

Análisis de Desempeño entre MONGODB y COUCHDB utilizando Norma ISO/IEC 25000

Performance Analysis between MONGODB and COUCHDB using ISO/IEC 25000 Standard

Estela Narváez*, Pablo Calapucha†, Marco Tarco‡, Pamela Buñay§
 Universidad Nacional de Chimborazo, 060108, Riobamba, Ecuador

Email: *miryan.narvaez@unach.edu.ec, †pcalapucha.fis@unach.edu.ec, ‡mtarco.fis@unach.edu.ec, §pbunay@unach.edu.ec

Resumen— Las bases de datos NoSQL han surgido para dar respuesta a problemas de escalabilidad y rendimiento, que en general, las bases de datos relacionales no pueden abarcar. La gran flexibilidad y las posibilidades de optimización en sus diseños las convierten en una atractiva variante a tener en cuenta para el desarrollo de aplicaciones de gestión de información. El objetivo principal del proyecto de investigación fue un estudio comparativo entre MongoDB y CouchDB utilizando la norma ISO/IEC 25010, con el fin de analizar y medir el desempeño de los gestores de base de datos NoSQL, en cuanto al consumo de recursos utilizados. Como resultado de la investigación se obtuvo que MongoDB fue ligeramente superior a CouchDB, demostrando que las bases de datos NoSQL tienen mejor rendimiento al momento de administrar grandes volúmenes de datos.

Palabras Clave— Base de datos NoSQL, MongoDB, CouchDB, ISO/IEC 25010, JMeter.

Abstract— The NoSQL databases have emerged to respond to problems of scalability and performance, those which in general, relational databases cannot cover. The great flexibility and the optimization possibilities in their designs make them an attractive variant to take into account when developing information management applications. The main objective of the research project was to carry out a comparative study between MongoDB and CouchDB using the ISO/IEC 25010 standard, in order to analyze and measure the performance of the NoSQL database managers, regarding the consumption of resources used. The result revealed that MongoDB was slightly higher than CouchDB, demonstrating that NoSQL databases have better performance when managing large volumes of data.

Keywords— Data Base NoSQL, MongoDB, CouchDB, ISO/IEC 25010, JMeter.

I. INTRODUCCIÓN

La respuesta a la necesidad de gestionar volúmenes masivos de información proviene de las bases de datos no relacionales NoSQL (Not only SQL). El término no es nada nuevo, pues fue acuñado a finales de los 90s que engloba las tecnologías que rompen el esquema relacional y que no implementan el lenguaje estructurado de consultas SQL (Structured Query Language). Este tipo de base de datos NoSQL surgen como una solución a los constantes requerimientos de procesamiento y análisis a gran escala de enormes cantidades de datos, para

los cuales los sistemas tradicionales de base de datos son insuficientes. [6], [17]

Existe un sin número de gestores de base de datos relacionales y no relacionales que son utilizados indistintamente para almacenar grandes cantidades de datos, los mismos que se transforman en información indispensable para la institución. En la actualidad el modelo no relacional de base de datos está predominando por que permite un alto grado de escalabilidad y con menor latencia en cuanto al acceso de datos. [1]

La aparición de las bases de datos NoSQL inicia con la web 2.0 y las compañías de Internet reconocidas a nivel mundial como: Facebook, Twitter, LinkedIn, Google, Youtube y Amazon, aplicaciones donde cualquier usuario podía subir contenido, provocando así un crecimiento exponencial de los datos. [9]

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. [17]

Morejon 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, dentro de las cuales se encuentran. Cassandra, Redis, MongoDB, CouchDB, Neo4J, HBase, entre otras.

Cassandra es escalable, basada en dos esquemas de almacenamiento columnar y clave-valor. Redis un motor de base de datos en memoria, almacena información en conjuntos Clave-Valor. MongoDB trabaja de forma nativa y permite el almacenamiento de datos en forma de documentos BSON. CouchDB trabaja mediante http como interfaz y trabaja con documentos JSON. Neo4J es ágil, flexible y altamente escalable, tiene un funcionamiento transaccional que almacena en estructura de grafos. HBase es una base de datos poderosa, combina capacidades de consulta en tiempo real con la velocidad de un almacén de clave/valor y procesamiento fuera de línea o por lotes a través de MapReduce. [6], [15]

Las bases de datos que predominan dentro de la familia

NoSQL son MongoDB, CouchDB, Cassandra y HBase. BD-Engines posiciona a MongoDB en el quinto puesto, Cassandra en décimo primer, HBase en el vigésimo segundo, mientras que CouchDB ocupa el trigésimo cuarto puesto, sin embargo MongoDB y CouchDB son las dos bases de datos que han sido pioneros, por lo tanto son populares y más utilizadas. [19]

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

- Clave-valor
- Columnas
- Grafos
- Documentos

Clave-valor: utiliza un método simple de clave-valor para almacenar datos, cada dato está identificado por una clave única, lo que permite la recuperación de la información rápidamente. Entre las bases de datos más reconocidas está Redis.

Columnas: los valores se almacenan en columnas en lugar de filas. Son útiles cuando se gestionan datos agregados. Las bases de datos más populares son HBase y Cassandra.

Grafos: los datos se almacenan en estructuras grafo para almacenar, correlacionar y consultar relaciones, está compuesto por bordes y nodos. La base de datos más reconocida es Neo4j.

Documentos: almacena la información como un documento, utilizando una estructura simple como JSON, BSON y XML usa una clave única para cada registro. Entre las bases de datos documentales más destacadas están MongoDB y CouchDB. [8]

MongoDB es una base de dato NoSQL multiplataforma de código abierto, está desarrollado en C++. Los distintos documentos en MongoDB se almacenan en formato *JSON* (JavaScript Object Notation, es un formato ligero de intercambio de datos) con un esquema dinámico como *BSON* (Binary JSON) compuestos de: nombre del campo, tipo de dato y valor, cada registro se denomina documento que puede ser agrupado en colecciones (tablas) similares a una base de datos relacionales, el programador tiene facilidad de adecuarlo según las necesidades. MongoDB funciona sin problema en sistemas operativos como: Windows, Linux, OS X y Solaris. [16], [5]

CouchDB es un gestor de bases de datos no relacional, aplicación Open Source que está escrito en Erlang (lenguaje de programación funcional robusto que permite un diseño flexible y fácilmente escalable y extensible), emplea *JSON* para almacenar los datos en forma de documentos, 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). Los sistemas operativos para los cuales esta disponible CouchDB son: Android, BSD, iOS, OS X, Solaris y Windows. [8]

Se detalla cómo funcionan cada una de las bases de datos (MongoDB, CouchDB), cuáles son sus principales características y ventajas. A continuación, se presenta un pequeño ejemplo en formato JSON (Fig. 1).

```
{
  "_id": 12345,
  "Nombre": "Maria",
  "Apellido": "Reyes",
  "Edad": 30,
  "Semestre": 10,
}
```

Figura 1. Ejemplo Base de Datos Documental

A. Características de MongoDB y CouchDB

Se realizó un cuadro comparativo entre los dos motores de bases de datos, con el propósito de determinar sus ventajas y desventajas, demostrando que las dos bases de datos tienen características similares, se presenta en la Tabla I.

Tabla I
MONGODB VS COUCHDB

Características	MongoDB	CouchDB
Desarrollador	MongoDB, Inc	Fundación Apache Software
Lenguaje	C++	Erlang
Modelo de datos	Orientado a documentos (BSON)	Orientado a documentos (JSON)
Interfaz	Usa el protocolo binario y TCP/IP	HTTP / RESTful
Objeto	Contiene colecciones	Contiene documentos
Método de consulta	MapReduce, lenguaje de consulta basado en objetos	MapReduce
Preferencias	Favorece la consistencia	Favorece la disponibilidad

El análisis de desempeño de las bases de datos MongoDB y CouchDB se basó en la norma ISO/IEC 25000. La Norma ISO/IEC 25000, es el resultado de la evolución de otras normas anteriores, especialmente de las normas ISO/IEC 9126. Es una familia de mormas que tiene como objetivo evaluar la calidad del software, tiene cinco 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

ISO/IEC 2500n – División de Gestión de Calidad, definen todos los modelos, términos y definiciones comunes referenciados por todas las otras normas de la familia 25000. Esta división se encuentra formada por ISO/IEC 25000 y ISO/IEC 25001.

ISO/IEC 2501n - División de Modelo de Calidad, 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 y ISO/IEC 25012.

La norma ISO/IEC 25010 se encuentra compuesto por ocho características de calidad (Adecuación Funcional, Eficiencia de desempeño, Compatibilidad, Usabilidad, Fiabilidad, Seguridad, Mantenibilidad, Portabilidad) y cada una de estas varias

Tabla II
CALIDAD DEL PRODUCTO SOFTWARE

CARACTERÍSTICAS	SUBCARACTERÍSTICAS
Adecuación Funcional	Completitud funcional Corrección funcional Pertinencia funcional
Eficiencia de desempeño	Comportamiento temporal Utilización de recursos Capacidad
Compatibilidad	Coexistencia Interoperabilidad
Usabilidad	Capacidad para reconocer su adecuación Capacidad de aprendizaje Capacidad para ser usado Protección contra errores de usuario Estética de la interfaz de usuario Accesibilidad
Fiabilidad	Madurez Disponibilidad Tolerancia a fallos Capacidad de recuperación
Seguridad	Confidencialidad Integridad No repudio Responsabilidad Autenticidad
Mantenibilidad	Modularidad Reusabilidad Analizabilidad Capacidad para ser modificado Capacidad para ser probado
Portabilidad	Adaptabilidad Capacidad para ser instalado Capacidad para ser reemplazado

subcaracterísticas. Según la norma ISO/IEC 25010 la Tabla II muestra las características y subcaracterísticas que debe tener un software para cumplir con los estándares establecidos y satisfacer las necesidades del usuario. [10]

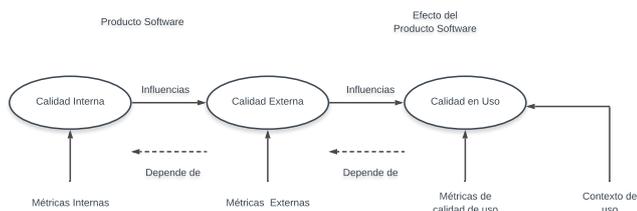


Figura 2. Modelo de calidad ISO/IEC 25010

La Fig. 2 muestra la relación entre la calidad del proceso, la calidad del producto y la calidad en uso, describe un modelo bipartito para la calidad del producto de software, siendo sus modelos principales: Calidad interna, Calidad externa y Calidad en el uso. La calidad interna se relaciona con las características estáticas del software; la calidad externa trata las características relacionadas con la ejecución del software; la calidad en uso considera la calidad del sistema en su ambiente operacional para usuarios específicos que realizan tareas específicas.

ISO/IEC 2502n - División de Medición de Calidad, estas

normas incluyen un modelo de referencia de calidad del software, con mediciones internas, externas y en uso. Se encuentra formada por cinco estándares: ISO/IEC 25020, ISO/IEC 25021, ISO/IEC 25022, ISO/IEC 25023 y ISO/IEC 25024.

En las últimas décadas la gestión de documentos ha evolucionado debido al auge de la tecnología y la cantidad de información generada. Un sistema de gestión documental asegura una información organizada y eficiente, que favorezca la productividad empresarial. Las organizaciones optan en buscar soluciones para desmaterializar los documentos. [14]

La gestión documental en las organizaciones públicas y privadas no responde de forma adecuada a las necesidades que se presentan según su actividad, gran parte de organizaciones realizan los procesos de forma manual. La documentación generada ha sido el mayor problema en estas instituciones por la ausencia de espacio físico para conservar dichos documentos 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. [7]

Los sistemas informáticos han 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 Ecuador 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 el seguimiento de los documentos; comprometiendo a las empresas a una escasa eficiencia en los procesos de resolución de documentos. [2]

El objetivo de la investigación fue comprobar 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 (software de código abierto, diseñado para cargar el comportamiento funcional de las pruebas y medir el rendimiento) para obtener los criterios: comportamiento de tiempos y utilización de recursos hardware (Procesador, Memoria RAM y Disco Duro), con los resultados obtenidos, se determinó cual de las dos bases de datos NoSQL (MongoDB y CouchDB) ocupa menos recursos para implementar en el aplicativo web GESTIÓN DOCUMENTAL CONAGOPARE CHIMBORAZO.

Para el desarrollo del Aplicativo Web se utilizó Programación Extrema o eXtreme Programming (XP) por ser una metodología ligera y flexible de desarrollo de software, se basa en la simplicidad, la comunicación y la realimentación del código, el objetivo principal es la satisfacción del cliente, consta de 4 fases (Fig. 3). [18]

- Planificación: esta primera fase se encarga de la recopilación de todos los requerimientos del proyecto.
- Diseño: debe ser simple y sencillo.
- Codificación: donde se plasma todas ideas a través del código.

- Pruebas: se implementa un test que son pruebas que se le hacen al proyecto para detectar y corregir errores.



Figura 3. Proceso de la Programación Extrema [18]

Las relaciones desarrollador-cliente y desarrolladores-desarrollador son fundamentales en esta metodología. La adopción del cliente como un miembro más del equipo de desarrollo es la clave del éxito. XP propone varios roles, se enumeran los más importantes: Programador, Cliente, Encargado de pruebas, Encargado de seguimiento, Entrenador, Consultor y Gestor. [3]

CONAGOPARE (Consejo Nacional de Gobiernos Parroquiales Rurales del Ecuador) CHIMBORAZO no dispone de un sistema de gestión de documentos. Esta problemática impide una rápida respuesta a los documentos que diariamente ingresan a la organización provenientes de las juntas parroquiales. Para dar respuesta al problema, se propuso desarrollar un aplicativo web GESTIÓN DOCUMENTAL CONAGOPARE CHIMBORAZO con el fin de dar seguimiento a los documentos que ingresan a secretaria, hasta el proceso de resolución del mismo; posteriormente, este resultado es notificado al solicitante mediante mensajería de texto que el documento fue gestionado.

El sistema tiene la capacidad de respaldar y almacenar la documentación digitalmente para cualquier proceso de auditoría de CONAGOPARE CHIMBORAZO.

II. METODOLOGÍA

La investigación se ajusta a un enfoque cuantitativo de tipo experimental. Se analizan los datos obtenidos de las pruebas realizadas a las dos bases de datos NoSQL (MongoDB y CouchDB) bajo las Normas ISO/IEC 25010/25023 para medir el desempeño de las mismas. Las pruebas se realizaron utilizando la herramienta informática *Apache JMeter*, instrumento electrónico que sirvió como método de recolección de los datos y para evaluar los resultados se utilizó la herramienta estadística *IBM SPSS 25*, permitiendo identificar cuál de las dos bases de datos es más eficiente y optimiza recursos.

Para analizar el desempeño de las bases de datos MongoDB y CouchDB se utilizó dos normas ISO/IEC 25010 y ISO/IEC 25023.

Tabla III
PONDERACIÓN DE MÉTRICAS DE EVALUACIÓN

Características	Subcaracterísticas	Nivel de Importancia
Eficiencia de Desempeño	Comportamiento Temporal	Alto
	Utilización de Recursos	Alto
Usabilidad	Capacidad para Recocer su Adecuación	Medio
	Estética de la Interfaz de Usuario	Medio

Norma ISO/IEC 25010 determina las características de calidad del producto de software que se van a tener en cuenta a la hora de evaluar. La calidad del producto software se puede interpretar como el grado en que el producto satisface los requisitos de sus usuarios aportando de esta manera un valor. Los requisitos (funcionalidad, rendimiento, seguridad, mantenibilidad, etc.) son precisamente los que se encuentran representados en el modelo de calidad, el cual categoriza la calidad del producto en características y subcaracterísticas (Tabla II). [10]

Norma ISO/IEC 25023 define específicamente las métricas para realizar la medición de la calidad de productos y sistemas software. 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.

Para establecer la cantidad de consultas que se va a utilizar se aplicó la fórmula de población infinita determinando el tamaño de muestra, y con la ayuda del software estadístico *IBM SPSS 25* se efectuó el análisis Mann Whitney. La prueba U de Mann Whitney compara los criterios de las muestras de MongoDB y CouchDB que son dos muestras independientes con una variable cuantitativa y determinar si las medianas de estas dos muestras difieren.

A. Criterios de Desempeño de las Bases de Datos NoSQL

Para evaluar del desempeño de las bases de datos NoSQL MongoDB y CouchDB 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 25010. La Tabla III presenta, las métricas de evaluación sujetos a los criterios del desarrollador.

- Se especifica el tipo de base de datos NoSQL a evaluar (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 ecuación 1 para medir la eficiencia de desempeño de la población de las bases de datos NoSQL:

$$n = \frac{Z^2 * p * q}{e^2} \tag{1}$$

Donde, n = Muestra, p = Probabilidad de éxito, q = Probabilidad de fracaso, Z = Nivel de confianza 95%, e = Error de muestra 5%.

- Se describe el proceso del plan de pruebas para medir los recursos y tiempo de respuesta, con una muestra de 353 sentencias en un tiempo estimado de 6 minutos.
- Ingresar los datos obtenidos en la matriz de calidad (Tabla IV).

B. Herramientas utilizadas

- ISO/IEC 25010: Modelo de Calidad de Software.
- ISO/IEC 25023: Medición de la Calidad del Producto Software y Sistemas. [10]
- Apache JMeter: Herramienta de testing para analizar y medir el rendimiento de las bases de datos. [11]
- IBM SPSS 25: Software estadístico utilizado para resolver problemas mediante análisis ad-hoc, pruebas de hipótesis y análisis predictivo. [20]
- Netbeans: Es un entorno de desarrollo integrado que permite escribir, compilar, depurar y ejecutar programas en diversos lenguajes. [13]
- Xampp: Es un servidor independiente multiplataforma, de software libre, fácil de usar y capaz de interpretar páginas dinámicas. [21]
- ActiveXperts SMS Component: Es una interfaz (API) rápida, confiable y fácil de usar para enviar y recibir mensajes SMS y Pager, a través de SMPP, módem GSM, HTTP o proveedor de acceso telefónico TAP / XIO. [12]

C. Metodología de Desarrollo

Se adoptó la metodología XP para el desarrollo del aplicativo, por ser eficiente, flexible y predecible; plantea de forma clara el proceso metodológico a seguir para la construcción del producto. XP propone un desarrollo ágil, disciplinado con soluciones sencillas y con un enfoque adaptativo. Se describe en detalle cada una de las fases de desarrollo:

- Fase I planificación: comienza la interacción entre el cliente y el grupo de desarrolladores para definir los requerimientos iniciales del aplicativo, utilizando las historias de usuario. En esta fase se tomó en cuenta cuatro valores (Comunicación, Sencillez, Realimentación e Historias de usuario) que al utilizarlos facilitó la especificación de requerimientos y plazos iniciales de entrega. El Objetivo de esta fase es que el equipo de desarrollo se familiarice con la metodología, los recursos físicos y tecnológicos necesarios para el desarrollo del proyecto. Se contó con la participación del cliente para el levantamiento de la información dentro y fuera de la empresa.
- Fase II diseño: con la información obtenida, se procede al análisis de la misma y al diseño de la estructura del sistema, se realiza el modelamiento de la base de datos donde se refleja los requerimientos funcionales y prototipos de las interfaces de usuario, descartando posibles problemas (técnico, arquitectura del software, tiempo) que puedan generarse durante el desarrollo del mismo. Dando prioridad a una eficiente programación, creando el código necesario y verificando que realice su trabajo correctamente. La metodología XP hace énfasis en los diseños simples y claros.

- Fase III desarrollo: Una vez aceptados los prototipos por el cliente, se procede al desarrollo, esta fase atiende los requerimientos de forma óptima, se entrega al cliente la última versión del aplicativo, el mismo que es contrastado con las historias de usuario inicial para pasar la prueba de unidad, logrando con el objetivo para el cual fue diseñado. Uno de los requerimientos de XP es tener al cliente disponible durante todo el proyecto.
- Fase IV pruebas: se realizaron pruebas de integración y validación diarias, esto proporcionó al equipo un indicador del progreso, revelando a tiempo si existe alguna falla en el sistema. Concluyendo el desarrollo con pruebas y ajustes finales al aplicativo.

D. Especificación de Requerimientos

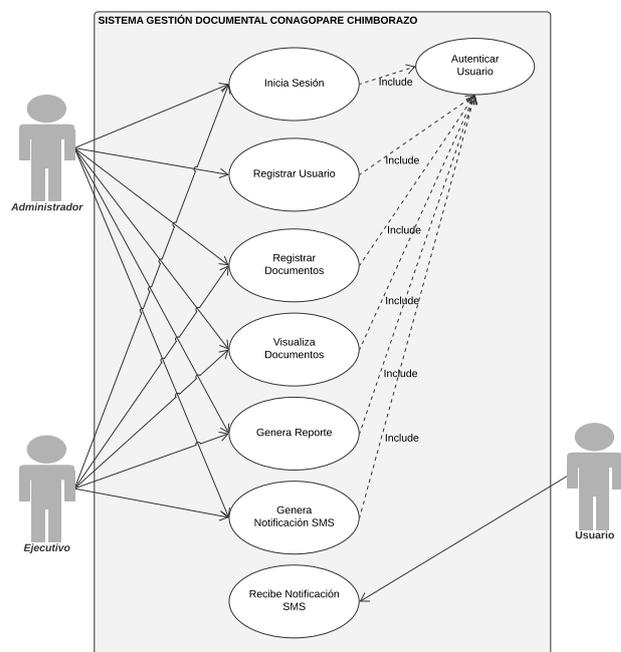


Figura 4. Diagrama de actividades

La Fig. 4 muestra el diagrama de actividades, representando los requerimientos que debe cumplir el aplicativo web para la Gestión de documentos CONAGOPARE Chimborazo, considerando que todas las actividades serán soportadas por el aplicativo. El diagrama contiene las historias de usuario más importante desde el punto de vista de la arquitectura del sistema, las cuales se usaron para llevar a cabo el desarrollo del sistema. En el diagrama se observa que los agentes (Administrador y Ejecutivo) están asociados a los roles con los que pueden colaborar mientras que el Usuario simplemente participa en el caso de uso asociado.

III. CONSTRUCCIÓN DEL SISTEMA

El Aplicativo Web de Gestión de Documentos utiliza uno de los patrones más utilizados en la actualidad para el desarrollo de interfaces de usuario enriquecidas, como es el patrón

Tabla IV
MATRIZ DE CALIDAD NORMA ISO/IEC 25000

Software	Características	Subcaracterísticas	Métricas	Fórmula	Calidad del Sw	Nivel de Importancia	Valor Obtenido	Parcial en %	Valor con respecto al 100%	Calidad del Sistema	Grado de Satisfacción
MongoDB	Eficiencia de Desempeño	Comportamiento Temporal	Tiempo de Respuesta				0,0747s	0,02%	99,98%	94,36%	Muy Satisfactorio
		Utilización de Recursos	Procesador CPU	$n = \frac{Z^2 \cdot s \cdot p \cdot q}{c^2}$	Externa	Alto	5,07%	11,26%	88,74%		
		Memoria RAM	0,26%								
		Disco Duro			0,007s	0,002%	99,99%				
CouchDB	Eficiencia de Desempeño	Comportamiento Temporal	Tiempo de Respuesta				14,52%	18,30%	90,84%	90,84%	Muy Satisfactorio
		Utilización de Recursos	Procesador CPU	$n = \frac{Z^2 \cdot s \cdot p \cdot q}{c^2}$	Externa	Alto	40,20%				
		Memoria RAM	0,19%								
		Disco Duro									

de diseño Modelo Vista Controlador (MVC). Se utilizó el patrón MVC en PHP por ser flexible, robusto y escalable en aplicaciones web. Proporciona al usuario una mayor facilidad a la hora de el desarrollo, su estructura es bien definida para dar soporte a un proyecto web ayudando a que el proyecto sea organizado y bien desarrollado. Permite crear aplicaciones interactivas las cuales se ejecutan desde el lado del cliente, ósea en el navegador. El MVC es el patrón más usado en la ingeniería de software. Este patrón separa el modelo de datos, el modelo de la capa de presentación (vista) y la parte de control, como presenta la Fig. 5. Modelo: es el encargado de representar los datos de la aplicación y la lógica para recuperar y persistir datos. Vista: se encarga de la parte visual de la aplicación. Controlador: es el responsable de controlar la interacción del usuario y luego actualizar el modelo o la vista. [22]

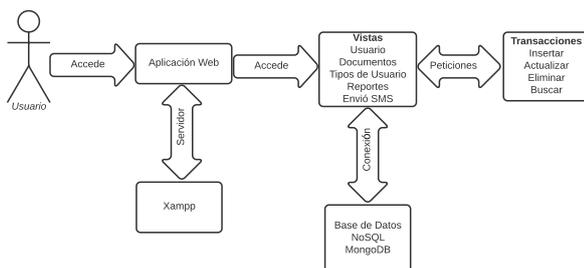


Figura 5. Arquitectura del Aplicativo Web

A. Diseño de la base de datos

MongoDB no maneja estándares gráficos, para el diseño de la base de datos, se utilizó el software Robo 3T (es una herramienta de administración de MongoDB multiplataforma centrada en shell, su principal objetivo es ejecutar consultas, crear índices y visualizar documentos), porque permite entender el manejo de los niveles de documentación de la base de

datos en MongoDB. En la base de datos se almacena todo el diccionario de los datos recolectados, se crea 8 colecciones, que exhibe la Fig. 6.

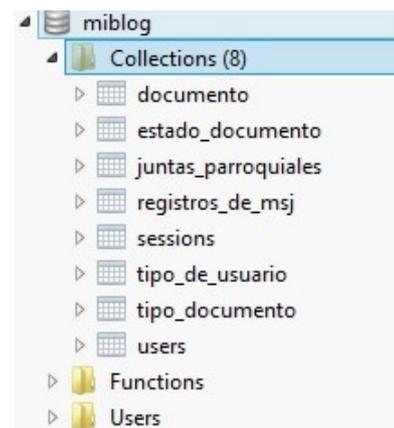


Figura 6. Colecciones MongoDB

Las consultas en la base de datos MongoDB son rápidas ahorrando recursos de procesamiento debido a la forma en que se almacenan los datos.

IV. RESULTADOS

Para realizar las pruebas de rendimiento de las bases de datos NoSQL MongoDB y CouchDB se utilizó un equipo portátil de las siguientes características, se muestran en la Tabla V.

Tabla V
CARACTERÍSTICAS DEL EQUIPO

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

A. Población y muestra

La muestra de estudio considerada corresponde al número de consultas que se realiza en cada una de las bases de datos (MongoDB y CouchDB), se ve necesario el uso de la fórmula población infinita porque no se conoce el número de elementos que la integran y se necesita tiempo y recursos para poder obtener información en un tiempo razonable.

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

$$n = \frac{(1.88)^2 * 0.5 * 0.5}{(0.005)^2}$$

$$n = 353.44 \cong 353$$

El valor de n obtenido por esta fórmula indica el tamaño de la muestra, se estableció el tamaño del error admisible en la estimación 5% con un nivel de confianza del 95%. El tamaño muestral resultante fue 353 sentencias para el estudio.

Se configuró el software Apache JMeter con una carga de 353 sentencias para 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) para medir el rendimiento del procesador, memoria RAM y Disco Duro en porcentaje. Los resultados se exportaron en un archivo con extensión (.csv).

Para demostrar las diferencias que existen entre las dos bases de datos se utilizó *U de Mann Whitney*, software estadístico que permite contrastar los criterios de las muestras independientes de MongoDB y CouchDB. Los resultados comprobaron que no existe una diferencia significativa en los criterios analizados de las bases de datos NoSQL MongoDB y CouchDB. [4]

B. Medición de Indicadores

Para el análisis de los datos recopilados se utilizó el software estadístico IBM SPSS 25, creando tablas y gráficas para presentar los resultados obtenidos, como se muestran en la Tabla VI y Fig. 7.

Tabla VI
TIEMPO DE RESPUESTA 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%

C. Eficiencia de Desempeño

La Fig. 8 presenta la suma del porcentaje general de los recursos (CPU, Memoria RAM, y Disco Duro); MongoDB tiene un porcentaje de desempeño de un 94,36%, mientras que CouchDB con un 90,84%, concluyendo que MongoDB es eficiente al optimizar los recursos.

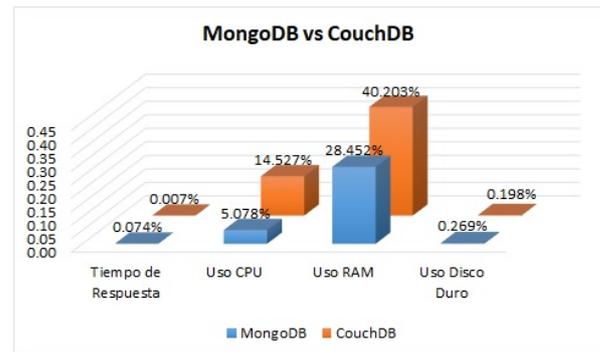


Figura 7. Tiempo de respuesta y uso de recursos

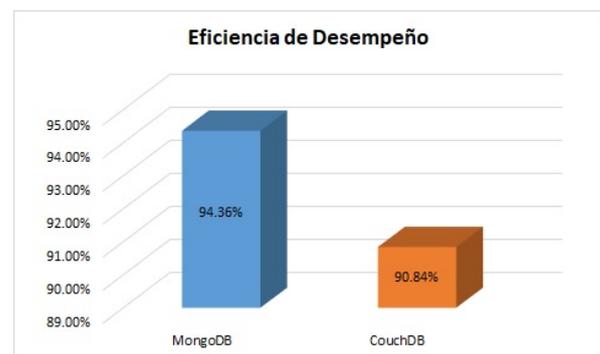


Figura 8. Eficiencia de Desempeño

D. Interfaz Aplicación Web

En la Fig. 9 se evidencia la interfaz diseñada y programada de la Aplicación Web de Gestión de Documentos CONAGOPARE CHIMBORAZO, el objetivo principal fue optimizar el proceso de entrega y recepción de documentos.



Figura 9. Interfaz del Sistema Web

V. CONCLUSIONES

La Norma ISO/IEC 25010/25023 determinó las métricas de eficiencia de desempeño para evaluar cada una de las bases de datos NoSQL (MongoDB y CouchDB). Obteniendo como resultados, Tiempo de respuesta CouchDB con una diferencia de 0,067s menos que MongoDB. Procesador (CPU) MongoDB presenta una diferencia de 9,449% menos que CouchDB. Memoria RAM MongoDB presenta una diferencia de 11,751% menos que CouchDB. Disco Duro CouchDB usa 0,071% menos que MongoDB. No existe una diferencia significativa de eficiencia de desempeño entre las dos bases de datos MongoDB y CouchDB.



Investigaciones previas ratifican que MongoDB es superior a CouchDB, y como resultado de la investigación se afirma que MongoDB presenta una leve ventaja de eficiencia de desempeño de 94,36% y CouchDB 90,84%, por lo que se consideró a MongoDB para implementarlo en el aplicativo Web de Gestión de Documentos CONAGOPARE CHIMBORAZO.

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.

REFERENCIAS

- [1] Angeles Antaño, José Castro, and René Valencia. *Migración de bases de datos sql a nosql*. Revista Tlamati, Especial, 3:144–148, 2014.
- [2] Odalys Aranda, Antonio Rodríguez, and Mayra Mugica. *Diagnóstico sobre la gestión documental y de archivos en la universidad central marta abreu de las villas. cuba: Caso de estudio*. Revista Española de Documentación Científica, 35(4):573–598, 2012.
- [3] Kent Beck. *Extreme Programming Explained: Embrace Change*. Addison-Wesley, 2000.
- [4] Vanessa Berlanga, and María Rubio. *Clasificación de pruebas no paramétricas. Cómo aplicarlas en spss*. REIRE. Revista d'Innovació i Recerca en Educació, vol. 5, num. 2, p. 101-113, 2012.
- [5] Kristina Chodorow. *MongoDB: the definitive guide*. Second Edition, USA: O'Reilly Media, Inc., 2013
- [6] Hansel del Busto, and Osmel Enríquez. *Bases de datos nosql*. Revista Telemática, 11(3):21–33, 2013.
- [7] Patricia Gallo. *Gestión documental en las organizaciones*. Editorial UOC, 2011.
- [8] Jing Han, Haihong E, Guan Le, and Jian Du. *Survey on nosql database*. In 6th international conference on pervasive computing and applications, pages 363–366. IEEE, 2011.
- [9] Harol H, and Carlos VALENZUELA. *NoSQL, la nueva tendencia en el manejo de datos*. Tecnología Investigación Y Academia, vol. 4, no 1, p. 147-150, 2016.
- [10] ISO, N. *Normas iso 25000*. Obtenido de <https://iso25000.com/index.php/normas-iso-25000/>, 2015.
- [11] Apache JMeter. *Apache JMeter*. Online. <https://jmeter.apache.org/>, 2016.
- [12] ActiveXperts, SMS. *ActiveXperts SMS Messaging Server*. Online. <https://www.activeexperts.com/sms-messaging-server/>, 2018.
- [13] NetBeans, I. D. E. *Netbeans IDE*. Online. <https://netbeans.org/>, 2018.
- [14] Juan Mogollón, and Telma de Carvalho. *La web semántica y sus usos en los procesos de documentación organizacional*. Informação & Informação, 22(2):437–448, 2017.
- [15] Marianela Morejón. *La información en bases de datos nosql y su incidencia en la generación documental de la secretaria general del honorable consejo universitario*. Master's thesis, Universidad Técnica de Ambato. Facultad de Ingeniería en Sistemas, 2018.
- [16] Francisco Moreno, Juan Quintero, and Robinson Rueda. *A performance comparison between oracle and mongodb*. Ciencia e Ingeniería Neogranadina, 26(1):109–129, 2016.
- [17] Alexander Castro, Juan González, and Mauro Callejas. *Utilidad y funcionamiento de las bases de datos nosql*. Facultad de Ingeniería, 21(33):21–32, 2012.
- [18] Roger Pressman. *Ingeniería del software. Un enfoque práctico*. Séptima Edición. McGrawHill, 2010.
- [19] Solid, I.T. *DB-Engines ranking*. Solid IT. Obtenido de <https://db-engines.com/en/ranking>, 2020
- [20] SPSS, IBM. *IBM SPSS Statistics for Windows, version 25*. Armonk, NY: IBM SPSS Corp, 2017.
- [21] Friends, Apache. *XAMPP*. Obtenido de <https://www.apachefriends.org/de/index.html>, 2018.
- [22] Yanette González and Yenisleidy Fernández. *Patrón Modelo-Vista-Controlador*. Revista Telemática 11.1, pp. 47-57, 2012.