

UNIVERSIDAD TÉCNICA DEL NORTE



Facultad de Ingeniería en Ciencias Aplicadas
Carrera de Ingeniería en Sistemas Computaciones

ESTUDIO COMPARATIVO SOBRE BASES DE DATOS NOSQL, QUE PERMITAN ANALIZAR LA VELOCIDAD DE RESPUESTA DE LOS DATOS EN UN PROTOTIPO DE RED SOCIAL UNIVERSITARIA

Trabajo de grado previo a la obtención del título de Ingeniero en Sistemas
Computacionales

Autor:

Bryan Lennin Cachimuel Loyo

Director:

Msc. Pablo Andrés Landeta López

Ibarra - Ecuador

Mayo 2020



UNIVERSIDAD TÉCNICA DEL NORTE

BIBLIOTECA UNIVERSITARIA

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
CÉDULA DE IDENTIDAD:	1003759634		
APELLIDOS Y NOMBRES:	CACHIMUEL LOYO BRYAN LENNIN		
DIRECCIÓN:	IBARRA – BARRIO AJAVÍ GRANDE AV. VICTOR MANUEL GUZMÁN Y PADRE ALBERTO ARO		
EMAIL:	blcachimuell@utn.edu.ec		
TELÉFONO FIJO:	062-606-147	TELÉFONO MÓVIL:	0967927800

DATOS DE LA OBRA	
TÍTULO:	ESTUDIO COMPARATIVO SOBRE BASES DE DATOS NOSQL, QUE PERMITAN ANALIZAR LA VELOCIDAD DE RESPUESTA DE LOS DATOS EN UN PROTOTIPO DE RED SOCIAL UNIVERSITARIA
AUTOR (ES):	CACHIMUEL LOYO BRYAN LENNIN
FECHA: DD/MM/AAAA	08/05/2020
SOLO PARA TRABAJOS DE GRADO	
PROGRAMA:	<input checked="" type="checkbox"/> PREGRADO <input type="checkbox"/> POSGRADO
TITULO POR EL QUE OPTA:	INGENIERO EN SISTEMAS COMPUTACIONALES
ASESOR /DIRECTOR:	Msc. PABLO LANDETA

2. CONSTANCIA

El autor (es) manifiesta (n) que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es (son) el (los) titular (es) de los derechos patrimoniales, por lo que asume (n) la responsabilidad sobre el contenido de la misma y saldrá (n) en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los ocho días del mes de mayo de 2020

EL AUTOR:

A handwritten signature in blue ink, appearing to read "Bryan Lennin Cachimuel Loyo". The signature is written in a cursive style with some loops and flourishes.

.....
BRYAN LENNIN CACHIMUEL LOYO

100375963-4

CERTIFICADO DEL DIRECTOR



UNIVERSIDAD TÉCNICA DEL NORTE FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

Ibarra, 08 de mayo de 2020

CERTIFICACIÓN DEL DIRECTOR

Por medio el presente yo MSc. Pablo Andrés Landeta López, certifico que le Sr. Bryan Lennin Cachimuel Loyo, portado de la cédula de identidad Nro. 1003759634. Ha trabajado en el desarrollo del proyecto de tesis “ESTUDIO COMPARATIVO SOBRE BASES DE DATOS NOSQL, QUE PERMITAN ANALIZAR LA VELOCIDAD DE RESPUESTA DE LOS DATOS EN UN PROTOTIPO DE RED SOCIAL UNIVERSITARIA”, previo a la obtención del título del Ingeniería en Sistemas Computacionales, lo cual ha realizado en su totalidad con responsabilidad.

Es todo en cuanto puedo certificar a la verdad.

Atentamente.

Msc. Pablo Andrés Landeta López

DIRECTO DE TESIS

AUTORÍA

Yo, BRYAN LENNIN CACHIMUEL LOYO, portador de la cédula de ciudadanía número 1003759634, declaro bajo juramento que el trabajo aquí descrito es de mi autoría, **ESTUDIO COMPARATIVO SOBRE BASES DE DATOS NOSQL, QUE PERMITAN ANALIZAR LA VELOCIDAD DE RESPUESTA DE LOS DATOS EN UN PROTOTIPO DE RED SOCIAL UNIVERSITARIA**, que no ha sido previamente presentada para ningún grado, ni calificación profesional, y que se ha respetado las diferentes fuentes y referencias.

A handwritten signature in blue ink, reading "Bryan Lennin Cachimuel Loyo". The signature is stylized and cursive, with a long horizontal stroke at the end.

BRYAN LENNIN CACHIMUEL LOYO

C.I 1003759634

DEDICATORIA

El trabajo de Grado que he desarrollado va dedicado principalmente para mi padre, madre y hermano ya que ellos han sido las personas que me han ayudado en todos los aspectos posibles durante mi vida universitaria. Con sus palabras de aliento me ayudaron muchas veces para salir adelante y no quedarme varado, es por ello que ustedes han sido la fuente de inspiración para poder realizar este proyecto y culminarlo, les estoy eternamente agradecido por todo lo me han ayudado hasta esta etapa, espero y estoy seguro de poder seguir cosechando triunfos y con ello ser un orgullo para ustedes.

Bryan Cachimuel

AGRADECIMIENTOS

A Dios por escuchar mis oraciones en todos estos años que he estado en la Universidad, de alguna u otra manera me ha enseñado que él nunca abandona y siempre está conmigo para ayudarme espiritualmente.

A mis padre Paco Cachimuel y mi madre Angélica Loyo que desde niño supieron guiarme por el buen camino, ustedes siempre se preocupan por mí y me ayudan en lo que sea necesario y de alguna manera u otra buscan la solución a los problemas, les agradezco por toda la dedicación que nos ponen a mi hermano y a mí ya que de esa manera nos han demostrado que siempre contaremos con ellos ya sea en la buenas y en las malas, muchas gracias por todo ello.

A mi mami Alicia que desde el cielo siempre me cuida y me protege sé que ya son muchos años desde que te fuiste, pero aun así te recuerdo con gran cariño ya que usted fue quien me cuido y me inculco que cuando tenga un problema ore al divino niño Jesús para que me ayude, es por ello por lo que siempre la recordare como uno de mis pilares fundamentales en mi niñez.

A mi hermanito Marlon Cachimuel, tú que as demostrado que un hermano también puede ser un amigo, me has dado muchas alegrías en mi vida con tus triunfos y logros y me has hecho dar cuenta de mis falencias y con ello superar mis debilidades, hermano espero que me sigas ayudando ya que tu muchas veces me has sacado una sonrisa cuando más la necesitaba, muchas gracias.

A mis amigos del team de nivelación: Cristoper, Nelson, Michelle y Franklin ustedes fueron mis primeros amigos en la universidad me han demostrado que la amistad es algo muy esencial y especial, me han ayudado en muchas cosas tanto académicas como personales de alguna u otra manera con la ayuda mutua que nos brindamos logramos cumplir nuestros propósitos.

A mis amigos Kevin R., Adriana, Josué, Maritza, Edgar, Katerin, Daysi, Evelin, Angelita y Kevin E, ustedes fueron los precursores y de la idea de qué, aunque somos pocos amigos pero de calidad, me enseñaron distintas cosas que me han ayudado a superar mis estigmas y forma de pensar, con sus cualidades y valores demostraron ser humildes y buenas personas, gracias por sus consejos y ayuda que me han dado.

A mi director de tesis Msc. Pablo Landeta que con sus conocimientos, consejos y paciencia me guio adecuadamente durante mi proceso de desarrollo de proyecto de tesis y con su ayuda logre culminar esté presente proyecto.

Finalmente tengo un agradecimiento especial para el Msc. Jorge Vásquez y Msc. Alexander Guevara quienes fueron los mejores docentes que me ayudaron en la primera etapa de mi proyecto de tesis que con su paciencia y experiencia me fundamentaron adecuadamente para iniciar mi tesis nunca cambien ingenieros y sigan enseñando tal como hacen ahora.

TABLA DE CONTENIDOS

AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE	II
CERTIFICADO DEL DIRECTOR.....	IV
AUTORÍA.....	V
DEDICATORIA	VI
AGRADECIMIENTOS	VII
RESUMEN	XIII
ABSTRACT	XIV
INTRODUCCIÓN.....	XV
Antecedentes	XV
Situación Actual.....	XV
Prospectiva.....	XVI
Planteamiento del Problema	XVII
Objetivos	XVII
Objetivo General.....	XVII
Objetivos Específicos.....	XVIII
Alcance	XVIII
Justificación.....	XVIII
Alcance	XIX
CAPÍTULO 1 MARCO TEÓRICO	1
1.1 Bases de Datos NoSQL y Bases de Datos SQL.....	1
1.1.1 Comparación entre las Bases de Datos NoSQL y SQL.....	3
1.1.2 Beneficios de las Bases de Datos NoSQL.....	4
1.1.3 La consistencia de los datos en las Bases de Datos NoSQL	5
1.2 Tipos de Bases de Datos NoSQL.....	6
1.2.1 Bases de Datos de clave-valor	6
1.2.2 Bases de Datos Documentales	7
1.2.3 Bases de Datos de Columnares.....	8
1.2.4 Bases de Datos de Grafos	9
1.3.1 Bases de Datos Apache Cassandra.....	11
1.3.2 Base de Datos MongoDB.....	13
1.3.3 Base de Datos Redis	15
1.4 Análisis Comparativo entre las Bases de Datos NoSQL	16
1.5 Metodología de desarrollo Scrum.....	17
1.5.1 Roles de Scrum	17
1.5.2 Eventos de Scrum.....	18
1.5.3 Artefactos de Scrum.....	19

1.6 Estándar ISO/IEC 25012.....	19
CAPÍTULO 2 DESARROLLO.....	22
2.1 Análisis de Requerimientos.....	22
2.1.1 Historias de Usuario.....	22
2.1.2 Product Backlog.....	30
2.1.3 Roles del proyecto	31
2.1.4 Roles del Develop team	31
2.1.5 Creación de los Sprints	32
2.1.6 Sprint 0.....	32
2.1.7 Sprint 1.....	33
2.1.8 Sprint 2.....	34
2.1.9 Sprint 3.....	36
2.1.10 Sprint 4.....	38
2.1.11 Sprint 5.....	41
2.2 Diseño	43
2.2.1 Arquitectura del Sistema Web.....	43
2.2.2 Diagrama del modelado de las bases de datos	44
2.3 Desarrollo del prototipo de red social	45
2.3.1 Sprint 1.....	46
2.3.2 Sprint 2.....	47
2.3.2 Sprint 3.....	51
2.3.2 Sprint 4.....	55
2.4 Pruebas.....	65
Capítulo III VALIDACIÓN DE RESULTADOS.....	69
3.1 Pruebas de Funcionamiento	69
3.2 Evaluación de la velocidad de respuesta de los datos	73
3.3 Análisis e Interpretación de resultados	80
3.4 Validación de Resultados.....	90
CONCLUSIONES.....	96
RECOMENDACIONES	97
GLOSARIO DE TÉRMINOS.....	98
REFERENCIAS BIBLIOGRÁFICAS.....	99

ÍNDICE DE FIGURAS

Fig. 1	Árbol de Problemas	XVII
Fig. 2	Bases de Datos NoSQL	1
Fig. 3	Bases de Datos SQL	3
Fig. 4	Bases de Datos clave-valor	6
Fig. 5	Ejemplo de bases de datos clave-valor	7
Fig. 6	Estructura de una base de datos documental	7
Fig. 7	Ejemplo de bases de datos documental	8
Fig. 8	Base de Datos columnares	8
Fig. 9	Ejemplo de bases de datos columnares	9
Fig. 10	Bases de Datos de grafos	10
Fig. 11	Ejemplo de bases de datos de grafos	11
Fig. 12	Arquitectura de Apache Cassandra	12
Fig. 13	Arquitectura de MongoDB	13
Fig. 14	Arquitectura de Redis	15
Fig. 15	Metodología Scrum	17
Fig. 16	ISO/IEC 25012	19
Fig. 17	Arquitectura del prototipo de red social	44
Fig. 18	Modelado de Datos de MongoDB	44
Fig. 19	Modelado de Datos de Apache Cassandra	45
Fig. 20	Modelado de Datos de Redis	45
Fig. 21	Configuración del Servidor	48
Fig. 22	Vista principal del prototipo de red social	48
Fig. 23	Vista de Registro de Usuarios	49
Fig. 24	Vista de Login de Usuarios	49
Fig. 25	Vista para Carga y despliegue de Imágenes	50
Fig. 26	Redirección hacia imágenes y etiquetas	50
Fig. 27	Comentarios en las imágenes	51
Fig. 28	Validación de campos de texto	52
Fig. 29	Validación de contraseñas	52
Fig. 30	Validación para el número de caracteres de contraseña	53
Fig. 31	Encriptación de contraseñas en MongoDB	53
Fig. 32	Encriptación de contraseñas en Apache Cassandra	53
Fig. 33	Sin disposición de dependencia para cifrado de contraseñas para Redis	54
Fig. 34	Validación de contraseñas en el Login	54
Fig. 35	Vista principal del prototipo de red social	55
Fig. 36	Lista de Contactos	56
Fig. 37	Ingreso de información de los Usuarios	56
Fig. 38	Despliegue de Información	57
Fig. 39	Ingreso de información para publicar	57
Fig. 40	Despliegue de las publicaciones	58
Fig. 41	Chat entre dos usuarios	59
Fig. 42	Chat Visto desde el Usuario Michelle Lara	59
Fig. 43	Chat visto desde el usuario Bryan Cachimuel	60
Fig. 44	Ingreso del usuario Nelson Cacoango al chat	60
Fig. 45	Chat privado visto desde el usuario Nelson Cacoango	61
Fig. 46	Chat visto desde el usuario Bryan Cachimuel	61
Fig. 47	Vista del chat de Michelle Lara	62
Fig. 48	Mejora de la vista principal al momento de Logearse	62
Fig. 49	Mejora de la vista de la carga de imágenes	63
Fig. 50	Mejora de la vista de publicaciones	63
Fig. 51	Mejora de la vista de contactos	64
Fig. 52	Mejora de la vista de ingreso de información	64
Fig. 53	Mejora de la vista de ingreso de información	76

Fig. 54 Muestra de MongoDB con mejor Rendimiento en mensajería.....	80
Fig. 55 Comparativa entre las variables Muestras, Media y Desviación Estándar en MongoDB.....	81
Fig. 56 Comparación entre las variables Muestras, Rendimiento y Error en MongoDB.....	82
Fig. 57 Comparativa entre las variables Muestras, Media y Desviación Estándar en Apache Cassandra.....	83
Fig. 58 Comparativa entre las variables Muestras, Rendimiento y Error en Apache Cassandra.....	83
Fig. 59 Comparativa entre las variables Muestras, Media y Desviación Estándar en Redis.....	84
Fig. 60 Comparación entre las variables Muestras, Rendimiento y Error en Redis.....	85
Fig. 61 Comparación de las variables Muestras, Media y Desviación Estándar en MongoDB.....	86
Fig. 62 Comparación de las variables Muestras, Rendimiento y Error en MongoDB.....	86
Fig. 63 Comparativa entre las variables Muestras, Media y Desviación Estándar en Apache Cassandra.....	88
Fig. 64 Comparativa entre las variables Muestras, Rendimiento y Error en Apache Cassandra.....	88
Fig. 65 Comparativa entre las Variables Muestras, Media y Desviación Estándar en Redis.....	89
Fig. 66 Comparativa entre las variables Muestras, Rendimiento y Error en Redis.....	90
Fig. 67 Matriz de Correlación de Pearson de la Base de Datos MongoDB.....	92
Fig. 68 Matriz de Correlación de Pearson de la Base de Datos Apache Cassandra.....	92
Fig. 69 Matriz de Correlación de Pearson de la Base de Datos Redis.....	92
Fig. 70 Resultado de Correlación Muestra-Rendimiento.....	93
Fig. 71 Resultado de Correlación Rendimiento-Error.....	93
Fig. 72 Resultado de Correlación Muestra-Rendimiento.....	94
Fig. 73 Resultado de Correlación Rendimiento-Error.....	94
Fig. 74 Resultado de Correlación Muestra-Rendimiento.....	95
Fig. 75 Resultado de Correlación Rendimiento-Error.....	95

ÍNDICE DE TABLAS

TABLA 1 COMPARACIÓN ENTRE LAS BASES DE DATOS NOSQL Y SQL.....	4
TABLA 2 INSERCIÓN DE DOCUMENTACIÓN MEDIANTE TUBERÍA DE AGREGACIÓN.....	14
TABLA 3 INSERCIÓN DE DOCUMENTACIÓN MEDIANTE MAPEO DE REDUCCIÓN.....	15
TABLA 4 COMPARACIÓN DE LAS BASES DE DATOS A ESTUDIARSE.....	16
TABLA 5 HISTORIA DE USUARIO RSB-001.....	22
TABLA 6 HISTORIA DE USUARIO RSB-002.....	23
TABLA 7 HISTORIA DE USUARIO RSB-003.....	23
TABLA 8 HISTORIA DE USUARIO RSB-004.....	24
TABLA 9 HISTORIA DE USUARIO RSB-005.....	24
TABLA 10 HISTORIA DE USUARIO RSB-006.....	25
TABLA 11 HISTORIA DE USUARIO RSB-007.....	25
TABLA 12 HISTORIA DE USUARIO RSB-008.....	26
TABLA 13 HISTORIA DE USUARIO RSB-009.....	26
TABLA 14 HISTORIA DE USUARIO RSB-010.....	27
TABLA 15 HISTORIA DE USUARIO RSB-011.....	27
TABLA 16 HISTORIA DE USUARIO RSB-012.....	28
TABLA 17 HISTORIA DE USUARIO RSB-013.....	28
TABLA 18 HISTORIA DE USUARIO RSB-014.....	29
TABLA 19 HISTORIA DE USUARIO RSB-015.....	29
TABLA 20 PRODUCT BACKLOG.....	30
TABLA 21 ROLES DEL PROYECTO.....	31
TABLA 22 ROLES DEL DEVELOP TEAM.....	31
TABLA 23 SPRING BACKLOG.....	32
TABLA 24 RETROSPECTIVA SPRINT 0.....	33
TABLA 25 SPRINT BACKLOG.....	33
TABLA 26 RETROSPECTIVA DEL SPRINT 1.....	34
TABLA 27 SPRING BACKLOG DEL SPRINT 2.....	35
TABLA 28 RETROSPECTIVA DEL SPRINT 2.....	36

TABLA 29 SPRING BACKLOG DEL SPRINT 3	37
TABLA 30 RETROSPECTIVA DEL SPRINT 3	38
TABLA 31 SPRING BACKLOG DEL SPRINT 4	39
TABLA 32 RETROSPECTIVA DEL SPRINT 4	41
TABLA 33 SPRING BACKLOG DEL SPRINT 5	41
TABLA 34 RETROSPECTIVA DEL SPRINT 5	42
TABLA 35 SPRINT 1	46
TABLA 36 SOFTWARE ESPECIALIZADO PARA EL DESARROLLO	46
TABLA 37 SPRINT 2	47
TABLA 38 SPRINT 3	51
TABLA 39 SPRINT 4	55
TABLA 40 SPRINT 5	58
TABLA 41 PRUEBA 1	65
TABLA 42 PRUEBA 2	65
TABLA 43 PRUEBA 3	66
TABLA 44 PRUEBA 4	67
TABLA 45 PRUEBA 5	68
TABLA 46 VALORES DE MONGODB EN REGISTROS DE USUARIOS	70
TABLA 47 VALORES DE APACHE CASSANDRA EN REGISTROS DE USUARIOS	70
TABLA 48 VALORES DE REDIS EN REGISTROS DE USUARIOS	71
TABLA 49 VALORES DE MONGODB EN MENSAJERÍA	71
TABLA 50 VALORES DE APACHE CASSANDRA EN MENSAJERÍA	72
TABLA 51 VALORES DE REDIS EN MENSAJERÍA	73
TABLA 52 SELECCIÓN DE VARIABLES PARA LA EVUALUACIÓN DE VELOCIDAD DE DATOS CON MONGODB	74
TABLA 53 SELECCIÓN DE VARIABLES PARA LA EVALUACIÓN DE VELOCIDAD DE DATOS CON APACHE CASSANDRA	74
TABLA 54 SELECCIÓN DE VARIABLES PARA LA EVALUACIÓN DE VELOCIDAD DE DATOS CON REDIS	75
TABLA 55 COMPARATIVA DEL RENDIMIENTO DE LAS TRES BASES DE DATOS NOSQL	76
TABLA 56 SELECCIÓN DE VARIABLES PARA LA EVALUACIÓN DE LA VELOCIDAD DE DATOS CON MONGODB	77
TABLA 57 SELECCIÓN DE VARIABLES PARA LA EVALUACIÓN DE LA VELOCIDAD DE DATOS CON APACHE CASSANDRA	78
TABLA 58 SELECCIÓN DE VARIABLES PARA LA EVALUACIÓN DE LA VELOCIDAD DE DATOS CON REDIS	78
TABLA 59 COMPARATIVA DEL RENDIMIENTO DE LAS TRES BASES DE DATOS NOSQL, PARA LA MENSAJERÍA	79
TABLA 60 SELECCIÓN DE VARIABLES PARA EL ANÁLISIS DE LOS DATOS EN MONGODB	80
TABLA 61 SELECCIÓN DE VARIABLES PARA EL ANÁLISIS DE LOS DATOS EN APACHE CASSANDRA	82
TABLA 62 SELECCIÓN DE VARIABLES PARA EL ANÁLISIS DE LOS DATOS EN REDIS	84
TABLA 63 SELECCIÓN DE VARIABLES PARA EL ANÁLISIS DE LO DATOS CON MONGOGO, PARA LA MENSAJERÍA	85
TABLA 64 SELECCIÓN DE VARIABLES PARA EL ANÁLISIS DE DATOS EN APACHE CASSANDRA, PARA LA MENSAJERÍA	87
TABLA 65 SELECCIÓN DE VARIABLES PARA EL ANÁLISIS DE DATOS EN REDIS, PARA LA MENSAJERÍA	89
TABLA 66 P-VALUE DE LAS TRES BASES DE DATOS	91

RESUMEN

El enfoque que sostiene este estudio es la comparativa entre Apache Cassandra, MongoDB y Redis, las tres bases de datos son alternativas que proporcionan un almacenamiento y tratamiento de información muy diferente a lo que son las bases de datos relacionales, es decir, que las bases de datos en estudio cuentan con las herramientas y estructuras necesarias para trabajar con grandes cantidades de información de una manera eficiente y eficaz, de tal manera que, al contar con los suficientes recursos a nivel software son ideales para trabajar con lo que es la Big Data.

Las bases de datos NoSQL proporcionan ciertas facilidades para su uso y al ser open source en ciertos casos ayudan a los desarrolladores a probar sus funcionalidades y sus diferentes entornos de almacenamiento, es por ello que en este trabajo se enfatizó en la creación de tres prototipos de red social que están conectados con Apache Cassandra, MongoDB y Redis, y con ello, verificar y analizar cómo las tres bases de datos pueden llegar a tener diferencias al almacenar los registros de usuarios y los mensajes que se enviarán entre los mismos usuarios registrados. De esta manera al finalizar el proceso de desarrollo se opta por realizar las respectivas pruebas de stress para verificar cómo cada una de las bases de datos NoSQL puede trabajar bajo presión de peticiones hacia el servidor y cuál es su calidad de respuesta ante los usuarios.

Al finalizar el proceso de pruebas se realizaron los respectivos análisis de resultados, con los cuales se pudo concluir cuál de las tres bases de datos era la apropiada para manejar lo que es la velocidad de respuesta de los datos ante las peticiones y consultas que realicen los usuarios, como también, se realizó la interpretación de los resultados mediante métodos estadísticos que ayudaron a encontrar la relación existente entre las variables cuantitativas obtenidas durante la fase de pruebas.

ABSTRACT

The focus of this study is the comparison between Apache Cassandra, MongoDB and Redis, the three databases are alternatives that provide different information storage and treatment than relational databases, that is, the databases under study count with the tools and structures necessary to work with large amounts of information in an efficient and effective way, in such a way that having sufficient resources at the software level are ideal for working with what Big Data is.

NoSQL databases provide certain facilities for their use and being open source in certain cases they help developers to test their functionalities and their different storage environments, which is why in this work the emphasis was on creating three prototypes of social network that are connected with Apache Cassandra, MongoDB and Redis and thereby verify and analyze how the three databases can have differences when storing user records and messages that are sent between the same registered users. In this way, at the end of the development process, it is chosen to carry out the respective stress tests to verify how each of the NoSQL databases can work under pressure from requests to the server and what is its quality of response to users.

At the end of the testing process, the respective analysis of results was carried out with which it was possible to conclude which of the three databases was appropriate to handle what is the speed of response of the data to requests and queries made by users. , as well as the interpretation of the results using statistical methods that helped to find the relationship between the quantitative variables obtained during the testing phase.

INTRODUCCIÓN

Antecedentes

El concepto como paradigma de Bases de Datos No Relacionales comenzaron a originarse en el año de 1998 y no fueron creadas por una empresa específica, sino que, fueron concebidas por diferentes empresas y grupos independientes que buscaban soluciones específicas a sus problemas como: bajo rendimiento en el procesamiento masivo de transacciones, dificultades con el almacenamiento de información, baja flexibilidad y escalabilidad. La aplicabilidad de las Bases de Datos NoSQL surge a causa del cambio que se ha dado en el manejo de la información, es así como las empresas han buscado soluciones para manejar este cambio y poder administrar y sacarle provecho a los volúmenes de información que ellos manejan y con ello ofrecer a sus usuarios un mejor servicio y disponibilidad de acceso a información como también proporcionar alta velocidad operativa para que los desarrolladores opten por trabajar con ellas de una manera adecuada y ajustándose a las necesidades en lo que es gestión de la información.(Reche,2016).

Las Instituciones de educación superior generan gran cantidad de información de tipo académico como documentos, tareas, evaluaciones, recursos de audio o video, imágenes de alta resolución, documentos tipo: pdf, word, excel, point y no es la excepción la Universidad Técnica del Norte dispone de sistemas de información robustos los cuales presentan una gestión baja para el control de diferentes datos, así mismo manejan sistemas de aprendizaje online como Moodle que almacenan información de diferente tipo, por ello su gestión no es adecuada, tarda mucho tiempo en procesar y almacenar la información y por ende llegan a generarse problemas ya que las bases de datos con las que trabajan comienzan a disminuir su rendimiento y escalabilidad lo que provoca deficiencia y fallos en los proceso de gestión tanto de almacenamiento y funcionamiento a nivel del sistema. (Torres, 2016).

Situación Actual

Actualmente en el sector académico de Imbabura se desarrollan sistemas y aplicaciones con Bases de Datos Relacionales (BDR) ya que los administradores de bases de datos (DBA) eligen diseñar sus soluciones sobre estructuras de datos relacionales, además cuentan con amplia experiencia y suficientes conocimientos para gestionar el proceso de almacenamiento e interconexión de los sistemas que vayan a desarrollar con las bases de datos (Alarcon, 2014), es por ello que los DBA al conocer los procesos correspondientes, continúan el desarrollo de sus proyectos con BDR y con esto excluyen alternativas tecnológicas que los pueda llegar a beneficiar a futuro.

En la Universidad Técnica del Norte en los últimos tres años se ha logrado identificar que el 5,87% de la comunidad académica ha crecido (estudiantes, docentes, investigadores, personal administrativo y otros trabajadores) y los integrantes de la misma demandan el uso de más almacenamiento para resguardar sus proyectos, investigaciones entre otra información (Universidad Técnica del Norte, 2018), esto ha provocado que las Bases de Datos (BD) con las cuales trabajan en el sector académico lleguen a tener dificultades debido a que no soportan controlar grandes volúmenes de datos, porque su estructura de almacenamiento puede llegar a tener conflictos cuando se presenta aglomeración de información, es por esto que los DBA optan por contratar o emplear otras herramientas como: bases de datos privadas (ORACLE, Microsoft SQL Server, Paradox, IBMDB2) que mejoren tanto la calidad del servicio como el almacenamiento y gestión de la información al realizar esto los ayudaría a solucionar sus problemas, pero con el inconveniente de que consumirían más recursos económicos de que lo que se había planificado para ese sector. (Graterol, 2017).

Según el Cuadrante Mágico de Gartner las Bases de Datos NoSQL están ganando terreno en lo que se refiere a control de información ya que hasta el momento una de las populares bases de datos NoSQL, MongoDB se encuentra ubicada entre las 10 mejores bases de datos. (Jimenez, 2017).

Prospectiva

De acuerdo con la selección de tecnología de almacenamiento adecuado involucra la consideración de numerosos aspectos, aunque el rendimiento suele ser el factor más importante, es necesario considerar ciertos aspectos como la funcionalidad, la facilidad de operación, sencillez de uso, disponibilidad de recursos para el trabajo continuo, los cuales son requerimientos que se necesita para solventar las necesidades de distribución de información, gestión y procesamiento de información. (Torre, 2018).

Es por ello que se ha visto la necesidad de implementar un estudio sobre las bases de datos NoSQL en el sector académico de Imbabura y especialmente en la Universidad Técnica del Norte, con el fin de indagar y comprobar si en la actualidad este sector se ve conforme con la manera de trabajar de las bases de datos relacionales y si en algún momento han considerado optar por la utilización de otras herramientas tecnológicas (Bases de Datos) que los puedan ayudar a mejorar su calidad de servicio.

Al realizar esta indagación se podrá recolectar información necesaria para determinar si es viable el uso de Bases de Datos NoSQL en los sistemas actuales y dicha información plasmarla en el estudio que se desarrollará, en donde se dará a conocer los resultados de la

investigación como también se optará por informar sobre las ventajas y oportunidades que se tiene al implementar este tipo de Bases de Datos.

Planteamiento del Problema

La falta de un estudio sobre las bases de datos NoSQL, puede traer inconvenientes como: los desarrolladores al no conocer sobre este tipo de bases de datos, optan por utilizar las tradicionales bases de datos relacionales y excluyen otras opciones que podrían servirles de mejor manera y les traería beneficios en sus proyectos a realizarse y con menos uso de recursos económicos para el sector donde trabajen.

A continuación, en la Fig.1 se indica el problema principal que se tiene al utilizar bases de datos relacionales con sus respectivas causas y efectos dentro de lo que es el almacenamiento de información.

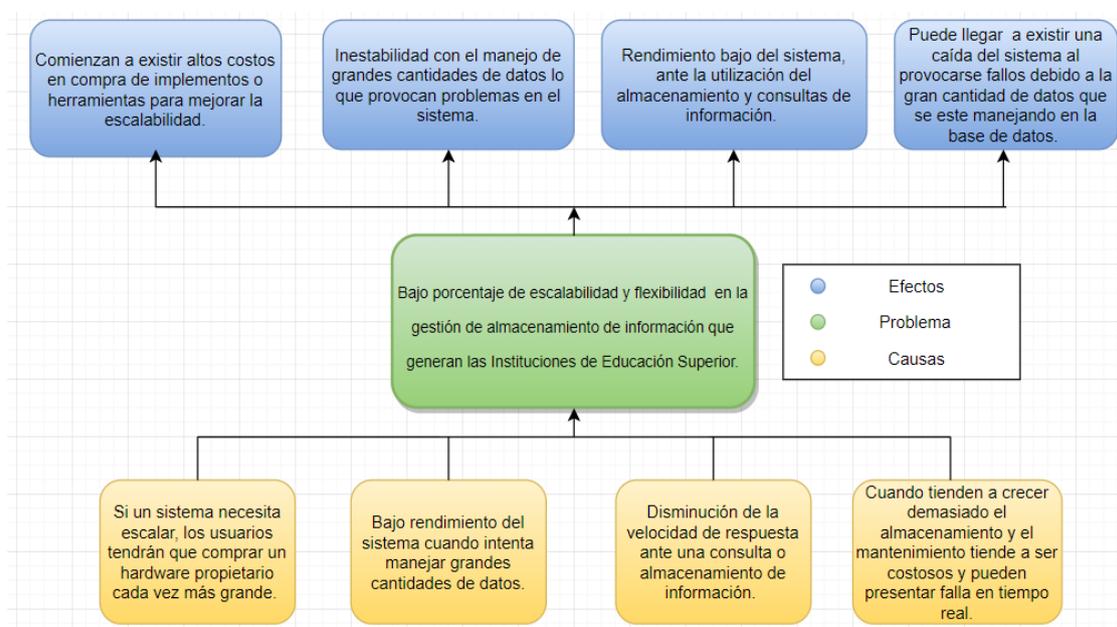


Fig. 1 Árbol de Problemas

Objetivos

Objetivo General

Realizar un estudio comparativo sobre Bases de Datos NoSQL que permitan analizar la velocidad de respuesta de los datos en un prototipo de red social universitaria.

Objetivos Específicos.

1. Desarrollar un marco teórico basado en la investigación de las Bases de Datos NoSQL Cassandra, MongoDB y Redis.
2. Evaluar la velocidad de respuesta de los datos dentro del prototipo de red social académica.
3. Desarrollar un prototipo de red social académica que permita almacenar e intercambiar información entre estudiantes.
4. Validar los resultados.

Alcance

Se desarrollará en este proyecto un prototipo de red social universitaria que contará con la implementación de tres Bases de Datos NoSQL (Apache Cassandra, MongoDB y Redis), a las cuales se las evaluará mediante métricas para verificar la velocidad de respuesta de los datos, los cuales serán valorados mediante el estándar ISO/IEC 25012 que se encargara de determinar la capacidad de las características de los datos de tener el potencial intrínseco para satisfacer la necesidades explícitas o implícitas del prototipo de red social. (Rodríguez, 2015).

El objetivo principal de este estudio es establecer una comparativa entre las bases de datos NoSQL para poder identificar la más apropiada que cuente con las características necesarias para poder utilizarla como una herramienta de comunicación entre estudiantes para ello la red social contará con un proceso de registro para crear una cuenta dentro de ella y con ello poder realizar un chat en donde se pueda enviar mensajes con el fin de comprobar que las bases de datos NoSQL cumplen un mejor desempeño en lo que respecta a control de la gestión de la información.

Justificación

Las razones para implementar este proyecto es para investigar y recolectar información, para validarla mediante una ecuación o método que revele el impacto de las bases de datos NoSQL en el sector académico de la Provincia de Imbabura y especialmente en la Universidad Técnica del Norte, como también dar a conocer la importancia de estas bases de datos en el ámbito de desarrollo de sistemas modernos, en donde se dará a conocer las ventajas que brindan estas bases de datos como son: el control y manipulación de datos debido a sus estructura, performance, escalabilidad, flexibilidad, etc (Rodríguez, Rodríguez & Díaz, 2016), todas estas ventajas hacen notar que este tipo de bases de datos sean las indicadas para trabajar en proyectos que controlen gran cantidad de información.

El presente proyecto tiene un enfoque hacia los siguientes objetivos de desarrollo sostenible:

Nº 9: Construir infraestructuras resilientes, promover la industrialización inclusiva y sostenible y fomentar la innovación.

9.b Apoyar el desarrollo de tecnologías, la investigación y las innovaciones nacionales en los países en desarrollo, incluso garantizando un entorno normativo propicio a la diversificación industrial y la adición de valor a los productos básicos, entre otras cosas. (Medina, 2015)

Este proyecto está justificado por la falta indagaciones y de implementación de estudios sobre las Bases de Datos NoSQL, de acuerdo a lo investigado en la localidad no se han hecho estudios del tema, pero existen compañías en el extranjero que ya han utilizado estas bases para el uso de grandes cantidades de información y en algunos casos estas empresas han desarrollado sus propias bases de datos no relacionales como son: Amazon, Facebook, Twitter entre otras compañías, que han optado por estas bases de datos debido a su escalabilidad y flexibilidad con los datos que trabajan. (Gómez & Maqueda, 2014).

Económico

El impacto a nivel económico implica disminuir gastos que se pueden presentar al utilizar las Bases de Datos tradicionales ya que las bases de datos NoSQL son en algunos casos open source lo que ayudaría a eliminar algunos procesos económicos innecesarios dentro del sector académico que decida cambiar de proceso de almacenamiento.

Social

Al realizar el estudio se va a poder contar con un impacto en la parte social ya que este estudio dará a conocer a los desarrolladores, todos los beneficios que tendrían al trabajar con estas bases de datos no relacionales y como podrían implementarlas en sus proyectos.

Alcance

Se desarrollará en este proyecto un prototipo de red social universitaria que contará con la implementación de tres Bases de Datos NoSQL (Apache Cassandra, MongoDB y Redis), a las cuales se las evaluará mediante métricas para verificar la velocidad de respuesta de los datos, los cuales serán valorados mediante el estándar ISO/IEC 25012 que se encargara de determinar la capacidad de las características de los datos de tener el potencial intrínseco para satisfacer la necesidades explicitas o implícitas del prototipo de red social. (Rodríguez, 2015).

El objetivo principal de este estudio es establecer una comparativa entre las bases de datos NoSQL para poder identificar la más apropiada que cuente con las características necesarias para poder utilizarla como una herramienta de comunicación entre estudiantes y para ello la red social contará con un proceso de registro para crear una cuenta dentro de ella y con ello poder realizar un chat en donde se pueda enviar mensajes con el fin de comprobar que las bases de datos NoSQL cumplen un mejor desempeño en lo que respecta a control de la gestión de la información.

CAPÍTULO 1

Marco Teórico

1.1 Bases de Datos NoSQL y Bases de Datos SQL

¿Qué son las Bases de Datos NoSQL?

En la actualidad los sistemas web y aplicaciones han cambiado drásticamente en el mundo del internet de las cosas, y debido a ello en los últimos quince años se han realizado cambios en las empresas ya que han tenido la obligación de automatizar todos sus procesos y reestructurar el manejo de su información e infraestructura tecnológica, es por ello que en la última década se ha venido mencionando lo que es Big Data, Big Users y Cloud Computing , dichas tecnologías controlan grandes cantidades información y por esta razón las mismas están teniendo como opción trasladar toda su información hacia las llamadas bases de datos NoSQL ya que ellas poseen herramientas y estructuras adecuadas para simplificar operaciones a gran escala. A demás se puede enfatizar que muchos desarrolladores opinan que es factible utilizar el tipo de modelo de datos que es sin esquema ya que es óptimo para controlar la gran variedad de datos que se procesan hoy en día dentro de las empresas. (Ali, 2017).

En la Fig. 2 se muestra la estructura que manejan las bases de datos NoSQL.

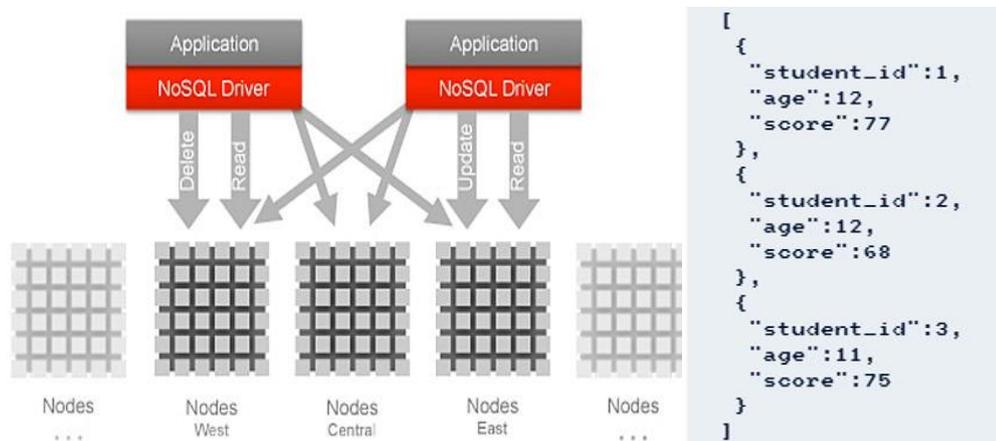


Fig. 2 Bases de Datos NoSQL

De acuerdo con su estructura y modo de operación con la información las bases de datos NoSQL no cumplen en su totalidad con las características de atomicidad, consistencia, aislamiento y durabilidad (ACID) y debido a ello escalan horizontalmente, por lo que proporcionan de esta manera eficiencia para ser utilizadas en entornos distribuidos. (Sarasa, 2016).

En cuanto a ventajas respecta se centran en aspectos relevantes para mejorar el almacenamiento, control de los datos y la productividad de los sistemas web y aplicaciones, de esta manera se podría citar las siguientes ventajas. (Sarasa, 2016):

- a) Puede resolver problemas de escalamiento de los grandes volúmenes de datos estructurados y no estructurados y a la vez controlar que el rendimiento que se proporciona sea el adecuado para la disponibilidad de la información almacenados.
- b) Los costos por el uso son reducidos ya que son de tipo open source (código abierto) y pueden llegar a ofertarse para arquitecturas escalables y dinámicas.
- c) Su operación sin esquemas puede ayudar a añadir diferentes tipos de datos a los registros sin especificar su estructura, lo que es benéfico para los administradores de bases de datos cuando ingresen información estructurada y no estructurada en los diferentes campos personalizados.
- d) La productividad en el desarrollo de aplicaciones contribuye a que los modelos de datos se adapten a las necesidades que tengan los sistemas web y aplicaciones modernas y con ello ayudar a simplificar las interacciones y contribuir al desarrollo y evolución de los sistemas web.

Bases de Datos SQL: Son también llamadas bases de datos relacionales aquellas que están compuestas por un conglomerado de información relacionada entre sí, debido a sus características y prestaciones son consideradas como un sistema de almacenamiento de datos integrado que puede ser manipulado mediante el uso de una serie de aplicaciones que cuenten con la estructura y herramientas adecuadas como para manejar la información que se encuentren dentro de la base de datos. (Silberschatz, Korth & Sudarshan, 2010).

En cuanto al modelo de datos puede ser utilizado para resolver problemas y administrar datos de forma dinámica, en donde su enfoque principal sea obtener relaciones entre entidades y exista una correcta normalización sin redundancia. (Silberschatz, Korth & Sudarshan, 2010).

En la Fig. 3 se muestra la estructura con la que cuentan las bases de datos relacionales.

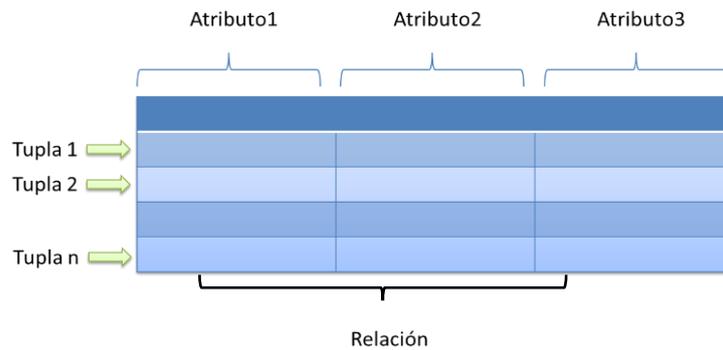


Fig. 3 Bases de Datos SQL

Las bases de datos relacionales pueden presentar ciertas características que las hacen favorables para su uso dentro de lo que es el desarrollo de sistemas web y aplicaciones y se las puede presentar de la siguiente manera. (Yue, Yuming, Yuefei, & Cungen, 2010):

- a) Dentro de las bases de datos relacionales la información puede ser tratada como como una colección de relaciones.
- b) Se cuenta con una variedad de métodos que permiten acceder a la información con la que cuente una base de datos.
- c) Enfatizan la organización de un conjunto de tablas con filas y columnas, en donde los datos se encuentran visibles y estrictamente organizados como tablas de valores.
- d) Se hace uso de operaciones de algebra relacional para poder modificar o combinar el contenido de la información de las bases de datos.

Dentro de la estructura las tablas son las encargadas de almacenar la información de los objetos que se representaran en la base de datos, mientras que sus componentes como las columnas almacenan un determinado tipo de dato con su respectivo valor de acuerdo con el atributo. En cuanto a las filas representan la reunión de los valores relacionados de un objeto. Cabe mencionar que en cada fila se podría remarcar un identificador que se lo podría representar como una clave principal y dentro de la fila se encontraría un atributo que puede ser identificado como una clave foránea. (Yue, Yuming, Yuefei, & Cungen, 2010).

1.1.1 Comparación entre las Bases de Datos NoSQL y SQL

De acuerdo con las investigaciones realizadas, se puede determinar que existen diferencias entre los dos tipos de bases de datos como puede ser su esquema, estructura, forma de realizar consultas hacia la base de datos y forma de almacenamiento las cuales se describirán en la siguiente tabla.

TABLA 1
COMPARACIÓN ENTRE LAS BASES DE DATOS NOSQL Y SQL

Bases de Datos NoSQL	Bases de Datos SQL
Posee un esquema dinámico.	Posee un esquema estático.
Posee una estructura flexible, en donde no es necesario definir una estructura de datos.	Posee una estructura bien definida de datos que tienen que ser uniformes, las propiedades de estos datos pueden especificarse por adelantado.
Su almacenamiento puede darse en colecciones de tipo clave-valor, documentos, grafos, etc.	La información es almacenada en tablas y por ende son reconocidas como bases de datos basadas en tablas.
Posee un escalamiento horizontal.	Posee un escalamiento vertical.
Para realizar consultas utiliza colecciones de documentos, este tipo de consultas se las conoce como lenguaje desestructurado.	Para realizar consultas utiliza un lenguaje estructurado llamado SQL que contiene códigos para manipular los datos de una base de datos.
Es utilizado cuando se necesita efectuar consultas sencillas.	Es utilizado cuando se necesita efectuar consultas complejas.
Puede llegar a perderse la integridad en las transacciones.	Utilizan ACID.
De acuerdo con su estructura no puede haber relaciones entre las colecciones, no obstante, puede depender del modelado de datos.	De acuerdo con su estructura pueden existir relaciones entre tablas las cuales debe estar ser referenciadas de forma metódica.

Fuente: Adaptado de (Silberschatz, Korth & Sudarshan, 2010)

1.1.2 Beneficios de las Bases de Datos NoSQL

Al contar con una estructura específica promueven beneficios importantes que hacen que se cumplan ciertas características apropiadas para ayudar a los sistemas modernos con el almacenamiento de datos estructurados y no estructurados, de esta manera dichos beneficios pueden ser los siguientes. (Castro, Romero, González, & Callejas Cuervo, 2012)

- a) Escalabilidad: Orientado hacia el diseño escalar de clústeres distribuidos donde los nodos se crean de acuerdo con la disponibilidad en que los usuarios necesiten utilizar el almacenamiento en la base de datos, permitiendo de esta manera que se elimine los costos y la complejidad de la fragmentación manual.
- b) Alto Rendimiento: Optimizado mediante el uso de modelos de datos que se enfocan al control de estructuras documentales, clave-valor y gráficos los cuales mediante patrones dan un acceso confiable y un rendimiento proporcional hacia la información que se esté almacenando.
- c) Control de datos con flexibilidad: Ofrece libertad para los administradores de bases de datos en el ámbito de velocidad y flexibilidad en el cambio de esquemas y consultas.
- d) Desarrollo con agilidad: Da soluciones a situaciones imprevistas ya que las bases de datos NoSQL satisfacen de una manera rápida y ágil a las consultas efectuadas para obtener información en grandes cantidades.

1.1.3 La consistencia de los datos en las Bases de Datos NoSQL

La consistencia radica en trabajar bajo un enfoque, en donde se plantea diferentes problemas que se han registrado en la seguridad de los datos, es por ello que se da a conocer cuatro tipos de consistencias que se presentan en las bases de datos NoSQL y son las siguientes: (Sarasa, 2016).

- a) **Consistencia en las escrituras:** Se presenta cuando existe un problema de escritura-escritura en donde varios usuarios realizan el proceso de actualización de un mismo dato al mismo tiempo, es decir que al realizar la actualización por parte de los usuarios las órdenes llegan al servidor y la serializan y vuelven a enviar la respuesta de una manera ordenada es allí donde radica el problema ya que un usuario recibe el dato actualizado mientras que al otro usuario se le produce una pérdida de actualización debido a que la última escritura que se aplica se sobrescribe sobre la primera.

Una solución que se ha planteado es tener bloqueos de escritura en donde si se desea realizar la actualización de un dato este se bloquee temporalmente hasta que el usuario termine de actualizarlo y después de realizar este proceso el dato dejará de estar bloqueado y estará disponible para ser utilizado por otros usuarios.

- b) **Consistencia en las lecturas:** Se presenta cuando existe un problema de lectura-escritura en donde un usuario realiza la lectura de un dato en medio de la escritura de otro usuario que en ese momento se encuentra leyendo o escribiendo en un dato, es por esto que sea planteado que las bases de datos relacionales apliquen las transacciones en donde se asegure que el usuario solo realice la lectura y luego la escritura o en su defecto la escritura y luego la lectura. (Sarasa, 2016).

Este concepto afirma que las Bases de Datos NoSQL no soportan transacciones lo que es incorrecto ya que las bases de datos orientadas a grafos pueden llegar a soportar cierto tipo de transacciones, como también las Bases de datos orientadas hacia agregados que puede soportar actualizaciones atómicas sobre un agregado. (Sarasa, 2016).

- c) **Consistencia de sesión:** Puede llegar a ser un problema las ventanas de inconsistencia ya que casi siempre se llega a mantener una carga que se encuentra ejecutándose en un cluster que mantenga la carga de peticiones a los diferentes nodos donde sea solicitada una ventana, para controlar esta clase de situaciones es necesario implementar un tipo de consistencia de tipo leer lo que se escribe, lo que implica aplicar el concepto de actualización en donde se pueda verificar que solamente se pueda ver la actualización ejecutada por el usuario.

1.2 Tipos de Bases de Datos NoSQL

Con la evolución de las Bases de Datos NoSQL se han ido creando diferentes tipos, en donde cada una representa y se adecua a las diferentes necesidades que se han ido encontrando en los sistemas web y aplicaciones actuales, es por ello por lo que se ha podido clasificar y definir de la siguiente manera.

1.2.1 Bases de Datos de clave-valor

La esquematización de este tipo de bases de datos presenta un modelo de datos donde la clave puede ser generada de una manera sintética y con diferente formato, pero con la condición de que la clave sea única para cada dato que sea registrado.

Los valores para almacenar pueden tener una estructura sencilla con formatos comunes como string, int, float, double, JSON como también otro tipo de estructuras complejas. Estas bases de datos pueden hacer uso de operaciones básicas como para obtener datos asociados a una clave (get), asociar un valor a una clave (put) y borrar una entrada con una clave específica. (delete) (Bosonit, 2018). En la Fig. 4 se hace una referencia a la estructura del modelo en lo que es clave-valor.



Fig. 4 Bases de Datos clave-valor

Se presentan ciertas ventajas como la simplicidad, eficiencia y flexibilidad que permiten realizar la búsqueda de la información almacenada de una manera rápida y concisa mientras se realiza la lectura a toda la base de datos. Cabe mencionar que la simplicidad puede llegar a ocasionar inconvenientes como por ejemplo que al carecer de una estructura específica no sería posible realizar consultas con queries ya que las bases de datos solo constan de colecciones lo cual puede llegar a complicar la implementación de modelos complejos. (Bosonit, 2018).

Para comprender como está estructurada se presenta en la Fig. 5 un ejemplo donde se puede apreciar las respectivas clave y valor que se maneja dentro de estas bases de datos.

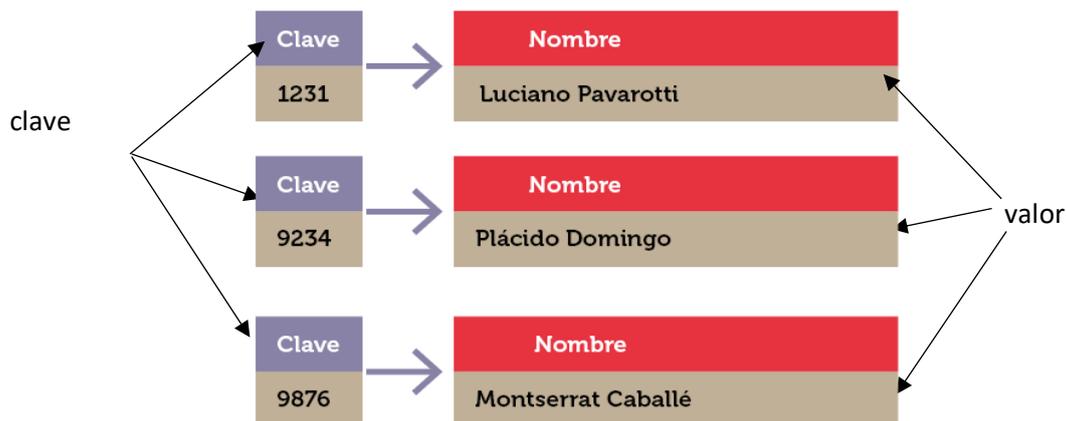


Fig. 5 Ejemplo de bases de datos clave-valor

1.2.2 Bases de Datos Documentales

Poseen cierta similitud con las bases de datos clave-valor pero con la diferencia de que estas tienen un nivel superior en el manejo de metadatos y que permiten la utilización de documentos donde consta un ID único que facilita el uso de métodos para realizar el proceso de indexación.

Pueden llegar a almacenar datos JSON, XML y BSON ya que son especializadas para controlar este tipo de archivos, pero en una sola colección con un valor y una clave especificada. Una vez almacenados los documentos, los campos públicos de los mismos pueden llegar a indexarse para su recuperación rápida en caso de no conocer la clave.

Los documentos que lleguen a ser almacenados pueden ser diferentes o iguales en su estructura. (Bosonit, 2018). En la Fig. 6 se hace una referencia a una estructura documental.



Fig. 6 Estructura de una base de datos documental

Muchos desarrolladores que proveen de este servicio han desarrollado lenguajes similares a SQL para poder realizar consultas en donde se pueda efectuar juntas entre colecciones, gracias a los caracteres estructurados que se afianzan en los documentos y los IDs únicos que se proporcionan a cada uno de ellos hacen que estas bases de datos

documentales respondan adecuadamente a consultas mediante filtrado y agregación. (Bosonit, 2018).

Una de las características principales que se pueden citar es que poseen un esquema proporcional por cada documento no relacional que puede llegar a variar y con ello ofrecer a los administradores más flexibilidad para las organizaciones, como también puede trabajar con la información mediante la aplicación de reglas estandarizadas para que la base de datos puede entenderlos durante el post procesamiento. (Kyocera, 2017).

En la Fig. 7 se muestra un ejemplo de cómo se presenta un documento JSON con su respectiva clave y valores.

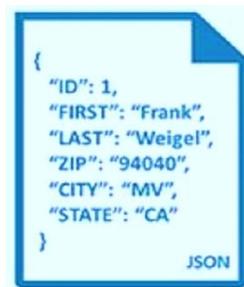


Fig. 7 Ejemplo de bases de datos documental

1.2.3 Bases de Datos de Columnares

Promueven el almacenamiento de los datos en tablas con filas y columnas, en las filas los formatos de las columnas pueden llegar a variar, pero identificándolas con sus respectivos nombres, mientras que las columnas se encargan de agrupar los datos que se encuentran en las columnas relacionándolos conjuntamente y facilitando así el acceso para realizar consultas hacia la base de datos. En la Fig. 8 se hace una referencia al manejo de una estructura columnar.

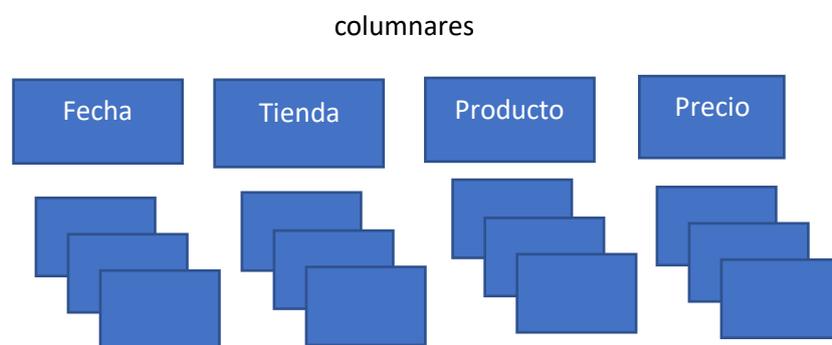


Fig. 8 Base de Datos columnares

El modelo de datos que se maneja es similar a las bases de datos relacionales, pero con la diferencia de que los datos se agrupan en las columnas mediante tuplas de valores que se encuentran asociadas hacia las filas de cada tabla. De acuerdo con ello la estructura

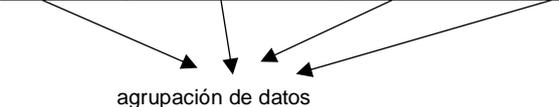
columnar establecida favorece a realizar consultas complejas sobre lecturas de información y extracción y agregado de la misma a cada una de las tablas de la base de datos columnar. (Bosonit, 2018).

Dentro de lo que son las características, poseen limitaciones estructurales en donde se puede utilizar ciertas técnicas que simulan la estructura de las bases de datos relacionales, a menudo al manejar estas técnicas en ciertos casos se requiere del uso de una misma clave principal para todas las tablas, lo que significaría que la jerarquía de la base de datos estaría limitada solo a dos niveles. (Garcete, 2014).

Mientras que para rendimiento los sistemas columnares pueden llegar a superar a los sistemas que se afianzan con las bases de datos relacionales, pero dicha superioridad en muchos casos puede variar en varios aspectos, como son las consultas en las que incluyen los cálculos o accesos individuales a los registros los cuales pueden llegar a ser lentos o rápidos en comparación con los sistemas relacionales que poseen una indexación adecuada. (Garcete, 2014).

En la Fig. 9 se muestra un ejemplo de cómo las columnas se encargan de agrupar los datos para poder realizar consultas.

clave fila	Datos personales		Datos profesional	
empid	Nombre	ciudad	designación	salario
1	raju	hyderabad	gerente	50,000
2	ravi	chennai	sr. ingeniero	30,000
3	rajesh	delhi	jr. ingeniero	25,000



 agrupación de datos

Fig. 9 Ejemplo de bases de datos columnares

1.2.4 Bases de Datos de Grafos

Representa la información almacenada como nodos de un grafo en donde las aristas representan a las relaciones, con el fin de que en caso de realizarse una revisión de la base de datos esta pueda ser accedida mediante una teoría de grafos para poder efectuar el respectivo recorrido tanto en los nodos como en las aristas de la base de datos. (Busto & Enríquez, 2013).

En la Fig. 10 se hace una referencia al manejo de una estructura de grafos.

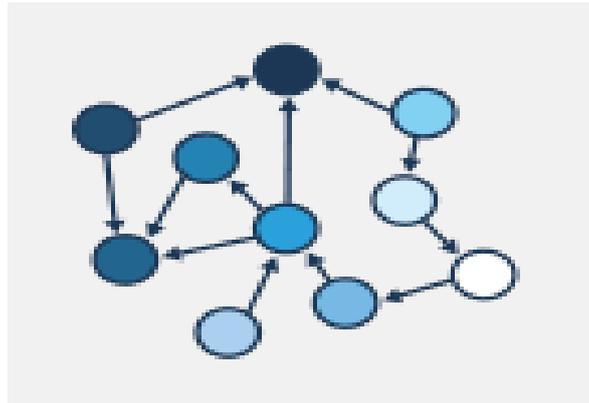


Fig. 10 Bases de Datos de grafos

Al hablar sobre la teoría de grafos es proporcionar evidencia sobre la gran utilidad de algoritmos de grafos que se orientan hacia varios tipos de cálculos que se centran en dar solución a diferentes problemáticas como: centralización de vectores que se impliquen en centralizaciones de los grafos, intermediación, HITS, entre otros.

En los últimos años se ha facilitado la realización de estudios sobre los algoritmos de aplicación en donde se ha detectado escenarios de alto rendimiento de usos de bases de datos de grafos, algunos proyectos que se nombraran a continuación poseen relevancia ya que son considerados como grandes escenarios en producción de bases de datos de grafos como son: Neo4j, Infinite Graph, HyperGraphDB, entre otros. (Busto & Enríquez, 2013).

Para la utilización de estas bases de datos es necesario que el modelado de datos este correctamente normalizado para que cada tabla pueda poseer una columna y cada relación adquiriera solamente dos, lo que ocasionara que los cambios que se realicen en la estructura de la información solo tengan efectos a nivel local y no afecte los procesos de modelamiento. (Busto & Enríquez, 2013).

En cuanto al diseño y consulta están orientado específicamente a datos en donde las relaciones están representadas con formas de grafos, lo que significa que la información representa a los datos que se encuentran interconectados con una cantidad no determinada de relaciones entre ellos. Mientras que las consultas a efectuarse deben ser mediante índices, ya que los sistemas que trabajan con este tipo de datos proporcionan una adyacencia libre de índices. (Busto & Enríquez, 2013).

En la Fig. 11 se muestra un ejemplo del recorrido que se realiza por los nodos mediante la utilización de una teoría de grafos.

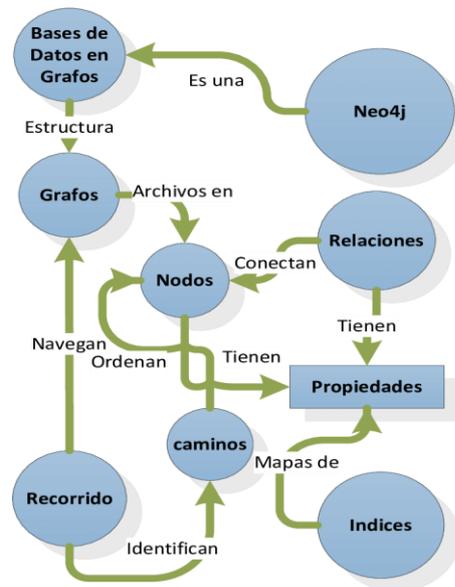


Fig. 11 Ejemplo de bases de datos de grafos

1.3 Bases de Datos NoSQL a estudiarse

Para efectuar este estudio se han seleccionado tres tipos de Bases de Datos NoSQL, las cuales fueron seleccionadas de acuerdo con la clasificación que ha dado el Cuadrante Mágico de Gartner en donde informan sobre los líderes y retadores en lo que respecta a sistema de gestión de bases de datos, es así como en este cuadrante cuando se hizo la respectiva investigación ubicaba a Apache Cassandra, MongoDB y Redis como una de las bases de datos NoSQL mejor ubicadas dentro del cuadrante (“Cuadrante Mágico para Sistemas Operativos de Gestión de Bases de Datos,” 2019). Al estimar las respectivas ubicaciones se consideró utilizar las tres bases de datos para desarrollar el estudio planteado y verificar cuál de las tres tiene la mejor capacidad de manejar grandes cantidades de información y control sobre la velocidad de procesamiento de registros de información. (Microsoft, 2019)

A continuación, se definen cada una de las Bases de Datos NoSQL seleccionadas para el estudio.

1.3.1 Bases de Datos Apache Cassandra

Es un tipo de Base de Datos distribuida, de alto rendimiento tanto en entrada como de salida de datos por lo que la hace productivamente escalable y congruente en manejar grandes cantidades de información, fue creada con el propósito de mantener la información respaldada con su protocolo de tolerancia a fallos.

Apache Cassandra es considerada como una solución post-relacional ya que cuenta con ciertas características que la hacen tener cierta similitud con las bases de datos relacionales como es su sistema de consultas que cuenta con ciertos comandos que se asemejan a los de las bases de datos relacionales. (Ladero, 2014).

En la Fig. 12 se muestra arquitectura de la base de datos Apache Cassandra.

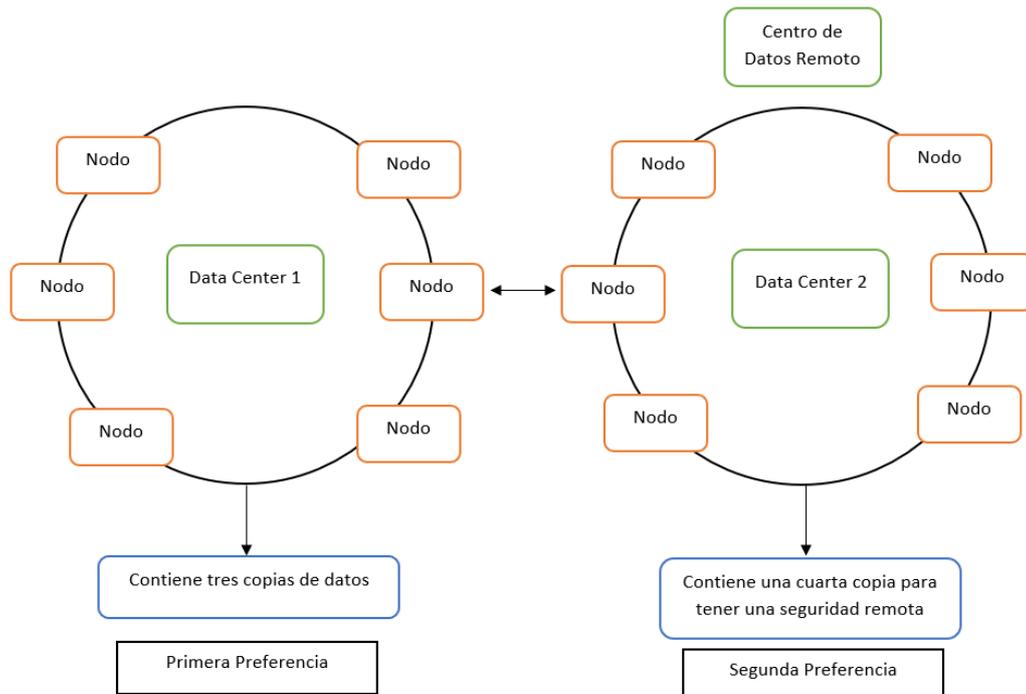


Fig. 12 Arquitectura de Apache Cassandra

La estructura que se maneja ayuda a que exista disponibilidad, tolerancia a fallos y consistencia que puede ser configurada de acuerdo con las necesidades de los usuarios, esta configuración puede estar repartida de la siguiente manera: (Zaforas, 2016).

- a) Escala Distribuida: Los datos se encuentran distribuidos a lo largo de todos los nodos del cluster, además de ello cuenta con alta disponibilidad lo que la hace favorable en caso de que existiera una caída de algún nodo, el servicio no sufriría cambios y seguiría con su ritmo normal en disponibilidad de información.
- b) Escala Linealmente: Proporciona un rendimiento de forma lineal con respecto al número de nodos que sean añadidos por los administradores de las bases de datos.
- c) Escala de forma Horizontal: Proporciona un escalamiento hacia los sistemas por medio del aumento de nodos los cuales están basados en hardware commodity de bajo costo.

Esta base de datos posee una arquitectura Peer-to-Peer que ayuda a eliminar puntos de fallo únicos dentro de establecimiento de información y provee con ello que todos los nodos tengan la misma importancia al ser almacenados, de acuerdo con este aspecto el patrón maestro-esclavo no podría ser aplicado ya que ningún nodo poseerá información relacionada con otros nodos.

Una característica esencial que cabe mencionar es que posee su propio lenguaje de consultas llamado Cassandra Query Language (CQL) que es utilizado para acceder a la base de datos mediante consultas, los datos que se encuentran dentro de la base de datos se encuentran desnormalizados por lo cual no se puede realizar junturas como las bases de datos relacionales. Para poder interactuar con la base de datos mediante CQL es necesario ingresar al intérprete de comandos shell, que permite ingresar comandos específicos del lenguaje de consultas de CQL. (Zaforas, 2016).

1.3.2 Base de Datos MongoDB

Se lo reconoce como un sistema de bases de datos multiplataforma que se orienta a documentos, por lo que al realizar sus registros con esquemas de datos diferentes y con atributos definidos hace que el mismo no se repita y no haya redundancia en los datos ingresados. Dentro de MongoDB se centran varias características como la velocidad de registro de datos y el control de los mismos, este proceso es realizado mediante un sistema sencillo de consultas que se efectúa hacia las bases de datos que se encuentran almacenadas dentro del gestor de MongoDB. (Paramio, 2011).

En la Fig. 13 se representa la arquitectura de la base de datos MongoDB.

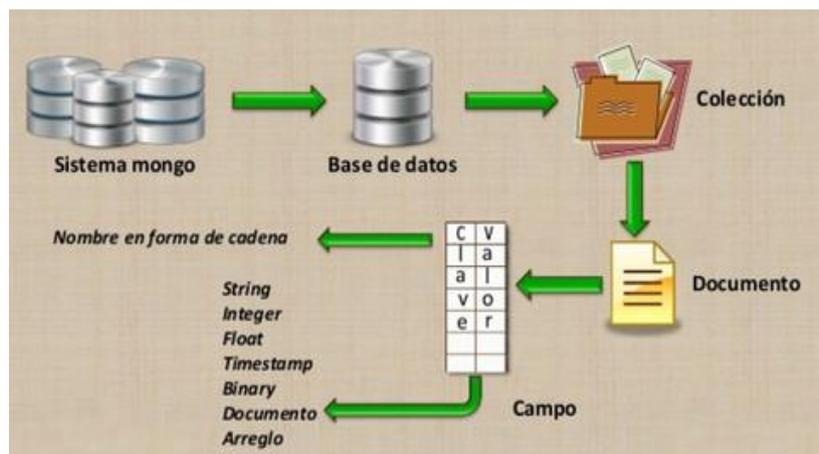


Fig. 13 Arquitectura de MongoDB

MongoDB posee un balance subjetivo en cuanto a lo que es rendimiento y funcionalidad, ya que cuenta con diferentes tipos de consultas las cuales podrían ser

eficientes en un sistema relacional definido y sin tener que sacrificar el rendimiento que provee el sistema en donde se esté efectuando el registro de información. (Paramio, 2011).

Ciertas características pueden ser descritas de la siguiente manera:

- a) Permite a los usuarios manipular la información almacenada en forma de documentos con un formato de tipo BSON. (Calvo, Durán, Quirós & Malinowski, 2017).
- b) Para la creación de las bases de datos no es necesario la definición de un esquema, lo que permite a los desarrolladores crear sistemas con alta disponibilidad y flexibles a modificaciones que se requieran realizar en la estructura de los documentos. (Calvo, Durán, Quirós & Malinowski, 2017).
- c) Ofrece diferentes tipos de prestaciones a nivel de implementación como es la definición de documentos anidados o referenciados o también el uso de comandos especializados como el operador lookup el cual sustituye las funciones del operador de juntura entre documentos. (Calvo, Durán, Quirós & Malinowski, 2017).

Las operaciones con las cuales se puede trabajar proporcionan a los usuarios la ejecución de múltiples consultas en documentos, además de ofrecer la facilidad de manipulación de los mismos como es la inserción de documentación mediante tuberías de agregación y mapeos de reducción.

De acuerdo con estudios esta manipulación de documentación puede ser ofrecida por diferentes herramientas que pueden determinar el proceso de manipulación de la documentación. (Calvo, Durán, Quirós & Malinowski, 2017).

En la Tabla 2 se muestra como la tubería de agregación permite a la documentación pasar por diferentes etapas para generar una colección o retomar un cursor, estas etapas indican algunas operaciones que pueden ser ejecutadas en MongoDB y pueden ser las siguientes:

TABLA 2
INSERCIÓN DE DOCUMENTACIÓN MEDIANTE TUBERÍA DE AGREGACIÓN

Tubería de Agregación	
Match	Verifica y selecciona la información que posea un criterio de búsqueda.
Project	Verifica la transformación de la estructura de los documentos con el fin de modificar o eliminar campos existentes.
Lookup	Permite ejecutar un left en vez de un join.
Unwind	Realiza la separación y creación de un nuevo documento para una entrada de un arreglo de algún documento existente.
Group	Encargado de agrupar los documentos mediante un campo de tipo llave.
Out	Especifica el lugar del almacenamiento de los resultados obtenidos.

Fuente: Adaptado de (Calvo, Durán, Quirós & Malinowski, 2017)

En la Tabla 3 indica que el mapeo de reducción es una metodología basada en una asociación de tipo llave-valor y para hacer uso de ella es necesario aplicar el uso de tres elementos los cuales se presentan de la siguiente manera. (Calvo, Durán, Quirós y Malinowski, 2017).

TABLA 3
INSERCIÓN DE DOCUMENTACIÓN MEDIANTE MAPEO DE REDUCCIÓN

Mapeo de Reducción	
Funciones de Mapeo	Transmite un arreglo de valores que se encuentran entrelazados a una misma llave.
Funciones de Reducción	Encargado de recibir una llave a la cual se la cataloga como un parámetro de un arreglo de valores entrelazados, en donde se tiene como fin retornar un valor asociado con la llave.
Salida de Datos	Encargado de retornar el resultado usando una de las dos opciones, el resultado obtenido es almacenado en una colección o se puede también devolver un documento.

Fuente: Adaptado de (Calvo, Durán, Quirós & Malinowski, 2017)

1.3.3 Base de Datos Redis

Es una Base de Datos NoSQL que almacena la información de forma estructurada, debido a su funcionalidad y capacidad puede guardar diferentes tipos de estructuras de datos tales como bitmap, hashes, listas e índices geoespaciales los mismos que pueden ser accedidos mediante radius queries y streams. Redis al poseer replicación incorporada, transacciones, secuencia de comandos de unión, diferencia e intersección y diferentes niveles de persistencia en disco ayuda a que exista una alta disponibilidad de acceso a información mediante redis sentinel y particionamiento automático mediante redis cluster. (Introduction to Redis – Redis, 2020)

En la Fig. 14 se muestra la arquitectura que maneja la base de datos Redis.

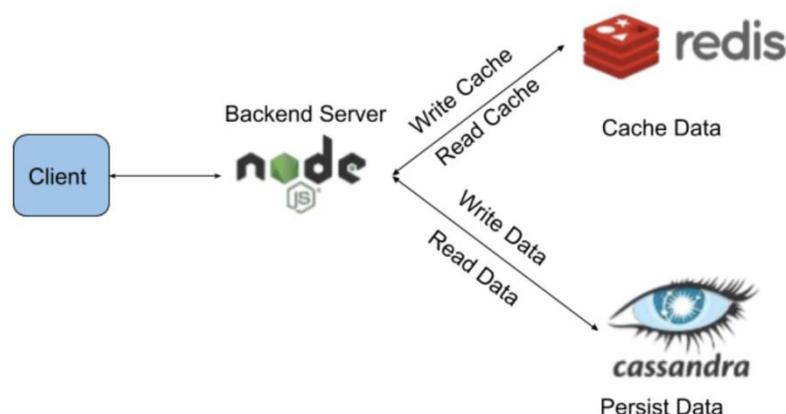


Fig. 14 Arquitectura de Redis

Una de las funcionalidades principales que presenta Redis es ejecutar operaciones atómicas en diferentes tipos como: incremento de valor a un hash, movimientos de datos en una lista, conjunción de cómputo de unión, intersección y diferencia, como también puede

ofrecer replicación asíncrona maestro-esclavo que puede ser accedida mediante una configuración en la primera sincronización rápida sin bloqueo y conexión automática con resincronización parcial en división de red.

Esta base de datos puede ofrecer un alto rendimiento cuando se trabaja mediante el uso de conjunto de datos los cuales pueden ser utilizados siempre y cuando exista una dependencia de caso de uso para poder realizar el volcamiento de un conjunto de datos en el disco o como a su vez se podría agregar comandos a los registros almacenados para poder acceder óptimamente a la información almacenada. (Introduction to Redis – Redis, 2020).

1.4 Análisis Comparativo entre las Bases de Datos NoSQL

Con la información obtenida y descrita anteriormente en los subtemas se pudo conseguir establecer un cuadro comparativo con las principales características que diferencian a cada una de las bases de datos NoSQL que se van a utilizar para el estudio a realizar y en la Tabla 4 se puede verificar dichas diferencias y se las puede describir de la siguiente manera.

TABLA 4
COMPARACIÓN DE LAS BASES DE DATOS A ESTUDIARSE

Descripción	Apache Cassandra	Redis	MongoDB
Modelo de Bases de Datos Primaria	Su columna ancha de almacenamiento es basada en la estructura de las Bases de Datos BigTable y DinamoDB	Su modelo de bases de datos permite almacenar los datos mediante tablas hashes.	Su modelo de Base de datos es documental y es una de las más populares.
Desarrollador	Es desarrollada por Apache Software Foundation.	Es desarrollada por RedisLabs.	Es desarrollada por MongoDB, Inc
Lenguaje de Programación	Java	ANSI C	C++
Servidor de Sistemas Operativos	Linux, OS X, Windws	Linux, * BSC, OS Windows	Linux, OS X, Solaris, Windws
Lenguajes de Programación soportados	C++, Java, JavaScript, PHP, Python, Haskell	C, C#, C++, Java, PHP, Python, Erlang, Go, Haskell, Javascript, Perl, Ruby, Scala	C#, Delphi, Haskell, Java, JavaScript, Matlab, Perl, PHP, Python
Conceptos de Usuario	Existe un acceso de los usuarios por objeto.	Existe una lista de control para acceso de los usuarios (ACL)	Existe un acceso para usuarios y roles.

Fuente: Adaptado de (Zaforas, Paramio & Gómez, 2017)

1.5 Metodología de desarrollo Scrum

Scrum es una metodología ágil que ayuda a solventar problemas complejos mediante la entrega de productos optimizados a su mayor expresión, lo cual es alcanzado mediante estrategias ordenadas y estrictamente cumplidas. Según (Porras, 2020) la arquitectura funcional de Scrum se puede ver como en la Fig. 15 donde se detalla cada proceso que se sigue.



Fig. 15 Metodología Scrum

De acuerdo con (Menzinsky, López, & Palacio, 2016) a esta metodología se la puede catalogar de las siguientes maneras:

- Ligera:** Al contener poca teoría se puede clasificar en actividades como reuniones, sistematización de roles y establecimientos de ciertos principios para una funcionalidad ordenada.
- Fácil de entender:** Desde la implementación de la metodología se establecen reglas específicas no tan complicadas que se acoplen al proyecto en desarrollo, estas reglas pueden ser establecidas de acuerdo con la comodidad del equipo de trabajo.
- Difícil de Dominar:** Es necesario buscar un método adecuado de integrar la metodología al entorno en concreto que se quiera desarrollar, es por esto que el rol del Scrum Master está diseñado para dominar el método a implementarse y ponerlo a disposición del equipo de trabajo.

1.5.1 Roles de Scrum

Aquello proyectos que se someten al uso de Scrum deben llegar a conformar un grupo clasificado en tres roles los cuales se detallan en: dueño del producto (product owner), Equipo de desarrollo (development team) y el Administrador del proyecto (Scrum Master), los cuales

conforman un grupo de trabajo comprometido a que se llegue a un resultado final satisfactorio en donde se cumpla con los requisitos establecidos.

A continuación con respecto a lo dicho por (Schwaber; & Sutherland, 2013) se describe a cada uno de los roles mencionados anteriormente:

- a) Dueño del Producto: Es el encargado de decidir qué requisitos deben ir en el backlog del proyecto de acuerdo con las funciones principales que se requieran para el negocio, de igual manera al implementar los requisitos estos deben ser explicados al equipo de trabajo para que los mismos puedan establecer dichos requisitos hacia la construcción de las historias de usuario.
- b) Scrum Master: Encargado de guiar, controlar y ayudar al equipo de trabajo con el cumplimiento de las reglas establecidas para el acatamiento de las historias de usuario, como también apoderado de dar las soluciones respectivas hacia los problemas que se presenten.
- c) Equipo de desarrollo: Compuesto por personal calificado en conocimientos y suficiente capacidad para cumplir con los requerimientos solicitados con el fin de entregar el producto terminado en el tiempo establecido.

1.5.2 Eventos de Scrum

Con respecto a lo dicho por (Porrás, 2020) Scrum está compuesto por cuatro eventos que se desarrollan de una manera específica y se los describe de la siguiente manera:

- a) Sprint: Considerado como la etapa trascendental de Scrum, es en donde se realiza el desarrollo de las tareas en un tiempo de dos a cuatro semanas.
- b) Planificación del sprint: Durante la planificación se realiza una reunión para planear el proceso a seguir para el siguiente sprint, en este se determina las tareas y asignaciones para cada miembro del equipo de desarrollo, por lo general la planificación puede durar alrededor de dos horas por semana.
- c) Scrum Diario: Se realiza reuniones diarias en donde cada uno de los miembros del equipo de trabajo es sometido a responder tres preguntas como: ¿Qué hice ayer?, ¿Qué haré hoy?, ¿Qué limitaciones ha tenido durante el desarrollo?, el interrogatorio debe ser máximo de hasta quince minutos.
- d) Revisión del sprint: Durante este evento se hace un análisis hacia las tareas terminadas y si es necesario se realiza las respectivas correcciones en el product backlog, por lo general la revisión tiende a durar una hora por semana.
- e) Retrospectiva del sprint: Es una reunión en donde se examina si existe la posibilidad de establecer cambios para optimizar la productividad del equipo de trabajo, el tiempo que puede llegar a durar esta reunión es de tres horas como máximo por sprint.

1.5.3 Artefactos de Scrum

Son los encargados de representar el valor del trabajo mediante diferentes formas en donde se establezca la transparencia de la información y las oportunidades adecuadas para que la misma sea diseñada y adaptada para el entendimiento de los artefactos que se van a utilizar durante el proceso de implantación de los artefactos que serán definidos de la siguiente manera:

- a) Pila del Producto (Product Backlog): Es en donde se detalla de manera concisa los requisitos establecidos mediante historias de usuario, este artefacto es entregado y explicado al dueño del producto con el fin de darle a conocer de forma detallada el proceso a seguir para el desarrollo del software.
- b) Pila del Sprint (Sprint Backlog): Posee las historias de usuario, las mismas que se encuentran desglosadas en tareas para su desarrollo, estas tareas son clasificadas en tres estados que puede ser: por hacer, en proceso y hecho, de esta manera la pila sprint trabaja de una manera dinámica y ordenada.
- c) Incremento: Es implementado durante la fase de desarrollo del sprint y se convierte en un artefacto utilizable que es presentando durante la revisión de un sprint.

1.6 Estándar ISO/IEC 25012

El estándar ISO/IEC 25012 pertenece a la familia del estándar ISO/IEC 25000 cuyas normas tienen como principal objetivo centralizar un marco de trabajo en el cual se pueda realizar evaluaciones de la calidad del producto del software, pero en este caso el estándar ISO/IEC 25012 se concentra en evaluar la calidad del producto de datos en donde se puede encontrar características las cuales se describen en la Fig. 16.

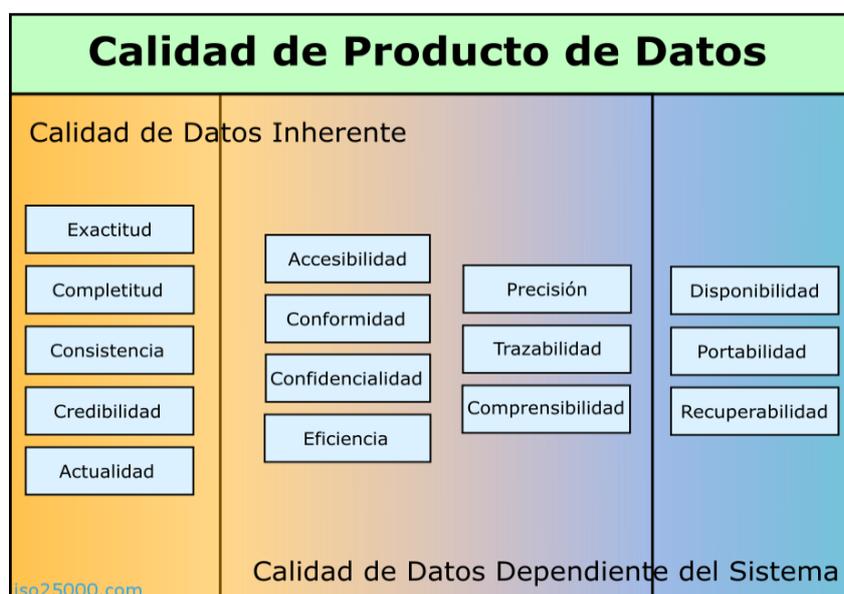


Fig. 16 ISO/IEC 25012

La forma en cómo se trabaja y se enfatiza el uso de la calidad de la información puede llegar a ser dentro de una organización un factor clave, ya que la toma de decisiones de las organizaciones en muchos casos depende de la calidad de datos. Es así como para sustentar y controlar la calidad de la información es necesario aplicar lineamientos, reglas, verificar características, control de procesos de información en cuanto a consistencia y credibilidad. Es así como las organizaciones que se encuentren centradas en obtener beneficios de la calidad datos aplican el estándar ISO/IEC 25012 con el fin de estimular y cumplir con las debidas reglas para mejorar la calidad de los datos que se obtienen de la recolección y almacenamiento de información de las organizaciones.

De acuerdo con el estándar vigente la norma ISO/IEC 25012, está enfocado en el modelo de la calidad de los datos en donde se llega a representar la base en que se construye un sistema para la apreciación de un producto de datos. Para poder constituir lo que es el modelo de la calidad de datos se establece características que controlen la calidad de datos las cuales se deben considerar al momento que se vaya a evaluar los resultados de los datos que se hayan obtenido en un sistema web o aplicación. (ISO 25012, 2008).

El estándar o normativa se centra en el control de la calidad del producto de datos, en donde se fija un objetivo principal que es satisfacer los requerimientos definidos del sistema mediante el cumplimiento de varias características como exactitud, completitud, consistencia, credibilidad, actualidad, accesibilidad. (ISO 25012, 2008).

La documentación de la (ISO 25012, 2008) presenta varias características de la calidad de datos que pueden ser clasificadas como:

- a) Calidad de Datos Inherentes: Mide el nivel de la calidad de los datos para satisfacer las necesidades establecidas mediante métricas específicas, desde un punto inherente la calidad de datos es medida de acuerdo con un grado de particularidad que puede ser: Posibles restricciones y valores de dominio, relaciones entre valores de dominio y metadatos.
- b) Calidad de Datos Dependiente del sistema: Se encarga de medir el grado de la calidad de datos alcanzada en un sistema informático, en el caso de que los datos sean utilizados bajo condiciones específicas que se centren un aspecto determinado. Enfatizando en el grado dependiente del sistema la calidad de los datos se enfoca en un dominio tecnológico en donde los datos se alcanzan mediante componentes informáticos como dispositivos de hardware y software.

Para realizar la evaluación de la calidad de datos es necesario partir de las características principales las cuales describen los procesos a seguir para cumplir con la calidad y se las puede describir de la siguiente manera: (ISO 25012, 2008).

- a) Exactitud: Los datos especificados contienen atributos los cuales representan el verdadero valor de un atributo deseado en un contexto especificado.
- b) Completitud: Los datos se encuentran asociados con una entidad la cual posee valores para cada uno de los atributos que se encuentren representados dentro de la entidad.
- c) Consistencia: Los datos deben ser coherentes y del mismo tipo de acuerdo con la entidad ya que van a estar constituidos mediante un contexto especificado que será dado a cada una de las entidades.
- d) Credibilidad: Los datos poseen atributos que son considerados verídicos para los usuarios.
- e) Actualidad: Los datos poseen atributos que contienen valores validos que van de acuerdo con el contexto establecido en la base de datos.

CAPÍTULO 2

Desarrollo

2.1 Análisis de Requerimientos

El sistema web a desarrollarse se enfoca en la creación de un prototipo de red social llamado BlackBelt en donde se pueda constituir y verificar el proceso de almacenamiento y control de la calidad del producto de los datos, como también enfatizar la utilización de nuevas tecnologías que innoven los sistemas web y optimicen sus procesos en cuanto a funcionalidad, fiabilidad y eficiencia respecta.

2.1.1 Historias de Usuario

De acuerdo con las estimaciones que sea podido establecer se presentan las siguientes historias de usuario de acuerdo con los requerimientos siguientes.

TABLA 5
HISTORIA DE USUARIO RSB-001

Historias de Usuario		
Código: RSB-001	Usuario: Desarrollador	
Nombre: Reunión del equipo de trabajo.		
Prioridad: alta	Riesgo: Alto	Estimación: 20 horas
Descripción: Reunión de planificación con el Scrum Master para establecer los requisitos necesarios para desarrollar el proyecto de trabajo de grado, en donde se acordará las tecnologías a necesitarse para realizar tanto el modelado de las bases de datos como el desarrollo del prototipo de red social.		
Pruebas de aceptación:		
<ul style="list-style-type: none">• Investigación sobre la base tecnológica a utilizarse para modelado y desarrollo del prototipo de red social.• Analizar cómo realizar el modelado de las bases de datos.• Investigación el software apropiado para el desarrollo del prototipo de red social.		

Fuente: Propia

TABLA 6
HISTORIA DE USUARIO RSB-002

Historias de Usuario

Código: RSB-002

Usuario: Desarrollador

Nombre: Establecer funcionalidades.

Prioridad: media

Riesgo: medio

Estimación: 12 horas

Descripción:

Definir las funcionalidades para el prototipo de red social, este proceso se lo realizará mediante la investigación de las funciones y herramientas con las cuales cuentan otras redes sociales.

Pruebas de aceptación:

- Investigación de funcionalidad de otras redes sociales.
- De acuerdo con el alcance establecido seleccionar las funcionalidades pertinentes para el prototipo de red social.
- Investigar el software necesario para poder desarrollar las funcionalidades seleccionadas

Fuente: Propia

TABLA 7
HISTORIA DE USUARIO RSB-003

Historias de Usuario

Código: RSB-003

Usuario: Desarrollador

Nombre: Establecer funcionalidades.

Prioridad: alta

Riesgo: medio

Estimación: 12 horas

Descripción:

Investigar la estructura de cada de las bases de datos NoSQL a utilizarse y con ello verificar como realizar el modelado de cada una de ellas y plasmarlo en el software DbSchema en donde se procederá a realizar el modelado y exportación de las mismas hacia las bases de datos.

Pruebas de aceptación:

- Hacer la conexión desde DbSchema hacia las bases de datos Apache Cassandra, MongoDB y Redis.
- Realizar el diseño de entidades que van a intervenir en el modelado de las bases de datos.
- Embeber las entidades correspondientes hacia la entidad principal Usuario.
- Verificar que los modelos de datos fueron exportados hacia los motores de las bases de datos.
- Ingreso de datos hacia las bases de datos.
- Realización de consultas mediante el uso de scripts para cada una de las bases de datos.
- Estimación y corrección de posibles errores a encontrarse en el modelado de las bases de datos.

Fuente: Propia

TABLA 8
HISTORIA DE USUARIO RSB-004

Historias de Usuario

Código: RSB-004

Usuario: Desarrollador

Nombre: Investigación de herramientas para el desarrollo del prototipo de red social

Prioridad: alta

Riesgo: Alto

Estimación: 6 horas

Descripción:

Búsqueda de tutoriales y manuales para que el desarrollador se capacite para manejar el lenguaje de programación y el software con el cual se proceda a desarrollar el prototipo de red social, de la misma manera se debe investigar las versiones del software para no tener inconvenientes al momento de realizar las conexiones desde el framework de desarrollo hacia las bases de datos.

Pruebas de aceptación:

- Control del proceso de capacitación hacia el desarrollador para probar su habilidad en el manejo del lenguaje de programación javascript.
- Revisión de las versiones con las cuales son compatibles cada software seleccionado para el desarrollo.
- Realizar pruebas de compatibilidad entre las versiones de software instaladas para probar que no exista problemas de conexión.

Fuente: Propia

TABLA 9
HISTORIA DE USUARIO RSB-005

Historias de Usuario

Código: RSB-005

Usuario: Desarrollador

Nombre: Inicio del desarrollo del Prototipo de red social

Prioridad: alta

Riesgo: alto

Estimación: 4 horas

Descripción:

Descarga del framework y las dependencias necesarias para efectuar el desarrollo del prototipo de red social, una vez instaladas las dependencias se efectúa la creación y configuración del servidor para la ejecución del sistema web, como también se realiza la conexión hacia las bases de datos para culminar con la configuración del servidor en node.js.

Pruebas de aceptación:

- Ejecutar el proyecto una vez se haya terminado de realizar la configuración del servidor.
- Comprobar la conexión de las bases de datos hacia el prototipo de red social mediante mensajes por consola que dirán si existen errores de configuración o conexión.
- Verificación de la configuración de servidor.

Fuente: Propia

TABLA 10
HISTORIA DE USUARIO RSB-006

Historias de Usuario

Código: RSB-006

Usuario: Desarrollador

Nombre: Gestión de Rutas y creación de directorios dentro del proyecto en desarrollo

Prioridad: alta

Riesgo: alto

Estimación: 3 horas

Descripción:

Para que el proyecto se vea ordenado se debe crear directorios en donde se almacenen los diferentes archivos de tipo javascript (js) y handlerbars (hbs) para el desarrollo del sistema web, una vez especificado el orden adecuado se crean todas las rutas con sus respectivas vistas.

Pruebas de aceptación:

- Comprobación de las rutas mediante la ejecución del proyecto en donde la URL que se muestre en el navegador indique el nombre dado en el archivo de rutas.
- Modificar las rutas con los respectivos nombres de las vistas creadas y enviar a ejecutar el proyecto para verificar la redirección de las rutas.

Fuente: Propia

TABLA 11
HISTORIA DE USUARIO RSB-007

Historias de Usuario

Código: RSB-007

Usuario: Desarrollador

Nombre: Almacenamiento de imágenes en la base de datos.

Prioridad: alta

Riesgo: alto

Estimación: 8 horas

Descripción:

Creación de un modelo de datos en donde se especifique los atributos necesarios para almacenar las imágenes en la base de datos, una vez creado el modelo se debe programar un método que sirva para almacenar las imágenes. Como también se procederá a mostrar en el sistema web las imágenes que se encuentren almacenadas en la base de datos.

Pruebas de aceptación:

- Controlar que los atributos creados en el modelo cumplan con los atributos de las entidades creadas en el modelo de la base de datos.
- Hacer las pruebas pertinentes del método especificado para subir las imágenes del sistema web hacia la base de datos.
- Mediante la ejecución del sistema web se podrá cargar una imagen y verificar si esta se despliega en pantalla mostrando su contenido.

Fuente: Propia

TABLA 12
HISTORIA DE USUARIO RSB-008

Historias de Usuario

Código: RSB-008

Usuario: Desarrollador

Nombre: Creación de etiquetas y panel de comentarios para las imágenes

Prioridad: alta

Riesgo: alto

Estimación: 8 horas

Descripción:

Las imágenes deben contar con tres etiquetas principales que son: me gusta, número de visualizaciones y tiempo en que se ha cargado las imágenes hacia la base de datos, como también debajo de cada imagen se cuenta con un panel para realizar los comentarios de usuarios.

Pruebas de aceptación:

- Una vez programadas las etiquetas se ejecutará el proyecto y se verifica que las etiquetas aparezcan debajo de la imagen marcando consigo un pequeño icono que mostrará el número de me gustas, numero de visualizaciones y tiempo de carga de las imágenes.
- Verificar que los métodos creados cuenten con los atributos y condiciones necesarias para cumplir con la consulta (leer, cargar y eliminar) que vaya a realizar el usuario dueño de las publicaciones.

Fuente: Propia

TABLA 13
HISTORIA DE USUARIO RSB-009

Historias de Usuario

Código: RSB-009

Usuario: Desarrollador

Nombre: Creación de Registro y Login

Prioridad: alta

Riesgo: alto

Estimación: 10 horas

Descripción:

Crear un modelo de datos en donde se especifique los atributos necesarios para registrar usuarios, una vez realizado el proceso pertinente se procede a crear las vistas necesarias para hacer el llamado al modelo y realizar los registros, como también se debe crear los métodos necesarios para la autenticación que permitan legitimar a los usuarios registrados, este método será invocado en la ruta de login que se creará para ingresar a la cuenta creada por el usuario. Invocar a las dependencias apropiadas para controlar los errores que se presentan al momento de realizar el login y registro de los nuevos usuarios.

Pruebas de aceptación:

- Controlar que los atributos creados en el modelo de datos cumplan con los atributos de la entidad creada en el modelo de datos en DbSchema.
- Controlar que las rutas especificadas para el registro y login sean redireccionadas hacia las vistas apropiadas.
- Verificar que los métodos de autenticación estén controlando el ingreso de los usuarios registrados.
- Los usuarios al registrarse deberán ingresar una contraseña que debe ser mayor a cuatro caracteres, en caso de que el usuario solo ingrese cuatro caracteres como contraseña le aparecerá un mensaje pidiéndole que ingrese una contraseña segura.
- Verificar que los métodos creados cuenten con los atributos y condiciones necesarias para cumplir con la consulta (leer, cargar y eliminar) que vaya a realizar el usuario dueño de las publicaciones.

Fuente: Propia

TABLA 14
HISTORIA DE USUARIO RSB-010

Historias de Usuario

Código: RSB-010

Usuario: Desarrollador

Nombre: Creación de vistas principales que tendrá el usuario logeado

Prioridad: alta

Riesgo: alto

Estimación: 4 horas

Descripción:

El usuario una vez logeado se le presenta como vista principal su muro de perfil con sus datos como nombre, correo y el tiempo cuando creo su cuenta en el prototipo de red social, como también podrá encontrar una sección de lado izquierdo con varias opciones de rutas a redirigirse como son: ruta para carga de imágenes, ruta para contactos, ruta para publicaciones, ruta para ingreso de información personal y ruta de mensajería.

Pruebas de aceptación:

- Verificar que al momento de que el usuario ingrese a su cuenta se muestre su muro con diferentes opciones de lado izquierdo para poder ingresar diferente tipo de información.
- Verificar que las opciones del lado izquierda se redirijan hacia las rutas especificadas.

Fuente: Propia

TABLA 15
HISTORIA DE USUARIO RSB-011

Historias de Usuario

Código: RSB-011

Usuario: Desarrollador

Nombre: Contactos dentro del prototipo de red social

Prioridad: alta

Riesgo: medio

Estimación: 5 horas

Descripción:

De acuerdo con los usuarios registrados en el prototipo de red social se debe crear un método para poder invocar a todos los usuarios y presentarlos en una tabla con sus respectivos datos para que los usuarios logeados puedan saber cuáles son los contactos que sean registrado en el prototipo de red social. **Pruebas de aceptación:**

- Verificar que la ruta de contactos redirija hacia la vista especificada.
- Verificar que se muestre en una tabla los usuarios registrados con sus respectivos datos dentro del prototipo de red social.

Fuente: Propia

TABLA 16
HISTORIA DE USUARIO RSB-012

Historias de Usuario

Código: RSB-012

Usuario: Desarrollador

Nombre: Ingreso y despliegue de Información del Usuario

Prioridad: alta

Riesgo: medio

Estimación: 5 horas

Descripción:

El usuario en el caso de querer ingresar su información personal podrá hacerlo desplegando una vista en donde podrá ir ingresando sus datos y después de ello su información será desplegada en un panel.

Pruebas de aceptación:

- Controlar que la ruta especificada para el ingreso de información esté definida adecuadamente.
- Realizar las pruebas pertinentes del método de ingreso de información para poder asegurar que las misma se vaya almacenando en la base de datos.
- Revisar que las opciones de selección en los campos de texto de fecha de nacimiento, carrera y sexo estén configuradas correctamente para que los usuarios puedan elegir la opción correspondiente.

Fuente: Propia

TABLA 17
HISTORIA DE USUARIO RSB-013

Historias de Usuario

Código: RSB-013

Usuario: Desarrollador

Nombre: Modelo de Publicaciones de estados y programación en el servidor y cliente

Prioridad: alta

Riesgo: medio

Estimación: 5 horas

Descripción:

Creación de un modelo de datos en donde se especificará los atributos necesarios para que el usuario pueda realizar la publicación de su estado de ánimo, para ello es necesario que se emplee un método para almacenar los estados en la base de datos y control de las vistas para el despliegue de los mismos.

Pruebas de aceptación:

- Controlar que los atributos creados en el modelo de datos cumplan con los atributos de la entidad creada en el modelo de la base de datos.
- Verificar que las rutas proporcionadas en el servidor sean ingresadas en el cliente y estén correctamente redirigidas hacia las vistas pertinentes.
- Realizar pruebas de la publicación de los estados de ánimos de los usuarios mediante el ingreso de información en los paneles y con ello verificar que los estados se estén desplegando en la vista adecuada.

Fuente: Propia

TABLA 18
HISTORIA DE USUARIO RSB-014

Historias de Usuario

Código: RSB-014

Usuario: Desarrollador

Nombre: Creación del Modelo con los atributos para Mensajería entre usuarios y configuración

Prioridad: alta

Riesgo: medio

Estimación: 12 horas

Descripción:

Investigar cómo realizar el proceso de mensajería dentro de una red social y después de ello crear el modelo en donde se especifique los atributos necesarios para los mensajes, este proceso va a ser controlado mediante una dependencia llamada socket.io.

Pruebas de aceptación:

- Controlar paulatinamente el desarrollo de la mensajería este proceso mediante la realización de pruebas tanto de la parte del servidor como del cliente.
- Mediante cuadros de dialogo ir controlando los errores que se puedan presentar durante el desarrollo del proceso de mensajería.

Fuente: Propia

TABLA 19
HISTORIA DE USUARIO RSB-015

Historias de Usuario

Código: RSB-015

Usuario: Desarrollador

Nombre: Modificar el diseño de las vistas mediante el uso de Bootstrap 4 y CSS

Prioridad: media

Riesgo: medio

Estimación: 5 horas

Descripción:

Para mejorar el diseño anterior de la red social se ha visto en la necesidad de mejorar las vistas del prototipo de red social para que la misma se vea más llamativa y tenga una mejor presentación para los usuarios y para ello se utilizará Bootstrap 4, css 3 y Font awesome icon para poder hacer animaciones, dar color y poner iconos en los respectivos campos del prototipo de red social.

Pruebas de aceptación:

- Implementar las librerías de Bootstrap 4 y CSS para realizar las mejoras del diseño en cada una de las vistas del prototipo de red social.
- Ejecutar el proyecto para verificar los cambios ocurridos en las vistas de la red social.
- Utilizar elementos para las etiquetas que correspondan solamente a Bootstrap 4 ya que si se utiliza elementos de versiones inferiores existirá problemas al cargar las vistas en la red social.

Fuente: Propia

2.1.2 Product Backlog

A continuación, se detalla una lista de las funcionalidades que tendrá el prototipo de red social.

TABLA 20
PRODUCT BACKLOG

Código	Historias de Usuario	Prioridad	Riesgo	Estimación (en horas)
RSB-001	Reunión del equipo de trabajo,	alta	alto	20 horas
RSB-002	Establecer funcionalidades	alto	medio	12 horas
RSB-003	Modelado de la Base de Datos	alta	medio	8 horas
RSB-004	Investigación de herramientas para el desarrollo del prototipo de red social	alta	alto	6 horas
RSB-005	Inicio del desarrollo del Prototipo de red social	alta	alto	4 horas
RSB-006	Gestión de Rutas y creación de directorios dentro del proyecto en desarrollo	alta	alto	3 horas
RSB-007	Almacenamiento de imágenes en la base de dato	alta	alto	5 horas
RSB-008	Creación de etiquetas y panel de comentarios para las imágenes	alta	alto	3 horas
RSB-009	Creación de Registro y Login	alta	alto	10 horas
RSB-010	Creación de vistas principales que tendrá el usuario logeado	alta	alto	4 horas
RSB-011	Contactos dentro del prototipo de red social	alta	alto	5 horas
RSB-012	Ingreso y despliegue de Información del Usuario	alta	alto	5 horas
RSB-013	Modelo de Publicaciones de estados y programación en el servidor y cliente	alta	alto	5 horas
RSB-014	Creación del Modelo para Mensajes y configuración en el cliente y el servidor	alta	alto	12 horas
RSB-015	Modificar el diseño de las vistas mediante el uso de Bootstrap 4 y CSS.	media	media	5 horas

Fuente: Propia

2.1.3 Roles del proyecto

Dentro del proceso de desarrollo del sistema se podrá encontrar diferentes agentes que se involucrarán durante este proceso, quienes tendrán un rol específico que ayudará a culminar el sistema web, a continuación, se presentan las personas implicadas en la tabla 21.

TABLA 21
ROLES DEL PROYECTO

Persona	Rol	Descripción
Msc. Pablo Landeta	Dueño del producto (Product Owner)	Persona que propuso la idea para el desarrollo del estudio.
Sr. Bryan Cachimuel	Líder del proyecto (Scrum Master)	Encargado de que el equipo de trabajo logre cumplir con los objetivos planteados y cumplir con cada sprint planteado.
Sr. Bryan Cachimuel	Equipo de desarrollo (Development team)	Encargado de realizar el trabajo teórico y práctico del proyecto de tesis.
Msc. Marco Pusda Msc. Silvia Arciniega	Testers	Encargados de comprobar el funcionamiento y control de errores del sistema web desarrollado.

Fuente: Propia

2.1.4 Roles del Develop team

Dentro de lo que corresponde al desarrollo se ha visto necesario mencionar las funciones que cumple las personas implicadas y se las describe en la cada una de ellas en la tabla 22.

TABLA 22
ROLES DEL DEVELOP TEAM

Rol	Descripción
Diseñador	Encargado de crear el diseño arquitectónico mediante los requerimientos establecidos y de acuerdo con ello generar prototipos de la arquitectura del sistema web.
Analista	Encargado de realizar el análisis respectivo de los requerimientos, estructura del sistema y los diagramas de arquitectura del sistema web.
Desarrollador	Encargado de explorar los diferentes ambientes, herramientas y lenguajes de programación en la que el sistema web podría ser desarrollado.
Usuario	Es necesario que el usuario este familiarizado con el sistema para que conozca las funcionalidades del sistema y las pueda evaluar
Testers	Son los encargados de llevar a cabo pruebas planificadas para comprobar el funcionamiento correcto de los sistemas web y verificar posibles errores y en caso de a verlos se comunican el personal encargado del desarrollo del sistema.

Fuente: (González, 2012)

2.1.5 Creación de los Sprints

De acuerdo con las historias de usuario se procederá a crear los sprints respectivos para obtener una mejor planificación para el desarrollo del sistema web.

2.1.6 Sprint 0

- Planificación del Sprint 0

Reunión del equipo de trabajo: Conjuntamente se reunirán el Scrum master y equipo de desarrolladores.

Objetivo: Definir con el Scrum Master que tipo de software se van a utilizar y que periodos de tiempo se realizará la revisión de los avances de acuerdo al seguimiento que se dará al aplicar la metodología SCRUM.

Miembros: Scrum Master y equipo de desarrollador

a) Spring Backlog

TABLA 23
SPRING BACKLOG

Historia de Usuario	Fases de Desarrollo	Actividad	Número de horas
RSB-001	Planificación	Reunión del equipo de trabajo.	6
	Planificación	Establecer el alcance del sistema web	4
	Investigación	Investigación de software para el modelamiento y desarrollo del sistema web	3
	Socialización	Dar a conocer al Scrum Master sobre el software investigado	2
	Planificación	Selección fija sobre software a utilizarse	2
Eventos	Planificación	Definir las tereas que desarrollaran en el sprint 0	2
	Retrospectiva	Análisis sobre los avances del sprint	1
Total de horas			20

Fuente: Propia

- Revisión del Sprint 0

Al finiquitar el Sprint 0 se ha podido determinar cómo los procesos de investigación ayudan a los desarrolladores a tener una idea clara y concisa del proyecto que se ha va a desarrollar y con ello se describir la información obtenida y plasmarla en el capítulo 1 de este trabajo de grado.

- Retrospectiva del Sprint 0

TABLA 24
RETROSPECTIVA SPRINT 0

		RETROSPECTIVA
Aciertos (¿Qué salió bien del Sprint?)	-	Obtener la información necesaria desde repositorios bibliográficos que ayudaron a encontrar información solvente para fundamentar la documentación del capítulo 1
Errores (¿Qué no salió bien del Sprint?)	-	La investigación tomo más tiempo del previsto y se tuvo que intensificar la investigación para cumplir con el tiempo especificado.
Mejoras (¿Qué mejoras se implementará?)	-	Utilizar otras técnicas para mejorar la calidad de captación de información que se va obteniendo al investigar los artículos dentro de las bases bibliográficas.

Fuente: Propia

2.1.7 Sprint 1

- Planificación del Sprint 1

Para el cumplimiento del Sprint 1 es necesario especificar un orden de investigación para analizar tanto las funcionalidades que tendrá el sistema web como también para iniciar con el desarrollo del mismo, de esta manera se procederá a realizar un avance ordenado y debidamente planificado.

a) Spring Backlog

TABLA 25
SPRINT BACKLOG

Historia de Usuario	Fases de Desarrollo	Actividad	Número de horas
RSB-002	Análisis	Analizar otras redes sociales y buscar las mejores funcionalidades para agregar al prototipo de red social.	12
RSB-003	Investigación	Buscar información y tutoriales de cómo realizar el modelado de la base de datos en Apache Cassandra, MongoDB y Apache Hbase	6
	Modelamiento	Realizar el modelamiento de las bases de datos e ingresar en cada colección y tablas los debidos atributos.	2
RSB-004	Investigación	Buscar las herramientas necesarias para poder iniciar con la fase de desarrollo.	6
Eventos	Planificación	Detallar las tareas a realizar en el Sprint actual	4
	Revisión	Revisión de los resultados obtenidos del Sprint con el Scrum Master.	2
	Retrospectiva	Analizar los resultados del Sprint	2
Total de horas			34

Fuente: Propia

- Revisión del Sprint 1

Al finiquitar el sprint 1 se puede determinar cuáles serían las mejores funcionalidades con las cuales podrá contar el prototipo de red social y con ello se pondrá énfasis en realizar

una investigación para cada funcionalidad seleccionada. Como también se ha hecho una investigación del software apropiado para realizar el modelado y el desarrollo del sistema web el cual se acopla a las necesidades del equipo de desarrollo.

Al investigar el proceso de tratamiento de los atributos dentro de cada base de datos se ha visto que existen múltiples diferencias en cada una de ellas, es por ello que se encontró cierto grado de dificultad ya que cada una cuenta con un sistema de modelamiento que se acopla a su estructura, es por ello que al investigar cada base de datos se visto en la necesidad de utilizar más tiempo del previsto para estudiar y comprender como seguir el debido proceso de modelamiento de las bases de datos.

- Retrospectiva del Sprint 1

TABLA 26
RETROSPECTIVA DEL SPRINT 1

		RETROSPECTIVA
Aciertos (¿Qué salió bien del Sprint?)	-	Las investigaciones realizadas ayudaron al equipo de desarrollo a despejar ciertas dudas y a encontrar información necesaria para conocer como es el funcionamiento de las bases de datos NoSQL.
Errores (¿Qué no salió bien del Sprint?)	-	Existieron imprevistos en las búsquedas de cracks para el software de modelado de bases de datos y debido a ello se usó un poco más del tiempo especificado.
Mejoras (¿Qué mejoras se implementará?)	-	Investigar en fuentes verídicas ya que en muchos casos ciertas fuentes contienen información caduca que no sirve para el sistema web que se planea desarrollar.

Fuente: Propia

2.1.8 Sprint 2

- Planificación del Sprint 2

Para el cumplimiento del Sprint 2 es necesario especificar un orden de investigación para analizar tanto las librerías necesarias como las rutas adecuadas que tendrá el prototipo de red social ya que con ello facilitará al usuario la navegación por las vistas de la red social. Como también será necesario codificar métodos para poder interactuar con las bases de datos Apache Cassandra, MongoDB y Redis. Se ha especificado otras tareas adecuadas para el cumplimiento de carga y verificación de imágenes en la base de datos, dicho de esta manera se procederá a realizar un avance ordenado y debidamente planificado.

a) Spring Backlog

TABLA 27
SPRING BACKLOG DEL SPRINT 2

Historia de Usuario	Fases de Desarrollo	Actividad	Número de horas
RSB-005	Análisis	De acuerdo con los manuales, videos investigados se debería seleccionar las dependencias que ayudaran a desarrollar el sistema web.	1
	Inicio del Desarrollo	Descargar el framework express y las dependencias para comenzar a realizar la configuración del servidor en node.js como también realizar la conexión hacia las bases de datos NoSQL.	3
RSB-006	Análisis	Analizar las rutas necesarias que podrá llegar a tener el prototipo de red social.	1
	Creación y Especificación	Se integra el script necesario para desarrollar las rutas principales que tendrá el prototipo de red social.	1
	Orden de archivos de acuerdo con los directorios creados	Se creará varios directorios con el fin de establecer un orden para que los archivos programados estén desarrollados en un debido orden.	1
RSB-007	Análisis	Especificar una vista en donde se pueda realizar un esquema para la subida de imágenes desde el ordenador hacia el prototipo de red social.	1
	Esquemmatización	Desarrollar un panel en donde se pueda pedir la selección de una imagen del ordenador y que al seleccionarla pida cierta información para poder almacenarla dentro de la base de datos.	5
	Verificación	Mostrar en un panel la imagen subida hacia la base de datos.	2
RSB-008	Análisis	Seleccionar las etiquetas necesarias con las cuales se presentarán las imágenes hacia los usuarios.	1
	Métodos y Comentarios	Creación de métodos para poder leer y eliminar las imágenes subidas hacia la red social, como también se creará un panel inferior para poder realizar comentarios de las imágenes subidas por los usuarios.	5
	Verificación	Mostrar en el panel de la imagen las etiquetas con los iconos de me gusta, número de vistas y hora de carga de la imagen hacia la base de datos. Como también verificar que los comentarios puedan ser ingresados.	2
Eventos	Planificación	Detallar las tareas a realizar en el Sprint actual	4
	Revisión	Revisión de los resultados obtenidos del Sprint con el Scrum Master.	2
	Retrospectiva	Analizar los resultados del Sprint	2
Total de horas			31

Fuente: Propia

- Revisión del Sprint 2

Al finalizar el sprint 2 se comprobó que la estimación de tiempo para cada tarea a tenido algunas variantes ya que en algunas actividades se ha visto en la necesidad de emplear un poco más de tiempo para culminar con las mismas, ya que en muchos casos se

ha encontrado con una cantidad significativa de errores al momento de ejecutar el sistema web, como también se tenía la necesidad de verificar el redireccionamiento de las rutas para el desplazamiento del usuario por las diferentes vistas creadas.

En cuanto a los métodos creados se realizó un análisis minucioso para comprender el proceso que se sigue para cargar, leer, modificar y eliminar las imágenes almacenadas en la base de datos y con ello proceder a utilizar como ejemplo esos métodos para hacer lo mismo para los comentarios que se van a escribir en las imágenes publicadas por los usuarios.

- Retrospectiva del Sprint 2

TABLA 28
RETROSPECTIVA DEL SPRINT 2

		RETROSPECTIVA
Aciertos (¿Qué salió bien del Sprint?)	-	Las investigaciones realizadas ayudaron a comprender de una manera óptima como es el funcionamiento de cada una de las bases de datos y con ello poder modelar las mismas dentro del software DbSchema. - Dentro de área de carga de imágenes el proceso a realizarse fue un éxito ya que el método utilizado ayudó a manipular imágenes y subirlas hacia la base de datos utilizada.
Errores (¿Qué no salió bien del Sprint?)	-	Al principio no se investigó en su totalidad las bases de datos Cassandra, MongoDB y Hbase y por ende se presentaron problemas al modelar las mismas, es por ello que se utilizó más tiempo para hacer una investigación más a fondo para comprender el trato de los atributos dentro de las bases de datos a utilizarse.
Mejoras (¿Qué mejoras se implementará?)	-	La investigación para cada área a utilizarse será analizada debidamente e investigada apropiadamente para que no se presenten problemas en él desarrollo del sistema web y uso de más tiempo de lo requerido para cada actividad.

Fuente: Propia

2.1.9 Sprint 3

- Planificación del Sprint 3

Para el cumplimiento del Sprint 3 es necesario investigar cómo realizar el registro y login de los usuarios y como estos podrán interactuar con la base de datos para obtener los permisos necesarios para poder ingresar a sus respectivas cuentas creadas y de acuerdo a ello los usuarios una vez realizado el proceso de login puedan acceder a las vistas principales que van a ser creadas y manipuladas por los usuarios logeados, dicho de esta manera se procederá a realizar un avance ordenado y debidamente planificado.

a) Spring Backlog

TABLA 29
SPRING BACKLOG DEL SPRINT 3

Historia de Usuario	Fases de Desarrollo	Actividad	Número de horas
RSB-009	Análisis	Especificar cómo funcionará el proceso de registro y login de cada usuario hacia el prototipo de red social, de esta manera se podrá controlar a cada usuario en su respectiva cuenta que va a utilizar.	2
	Desarrollo	Se desarrollará para el registro y login las debidas rutas para poder acceder a la cuenta que cada usuario. Se deberá comprobar que los métodos de autenticación estén controlando la redirección de los usuarios hacia la vista correspondiente	4
	Control	Se desarrollará un método que permita autenticar a los usuarios para que el mismo pueda ingresar a su cuenta con la debida seguridad y no exista robo de información o acceso a cuenta por parte de usuarios no autorizados.	4
RSB-010	Análisis	Especificar como se aplicará el proceso de login hacia las vistas principales que tendrá el usuario una vez que se haya logeado en su cuenta.	1
	Creación	Se crearán las vistas pertinentes en donde el usuario podrá acceder a la información a su cuenta mediante redireccionamiento a las vistas principales que se encontraran dentro de la vista que se presenta después de a ver realizado el login de usuario.	2
	Control	Para verificar el funcionamiento se revisará que cada botón u opción seleccionada del menú principal redirija hacia las vistas pertinentes.	2
Eventos	Planificación	Detallar las tareas a realizar en el Spint actual	5
	Revisión	Revisión de los resultados obtenidos del Sprint con el Scrum Master.	2
	Retrospectiva	Analizar los resultados del Sprint	2
Total de horas			24

Fuente: Propia

- Revisión del Sprint 3

Al culminar el sprint 3 se pudo verificar como proceso de rutas ayuda a que el sistema web tenga una dinámica de ordenada de registro y login de los usuarios, dentro de este proceso se pudo observar que es necesario aplicar ciertas reglas de autenticación que permitan a los usuarios tener cierto grado de seguridad al momento de ingresar a sus cuentas.

Para probar que el proceso de registro y login se encuentran codificados correctamente se han realizado ciertas pruebas para verificar que los mensajes de error están

funcionando debidamente es por ello que a continuación se describen cada uno de los errores que se controlaran dentro del registro y login.

- El usuario al registrarse deberá ingresar una contraseña y la tendrá que reconfirmar y en el caso de que estas no coincidan se presentara un error por pantalla informando al usuario que las contraseñas ingresadas no coinciden.
- El usuario al registrar las contraseñas deberá ingresarlas con un número mayor a cuatro caracteres para poder tener una mayor seguridad en su cuenta, en el caso de intentar ingresar solamente cuatro caracteres como contraseña se presentará un error que informará al usuario que debe ingresar una contraseña mayor a cuatro caracteres.
- El usuario una vez registrado podrá ingresar a su cuenta en el caso de que este ingrese una contraseña errónea el no podrá ingresar a su cuenta ya que se presentará un error que informe al usuario que está ingresando una contraseña incorrecta.

La creación de las vistas fue un aspecto importante dentro del proceso de logeo de los usuarios ya que con ello se pudo comprobar que las rutas creadas han sido perfectamente desarrolladas y que están redirigiendo hacia las vistas apropiadas.

- Retrospectiva del Sprint 3

TABLA 30
RETROSPECTIVA DEL SPRINT 3

		RETROSPECTIVA
Aciertos (¿Qué salió bien del Sprint?)	-	La investigación realizada ayudó a especificar bien los parámetros para el registro y login de los usuarios es por ello por lo que al realizar las debidas pruebas de registro y logeo fueron exitosas ya que los métodos utilizados fueron programados de acuerdo con los patrones especificados en las investigaciones realizadas.
Errores (¿Qué no salió bien del Sprint?)	-	Al principio se presentaron errores con los mensajes que informaban a los usuarios sobre el mal ingreso de contraseña al registrarse, pero después de un análisis hacia la codificación los errores pudieron resolverse sin problemas.
Mejoras (¿Qué mejoras se implementará?)	-	Es necesario optimizar el tiempo para cada actividad y verificar que las actividades se estén cumpliendo en los plazos establecidos.

Fuente: Propia

2.1.10 Sprint 4

- Planificación del Sprint 4

Para el cumplimiento del Sprint 4 es necesario analizar todos los atributos necesarios para especificar los modelos correspondientes que serán almacenados en cada una de las bases de datos NoSQL a utilizarse, como también ir especificando los métodos y promesas que serán importantes para la importación de los modelos hacia las bases de datos.

También será necesario establecer un área donde los usuarios puedan ingresar su información, este proceso será analizado cuidadosamente para que los usuarios autoricen que su información pueda ser ingresada como no ingresada, en caso de ser ingresada esta información podrá ser tratada con los debidos niveles de seguridad establecidos.

a) Spring Backlog

TABLA 31
SPRING BACKLOG DEL SPRINT 4

Historia de Usuario	Fases de Desarrollo	Actividad	Número de horas
RSB-011	Análisis	Considerar los métodos necesarios para poder extraer desde la base de datos la información de los usuarios registrados del prototipo de red social.	1
	Desarrollo	Implementar el método para extraer los datos de los usuarios y desplegarlos dentro de una tabla el nombre y correo electrónico de los usuarios.	3
	Control	Verificar que los datos pedidos dentro del método sean desplegados dentro de la tabla y controlar que no exista errores dentro de la consulta.	1
RSB-012	Análisis	Considerar los atributos del modelo de la base de datos creado con DbSchema y programarlo en el archivo pertinente para poder crear esquema e invocarlo dentro de las clases del proyecto en desarrollo.	1
	Desarrollo	Crear los métodos adecuados para poder ingresar y desplegar la información dentro de las vistas del prototipo de red social.	3
	Control	Verificar mediante pruebas que los métodos estén cumpliendo con las consultas respectivas hacia la base de datos. Dentro de las vistas validar los respectivos campos para que las vistas sean comprensibles para los usuarios.	1
RSB-013	Análisis	Considerar los atributos del modelo de la base de datos creado con DbSchema y programarlo en el archivo pertinente para poder crear esquema e invocarlo dentro de las clases del proyecto en desarrollo.	1
	Desarrollo	Crear los métodos adecuados para poder ingresar y desplegar las publicaciones dentro de las vistas del prototipo de red social. Crear una vista que sea capaz de desplegar cada una de las publicaciones que el usuario requiera realizar.	3
	Control	Verificar mediante pruebas que los métodos estén cumpliendo con las consultas respectivas hacia la base de datos. Comprobar que la vista estilizada sea capaz de desplegar cada una de las publicaciones realizadas con los respectivos atributos ingresados.	1

Eventos	Planificación	Detallar las tareas a realizar en el Spint actual	5
	Revisión	Revisión de los resultados obtenidos del Sprint con el Scrum Master	2
	Retrospectiva	Analizar los resultados del Sprint	2
Total de horas			24

Fuente: Propia

- Revisión del Sprint 4

Los procesos desarrollados durante este sprint contemplaron analizar como los usuarios dentro de sus cuentas podían ver que personas están registradas dentro del prototipo de red social es así como se creó un método para poder obtener esta información mediante una consulta hacia la base de datos para poder desplegar los datos de los usuarios registrados y con ello ver mediante una tabla la información obtenida. Durante este proceso también se pudo crear un modelo de datos para el ingreso de información para los usuarios, este modelo se lo obtiene mediante la verificación del modelo de datos creado en DbSchema, una vez realizado el modelado se procedió a crear un método para guardar la información y otro para desplegarla en una vista, para probar este proceso se realizó los siguientes procedimientos.

- Validación de campos de texto como es para el ingreso de fechas de nacimiento donde existe un límite máximo y mínimo para las fechas, como también se hizo la validación del campo de ingreso de tipo de sexo y selección de carrera.
- Se ingresó la información requerida dentro de los campos de texto y se envió una petición para almacenar dicha información ingresada.
- Control del despliegue de información tanto en el muro del usuario como en la vista de ingreso de información.

En cuanto a las publicaciones respecta se creó un modelo de datos para el ingreso de publicaciones para los usuarios, este modelo se lo obtiene mediante la verificación del modelo de datos creado en DbSchema, una vez realizado el modelado se procedió a crear un método para guardar la información y otro para desplegarla en una vista, para probar este proceso se realizó los siguientes procedimientos.

- Se ingresó la información requerida dentro de los campos de texto y se envió una petición para almacenar dicha información ingresada.
- Se realizó el control del despliegue de información de las publicaciones realizadas.

Al término del cumplimiento de ingreso de información y publicaciones se puede enfatizar que es necesario establecer un modelo de datos adecuado y que cumpla con los atributos necesarios para cumplir con los parámetros señalados en cada historia de usuario.

- Retrospectiva del Sprint 4

TABLA 32
RETROSPECTIVA DEL SPRINT 4

RETROSPECTIVA	
Aciertos (¿Qué salió bien del Sprint?)	- Mediante consultas se pudo obtener la información deseada y se la pudo desplegar tanto en las vistas de contactos, de despliegue de información y de publicaciones, todo este proceso se lo pudo obtener mediante una investigación orientada a como realizar consultas hacia la base de datos.
Errores (¿Qué no salió bien del Sprint?)	- Por cuestiones de redireccionamiento de rutas se tubo ciertos inconvenientes en el despliegue de información en las vistas y debido a ello se tomó más tiempo de lo planificado.
Mejoras (¿Qué mejoras se implementará?)	- Poner más atención a los errores que se muestran por consola y verificar que las rutas que se pidan vayan de acuerdo con las descritas dentro del archivo de rutas.

Fuente: Propia

2.1.11 Sprint 5

- Planificación del Sprint 5

Para el cumplimiento del Sprint 5 es necesario investigar cual es el proceso para implementar un chat entre usuarios y cuáles serían las dependencias adecuadas para poder desarrollar este módulo de mensajería. Como también se debería investigar cuales serían las mejores herramientas en diseño para mejorar las vistas y dar un realce apropiado a cada vista del prototipo de red social.

a) Spring Backlog

TABLA 33
SPRING BACKLOG DEL SPRINT 5

Historia de Usuario	Fases de Desarrollo	Actividad	Número de horas
RSB-014	Análisis	Analizar con que atributos contará el modelo de mensajes y definirlos adecuadamente. Realizar una investigación para ir dando las debidas validaciones al chat entre usuarios y con ello precautelar posibles errores que se puedan encontrar al enviar mensajes.	3
	Desarrollo	Descargar las dependencias necesarias para poder realizar el chat entre usuarios de la red social y con ello darle al prototipo mejores prestaciones y servicios para el usuario. Implementar en el script métodos apropiados para controlar las validaciones correspondientes en los chats.	8
	Control	Verificar que las validaciones programadas estén dadas tanto para la parte del servidor como del cliente.	5
	Pruebas	Ejecutar el sistema web y realizar envíos de mensajes entre usuarios para comprobar que los	4

		mismo no presenten fallas al momento de ser enviados. Mediante manipulación de las cuentas controlar que las validaciones impuestas tengan la valía necesaria para fortalecer el envío de mensajes de texto.	
RSB-015	Análisis	Analizar qué tipo de cambios se le puede realizar a las vistas del prototipo de red social y seleccionar las mejores herramientas de diseño de Bootstrap 4 y CSS.	2
	Desarrollo	Descargar las extensiones apropiadas y con ello hacer el llamado de los componentes necesarios en las etiquetas de los archivos handlerbars que han sido creados en el prototipo de red social.	3
Eventos	Planificación	Detallar las tareas a realizar en el sprint actual	5
	Revisión	Revisión de los resultados obtenidos del Sprint con el Scrum Master	2
	Retrospectiva	Analizar los resultados del Sprint	2
Total de horas			34

Fuente: Propia

- Revisión del Sprint 5

Creación de un modelo de datos para realizar el chat entre los usuarios, este modelo se lo obtiene mediante la verificación del modelo de datos creado en DbSchema, una vez realizado el modelado se procedió a implementar las configuraciones necesarios tanto de la parte del servidor como del cliente, dichas configuraciones fueron desarrolladas mediante la utilización de socket.io y jquery para poder controlar las funcionalidades y atributos del modelos de datos, de esta manera se pudo también realizar las validaciones para poder enviar los mensajes en tiempo real a cada una de las cuentas de los usuarios que se encuentren registrados dentro de la base de datos.

Para cumplir con el desarrolló se enfocó en la mejora de las vistas mediante la utilización de Bootstrap 4 y CSS las cuales han sido herramientas fundamentales para diseñar el prototipo de red social y que cuente con ciertas funcionalidades indispensables para el mejoramiento de las vistas del prototipo de red social.

- Retrospectiva del Sprint 5

TABLA 34
RETROSPECTIVA DEL SPRINT 5

RETROSPECTIVA	
Aciertos (¿Qué salió bien del Sprint?)	- Con la investigación y utilización de socket.io y jquery se pudo manipular óptimamente las funcionalidades de envío y recepción de mensajes en tiempo real para los chats entre usuarios.
Errores (¿Qué no salió bien del Sprint?)	- Se tuvo ciertos inconvenientes al momento de almacenar los mensajes dentro de la base de datos debido a la complejidad que se presentó al momento de utilizar jquery.

Mejoras (¿Qué mejoras se implementará?)	- Investigar sobre las funcionalidades y modo de uso de jquery para poder desarrollar mejor los proyectos a futuro donde se pueda utilizar esta dependencia.
------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------

Fuente: Propia

2.2 Diseño

Es una fase del desarrollo del software, que se encarga de estructurar la arquitectura del software mediante procesos de calidad para cumplir con los requerimientos del cliente que han sido remarcados en las historias de usuario. Los procesos del diseño permiten a los diseñadores interpretar todos los aspectos necesarios de los sistemas web a construir. (Mero,2020).

Durante el diseño del software se evalúa la calidad del desarrollo del proyecto y para ello se realizará procedimientos de revisión técnica como los siguientes:

- a) Los diseñadores deben efectuar todos los requerimientos explícitos y acumular los requerimientos implícitos ya que al implementarlos en las guías correspondientes deben ser claros y concisos para aquellas personas que se encarguen de desarrollar el código y los encargados de realizar las pruebas mantengan la integridad del código desarrollado. (Mero,2020).
- b) Para que el diseño sea efectivo este debe tener una idea íntegra de lo que va a realizar el sistema web y con ello contemplar en la totalidad la funcionalidad, comportamiento y dominio de los datos es decir que debe cumplir con todos los estándares de calidad que espera el cliente. (Mero,2020).

2.2.1 Arquitectura del Sistema Web

El sistema web a desarrollarse utiliza el lenguaje de programación javascript como base para desarrollar el prototipo de red social, mientras que para el entorno de desarrollo se ha visto necesario el uso de node.js con el framework express, en cuanto a las bases de datos NoSQL se seleccionó tres tipos como: Apache Cassandra, MongoDB y Redis las cuales fueron definidas desde el principio para realizar el estudio de las mismas.

Para entender mejor el proceso a seguir se ha diseñado una arquitectura que contempla como estará funcionando el prototipo de red social, tal como se puede observar en la Fig. 17.

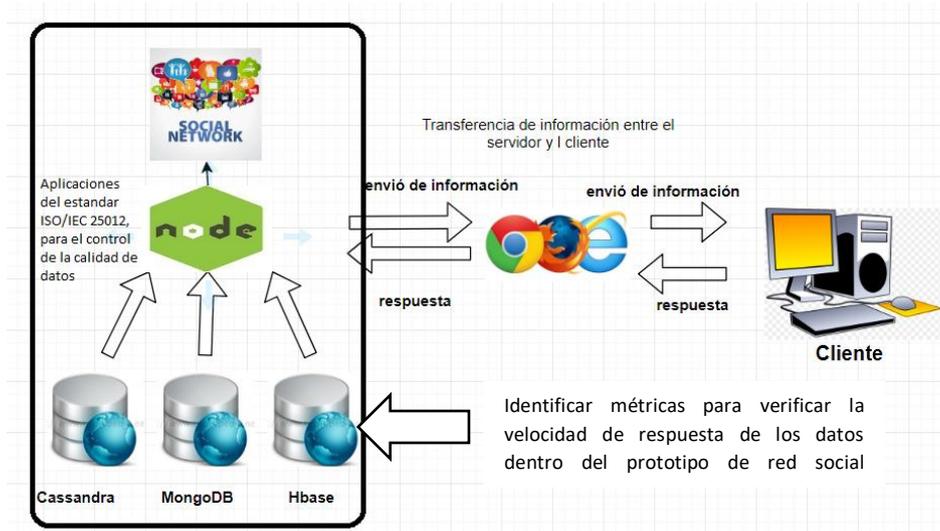


Fig. 17 Arquitectura del prototipo de red social

2.2.2 Diagrama del modelado de las bases de datos

En la Fig. 18 se identifica la elaboración del modelo de datos de MongoDB en la herramienta DbSchema.

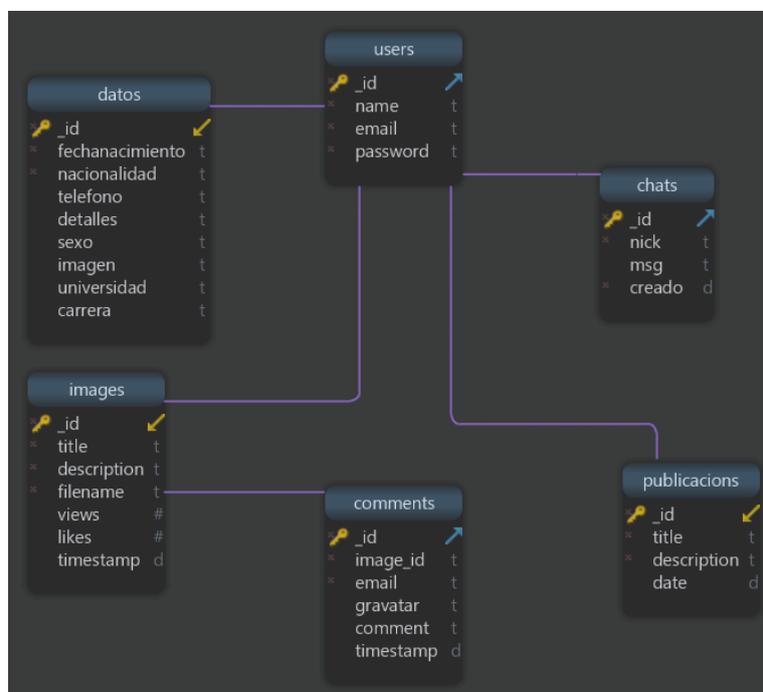


Fig. 18 Modelado de Datos de MongoDB

Mientras que en la Fig. 19 se identifica la elaboración del modelo de datos de Apache Cassandra en la herramienta DbSchema.

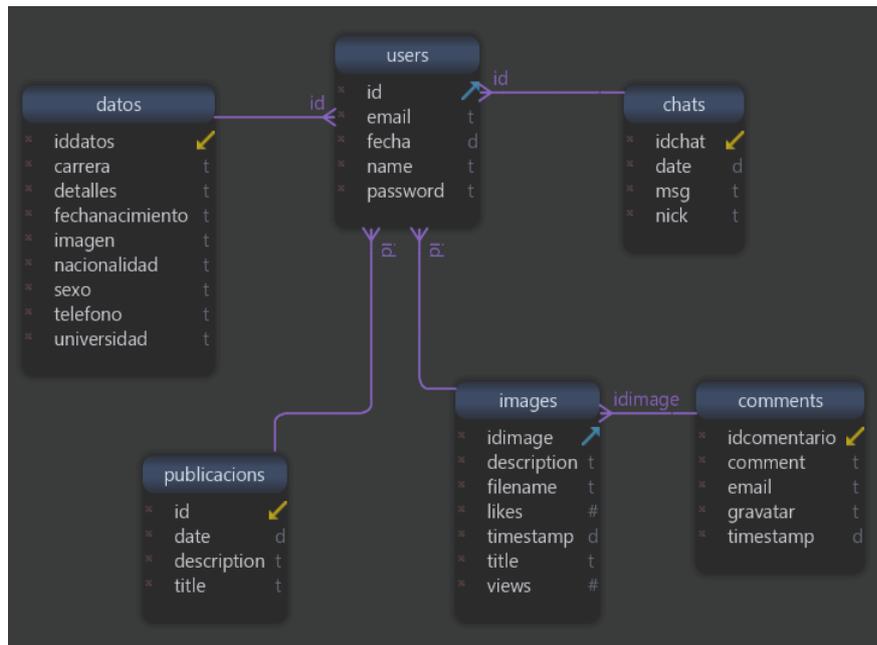


Fig. 19 Modelado de Datos de Apache Cassandra

En cuanto a la Fig. 20 se identifica la elaboración del modelo de datos de Redis en la herramienta DbSchema.

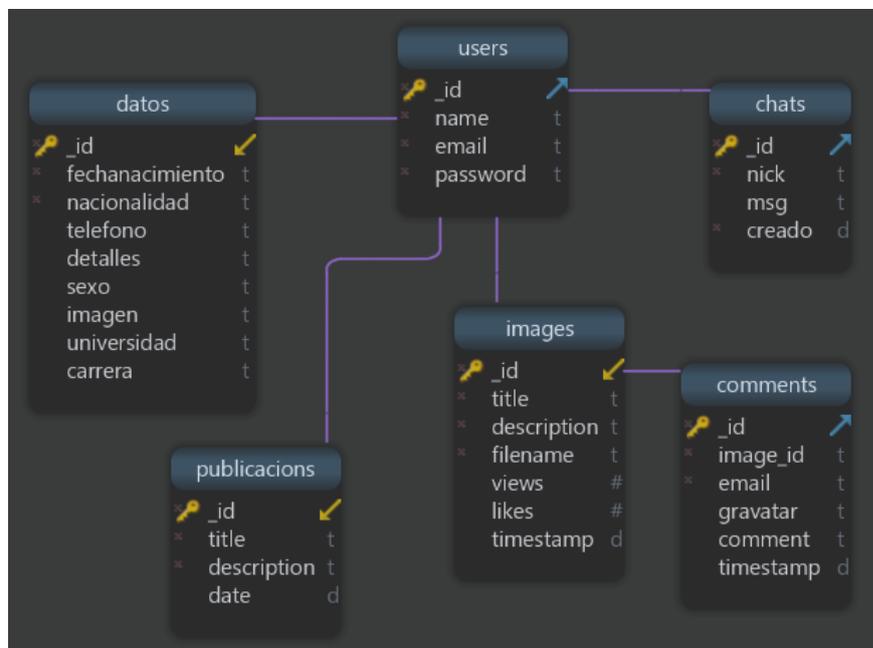


Fig. 20 Modelado de Datos de Redis

2.3 Desarrollo del prototipo de red social

De acuerdo con lo realizado en el proceso de desarrollo y una vez finiquitado cada sprint se presentan los resultados obtenidos en cada uno de ellos.

2.3.1 Sprint 1

TABLA 35
SPRINT 1

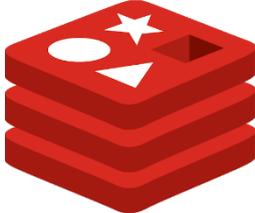
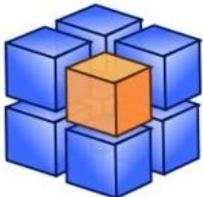
Historias de Usuario	Descripción
RSB-002	Establecer funcionalidades
RSB-003	Modelado de Bases de Datos
RSB-004	Investigación de herramientas para el desarrollo del sistema web

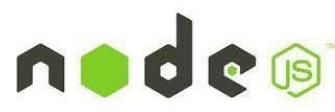
Fuente: Propia

Con las tareas designadas en el Sprint 1 las cuales se puede ver en la tabla 35 y de acuerdo con ella se pudo establecer las funcionalidades que va a tener el prototipo de red social en donde se definió que la misma prestará diferentes servicios como publicación de imágenes, videos, documentos, chatear y verificación de seguidores, de esta manera al tener claro las funcionalidades, se procedió a crear los respectivos atributos que tendrá cada documento y tabla dentro de las bases de datos Apache Cassandra, MongoDB y Redis .

Se prosiguió con la realización de una investigación donde se definían las herramientas necesarias que se van a utilizar para el desarrollo del sistema web, tal como se puede observar en la tabla 36.

TABLA 36 SOFTWARE ESPECIALIZADO PARA EL DESARROLLO

Software	Descripción
	Es un tipo de Base de Datos distribuida, de alto rendimiento tanto en entrada como de salida de datos por lo que la hace productivamente escalable y congruente en manejar grandes de cantidades de información.
	Se lo reconoce como un sistema de bases de datos multiplataforma que se orienta a documentos, por lo que al realizar sus registros con esquemas de datos diferentes y con atributos definidos hace que el mismo no se repita y no haya redundancia en los datos ingresados.
	Es considerado un agente de almacenamiento de información de base de datos, caché y mensajes, posee una estructura interna que permite utilizar estructuras de datos como bitmaps, listas, hashes e índices geoespaciales los cuales permiten hacer consultas de radius queries y streams.
	DbSchema es un software especializado para el modelamiento de diferentes tipos de bases de datos SQL como NoSQL, al poseer una integración de diferentes esquemas y facilidad de descarga de controladores de las diferentes bases de datos hace que este software sea adecuado y eficaz para los diseñadores de bases de datos.

	<p>Visual Studio Code es un editor de código fuente que incluye entre sus servicios soporte de depuración, control de Git integrado, como también puede llegar a soportar diferentes lenguajes de programación como: C#, F#, Visual Basic, PHP, Python, Perl, SQL, Java, Git como también acepta lo que es programación de hojas de estilo con html, CSS y JavaScript.</p>
	<p>Es un software de código abierto multiplataforma que posee herramientas adecuadas para desarrollar cualquier tipo de proyecto, este software está basado en el lenguaje de programación ECMAScript que fortalece la sintaxis y características que brinda javascript.</p>
	<p>Es el framework más conocido de node.js, cuenta con varias características que lo hacen un framework robusto, rápido y flexible, además de contar con once middleware poderosos y session-handler que lo hacen único en su clase.</p>
 <pre> }, "keywords": [], "author": "", "license": "ISC", "dependencies": { "bcryptjs": "^2.4.3", "connect-flash": "^0.1.1", "errorhandler": "^1.5.1", "express": "^4.17.1", "express-handlebars": "^3.1.0", "express-session": "^1.16.2", "fs-extra": "^8.1.0", "md5": "^2.2.1", "method-override": "^3.0.0", "mongoose": "^5.6.2", "morgan": "^1.9.1", "multer": "^1.4.1", "passport": "^0.4.0", "passport-local": "^1.0.0", "timeago.js": "^4.0.0-beta.2", "uuid": "^3.3.2" }, </pre>	<p>Librerías o módulos de node.js que se especializan cada uno en cumplir una función respectiva dentro del desarrollo del prototipo de red social. Cada librería será invocada en las clases respectivas para poder usar sus características y poder ir estructurando los servicios que proporcionará la red social.</p>

Fuente: (Saraza,2019)

2.3.2 Sprint 2

TABLA 37
SPRINT 2

Historias de Usuario	Descripción
RSB-005	Inicio del desarrollo del sistema web
RSB-006	Gestión de Rutas y creación de directorios dentro del proyecto en desarrollo
RSB-007	Almacenamiento de imágenes en la base de datos
RSB-008	Creación de etiquetas y panel de comentarios para las imágenes

Fuente: Propia

Con las tareas designadas en el Sprint 2 se consiguió dar el inicio al desarrollo del prototipo de red social en el cual como primer paso fue la configuración del servidor y la conexión de node.js con las bases de datos de MongoDB y de igual manera con Apache Cassandra y Redis, en la Fig. 21 se muestra los mensajes por consola que muestran que el servidor y la conexión hacia la base de datos se encuentran en perfecto estado.

```

26 //Settings
27 app.set('port',process.env.PORT ||3003);
28 app.set('views',path.join(__dirname,'../views'));
29 app.engine('.hbs',exphbs({
30   defaultLayout:'main',
31   layoutsDir:path.join(app.get('views'),'layouts'),
32   partialsDir:path.join(app.get('views'),'partials'),
33   extname:'.hbs',
34   helpers:require('./helpers')
35 }));
36 app.set('view engine','.hbs');
37
38 //middlewares
39 app.use(morgan('dev'));
40 app.use(multer({dest:path.join(__dirname,'../public/upload/temp')}}).single('image'));
41 app.use(express.urlencoded({extended:false}));
42 app.use(express.json());
43
44 //Middlewares del otro proyecto
45 app.use(express.urlencoded({extended:false}));
46 app.use(methodOverride('_method'));
47 app.use(session({
48   secret:'secret',
49   resave:true,
50   saveUninitialized:true
51 }));
52 app.use(passport.initialize());
53 app.use(passport.session());
54 app.use(flash());

```

Fig. 21 Configuración del Servidor

Se crea una vista principal que da la bienvenida al prototipo de red social con su respectiva ruta, ya que al momento de ejecutar el proyecto saldrá como vista principal la Fig. 22.

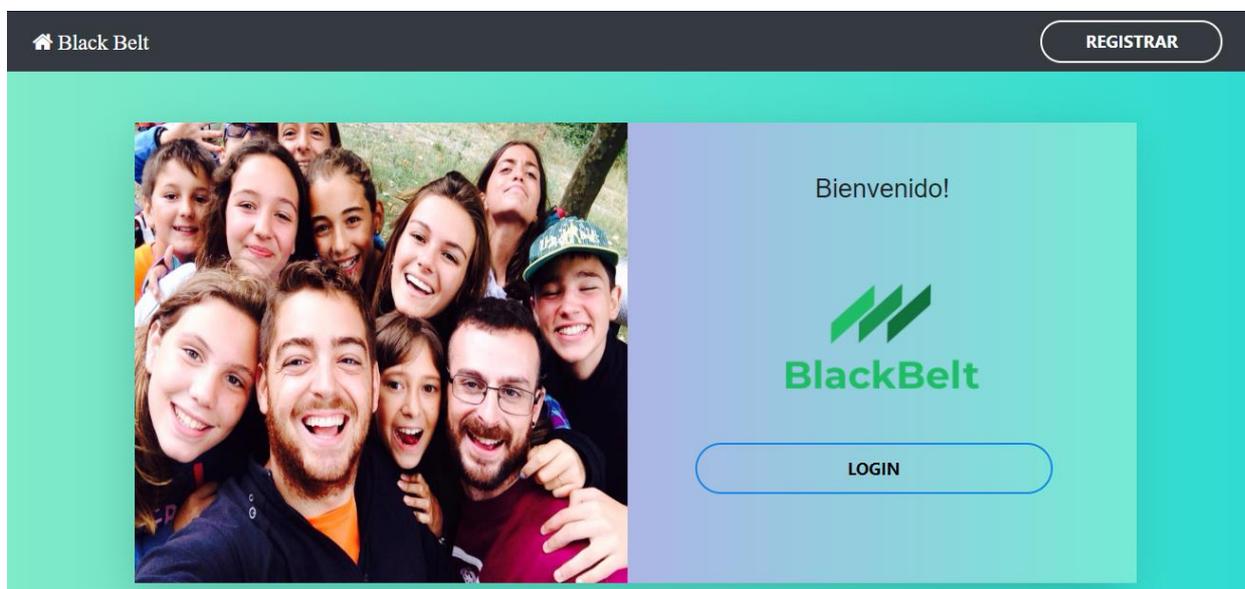


Fig. 22 Vista principal del prototipo de red social

En el archivo index.js de las rutas se crea la línea de código para la ruta de registro tal como se presenta en la Fig. 23.

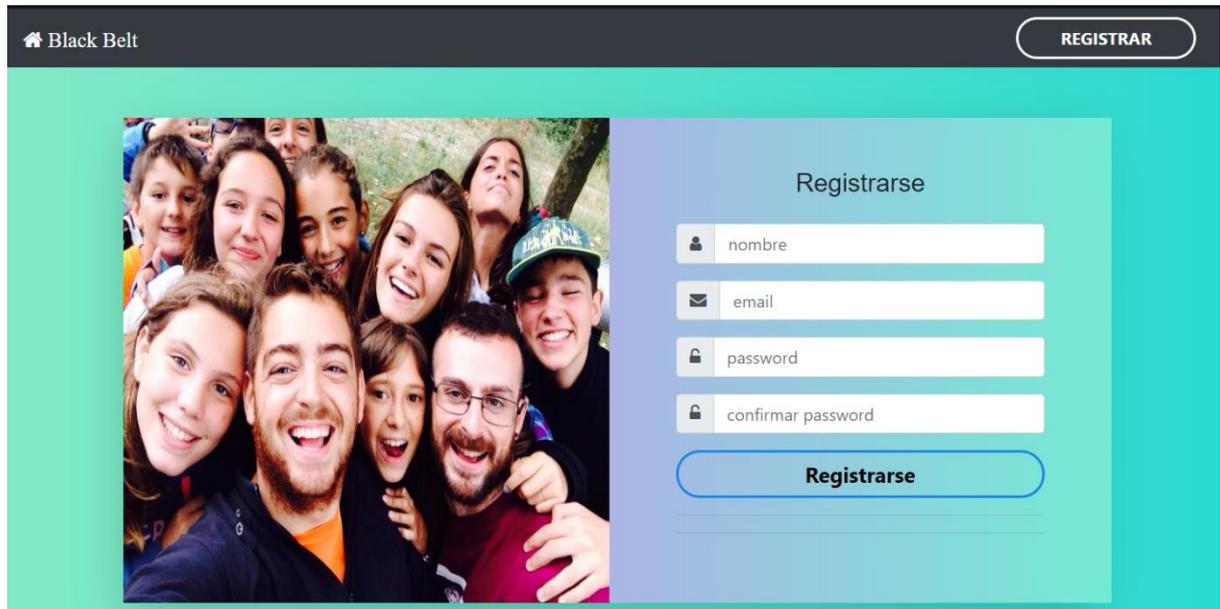


Fig. 23 Vista de Registro de Usuarios

En el archivo index.js de las rutas se crea la línea de código para la ruta de Login tal como se observa en la Fig. 24.

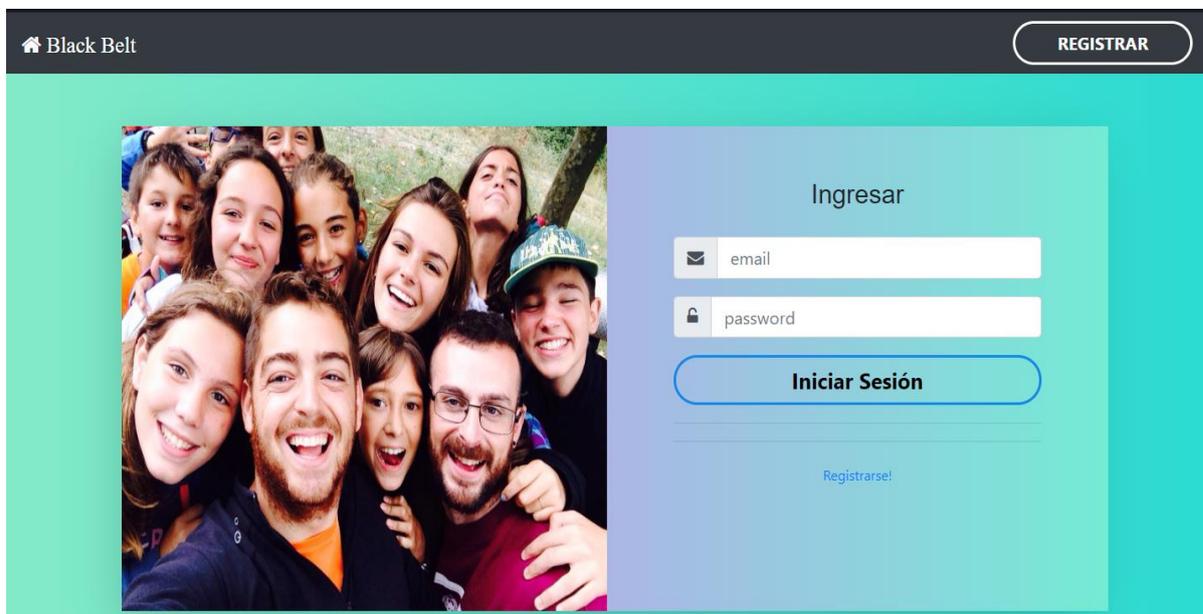


Fig. 24 Vista de Login de Usuarios

De acuerdo con las historias de usuario, como siguiente paso en el desarrollo es crear una ruta que especifica el redireccionamiento a la vista de carga y presentación de imágenes tal como se identifica en la Fig. 25.



Fig. 25 Vista para Carga y despliegue de Imágenes

Con respecto a la figura anterior se añadirá una ruta que al dar clic en una de las imágenes esta se redireccionará a una vista en donde se muestre la imagen seleccionada con sus respectivas etiquetas tal como se muestra en la Fig. 26.

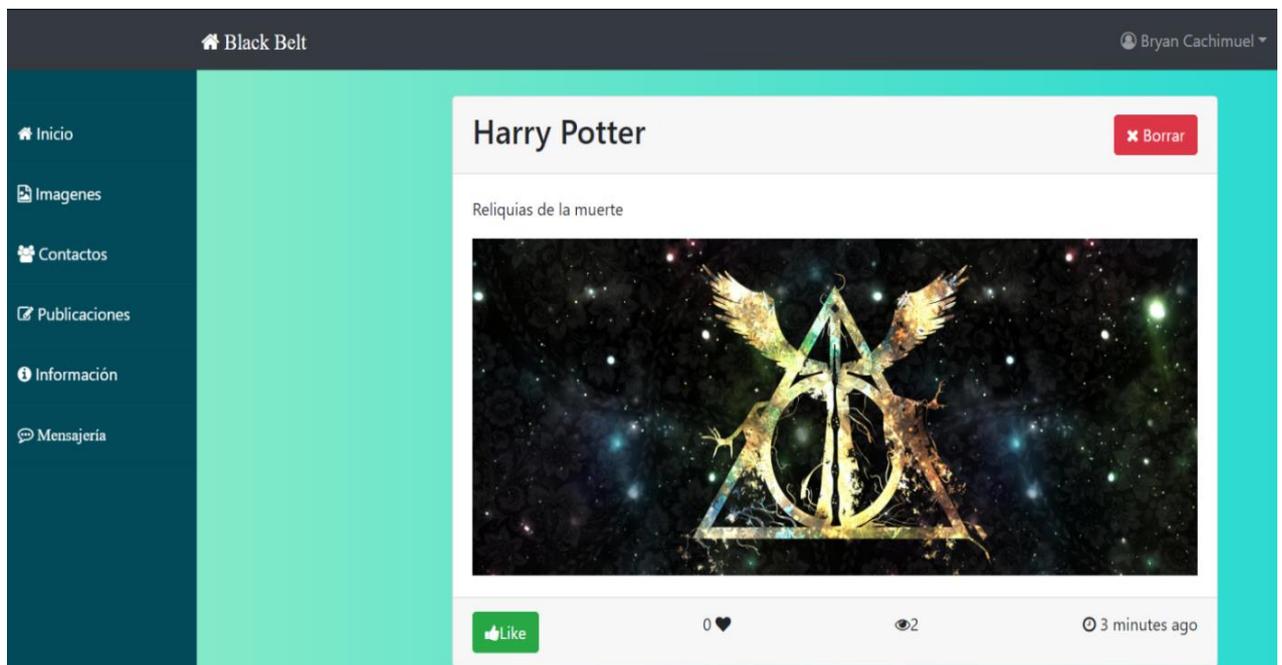


Fig. 26 Redirección hacia imágenes y etiquetas

Para completar el procedimiento de carga de imágenes se ha creado un panel para la escritura de comentarios tal como se muestra en la Fig. 27.

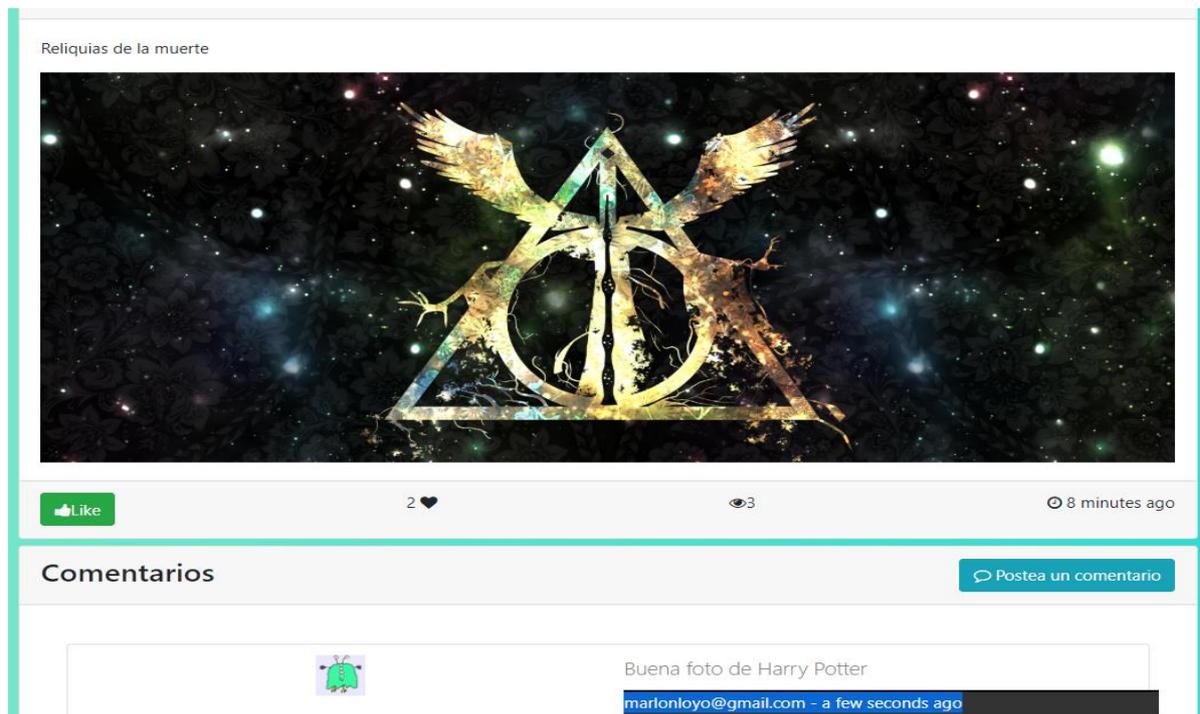


Fig. 27 Comentarios en las imágenes

2.3.2 Sprint 3

TABLA 38
SPRINT 3

Historias de Usuario	Descripción
RSB-009	Creación de Registro y Login
RSB-010	Creación de vistas principales que tendrá el usuario logeado

Fuente: Propia

Con las tareas designadas en el Sprint 3 se consiguió cumplir con los parámetros necesarios para realizar el registro y login de los usuarios del prototipo de red social y para cumplir con estas especificaciones sea realizado las siguientes validaciones.

Un usuario nuevo al intentar registrarse debe ingresar toda la información que se le solicite, en caso de no ingresar la información en uno de los campos el sistema web no le permitirá realizar su registro tal como se observa en la Fig. 28.

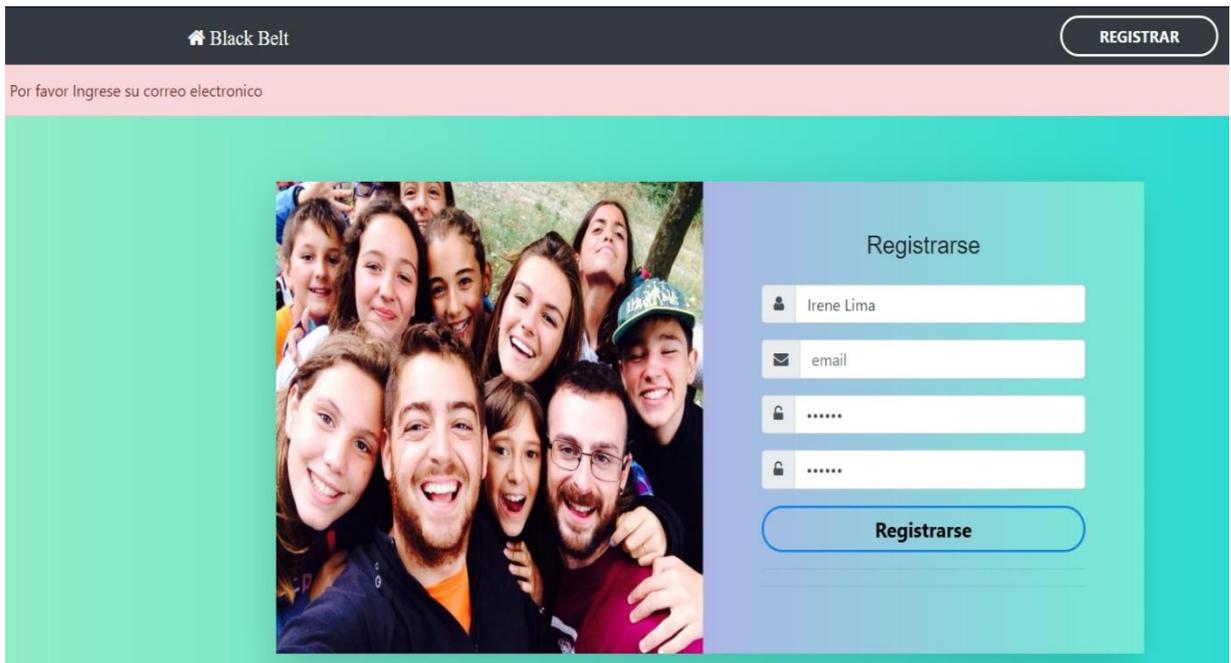


Fig. 28 Validación de campos de texto

Al registrarse se presentan dos campos para ingresar la contraseña en donde se pide en un campo el ingreso de la contraseña y en el otro la confirmación de la misma en caso de que se ingrese diferente contraseña se presentara una advertencia tal como se muestra en la Fig. 29.

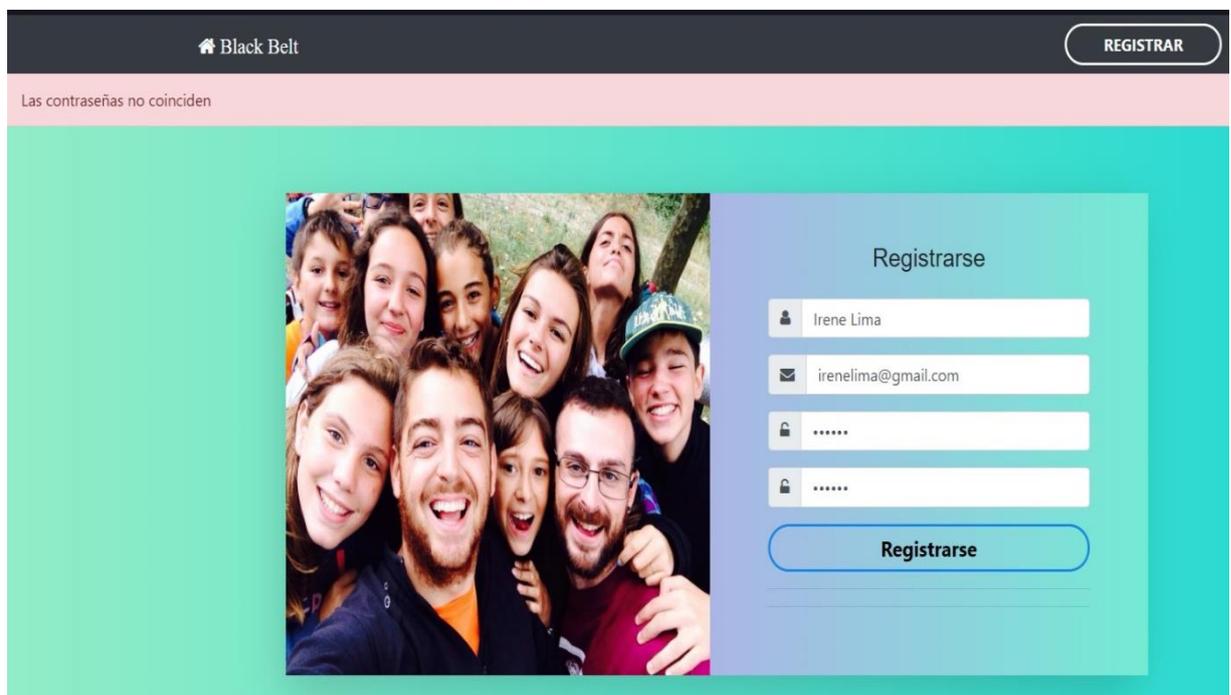


Fig. 29 Validación de contraseñas

Otro aspecto validado dentro de los campos de texto es que debe ingresar una contraseña mayor a cuatro caracteres para que esta pueda ser almacenada en la base de datos, en el caso de que el usuario solo menos caracteres el sistema web mostrara una advertencia tal como se muestra en la Fig. 30.

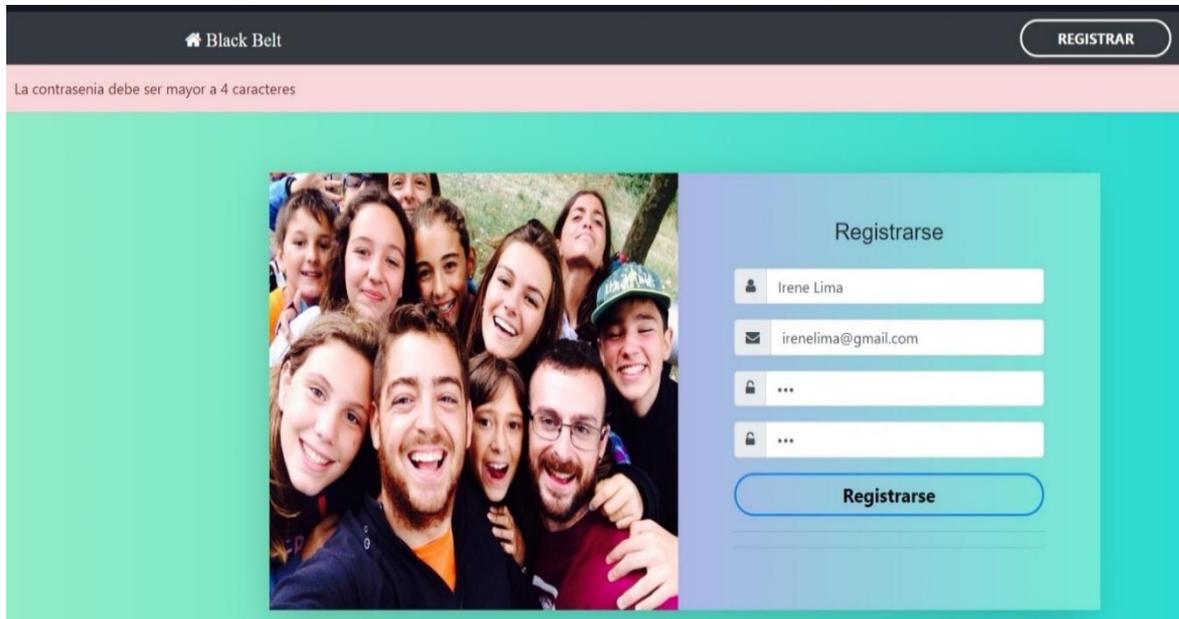


Fig. 30 Validación para el número de caracteres de contraseña

En la Fig. 31 se puede identificar que para la sección de contraseñas se ha utilizado una dependencia llamada bcrypt que permite cifrar la contraseña en MongoDB y las almacena dentro de las misma como un código de tipo md5.

```
_id: ObjectId("5de69a7350eeca1f0441c52a")
name: "Adriana Diaz"
email: "adrianadiaz@gmail.com"
password: "$2a$10$DiJclYUT63FCIeBj3Cm308TXDOEtAsTiLK23qH57eI/SnTfjmqbK"
date: 2019-12-03T17:25:07.736+00:00
__v: 0
```

Fig. 31 Encriptación de contraseñas en MongoDB

Mientras que para el sistema web conectado hacia Apache Cassandra se ha seleccionado para las contraseñas una dependencia llamada md5 que permite cifrar la contraseña como un dato tipo hash, tal como se identifica en la Fig. 32.

```
cqlsh:social> select*from users;

id | email | fecha | name | password
-----+-----+-----+-----+-----
4fb80a41-2968-47e9-9621-92d88b34bec2 | bryanloyo56@gmail.com | 2020-02-07 03:52:56.672000+0000 | Bryan Cachimuel | e10adc3949ba59abbe56e057f20f883e

(1 rows)
cqlsh:social> 
```

Fig. 32 Encriptación de contraseñas en Apache Cassandra

En cuanto al sistema web conectado a redis respecta, en este caso, javascript no cuenta con una librería adecuada para almacenar contraseñas cifradas y debido a ello las contraseñas se almacenan tal como ingresa en el registro de usuarios, tal como se muestra en la Fig. 33.

```
127.0.0.1:6379> hgetall bryanloyo@gmail.com_
1) "name"
2) "Bryan Cachimuel"
3) "email"
4) "bryanloyo@gmail.com"
5) "password"
6) "123456"
```

Fig. 33 Sin disposición de dependencia para cifrado de contraseñas para Redis

El usuario una vez que haya realizado el registro podrá ingresar a la vista de login para poder logearse a su cuenta, en caso de que el usuario ingrese una contraseña incorrecta se presentara una advertencia, tal como se muestra en la Fig. 34.

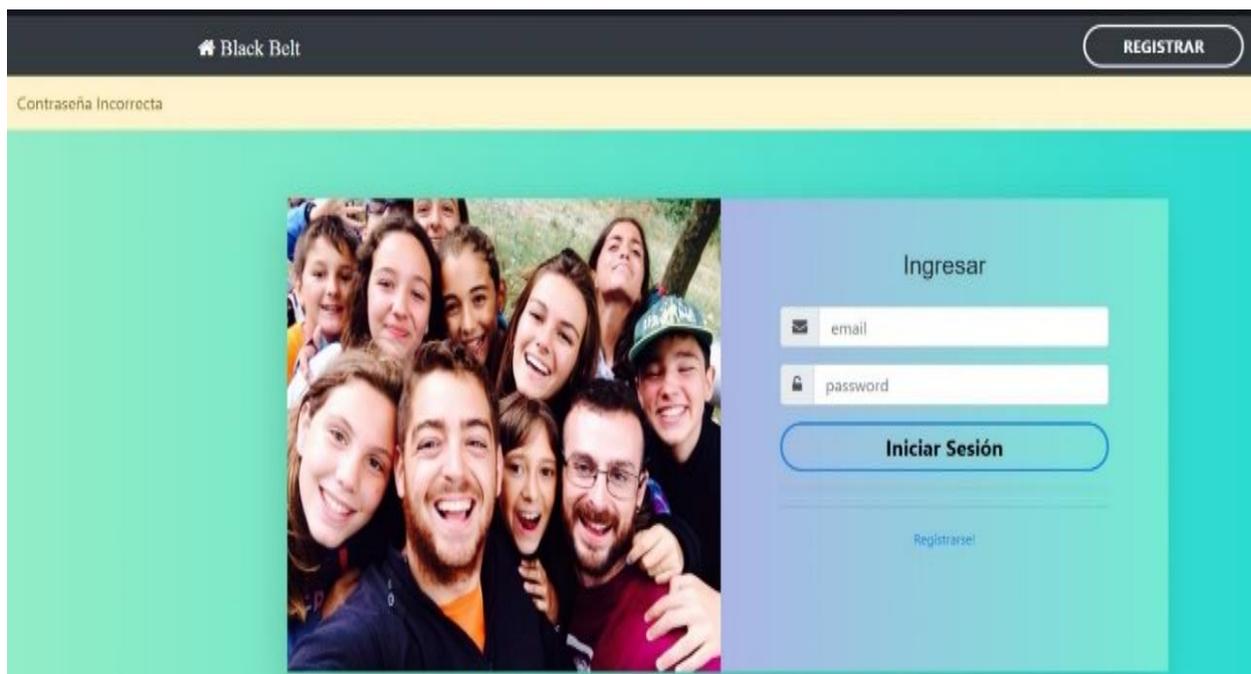


Fig. 34 Validación de contraseñas en el Login

El usuario en el caso de que haya ingresado sus datos correspondientes para logearse podrá ingresar a su cuenta y podrá ver la vista principal del prototipo de red social, tal como se muestra en la Fig. 35.

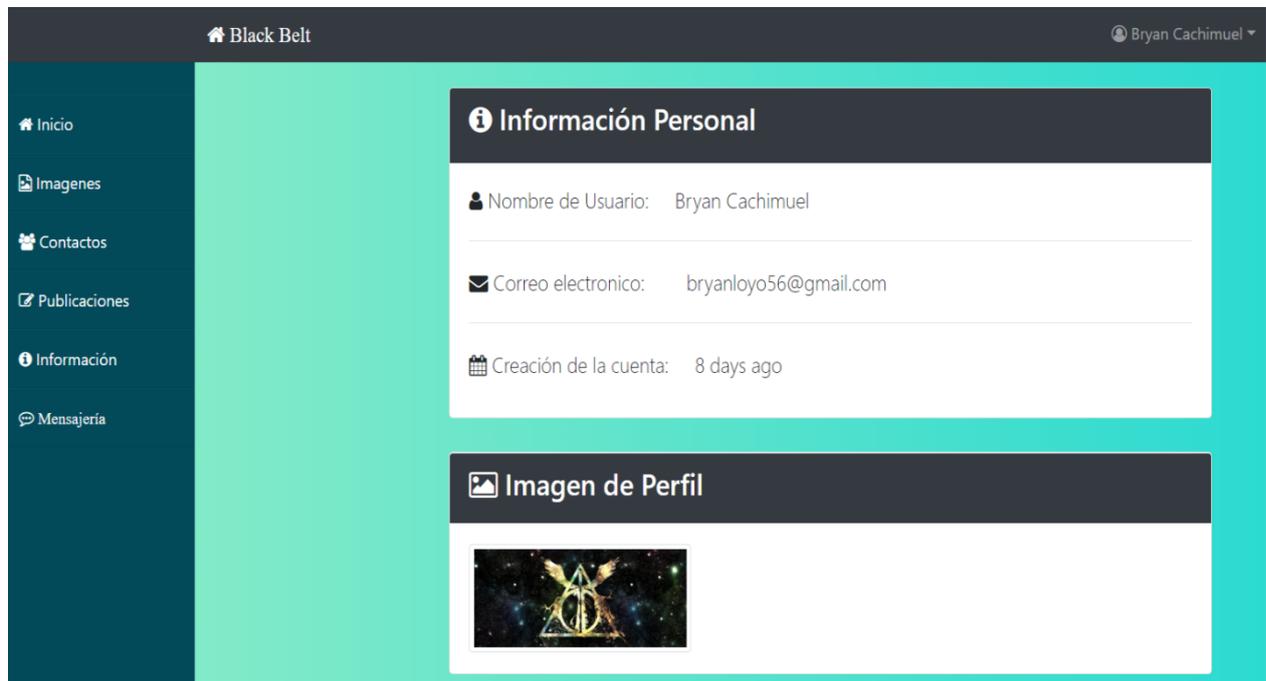


Fig. 35 Vista principal del prototipo de red social

2.3.2 Sprint 4

TABLA 39
SPRINT 4

Historias de Usuario	Descripción
RSB-011	Contactos dentro del prototipo de red socia
RSB-012	Ingreso y despliegue de Información del Usuario
RSB-013	Modelo de Publicaciones de estados y programación en el servidor y cliente

Fuente: Propia

Con las tareas designadas en el Sprint 4 se consiguió cumplir con los parámetros necesarios para realizar el registro y login de los usuarios del prototipo de red social y para cumplir con estas especificaciones se ha realizado las siguientes validaciones.

En la sección de contacto se creó un método para desplegar la información de todos los usuarios registrados dentro de la base de datos y con ello verificar en una vista dicha información, la misma que se encuentra reflejada en una tabla de contactos de la universidad la cual cuenta con dos secciones para el nombre y email, así como se ve en la Fig. 36.

Contactos de la Universidad	
Nombre	Email
Marlon Cachimuel	marlonloyo@gmail.com
Josue Alba	josuealba@gmail.com
Adriana Diaz	adrianadiaz@gmail.com
Angelita Ruano	angelitaruano@gmail.com
Fernanda Ruano	fernandaruano@gmail.com
Bryan Cachimuel	bryanloyo56@gmail.com
Joselyn Quiroz	joselynquiroz@gmail.com
Ismael Quiroz	ismaelquiroz@utn.edu.ec
Michelle Lara	michellelara@outlook.com
Francisco Egas	franciscoegas@gmail.com

Fig. 36 Lista de Contactos

Se realizó las respectivas validaciones de cuadros de texto como lo es para ingreso de fecha de nacimiento, selección de carrera y selección de género del usuario, tal como se demuestra en la Fig. 37.

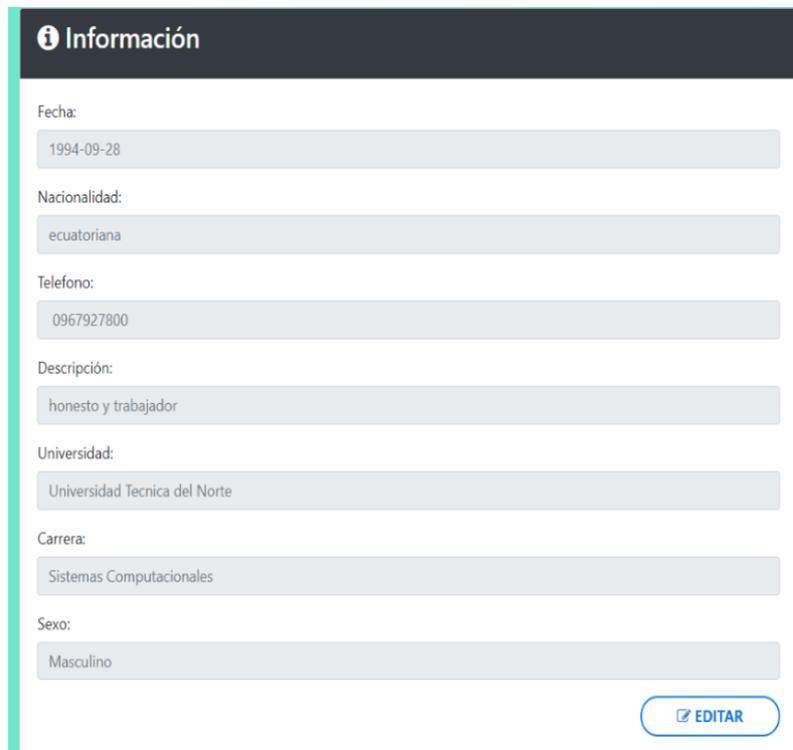
The screenshot shows a web application interface with a sidebar on the left containing menu items: Inicio, Imágenes, Contactos, Publicaciones, Información, and Mensajería. The main content area displays a modal form titled "Complete la Información" with the following fields:

- Fecha:** A date picker showing "febrero de 2020" with a calendar view. The date "6" is selected.
- Descripción:** A text input field containing "honesto y trabajador".
- Universidad:** A text input field containing "Universidad Tecnica del Norte".
- Carrera:** A dropdown menu with "Sistemas Computacionales" selected.
- Sexo:** A dropdown menu with "Masculino" selected.

A green "Guardar" button is located at the bottom right of the form.

Fig. 37 Ingreso de información de los Usuarios

Una vez ingresada la información de procedió a realizar el despliegue de la misma dentro de un panel donde se podía ver la información ingresada anteriormente, tal como se muestra en la Fig. 38.



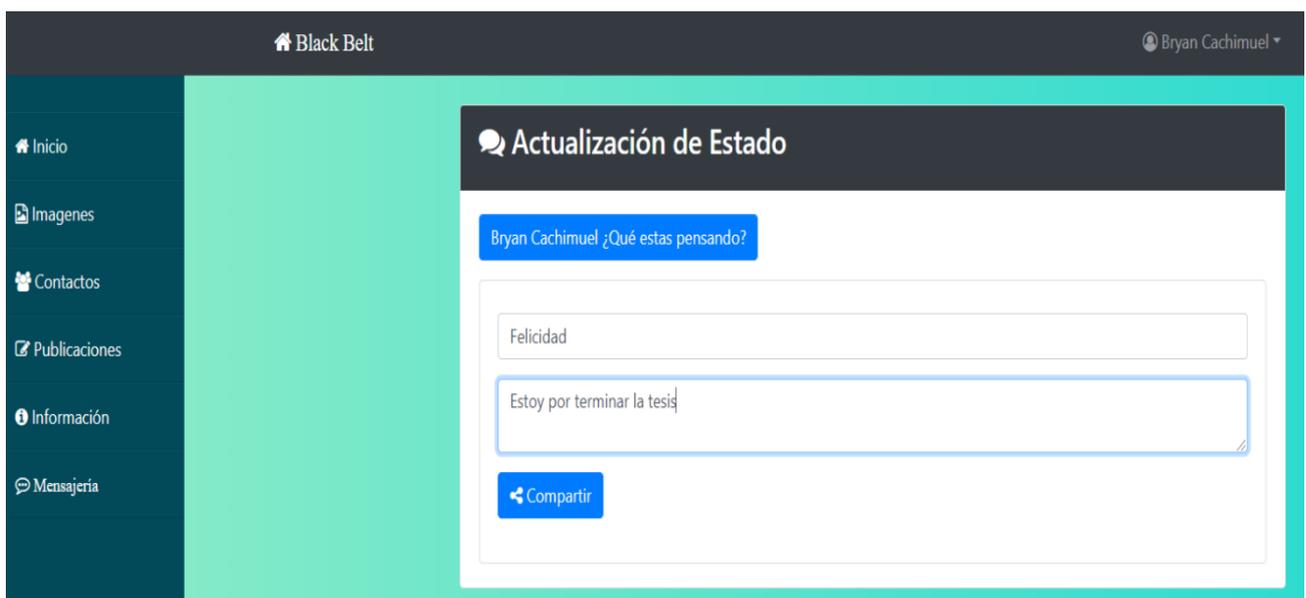
The screenshot shows a user profile information panel with the following fields and values:

- Fecha: 1994-09-28
- Nacionalidad: ecuatoriana
- Telefono: 0967927800
- Descripción: honesto y trabajador
- Universidad: Universidad Tecnica del Norte
- Carrera: Sistemas Computacionales
- Sexo: Masculino

An 'EDITAR' button is located at the bottom right of the panel.

Fig. 38 Despliegue de Información

Otra opción desarrollada es la de realizar publicaciones de estados en donde el usuario podrá ingresar información de su estado de ánimo y la podrá publicar en su muro, tal como se identifica en la Fig. 39.



The screenshot shows a social media interface for creating a post. The header includes 'Black Belt' and the user name 'Bryan Cachimuel'. The main content area is titled 'Actualización de Estado' and contains the following elements:

- A blue button with the text 'Bryan Cachimuel ¿Qué estas pensando?'.
- A dropdown menu for mood selection, currently showing 'Felicidad'.
- A text input field containing the text 'Estoy por terminar la tesis'.
- A blue 'Compartir' button.

A sidebar on the left contains navigation options: Inicio, Imagenes, Contactos, Publicaciones, Información, and Mensajería.

Fig. 39 Ingreso de información para publicar

Una vez ingresada la información se la pública y se despliega la misma en un recuadro en la parte inferior, tal como se muestra en la Fig. 40.



Fig. 40 Despliegue de las publicaciones

2.3.2 Sprint 5

TABLA 40
SPRINT 5

Historias de Usuario	Descripción
RSB-014	Creación del Modelo para Mensajes con sus respectivas configuraciones
RSB-015	Modificar el diseño de las vistas mediante el uso de Bootstrap 4 y CSS

Fuente: Propia

Con las tareas designadas en el Sprint 5 se consiguió crear el módulo de chats entre usuarios en este proceso se definió los atributos necesarios para poder realizar el chat entre los usuarios registrados.

De acuerdo con la Fig. 41 se puede verificar que dos usuarios se encuentran conectados y se encuentran listos para chatear en este caso se puede ver que solo dos usuarios están conectados tal como se muestra en la sección que dice Usuarios donde se ve que Bryan y Michelle están en línea.

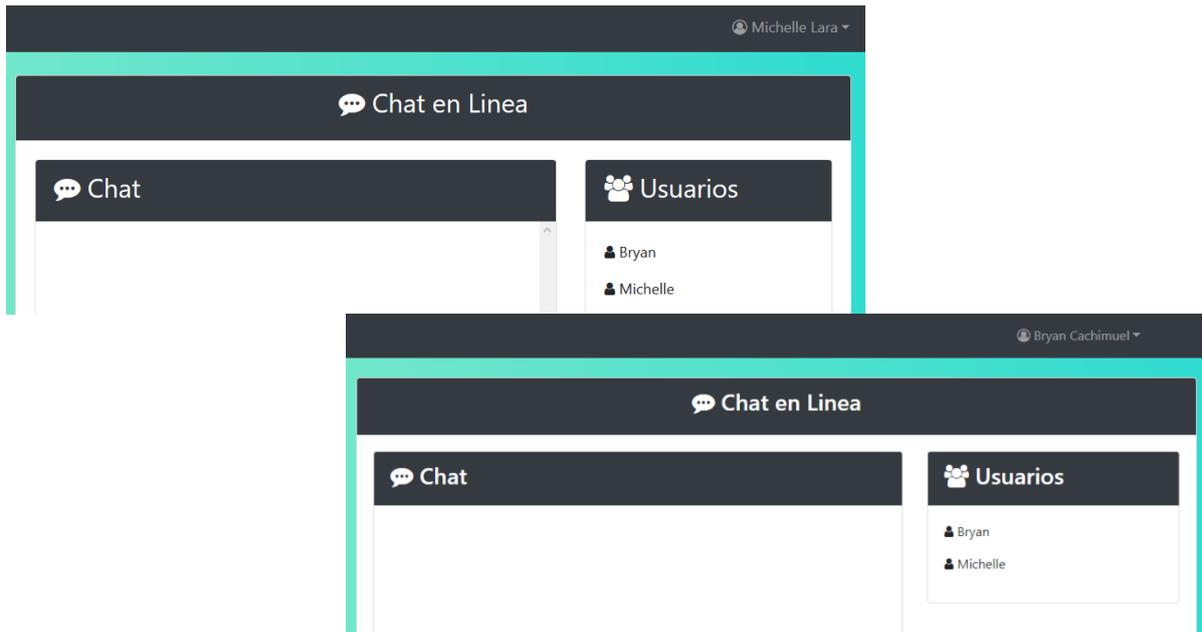


Fig. 41 Chat entre dos usuarios

Una vez los usuarios se encuentren conectados al prototipo de red social ellos podrán chatear tal como se la Fig. 42.

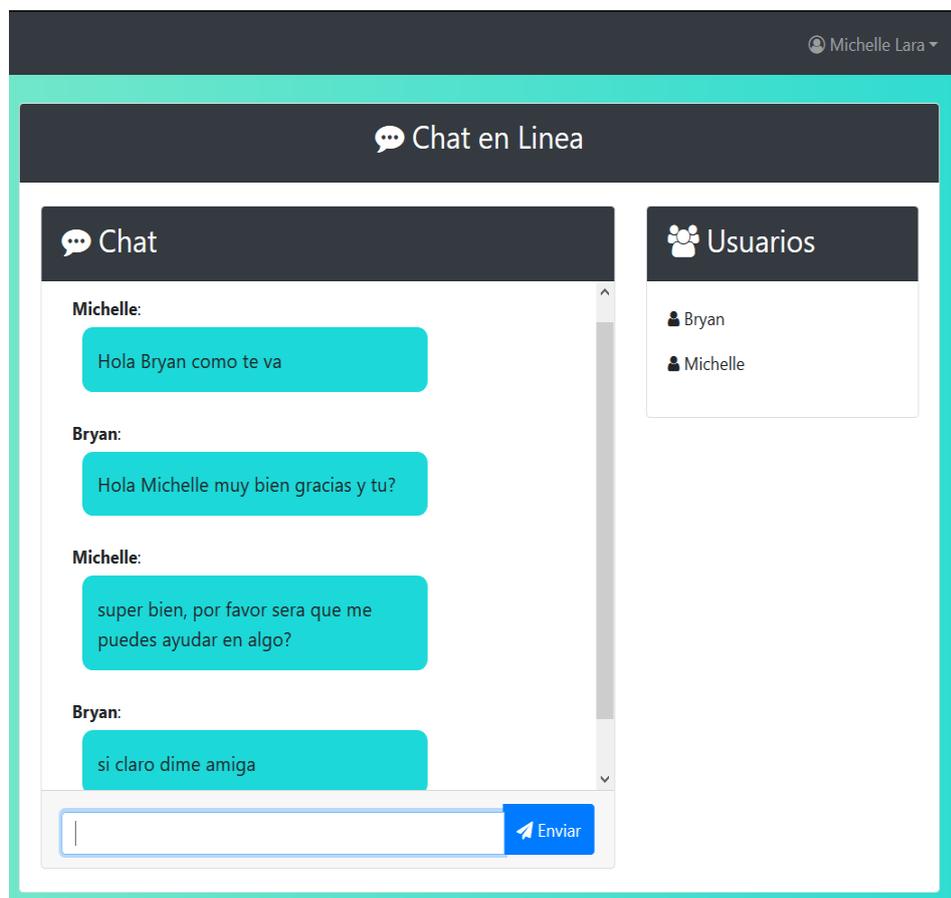


Fig. 42 Chat Visto desde el Usuario Michelle Lara

El usuario Bryan Cachimuel está recibiendo los mensajes que el usuario Michelle le envía, tal como se muestra en la Fig. 43.

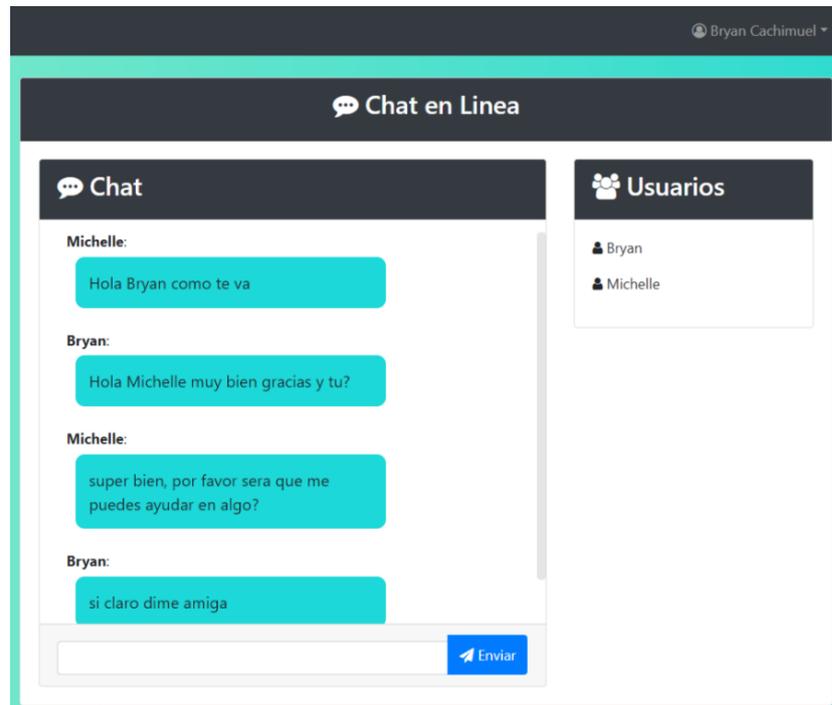


Fig. 43 Chat visto desde el usuario Bryan Cachimuel

Ahora se conecta un nuevo usuario el cual podrá unirse al chat que se estaba realizado., tal como se verifica en la Fig. 44 donde se presenta al usuario Nelson como nuevo miembro del chat.

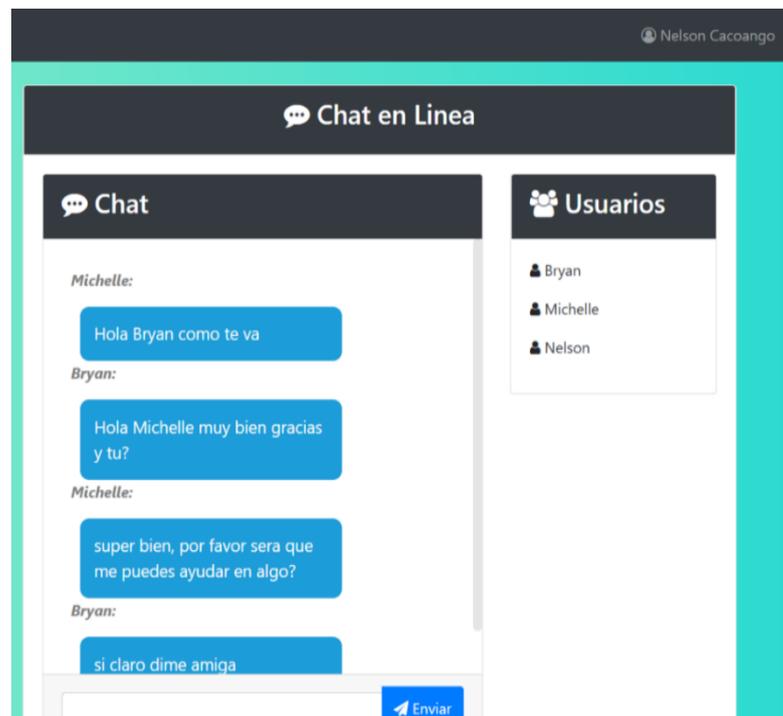


Fig. 44 Ingreso del usuario Nelson Cacoango al chat

Con el nuevo usuario conectado se procederá a realizar un chat privado entre dos personas en este caso se enviará un mensaje desde el usuario Nelson, tal como se ve en la Fig. 45.

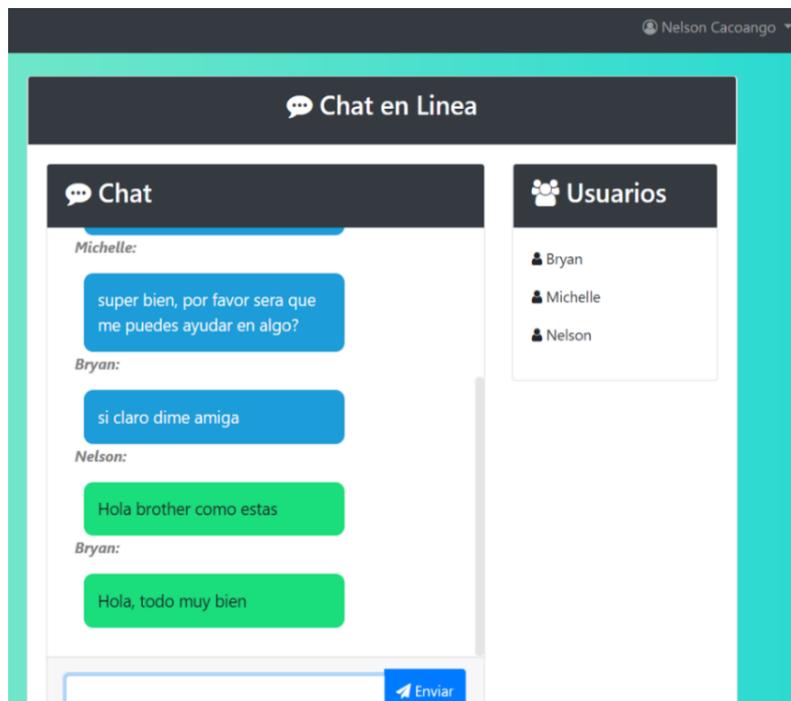


Fig. 45 Chat privado visto desde el usuario Nelson Cacoango

El usuario Bryan recibirá el mensaje de Nelson, tal como se ve en la Fig. 46.

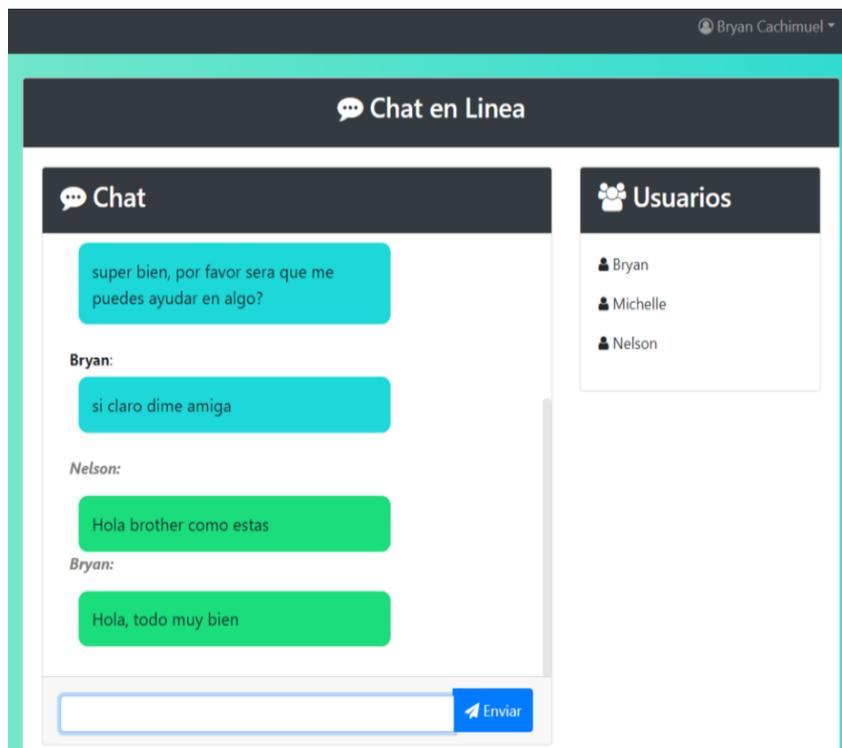


Fig. 46 Chat visto desde el usuario Bryan Cachimuel

Mientras que el usuario Michelle no vera ningún mensaje ya que ella no forma parte del chat privado realizado entre los usuarios Bryan y Nelson, tal como se ve en la Fig. 47.

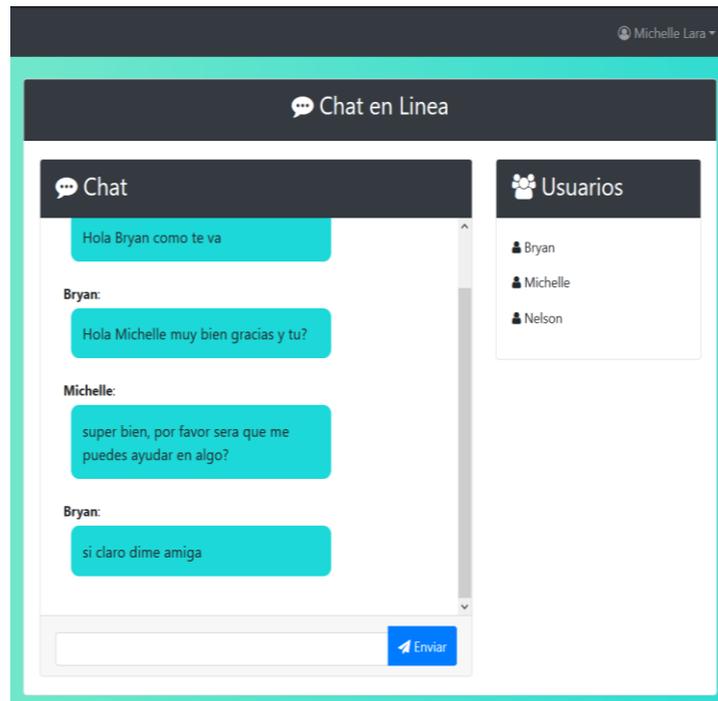


Fig. 47 Vista del chat de Michelle Lara

Al terminar las tareas emprendidas dentro de cada sprint es necesario mejorar todas las vistas del prototipo de red social con el fin de optimizar la presentación de la misma, este proceso se lo llevo a cabo mediante el uso de Bootstrap 4 y CSS, tal como se muestra en la Fig. 48.

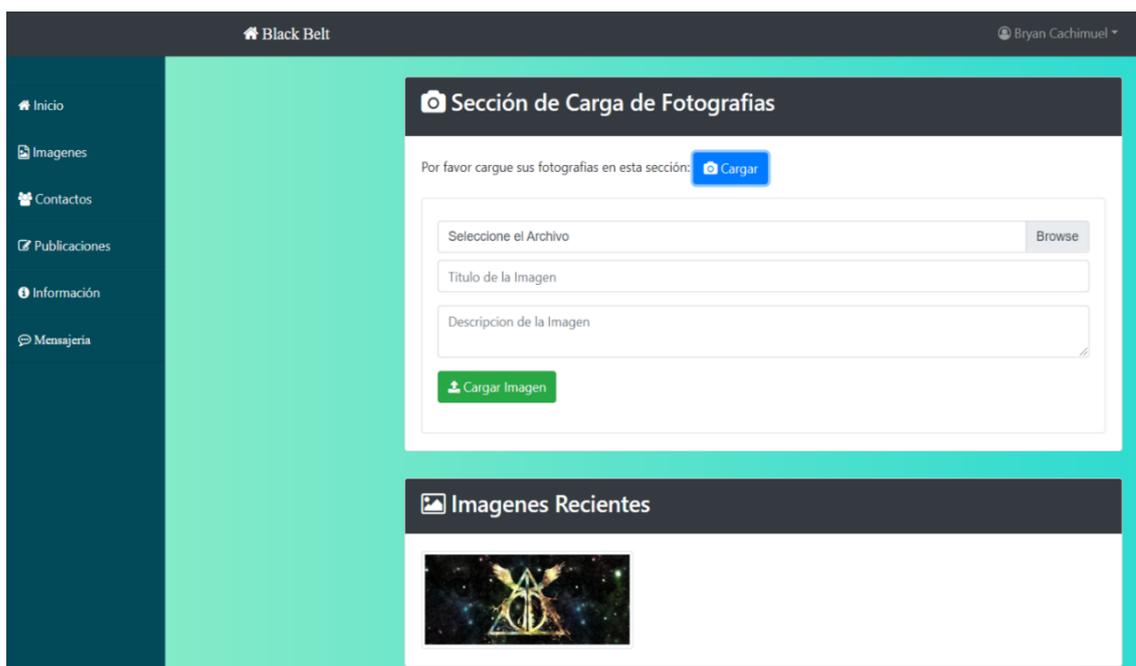


Fig. 48 Mejora de la vista principal al momento de Logearse

Mejora de la vista para cargar imágenes de los usuarios, tal como se ve en la Fig. 49.

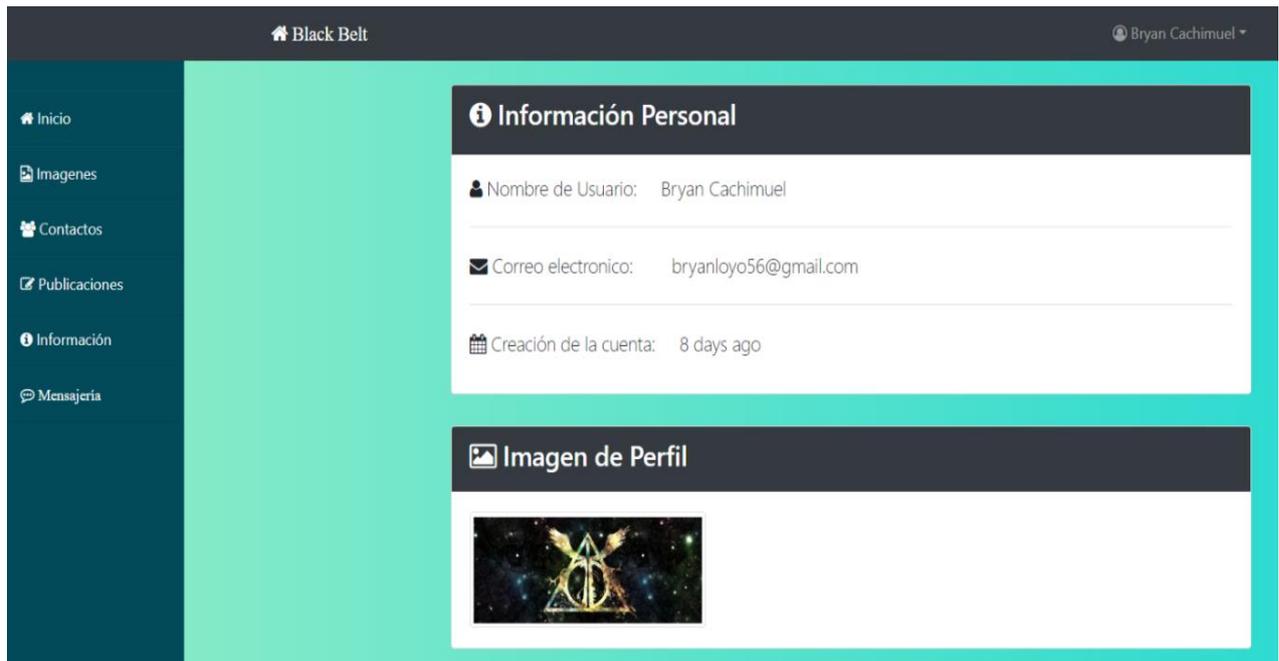


Fig. 49 Mejora de la vista de la carga de imágenes

Mejora de la vista de publicaciones de los usuarios, tal como se ve en la Fig. 50.

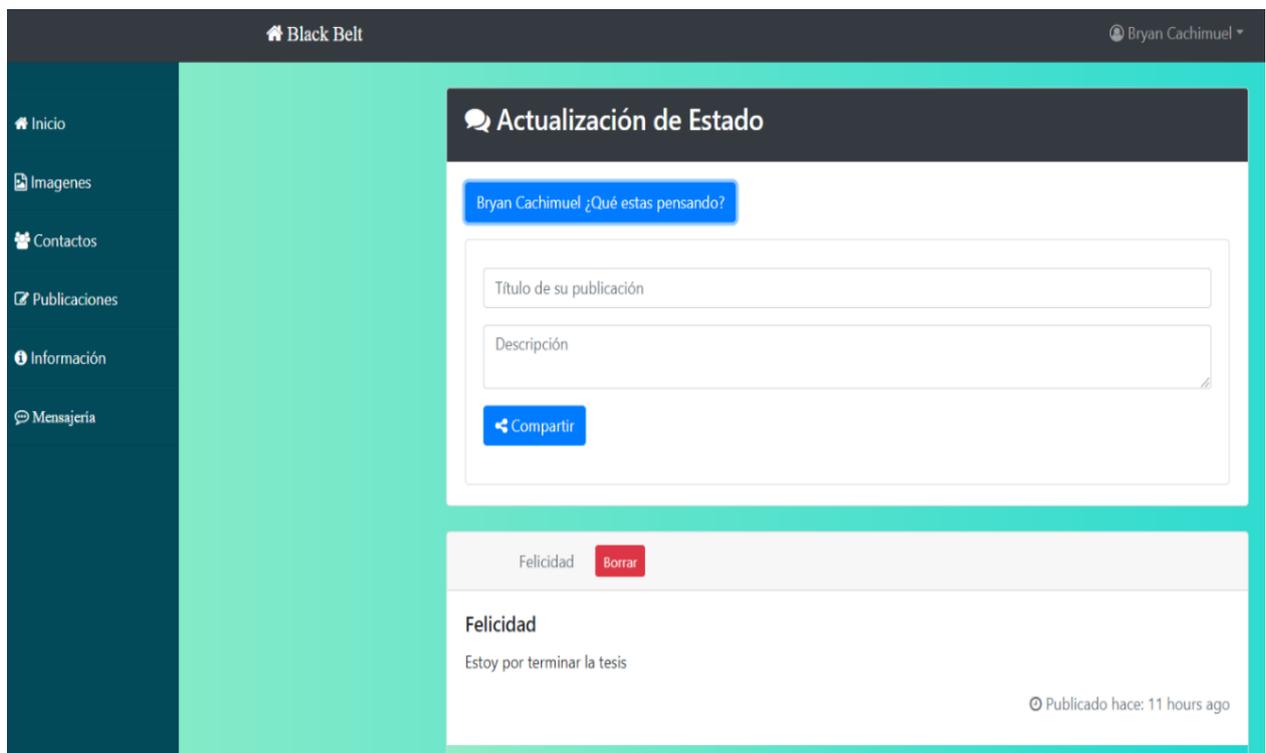
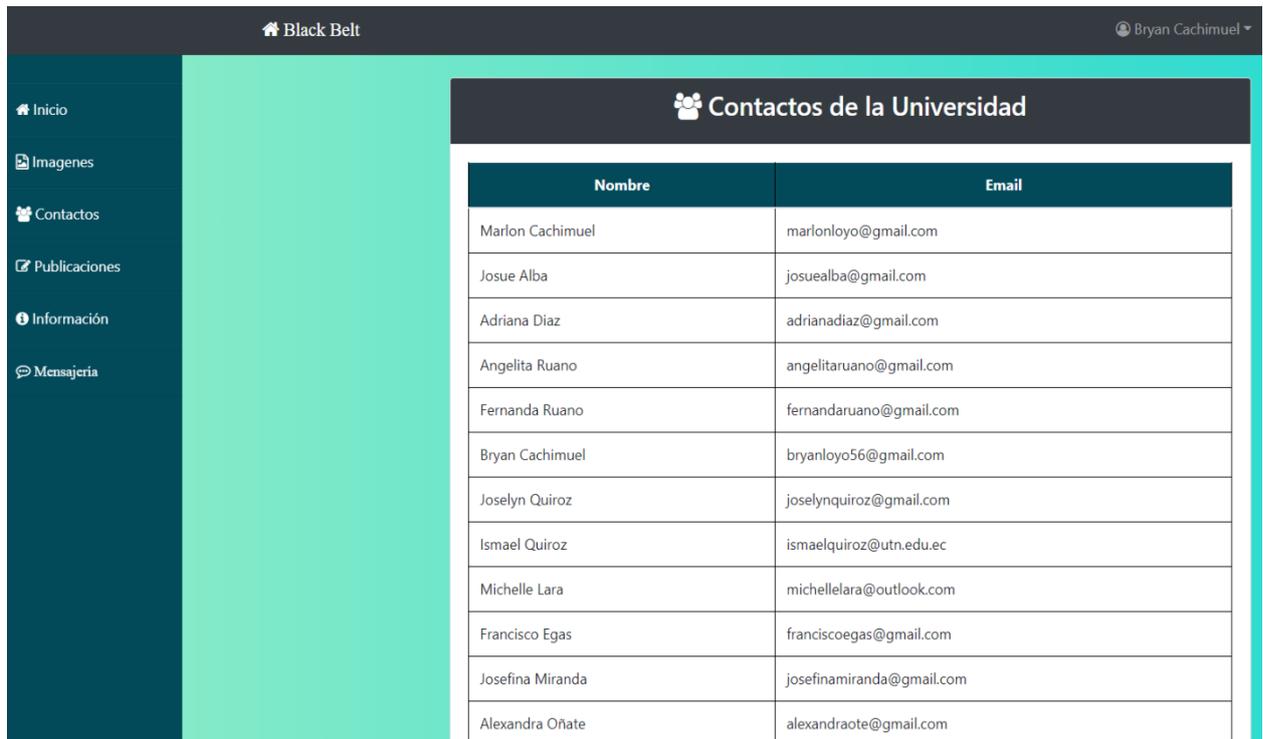


Fig. 50 Mejora de la vista de publicaciones

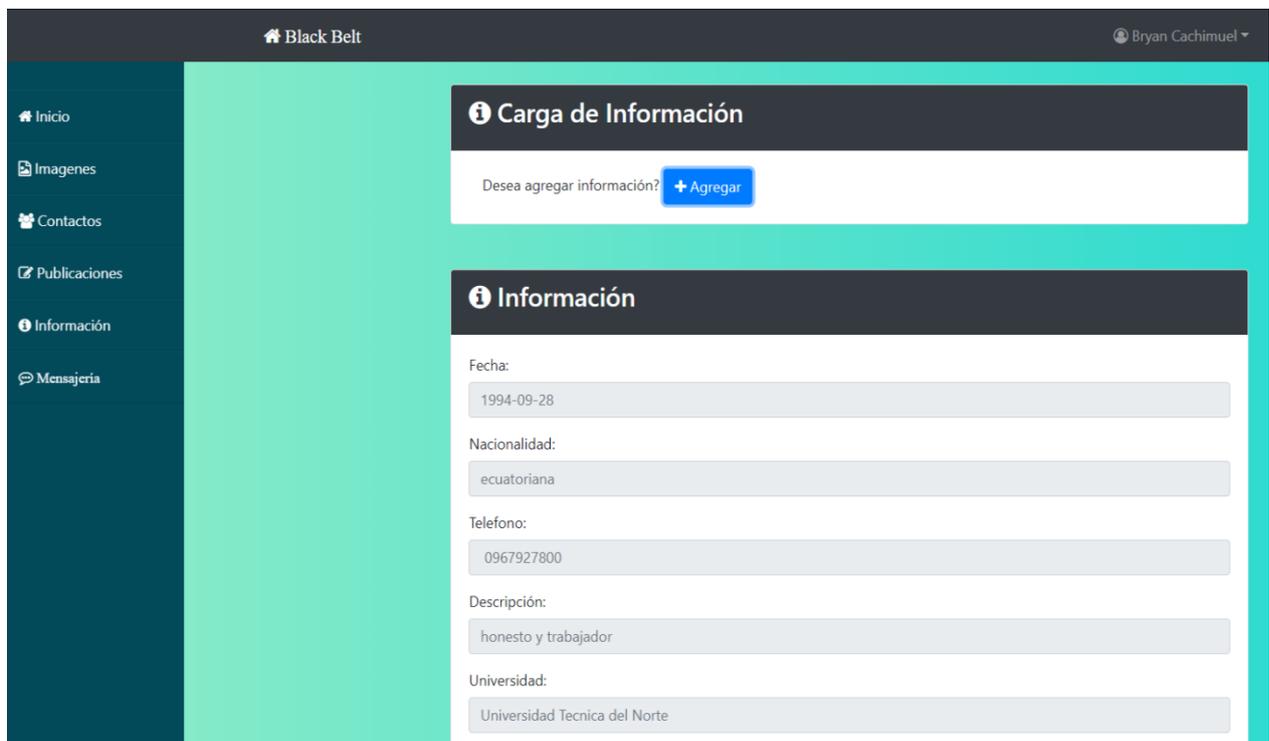
Mejora de la vista de contactos para que los usuarios puedan ver a todos los usuarios que están registrados en el prototipo de red social, tal como se ve la Fig. 51.



Nombre	Email
Marlon Cachimuel	marlonloyo@gmail.com
Josue Alba	josuealba@gmail.com
Adriana Diaz	adrianadiaz@gmail.com
Angelita Ruano	angelitaruano@gmail.com
Fernanda Ruano	fernandaruano@gmail.com
Bryan Cachimuel	bryanloyo56@gmail.com
Joselyn Quiroz	joselynquiroz@gmail.com
Ismael Quiroz	ismaelquiroz@utn.edu.ec
Michelle Lara	michellelara@outlook.com
Francisco Egas	franciscoegas@gmail.com
Josefina Miranda	josefinamiranda@gmail.com
Alexandra Oñate	alexandraote@gmail.com

Fig. 51 Mejora de la vista de contactos

Mejora de la vista para ingreso y verificación de información de los usuarios, tal como se ve en la Fig. 52.



Carga de Información

Desea agregar información? [+ Agregar](#)

Información

Fecha:
1994-09-28

Nacionalidad:
ecuatoriana

Teléfono:
0967927800

Descripción:
honesto y trabajador

Universidad:
Universidad Tecnica del Norte

Fig. 52 Mejora de la vista de ingreso de información

2.4 Pruebas

Una vez culminado el desarrollo se han realizado las debidas pruebas de funcionalidad del prototipo de red social y se ha encontrado con los siguientes resultados.

TABLA 41
PRUEBA 1

Procesos	Actividad	Porcentaje de aceptación
Modelado de Bases de Datos	Modelado de la Base de datos en Apache Cassandra en el software DbSchema.	100%
	Modelado de la Base de datos en MongoDB en el software DbSchema.	100%
	Modelado de la Base de datos en Redis en el software DbSchema.	100%
Pruebas de compatibilidad	Realizar pruebas de compatibilidad entre las versiones del software a utilizarse.	100%

Fuente: Propia

TABLA 42
PRUEBA 2

Procesos	Actividad	Porcentaje de aceptación
Utilización de dependencias	Descarga y utilización de dependencias para el cumplimiento de parámetros en cada clase creada en el prototipo de red social.	100%
Configuraciones	Configuración del servidor en node.js.	100%
Conexión	Conexión desde node.js a la base de datos Apache Cassandra.	100%
	Conexión desde node.js a la base de datos MongoDB.	100%
	Conexión desde node.js a la base de datos Redis.	100%
Rutas	Creación de las rutas para el index.	100%
	Creación de las rutas para el usuario.	100%
	Creación de las rutas para las imágenes.	100%
Modelo en node.js	Creación de un modelo donde se especifique los atributos necesarios para migrar hacia la base de datos.	100%
Creación de Métodos	Creación de un método especializado para el almacenamiento de imágenes dentro de la base de datos	100%
Etiquetación y comentarios	Creación de etiquetas e iconos para verificar la cantidad de me gustas, número de vistas y tiempo de carga de la imagen hacia la base de datos.	100%
	Creación de un panel para el ingreso de comentarios de los usuarios.	100%

Fuente: Propia

TABLA 43
PRUEBA 3

Procesos	Actividad	Porcentaje de aceptación
Modelo en node.js	Creación de un modelo donde se especifique los atributos necesarios para migrar hacia la base de datos.	100%
Creación de vistas	Creación de vistas adecuadas para redirigir desde la vista del index, hacia las vistas de registro y login.	100%
Creación de método	Creación de un método de autenticación para el control seguridad en el login.	100%
Uso de dependencias	Utilización de la dependencia bcrypt para el cifrado de contraseñas en el registro de usuarios.	100%
	Utilización de la dependencia md5 para el cifrado de contraseñas en el registro de usuarios para la base de datos de Apache Cassandra.	100%
	Utilización de dependencias para el despliegue de alertas por pantalla en el caso de presentarse errores en registro y login.	100%
Control de contraseñas	Mediante sentencias se definió la cantidad de caracteres necesarios para el registro de usuarios.	100%
	Mediante sentencias hacer una comparativa entre los campos de ingreso y reingreso de contraseña.	100%
	Mediante sentencias se controla que la contraseña en login sea igual a la contraseña almacenada en la base de datos.	100%
Control de rutas	Verificación de las rutas creadas anteriormente cuenten con la autenticación de usuarios para que exista un control de seguridad apropiado.	100%
Creación de rutas	Creación de rutas necesarias para mostrar por pantalla una vista principal del muro del usuario logeado.	100%
Creación de estados	Creación de un panel donde se especifique que los usuarios puedan escribir sus estados o publicar algún contenido que pueda escribir el usuario.	100%

Fuente: Propia

TABLA 44
PRUEBA 4

Procesos	Actividad	Porcentaje de aceptación
Creación de métodos	Crear un método para Apache Cassandra, MongoDB y Redis que permita realizar una consulta hacia la base de datos para poder obtener la información de todos los usuarios registrados en el prototipo de red social.	100%
Despliegue de datos	Crear una tabla donde se vaya almacenado la información obtenida de los usuarios.	100%
Modelo en node.js y métodos	Crear un modelo de datos con los atributos correspondientes para poder utilizarlos en los métodos que permitirán ingresar la información.	100%
Ingreso de Información	Realizar el diseño de una vista en donde se pida ingresar la información para los usuarios.	100%
Validación en los campos de texto	Mediante jquery realizar las validaciones de los campos de texto tanto en el registro de usuarios, ingreso de información de los usuarios, edición de información de los usuarios, ingreso de información para carga de imágenes y de las publicaciones que realicen los usuarios.	100%
Validación en los campos de texto	Realizar las respectivas validaciones para los campos de texto de la fecha de nacimiento, selección de carrera y selección de género de los usuarios.	100%
Despliegue de datos personales	Mediante una consulta hacia las bases de datos se obtiene información ingresada por el usuario al momento del registro y se colocó la misma en un recuadro en la vista principal para dar a conocer los datos principales del usuario.	100%
Despliegue de imágenes cargadas	Mediante una consulta hacia las bases de datos se obtiene las imágenes cargadas por los usuarios y se las coloca en un recuadro en la vista principal para dar a conocer las imágenes cargadas por los usuarios.	100%
Modelo en node.js y métodos	Crear un modelo de datos con los atributos correspondientes para poder utilizarlos en los métodos que permitirán ingresar la información para poder realizar una publicación.	100%
Ingreso de Información	Realizar el diseño de una vista en donde se pida ingresar la información para realizar una publicación.	100%
Despliegue de datos	Mediante una consulta hacia la base de datos se obtuvo la información necesaria y se la plasmó en una vista donde se pedía la información consultada.	100%

Fuente: Propia

TABLA 45
PRUEBA 5

Procesos	Actividad	Porcentaje de aceptación
Modelo en node.js y métodos	Creación de un modelo para mensajes especificando los atributos necesarios para migrar hacia la base de datos y debidamente crear los métodos para crear y listar los mensajes	100%
Control de envío y recepción	Configurar los archivos necesarios tanto para la parte del cliente como del servidor	100%
	Mediante la utilización de la dependencia socket.io se verifico el envío y recepción de mensajes en tiempo real.	100%
Creación de vistas	Crear una vista en donde se cree nombres específicos para los campos de escritura de chat, panel del chat para poder manejarlos con la herramienta jquery que ayudara con la configuración tanto para el servidor y cliente.	100%
Actualización de vistas	Revisión de los cambios a efectuados en cada vista utilizando Bootstrap.	100%
	Revisión de los cambios a efectuados en cada vista utilizando CSS.	100%

Fuente: Propia

Capítulo III

Validación de Resultados

El sistema web será validado mediante la aplicación de la norma ISO/IEC 25012 de la cual sea elegido como principal característica a evaluar lo que es la eficiencia, ya que es la que se acopla adecuadamente para realizar la validación de los resultados que se espera obtener de los prototipos de red social desarrollados juntamente con las bases de datos NoSQL como son Apache Cassandra, MongoDB y Redis.

Para cumplir con la validación de los resultados se ha optado por utilizar el software Jmeter, ya que cuenta con las herramientas necesarias para poder realizar pruebas de estrés y mediante las mismas verificar que los sistemas web desarrollados cuentan con un alto performance y eficiencia al momento de efectuar un número alto de peticiones y respuestas hacia el servidor. (Apache JMeter, 2019). De esta manera al momento de implementar las pruebas de estrés se efectúa una simulación de un ambiente de producción con un alto índice de prestaciones por parte de los usuarios hacia los sistemas web.

De acuerdo con lo mencionado anteriormente se ha descrito a continuación los procesos necesarios para obtener los resultados y con ello realizar los análisis respectivos para poder llegar a una conclusión con que se determinara el resultado esperado para el estudio en proceso.

3.1 Pruebas de Funcionamiento

Las pruebas de funcionamiento se las ha conseguido mediante el uso de la herramienta Jmeter ya que con ella se ha podido obtener el rendimiento (velocidad de respuesta) de los registros de usuarios y de los mensajes enviados entre usuarios en los prototipos de red social, es de esta manera que durante el proceso de pruebas se ha logrado conseguir diferentes tipos de resultados para las variables que Jmeter evalúa tales como: la media, mediana, mínimo, máximo, desviación estándar, porcentaje de error, rendimiento, paquetes recibidos, paquetes enviados y el promedio total en bytes de todas las muestras (avg. bytes). Es decir que todas las variables antes mencionadas serán las encargadas de dar a conocer los valores que se requieren para analizar y concluir cuál de las tres bases de datos NoSQL es mejor dentro del manejo de grandes cantidades de información.

A continuación, en las tablas siguientes se describen los valores consignados por Jmeter durante la fase de pruebas, donde se puede apreciar las diferencias que existen entre las tres bases de datos NoSQL: MongoDB, Apache Cassandra y Redis.

Las pruebas de estrés realizadas hacia el sistema web conectado a la base de datos MongoDB se describen en la tabla 46, donde se puede identificar que cada una de las variables oscilan a tener diferentes valores ya que van de la mano con el aumento progresivo que va teniendo la variable independiente muestras, ya que de acuerdo con ella las demás variables dependientes van adquiriendo un valor calculado del análisis que realiza Jmeter al momento de ingresar los datos correspondientes.

TABLA 46
VALORES DE MONGODB EN REGISTROS DE USUARIOS

Base de Datos MongoDB									
Muestras	Media	Mínimo	Máximo	Desviación Estándar	Error %	Rendimiento (bit/sec)	Recibidos (kb/sec)	Enviados (kb/sec)	Avg Bytes
100	12321	12093	12702	142,38	0	7,6	56,91	9,16	7663,6
200	23653	21753	23933	181,25	0	8,1	60,67	9,77	7663,6
300	34715	32646	35084	313,14	0	8,4	62,52	10,07	7663,6
400	30958	4002	41461	15541,23	29,5	9,4	58,18	9,05	7663,6
500	27195	4000	41829	17410,54	38,6	11,8	67,09	9,72	7663,6
600	23461	4000	43474	18963,45	51,8	13,5	69,24	9,28	7663,6
700	20914	4000	39675	17539,01	51,9	17,4	86,93	10,12	7663,6
800	18450	4000	42103	17940,04	62,5	18,7	84,79	9,53	7663,6
900	16791	4000	41324	17450,09	66	21,4	92,9	9,67	7663,6
1000	16879	4000	44556	18391,56	68,2	22,2	94,29	9,31	7663,6
1100	14634	4000	42162	16780,84	72,2	25,7	104,07	9,52	7663,6
1200	13451	4000	41501	15826,45	75	28,5	111,81	9,56	7663,6
1300	13172	4000	41193	15735,31	77,9	31,2	121,84	9,93	7663,6
1400	11929	4000	43051	15089,78	79,7	32	117,53	9,14	7663,6
1500	11742	4000	41769	14858,39	80,5	35,4	130,4	9,61	7663,6
Total	290265	114494	575817	202163,5	754	291	1319,2	143,4	114954

Fuente: Propia

De igual manera el sistema web conectado hacia la base de datos Apache Cassandra fue sometido a pruebas de estrés donde se pudo conseguir la respectiva información que es explicada en la tabla 47, en donde se define que cada una de las variables poseen diferentes valores debido al aumento progresivo que va teniendo la variable independiente muestras, ya que la misma tiene una relación con las demás variables dependientes.

TABLA 47
VALORES DE APACHE CASSANDRA EN REGISTROS DE USUARIOS

Base de Datos Apache Cassandra									
Muestras	Media	Mínimo	Máximo	Desviación Estándar	Error %	Rendimiento (bit/sec)	Recibidos (kb/sec)	Enviados (kb/sec)	Avg Bytes
100	12563	12171	12886	193,22	0	7,4	53,63	8,98	7383,2
200	22734	15463	23087	528,67	0	8,4	60,79	10,18	7383,2
300	29329	5563	34078	10333,05	16	8,6	56,59	9,63	7383,2
400	24276	4000	34929	14333,57	34,5	11,2	63,78	9,73	7383,2
500	27629	4000	39181	16141,45	32	12,5	72,25	10,35	7383,2
600	22369	4000	40187	17634,58	49,17	14,7	73,87	9,83	7383,2
700	19847	4000	41062	17810,32	57,43	16,9	78,81	9,71	7383,2
800	18268	4000	41519	17565,07	62,25	19,1	84,96	9,59	7383,2
900	14429	4000	38036	15353,53	69,56	23,4	95,66	9,59	7383,2
1000	20708	4000	56983	24049,04	68,4	17,3	71,76	7,25	7383,2
1100	16398	4000	48505	19423,21	72,36	22,5	89,4	8,37	7383,2

1200	14263	4000	43008	16816,07	74	27,5	107,24	9,55	7383,2
1300	11577	4000	36325	13541,21	76,15	35,3	132,62	10,16	7383,2
1400	12284	4000	39575	14810,44	76,36	35	131,3	10,11	7383,2
1500	10097	4000	37854	12782,21	82	39,1	137,84	9,49	7383,2
Total	276771	81197	567215	211315,64	770,18	298,9	1310,5	142,52	110748

Fuente: Propia

En cuanto al sistema web conectado a la base de datos Redis se le realizó las pruebas de estrés en donde se obtuvo los valores correspondientes para cada variable y los mismos se muestra en la tabla 48, donde se puede observar que cada una de las variables llegan a tener diferentes valores debido a variable independiente muestras que va aumentando progresivamente para verificar el performance del sistema web.

TABLA 48
VALORES DE REDIS EN REGISTROS DE USUARIOS

Base de Datos Redis									
Muestras	Media	Mínimo	Máximo	Desviación Estándar	Error %	Rendimiento (bit/sec)	Recibidos (kb/sec)	Enviados (kb/sec)	Avg Bytes
100	12600	12271	12817	135,33	0	7,5	55,78	9,01	7615,2
200	23010	22431	23246	126,41	0	8,3	61,72	9,97	7615,2
300	33920	23421	34237	661,15	0	8,5	63,53	10,23	7615,2
400	31433	4001	40224	14402,19	24,75	9,7	61,97	9,59	7615,2
500	22371	4000	37334	16058,67	45,4	13,2	70,39	9,54	7615,2
600	19981	4000	38481	16723,89	53,83	15,3	75,73	9,48	7615,2
700	20296	4000	41622	18173,27	57	16,6	79	9,53	7615,2
800	18914	4000	42995	18337,51	61,75	18,4	83,64	9,42	7615,2
900	16933	4000	41765	17592,63	65,89	21,2	91,66	9,55	7615,2
1000	15607	4000	43016	17876,31	69,4	23	94,51	9,4	7615,2
1100	14879	4000	43265	17114,92	72,36	25,2	101,76	9,33	7615,2
1200	12038	4000	39054	14340,82	77,33	30,3	115	9,31	7615,2
1300	12243	4000	37855	14336,71	75,15	34	131,08	10,14	7615,2
1400	10854	4000	38296	13403,33	80,29	36,1	131,33	9,52	7615,2
1500	10563	4000	39162	13410,36	81,27	37,8	136,11	9,46	7615,2
Total	275642	106124	553369	192693,5	764,42	305,1	1353,21	143,48	114228

Fuente: Propia

Las pruebas de estrés fueron realizadas también a lo que es la mensajería donde se obtuvieron los siguientes resultados del sistema web conectado hacia la base de datos MongoDB los cuales se describen en la tabla 49, en la cual se puede identificar que cada una de las variables tienen distintos valores ya que la variable independiente muestras llega a tener un aumento de 100 por cada prueba que se le realiza realizando.

TABLA 49
VALORES DE MONGODB EN MENSAJERÍA

Base de Datos MongoDB									
Muestras	Media	Mínimo	Máximo	Desviación Estándar	Error %	Rendimiento (bit/sec)	Recibidos (kb/sec)	Enviados (kb/sec)	Avg Bytes
100	3	1	34	4,42	0	100,4	28,73	50,69	293
200	5	1	134	12,65	0	186,2	53,28	94,75	293
300	219	9	473	71,02	0	239,2	68,45	121,72	293
400	62	1	325	69,06	29,21	155,8	44,59	79,28	293
500	230	18	584	134,67	36,1	163	46,65	82,95	293
600	594	38	1366	245,64	51,75	213,3	61,03	108,52	293
700	254	1	837	212,89	51,24	277,8	79,48	141,33	293

800	529	1	1158	280,24	58,9	249,8	71,49	127,12	293
900	450	1	1009	363,49	69,1	315,2	90,2	160,39	293
1000	164	2	983	200,06	68,05	336,7	96,34	171,31	293
1100	159	1	988	183,67	71,85	292,5	83,69	148,81	293
1200	142	2	684	118,77	78,14	414,8	118,69	211,04	293
1300	435	2	976	252,91	78,25	326,1	93,32	165,94	293
1400	203	2	681	179,08	80	406	116,18	206,58	293
1500	269	2	1753	316,19	80,42	382,1	109,32	194,39	293
Total	3718	82	11985	2644,76	753,01	4058,9	1161,44	2064,82	4395

Fuente: Propia

Como siguiente proceso se ha realizado las pruebas de estrés al sistema web conectado hacia la base de datos Apache Cassandra en la cual se ha conseguido los siguientes resultados descritos en la tabla 50, donde se puede identificar los valores obtenidos para cada una de las variables independientes, los cuales tienden a aumentar debido al incremento gradual de las muestras ingresadas durante la fase de pruebas de Jmeter.

TABLA 50
VALORES DE APACHE CASSANDRA EN MENSAJERÍA
Base de Datos Apache Cassandra

Muestras	Media	Mínimo	Máximo	Desviación Estándar	Error %	Rendimiento (bit/sec)	Recibidos (kb/sec)	Enviados (kb/sec)	Avg Bytes
100	1	1	4	0,64	0	101,2	28,96	51,19	293
200	1	1	6	0,79	0	197,8	56,6	102	293
300	4	1	71	7,73	15,8	314,5	89,98	159,52	293
400	90	1	409	66,07	34,1	304,4	87,1	156,96	293
500	100	2	369	56,21	32,11	379,7	108,53	195,76	293
600	142	2	650	97,28	48,93	367	105	189,22	293
700	130	11	377	100,66	57,25	413,7	118,38	213,22	293
800	109	3	395	78,39	62	428,5	122,61	220,94	293
900	265	8	665	133,26	69,35	404,5	115,74	208,57	293
1000	100	2	658	88,57	68,07	394,8	112,96	203,56	293
1100	142	1	758	134,74	72,05	417,9	119,58	215,5	293
1200	221	1	904	209,9	73,55	398,7	114,07	205,56	293
1300	301	3	824	163,19	76,02	431,2	123,37	222,33	293
1400	143	1	681	139,04	76,12	418,9	119,86	216	293
1500	207	1	1012	234,02	81,7	402,1	115,07	207,36	293
Total	1956	39	7783	1510,49	767,05	5374,9	1537,81	2767,69	4395

Fuente: Propia

Para finalizar con las pruebas de estrés sea a obtenido los valores del sistema web conectado hacia la base de datos Redis los mismos que pueden ser identificados en la tabla 51, donde se puede comprobar que las variables independientes poseen valores que van incrementándose conforme se vaya ingresando un número determinado de muestras las cuales van en aumento progresivo.

TABLA 51
VALORES DE REDIS EN MENSAJERÍA

Base de Datos Redis									
Muestras	Media	Mínimo	Máximo	Desviación Estándar	Error %	Rendimiento (bit/sec)	Recibidos (kb/sec)	Enviados (kb/sec)	Avg Bytes
100	5	1	72	8,48	0	101,8	29,14	53,5	293
200	69	3	349	57,43	0	183,3	52,45	96,31	293
300	120	6	345	67,38	0	214,7	61,45	112,83	293
400	320	7	1012	174,07	24,5	193,3	55,32	101,57	293
500	319	16	1279	154,94	45,2	218,2	62,42	114,61	293
600	250	16	1259	241,54	53,15	237,8	68,05	124,94	293
700	213	35	631	159,09	56,8	266,3	76,19	139,89	293
800	633	4	1134	400,2	61,5	242,6	69,41	127,44	293
900	292	2	979	290,08	65,32	301	86,13	158,14	293
1000	349	1	874	259,79	69,15	319,5	91,42	167,86	293
1100	302	3	817	238,85	72,16	305,6	87,43	160,54	293
1200	191	5	705	178,21	77,22	384,9	110,12	202,2	293
1300	131	1	633	115,64	75,05	429,3	122,84	225,56	293
1400	114	2	526	96,24	80	419,4	120,01	220,36	293
1500	387	1	779	168,08	81,07	391,4	112	205,66	293
Total	3695	103	11394	2610,02	761,12	4209,1	1204,38	2211,41	4395

Fuente: Propia

3.2 Evaluación de la velocidad de respuesta de los datos

Con los valores obtenidos y explicados en las tablas 46, 47 ,48 ,49 ,50 y 51 se ha logrado extraer las variables independientes con sus respectivos valores cuantificables, para poder con ellos definir cuál de las tres bases de datos posee la velocidad de respuesta adecuada que permitirá concluir cuál de ellas puede manejar grandes cantidades de información de una manera eficiente de acuerdo con la característica escogida de la norma ISO/IEC 25012.

Con lo mencionado anteriormente se procede a dar inicio a la fase de evaluación de las herramientas, en donde se ha ingresado en el sistema web conectado hacia la base de datos MongoDB un usuario el cual sirve como base para poder efectuar el análisis dentro del software Jmeter, una vez analizado el usuario se procede de la siguiente manera: se da inicio con el ingreso de 100 usuarios y se va aumentando de 100 en 100 hasta llegar a 1500 registros de usuarios conforme se ha establecido el incremento gradual de las muestras, es así como en la tabla 52 se puede identificar los valores alcanzados para las variables dependientes de la muestra, cabe mencionar que dentro de esta tabla se encuentra el valor total de cada una las variables y en especial se identifica como principal variable de análisis lo que es el rendimiento en donde se obtiene como valor total 291.3 bit/sec, este valor al final servirá para comparar con los otros valores totales a conseguirse para el estudio.

TABLA 52
SELECCIÓN DE VARIABLES PARA LA EVALUACIÓN DE VELOCIDAD DE DATOS CON MONGODB

Base de Datos MongoDB					
Muestras	Desviación Estándar	Rendimiento (bit/sec)	Recibidos (kb/sec)	Enviados (kb/sec)	Error %
100	142,38	7,6	56,91	9,16	0
200	181,25	8,1	60,67	9,77	0
300	313,14	8,4	62,52	10,07	0
400	15541,23	9,4	58,18	9,05	29,5
500	17410,54	11,8	67,09	9,72	38,6
600	18963,45	13,5	69,24	9,28	51,83
700	17539,01	17,4	86,93	10,12	51,86
800	17940,04	18,7	84,79	9,53	62,5
900	17450,09	21,4	92,9	9,67	66
1000	18391,56	22,2	94,29	9,31	68,2
1100	16780,84	25,7	104,07	9,52	72,18
1200	15826,45	28,5	111,81	9,56	75
1300	15735,31	31,2	121,84	9,93	77,92
1400	15089,78	32	117,53	9,14	79,73
1500	14858,39	35,4	130,4	9,61	80,5
Total	202163,46	291,3	1319,17	143,44	753,82

Fuente: Propia

Continuando con el análisis se procedió a realizar las pruebas de estrés hacia el sistema con la base de datos Apache Cassandra en donde se adquirió diferentes valores para las variables los cuales son descritos en la tabla 53, donde se puede identificar los valores obtenidos para las variables dependientes de la muestra, es necesario mencionar que dentro de esta tabla se encuentra el valor total de cada una las variables en donde se identifica como principal variable de análisis lo que es el rendimiento el cual tiene un valor de 5374,9 bit/sec.

TABLA 53
SELECCIÓN DE VARIABLES PARA LA EVALUACIÓN DE VELOCIDAD DE DATOS CON APACHE CASSANDRA

Base de Datos Apache Cassandra					
Muestras	Desviación Estándar	Rendimiento (bit/sec)	Recibidos KB/sec	Enviados KB/sec	Error %
100	193,22	7,4	53,63	8,98	0
200	528,67	8,4	60,79	10,18	0
300	10333,05	8,6	56,59	9,63	16
400	14333,57	11,2	63,78	9,73	34,5
500	16141,45	12,5	72,25	10,35	32
600	17634,58	14,7	73,87	9,83	49,17
700	17810,32	16,9	78,81	9,71	57,43
800	17565,07	19,1	84,96	9,59	62,25
900	15353,53	23,4	95,66	9,59	69,56
1000	24049,04	17,3	71,76	7,25	68,4
1100	19423,21	22,5	89,4	8,37	72,36
1200	16816,07	27,5	107,24	9,55	74
1300	13541,21	35,3	132,62	10,16	76,15
1400	14810,44	35	131,3	10,11	76,36
1500	12782,21	39,1	137,84	9,49	82
Total	211315,64	298,9	1310,5	142,52	770,18

Fuente: Propia

Como prueba de estrés final se realizó al sistema web conectado hacia la base de datos Redis, en donde se obtuvo los respectivos valores cuantificables para cada una de las variables, dichos valores son descritos en la tabla 54, en la cual se identifica los valores obtenidos para las variables dependientes de la muestra, cabe mencionar que dentro de esta tabla se encuentra el valor total de cada una las variables, en donde se enfatiza en especial el rendimiento cuyo valor total es de 305,1 bit/sec.

TABLA 54
SELECCIÓN DE VARIABLES PARA LA EVALUACIÓN DE VELOCIDAD DE DATOS CON REDIS

Base de Datos Redis					
Muestras	Desviación Estandar	Rendimiento (bit/sec)	Recibidos KB/sec	Enviados KB/sec	Error %
100	135,33	7,5	55,78	9,01	0
200	126,41	8,3	61,72	9,97	0
300	661,15	8,5	63,53	10,23	0
400	14402,19	9,7	61,97	9,59	24,75
500	16058,67	13,2	70,39	9,54	45,4
600	16723,89	15,3	75,73	9,48	53,83
700	18173,27	16,6	79	9,53	57
800	18337,51	18,4	83,64	9,42	61,75
900	17592,63	21,2	91,66	9,55	65,89
1000	17876,31	23	94,51	9,4	69,4
1100	17114,92	25,2	101,76	9,33	72,36
1200	14340,82	30,3	115	9,31	77,33
1300	14336,71	34	131,08	10,14	75,15
1400	13403,33	36,1	131,33	9,52	80,29
1500	13410,36	37,8	136,11	9,46	81,27
Total	192693,5	305,1	1353,21	143,48	764,42

Fuente: Propia

Una vez finalizado el proceso de pruebas de stress y orden de información dentro de las respectivas tablas donde se describen los valores obtenidos para cada variable, se procedió a efectuar un análisis comparativo dentro de la tabla 55 en la que se verifica cómo los valores de rendimiento cambian de una forma mínima en cada una de las muestras ingresadas dentro del software Jmeter, es de esta manera como se obtiene un resultado final de la comparación elaborada de las tres bases de datos NoSQL y se llega a la conclusión de que la base de datos con mejor rendimiento dentro del análisis efectuado es: MongoDB ya que cuenta con un rendimiento (velocidad de respuesta) de 291,3 bit/sec que es más bajo que los resultados obtenidos para los sistemas web conectados a Apache Cassandra y Redis

TABLA 55
COMPARATIVA DEL RENDIMIENTO DE LAS TRES BASES DE DATOS NOSQL

Rendimiento de las tres Bases de Datos NoSQL (bit/sec)			
Muestras	MongoDB	Apache Cassandra	Redis
100	7,6	7,4	7,5
200	8,1	8,4	8,3
300	8,4	8,6	8,5
400	9,4	11,2	9,7
500	11,8	12,5	13,2
600	13,5	14,7	15,3
700	17,4	16,9	16,6
800	18,7	19,1	18,4
900	21,4	23,4	21,2
1000	22,2	17,3	23
1100	25,7	22,5	25,2
1200	28,5	27,5	30,3
1300	31,2	35,3	34
1400	32	35	36,1
1500	35,4	39,1	37,8
Total	291,3	298,9	305,1

Fuente: Propia

Con la información presentada en la tabla 55 se puede establecer en la Fig. 53 una comparativa gráfica, donde se da a conocer las fluctuaciones que se presentan en el rendimiento de cada base de datos NoSQL y de acuerdo con la misma se puede decir que MongoDB posee una ligera diferencia de rendimiento en comparación con Apache Cassandra y Redis.

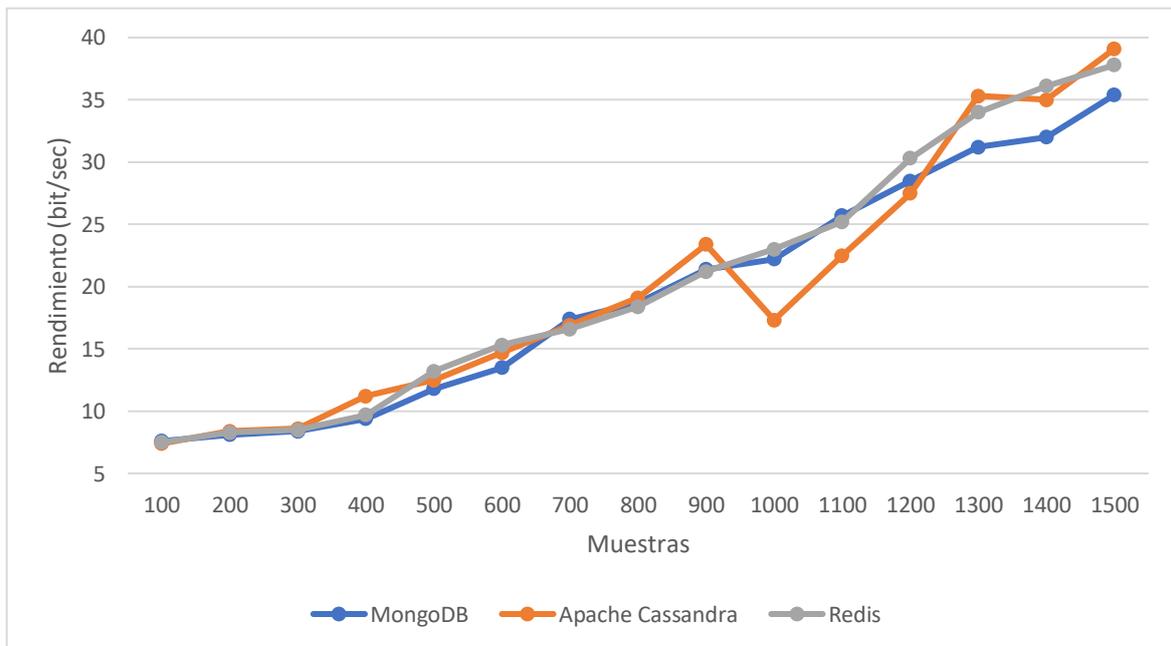


Fig. 53 Mejora de la vista de ingreso de información

Para complementar la fase de evaluación de las herramientas se realizó las mismas a lo que es la mensajería para comprobar el rendimiento obtenido al enviar mensajes de un usuario a otro, de esta manera se procedió a hacer él envió de 100 mensajes los cuales fueron aumentando progresivamente de 100 en 100 hasta llegar al envió de 1500 mensajes para la verificación de la velocidad de respuesta en cada uno de los prototipos de red social conectados hacia las bases de datos NoSQL.

Las pruebas efectuadas hacia el sistema web conectado a la base de datos MongoDB se especifican en la tabla 56, donde se identifica los valores conseguidos para las variables dependientes de la muestra, cabe indicar que dentro de esta tabla se encuentra el valor total de cada una las variables, en especial la variable de análisis rendimiento que posee un valor total de 4058,9 bit/sec.

TABLA 56
SELECCIÓN DE VARIABLES PARA LA EVALUACIÓN DE LA VELOCIDAD DE DATOS CON MONGODB

Base de Datos MongoDB					
Muestras	Desviación Estándar	Rendimiento (bit/sec)	Recibidos (kb/sec)	Enviados (kb/sec)	Error %
100	4,42	100,4	28,73	50,69	0
200	12,65	186,2	53,28	94,75	0
300	71,02	239,2	68,45	121,72	0
400	69,06	155,8	44,59	79,28	29,21
500	134,67	163	46,65	82,95	36,1
600	245,64	213,3	61,03	108,52	51,75
700	212,89	277,8	79,48	141,33	51,24
800	280,24	249,8	71,49	127,12	58,9
900	363,49	315,2	90,2	160,39	69,1
1000	200,06	336,7	96,34	171,31	68,05
1100	183,67	292,5	83,69	148,81	71,85
1200	118,77	414,8	118,69	211,04	78,14
1300	252,91	326,1	93,32	165,94	78,25
1400	179,08	406	116,18	206,58	80
1500	316,19	382,1	109,32	194,39	80,42
Total	2644,76	4058,9	1161,44	2064,82	753,01

Fuente: Propia

Como siguiente punto de análisis se procedió a cumplir las pruebas para el sistema web conectado hacia la base de datos Apache Cassandra en donde se obtuvo diferentes valores para las variables cuantitativas los cuales son descritos en la tabla 57, donde se puede identificar los valores obtenidos para las variables dependientes de la muestra, cabe mencionar que dentro de esta tabla se encuentra el valor total de cada una las variables en donde se identifica como principal variable de análisis al rendimiento el cual tiene un valor de 5374,9 bit/sec.

TABLA 57
SELECCIÓN DE VARIABLES PARA LA EVALUACIÓN DE LA VELOCIDAD DE DATOS CON APACHE CASSANDRA

Base de Datos Apache Cassandra					
Muestras	Desviación Estándar	Rendimiento (bit/sec)	Recibidos KB/sec	Enviados KB/sec	Error %
100	0,64	101,2	28,96	51,19	0
200	0,79	197,8	56,6	102	0
300	7,73	314,5	89,98	159,52	15,8
400	66,07	304,4	87,1	156,96	34,1
500	56,21	379,7	108,53	195,76	32,11
600	97,28	367	105	189,22	48,93
700	100,66	413,7	118,38	213,22	57,25
800	78,39	428,5	122,61	220,94	62
900	133,26	404,5	115,74	208,57	69,35
1000	88,57	394,8	112,96	203,56	68,07
1100	134,74	417,9	119,58	215,5	72,05
1200	209,9	398,7	114,07	205,56	73,55
1300	163,19	431,2	123,37	222,33	76,02
1400	139,04	418,9	119,86	216	76,12
1500	234,02	402,1	115,07	207,36	81,7
Total	1510,49	5374,9	1537,81	2767,69	767,05

Fuente: Propia

En cuanto al análisis realizado al sistema web conectado a la base de datos Redis se puede identificar en la tabla 58 que los valores correspondientes a cada una de las variables dependientes de la muestra tienden a cambiar en mínimas proporciones, cabe mencionar que dentro de esta tabla se encuentra el valor total de cada una las variables en donde se identifica como principal variable al rendimiento que cuenta con un valor total de 5374,9 bit/sec.

TABLA 58
SELECCIÓN DE VARIABLES PARA LA EVALUACIÓN DE LA VELOCIDAD DE DATOS CON REDIS

Base de Datos Redis					
Muestras	Desviación Estándar	Rendimiento (bit/sec)	Recibidos KB/sec	Enviados KB/sec	Error %
100	8,48	101,8	29,14	53,5	0
200	57,43	183,3	52,45	96,31	0
300	67,38	214,7	61,45	112,83	0
400	174,07	193,3	55,32	101,57	24,5
500	154,94	218,2	62,42	114,61	45,2
600	241,54	237,8	68,05	124,94	53,15
700	159,09	266,3	76,19	139,89	56,8
800	400,2	242,6	69,41	127,44	61,5
900	290,08	301	86,13	158,14	65,32
1000	259,79	319,5	91,42	167,86	69,15
1100	238,85	305,6	87,43	160,54	72,16
1200	178,21	384,9	110,12	202,2	77,22
1300	115,64	429,3	122,84	225,56	75,05
1400	96,24	419,4	120,01	220,36	80
1500	168,08	391,4	112	205,66	81,07
Total	2610,02	4209,1	1204,38	2211,41	761,12

Fuente: Propia

Al finalizar el proceso de pruebas para la sección de mensajería se ha podido realizar un ordenamiento de información el mismo que se da a conocer en la tabla 59, en donde se puede observar cómo los valores de rendimiento cambian de una forma mínima en cada una de las muestras ingresadas dentro del software Jmeter, es de esta manera como se ha podido conseguir un resultado final de la comparación realizada de las tres bases de datos NoSQL y por ende se ha llegado a la conclusión de que la base de datos con mejor rendimiento dentro del análisis realizado es: MongoDB ya que la misma cuenta con un rendimiento (velocidad de respuesta) de 4058,9 bit/sec que en comparación con las otras bases de datos es más bajo que los resultados obtenidos para las bases de datos Apache Cassandra y Redis.

TABLA 59
COMPARATIVA DEL RENDIMIENTO DE LAS TRES BASES DE DATOS NOSQL, PARA LA MENSAJERÍA

Rendimiento de las tres Bases de Datos NoSQL (bit/sec)			
Muestras	MongoDB	Apache Cassandra	Redis
100	100,4	101,2	101,8
200	186,2	197,8	183,3
300	239,2	314,5	214,7
400	155,8	304,4	193,3
500	163	379,7	218,2
600	213,3	367	237,8
700	277,8	413,7	266,3
800	249,8	428,5	242,6
900	315,2	404,5	301
1000	336,7	394,8	319,5
1100	292,5	417,9	305,6
1200	414,8	398,7	384,9
1300	326,1	431,2	429,3
1400	406	418,9	419,4
1500	382,1	402,1	391,4
Total	4058,9	5374,9	4209,1

Fuente: Propia

Con la información descrita en la tabla 59 se puede manifestar que en la Fig. 54 existe una comparación gráfica que demuestra las fluctuaciones que se presentan en el rendimiento de cada base de datos NoSQL y de acuerdo con la misma se puede decir que MongoDB cuenta con una ligera diferencia de rendimiento en comparación con Apache Cassandra y Redis.

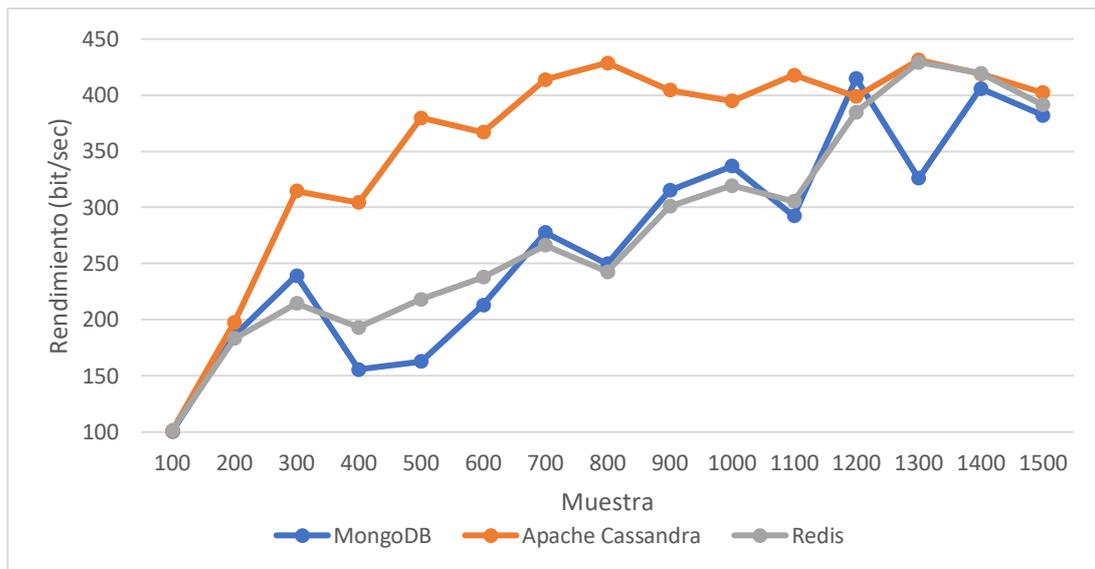


Fig. 54 Muestra de MongoDB con mejor Rendimiento en mensajería

3.3 Análisis e Interpretación de resultados

Considerando los valores descritos en las tablas 46, 47 ,48 ,49 ,50 y 51 se ha procedido a seleccionar las variables dependientes a analizar para poder comprobar la característica seleccionada de la norma ISO/IEC 25012 que se refiere a la eficiencia, de esta manera se puede comprobar que el rendimiento no se ve alterado drásticamente cuando se produce un incremento gradual de acuerdo al número de peticiones que se realice al servidor por parte del registro de usuarios y envió de mensajes entre los mismos usuarios registrados dentro de los prototipos de red social.

Analizando detenidamente y considerando los factores necesarios para realizar la correcta interpretación de los resultados se ha establecido en la tabla 60 tomar en cuenta las variables muestra, media, desviación estándar, rendimiento y error porcentual ya que las mismas identifican los valores adecuados para verificar la eficiencia que presenta el sistema web conectado hacia la base de datos MongoDB.

TABLA 60
SELECCIÓN DE VARIABLES PARA EL ANÁLISIS DE LOS DATOS EN MONGODB

Base de Datos MongoDB				
Muestras	Media	Desviación Estándar	Rendimiento (bit/sec)	Error %
100	12321	142,38	7,6	0
200	23653	181,25	8,1	0
300	34715	313,14	8,4	0
400	30958	15541,23	9,4	17,5
500	27195	17410,54	11,8	38,6
600	23461	18963,45	13,5	51,5
700	20914	17539,01	17,4	56,29
800	18450	17940,04	18,7	62,5

900	16791	17450,09	21,4	66
1000	16879	18391,56	22,2	68,2
1100	14634	16780,84	25,7	71,45
1200	13451	15826,45	28,5	75
1300	13172	15735,31	31,2	77,92
1400	11929	15089,78	32	79,73
1500	11742	14858,39	35,4	80,5

Fuente: Propia

De acuerdo con la tabla 60 se obtiene la Fig. 55, en la cual se establece una comparativa entre las variables, en donde se confirma que la media da a conocer un tiempo promedio en milisegundos los mismos que van de acuerdo con el incremento que se estableció en la muestra, como también se comprueba como la desviación estándar indica la cantidad de variación existente entre los tiempos de respuesta que se vayan obteniendo conforme la muestra aumente.

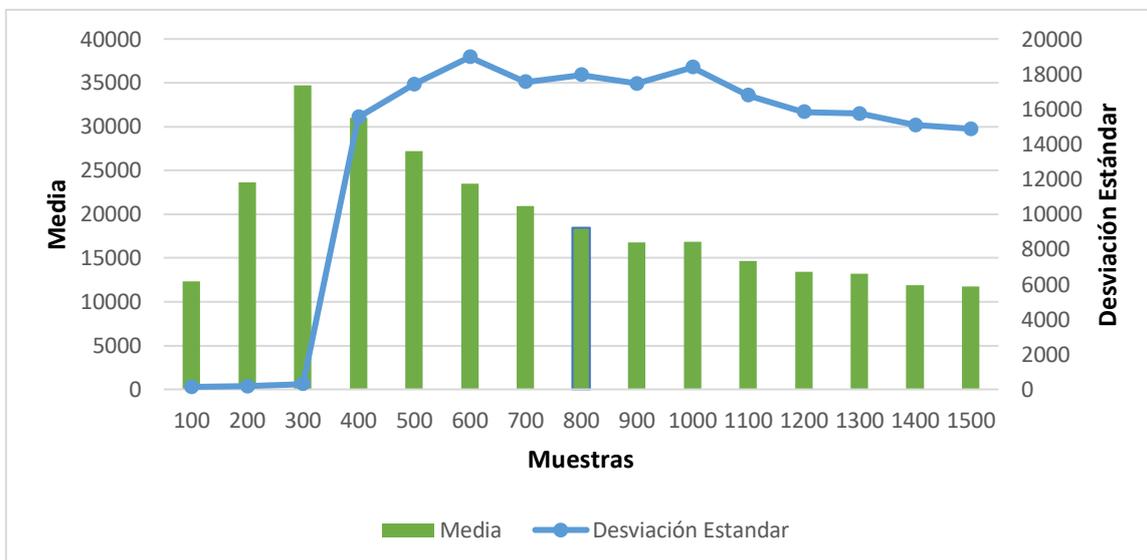


Fig. 55 Comparativa entre las variables Muestras, Media y Desviación Estándar en MongoDB

Con respecto a la tabla 60 es factible establecer una comparativa que se demuestra en la Fig. 56, en donde se puede apreciar el porcentaje de error que se puede encontrar conforme se van efectuando los ingresos de las muestras, como también se identifica el rendimiento que consiste en representar el número de transacciones por minuto que es capaz de soportar el sistema web conforme al incremento de las muestras.

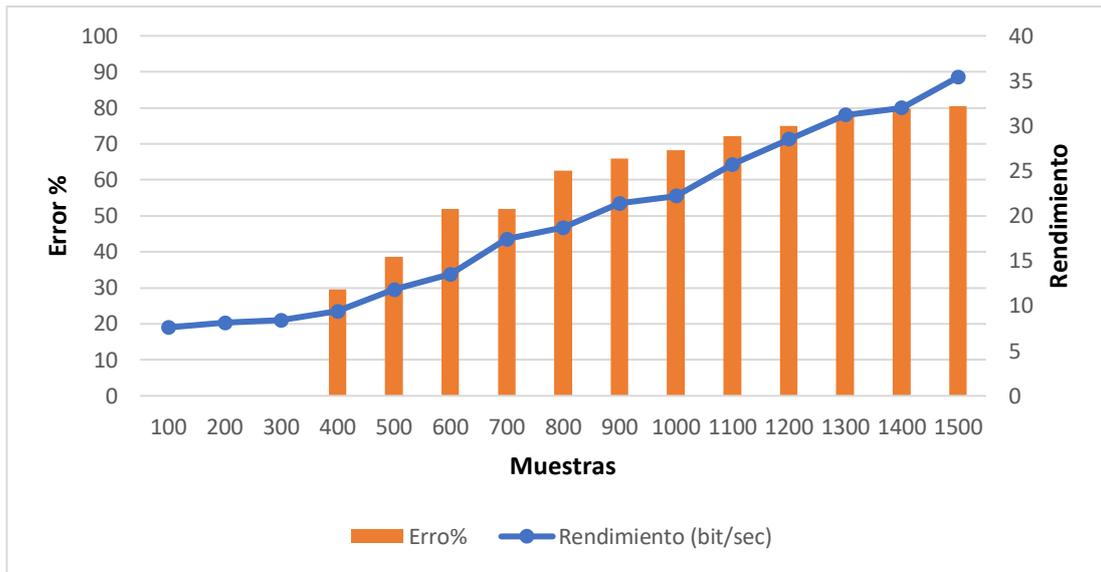


Fig. 56 Comparación entre las variables Muestras, Rendimiento y Error en MongoDB

El proceso de análisis siguiente ha sido efectuado para el sistema web conectado hacia la base de datos de Apache Cassandra, en el que se ha considerado los factores necesarios para realizar la correcta interpretación de los resultados establecidos en la tabla 61 donde se toma en cuenta las variables como: muestra, media, desviación estándar, rendimiento y error porcentual ya que ellas identifican los valores apropiados para verificar la eficiencia que presenta el sistema web.

TABLA 61
SELECCIÓN DE VARIABLES PARA EL ANÁLISIS DE LOS DATOS EN APACHE CASSANDRA

Base de Datos Apache Cassandra				
Muestras	Media	Desviación Estándar	Rendimiento (bit/sec)	Error %
100	12563	193,22	7,4	0
200	22734	528,67	8,4	0
300	29329	10333,05	8,6	16
400	24276	14333,57	11,2	34,5
500	27629	16141,45	12,5	32
600	22369	17634,58	14,7	49,17
700	19847	17810,32	16,9	57,43
800	18268	17565,07	19,1	62,25
900	14429	15353,53	23,4	69,56
1000	20708	24049,04	17,3	68,4
1100	16398	19423,21	22,5	72,36
1200	14263	16816,07	27,5	74
1300	11577	13541,21	35,3	76,15
1400	12284	14810,44	35	76,36
1500	10097	12782,21	39,1	82

Fuente: Propia

De acuerdo con la tabla 61 se obtiene la Fig. 57 que manifiesta una comparativa entre variables, en donde se confirma que la media da a conocer un tiempo promedio en milisegundos los mismos que van de acuerdo con el incremento que se estableció en la muestra, como también se comprueba como la desviación estándar indica la cantidad de variación existente entre los tiempos de respuesta que se vayan obteniendo conforme la muestra aumente.

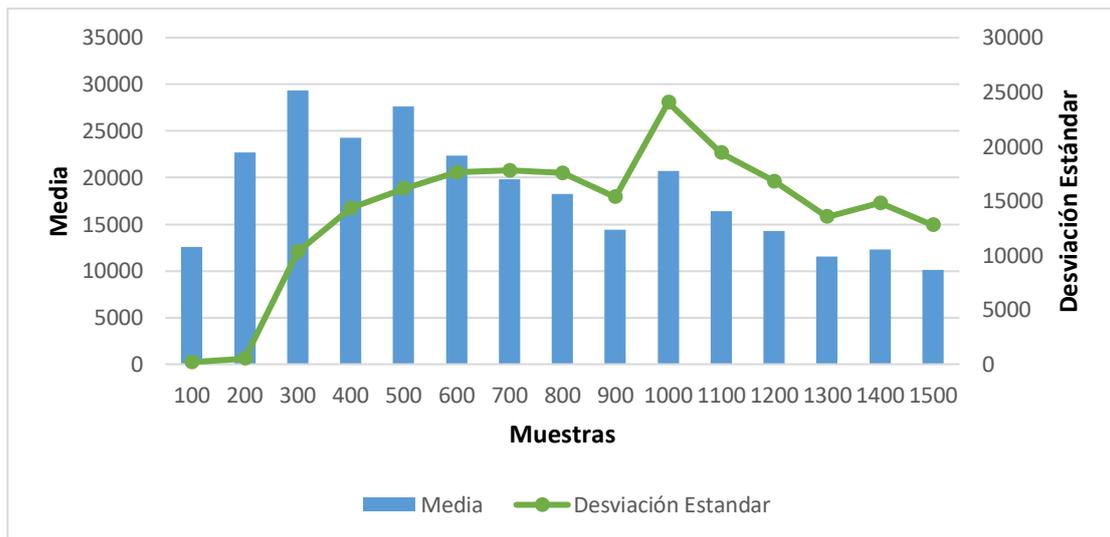


Fig. 57 Comparativa entre las variables Muestras, Media y Desviación Estándar en Apache Cassandra

De acuerdo con la tabla 61 es factible establecer una comparativa que se muestra en la Fig. 58, en donde se puede apreciar el porcentaje de error que se puede hallar conforme se van haciendo los ingresos de las muestras, como también se identifica el rendimiento que llega a representar el número de transacciones por minuto que es capaz de soportar el sistema web conforme las muestras van aumentando paulatinamente.

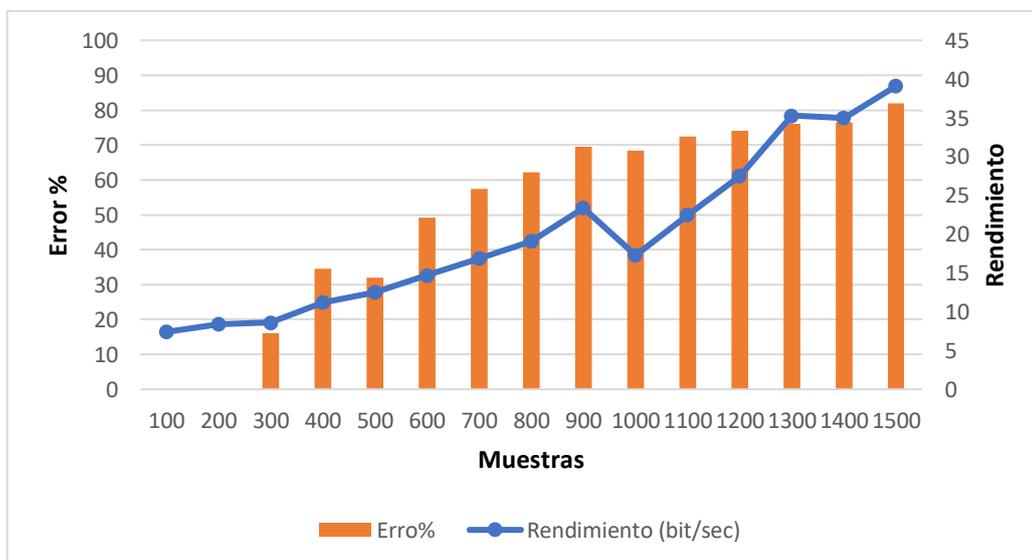


Fig. 58 Comparativa entre las variables Muestras, Rendimiento y Error en Apache Cassandra

Como análisis final se ha efectuado al sistema web conectado hacia la base de datos Redis, en donde se ha considerado los factores necesarios para realizar la correcta interpretación de los resultados los mismos que se manifiestan en la tabla 62, en donde se toma en cuenta las variables muestra, media, desviación estándar, rendimiento y error porcentual ya que las mismas identifican los valores adecuados para verificar la eficiencia que presenta el sistema web.

TABLA 62
SELECCIÓN DE VARIABLES PARA EL ANÁLISIS DE LOS DATOS EN REDIS

Base de Datos Redis				
Muestras	Media	Desviación Estándar	Rendimiento (bit/sec)	Error%
100	12600	135,33	7,5	0
200	23010	126,41	8,3	0
300	33920	661,15	8,5	0
400	31433	14402,19	9,7	24,75
500	22371	16058,67	13,2	45,4
600	19981	16723,89	15,3	53,83
700	20296	18173,27	16,6	57
800	18914	18337,51	18,4	61,75
900	16933	17592,63	21,2	65,89
1000	15607	17876,31	23	69,4
1100	14879	17114,92	25,2	72,36
1200	12038	14340,82	30,3	77,33
1300	12243	14336,71	34	75,15
1400	10854	13403,33	36,1	80,29
1500	10563	13410,36	37,8	81,27

Fuente: Propia

De acuerdo con la tabla 62 se obtiene la Fig. 59, en la cual se establece una comparativa entre las variables, en donde se confirma que la media da a conocer un tiempo promedio en milisegundos los cuales van de acuerdo con el incremento que se vaya estableciendo en la muestra, como también se comprueba como la desviación estándar indica la cantidad de variación existente entre los tiempos de respuesta que se vayan obteniendo conforme se incremente las muestras.

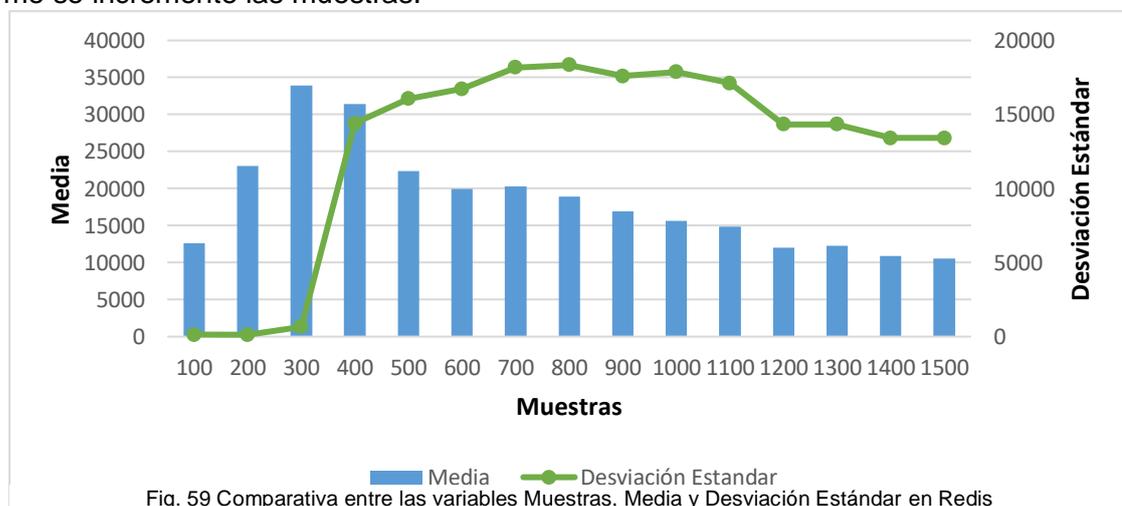


Fig. 59 Comparativa entre las variables Muestras, Media y Desviación Estándar en Redis

Con respecto a la tabla 62 es factible establecer otra comparativa que se la obtiene en la Fig. 60, en donde se puede apreciar el porcentaje de error que se puede hallar conforme se van realizando los ingresos de las muestras, como también se identifica el rendimiento que representa el número de transacciones por minuto que es capaz de soportar el sistema web conforme las muestras van aumentando.

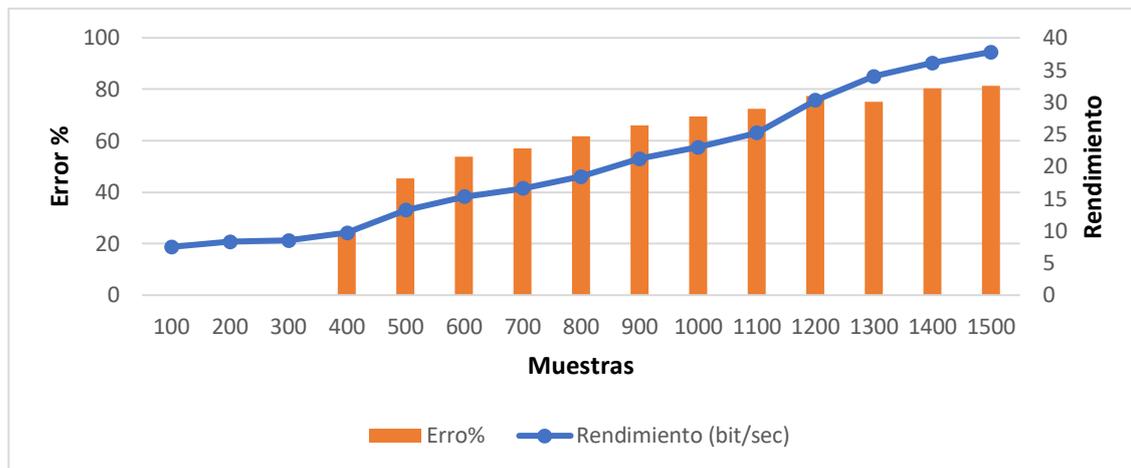


Fig. 60 Comparación entre las variables Muestras, Rendimiento y Error en Redis

El proceso de análisis de los resultados también se lo realizó para la mensajería del sistema web conectado hacia la base de datos MongoDB, en el cual se analizaron los datos enviados desde un usuario a otro con los cuales se fundamentó la tabla 63 que contiene los valores obtenidos para las variables cuantitativas como son: media, desviación Estándar, Rendimiento y porcentaje de error ya que las mismas poseen los valores necesarios para verificar la eficiencia que presenta el sistema web.

TABLA 63
SELECCIÓN DE VARIABLES PARA EL ANÁLISIS DE LO DATOS CON MONGOGO, PARA LA MENSAJERÍA

Base de Datos MongoDB				
Muestras	Media	Desviación Estándar	Rendimiento (bit/sec)	Error %
100	3	4,42	100,4	0
200	5	12,65	186,2	0
300	219	71,02	239,2	0
400	62	69,06	155,8	29,21
500	230	134,67	163	36,1
600	594	245,64	213,3	51,75
700	254	212,89	277,8	51,24
800	529	280,24	249,8	58,9
900	450	363,49	315,2	69,1
1000	164	200,06	336,7	68,05
1100	159	183,67	292,5	71,85
1200	142	118,77	414,8	78,14
1300	435	252,91	326,1	78,25
1400	203	179,08	406	80
1500	269	316,19	382,1	80,42

Fuente: Propia

De acuerdo con la tabla 63 se obtiene la Fig. 61, que establece una comparativa entre las variables, en donde se confirma que la media da a conocer un tiempo promedio en milisegundos los cuales van de acuerdo con el incremento que se vaya estableciendo en la muestra, como también se comprueba como la desviación estándar indica la cantidad de variación existente entre los tiempos de respuesta que se vayan obteniendo conforme la muestra se incremente.

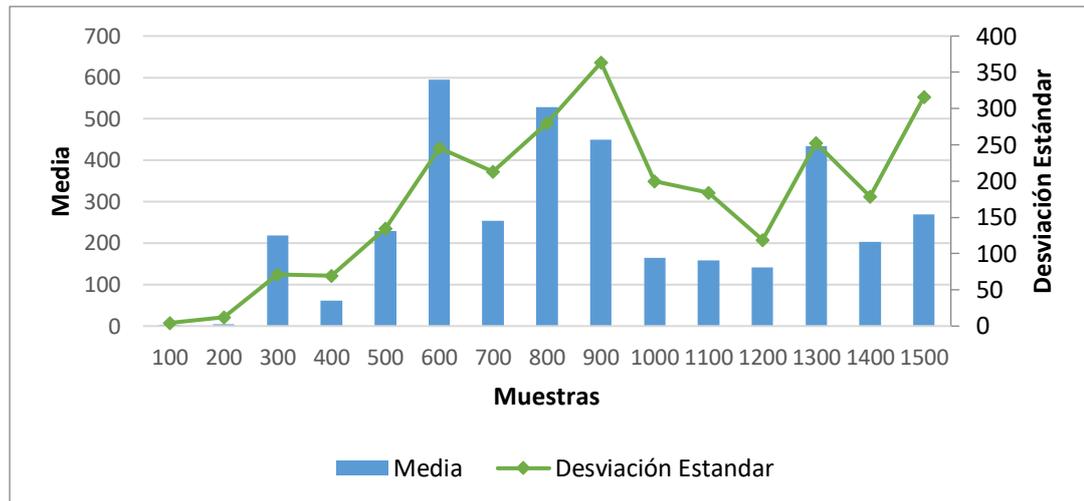


Fig. 61 Comparación de las variables Muestras, Media y Desviación Estándar en MongoDB

Con respecto a la tabla 63 es factible establecer una comparativa diferente que se indica en la Fig. 62, en donde se puede apreciar que el porcentaje de error se incrementa conforme se van aumentando los ingresos de las muestras, como también se identifica el rendimiento que representa el número de transacciones por minuto que es capaz de soportar el sistema web conforme las muestras van aumentando progresivamente.

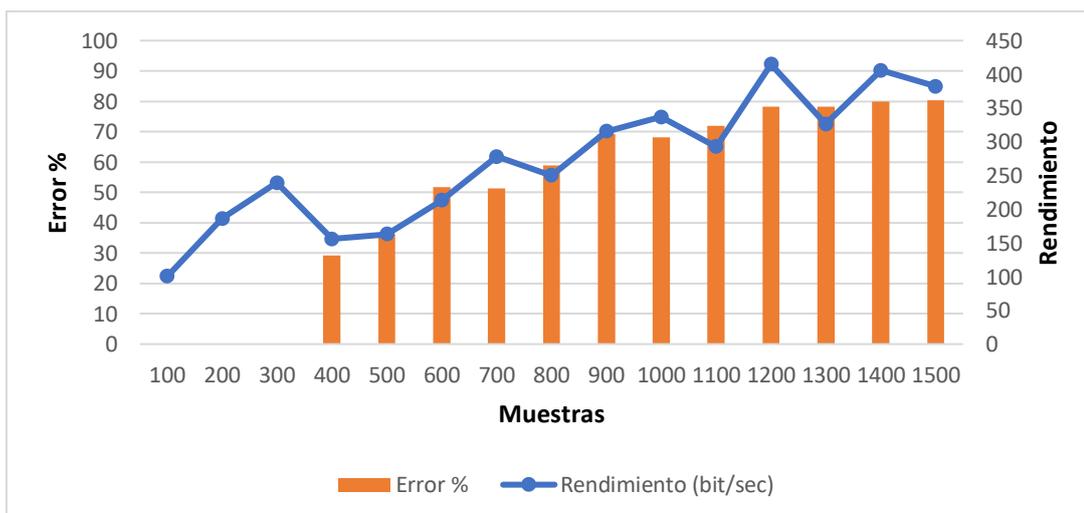


Fig. 62 Comparación de las variables Muestras, Rendimiento y Error en MongoDB

El proceso de análisis siguiente ha sido efectuado para el sistema web conectado hacia la base de datos de Apache Cassandra en donde se ha considerado los factores

necesarios para realizar la correcta interpretación de los resultados establecidos en la tabla 64, para lo cual se ha tomado en cuenta las siguientes variables: muestra, media, desviación estándar, rendimiento y error porcentual ya que las mismas identifican los valores adecuados para verificar la eficiencia que presenta el sistema web.

TABLA 64
SELECCIÓN DE VARIABLES PARA EL ANÁLISIS DE DATOS EN APACHE CASSANDRA, PARA LA MENSAJERÍA

Base de Datos Apache Cassandra				
Muestras	Media	Desviación Estándar	Rendimiento (bit/sec)	Error %
100	1	0,64	101,2	0
200	1	0,79	197,8	0
300	4	7,73	314,5	15,8
400	90	66,07	304,4	34,1
500	100	56,21	379,7	32,11
600	142	97,28	367	48,93
700	130	100,66	413,7	57,25
800	109	78,39	428,5	62
900	265	133,26	404,5	69,35
1000	100	88,57	394,8	68,07
1100	142	134,74	417,9	72,05
1200	221	209,9	398,7	73,55
1300	301	163,19	431,2	76,02
1400	143	139,04	418,9	76,12
1500	207	234,02	402,1	81,7

Fuente: Propia

De acuerdo con la tabla 64 se obtiene la Fig. 63, en la cual se establece una comparativa entre las variables, en donde se confirma que la media da a conocer un tiempo promedio en milisegundos los cuales van de acuerdo con el incremento que se vaya estableciendo en la muestra, como también se comprueba como la desviación estándar indica la cantidad de variación existente entre los tiempos de respuesta que se vayan obteniendo conforme la muestra se incremente.

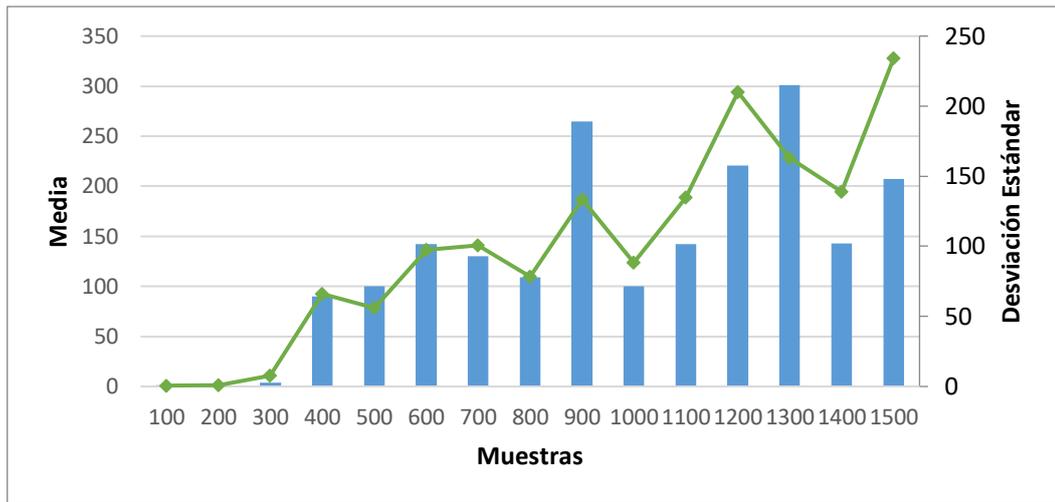


Fig. 63 Comparativa entre las variables Muestras, Media y Desviación Estándar en Apache Cassandra

Con respecto a la tabla 64 es factible establecer otra comparativa que se demuestra en la Fig. 64, en donde se puede apreciar el porcentaje de error que se puede encontrar conforme se van incrementando los ingresos de las muestras, como también se identifica el rendimiento que representa el número de transacciones por minuto que es capaz de soportar el sistema web conforme las muestras van aumentando progresivamente.

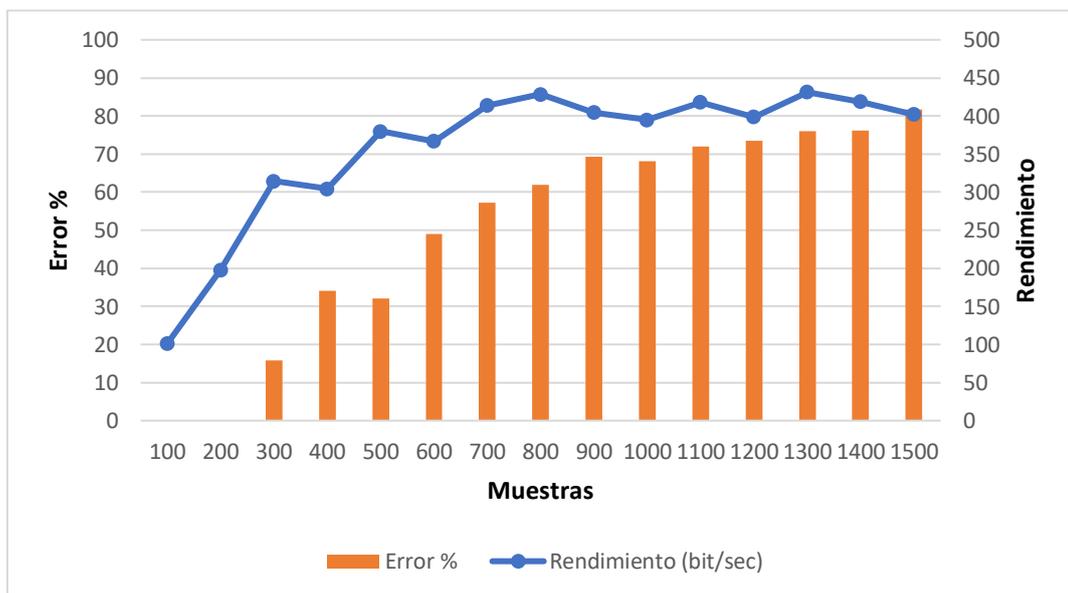


Fig. 64 Comparativa entre las variables Muestras, Rendimiento y Error en Apache Cassandra

Como punto final de análisis se ha efectuado al sistema web conectado hacia la base de datos Redis, en donde se ha considerado los factores necesarios para realizar la correcta interpretación de los resultados los cuales se han establecido en la tabla 65, en donde se tomó en cuenta las variables: muestra, media, desviación estándar, rendimiento y error porcentual ya que las mismas identifican los valores adecuados para verificar la eficiencia que presenta el sistema web.

TABLA 65
SELECCIÓN DE VARIABLES PARA EL ANÁLISIS DE DATOS EN REDIS, PARA LA MENSAJERÍA

Base de Datos Redis				
Muestras	Media	Desviación Estándar	Rendimiento (bit/sec)	Error %
100	5	8,48	101,8	0
200	69	57,43	183,3	0
300	120	67,38	214,7	0
400	320	174,07	193,3	24,5
500	319	154,94	218,2	45,2
600	250	241,54	237,8	53,15
700	213	159,09	266,3	56,8
800	633	400,2	242,6	61,5
900	292	290,08	301	65,32
1000	349	259,79	319,5	69,15
1100	302	238,85	305,6	72,16
1200	191	178,21	384,9	77,22
1300	131	115,64	429,3	75,05
1400	114	96,24	419,4	80
1500	387	168,08	391,4	81,07

Fuente: Propia

De acuerdo con la tabla 65 se obtiene la Fig. 65, en que se establece una comparativa entre variables, en donde se confirma que la media da a conocer un tiempo promedio en milisegundos los cuales van de acuerdo con el incremento que se vaya estableciendo en la muestra, como también se comprueba como la desviación estándar indica la cantidad de variación existente entre los tiempos de respuesta que se vayan obteniendo conforme la muestra se incremente.

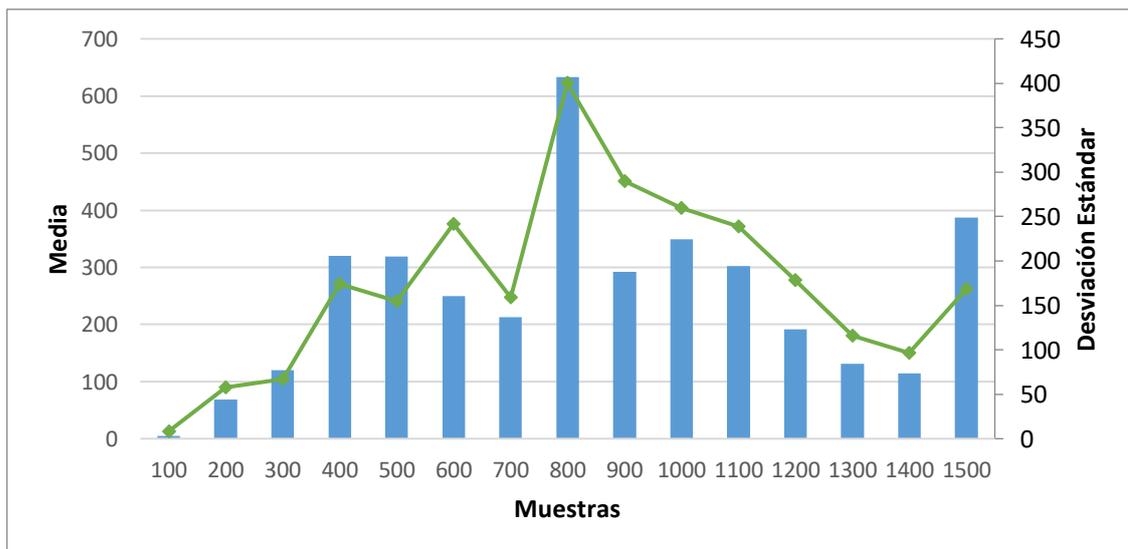


Fig. 65 Comparativa entre las Variables Muestras, Media y Desviación Estándar en Redis

Con respecto a la tabla 65 es factible establecer una comparativa que se manifiesta en la Fig. 66, en donde se puede apreciar el porcentaje de error que se puede encontrar conforme se van incrementando los ingresos de las muestras, como también se identifica el rendimiento que consiste en representar el número de transacciones por minuto que es capaz de soportar el sistema web conforme las muestras van aumentando progresivamente.

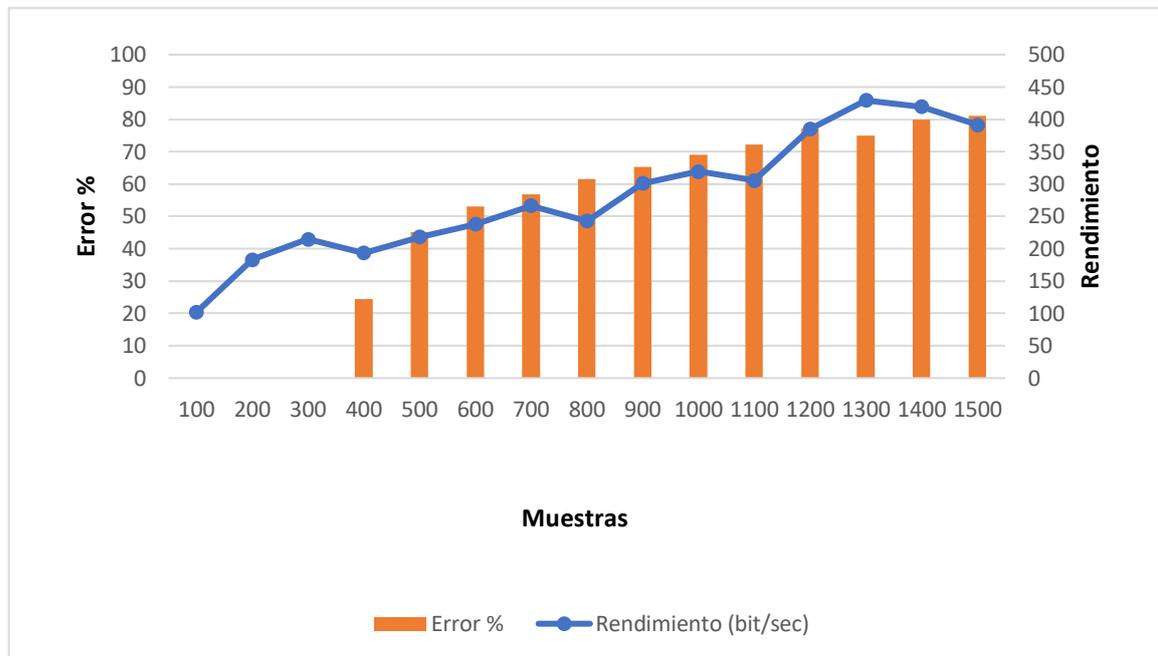


Fig. 66 Comparativa entre las variables Muestras, Rendimiento y Error en Redis

3.4 Validación de Resultados

Los datos analizados en las secciones anteriores sirvieron para identificar la eficiencia obtenida en cada una de las redes sociales conectadas hacia las bases de datos NoSQL y mediante ese proceso se determinó cuál de las tres es la mejor opción para trabajar con grandes cantidades de información, pero para esta sección de validación ha sido necesario comprobar la relación existente entre los valores y para realizar esta interpretación, se ha utilizado lo que es la regresión lineal con el objetivo de analizar las variables como media, desviación estándar, error porcentual, rendimiento y kb por segundo recibidos y enviados.

Para el análisis de los resultados es necesario verificar el tipo de regresión lineal oportuno a utilizarse para los valores obtenidos de las tres bases de datos NoSQL, por lo tanto, para probar este punto se ha optado por manejar las pruebas de Shapiro Wilk y para ello se ha utilizado el software estadístico R que contiene herramientas para hacer cálculos estadístico, además posee una gran variedad de técnicas estadísticas como modelos lineales y no lineales, pruebas estadísticas y análisis, clasificación y agrupamiento de series

temporales para el cálculo estadístico de variables de estudio. (El proyecto R para computación estadística, 2019).

Con la descripción dada anteriormente se pudo ingresar los datos obtenidos de Jmeter dentro del análisis que realiza R y se pudo conseguir mediante cálculos los siguientes datos que se encuentran descritos en la tabla 66. En donde se definió que el p-value debe ser menor o igual a 0,05 para que los datos presenten una distribución normal y en caso de que no se cumpla con esta regla los datos presentados no tendrán una distribución normal.

TABLA 66
P-VALUE DE LAS TRES BASES DE DATOS

Prueba de Shapiro Wilk				
Bases de Datos	MongoDB	Apache	Redis	Cumple con
NoSQL		Cassandra		p-value
Media	0,08479	0,5412	0,07176	Si
Desviación Estándar	0,00896	0,02031	0,03585	Si
Error %	0,01183	0,02512	0,007807	Si
Rendimiento (bit/sec)	0,2272	0,1812	0,2047	Si
Recibidos (kb/sec)	0,2139	0,07585	0,1283	Si
Enviados (kb/sec)	0,7196	0,003755	0,04877	Si

Fuente: Propia

Con los valores obtenidos en la tabla 66 se puede decir que las variables si cumplen con la propiedad de p-value ya que si tienen una distribución normal, por lo tanto cumplen también con la regla de la normalidad de datos, esto quiere decir que de acuerdo con los valores consignados se puede aplicar las pruebas de correlación de Pearson la misma que ayudará a verificar si existe una covariación lineal entre dos variables cuantitativas, y para designar las variables a analizarse hay que ver su grado relación existen para lo cual es necesario utilizar la siguiente formula:

$$r_{xy} = \frac{\sum xy}{\sqrt{\sum x^2} \sqrt{\sum y^2}}$$

Para verificar el método de correlación de Pearson es necesario aplicar la siguiente regla:

$$1 \geq r_{xy} \geq -1 \text{ donde } r_{xy} = \text{Coeficiente de correlación}$$

Con la regla anterior puntualizada se procede a realizar el análisis entre las variables y se obtiene la figura 67, donde se muestra la matriz generada en RStudio de la comparación de las variables cuantitativas del sistema web conectado a la base de datos MongoDB.

	Muestras	Media	Desviacion	Error	Rendimiento	Recibidos	Enviados
Muestras	1.00	-0.67	0.61	0.94	0.99	0.98	-0.01
Media	-0.67	1.00	-0.25	-0.63	-0.74	-0.73	0.23
Desviacion	0.61	-0.25	1.00	0.83	0.53	0.49	-0.15
Error	0.94	-0.63	0.83	1.00	0.91	0.88	-0.08
Rendimiento	0.99	-0.74	0.53	0.91	1.00	1.00	0.01
Recibidos	0.98	-0.73	0.49	0.88	1.00	1.00	0.09
Enviados	-0.01	0.23	-0.15	-0.08	0.01	0.09	1.00

Fig. 67 Matriz de Correlación de Pearson de la Base de Datos MongoDB

Es necesario obtener el análisis entre las variables del sistema web conectado hacia la base de datos de Apache Cassandra donde se obtiene la Fig. 68 en donde se explica la comparación obtenida entre las variables cuantitativas.

	Muestras	Media	Desviacion	Error	Rendimiento	Recibidos	Enviados
Muestras	1.00	-0.67	0.55	0.94	0.96	0.93	-0.12
Media	-0.67	1.00	0.05	-0.56	-0.76	-0.74	0.10
Desviacion	0.55	0.05	1.00	0.75	0.34	0.27	-0.37
Error	0.94	-0.56	0.75	1.00	0.86	0.81	-0.19
Rendimiento	0.96	-0.76	0.34	0.86	1.00	0.99	0.08
Recibidos	0.93	-0.74	0.27	0.81	0.99	1.00	0.20
Enviados	-0.12	0.10	-0.37	-0.19	0.08	0.20	1.00

Fig. 68 Matriz de Correlación de Pearson de la Base de Datos Apache Cassandra

Para culminar el análisis, se lo ha realizado al sistema web conectado hacia la base de datos Redis en donde se ha obtenido la Fig. 69, en donde se explica la comparación existente entre las variables cuantitativas.

	Muestras	Media	Desviacion	Error	Rendimiento	Recibidos	Enviados
Muestras	1.00	-0.69	0.59	0.93	0.99	0.98	-0.09
Media	-0.69	1.00	-0.28	-0.67	-0.75	-0.72	0.50
Desviacion	0.59	-0.28	1.00	0.82	0.47	0.42	-0.30
Error	0.93	-0.67	0.82	1.00	0.88	0.85	-0.25
Rendimiento	0.99	-0.75	0.47	0.88	1.00	1.00	-0.08
Recibidos	0.98	-0.72	0.42	0.85	1.00	1.00	-0.01
Enviados	-0.09	0.50	-0.30	-0.25	-0.08	-0.01	1.00

Fig. 69 Matriz de Correlación de Pearson de la Base de Datos Redis

Con las matrices de Correlación de Pearson conseguidas de forma general, se procede a verificar la correlación que existe entre dos variables que están fuertemente relacionadas, en este caso para las tres bases de datos sea elegido realizar las pruebas de la siguiente manera: muestra-rendimiento y rendimiento-error.

A continuación, se da a conocer mediante figuras cuales fueron los resultados conseguidos de la correlación entre dos variables fuertemente relacionadas.

Los resultados obtenidos para la base de datos MongoDB se detallan en la Fig. 70, donde se especifica que existen una correlación de 0.99 entre las variables muestra y rendimiento lo que indica que las dos variables tienen una relación fuerte alta positiva, por lo tanto, se determina que a mayor número de usuarios registrados mayor será el rendimiento.

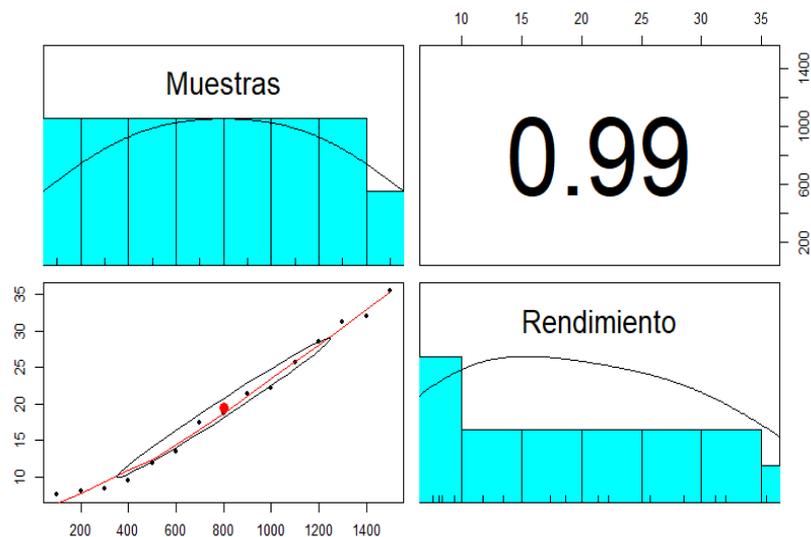


Fig. 70 Resultado de Correlación Muestra-Rendimiento

Mientras que el resultado obtenido para el análisis de correlación entre las variables rendimiento y porcentaje de error es de 0.91 el mismo que es definido en la Fig. 71, indica que existe una fuerte relación alta positiva, por lo cual se concluye que a mayor rendimiento obtenido mediante las muestras el porcentaje de error aumentará.

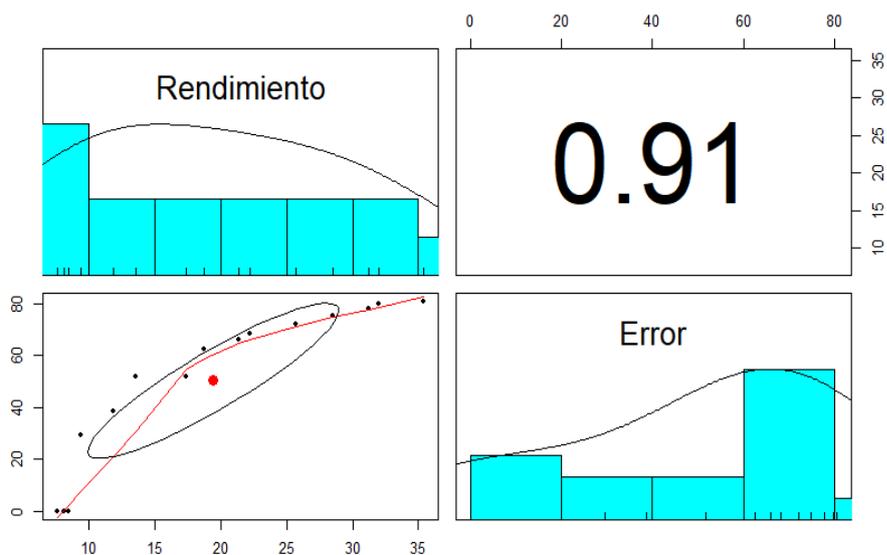


Fig. 71 Resultado de Correlación Rendimiento-Error

En cuando a los análisis realizados para el sistema web conectando a la base de datos de Apache Cassandra, determina que la correlación existente entre las variables muestra y rendimiento es de 0.96, cuyo resultado está definido en la Fig. 72 que indica que las dos variables tienen una relación fuerte alta positiva, por lo tanto, se determina que a mayor número de usuarios registrados mayor será el rendimiento.

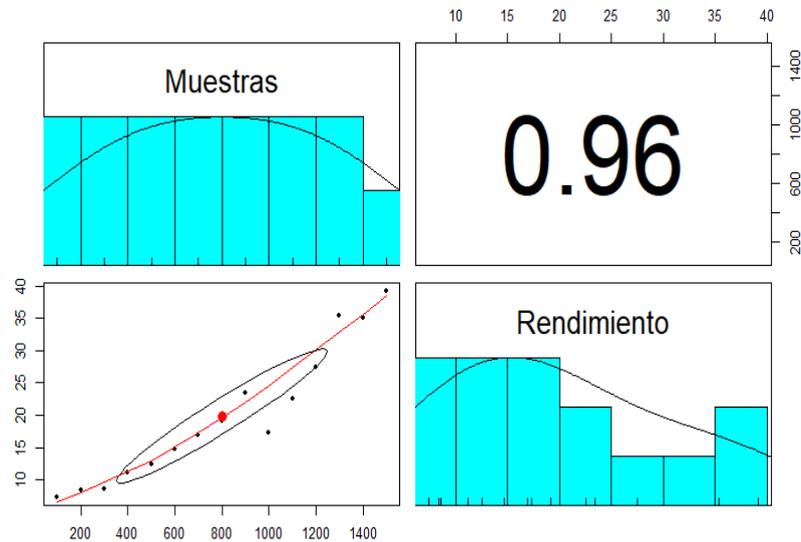


Fig. 72 Resultado de Correlación Muestra-Rendimiento

Para cumplir con el análisis hacia las variables cuantitativas se ha determinado obtener la correlación existente entre las variables rendimiento y porcentaje de error cuyo valor obtenido es de 0.86 el cual se lo ve definido en la Fig. 73 y mediante la misma se puede definir que existe una fuerte relación alta positiva, por lo cual se concluye que a mayor rendimiento obtenido mediante las muestras el porcentaje de error aumentara.

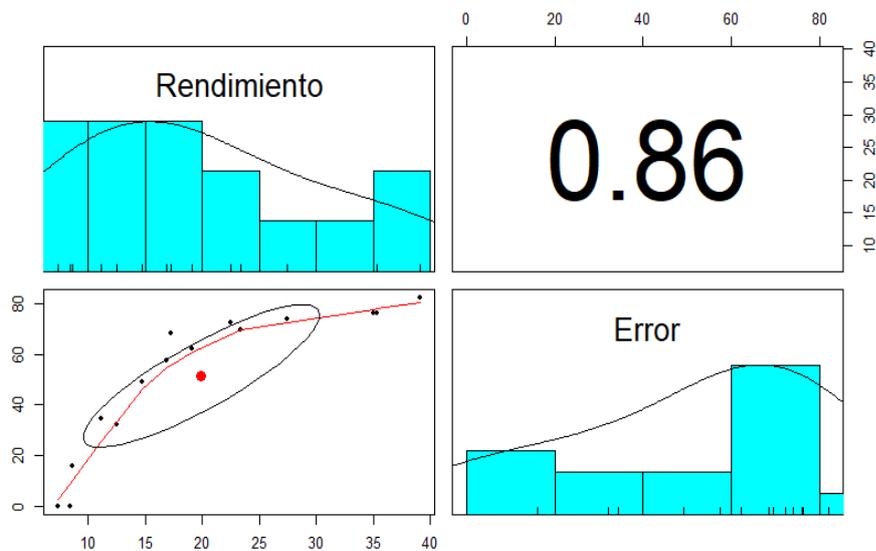


Fig. 73 Resultado de Correlación Rendimiento-Error

Como parte final del análisis se lo realizo al sistema web conectado hacia la base de datos Redis, en donde se determina que la correlación existente entre las variables muestra y rendimiento es de 0.99, cuyo resultado está definido en la Fig. 74, indica que las dos variables tienen una relación fuerte alta positiva, por lo tanto, se determina que a mayor número de usuarios registrados mayor será el rendimiento

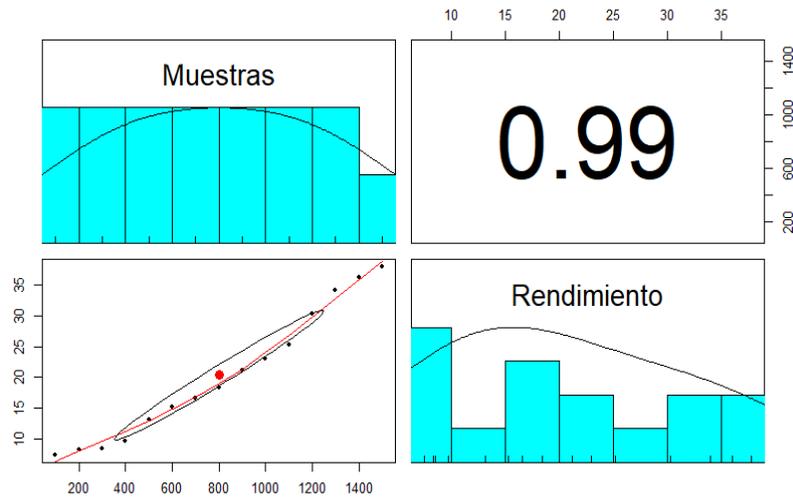


Fig. 74 Resultado de Correlación Muestra-Rendimiento

En cuanto al análisis entre las variables cuantitativas respecta se ha podido definir que la correlación existente entre las variables rendimiento y error porcentual tiene como valor 0.88 el cual es descrito en la Fig. 75 en donde se puede identificar que las dos variables tienen una fuerte relación alta positiva, por lo tanto, se concluye que a mayor rendimiento obtenido mediante las muestras el porcentaje de error aumentará.

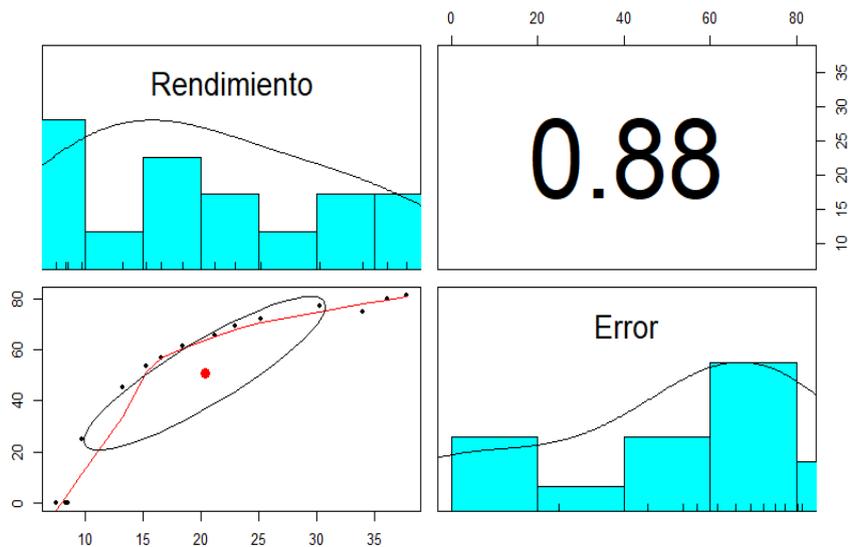


Fig. 75 Resultado de Correlación Rendimiento-Error

CONCLUSIONES

- Mediante la indagación en diferentes bases de datos bibliográficas como IEEE y SCOPUS se pudo recolectar información que proporcionó datos relevantes sobre cada una de las bases de datos a estudiarse y mediante la misma se dedujo que las bases de datos NoSQL cuentan con beneficios tanto para el almacenamiento como tratamiento de grandes cantidades de información en diferentes entornos web de producción de datos.
- De acuerdo con el alcance establecido en el anteproyecto se investigó las funcionalidades con las que cuenta los prototipos de red social y de acuerdo con su complejidad se las repartió en las historias de usuario y con ello se pudo establecer el orden apropiado para desarrollar los prototipos con las funcionalidades establecidas para que los usuarios registrados puedan interactuar entre sí.
- De acuerdo con la norma ISO/IEC 25012 y a la característica de eficiencia se establece a que aspecto se debe regir la fase de pruebas para encontrar el resultado de la velocidad de respuesta de los datos.
- Con los resultados obtenidos de las pruebas realizadas con el software Jmeter se pudo conseguir los valores para las variables: muestras, media, desviación estándar, error porcentual y rendimiento con las cuales se pudo realizar el respectivo análisis y aplicando la característica de eficiencia de la Norma ISO/IEC 25012 se pudo con llegar a la conclusión de que MongoDB es la base de datos NoSQL que controla mejor lo que es la velocidad de respuesta de los datos, ya que se pudo comprobar que al momento de registrar los datos MongoDB tuvo un total de 291,3 bit/sec y para el envío de mensajes de usuario a usuario se tuvo un total de 4058,9 bit/sec, los dos totales fueron comparados con los resultados de las otras dos bases de datos y se pudo comprobar mediante estas pruebas a la mejor base de datos NoSQL dentro de este marco de estudio.
- El software R al contar con herramientas diferentes ayudo a calcular los valores estadísticos necesarios para determinar el tipo de correlación existente entre las variables cuantitativas y de acuerdo con ello se consiguió establecer el grado de correlación de Pearson para el grupo de variables que se encuentran fuertemente relacionadas.

RECOMENDACIONES

- Para la implementación del marco teórico es necesario realizar las investigaciones dentro de fuentes bibliográficas que cuenten con investigaciones de alto impacto, ya que es ahí donde se puede encontrar ya sea artículos o libros científicos que pueden ayudar con las investigaciones de los estudiantes.
- Es adecuado verificar el alcance establecido en el anteproyecto para poder realizar un levantamiento de requisitos y poder con ello adecuar las historias de usuario de una forma ordenada y con los respectivos tiempos para que el desarrollo de los prototipos de red social esté con el tiempo estipulado y con las respectivas funcionalidades establecidas por el product owner.
- Es necesario comprender como está estructurada la norma ISO/IEC 25012, ya que de acuerdo con ella se establece que característica es la esencial para poder realizar tanto las pruebas, análisis e interpretación de los resultados y con ello poder concluir cuál de las tres bases de datos NoSQL cumple con verificar la mejor velocidad de respuesta en lo que respecta al tratamiento de datos.
- Con las pruebas, variables cuantitativas y la característica de eficiencia de la Norma ISO/IEC 25012, se debe realizar las observaciones oportunas para seleccionar las variables con mayor importancia y con ello poder analizar e interpretar los resultados de forma correcta con el fin de poder llegar a una conclusión exacta y verídica.
- Dentro del área de la ingeniería es indispensable que los profesionales y estudiantes tengan conocimiento sobre lo que es la estadística ya que con ella se puede dar a conocer a las personas que los productos que se desarrollan tienen la valía y la calidad adecuada ya que cuentan con un respaldo teórico donde se prueba que mediante métodos, técnicas o modelos estadísticos el software desarrollado es completamente aceptable o tiene cierto grado de falencias a ser corregidos.

GLOSARIO DE TÉRMINOS

Flexibilidad: Es el hecho en donde se puede agregar datos al sistema posteriormente sin tener que rescribir lo que ya se tiene dentro de las bases de datos.

Escalabilidad: Es la propiedad de aumentar la capacidad de trabajo o de tamaño de un sistema sin comprometer el funcionamiento y la calidad de un sistema web.

Atomicidad: Es una propiedad que asegura que una operación se ha realizado o no y por lo tanto ante un fallo del sistema no puede quedar a medias.

Consistencia: Es el estado coherente de la información o datos que contiene y que se relacionan en el cual la información cumple con las necesidades o expectativas de quien lo requiera.

Cluster: Es una tecnología que permite el clustering de base de datos en memoria en un ambiente de no compartición, la arquitectura de no compartición permite que el sistema gestor de bases de datos funcione utilizando hardware no muy costoso y con requerimientos mínimos tanto en software como en hardware.

Framework: Considerado como una plataforma de software reutilizable ya que cuenta con una base de códigos estructurada como base para el inicio del desarrollo de aplicaciones y soluciones web.

Multiplataforma: Es un atributo conferido para programas informáticos o métodos y conceptos de cómputo que son implementados en varias plataformas informáticas.

Dependencias: Son elementos clave dentro de las aplicaciones en node.js ya que son una parte lógica de la aplicación y son necesarias para que sistema web funcione correctamente en producción.

Seguridad: Es la protección que se da para resguardar la información u objetos mediante la aplicación de normas que se ajusten para controlar la no autorización a la misma por parte de agentes externos.

Escalabilidad: Es el proceso de un software para adaptarse al rendimiento producido por el incremento de números de usuarios y su alta demanda en niveles de transacciones, lo que produce que las bases de datos sufran una degradación en el performance.

Rendimiento: Es el encargado de medir mediante pruebas la velocidad, fiabilidad y estabilidad de un sistema web ante demanda de transacciones de datos.

Performance: Es el desempeño realizado con respecto al rendimiento de un software.

Metodología: Mediante estrategias, técnicas y formas de trabajo de desarrollo de software se establece prácticas adaptativas que se centren en la funcionalidad del producto a entregarse.

REFERENCIAS BIBLIOGRÁFICAS

- Ali, K. (2017). A Study of Software Development Life Cycle Process Models Kazim. *International Journal of Advanced Research in Computer Science*, 8(1), 15–23.
- Saraza A. (2016). Introducción a las bases de datos NoSQL usando MongoDB. In *Journal of Chemical Information and Modeling* (Vol. 53). <https://doi.org/10.1017/CBO9781107415324.004>
- Bases de datos no relacionales | Bases de datos de gráficos | AWS. (2017). Retrieved February 12, 2020, from <https://aws.amazon.com/es/nosql/>
- Bases de datos SQL | AWS. (2017). Retrieved February 12, 2020, from <https://aws.amazon.com/es/relational-database/>
- Bosonit. (2018). Tipos de bases de datos NoSQL 2: Clave-valor, documentales y columnares. - Bosonit. Retrieved May 8, 2019, from <https://bosonit.com/tipos-de-bases-de-datos-nosql-2-clave-valor-documentales-y-columnares/>
- Silberschatz, A. (Bell L., Korth, H. F. (Bell L., & Sudarshan, S. (2010). Fundamentos de bases de datos. In Victoria. <https://doi.org/10.1017/CBO9781107415324.004>
- ¿Qué son las bases de datos documentales? | Kyocera.(2017). Retrieved February 12, 2020, from <https://www.kyoceradocumentsolutions.es/smarter-workspaces/business-challenges/paperless/que-son-las-bases-de-datos-documentales.html>
- ¿Qué es NoSQL y MongoDB? Bases de datos no estructuradas. (2020). Retrieved February 12, 2020, from <https://revistadigital.inesem.es/informatica-y-tics/bases-datos-nosql-mongodb/>
- Lourenço, J. R., Cabral, B., Carreiro, P., Vieira, M., & Bernardino, J. (2015). Choosing the right NoSQL database for the job: a quality attribute evaluation. *Journal of Big Data*, 2(1). <https://doi.org/10.1186/s40537-015-0025-0>
- Zhao, Y. (2013). Research on mongodb design and query optimization in vehicle management information system. *Applied Mechanics and Materials*, 246–247, 418–422. <https://doi.org/10.4028/www.scientific.net/AMM.246-247.418>
- hao, Y. (2013). Research on mongodb design and query optimization in vehicle management information system. *Applied Mechanics and Materials*, 246–247, 418–422. <https://doi.org/10.4028/www.scientific.net/AMM.246-247.418>
- Cuadrante Mágico para Sistemas Operativos de Gestión de Bases de Datos. (2019). Retrieved April 22, 2020, from <https://www.gartner.com/en/documents/3975492>
- Chen, Z., Yang, S., Tan, S., He, L., Yin, H., & Zhang, G. (2015). A new fragment re-allocation strategy for NoSQL database systems. *Frontiers of Computer Science*, 9(1), 111–127. <https://doi.org/10.1007/s11704-014-3480-4>
- Trigas Gallego, M., & Domingo Troncho, A. C. (2012). Gestión de Proyectos Informáticos. Metodología Scrum. *Openaccess.Uoc.Edu*, 56. Retrieved from <http://www.quimbiotec.gob.ve/sistem/auditoria/pdf/ciudadano/mtrigasTFC0612memoria.pdf%5Chttp://openaccess.uoc.edu/webapps/o2/bitstream/10609/17885/1/mtrigasTFC0612memoria.pdf>
- Rocha, Á., Correia, A. M., Costanzo, S., & Reis, L. P. (2015). New Contributions in Information Systems and Technologies. *Advances in Intelligent Systems and Computing*, 353, III–IV. <https://doi.org/10.1007/978-3-319-16486-1>
- Imam, A. A., Basri, S., Ahmad, R., Watada, J., & González-Aparicio, M. T. (2018). Automatic schema suggestion model for NoSQL document-stores databases. *Journal of Big Data*, 5(1), 1–17. <https://doi.org/10.1186/s40537-018-0156-1>
- Calvo, K., Durán, J., Quirós, E., & Malinowski, E. (2017). MongoDB: alternativas de implementar y consultar documentos. *IX Congreso Internacional de Computación y Telecomunicaciones*, (February 2018), 78–86. <https://doi.org/ISSN 2519-0679>
- Ladero I. ¿Qué es Cassandra? - BAOSS. (2020). Retrieved February 12, 2020, from

- <https://www.baoss.es/que-es-cassandra/>
- Colombo, P., & Ferrari, E. (2016). Fine-Grained Access Control Within NoSQL Document-Oriented Datastores. *Data Science and Engineering*, 1(3), 127–138. <https://doi.org/10.1007/s41019-016-0015-z>
- Mehmood, N. Q., Culmone, R., & Mostarda, L. (2017). Modeling temporal aspects of sensor data for MongoDB NoSQL database. *Journal of Big Data*, 4(1). <https://doi.org/10.1186/s40537-017-0068-5>
- Paramio C. Una introducción a MongoDB. (2011). Retrieved February 12, 2020, from <https://www.genbeta.com/desarrollo/una-introduccion-a-mongodb>
- ISO 25012. (2008). Retrieved February 12, 2020, from <https://iso25000.com/index.php/normas-iso-25000/iso-25012>
- Mero K. Diseño y Desarrollo del Software | Sistemas de Información. (2020). Retrieved March 26, 2020, from <https://blogereducativo.wordpress.com/disen-y-desarrollo-del-software>
- Gracia del Busto, H., & Enríquez, O. Y. (2013). Bases de datos NoSQL. *Revista Telemática*, 11(3), 21–33. Retrieved from <http://revistatelematica.cujae.edu.cu/index.php/tele/article/view/74/74>
- Elghamrawy, S. M., & Hassanien, A. E. (2017). A partitioning framework for Cassandra NoSQL database using Rendezvous hashing. *Journal of Supercomputing*, 73(10), 4444–4465. <https://doi.org/10.1007/s11227-017-2027-5>
- Ben Brahim, M., Drira, W., Filali, F., & Hamdi, N. (2016). Spatial data extension for Cassandra NoSQL database. *Journal of Big Data*, 3(1), 1–16. <https://doi.org/10.1186/s40537-016-0045-4>
- Garcete, A. (2014). Base de Datos Orientado a Columnas. 2014.
- Carlson, J. L. (2013). Redis in Action. In *Media.johnwiley.com.au*.
- Arefin, M. S., & Hossain, K. N. (2015). Efficiency of NoSQL Databases. 213–227.
- S, P. F. D. C. G., Juan, N., & López, J. (2018). Cassandra_Nosql.
- Guía, L., Scrum, D. De, & Reglas, L. (2013). La Guía de Scrum
- Rossel, G. (2016). NoSQL- Column Family.
- Schwaber & Sutherland. (2013). La Guía de Scrum. 21
- Rossel, G. (2016). NoSQL- Column Family
- López, J. (2018). Cassandra_Nosql. 116
- Services, A. W. (2017). Database Caching Strategies Using Redis. (May).
- Lewandowski, C. M. (2015). the Little Redis Book. The Effects of Brief Mindfulness Intervention on Acute Pain Experience: An Examination of Individual Difference, 1. <https://doi.org/10.1017/CBO9781107415324.004>
- Khatibi, E., & Mirtaheri, S. L. (2019). A dynamic data dissemination mechanism for Cassandra NoSQL data store. *Journal of Supercomputing*, 75(11), 7479–7496. <https://doi.org/10.1007/s11227-019-02959-7>
- Lee, M., Jeon, S., & Song, M. (2018). Characterizing user interest in NoSQL databases of social question and answer data. *Journal of Supercomputing*, 76(5), 1–16. <https://doi.org/10.1007/s11227-018-2293-x>
- Porras, J. (2020). Primeros pasos scrum. Retrieved May 20, 2020, from <https://synapptica.net/metodologia-scrum.html>
- Mardan, A. (2018). Persistence with MongoDB and Mongoskin. *Practical Node.js*, 165–204. https://doi.org/10.1007/978-1-4842-3039-8_5
- Imam, A. A., Basri, S., Ahmad, R., Watada, J., & González-Aparicio, M. T. (2018). Automatic schema suggestion model for NoSQL document-stores databases. *Journal of Big Data*, 5(1), 1–18. <https://doi.org/10.1186/s40537-018-0156-1>
- Gessert, F., Wingerath, W., Friedrich, S., & Ritter, N. (2017). NoSQL database systems: a survey and decision guidance. *Computer Science - Research and Development*, 32(3–4), 353–365. <https://doi.org/10.1007/s00450-016-0334-3>
- Saxena, U., Sachdeva, S., & Batra, S. (2015). Moving from relational data storage to decentralized structured storage system. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 8999, 180–194. https://doi.org/10.1007/978-3-319-16313-0_13

- Branagan, C., & Crosby, P. (2013). Understanding the Top 5 Redis Performance Metrics. Retrieved from <http://en.wikipedia.org/wiki/Publish/subscribe>
- Vivas, H., Muñoz, H., Cambarieri, M., & Petroff, M. (2014). Arquitectura de software con websocket para aplicaciones web multiplataforma. *ResearchGate*, (October), 11. <https://doi.org/10.13140/2.1.5084.3845>
- Castro Romero, A., González, J., & Callejas Cuervo, M. (2012). Utilidad y funcionamiento de las bases de datos NoSQL. *Facultad de Ingeniería*, 21(33), 21–32. <https://doi.org/10.19053/01211129.2115>
- Yue, M., Yuming, S., Yuefei, S., & Cungen, C. (2010). The description logic for relational databases. *IFIP Advances in Information and Communication Technology*, 340 AICT(2007), 64–71. <https://doi.org/10.1007/978-3-642-16327-2-11>
- Menzinsky, A., López, G., & Palacio, J. (2016). Scrum Manager: Guía de formación. In *Gestión de proyectos*. Retrieved from http://www.scrummanager.net/files/sm_proyecto.pdf
http://www.scrummanager.net/files/scrum_manager.pdf
- Tutorialspoint. (2019). About the Tutorial Copyright & Disclaimer. Tutorialspoint (I) Pvt. Ltd.
- Calvo, K., Durán, J., Quirós, E., & Malinowski, E. (2017). MongoDB: alternativas de implementar y consultar documentos. *IX Congreso Internacional de Computación y Telecomunicaciones*, (February 2018), 78–86. <https://doi.org/ISSN 2519-0679>
- Herranz Gómez, R. (2014). Proyecto Fin De Carrera Bases De Datos Nosql : Arquitectura Y. 1–159.
- Apache JMeter - Apache JMeter™. (2019). Retrieved February 11, 2020, from <http://jmeter.apache.org/>
- R: El proyecto R para computación estadística. (2019). Retrieved April 22, 2020, from <https://www.r-project.org/>
- Castillo, J. N., Garcés, J. R., Navas, M. P., Segovia, D. F. J., & Naranjo, J. E. A. (2017). Base de Datos NoSQL: MongoDB vs. Cassandra en operaciones CRUD (Create, Read, Update, Delete). *Revista Publicando*, 4(11 (1)), 79–107.
- Chebotko, A., Kashlev, A., & Lu, S. (2015). A Big Data Modeling Methodology for Apache Cassandra. *Proceedings - 2015 IEEE International Congress on Big Data, BigData Congress 2015*, (June), 238–245. <https://doi.org/10.1109/BigDataCongress.2015.41>
- Pokorny, J. (2013). NoSQL databases: A step to database scalability in web environment. *International Journal of Web Information Systems*, 9(1), 69–82. <https://doi.org/10.1108/17440081311316398>
- Eisinga, R., Grotenhuis, M. Te, & Pelzer, B. (2013). The reliability of a two-item scale: Pearson, Cronbach, or Spearman-Brown? *International Journal of Public Health*, 58(4), 637–642. <https://doi.org/10.1007/s00038-012-0416-3>
- Feng, W., Zhu, Q., Zhuang, J., & Yu, S. (2019). An expert recommendation algorithm based on Pearson correlation coefficient and FP-growth. *Cluster Computing*, 22(s3), 7401–7412. <https://doi.org/10.1007/s10586-017-1576-y>
- Giroladini, W., Pederzoli, L., Bilucaglia, M., Melloni, S., & Tressoldi, P. (2016). A new method to detect event-related potentials based on Pearson's correlation. *Eurasip Journal on Bioinformatics and Systems Biology*, 2016(1). <https://doi.org/10.1186/s13637-016-0043-z>
- Tangen, Ø. R. (2015). Real-Time Web with WebSocket. 28–29. Retrieved from <https://www.duo.uio.no/handle/10852/44808>
- Diaz, F. J., Banchoff, C. M. T., & Soria, V. (2012). Usando Jmeter para pruebas de rendimiento. (January), 1–14.