

UNIVERSIDAD NACIONAL DE CHIMBORAZO



FACULTAD DE INGENIERIA CARRERA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

Proyecto de Investigación previo a la obtención del título de ingeniero en Sistemas y Computación.

TRABAJO DE TITULACIÓN

ANÁLISIS DEL DESEMPEÑO ENTRE MONGODB Y COUCHDB UTILIZANDO NORMA ISO/IEC 25000. CASO PRÁCTICO: APLICACIÓN WEB DE GESTIÓN DE DOCUMENTOS CONAGOPARE CHIMBORAZO.

AUTORES:

Pablo Ronny Calapucha Grefa
Marco Vinicio Tarco Caisa

TUTOR:

PhD. Miryan Estela Narváez Vilema

Riobamba - Ecuador

2019

VEREDICTO DE LA INVESTIGACIÓN

Los miembros del Tribunal de Graduación del proyecto de investigación de título: “ANÁLISIS DEL DESEMPEÑO ENTRE MONGODB Y COUCHDB UTILIZANDO NORMA ISO/ICE 25000, CASO PRÁCTICO: APLICACIÓN WEB DE GESTIÓN DE DOCUMENTOS CONAGOPARE CHIMBORAZO”, presentado por: Pablo Ronny Calapucha Grefa y Marco Vinicio Tarco Caisa, dirigida por: PhD. Miryan Estela Narváez Vilema.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso de custodia en la biblioteca de la facultad de Ingeniería de la UNACH.

Para constancia lo expuesto firman:

PhD. Miryan Estela Narváez Vilema
Director de Proyecto


Firma

MsC. Cristian Hugo Morales Alárcon
Miembro de Tribunal


Firma

MsC. Ana Elizabeth Congacha Aushay
Miembro de Tribunal


Firma

INFORME TUTOR

En calidad de tutor del proyecto de investigación: “ANÁLISIS DEL DESEMPEÑO ENTRE MONGODB Y COUCHDB UTILIZANDO NORMA ISO/IEC 25000. CASO PRÁCTICO: APLICACIÓN WEB DE GESTIÓN DE DOCUMENTOS CONAGOPARE CHIMBORAZO”; luego de haber revisado el desarrollo de la investigación por el Sr. Pablo Ronny Calapucha Grefa y el Sr. Marco Vinicio Tarco Caisa, tengo a bien informar que el trabajo de investigación indicado, cumple los requisitos exigidos para que pueda ser expuesto al público, luego de ser evaluado por el tribunal designado. **Riobamba, 2019**



PhD. Miryan Estela Narváez Vilema

TUTOR

AUTORIA DE LA INVESTIGACION

“La responsabilidad del contenido de este Proyecto de Graduación corresponde exclusivamente a: Pablo Ronny Calapucha Grefa y Marco Vinicio Tarco Caisa con la dirección de la PhD. Miryan Estela Narváez Vilema y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo”



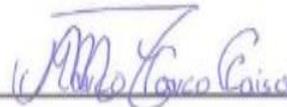
Myrian Estela Narváez Vilema
060357677-8

Tutor del Proyecto de Investigación



Pablo Ronny Calapucha Grefa
150071344-9

Autor del Proyecto de Investigación



Marco Vinicio Tarco Caisa
172591127-3

Autor del Proyecto de Investigación

DEDICATORIA

Este trabajo dedicamos a nuestras familias quienes han sido un pilar fundamental en cada paso de nuestras vidas, gracias por todo el apoyo que nos han brindado, han sido nuestra inspiración para no darnos por vencidos, y poder llegar a este momento tan importante de nuestra vida profesional.

A la comunidad administrativa y académica de nuestra querida alma mater UNACH, pues es por ellos que la universidad está vigente hoy por hoy.

PABLO RONNY CALAPUCHA GREFA

MARCO VINICIO TARCO CAISA

AGRADECIMIENTO

Agradecemos infinitamente a nuestro Creador, por habernos brindado vida y prosperidad con nuestras familias, y por permitirnos llegar al final de nuestra carrera.

Agradezco a la Universidad Nacional de Chimborazo por abrirnos las puertas hacia el conocimiento y a los docentes por compartir sus experiencias y orientaciones con profesionalismo para nuestra formación profesional.

Agradecemos a nuestra querida tutora PhD. Estela Narváez por su amabilidad y tiempo que dedico en la elaboración de esta investigación de tesis.

PABLO RONNY CALAPUCHA GREFA
MARCO VINICIO TARCO CAISA

ÍNDICE

VEREDICTO DE LA INVESTIGACIÓN	II
INFORME TUTOR	III
AUTORÍA DE LA INVESTIGACIÓN.....	IV
DEDICATORIA	V
AGRADECIMIENTO	VI
RESUMEN	XII
ABSTRACT.....	XIII
INTRODUCCIÓN	1
CAPITULO I	3
1. PLANTEAMIENTO DEL PROBLEMA.....	3
1.1. Problema.....	3
1.2. Justificación.....	4
1.3. OBJETIVOS.....	5
1.3.1. Objetivo General	5
1.3.2. Objetivos Específicos	5
CAPITULO II.....	6
Antecedentes del Estudio.....	6
2. MARCO TEORICO	8
2.1. Base de datos NoSQL.....	8
2.2. Clasificación de las Bases de datos NoSQL.....	8
2.2.1. Clave-valor:	8
2.2.2. Columnas:.....	9
2.2.3. Grafos:	9
2.2.4. Documentos:.....	10
2.2.4.1. MongoDB	10
2.2.4.1.1. Arquitectura MongoDB	11
2.2.4.2. CouchDB.....	12
2.2.4.2.1. Arquitectura CouchDB	13
2.2.4.2.2. Operaciones CRUD en CouchDB.....	14
2.2.4.3. Características entre MongoDB y CouchDB.....	14
2.3. Norma ISO/IEC 25000	15
2.3.1. ISO/IEC 2500n – División de Gestión de Calidad.....	15
2.3.2. ISO/IEC 2501n – División de Modelo de Calidad.....	15
2.3.3. Norma ISO/IEC 25010	16
2.3.4. ISO/IEC 2502n – División de Medición de Calidad	16
2.3.5. ISO/IEC 2503n – División de Requisitos de Calidad	17
2.3.6. ISO/IEC 2504n - División de Evaluación de Calidad	18
2.4. Lenguajes de programación para la Web e IDE	19
2.4.1. PHP.....	19
2.4.2. Netbeans	19
2.4.3. Robo 3T.....	19
2.5. Servidor de Aplicaciones.....	19
2.5.1. Xampp	20
2.6. Software JMeter.....	20
2.7. ActiveXperts SMS Component	20
2.8. Metodología ágil XP.....	20
2.8.1. Fases de la metodología XP	20

CAPITULO III.....	22
3. METODOLOGÍA.....	22
3.1. Enfoque.....	22
3.1.1. Hipótesis.....	22
3.1.2. Identificación de variables.....	22
3.1.3. Operacionalización de Variables.....	23
3.2. Tipo y diseño de investigación.....	24
3.2.1. Investigación experimental.....	24
3.2.2. Investigación bibliográfica.....	24
3.3. Unidad de análisis.....	24
3.4. Población de estudio y tamaño de la muestra.....	25
3.5. Técnicas de recolección de datos.....	25
3.5.1. Observación.....	25
3.5.2. Entrevista.....	26
3.6. Técnicas de análisis e interpretación de la información.....	26
3.6.1. Herramientas utilizadas.....	26
3.6.2. Prueba U de Mann Whitney.....	27
3.6.3. Desarrollo de la aplicación web.....	27
CAPITULO IV.....	28
4. CRITERIOS DE DESEMPEÑO DE LAS BASES DE DATOS NOSQL.....	28
4.1. Metodología para determinar los criterios de desempeño norma ISO/IEC 25000.....	28
4.2. Ponderación de las características de calidad externas.....	30
4.3. Procedimiento de Ejecución de la matriz de calidad.....	30
CAPITULO V.....	32
5. RESULTADOS Y DISCUSIÓN.....	32
5.1. Material y condición con el cual se realizaron las pruebas de desempeño.....	32
5.1.1. Consideración adicional:.....	32
5.2. Resultados.....	33
5.2.1. Tiempo de Respuesta en Segundo.....	34
5.2.2. Uso de CPU.....	35
5.2.3. Uso de memoria RAM.....	35
5.2.4. Uso de disco.....	36
5.2.5. Eficiencia de desempeño.....	36
5.3. Comprobación de la hipótesis.....	37
5.3.1.1. Comprobación de hipótesis general.....	37
5.4. Discusión.....	38
CAPITULO VI.....	40
6. CONCLUSIONES Y RECOMENDACIONES.....	40
6.1. CONCLUSIONES.....	40
6.2. RECOMENDACIONES.....	41
BIBLIOGRAFÍA.....	42
ANEXOS.....	44
ANEXO I:.....	45
ANEXO II:.....	48
ANEXO III:.....	48
ANEXO IV:.....	49
ANEXO V:.....	57

ANEXO VI:	58
ANEXO VII:	59
ANEXO VIII:	67
ANEXO IX:	68
ANEXO X:	82
ANEXO XI:	83
ANEXO XII:	93

ÍNDICE DE TABLA

Tabla 1: Estructura clave-valor	8
Tabla 2: Estructura de datos	9
Tabla 3: Operaciones CRUD MongoDB	12
Tabla 4: Operaciones CRUD CouchDB	14
Tabla 5: Diferencia entre MongoDB y CouchDB	14
Tabla 6: Características de ISO/IEC 25010	16
Tabla 7: Hipótesis general	22
Tabla 8: Identificación de variables	23
Tabla 9: Unidad de análisis	24
Tabla 10: Grado de importancia y escala de métricas de Calidad	28
Tabla 11: Grado de importancia y escala de métricas de Calidad	28
Tabla 12: Grado de importancia y escala de métricas de Calidad	29
Tabla 13: Ponderación de Métricas de evaluación	30
Tabla 14: Aplicación de la matriz de calidad de MongoDB Norma ISO/IEC 25000	31
Tabla 15: Características del equipo	32
Tabla 16: Tiempo de respuesta y porcentajes de recursos entre MongoDB y CouchDB	33
Tabla 17: Prueba U de Mann-Whitney	37
Tabla 18: Medición de indicadores MongoDB	45
Tabla 19: Medición de indicadores CouchDB	46
Tabla 20: Comprobación de la normalidad	48
Tabla 21: Comprobación de cada indicador	48
Tabla 22: Especificación funcional 001	50
Tabla 23: Especificación funcional 001-1	51
Tabla 24: Especificación funcional 002	52
Tabla 25: Especificación funcional 002-1	53
Tabla 26: Especificación funcional 003	54
Tabla 27: Especificación funcional 004	55
Tabla 28: Login	93
Tabla 29: Registro de documentos	93
Tabla 30: Registro de envío de mensajes	94

ÍNDICE DE ILUSTRACIONES

Ilustración 1: Estructura de grafos	9
Ilustración 2: Estructura de documentos.....	10
Ilustración 3: Arquitectura funcional MongoDB	11
Ilustración 4: Arquitectura de MongoDB	11
Ilustración 5: Arquitectura funcional de CouchDB	12
Ilustración 6: Arquitectura de CouchDB	13
Ilustración 7: Características de ISO/IEC 25010	17
Ilustración 8: Evaluación de calidad del producto de software	18
Ilustración 9: Tiempo de respuesta y uso de recursos.....	33
Ilustración 10: Tiempo de respuesta MongoDB vs CouchDB.....	34
Ilustración 11: Tiempo de respuesta MongoDB vs CouchDB.....	34
Ilustración 12: Uso de CPU MongoDB vs CouchDB.....	35
Ilustración 13: Uso de memoria RAM MongoDB vs CouchDB	35
Ilustración 14: Uso de disco duro MongoDB vs CouchDB.....	36
Ilustración 15: Eficiencia de desempeño MongoDB vs CouchDB.....	37
Ilustración 16: Interfaz del Sistema Web.....	49
Ilustración 17: Interfaz del Sistema Web.....	50
Ilustración 18: Acta de Compromiso CONAGOPARE CHIMBORAZO	57
Ilustración 19: Modelado de la base de datos NoSQL.....	58
Ilustración 20: Archivo de la base de datos MongoDB	59
Ilustración 21: Dirección del db.....	59
Ilustración 22: Configuración avanzada del sistema	60
Ilustración 23: Configuración variables de entorno.....	60
Ilustración 24: Ruta de MongoDB ingresado en la Variable Path.....	61
Ilustración 25: Inicio del servidor mongod	61
Ilustración 26: Creación de la carpeta Log y Data.....	62
Ilustración 27: Inicio del servicio automático MongoDB.....	62
Ilustración 28: Conexión a la base de datos	63
Ilustración 29: Interfaz Robo 3T.....	63
Ilustración 30: Instalación del servidor Xampp	64
Ilustración 31: Complemento php_mongo.dll	64
Ilustración 32: Archivo php.ini	65
Ilustración 33: MongoDB en el servidor Xampp.....	65
Ilustración 34: Maximizar el tamaño de los archivos	65
Ilustración 35: Instalación del ACTIVEXPERTS SMS COMPONENT	66
Ilustración 36: Archivo php.ini	66
Ilustración 37: Configuración del php_com_dotnet	66
Ilustración 38: Creación de la Base de datos en Robo 3T	67
Ilustración 39: Creación de colecciones en Robo 3T.....	67
Ilustración 40: Estructura de la aplicación web	68
Ilustración 41: Conexión Base de datos.....	69
Ilustración 42: Interfaz de Login.....	69
Ilustración 43: Proceso de inicio de sesión	70
Ilustración 44: Contenido de la página principal	70
Ilustración 45: Interfaz de agregar usuarios	71
Ilustración 46: Contenido de la página principal	71

Ilustración 47: Interfaz de modificar usuario.....	72
Ilustración 48: Actualización del usuario	72
Ilustración 49: Interfaz de modificar usuario.....	74
Ilustración 50: Visualización de los usuarios.....	75
Ilustración 51: Insertar y actualizar tipos de usuarios y tipo de documento	76
Ilustración 52: Eliminar tipos de usuarios y tipo de documentos	77
Ilustración 53: Interfaz de insertar documento	77
Ilustración 54: Insertar documento	78
Ilustración 55: Interfaz de actualizar documento.....	79
Ilustración 56: Actualizar Documento.....	79
Ilustración 57: Interfaz de visualización documentos.....	81
Ilustración 58: Visualizar Documentos.....	81
Ilustración 59: Plantilla ActiveXperts SMS.....	82
Ilustración 60: Login.....	84
Ilustración 61: Menú	84
Ilustración 62: Menú administrador.....	85
Ilustración 63: Menú ejecutivo	86
Ilustración 64: Menú de usuarios	88
Ilustración 65: Menú de usuarios - Administrador	88
Ilustración 66: Ventana para agregar usuarios	89
Ilustración 67: Modificar datos del usuario	89
Ilustración 68: Modificar datos de usuario	90
Ilustración 69: Ventana para agregar documento	90
Ilustración 70: Visualizar documentos.....	91
Ilustración 71: Vista de documentos en espera.....	92
Ilustración 72: Formulario Login.....	93
Ilustración 73: Registro de documentos.....	94
Ilustración 74: Registro de Envió de Mensajes.....	94

RESUMEN

El objetivo principal de este proyecto de investigación fue analizar el desempeño entre las bases de datos NoSQL (MongoDB y CouchDB) utilizando la norma ISO/IEC 25010, con la cual se estableció los parámetros para evaluar la eficiencia de las bases de datos NoSQL determinando el punto base para implementar en el APLICATIVO WEB DE GESTIÓN DE DOCUMENTOS CONAGOPARE CHIMBORAZO, con el fin de ofrecer un aplicativo web de calidad. Luego de evaluar las dos bases de datos se obtuvo que MongoDB fue ligeramente superior a CouchDB. Este trabajo es de enfoque cuantitativo y a través de una investigación experimental proporcionan un entorno de trabajo para evaluar el rendimiento de las bases de datos NoSQL. Como resultado de la investigación se obtuvo que las bases de datos NoSQL tienen mejor rendimiento al momento de administrar grandes volúmenes de datos.

Anteriormente la información de los documentos en CONAGOPARE Chimborazo eran almacenados en carpetas que utilizaban espacios físicos y mediante esta investigación se determinó la base de datos NoSQL óptima para implantar en el aplicativo Web de GESTIÓN DOCUMENTAL CONAGOPARE CHIMBORAZO. Para el desarrollo del aplicativo web se empleó la metodología XP, por ser flexible a cambios constantes durante el desarrollo, con la retroalimentación del cliente y el desarrollador, utilizando como lenguaje de programación PHP, MongoDB como sistema de gestión de base de datos NoSQL y el servidor web XAMP. Se concluye en la importancia de implantar el aplicativo web como base para garantizar la calidad de gestión de documentos, agilizando los procesos de los documentos enviados por las juntas parroquiales a CONAGOPARE Chimborazo y así contribuya en la toma de decisiones, a nivel institucional.

Palabras Clave: NoSQL, MongoDB, CouchDB, ISO/IEC 25010, JMeter.

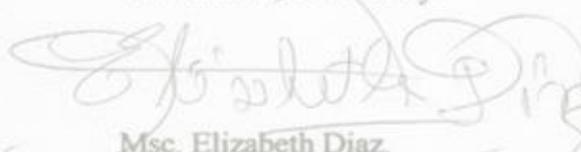
ABSTRACT

The main objective of this research project was to analyze the performance between the NoSQL (MongoDB and CouchDB) databases using the ISO/ IEC 25010 standard, through which the parameters to evaluate the efficiency of the NoSQL databases were established, determining the base point to implement in the CONAGOPARE CHIMBORAZO DOCUMENT MANAGEMENT WEB APPLICATION, in order to offer a quality web application. After evaluating the two databases, it was found that MongoDB was slightly higher than CouchDB. This work has a quantitative approach and through an experimental research provides a working environment to evaluate the performance of NoSQL databases. As a result of the research it was found that the NoSQL databases have better performance when managing large volumes of data.

Formerly, the information of the documents in CONAGOPARE Chimborazo were stored in folders that used physical spaces but through this investigation the optimal NoSQL database was determined to be implemented in the Web application of GESTION DOCUMENTAL CONAGOPARE CHIMBORAZO. For the development of the web application the XP methodology was used, because it seems to be flexible to constant changes during its development, with the client feedback and the developer, using a PHP programming as language, MongoDB as NoSQL database management system and the XAMP as web server. The importance of implementing the web application as a basis to guarantee the quality of document management it is concluded, streamlining the documents processes, sent by the parish councils to CONAGOPARE Chimborazo and thus contributes to decision-making, at the institutional level.

Keywords: NoSQL, MongoDB, CouchDB, ISO/IEC 25010, JMeter.

Translation reviewed by:


Msc. Elizabeth Diaz

INTRODUCCIÓN

Día tras día se recopila gran cantidad de datos a nivel mundial, los mismos que son almacenados en distintos gestores de base de datos; no obstante el crecimiento exponencial de información ha provocado que exista un rompimiento en el paradigma relacional de los gestores de base de datos, debido a que el modelo relacional empezó a presentar inconvenientes. (Castro, González, & Callejas, 2012).

La gestión de documentos ha evolucionado en las últimas décadas debido al auge de la tecnología, la documentación ha sido el mayor problema en las organizaciones públicas y privadas por la ausencia de espacio físico para conservar dichos documentos, las organizaciones optan en buscar soluciones para desmaterializar los documentos (Montoya & Carvalho, 2017)

Morejon (2018) en su estudio cualitativo de bases de datos NoSQL determina que existe una mejora notable al utilizarlas en sistemas con alta demanda de información. Existe una gran cantidad de bases de datos NoSQL todas trabajan sin necesidad de un lenguaje de consultas estructurado como: MongoDB y CouchDB. La base de datos MongoDB trabaja de forma nativa y permite el almacenamiento de datos en forma de documentos BSON, y CouchDB trabaja mediante http como interfaz y trabaja con documentos JSON. (García del Busto & Uanes Enríquez, 2012)

Esta investigación tiene como objetivo determinar cuál de las bases de datos no relacionales MongoDB y CouchDB utiliza menos recursos con respecto al factor de eficiencia bajo la Norma de calidad ISO/IEC 25000-25010. Se utilizó el software JMeter para obtener los criterios de: comportamiento de tiempos, utilización de recursos hardware (Procesador, Memoria RAM y Disco Duro), con los resultados que se obtuvo, se determinó que base de datos NoSQL (MongoDB y CouchDB) ocupa menos recursos para el sistema de **GESTIÓN DOCUMENTAL CONAGOPARE CHIMBORAZO**.

En el capítulo I, detalla el problema, la justificación, el objetivo general y los objetivos específicos de la investigación.

En el capítulo II, se presenta una descripción general del marco teórico relacionado con la temática de la investigación.

En el capítulo III, se describe la metodología aplicada durante el desarrollo de la investigación tomando en cuenta la documentación y la investigación analizada previamente.

En el capítulo IV, se establece los criterios de desempeño siguiendo el estándar de calidad de la norma ISO/IEC 25000.

En el capítulo V, se da a conocer los resultados y la discusión de la investigación.

En el capítulo VI, se establecen las conclusiones y recomendaciones del estudio.

CAPITULO I

1. PLANTEAMIENTO DEL PROBLEMA

1.1. Problema

El desarrollo de aplicaciones ha ido evolucionando constantemente, la aparición de diversos lenguajes de programación y sistemas de bases de datos libres y privados, han permitido desarrollar aplicaciones multiplataforma. Aun así en nuestro país la mayoría de empresas públicas y privadas mantienen una organización manual ocasionando que la gestión de documentos sea ineficiente, esto a su vez conlleva tiempo de búsqueda, uso de espacio físico y no permite realizar seguimiento de estos documentos; comprometiendo a las empresas a una escasa eficiencia en los procesos de resolución de documentos. (Font Aranda, Ruiz Rodríguez, & Mena Mugica, 2012).

CONAGOPARE CHIMBORAZO no dispone de un sistema de gestión de documentos, siendo esa la problemática que impide una rápida respuesta a los documentos que diariamente ingresan a la organización provenientes de alguna de las juntas parroquiales, como propuesta al problema, la investigación se orienta a realizar, un estudio de comparación de base de datos NoSQL (MongoDB y CouchDB), utilizando la Norma ISO/EC 25000-25010 y con la ayuda del software Apache JMeter se evalúa la eficiencia de desempeño con las siguientes características: comportamiento temporal y utilización de recurso. El aplicativo web GESTIÓN DOCUMENTAL CONAGOPARE CHIMBORAZO permite el seguimiento de los documentos que ingresan a secretaria hasta el proceso de resolución del documento, notificando al solicitante mediante mensajería de texto que el documento fue gestionado, el sistema tiene la capacidad de respaldar la documentación y almacenar digitalmente para cualquier proceso de auditoría de CONAGOPARE CHIMBORAZO.

1.2. Justificación

La gestión documental en las organizaciones públicas y privadas de Riobamba no responde de forma adecuada a las necesidades que se presentan según su actividad, debido a que gran parte de organizaciones realizan los procesos de forma manual, por lo tanto es importante plantearse una nueva política que permita la creación de aplicaciones que automaticen los procesos, para el almacenamiento y la manipulación de la información que permitirán ayudar a cumplir los objetivos de las empresas. (Russo, 2009)

Existe un sin número gestores de base de datos relacionales y no relacionales que son utilizados indistintamente para almacenar grandes cantidades de datos para luego ser transformados en información indispensable de la institución, en la actualidad el modelo no relacional de base de datos esta predominando ya que permite un alto grado de escalabilidad a diferencia de las relacionales y con menor latencia en cuanto al acceso hacia los datos. (Silicon, 2016)

La gestión de documentos asegura la eficiencia y el rápido acceso hacia ellos, permitiendo también realizar un seguimiento de los procesos por los cuales debe pasar, CONAGOPARE CHIMBORAZO no dispone de un aplicativo que permita gestionar sus documentos por lo que la implementación del sistema GESTIÓN DOCUMENTAL CONAGOPARE CHIMBORAZO brinda respuestas eficientes, y tiene capacidad de almacenar los documentos evitando así el almacenamiento físico, todos los documentos serán digitalizados y almacenados en un repositorio local.

1.3. OBJETIVOS

1.3.1. Objetivo General

- Analizar el desempeño entre MongoDB y CouchDB, utilizando NORMA ISO/IEC 25000/25010 para la implementación en la aplicación web de gestión de documentos CONAGOPARE CHIMBORAZO.

1.3.2. Objetivos Específicos

- Determinar los criterios de eficiencia de desempeño establecidos en la NORMA ISO/IEC 25010 para identificar los indicadores a ser evaluados.
- Evaluar los criterios de eficiencia de desempeño de la base de datos no relacionales MongoDB y CouchDB mediante el software JMeter para realizar el comparativo de desempeño.
- Desarrollar un aplicativo web de gestión de documentos para CONAGOPARE CHIMBORAZO utilizando la base de datos no relacional que tenga mejor eficiencia de desempeño.

CAPITULO II

Antecedentes del Estudio

En la actualidad las bases de datos NoSQL representan un cambio de los tradicionales sistemas de gestión de bases de datos relacionales, abriendo un amplio campo de estudio a los investigadores para realizar un análisis detallado sobre las bases de datos no relacionales, encontrando así diferentes estudios realizados por varios autores. En la Universidad Politécnica de Valencia de España, Bas (2015), realizó un estudio comparativo de base de datos relacionales y no relacionales con el objetivo de determinar si existe una mejora al utilizar bases de datos NoSQL en el sistema de minería de datos, donde se recolecta información de distintos indicadores, el sistema de nombre Regedate utilizaba la tecnología clásica mediante bases de datos relacionales, la investigación buscaba la forma de incrementar el rendimiento sin la necesidad de invertir en hardware de mejor prestaciones, después de realizar el análisis respectivo se determinó que existía carencias en las bases de datos relacionales debido a que requieren más tiempo a la hora de procesar gran cantidad de datos, a diferencia de las bases de datos NoSQL que fueron eficientes y pueden trabajar incluso con más sobrecarga de datos y tiene relación con lo que sostiene Córdova & Cuzco (2013) que ratifican que ese tipo de base de datos son capaces de procesar un volumen grande de datos en menor tiempo a diferencia de las bases de datos relacionales. Gaona & Saldoval (2013) realizaron un estudio comparativo de modelo de bases de datos relacionales y no relacionales y su incidencia de los procesos en sistemas informáticos, determinando que las bases no relacionales se desenvuelven mejor al momento de procesar gran cantidad de datos. Esos estudios previos orientan esta investigación porque hace énfasis en que las bases de datos no relacionales brindan un mejor tiempo de respuesta al implementarse en los sistemas informáticos de hoy en día, sirvieron como punto base para realizar un estudio exclusivo de bases de datos no relacionales.

En cuanto a los estudios realizados de las bases de datos no relacionales Morejón (2018) presenta un estudio cualitativo para la Universidad Técnica de Ambato con el objetivo de determinar cómo la información cargada en base de datos NoSQL incide en la generación de documentos de la secretaría del Honorable Consejo Universitario, los resultados en base a las encuestas que se realizó en esa investigación arrojan que el 100% de personal considera almacenar los datos en un gestor de base de datos NoSQL. En la Univesidad de Bergensis,

Bhamra (2017) presento su estudio denominado A Comparative Analysis Of MongoDB and Cassandra, con el fin de determinar cuál presenta mejores características con respecto al enfoque cualitativo de cada una de ellas, dando como resultado que MongoDB es mejor pues existe soporte y guías de manejo. Una investigación profundizada presenta Niteshwar (2016), donde se compara el comportamiento de dos bases de datos NoSQL MongoDB y CouchDB, teniendo como resultado que MongoDB presenta 58.4% rendimiento, frente a CouchDB que tiene 9.8%, lo que da a conocer que MongoDB realiza las operaciones más rápido que CouchDB. Estas investigaciones incitan a profundizar y abordar otros aspectos que permitan identificar de forma precisa la base de datos NoSQL que tengan mejores prestaciones a la hora de procesar datos, pues en la mayoría de los casos de investigación antes mencionados se consideró el aspecto cualitativo y es por ello que es viable realizar una investigación con enfoque cuantitativo que considere implementar la norma ISO/IEC 25010 lo que permitió evaluar la calidad del software.

2. MARCO TEORICO

2.1. Base de datos NoSQL

La aparición de las bases de datos NoSQL inicia con compañías de Internet reconocidas a nivel mundial en redes sociales como Facebook, Google, Twitter y Amazon. Las bases de datos NoSQL se caracterizan principalmente por no utilizar SQL como lenguaje principal de consultas, el esquema y los datos no necesariamente deben estar estructurados como en las bases de datos relacionales. (Castro, González, & Callejas, 2012)

2.2. Clasificación de las Bases de datos NoSQL

Las bases de datos NoSQL se clasifican según la forma de almacenar los datos:

2.2.1. Clave-valor:

El almacenamiento de la información de estas bases de datos se lo realiza con una sola clave y un valor Tabla 1, lo que permite realizar búsquedas de información eficaz, las bases de datos que utilizan este tipo de almacenamiento son: Redis, Dynamite y Voldemort DB. (Castro, González, & Callejas, 2012)

Ejemplo:

Tabla 1: Estructura clave-valor

Clave	Valor
1245322	Nombre: Pablo Calapucha Edad: 24 Semestre: 10
535352	Nombre: Marco Tarco Edad: 23 Semestre: 10

Realizado por: Los autores

2.2.2. Columnas:

La información ingresada se almacena en columnas, manteniendo inicialmente la clave y a su vez se agrupan valores para un mismo atributo en filas Tabla 2. Las bases de datos que utilizan este tipo de almacenamiento son: Apache Hbase, Cassandra y Hypertable. (Castro, González, & Callejas, 2012)

Ejemplo:

Tabla 2: Estructura de datos

Id	1245322	535352
Nombres	Pablo Calapucha	Marco Tarco
Edades	24	23
Semestre	10	10

Realizado por: Los autores

2.2.3. Grafos:

El almacenamiento se lo realiza por grafos que contiene nodos y estos nodos contienen entidades, que se relacionan por aristas entre nodos Ilustración 1. Permitiendo una rápida búsqueda de información por medio de su estructura normalizada. Las bases de datos que utilizan el tipo de almacenamiento son: Neo4j, InfoGrid y Virtuoso. (Castro, González, & Callejas, 2012)

Ejemplo:



2.2.4. Documentos:

Almacena datos semiestructurados como documentos se los agrupa en colecciones, estas tienen como formato JSON, BSON, ó XML donde se utiliza una clave única por cada registro, al encapsular llave y valor en estructura de documentos Ilustración 2. Las bases de datos que utilizan este tipo de almacenamiento son: MongoDB y CouchDB. (Castro, González, & Callejas, 2012)

Ejemplo:

Ilustración 2: Estructura de documentos

```
{
  "_id": 1232,
  "Nombres": "Pablo Calapucha",
  "Edades": 24,
  "Semestre": 10,
}
```

JSON

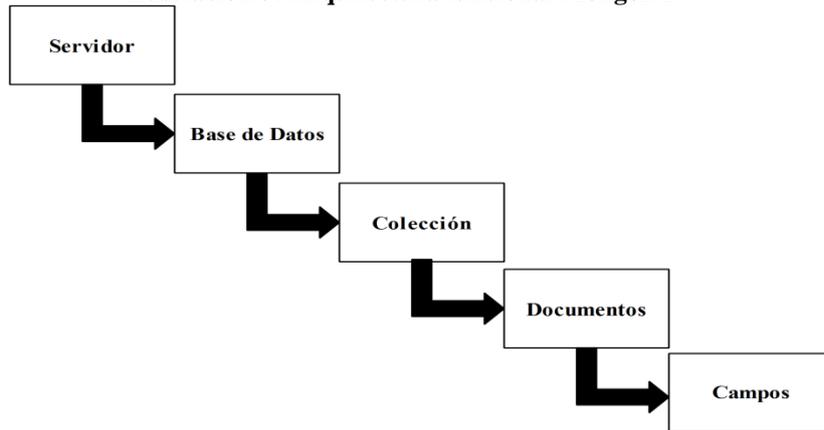
Realizado por: Los autores

Las bases de datos que predominan dentro de la familia NoSQL son MongoDB, CouchDB, Cassandra, HBase, entre otros. BD-Engines (2019) posiciona a MongoDB en el quinto puesto mientras que CouchDB ocupa el trigésimo segundo puesto, sin embargo son las dos bases de datos quienes han sido pioneros, por lo tanto son populares y más utilizadas.

2.2.4.1. MongoDB

Es una base de dato NoSQL de código abierto licencia GNU AGPL, está desarrollado en el lenguaje de C++. Los datos ingresados en MongoDB se guardan en documentos tipo JSON con un esquema dinámico como BSON compuestos de: nombre del campo, el tipo de dato y el valor, cada registro se denomina como documento que puede ser agrupado en colecciones Ilustración 3 que pueden ser similares a una base de datos relacionales, el programador tiene facilidad de adecuarlo según las necesidades. (Moreno, Quintero, & Vásquez, 2016)

Ilustración 3: Arquitectura funcional MongoDB



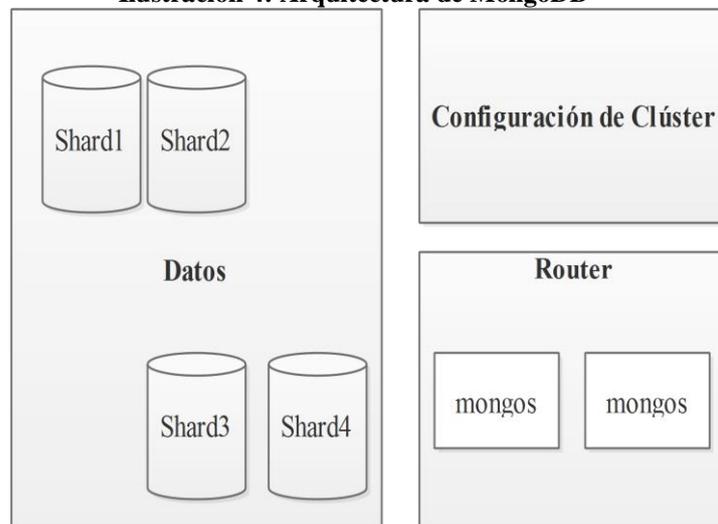
Realizado por: Los autores

2.2.4.1.1. Arquitectura MongoDB

Se basa en tres procesos:

- **Sharding:** Almacenamiento distribuido de los datos (Colecciones).
- **Mongos:** Envía las peticiones hacia el dato correcto Shard.
- **Configuración de Servidor:** Tiene toda la configuración y hace seguimiento al estado del Clúster. (Moreno, Quintero, & Vásquez, 2016)

Ilustración 4: Arquitectura de MongoDB



Realizado por: Los autores

2.2.4.1.2. Operaciones CRUD en MongoDB

La tabla 3 presenta las operaciones CRUD de MongoDB:

Tabla 3: Operaciones CRUD MongoDB

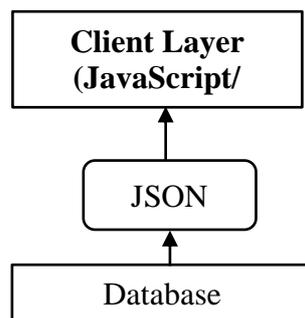
CRUD	Sintaxis	Definición
Crear	<code>db.usuarios.insert({nombre:"Pablo Calapucha", Edad: 24, Semestre: 10})</code>	Agrega un arreglo dentro de una colección.
Leer	<code>db.usuarios.findOne({"Nombre":"Pablo Calapucha"})</code>	Consulta información almacenada en la base de datos.
Actualizar	<code>db.usuarios.update({_id:ObjectId("S1e6681a2b7e6da6")}, { \$set:{Genero: "Masculino"} })</code>	Actualiza o añade a un campo existente.
Eliminar	<code>db.usuarios.remove({Edad:{\$gt:24}})</code>	Remueve un documento definido por un parámetro específico.

Realizado por: Los autores

2.2.4.2. CouchDB

Emplea JSON para almacenar los datos, utiliza JavaScript como lenguaje de consulta y HTTP como interfaz de programación principal. Los recursos que contiene la base de datos tienen una representación de una URL única que expone servicios a través de los métodos que implementa HTTP (Post, Get, Put y Deleted). (Chacaltana, 2015)

Ilustración 5: Arquitectura funcional de CouchDB



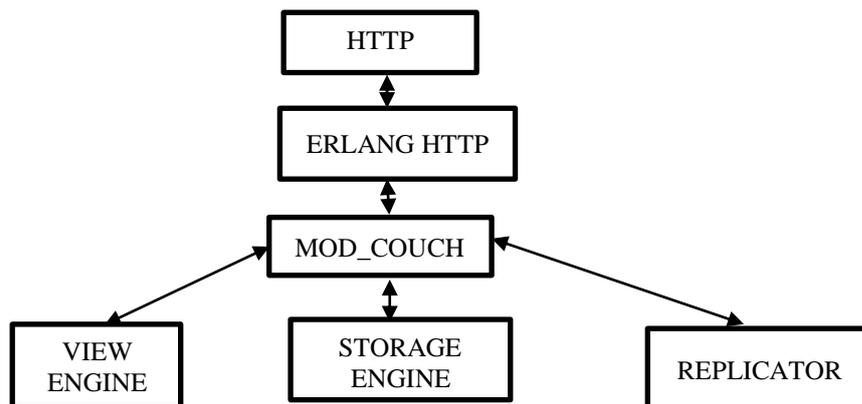
Realizado por: Los autores

2.2.4.2.1. Arquitectura CouchDB

Se basa en seis procesos:

- **HTTP Client:** Realiza peticiones a la base de datos, se puede utilizar cualquier aplicación que tenga el servicio HTTP. (Chacaltana, 2015)
- **Erlang HTTP:** Controla y gestiona el HTTP como: concurrencias, peticiones, autenticación entre otros.
- **Mod_Couch:** Encargado de gestionar y entender las peticiones solicitadas por el cliente HTTP.
- **View_Engine:** Realiza el proceso de la vista, el diseño de la documentación.
- **Storage_Engine:** Se comunica con el *View_Engine* y el replicador.
- **Replicador:** Permite la replicación de los documentos y la distribución entre diferentes dispositivos.

Ilustración 6: Arquitectura de CouchDB



Realizado por: Los autores

2.2.4.2. Operaciones CRUD en CouchDB

La tabla 4 presenta las operaciones CRUD de CouchDB:

Tabla 4: Operaciones CRUD CouchDB

CRUD	Sintaxis	Definición
Crear	<code>curl -X PUT http://127.0.0.1:5984/supermercado/ UUID -d '{"nombre": "Manzana", "tipo": "fruta", "precio": 5}'</code>	Insertar un documento con sus atributos en formato JSON.
Leer	<code>curl -X GET http://127.0.0.1:5984/supermercado/UUID</code>	Permite leer un documento mediante su ID.
Actualizar	<code>curl -X PUT http://127.0.0.1:5984/supermercado/98c003b03bc8aa87cb05983d1c000713 -d '{"_rev": "1-eba25568090eb2dfffad770b55147a67", "nombre": "Manzana", "tipo": "fruta", "precio": 4}'</code>	Permite reescribir un documento.
Eliminar	<code>Curl -X DELETE http://127.0.0.1:5984/supermercado/98c003b03bc8aa87cb05983d1c000713?rev=2-298fdb46385be60609b242b3e5cc3566</code>	Permite borrar un documento mediante su ID.

Realizado por: Los autores

2.2.4.3. Características entre MongoDB y CouchDB

La diferencia entre ambas bases de datos se explica en la tabla 5.

Tabla 5: Diferencia entre MongoDB y CouchDB

Características	MongoDB	CouchDB
Desarrollador	Desarrollado por 10gen.	Es un producto de Apache.
Idioma	Escrito en C++.	Escrito en Erlang.
Formato de datos	Modelo orientado a documentos, pero los datos se presentan en formato BSON.	Modelo orientado a documentos y los datos se presentan en formato JSON.
Interfaz	MongoDB usa el protocolo binario y TCP/IP.	CouchDB utiliza HTTP / REST-based interfaz. Es muy intuitivo y muy bien Diseñado.
Modelo de datos	La base de datos contiene colecciones y dentro documentos.	La base de datos contiene documentos.
Método de consulta	Sigue el método de consulta map-reduce y basado en objetos de lenguaje de consulta.	Sigue el método de consulta map-reduce
Preferencias	Favorece la consistencia	Favorece la disponibilidad

Realizado por: (Kumari, 2018)

2.3. Norma ISO/IEC 25000

Sirve para evaluar la calidad del software, cuya familia de normas es resultado de la evolución de algunas normas anteriores, especialmente de la norma ISO/IEC 9126, no obstante se tiene cuatro divisiones.

- ISO/IEC 2500n - División de Gestión de Calidad
- ISO/IEC 2501n - División de Modelo de Calidad
- ISO/IEC 2502n - División de Medición de Calidad
- ISO/IEC 2503n - División de Requisitos de Calidad
- ISO/IEC 2504n - División de Evaluación de Calidad

2.3.1. ISO/IEC 2500n – División de Gestión de Calidad

Este apartado de Normas define modelos, términos y definiciones para los demás estándares que los conforman que son:

- **ISO/IEC 25000 - Guía SQuaRE:** Establece la terminología y los modelos de referencia en la arquitectura SQuaRE.
- **ISO/IEC 25001 - Planificación y Gestión:** Establece los requisitos para gestionar en la evaluación y especificación de los requisitos del software.

2.3.2. ISO/IEC 2501n – División de Modelo de Calidad

Este apartado de normas presenta modelos de calidad donde interviene calidad interna, externa y en si el uso del producto de software, se encuentra conformada por:

- **ISO/IEC 25010:** Modelo de calidad para el producto software y la calidad en operación, presenta características y subcaracterísticas para evaluar el software. (ISO 25000, 2005)

2.3.3. Norma ISO/IEC 25010

Determina la calidad interna y externa de un producto de software, está compuesto por ocho características de calidad que se muestra en la tabla 6. (ISO 25000, 2005)

Tabla 6: Características de ISO/IEC 25010

Factor	Descripción	Criterios
Adecuación funcional	Capacidad del producto para proporcionar funciones que satisfacen los requerimientos declarados.	Complejidad Funcional Corrección Funcional Pertinencia Funcional
Eficiencia de Desempeño	Grado de utilización de recursos bajo determinadas condiciones.	Comportamiento Temporal Utilización de Recursos Capacidad
Compatibilidad	Capacidad de intercambio de información que comparte el mismo hardware.	Coexistencia Interoperabilidad
Usabilidad	Capacidad para que el producto sea entendible, aprendido, usado y resultar atractivo para el usuario.	Capacidad de aprendizaje y uso. Protección contra errores de usuario. Estética de interfaz de usuario. Accesibilidad.
Fiabilidad	Capacidad del sistema para realizar sus funciones cuando se use bajo condiciones y en un periodo determinado.	Madurez. Disponibilidad. Tolerancia a Fallos.
Seguridad	Capacidad para la protección de los datos.	Confidencialidad Integridad. Responsabilidad. Autenticidad.
Mantenibilidad	Capacidad del software para la modificación y evolución debido a las necesidades.	Modularidad. Reusabilidad.
Portabilidad	Capacidad del software para ser transferido de forma efectiva a otro entorno hardware y software.	Adaptabilidad. Capacidad para ser instalado. Capacidad para ser reemplazado.

Realizado por: (ISO 25000, 2005)

- **ISO/IEC 25012:** Modelo de calidad para los datos, precisamente para aquellos que están alojados de manera estructurada y que forman parte de un sistema. (ISO 25000, 2005)

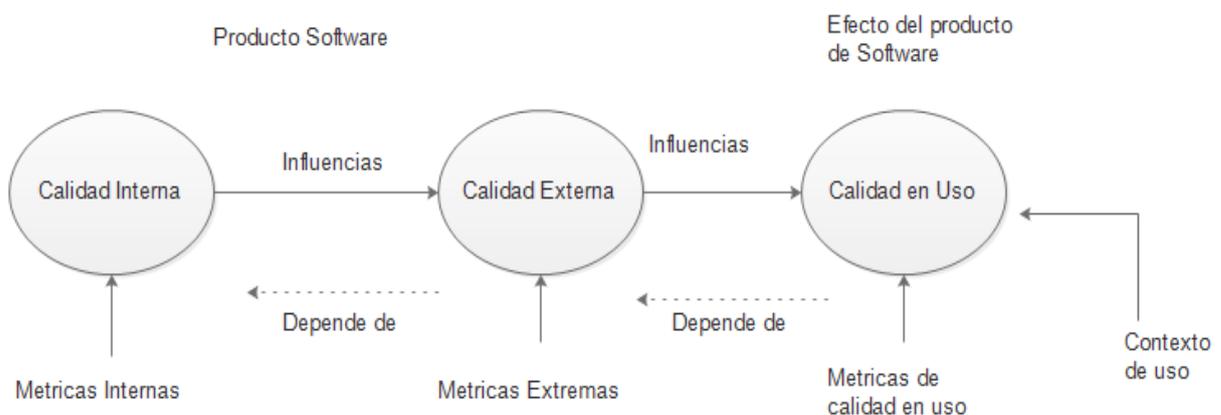
2.3.4. ISO/IEC 2502n – División de Medición de Calidad

En esta división de calidad, incluyen un modelo de referencia de calidad del software, con mediciones internas, externas y en uso. Los estándares son:

- **ISO/IEC 25020 - Guía y Modelo de Referencia para la medición:** Proporciona una explicación introductoria a los elementos de medida de la calidad, crea guías para los usuarios.

- **ISO/IEC 25021 – Elementos de Medida de Calidad:** Define y especifica métricas base y derivadas que puede ser utilizadas en el proceso de desarrollo del software.
- **ISO/IEC 25022 – Medición de Calidad en Uso:** Especifica las métricas que se utilizaran para la medición en el uso del software.
- **ISO/IEC 25023 – Medición de la Calidad del Producto Software y Sistemas:** Especifica la medición de la calidad del software.

Ilustración 7: Características de ISO/IEC 25010



Realizado por: (ISO 25000, 2005)

- **ISO/IEC 25024 – Medición de la Calidad de Datos:** Describe métricas para la medición de datos.

2.3.5. ISO/IEC 2503n – División de Requisitos de Calidad

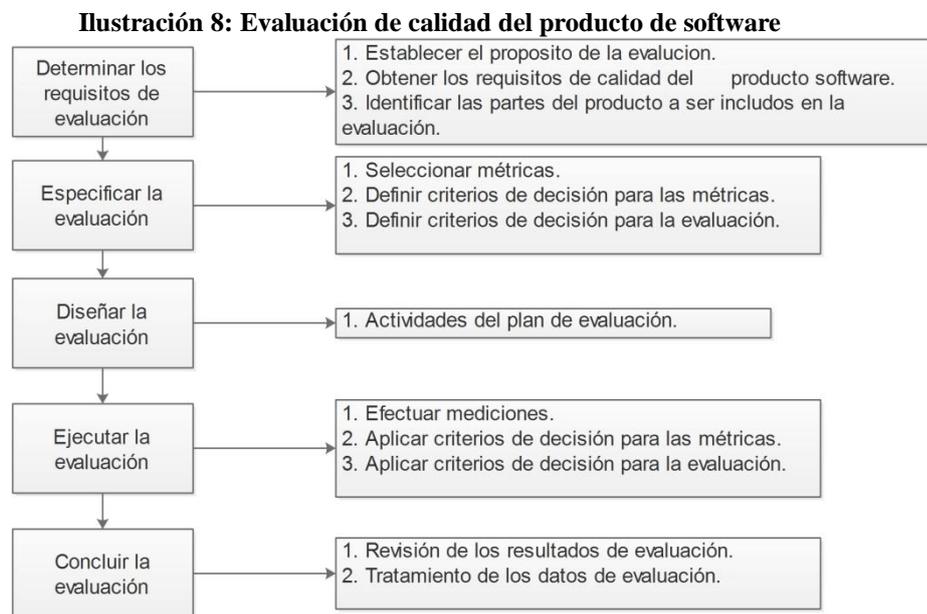
Especifica los requisitos de calidad que pueden ser utilizados en la entrada en el proceso de evaluación.

- **ISO/IEC 25030 – Requisitos de Calidad:** Este estándar permite a especificar los requisitos de calidad del software

2.3.6. ISO/IEC 2504n - División de Evaluación de Calidad

Las normas que pertenecen a este apartado son las que aportan con requisitos, recomendaciones y guías para llevar a cabo el proceso de evaluación del producto software, a continuación se presentan las divisiones:

- **ISO/IEC 25040(2010) – Guía y Modelo de Referencia de la Evaluación:** Establece la referencia para la evaluación, que tiene en cuenta las entradas, restricciones y recursos en el proceso de evaluación, para obtener las salidas.



Realizado por: (ISO 25000, 2005)

- **ISO/IEC 25041(2012) – Guía de Evaluación para Desarrolladores y Evaluadores Independientes:** Pondera requisitos y recomendaciones para la práctica de evaluación del producto software desde el punto de vista tanto del desarrollador, adquirentes y de los evaluadores independientes.
- **ISO/IEC 25042 – Módulos de Evaluación:** Consiste en establecer los módulos de evaluación y documentación.
- **ISO/IEC 25045 – Módulos de Evaluación para Recuperabilidad:** Define un módulo para la evaluación de los sistemas que ejecutan transacciones concurrentes.

2.4. Lenguajes de programación para la Web e IDE

Los lenguajes de programación han ido surgiendo según las necesidades y tendencias de las plataformas. Con el pasar de los años el desarrollo de páginas web estáticas fueron evolucionando, permitiendo a los usuarios no solo mostrar información sino manipular páginas dinámicas mejorando la experiencia en la navegación con la utilización de bases de datos. (Gutierrez, 2012)

2.4.1. PHP

Lenguaje que permite crear webs dinámicas que actúa en el lado del servidor, soporta la orientación a objetos y es un lenguaje multiplataforma, puede conectarse a una gran cantidad de bases de datos, relacionales y no relacionales. Su código fuente no es visible en los navegadores permitiendo mayor seguridad al cliente y confiabilidad. (Cecilia & Leite, 2015)

2.4.2. Netbeans

Es un entorno de desarrollo libre que permite a los programadores escribir, compilar, depurar y ejecutar programas. Para desarrollar un software puede utilizar diversos lenguajes de programación entre los más destacados esta: Java, PHP, C++, C# y Ruby. (NetBeans, 2018)

2.4.3. Robo 3T

Es un software de gestión de MongoDB en interfaz gráfica, de licencia gratuita (Robo Mongo, 2019)

2.5. Servidor de Aplicaciones

Es un ordenador físico o virtual al cual se puede acceder desde una red, trabaja con la aplicación cargada en su repositorio, el cual permite realizar las tareas que son solicitadas por los clientes. (Orti, 2015)

2.5.1. Xampp

Es un servidor virtual multiplataforma, bajo la licencia GNU el cual permite virtualizar un servidor Web que interpreta páginas estáticas o dinámicas. (Gómez, Candela, & Sepúlveda, 2013)

2.6. Software JMeter

Permite ejecutar pruebas de rendimiento y funcionalidad sobre aplicaciones de tipo cliente/servidor, JMeter es muy utilizado para realizar pruebas de conexión a bases de datos mediante: Servicios Web, JDBC, LDAP, HTTP. Siendo una de las características fundamentales, la capacidad de ingresar el número de usuarios o procesos concurrentes que se realizan en el tiempo de peticiones en segundos y el número de iteraciones, es decir el número de usuarios determinados. (JMeter, 2018)

2.7. ActiveXperts SMS Component

Es una API que proporciona una sencilla forma de enviar y recibir mensajes a través de un modem GSM, SMPP, HTTP y/o acceso telefónico, es compatible con los distintos lenguajes de programación: C++, .Net, PHP, JAVA, etc. (Active X perts Software BV, 2018)

2.8. Metodología ágil XP

Es una metodología ágil y flexible que está enfocado a la ingeniería de software cuyo objetivo es la gestión y desarrollo de proyectos con eficiencia, flexibilidad y control. La metodología ágil XP se centra en las relaciones interpersonales del grupo de trabajo y el cliente, dando un ambiente de trabajo cómodo. (Ávila & Meneses, 2013)

2.8.1. Fases de la metodología XP

Los procesos de la metodología XP son:

- **Planificación:** El cliente plantea los requerimientos necesarios para establecer una idea para la entrega del primer producto. Al mismo tiempo los

desarrolladores especifican las tecnologías, costo y fechas de entregas de las primeras versiones en el desarrollo del software.

- **Diseño:** El diseño de las interfaces debe ser simple y claro, para que el cliente pueda entender y adaptarse con facilidad.
- **Desarrollo:** Los desarrolladores establecen estándares de calidad, de manera que sean fácil entender por el equipo de desarrollo, en caso de recodificar el código.
- **Pruebas:** Todos los módulos son pasados por pruebas unitarias en base a los requisitos del cliente antes de ser la entrega final al cliente. (Ávila & Meneses, 2013)

CAPITULO III

3. METODOLOGÍA

3.1. Enfoque

La presente investigación se ajusta a un enfoque cuantitativo, porque se analiza los datos recopilados de las pruebas a cada base de datos NoSQL (MongoDB y CouchDB), conjuntamente con herramientas informáticas y estadística, para identificar cuál de las bases de datos es más eficiente y optimiza los recursos.

3.1.1. Hipótesis

Tabla 7: Hipótesis general

Identificador	Descripción	Simbología
H₀	No existe una diferencia en el análisis de desempeño entre bases de datos NoSQL MONGODB y COUCHDB.	H₀ : $M_x=M_y$
H₁	Existe una diferencia en el análisis de desempeño entre bases de datos NoSQL MONGODB y COUCHDB.	H₁ : $M_x \neq M_y$

Realizado por: Los autores

3.1.2. Identificación de variables

- **Variables Independientes**

Análisis del desempeño entre MongoDB y CouchDB.

- **Variables Dependientes**

Aplicación web de gestión de documentos.

3.1.3. Operacionalización de Variables

Tabla 8: Identificación de variables

Variable	Tipo	Definición Conceptual	Dimensión	Indicadores
Análisis de desempeño entre MongoDB y CouchDB.	Independiente	<p>MongoDB: Es una base de datos orientado a documentos, que almacena datos en documentos flexibles, similares a JSON. (mongodb, 2018)</p> <p>CouchDB: Emplea JSON para almacenar los datos tiene una representación de una URL única que expone servicios a través de los métodos que implementa HTTP. (Chacaltana, 2015)</p>	<ul style="list-style-type: none"> • Comportamiento de Tiempos. • Utilización de Recursos. 	<ul style="list-style-type: none"> • Promedio de tiempo de respuesta de las bases de datos NoSQL. • Promedio de utilización de CPU. • Promedio de utilización de RAM. • Promedio de utilización de Disco Duro.
Aplicación web de gestión de documentos.	Dependiente	Conexión de la base de datos NoSQL a la aplicación web gestión de documentos CONAGOPARE Chimborazo.	<ul style="list-style-type: none"> • Pertinencia Funcional 	<ul style="list-style-type: none"> • Operaciones de las funciones básicas CRUD.

Realizado por: Los autores

3.2. Tipo y diseño de investigación

3.2.1. Investigación experimental

El diseño usado es de tipo experimental porque se analiza el desempeño de las bases de datos NoSQL MongoDB y CouchDB, utilizando herramientas informáticas y estadísticas, para obtener los datos necesarios y medir la relación entre las variables y validar la hipótesis.

3.2.2. Investigación bibliográfica

Permite la recolección de datos científicos que se encuentran en documentos, libros y revistas científicas, referentes al tema de estudio, para sustentar el tema de investigación.

3.3. Unidad de análisis

Tabla 9: Unidad de análisis

Población	Muestra	Unidad de Análisis	Variabes
Bases de datos NoSQL(MongoDB y CouchDB)	Número de consultas	MongoDB y CouchDB	<ul style="list-style-type: none">• Tiempo de respuesta• Procesador(CPU)• RAM• Disco Duro

Realizado por: Los autores

3.4. Población de estudio y tamaño de la muestra

En la investigación se establece como población a las base de datos NoSQL MongoDB y CouchDB y la muestra el número de consultas que se realiza, se ve necesario el uso de la formula población infinita.

Identificadores:

n = Muestra

p = Probabilidad de éxito

q = Probabilidad de fracaso

Z = Nivel de confianza 95%

e = Error de muestra 5%

n = Muestra

Formula

$$n = \frac{Z^2 * p * q}{e^2}$$

$$n = \frac{(1.88)^2 * 0.5 * 0.5}{(0.05)^2} \cong 353.44 \cong 353$$

3.5. Técnicas de recolección de datos

Para la recopilación de la información se aplicó la técnica de observación.

3.5.1. Observación

Esta técnica ayuda inicialmente a identificar el problema de la institución, se identificó que no existía ningún mecanismo de seguimiento que garantice la resolución en el proceso de gestión de los documentos. Posteriormente se estableció una investigación comparativa utilizando la Norma ISO/IEC 25010 para medir el desempeño de base de datos NoSQL MongoDB y CouchDB, para identificar cual optimiza recursos. Para el desarrollo de la aplicación web se observa que la metodología XP garantiza que el desarrollador y cliente interactúen durante las fases de desarrollo.

3.5.2. Entrevista

Se utilizó la técnica de entrevista informal para establecer una relación directa con el representante de la organización, con el objetivo de determinar la situación actual de los procesos de gestión de los documentos, tras obtener la información se realizó una planificación para la elaboración del presente proyecto.

3.6. Técnicas de análisis e interpretación de la información

3.6.1. Herramientas utilizadas

- ISO/IEC 25010: Modelo de Calidad de Software.
- ISO/IEC 25023: Medición de la Calidad del Producto Software y Sistemas.
- ISO/IEC 25040: Guía y Modelo de Referencia de la Evaluación.
- Software Apache JMeter
- Software IBM SPSS 25
- Matriz de medición de Indicadores(formato Excel)
- Netbeans
- Xampp
- ActiveXperts SMS Component.

Se procedió a realizar el modelo de evaluación de las bases de datos NoSQL con la Norma ISO/IEC 25040, se identifica las métricas externas de calidad desempeño de la Norma ISO/IEC 25023 con un grado de importancia expresadas en la Tabla 11 que son: comportamiento temporal y utilización de recursos. Se realizó un plan de pruebas de los criterios a ser evaluados: Tiempo de Respuesta en segundos, Procesamiento CPU, Memoria RAM y Disco Duro en porcentaje, representados en el (ANEXO I). Al no saber la cantidad de sentencias que puede realizar el usuario se aplicó la fórmula de población infinita, determinando el tamaño de muestra adecuado para el sustento de la hipótesis, y con la ayuda del software estadístico *IBM SPSS 25* se efectuó el análisis *Mann Whitney*.

3.6.2. Prueba U de Mann Whitney

Se implementó el análisis de prueba U de Mann Whitney para comparar los criterios de las muestras de MongoDB y CouchDB que son dos muestras independientes con los resultados obtenidos se verifica, si la hipótesis es nula o alternativa. (Berlanga & Rubio, 2012)

3.6.3. Desarrollo de la aplicación web

Durante el desarrollo de la aplicación web y la implementación de la base de datos NoSQL de la gestión documental de CONAGOPARE Chimborazo, se aplicó la metodología XP con las siguientes fases:

- **Fase I planificación:** Se realizó una reunión con el cliente donde se identificó los aspectos principales para el funcionamiento del aplicativo (ANEXO IV), y se estableció el acta de compromiso CONAGOPARE CHIMBORAZO (ANEXO V).
- **Fase II diseño:** Se realizó el modelado base de datos NoSQL (ANEXO VI).
- **Fase III desarrollo:** Se realizó la instalación y configuración de las herramientas necesarias para el desarrollo del aplicativo web (ANEXO VII), mediante el software Robo3T se creó la base de datos (ANEXO VIII), se aplicó la estructura MVC (modelo, vista, controlador), utilizando el lenguaje de programación PHP, XAMP para desplegar el aplicativo (ANEXO IX), y el complemento ActiveXperts SMS Component para la mensajería de texto (ANEXO X).
- **Fase IV pruebas:** Con el manual guía de funcionamiento de la aplicación, se procedió a verificar el funcionamiento de los módulos de la aplicación web (ANEXO XI), y mediante la prueba de caja negra se pudo verificar el funcionamiento correcto de la aplicativo web (ANEXO XII).

CAPITULO IV

4. CRITERIOS DE DESEMPEÑO DE LAS BASES DE DATOS NOSQL

4.1. Metodología para determinar los criterios de desempeño norma ISO/IEC 25000

Para proceder en la evaluación del desempeño de las bases de datos NoSQL se estableció un rango en porcentaje impuesta por el evaluador, que aplica a los niveles de puntuación y grados de satisfacción propuestas para la Norma ISO/IEC 25040.

Tabla 10: Grado de importancia y escala de métricas de Calidad

Rango %	Niveles de puntuación	Grado de Satisfacción
80,0 - 10,0	Cumple con los requisitos	Muy satisfactorio
50,0 - 70,0	Aceptable	Satisfactorio
0 - 40,0	Inaceptable	Insatisfactorio

Realizado por: Los autores

Para medir el criterio de desempeño se debe establecer métricas de calidad del modelo externo ISO/IEC 25023. Serán definidas en la tabla 11, con el objetivo de identificar las características más relevantes que se analizaron y se ejecutaron en la evaluación.

Tabla 11: Grado de importancia y escala de métricas de Calidad

Escala	Significado	Grado de Importancia
8 - 10	Criterio relevante.	Alto
5 - 7	Criterio de calidad no indispensable.	Medio
1 - 4	Criterio de calidad no necesario.	Bajo

Realizado por: Los autores

La tabla 12 describe la calidad externa de la ISO/IEC 25010, se adjuntan las subcaracterísticas para establecer el grado de importancia, basada en la tabla 11.

Tabla 12: Grado de importancia y escala de métricas de Calidad

Características	Subcaracterísticas	Grado de Importancia
Adecuación funcional	Complejidad funcional	Bajo
	Corrección funcional	Bajo
	Pertinencia funcional	Medio
Eficiencia de Desempeño	Comportamiento temporal	Alto
	Utilización de Recursos	Alto
	Capacidad	Bajo
Compatibilidad	Coexistencia	Bajo
	Interoperabilidad	Bajo
Usabilidad	Capacidad para reconocer su adecuación	Bajo
	Capacidad de Aprendizaje	Bajo
	Capacidad para ser usado	Bajo
	Protección contra errores de usuario	Bajo
	Estética de la interfaz de usuario	Bajo
	Accesibilidad	Bajo
Fiabilidad	Madurez	Bajo
	Disponibilidad	Bajo
	Tolerancia a fallos	Bajo
	Capacidad de recuperación	Bajo
Seguridad	Confidencialidad	Bajo
	Integridad	Bajo
	No repudio	Bajo
	Responsabilidad	Bajo
	Autenticidad	Bajo
Mantenibilidad	Modularidad	Bajo
	Reusabilidad	Bajo
	Inaplicabilidad	Bajo
	Capacidad para ser modificado	Bajo
	Capacidad para ser probado	Bajo
Portabilidad	Adaptabilidad	Bajo
	Capacidad para ser instalado	Bajo
	Capacidad para ser reemplazado	Bajo

Realizado por: Los autores

4.2. Ponderación de las características de calidad externas

En la tabla 13 presenta, los valores a medir sujetos a los criterios del desarrollador.

Tabla 13: Ponderación de Métricas de evaluación

Características	Subcaracterísticas	Nivel de Importancia
Eficiencia de Desempeño	Comportamiento temporal	Alto
	Utilización de Recursos	Alto
Usabilidad	Capacidad para reconocer su adecuación	Medio
	Estética de la interfaz de usuario	Medio

Realizado por: Los autores

4.3. Procedimiento de Ejecución de la matriz de calidad

Se aplica una matriz de calidad, donde el evaluador realizara la evaluación de calidad de las bases de datos NoSQL.

- Se especifica el tipo de base de dato NoSQL a evaluarse (MongoDB y CouchDB)
- Definir las características de calidad externas a evaluarse y su nivel de importancia.
- Definir la ponderación de las características y subcaracterísticas externas definidos en el punto anterior.
- Se aplica la fórmula para medir la eficiencia de desempeño de la población de las bases de las bases de datos NoSQL:

$$n = \frac{Z^2 * p * q}{e^2}$$

- Se describe el proceso del plan de pruebas para la medición de los recursos y tiempo de respuesta, con una muestra de 353 sentencias en un tiempo estimado de 6 minutos.
- Ingresar los datos anteriores en la matriz de calidad que debe seguir la siguiente matriz.

Tabla 14: Aplicación de la matriz de calidad de MongoDB Norma ISO/IEC 25000

Software	Características	Subcaracterísticas	Calidad del SW	Métrica	Fórmula	Nivel de Importancia	Valor Obtenido	Parcial en porcentaje	Valor con respecto al 100%	Calidad del Sistema	Grado de Satisfacción
MongoDB	Eficiencia de Desempeño	Comportamiento temporal	Externa	Tiempo de Respuesta	$n = \frac{Z^2 * p * q}{e^2}$	Alto	0,0747s	0,02%	99,98%	94,36%	Muy satisfactorio
		Utilización de Recursos	Externa	Procesador CPU	$n = \frac{Z^2 * p * q}{e^2}$	Alto	5,07%	11,26%	88,74%		
				Memoria RAM		Alto	28,45%				
				Disco duro		Alto	0,26%				
CouchDB	Eficiencia de Desempeño	Comportamiento temporal	Externa	Tiempo de Respuesta	$n = \frac{Z^2 * p * q}{e^2}$	Alto	0,0075s	0,002%	99,99%	90,84%	Muy satisfactorio
		Utilización de Recursos	Externa	Procesador CPU	$n = \frac{Z^2 * p * q}{e^2}$	Alto	14,52%	18,30%	81,7%		
				Memoria RAM		Alto	40,20%				
				Disco duro		Alto	0,19%				
Aplicación web de gestión de documentos.	Adecuación Funcional	Pertinencia Funcional	Interno	Operaciones de las funciones básicas CRUD	N/A	Medio	N/A	N/A	N/A	N/A	N/A

Realizado por: Los autores

CAPITULO V

5. RESULTADOS Y DISCUSIÓN

5.1. Material y condición con el cual se realizaron las pruebas de desempeño

Para obtener los datos de análisis de desempeño de las bases de datos NoSQL MongoDB y CouchDB se utilizó un equipo portátil que tiene las siguientes características:

Tabla 15: Características del equipo

Característica	Descripción
CPU	Intel Core™ i5 de Tercera Generación
RAM	8 Gb
SO	Windows 8.1 Pro de 64 Bits

Realizado por: Los autores

5.1.1. Consideración adicional:

En el software Apache JMeter se configuro la prueba de carga de 353 sentencias procesadas en cada una de las bases de datos NoSQL (MongoDB y CouchDB) en un tiempo estimado de 6 minutos, se utilizó el complemento PerfMon (Server Performance Monitoring) que permitió medir los resultados de procesador, memoria RAM y Disco Duro en porcentaje, los resultados se exportaron en un archivo con extensión (.csv). Se analizó los datos previamente extraídos del software Apache JMeter utilizando el software estadístico IBM SPSS 25, estos resultados permitirán sustentar la hipótesis, sacar las conclusiones y recomendaciones de cada base de datos NoSQL (MongoDB y CouchDB).

5.2. Resultados

Una vez realizado el análisis de normalidad Anexo II a través del software estadístico IBM SPSS 25 se presenta los siguientes resultados obtenidos de las medias respectivo a los indicadores Anexo III e Ilustración 9, de un total de 353 sentencias en un tiempo de 6 minutos para cada base de datos NoSQL (MongoDB y CouchDB).

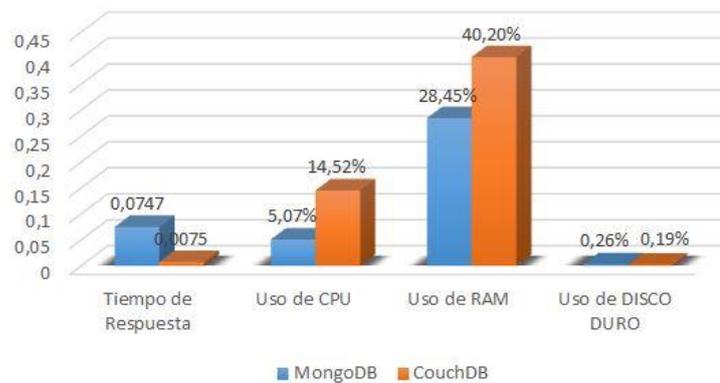
Tabla 16: Tiempo de respuesta y porcentajes de recursos entre MongoDB y CouchDB

Indicadores	MongoDB	CouchDB	Diferencia
Tiempo de Respuesta	0,074s	0,007s	0,067s
Uso de CPU	5,078%	14,527%	9,449%
Uso de RAM	28,452%	40,203%	11,751%
Uso de DISCO DURO	0,269%	0,198%	0,071%

Realizado por: Los autores

Ilustración 9: Tiempo de respuesta y uso de recursos

MongoDB vs CouchDB



Realizado por: Los autores

5.2.1. Tiempo de Respuesta en Segundo

La Ilustración 10, muestra los resultados de las bases de datos NoSQL CouchDB tiene una diferencia de 0,067s que permite afirmar que CouchDB realiza las transacciones en un menor tiempo, al realizar las 353 transacciones.

Ilustración 10: Tiempo de respuesta MongoDB vs CouchDB



Realizado por: Los autores

Para la medición del desempeño se realizó la conversión de del tiempo de respuesta de segundos a porcentaje aplicando la regla de tres con un resultado de 0,02% y CouchDB 0,003% y con respecto al 100%, lo que dio como resultado para MongoDB 99,98% y CouchDB 99,99%.

Ilustración 11: Tiempo de respuesta MongoDB vs CouchDB

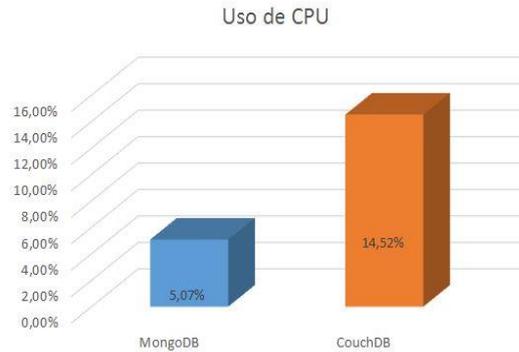


Realizado por: Los autores

5.2.2. Uso de CPU

En la Ilustración 12, se presenta los resultados de las bases de datos NoSQL donde MongoDB tiene una diferencia de 9,449%, por lo que se afirma que MongoDB ocupa menos recurso de procesador (CPU).

Ilustración 12: Uso de CPU MongoDB vs CouchDB

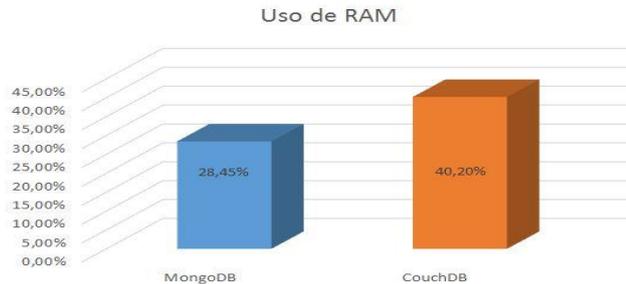


Realizado por: Los autores

5.2.3. Uso de memoria RAM

La Ilustración 13, muestra los resultados de la base de datos NoSQL MongoDB tiene una diferencia de 11,751%, por lo que se afirma que MongoDB ocupa menos recursos de memoria RAM.

Ilustración 13: Uso de memoria RAM MongoDB vs CouchDB



Realizado por: Los autores

5.2.4. Uso de disco

La Ilustración 14, muestra los resultados de la bases de datos NoSQL donde CouchDB tiene una diferencia 0,071%, por lo que se afirma que CouchDB ocupa menos recurso de disco duro.

Ilustración 14: Uso de disco duro MongoDB vs CouchDB

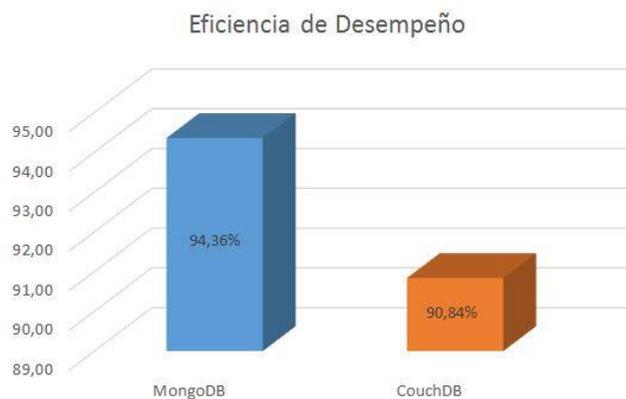


Realizado por: Los autores

5.2.5. Eficiencia de desempeño

La Ilustración 15 se presenta la suma de los porcentajes de recursos (CPU, Memoria RAM, y Disco Duro) de las bases de datos NoSQL MongoDB y CouchDB, el total de los recursos se divide por la cantidad de variables ($n = 3$) con el fin de obtener el porcentaje general, el resultado para MongoDB es de 11,266% y para CouchDB es de 18,309%; con respecto al 100%, MongoDB tiene un porcentaje de desempeño de un 88,74%, mientras que CouchDB con un 81,7%. Relacionando el porcentaje del tiempo de respuesta ilustración 10 y el porcentaje de los recursos se tiene a MongoDB con 94,36% y CouchDB con 90,84 concluyendo que MongoDB es eficiente al optimizar los recursos.

Ilustración 15: Eficiencia de desempeño MongoDB vs CouchDB



Realizado por: Los autores

5.3. Comprobación de la hipótesis

Tabla 17: Prueba U de Mann-Whitney

Hipótesis de prueba de hipótesis	Prueba	Significancia	Significancia establecida	Decisión
La distribución de Tiempo_Respuesta es la misma entre las categorías de NoSQL.	Prueba U de Mann-Whitney para muestras Independientes.	0,000	0,05	Rechazar la hipótesis nula.
La distribución de RAM es la misma entre las categorías de NoSQL.		0,000	0,05	Rechazar la hipótesis nula.
La distribución de CPU es la misma entre las categorías de NoSQL.		0,000	0,05	Rechazar la hipótesis nula.
La distribución de Disco Duro es la misma entre las categorías de NoSQL.		0,000	0,05	Rechazar la hipótesis nula.

Realizado por: Los autores

5.3.1.1. Comprobación de hipótesis general

Los resultados obtenidos en la prueba de la prueba U de Mann Whitney indican que no existe una diferencia significativa en los criterios analizados de ambas bases de datos NoSQL MongoDB y CouchDB, por lo tanto, al observar que el valor de significancia es menor al valor α (alfa) $\alpha=0.05\%$, se rechaza la hipótesis nula (H_0), y se acepta la hipótesis alternativa (H_1).

5.4. Discusión

En la presente tesis se realizó un análisis comparativo entre las bases de datos NoSQL MongoDB y CouchDB con el propósito de identificar cuál de las dos bases de datos antes mencionadas presenta un mejor desempeño a la hora de procesar los datos, de acuerdo con el análisis de los resultados obtenidos, se aceptó la hipótesis alternativa (H_1) que establece que existe una diferencia de desempeño entre las bases de datos NoSQL MongoDB y CouchDB, este proyecto de investigación tiene relación con lo que sostiene Salazar (2014) que afirma que las bases de datos NoSQL presentan un menor tiempo en cuanto a la respuesta de peticiones y menor uso de los recursos de hardware del computador, concuerda también con lo que sostiene Salazar & Germano (2017) que establecen que MongoDB da respuesta a las peticiones con menor uso de recursos, para lo cual se acude a los resultados obtenidos del criterio uso de CPU y memoria RAM de la presente investigación, con un porcentaje correspondiente de CPU MongoDB requiere 5,07% y CouchDB 14,52%, para procesar la misma cantidad de sentencias durante un tiempo establecido, en cambio en el uso de memoria RAM MongoDB requiere de 28,45% y CouchDB 40,20%, por ser los recursos que más inciden se puede ratificar que MongoDB es superior a CouchDB a la hora de procesar datos.

A diferencia de lo que sostiene Datt (2016), donde afirma que MongoDB presenta una tasa de respuesta de hasta 10 veces mejor que CouchDB, bajo las mismas condiciones en la investigación que se ha realizado para las dos bases de datos MongoDB y CouchDB, los resultados demuestran que en realidad no existe esa diferencia, pues el tiempo que requiere MongoDB es 0,074 segundos mientras que CouchDB 0,007, por consiguiente se tiene que CouchDB es ligeramente superior a MongoDB en cuanto al tiempo de respuesta se refiere. No obstante en la utilización de Disco Duro los resultados demuestran que CouchDB requiere 0,198% mientras que MongoDB 0,269% siendo claro que CouchDB presenta una leve ventaja. A diferencia de lo que Morejon (2018) da a conocer las cualidades de las bases de datos NoSQL, esta investigación tuvo un enfoque cuantitativo que permite conocer de forma exacta los indicadores de los criterios evaluados.

Por lo tanto, se puede concluir que a pesar de no existir una diferencia significativa en cada uno de los criterios evaluados, al momento de analizar los resultados se comprobó que existe al menos dos indicadores de suma importancia (CPU y Memoria RAM) que ayuda a ratificar que MongoDB es la mejor alternativa para implementarlo en sistemas que requieran alto nivel de concurrencia.

CAPITULO VI

6. CONCLUSIONES Y RECOMENDACIONES

6.1. CONCLUSIONES

- Se determinó las métricas de eficiencia de desempeño establecidos en la ISO/IEC 25010/25023 lo que permitió evaluar el desempeño de cada una de las bases de datos NoSQL (MongoDB y CouchDB).
- Los resultados obtenidos para la eficiencia de desempeño son: Tiempo de respuesta CouchDB con una diferencia de 0,06s menos que MongoDB; En procesador (CPU) MongoDB presenta una diferencia de 9,45% menos que CouchDB; En memoria RAM MongoDB presenta una diferencia de 11,75% menos que CouchDB; En Disco Duro CouchDB usa 0,07% menos que MongoDB.
- No existe una diferencia significativa de eficiencia de desempeño entre las bases de datos MongoDB y CouchDB, pero existe al menos un valor de criterio distinto entre las bases analizadas, por lo cual se rechaza la hipótesis nula (H_0), y se acepta la hipótesis alternativa (H_1) con un nivel de significancia menor a $\alpha=0,05\%$, en base a la Ilustración 15, MongoDB presenta una leve ventaja con respecto a CouchDB por lo que se consideró para implementarlo en el aplicativo Web de Gestión de Documentos CONAGOPARE CHIMBORAZO.
- En investigaciones previas ratifican que MongoDB es superior a CouchDB, y como resultado de nuestra investigación afirmamos que MongoDB presenta una leve ventaja de eficiencia de desempeño de 94,36% y CouchDB 90,84%.

6.2. RECOMENDACIONES

- Utilizar la Norma ISO/EC 25010, la misma que ayuda a evaluar de forma correcta el grado de desempeño de cualquier software, pues contiene muchas dimensiones a ser consideradas a la hora de implementar en los estudios de calidad de software y que se encuentra vigente en la actualidad.
- Usar Apache JMeter, puesto que ayuda obtener resultados de los recursos utilizados del software a ser estudiado, por ser un aplicativo compatible con la mayoría de aplicaciones que pueden ser analizadas, y con la libertad de ser de código abierto para poderlo utilizar según lo requiera.
- Según (Salazar & Germano, 2017) MongoDB realiza las operaciones solicitadas utilizando menos recurso, y que en esta investigación se pudo comprobar que en realidad si existe una leve ventaja con respecto al uso de los recursos por lo tanto MongoDB es la mejor opción para implementarlo en los sistemas informáticos.
- Tener pasión por aprender acerca de las bases de datos no relacionales, debido a que las bases de datos tradicionales en la actualidad ya no son tan efectivos debido a que no soportan una gran cantidad de datos, o una alta concurrencia.

BIBLIOGRAFÍA

- Active X perts Software BV. (2018). *Active X perts Software BV*. Obtenido de <https://www.activexperts.com/sms-component/>
- Ávila, E., & Meneses, A. (03 de 2013). Delfdroid y su comparación evaluativa con XP y Scrum mediante el método 4-DAT. *Revista Cubana de Ciencias Informaticas*, 07. Obtenido de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992013000100003&lang=pt
- Bas Abad, V. J. (2015). *Estudio comparativo de BBDD Relacionaes y NoSQL en un entorno industrial*. Valencia.
- BD-Engines. (2019). *BD-Engines*. Obtenido de <https://db-engines.com/en/ranking>
- Berlanga, V., & Rubio, M. (2012). Clasificación de pruebas no paramétricas. Como aplicarlas en SPSS. *ICE Universitat de Barcelona*, 104.
- Castro, A., González, J., & Callejas, M. (2012). Utilidad y Funcionamiento de las bases de datos NoSQL. *Facultad de Ingeniería UPTC*.
- Cecilia, T., & Leite, E. (2015). LAS TECNOLOGÍAS DE LA INFORMACIÓN Y LAS COMUNICACIONES COMO HERRAMIENTAS PARA LA GESTIÓN DEL PATRIMONIO CULTURAL CON UNA VISIÓN EMPRENDEDORA/THE INFORMATION TECHNOLOGY AND COMMUNICATIONS AS TOOLS FOR CULTURAL HERITAGE MANAGEMENT WITH AN ENTREPRENEURIAL VI. *Engineering--Civil Engineering*. Obtenido de <https://search.proquest.com/docview/1766025870/citation/F4D91763844C4AB6PQ/14?accountid=36757>
- Chacaltana, G. (2015). *Optimización de aplicaciones web con base de datos NoSQL In-Memory*. Obtenido de <http://normalizacion-bd.blogspot.com/2012/11/6-arquitectura-clienteservidor.html>
- Córdova Espinoza, R. F., & Cuzco Sarango, B. E. (2013). *Análsisi comparativo entre bases de datos relacionales con bases de datos no relacionaes*. Univerdisad Politécnica Salesiana Sede Cuenca, Cuenca.
- Datt, N. (2016). Comparative Study of CouchDB and MongoDB - NoSQL Document Oriented Database. *International Journal of Computer Application*, 136.
- Font Aranda, O., Ruiz Rodríguez, A., & Mena Mugica, M. (2012). Diagnóstico sobre la gestión documental y de archivos en la Universidad Central Marta Abreu de las Villas. Obtenido de http://diposit.ub.edu/dspace/bitstream/2445/117724/1/TFM_Perez_Pineda_Ivan.pdf
- García del Busto, H., & Uanes Enríquez, O. (2012). Base de Datos NoSQL. *Telematica*.
- Gómez, E., Candela, C., & Sepúlveda, L. (2013). Seguridad en la configuración del Servidor Web Apache. *INGE CUC*, 32-38.
- Gutierrez, R. (2012). El inicio de la Web: historia y cronología del hipertexto hasta HTML 4.0 (1990-99). *Revista de estudios sobre la ciencia y la tecnología*, 57-82. Obtenido de https://s3.amazonaws.com/academia.edu.documents/36585759/El_inicio_de_la_Web.pdf?AWSAccessKeyId=AKIAIWOWYYGZ2Y53UL3A&Expires=1538679804&Signature=jM3%2FKs4VS%2FUBu3RKWS0e5Tah6yk%3D&response-content-disposition=inline%3B%20filename%3DHow_the_Web_started_Hi
- ISO 25000. (2005). *ISO 25000*. Obtenido de <https://iso25000.com/index.php/normas-iso-25000>
- JMeter. (2018). *JMeter*. Obtenido de <https://jmeter.apache.org/>

- Kumar, S., & Sharma, D. (2015). Software Quality Appraisal Using Multi-Criteria. *Information Engineering and Electronic Business*.
- Kumari, S. (2018). IMPLEMENTACIÓN DEL PARADIGMA DE REDUCCIÓN DE MAP EN MONGODB Y COUCHDB. *Revista Internacional de Investigación Avanzada en Ciencias de la Computación*, 728.
- mongodb. (2018). *mongodb*. Obtenido de <https://www.mongodb.com/what-is-mongodb>
- Montoya, J., & Carvalho, T. (2017). LA WEB SEMÁNTICA Y SUS USOS EN LOS PROCESOS DE DOCUMENTACIÓN ARCHIVÍSTICA ORGANIZACIONAL. 438. Obtenido de https://www.researchgate.net/publication/322376789_La_web_semantica_y_sus_usos_en_los_procesos_de_documentacion_organizacional
- Morejon, M. D. (2018). *La información en bases de datos NOSQL y su incidencia en la generación documental de la secretaria general del honorable consejo universitario*. Ambato.
- Moreno, F., Quintero, J., & Vásquez, R. (15 de 03 de 2016). UNA COMPARACIÓN DE RENDIMIENTO ENTRE ORACLE Y MONGODB . 26-1, 110-112. Obtenido de <http://www.scielo.org.co/pdf/cein/v26n1/v26n1a07.pdf>
- NetBeans. (2018). *NetBeans*. Obtenido de <https://netbeans.org/>
- Orti, C. B. (2015). *Las tecnologías de la información y comunicación*. Valencia.
- Robo Mongo. (2019). *Robo Mongo*. Obtenido de <https://robomongo.org/>
- Rodríguez , A., Rodríguez , D., & Elizabeth , E. (2016). Selección de Base de Datos No SQL para almacenamiento de Históricos en Sistemas de Supervisión. *Revista Cubana de Ciencias Informáticas*, 161. Obtenido de http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S2227-18992016000300012
- Russo, P. (2009). *Gestión de Documentos en las Organizaciones*. Barcelona: UOC.
- Salazar , J. E. (2014). *Análisis comparativo entre dos bases de datos SQL y dos bases de datos NoSQL*. Pereira.
- Salazar, E., & Germano, K. (2017). Uma Comparação entre MongoDB E CouchDB. *IVSIMTEC*.
- Silicon. (2016). *Silicon*. Obtenido de https://www.silicon.es/bases-datos-no-relacionales-nosql-cuando-usarlas-2324948?inf_by=5c1fe0bf671db828348b4747

ANEXOS

ANEXO I:

Medición de Indicadores de las Bases de datos NoSQL (Mongodb y CouchDB)

Medición del tiempo de respuesta y medición de los recursos en un total de 353 sentencias, en la tabla 18 se muestra 50 datos y los restantes de las sentencias se encuentran en el CD.

Tabla 18: Medición de indicadores MongoDB

N	Base de Datos NoSQL	Tiempo de Resp	RAM %	CPU %	DISCO %
1	MongoDB	0,132	29,986	6,936	3
2	MongoDB	0,083	30,048	19,354	0
3	MongoDB	0,077	30,047	11,154	0
4	MongoDB	0,124	30,061	7,446	0
5	MongoDB	0,1	30,074	4,751	0
6	MongoDB	0,094	30,062	10,799	1
7	MongoDB	0,076	29,999	6,226	2
8	MongoDB	0,077	29,995	5,451	1
9	MongoDB	0,082	29,995	6,226	2
10	MongoDB	0,083	29,988	6,25	0
11	MongoDB	0,105	30,049	8,222	0
12	MongoDB	0,093	30,045	10,194	0
13	MongoDB	0,059	30,044	5,475	0
14	MongoDB	0,09	30,049	6,978	0
15	MongoDB	0,089	30,042	5,122	0
16	MongoDB	0,072	29,983	7,401	0
17	MongoDB	0,095	29,981	3,076	0
18	MongoDB	0,086	29,967	7,448	0
19	MongoDB	0,103	29,984	10,549	0
20	MongoDB	0,094	29,989	11,327	0
21	MongoDB	0,084	30,003	6,623	0
22	MongoDB	0,081	30,013	14,819	1
23	MongoDB	0,075	28,783	9,713	1
24	MongoDB	0,086	28,801	7,846	2
25	MongoDB	0,08	28,862	9,352	0
26	MongoDB	0,099	29,015	14,85	1
27	MongoDB	0,092	28,872	4,322	0
28	MongoDB	0,091	28,888	4,652	0
29	MongoDB	0,077	28,89	5,897	0
30	MongoDB	0,073	28,892	5,051	0
31	MongoDB	0,088	28,904	10,549	0
32	MongoDB	0,07	28,719	4,676	0
33	MongoDB	0,084	28,756	6,65	0
34	MongoDB	0,05	28,757	3,9	0

35	MongoDB	0,045	28,756	5,85	1
36	MongoDB	0,095	29,114	14,825	1
37	MongoDB	0,061	28,9	4,324	0
38	MongoDB	0,089	28,899	2,34	0
39	MongoDB	0,081	28,945	16,779	0
40	MongoDB	0,057	28,985	19,99	1
41	MongoDB	0,095	28,934	5,098	1
42	MongoDB	0,078	28,933	5,873	1
43	MongoDB	0,074	28,956	9,319	0
44	MongoDB	0,075	28,951	5,851	0
45	MongoDB	0,089	28,95	7,446	0
46	MongoDB	0,089	28,95	2,725	0
47	MongoDB	0,086	28,949	4,298	0
48	MongoDB	0,075	28,948	3,901	0
.....	MongoDB
353	MongoDB	0,081	27,965	2,326	0

Realizado por: Los autores.

Tabla 19: Medición de indicadores CouchDB

N	Base de Datos NoSQL	Tiempo de Resp	RAM %	CPU %	DISCO %
1	COUCHDB	0,009	39,339	16,366	0
2	COUCHDB	0,005	39,339	6,121	0
3	COUCHDB	0,009	39,345	9,104	0
4	COUCHDB	0,009	39,339	10,421	0
5	COUCHDB	0,008	39,34	8,864	0
6	COUCHDB	0,007	39,278	6,27	0
7	COUCHDB	0,003	39,272	5,427	0
8	COUCHDB	0,007	39,244	5,475	0
9	COUCHDB	0,011	39,242	9,445	0
10	COUCHDB	0,009	39,242	8,202	0
11	COUCHDB	0,009	39,242	3,87	0
12	COUCHDB	0,01	39,242	6,621	0
13	COUCHDB	0,01	39,23	5,499	1
14	COUCHDB	0,003	39,217	8,597	0
15	COUCHDB	0,007	39,193	7,833	0
16	COUCHDB	0,008	39,179	5,078	0
17	COUCHDB	0,007	39,166	5,873	0
18	COUCHDB	0,007	39,166	5,828	0
19	COUCHDB	0,007	39,154	7,427	0
20	COUCHDB	0,007	39,142	6,274	0
21	COUCHDB	0,007	39,133	3,876	0
22	COUCHDB	0,008	39,133	10,966	0

23	COUCHDB	0,009	39,132	9,736	0
24	COUCHDB	0,009	39,085	12,453	0
25	COUCHDB	0,01	39,068	4,639	0
26	COUCHDB	0,009	39,068	8,579	0
27	COUCHDB	0,009	39,062	7,421	0
28	COUCHDB	0,007	39,056	7,848	0
29	COUCHDB	0,008	39,056	3,487	1
30	COUCHDB	0,007	39,049	5,85	0
31	COUCHDB	0,008	39,049	6,228	0
32	COUCHDB	0,008	39,049	7,018	0
33	COUCHDB	0,007	39,049	5,499	0
34	COUCHDB	0,008	39,049	4,656	0
35	COUCHDB	0,008	39,044	8,202	0
36	COUCHDB	0,076	39,044	8,197	0
37	COUCHDB	0,008	39,04	6,248	0
38	COUCHDB	0,007	39,04	8,577	0
39	COUCHDB	0,008	39,034	5,475	0
40	COUCHDB	0,008	39,138	13,675	0
41	COUCHDB	0,007	39,402	21,06	0
42	COUCHDB	0,007	39,402	4,723	0
43	COUCHDB	0,007	39,4	4,274	0
44	COUCHDB	0,008	39,399	4,253	0
45	COUCHDB	0,008	39,399	6,246	0
46	COUCHDB	0,008	39,399	10,577	0
47	COUCHDB	0,006	39,394	8,22	0
48	COUCHDB	0,007	39,394	8,939	0
.....	COUCHDB
353	COUCHDB	0,008	41,195	11,259	0

Realizado por: Los autores.

ANEXO II:

Comprobación de la normalidad

Para comprobar la normalidad de los datos cuantitativos se verifico con **Kolmogorov-Smirnov^a** por que los datos son mayor 50 datos y la significancia para cada criterio es ,000 por lo tanto la muestra aplicada no tiene normalidad por ser menor al valor de la significancia $p < 0,005$.

Tabla 20: Comprobación de la normalidad

Criterios	Kolmogorov-Smirnov ^a			Shapiro-Wilk		
	Estadísticos	gl	Sig.	Estadísticos	gl	Sig.
Tiempo de Respuesta	,278	706	,000	,797	706	,000
RAM	,278	706	,000	,705	706	,000
CPU	,151	706	,000	,809	706	,000
Disco Duro	,509	706	,000	,323	706	,000

Realizado por: Los autores

ANEXO III:

Comprobación de cada indicador

En la tabla 20 se realiza estadística descriptiva de las base de datos MongoDB y CouchDB de los indicadores tiempo de respuesta, Ram, CPU, disco.

Tabla 21: Comprobación de cada indicador

NOSQL		Tiempo de Respuesta	RAM	CPU	DISCO
MongoDB	Media	,07466	28,45280	5,07875	,27
	N	353	353	353	353
	Desviación estándar	,012042	,487191	3,091538	,660
	Mediana	,07300	28,26000	4,30100	,00
CouchDB	Media	,00750	40,20300	14,52754	,02
	N	353	353	353	353
	Desviación estándar	,006958	,560822	7,879753	,140
	Mediana	,00700	40,07100	12,50000	,00

Realizado por: Los autores

ANEXO IV:

Planificación y funcionamiento del Software

- **Nombre Proyecto:** Análisis del desempeño entre MongoDB Y CouchDB, utilizando Norma ISO/IEC 25000. Caso Práctico: Aplicación web de gestión de documentos CONAGOPARE CHIMBORAZO.
- **Nombre del Producto:** GESTIÓN DOCUMENTAL CONAGOPARE CHIMBORAZO.
- **Objetivo:** Optimizar el proceso de entrega y recepción de documentos.
- **Proveedor:** Desarrollador Externo.
- **Logo del Producto:** Sello Oficial de CONAGOPARE
- **Interfaz Web:**

Ilustración 16: Interfaz del Sistema Web

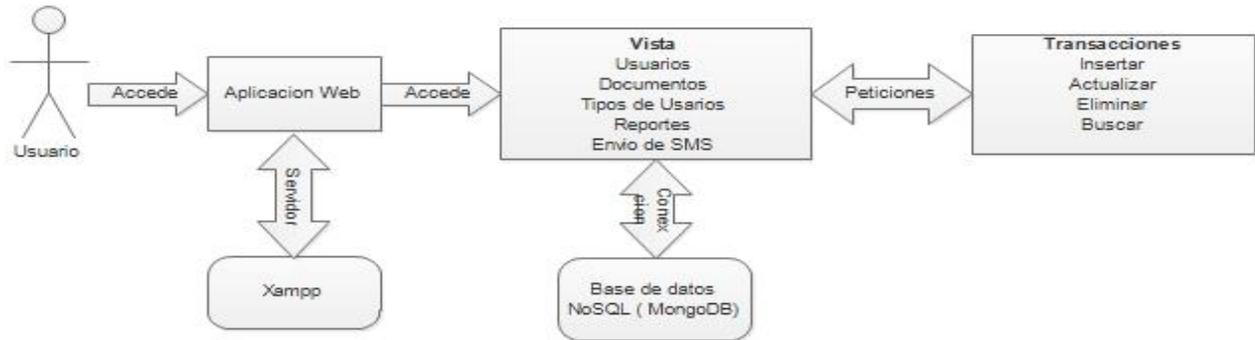


Realizado por: Los autores

- **Motor de Base de Datos NoSQL:** MongoDB
- **Lenguaje de Programación:** PHP
- **Servidor:** Xampp.
- **Herramientas Adicionales:** ActiveXperts SMS Component, Modem Huawei
- **Manual de Usuario y Técnico:** Se encuentra en el CD.
- **Metodología:** XP.

Funcionamiento de la Aplicación

Ilustración 17: Interfaz del Sistema Web



Realizado por: Los autores

Tabla 22: Especificación funcional 001

ESPECIFICACIÓN FUNCIONAL

ID Requerimiento	001	Título	Caso de Uso
------------------	-----	--------	-------------

Modelo de casos de uso

En la siguiente matriz se da a conocer los casos de uso y el actor involucrado para el SISTEMA DOCUMENTAL CONAGOPARE CHIMBORAZO.

Modelo de caso de uso

Descripción	Actor
Registro y modificación de usuarios.	Administrador
Registro de Documentos	Administrador & Ejecutivo
Visualizar Documentos	Administrador & Ejecutivo
Generar Reportes	Administrador & Ejecutivo
Generar Notificación SMS	Administrador & Ejecutivo
Recepción de notificación SMS	Usuario

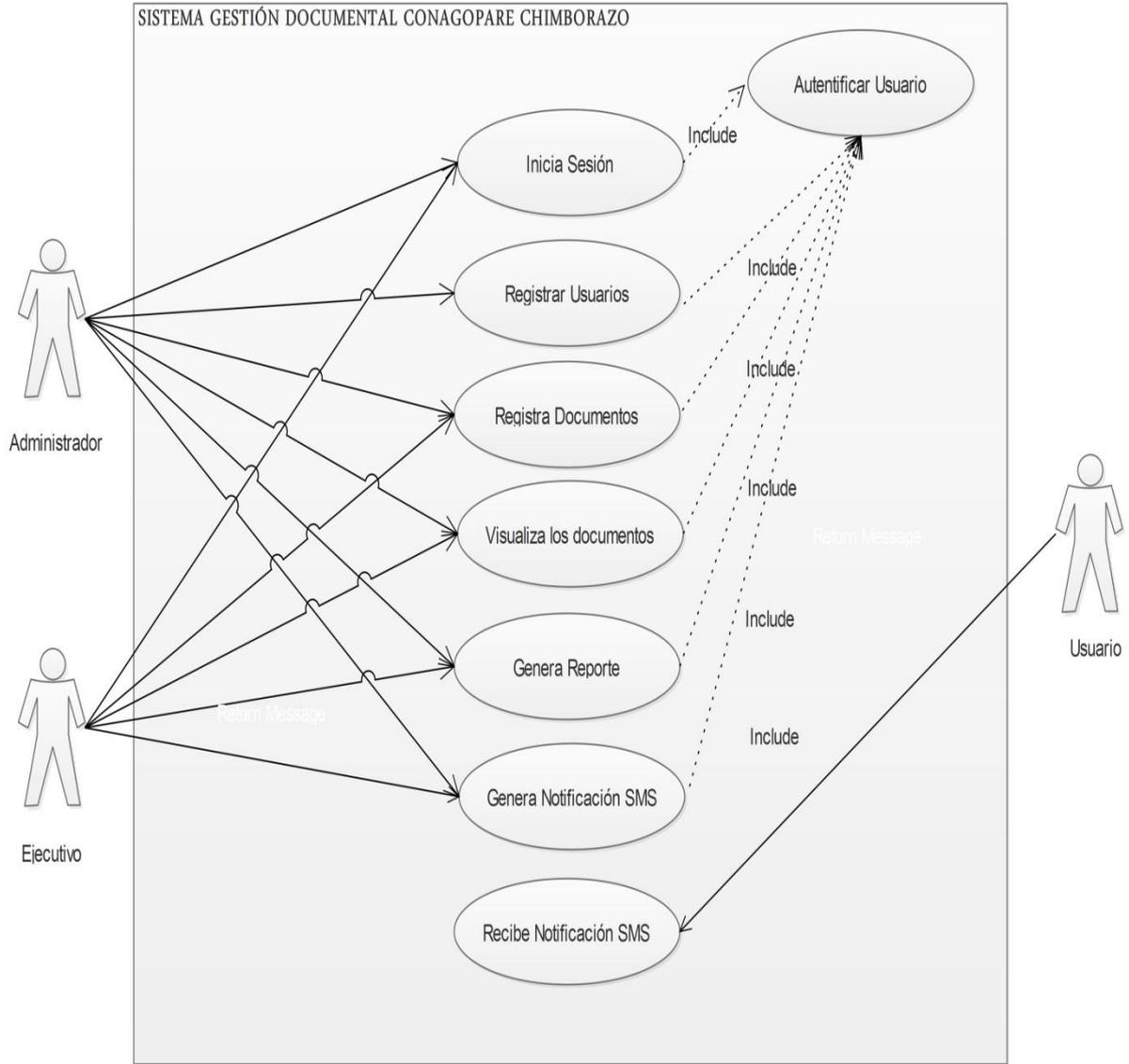
Los perfiles de Administrador y Ejecutivo deberán estar registrados en el sistema.

Realizado por: Los autores

Tabla 23: Especificación funcional 001-1

ESPECIFICACIÓN FUNCIONAL			
ID Requerimiento	001-1	Título	Diagrama de Casos de Uso

Diagrama caso de uso



Realizado por: Los autores

Tabla 24: Especificación funcional 002

ESPECIFICACIÓN FUNCIONAL			
ID Requerimiento	002	Título	Login

Login

Autenticación del usuario registrado en el SISTEMA DOCUMENTAL CONAGOPARE CHIMBORAZO.

Pre-requisito:

Deberá cumplir lo siguiente:

- Estar registrado en el sistema.

Consideraciones:

- Al ingresar al sistema el usuario podrá visualizar los módulos de acuerdo a su rol.
- Para cada registro de documento se genera una ID única.
- El aplicativo funciona de forma local.

Descripción

Los módulos se visualizan acorde al perfil o rol de usuario

Perfiles de usuarios

Perfil	Acceso
Administrador	Usuarios Tipo de Usuario Documentos Tipo de Documentos Emisor
Ejecutivo	Documentos Documentos en Espera Documentos Finalizados Emisor

Realizado por: Los autores

Tabla 25: Especificación funcional 002-1

ESPECIFICACIÓN FUNCIONAL			
ID Requerimiento	002-1	Título	Login

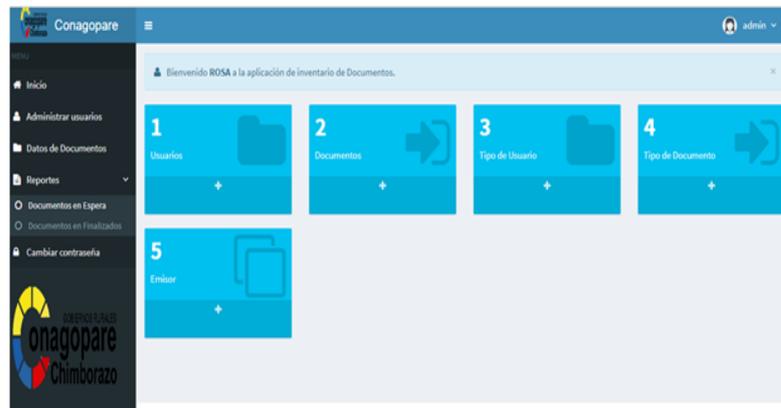
Interfaz Login

Interfaz login



Interfaz Administrador

Interfaz Administrador



Interfaz Ejecutivo

Interfaz ejecutivo



Realizado por: Los autores

Tabla 26: Especificación funcional 003

ESPECIFICACIÓN FUNCIONAL

ID Requerimiento	003	Titulo	Registro de Documentos
------------------	-----	--------	------------------------

Modulo de Documentos

El usuario podrá receiptar los documentos y registrarlos en el aplicativo web para el correspondiente seguimiento.

Pre-requisito:

Deberá cumplir lo siguiente:

- Tener iniciada la sesión en el aplicativo web.

Consideraciones:

- El usuario deberá escoger la junta parroquial de la que proviene el documento.
- El usuario deberá pedir un número de contacto celular para enviar la notificación.

Descripción

El módulo de documentos, presenta algunos campos el cual el usuario debe llenarlos con la información solicitada.

Interfaz de Documentos

Interfaz de documentos

The screenshot shows a web application interface for adding a document. The header is 'Conagopare' with a user profile icon. The main title is 'Agregar Documento'. The form contains the following fields: 'Numero del Documento' (text input), 'Fecha' (text input with value '01-08-2018'), 'Emisor' (text input), 'Telefono' (text input), 'Descripcion' (text input), 'Doc Adjunto' (file selection button showing 'Seleccionar archivo' and 'Ningún archivo seleccionado'), 'Tipo de Documento' (dropdown menu), and 'Tipo de Accion' (dropdown menu). At the bottom, there are 'Guardar' and 'Cancelar' buttons. A sidebar on the left contains navigation options: 'Inicio', 'Datos de Documentos', 'Reportes', 'Documentos en Espera', 'Documentos en Finalizados', and 'Cambiar contraseña'. The Conagopare logo is also visible in the sidebar.

Realizado por: Los autores

Tabla 27: Especificación funcional 004

ESPECIFICACIÓN FUNCIONAL

ID	Requerimiento	004	Titulo	Envío de Notificación SMS
----	---------------	-----	--------	---------------------------

Notificaciones de Texto SMS

El usuario podrá emitir la notificación una vez que el documento este gestionado.

Pre-requisito:

Deberá cumplir lo siguiente:

- Tener iniciada la sesión en el aplicativo web.

Consideraciones:

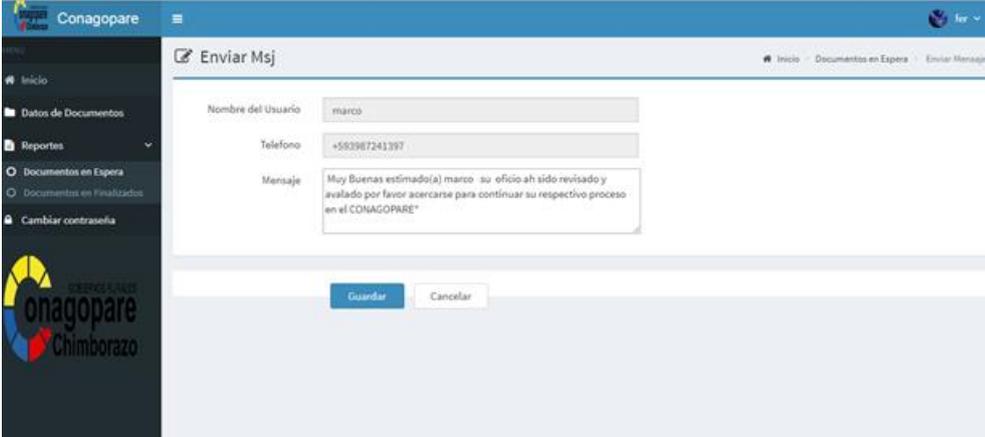
- El usuario deberá emitir la correspondiente notificación al número de contacto celular el cual se solicitó al ingresar el documento.

Descripción

Para el envío de la notificación se usa la interfaz de la aplicación web.

Interfaz de Documentos

Interfaz de documentos



The screenshot shows the 'Enviar Msj' (Send Message) interface of the Conagopare web application. The interface includes a sidebar menu on the left with options like 'Inicio', 'Datos de Documentos', 'Reportes', 'Documentos en Espera', 'Documentos en Finalizados', and 'Cambiar contraseña'. The main content area has a header with 'Enviar Msj' and a breadcrumb trail: 'Inicio > Documentos en Espera > Enviar Mensaje'. Below the header, there are three input fields: 'Nombre del Usuario' with the value 'marco', 'Telefono' with the value '+593987241397', and 'Mensaje' with the text 'Muy Buenas estimado(a) marco su oficio ah sido revisado y avalado por favor acercarse para continuar su respectivo proceso en el CONAGOPARE*'. At the bottom of the form, there are two buttons: 'Guardar' (Save) and 'Cancelar' (Cancel).

Realizado por: Los autores

ANEXO V:

Acta de Compromiso CONAGOPARE CHIMBORAZO

Ilustración 18: Acta de Compromiso CONAGOPARE CHIMBORAZO



CARTA DE COMPROMISO

Riobamba, 30 de noviembre de 2018

Quien suscribe, Ing. Marco Alvarez Coordinador Técnico del Conagopare Chimborazo tiene a bien suscribir la siguiente carta de compromiso.

En virtud de la importancia que representa el proyecto **ANÁLISIS DEL DESEMPEÑO ENTRE MONGODB Y COUCHDB UTILIZANDO NORMA ISO/IEC 25000. CASO PRÁCTICO: APLICACIÓN WEB DE GESTIÓN DE DOCUMENTOS CONAGOPARE CHIMBORAZO** el cual servirá para la automatizar el seguimiento de los documentos hasta su resolución, es por eso que a través de esta carta manifestamos nuestro compromiso con la realización de dicho proyecto. Cabe indicar que el proyecto de tesis se ha venido trabajando desde mes de Marzo del presente año.

De esta forma, nos comprometemos en realizar las tareas que conforman cada una de las etapas de desarrollo y en la realización de todos los aportes indicados de tal manera que quede implantado en la infraestructura del CONAGOPARE.

Para constancia de lo actuado, firman por duplicado el presente documento.

Ing. Marco Alvarez
UNIDAD DE PLANIFICACIÓN



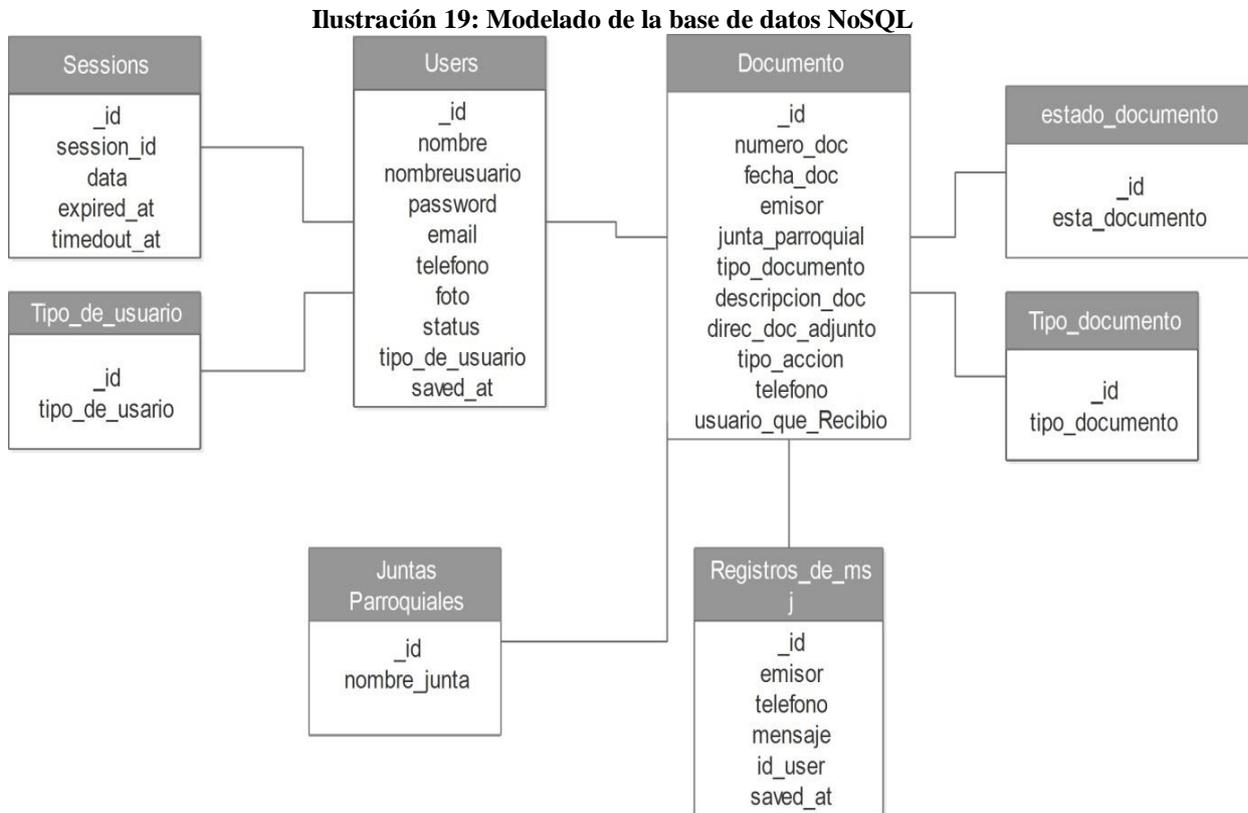
Sr. Pablo Calapucha
C.I. 1500713449

Sr. Marco Vinicio Tarco Caisa
C.I. 1725911273

Realizado por: CONAGOPARE CHIMBORAZO

ANEXO VI:

Modelado de la base de datos NoSQL



Realizado por: Los autores

ANEXO VII:

Instalación y configuración de las herramientas para el desarrollo

Requerimientos técnicos

Requerimientos mínimos de hardware

- **Procesador:** Dual Core
- **Memoria RAM:** Mínimo de 2GB RAM
- **Disco duro:** 200GB

Requerimientos mínimos de software

- **Sistema operativo:** Cualquier sistema Operativo de Windows

Instalación de la base de datos MongoDB

Descargar el instalador del software MongoDB `mongodb-win32-x86_64-2008plus-ssl-3.6.8` desde la página oficial <https://www.mongodb.com/> una vez descargado, se procede a crear una carpeta en la carpeta `C://mongodb` y descomprimir.

Ilustración 20: Archivo de la base de datos MongoDB

Nombre	Fecha de modifica...	Tipo	Tamaño
bin	25/02/2019 7:49	Carpeta de archivos	
GNU-AGPL-3.0	18/09/2018 13:56	Archivo 0	34 KB
mongodb-win32-x86_64-2008plus-ssl-3.6.8	24/10/2018 12:55	Archivo WinRAR Z...	300.200 KB
MPL-2	18/09/2018 13:56	Archivo	17 KB
README	18/09/2018 13:56	Archivo	3 KB
THIRD-PARTY-NOTICES	18/09/2018 13:56	Archivo	56 KB

Realizado por: Los autores

Se procede a crear la carpeta `C:\data\db` donde se guardaran las bases de datos

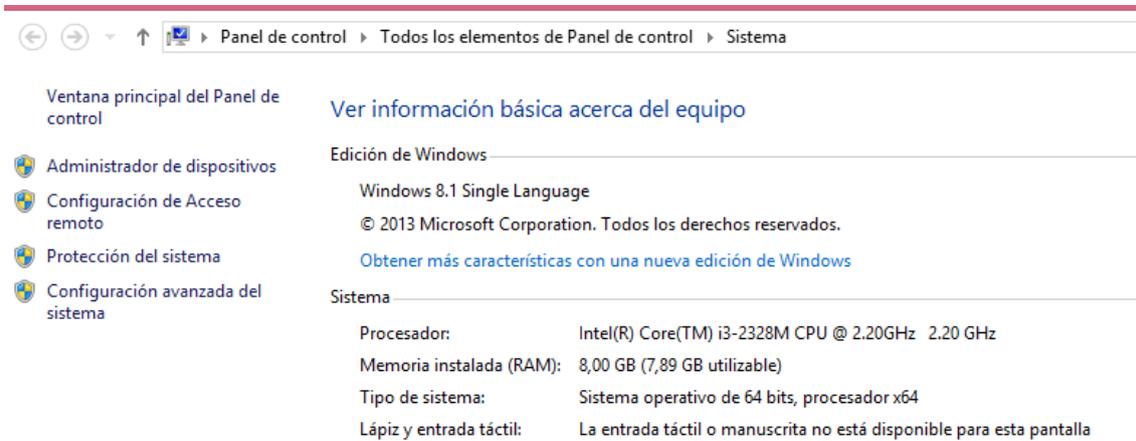
Ilustración 21: Dirección del db



Realizado por: Los autores

Iniciar MongoDB desde cualquier directorio del cmd se procede a ingresar a sistemas, configuración avanzada del sistema.

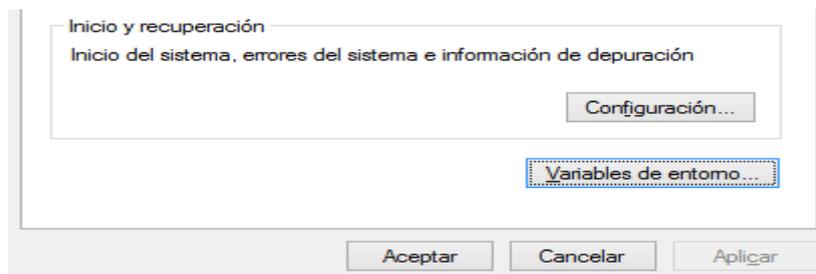
Ilustración 22: Configuración avanzada del sistema



Realizado por: Los autores

En configuración avanzada del sistema ingresamos y damos clic en variables de entorno.

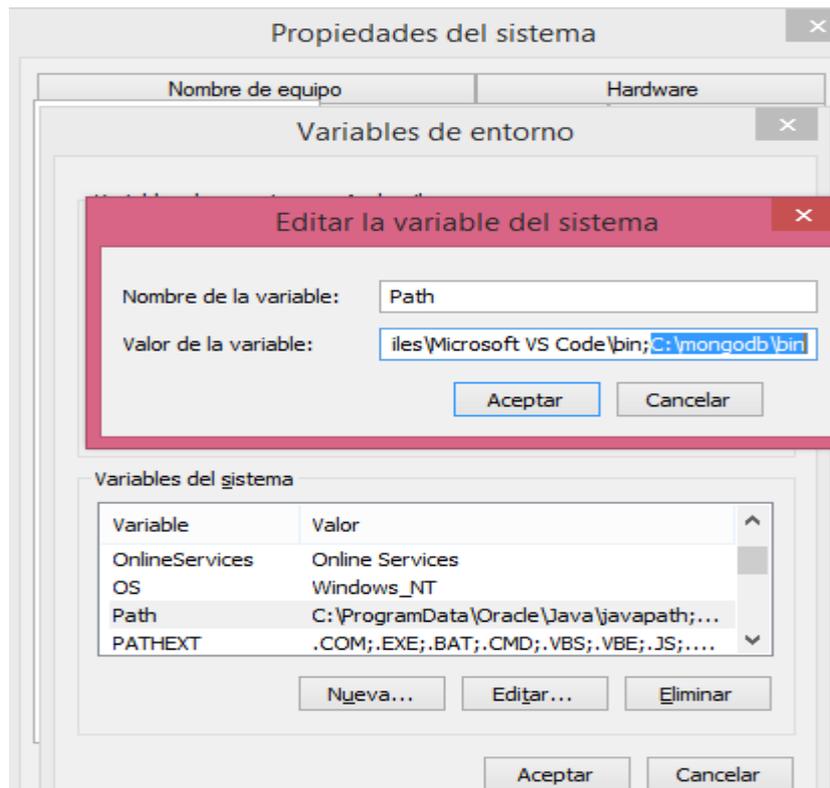
Ilustración 23: Configuración variables de entorno



Realizado por: Los autores

Ingresamos la ruta en la variable Path e insertamos la ruta de MongoDB y aceptamos.

Ilustración 24: Ruta de MongoDB ingresado en la Variable Path



Realizado por: Los autores

Para iniciar el servicio de MongoDB se procede abrir un terminal y se inserta el comando **mongod**.

Ilustración 25: Inicio del servidor mongod

```
C:\Users\Andronik>mongod
2019-03-10T23:33:32.920-0500 I CONTROL [initandlisten] MongoDB starting : pid=7
348 port=27017 dbpath=C:\data\db\ 64-bit host=Andronica
2019-03-10T23:33:32.921-0500 I CONTROL [initandlisten] targetMinOS: Windows 7/W
indows Server 2008 R2
2019-03-10T23:33:32.921-0500 I CONTROL [initandlisten] db version v3.6.8
2019-03-10T23:33:32.921-0500 I CONTROL [initandlisten] git version: 6bc9ed599c3
fa164703346a22bad17e33fa913e4
2019-03-10T23:33:32.922-0500 I CONTROL [initandlisten] OpenSSL version: OpenSSL
1.0.2o-fips 27 Mar 2018
2019-03-10T23:33:32.922-0500 I CONTROL [initandlisten] allocator: tcmalloc
2019-03-10T23:33:32.922-0500 I CONTROL [initandlisten] modules: none
2019-03-10T23:33:32.922-0500 I CONTROL [initandlisten] build environment:
2019-03-10T23:33:32.922-0500 I CONTROL [initandlisten] distmod: 2008plus-ss
l
2019-03-10T23:33:32.922-0500 I CONTROL [initandlisten] distarch: x86_64
```

Realizado por: Los autores

Para que el servicio de mongod este automáticamente se debe realizar lo siguiente:

Crear dos carpetas **log**, **data** en la carpeta mongodb.

Ilustración 26: Creación de la carpeta Log y Data

	bin	25/02/2019 7:49	Carpeta de archivos	
	data	10/03/2019 23:36	Carpeta de archivos	
	log	10/03/2019 23:35	Carpeta de archivos	
	GNU-AGPL-3.0	18/09/2018 13:56	Archivo 0	34 KB
	mongodb-win32-x86_64-2008plus-ssl-3.6.8	24/10/2018 12:55	Archivo WinRAR Z...	300.200 KB

Realizado por: Los autores

Se crea el archivo mongod.cfg y se ingresa lo siguiente.

```
echo logpath=C:\mongodb\log\mongo.log > C:\mongodb\mongod.cfg
```

Ejecutar en un terminal eh iniciamos el servicio:

```
mongod --config "C:\mongodb\mongod.cfg" --logpath "C:\mongodb\log\mongo.log" --install
```

Iniciar el servicio MongoDB

```
net start MongoDB
```

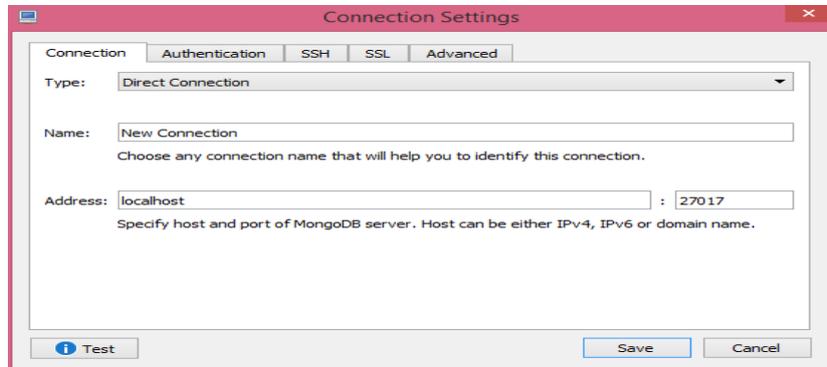
Ilustración 27: Inicio del servicio automático MongoDB

```
C:\WINDOWS\system32>mongod --config "C:\mongodb\mongod.cfg" --logpath "C:\mongodb\log\mongo.log" --install
2019-03-11T00:00:15.573-0500 I CONTROL [main] log file "C:\mongodb\log\mongo.log" exists; moved to "C:\mongodb\log\mongo.log.2019-03-11T05-00-15".
C:\WINDOWS\system32>net start MongoDB
El servicio de MongoDB está iniciándose..
El servicio de MongoDB se ha iniciado correctamente.
```

Realizado por: Los autores

Con el software Robo 3T se crea la conexión a MongoDB y lo comprueba dando clic en Test.

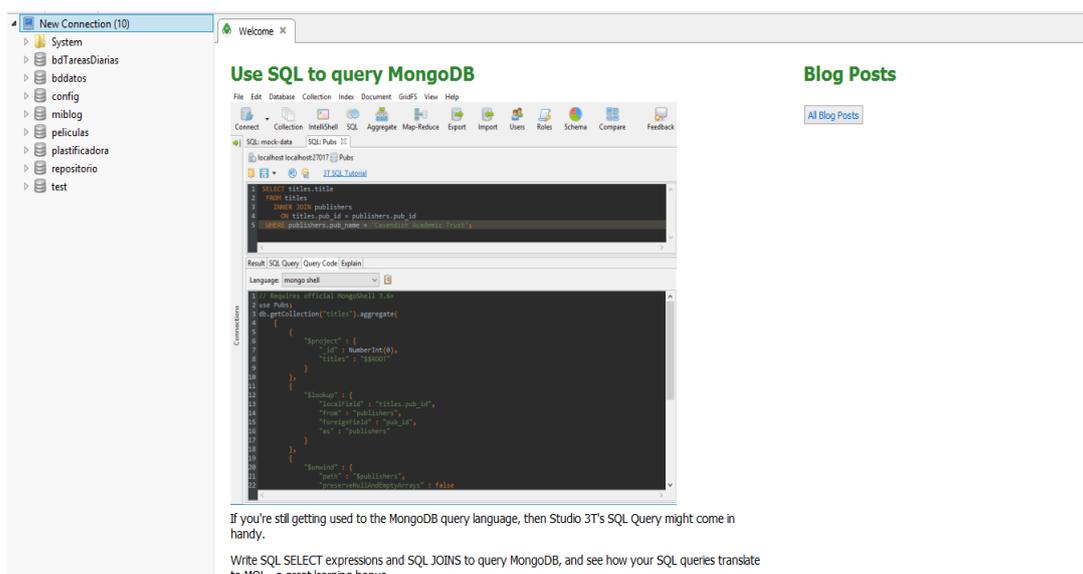
Ilustración 28: Conexión a la base de datos



Realizado por: Los autores

Por último se visualiza la interfaz de Robo 3T

Ilustración 29: Interfaz Robo 3T

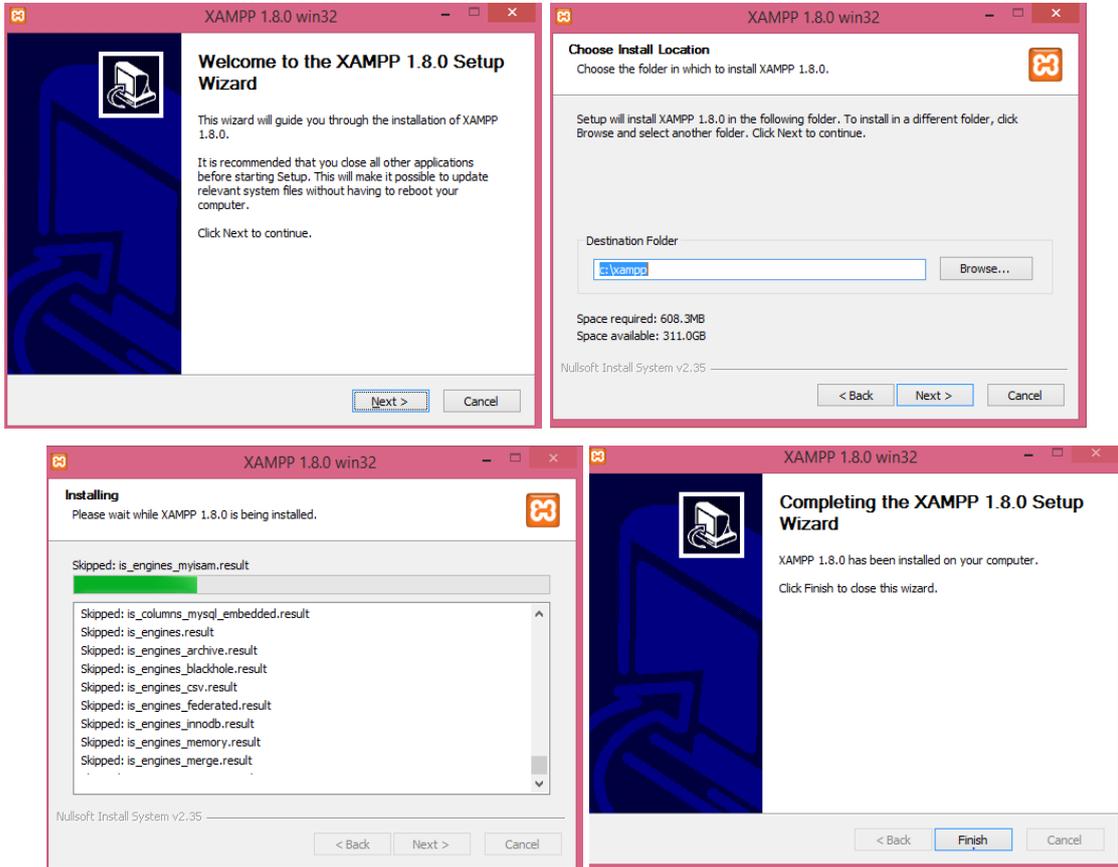


Realizado por: Los autores

Instalación de web server

Descargar desde la página oficial el instalador `xampp-win32-1.8.0-VC9-installer` <https://www.apachefriends.org/es/index.html> y se procede a instalar.

Ilustración 30: Instalación del servidor Xampp



Realizado por: Los autores

Una vez instalado el servidor se procede a configurar para mongodb, se descarga el complemento `php_mongo.dll` y se inserta en la carpeta `C:\xampp\php\ext`.

Ilustración 31: Complemento php_mongo.dll



Realizado por: Los autores

Se configura el archivo php.ini en la dirección C:\xampp\php.

Ilustración 32: Archivo php.ini



Realizado por: Los autores

Se ingresa el nuevo complemento y se guarda los cambios.

Extension = php_mongodb.dll

Verificamos en el servidor Xampp sí reconoce el nuevo complemento.

Ilustración 33: MongoDB en el servidor Xampp

MongoDB Support		enabled
Version		1.3.2RC1

Directive	Local Value	Master Value
mongo.allow_empty_keys	0	0
mongo.chunk_size	262144	262144
mongo.cmd	\$	\$
mongo.default_host	localhost	localhost
mongo.default_port	27017	27017
mongo.is_master_interval	no value	no value
mongo.long_as_object	0	0
mongo.native_long	0	0
mongo.ping_interval	no value	no value
mongo.utf8	1	1

Realizado por: Los autores

Ahora para culminar la configuración del Xampp se procede aumentar el tamaño max de importación y reiniciamos los servicios.

Ilustración 34: Maximizar el tamaño de los archivos

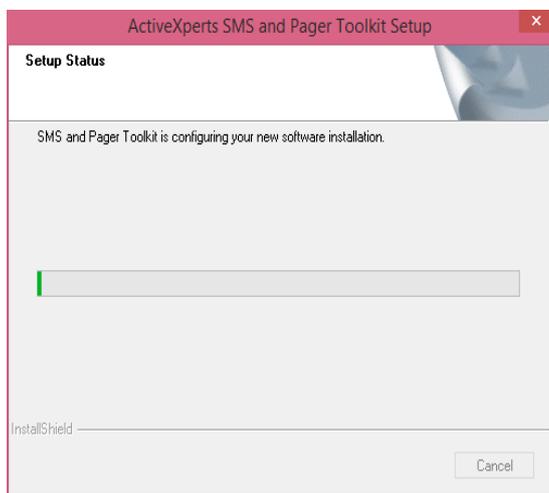
```
; Maximum allowed size for uploaded files.  
; http://php.net/upload-max-filesize  
upload_max_filesize = 20M  
post_max_size = 13M  
memory_limit = 15M
```

Realizado por: Los autores

Instalación del componente Activexperts SMS Component

Se descarga el instalador SMS_and_Pager_Toolkit_24399 de la página oficial <https://www.activexperts.com/sms-component/> y se procede instalar.

Ilustración 35: Instalación del ACTIVEEXPERTS SMS COMPONENT



Realizado por: Los autores

Una vez instalado se procede a configurar en el servidor Xampp en la dirección C:\xampp\php

Ilustración 36: Archivo php.ini



Realizado por: Los autores

Se ingresa los siguientes comandos y se debe reiniciar el servidor.

Ilustración 37: Configuración del php_com_dotnet

```
[PHP_COM_DOTNET]
extension=php_com_dotnet.dll
extension_dir = "ext"
enable_dl = upload_max_filesize0n
[curl]
curl.cainfo=c:/xampp/php/cacert.pem?
```

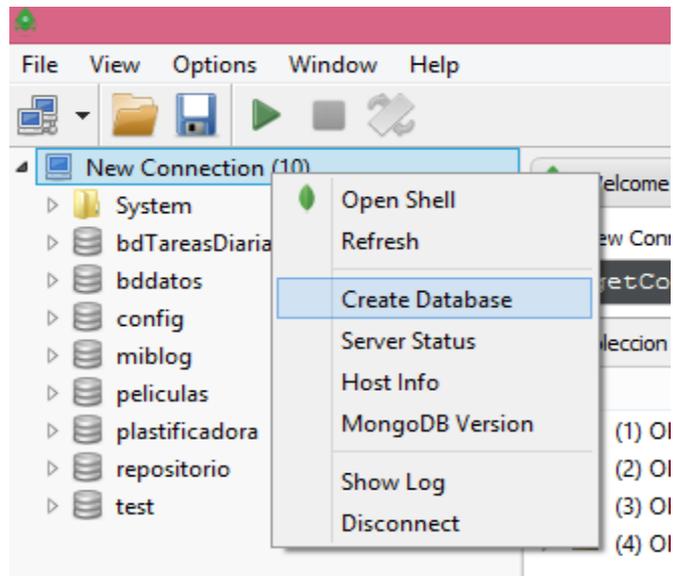
Realizado por: Los autores

ANEXO VIII:

Creación de la base de datos en Robo 3T

Se crea la base de datos miblog en software Robo 3T.

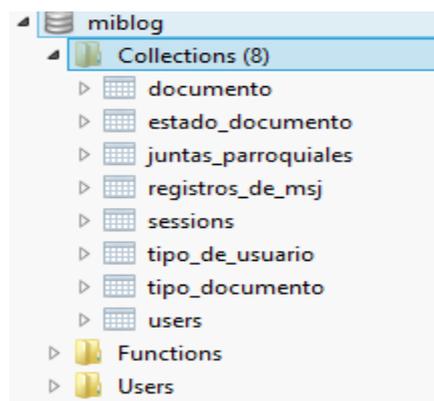
Ilustración 38: Creación de la Base de datos en Robo 3T



Realizado por: Los autores

Se crea las colecciones: documento, estado_documento, juntas_parroquiales, registros_de_msj, sessions, tipo_de_usuario y tipo_de_documento.

Ilustración 39: Creación de colecciones en Robo 3T



Realizado por: Los autores

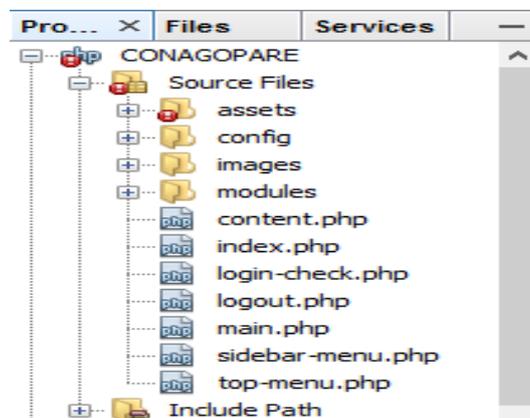
ANEXO IX:

Creación de la aplicación web

En el desarrollo de la aplicación se procedió a seguir la estructura MVC (Modelo, Vista, Controlador), que se encarga de realizar las operaciones CRUD (Crear, Leer, Actualizar, Eliminar). La aplicación cuenta con los siguientes archivos.

- Assets: se encuentra los plugins de la interfaz gráfica css, js.
- Config: en esta carpeta está la conexión a la base de datos.
- Images: se encuentran las imágenes del sistema.
- Modules: se encuentran los módulos MVC.

Ilustración 40: Estructura de la aplicación web



Realizado por: Los autores

Conexión a la base de datos MongoDB

La conexión a la base de datos se lo encuentra en el archivo config/dbconnection.php.

Ilustración 41: Conexión Base de datos

```
<?php
class DBConnection{
const HOST = 'localhost';
const PORT = 27017;
const DBNAME = 'miblog';
private static $instance;
public $connection;
public $database;
private function __construct(){
$connectionString = sprintf('mongodb://%s:%d', DBConnection::HOST, DBConnection::PORT);
try{ $this->connection = new MongoClient($connectionString);
$this->database = $this->connection->selectDB(DBConnection::DBNAME); } catch
(MongoConnectionException $e) { throw $e; }
static public function instantiate(){
if(!isset(self::$instance)){
$class = __CLASS__;
self::$instance = new $class;}
return self::$instance;}
public function getCollection($name){
return $this->database->selectCollection($name);}}
?>
```

Realizado por: Los autores

Login

El login inicia en el archivo index.php.

Ilustración 42: Interfaz de Login



Realizado por: Los autores

El proceso de verificar los datos se lo visualiza en el archivo login-check.php.

Ilustración 43: Proceso de inicio de sesión

```
<?php
require_once "config/dbconnection.php";
$mongo = DBConnection::instantiate();
$collection = $mongo->getCollection('users');
$username = $_POST['username'];
$password = $_POST['password'];
if (!ctype_alnum($username) OR !ctype_alnum($password)) {
header("Location: index.php?alert=1");}else{
$query = array('nombreusuario'=> $username, 'password'=> md5($password), 'status' => 'activo');
$articulo = $collection->findOne($query);if ($articulo > 0) {session_start();
$_SESSION['id_user'] = $articulo['_id'];
$_SESSION['username'] = $articulo['nombreusuario'];
$_SESSION['password'] = $articulo['password'];
$_SESSION['name_user'] = $articulo['nombre'];
$_SESSION['permisos_acceso'] = $articulo['tipo_de_usuario'];
header("Location: main.php?module=start");}else {
header("Location: index.php?alert=1");}}?>
```

Realizado por: Los autores

Contenido de la página principal: El archivo content.php permiten direccionar a los módulos de las vistas.

Ilustración 44: Contenido de la página principal

```
<?php
require_once "config/dbconnection.php";
if (empty($_SESSION['username']) && empty($_SESSION['password'])){
echo "<meta http-equiv='refresh' content='0'; url=index.php?alert=1'>";
}else {if ($_GET['module'] == 'start') {
include "modules/start/view.php";
}elseif ($_GET['module'] == 'stock_inventory') {
include "modules/stock_inventory/view.php";
}elseif ($_GET['module'] == 'form_stock_inventory') {
include "modules/stock_inventory/form.php";
}elseif ($_GET['module'] == 'stock_report') {
include "modules/stock_report/view.php";
}elseif ($_GET['module'] == 'stock_report2') {
include "modules/stock_report/view2.php";
}elseif ($_GET['module'] == 'user') {
include "modules/user/view.php";
}elseif ($_GET['module'] == 'form_user') {
include "modules/user/form.php";}
```

Realizado por: Los autores

Agregar usuario

Ilustración 45: Interfaz de agregar usuarios

Nombre del Proveedor

Telefono

Direccion

Correo

Realizado por: Los autores

El proceso que realiza la inserción de un usuario es el archivo `modules/user/proses.php` obtenido los datos ingresados en la interfaz `modules/user/form.php`.

Ilustración 46: Contenido de la página principal

```
<?php
session_start();
require_once "../config/dbconnection.php";
$mongo = DBConnection::instantiate();
$collection = $mongo->getCollection('users');
if ($_GET['act']=='insert') {
if (isset($_POST['Guardar'])) {
$article = array();
$article['nombreusuario'] = $_POST['username'];
$article['password'] = md5($_POST['password']);
$article['nombre'] = $_POST['name_user'];
$article['tipo_de_usuario'] = $_POST['permisos_acceso'];
$article['status']=$_POST['status'];
$article['saved_at']= new MongoDate();
$collection->insert($article);
if ($collection) {
header("location: ../main.php?module=user&alert=1");
}}}
```

Realizado por: Los autores

Actualizar usuario

Ilustración 47: Interfaz de modificar usuario

The screenshot shows a web application interface for modifying user data. The header includes the 'Conagopare' logo and a user profile 'admin'. A sidebar menu on the left lists options like 'Inicio', 'Administrar usuarios', and 'Cambiar contraseña'. The main content area is titled 'Modificar datos de Usuario' and contains a form with the following fields: 'Nombre de Usuario' (text input with 'fer'), 'Nombre' (text input with 'fernando Basantes'), 'Email' (text input with 'fer@hotmail.com'), 'Telefono' (text input with '0923823850'), 'Foto' (file selection button and a preview image), and 'Permisos de acceso' (dropdown menu with 'planificacion'). At the bottom of the form are 'Guardar' and 'Cancelar' buttons.

Realizado por: Los autores

El proceso de actualizar un usuario es mediante la búsqueda del `_id` de un usuario, el archivo `modules/user/proses.php` realiza la actualización del usuario.

Ilustración 48: Actualización del usuario

```
elseif ($_GET['act']=='update') {
if (isset($_POST['Guardar'])) {
if (isset($_POST['id_user'])) {
$id_user      = $_POST['id_user'];
$username     = $_POST['username'];
$password     = md5($_POST['password']);
$name_user    = $_POST['name_user'];
$email        = $_POST['email'];
$telefono     = $_POST['telefono'];
$permisos_acceso = $_POST['tipo_de_usuario'];
$name_file    = $_FILES['foto']['name'];
$kukuran_file = $_FILES['foto']['size'];
$tipe_file    = $_FILES['foto']['type'];
$tmp_file     = $_FILES['foto']['tmp_name'];
$allowed_extensions = array('jpg','jpeg','png');
$path_file    = "../images/user/".$name_file;
$file         = explode(".", $name_file);
$extension    = array_pop($file);
```

```

if (empty($_POST['password']) && empty($_FILES['foto']['name'])) {
$status = array('$set' => array("nombreusuario"=> $username , "nombre"=>$name_user
,"email"=>$email, "telefono" => $telefono, "tipo_de_usuario"=>$permisos_acceso));
$collection->update(array('_id'=> new MongoClient($id_user)), $status);
if ($collection) {header("location: ../../main.php?module=user&alert=2");}}

elseif (!empty($_POST['password']) && empty($_FILES['foto']['name'])) {
$status = array('$set' => array("nombreusuario"=> $username
,"nombre"=>$name_user, "password"=>$password, "email"=>$email, "telefono" => $telefono,
"tipo_de_usuario"=>$permisos_acceso));
$collection->update(array('_id'=> new MongoClient($id_user)), $status);if ($collection) {
header("location: ../../main.php?module=user&alert=2");}}
elseif (empty($_POST['password']) && !empty($_FILES['foto']['name'])) {
if (in_array($extension, $allowed_extensions)) {
if($ukuran_file <= 1000000) { if(move_uploaded_file($tmp_file, $path_file)) {
$status = array('$set' => array("nombreusuario"=> $username
,"nombre"=>$name_user, "email"=>$email, "telefono" => $telefono, "foto"=>$name_file,
"tipo_de_usuario"=>$permisos_acceso));
$collection->update(array('_id'=> new MongoClient($id_user)), $status);if ($collection) {
header("location: ../../main.php?module=user&alert=2");}} else {
header("location: ../../main.php?module=user&alert=5");}} else {
header("location: ../../main.php?module=user&alert=6");}} else {
header("location: ../../main.php?module=user&alert=7");}} else {
if (in_array($extension, $allowed_extensions)) { if($ukuran_file <= 1000000) {
if(move_uploaded_file($tmp_file, $path_file)) {
$status = array('$set' => array("nombreusuario"=> $username
,"nombre"=>$name_user, "password"=>$password, "email"=>$email, "telefono" =>
$telefono, "foto"=>$name_file, "tipo_de_usuario"=>$permisos_acceso));
$collection->update(array('_id'=> new MongoClient($id_user)), $status);
if ($collection) {header("location: ../../main.php?module=user&alert=2");}} else {
header("location: ../../main.php?module=user&alert=5");}} else {
header("location: ../../main.php?module=user&alert=6");}} else {
header("location: ../../main.php?module=user&alert=7");}}}}}}

```

Realizado por: Los autores

Vista de usuarios

Ilustración 49: Interfaz de modificar usuario

The screenshot displays the 'Gestión de Usuarios' (User Management) interface. On the left is a dark sidebar menu with options: Inicio, Administrar usuarios, Datos de Documentos, Reportes, Documentos en Espera, Documentos en Finalizados, and Cambiar contraseña. The main header is blue with the 'Conagopare' logo and a user profile 'admin'. The main content area has a title 'Gestión de Usuarios' and an 'Agregar' button. Below the title is a search bar and a 'Mostrar 10 registros' dropdown. The main part of the interface is a table with the following data:

No.	Foto	Nombre de usuario	Nombre	Permisos de acceso	Status	
1		admin	admin	Super Admin	activo	
2		fer	fernando Basantes	planificacion	activo	
3		walter	wilmer valle	gerente	activo	
4		estelita	estelita	gerente	bloqueado	
5		marco	mark	planificacion	activo	
6		andres11	andres11	Super Admin	bloqueado	
7		jared	jared	planificacion	activo	

Realizado por: Los autores

La visualización de los usuarios está en el archivo modules/user/view.php.

Ilustración 50: Visualización de los usuarios

```
$no = 1;
$mongodb= new Mongo();
$articleCollection = $mongodb->miblog->users;
$cursor = $articleCollection->find(array(),
array('_id','nombre','nombreusuario','password','foto','tipo_de_usuario','status','saved_at'));
while ($cursor->hasNext()): $article = $cursor->getNext();
echo "<tr>
<td width='50' class='center'>$no</td>";
if ($article['foto']== "") { ?>
<td class='center'><img class='img-user' src='images/user/user-default.png' width='45'></td>
<?php} else { ?>
<td class='center'><img class='img-user' src='images/user/<?php echo $article['foto']; ?>'
width='45'></td>
<?php} ?> <td><?php echo substr($article['nombreusuario'], 0, 35); ?></td>
<td><?php echo substr($article['nombre'], 0, 35) ; ?></td>
<td><?php echo substr($article['tipo_de_usuario'], 0, 35) ; ?></td><td>
<?php echo substr($article['status'], 0, 35) ; ?></td>
<td class='center' width='100'> <?php
if ($article['status']=='activo') { ?>
<a data-toggle="tooltip" data-placement="top" title="Bloqueado" style="margin-right:5px"
class="btn btn-warning btn-sm" href="modules/user/proses.php?act=off&id=<?php echo
$article['_id'];?>">
<i style="color:#fff" class="glyphicon glyphicon-off"></i></a>
<?php} else { ?>
<a data-toggle="tooltip" data-placement="top" title="activo" style="margin-right:5px" class="btn
btn-success btn-sm" href="modules/user/proses.php?act=on&id=<?php echo $article['_id'];?>">
<i style="color:#fff" class="glyphicon glyphicon-ok"></i>
</a><?php
} ?>
<a data-toggle='tooltip' data-placement='top' title='Modificar' class='btn btn-primary btn-sm'
href='?module=form_user&form=edit&id=<?php echo $article['_id'];?>'>
<i style='color:#fff' class='glyphicon glyphicon-edit'></i>
</a></div></td>
<?php
$no++;endwhile;
?>
```

Realizado por: Los autores

Insertar, actualizar tipos de usuarios y tipo de documentos

Las inserciones y las actualizaciones son similares al del usuario pero para encontrar los archivos de inserción es en `modules/tipuser/proses.php` y de tipo de documento es en `modules/tipudoc/proses.php`.

Ilustración 51: Insertar y actualizar tipos de usuarios y tipo de documento

```
<?php
session_start();
require_once "../config/dbconnection.php";
$mongo = DBConnection::instantiate();
$collection = $mongo->getCollection('tipo_de_usuario');
if ($_GET['act']=='insert') { if (isset($_POST['Guardar'])) {
$article = array();
$article['tipo_de_usuario'] = $_POST['tipousuario'];
$article['saved_at'] = new MongoDate();
$collection->insert($article);
if ($collection) { header("location: ../main.php?module=tipuser&alert=1");} } }
elseif ($_GET['act']=='update') { if (isset($_POST['Guardar'])) {
if (isset($_POST['id_user'])) {
$id_user = $_POST['id_user'];
$username = $_POST['tipo_de_usuario'];
$status = array('$set' => array("tipo_de_usuario" => $username));
$collection->update(array('_id' => new MongoClient($id_user)), $status);
if ($collection) { header("location: ../main.php?module=tipuser&alert=2");} } } }
elseif ($_GET['act']=='delete') { if (isset($_GET['id'])) {
$codigo = $_GET['id'];
$collection->remove(array('_id' => new MongoClient($codigo)));
if ($collection) { header("location: ../main.php?module=tipuser&alert=3");} } }
elseif ($_GET['act']=='on') { if (isset($_GET['_id'])) { $connection = new MongoClient();
$db = $connection->selectDB('miblog');
$collection = $db->selectCollection('users');
$collection->update(array('_id' => new MongoClient($id_user)), $status);
$id_user = $_GET['_id'];
$status = 'activo';
if ($collection) { header("location: ../main.php?module=user&alert=3");} } }
elseif ($_GET['act']=='off') { if (isset($_GET['_id'])) {
$connection = new MongoClient();
$db = $connection->selectDB('miblog');
$collection = $db->selectCollection('users');
$collection->update(array('_id' => new MongoClient($id_user)), $status);
$id_user = $_GET['_id'];
$status = 'bloqueado';
if ($collection) { header("location: ../main.php?module=user&alert=4");} } }
?>
```

Realizado por: Los autores

Eliminar tipos de usuarios y tipo de documentos

El eliminar un tipo de usuario o tipo de documentos están en los modules/ tipuser/proses.php y modules/tipudoc/proses.php.

Ilustración 52: Eliminar tipos de usuarios y tipo de documentos

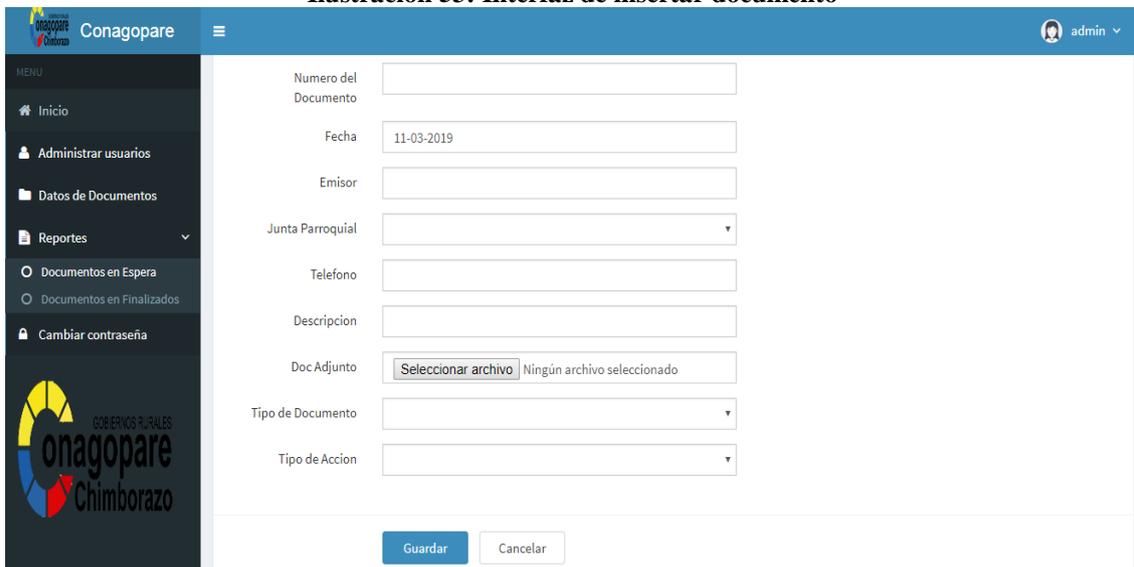
```
elseif ($_GET['act']=='delete') {  
    if (isset($_GET['id'])) {  
        $codigo = $_GET['id'];  
        $collection->remove(array('_id'=> new MongoClient($codigo)));  
        if ($collection) {  
            header("location: ../../main.php?module=tipuser&alert=3");  
        }  
    }  
}
```

Realizado por: Los autores

Insertar documentos

La inserción de datos se lo realiza en el archivo modules/stock_inventory/form.php.

Ilustración 53: Interfaz de insertar documento



The screenshot shows the 'Conagopare' web application interface. On the left is a dark sidebar menu with the following items: 'Inicio', 'Administrar usuarios', 'Datos de Documentos', 'Reportes', 'Documentos en Espera', 'Documentos en Finalizados', and 'Cambiar contraseña'. The main content area is a form for inserting a document. The form fields are: 'Numero del Documento' (text input), 'Fecha' (text input with value '11-03-2019'), 'Emisor' (text input), 'Junta Parroquial' (dropdown menu), 'Telefono' (text input), 'Descripción' (text input), 'Doc Adjunto' (file selection button labeled 'Seleccionar archivo' and 'Ningún archivo seleccionado'), 'Tipo de Documento' (dropdown menu), and 'Tipo de Accion' (dropdown menu). At the bottom of the form are two buttons: 'Guardar' (blue) and 'Cancelar' (white).

Realizado por: Los autores

En el archivo modules/stock_inventory/proses.php, realiza el proceso de inserción y a su vez realiza el envío de mensaje si finaliza.

Ilustración 54: Insertar documento

```
<?php
session_start();
if ($_GET['act']== 'insert') {
if (isset($_POST['Guardar'])) {
$connection = new Mongo();
$database = $connection->selectDB('miblog');
$collection = $database->selectCollection('documento');
$collection1 = $database->selectCollection('registros_de_msj');
$article = array();
$article['numero_doc']= $_POST['numero_doc'];
$article['fecha_doc']= $_POST['fecha_doc'];
$article['emisor']= $_POST['emisor'];
$article['junta_parroquial']= $_POST['junta_parroquial'];
$article['telefono']= $_POST['telefono'];
$article['descripcion_doc']= $_POST['descripcion_doc'];
$article['direc_doc_adjunto'] = $_FILES['direc_doc_adjunto']['name'];
$styp = $_FILES['direc_doc_adjunto']['type']; if ($article['direc_doc_adjunto'] != "") {
$dir = "./documentos/";
move_uploaded_file($_FILES['direc_doc_adjunto']['tmp_name'], "documentos/" .
$_FILES['direc_doc_adjunto']['name']);
$ar = $_FILES['direc_doc_adjunto']['name'];} else {echo "<script> alert('Formato no
válido!');</script>";}
$article['tipo_documento']= $_POST['tipo_documento'];
$article['tipo_accion']= $_POST['tipo_accion'];
$article['usuario_que_recibio']= $_POST['usuario_que_recibio'];
$article['saved_at']= new MongoDate(); $article1 = array();
$article1['emisor'] = $_POST['emisor'];
$article1['telefono'] = "+593".$_POST['telefono'];
$article1['mensaje'] = "Muy Buenas Señor(a) ".$article1['emisor']." su ".$article['tipo_documento']."
ah sido revisado y aprobado por favor acercarse a CONAGOPARE Riobamba";
$article1['id_user'] = $_POST['usuario_que_recibio'];
$article1['saved_at']= new MongoDate();
if($article['tipo_accion']=="espera"){ $collection->insert($article); if ($collection) {
header("location: ../../main.php?module=stock_inventory&alert=1");} } else {
$collection->insert($article); $collection1->insert($article1);
$objGsmOut = new COM ("ActiveXperts.GsmOut"); $archivo = 'D:\mensajes';
$dispositivo = 'HUAWEI Mobile Connect - 3G Modem'; $velocidad = 0;
$objGsmOut->LogFile = $archivo;
$objGsmOut->Device = $dispositivo;
$objGsmOut->DeviceSpeed = $velocidad;
$objGsmOut->MessageRecipient = $article1['telefono'];
$objGsmOut->MessageData = $article1['mensaje'];
if($objGsmOut->LastError == 0){ $objGsmOut->Send;
echo 'OK';} if ($collection) {
header("location: ../../main.php?module=stock_inventory&alert=1");} } } }
```

Realizado por: Los autores

Actualizar un documento

Ilustración 55: Interfaz de actualizar documento

The screenshot shows a web application interface for updating a document. The header includes the logo for 'GOBIERNO AUTÓNOMO DESCENTRALIZADO Conagopare Chimborazo' and a user profile 'admin'. The main content area is titled 'Modificar datos del Documento' and contains several input fields: 'Numero del Documento' with the value 'doc_1', 'Fecha' with '17-07-2018', 'Emisor' with 'Pablo', 'Junta Parroquial' with a dropdown menu showing 'COLUMBE', 'Descripcion' (empty), 'Doc Adjunto' with a file selection button and the name 'EstudianteReport.pdf', and 'Tipo de Documento' with a dropdown menu showing 'Informe'. At the bottom of the form are two buttons: 'Guardar' and 'Cancelar'.

Realizado por: Los autores

Para la modificación se realiza en el `modules/stock_inventory/proses.php`.

Ilustración 56: Actualizar Documento

```
elseif ($_GET['act']=='update') {
if (isset($_POST['Guardar'])) {
if (isset($_POST['id_doc'])) {
$id_doc = $_POST['id_doc'];
$numero_doc = $_POST['numero_doc'];
$fecha_doc = $_POST['fecha_doc'];
$emisor = $_POST['emisor'];
$junta_parroquial = $_POST['junta_parroquial'];
$telefono = $_POST['telefono'];
$descripcion_doc = $_POST['descripcion_doc'];
$documento = $_FILES['direc_doc_adjunto']['name'];
$typ = $_FILES['direc_doc_adjunto']['type'];
if ($article['direc_doc_adjunto'] != "") {
$dir = "./documentos/";
move_uploaded_file($_FILES['direc_doc_adjunto']['tmp_name'], "documentos/" .
$_FILES['direc_doc_adjunto']['name']);
$ar = $_FILES['direc_doc_adjunto']['name'];
} else {echo "<script> alert('Formato no válido!');</script>";}
$tipo_documento=$_POST['tipo_documento'];
$tipo_accion=$_POST['tipo_accion'];
$usuario_que_recibio=$_POST['usuario_que_recibio'];
if (empty($_POST['numero_doc']) && empty($_FILES['direc_doc_adjunto']['name'])) {
$status = array('$set' => array("numero_doc"=> $numero_doc , "fecha_doc"=>$fecha_doc
, "emisor"=>$emisor, "junta_parroquial" => $junta_parroquial, "telefono"=>$telefono,
"descripcion_doc"=>$descripcion_doc, "direc_doc_adjunto"=>"$documento", "tipo_documento"=
>$tipo_documento, "tipo_accion"=>$tipo_accion, "usuario_que_recibio"=>$usuario_que_recibio))
;
}
```

```

$collection->update(array('_id'=> new MongoId($id_doc)), $status);
if ($collection) {
header("location: ../../main.php?module=user&alert=2");}
elseif (!empty($_POST['numero_doc']) && empty($_FILES['direc_doc_adjunto']['name'])) {
$status = array('$set' => array("numero_doc"=> $numero_doc , "fecha_doc"=>$fecha_doc
,"emisor"=>$emisor, "junta_parroquial" => $junta_parroquial, "telefono"=>$telefono,
"descripcion_doc"=>$descripcion_doc,"direc_doc_adjunto"=>"$documento","tipo_documento"=
>$tipo_documento,"tipo_accion"=>$tipo_accion,"usuario_que_recibio"=>$usuario_que_recibio))
$collection->update(array('_id'=> new MongoId($id_doc)), $status);
if ($collection) {header("location: ../../main.php?module=user&alert=2");}}
elseif (empty($_POST['numero_doc']) && !empty($_FILES['direc_doc_adjunto']['name'])) {
if (in_array($extension, $allowed_extensions)) {
if($ukuran_file <= 1000000) {
if(move_uploaded_file($tmp_file, $path_file)) {
$status = array('$set' => array("numero_doc"=> $numero_doc , "fecha_doc"=>$fecha_doc
,"emisor"=>$emisor, "junta_parroquial" => $junta_parroquial, "telefono"=>$telefono,
"descripcion_doc"=>$descripcion_doc,"direc_doc_adjunto"=>"$documento","tipo_documento"=
>$tipo_documento,"tipo_accion"=>$tipo_accion,"usuario_que_recibio"=>$usuario_que_recibio));
$collection->update(array('_id'=> new MongoId($id_doc)), $status);

if ($collection) {
header("location: ../../main.php?module=user&alert=2");} } else {
header("location: ../../main.php?module=user&alert=5");} } else {
header("location: ../../main.php?module=user&alert=6");} } else {
header("location: ../../main.php?module=user&alert=7");} }
else { if (in_array($extension, $allowed_extensions)) {
if($ukuran_file <= 1000000) {
if(move_uploaded_file($tmp_file, $path_file)) {
$status = array('$set' => array("numero_doc"=> $numero_doc , "fecha_doc"=>$fecha_doc
,"emisor"=>$emisor, "junta_parroquial" => $junta_parroquial, "telefono"=>$telefono,
"descripcion_doc"=>$descripcion_doc,"direc_doc_adjunto"=>"$documento","tipo_documento"=
>$tipo_documento,"tipo_accion"=>$tipo_accion,"usuario_que_recibio"=>$usuario_que_recibio))
$collection->update(array('_id'=> new MongoId($id_doc)), $status);
if ($collection) {
header("location: ../../main.php?module=user&alert=2");} } else {
header("location: ../../main.php?module=user&alert=5");}
} else {
header("location: ../../main.php?module=user&alert=6");} } else {
header("location: ../../main.php?module=user&alert=7");} } } }

```

Realizado por: Los autores

Visualizar los documentos

En la visualización de documentos finalizados.

Ilustración 57: Interfaz de visualización documentos

No.	fecha_doc	numero Documento	emisor	tipo de Documento	Descripción	Documento Adjunto	tipo de Accion	telefono	usuario que recibio
11	16-07-2018	otra mas	goliso	Informe	lpo	PIC-15-INAMHI-001.pdf	finalizado	0988243748	rosa
16	29-07-2018	doc-23	Ronny Grefa	oficio	prueba de mensaje	Formularios De Proyecto.pdf	finalizado	987241397	fernando Basantes

Realizado por: Los autores

Par visualizar los documentos se lo realiza en archivo modules/stock_inventory/view.php.

Ilustración 58: Visualizar Documentos

```
$collection = $mongo->getCollection('documento');$cursor = $collection->find(array(), array('_id','numero_doc','fecha_doc','emisor','tipo_documento','descripcion_doc','direc_doc_adjunto','tipo_accion','telefono','usuario_que_recibio')); // while ($cursor->hasNext()): $article = $cursor->getNext(); echo "<tr><td width='50' class='center'>$no</td>";?> <td><?php echo substr($article['fecha_doc'], 0, 35); ?></td> <td><?php echo substr($article['numero_doc'], 0, 35); ?></td> <td><?php echo substr($article['emisor'], 0, 35); ?></td> <td><?php echo substr($article['tipo_documento'], 0, 35); ?></td> <td><?php echo substr($article['descripcion_doc'], 0, 35); ?></td><td><a target="_blank" href="modules/stock_inventory/visualizar_documento.php?id=<?php echo $article['direc_doc_adjunto']?>" > <?php echo substr($article['direc_doc_adjunto'], 0, 35); ?></a></td> <td><?php echo substr($article['tipo_accion'], 0, 35); ?></td> <td><?php echo substr($article['telefono'], 0, 35); ?></td> <td><?php echo substr($article['usuario_que_recibio'], 0, 35); ?></td> <?php $no++;endwhile;
```

Realizado por: Los autores

ANEXO X:

Plantilla ActiveXperts SMS

Es una API que proporciona la comunicación entre el modem y a la aplicación web.

Ilustración 59: Plantilla ActiveXperts SMS

```
<?php
session_start();
require_once "../config/dbconnection.php";
$mongo = DBConnection::instantiate();
$collection = $mongo->getCollection('registros_de_msj');
$collection1 = $mongo->getCollection('documento');
$article = array();
$article['emisor'] = $_POST['emisor'];
$article['telefono'] = $_POST['telefono'];
$article['mensaje'] = $_POST['mensaje'];
$article['id_user'] = $_POST['id_user'];
$article['saved_at'] = new MongoDate();
$status = array('$set' => array("tipo_accion" => "finalizado"));
$collection1->update(array('_id'=> new MongoId($id_user)), $status);
$collection->insert($article);
$objGsmOut = new COM ("ActiveXperts.GsmOut");
$archivo = 'D:\mensajes';
$dispositivo = 'HUAWAI Mobile Connect - 3G Modem';
$velocidad = 0;
$objGsmOut->LogFile = $archivo;
$objGsmOut->Device = $dispositivo;
$objGsmOut->DeviceSpeed = $velocidad;
$objGsmOut->MessageRecipient = $article['telefono'];
$objGsmOut->MessageData = $article['mensaje'];
if($objGsmOut->LastError == 0){
    $objGsmOut->Send;
```

Realizado por: Los autores

ANEXO XI:

Funcionamiento de la aplicación

Descripción del sistema

Objeto

El propósito de esta guía de usuario es facilitar al usuario una interacción con las distintas interfaces que se presenta en el aplicativo web GESTIÓN DOCUMENTAL CONAGOPARE CHIMBORAZO.

Alcance

Esta guía está orientada a los usuarios que deseen aprender a utilizar el aplicativo web, de otro modo también para quienes ya hayan tenido un contacto con sistema y que por algún motivo olvidaron el funcionamiento del mismo.

Requerimientos

a) Requerimientos de hardware

Se deberá contar con:

- Estación de Trabajo (Computadora Personal)
- Conexión Local (Internet NO indispensable)

b) Requerimientos de Software

Se deberá contar con:

- Cualquier Sistema Operativo.
- Navegador Web.

Contacto con el sistema

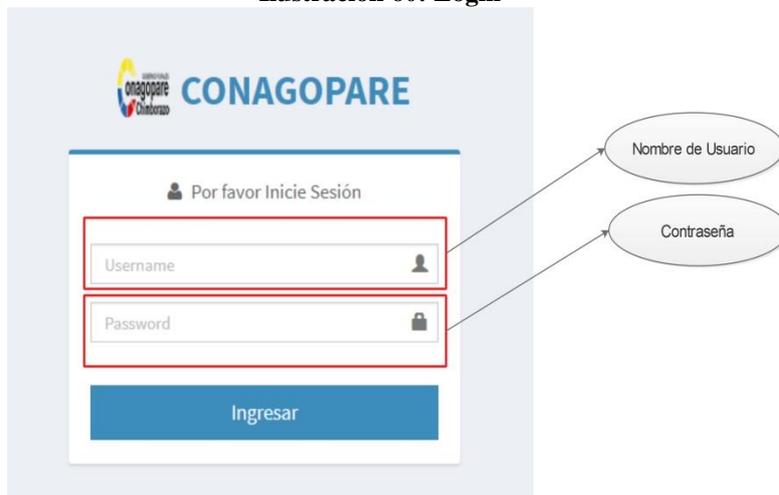
Ingreso al Sistema

Para acceder al sistema se debe ejecutar el navegador de su preferencia y en el apartado de URL, pegar el siguiente enlace:

<localhost/CONAGOPARE/index.php>

Se verá la siguiente interfaz donde debemos proporcionar las credenciales, dependiendo el tipo de usuario se accederá a los módulos de la aplicación.

Ilustración 60: Login



Realizado por: Los autores

Al iniciar sesión se presenta la siguiente interfaz, donde el usuario podrá desplazarse por los menús.

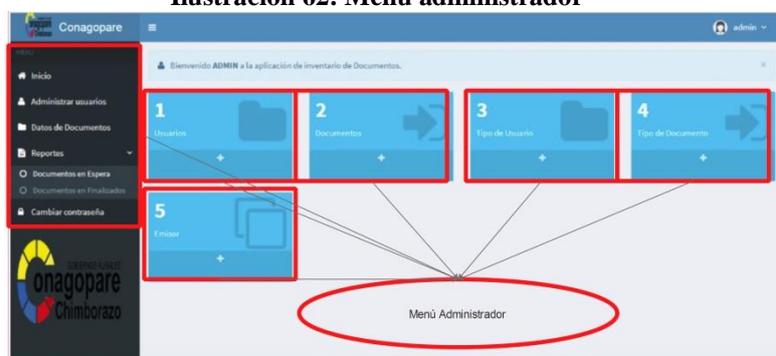
Ilustración 61: Menú



Realizado por: Los autores

Menú del Aplicativo

Ilustración 62: Menú administrador



Realizado por: Los autores

Opciones de la barra de menú:

- En el menú **Usuarios** se podrá:
 - Agregar, Eliminar, Actualizar cualquier usuario.
- En el menú **Documentos** se podrá:
 - Registrar un documento proveniente de cualquiera de las juntas parroquiales.
- En el menú **Tipo de Usuario** se podrá:
 - Visualizar aquellos usuarios registrados, y se podrá modificar el rol.
- En el menú **Tipo de Documento** se podrá:
 - Agregar otra división de documentos.
- En el menú **Emisor** se podrá:
 - Visualizar el remitente de los documentos registrados.

Ilustración 63: Menú ejecutivo



Realizado por: Los autores

Opciones de la barra de menú:

- En el menú **Documentos** se podrá:
 - Registrar un documento proveniente de cualquiera de las juntas parroquiales.
- En el menú **Documentos en Espera** se podrá:
 - Visualizar aquellos documentos que fueron registrados y que están a la espera de la firma para el correspondiente trámite que soliciten las juntas parroquiales.
 - Cambiar el estado del documento.
- En el menú **Documentos Finalizados** se podrá:
 - Visualizar aquellos documentos que estén en estado de entrega, los mismos que son traspasados del menú *documentos en espera*.
- En el menú **Emisor** se podrá:
 - Visualizar el remitente de los documentos registrados.

Comenzar a trabajar

Se puede empezar a trabajar con el aplicativo de la siguiente manera:

- Iniciar la sesión dependiendo del rol que desempeñe en la organización.
- Si es administrador, podrá realizar cambios dentro de los siguientes menús: usuarios, tipo de usuarios, documentos, tipo de documentos y emisor.
- Si es ejecutivo, solamente tendrá acceso a los siguientes menús: documentos, documentos en espera, documentos finalizados y emisor.

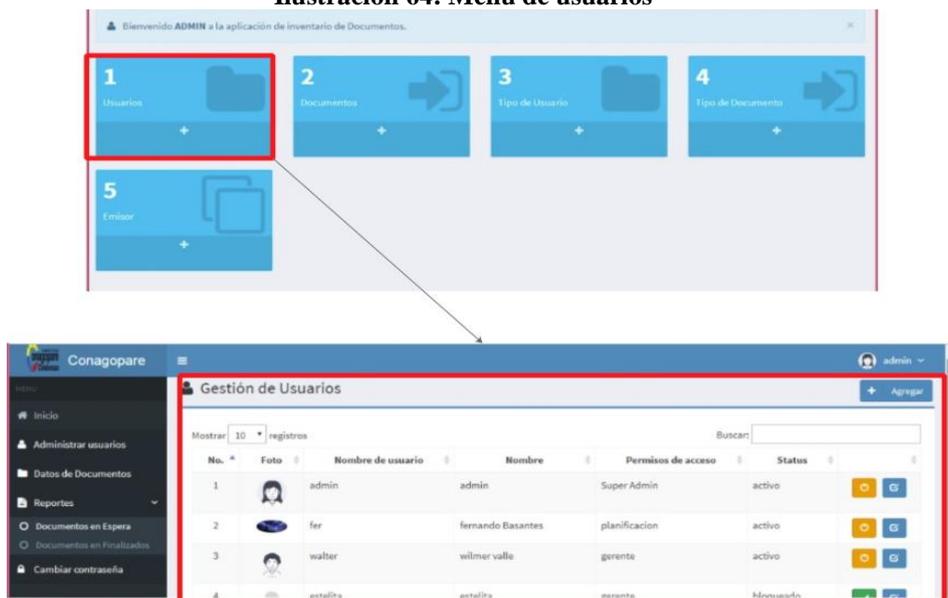
Interfaces de cada menú

A continuación se detalla las interfaces de cada menú correspondiente al tipo de usuario.

Menú Usuarios – Administrador

En la siguiente ilustración se ingresa al menú de Usuarios, donde se podrá ver todos los usuarios actualmente registrados y el rol que desempeñan dentro del sistema, si desea agregar un nuevo usuario, debe seleccionar la opción *agregar*.

Ilustración 64: Menú de usuarios

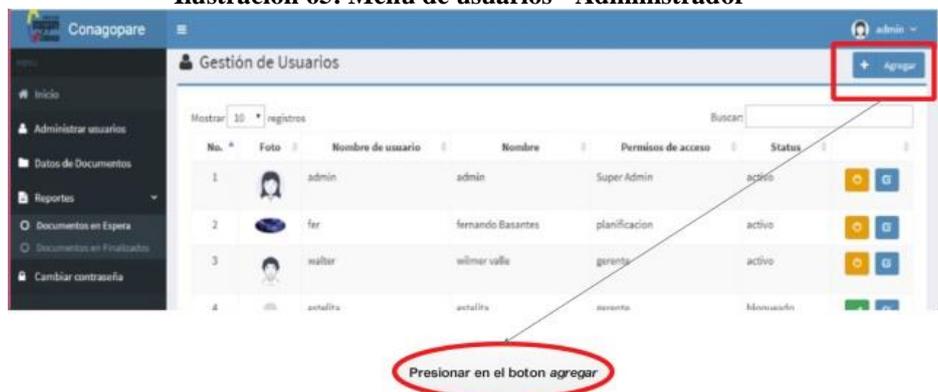


Realizado por: Los autores

Menú usuarios (agregar usuario) – Administrador

En la siguiente ventana se selecciona el botón *agregar* se muestra en la siguiente figura:

Ilustración 65: Menú de usuarios - Administrador



Realizado por: Los autores

Aparecerá la siguiente ventana donde se deberá llenar con los datos que se solicita.

Ilustración 66: Ventana para agregar usuarios

Nombre

Telefono

Direccion

Correo

Realizado por: Los autores

Para modificar los datos de usuario se deberá seleccionar el botón de editar.

Ilustración 67: Modificar datos del usuario

Mostrar 10 registros

Buscar:

No.	Foto	Nombre de usuario	Nombre	Permisos de acceso	Status	
1		admin	admin	Super Admin	activo	<input type="button" value="Editar"/>
2		fer	fernando Basantes	planificación	activo	<input type="button" value="Editar"/>

Realizado por: Los autores

Aparecerá la siguiente ventana, donde se deberá realizar los cambios pertinentes.

Ilustración 68: Modificar datos de usuario

Conagopare

Inicio - Usuario - Modificar

Modificar datos de Usuario

Nombre de Usuario:

Nombre:

Email:

Telefono:

Foto: Ningún archivo seleccionado

Permisos de acceso:

Se realizará los cambios necesarios

Realizado por: Los autores

Documentos

Para ingresar documentos deberemos dar clic en el menú Documentos, y aparecerá la siguiente ventana donde se llenará los campos requeridos.

Ilustración 69: Ventana para agregar documento

Conagopare

Inicio - Documentos - Agregar

Agregar Documento

Numero del Documento:

Fecha:

Emisor:

Junta Parroquial:

Telefono:

Descripción:

Doc Adjunto: Ningún archivo seleccionado

Tipo de Documento:

Tipo de Acción:

Se llenará los campos requeridos

Realizado por: Los autores

Para visualizar los documentos en espera se deberá ingresar al menú Documentos en Espera, donde se podrá visualizar todos los documentos ingresados y que están en proceso de revisión.

Ilustración 70: Visualizar documentos

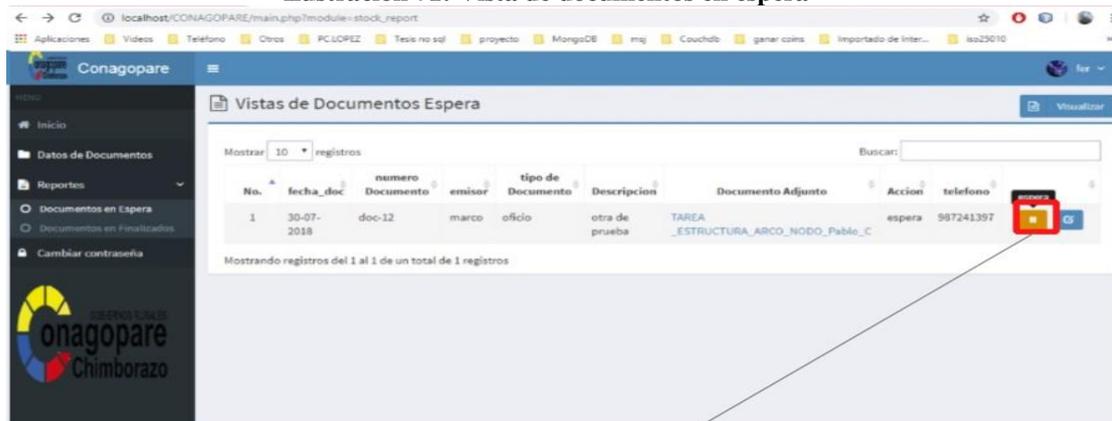
No.	fecha_doc	numero Documento	emisor	tipo de Documento	Descripción	Documento Adjunto	tipo de Accion	telefono	usuario que recibio
1	30-05-2018	doc-01	pablo	oficio	ni idea q sea esto	c:/doc/document/ddds.pdf	espera	0988247119	rosa
2	30-05-2018	doc-1542	pedro	Informe	manian welcome to the club	video chas	espera	0988247119	rosa
3	15-07-2018	col	col	Informe	col	Carta de Aceptacion.pdf	espera	0918277878	rosa
4	23-07-2018	sdsdfs	1212	oficio	dsdsd	T4Apuntes.pdf	espera	976565	rosa
5	30-07-2018	doc-12	marco	oficio	otra de prueba	TAREA _ESTRUCTURA_ARCO_NODO_Pablo_C	espera	967241397	fernando Basantes
6	2019,03,15	doc-232431	cordova	oficio	nadess	1483 Texto del articulo-6932-1-10-	espera	0985462323	admin
7	10-03-2019	dfcdcd	polutyrt	oficio	xcdcd	Urkund Report - Informe Final Tesis	espera	0987678645	admin

Se visualiza todos los documentos en espera.

Realizado por: Los autores

Una vez que se verifique que el documento esta revisado y gestionado, se procede a cambiar el estado del documento para ello en la misma ventana seleccionamos el icono de estado para ponerlo en finalizar.

Ilustración 71: Vista de documentos en espera



Una vez que demos clic en ese boton cambiara de estado a **Finalizado**

Realizado por: Los autores

Por último se genera la notificación, eso sucede cuando se realiza el cambio de estado del documento de espera a finalizado, nos aparecerá una ventana emergente con lo siguiente:

ANEXO XII:

Evaluación de la Caja Negra

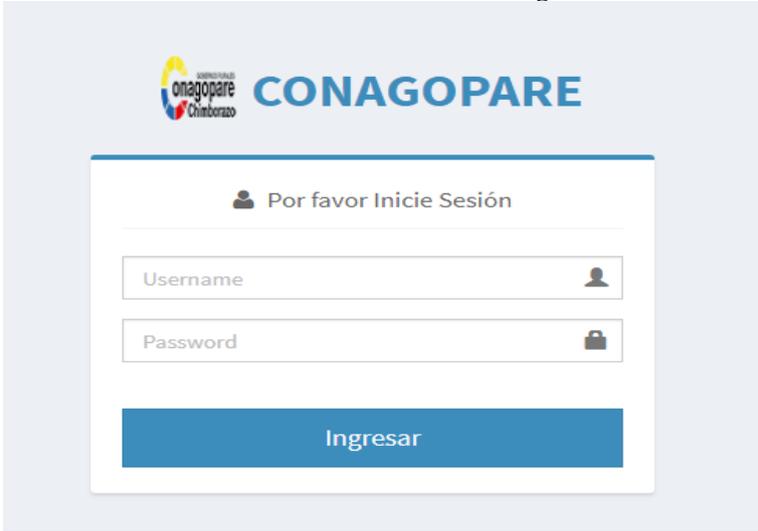
- **Inicio de sesión.**

Tabla 28: Login

DESCRIPCIÓN PRUEBA CAJA NEGRA	
Datos de Entrada: Login – Usuario y Password	Resultado: Acceso al sistema
Tipo de flujo de datos Formulario Interno	
Detalle: La información ingresada de Usuario y Password se validara los campos en formulario y en la bases de datos NoSQL.	

Realizado por: Los autores

Ilustración 72: Formulario Login



Realizado por: Los autores

- **Registro de Documentos**

Tabla 29: Registro de documentos

DESCRIPCIÓN PRUEBA CAJA NEGRA	
Datos de Entrada: numero_doc - fecha_doc – emisor – junta_parroquial – tipo_documento- descripción_doc – direc_doc_adjunto – tipo_accion – telefno – usuario_que_Recibio	Resultado: Registro de los documentos
Tipo de flujo de datos Formulario Interno Archivo	
Detalle: La información ingresada en el formulario será validada y registro de la información del documento.	

Ilustración 73: Registro de documentos

Realizado por: Los autores

- **Registro de Envió de Mensajes**

Tabla 30: Registro de envío de mensajes
DESCRIPCIÓN PRUEBA CAJA NEGRA

Datos de Entrada: emisor – teléfono – mensaje – id_user – saved_at	Resultado: Registro y envío de mensaje de texto
Tipo de flujo de datos Formulario Interno Archivo	
Detalle: la información ingresada en el formulario se envía al emisor del documento, notificando la resolución de proceso de notificación.	

Realizado por: Los autores

Ilustración 74: Registro de Envió de Mensajes

Realizado por: Los autores