. *Interfaz Gráfica JMeter. Pantalla con ejemplo de petición login a la aplicación MEAN SN.* ***Fuente Autor. Ver ANEXO 3***

Para revisar el status de cada petición según se vayan ejecutando se agregó también un árbol general de resultados que muestra cada una de las peticiones realizadas y la respuesta devuelta por parte de la aplicación.

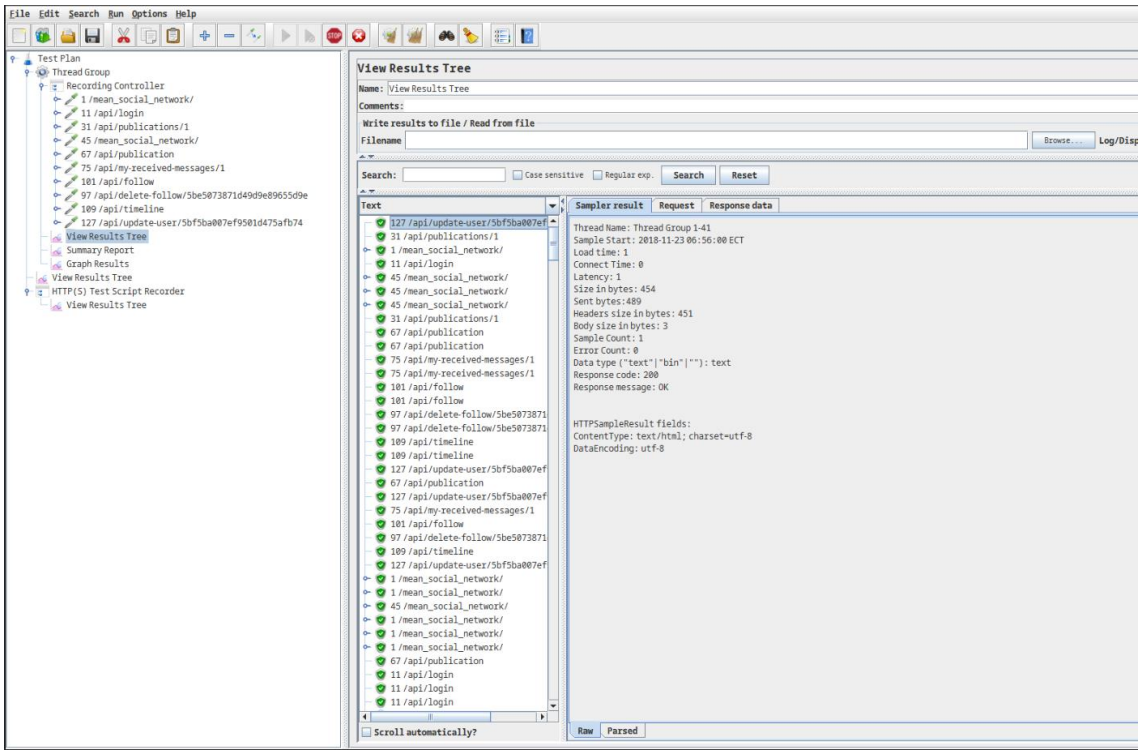


Figura 32. *Interfaz Gráfica JMeter. Pantalla con resultados de las peticiones realizadas.*
Fuente Autor.

Finalmente para la visualiación de los resultados se generó una tabla y un gráfico explicativo sobre los que se puede visualizar las estadisticas de tiempos de respuestas totales para poder llevar a cabo el análisis correspondiente.

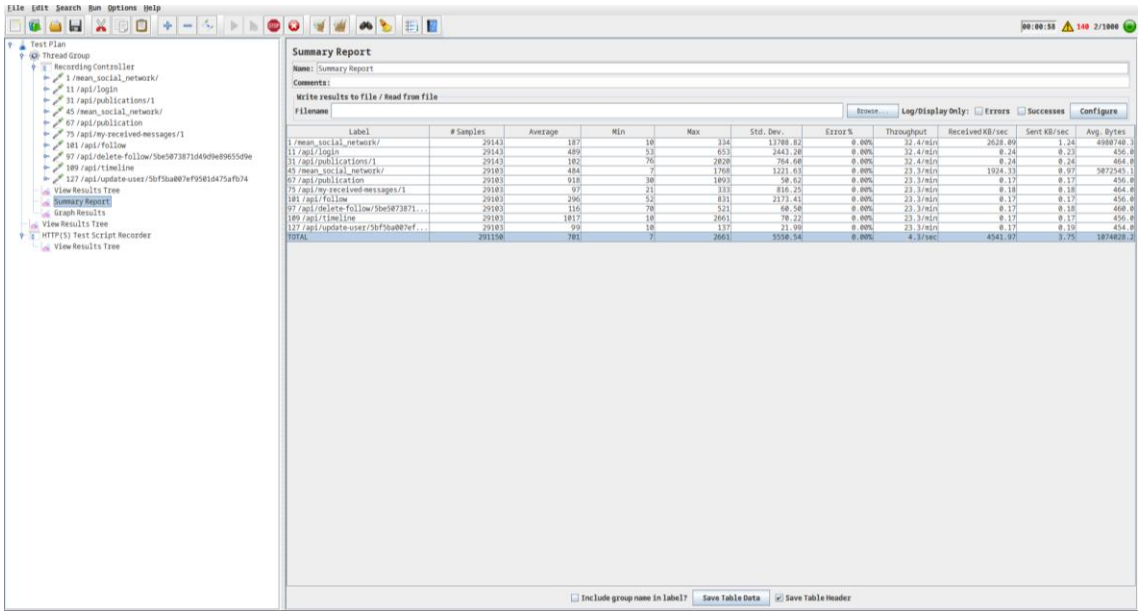


Figura 33. *Interfaz Gráfica JMeter. Pantalla tabla de resumen de resultados obtenidos.* ***Fuente Autor.***

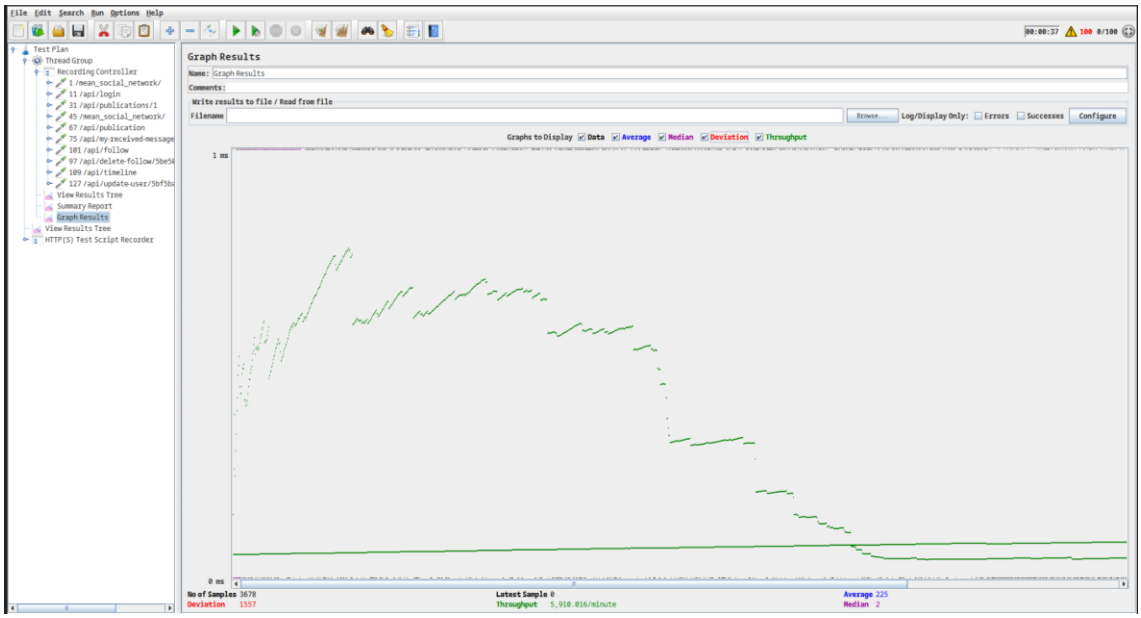


Figura 34. *Interfaz Gráfica JMeter. Pantalla del gráfico de resumen.* ***Fuente Autor.***

3.5. ANÁLISIS E INTERPRETACIÓN DE LOS RESULTADOS OBTENIDOS EN LAS PRUEBAS

El propósito de realizar la prueba de carga consitió en determinar hasta que punto la base de datos puede soportar una determinada cantidad simultánea de operaciones con usuarios concurrentes, sin provocar tiempos de respuesta por encima de lo considerado normal para el correcto funcionamiento de la aplicación.

Como referencia para evaluar los resultados obtenidos durante las pruebas, serán tomados en cuenta los datos de PageSpeed Insights, página desarrollada por Google, la cual clasifica el tiempo de respuesta promedio para bases de datos en aplicaciones web en condiciones normales de la siguiente manera:

	Rápido (ms)	**Normal (ms)**	**Lento (ms)**
Tiempo de Respuesta	[0, 200]	(200, 500]	Sobre los 500

Tabla 5. *Tiempos de respuesta promedio.* ***Fuente Google PageSpeed Insights.***

La ***Tabla 5*** muestra los tiempos de respuesta adecuados para el correcto funcionamiento de una base de datos promedio bajo condiciones normales según Google PageSpeed Insights.

Por otro lado, la prueba de estrés consistió, de manera similar, en provocar una sobrecarga en las operaciones realizadas sobre la base de datos para determinar un punto en el cuál se llegarán a producir errores en las respuestas de las diferentes consultas y operaciones consideradas básicas durante el funcionamiento común de la aplicación, teniendo en cuenta su comportamiento temporal y una cantidad de usuarios concurrentes.

Dicho lo anterior, tanto la prueba de carga, como la prueba de estrés fueron realizadas bajo las siguientes condiciones:

- 20 min de Duración Máxima de la Prueba.
- Máximo 5000 Usuarios Concurrentes.
- 28 usuarios/s como tasa promedio de aumento hasta el pico (5000 usuarios).
- Una vez alcanzado el pico de usuarios se mantendrá la configuración.

Finalmente, el obtjetivo de la prueba de Tiempo de Respuesta, es determinar bajo que condiciones la aplicación completa y sus componentes responderán de manera inadecuada teniendo en cuenta su comportamiento netamente temporal. Para esta prueba se dispusieron las siguientes condiciones:

- 40 min de Duración Máxima de la Prueba
- Máximo 5000 Usuarios Concurrentes
- Cada petición http es realizada después de la anterior para simular el comportamiento que tendría un usuario promedio al momento de utilizar la aplicación.
- El tiempo de respuesta obtenido será el correspondiente al FCP (First Contentful Paint), definido como el tiempo que demora el navegador en cargar el primer byte recibido por parte de los controladores de la aplicación.

Según la antes mencionada aplicación web, PageSpeed Insights, la velocidad de respuesta adecuada para una experiencia óptima, y tomando como punto de referencia el FCP es la siguiente:

	Rápido (ms)	**Normal (ms)**	**Lento (ms)**
Tiempo de Respuesta del FCP	[0, 1000]	(1000, 2500]	Sobre los 2500

Tabla 6. *Tiempos de respuesta promedio.* ***Fuente Google PageSpeed Insights.***

La ***Tabla 6*** muestra los tiempos de respuesta adecuados para el correcto funcionamiento de una aplicación web promedio bajo condiciones normales según Google PageSpeed Insights.

3.5.1. ANÁLISIS DE RESULTADOS DE LA PRUEBA DE CARGA

Con las condiciones descritas anteriormente se consiguieron los siguientes resultados:

- 18 min de Duración Total de la Prueba.

- 3 min hasta llegar al Pico de Usuarios.
- El pico máximo de Tiempo de Respuesta obtenido ocurrió bajo las siguientes condiciones:
    - Se dio entre el minuto 16 y minuto 17 desde el Inicio de la Prueba,
    - Ocurrió durante la ejecución de 5000 usuarios concurrentes.
    - Cada usuario realizaba un promedio de 242 operaciones por segundo.
    - La operación sobre la que ocurrió fue la iniciada por el Script correspondiente a Insersión en Messages.
- Durante toda la prueba se realizaron un total de 28422 operaciones.

Script	#Muestras	Respuesta Promedio (ms)	Respuesta Min. (ms)	Respuesta Max. (ms)
Consulta Simple	7122	256.37	8	388.76
Inserción en Follows	7102	303.8	19	511.69
Inserción en Messages	7085	314.6	25	695.3
Inserción en Publications	7113	377.43	20	455.6
TOTAL	**28422**	**313.05**	**8**	**695.3**

Tabla 7. *Resultado de la Prueba de Carga.* ***Fuente Autor.***

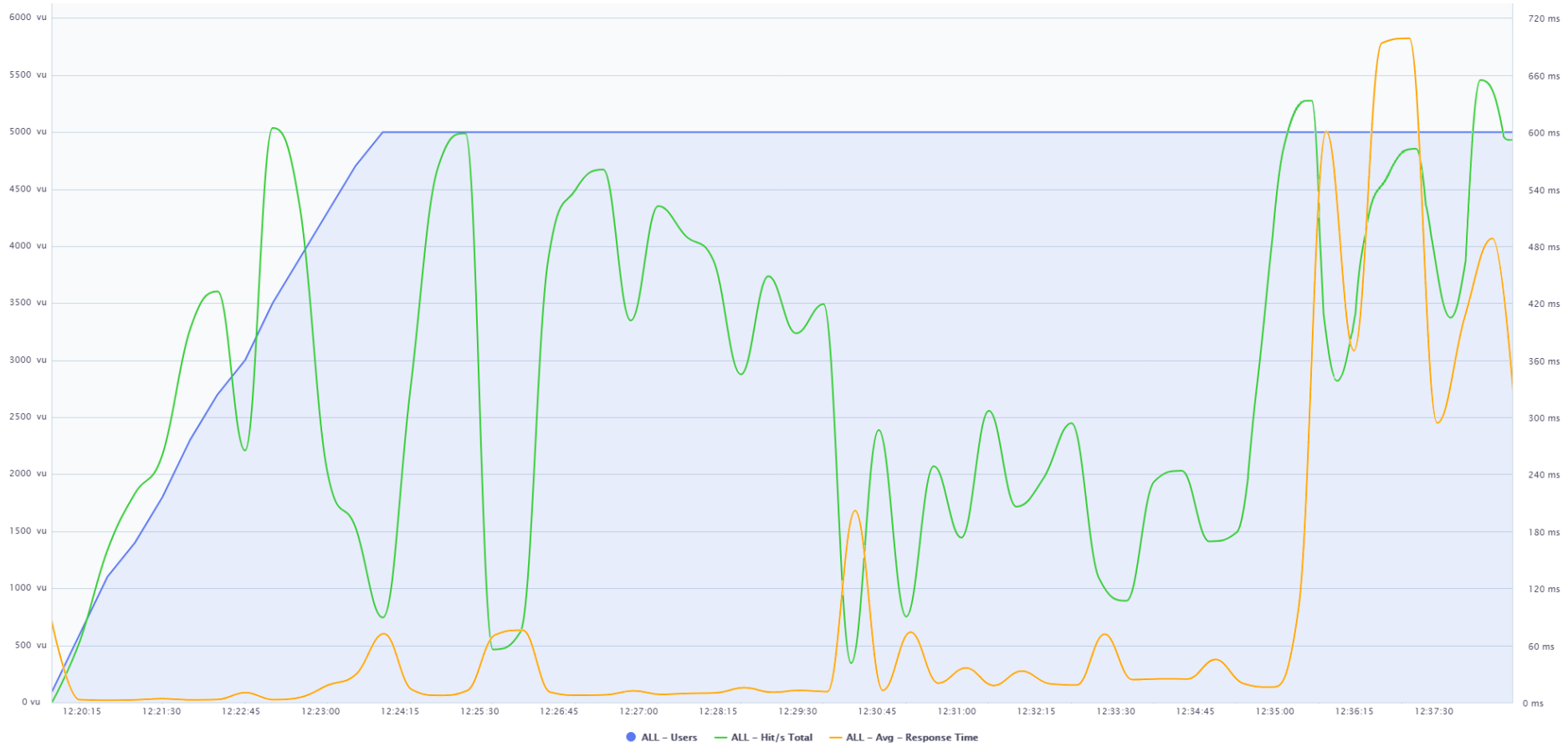


Figura 35. *Interpretación Gráfica de los resultados obtenidos con la Prueba de Carga.* ***Fuente Autor.***

Cantidad de Usuarios Concurrentes — Operaciones por Segundo — Tiempo de Respuesta (ms)

La ***Tabla 7*** mustra los resultados obtenidos en las pruebas de carga ejecutadas, por cada script realizado mediante JMeter se puede observar los valores correspondientes al número de muestras ejecutadas, es decir las veces que se ejecutó ese script, y además los valores de respuesta promedio, respuesta mínima y respuesta máxima recibidos por parte de la base de datos en milisegundos. Por otro lado, en el gráfico anterior correspondiente a la ***Figura 35***, se puede apreciar la siguiente información:

- La línea azul muestra el aumento progresivo de la cantidad de usuarios concurrentes durante el transcurso de la prueba, empezando en cero hasta llegar al pico de 5000 usuarios y manteniéndose constante hasta el final.
- La línea verde corresponde a la representación temporal de la cantidad promedio de veces que fueron ejecutados de manera simultánea cada uno de los scripts durante el transcurso de la prueba.
- La línea amarilla muestra el tiempo promedio de respuesta, en milisegundos, a las peticiones ejecutadas por cada uno de los scripts en el tiempo que duró la prueba.

Observando los datos desde el punto de vista del Comportamiento Temporal en la Eficiencia según la norma ISO/IEC 9126 se obtiene el siguiente análisis:

SCRIPT	Rápido (X < 200 ms)	Normal (200 < X < 500 ms)	Lento (X > 500 ms)
Consulta Simple	85 %	15 %	-
Inserción en Follows	70 %	23 %	7 %
Inserción en Messages	77 %	20 %	3 %
Inserción en Publications	73 %	17 %	-
TOTAL	**76 %**	**19 %**	**5 %**

Tabla 8. *Porcentajes de Tiempo de Respuesta.* ***Fuente Autor.***

La ***Tabla 8*** muestra el análisis en porcentajes de tiempo de respuesta obtenidos, de cada uno de los scripts ejecutados durante la prueba de carga según los parámetros establecidos por Google PageSpeed Insights, y mostrados en la ***Tabla 5***. Teniendo en cuenta los resultados es posible expresar que la base de datos de la aplicación presenta en su mayoría

tiempos de respuesta óptimos entre Rápido y Normal, cabe recalcar también que es necesario tener muy en cuenta que existen determinados momentos en los que se presentan resultados no deseados debido a la sobrecarga de peticiones, esto puede provocar errores en las respuestas, o en el peor de los casos un fallo total y caida en la base de datos.

3.5.2. ANÁLISIS DE RESULTADOS DE LA PRUEBA DE ESTRÉS

Con las condiciones descritas anteriormente se consiguieron los siguientes resultados:

- 18 min de Duración Total de la Prueba.
- 3 min hasta llegar al Pico de Usuarios.
- Primer Error obtenido bajo las siguientes condiciones:
    - 15 min desde el Incio de la Prueba.
    - Error en el Script de Consulta simple.
    - 5000 usuarios Concurrentes.
    - En ese momento cada usuario realizaba aproximadamente 250 operaciones por segundo.
- Durante toda la prueba se realizaron un total de 28422 operaciones.
- Durante toda la prueba se detectó un total de 179 errores en operaciones en ejecución.

Observando los datos desde el punto de vista del Cumplimiento de la Eficiencia del Software en la Eficiencia según la norma ISO/IEC 9126 se obtiene que, en total de 28422 peticiones realizadas a la base de datos, hubo 179 respuestas fallidas, correspondiente a un 0.63% del total:

Script	#Muestras	Porcentaje de Error (%)	Conteo de Errores
Consulta Simple	7122	0.58	41
Inserción en Follows	7102	0.66	47
Inserción en Messages	7085	0.63	45

Inserción en Publications	7113	0.65	46
TOTAL	**28422**	**0.63**	**179**

Tabla 9. *Resultado de la Prueba de Estrés.* ***Fuente Autor.***

La ***Tabla 9*** mustra los resultados obtenidos en las pruebas de estrés ejecutadas, por cada script realizado mediante JMeter se puede observar los valores correspondientes al número de muestras ejecutadas, es decir las veces que se ejecutó ese script, y además la cantidad de errores que se presentaron durante las pruebas y el porcentaje que representan en base a la cantidad de veces que fue ejecutado dicho script.



Figura 36. *Interpretación Gráfica de los resultados obtenidos con la Prueba de Estrés.* ***Fuente Autor.***

Cantidad de Usuarios Concurrentes — Operaciones por Segundo — Errores Encontrados

En el gráfico anterior correspondiente a la ***Figura 36***, se puede apreciar la siguiente información:

- La línea azul muestra el aumento progresivo de la cantidad de usuarios concurrentes durante el transcurso de la prueba, empezando en cero hasta llegar al pico de 5000 usuarios y manteniéndose constante hasta el final.
- La línea verde corresponde a la representación temporal de la cantidad promedio de veces que fueron ejecutados de manera simultánea cada uno de los scripts durante el transcurso de la prueba.
- La línea roja muestra la cantidad de errores en función del tiempo que hubo durante el transcurso de la prueba, recalcando que se mantenía en cero hasta el minuto 15.

Teniendo en cuenta los resultados anteriores se procede a realizar el cálculo correspondiente para determinar la confiabilidad de la aplicación en base a la eficiencia y según la norma ISO/IEC 9126.

- X = 1 – A/N
- A = Número de veces que ocurrió un evento de error en la respuesta esperada.
- N = Número de casos de prueba realizados para generar un error.

Donde:

- X = 1 – (179/28422)
- X = 0.9937

Dado este resultado es posible afirmar que la aplicación es un 99.37 % confiable, bajo las condiciones descritas anteriormente.

3.5.3. ANÁLISIS DE RESULTADOS DE LA PRUEBA DE TIEMPO DE RESPUESTA

Realizando la respectiva prueba con las condiciones descritas anteriormente se obtuvieron los siguientes resultados:

Script	#Muestras	Respuesta Promedio (ms)	Respuesta Min. (ms)	Respuesta Max. (ms)
mean_social_network/	29143	187	10	334
api/login/	29143	489	53	653
api/publications/1	29143	102	76	2020
mean_social_network/	29103	484	7	1768
api/publication	29103	918	30	1093
api/my-received-messages/1	29103	97	21	333
api/follow	29103	296	52	831
api/delete-follow/ID	29103	116	70	521
api/timeline	29103	1017	10	2661
api/update-user/USER_ID	29103	99	10	137
TOTAL	**291150**	**701**	**7**	**2661**

Tabla 10. *Resultado de la Prueba de Tiempo de Respuesta.* ***Fuente Autor.***

La ***Tabla 10*** mustra los resultados obtenidos en las pruebas de tiempo de respuesta ejecutadas, por cada script realizado mediante JMeter se puede observar los valores correspondientes al número de muestras ejecutadas, es decir las veces que se ejecutó ese script, y además los valores de respuesta promedio, respuesta mínima y respuesta máxima recibidos por parte de la aplicación en milisegundos.

Observando los datos desde el punto de vista del Comportamiento Temporal en la Eficiencia según la norma ISO/IEC 9126 se obtiene el siguiente análisis:

SCRIPT	Rápido (X < 1000 ms)	Normal (1000 < X < 2500 ms)	Lento (X > 2500 ms)
mean_social_network/	100 %	-	-
api/login/	100 %	-	-
api/publications/1	93 %	7 %	-
mean_social_network/	98 %	2 %	-
api/publication	90 %	10 %	-
api/my-received-messages/1	100 %	-	-
api/follow	100 %	-	-
api/delete-follow/ID	100 %	-	-
api/timeline	87 %	9 %	1 %
api/update-user/USER_ID	100 %	-	-
TOTAL	**97 %**	**2 %**	**1 %**

Tabla 11. *Porcentaje de Tiempo de Respuesta.* ***Fuente Autor.***

La ***Tabla 11*** muestra el análisis en porcentajes de tiempo de respuesta obtenidos, de cada uno de los scripts ejecutados durante la prueba de tiempo de respuesta según los parámetros establecidos por Google PageSpeed Insights, y mostrados en la ***Tabla 6***.

Teniendo en cuenta los resultados obtenidos es posible expresar que la aplicación presenta en su mayoría tiempos de respuesta óptimos entre Rápido y Normal, cabe

recalcar también que es necesario tener muy en cuenta que existen determinados momentos en los que se presentan resultados no deseados debido a la cantidad de registros existentes y la complejidad de la búsqueda dentro de la base de datos para obtener la respuesta deseada, de no tomar las debidas precauciones se pueden presentar caídas en el servidor de la aplicación.

CAPÍTULO IV

DISCUSIÓN

VENTAJAS Y DESVENTAJAS DEL STACK DE DESARROLLO MEAN COMO HERRAMIENTA PARA BIG DATA

Ciertamente, el stack de desarrollo MEAN, presenta varias ventajas frente a otras herramientas dedicadas al desarrollo web, partiendo del hecho que el stack completo es puesto en marcha con un mismo lenguaje de programación, facilitando la comunicación entre sus componentes, y creando un ambiente con una sinergia capaz de procesar datos sin mayor inconveniente. Dentro de este contexto, existen autores que respaldan el uso del stack MEAN para aplicaciones escalables, así como aquellas que involucren un cierto nivel de análisis y procesamiento de información.

En su investigación "Web Application Development by Using MEAN Stack", los autores Kadam, Chaudhari, Patil, & Chavhan (2017) por ejemplo, realizan un análisis de las definiciones de cada componente de MEAN, y su estructura básica, además explican sus beneficios al momento de construir aplicaciones web escalables, para finalmente concluir con la premisa de que dicho stack, es una tecnología relativamente nueva, que supone la superación de ciertos inconvenientes en comparación al típico stack de desarrollo web LAMP, y afirmando que se trata de una herramienta altamente escalable y eficiente.

De forma similar, Duy Khue, Thanh Binh, Chang, Kim, & Chung (2017), en su artículo científico "Design and Implementation of MEAN Stack-based Scalable Real-time Digital Signage System", describen como usaron el stack MEAN para la construcción de un Sistema de Señalización Digital o DSS, los autores realizan una pequeña comparación de MEAN con otras herramientas, caracterizándolo como más simple, flexible, rápido y adecuado para el desarrollo web, todo esto a manera de justificación del porqué del uso del mismo, explicando que según las pruebas realizadas, debido a su ligereza era capaz de soportar el incremento de tráfico sobre la base de datos y el manejo y monitorización propio del sistema desarrollado en tiempo real.

Así como los ejemplos anteriores, existen muchos otros autores como, Karanjit (2016), German, Salmeron, Ha, & Henderson (2016), Jameel, Khanna, Rangappa, Patil, & Khalkar (2017), o Poulter, Johnston, & Cox (2015) que en sus investigaciones coinciden en analizar MEAN como una herramienta poderosa y de mucha consideración para el desarrollo web en general, así como caracterizarla de mejor manera en comparación a otras herramientas usadas comúnmente para la construcción de aplicaciones y el desarrollo web.

Entrando en el contexto del análisis Big Data, autores como Stajcer, Stajcer, & Orescanin (2016), en su investigación "Using MEAN stack for development of GUI in real-time big data architecture" exponen el stack MEAN desde un punto de vista centrado en el procesamiento de la información dentro de la aplicación desarrollada, los autores generan una propuesta para la creación de una aplicación con una interfaz gráfica de usuario para un sistema de detección de fraudes enfocado en el análisis de ficheros, usando para esto una estructura basada en tres capas, el artículo concluye con la caracterización de MEAN como una herramienta con el potencial suficiente para soportar el procesamiento que supone la arquitectura de una aplicación Big Data.

En base a lo anterior y a los resultados que arrojó la investigación descrita en el presente documento, es posible definir las siguientes ventajas acerca del stack MEAN como herramienta de desarrollo de aplicaciones web orientadas al Big Data:

- La estructura y el diseño de MEAN permiten crear aplicaciones de principio a fin con el mismo lenguaje de programación, tanto el desarrollo del backend, como del frontend y la base de datos se lo realiza mediante Javascript, característica que, como se ha mencionado varias veces en este documento, crea una sinergia y capacidad de comunicación muy potente entre sus componentes.
- El hecho de usar Javascript como único lenguaje tiene como resultado la creación de aplicaciones muy completas en cuanto a funcionalidad, con un peso sumamente ligero, por lo que la velocidad de comunicación entre el backend y el frontend se ve mejorada frente a otras herramientas, eso sin mencionar el relativo ahorro de recursos al momento de poner la aplicación en producción.
- Los frameworks AngularJS y ExpressJS usados para la construcción del frontend y backend respectivamente, presentan características multiparadigma propias de Javascript, pero dejando un poco de lado esto, ambos tienen una gran

compatibilidad con el paradigma Modelo – Vista – Controlador muy usado en el desarrollo web, mismo que permite llevar un muy buen control de cada uno de los componentes de la aplicación.

- La base de datos utilizada por el stack MEAN, MongoDB, incrementa más aun la velocidad de respuesta en cuanto al funcionamiento de la aplicación en general, no solo porque soporta operaciones embebidas con el lenguaje por excelencia del stack, Javascript, sino también gracias a la arquitectura documento – colección con la que trabaja, y el formato JSON, en el cual es guardada la información, que al igual que el resto de los componentes tiene un peso de almacenamiento reducido en comparación a otros tipos de bases de datos.
- Otro punto a favor de MongoDB es el conjunto de funcionalidades relacionadas al análisis de datos que presenta de forma nativa, incluyendo su propia versión del algoritmo MapReduce, funciones propias de agregación, consultas estructuradas para búsquedas de campos específicos, indexación, replicación, balanceo de carga, y una fuerte compatibilidad con Hadoop como herramienta de procesamiento.

A pesar de lo dicho en párrafos anteriores, existen también autores que contradicen el uso de uno o varios componentes del stack MEAN, González, González, & López (2014) por ejemplo, presentan un artículo titulado "JAVASCRIPT SEAN STACK: UNA MIRADA AL FUTURO DEL DESARROLLO WEB", en el cual refutan el uso de una base de datos NoSql para el desarrollo web, caracterizando MongoDB, como insuficiente y proponiendo el uso de bases de datos estructuradas como SQLite para obtener mejoras en el procesamiento. En este mismo sentido (Oliver, 2014), en su artículo "The 10 worst big data practices", publicado para el sitio web InfoWorld, promueve el uso de MongoDB como base de datos para almacenar grandes cantidades de información, pero a su vez reniega totalmente del uso de esta como herramienta para el procesamiento de datos, manteniendo la premisa de que este no es un sistema construido específicamente para llevar a cabo operaciones analíticas lo suficientemente complejas.

Dicho lo anterior, y tomando en cuenta la investigación realizada a lo largo de este documento, se plantean las siguientes desventajas en cuanto al uso del stack de desarrollo MEAN:

- Si se habla completamente del stack MEAN, es posible notar una curva de aprendizaje superior en cuanto a otras herramientas similares, además que la forma de programación de todos sus componentes no es lo suficientemente intuitiva en cuanto a aprendizaje.
- El mantenimiento de la base de datos construida con MongoDB y el resto de las componentes del stack puede ser un poco complejo y depende en gran medida de la experiencia que se tenga en este ámbito específico, dicha base de datos consta de herramientas muy poderosas para el desarrollo y análisis Big Data, pero se salen un poco del contexto debido a su complejidad de configuración y uso.
- Como acotación al punto anterior, una aplicación de tipo Big Data depende en un gran porcentaje del uso de sus datos, y la calidad de sus funciones de análisis y procesamiento, por lo que, si las operaciones que se necesiten realizar sobre la información son complejas, o difíciles de llevar a cabo, la aplicación desarrollada no tendrá el impacto y usabilidad necesarios.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

Dentro de las técnicas, software y algoritmos utilizados para el análisis de grandes cantidades de información usados en la tecnología Big Data, resaltan el algoritmo MapReduce como base de lo que ahora se conoce como procesamiento de información compleja, y los lenguajes de programación actuales y no tan actuales, que representan una mejora sustancial en la capacidad de análisis de datos, como son por ejemplo, Python, R, Javascript, Golang, etc., por lo que es posible decir que a pesar que el Big Data tiene sus orígenes en herramientas no tan actuales, las técnicas usadas para la aplicación de esta tecnología se mantendrán en constante desarrollo buscando siempre mejorar la capacidad de procesamiento de información y la facilidad de uso.

Aplicando el paradigma Business Intelligence sobre los datos obtenidos en la aplicación se logró generar varios reportes con diferentes propósitos que van desde una simple consulta para determinar la cantidad de personas que usan la aplicación clasificadas por género, hasta reportes más complejos sobre la cantidad de usuarios a los cuales les guste o no un determinado tema, en base a las diferentes actividades que realicen en la aplicación y relacionándolo con los usuario con los cuales tiene "interacción", mencionando únicamente estos dos ejemplos ya es posible tener una idea del impacto que tiene esta tecnología en campos específicos como el marketing, la publicidad, e-commerce, etc.

Por otro lado, teniendo en cuenta las pruebas realizadas, se puede llegar a concluir que la aplicación desarrollada presenta cifras y porcentajes muy buenos en cuanto a tiempo de respuesta general, y tiempo de respuesta de la base de datos, además de un alto porcentaje en cuanto a confiabilidad de uso, y una excelente capacidad de procesamiento de datos desde su obtención hasta su análisis, por lo que es posible afirmar que una web orientada a la tecnología Big Data resultaría muy eficiente al ser construida mediante el stack MEAN

De igual manera, en base a la investigación realizada acerca del stack de desarrollo MEAN, así como de la tecnología Big Data y sus aplicaciones, y una vez expuesta la discusión y definidas las ventajas y desventajas del stack de desarrollo MEAN, es posible concluir que el stack MEAN es una herramienta sumamente escalable, y aunque sus características dependen en gran parte del hardware, puede definirse como flexible, completa, ligera, con un rápido acceso sobre las consultas a la información almacenada, con un nivel de fallos relativamente bajo cuando es sometida a condiciones de sobrecarga, y con las suficientes características y ventajas para ser considerada al momento de desarrollar aplicaciones web orientadas a la tecnología Big Data, de igual manera tiene ciertas desventajas mencionadas anteriormente, que deben ser consideradas y superadas de manera adecuada, haciendo especial énfasis en la complejidad del uso de sus componentes y la alta curva de aprendizaje que presenta.

5.2. RECOMENDACIONES

A manera de recomendación cabe resaltar que es necesario llevar buenas prácticas en cuanto al desarrollo del código, y tener un paradigma de programación definido, esto teniendo en cuenta que el desarrollo en Javascript debe ser tan estable como para poder obtener una aplicación lo suficiente limpia, escalable, y evitando redundancias en lo posible.

Realizar pruebas específicas relacionadas al hardware sobre el cuál la aplicación va a ser desplegada, esto permitirá determinar puntos en los cuales la aplicación podría fallar, dando una visión global de en que momento podría ser necesario un aumento en las capacidades del servidor de ejecución.

Teniendo en cuenta lo anterior, es necesario recomendar también distribuir la carga relacionada a la base de datos de ser necesario, MongoDB de por si tiene una estructura ligera y una forma de indexar los documentos en las colecciones que mejora sustancialmente su capacidad de búsqueda debido a su naturaleza de escalamiento horizontal, pero al llegar a un determinado límite dependiendo del hardware, las consultas pueden llegar a ser lentas y provocar fallos tanto en la respuesta obtenida como en el tiempo que se demora en hacerlas.

Finalmente, se propone como trabajo futuro el desarrollo de aplicaciones orientadas al Big Data mediante el stack MEAN, en ámbitos diferentes a lo relacionado a redes sociales, siendo un ejemplo a considerar ambientes más empresariales que también puedan generar grandes cantidades de información con lo cual se pueda aplicar el paradigma Business Intelligence y generar conocimiento que sea de utilidad para una determinada compañía, ya sea acerca de la publicidad o de la toma de decisiones futuras como fue señalado a manera de ejemplo anteriormente en este documento.

CAPÍTULO VI

REFERENCIAS

6.1. REFERENCIAS BIBLIOGRÁFICAS

1&1. (17 de Octubre de 2016). *1&1 Internet España S.L.U.* Obtenido de https://www.1and1.es/digitalguide/servidores/know-how/desarrollo-web-con-stacks-de-software/

1, 1. &. (17 de Octubre de 2016). *1 & 1 Internet España S.L.U.* Obtenido de https://www.1and1.es/digitalguide/servidores/know-how/desarrollo-web-con-stacks-de-software/

Barranco, R. (18 de Junio de 2012). *IBM*. Obtenido de https://www.ibm.com/developerworks/ssa/local/im/que-es-big-data/

Chung, L., & Sampaio, J. (2009). On Non-Functional Requirements in Software. *Book Conceptual Modeling: Foundations and Applications*, 363-379. doi:10.1007/978-3-642-02463-4_19

Dresner, H. (2008). *The Perfomance Management Revolution.* New Jersey: John Wiley & Sons, Inc.

Haviv, A. (15 de Febrero de 2014). *MEAN.JS Official Blog*. Obtenido de http://blog.meanjs.org/post/76726660228/forking-out-of-an-open-source-conflict

Karpov, V. (30 de Abril de 2013). *The MEAN Stack: MongoDB, ExpressJS, AngularJS and Node.js.* Obtenido de https://www.mongodb.com/blog/post/the-mean-stack-mongodb-expressjs-angularjs-and

Mera, J. (2016). Análisis del proceso de pruebas de calidad de software. *Ingeniería Solidaria, 12*(20), 163-176. doi: http://dx.doi.org/10.16925/in.v12i20.1482

Oliver, A. (17 de Julio de 2014). *InfoWorld.* Obtenido de https://www.infoworld.com/: https://www.infoworld.com/article/2608460/application-development/the-10-worst-big-data-practices.html

Pankaj, K. (17 de Mayo de 2018). *LinkedIn.* Obtenido de https://www.linkedin.com/pulse/golang-big-data-pankaj-khairnar

Paruchuri, V. (15 de Diciembre de 2016). *Quora.* Obtenido de https://www.quora.com/Which-is-better-for-data-analysis-R-or-Python-Is-R-still-a-better-data-analysis-language-than-Python-Has-anyone-else-used-Python-with-Pandas-to-a-large-extent-in-data-analysis-projects

SINNEXUS. (2016). *Sinnexus*. Obtenido de http://www.sinnexus.com/business_intelligence/

Arredondo, R., & Muñoz, L. (2017). Reparación de Data Warehouses con sentido semántico. *Revista Cubana de Ciencias Informáticas*, *11*(2), 73–85.

Camargo, J., Camargo, J., & Joyanes, L. (2015). Conociendo Big data. *Revista Facultad de Ingeniería (Fac. Ing.), Enero-Abril*, *24*(38), 63–77. https://doi.org/10.15517/eci.v6i1.19005

Chen, H., Chinag, R., & Storey, V. (2012). Business Intelligence and Analytics: From Big Data To Big Impact. *Mis Quarterly*, *36*(4), 1165–1188. https://doi.org/10.1145/2463676.2463712

Cotino, L. (2017). Big data e inteligencia artificial . Una aproximación a su tratamiento jurídico desde los derechos fundamentales Big Data and Artificial Intelligence . An Approach from a Legal Point of View about Fundamental Rights. *Dilemata*, *24*(0), 131–150. Retrieved from http://www.dilemata.net/revista/index.php/dilemata/article/view/412000104

Cravero Leal, A., Sepulveda, S., Mate, A., Mazon, J. N., & Trujillo, J. (2014). Goal oriented requirements engineering in data warehouses: A comparative study. *Ingenieria E Investigacion*, *34*(2), 66–70.

Dean, J., & Ghemawat, S. (2008). MapReduce: Simplified Data Processing on Large Clusters. *Communications of the ACM*, *51*(1), 107–113. https://doi.org/10.1145/1327452.1327492

Dominguez, A. H., & Yeja, A. H. (2015). Acerca de la aplicación de MapReduce + Hadoop en el tratamiento de Big Data. *Rcci*, *9*(3), 49–63. Retrieved from http://www.redalyc.org/pdf/3783/378343683010.pdf

Duy Khue, T., Thanh Binh, N., Chang, W., Kim, C., & Chung, S. (2017). Design and Implementation of MEAN Stack-based Scalable Real-time Digital Signage System. *International Conference of Information and Communication Technology for Embedded Systems, 8*, 1–6. https://doi.org/10.1109/ICTEmSys.2017.7958779

German, A., Salmeron, S., Ha, W., & Henderson, B. (2016). MEAN Web Development.

Proceedings of the 17th Annual Conference on Information Technology Education - SIGITE '16, 128–129. https://doi.org/10.1145/2978192.2978247

González, R., González, R., & López, R. (2014). Javascript sean stack: una mirada al futuro del desarrollo web. *ResearchGate*, 12. Retrieved from https://www.researchgate.net/profile/Rubiel_Gonzalez_Labarta/publication/261988203_JAVASCRIPT_SEAN_STACK_UNA_MIRADA_AL_FUTURO_DEL_DESARROLLO_WEB/links/0c9605361d5843c7d4000000/JAVASCRIPT-SEAN-STACK-UNA-MIRADA-AL-FUTURO-DEL-DESARROLLO-WEB.pdf

Hernández, E., Duque, N., & Moreno, J. (2017). Big Data : una exploración de investigaciones , tecnologías y casos de aplicación. *Tecno Lógicas*, *20*(39), 25. Retrieved from www.scielo.org.co/pdf/teclo/v20n39/v20n39a02.pdf

Hernández Lamillar, R. M. (2005). Herramientas del Marketing: Data Warehousing, Tecnologia necesaria para el Comercio Internacional. *Revista CENIC Ciencias Biologicas.*, *36*, 1–5. Retrieved from http://web.ebscohost.com/ehost/pdfviewer/pdfviewer?sid=57889979-87f8-459a-a673-b0c89dc4f300%40sessionmgr114&vid=1&hid=118

Jameel, S., Khanna, S., Rangappa, S., Patil, S., & Khalkar, R. G. (2017). MEAN-vs-JAX RS for RESTful Web Services : Performance and Efficiency Analysis, *6*(3), 155–161.

John, S. (2014). Big Data: A Revolution That Will Transform How We Live, Work, and Think. *International Journal of Advertising*, *33*(1), 181–183. https://doi.org/10.2501/IJA-33-1-181-183

Kadam, A., Chaudhari, H., Patil, C., & Chavhan, S. (2017). Web Application Development by Using MEAN Stack. *International Journal of Science Technology Management and Research*, *2*(4), 15–18. Retrieved from www.ijstmr.com/wp-content/uploads/.../IJSTMR_V2I4_13106.pdf

Karanjit, A. (2016). *MEAN vs. LAMP Stack*. St. Cloud State University. Retrieved from http://repository.stcloudstate.edu/csit_etds/11/

Martínez, J. (2015). Big Data aplicación y utilidad para el sistema sanitario. *Farm Hosp*, *39*(2), 69–70. https://doi.org/10.7399/fh.2015.39.2.8835

Martínez, R. (2016). Big Data en humanidades digitales: de la escritura digital a la "lectura distante." *Revista Chilena de Literatura*, (94), 39–58. Retrieved from http://www.revistaliteratura.uchile.cl/index.php/RCL/article/viewFile/44969/47051

Mayer-Schönberger, V., & Cukier, K. (2013). *Big Data La revolución de los dato masivos*. Madrid: Turner.

McAfee, A., & Brynjolfsson, E. (2012). Big data: the management revolution. *Harvard Business Review*, *90*(10), 61–67. https://doi.org/00475394

Moody, D., & Kortink, M. (2000). From Enterprise Models to Dimensional Models: A Methodology for Data Warehouse and Data Mart Design. *Proceedings of the International Workshop on Design and Management of Data Warehouses (DMDW'2000)*, *2000*, 5–16.

Nuñez, Y., Díaz de la Paz, L., & García, J. (2016). Algoritmo para corregir anomalías a nivel de instancia en grandes volúmenes de datos utilizando MapReduce. *Revista Cubana de Ciencias Informáticas*, *10*(3), 105–118. Retrieved from http://rcci.uci.cu/?journal=rcci&page=article&op=download&path%5B%5D=1344&path%5B%5D=452

Poulter, A., Johnston, S., & Cox, S. (2015). Using the MEAN stack to implement a RESTful service for an Internet of Things application. *IEEE World Forum on Internet of Things, WF-IoT 2015 - Proceedings*, *2*, 6. https://doi.org/10.1109/WF-IoT.2015.7389066

Ricci, D., Sprimont, P., Ayala, C., Ramón Fox, F., Michel, R., Navarro, S., … Reyes Ruiz, M. (2014). WEB-BASED TOOLS FOR THE ANALYSIS OF TAOS DATA AND MUCH. *Revista Mexicana de Astronomía Y Astrofísica*, *45*, 42–44. Retrieved from http://www.redalyc.org/articulo.oa?id=57132995015

Serrano, J. (2013). Big Data y Not So Big Data. *Anuario ThinkEPI*, *7*(1), 161–163. Retrieved from https://dialnet.unirioja.es/descarga/articulo/4234739.pdf

Serrano, J. (2014). Big Data. *El Profesional de La Información*, *23*(6), 561–566. https://doi.org/10.1080/15228959.2012.730411

Stajcer, M., Stajcer, M., & Orescanin, D. (2016). Using MEAN stack for development of GUI in real-time big data architecture. *International Convention on Information*

and Communication Technology, Electronics and Microelectronics, MIPRO, 39, 524–529. https://doi.org/10.1109/MIPRO.2016.7522200

6.2. ANEXOS

ANEXO 1

FORMATO IEEE 830 PARA ESPECIFICACIONES DE REQUERIMIENTOS DE SOFTWARE SRS

1. Introducción

Este documento es una Especificación de Requisitos Software (ERS) para la aplicación web MEAN Social Network como parte del trabajo final de tesis de grado titulado “USO DEL STACK MEAN PARA EL DESARROLLO DE UNA APLICACIÓN WEB ORIENTADA A LA TECNOLOGÍA BIG DATA”. Esta especificación se ha estructurado basándose en las directrices dadas por el estándar IEEE Práctica Recomendada para Especificaciones de Requisitos Software ANSI/IEEE 830, 1998.

1.1. Propósito

El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales para el desarrollo de una aplicación web orientada al uso de la tecnología Big Data. Ésta podrá ser utilizada por cualquier tipo de usuario

1.2. Alcance

Esta especificación de requisitos está dirigida a usuarios de la aplicación y lectores del trabajo final de tesis de grado antes mencionados, para profundizar en el entendimiento de la aplicación y sus funcionalidades, la cual tiene por objetivo principal la generación, análisis y uso de datos de distintos tipos de usuarios.

1.3. Personal involucrado

Nombre	Marcelo Patricio Sánchez Tarira
Rol	Diseñador, Desarrollador y Programador
Categoría profesional	Egresado Ingeniería en Sistemas
Responsabilidades	Análisis de información, diseño y programación
Información de contacto	m.sancheztarira@gmail.com
Aprobación	✓

1.4. Definiciones, acrónimos y abreviaturas

Nombre	***Descripción***
Usuario	Persona registrada capaz de acceder a la aplicación
No Usuario	Persona no registrada en la aplicación

MEAN	Acrónimo para definir el stack: Mongo, Express, Angular, NodeJS
MEAN SN	Abreviatura del nombre dado a la aplicación MEAN Social Network

1.5. Referencias

Título del Documento	Referencia
Standard IEEE 830 - 1998	IEEE

1.6. Resumen

Este documento consta de tres secciones. En la primera sección se realiza una introducción al mismo y se proporciona una visión general de la especificación de recursos del sistema.

En la segunda sección del documento se realiza una descripción general del sistema, con el fin de conocer las principales funciones que éste debe realizar, los datos asociados y los factores, restricciones, supuestos y dependencias que afectan al desarrollo, sin entrar en excesivos detalles.

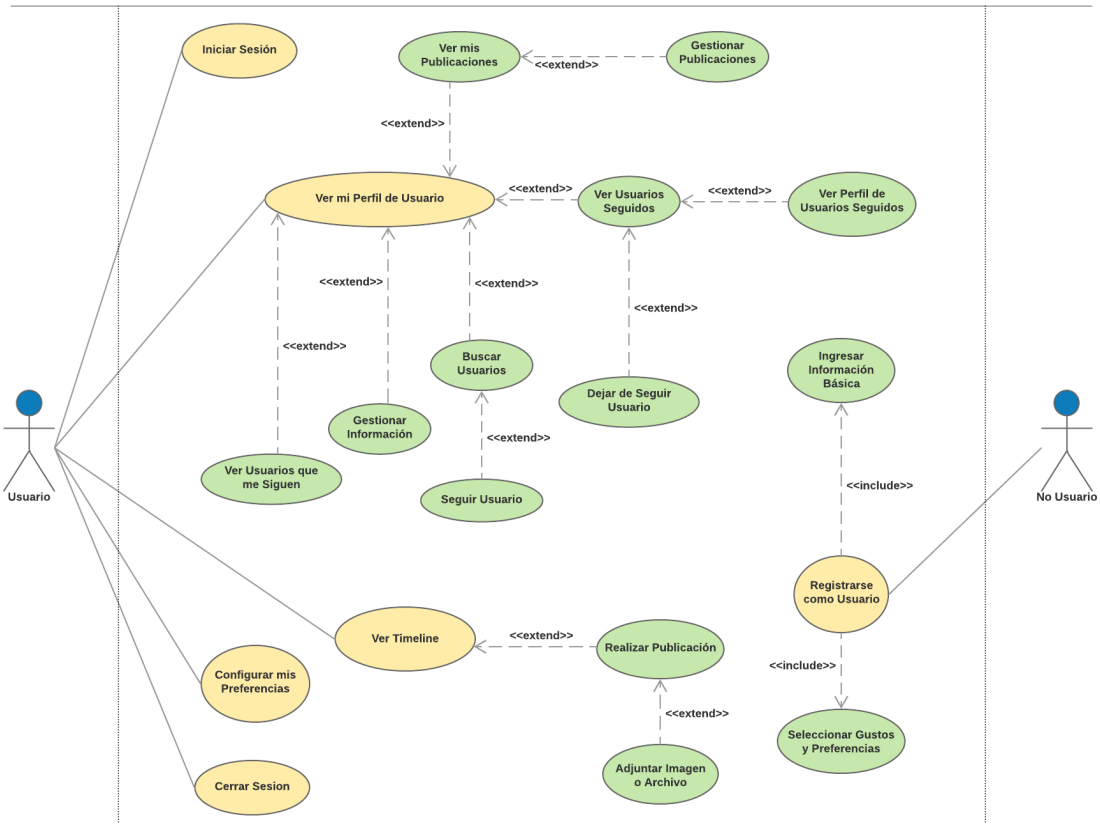
Por último, la tercera sección del documento es aquella en la que se definen detalladamente los requisitos que debe satisfacer el sistema.

2. Descripción general

2.1. Perspectiva del producto

MEAN Social Network es una aplicación web creada con la estructura y características básicas de una red social, con la particularidad que el entorno completo sobre el cual funciona fue diseñado únicamente con frameworks cuyo lenguaje de programación es Javascript, y la base de datos con la cual funciona es MongoDB.

2.2. Funcionalidad del producto



2.3. Características de los usuarios

Tipo de usuario	Usuario
Formación	N/A
Habilidades	N/A
Actividades	Publicaciones, Timeline, Manejo de Seguidores, Manejo de su información y gustos ingresados

Tipo de usuario	No Usuario
Formación	N/A
Habilidades	N/A
Actividades	Registro como Usuario

2.4. Restricciones

- Lenguajes y tecnologías en uso: Stack MEAN, Javascript como lenguaje base.
- Los servidores deben ser capaces de atender consultas concurrentemente.
- El sistema esta construido según un modelo cliente/servidor y cuenta con un api que comunica el frontend con las funcionalidades en el backend.

2.5. Suposiciones y dependencias

- Se asume que los requisitos aquí descritos son estables
- Los equipos en los que se vaya a ejecutar el sistema deben cumplir los requisitos indicados para garantizar una ejecución correcta de la misma

3. Requisitos específicos

Requerimientos Funcionales

Identificación del requerimiento:	RF01
Nombre del Requerimiento:	Autentificación de Usuario.
Características:	Los usuarios deberán identificarse para acceder a cualquier parte del sistema.
Descripción del requerimiento:	El sistema podrá ser accedido por cualquier usuario autentificado siempre y cuando este haya seguido los pasos de registro correspondientes.
Requerimiento NO funcional:	• RNF01 • RNF02 • RNF05
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF02
Nombre del Requerimiento:	Registrar Usuarios.
Características:	Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del mismo.
Descripción del requerimiento:	El sistema permitirá a un no usuario registrarse. El usuario debe suministrar datos como: Nombre, Apellido, E-mail, Usuario y Password.
Requerimiento NO funcional:	• RNF01 • RNF02 • RNF05
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF03
Nombre del Requerimiento:	Perfil de Usuario.
Características:	La aplicación contará con un apartado donde se mostrará específicamente el perfil, configuraciones, publicaciones, etc. Del usuario identificado.
Descripción del requerimiento:	**Ver Publicaciones:** Muestra las publicaciones realizadas por el usuario, y brinda la posibilidad de crear, editar o eliminar publicaciones.
Requerimiento NO funcional:	• RNF01 • RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF04
Nombre del Requerimiento:	Perfil de Usuario.
Características:	La aplicación contará con un apartado donde se mostrará específicamente el perfil, configuraciones, publicaciones, etc. Del usuario identificado.

Descripción del requerimiento:	**Ver Usuarios Seguidos:** Muestra los usuarios seguidos actualmente por el usuario identificado, y brinda la posibilidad de dejar de seguir, o ver otro perfil de usuario con sus publicaciones.
Requerimiento NO funcional:	• RNF01 • RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF05
Nombre del Requerimiento:	Perfil de Usuario.
Características:	La aplicación contará con un apartado donde se mostrará específicamente el perfil, configuraciones, publicaciones, etc. Del usuario identificado.
Descripción del requerimiento:	**Buscar Usuarios:** Permite consultar y mostrar los usuarios que se encuentran actualmente en uso de la aplicación web, y la posibilidad de empezar a seguirlos.
Requerimiento NO funcional:	• RNF01 • RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF06
Nombre del Requerimiento:	Perfil de Usuario.
Características:	La aplicación contará con un apartado donde se mostrará específicamente el perfil, configuraciones, publicaciones, etc. Del usuario identificado.
Descripción del requerimiento:	**Gestionar Información:** Muestra la configuración actual de la información ingresada del usuario al momento de su registro, y permite editar.
Requerimiento NO funcional:	• RNF01 • RNF02 • RNF03 • RNF05
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF07
Nombre del Requerimiento:	Perfil de Usuario.
Características:	La aplicación contará con un apartado donde se mostrará específicamente el perfil, configuraciones, publicaciones, etc. Del usuario identificado.
Descripción del requerimiento:	**Ver Usuarios que me Siguen:** Muestra una lista de los usuarios que siguen al usuario actual identificado.
Requerimiento NO funcional:	• RNF01 • RNF03
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF08
Nombre del Requerimiento:	Timeline.

Características:	La aplicación contará con un apartado donde se mostrará en orden de fecha las publicaciones realizadas por el usuario registrado, así como por los usuarios a los cuales sigue.
Descripción del requerimiento:	**Timeline:** Permite ver las publicaciones propias del usuario y de los usuarios a los cuales sigue.
Requerimiento NO funcional:	• RNF01 • RNF02 • RNF03 • RNF04
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF09
Nombre del Requerimiento:	Timeline.
Características:	La aplicación contará con un apartado donde se mostrará en orden de fecha las publicaciones realizadas por el usuario registrado, así como por los usuarios a los cuales sigue.
Descripción del requerimiento:	**Realizar Publicación:** Permite al usuario identificado realizar una publicación, así como adjuntar una imagen o archivo de texto de ser requerido por la publicación.
Requerimiento NO funcional:	• RNF01 • RNF02 • RNF03 • RNF04
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RF10
Nombre del Requerimiento:	Configurar Preferencias.
Características:	La aplicación contará con un apartado donde será posible listar las configuraciones del usuario en cuanto a las preferencias de sus gustos.
Descripción del requerimiento:	Muestra la configuración actual de la información ingresada del usuario sobre las cosas que le gustan, y las que no, además permite editar en base a sus preferencias.
Requerimiento NO funcional:	• RNF01 • RNF02 • RNF03 • RNF05
Prioridad del requerimiento: Alta	

Requerimientos No Funcionales.

Identificación del requerimiento:	RNF01
Nombre del Requerimiento:	Interfaz del sistema.
Características:	El sistema presentara una interfaz de usuario sencilla para que sea de fácil manejo a los usuarios del sistema.
Descripción del requerimiento:	El sistema debe tener una interfaz de uso intuitiva y sencilla.
Prioridad del requerimiento:	

Alta

Identificación del requerimiento:	RNF02
Nombre del Requerimiento:	Desempeño
Características:	El sistema garantizara a los usuarios un desempeño en cuanto a los datos almacenado en el sistema ofreciéndole una confiabilidad a esta misma.
Descripción del requerimiento:	Garantizar el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF03
Nombre del Requerimiento:	Usuario Identificado
Características:	Garantizara que usuario tenga acceso únicamente a su información, y a la información mostrada por los usuarios que sigue.
Descripción del requerimiento:	Facilidades y controles para permitir el acceso a la información al usuario identificado a través de Internet, con la intención de consultar y subir información pertinente para cada una de ellas.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF04
Nombre del Requerimiento:	Confiabilidad continúa del sistema.
Características:	El sistema tendrá que estar en funcionamiento las 24 horas los 7 días de la semana. Ya que es una página web diseñada para la carga de datos y comunicación entre usuarios.
Descripción del requerimiento:	La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios de 7 días por 24 horas, garantizando un esquema adecuado que no permita posibles fallas en la medida en que la aplicación pueda controlarlas.
Prioridad del requerimiento: Alta	

Identificación del requerimiento:	RNF05
Nombre del Requerimiento:	Seguridad en información
Características:	El sistema garantizara a los usuarios una seguridad en cuanto a la información que se procede en el sistema.
Descripción del requerimiento:	Garantizar la seguridad del sistema con respecto a la información y datos que se manejan tales sean documentos, archivos y contraseñas.
Prioridad del requerimiento: Alta	

3.1. Requisitos comunes de los interfaces

3.1.1. Interfaces de usuario

La interfaz de usuario para un usuario identificado presenta un total tres pantallas principales según los requerimientos anteriores, en las cuales existen enlaces hacia diferentes perfiles de usuario, así como al propio, además de campos de texto para búsquedas y publicaciones, así como para adjuntar archivos, de igual manera se pueden apreciar botones de acceso entre las pantallas y botones de envío de datos.

3.1.2. Interfaces de hardware

Para el caso de la investigación presentada sobre el tratamiento de datos se dispuso de un servidor con las siguientes características:

- 8 Gb en memoria RAM
- 4 vCpu's
- Disco de estado sólido de 160GB.

3.1.3. Interfaces de software

Sistema operativo Ubuntu 18.04 corriendo Apache como web server.

3.1.4. Interfaces de comunicación

El frontend construido sobre Angular se comunica con el backend y la base de datos mediante un api construida mediante Express como manejador de rutas.

3.2. Requisitos funcionales

3.2.1. Requisito funcional 1

- **Autentificación de Usuarios:** los usuarios deberán identificarse para acceder a cualquier parte del sistema.

    - ✓ El sistema podrá ser accedido por cualquier usuario autentificado siempre y cuando este haya seguido los pasos de registro correspondientes.

3.2.2. Requisito funcional 2

- **Registrar Usuarios:** Los usuarios deberán registrarse en el sistema para acceder a cualquier parte del mismo.

    - ✓ El sistema permitirá a un no usuario registrarse. El usuario debe suministrar datos como: Nombre, Apellido, E-mail, Usuario y Password.

3.2.3. Requisito funcional 3

- **Perfil de Usuario:** La aplicación contará con un apartado donde se mostrará específicamente el perfil, configuraciones, publicaciones, etc. Del usuario identificado.

    - ✓ **Ver Publicaciones:** Muestra las publicaciones realizadas por el usuario, y brinda la posibilidad de crear, editar o eliminar publicaciones.

- ✓ **Ver Usuarios Seguidos:** Muestra los usuarios seguidos actualmente por el usuario identificado, y brinda la posibilidad de dejar de seguir, o ver otro perfil de usuario con sus publicaciones.
- ✓ **Buscar Usuarios:** Permite consultar y mostrar los usuarios que se encuentran actualmente en uso de la aplicación web, y la posibilidad de empezar a seguirlos.
- ✓ **Gestionar Información:** Muestra la configuración actual de la información ingresada del usuario al momento de su registro, y permite editar.
- ✓ **Ver Usuarios que me Siguen:** Muestra una lista de los usuarios que siguen al usuario actual identificado.

3.2.4. Requisito funcional 8

- **Timeline:** La aplicación contará con un apartado donde se mostrará en orden de fecha las publicaciones realizadas por el usuario registrado, así como por los usuarios a los cuales sigue.

    - ✓ **Timeline:** Permite ver las publicaciones propias del usuario y de los usuarios a los cuales sigue.
    - ✓ **Realizar Publicación:** Permite al usuario identificado realizar una publicación, así como adjuntar una imagen o archivo de texto de ser requerido por la publicación.

3.2.5. Requisito funcional 10

- **Configurar Preferencias:** La aplicación contará con un apartado donde será posible listar las configuraciones del usuario en cuanto a las preferencias de sus gustos.

    - ✓ Muestra la configuración actual de la información ingresada del usuario sobre las cosas que le gustan, y las que no, además permite editar en base a sus preferencias.

3.3. Requisitos no funcionales

3.3.1. Requisitos de rendimiento

- Garantizar que el diseño del sistema y las consultas realizadas sobre las bases de datos, entre otros procesos simultáneos, no afecten el desempeño de la aplicación misma ni afecte el tiempo de respuesta de la base de datos.

3.3.2. Seguridad

- Garantizar la confiabilidad, la seguridad y el desempeño del sistema informático a los diferentes usuarios. En este sentido la información almacenada o registros realizados podrán ser consultados y actualizados permanente y simultáneamente, sin que se afecte el tiempo de respuesta.

- Garantizar la seguridad del sistema con respecto a la información y datos que se manejan tales sean documentos, archivos y contraseñas.

3.3.3. Fiabilidad

- El sistema debe tener una interfaz de uso intuitiva y sencilla
- La interfaz de usuario debe ajustarse a las características de la web de la institución, dentro de la cual estará incorporado el sistema de gestión de procesos y el inventario

3.3.4. Disponibilidad

- La disponibilidad del sistema debe ser continua con un nivel de servicio para los usuarios de 7 días por 24 horas, garantizando un esquema adecuado que no permita posibles fallas en la medida en que la aplicación pueda controlarlas.

3.3.5. Portabilidad

- La aplicación debe ser correctamente mostrada en navegadores Chrome, Mozilla Firefox, Internet Explorer y Opera.

ANEXO 2

SCRIPT DE EJECUCIÓN PARA PRUEBAS DE CARGA Y ESTRES

```
<?xml version="1.0" encoding="UTF-8"?>
<jmeterTestPlan version="1.2" properties="5.0" jmeter="5.0 r1840935">
 <hashTree>
  <TestPlan guiclass="TestPlanGui" testclass="TestPlan" testname="Test Plan" enabled="true">
   <stringProp name="TestPlan.comments"></stringProp>
   <boolProp name="TestPlan.functional_mode">false</boolProp>
   <boolProp name="TestPlan.tearDown_on_shutdown">true</boolProp>
   <boolProp name="TestPlan.serialize_threadgroups">false</boolProp>
   <elementProp name="TestPlan.user_defined_variables" elementType="Arguments" guiclass="ArgumentsPanel"
testclass="Arguments" testname="User Defined Variables" enabled="true">
    <collectionProp name="Arguments.arguments"/>
   </elementProp>
   <stringProp name="TestPlan.user_define_classpath"></stringProp>
  </TestPlan>
  <hashTree>
   <ThreadGroup guiclass="ThreadGroupGui" testclass="ThreadGroup" testname="Thread Group" enabled="true">
    <stringProp name="ThreadGroup.on_sample_error">continue</stringProp>
    <elementProp name="ThreadGroup.main_controller" elementType="LoopController" guiclass="LoopControlPanel"
testclass="LoopController" testname="Loop Controller" enabled="true">
     <boolProp name="LoopController.continue_forever">false</boolProp>
     <intProp name="LoopController.loops">-1</intProp>
    </elementProp>
    <stringProp name="ThreadGroup.num_threads">1</stringProp>
    <stringProp name="ThreadGroup.ramp_time">1</stringProp>
    <boolProp name="ThreadGroup.scheduler">false</boolProp>
    <stringProp name="ThreadGroup.duration"></stringProp>
    <stringProp name="ThreadGroup.delay"></stringProp>
   </ThreadGroup>
   <hashTree>
    <Arguments guiclass="ArgumentsPanel" testclass="Arguments" testname="User Defined Variables" enabled="true">
     <collectionProp name="Arguments.arguments">
      <elementProp name="server" elementType="Argument">
       <stringProp name="Argument.name">server</stringProp>
       <stringProp name="Argument.value">142.93.248.176</stringProp>
       <stringProp name="Argument.desc">Direccion del servidor</stringProp>
       <stringProp name="Argument.metadata">=</stringProp>
      </elementProp>
      <elementProp name="port" elementType="Argument">
       <stringProp name="Argument.name">port</stringProp>
       <stringProp name="Argument.value">27017</stringProp>
       <stringProp name="Argument.desc">Puerto MongoDb</stringProp>
       <stringProp name="Argument.metadata">=</stringProp>
      </elementProp>
      <elementProp name="database" elementType="Argument">
       <stringProp name="Argument.name">database</stringProp>
       <stringProp name="Argument.value">mean_social_network</stringProp>
       <stringProp name="Argument.desc">Base de Datos</stringProp>
       <stringProp name="Argument.metadata">=</stringProp>
      </elementProp>
      <elementProp name="user" elementType="Argument">
       <stringProp name="Argument.name">user</stringProp>
       <stringProp name="Argument.value">mean_user</stringProp>
       <stringProp name="Argument.desc">Usuario de la Base de Datos</stringProp>
       <stringProp name="Argument.metadata">=</stringProp>
      </elementProp>
      <elementProp name="password" elementType="Argument">
       <stringProp name="Argument.name">password</stringProp>
       <stringProp name="Argument.value">mean_pass</stringProp>
       <stringProp name="Argument.desc">Pass para la Base de Datos</stringProp>
       <stringProp name="Argument.metadata">=</stringProp>
      </elementProp>
     </collectionProp>
    </Arguments>
    <hashTree/>
    <JSR223Sampler guiclass="TestBeanGUI" testclass="JSR223Sampler" testname="Consulta Simple" enabled="true">
     <stringProp name="cacheKey">true</stringProp>
     <stringProp name="filename"></stringProp>
     <stringProp name="parameters"></stringProp>
     <stringProp name="script">
```

```
import com.mongodb.MongoClient;
import com.mongodb.ServerAddress;
import com.mongodb.DB;
import com.mongodb.DBCollection;

//Conexion y Autentificacion
MongoClient mongoClient = new MongoClient(new ServerAddress(&quot;${server}:${port}&quot;));
DB mean_sn = mongoClient.getDB(&quot;${database}&quot;);

//Colecciones
DBCollection users = mean_sn.getCollection(&quot;users&quot;);
DBCollection follows = mean_sn.getCollection(&quot;follows&quot;);
DBCollection publications = mean_sn.getCollection(&quot;publications&quot;);

//Obtener Datos
long users_count = users.getCount();
String users_count_parsed = String.valueOf(users_count);
SampleResult.setResponseData(users_count_parsed.getBytes());
long follows_count = follows.getCount();
String follows_count_parsed = String.valueOf(follows_count);
SampleResult.setResponseData(follows_count_parsed.getBytes());
long publications_count = publications.getCount();
String publications_count_parsed = String.valueOf(publications_count);
SampleResult.setResponseData(publications_count_parsed.getBytes());
</stringProp>
<stringProp name="scriptLanguage">groovy</stringProp>
</JSR223Sampler>
<hashTree>
<ResponseAssertion guiclass="AssertionGui" testclass="ResponseAssertion" testname="Response Assertion"
enabled="true">
<collectionProp name="Asserion.test_strings">
<stringProp name="91555">\d+</stringProp>
</collectionProp>
<stringProp name="Assertion.test_field">Assertion.response_data</stringProp>
<boolProp name="Assertion.assume_success">false</boolProp>
<intProp name="Assertion.test_type">1</intProp>
<stringProp name="Assertion.custom_message"></stringProp>
</ResponseAssertion>
<hashTree/>
</hashTree>
<JSR223Sampler guiclass="TestBeanGUI" testclass="JSR223Sampler" testname="Insercion en Publications"
enabled="true">…….
```

En caso de necesitar el script de ejecución completo comunicarse con el autor de este documento.

ANEXO 3

SCRIPT DE EJECUCIÓN DE PETICIONES PARA PRUEBA DE TIEMPO DE RESPUESTA

```
<?xml version="1.0" encoding="UTF-8"?>
<jmeterTestPlan version="1.2" properties="5.0" jmeter="5.0 r1840935">
  <hashTree>
    <TestPlan guiclass="TestPlanGui" testclass="TestPlan" testname="Test Plan" enabled="true">
      <stringProp name="TestPlan.comments"></stringProp>
      <boolProp name="TestPlan.functional_mode">false</boolProp>
      <boolProp name="TestPlan.serialize_threadgroups">false</boolProp>
      <elementProp name="TestPlan.user_defined_variables" elementType="Arguments" guiclass="ArgumentsPanel"
            testclass="Arguments" testname="User Defined Variables" enabled="true">
        <collectionProp name="Arguments.arguments"/>
      </elementProp>
      <stringProp name="TestPlan.user_define_classpath"></stringProp>
    </TestPlan>
    <hashTree>
      <ThreadGroup guiclass="ThreadGroupGui" testclass="ThreadGroup" testname="Thread Group" enabled="true">
        <stringProp name="ThreadGroup.on_sample_error">continue</stringProp>
        <elementProp name="ThreadGroup.main_controller" elementType="LoopController" guiclass="LoopControlPanel"
            testclass="LoopController" testname="Loop Controller" enabled="true">
          <boolProp name="LoopController.continue_forever">false</boolProp>
          <intProp name="LoopController.loops">-1</intProp>
        </elementProp>
        <stringProp name="ThreadGroup.num_threads">5000</stringProp>
        <stringProp name="ThreadGroup.ramp_time">60</stringProp>
        <longProp name="ThreadGroup.start_time">1370726934000</longProp>
        <longProp name="ThreadGroup.end_time">1370726934000</longProp>
        <boolProp name="ThreadGroup.scheduler">false</boolProp>
        <stringProp name="ThreadGroup.duration"></stringProp>
        <stringProp name="ThreadGroup.delay"></stringProp>
      </ThreadGroup>
      <hashTree>
        <RecordingController guiclass="RecordController" testclass="RecordingController" testname="Recording Controller"
            enabled="true"/>
        <hashTree>
          <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="1 /mean_social_network/"
            enabled="true">
            <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel"
            testclass="Arguments" enabled="true">
              <collectionProp name="Arguments.arguments"/>
            </elementProp>
            <stringProp name="HTTPSampler.domain">localhost</stringProp>
            <stringProp name="HTTPSampler.port"></stringProp>
            <stringProp name="HTTPSampler.protocol">http</stringProp>
            <stringProp name="HTTPSampler.contentEncoding"></stringProp>
            <stringProp name="HTTPSampler.path">/mean_social_network/</stringProp>
            <stringProp name="HTTPSampler.method">GET</stringProp>
```

```
  <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
  <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
  <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
  <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
  <boolProp name="HTTPSampler.image_parser">true</boolProp>
  <stringProp name="HTTPSampler.embedded_url_re"></stringProp>
  <stringProp name="HTTPSampler.connect_timeout"></stringProp>
  <stringProp name="HTTPSampler.response_timeout"></stringProp>
</HTTPSamplerProxy>
<hashTree>
  <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header Manager" enabled="true">
    <collectionProp name="HeaderManager.headers">
      <elementProp name="Accept-Language" elementType="Header">
        <stringProp name="Header.name">Accept-Language</stringProp>
        <stringProp name="Header.value">en-US,en;q=0.5</stringProp>
      </elementProp>
      <elementProp name="Upgrade-Insecure-Requests" elementType="Header">
        <stringProp name="Header.name">Upgrade-Insecure-Requests</stringProp>
        <stringProp name="Header.value">1</stringProp>
      </elementProp>
      <elementProp name="Accept-Encoding" elementType="Header">
        <stringProp name="Header.name">Accept-Encoding</stringProp>
        <stringProp name="Header.value">gzip, deflate</stringProp>
      </elementProp>
      <elementProp name="User-Agent" elementType="Header">
        <stringProp name="Header.name">User-Agent</stringProp>
        <stringProp name="Header.value">Mozilla/5.0 (X11; Linux x86_64; rv:64.0) Gecko/20100101
        Firefox/64.0</stringProp>
      </elementProp>
      <elementProp name="Accept" elementType="Header">
        <stringProp name="Header.name">Accept</stringProp>
        <stringProp name="Header.value">text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8</stringProp>
      </elementProp>
    </collectionProp>
  </HeaderManager>
  <hashTree/>
</hashTree>
<HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="11 /api/login"
      enabled="true">
  <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel"
      testclass="Arguments" enabled="true">
    <collectionProp name="Arguments.arguments">
      <elementProp name="email" elementType="HTTPArgument">
        <boolProp name="HTTPArgument.always_encode">true</boolProp>
        <stringProp name="Argument.value">m.sancheztarira@gmail.com</stringProp>
        <stringProp name="Argument.metadata">=</stringProp>
        <boolProp name="HTTPArgument.use_equals">true</boolProp>
        <stringProp name="Argument.name">email</stringProp>
      </elementProp>
      <elementProp name="password" elementType="HTTPArgument">
        <boolProp name="HTTPArgument.always_encode">true</boolProp>
```

```
        <stringProp name="Argument.value">123456</stringProp>
        <stringProp name="Argument.metadata">=</stringProp>
        <boolProp name="HTTPArgument.use_equals">true</boolProp>
        <stringProp name="Argument.name">password</stringProp>
        <stringProp name="HTTPArgument.content_type">password</stringProp>
       </elementProp>
      </collectionProp>
     </elementProp>
     <stringProp name="HTTPSampler.domain">localhost</stringProp>
     <stringProp name="HTTPSampler.port">3800</stringProp>
     <stringProp name="HTTPSampler.protocol">http</stringProp>
     <stringProp name="HTTPSampler.contentEncoding"></stringProp>
     <stringProp name="HTTPSampler.path">/api/login</stringProp>
     <stringProp name="HTTPSampler.method">POST</stringProp>
     <boolProp name="HTTPSampler.follow_redirects">false</boolProp>
     <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
     <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
     <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
     <boolProp name="HTTPSampler.image_parser">true</boolProp>
     <stringProp name="HTTPSampler.embedded_url_re"></stringProp>
     <stringProp name="HTTPSampler.connect_timeout"></stringProp>
     <stringProp name="HTTPSampler.response_timeout"></stringProp>
    </HTTPSamplerProxy>
    <hashTree>
     <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header Manager" enabled="true">
      <collectionProp name="HeaderManager.headers">
       <elementProp name="Referer" elementType="Header">
        <stringProp name="Header.name">Referer</stringProp>
        <stringProp name="Header.value">http://localhost/login</stringProp>
       </elementProp>
       <elementProp name="Accept-Language" elementType="Header">
        <stringProp name="Header.name">Accept-Language</stringProp>
        <stringProp name="Header.value">en-US,en;q=0.5</stringProp>
       </elementProp>
       <elementProp name="Origin" elementType="Header">
        <stringProp name="Header.name">Origin</stringProp>
        <stringProp name="Header.value">http://localhost</stringProp>
       </elementProp>
       <elementProp name="Access-Control-Request-Headers" elementType="Header">
        <stringProp name="Header.name">Access-Control-Request-Headers</stringProp>
        <stringProp name="Header.value">content-type</stringProp>
       </elementProp>
       <elementProp name="Access-Control-Request-Method" elementType="Header">
        <stringProp name="Header.name">Access-Control-Request-Method</stringProp>
        <stringProp name="Header.value">POST</stringProp>
       </elementProp>
       <elementProp name="Accept-Encoding" elementType="Header">
        <stringProp name="Header.name">Accept-Encoding</stringProp>
        <stringProp name="Header.value">gzip, deflate</stringProp>
       </elementProp>
       <elementProp name="User-Agent" elementType="Header">
```

```
        <stringProp name="Header.name">User-Agent</stringProp>
        <stringProp name="Header.value">Mozilla/5.0 (X11; Linux x86_64; rv:64.0) Gecko/20100101
        Firefox/64.0</stringProp>
       </elementProp>
       <elementProp name="Accept" elementType="Header">
        <stringProp name="Header.name">Accept</stringProp>
        <stringProp name="Header.value">text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8</stringProp>
       </elementProp>
      </collectionProp>
    </HeaderManager>
    <hashTree/>
  </hashTree>
  <HTTPSamplerProxy guiclass="HttpTestSampleGui" testclass="HTTPSamplerProxy" testname="31 /api/publications/1"
        enabled="true">
    <elementProp name="HTTPsampler.Arguments" elementType="Arguments" guiclass="HTTPArgumentsPanel"
        testclass="Arguments" enabled="true">
      <collectionProp name="Arguments.arguments"/>
    </elementProp>
    <stringProp name="HTTPSampler.domain">localhost</stringProp>
    <stringProp name="HTTPSampler.port">3800</stringProp>
    <stringProp name="HTTPSampler.protocol">http</stringProp>
    <stringProp name="HTTPSampler.contentEncoding"></stringProp>
    <stringProp name="HTTPSampler.path">/api/publications/1</stringProp>
    <stringProp name="HTTPSampler.method">GET</stringProp>
    <boolProp name="HTTPSampler.follow_redirects">true</boolProp>
    <boolProp name="HTTPSampler.auto_redirects">false</boolProp>
    <boolProp name="HTTPSampler.use_keepalive">true</boolProp>
    <boolProp name="HTTPSampler.DO_MULTIPART_POST">false</boolProp>
    <boolProp name="HTTPSampler.image_parser">true</boolProp>
    <stringProp name="HTTPSampler.embedded_url_re"></stringProp>
    <stringProp name="HTTPSampler.connect_timeout"></stringProp>
    <stringProp name="HTTPSampler.response_timeout"></stringProp>
  </HTTPSamplerProxy>
  <hashTree>
    <HeaderManager guiclass="HeaderPanel" testclass="HeaderManager" testname="HTTP Header Manager" enabled="true">
      <collectionProp name="HeaderManager.headers">
        <elementProp name="Referer" elementType="Header">
          <stringProp name="Header.name">Referer</stringProp>
          <stringProp name="Header.value">http://localhost/timeline</stringProp>
        </elementProp>
        <elementProp name="Accept-Language" elementType="Header">
          <stringProp name="Header.name">Accept-Language</stringProp>
          <stringProp name="Header.value">en-US,en;q=0.5</stringProp>
        </elementProp>……
```

En caso de necesitar el script de ejecución completo comunicarse con el autor de este documento.