



**FACULTAD DE INGENIERÍA, ARQUITECTURA Y  
URBANISMO**

**ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS  
TESIS**

**ANÁLISIS COMPARATIVO DE RENDIMIENTO DE  
GESTORES DE BASE DE DATOS NOSQL  
DOCUMENTALES**

**PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO  
DE SISTEMAS**

**Autor:**

**Bach. Venegas Bravo José Alexander  
ORCID: <https://orcid.org/0000-0003-0515-5645>**

**Asesor:**

**Mg. Bravo Ruiz Jaime Arturo  
ORCID: <https://orcid.org/0000-0003-1929-3969>**

**Línea de Investigación:**

**Infraestructura, Tecnología y Medio Ambiente**

**Pimentel – Perú  
2022**

**ANÁLISIS COMPARATIVO DE RENDIMIENTO GESTORES DE BASE DE  
DATOS NOSQL DOCUMENTALES**

**PRESENTADO POR:**

---

**Bach. Venegas Bravo José Alexander**  
**Autor**

---

**Mg. Bravo Ruiz Jaime Arturo**  
**Asesor Metodológico**

**APROBADO POR:**

---

**Dr. Sánchez Chero Manuel Jesús**  
**Presidente**

---

**Mg. Bravo Ruiz Jaime Arturo**  
**Secretario**

---

**Mg. Diaz Vidarte Miguel Orlando**  
**Vocal**

## **DEDICATORIA**

### **A Dios**

Por fortalecerme todos los días, por permitirme creer en él y por darme sabiduría para permanecer en el camino correcto y lograr completar mi carrera profesional.

### **A mis Padres**

Ana y José, por su amor y dedicación a lo largo de mi vida; por orientarme día a día para la culminación de mi etapa profesional, a través de esta Tesis, les recuerdo lo que han hecho y lo que hacen por mí.

### **A mis Hermanos**

Juan y Wilmer, porque siempre fueron mi motivación para salir adelante y por darme el aliento de superación.

### **A mi esposa**

Thalía, por el maravilloso apoyo que me brinda todos los días para alcanzar nuevas metas. También por lo que gran parte de esta Tesis se lo debo a ella por los ánimos y el aliento que me brinda a diario.

### **A mi Hijo**

Matheus, por ser mi fortaleza de seguir superándome.

Porque yo sé muy bien los planes que tengo para ustedes – afirma el señor -, planes de bienestar y no de calamidad, a fin de darles un futuro y una esperanza.

**Jeremías 29:11**

## **AGRADECIMIENTO**

Agradezco a Dios por iluminarme siempre y permitirme seguir en el camino del bien, por darme la oportunidad, que con mi vida profesional acompañe y ayude a mis semejantes; y a nuestro señor Jesucristo por acogernos siempre bajo su manto y tenerlo como ejemplo para mi vida.

Agradezco a mis amigos y familiares; que me brindaron su apoyo y su tiempo para poder cumplir esta meta trazada.

Mira que te mando que te esfuerces y seas valiente, no temas ni desmayes, porque Jehová tu Dios estará contigo en dondequiera que vayas.

**Josué 1:9**

## RESUMEN

Un sistema de gestión de bases de datos (DBMS) es una plataforma de software de sistema que almacena y recupera una colección de datos relacionados, se utiliza generalmente en áreas de gestión de datos específicos de la aplicación, donde el rendimiento, la disponibilidad y la escalabilidad tienen la máxima prioridad. NoSQL es un método alternativo de base de datos relacional, que almacena los datos no estructurados o semiestructurados en un formato sin esquema. Según las estimaciones, el volumen de datos comerciales se duplica cada 1,2 años. El aumento de los datos no estructurados recogidos por las empresas es innegable. Las nuevas épocas de los marcos exigen la escalada horizontal de las bases de datos. Los datos con diversos formatos, incluidos los datos no estructurados, deben almacenarse y procesarse en las bases de datos. El enfoque NoSQL es la respuesta a estas preguntas. Este estudio se basa en la comparación del rendimiento de gestores de base de datos NoSQL del tipo documental en el procesamiento de big data; para esta investigación se eligió dos gestores de base de datos NoSQL del tipo documental. se eligió trabajar con el tipo Documental; porque están diseñadas para almacenar, recuperar y gestionar datos orientados a documentos; luego se presentó un ranking con 08 de gestores de base de datos documental elaborado por DB-Engines; después se seleccionó a los dos gestores de base de datos mediante las siguientes características: Tipo de datos, Modelo de datos, plataforma y tipo de licencia. Asimismo, se construyó un ambiente físico para las pruebas de rendimiento y se construyó una plataforma web para medir el tiempo de respuesta. Las pruebas de rendimiento se realizaron con dos colecciones de datos, una colección con 4056 documentos y la otra con 1'566,188 documentos. Los indicadores de evaluación de las pruebas de rendimiento fueron Consumo de memoria RAM, Consumo CPU, Tiempo de Respuesta, Espacio usado de Disco y cantidad de errores. Se analizaron los datos y se concluye que el administrador de base de datos NoSQL MongoDB ofreció mejores resultados utilizando menos uso de CPU y tiempo de Respuesta; mientras CouchDB ofreció menos consumo de Memoria RAM y Espacio usado de Disco cuando se usa grandes conjuntos de datos, en ninguna de las pruebas se presentaron errores.

**Palabras claves:** Base de Datos, NoSQL, Documento, MongoDB, CouchDB.

## ABSTRACT

A database management system (DBMS) is a system software platform that stores and retrieves a collection of related data, typically used in application-specific data management areas where performance, availability and scalability have the highest priority. NoSQL is an alternative relational database method, which stores unstructured or semi-structured data in a schema-less format. According to estimates, the volume of business data is doubling every 1.2 years. The increase in unstructured data collected by companies is undeniable. The new epochs of frameworks demand horizontal scaling of databases. Data in various formats, including unstructured data, must be stored and processed in databases. The NoSQL approach is the answer to these questions. This study is based on comparing the performance of NoSQL database managers of the Document type in processing big data; for this research, two NoSQL database managers of the Document type were chosen. it was chosen to work with the Document type; because they are designed to store, retrieve and manage document-oriented data; then a ranking with 08 of Document database managers prepared by DB-Engines was presented; then the two database managers were selected by the following characteristics: Data type, Data model, platform and license type. Also, a physical environment was built for performance testing and a web platform was built to measure response time. The performance tests were performed with two data collections, one collection with 4056 documents and the other with 1'566,188 documents. The evaluation indicators of the performance tests were RAM Consumption, CPU Consumption, Response Time, Used Disk Space and number of errors. The data were analyzed and it was concluded that the NoSQL database manager MongoDB offered better results using less CPU usage and response time; while CouchDB offered less RAM consumption and less used disk space when using large datasets, in none of the test's errors were presented.

**Keywords:** Database, NoSQL, Document, MongoDB, CouchDB

## ÍNDICE

<b>I. INTRODUCCIÓN</b>	12
1.1. Realidad problemática.	12
1.2. Antecedentes de Estudio.	13
1.3. Teorías relacionadas al tema.	18
1.4. Formulación del problema.	37
1.5. Justificación e importancia del estudio.	37
1.6. Hipótesis.	38
1.7. Objetivos.	38
1.7.1. Objetivos General.	38
1.7.2. Objetivos Específicos.	38
<b>II. MATERIAL Y MÉTODO</b>	39
2.1. Tipo y Diseño de Investigación.	39
2.2. Población y muestra.	39
2.3. Variables, Operacionalización.	42
2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.	43
2.5. Procedimiento de análisis de datos.	44
2.6. Criterios éticos.	46
2.7. Criterios de rigor científico.	46
<b>III. RESULTADOS</b>	48
3.1. Resultado en Tablas y Figuras.	48
3.2. Discusión de resultados.	66
3.3. Aporte práctico.	68
<b>IV. CONCLUSIONES Y RECOMENDACIONES</b>	122
4.1. Conclusiones.	122
4.2. Recomendaciones.	123
<b>REFERENCIAS</b>	124
<b>ANEXOS</b>	126

## ÍNDICE DE FIGURAS

<b>Figura 1</b> - Teorema CAP.....	20
<b>Figura 2</b> - Estructura de una Base de Datos NoSQL Clave - Valor. ....	23
<b>Figura 3</b> - Ejemplo de escritura en una BBDD clave-valor. ....	23
<b>Figura 4</b> - Ejemplo de un documento XML. ....	24
<b>Figura 5</b> - Ejemplo de un documento JSON. ....	25
<b>Figura 6</b> - Estructura de una Base de Datos NoSQL Documental.....	26
<b>Figura 7</b> - Estructura de una Base de Datos NoSQL orientado a Grafos. ....	34
<b>Figura 8</b> - Ejemplo de tabla de una BBDD Columnar. ....	35
<b>Figura 9</b> - Ranking de los Gestores de Base de Datos NoSQL Documental. ....	40
<b>Figura 10</b> - Gráfico resumen del uso de memoria RAM en la prueba uno.....	49
<b>Figura 11</b> - Gráfico resumen del uso de CPU en la prueba uno.....	50
<b>Figura 12</b> - Gráfico resumen del uso de Tiempo de respuesta en la prueba uno.....	51
<b>Figura 13</b> - Gráfico resumen del uso de Espacio usado en Disco de la prueba uno. ....	52
<b>Figura 14</b> - Gráfico resumen del uso de memoria RAM en la prueba dos. ....	53
<b>Figura 15</b> - Gráfico resumen del uso de CPU en la prueba dos. ....	54
<b>Figura 16</b> - Gráfico resumen del uso de Tiempo de Respuesta en la prueba dos. ....	55
<b>Figura 17</b> - Gráfico resumen del uso de memoria RAM en la prueba tres.....	56
<b>Figura 18</b> - Gráfico resumen del uso de CPU en la prueba tres. ....	57
<b>Figura 19</b> - Gráfico resumen del uso de Tiempo de respuesta en la prueba tres.....	58
<b>Figura 20</b> - Gráfico resumen del uso de memoria RAM en la prueba cuatro.....	59
<b>Figura 21</b> - Gráfico resumen del uso de CPU en la prueba cuatro. ....	60
<b>Figura 22</b> - Gráfico resumen del uso de tiempo de respuesta en la prueba cuatro. ....	61
<b>Figura 23</b> - Promedio de Uso de Memoria RAM.....	62
<b>Figura 24</b> - Promedio de Uso de CPU. ....	63
<b>Figura 25</b> - Promedio de Tiempo de Respuesta. ....	64
<b>Figura 26</b> - Espacio Usado en Disco en Operaciones de Volcado. ....	65
<b>Figura 27</b> - Base de Datos Especializada IEEE Xplore. ....	68
<b>Figura 28</b> - Base de Datos Especializada ScieDirect. ....	69
<b>Figura 29</b> - Ranking de Base de Datos NoSQL Documental. ....	69
<b>Figura 30</b> - Descripción del Sistema Operativo ....	72
<b>Figura 31</b> - Código Fuente de la plataforma web Cronometro.....	74
<b>Figura 32</b> - Plataforma Web de Cronometro para medir tiempo de Respuesta. (CouchDB).....	74
<b>Figura 33</b> - Plataforma Web de Cronometro para medir tiempo de Respuesta. (MongoDB).....	75
<b>Figura 34</b> - Línea de comando de MongoDB para calcular el tamaño de una colección. ....	75
<b>Figura 35</b> - Apartado de propiedades de CouchDB.....	75
<b>Figura 36</b> - Sentencia try catch para medir el indicador cantidad de errores.....	76
<b>Figura 37</b> - Gráfico de uso de memoria RAM (Volcado 1).....	77
<b>Figura 38</b> - Gráfico de uso de CPU (Volcado 1). ....	78



<b>Figura 39</b> - Gráfico del Uso del tiempo de respuesta (Volcado 1). ....	79
<b>Figura 40</b> - Gráfico de Espacio usado en Disco (Volcado 1). ....	80
<b>Figura 41</b> - Gráfico de uso de memoria RAM (Volcado 2). ....	81
<b>Figura 42</b> - Gráfico de la cantidad de Uso de CPU (Volcado 2). ....	82
<b>Figura 43</b> - Gráfico del Uso de Tiempo de Respuesta (Volcado 2). ....	83
<b>Figura 44</b> - Gráfico del Espacio usado en Disco (Volcado 2). ....	84
<b>Figura 45</b> - Gráfico de consumo de Memoria RAM (Listado 1). ....	86
<b>Figura 46</b> - Gráfico de cantidad de uso de CPU (Listado 1). ....	86
<b>Figura 47</b> - Gráfico del Uso de Tiempo de respuesta (Listado 1). ....	87
<b>Figura 48</b> - Gráfico cantidad de uso de Memoria RAM (Listado 2). ....	88
<b>Figura 49</b> - Gráfico de cantidad de CPU (Listado 2). ....	89
<b>Figura 50</b> - Gráfico del Uso de tiempo de respuesta (Listado 2). ....	90
<b>Figura 51</b> - Gráfico de cantidad de uso Memoria RAM (Listado 3). ....	91
<b>Figura 52</b> - Gráfico Cantidad de uso de CPU (Listado 3). ....	92
<b>Figura 53</b> - Gráfico de Uso de Tiempo de Respuesta (Listado 3). ....	93
<b>Figura 54</b> - Gráfico de Cantidad de Uso de Memoria RAM (Listado 4). ....	94
<b>Figura 55</b> - Gráfico Cantidad de Uso de CPU (Listado 4). ....	95
<b>Figura 56</b> - Gráfico de Uso de Tiempo de Respuesta (Listado). ....	96
<b>Figura 57</b> - Gráfico de Consumo de Memoria RAM (Actualización 1). ....	97
<b>Figura 58</b> - Gráfico de la Cantidad de Uso de CPU (Actualización 1). ....	98
<b>Figura 59</b> - Gráfico del Uso del Tiempo de Respuesta (Actualización 1). ....	99
<b>Figura 60</b> - Gráfico de Cantidad de Uso de Memoria RAM (Actualización 2). ...	100
<b>Figura 61</b> - Gráfico de Cantidad de Uso de CPU (Actualización 2). ....	101
<b>Figura 62</b> - Gráfico del Uso de Tiempo de Respuesta (Actualización 2). ....	102
<b>Figura 63</b> - Gráfico de Cantidad de uso de Memoria RAM (Actualización 3) ...	103
<b>Figura 64</b> - Gráfico de la Cantidad de Uso de CPU (Actualización 3). ....	104
<b>Figura 65</b> - Gráfico del Uso del Tiempo de Respuesta (Actualización 3). ....	105
<b>Figura 66</b> - Gráfico de Cantidad de Uso de Memoria RAM (Actualización 4). ...	106
<b>Figura 67</b> - Gráfico de la Cantidad de Uso de CPU (Actualización 4). ....	107
<b>Figura 68</b> - Gráfico del Uso del Tiempo de Respuesta (Actualización 4). ....	108
<b>Figura 69</b> - Gráfico de Cantidad de Uso de Memoria RAM (Eliminación 1). ....	109
<b>Figura 70</b> - Gráfico de la Cantidad de Uso de CPU (Eliminación 1). ....	110
<b>Figura 71</b> - Gráfico del Uso de Tiempo de Respuesta (Eliminación 1). ....	111
<b>Figura 72</b> - Gráfico de la Cantidad de Uso de Memoria RAM (Eliminación 2). ...	112
<b>Figura 73</b> - Gráfico de la Cantidad de Uso de CPU (Eliminación 2). ....	113
<b>Figura 74</b> - Gráfico del Uso de Tiempo de Respuesta (Eliminación 2). ....	114
<b>Figura 75</b> - Gráfico de la Cantidad de Uso de Memoria RAM (Eliminación 3). ...	115
<b>Figura 76</b> - Gráfico de la Cantidad de Uso de CPU (Eliminación 3). ....	116
<b>Figura 77</b> - Gráfico del Uso de Tiempo de Respuesta (Eliminación 3). ....	117
<b>Figura 78</b> - Gráfico de la Cantidad de Uso de Memoria RAM (Eliminación 4). ...	118
<b>Figura 79</b> - Gráfico de la Cantidad de Uso de CPU (Eliminación 4). ....	119
<b>Figura 80</b> - Gráfico del Uso de Tiempo de Respuesta (Eliminación 4). ....	120

## ÍNDICE DE TABLAS

<b>Tabla 1</b> - Clasificación de bases de Datos NoSQL en función de la condición que cumplen.....	21
<b>Tabla 2</b> - Características de ISO/IEC 25010.....	36
<b>Tabla 3</b> - Población de los Gestores de Base de Datos NoSQL Documental.....	40
<b>Tabla 4</b> – Características de los Gestores de Base de Datos NoSQL Documental.....	41
<b>Tabla 5</b> - Muestra de los Gestores de Base de Datos NoSQL Documental.....	42
<b>Tabla 6</b> – Operacionalización de variables .....	43
<b>Tabla 7</b> - Resumen del Uso de Memoria RAM - Prueba uno.....	48
<b>Tabla 8</b> - Resumen del Uso de CPU - Prueba uno. ....	49
<b>Tabla 9</b> - Resumen del Tiempo de Respuesta - Prueba uno.....	50
<b>Tabla 10</b> - Resumen del Espacio Usado en Disco - Prueba uno .....	51
<b>Tabla 11</b> - Resumen del Uso de Memoria RAM - Prueba dos.....	52
<b>Tabla 12</b> - Resumen del Uso de CPU - Prueba dos. ....	53
<b>Tabla 13</b> - Resumen del Tiempo de Respuesta - Prueba dos.....	54
<b>Tabla 14</b> - Resumen del Uso de Memoria RAM - Prueba tres.....	56
<b>Tabla 15</b> - Resumen del Uso de CPU - Prueba tres.....	57
<b>Tabla 16</b> - Resumen del Tiempo de Respuesta - Prueba tres.....	58
<b>Tabla 17</b> - Resumen del Uso de Memoria RAM - Prueba cuatro.....	59
<b>Tabla 18</b> - Resumen del Uso de CPU - Prueba cuatro.....	60
<b>Tabla 19</b> - Resumen del Tiempo de Respuesta - Prueba cuatro.....	61
<b>Tabla 20</b> - Resultados Obtenidos de la Colección Establecimientos.....	66
<b>Tabla 21</b> - Resultados Obtenidos de la Colección Egresoshospitalarios.....	66
<b>Tabla 22</b> - Características de los Gestores de Base de Datos NoSQL Documental.....	70
<b>Tabla 23</b> - Evaluación de características con Puntaje.....	71
<b>Tabla 24</b> – Características y descripción del Hardware del entorno de Trabajo ..	72
<b>Tabla 25</b> - Cantidad de Uso de Memoria RAM (Volcado 1).....	77
<b>Tabla 26</b> - Cantidad de Uso de CPU (Volcado 1).....	78
<b>Tabla 27</b> – Uso del Tiempo de Respuesta (Volcado 1).....	79
<b>Tabla 28</b> - Espacio Usado en Disco (Volcado 1).....	80
<b>Tabla 29</b> - Cantidad de Uso de Memoria RAM (Volcado 2).....	81
<b>Tabla 30</b> - Cantidad de uso de CPU (Volcado 2).....	82
<b>Tabla 31</b> – Uso de Tiempo de Respuesta (Volcado 2).....	83
<b>Tabla 32</b> - Espacio Usado en disco (Volcado 2).....	84
<b>Tabla 33</b> - Cantidad de Uso de Memoria RAM (Listado 1).....	85
<b>Tabla 34</b> - Cantidad de Uso de CPU (Listado 1).....	86
<b>Tabla 35</b> – Uso del Tiempo de Respuesta (Listado 1).....	87
<b>Tabla 36</b> – Cantidad de uso de Memoria RAM (Listado 2).....	88
<b>Tabla 37</b> - Cantidad de uso de CPU (Listado 2).....	89
<b>Tabla 38</b> – Uso de Tiempo de Respuesta (Listado 2).....	90
<b>Tabla 39</b> - Cantidad de uso de Memoria RAM (Listado 3).....	91
<b>Tabla 40</b> - Cantidad de uso de CPU (Listado 3).....	92
<b>Tabla 41</b> – Uso de Tiempo de Respuesta (Listado 3).....	93

Tabla 42 - Cantidad de Uso de Memoria RAM (Listado 4).....	94
<b>Tabla 43</b> - Cantidad de Uso de CPU (Listado 4).....	95
<b>Tabla 44</b> – Uso de Tiempo de Respuesta (Listado 1).....	96
<b>Tabla 45</b> - Cantidad de Uso de Memoria RAM (Actualización 1).....	97
<b>Tabla 46</b> - Cantidad de Uso de CPU (Actualización 1). ....	98
<b>Tabla 47</b> – Uso del Tiempo de Respuesta (Actualización 1).....	99
<b>Tabla 48</b> - Cantidad de Uso de Memoria RAM (Actualización 2).....	100
<b>Tabla 49</b> - Cantidad de Uso de CPU (Actualización 2). ....	101
<b>Tabla 50</b> - Uso de Tiempo de Respuesta (Actualización 2). ....	102
<b>Tabla 51</b> - Cantidad de Uso de Memoria RAM (actualización 3) .....	103
<b>Tabla 52</b> - Cantidad de Uso de CPU (Actualización 3). ....	104
<b>Tabla 53</b> - Uso de Tiempo de Respuesta (Actualización 3). ....	105
<b>Tabla 54</b> - Cantidad de Uso de Memoria RAM (Actualización 4).....	106
<b>Tabla 55</b> - Cantidad de Uso de CPU (Actualización 4). ....	107
<b>Tabla 56</b> - Uso del Tiempo de Respuesta (Actualización 4). ....	108
<b>Tabla 57</b> - Cantidad de Uso de Memoria RAM (Eliminación 1).....	109
<b>Tabla 58</b> - Cantidad de Uso de CPU (Eliminación 1). ....	110
<b>Tabla 59</b> - Uso de Tiempo de Respuesta (Eliminación 1).....	111
<b>Tabla 60</b> - Cantidad de Uso de Memoria RAM (Eliminación 2).....	112
<b>Tabla 61</b> - Cantidad de Uso de CPU (Eliminación 2). ....	113
<b>Tabla 62</b> - Uso de Tiempo de Respuesta (Eliminación 2).....	114
<b>Tabla 63</b> - Cantidad de Uso de Memoria RAM (Eliminación 3).....	115
<b>Tabla 64</b> - Cantidad de Uso de CPU (Eliminación 3). ....	116
<b>Tabla 65</b> - Uso de Tiempo de Respuesta (Eliminación 3).....	117
<b>Tabla 66</b> - Cantidad de Uso de Memoria RAM (Eliminación 4).....	118
<b>Tabla 67</b> - Cantidad de Uso de CPU (Eliminación 4). ....	119
<b>Tabla 68</b> - Uso de Tiempo de Respuesta (Eliminación 4).....	120

## **I. INTRODUCCIÓN**

### **1.1. Realidad problemática.**

Actualmente, las bases de datos son ampliamente utilizadas en el campo de los sistemas de información, los cuales nos permiten trabajar con múltiples niveles de datos, siendo una de estas las bases de datos relacionales, asimismo son utilizados en distintas organizaciones debido a su fácil uso y la integridad con la que manejan los datos; pero muestra ineficiencia al manejar un gran volumen de datos, debido a este problema surgieron las bases de datos NoSQL, que, a diferencia de las bases de datos relacionales, pueden manejar gran cantidad de datos, entre ellos los datos de tipo no estructurados como: documentos, multimedia, correos electrónicos y medios sociales de manera eficaz.

Así mismo Jung, Youn, Bae, & Choi (2016) describen a NoSQL como un sistema de gestión de base de datos que facilita el procesamiento de datos masivos estructurados y no estructurados y que hoy en día muchas empresas tienden a iniciar proyectos utilizando NoSQL; ya que los sistemas de gestión de base de datos relacionales que se han venido utilizando, tienen problemas en el procesamiento de datos de tipo no estructurados, así como también el rendimiento y coste en el procesamiento de datos masivos. Las bases de datos NoSQL realizan la lectura y escritura rápidamente, gracias a la función de mapeo de memoria; haciendo de NoSQL un sistema de gestión de base de datos adecuado para el procesamiento de grandes volúmenes de datos ya sean estructurados como no estructurados con mayor facilidad.

Dado este contexto, en la actualidad las empresas aún siguen apostando por los sistemas de gestión de base de datos relacionales, dado que estas muestran ciertas características importantes para la empresa como agrupar y almacenar la información de manera fiable y segura.

Por otra parte, los almacenes de datos, son considerados una parte importante en la inteligencia de negocios el cual se define como un conjunto de operaciones, tecnologías y aplicaciones que proporcionan una recuperación de acceso rápido y sencillo a los datos del sistema de gestión empresarial para su análisis e interpretación.

Existen organizaciones que generan terabytes de datos, los cuales necesitan ser almacenados y a la misma vez ser analizados para generar información útil; siendo esto un reto para los almacenes de datos relacionales. (Pereira, Oliveira, & Rodrigues, 2015)

De igual manera Murazza & Nurwidyantoro (2017) en su trabajo de investigación nos dicen que los sistemas de gestión de base de datos relacionales, están empezando a mostrar sus limitaciones al momento almacenar y gestionar big data, y debido a este problema las bases de datos NoSQL, están empezando a ser consideradas como una elección a las bases de datos relacionales porque ofrece un rendimiento mucho mayor que estas.

Según la empresa SolidQ (2018) a medida que aumenta la cantidad de datos en una organización, surgirá la pregunta de cómo ejecutar de manera rápida y eficiente cargas de trabajo cada vez más grandes y masivas.

Según IBM (2019) Las inserciones masivas de datos pueden aumentar los resultados operativos de inserción en el origen de datos.

De acuerdo a lo expuesto en las investigaciones, queda claro que las bases de datos NoSQL son claramente la principal herramienta para gestionar grandes cantidades de datos, asimismo existe una problemática acerca del rendimiento en el manejo de grandes conjunto de datos en gestores de base de datos NoSQL documental, pero no existe una investigación acerca del rendimiento en el procesamiento de grandes conjuntos de datos, de acuerdo a los siguientes criterios: Consumo de Memoria RAM, Consumo de CPU, Tiempo de Respuesta, Espacio usado en Disco y Cantidad de Errores; por lo que genera incertidumbre al escoger que base de datos NoSQL Documental sea mejor en el procesamiento de grandes conjuntos de datos.

## **1.2. Antecedentes de Estudio.**

Sundhara & Srividya, (2017), en su estudio titulado como “A Performance Comparison of Document Oriented NoSQL Databases”. El propósito de este trabajo es la comparación de MongoDB y CouchBase, las dos bases de datos son tomadas bajo experimentación. Estas dos son bases de datos NoSQL orientadas a documentos ampliamente utilizadas. La razón que consideraron

estas dos bases de datos es para entender para qué aplicación, qué base de datos se adapta bien para las aplicaciones de streaming.

La Arquitectura de este sistema se accede a los datos de streaming a través de la plataforma NodeJS. Una vez capturados los tweets, aparecen en forma de documentos JSON. Estos documentos JSON se almacenan en MongoDB y CouchBase para llevar a cabo la comparación de rendimiento. En su marco de trabajo sugiere que se puede acceder a la información del flujo de twitter a través de la plataforma NodeJS. Una vez capturados los tweets, aparecen en forma de documentos JSON. Algunas de las características que se han tomado son: Replicación, Tipo de almacenamiento, Características del CAP, MapReduce. Las características que pertenecen a la base de datos bajo estudio se toman y para cualquier aplicación de streaming se discute cómo se comportan.

Asimismo, Analizaron las características cualitativas de ambas bases de datos Documentales. Los resultados sugieren que para las aplicaciones de streaming, MongoDB claramente tiene una ventaja sobre CouchBase. Dado que CouchBase no tiene una opción para importar documentos JSON de forma masiva, demuestra ser un modelo de fallo para este tipo de aplicaciones de streaming.

Gupta & Rani, (2016), en su investigación titulada como “A Comparative Study of Elasticsearch and CouchDB Document Oriented Databases”, el propósito fue familiarizarse primero con dos de las bases de datos NoSQL más usadas: Búsqueda elástica y CouchDB. Luego compararon el rendimiento de dos bases de datos: Búsqueda elástica y CouchDB en conjunto de datos de imágenes. La implementación lo realizaron en LINUX. La herramienta utilizada para la implementación es Eclipse IDE. Su codificación se ha realizado mediante JAVA. Las librerías Elasticsearch, CouchDB y Maven java han sido utilizadas para hacer la conexión y realizar todas las operaciones necesarias para la comparación. Rest Client API la utilizan para acceder al servicio de Elasticsearch y CouchDB.

También han realizado operaciones de Inserción, Selección, Actualización y Borrado y se han comparado los resultados en base al tiempo que ambas bases de datos han tardado en realizar estas operaciones.

Este análisis se basó en los resultados obtenidos al insertar, leer, actualizar y eliminar operaciones en ambas bases de datos orientadas a documentos, lo que justifica que CouchDB sea más eficaz que Elasticsearch durante las operaciones CRUD de inserción, actualización y eliminación, pero Elasticsearch está muy por delante de CouchDB en términos de operaciones de selección.

Chopade & Dhavase, (2017), en su investigación titulada como “MongoDB, CouchBase: Performance Comparison for Image Dataset”, La motivación de este trabajo es comparar el rendimiento de la base de datos en términos de tiempo necesario para almacenar y descargar imágenes de la base de datos. Realizaron la comparación de dos bases de datos NoSQL, MongoDB y CouchBase, para la inserción y recuperación de conjuntos de datos de imágenes. Para consultas normales, el rendimiento de la base de datos no importó, ya que se trata de datos de pequeño tamaño.

Para documentos de gran tamaño como imágenes o archivos, generalmente encontraron la diferencia, el tiempo requerido por MongoDB es mayor en comparación con CouchBase y para el caso de recuperación de conjuntos de datos de imágenes es exactamente diferente. El tiempo requerido para la recuperación es mayor para CouchBase que para MongoDB.

Truic, Rădulescu, Boicea, & Bucur, (2015), en su investigación titulada como “Performance evaluation for CRUD operations in asynchronously replicated document oriented database”, Este trabajo examina la replicación asincrónica, una de las características clave de un sistema escalable y flexible. Se examinan tres de las bases de datos alineadas a documentos más populares, Mongo DB, CouchDB y Couchbase. Para las pruebas, se considera el tiempo que lleva realizar una operación CRUD para una sola petición de base de datos y un entorno distribuido con dos nodos y los resultados se contrastan con los resultados de las pruebas obtenidas para tres sistemas de administración de bases de datos relacionales: Microsoft SQL Server, MySQL y PostgreSQL.

En ambos ambientes probados, el rendimiento de Couchbase y PostgreSQL son muy equivalentes. El entorno distribuido principal de Couchbase se basa en una arquitectura compartida de todo y cada vez que se añade un nuevo nodo, sus recursos se añaden al pool de recursos compartidos. Esta característica hace de Couchbase una buena opción cuando se trata de aplicaciones intensivas de procesamiento de datos. Los sistemas de administración de bases de datos son aplicaciones que pueden modificar la forma en que se manejan las operaciones CRUD con el tiempo, posiblemente mejorando o disminuyendo su rendimiento. A la hora de crear aplicaciones, se recomendó hacer la elección que mejor se adapte a los requisitos, tras un cuidadoso análisis de todas las características que ofrece.

Neto, Fernandes, Holanda, Ribeiro, & Carvalho, (2017), en su investigación titulada como “Análise de Desempenho do MongoDB Um comparativo entre instalações standalone e sharded cluster na análise de dados abertos do Programa Bolsa Família”, El propósito de este documento es evaluar los aspectos de rendimiento de MongoDB cuando se aplica al análisis de grandes volúmenes de datos abiertos relacionados con Bolsa Família, y evaluar las ganancias derivadas de la adopción de la arquitectura de clúster en comparación con el modo autónomo.

Gracias a los resultados obtenidos, fue posible verificar un aumento considerable del rendimiento en la ejecución de consultas agregadas o resumidas con la adopción de la solución en clúster, en comparación con la instalación autónoma. A efectos de rendimiento en consultas agregadas en grandes volúmenes de datos, la adopción de sharding basado en hash demostró ser una estrategia adecuada, al distribuir los datos de manera uniforme y, por lo tanto, dividir adecuadamente la carga de procesamiento en el clúster. La demanda de memoria y CPU en los componentes del nodo central es menor que en los fragmentos. Sin embargo, la demanda de redes es mayor. Estas características pueden ser explotadas para la optimización de los recursos. En contraste con el aumento de rendimiento durante la ejecución de consultas, el tiempo de carga de grandes volúmenes de datos en modo masivo fue considerablemente mayor en el clúster, con una alta demanda de recursos de red. La desactivación del



equilibrado durante la carga por lotes resultó ser una solución de contorno adecuada.

Gyrödi, Gyrödi, Pecherle, & Olah, (2015), en su investigación titulada como “A Comparative Study: MongoDB vs. MySQL”, En este trabajo realizan una tesis comparativa entre bases de datos relacionales y no relacionales. Centrando su presentación principalmente en una implementación de las ciencias aplicadas de bases de datos NoSQL, MongoDB, y una comparación con otra implementación de bases de datos relacionales, MySQL, justificando así por qué MongoDB es más eficiente que MySQL. También presentaron las ventajas de utilizar una base de datos no relacional en comparación con una base de datos relacional, integrada en un foro en el campo del desarrollo personal y profesional. La base de datos NoSQL utilizada para desarrollar el foro es MongoDB, fue elegida entre una diversidad de bases de datos no relacionales.

Pereira, Oliveira, & Rodrigues, (2015), en su investigación titulada: Iberian Conference on Information Systems and Technologies (CISTI), Portugal, presentaron Data warehouses in MongoDB vs SQL Server: A comparative analysis of the query performance; en su indagación se realizó un análisis comparativo del rendimiento de las consultas de MongoDB y MySQL, se realizaron distintas pruebas para tramitar y examinar grandes volúmenes de datos, para ello se establecieron 6 diferentes cargas de datos y 4 tipos de consultas con cierto grado de complejidad para determinar el nivel de rendimiento. Obteniendo distintos tipos de resultados mostrando SQL server el mejor resultado al ejecutar consultas complejas.

Jung, Youn, Bae, & Choi, (2016), en su investigación titulada; International Conference on Database Theory and Application (DTA), presentaron su investigación A Study on Data Input and Output Performance Comparison of MongoDB and PostgreSQL in the Big Data Environment; su investigación tuvo como objetivo analizar el rendimiento de los datos de acceso y salida en un entorno de big data, para ello se comparó los sistemas MongoDB y PostgreSQL al procesar grandes cantidades de datos, se ejecutaron operaciones de inserción, selección, actualización y eliminación utilizando 5 conjuntos de datos;

dando como resultado que en las operaciones realizadas MongoDB mostro mejor desempeño que PostgreSQL.

### **1.3. Teorías relacionadas al tema.**

#### **1.3.1. Base de datos**

Gómez, (2013), afirma las bases de datos son una recopilación organizada de información y un conjunto de programas que acceden a esos datos. Los datos generalmente se muestran como imágenes, gráficos, texto, números, etc. y a menudo se llevan a cabo para modelar aspectos importantes de la realidad.

#### **1.3.2. Lenguaje de Base de Datos**

El sistema de base de datos provee un lenguaje de definición de datos para detallar el esquema de la base de datos y un lenguaje de administración de datos para consultar y modificar la base de datos. De hecho, la definición de datos y el lenguaje de administración de datos no son dos lenguajes diferentes, sino solo parte de un lenguaje de base de datos, como el SQL de uso común. Silberschatz, Korth, & Sudarhan, (2006).

#### **1.3.3. Sistemas gestores de base de datos**

Un SGBD, tiene como objetivo proporcionar al usuario herramientas que le permitan manipular datos con requisitos abstractos, es decir, de tal manera que no tenga que saber cómo se almacenan los datos en la computadora o la tecnología de acceso que está utilizando. Las aplicaciones procesan los datos almacenados en la gestión de la información para facilitar el trabajo del usuario. Sarasa, (2016).

#### **1.3.4. Base de datos NoSQL**

En el libro “Introducción a las bases de datos NoSQL utilizando MongoDB” Cabezuelo, (2016). Nos dice que, durante décadas, las bases de datos relacionales han sido el modelo de perseverancia más manejado en la mayoría de las aplicaciones, pero no son perfectas ya que presentan limitaciones al procesar grandes cantidades de datos. Debido a esto algunas empresas

dispusieron usar como opciones a las bases de datos relacionales, tales como Google y Amazon que desplegaron sus propias técnicas de almacenamiento basado en clústeres. Estos sistemas de bases de datos permiten la gestión y el procesamiento de grandes cantidades de datos en un entorno distribuido y, por lo tanto, se consideran la fuente de las llamadas "bases de datos NoSQL". Las bases de datos NoSQL proveen el modelo de datos que mejor se adapta a las necesidades de su aplicación; Entre las bases de datos NoSQL, se pueden distinguir las siguientes: bases de datos clave - valor, bases de datos de documentos, bases de datos basadas en columnas y bases de datos de gráficos.

#### **1.3.4.1. Características de las Base de Datos NoSQL**

Si bien se han identificado varios métodos NoSQL diferentes, puede encontrar características comunes.

**Escalabilidad horizontal y habilidad de distribución.** - En este tipo de sistema se pueden quitar, agregar o realizar operaciones a través de sus componentes (hardware) sin afectar su desempeño. Esta escalabilidad facilita la copia de seguridad y la distribución de datos en el servidor. Este almacenamiento distribuido mejora la confiabilidad y la escalabilidad, pero aumenta los costos.

**Libertad de esquema.** - Tener un esquema elástico le brinda más libertad para crear sus datos. Esto facilita la combinación con lenguajes de programación orientados a objetos.

**Modelo de concurrencia débil.** - No implementan ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad), por lo que se necesita mucho cuidado para garantizar la consistencia transaccional. En su lugar, realiza BASE, que se explica con más detalle a continuación; Guy, (2015).

**Consultas simples.** - Las consultas más simples requieren menos operaciones, son más eficientes y simples. (Castro Romero, González Sanabria, & Callejas Cuervo, 2012)

#### 1.3.4.2. Teorema CAP (Consistency, Availability and Partition Tolerance)

Eric Brewer introdujo el concepto de entornos distribuidos en el año 2000 y comentó que no pueden avalar la consistencia, la disponibilidad y la tolerancia de las particiones al mismo tiempo. (MacCreary & Ann, 2013)

**Consistencia:** Cuando se efectúa una inserción o una consulta en la base de datos, persistentemente debe obtener la misma información sin importar de qué nodo o servidor provenga. Esto significa que los usuarios pueden acceder a los mismos registros al mismo tiempo.

**Disponibilidad:** Es la garantía de que cualquier solicitud al nodo obtenga una respuesta de manera satisfactoria o no satisfactoria.

**Tolerancia a particiones:** Dado que el sistema distribuido está particionado, el sistema debe seguir marchando en caso de que se origine un fallo total o una caída parcial del sistema.



*Figura 1 - Teorema CAP*

Para ser distribuida y escalable, una base de datos no puede satisfacer las tres características paralelamente. La imagen muestra los escenarios que se cumplen en el teorema CAP (Space, 2017), las bases de datos NoSQL se pueden clasificar según las condiciones que cumplen.

Tenga en recuento que esta clasificación no es decisiva, ya que algunas bases de datos NoSQL se pueden configurar para que ya no cumplan ciertas condiciones para diferentes bases de datos.

El teorema CAP satisface las siguientes condiciones:

**AP:** Garantiza la disponibilidad y tolerancia a particiones, pero no la consistencia general, algo de lo cual se puede lograr recreando partes. Esto significa que no hay garantía de que los datos sean siempre los mismos en todos los nodos.

**CP:** Garantiza la consistencia y la tolerancia a particiones, pero debe sacrificar la disponibilidad. Si bien no se puede garantizar una disponibilidad del 100 %, de ninguna manera es inalcanzable o está en un nivel muy bueno.

**CA:** Asegura la consistencia y la disponibilidad, lo cual es un problema en la tolerancia a fallas de la partición, ya que asegura que los datos sean siempre los mismos y que el sistema pueda satisfacer todas las solicitudes. (Fernández, 2017)

***Tabla 1** - Clasificación de bases de Datos NoSQL en función de la condición que cumplen.*

<b>Clasificación de bases de datos NoSQL en función de la condición que cumplen</b>		
<b>Condición</b>	<b>Características</b>	<b>Ejemplos</b>
<b>CA.</b> Consistencia y disponibilidad.	Revalidaciones dobles. Protocolos de validación de cachés.	Relacionales (Oracle, Mysql, SQL Server), Neo4J.
<b>CP.</b> Consistencia y particiones	Bloqueos “pesimistas”. Omitir las particiones más pequeñas.	MongoDB, HBase, Redis.
<b>CP.</b> Consistencia y particiones	Invalidaciones de caché. Resolución de conflictos	DynamoDB, CouchDB, Couchbase, Cassandra.

**Fuente:** (MacCreary & Ann, 2013)

### 1.3.4.3. Tipos de Base de Datos NoSQL

#### 1.3.4.3.1. Base de datos Clave / Valor

Este es uno de los modelos más populares en BBDD NoSQL y uno de los más simples. Este estilo funciona así: cada elemento se identifica con una clave única, lo que agiliza las operaciones de lectura y escritura. La información generalmente se almacena como objetos binarios (BLOB). (AWS, 2020)

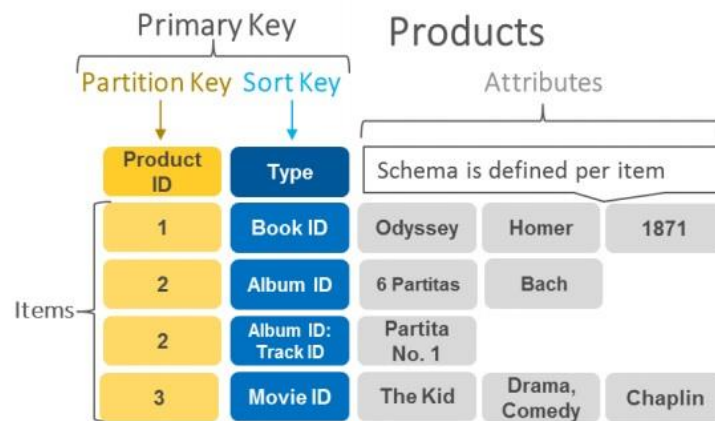
Las características primordiales de una base de datos de clave-valor son:

**Simplicidad.** Las bases de datos clave-valor manejan estructuras de datos muy simples. Las bases de datos estructuradas complejas a veces son innecesarias y siempre debe ahorrarse el trabajo de definir esquemas. Un buen ejemplo es su implementación para registrar datos sobre los estudiantes de GSI y el año en que asistieron. Para manipular los datos, todo lo que necesita hacer es nombrar el conjunto, en nuestro caso "alumno". Cada vez que se incrusta un valor en la colección Student, se asigna su clave. Cada clave puede tener múltiples atributos como nombre y apellido, dirección de correo electrónico, DNI y apellido del alumno asociado. Para hacer una consulta solo necesitas acceder al registro con la clave 001, que revelará todos los valores agrupados a esa clave y también puedes modificar el valor.

**Velocidad.** Esta es una de sus primordiales características ya que gracias a su sencilla estructura de datos puede proporcionar operaciones de alta utilidad en poco tiempo. Una forma de realizar operaciones rápidas es almacenar datos en la memoria porque examinar y escribir datos es mucho más rápido en la RAM que en el disco. Sin embargo, cuando los datos se almacenan en la RAM en lugar de en el disco, la RAM tiene la desventaja de la falta de capacidad de almacenamiento persistente y las dificultades que pueden nacer como resultado de la pérdida de datos, como en el caso de una falla de energía o el servidor está inactivo. Si la aplicación cambia el valor agrupado con una clave, la base de datos de clave-valor solo actualiza la entrada de RAM que acopia ese valor. Asimismo, envía un mensaje a la aplicación de que el valor se ha actualizado y

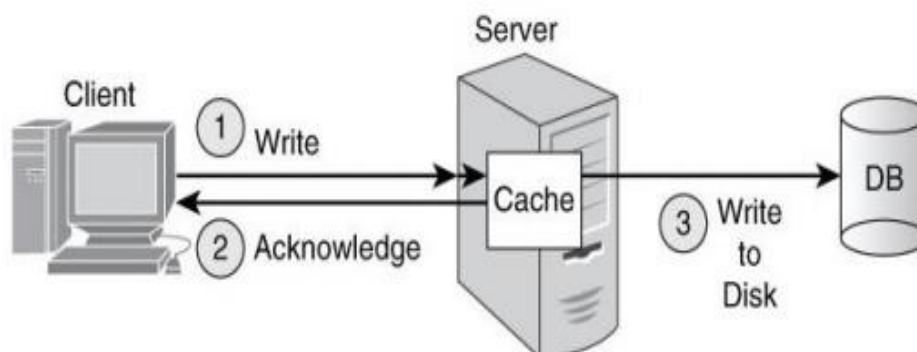
la aplicación continúa haciendo otras cosas mientras la base de datos escribe el valor actualizado en el disco en la RAM, por lo que no se pierden datos.

La ejecución de operaciones de lectura también es más aligera si los datos se almacenan en la memoria. El inconveniente que puede surgir es que la cantidad de memoria puede ser mucho menor que el disco, por lo que hay que usar un algoritmo para elegir qué parte de la memoria hay que liberar para que otros datos ocupen espacio. Por lo general, con el algoritmo LRU (Least Recently Used), los datos que no se han utilizado durante más tiempo se eliminan y se libera espacio para almacenar nuevos datos. (AWS, 2020)



**Figura 2** - Estructura de una Base de Datos NoSQL Clave - Valor.

**Escalabilidad.** Debe poder añadir o eliminar servidores del conjunto en función de la carga del sistema. Cuando se agregan o eliminan servidores, los datos y la actividad deben replicarse. Los dos métodos de extensión son la replicación maestro-esclavo y la replicación sin maestro-esclavo. (AWS, 2020)



**Figura 3** - Ejemplo de escritura en una BBDD clave-valor.

#### 1.3.4.3.2. Base de Datos Documental

"Una base de datos documental es una base de datos no relacional diseñada para acumular datos parcialmente estructurados".

Los desarrolladores pueden utilizar las bases de datos de documentos de manera intuitiva porque los datos a nivel de aplicación a menudo se representan en un formato de documento mejor que en forma tabular. Cada documento es autodescriptivo, con o sin la misma estructura de datos, puede tener un esquema único y no depende completamente de ningún otro documento. Los documentos se congregan en colecciones que cumplen el mismo propósito que las tablas en una base de datos relacional.

Estos tipos de bases de datos utilizan documentos para acumular información. En cuanto al documento, no se refiere a un documento de texto, normalmente un archivo XML o JSON, aunque se logran utilizar otros tipos como YAML.

```
<writer>
  <writername>Gabriel García Márquez</writername>
  <books>
    <book>
      <bookname>Crónicas de una muerte anunciada</bookname>
      <datereleased>1981</datereleased>
      <genre>Novela de ficción</genre>
      <numpag>144</numpag>
    </book>
    <book>
      <bookname>Cien años de soledad</bookname>
      <datereleased>1967</datereleased>
      <genre>Novela de ficción</genre>
      <numpag>496</numpag>
    </book>
  </books>
</writer>
```

*Figura 4 - Ejemplo de un documento XML.*



```

{
  "id" : 1,
  "writername": "Gabriel García Márquez",
  "books" : [
    {
      "bookname" : "Crónicas de una muerte anunciada",
      "datereleased" : 1981,
      "genre" : "Novela de ficción",
      "numpag" : 144
    },
    {
      "bookname" : "Cien años de soledad",
      "datereleased" : 1967,
      "genre" : "Novela de ficción",
      "numpag" : 496
    }
  ]
}

```

*Figura 5 - Ejemplo de un documento JSON.*

Las colecciones se crean en la base de datos donde se acumulan los documentos a medida que se insertan. Por lo general, los documentos almacenados en la colección están afines con la misma entidad, es decir, en el "libro" de la colección, los documentos que almacenan datos contables, no datos de animales, no deben almacenar documentos en una colección que no está relacionada con ellos. (información, 2018)

Estas bases de datos están diseñadas para permitirle distinguir entre diferentes documentos en una colección para que pueda encontrar una gran cantidad de documentos con diferente distribución, de ahí el tipo de base de datos. Estos datos son muy flexibles o están libres de esquemas y no necesitan comience a guardar datos como otros tipos, predefinidos. (Sullivan, 2015)

En las bases de datos de documentos, no es necesario utilizar ID de documentos con pares clave-valor. Aquí puede buscar documentos por valor, por ejemplo, en las Figuras 4 y 5, si estuviera buscando un libro escrito en 1981, el resultado sería "Crónicas de la Profecía". Esto se debe al uso de listas para mejorar la eficiencia de las operaciones en conjuntos de documentos. Indexa todos los datos en un solo documento en la base de datos, para que pueda

encontrar rápidamente grupos de documentos con los mismos atributos. (MacCreary & Ann, 2013)

#### 1.3.4.3.2.1. Diseño de una base de datos documental

Alguien comentó anteriormente que las bases de datos de documentos son muy flexibles. Comenzar con el modelado de base de datos requiere una comprensión de la normalización y la desnormalización. Estos conceptos se interpretan de manera diferente en una base de datos de este tipo que en el caso de una base de datos relacional.

Estos archivos se normalizarán si se refieren a otros archivos donde se puede buscar información adicional. Por ejemplo, hay una colección de coleccionistas y hay una colección de documentos académicos, para no duplicar su material en los registros de cada persona, se recomienda que se refiera a un documento archivado acumulado en otra colección mostrando su documento único identificador, Por lo tanto, cuando usamos varias colecciones para acumular datos relacionados, se dice que normalizamos.



**Figura 6 - Estructura de una Base de Datos NoSQL Documental.**

Estas bases de datos brindan diferentes opciones para la recuperación de datos, como revelar todos los documentos posteriores a una fecha específica, documentos de un tipo específico, documentos con atributos específicos, etc.

Para realizar la búsqueda se utiliza un procesador de consultas, que acepta consultas de entrada, datos de documentos y colecciones, y crea una serie de operaciones de recuperación de datos.

Este procesador de consultas puede tomar sus propias decisiones para optimizar el rendimiento de las consultas.

Tenga en cuenta también que los documentos están sujetos a cambios, por lo que cuando los crea, el sistema de administración de la base de datos les asigna espacio. El volumen de este espacio incluirá el tamaño del documento más otras dimensiones asignadas cuando modifique y amplíe el documento. Si el documento supera el tamaño reservado, el sistema tendrá que moverlo a otro lugar, con una penalización de rendimiento, ya que el documento debe leerse, copiarse en otro lugar y liberar el espacio que ha ocupado antes. Para evitar la pérdida de rendimiento, debe asignar suficiente espacio para su crecimiento para que no tenga que cambiar el lugar del documento.

Al diseñar una base de datos con estas tipologías, es importante elegir el número adecuado de índices según sea necesario. Si su aplicación lee la mayoría de las operaciones, tendrá que agregar más índices e indexar la totalidad de las propiedades del documento, ya que es probable que se utilicen muchos campos. Filtra tu búsqueda. Por otro lado, si la mayoría de las operaciones en nuestra base de datos son registros, habrá menos índices implementados porque también ejecutan recursos. Dicho esto, lo correcto es equilibrar la cantidad de índices realizados en la base de datos con la funcionalidad proporcionada.

Las bases de datos documentales tienen muchos usos y características comunes:

**Uso de un modelo de datos basado en árboles:** En una base de datos NoSQL de documentos, se puede cargar cualquier tipo de datos sin que la base de datos tenga conocimiento previo de la estructura de los datos o de lo que significan los valores. Esta flexibilidad hace que estas bases de datos sean excelentes tanto para crear prototipos de una solución en un proceso de desarrollo "ágil" como para permitir cambios en los datos almacenados tras la puesta en marcha de un sistema. No es necesario rediseñar un esquema complejo dentro de la base de datos para cada pequeño cambio. Este no es el

caso de los sistemas tradicionales de gestión de bases de datos relacionales (RDBMS).

Los clones Bigtable le permiten administrar conjuntos de datos; es decir, cuando un clon Bigtable recibe una clave de fila, devuelve un conjunto de familias de columnas. Cuando una mesa grande recibe una clave de fila y un calificador de familia de columnas, devuelve un conjunto de columnas, cada una con un valor de celda (algunas con múltiples valores de celda con diferentes marcas de tiempo).

Los clones Bigtable le ofrecen tres niveles de conjuntos: fila, familia y columna, que se pueden representar en un modelo de árbol. La Figura 1 muestra un ejemplo de la aplicación de pedidos en línea estructurada para un clon de Bigtable.

Cuando queremos bajar otro nivel es aquí en donde entra en juego una base de datos documental, proporcionando la flexibilidad que se necesita. En la figura 2 es un ejemplo de un documento de pedido con información mejorada y amplia flexibilidad.

Es libre de usar una estructura menos jerárquica y más plana. En la figura 3 muestra la gestión de un archivo de registro como una estructura de árbol.

En este caso, un informe de error de seguimiento de pilas puede ser una estructura de árbol. Podría, por ejemplo, volcar información sobre cada proceso de ejecución en vivo en un archivo, en lugar de sólo la sección de código que reportó el error. Este enfoque es particularmente útil para la depuración paralela.

En un clon de Bigtable, su aplicación debe administrar la conversión de la estructura jerárquica de documentos anterior a y desde una estructura de árbol, mientras que una base de datos de documentos puede administrarla de forma nativa. El almacenamiento de datos de documentos sin "destruir" datos a través de un conjunto de tablas es potencialmente útil si desea consultar un recuento de informes de error para ver qué aplicaciones causaron la mayoría de los errores en un módulo de código en particular. Esto requiere un trabajo de

análisis de lotes en una Bigtable, pero una simple consulta de agregación en una base de datos de documentos con indexación de elementos habilitada.

**Gestión de documentos de artículo:** Las estructuras de los árboles varían mucho. Un formato semiestructurado como el XHTML (el formato utilizado por las páginas web) tiene, como su nombre indica, cierta estructura, pero se pueden modelar los párrafos, encabezados y secciones individuales de varias maneras.

Una práctica común en las bases de datos de documentos es indexar una propiedad, o un elemento, sin importar dónde se encuentre dentro de un documento. Por ejemplo, puede indexar todos los elementos h1 (encabezados principales) y h2 (subencabezados) independientemente de dónde se encuentren. Tanto MarkLogic Server como Microsoft DocumentDB permiten este estilo de indexación.

Las bases de datos de documentos son excelentes para proporcionar consultas consistentes sobre estas estructuras de variables. Hay muchos ejemplos de la vida real de consultas sobre estructuras de documentos:

Las trazas de pila en los archivos de registro pueden tener un campo de nombre de clase a diferentes profundidades en una estructura de árbol.

Los registros médicos de un paciente pueden mencionar un medicamento o una afección en cualquier parte del campo de notas de texto.

**Descubrir la estructura del documento:** Las bases de datos documental tienden a almacenar documentos en un formato comprimido en disco. Para ello, las bases de datos deben comprender el formato de los documentos que reciben. Cuando se envían documentos JSON o XML, por ejemplo, una base de datos utiliza esa estructura para gestionar mejor los datos en el disco.

He aquí un ejemplo de un documento JSON que representa un pedido en línea:

MongoDB almacena los documentos en su propia representación binaria BSON, lo cual es útil porque JSON, como en el ejemplo anterior de JSON, tiene

mucho texto en los nombres de las propiedades. Puede ahorrar espacio comprimiendo o gestionando estos nombres de propiedad como simples identificadores numéricos, en lugar de como cadenas largas.

**Soporte de documentos no estructurados:** La información totalmente desestructurada es realmente rara. Es más típico utilizar un formato de contenedor como JSON o XML y almacenar grandes cantidades de texto plano en propiedades y elementos particulares (Anderson & Slater, 2010).

Sin embargo, hay situaciones en las que se recibe mucho texto o datos binarios (piense en la carpeta Mis Documentos promedio). He aquí un par de formas de gestionar grupos de documentos recopilados desde un dispositivo de almacenamiento jerárquico de este tipo:

Las colecciones de archivos pueden contener una combinación de:

- a. Archivos estructurados (como información de gastos CSV)
- b. Ficheros semiestructurados (como XML y páginas web HTML guardadas)
- c. Archivos no estructurados (como imágenes JPEG, películas, MP3s, Word documents, PDFs y archivos de texto plano).

En realidad, los formatos no estructurados son en realidad semiestructurados. que normalmente no le das instrucciones a tu base de datos para que las entienda. Sin embargo, es posible que le falte alguna información útil que usted podría quieren extraer y buscar. Por ejemplo, las imágenes JPEG pueden contener metadatos sobre:

- a. La cámara que tomó las imágenes
- b. Las condiciones prevalecientes cuando se tomó la imagen
- c. Las coordenadas GPS y la elevación donde se tomó la imagen

Algunas bases de datos vienen con soporte incorporado para extraer estos metadatos y texto plano de archivos binarios. Esto es ideal para indexar el texto

plano para la búsqueda o para proporcionar una gestión más estructurada de los archivos". metadatos.

MarkLogic Server, por ejemplo, incluye soporte para más de 200 formatos binarios a través de su uso de librerías de extracción de datos binarios. Estos proporcionar una salida XHTML, usando meta tags para los metadatos y el cuerpo para contenido de texto. Puede integrar otras soluciones con otros documentos bases de datos para permitir la extracción automática de información sobre la ingesta.

Muchas bases de datos documentales apoyan el concepto de una ruta URI como una ruta primaria o un ID de documento unívoco. Piensa en esta ruta como si fuera un archivo con una jerarquía innata. Algunas bases de datos de documentos permiten para listar los documentos almacenados bajo una URI particular y parcial. En este puede representar un sistema de ficheros en una base de datos de documentos. Algunos Las bases de datos NoSQL consideran el ID único como externo al documento (MarkLogic), mientras que otros (MongoDB y Microsoft DocumentDB) utilizar una propiedad de identificación especial dentro del documento. (Folwer, 2015)

#### **1.3.4.3.2.2. Base de Datos Documental en una Empresa**

Si una base de datos le dice que los datos se guardan, entonces es probable que confíe en esa garantía. En casos de uso crítico, esta dependencia es vital. Además, muchas industrias no sólo quieren que los cambios en sus datos sean accesibles inmediatamente después de una actualización; también quieren que los índices utilizados para encontrar los datos se mantengan actualizados, reflejando el estado actual de la base de datos.

Otro problema es la distribución de los datos en torno a un grupo de bases de datos. Mediante la distribución de datos, se mejora la velocidad de escritura. Sin embargo, la distribución de datos también significa que puede sacrificar la velocidad de las operaciones de lectura y también de las consultas. Esto se debe a que las consultas deben ser manejadas por muchos servidores. Es importante entender las ventajas y desventajas de cada situación.

Las bases de datos NoSQL de documentos difieren entre sí en la forma en que proporcionan las características anteriores. Son fundamentales para las empresas que quieren apostar por las nuevas tecnologías para obtener ventajas competitivas.

**Sharding:** Si todos sus nuevos datos llegan al último servidor de su clúster, las operaciones de escritura se verán afectadas, y a medida que el servidor sobrecargado se esfuerza por mantenerse al día, las operaciones de lectura también se verán afectadas. Aquí es donde interviene Sharding.

Sharding es el proceso de asegurar que los datos se distribuyan uniformemente a través de un grupo entero de servidores. Los fragmentos se pueden configurar en el momento en que se implementa un cluster (MongoDB), se pueden fijar en un número de cubos o particiones que se pueden mover más tarde (Couchbase, Microsoft DocumentDB), o se pueden gestionar automáticamente simplemente moviendo un número de documentos entre servidores para mantener las cosas equilibradas (MarkLogic Server).

**XML y bases de datos XML:** Las primeras bases de datos de documentos se crearon en torno al estándar de documentos XML. Las bases de datos XML nos interesan principalmente como precursores arquitectónicos de la moderna base de datos documental de JSON; las bases de datos XML representan hoy en día un nicho importante pero pequeño en el mercado general de bases de datos.

XML (eXtensible Markup Language) surgió como resultado de la convergencia de esfuerzos para desarrollar un lenguaje de marcado generalizado como sucesor de varios formatos especializados como SGML y de la comprensión de que HTML -la base de la Web 1.0- combinaba de manera incómoda el diseño y los datos. XML era capaz de representar casi cualquier tipo de información y, junto con las hojas de estilo en cascada (CSS) que controlaban el renderizado, permitían a los sitios web de segunda generación separar datos y formatos.

XML fue ampliamente utilizado más allá de estos casos de uso de la Web 2.0 y se convirtió en un formato estándar para muchos tipos de documentos, incluyendo eventualmente documentos de procesamiento de texto y hojas de



cálculo. XML es también la base de muchos protocolos de intercambio de datos y, en particular, fue la base de especificaciones de servicios web como SOAP (Simple Object Access Protocol).

A principios de la década de 2000, se esperaba que la mayoría de los documentos de una organización que no formaban parte de una base de datos relacional acabaran representados como XML. Y aunque el impulso de XML se ha ralentizado, hoy en día una gran variedad de tipos de documentos utiliza XML bajo la cubierta.

**Herramientas y estándares XML:** XML está soportado por un rico ecosistema que incluye una variedad de estándares y herramientas para ayudar en la creación, validación, búsqueda y transformación de documentos XML. Estos incluyen:

- a. **XPath:** Una sintaxis para recuperar elementos específicos de un documento XML. XPath proporciona una manera simple y conveniente de filtrar documentos usando comodines y referencias de etiquetas.
- b. **XQuery:** Un lenguaje de consulta para interrogar documentos XML. La especificación relacionada XQuery Update proporciona mecanismos para modificar un documento. A XQuery a veces se le llama "el SQL de XML".
- c. **Esquema XML:** Un tipo especial de documento XML que describe los elementos que pueden estar presentes en una clase específica de documentos XML. El esquema XML se puede utilizar para validar que un documento está en el formato correcto, o para ayudar a los programas que deseen interrogar documentos XML que se ajusten a dicho formato.
- d. **XSLT (Extensible Stylesheet Language Transformations):** Un lenguaje para convertir documentos XML en formatos alternativos, incluyendo formatos no XML como HTML.

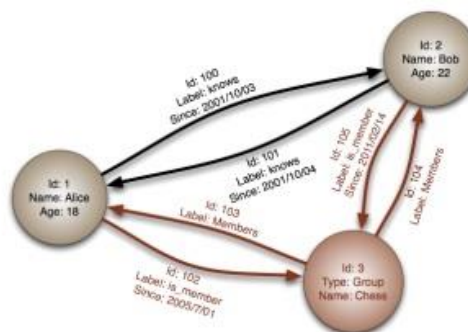
- e. **DOM (modelo de objeto de documento):** Una API orientada a objetos que los programas pueden utilizar para interactuar con XML, XHTML y documentos de estructura similar

#### 1.3.4.3.3. Base de datos orientadas a grafos

Desde el principio, es muy bueno estar al tanto y entender qué es un esquema. Un gráfico es una colección de objetos llamados nodos que están conectados por arcos o aristas para que pueda mostrar las relaciones entre estos nodos. Entonces, este tipo de base de datos recibió este nombre porque la información que almacena se representa como nodos del gráfico y las relaciones como bordes para que pueda atravesarse a través de la teoría de grafos. Cada botón tiene propiedades diferentes.(acens, 2014)

Son útiles cuando tiene elementos afines entre sí de forma compleja y estas relaciones tienen propiedades similares a las de los nodos. Pueden realizar consultas muy simples, como consultar las propiedades de los nodos, y consultas muy complejas sobre las relaciones entre los nodos para encontrar patrones. A los nodos y relaciones se les asigna un identificador único. La desventaja es que son difíciles de escalar a muchos servidores, se pueden escalar a muchos servidores para mejorar el rendimiento de lectura, pero escribir en varios servidores es difícil debido a la relación de los nodos.

La consulta se ejecuta más rápido en una base de datos relacional porque en lugar de realizar la unión, sigue un arco de nodo a nodo, lo que es un trabajo más sencillo y menos costoso. Su modelo es muy simple, pero requiere estandarización.(Sullivan, 2015)



**Figura 7 - Estructura de una Base de Datos NoSQL orientado a Grafos.**

#### 1.3.4.3.4. Base de Datos Columnares

Usan almacenamiento establecido en columnas para comprimir la sobrecarga de E/S del disco. Se utilizan en muchos tipos diferentes de aplicaciones metódicas porque admiten guardar rápidamente y recuperar fácilmente grandes cantidades de datos. Estas bases de datos comparten algunas similitudes con las bases de datos relacionales y, de hecho, algunas de sus tipos son comunes a ambas. Un ejemplo es que ambos utilizan una clave principal, que se denomina clave de fila en el caso de las bases de datos en columnas. En ambas bases de datos, estas claves están indexadas para una recuperación ágil de registros.

Hay otras funciones que tienen el mismo aspecto, pero se implementan de manera diferente. En este argumento se almacena en columnas. Mientras está viendo los datos en la columna, los datos en realidad se almacenan en un mapa que apunta al mapa de valores en la columna. La diferencia entre una base de datos y otra es que en las bases de datos columnares no existen relaciones y no se pueden realizar transacciones de varias filas.

38383	48282	59595
Dell Laptop	Apple iPhone	Galaxy Tab S

*Figura 8 - Ejemplo de tabla de una BBDD Columnar.*

#### 1.3.5. Norma ISO/IEC 25000

Sirve para evaluar la calidad del software, cuya familia de normas es derivación de la evolución de algunas normas anteriores, principalmente de la norma ISO/IEC 9126, no obstante, se tiene cuatro divisiones. Pero solo hablaremos de la ISO/IEC 2501n, que es donde nos centraremos en nuestra investigación.

##### 1.3.5.1. ISO/IEC 2501n – División de Modelo de Calidad

Esta sección de normas presenta modelos de calidad donde interviene calidad externa, interna y en si el uso del producto de software, está conformada por:

**ISO/IEC 25010:** Modelo de calidad para el producto software y la calidad en operación, presenta características y sub-características para evaluar el software. (NORMA ISO, 2020)

### 1.3.5.2. 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 2. (ISO 25000, 2005)

**Tabla 2 - Características de ISO/IEC 25010.**

<b>Factor</b>	<b>Descripción</b>	<b>Criterios</b>
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.

*Fuente:* (NORMA ISO, 2020)

#### **1.4. Formulación del problema.**

¿Qué gestor de base de datos NoSQL tendrá mejor rendimiento en el procesamiento de grandes conjuntos de datos?

#### **1.5. Justificación e importancia del estudio.**

##### **1.5.1. Justificación tecnológica**

La presente investigación está enmarcada dentro de la línea de investigación de Infraestructura, Tecnología y Medio Ambiente de la Escuela profesional de ingeniería de sistemas, de la USS, y aporta conocimientos sobre el rendimiento de los gestores de base de datos NoSQL Documental en lo que es procesamiento de volcado de información. Está diseñado para realizar un análisis comparativo de rendimiento de dos sistemas de gestión de base de datos NoSQL, en el procesamiento de volcado de información; pues la inserción masiva proporciona una forma muy eficaz de copiar grandes cantidades de datos a una tabla esto ayudará a las empresas a obtener nuevos conocimientos para la mejora de los procesos, que conduzcan a la toma de mejores decisiones.

Según (IBM, 2019) las inserciones masivas de datos pueden aumentar el rendimiento de las operaciones de inserción en el origen de datos.

##### **1.5.2. Justificación económica**

Este estudio proporcionará a las empresas una comprensión más amplia del rendimiento de los administradores de bases de datos NoSQL MongoDB y CouchDB, de modo que las empresas puedan elegir la base de datos NoSQL

que sea mejor para su negocio sin incurrir en la sobrecarga innecesaria de implementar un administrador de bases de datos NoSQL sin saber qué resultados Puede ser obtenido.

### **1.5.3. Justificación ambiental**

Cuando se trata del uso de computadoras, cualquier forma de electricidad que genere calor contribuye al calentamiento global, no en grandes cantidades, pero si se convierte en parte del problema global. La identificación de ciertas deficiencias, como la redundancia innecesaria de procesos, guiará en cierta medida la optimización de los recursos de hardware y el tiempo, lo que reducirá el consumo de energía.

## **1.6. Hipótesis.**

MongoDB tendrá mejor rendimiento en el procesamiento de grandes conjuntos de datos?

## **1.7. Objetivos.**

### **1.7.1. Objetivos General.**

Comparar los gestores de base de datos NoSQL Documental para determinar su rendimiento en el procesamiento de grandes conjuntos de datos.

### **1.7.2. Objetivos Específicos.**

- a. Seleccionar los gestores de bases de datos NoSQL en base a la evaluación de sus características.
- b. Construir ambiente para pruebas.
- c. Realizar las pruebas de rendimiento.
- d. Interpretar los resultados.

## **II. MATERIAL Y MÉTODO**

### **2.1. Tipo y Diseño de Investigación**

#### **2.1.1. Tipo de investigación**

El tipo de investigación de este proyecto de acuerdo a su naturaleza, corresponde al tipo Cuantitativa, porque los indicadores están representados por valores numéricos, con el cual se medirán resultados para poder emitir estadísticas.

#### **2.1.2. Diseño de investigación**

El diseño de la investigación corresponde al tipo cuasi experimental porque la población y la muestra han sido definidas por el investigador con base a los artículos científicos de los gestores de base de datos NoSQL documental. Asimismo, la variable independiente será manipulada, lo que tendrá un impacto en la variable dependiente.

### **2.2. Población y muestra.**

#### **2.2.1. Población**

Para determinar la población se seleccionó los 08 primeros gestores de base de datos más populares de acuerdo al ranking elaborado por la página DB-Engines, asimismo se realizó un estudio teórico de ellos, y estos son: MongoDB, Amazon DynamoDB, Microsoft Azure CosmosDB, CouchBase, Apache CouchDB, Firebase Realtime Database, MarkLogic y Reino.

**Tabla 3 - Población de los Gestores de Base de Datos NoSQL Documental.**

Nº	BASE DE DATOS NOSQL	LICENCIA
1	Mongo DB	Libre
2	Amazon DynamoDB	Propietario
3	Microsoft Azure Cosmos DB	Propietario
4	Couchbase	Propietario/ Libre
5	Apache CouchDB	Libre
6	Firebase Realtime Database	Propietario
7	MarkLogic	Propietario
8	Reino	Libre

**Fuente:** Elaboración Propia.

Rango			DBMS	Modelo de base de datos	Puntuación		
Oct. De 2020	Sep. De 2020	Oct. De 2019			Oct. De 2020	Sep. De 2020	Oct. De 2019
1.	1.	1.	MongoDB	Documento	448,02	+1,54	+35,93
2.	2.	2.	Amazon DynamoDB	Multi-modelo	68,41	+2,23	+8,24
3.	3.	4.	Microsoft Azure Cosmos DB	Multi-modelo	32,01	+0,34	+0,68
4.	4.	3.	Couchbase	Documento , multimodelo	30,33	-0,28	-1,88
5.	5.	5.	CouchDB	Documento	17,41	+0,16	-0,63
6.	6.	7.	Firebase Realtime Database	Documento	16,26	+0,65	+4,52
7.	7.	6.	MarkLogic	Multi-modelo	11,73	-0,21	-1,33
8.	8.	8.	Reino	Documento	8,74	0,00	+0,82

**Figura 9 - Ranking de los Gestores de Base de Datos NoSQL Documental.**

### 2.2.2. Muestra

La muestra de esta investigación está compuesta por dos Gestores de Base de Datos Documental: MongoDB y Apache CouchDB, las cuales han sido seleccionadas por las siguientes características: Tipo de datos, Modelo de datos, plataforma y tipo de licencia.



**Tabla 4 – Características de los Gestores de Base de Datos NoSQL Documental.**

Nº	BASE DE DATOS NOSQL	TIPO DE DATOS	MODELO DE DATOS	PLATAFORMA	LICENCIA
1	Mongo DB	BSON /JSON	Documento	Linux/ OS X/Solaris/Windows	Libre
2	Amazon DynamoDB	JSON	Multimodelo	Alojado (Nube)	Propietario
3	Microsoft Azure Cosmos DB	JSON	Multimodelo	Alojado (Nube)	Propietario
4	Couchbase	JSON	Multimodelo	Linux/ OS X/Windows	Propietario/ Libre
5	Apache CouchDB	JSON	Documento	Android/ BSD/ Linux/OS X/ Solaris/ Windows	Libre
6	Firebase Realtime Database	JSON	Documento	Alojado (Nube)	Propietario
7	MarkLogic	JSON	Multimodelo	Linux/ OS X/Windows	Propietario
8	Reino	JSON	Documento	Windows	Libre

**Fuente:** *Elaboración Propia.*

MongoDB y Apache CouchDB fueron seleccionados porque obtuvieron un mayor puntaje en la evaluación de sus características:

- a) **Modelo de datos:** porque los dos gestores de base de datos NoSQL seleccionados son netamente del tipo JSON.
- b) **Modelo de datos:** porque los dos gestores de base de datos NoSQL seleccionados son netamente del tipo documental.

c) **Plataforma:** porque los dos gestores de base de datos NoSQL seleccionados trabajaron con el sistema operativo Linux.

d) **Tipo de licencia:** porque los dos gestores de base de datos NoSQL seleccionados son de licencia libre.

No se consideraron los otros gestores de base de datos NoSQL documental, porque no alcanzaron el puntaje mayor en la evaluación de sus características.

*Tabla 5 - Muestra de los Gestores de Base de Datos NoSQL Documental.*

N°	BASE DE DATOS NOSQL	TIPO DE DATOS	MODELO DE DATOS	PLATAFORMA	LICENCIA
1	Mongo DB	BSON /JSON	Documento	Linux/ OS X/Solaris/Windows	Libre
2	Apache CouchDB	JSON	Documento	Android/ BSD/ Linux/OS X/ Solaris/ Windows	Libre

*Fuente: Elaboración Propia.*

### **2.3. Variables, Operacionalización.**

#### **2.3.1. Variable independiente**

Comparación de Gestores de Bases de Datos NoSQL.

#### **2.3.2. Variable dependiente**

Rendimiento en el procesamiento de grandes conjuntos de datos.

### 2.3.3. Operacionalización

*Tabla 6 – Operacionalización de variables*

Variable independiente	Dimensiones	Indicadores	Fórmulas	Técnicas e instrumentos de recolección de datos
Comparación de Gestores de Bases de Datos NoSQL.	Utilización de Recursos	Promedio de Utilización RAM	$PUR = UF - UI$	Ficha de Registro
		Promedio de Utilización CPU	$PUC = UF - UI$	
Variable dependiente	Dimensiones	Indicadores	Fórmulas	Técnicas e instrumentos de recolección de datos
Rendimiento en el procesamiento de grandes conjuntos de datos.	Comportamiento en el Tiempo	Tiempo de Respuesta en el procesamiento de datos	$TR = TF - TI$	Ficha de Registro
	Volumen	Espacio en disco para almacenamiento de base de datos	$ED = EUF - EUI$	
	Tolerancia a Fallos	Cantidad de errores en el procesamiento de datos	$CE = NEE$	

*Fuente: Elaboración Propia.*

### 2.4. Técnicas e instrumentos de recolección de datos, validez y confiabilidad.

La técnica que se utilizó en este estudio es la observación científica y como instrumento la ficha de registro. A continuación, se explican los componentes de esta técnica y herramienta.

## **2.4.1. Técnica de recolección de datos**

### **2.4.1.1. Observación científica**

Es una de las técnicas que se utilizó en el análisis del proceso de investigación. Consiste en visualizar los eventos que se desarrollan durante la investigación, en el cual el investigador es participe de las actividades llevadas a cabo.

La observación hace que el investigador determine que se está llevando a cabo, como se está desarrollando, quien lo está haciendo, cuando se está desarrollando, que tiempo está tomando, donde se desarrolla y porque se desarrolla.

## **2.4.2. Instrumento de recolección de datos**

### **2.4.2.1. Ficha de registro**

La ficha de registro, es el instrumento donde se registró los datos obtenidos de los sucesos presentados durante la observación, con el fin de presentar resultados.

## **2.5. Procedimiento de análisis de datos.**

Una vez seleccionado nuestros gestores de base de datos y preparado nuestro entorno de trabajo, se procederá a la recolección de datos de la siguiente manera:

**2.5.1. Observación científica:** esta técnica nos permitió observar los sucesos transcurrido durante las pruebas realizadas, como el comportamiento de los gestores de base de datos NoSQL documental durante su análisis, el tiempo que toman en la realización de las pruebas, la utilización de recursos, tolerancia a fallos y volumen utilizado.

**2.5.2. Ficha de Registro:** este instrumento nos mostró los resultados obtenidos de las pruebas realizadas a los gestores de base de datos, como: tiempo de respuesta de las transacciones, volumen utilizado, cantidad de errores

presentados, cantidad de uso de memoria RAM y cantidad de uso de CPU. Mayor detalle en ANEXO 1.

### 2.5.3. Análisis estadístico e Interpretación de los datos

Después de obtener los datos de la ficha de registro, se detalló los datos obtenidos realizando cuadros estadísticos del análisis e interpretación de los resultados obtenidos durante la investigación. A continuación, se muestra las fórmulas para procesar nuestros datos:

- a) **Promedio de utilización de RAM:** Es la resta del uso final y el uso inicial de la memoria RAM. Siendo su unidad de medida en megabyte.

$$PUR = UF - UI$$

**Donde:** *PUR* = Promedio de utilización de RAM; *UF* = Uso Final; *UI* = Uso Inicial.

- b) **Promedio de utilización de CPU:** Es la resta del uso final y el uso inicial del CPU. Siendo su unidad de medida en megabyte.

$$PUC = UF - UI$$

**Donde:** *PUC* = Promedio de Utilización de CPU; *UF* = Uso Final; *UI* = Uso Inicial.

- c) **Tiempo de Respuesta en el procesamiento de datos:** Es la resta del tiempo final y el tiempo inicial del volcado de información. Siendo su unidad de medida en segundos o minutos.

$$TR = TF - TI$$

**Donde:** *TR* = Tiempo de Respuesta; *TF* = Tiempo Final; *TI* = Tiempo Inicial

- d) **Espacio en disco para almacenamiento de base de datos:** Es la resta del espacio utilizado final y el espacio utilizado Inicial. Siendo su unidad de medida en porcentaje.

$$ED = EUD - EUI$$

**Donde:** *ED= Espacio de Disco; EUD= Espacio Utilizado Final;*

*EUI = Espacio Utilizado Inicial*

- e) **Cantidad de errores en el procesamiento de datos:** Es la cantidad de errores encontrados durante el procesamiento de grandes conjuntos de datos. Siendo su unidad de medida en números.

$$CE = NEE$$

**Donde:** *CE= Cantidad de Errores; NEE= Número de errores encontrados*

## 2.6. Criterios éticos.

Los criterios éticos que se tomó en este trabajo de Investigación, están relacionados con cualquier suceso que se abarcó esta investigación. Por ellos se tomó los siguientes criterios éticos.

### 2.6.1. Confidencialidad

Los códigos de ética hicieron énfasis a la seguridad y el resguardo de la identidad de las fuentes como también de los autores a quien hice referencia.

### 2.6.2. Derechos de Autor

El material que se usó para el proceso del trabajo de investigación estuvo referenciado y citado con sus respectivos autores como colaboradores en dar aportes a esta investigación.

## 2.7. Criterios de rigor científico.

### 2.7.1. Fiabilidad

El proyecto es consistente con las opiniones expresadas en su contenido y su implementación es consistente con las políticas y estándares de desarrollo del proyecto.

### **2.7.2. Validez**

Los datos que se logró obtener por el proyecto fueron evaluados y analizados por ingenieros especializados en el tema para determinar su autenticidad.

### **2.7.3. Consistencia**

El trabajo de investigación es bibliografía coherente avalada por la comunidad científica.

### III. RESULTADOS

#### 3.1. Resultado en Tablas y Figuras.

Para la evaluación de resultados, se realizó cuatro pruebas: volcado de datos, listado de datos, actualización de datos y eliminación de datos; asimismo se ejecutó distintas consultas en los gestores de base de datos NoSQL Documental, de los cuales se obtuvo los siguientes indicadores: Uso de memoria RAM, Uso de CPU, Tiempo de respuesta, Espacio usado en disco y Cantidad de errores.

##### 3.1.1. Resumen de evaluación cuantitativa de los gestores de base de datos NoSQL.

###### A) Prueba uno

En la prueba uno, se realizó dos operaciones de volcado de datos para ambos gestores de base de datos NoSQL documental con las colecciones establecimientos y egresoshospitalarios; donde se evaluaron los siguientes indicadores: Uso de Memoria RAM, uso de CPU, Tiempo de Respuesta, Espacio Usado en disco y Cantidad de errores. Asimismo, se mostró los valores que arrojó la prueba uno

###### 1. Uso de memoria RAM

*Tabla 7 - Resumen del Uso de Memoria RAM - Prueba uno.*

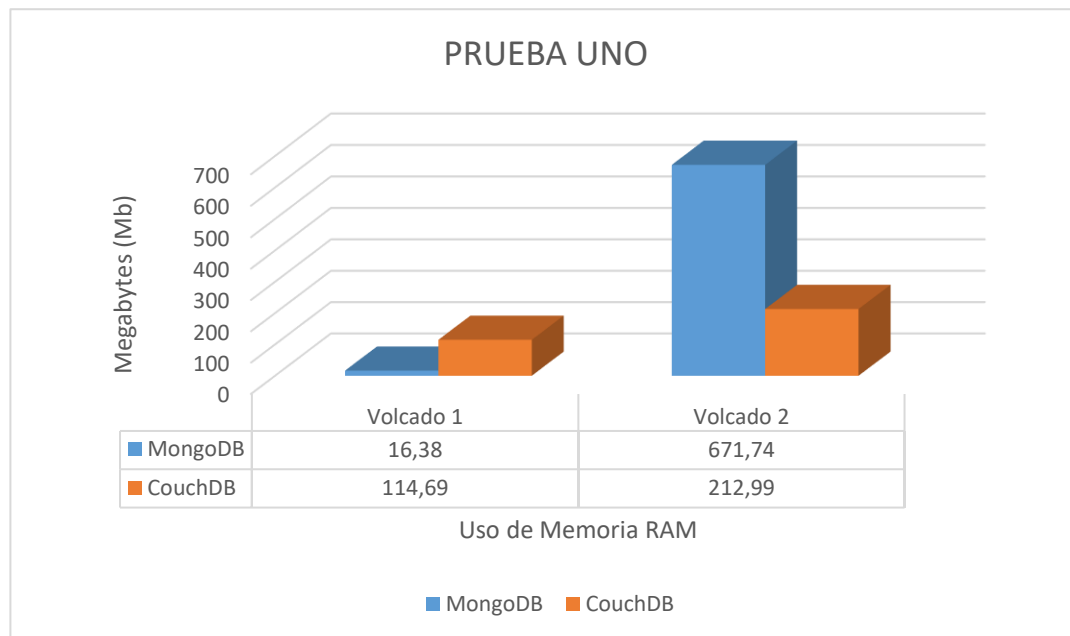
Gestores de Base de Datos NoSQL	Volcado 1	Volcado 2
MongoDB	16.38 Mb	671.74 MB
CouchDB	114.69 MB	212.99 MB

*Fuente: Elaboración Propia.*

Según esta tabla, se observó que MongoDB empleó menos uso de memoria RAM para el volcado 1, mientras que para el Volcado 2 uso más memoria RAM que CouchDB; el uso de memoria RAM se mostró en Megabytes.



En la siguiente figura se mostró el resumen del uso de memoria RAM empleado en el volcado de datos 1 y 2 donde se usó la colección establecimientos y egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 10** - Gráfico resumen del uso de memoria RAM en la prueba uno.

## 2. Uso de CPU

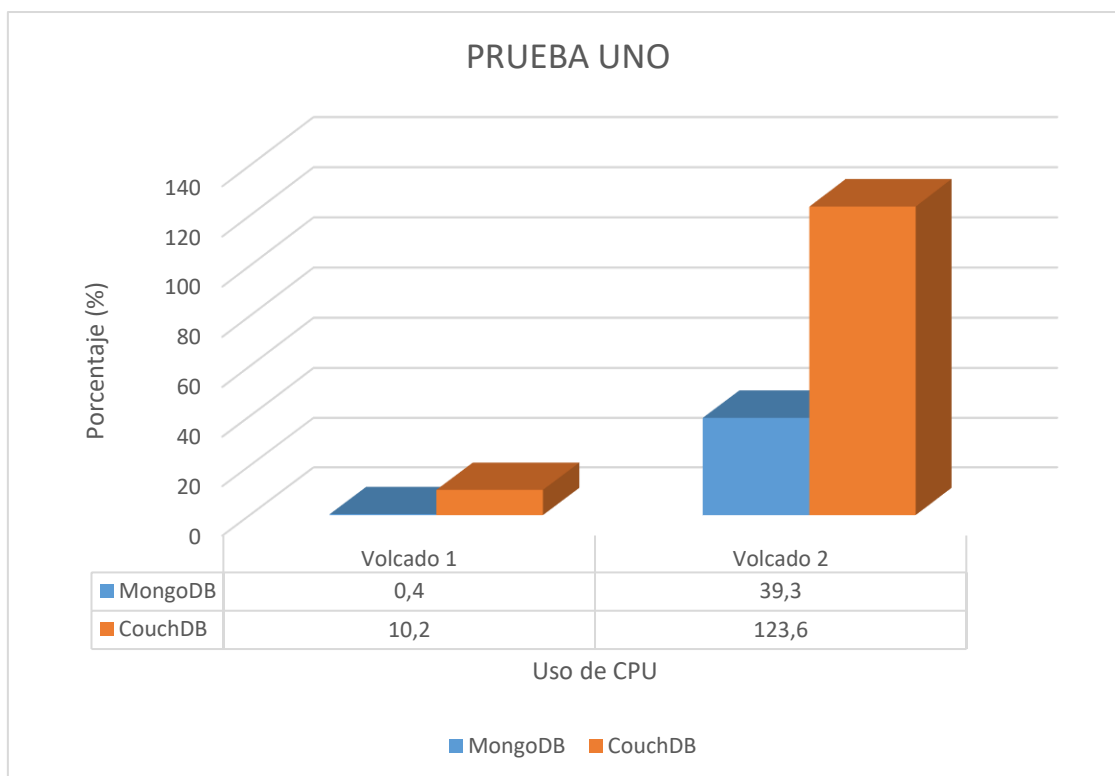
**Tabla 8** - Resumen del Uso de CPU - Prueba uno.

Gestores de Base de Datos NoSQL	Volcado 1	Volcado 2
MongoDB	0.4 %	39.3 %
CouchDB	10.2 %	123.6 %

**Fuente:** Elaboración Propia

Según esta tabla, se observó que MongoDB empleó menos uso de CPU para el volcado 1 y 2, mientras que CouchDB empleó más uso de CPU; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el resumen del uso de CPU empleado en el volcado de datos 1 y 2; donde se usó la colección establecimientos y egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 11** - Gráfico resumen del uso de CPU en la prueba uno.

### 3. Tiempo de Respuesta

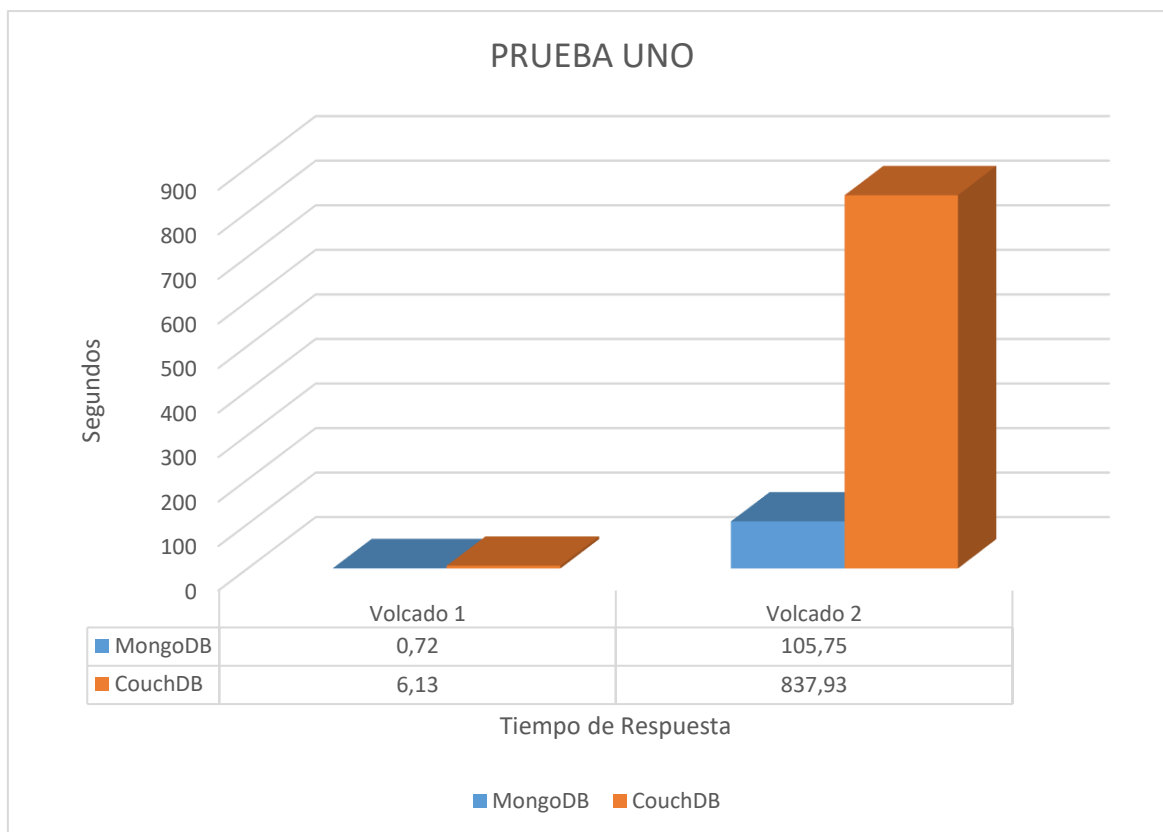
**Tabla 9** - Resumen del Tiempo de Respuesta - Prueba uno.

Gestores de Base de Datos NoSQL	Volcado 1	Volcado 2
MongoDB	0.72 Seg.	105.75 Seg.
CouchDB	6.13 Seg.	837.93 Seg.

**Fuente:** Elaboración Propia

Según esta tabla, se observó que MongoDB empleó menos tiempo de respuesta para el volcado 1 y 2, mientras que CouchDB empleó más tiempo de respuesta. El tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el resumen del tiempo de respuesta empleado en el volcado de datos 1 y 2; donde se usó la colección establecimientos y egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 12** - Gráfico resumen del uso de Tiempo de respuesta en la prueba uno.

#### 4. Espacio usado en disco

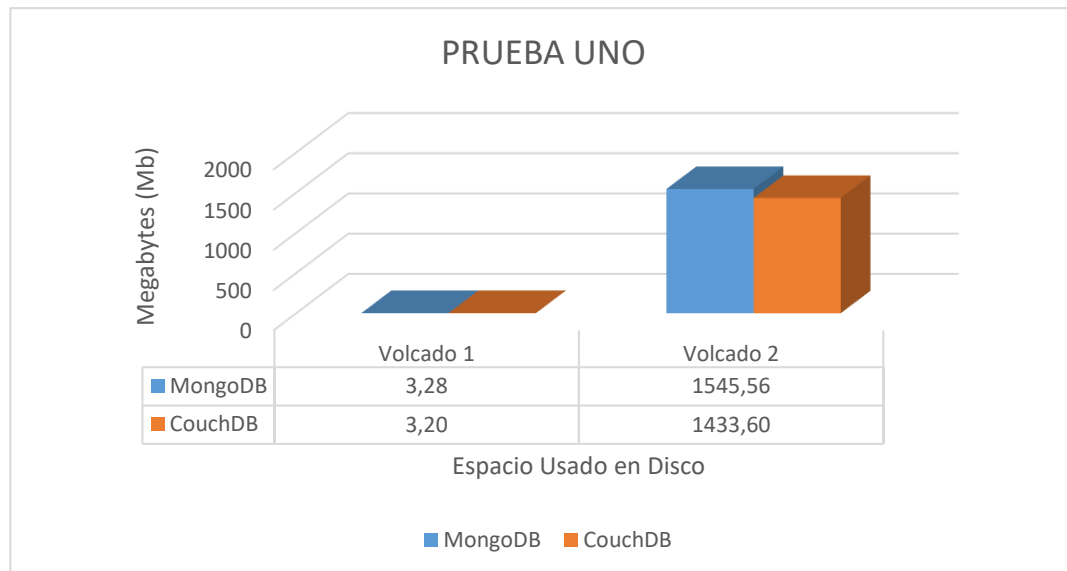
**Tabla 10** - Resumen del Espacio Usado en Disco - Prueba uno

Gestores de Base de Datos NoSQL	Volcado 1	Volcado 2
MongoDB	3.28 Mb	1545.56 Mb
CouchDB	3.20 Mb	1433.60 Mb

**Fuente:** Elaboración Propia

Según esta tabla, se observó que CouchDB empleó menos Espacio usado en disco para el volcado 1 y 2, mientras que MongoDB empleo más tiempo de respuesta. El Espacio usado en disco se mostró en megabytes.

En la siguiente figura se mostró el resumen del espacio usado en disco empleado en el volcado de datos 1 y 2; donde se usó la colección establecimientos y egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 13** - Gráfico resumen del uso de Espacio usado en Disco de la prueba uno.

## 5. Cantidad de errores

No se presentó errores en la prueba uno.

### B) Prueba dos

En la prueba dos, se realizó cuatro operaciones de listado de datos para ambos gestores de base de datos NoSQL documental con las colecciones establecimientos y egresoshospitalarios; donde se evaluaron los siguientes indicadores: Uso de Memoria RAM, uso de CPU, Tiempo de Respuesta y Cantidad de errores. Asimismo, se mostró los valores que arrojó la prueba dos.

#### 1. Uso de memoria RAM

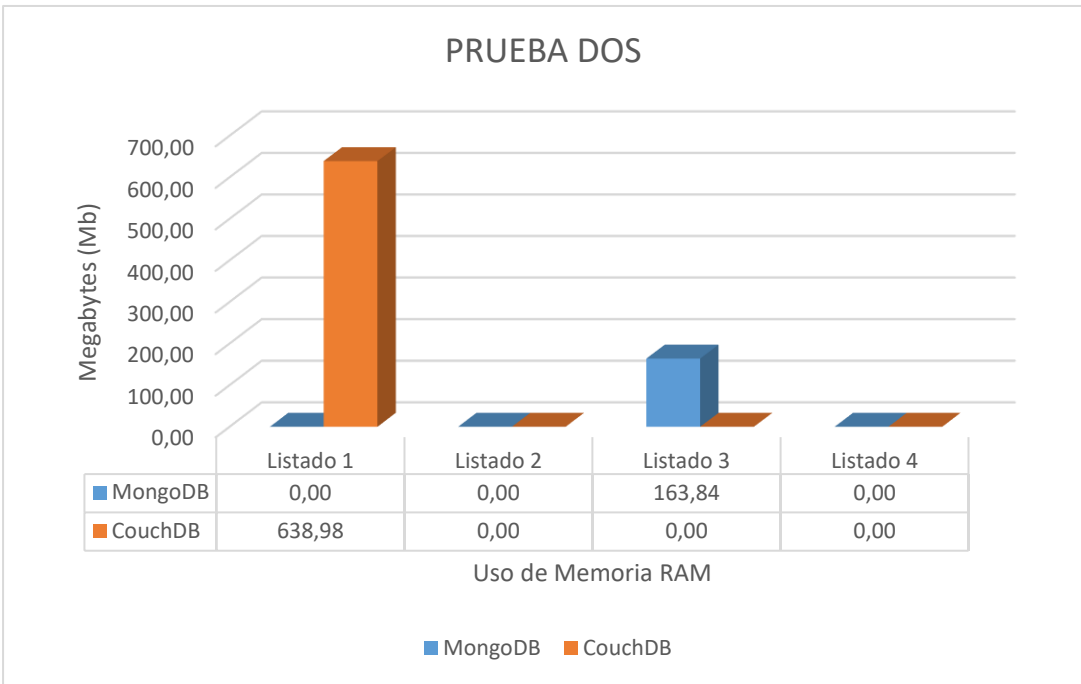
**Tabla 11** - Resumen del Uso de Memoria RAM - Prueba dos.

Gestores de				
Base de Datos	Listado 1	Listado 2	Listado 3	Listado 4
NoSQL				
MongoDB	0.00 Mb	0.00 Mb	163.84 Mb	0.00 Mb
CouchDB	638.98 Mb	0.00 Mb	0.00 Mb	0.00 Mb

**Fuente:** Elaboración Propia

Según esta tabla, se observó que MongoDB empleó menos uso de memoria RAM para el listado 1, mientras para el listado 2 y 4 mostró igualdad con CouchDB, asimismo CouchDB mostro menos consumo de memoria RAM para el listado 3; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el resumen del uso de memoria RAM empleado en el listado de datos 1, 2, 3 y 4; donde se usó la colección establecimientos y egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 14 - Gráfico resumen del uso de memoria RAM en la prueba dos.**

## 2. Uso de CPU

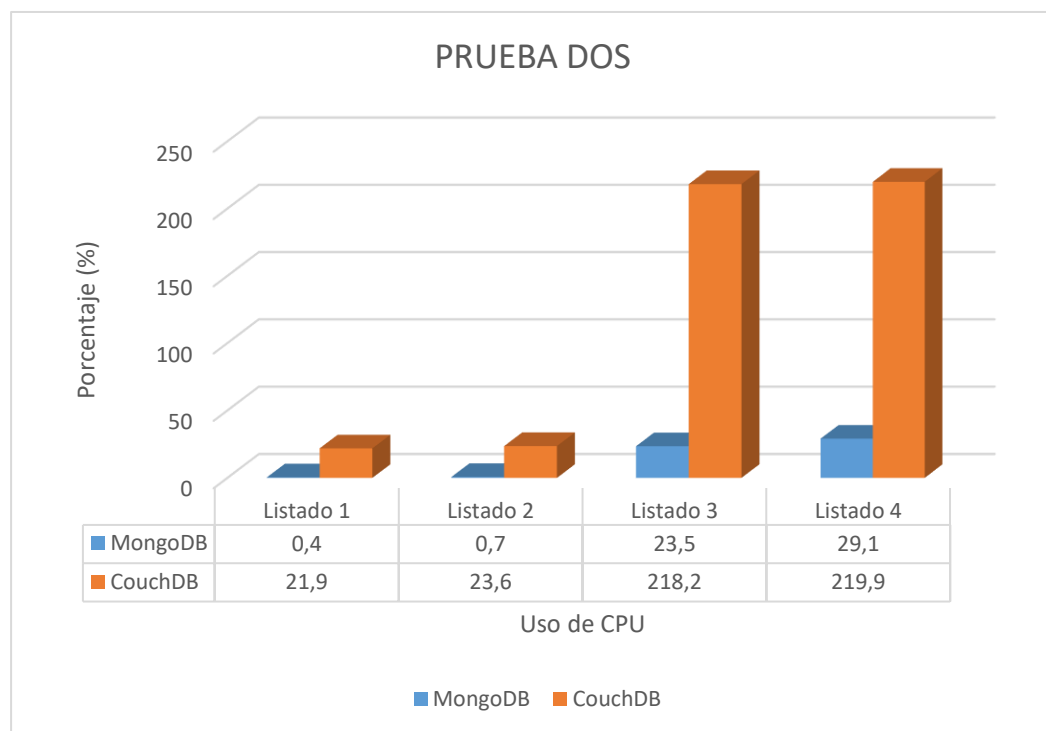
**Tabla 12 - Resumen del Uso de CPU - Prueba dos.**

<b>Gestores de Base de Datos NoSQL</b>	<b>Listado 1</b>	<b>Listado 2</b>	<b>Listado 3</b>	<b>Listado 4</b>
MongoDB	0.4 %	0.7 %	23.5 %	29.1 %
CouchDB	21.9 %	23.6 %	218.2 %	219.9 %

**Fuente:** Elaboración Propia

Según esta tabla, se observó que MongoDB empleó menos uso de CPU para el Listado 1, 2, 3 y 4; mientras que CouchDB empleó más uso de CPU. El uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el resumen del uso de CPU empleado en el Listado de datos 1, 2, 3 y 4; donde se usó la colección establecimientos y egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 15 - Gráfico resumen del uso de CPU en la prueba dos.**

### 3. Tiempo de Respuesta

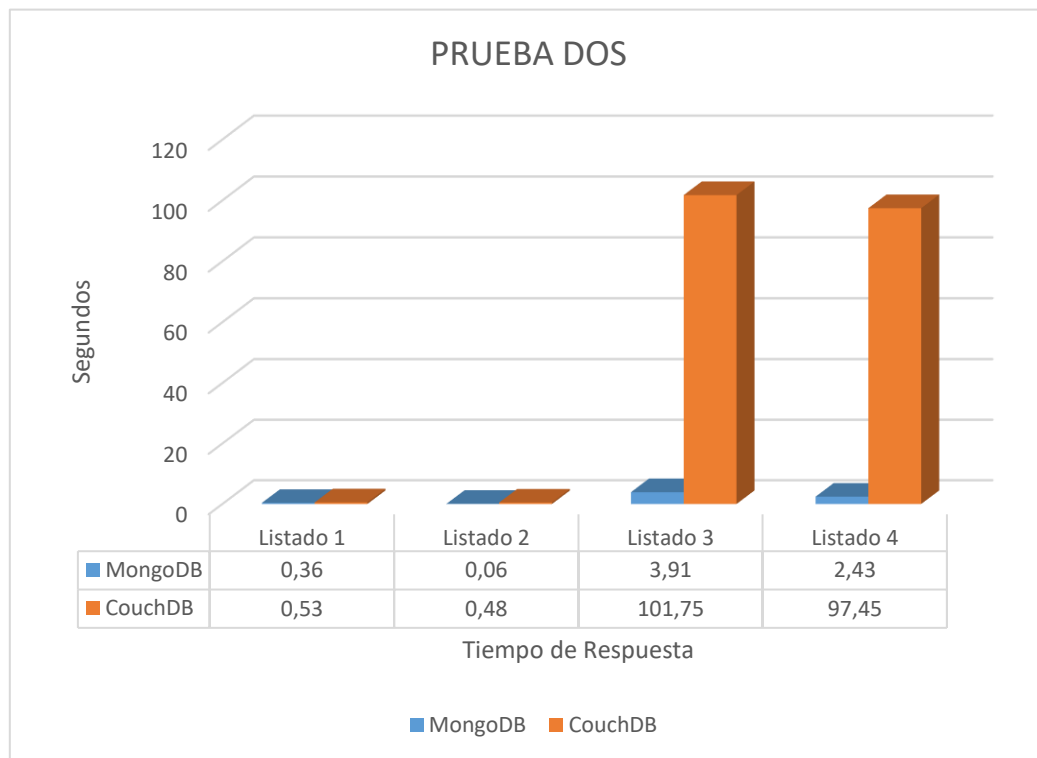
**Tabla 13 - Resumen del Tiempo de Respuesta - Prueba dos.**

Gestores de Base de Datos NoSQL	Listado 1	Listado 2	Listado 3	Listado 4
MongoDB	0.36 Seg.	0.06 Seg.	3.91 Seg.	2.43 Seg.
CouchDB	0.53 Seg.	0.48 Seg.	101.75 Seg.	97.45 Seg.

**Fuente:** Elaboración Propia

Según esta tabla, se observó que MongoDB empleó menos tiempo de respuesta para el listado 1, 2, 3 y 4; mientras que CouchDB empleó más tiempo de respuesta. El tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el resumen del tiempo de respuesta empleado en el listado 1, 2, 3 y 4; donde se usó la colección establecimientos y egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 16 - Gráfico resumen del uso de Tiempo de Respuesta en la prueba dos.**

#### **4. Cantidad de errores**

No se presentó errores en la prueba dos.

#### **C) Prueba tres**

En la Prueba tres, se realizó cuatro operaciones de actualizar datos para ambos administradores de base de datos NoSQL documental con las colecciones establecimientos y egresoshospitalarios; donde se evaluaron los siguientes indicadores: Uso de Memoria RAM, uso de CPU, Tiempo de Respuesta y Cantidad de errores. Asimismo, se mostró los valores que arrojó la prueba tres.

## 1. Uso de memoria RAM

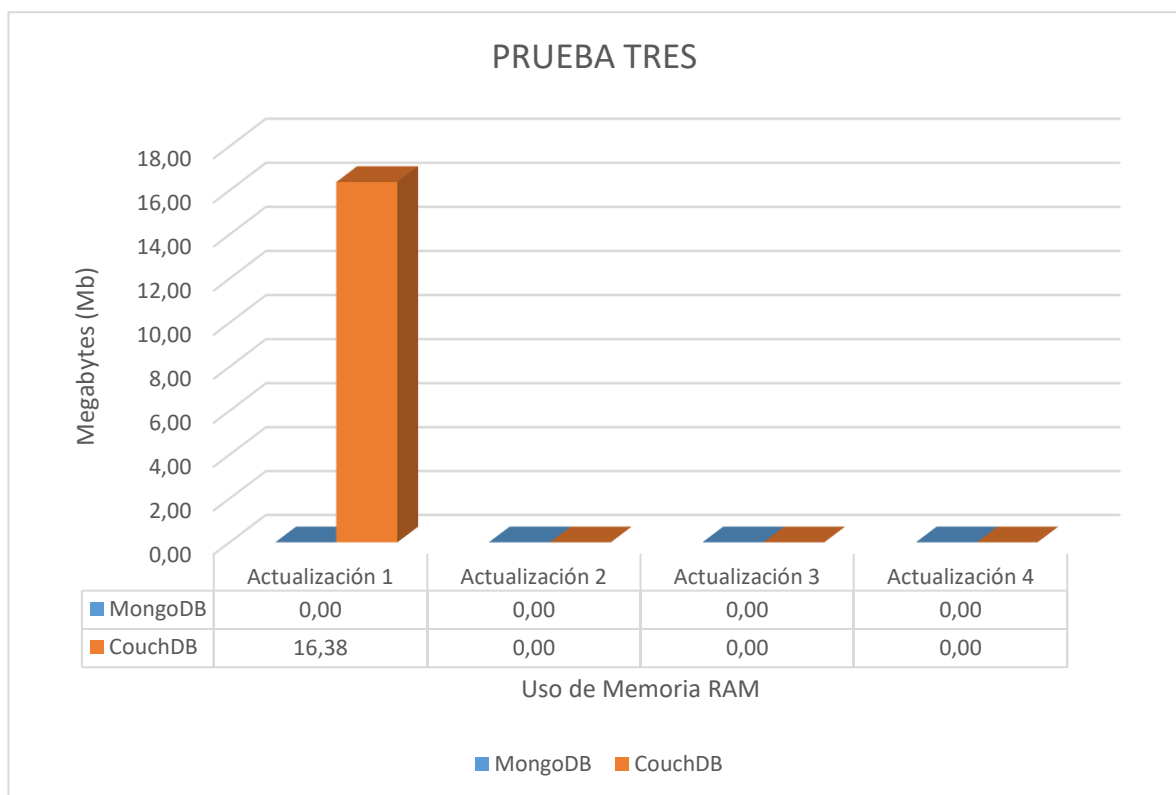
**Tabla 14 - Resumen del Uso de Memoria RAM - Prueba tres.**

Gestores de Base de Datos NoSQL	Actualización 1	Actualización 2	Actualización 3	Actualización 4
MongoDB	0.00 Mb	0.00 Mb	0.00 Mb	0.00 Mb
CouchDB	16.38 Mb	0.00 Mb	0.00 Mb	0.00 Mb

**Fuente:** Elaboración Propia

Según esta tabla, se observó que MongoDB empleó menos uso de memoria RAM para la actualización 1, mientras que para la actualización 2, 3 y 4 mostró igualdad que CouchDB; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el resumen del uso de memoria RAM empleado en la actualización de datos 1, 2, 3 y 4; donde se usó la colección establecimientos y egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 17 - Gráfico resumen del uso de memoria RAM en la prueba tres**



## 2. Uso de CPU

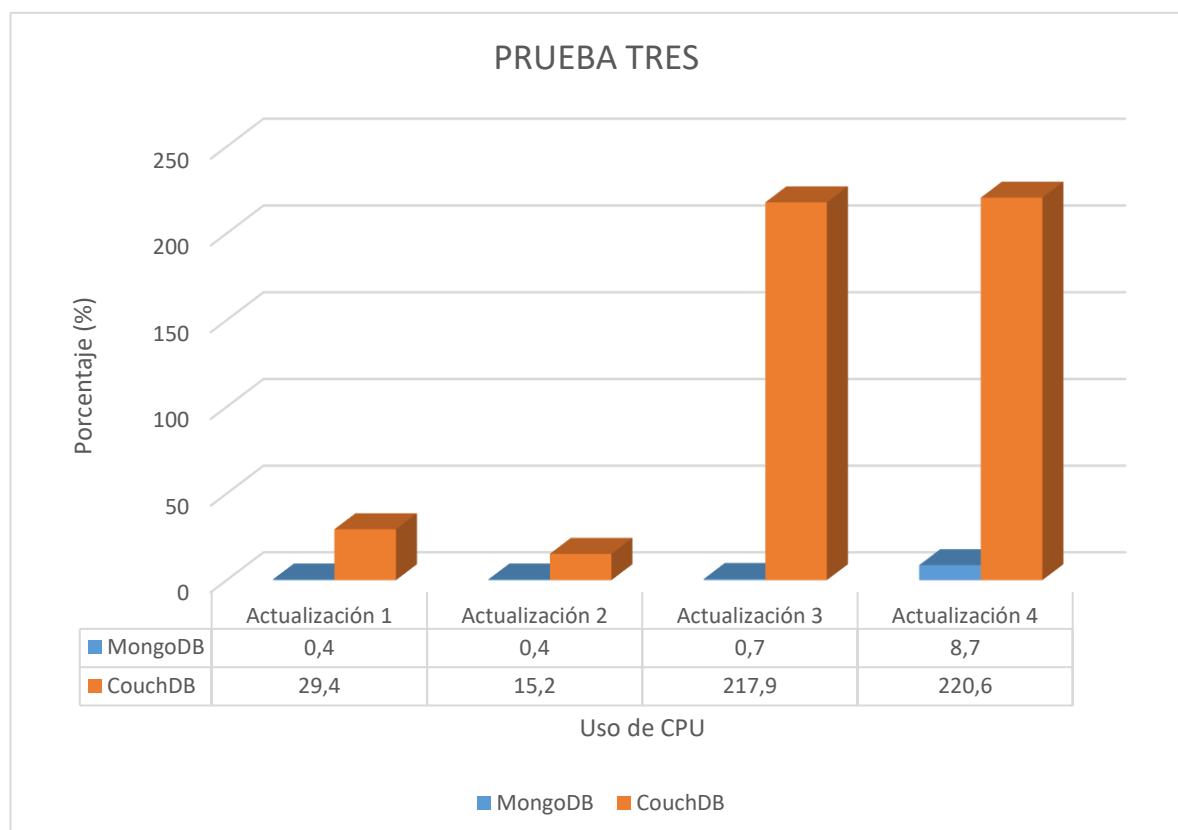
**Tabla 15 - Resumen del Uso de CPU - Prueba tres.**

Gestores de Base de Datos NoSQL	Actualización 1	Actualización 2	Actualización 3	Actualización 4
MongoDB	0.4 %	0.4 %	0.7 %	8.7 %
CouchDB	29.4 %	15.2 %	217.9 %	220.6 %

**Fuente:** Elaboración Propia

Según esta tabla, se observó que MongoDB empleó menos uso de CPU para la actualización 1, 2, 3 y 4; mientras que CouchDB empleó más uso de CPU. El uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el resumen del uso de CPU empleado en la actualización de datos 1, 2, 3 y 4; donde se usó la colección establecimientos y egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 18 - Gráfico resumen del uso de CPU en la prueba tres.**

### 3. Tiempo de Respuesta

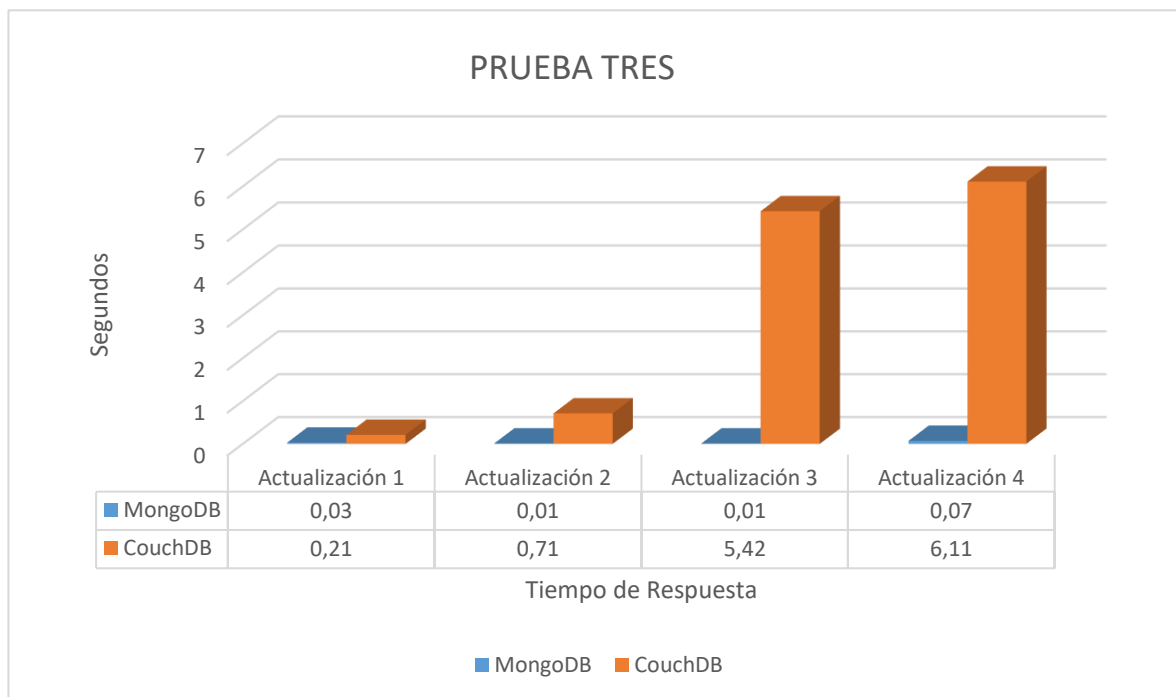
**Tabla 16 - Resumen del Tiempo de Respuesta - Prueba tres.**

Gestores de Base de Datos NoSQL	Actualización 1	Actualización 2	Actualización 3	Actualización 4
MongoDB	0.03 Seg.	0.01 Seg.	0.01 Seg.	0.07 Seg.
CouchDB	0.21 Seg.	0.71 Seg.	5.42 Seg.	6.11 Seg.

**Fuente:** Elaboración Propia

Según esta tabla, se observó que MongoDB empleó menos tiempo de respuesta para la actualización 1, 2, 3 y 4; mientras que CouchDB empleó más tiempo de respuesta. El tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el resumen del tiempo de respuesta empleado en la actualización 1, 2, 3 y 4; donde se usó la colección establecimientos y egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 19 - Gráfico resumen del uso de Tiempo de respuesta en la prueba tres.**

### 4. Cantidad de errores

No se presentó errores en la prueba tres.

## D) Prueba cuatro

En la Prueba cuatro, se realizó cuatro operaciones de eliminar datos para ambos administradores de base de datos NoSQL documental con las colecciones establecimientos y egresoshospitalarios; donde se evaluaron los siguientes indicadores: Uso de Memoria RAM, uso de CPU, Tiempo de Respuesta y Cantidad de errores. Asimismo, se mostró los valores que arrojó la prueba cuatro.

### 1. Uso de memoria RAM

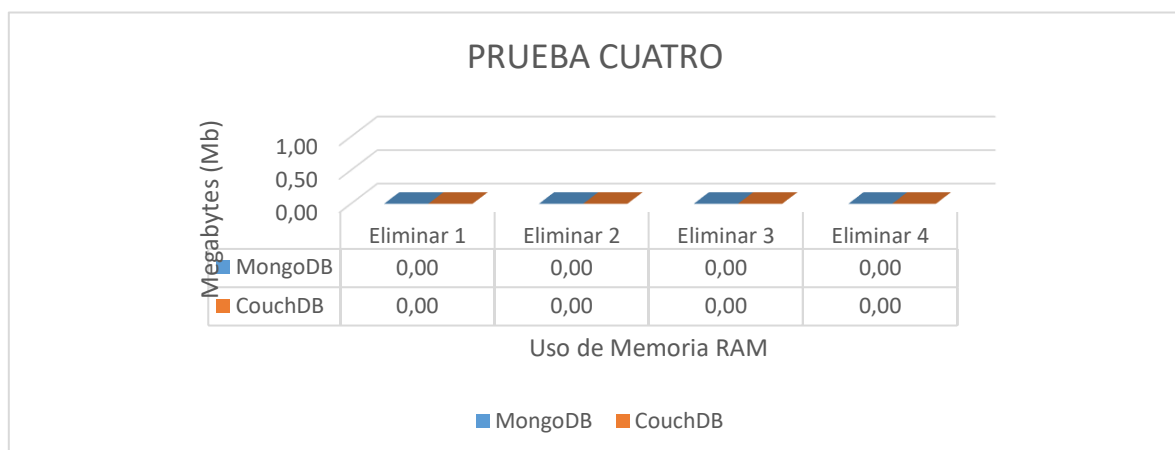
*Tabla 17 - Resumen del Uso de Memoria RAM - Prueba cuatro.*

Gestores de				
Base de Datos	Eliminar 1	Eliminar 2	Eliminar 3	Eliminar 4
NoSQL				
MongoDB	0.00 Mb	0.00 Mb	0.00 Mb	0.00 Mb
CouchDB	0.00 Mb	0.00 Mb	0.00 Mb	0.00 Mb

*Fuente: Elaboración Propia*

Según esta tabla, se observó que MongoDB empleó igualdad de uso de memoria RAM para el eliminar 1, 2, 3 y 4 que CouchDB; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el resumen del uso de memoria RAM empleado en el eliminar datos 1, 2, 3 y 4; donde se usó la colección establecimientos y egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



*Figura 20 - Gráfico resumen del uso de memoria RAM en la prueba cuatro.*

## 2. Uso de CPU

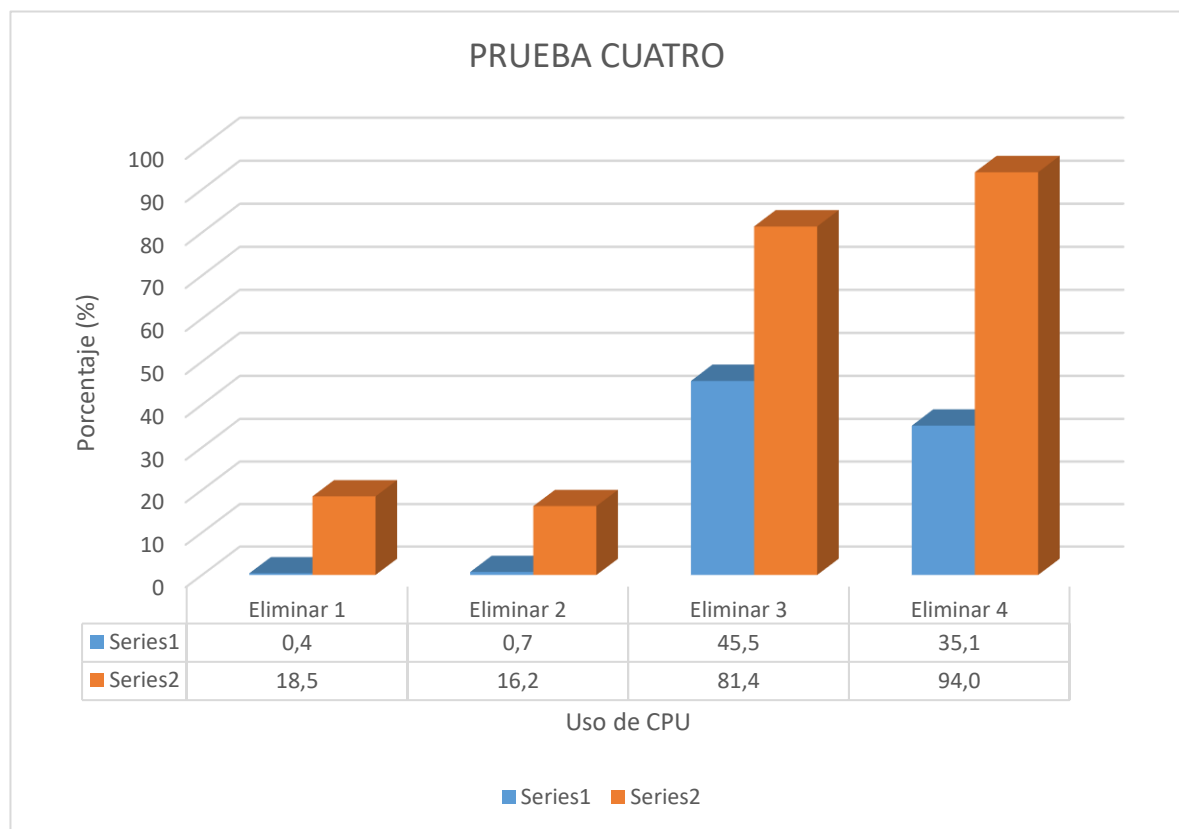
**Tabla 18 - Resumen del Uso de CPU - Prueba cuatro.**

<b>Gestores de Base de Datos NoSQL</b>	<b>Eliminar 1</b>	<b>Eliminar 2</b>	<b>Eliminar 3</b>	<b>Eliminar 4</b>
MongoDB	0.4 %	0.7 %	45.5 %	35.1 %
CouchDB	18.5 %	16.2 %	81.4 %	94.0 %

**Fuente:** Elaboración Propia

Según esta tabla, se observó que MongoDB empleó menos uso de CPU para el eliminar 1, 2, 3 y 4; mientras que CouchDB empleo más uso de CPU. El uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el resumen del uso de CPU empleado en el eliminar datos 1, 2, 3 y 4; donde se usó la colección establecimientos y egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 21 - Gráfico resumen del uso de CPU en la prueba cuatro.**

### 3. Tiempo de Respuesta

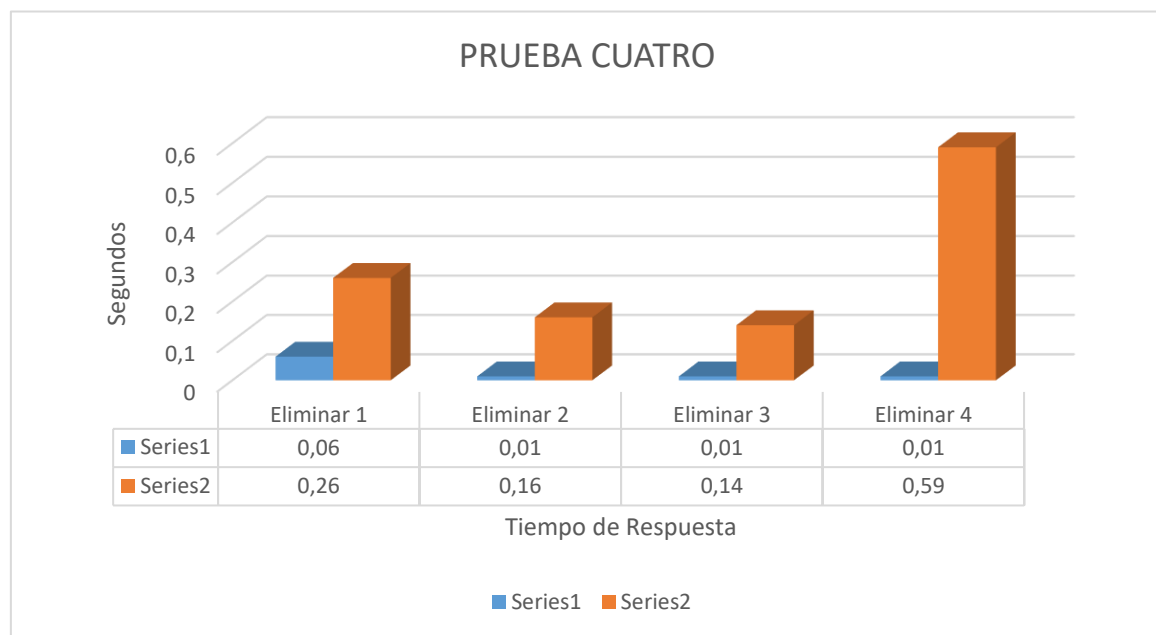
**Tabla 19 - Resumen del Tiempo de Respuesta - Prueba cuatro.**

Gestores de Base de Datos NoSQL	Eliminar 1	Eliminar 2	Eliminar 3	Eliminar 4
MongoDB	0.06 Seg.	0.01 Seg.	0.01 Seg.	0.01 Seg.
CouchDB	0.26 Seg.	0.16 Seg.	0.14 Seg.	0.59 Seg.

**Fuente:** Elaboración Propia

Según esta tabla, se observó que MongoDB empleó menos tiempo de respuesta para Eliminar 1, 2, 3 y 4; mientras que CouchDB empleo más tiempo de respuesta. El tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el resumen del tiempo de respuesta empleado en eliminar 1, 2, 3 y 4; donde se usó la colección establecimientos y egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 22 - Gráfico resumen del uso de tiempo de respuesta en la prueba cuatro.**

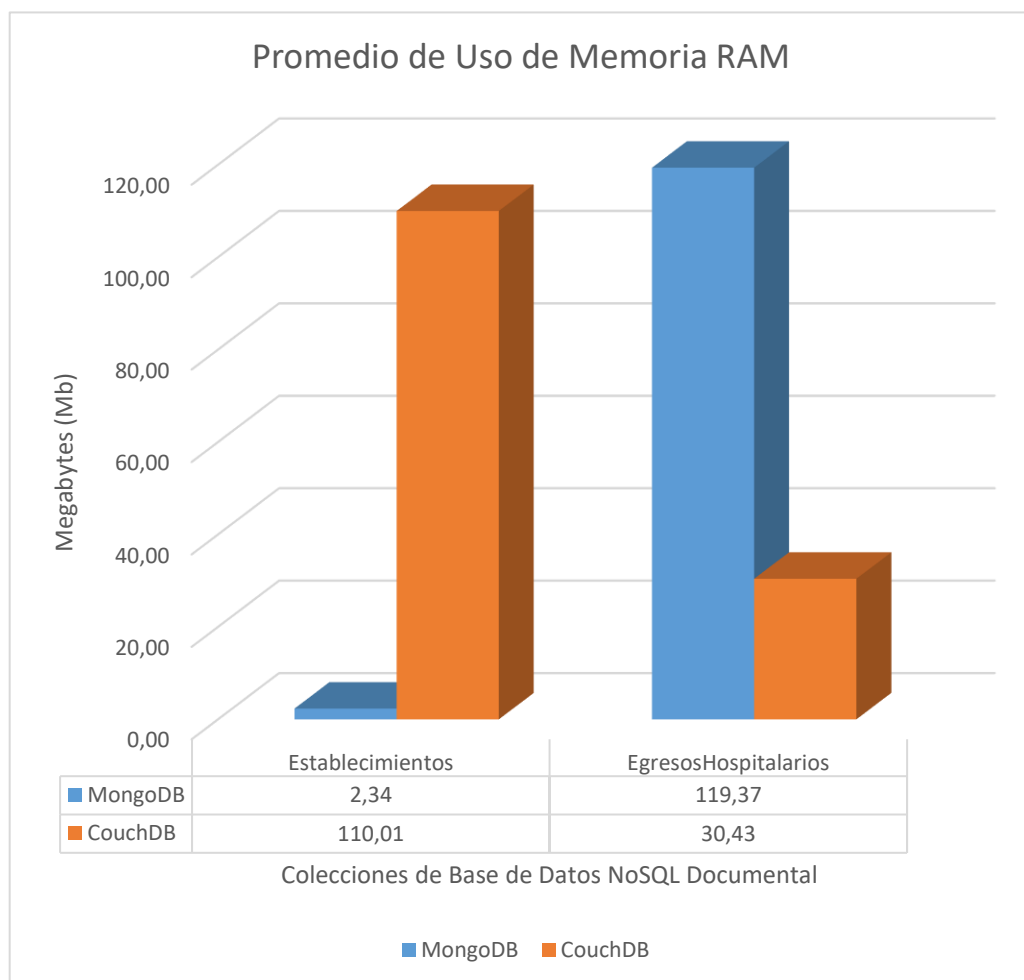
### 4. Cantidad de errores

No se presentó errores en la prueba cuatro

### 3.1.2. Promedio de la evaluación cuantitativa de los gestores de base de datos NoSQL.

#### A) Uso de Memoria RAM

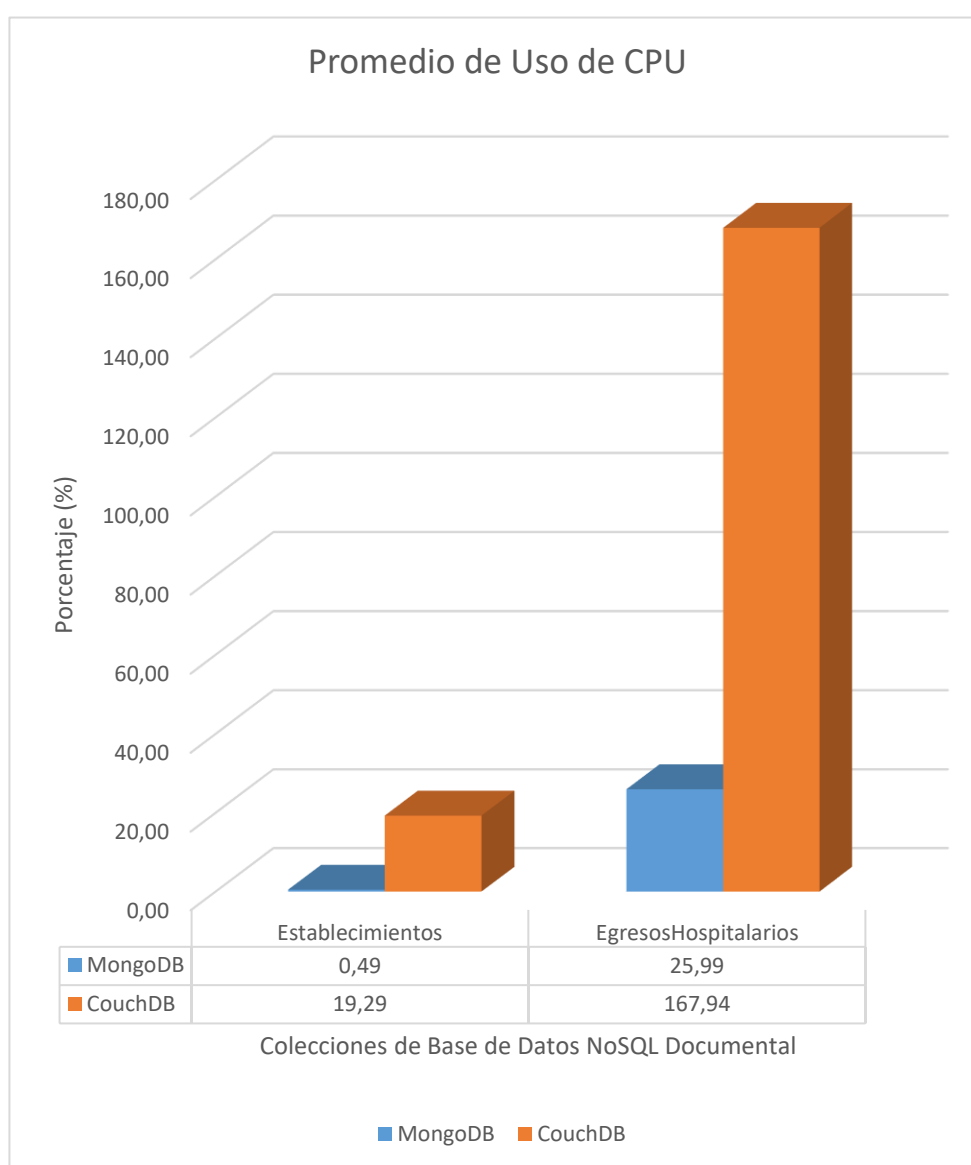
En el siguiente gráfico se mostró el promedio del uso de Memoria RAM de todas operaciones realizadas en las pruebas de rendimiento; donde se observó que MongoDB usó menos memoria RAM cuando se usó la colección establecimientos (4056 documentos), pero cuando se usó la colección egresoshospitalarios (1'566,188 documentos) MongoDB mostró más consumo de Memoria RAM que CouchDB. Asimismo, se determinó que CouchDB usó menos Memoria RAM cuando se ejecutó el procesamiento de grandes conjuntos de datos.



**Figura 23 - Promedio de Uso de Memoria RAM**

## B) Uso de CPU

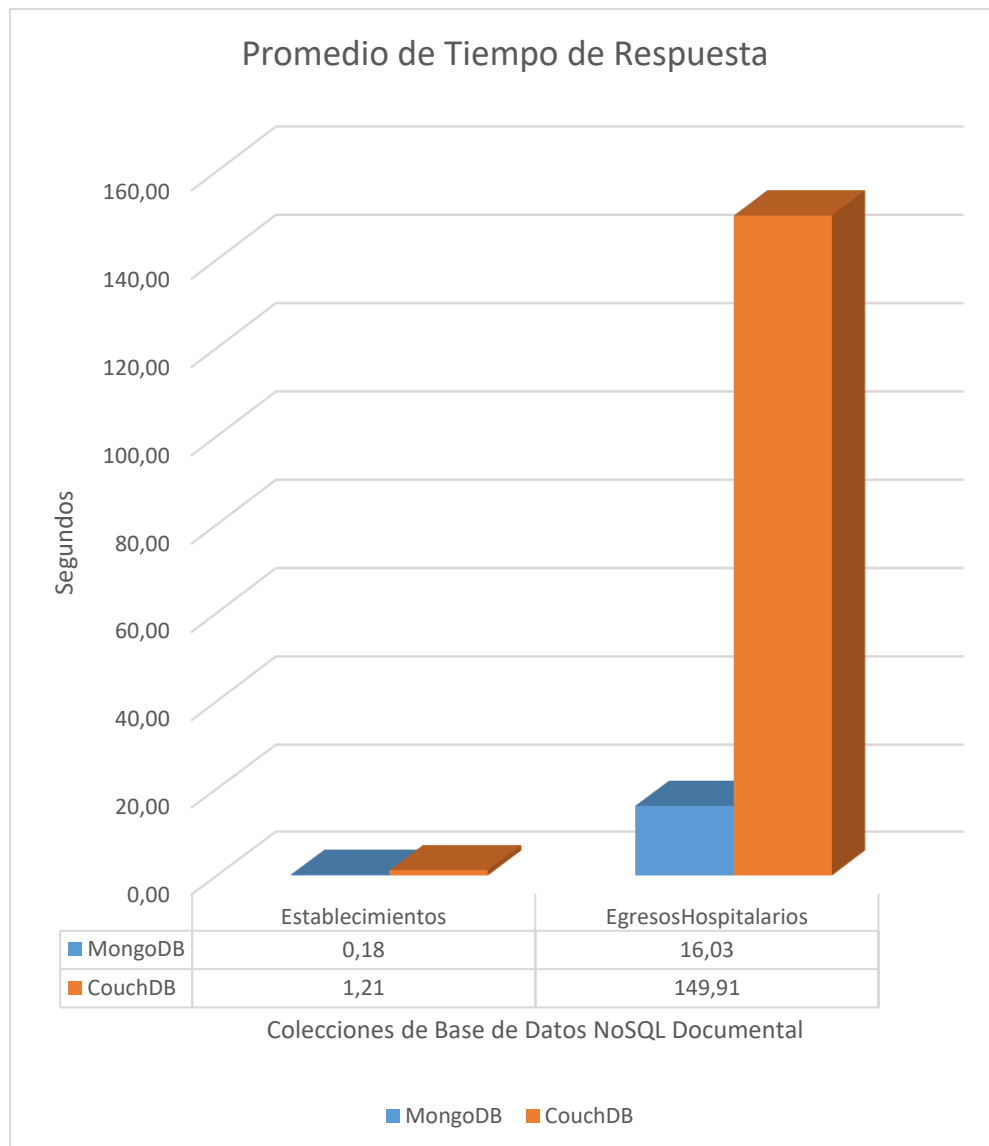
En el siguiente gráfico se mostró el promedio del uso de CPU de todas operaciones realizadas en las pruebas de rendimiento; donde se observó que MongoDB usó menos CPU cuando se usó la colección establecimientos (4056 documentos), asimismo cuando se usó la colección egresoshospitalarios (1'566,188 documentos) MongoDB mostró de igual forma menos consumo de CPU que CouchDB. Asimismo, se determinó que MongoDB usó menos CPU cuando se ejecutó el procesamiento de grandes conjuntos de datos.



**Figura 24 - Promedio de Uso de CPU.**

### C) Tiempo de Respuesta

En el siguiente gráfico se mostró el promedio del tiempo de respuesta de todas operaciones realizadas en las pruebas de rendimiento; donde se observó que MongoDB usó menos tiempo de respuesta cuando se usó la colección establecimientos (4056 documentos), asimismo cuando se usó la colección egresoshospitalarios (1'566,188 documentos) MongoDB mostró de igual forma menos consumo de tiempo de respuesta que CouchDB. Asimismo, se determinó que MongoDB usó menos tiempo de respuesta cuando se ejecutó el procesamiento de grandes conjuntos de datos.

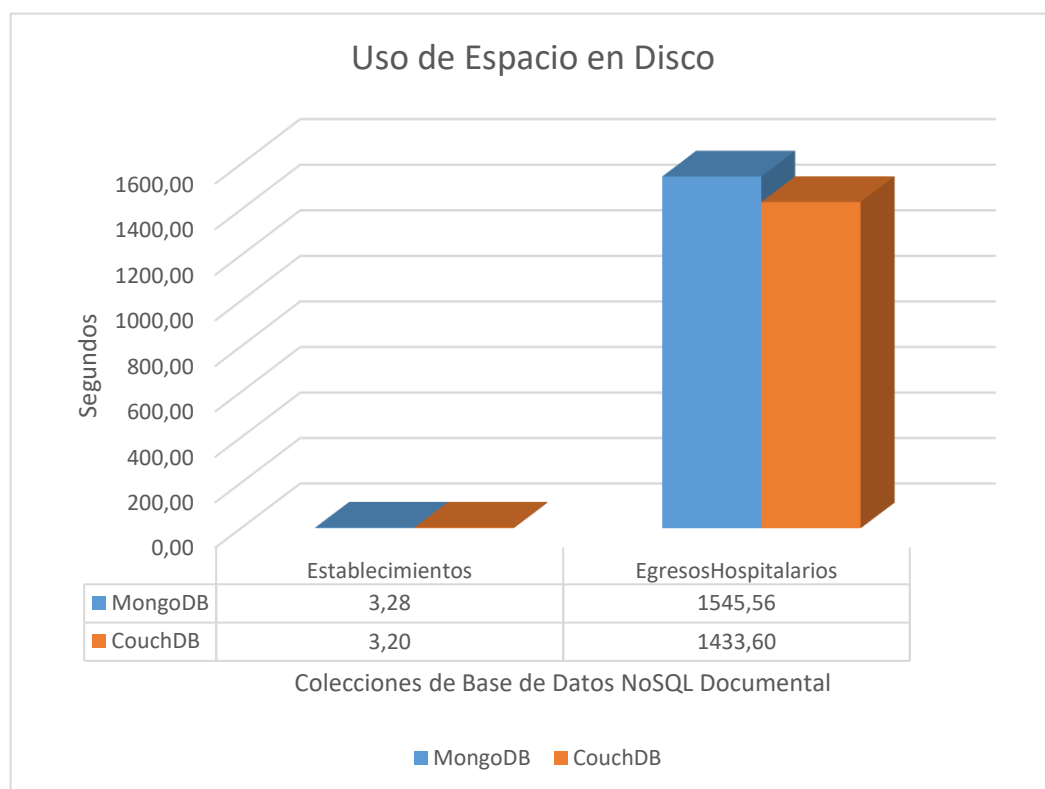


**Figura 25 - Promedio de Tiempo de Respuesta.**



## D) Espacio Usado en Disco

En el siguiente gráfico se mostró el espacio usado de Disco cuando se ejecutaron las operaciones de volcado de datos; donde se observó que MongoDB usó más espacio en disco cuando se usó la colección establecimientos (4056 documentos), asimismo cuando se usó la colección egresoshospitalarios (1'566,188 documentos) MongoDB mostró de igual forma más consumo de espacio en disco que CouchDB. Asimismo, se determinó que CouchDB usó menos espacio en disco cuando se ejecutó las operaciones de volcado de datos siendo ejecutados con la misma cantidad de documentos.



**Figura 26 - Espacio Usado en Disco en Operaciones de Volcado.**

## E) Cantidad de Errores

Todas las operaciones realizadas en el procesamiento de grandes conjuntos de datos no presentaron ningún error por parte de ninguno de los administradores de base de datos NoSQL Documental.

### 3.2. Discusión de resultados.

*Tabla 20 - Resultados Obtenidos de la Colección Establecimientos.*

Indicadores	MongoDB	CouchDB
Uso de Memoria RAM	2.34 MB	110.01 MB
Uso de CPU	0.49 %	19.29 %
Tiempo de Respuesta	0.18 Seg.	1.21 Seg.
Espacio usado en Disco	3.28 MB	3.20 MB
Cantidad de errores	0	0

**Fuente:** Elaboración Propia.

La siguiente tabla muestra el promedio de la estimación del rendimiento de los indicadores, de las distintas pruebas realizadas a los gestores de base de datos NoSQL Documental, en donde MongoDB mostró menos promedio en el consumo de Memoria RAM, consumo de CPU y Tiempo de Respuesta; mientras que CouchDB mostró menos espacio usado en Disco. Ningún gestor de base de datos NoSQL Documental mostro errores, se usó la colección establecimientos de 4056 documentos, de la cual se obtuvieron distintos volúmenes de datos.

*Tabla 21 - Resultados Obtenidos de la Colección Egresoshospitalarios.*

Indicadores	MongoDB	CouchDB
Uso de Memoria RAM	119.37 MB	30.43 MB
Uso de CPU	25.99 %	167.94 %
Tiempo de Respuesta	16.03 Seg.	149.91 Seg.
Espacio usado en Disco	1545.56 MB	1433.60 MB
Cantidad de errores	0	0

**Fuente:** Elaboración Propia.

La siguiente tabla muestra el promedio de la estimación del rendimiento de los indicadores, de las distintas pruebas realizadas a los gestores de base de datos NoSQL Documental, en donde MongoDB mostró menos promedio en el consumo CPU y Tiempo de Respuesta; mientras que CouchDB mostró menos consumo en memoria RAM y espacio usado en Disco. Ningún gestor de base de datos NoSQL

Documental mostro errores, se usó la colección egresoshospitalarios de 1'566,188 documentos, de la cual se obtuvieron distintos volúmenes de datos.

En esta investigación se comparó el rendimiento de dos gestores de base de datos NoSQL: MongoDB y Apache CouchDB. El ambiente de pruebas se implementó en LINUX (Ubuntu). La herramienta utilizada para la implementación fue el Servidor web apache y la codificación se realizó en PHP. Se utilizó librerías de MongoDB y Apache CouchDB para hacer la conexión; asimismo se construyó una plataforma web con un cronometro para medir el tiempo de respuesta y a la vez para poder realizar todas las operaciones necesarias para la comparación. Se han realizado las operaciones de Volcado, Listado, Actualización y Eliminación; y los resultados se han comparado en base al tiempo de Respuesta, Uso de CPU, Uso de Memoria RAM, Espacio usado en disco y cantidad de errores por ambas bases de datos para realizar estas operaciones. Se trabajó con dos conjuntos de datos de muestra, que consisten en 4056 documentos y 1'566,188 documentos. Asimismo, se concluyó que CouchDB proporciona mejor desempeño en Consumo de Memoria RAM cuando se realizan pruebas de grandes conjuntos de datos y utilizó menos consumo de Espacio en Disco cuando se realiza la operación de volcado de datos ya sea de grandes o pequeños conjuntos de datos. MongoDB por otro lado proporcionó mejor rendimiento en el Consumo de CPU y Tiempo de Respuesta cuando se realizan las operaciones de grandes conjuntos de datos; asimismo cuando se trata de pequeños conjuntos de datos MongoDB muestra resultados óptimos en cuanto al uso de Memoria RAM, Uso de CPU y Tiempo de Respuesta; resultando utilizar menos consumo de estos indicadores que el gestor de base de datos NoSQL CouchDB.

Teniendo en cuenta la investigación de (Gupta & Rani, 2016), donde Compararon el rendimiento de dos bases de datos: Elasticsearch y Apache CouchDB. Su implementación se ha hecho en LINUX. La herramienta utilizada para la implementación es Eclipse IDE. La codificación se ha hecho con JAVA. Se han utilizado las librerías javas Elasticsearch, CouchDB y Maven para hacer la conexión y realizar todas las operaciones necesarias para la comparación. La API del Rest Client se utiliza para realizar el servicio de Elasticsearch y CouchDB. Se han realizado las operaciones de Inserción, Selección, Actualización y Eliminación y los

resultados se han comparado en base al tiempo empleado por ambas bases de datos para realizar estas operaciones. Trabajaron con un conjunto de datos de muestra que consiste en 20.000 documentos. Asimismo, concluyeron a partir de los resultados observados que Elasticsearch tarda mucho más tiempo que CouchDB en términos de inserción, eliminación y actualización de datos, pero para la selección de datos, Elasticsearch realiza la búsqueda de forma más eficiente que CouchDB.

### 3.3. Aporte practico

Para la elección del administrador de base de datos NoSQL se ha probado en bases de datos dedicadas IEEE Xplore y ScienceDirect. Las encuestas seleccionadas incluyen resultados obtenidos a través de un proceso de ingeniería experimental durante los últimos 5 años (2015-2020), a nivel comparativo de dos o más gestores de bases de datos NoSQL, o aplicados a casos de estudio.

Las bases de datos profesionales se eligen por su reputación y contribuciones al campo técnico. De igual manera, se tuvo en cuenta el número de publicaciones relacionadas con el tema de investigación bajo los términos “NoSQL” y “NoSQL Database”. Por otro lado, IEEE Xplore cuenta con más de 25.000 revisiones en su base de datos, el principal insumo son investigaciones del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE); Hay 1120 publicaciones relacionadas en esta base de datos. ScienceDirect, por su parte, indexa más de 2.500 reseñas en su base de datos en modo peer-review y cuenta con un amplio apartado técnico. Entre ellos, se obtuvieron alrededor de 300 resultados relacionados con el tema de investigación.



*Figura 27 - Base de Datos Especializada IEEE Xplore.*

**Fuente:** <https://ieeexplore.ieee.org/Xplore/home.jsp>



ScienceDirect

**Figura 28** - Base de Datos Especializada SciencieDirect.

**Fuente:** <https://www.sciencedirect.com/>

Después se eligió el tipo de Base de Datos NoSQL con el que se trabajó en esta investigación, siendo elegido los Gestores de Base de Datos NoSQL del tipo Documental; porque es uno de los principales tipos de bases de datos NoSQL y están diseñadas para almacenar, recuperar y gestionar datos orientados a documentos o semiestructurados. Su ventaja oculta de los bases de datos NoSQL Documentales es que son una colección de colecciones de clave-valor. Es decir, dentro de un contenedor similar a las bases de datos NoSQL de Clave-Valor, hay un nivel adicional de indexación de clave-valor que permite consultas mucho más eficientes.

Asimismo, se presentó un ranking de los 08 primeros gestores de base de Datos NoSQL documental, elaborado por la página DB-Engines. Se trabajó con los gestores de base de Datos NoSQL documentales más populares. Asimismo, este ranking es respaldado por la mayoría de gestores de bases de datos comerciales y no comerciales, además cuenta con un blog donde promueve la colaboración activa de miembros, una enciclopedia donde se clasifican a los gestores de acuerdo a su forma de trabajo. Esta web se encarga de promocionar eventos, noticias y publicar artículos nuevos para la comunidad, es un producto de SolidIT GmbH empresa austriaca constituida en el 2011 dedicada a consultoría IT y desarrollo empresarial con certificaciones ISO/ IEC 25012, ISO/IEC 25024 y CompTIA Network.

Rango			DBMS	Modelo de base de datos	Puntuación		
Oct. De 2020	Sep. De 2020	Oct. De 2019			Oct. De 2020	Sep. De 2020	Oct. De 2019
1.	1.	1.	MongoDB +	Documento	448.02	+1,54	+35,93
2.	2.	2.	Amazon DynamoDB +	Multi-modelo	68,41	+2,23	+8,24
3.	3.	↑ 4.	Microsoft Azure Cosmos DB +	Multi-modelo	32.01	+0,34	+0,68
4.	4.	↓ 3.	Couchbase +	Documento , multimodelo	30,33	-0,28	-1,88
5.	5.	5.	CouchDB	Documento	17,41	+0,16	-0,63
6.	6.	↑ 7.	Firebase Realtime Database	Documento	16.26	+0,65	+4.52
7.	7.	↓ 6.	MarkLogic +	Multi-modelo	11,73	-0,21	-1,33
8.	8.	8.	Reino +	Documento	8,74	0,00	+0,82

**Figura 29** - Ranking de Base de Datos NoSQL Documental.

Después, se elaboró una tabla con las características de los gestores de base de datos NoSQL documental.

**Tabla 22 - Características de los Gestores de Base de Datos NoSQL Documental.**

N°	BASE DE DATOS NOSQL	TIPO DE DATOS	MODELO DE DATOS	PLATAFORMA	LICENCIA
1	Mongo DB	BSON /JSON	Documento	Linux/ OS X/Solaris/Windows	Libre
2	Amazon DynamoDB	JSON	Multimodelo	Alojado (Nube)	Propietario
3	Microsoft Azure Cosmos DB	JSON	Multimodelo	Alojado (Nube)	Propietario
4	Couchbase	JSON	Multimodelo	Linux/ OS X/Windows	Propietario/ Libre
5	Apache CouchDB	JSON	Documento	Android/ BSD/ Linux/OS X/ Solaris/ Windows	Libre
6	Firebase Realtime Database	JSON	Documento	Alojado (Nube)	Propietario
7	MarkLogic	JSON	Multimodelo	Linux/ OS X/Windows	Propietario
8	Reino	JSON	Documento	Windows	Libre

**Fuente:** Elaboración Propia

A partir del ranking presentado y la tabla de características de los gestores de base de datos NoSQL Documental, se procedió a elegir los gestores de base de datos NoSQL documental teniendo en cuenta sus características: Tipo de datos, Modelo de datos, plataforma y tipo de licencia.

**A) Tipo de datos:** Los gestores de base de datos NoSQL documental que se eligió son del tipo de datos JSON.

- B) Modelo de datos:** Los gestores de base de datos NoSQL documental que se eligió son solo del modelo documental y no del modelo de datos Multimodelo.
- C) Plataforma:** Los gestores de base de datos NoSQL documental que se eligió trabajan con el sistema operativo Linux.
- D) Tipo de licencia:** Los gestores de base de datos NoSQL documental que se eligió son de licencia libre.

En la siguiente tabla se evaluaron los gestores de bases de datos NoSQL documental de acuerdo a características, dando resultado a un puntaje de evaluación.

**Tabla 23 - Evaluación de características con Puntaje.**

N°	BASE DE DATOS NOSQL	TIPO DE DATOS	MODELO DE DATOS	PLATAFORMA	LICENCIA	PUNTAJE
1	Mongo DB	x	x	x	x	4
2	Amazon DynamoDB	x	-	-	-	1
3	Microsoft Azure Cosmos DB	x	-	-	-	1
4	Couchbase	x	-	x	x	3
5	Apache CouchDB	x	x	x	x	4
6	Firebase Realtime Database	x	x	-	-	2
7	MarkLogic	x	-	x	-	2
8	Reino	x	x	-	x	3

**Fuente:** *Elaboración Propia*

De acuerdo a la tabla de evaluación de características; los gestores de base de datos NoSQL documental seleccionados fueron MongoDB y Apache CouchDB, estos han sido seleccionados porque ambos obtuvieron un puntaje de número

cuatro en la evaluación de características; con estos dos gestores de base de datos se trabajó la siguiente investigación. Después de seleccionar los dos gestores de base de datos NoSQL documental se procedió a construir un ambiente para pruebas, el cual se realizó de la siguiente manera.

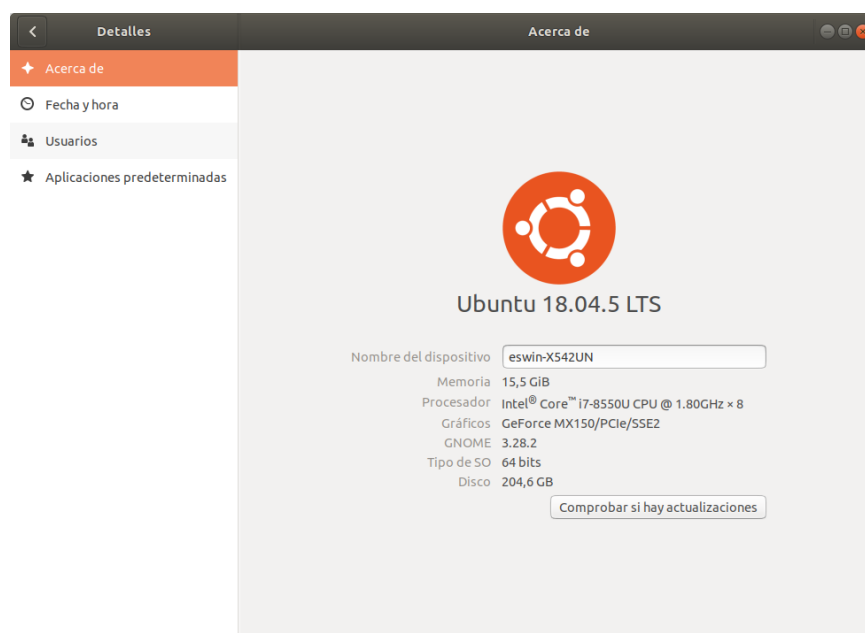
En primer lugar, se realizó la selección de características y descripción del hardware para construir el ambiente de pruebas:

**Tabla 24** – Características y descripción del Hardware del entorno de Trabajo

CARACTERÍSTICAS	DESCRIPCIÓN
Marca	ASUS
Procesador	Core i7-8550u 1.8 GHZ
Memoria RAM	16 GB
Disco Duro	2 TB
Tarjeta Gráfica	2 GB
Sistema Operativo	Ubuntu Versión 18.4.5

**Fuente:** Elaboración Propia

Después que se realizó la adquisición, se procedió a instalar el sistema operativo Ubuntu Versión 18.4.5 del tipo de Sistema operativo x 64 bits.



**Figura 30** - Descripción del Sistema Operativo



Luego se procedió a instalar el servidor web Apache (Ver anexo 2); después se procedió a instalar PHP versión 7.4; finalmente se procedió a instalar MongoDB versión 4.4.1 y Apache CouchDB versión 3.1.1.

Después de haber instalado nuestras herramientas y gestores de base de datos se construyó una plataforma web con un cronometro para calcular el tiempo de las pruebas de rendimiento (se codificó de acuerdo a los gestores de base de datos que se trabajó); se construyó esta herramienta porque CouchDB no tenía una herramienta para medir el tiempo de respuesta del procesamiento de conjunto de datos; por lo tanto también se tuvo que construir la misma herramienta para MongoDB para tener igualdad en la medición de sus criterios. A continuación, se mostró el código fuente del cronometro.

```
public function update(){

    $docs = array (
        array('_id'=>'1b627e6291afd1dd222ec16f5c79764e', '_rev'=>'9-48c7dc7c5c2556ab17a77623580a451c', 'CodigoAntiguo'=>'NA', 'CodigoNuevo'=>'NA'),
        array('_id'=>'1b627e6291afd1dd222ec16f5c7984d9', '_rev'=>'1-900174fce6b7e4efc4ac8a46f2c8d639', 'CodigoAntiguo'=>'NA', 'CodigoNuevo'=>'NA',)
    );

    $this->client->storeDocs($docs);

    //var_dump($docs);
}
public function cargartxt($archivo){
    $this->client->storeDocs($archivo);
}
}
$fp = fopen('establecimientos.csv','r');
$ArrDat = array();
$loop = 0;
while (!feof($fp)) {
    $loop++;
    $line = fgets($fp);
    $field[$loop] = explode (';', $line);
    if(strcmp($field[$loop][0], 'Codigo Antiguo') != 0){
        if(!empty($field[$loop][0])){
```

```

array_push($ArrDat, array('CodigoAntiguo'=>$field[$loop][0], 'CodigoNuevo'=>$field[$loop][1], 'CodigoMadre'=>$field[$loop][2], 'CodigoNuevoMadre'=>$field[$loop][3], 'CodigoRegion'=>$field[$loop][4], 'Region'=>$field[$loop][5], 'SEREMI'=>$field[$loop][6], 'Perteneiente'=>$field[$loop][7], 'TipoEstablecimiento'=>$field[$loop][8], 'TipoEstrategia'=>$field[$loop][9], 'Certificacion'=>$field[$loop][10], 'Dependencia'=>$field[$loop][11], 'NiveldeAtencion'=>$field[$loop][12], 'NombreOficial'=>$field[$loop][13], 'Alias'=>$field[$loop][14], 'CodigoComuna'=>$field[$loop][15], 'NombreComuna'=>$field[$loop][16], 'Via'=>$field[$loop][17], 'Numero'=>$field[$loop][18], 'Direccion'=>$field[$loop][19], 'Telefono'=>$field[$loop][20], 'FechaVigencia'=>$field[$loop][21], 'FechaReapertura'=>$field[$loop][22], 'CodigoServiciodeSalud'=>$field[$loop][23], 'SAPU'=>$field[$loop][24], 'FechaCambio'=>$field[$loop][25], 'Observacion'=>$field[$loop][26], 'LONGITUD'=>$field[$loop][27], 'LATITUD'=>$field[$loop][28]));
    }
}
fclose($fp);

$serv = new Servicio();
//$serv->cargartxt($ArrDat);
$serv->update();
?>

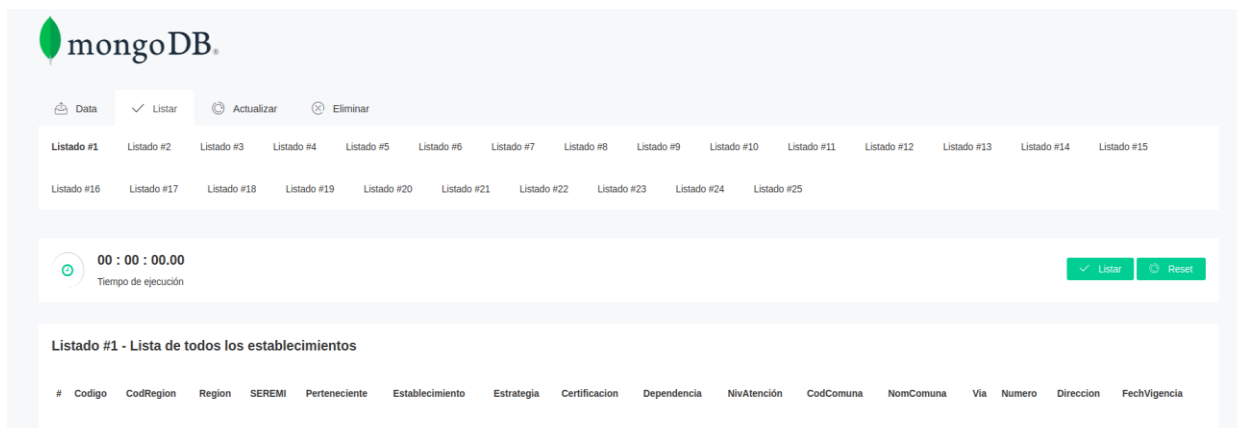
```

**Figura 31** - Código Fuente de la plataforma web Cronometro

The screenshot shows the CouchDB web interface in a browser. The URL is localhost:8082/couchdb/listar-1.php. The interface includes a CouchDB logo, a navigation bar with 'Data', 'Listar' (selected), 'Actualizar', and 'Eliminar' buttons. Below this is a grid of 25 'Listado #' buttons. A timer section shows '00 : 00 : 00.00' with 'Listar' and 'Reset' buttons. At the bottom, there is a table titled 'Listado #1 - Lista de todos los establecimientos' with columns for various attributes like Codigo, CodRegion, Region, SEREMI, etc.

#	Codigo	CodRegion	Region	SEREMI	Perteneiente	Establecimiento	Estrategia	Certificacion	Dependencia	NivAtención	CodComuna	NomComuna	Via	Numero	Direccion	FechVigencia
---	--------	-----------	--------	--------	--------------	-----------------	------------	---------------	-------------	-------------	-----------	-----------	-----	--------	-----------	--------------

**Figura 32** - Plataforma Web de Cronometro para medir tiempo de Respuesta. (CouchDB)



**Figura 33** - Plataforma Web de Cronometro para medir tiempo de Respuesta. (MongoDB)

Luego, para medir el indicador Espacio en disco, en los administradores de base de datos NoSQL Documental se realizó lo siguiente:

Para el gestor de base de datos NoSQL Documental MongoDB se ejecutó la siguiente función:

```
db.getCollection('egresoshospitalarios').dataSize({})
```

**Figura 34** - Línea de comando de MongoDB para calcular el tamaño de una colección.

Para el gestor de base de datos NoSQL Documental CouchDB se empleó el apartado que me mostró las propiedades del gestor de base datos NoSQL.

Name	Size	# of Docs
<a href="#">_replicator</a>	2.3 KB	1
<a href="#">_users</a>	2.3 KB	1
<a href="#">egresoshospitalarios</a>	1.4 GB	1566188
<a href="#">establecimientos</a>	3.2 MB	4056

**Figura 35** - Apartado de propiedades de CouchDB.

Después, para medir el indicador Cantidad de errores en el procesamiento de datos, se utilizó la sentencia try catch para detectar posibles errores en el momento de la ejecución; este try catch fue implementado en la misma plataforma web construida para medir el tiempo de respuesta. Se utiliza Try cuando desee anticipar excepciones durante la ejecución del programa. Esto es necesario porque las

declaraciones se ejecutarán, aunque la excepción esté establecida allí, por lo que la función catch se usa más tarde cuando sabe que se ha producido la excepción.

La instrucción try catch se ve así:

```
1  <?php
2  try {
3      require ('conexion-db2.php');
4      $serv = new Servicio();
5
6      $DisplayMember = $_POST["DisplayMember"];
7      $ValueMember = $_POST["ValueMember"];
8
9      $selector = [
10     $DisplayMember=>$ValueMember
11 ];
12
13     $docs = $serv->listaparametro($selector);
14
15     $ArrDat = array();
16     foreach ($docs as $key => $value) {
17         array_push($ArrDat, array('_id'=>$value->_id, '_rev'=>$value->_rev));
18     }
19
20     $serv->deleteBulk($ArrDat);
21 } catch (Exception $e) {
22     echo 'Excepción capturada: ', $e->getMessage(), "\n";
23 }
24 ?>
```

*Figura 36 - Sentencia try catch para medir el indicador cantidad de errores.*

En la presente investigación se realizó las pruebas de Rendimiento seleccionando una base de datos hospitalarios del Ministerio de Salud de Chile, las cuales contienen 02 colecciones denominadas: establecimientos con 4096 documentos y egresoshospitalarios con 1'566,188 documentos.

Asimismo, se realizó las pruebas de rendimiento las cuales se realizaron de la siguiente manera:

**A) Prueba uno:** En la prueba uno, se realizó dos operaciones de volcado de datos para ambos gestores de base de datos NoSQL documental con las colecciones establecimientos y egresoshospitalarios; donde se evaluaron los siguientes indicadores: Uso de Memoria RAM, uso de CPU, Tiempo de Respuesta, Espacio Usado en disco y Cantidad de errores. Asimismo, se mostró los valores que arrojó la prueba uno.

## Volcado 1

Se realizó el volcado de datos con la colección establecimientos que contiene 4056 documentos.

### 1. Consumo de Memoria RAM

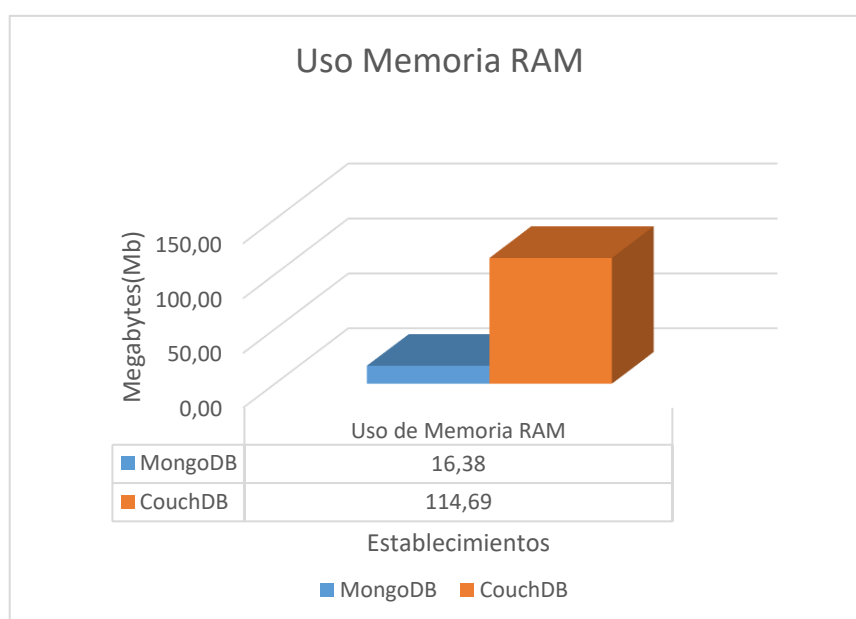
**Tabla 25** - Cantidad de Uso de Memoria RAM (Volcado 1).

Gestor de Base de Datos NoSQL	UI	UF	PUR	Uso de Memoria RAM (Mb)
MongoDB	0.4%	0.5%	0.1%	16.38
CouchDB	0.3%	1.0%	0.7%	114.69

**Fuente:** Elaboración Propia.

Según esta tabla observamos que MongoDB empleo menos uso de memoria RAM para el volcado de datos con la colección establecimientos; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el uso de memoria RAM empleado en el volcado de datos donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 37** - Gráfico de uso de memoria RAM (Volcado 1).

## 2. Consumo de CPU

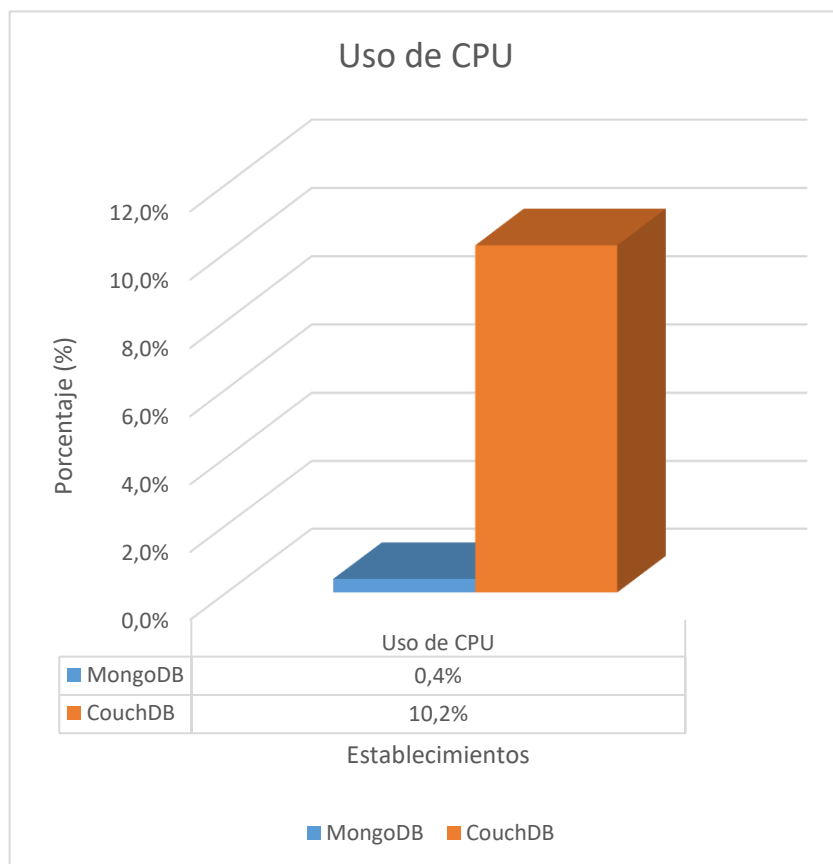
**Tabla 26** - Cantidad de Uso de CPU (Volcado 1).

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	0.7%	0.4%
CouchDB	2.0%	12.2%	10.2%

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB emplea menos uso de CPU para el volcado de datos con la colección establecimientos; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el uso de CPU empleado en el volcado de datos donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 38** - Gráfico de uso de CPU (Volcado 1).

### 3. Tiempo de respuesta

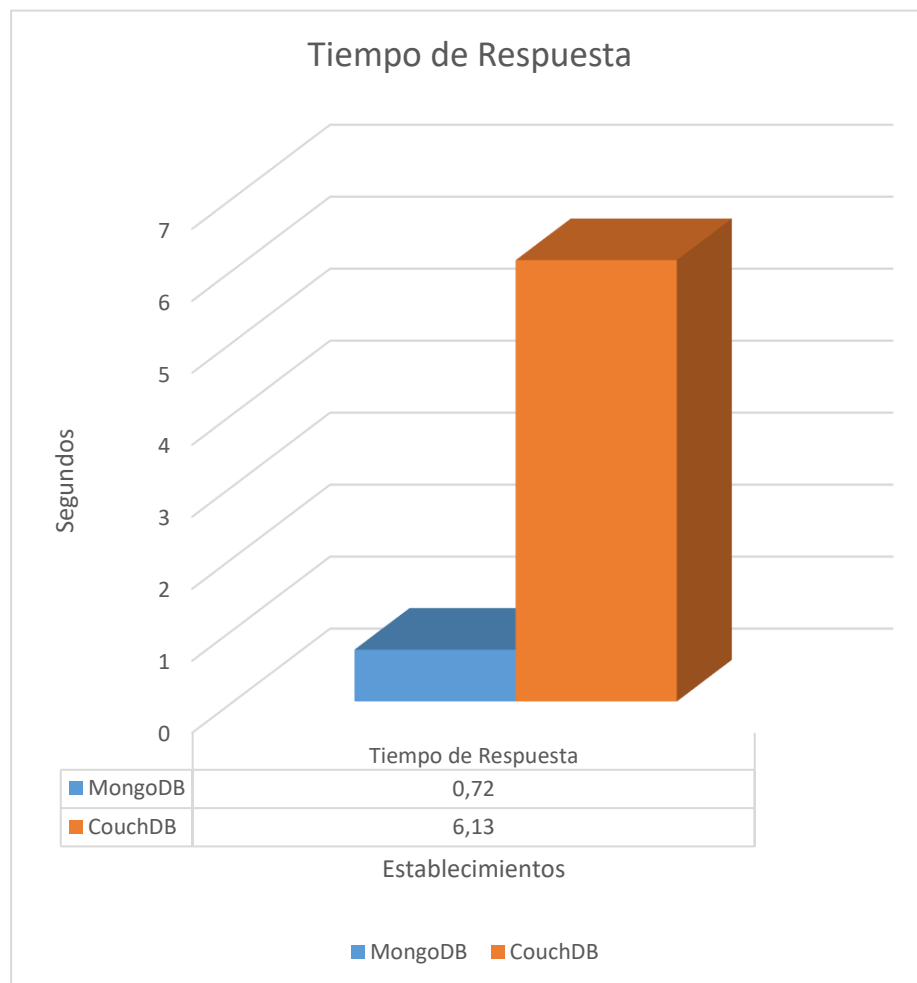
*Tabla 27 – Uso del Tiempo de Respuesta (Volcado 1).*

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	39.6%	39.3%
CouchDB	0.7%	124.3%	123.6%

**Fuente:** *Elaboración Propia.*

Según esta tabla, observamos que MongoDB empleo menos tiempo de respuesta para el volcado de datos con la colección establecimientos; el tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el tiempo de respuesta empleado en el volcado de datos donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 39 - Gráfico del Uso del tiempo de respuesta (Volcado 1).**

#### 4. Espacio Usado en Disco

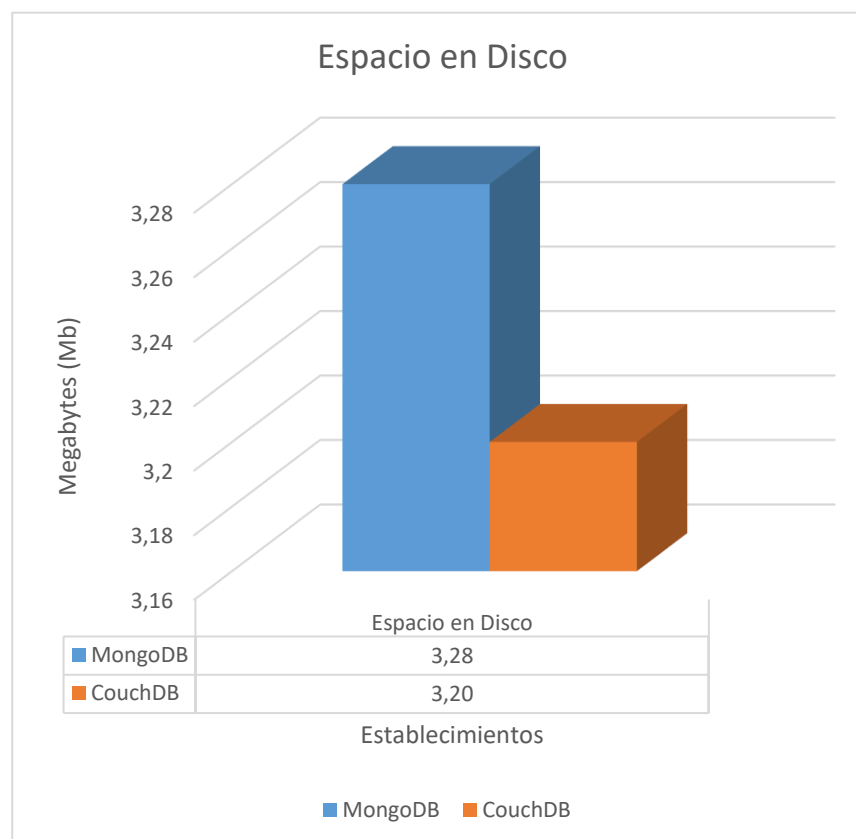
*Tabla 28 - Espacio Usado en Disco (Volcado 1).*

Gestor de Base de Datos NoSQL	EUI	EUF	ED
<b>MongoDB</b>	0	3.28	3.28 Mb
<b>CouchDB</b>	0	3.20	3.20 Mb

*Fuente: Elaboración Propia.*

Según esta tabla, observamos que CouchDB empleó menos espacio en disco para el volcado de datos con la colección establecimientos; el espacio en disco se mostró en megabytes.

En la siguiente figura se mostró el espacio en disco empleado en el volcado de datos donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



*Figura 40 - Gráfico de Espacio usado en Disco (Volcado 1).*

#### 5. Cantidad de errores

No se presentó errores en el volcado uno de la colección establecimientos.



## Volcado 2

Se realizó el volcado de datos con la colección egresoshospitalarios que contiene 1'566,188 documentos.

### 1. Consumo de Memoria RAM

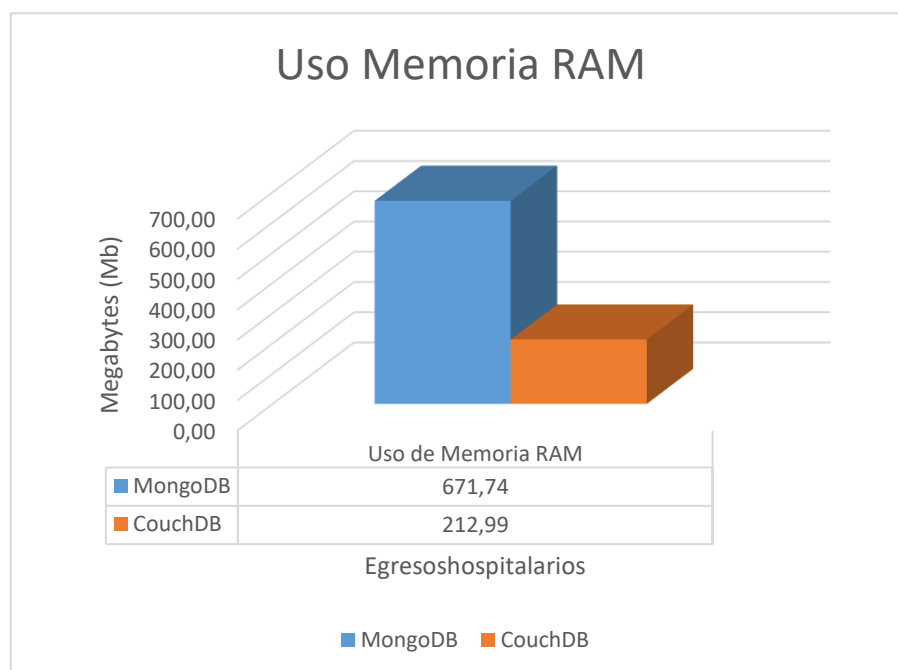
**Tabla 29** - Cantidad de Uso de Memoria RAM (Volcado 2).

Gestor de Base de Datos NoSQL	UI	UF	PUR	Uso de Memoria RAM (Mb)
MongoDB	0.5%	4.6%	4.1%	671.74
CouchDB	0.4%	1.7%	1.3%	212.99

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que CouchDB empleo menos uso de memoria RAM para el volcado de datos con la colección egresoshospitalarios; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el uso de memoria RAM empleado en el volcado de datos donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 41** - Gráfico de uso de memoria RAM (Volcado 2).

## 2. Consumo de CPU

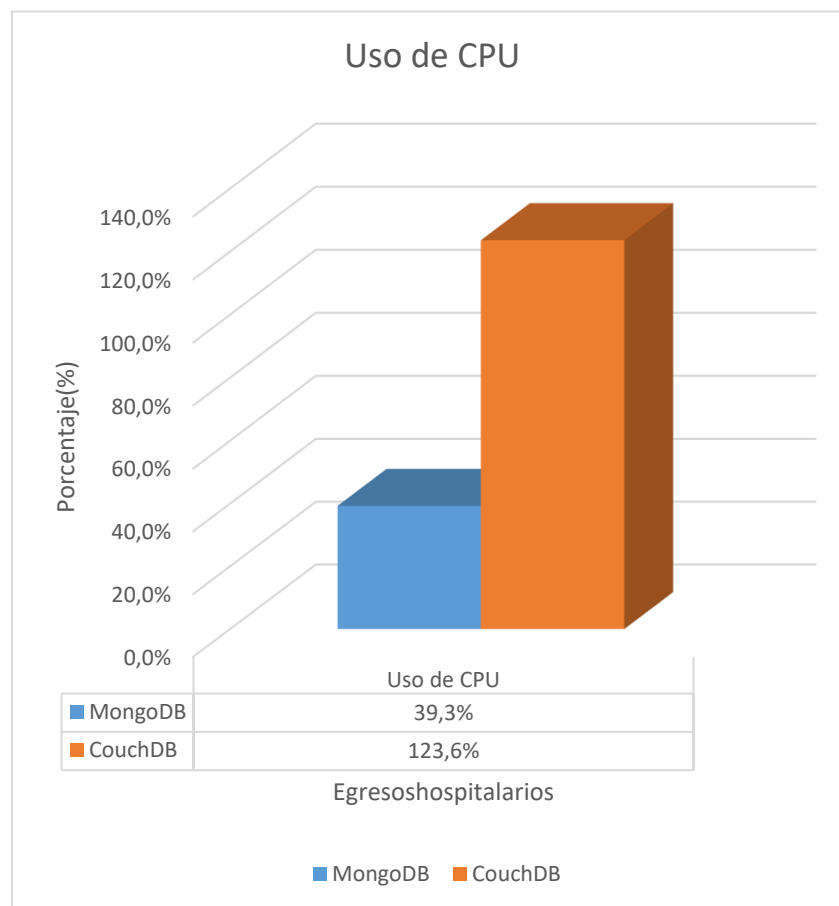
**Tabla 30** - Cantidad de uso de CPU (Volcado 2).

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	39.6%	39.3%
CouchDB	0.7%	124.3%	123.6%

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB empleó menos uso de CPU para el volcado de datos con la colección egresoshospitalarios; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el uso de CPU empleado en el volcado de datos donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 42** - Gráfico de la cantidad de Uso de CPU (Volcado 2).

### 3. Tiempo de Respuesta

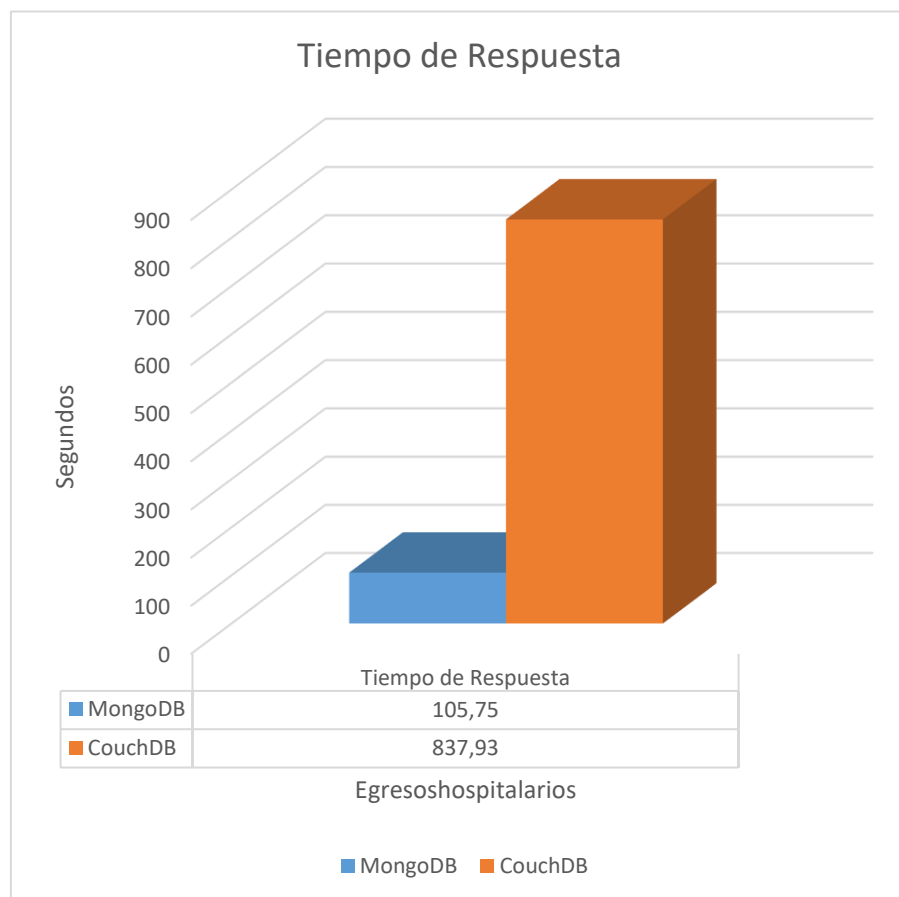
**Tabla 31** – *Uso de Tiempo de Respuesta (Volcado 2).*

Gestor de Base de Datos NoSQL	TI	TF	TR
MongoDB	0	105.75	105.75
CouchDB	0	837.93	837.93

**Fuente:** *Elaboración Propia.*

Según esta tabla, observamos que MongoDB empleó menos tiempo de respuesta para el volcado de datos con la colección egresoshospitalarios; el tiempo de respuesta se mostró en segundos

En la siguiente figura se mostró el tiempo de respuesta empleado en el volcado de datos donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 43** - *Gráfico del Uso de Tiempo de Respuesta (Volcado 2).*

#### 4. Espacio en Disco

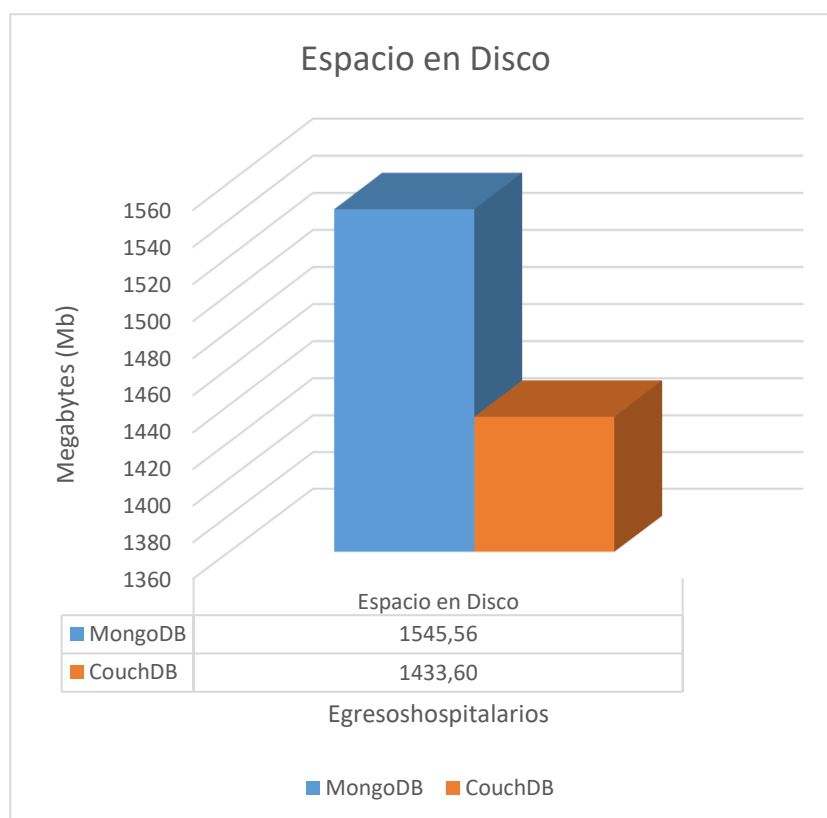
**Tabla 32** - *Espacio Usado en disco (Volcado 2).*

Gestor de Base de Datos NoSQL	EUI	EUf	ED
MongoDB	0	1545.56	1545.56 Mb
CouchDB	0	1433.60	1433.60 Mb

**Fuente:** *Elaboración Propia.*

Según esta tabla, observamos que CouchDB empleó menos espacio en disco para el volcado de datos con la colección egresoshospitalarios; el espacio en disco se mostró en megabytes.

En la siguiente figura se mostró el espacio en disco empleado en el volcado de datos donde se usó la colección egresoshospitalarios; en ambos gestores de base de datos NoSQL Documental.



**Figura 44** – *Gráfico del Espacio usado en Disco (Volcado 2).*

## 5. Cantidad de Errores

No se presentó errores en el volcado dos de la colección egresoshospitalarios.

**B) Prueba Dos:** En la prueba dos, se realizó cuatro operaciones de listado de datos para ambos gestores de base de datos NoSQL documental con las colecciones establecimientos y egresoshospitalarios; donde se evaluaron los siguientes indicadores: Uso de Memoria RAM, uso de CPU, Tiempo de Respuesta y Cantidad de errores. Asimismo, se mostró los valores que arrojó la prueba dos.

### Listado 1:

En el listado 1, se usó la colección establecimientos, donde se realizó el listado de todos los establecimientos de la colección.

#### 1. Consumo de Memoria RAM

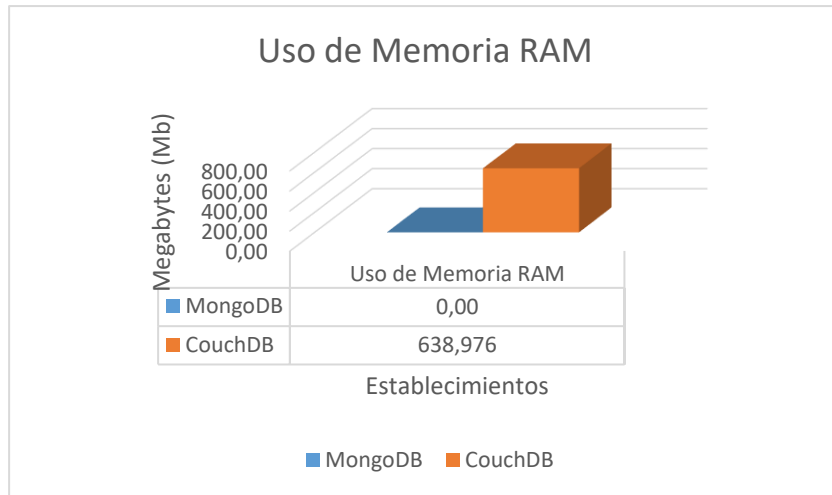
*Tabla 33 - Cantidad de Uso de Memoria RAM (Listado 1).*

Gestor de Base de Datos NoSQL	UI	UF	PUR	Uso de Memoria RAM (Mb)
MongoDB	11.9%	11.9%	0.0%	0.00
CouchDB	0.6%	4.5%	3.9%	638.98

**Fuente:** *Elaboración Propia.*

Según esta tabla, observamos que MongoDB empleo menos uso de memoria RAM para el listado de datos 1, con la colección establecimientos; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el uso de memoria RAM empleado en el listado de datos 1, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 45 - Gráfico de consumo de Memoria RAM (Listado 1).**

## 2. Consumo de CPU

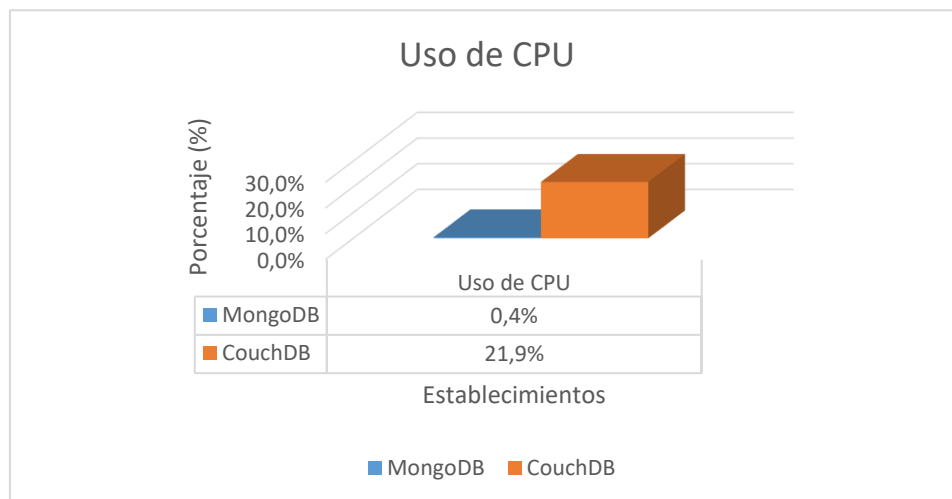
**Tabla 34 - Cantidad de Uso de CPU (Listado 1).**

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	0.7%	0.4%
CouchDB	4.3%	26.2%	21.9%

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB emplea menos uso de CPU para el listado de datos 1, con la colección establecimientos; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el uso de CPU empleado en el listado de datos 1, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 46 - Gráfico de cantidad de uso de CPU (Listado 1).**

### 3. Tiempo de Respuesta

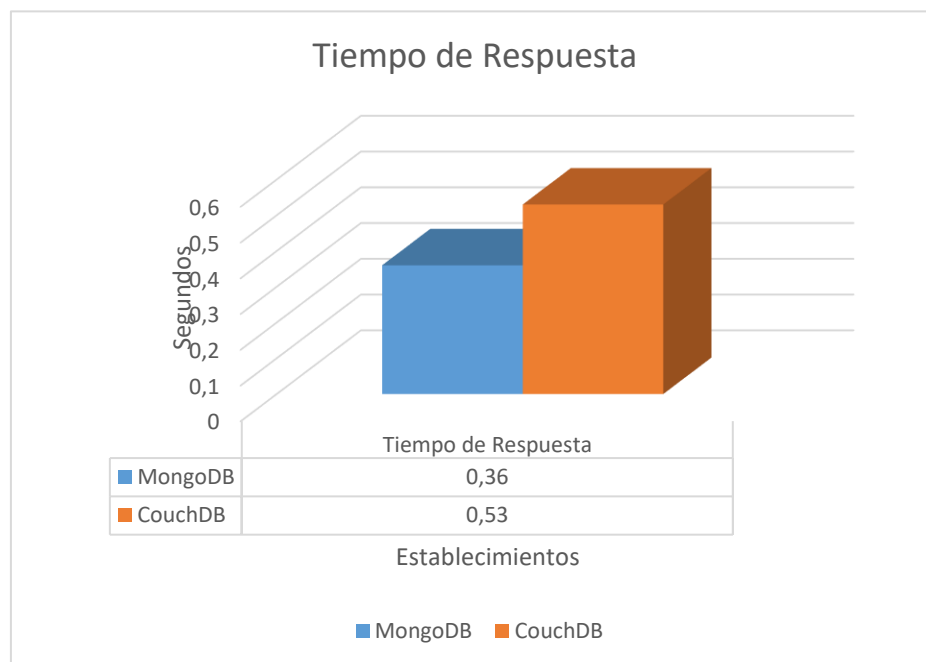
*Tabla 35 – Uso del Tiempo de Respuesta (Listado 1).*

Gestor de Base de Datos NoSQL	TI	TF	TR
MongoDB	0	0.36	0.36
CouchDB	0	0.53	0.53

**Fuente:** *Elaboración Propia.*

Según esta tabla, observamos que MongoDB empleó menos tiempo de respuesta para el listado de datos 1, con la colección establecimientos; el uso del tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el uso del tiempo de respuesta empleado en el listado de datos 1, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 47 - Gráfico del Uso de Tiempo de respuesta (Listado 1).**

### 4. Cantidad de Errores

No se presentó errores en la prueba dos – Listado 1.

## Listado 2:

En el listado 2, se usó la colección establecimientos, donde se realizó el listado de todos los establecimientos de la región Tarapacá.

### 1. Consumo de Memoria RAM

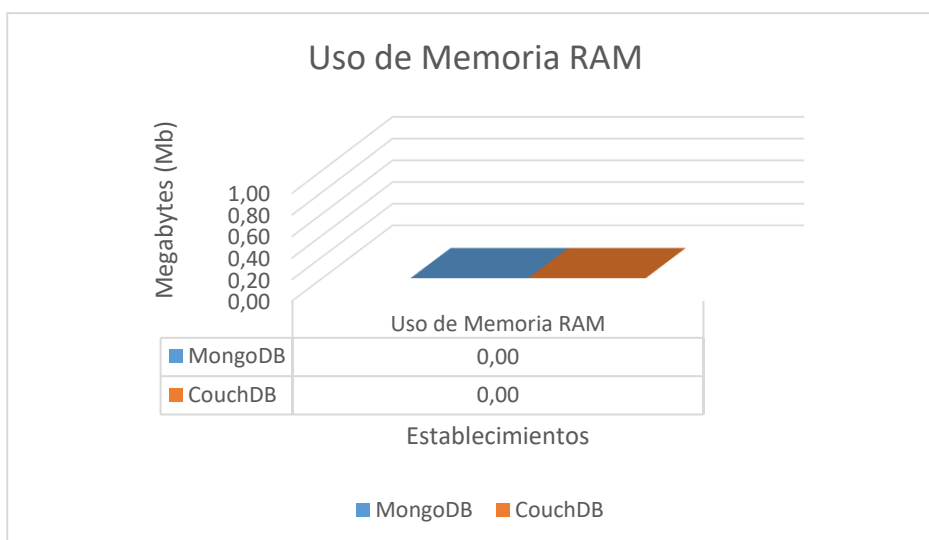
**Tabla 36** – Cantidad de uso de Memoria RAM (Listado 2).

Gestor de Base de Datos NoSQL	UI	UF	PUR	Uso de Memoria RAM (Mb)
MongoDB	11.9%	11.9%	0.0%	0.00
CouchDB	0.6%	0.6%	0.0%	0.00

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB y CouchDB ambos gestores de base de datos NoSQL documental usaron la misma cantidad de memoria RAM para el listado de datos 2, con la colección establecimientos; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el uso de memoria RAM empleado en el listado de datos 2, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 48** - Gráfico cantidad de uso de Memoria RAM (Listado 2).



## 2. Consumo de CPU

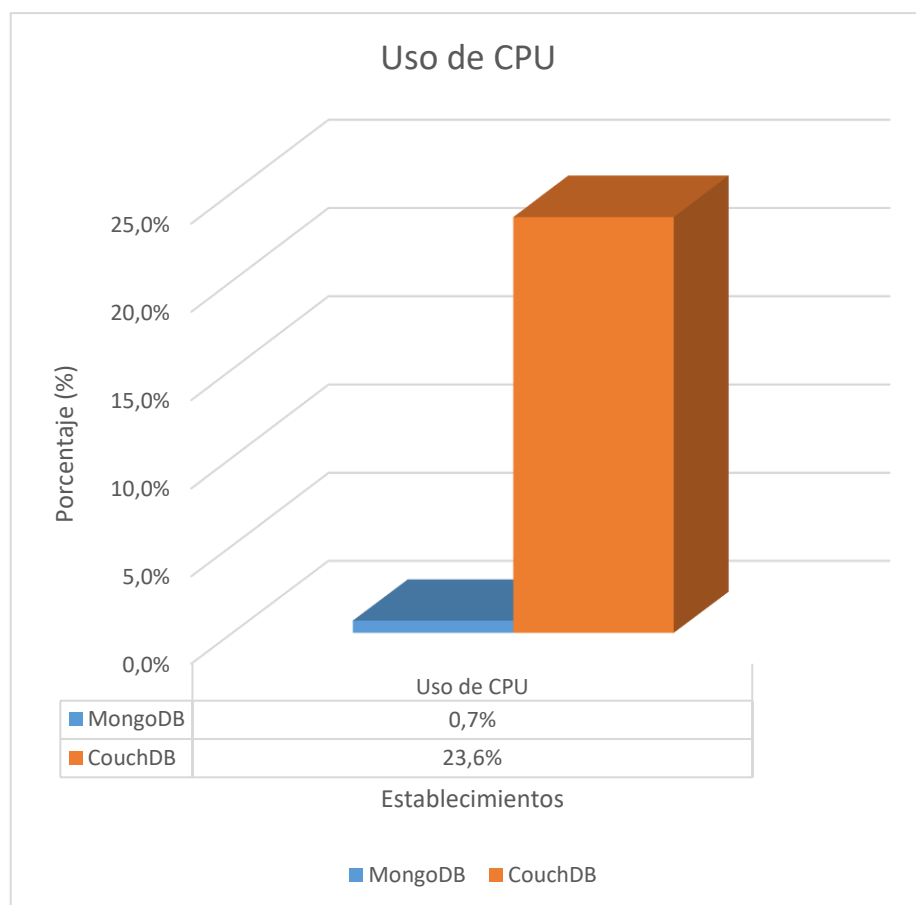
**Tabla 37** - Cantidad de uso de CPU (Listado 2).

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	1.0%	0.7%
CouchDB	1.7%	25.3%	23.6%

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB empleo menos uso de CPU para el listado de datos 2, con la colección establecimientos; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el uso de CPU empleado en el listado de datos 2, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 49** - Gráfico de cantidad de CPU (Listado 2).

### 3. Tiempo de Respuesta

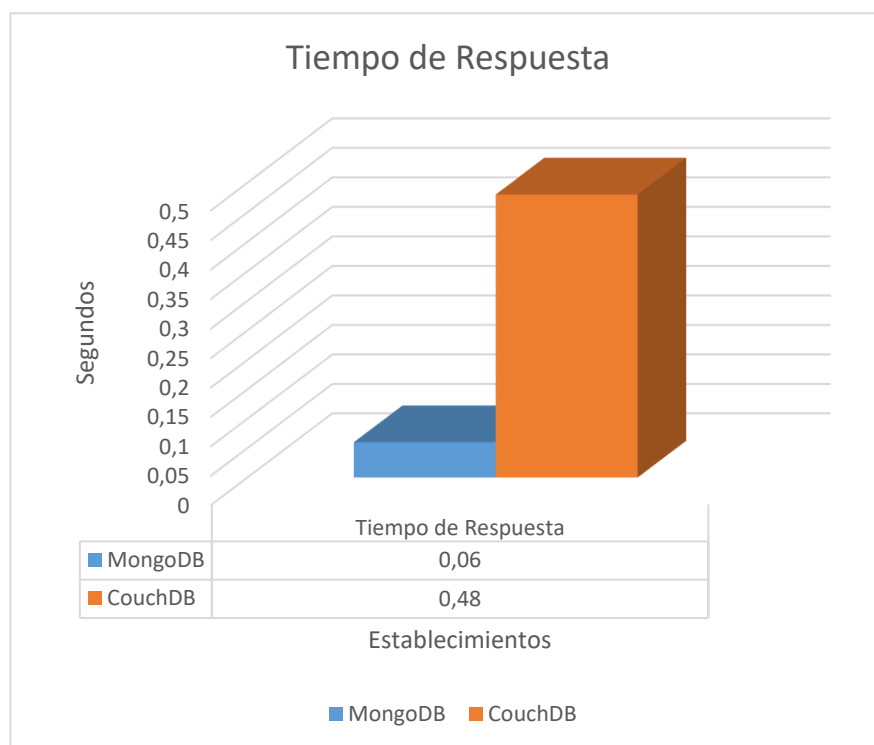
**Tabla 38** – *Uso de Tiempo de Respuesta (Listado 2).*

Gestor de Base de Datos NoSQL	TI	TF	TR
MongoDB	0	0.06	0.06
CouchDB	0	0.48	0.48

**Fuente:** *Elaboración Propia.*

Según esta tabla, observamos que MongoDB empleó menos tiempo de respuesta para el listado de datos 2, con la colección establecimientos; el uso del tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el uso del tiempo de respuesta empleado en el listado de datos 1, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 50** - *Gráfico del Uso de tiempo de respuesta (Listado 2).*

### 4. Cantidad de Errores

No se presentó errores en la prueba dos – Listado 2.

### Listado 3:

En el listado 3, se usó la colección egresoshospitalarios, donde se realizó el listado de todos los egresos hospitalarios de la Comuna de Residencia de Antofagasta.

#### 1. Consumo de Memoria RAM

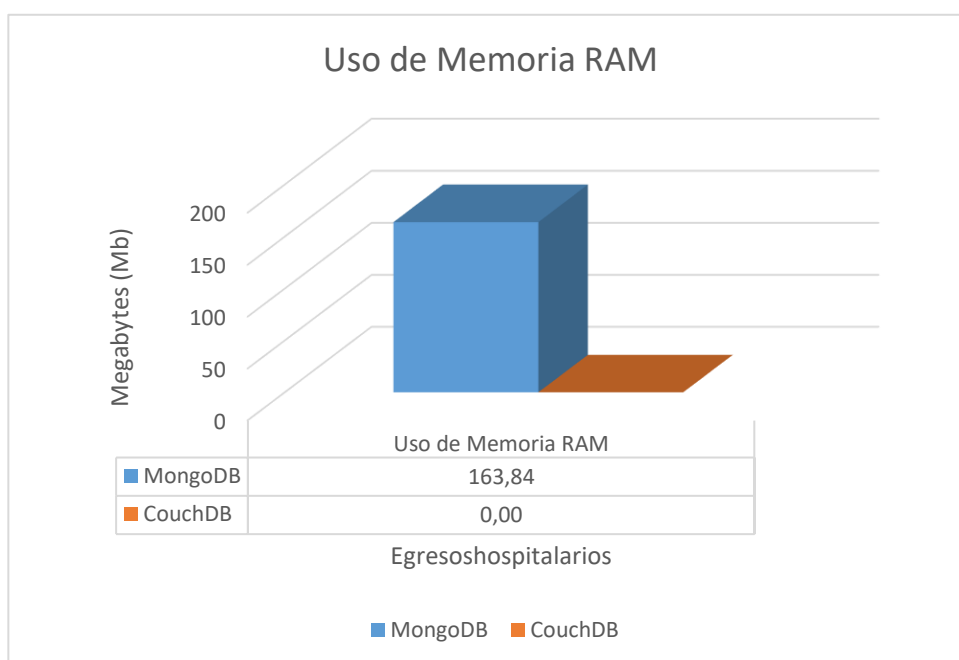
**Tabla 39** - Cantidad de uso de Memoria RAM (Listado 3).

Gestor de Base de Datos NoSQL	UI	UF	PUR	Uso de Memoria RAM (Mb)
MongoDB	11.9%	12.9%	1.0%	163.84
CouchDB	0.6%	0.6%	0.0%	0.00

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que CouchDB emplea menos uso de memoria RAM para el listado de datos 3, con la colección egresoshospitalarios; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el uso de memoria RAM empleado en el listado de datos 3, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 51** - Gráfico de cantidad de uso Memoria RAM (Listado 3).

## 2. Consumo de CPU

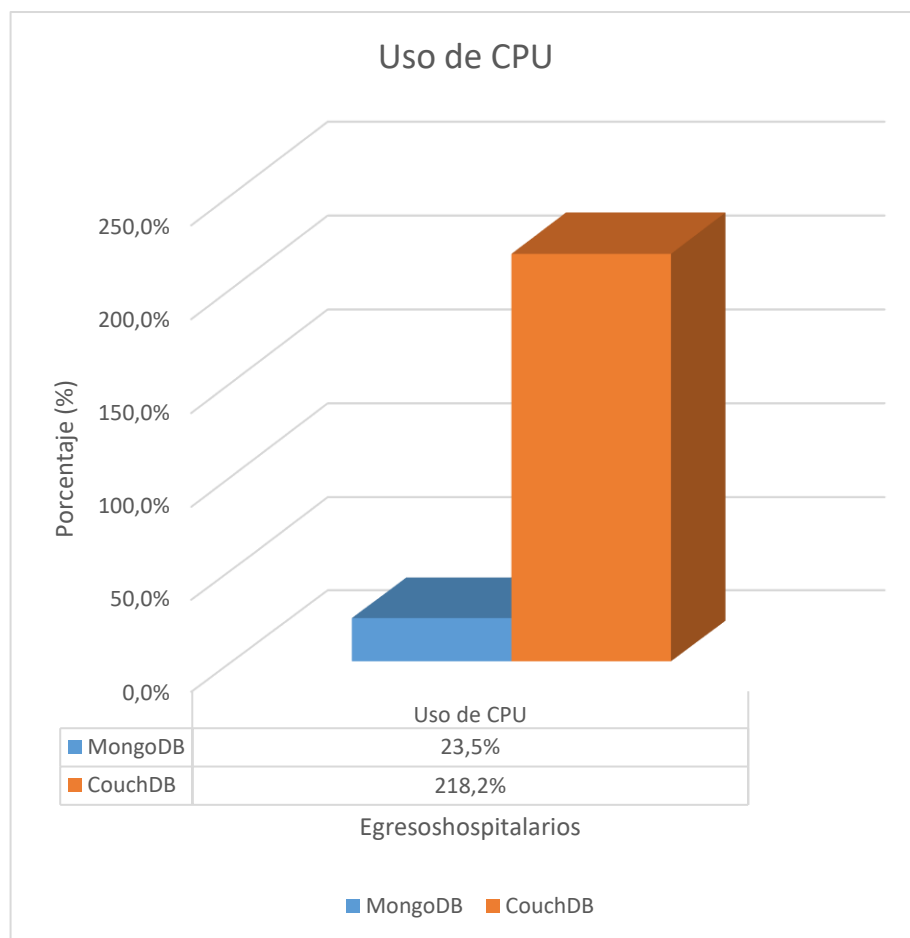
**Tabla 40** - Cantidad de uso de CPU (Listado 3).

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	23.8%	23.5%
CouchDB	3.0%	221.2%	218.2%

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB emplea menos uso de CPU para el listado de datos 3, con la colección egresoshospitalarios; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el uso de CPU empleado en el listado de datos 3, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 52** - Gráfico Cantidad de uso de CPU (Listado 3).

### 3. Tiempo de Respuesta

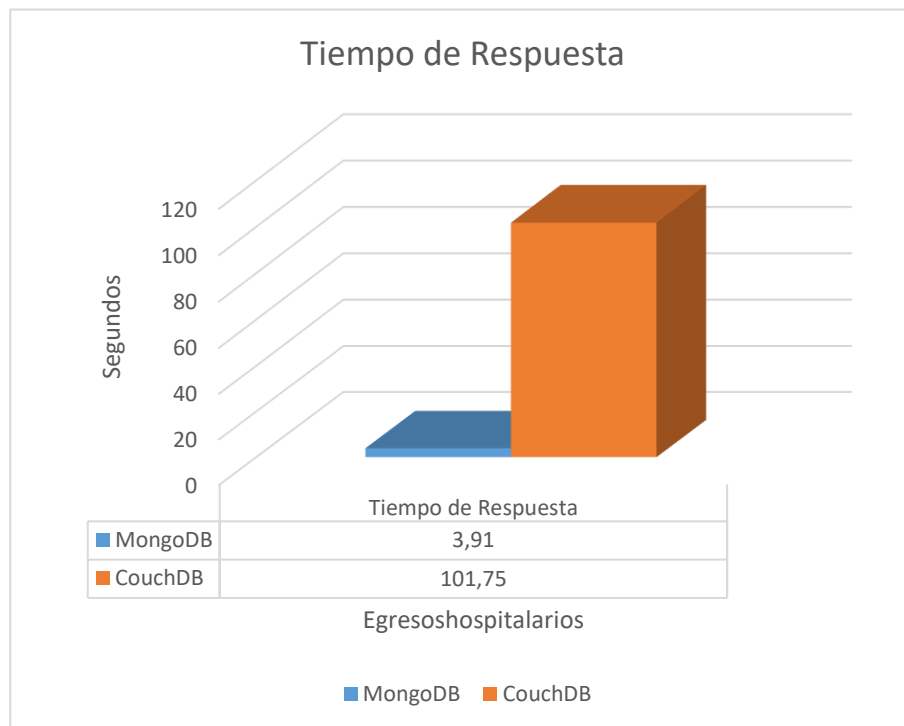
**Tabla 41** – *Uso de Tiempo de Respuesta (Listado 3).*

Gestor de Base de Datos NoSQL	TI	TF	TR
MongoDB	0	3.91	3.91
CouchDB	0	101.75	101.75

**Fuente:** *Elaboración Propia.*

Según esta tabla, observamos que MongoDB empleo menos tiempo de respuesta para el listado de datos 3, con la colección egresoshospitalarios; el uso del tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el uso del tiempo de respuesta empleado en el listado de datos 3, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 53** - *Gráfico de Uso de Tiempo de Respuesta (Listado 3).*

### 4. Cantidad de Errores

No se presentó errores en la prueba dos – Listado 3.

#### Listado 4:

En el listado 4, se usó la colección egresoshospitalarios, donde se realizó el listado de todos los egresos hospitalarios de la Comuna de Residencia de Quilicura.

#### 1. Consumo de Memoria RAM

Tabla 42 - Cantidad de Uso de Memoria RAM (Listado 4).

Gestor de Base de Datos NoSQL	UI	UF	PUR	Uso de Memoria RAM (Mb)
MongoDB	12.9%	12.9%	0.0%	0.00
CouchDB	0.6%	0.6%	0.0%	0.00

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB y CouchDB ambos gestores de base de datos NoSQL documental usaron la misma cantidad de memoria RAM para el listado de datos 4, con la colección egresoshospitalarios; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el uso de memoria RAM empleado en el listado de datos 4, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental

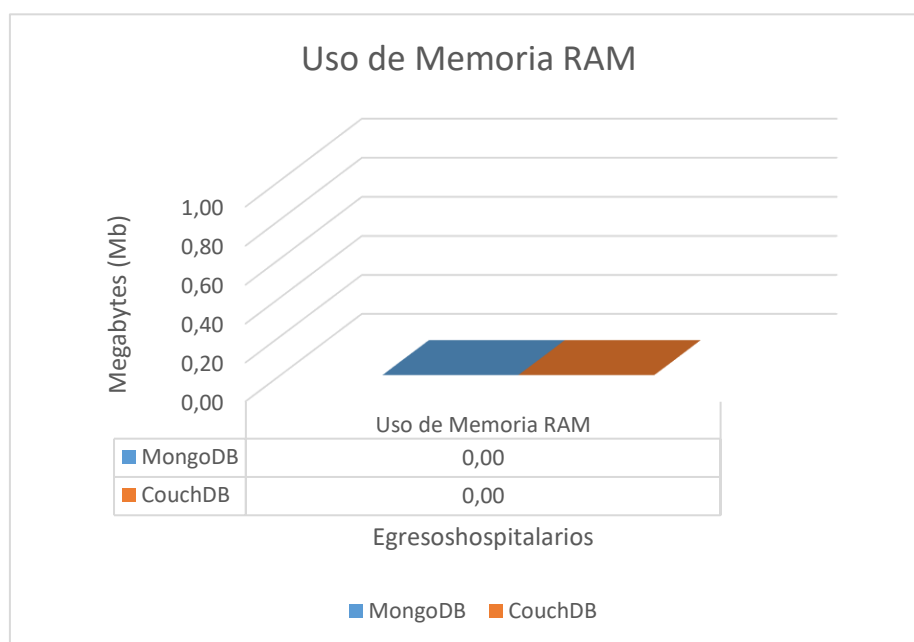


Figura 54 - Gráfico de Cantidad de Uso de Memoria RAM (Listado 4).

## 2. Consumo de CPU

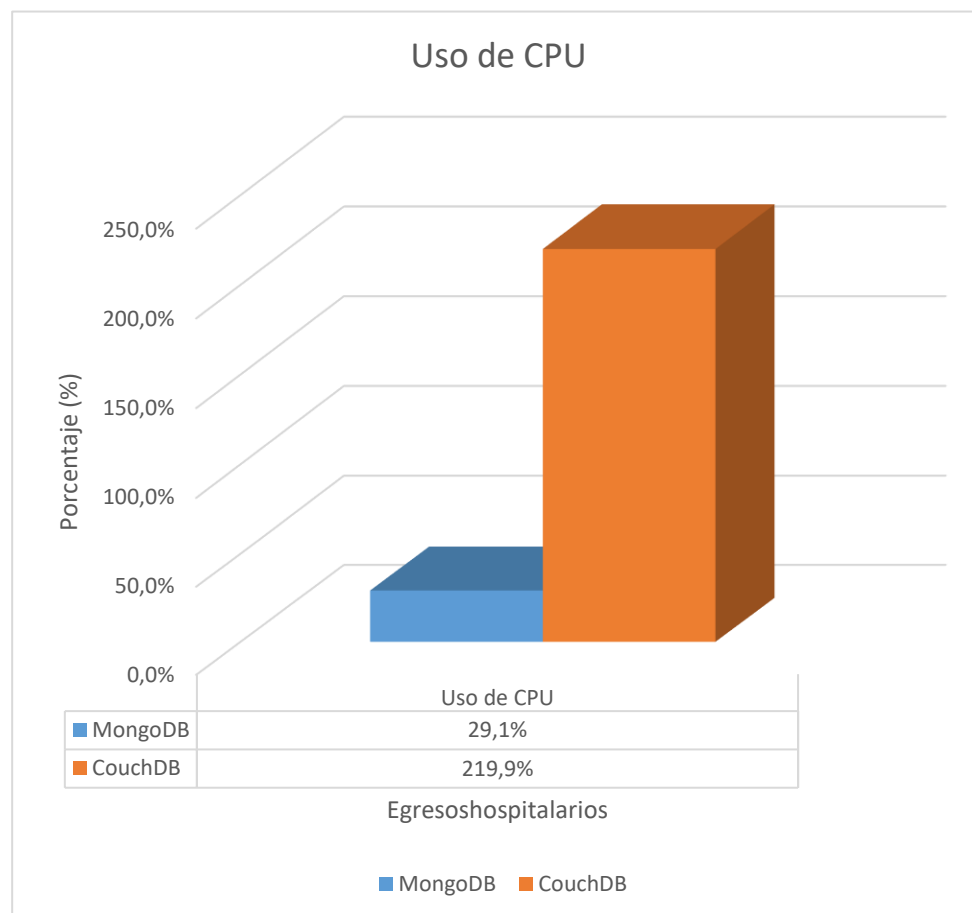
**Tabla 43** - Cantidad de Uso de CPU (Listado 4).

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	29.4%	29.1%
CouchDB	5.3%	225.2%	219.9%

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB emplea menos uso de CPU para el listado de datos 4, con la colección egresoshospitalarios; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el uso de CPU empleado en el listado de datos 4, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 55** - Gráfico Cantidad de Uso de CPU (Listado 4).

### 3. Tiempo de Respuesta

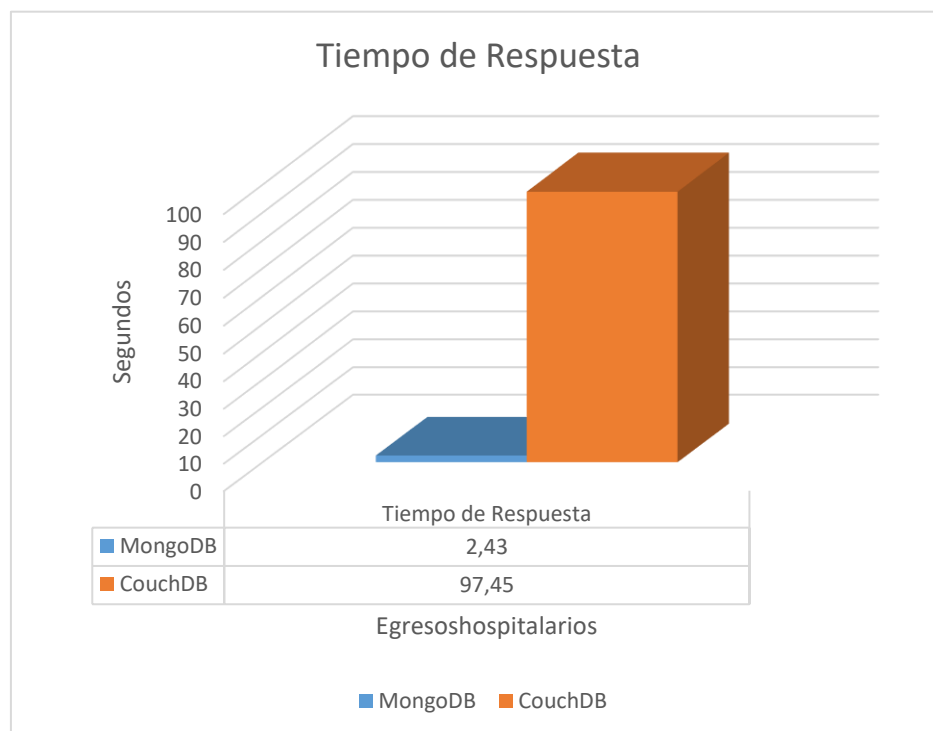
**Tabla 44** – *Uso de Tiempo de Respuesta (Listado 1).*

Gestor de Base de Datos NoSQL	TI	TF	TR
MongoDB	0	2.43	2.43
CouchDB	0	97.45	97.45

**Fuente:** *Elaboración Propia.*

Según esta tabla, observamos que MongoDB empleo menos tiempo de respuesta para el listado de datos 4, con la colección egresoshospitalarios; el uso del tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el uso del tiempo de respuesta empleado en el listado de datos 4, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 56** - *Gráfico de Uso de Tiempo de Respuesta (Listado).*

### 4. Cantidad de Errores

No se presentó errores en la prueba dos – Listado 4.



**C) Prueba Tres:** En la Prueba tres, se realizó cuatro operaciones de actualizar datos para ambos gestores de base de datos NoSQL documental con las colecciones establecimientos y egresoshospitalarios; donde se evaluaron los siguientes indicadores: Uso de Memoria RAM, uso de CPU, Tiempo de Respuesta y Cantidad de errores. Asimismo, se mostró los valores que arrojó la prueba tres.

## Actualización 1

En la actualización 1, se usó la colección establecimientos, donde se realizó la actualización de todos los establecimientos de la Región Metropolitana de Santiago.

### 1. Consumo de Memoria RAM

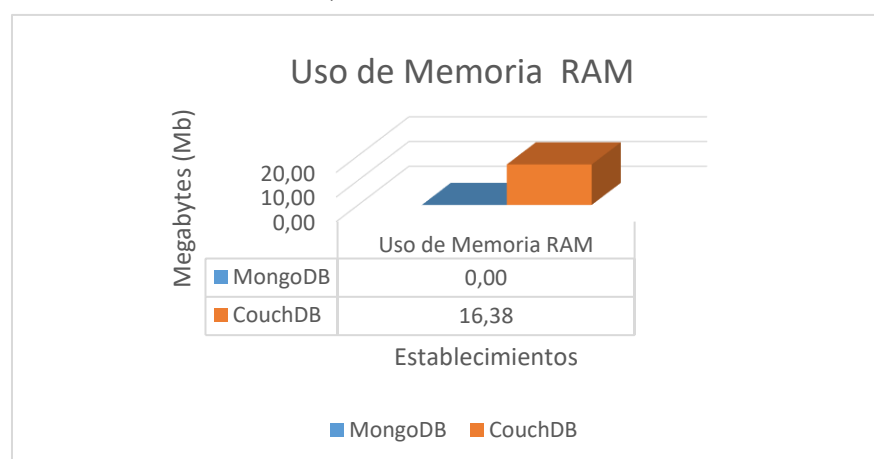
**Tabla 45 - Cantidad de Uso de Memoria RAM (Actualización 1)**

Gestor de Base de Datos NoSQL	UI	UF	PUR	Uso de Memoria RAM (Mb)
MongoDB	13.0%	13.0%	0.0%	0.00
CouchDB	0.6%	0.7%	0.1%	16.38

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB empleo menos uso de memoria RAM para la actualización 1, con la colección establecimientos; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el uso de memoria RAM empleado en la actualización de datos 1, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 57 - Gráfico de Consumo de Memoria RAM (Actualización 1).**

## 2. Consumo de CPU

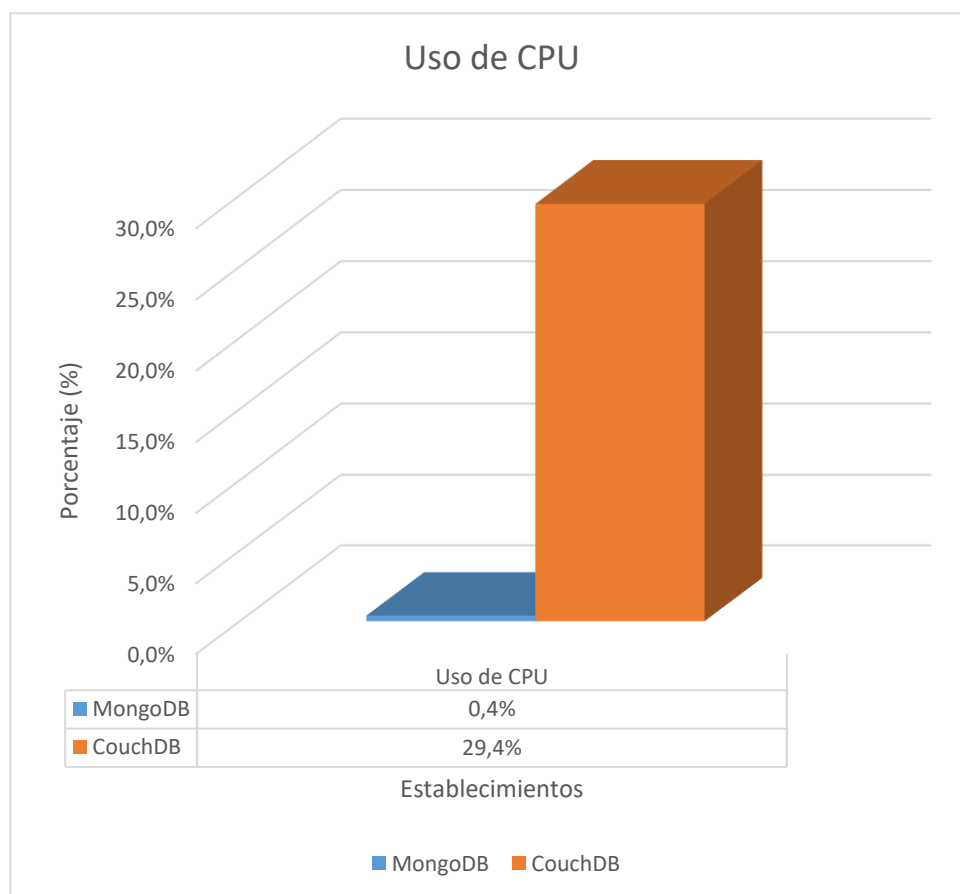
**Tabla 46** - Cantidad de Uso de CPU (Actualización 1).

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	0.7%	0.4%
CouchDB	3.6%	33.0%	29.4%

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB emplea menos uso de CPU para la actualización de datos 1, con la colección establecimientos; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el uso de CPU empleado en la actualización de datos 1, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 58** - Gráfico de la Cantidad de Uso de CPU (Actualización 1).

### 3. Tiempo de respuesta

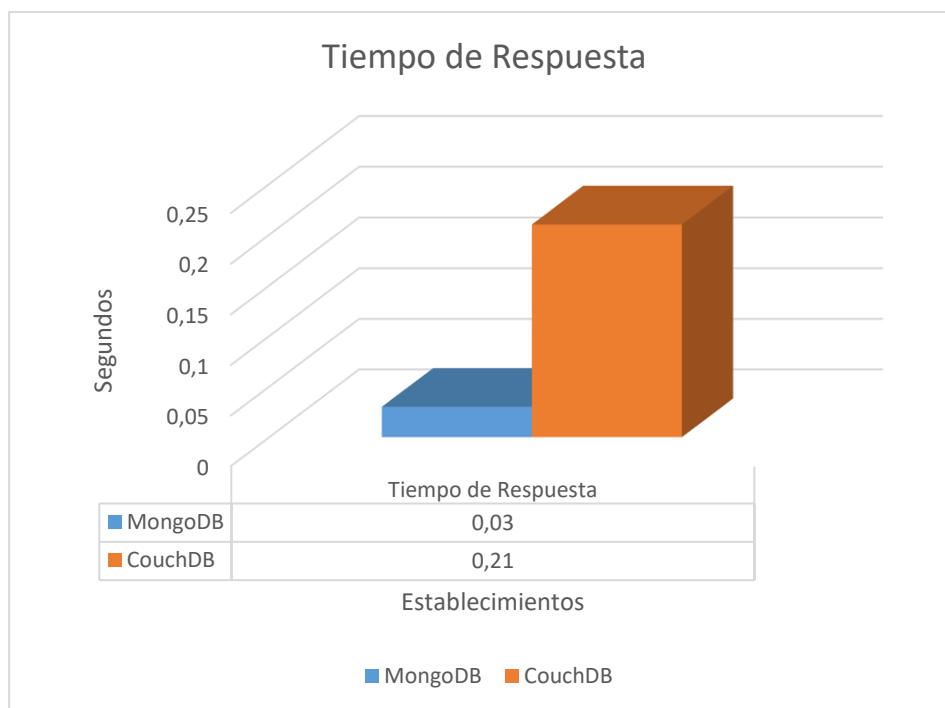
**Tabla 47 – Uso del Tiempo de Respuesta (Actualización 1)**

Gestor de Base de Datos NoSQL	TI	TF	TR
MongoDB	0	0.03	0.03
CouchDB	0	0.21	0.21

**Fuente:** *Elaboración Propia.*

Según esta tabla, observamos que MongoDB empleó menos tiempo de respuesta para la actualización de datos 1, con la colección establecimientos; el uso del tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el uso del tiempo de respuesta empleado en la actualización de datos 1, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 59 - Gráfico del Uso del Tiempo de Respuesta (Actualización 1).**

### 4. Cantidad de errores.

No se presentó errores en la prueba tres – actualización 1.

## Actualización 2

En la actualización 2, se usó la colección establecimientos, donde se realizó la actualización de todos los establecimientos de la Región de los Ríos.

### 1. Consumo de Memoria RAM

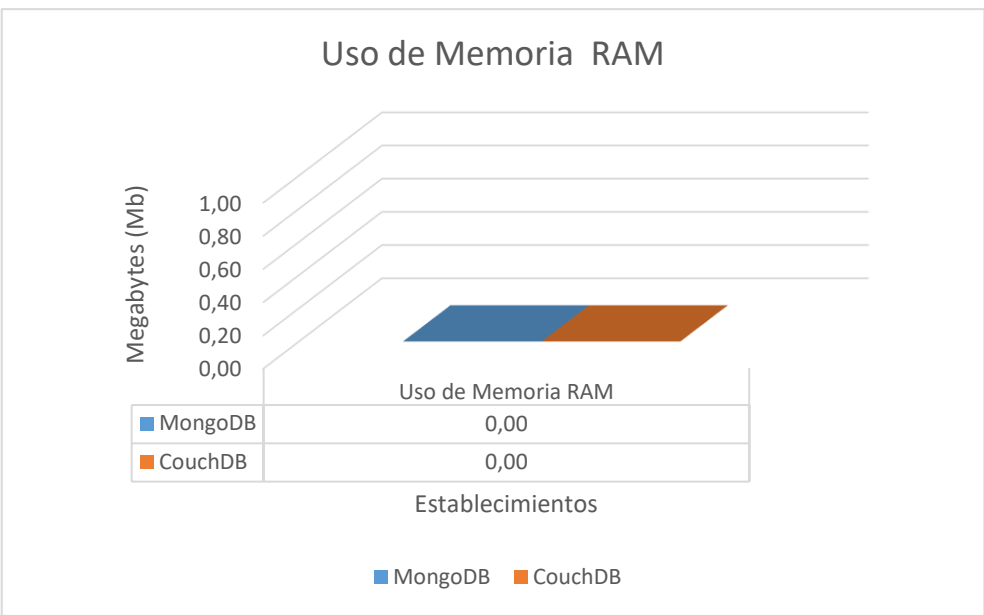
**Tabla 48** - Cantidad de Uso de Memoria RAM (Actualización 2).

Gestor de Base de Datos NoSQL	UI	UF	PUR	Uso de Memoria RAM (Mb)
MongoDB	13.0%	13.0%	0.0%	0.00
CouchDB	0.6%	0.6%	0.0%	0.00

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB y CouchDB ambos gestores de base de datos NoSQL documental usaron la misma cantidad de memoria RAM para la actualización de datos 2, con la colección establecimientos; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el uso de memoria RAM empleado en la actualización de datos 2, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 60** - Gráfico de Cantidad de Uso de Memoria RAM (Actualización 2).

## 2. Consumo de CPU

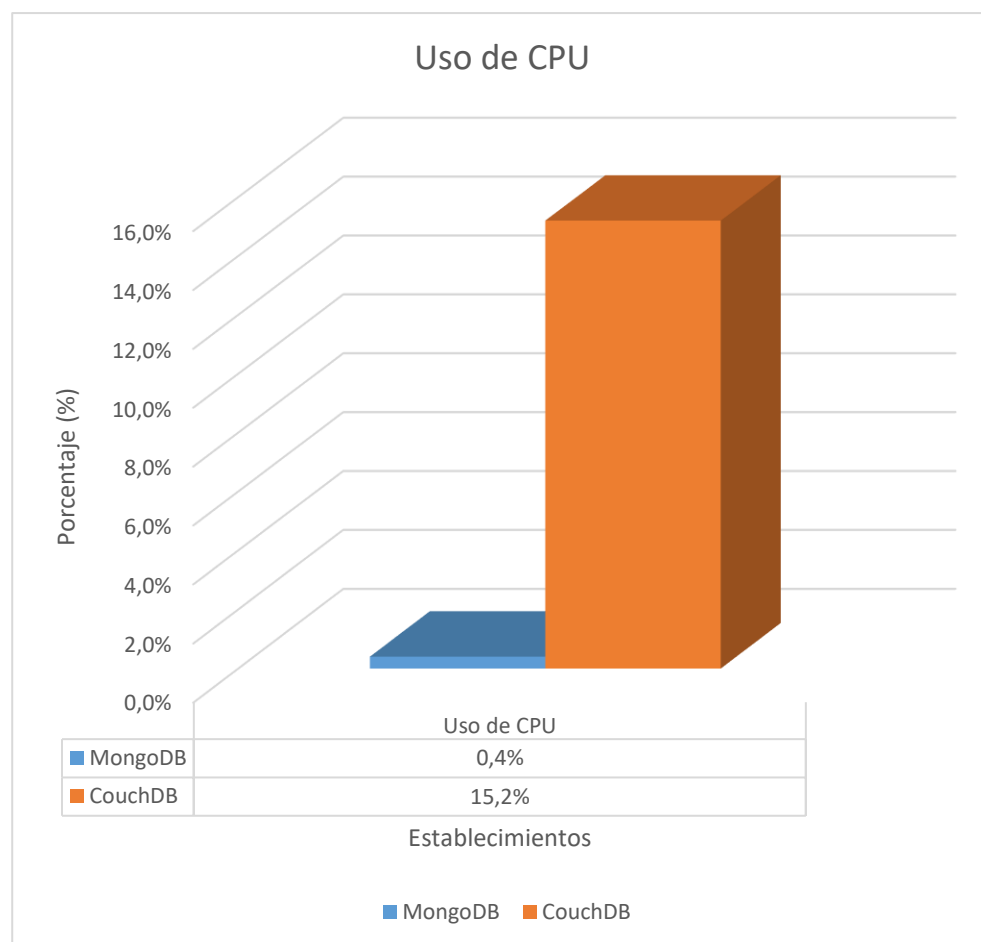
**Tabla 49** - Cantidad de Uso de CPU (Actualización 2).

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	0.7%	0.4%
CouchDB	6.6%	21.8%	15.2%

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB empleo menos uso de CPU para la actualización de datos 2, con la colección establecimientos; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el uso de CPU que se empleó en la actualización de datos 2, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 61** - Gráfico de Cantidad de Uso de CPU (Actualización 2).

### 3. Tiempo de respuesta

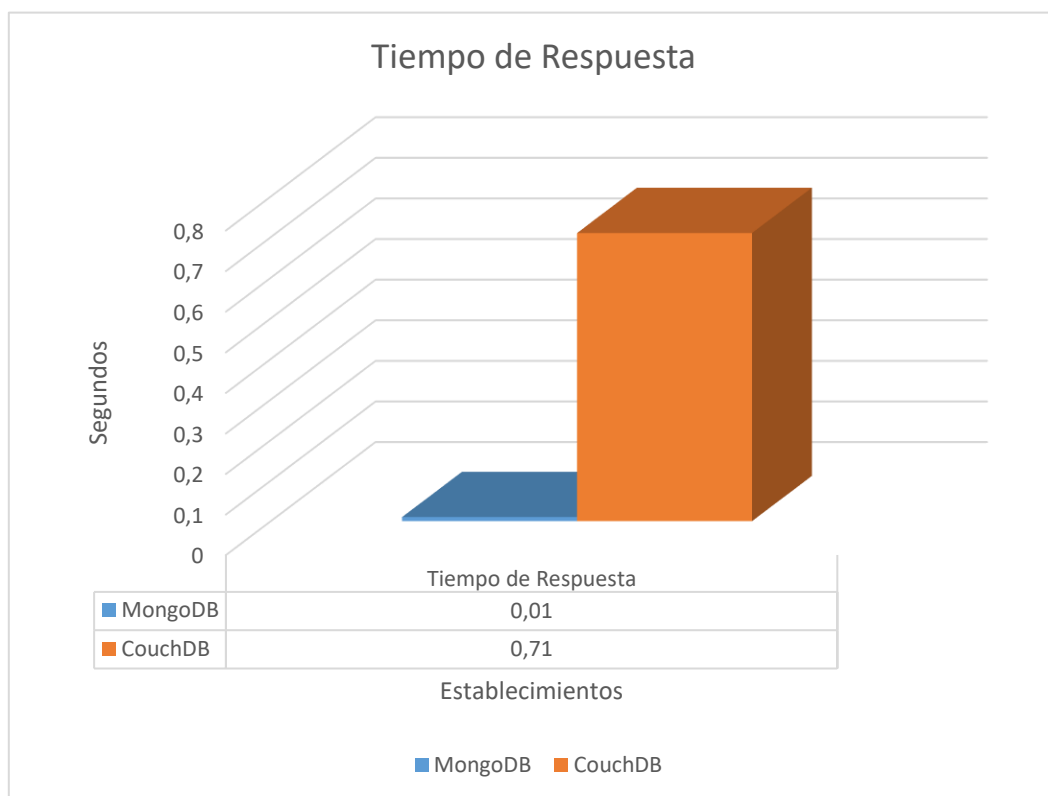
**Tabla 50 - Uso de Tiempo de Respuesta (Actualización 2).**

Gestor de Base de Datos NoSQL	TI	TF	TR
MongoDB	0	0.01	0.01
CouchDB	0	0.71	0.71

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB empleo menos tiempo de respuesta para la actualización de datos 2, con la colección establecimientos; el uso del tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el uso del tiempo de respuesta que se empleó en la actualización de datos 2, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 62 - Gráfico del Uso de Tiempo de Respuesta (Actualización 2).**

### 4. Cantidad de errores.

No se presentó errores en la prueba tres – actualización 2.

### Actualización 3

En la actualización 3, se usó la colección egresoshospitalarios, donde se realizó la actualización de todos los egresos hospitalarios del establecimiento de Salud Hospital Dr. Leonardo Guzmán (Antofagasta).

#### 1. Consumo de Memoria RAM

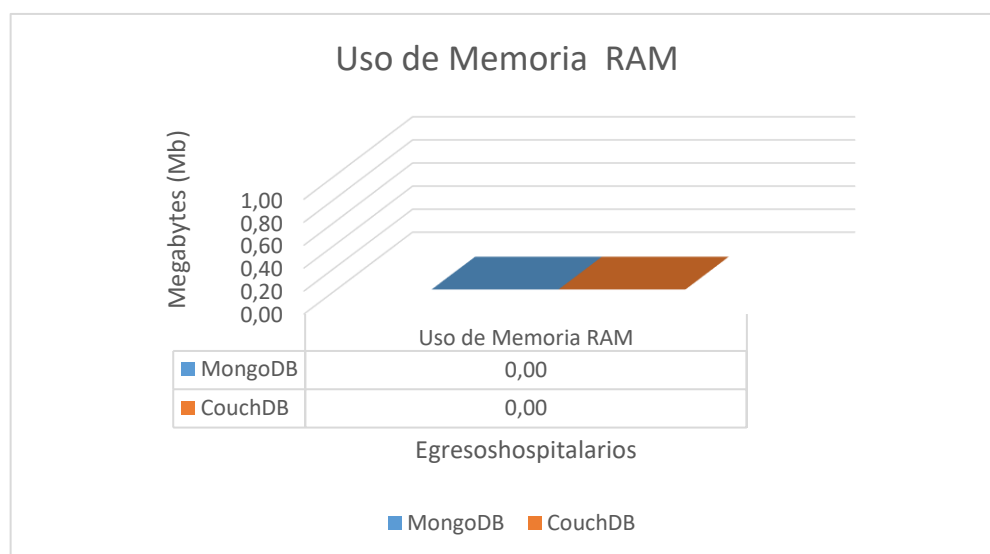
**Tabla 51** - Cantidad de Uso de Memoria RAM (actualización 3)

Gestor de Base de Datos NoSQL	UI	UF	PUR	Uso de Memoria RAM (Mb)
MongoDB	13.0%	13.0%	0.0%	0.00
CouchDB	0.6%	0.6%	0.0%	0.00

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB y CouchDB ambos gestores de base de datos NoSQL documental usaron la misma cantidad de memoria RAM para la actualización de datos 3, con la colección egresoshospitalarios; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el uso de memoria RAM empleado en la actualización de datos 3, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 63** - Gráfico de Cantidad de uso de Memoria RAM (Actualización 3)

## 2. Consumo de CPU

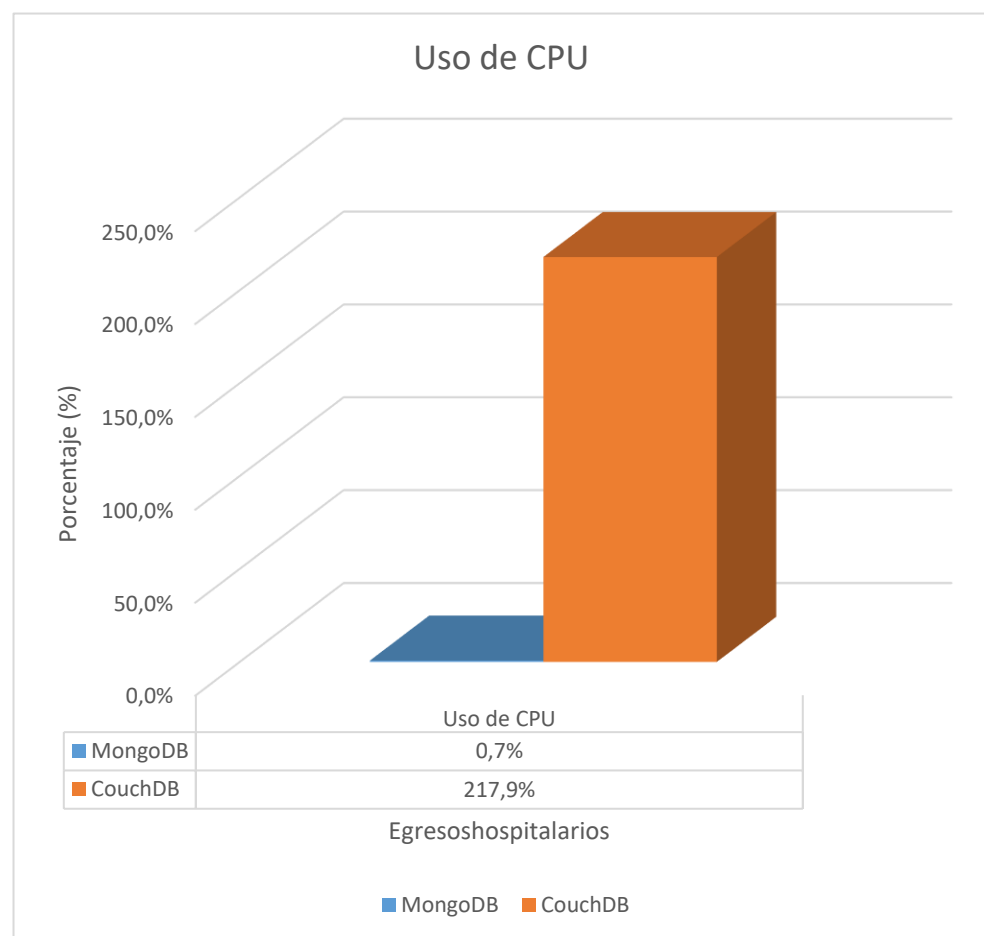
**Tabla 52 - Cantidad de Uso de CPU (Actualización 3).**

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	1.0%	0.7%
CouchDB	0.7%	218.6%	217.9%

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB emplea menos uso de CPU para la actualización de datos 3, con la colección egresoshospitalarios; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el uso de CPU que se empleó en la actualización de datos 3, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 64 - Gráfico de la Cantidad de Uso de CPU (Actualización 3).**



### 3. Tiempo de respuesta

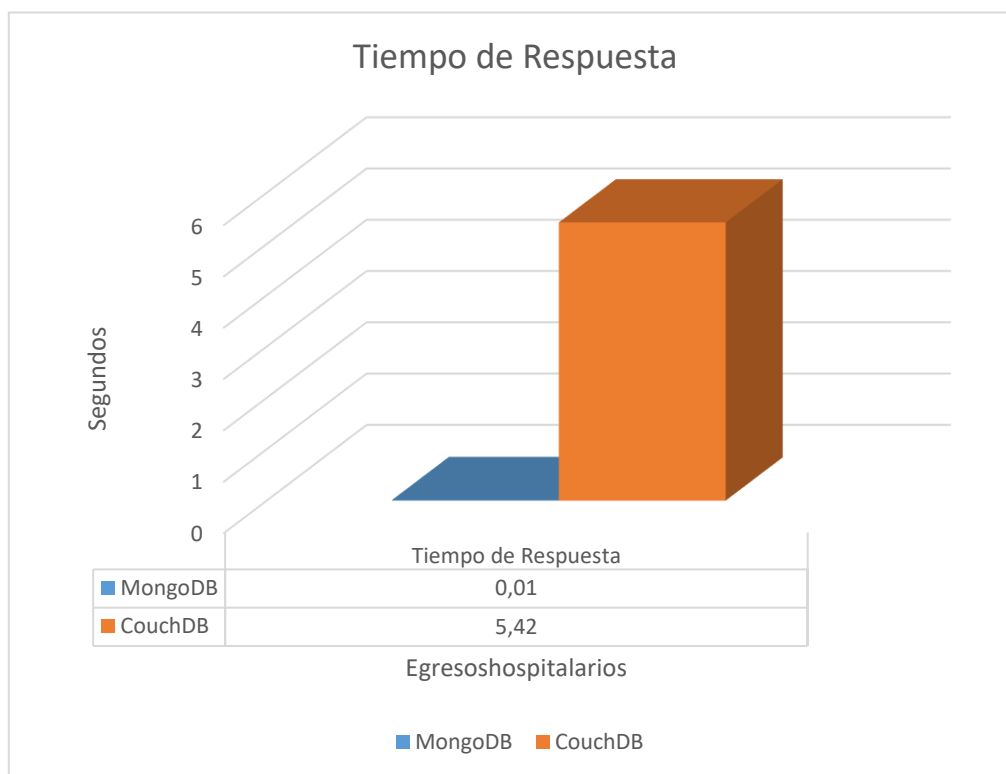
**Tabla 53 - Uso de Tiempo de Respuesta (Actualización 3).**

Gestor de Base de Datos NoSQL	TI	TF	TR
MongoDB	0	0.01	0.01
CouchDB	0	5.42	5.42

**Fuente:** Elaboración Propia.

Según esta tabla, observamos que MongoDB empleó menos tiempo de respuesta para la actualización de datos 3, con la colección egresoshospitalarios; el uso del tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el uso del tiempo de respuesta que se empleó en la actualización de datos 3, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 65 – Gráfico del Uso del Tiempo de Respuesta (Actualización 3).**

### 4. Cantidad de errores.

No se presentó errores en la prueba tres – actualización 3.

Actualización 4

En la actualización 4, se usó la colección egresoshospitalarios, donde se realizó la actualización de todos los egresos hospitalarios del establecimiento de Salud Hospital San Juan de Dios (San Fernando).

1. Consumo de Memoria RAM

Tabla 54 - Cantidad de Uso de Memoria RAM (Actualización 4).

Gestor de Base de Datos NoSQL	UI	UF	PUR	Uso de Memoria RAM (Mb)
MongoDB	13.0%	13.0%	0.0%	0.00
CouchDB	0.6%	0.6%	0.0%	0.00

Fuente: Elaboración Propia.

Según esta tabla, se observó que MongoDB y CouchDB ambos gestores de base de datos NoSQL documental usaron la misma cantidad de memoria RAM para la actualización de datos 4, con la colección egresoshospitalarios; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el uso de memoria RAM empleado en la actualización de datos 4, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental

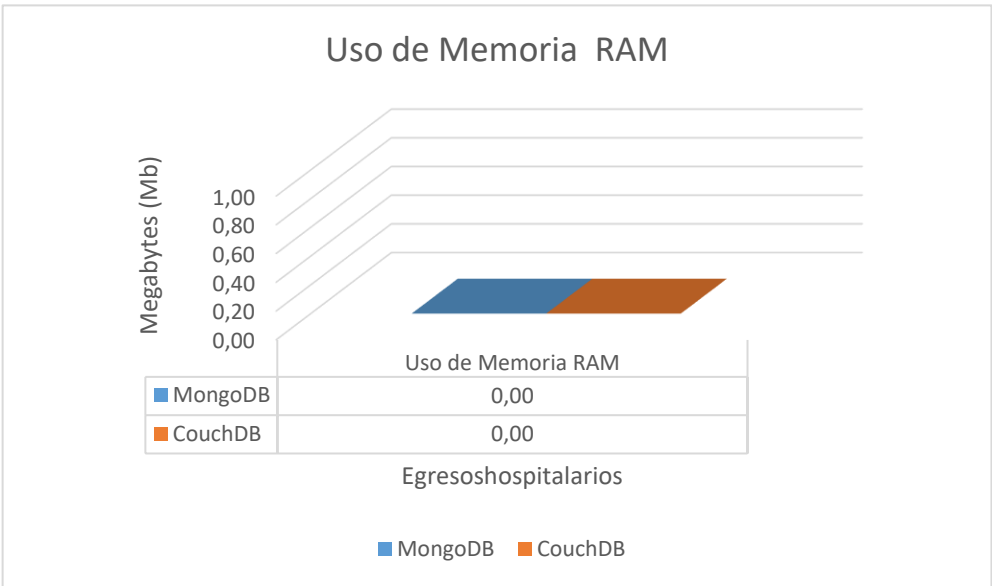


Figura 66 - Gráfico de Cantidad de Uso de Memoria RAM (Actualización 4).

## 2. Consumo de CPU

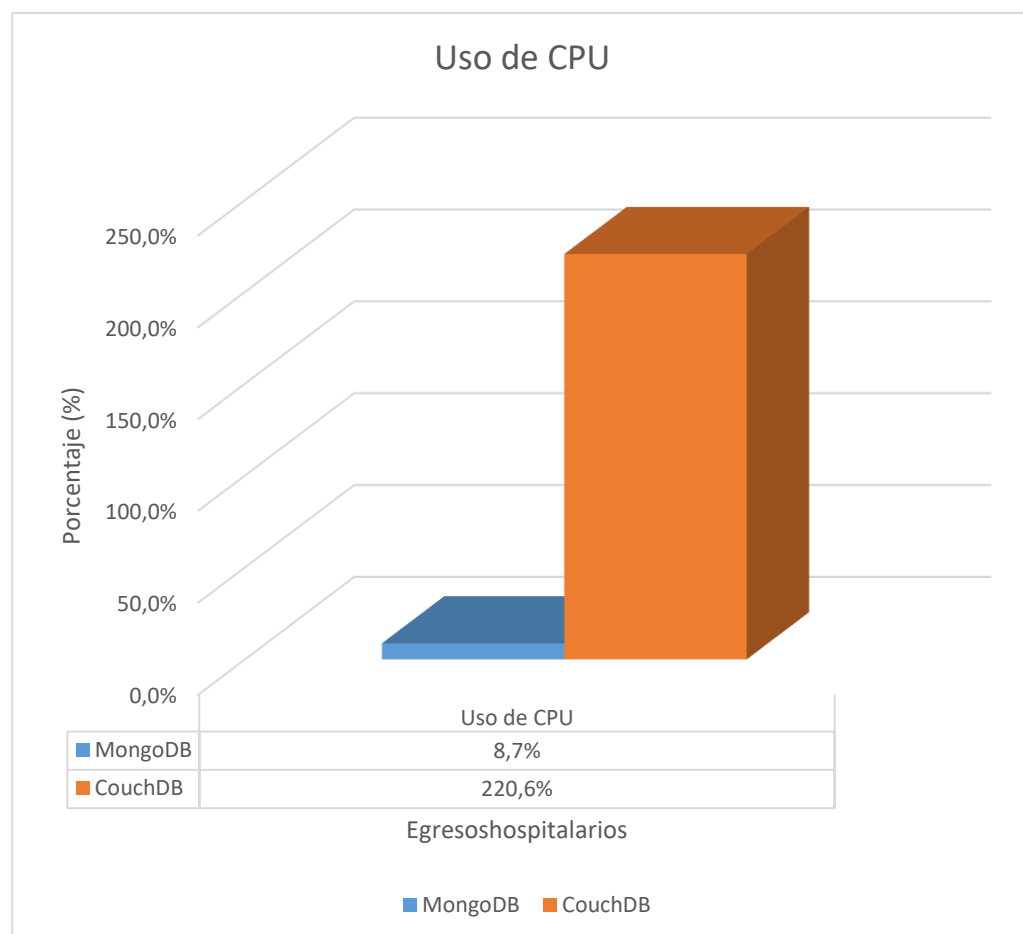
**Tabla 55 - Cantidad de Uso de CPU (Actualización 4).**

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	9.0%	8.7%
CouchDB	1.7%	222.3%	220.6%

**Fuente:** Elaboración Propia.

Según esta tabla, se observó que MongoDB empleó menos uso de CPU para la actualización de datos 4, con la colección egresoshospitalarios; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el uso de CPU que se empleó en la actualización de datos 4, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 67 - Gráfico de la Cantidad de Uso de CPU (Actualización 4).**

### 3. Tiempo de respuesta

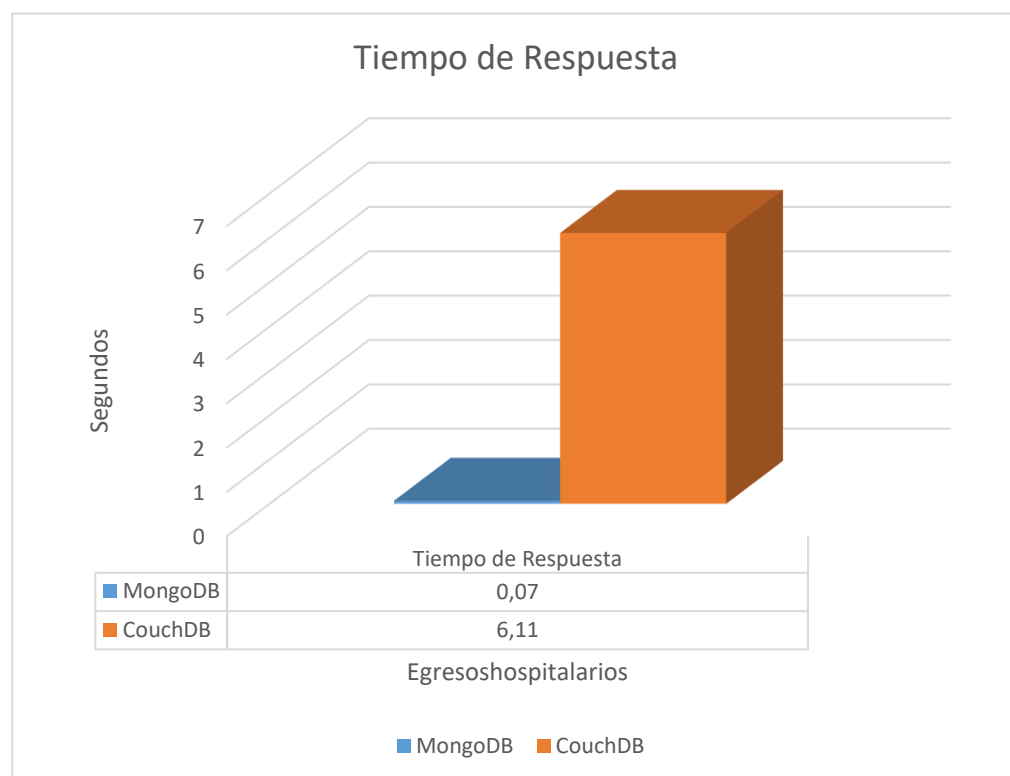
**Tabla 56 - Uso del Tiempo de Respuesta (Actualización 4).**

Gestor de Base de Datos NoSQL	TI	TF	TR
MongoDB	0	0.07	0.07
CouchDB	0	6.11	6.11

**Fuente:** Elaboración Propia.

Según esta tabla, se observó que MongoDB empleó menos tiempo de respuesta para la actualización de datos 4, con la colección egresoshospitalarios; el uso del tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el uso del tiempo de respuesta que se empleó en la actualización de datos 4, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 68 - Gráfico del Uso del Tiempo de Respuesta (Actualización 4).**

### 4. Cantidad de errores.

No se presentó errores en la prueba tres – actualización 4.

**D) Prueba Cuatro:** En la Prueba cuatro, se realizó cuatro operaciones de eliminar datos para ambos gestores de base de datos NoSQL documental con las colecciones establecimientos y egresoshospitalarios; donde se evaluaron los siguientes indicadores: Uso de Memoria RAM, uso de CPU, Tiempo de Respuesta y Cantidad de errores. Asimismo, se mostró los valores que arrojó la prueba cuatro.

### Eliminación 1

En la Eliminación 1, se usó la colección establecimientos, donde se realizó la eliminación de todos los establecimientos de la Comuna Curaco de Vélez.

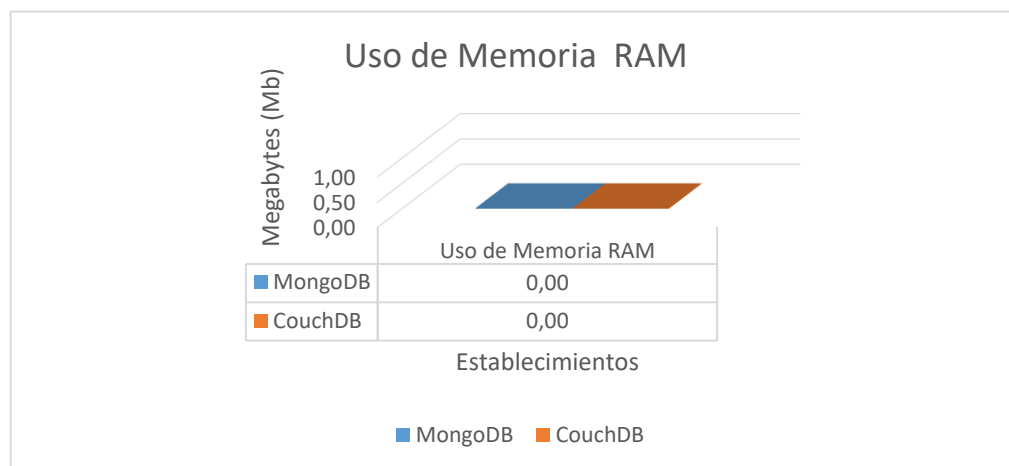
#### 1. Consumo de Memoria RAM

**Tabla 57 - Cantidad de Uso de Memoria RAM (Eliminación 1).**

Gestor de Base de Datos NoSQL	UI	UF	PUR	Uso de Memoria RAM (Mb)
MongoDB	13.0%	13.0%	0.0%	0.00
CouchDB	0.6%	0.6%	0.0%	0.00

**Fuente:** Elaboración Propia.

Según esta tabla, se observó que MongoDB y CouchDB ambos gestores de base de datos NoSQL documental usaron la misma cantidad de memoria RAM para la eliminación de datos 1, con la colección establecimientos; el uso de memoria RAM se mostró en Megabytes. En la siguiente figura se mostró el uso de memoria RAM empleado en la eliminación de datos 1, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 69 - Gráfico de Cantidad de Uso de Memoria RAM (Eliminación 1).**

## 2. Consumo de CPU

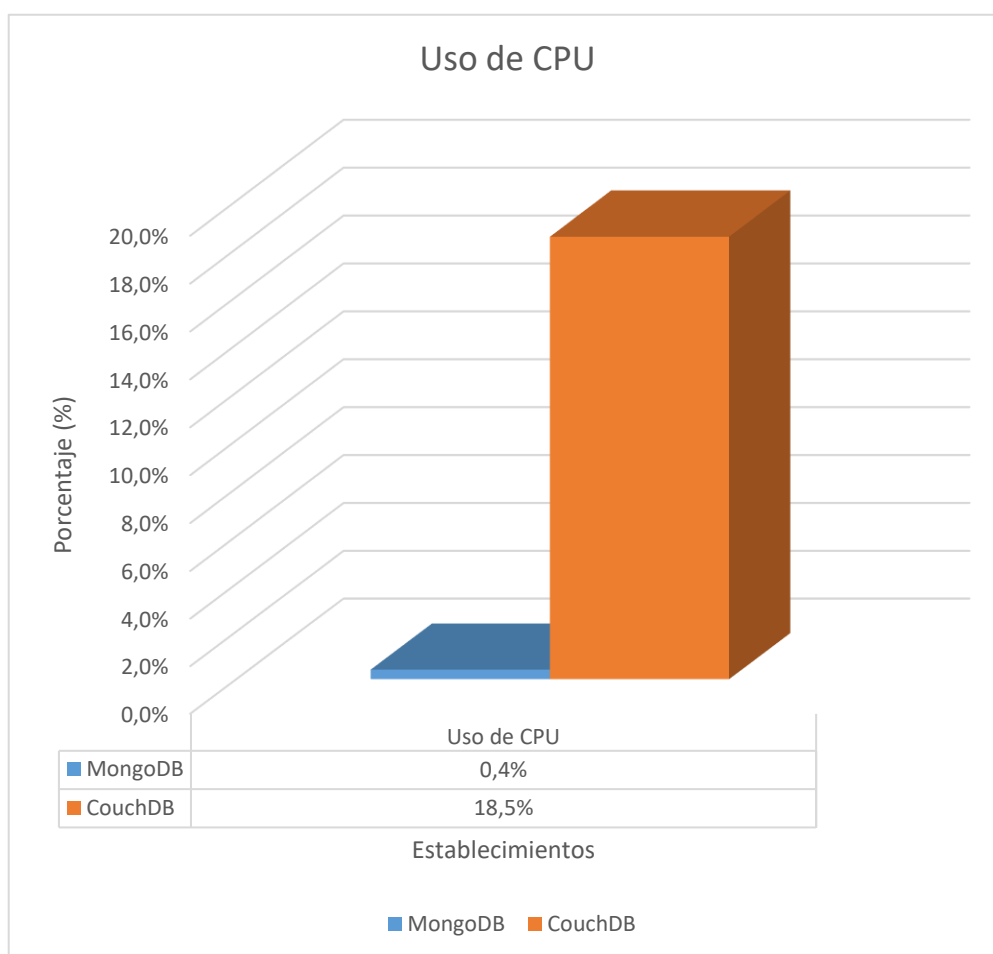
**Tabla 58 - Cantidad de Uso de CPU (Eliminación 1).**

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	0.7%	0.4%
CouchDB	3.6%	22.1%	18.5%

**Fuente:** Elaboración Propia.

Según esta tabla, se observó que MongoDB empleó menos uso de CPU para la eliminación de datos 1, con la colección establecimientos; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el uso de CPU que se empleó en la actualización de datos 1, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 70 - Gráfico de la Cantidad de Uso de CPU (Eliminación 1).**

### 3. Tiempo de Respuesta

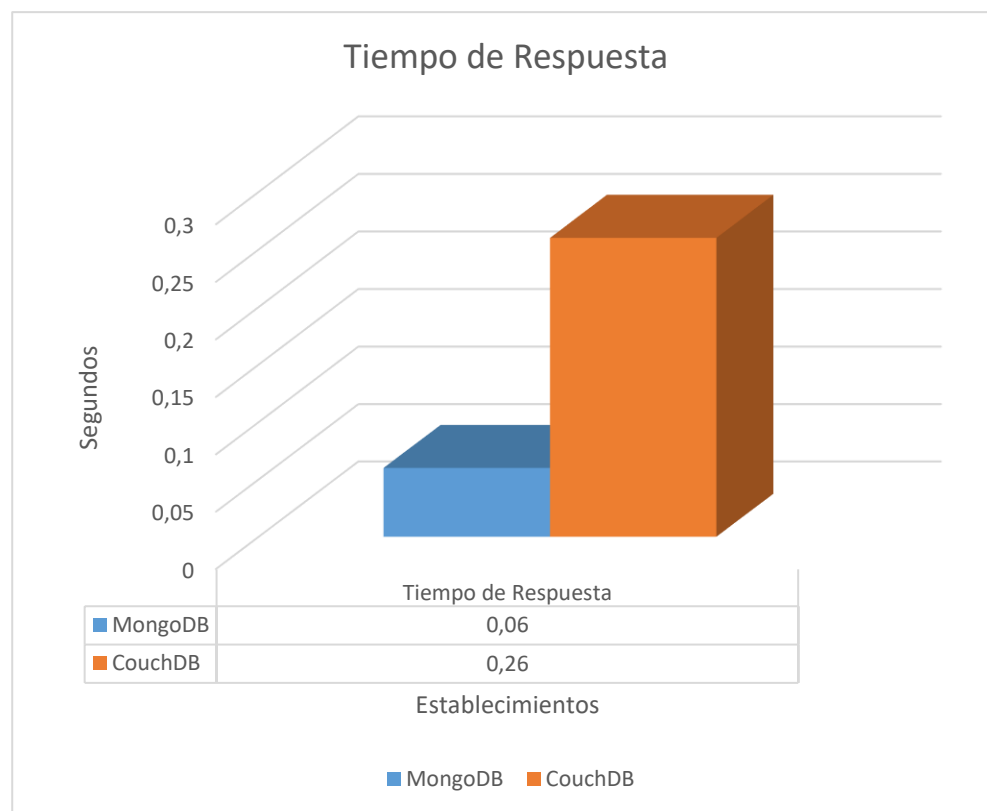
**Tabla 59 - Uso de Tiempo de Respuesta (Eliminación 1).**

Gestor de Base de Datos NoSQL	TI	TF	TR
MongoDB	0	0.06	0.06
CouchDB	0	0.26	0.26

**Fuente:** *Elaboración Propia.*

Según esta tabla, se observó que MongoDB empleó menos tiempo de respuesta para la eliminación de datos 1, con la colección establecimientos; el uso del tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el uso del tiempo de respuesta que se empleó en la eliminación de datos 1, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 71 - Gráfico del Uso de Tiempo de Respuesta (Eliminación 1).**

### 4. Cantidad de Errores

No se presentó errores en la prueba cuatro – Eliminación 1.

## Eliminación 2

En la Eliminación 2, se usó la colección establecimientos, donde se realizó la eliminación de todos los establecimientos de la Región de Magallanes y de la Antártica Chilena.

### 1. Consumo de Memoria RAM

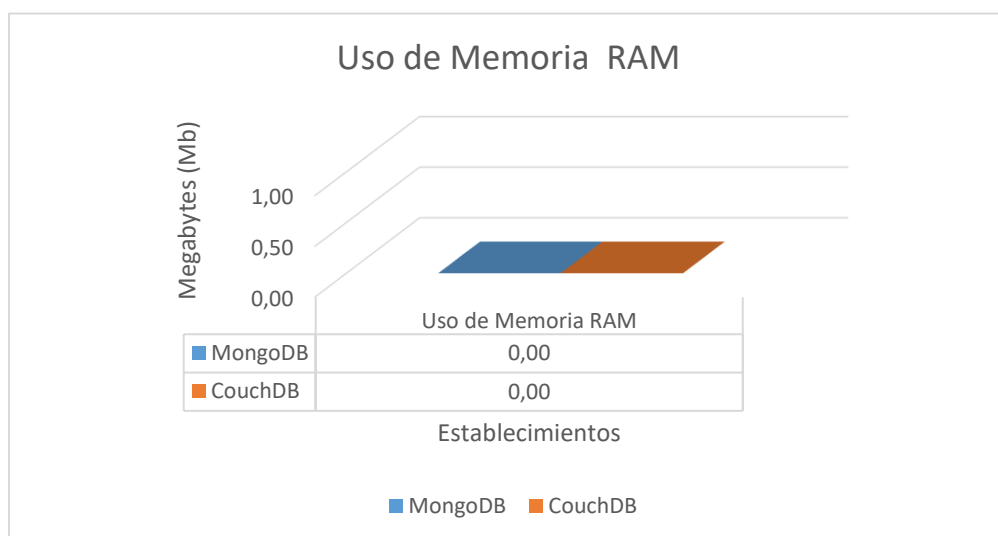
**Tabla 60** - Cantidad de Uso de Memoria RAM (Eliminación 2).

Gestor de Base de Datos NoSQL	UI	UF	PUR	Uso de Memoria RAM (Mb)
MongoDB	13.0%	13.0%	0.0%	0.00
CouchDB	0.6%	0.6%	0.0%	0.00

**Fuente:** Elaboración Propia.

Según esta tabla, se observó que MongoDB y CouchDB ambos gestores de base de datos NoSQL documental usaron la misma cantidad de memoria RAM para la eliminación de datos 2, con la colección establecimientos; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el uso de memoria RAM empleado en la eliminación de datos 2, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 72** - Gráfico de la Cantidad de Uso de Memoria RAM (Eliminación 2).



## 2. Consumo de CPU

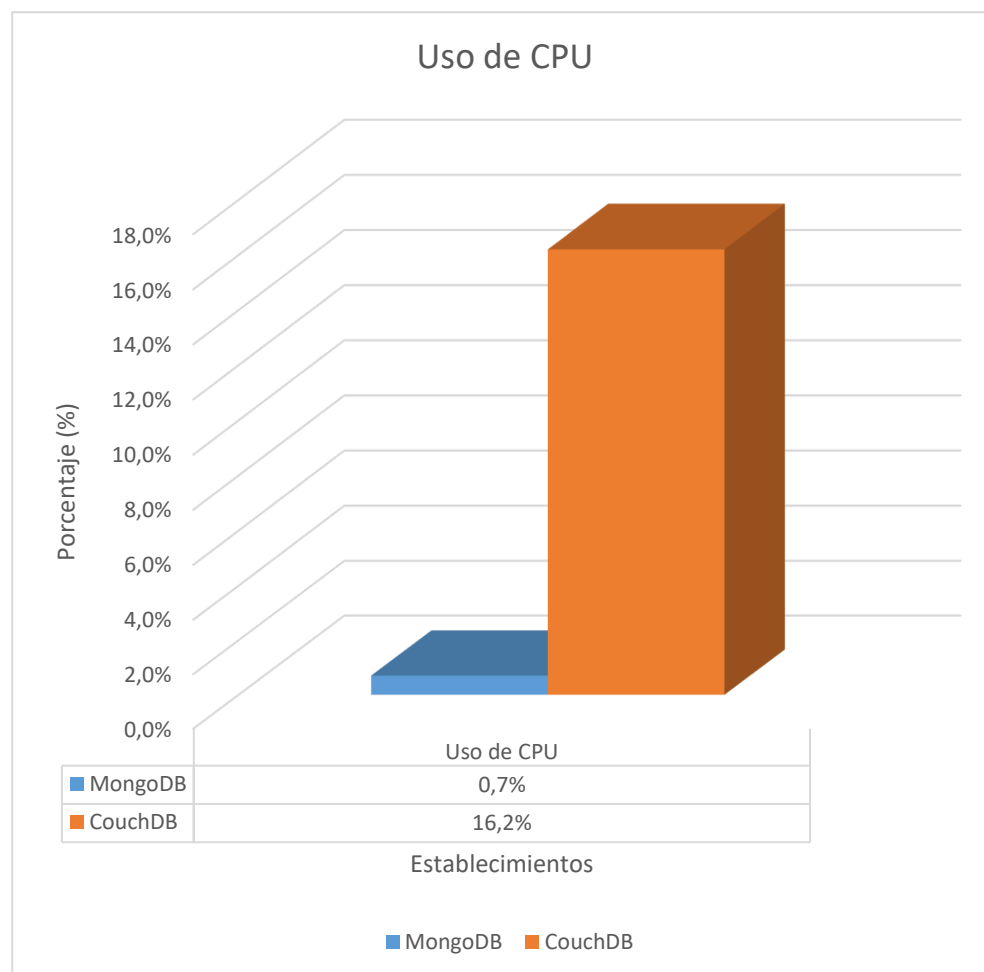
**Tabla 61** - Cantidad de Uso de CPU (Eliminación 2).

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	1.0%	0.7%
CouchDB	4.0%	20.2%	16.2%

**Fuente:** Elaboración Propia.

Según esta tabla, se observó que MongoDB empleó menos uso de CPU para la eliminación de datos 2, con la colección establecimientos; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el uso de CPU que se empleó en la actualización de datos 2, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 73** - Gráfico de la Cantidad de Uso de CPU (Eliminación 2).

### 3. Tiempo de Respuesta

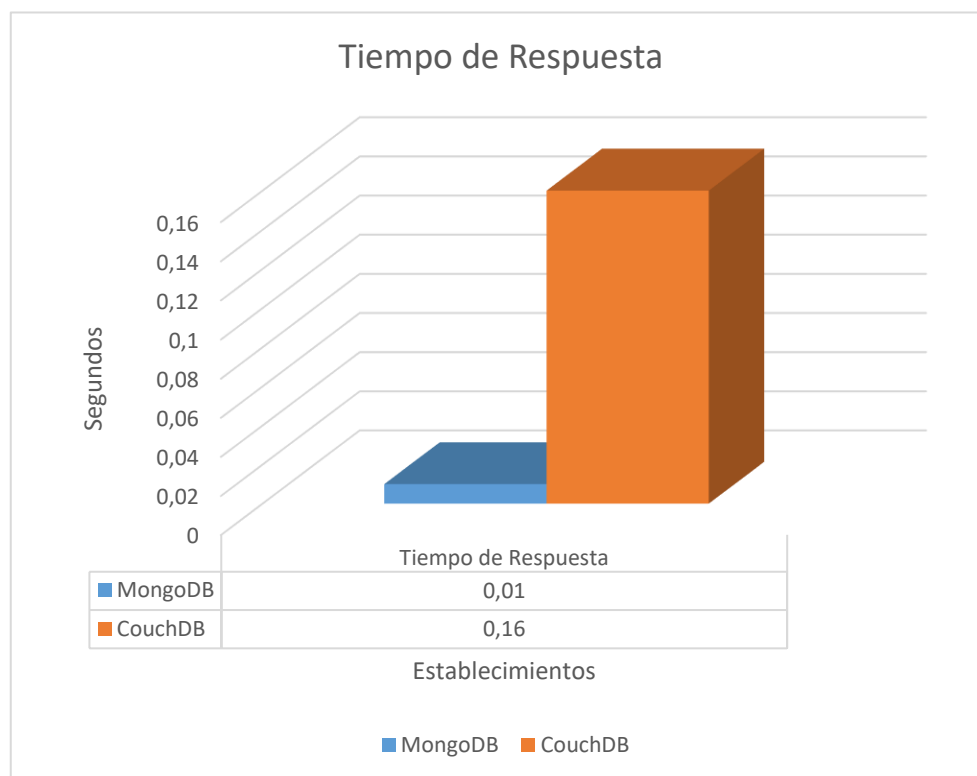
**Tabla 62 - Uso de Tiempo de Respuesta (Eliminación 2).**

Gestor de Base de Datos NoSQL	TI	TF	TR
MongoDB	0	0.01	0.01
CouchDB	0	0.16	0.16

**Fuente:** Elaboración Propia.

Según esta tabla, se observó que MongoDB empleó menos tiempo de respuesta para la eliminación de datos 2, con la colección establecimientos; el uso del tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el uso del tiempo de respuesta que se empleó en la eliminación de datos 2, donde se usó la colección establecimientos en ambos gestores de base de datos NoSQL Documental.



**Figura 74 - Gráfico del Uso de Tiempo de Respuesta (Eliminación 2).**

### 4. Cantidad de Errores

No se presentó errores en la prueba cuatro – Eliminación 2.

Eliminación 3

En la Eliminación 3, se usó la colección egresoshospitalarios, donde se realizó la eliminación de todos los egresos hospitalarios de la Comuna Juan Fernández.

1. Consumo de Memoria RAM

Tabla 63 - Cantidad de Uso de Memoria RAM (Eliminación 3)

Gestor de Base de Datos	UI	UF	PUR	Uso de Memoria RAM (Mb)
NoSQL				
MongoDB	13.0%	13.0%	0.0%	0.00
CouchDB	0.6%	0.6%	0.0%	0.00

Fuente: Elaboración Propia.

Según esta tabla, se observó que MongoDB y CouchDB ambos gestores de base de datos NoSQL documental usaron la misma cantidad de memoria RAM para la eliminación de datos 3, con la colección egresoshospitalarios; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el uso de memoria RAM empleado en la eliminación de datos 3, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.

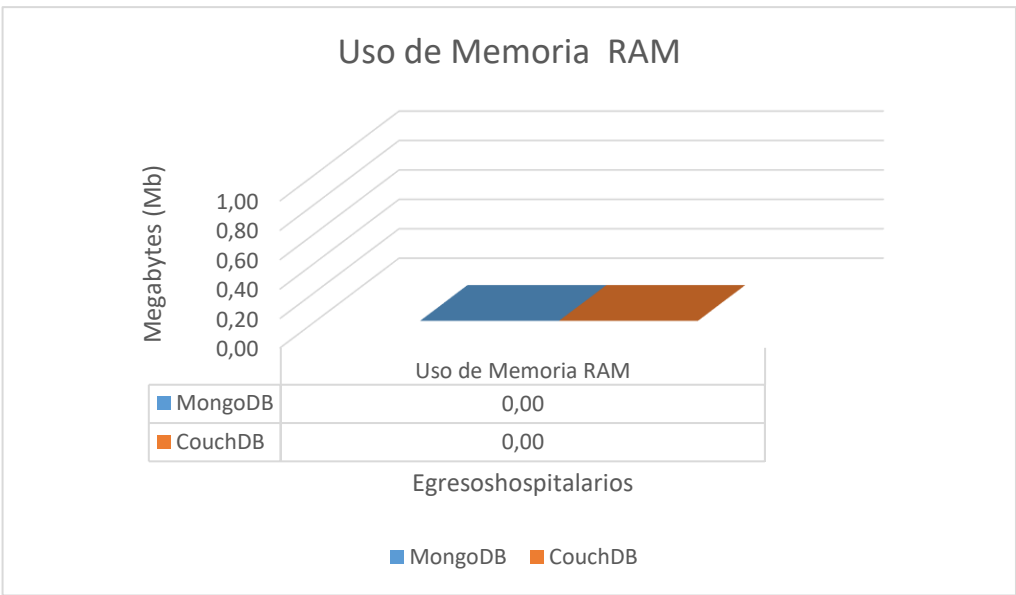


Figura 75 - Gráfico de la Cantidad de Uso de Memoria RAM (Eliminación 3).

## 2. Consumo de CPU

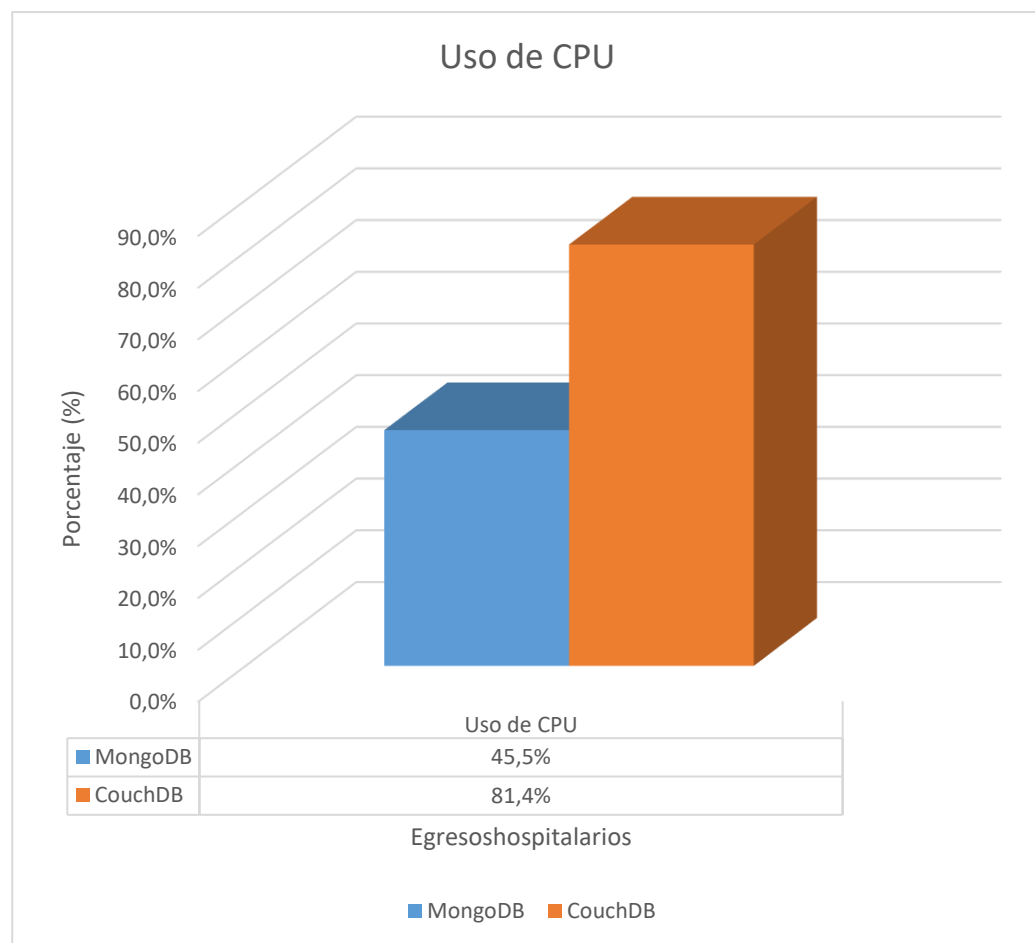
**Tabla 64** - Cantidad de Uso de CPU (Eliminación 3).

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	45.8%	45.5%
CouchDB	3.6%	85.0%	81.4%

**Fuente:** Elaboración Propia.

Según esta tabla, se observó que MongoDB empleó menos uso de CPU para la eliminación de datos 3, con la colección egresoshospitalarios; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el uso de CPU que se empleó en la actualización de datos 3, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 76** - Gráfico de la Cantidad de Uso de CPU (Eliminación 3).

### 3. Tiempo de Respuesta

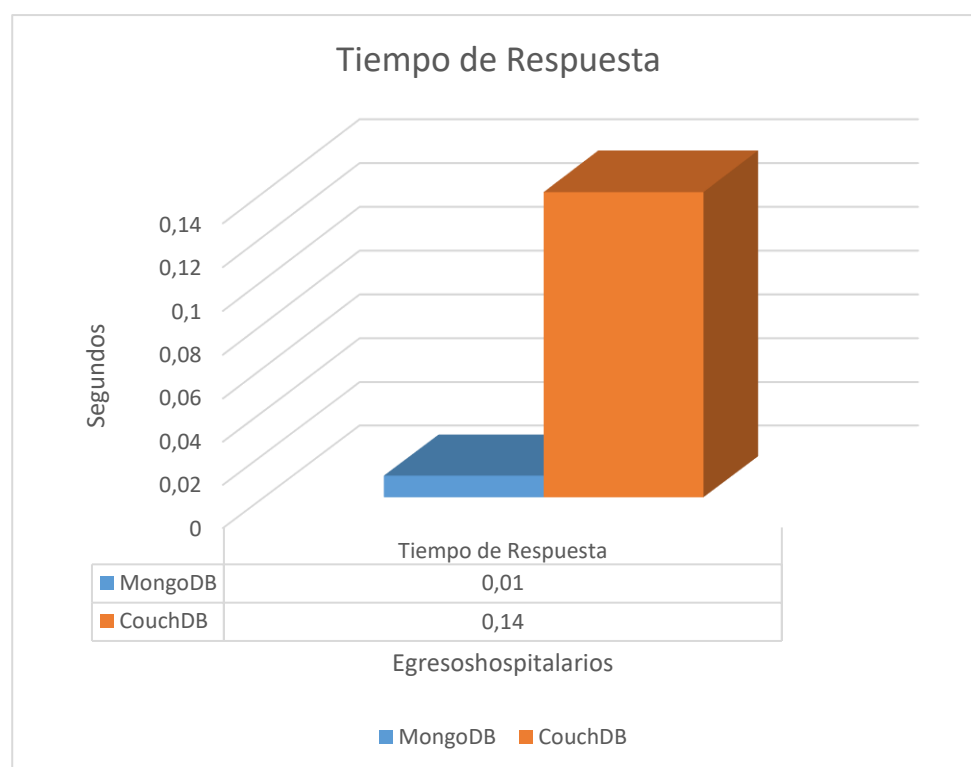
**Tabla 65** - Uso de Tiempo de Respuesta (Eliminación 3).

Gestor de Base de Datos NoSQL	TI	TF	TR
MongoDB	0	0.01	0.01
CouchDB	0	0.14	0.14

**Fuente:** Elaboración Propia.

Según esta tabla, se observó que MongoDB empleó menos tiempo de respuesta para la eliminación de datos 3, con la colección egresoshospitalarios; el uso del tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el uso del tiempo de respuesta que se empleó en la eliminación de datos 3, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 77** - Gráfico del Uso de Tiempo de Respuesta (Eliminación 3).

### 4. Cantidad de Errores

No se presentó errores en la prueba cuatro – Eliminación 3.

## Eliminación 4

En la Eliminación 4, se usó la colección egresoshospitalarios, donde se realizó la eliminación de todos los egresos hospitalarios del Centro de Salud Mutual CChc Copiapó.

### 1. Consumo de Memoria RAM

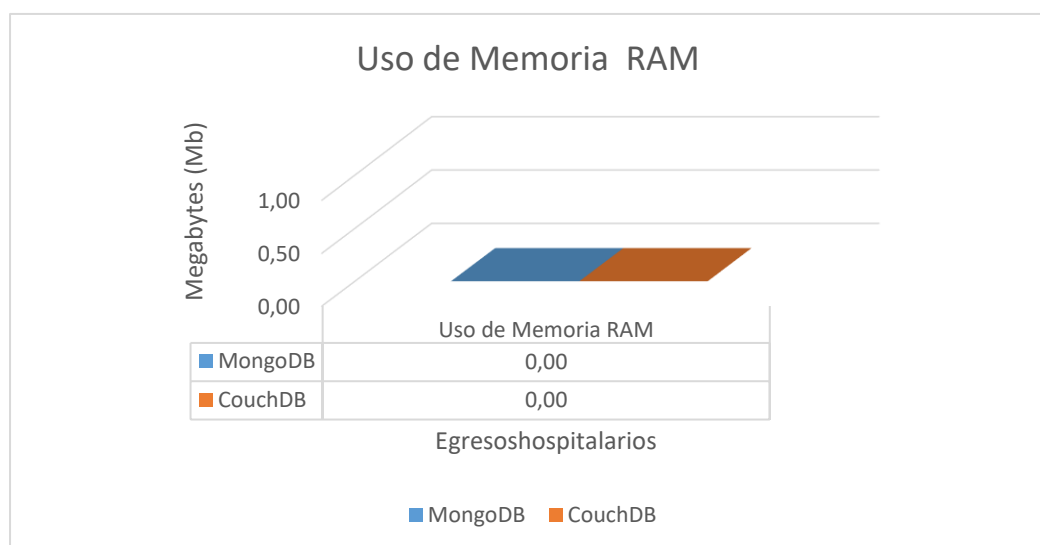
**Tabla 66** - Cantidad de Uso de Memoria RAM (Eliminación 4).

Gestor de Base de Datos NoSQL	UI	UF	PUR	Uso de Memoria RAM (Mb)
MongoDB	13.0%	13.0%	0.0%	0.00
CouchDB	0.6%	0.6%	0.0%	0.00

**Fuente:** Elaboración Propia.

Según esta tabla, se observó que MongoDB y CouchDB ambos gestores de base de datos NoSQL documental usaron la misma cantidad de memoria RAM para la eliminación de datos 4, con la colección egresoshospitalarios; el uso de memoria RAM se mostró en Megabytes.

En la siguiente figura se mostró el uso de memoria RAM empleado en la eliminación de datos 3, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 78** - Gráfico de la Cantidad de Uso de Memoria RAM (Eliminación 4).

## 2. Consumo de CPU

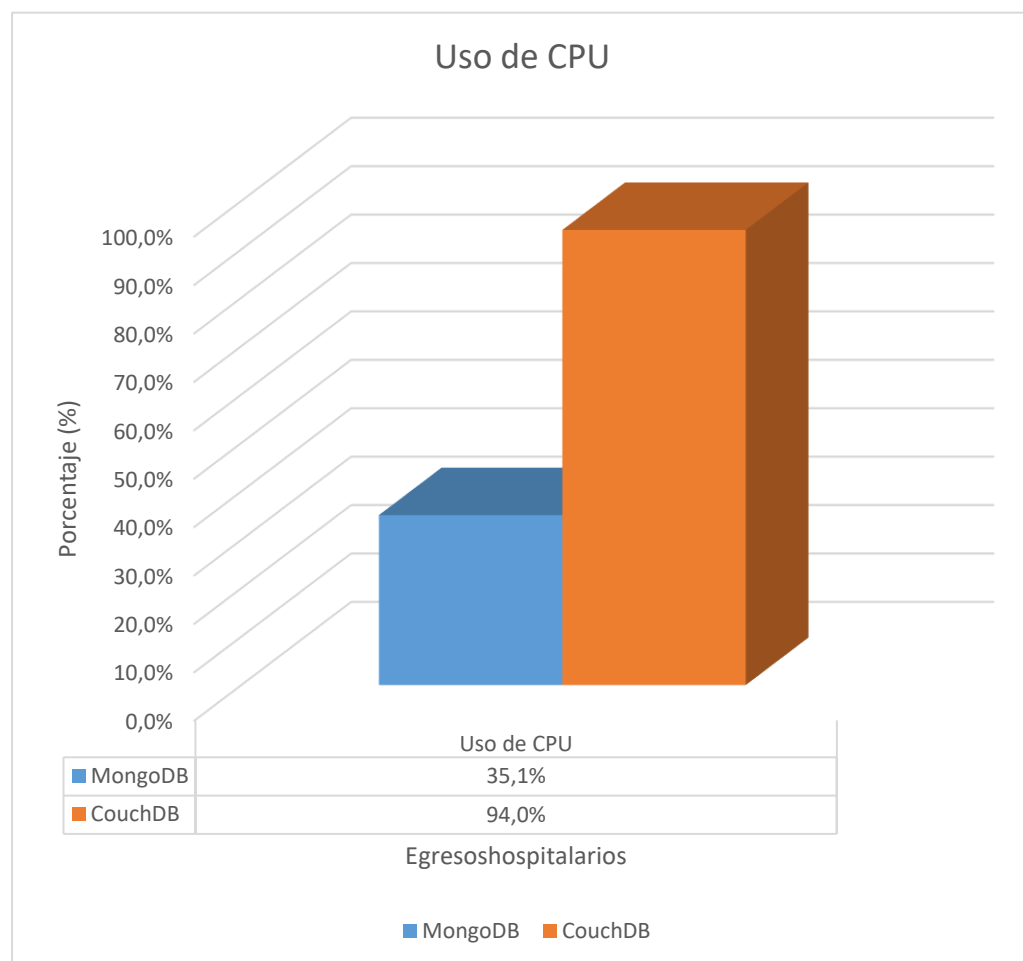
**Tabla 67** - Cantidad de Uso de CPU (Eliminación 4).

Gestor de Base de Datos NoSQL	UI	UF	PUC
MongoDB	0.3%	35.4%	35.1%
CouchDB	2.0%	96.0%	94.0%

**Fuente:** Elaboración Propia.

Según esta tabla, se observó que MongoDB empleó menos uso de CPU para la eliminación de datos 4, con la colección egresoshospitalarios; el uso de CPU se mostró en porcentaje.

En la siguiente figura se mostró el uso de CPU que se empleó en la actualización de datos 4, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 79** - Gráfico de la Cantidad de Uso de CPU (Eliminación 4).

### 3. Tiempo de Respuesta

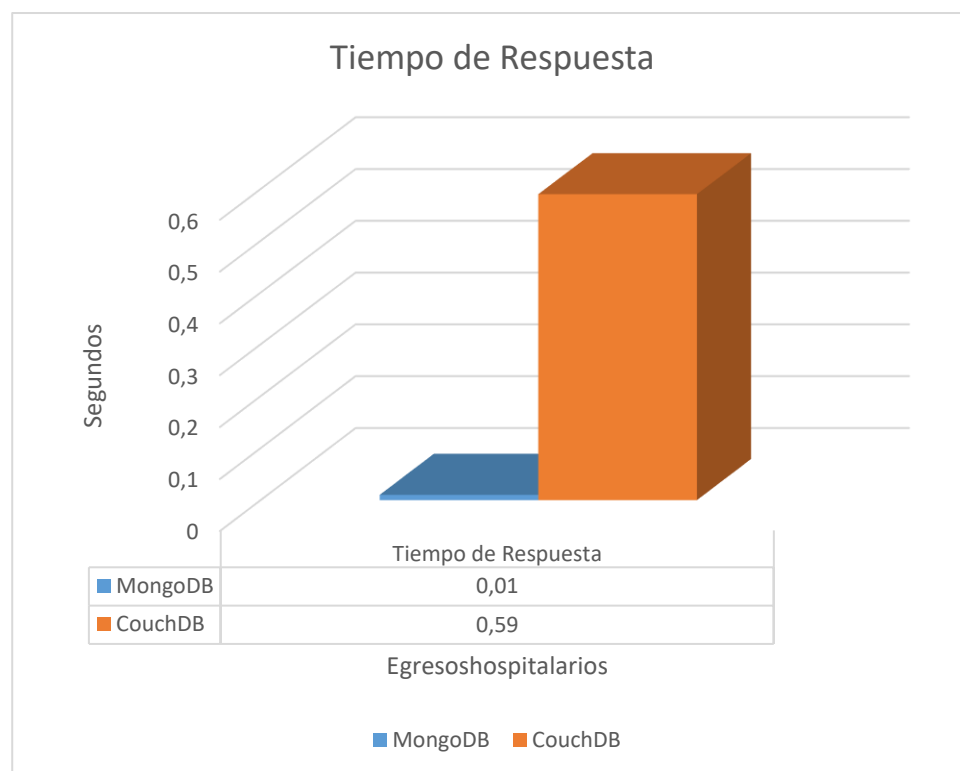
**Tabla 68 - Uso de Tiempo de Respuesta (Eliminación 4).**

Gestor de Base de Datos NoSQL	TI	TF	TR
MongoDB	0	0.01	0.01
CouchDB	0	0.59	0.59

**Fuente:** Elaboración Propia.

Según esta tabla, se observó que MongoDB empleó menos tiempo de respuesta para la eliminación de datos 4, con la colección egresoshospitalarios; el uso del tiempo de respuesta se mostró en segundos.

En la siguiente figura se mostró el uso del tiempo de respuesta que se empleó en la eliminación de datos 4, donde se usó la colección egresoshospitalarios en ambos gestores de base de datos NoSQL Documental.



**Figura 80 - Gráfico del Uso de Tiempo de Respuesta (Eliminación 4).**

### 4. Cantidad de Errores

No se presentó errores en la prueba cuatro – Eliminación 4.



Finalmente se interpretó los resultados en tablas y figuras; los datos anotados en la ficha de registro de datos, fueron interpretados utilizando la herramienta Microsoft Excel 2019; donde se obtuvo los resúmenes, promedios finales los datos y se convirtió los datos de kilobytes a Megabytes del uso de memoria RAM y Espacio usado en Disco. Asimismo, se determinó en los resultados que CouchDB proporciona mejor desempeño en Consumo de Memoria RAM cuando se realizan pruebas de grandes conjuntos de datos y utilizó menos consumo de espacio en disco cuando se realiza la operación de volcado de datos ya se de grandes o pequeños conjuntos de datos. MongoDB por otro lado proporcionó mejor rendimiento en el Consumo de CPU y Tiempo de Respuesta cuando se realizan las operaciones de grandes conjuntos de datos; asimismo cuando se trata de pequeños conjuntos de datos MongoDB muestra resultados óptimos en cuanto al uso de Memoria RAM, Uso de CPU y Tiempo de Respuesta; resultando utilizar menos consumo de estos indicadores que el gestor de base de datos NoSQL CouchDB.

## **IV. CONCLUSIONES Y RECOMENDACIONES**

### **4.1. Conclusiones**

- a) Luego de un análisis comparativo, se confirmó que MongoDB y CouchDB poseen características similares como tipo de datos, modelo de datos, plataforma y tipo de licencia para ser utilizados en el rendimiento de procesamiento de grandes conjuntos de datos.
- b) Se construyó un ambiente para pruebas de software libre porque nos permitió reducir costos en la implementación del servidor y porque nos ofreció mejor soporte y compatibilidad con el hardware del servidor.
- c) En las pruebas de rendimiento se realizó cuatro operaciones: Volcado, listado, actualización y eliminación; las cuales nos permitió determinar que gestor de base de datos NoSQL Documental seleccionado es más eficiente en cuanto al uso de Memoria RAM, uso de CPU, Tiempo de Respuesta, Espacio usado en disco y cantidad de errores.
- d) Luego de realizar la interpretación de resultados, se determinó que CouchDB proporcionó mejor desempeño en el uso de Memoria RAM cuando se realizan pruebas de grandes conjuntos de datos y utilizó menos consumo de espacio en disco cuando se realiza la operación de volcado de datos ya se de grandes o pequeños conjuntos de datos. MongoDB por otro lado proporcionó mejor rendimiento en el Consumo de CPU y Tiempo de Respuesta cuando se realizan las operaciones de grandes conjuntos de datos; asimismo cuando se trata de pequeños conjuntos de datos MongoDB muestra resultados óptimos en cuanto al uso de Memoria RAM, Uso de CPU y Tiempo de Respuesta; resultando utilizar menos consumo de estos indicadores que el gestor de base de datos NoSQL CouchDB.

## **4.2. Recomendaciones**

- a) Se recomienda utilizar otros gestores de base de datos NoSQL documental para realizar las pruebas de rendimiento de grandes conjuntos de datos con diferentes consultas.
- b) Se recomienda utilizar otro lenguaje de programación para la evaluación del tiempo de respuesta de los gestores de base de datos NoSQL documental.
- c) Se recomienda utilizar pruebas de rendimiento con otros tipos de datos.
- d) Se recomienda ante cualquier consulta o duda revisar las fuentes documentales de las páginas oficiales de las bases de datos NoSQL para guiarte.
- e) Se recomienda realizar pruebas utilizando diferentes volúmenes de datos.

## REFERENCIAS

- acens, B. (28 de Febrero de 2014). *Bases de datos NoSQL. Qué son y tipos que nos podemos*. Obtenido de <https://www.acens.com/wp-content/images/2014/02/bbdd-nosql-wp-acens.pdf>
- Anderson, J. C., & Slater, J. L. (2010). *CouchDB: The Definitive Guide*. San Francisco, California: Repkover.
- AWS, B. (2020). *¿Qué es una base de datos clave-valor?* Obtenido de <https://aws.amazon.com/es/nosql/key-value/>
- Busto, H. G., & Enríquez, O. Y. (2012). Bases de datos NoSQL. *Telemática*, 13.
- Cabezuelo, A. S. (2016). *Introducción a las bases de datos NoSQL usando MongoDB*. Editorial UOC.
- Castro Romero, A., González Sanabria, J. S., & Callejas Cuervo, M. (2012). *Utilidad y funcionamiento de las bases de datos NoSQL*. Boyacá - Colombia.
- Chopade, M. R., & Dhavase, M. N. (2017). MongoDB y CouchBase Comparación de rendimiento para el conjunto de datos de imágenes . pune, india.
- Daniel Pereira, P. O. (2015). *Data warehouses in MongoDB vs SQL Server: A comparative analysis of the querie performance. 10th Iberian Conference on Information Systems and Technologies (CISTI)*. Aveiro, Portugal.
- Fernández, R. (30 de Mayo de 2017). *NoSQL: clasificación de las bases de datos según el teorema*. Obtenido de <https://www.genbeta.com/desarrollo/nosql-clasificacion-de-las-bases-de-datos-segun-el-teorema-cap>
- Folwer, A. (2015). *Nosql For Dummies*. Canadá.
- Gómez, J. M. (2013). *Base de datos relacionales y modelado de datos*. Parainfo.
- Gupta, S., & Rani, R. (2016). Un estudio comparativo de las bases de datos orientadas a documentos CouchDB y Elasticsearch . *A Comparative Study of Elasticsearch and CouchDB Document Oriented Databases*, (pág. 4). India.
- Guy, H. (2015). *BBases de datos de la próxima generación*. Apress.
- Gyrödi, C., Gyrödi, R., Pecherle, G., & Olah, A. (2015). Un estudio comparativo: MongoDB vs MySQL. Rumania.
- IBM. (2019). Obtenido de IBM: [https://www.ibm.com/support/knowledgecenter/es/SSEP GG\\_10.5.0/com.ibm.swg.im.iis.fed.tuning.doc/topics/iifytbulkinsntz.html](https://www.ibm.com/support/knowledgecenter/es/SSEP GG_10.5.0/com.ibm.swg.im.iis.fed.tuning.doc/topics/iifytbulkinsntz.html)
- información, T. (2018). *Bases de datos documental*. Obtenido de <https://www.tecnologias-informacion.com/documentosbd.html>

- Jung, M.-G., Youn, S.-A., Bae, J., & Choi, Y.-L. (2016). Rendimiento de entrada y salida en el gran entorno de datos. *8th International Conference on Database Theory and Application (DTA)*. Korea del Sur.
- MacCreary, D., & Ann, K. (2013). *Making Sense of NoSQL: A guide for managers and the rest of us*. Manning Publications.
- Morejón Abril, M. d. (2018). *La información en bases de datos NoSQL y su incidencia en la generación documental de la secretaría general del honorable consejo universitario*. Ambato, Ecuador.
- Murazza, M. R., & Nurwidyantoro, A. (2017). Cassandra and SQL database comparison for near real-time Twitter data warehouse. *International Seminar on Intelligent Technology and Its Applications (ISITIA)*. Lombok, Indonesia.
- Neto, A. D., Fernandes, E. P., Holanda, M., Ribeiro, E., & Carvalho, R. N. (2017). ANÁLISIS DE RENDIMIENTO DE MONGODB - UN ESTUDIO COMPARATIVO ENTRE DESPLIEGUES DE CLÚSTERES INDEPENDIENTES Y FRAGMENTADOS. Brasilia, Brasil.
- NORMA ISO, 2. (2020). *ISO 25000*. Obtenido de ISO 25000: <https://iso25000.com/index.php/normas-iso-25000>
- Pereira, D., Oliveira, P., & Rodrigues, F. (2015). Análisis comparativo de rendimiento de consultas. *10th Iberian Conference on Information Systems and Technologies (CISTI)*. Portugal.
- S. K., & Srividya, M. (2017). *Una comparación de rendimiento de bases de datos NoSQL orientadas a documentos*. Tamil Nadu, India.
- Sarasa, A. (2016). *Introducción a las bases de datos NoSQL usando MongoDB*. Barcelona: UOC.
- Silberschatz, A., Korth, H., & Sudarhan, f. (2006). *Fundamentos de Base de Datos*. Madrid: McGraw-Hill/Interamericana.
- SolidQ. (5 de Setiembre de 2018). Obtenido de SolidQ: <https://blogs.solidq.com/es/sql-server/la-insercion-masiva-de-datos-en-sql-server-tiene-sus-trucos-que-te-los-contamos-aqui/>
- Space, C. F. (01 de Diciembre de 2017). *Future Bites*. Obtenido de <https://bites.futurespace.es/teorema-cap-2/>
- Sullivan, D. (2015). *NoSQL for mere mortals*. Addison-Wesley Profesionals,.
- Truic, C.-O., Rădulescu, F., Boicea, A., & Bucur, I. (2015). Evaluación del rendimiento de las operaciones CRUD en una base de datos asincrónica orientada a documentos. Bucharest, Romania.

## ANEXOS

### Anexo 1: Resolución de Aprobación de Proyecto



#### FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO RESOLUCIÓN N°0018-2022/FIAU-USS

Pimentel, 3 de febrero de 2022

#### VISTO:

El Acta de reunión N°3101 - 2022 del Comité de investigación de la Escuela profesional de INGENIERÍA DE SISTEMAS remitida mediante Oficio N°0004-2022/FIAU-IS-USS de fecha 2 de febrero de 2022, y;

#### CONSIDERANDO:

Que, de conformidad con la Ley Universitaria N° 30220 en su artículo 48° que a letra dice: "La investigación constituye una función esencial y obligatoria de la universidad, que la fomenta y realiza, respondiendo a través de la producción de conocimiento y desarrollo de tecnologías a las necesidades de la sociedad, con especial énfasis en la realidad nacional. Los docentes, estudiantes y graduados participan en la actividad investigadora en su propia institución o en redes de investigación nacional o internacional, creadas por las instituciones universitarias públicas o privadas.";

Que, de conformidad con el Reglamento de grados y títulos en su artículo 21° señala: "Los temas de trabajo de investigación, trabajo académico y tesis son aprobados por el Comité de Investigación y derivados a la facultad o Escuela de Posgrado, según corresponda, para la emisión de la resolución respectiva. El periodo de vigencia de los mismos será de dos años, a partir de su aprobación. En caso un tema perdiera vigencia, el Comité de Investigación evaluará la ampliación de la misma."

Que, de conformidad con el Reglamento de grados y títulos en su artículo 24° señala: La tesis es un estudio que debe denotar rigurosidad metodológica, originalidad, relevancia social, utilidad teórica y/o práctica en el ámbito de la escuela profesional. Para el grado de doctor se requiere una tesis de máxima rigurosidad académica y de carácter original. Es individual para la obtención de un grado; es individual o en pares para obtener un título profesional. Asimismo, en su artículo 25° señala: "El tema debe responder a alguna de las líneas de investigación institucionales de la USS S.A.C.".

Que, mediante documentos de vistos, el Comité de investigación de la referida Escuela profesional acordó aprobar la ampliación de la vigencia de las tesis que se detallan en el Acta de reunión N°3101 - 2022, de la línea de investigación de INFRAESTRUCTURA, TECNOLOGÍA Y AMBIENTE, a cargo de egresados del Programa de estudios INGENIERÍA DE SISTEMAS, hasta el 31 de julio de 2022.

Estando a lo expuesto, y en uso de las atribuciones conferidas y de conformidad con las normas y reglamentos vigentes;

#### SE RESUELVE:

**ARTÍCULO ÚNICO: AMPLIAR VIGENCIA**, de la Tesis a cargo de los egresados del Programa de estudios de **INGENIERÍA DE SISTEMAS** que se detallan en el anexo de la presente Resolución, hasta el 31 de julio de 2022.

#### REGÍSTRESE, COMUNÍQUESE Y ARCHÍVESE



Cc: Interesado, Archivo

**FACULTAD DE INGENIERÍA, ARQUITECTURA Y URBANISMO**  
**RESOLUCIÓN N°0018-2022/FIAU-USS**

Pimentel, 3 de febrero de 2022

**ANEXO**

<b>N°</b>	<b>APELLIDOS Y NOMBRES</b>	<b>TEMA DE TESIS</b>	<b>RESOLUCIÓN DE APROBACIÓN/MODIFICACIÓN/AMPLIACIÓN TEMA DE TESIS</b>	<b>FECHA RESOLUCIÓN DE APROBACIÓN TEMA DE TESIS</b>
1	CAMPOVERDE VEGA VICTOR ANDREE	ANÁLISIS COMPARATIVO DE RENDIMIENTO EN GESTORES DE BASES DE DATOS RELACIONALES Y NO RELACIONALES	2325-2020/FIAU-USS	17 de Noviembre de 2020 (*)
2	CLAVO TAFUR CRISTIAN JESUS	ANÁLISIS COMPARATIVO DE TÉCNICAS DE MITIGACIÓN DE ATAQUE DE DDOS EN CLOUD COMPUTING	2325-2020/FIAU-USS	17 de Noviembre de 2020 (*)
3	DIAZ CASTAÑEDA RUDY SLAYTONW	MODELO DE PROCESOS PARA EL DESARROLLO DE SOFTWARE CON CARACTERÍSTICAS DE SEGURIDAD PARA VULNERABILIDADES MÁS RECURRENTES	0933-2019/FIAU-USS	22 de julio de 2019
4	VENEGAS BRAVO JOSE ALEXANDER	ANÁLISIS COMPARATIVO DE RENDIMIENTO DE GESTORES DE BASE DE DATOS NOSQL DOCUMENTALES	2325-2020/FIAU-USS	17 de noviembre de 2020 (*)

(\*) La Resolución en referencia ampliaba la vigencia hasta el 31 de julio 2021

**Anexo 2:** ficha de registro para la recolección de datos.

FICHA DE REGISTRO DE DATOS

N:

PROYECTO:

ENCARGADO:

FECHA:

TIPO DE OPERACIÓN:

VOLCADO

LISTAR

ACTUALIZAR

ELIMINAR

COLECCIÓN DE BD

Nº DOCUMENTOS:

HORA INICIO:

HORA FIN:

SISTEMA GESTOR

DE BASE DE DATOS

NOSQL:

SOFTWARE

UTILIZADO:

DESCRIPCION DEL

PROCESO:

OBSERVACIONES:

1.LA FICHA TIENE QUE SER LLENADA CON LETRA LEGIBLE.

2.ES NECESARIO QUE SE REGISTRE EL ENCARGADO QUE EJECUTA CADA PROCESO.

3.EL TIPO DE OPERACIÓN TIENE QUE MARCARSE CON UNA X O UN CHECK.

4.ES NECESARIO QUE SE DETALLE LA DESCRIPCION DEL PROCESO



## Anexo 3: Instalación de Apache en Ubuntu 18.04

### Introducción

El servidor HTTP Apache es el servidor web más usado en el mundo. Provee muchas características poderosas, incluyendo módulos de carga dinámica, soporte robusto a medios, así como amplia integración a otros programas comúnmente utilizados.

En esta guía, explicaremos cómo instalar un servidor web Apache en tu servidor Ubuntu 18.04.

### Prerrequisitos

Antes de empezar a realizar estos pasos, se debe tener un usuario regular configurado en su servidor, éste debe corresponder a una cuenta con privilegios de sudo, que no sea superusuario (root). Adicionalmente, necesitarás habilitar un cortafuegos básico que bloquee los puertos no esenciales. Puedes aprender cómo configurar una cuenta de usuario regular y cómo ajustar el cortafuegos para tu servidor, siguiendo nuestra guía inicial de configuración para Ubuntu 18.04.

Cuando se tenga una cuenta disponible, ingresa con el usuario diferente a superusuario que mencionamos anteriormente y podrás empezar.

### Paso 1 — Instalar Apache

Apache se encuentra disponible dentro de los repositorios de software predeterminados de Ubuntu, haciendo posible la instalación mediante las herramientas convencionales de administración de paquetes.

Empezaremos por actualizar el índice de los paquetes locales. Esto, para garantizar que en él se refleje las cargas más recientes de las nuevas versiones de los paquetes.

```
sudo apt update
```

A continuación, instala el paquete apache2:

```
sudo apt install apache2
```

Después de confirmar la instalación, apt instalará Apache al igual que todas las dependencias requeridas.

## Paso 2 — Configurar el cortafuegos

Antes de probar el Apache, es necesario modificar los ajustes del cortafuegos de tal manera que se garantice el acceso externo a los puertos web por defecto. Asumiendo que seguiste las instrucciones de los prerequisites, tendrás un cortafuegos UFW configurado para restringir el acceso a tu servidor.

Durante la instalación, Apache por sí mismo, se registra en el UFW para proveer los perfiles que permitan habilitar o deshabilitar su acceso a través del cortafuego.

Lista los perfiles de aplicación dentro ufw digitando:

```
sudo ufw app list
```

Se debería desplegar una lista de perfiles de aplicación:

```
Salida
Available applications:
  Apache
  Apache Full
  Apache Secure
  OpenSSH
```

Como te has podido dar cuenta, existen tres perfiles disponibles para Apache:

**Apache:** este perfil habilita únicamente el puerto 80 (normal, tráfico web sin encriptar).

**Apache Full:** este perfil habilita dos puertos: puerto 80 (normal, tráfico web sin encriptar) y el puerto 443 (tráfico encriptado mediante TLS/SSL).

**Apache Secure:** este perfil habilita únicamente el puerto 443 (tráfico encriptado mediante TLS/SSL).

Se recomienda que siempre habilites el perfil con más restricciones dependiendo del tráfico requerido y cómo se ha configurado tu máquina. Como aún no hemos configurado el SSL para nuestro servidor en esta guía, solo permitiremos el tráfico a través del puerto 80:

```
sudo ufw allow 'Apache'
```

Se puede verificar el cambio digitando:

```
sudo ufw status
```

Se te debería desplegar que el tráfico HTTP se encuentra permitido:

```
Salida
Status: active

To Action From
--
OpenSSH ALLOW Anywhere
Apache ALLOW Anywhere
OpenSSH (v6) ALLOW Anywhere (v6)
Apache (v6) ALLOW Anywhere (v6)
```

Como puedes observar, el perfil ha sido activado, y el acceso al servidor web es permitido.

### Paso 3 — Verificar el servidor web

Al finalizar el proceso de instalación, Ubuntu 18.04 inicia Apache. Entonces, el servidor web debería encontrarse activo y en ejecución.

Verifica con el sistema de base systemd que el servicio se está ejecutando al digitar:

```
sudo systemctl status apache2
```

Salida

```
● apache2.service - The Apache HTTP Server
   Loaded: loaded (/lib/systemd/system/apache2.service; enabled; vendor
  preset: enabled)
   Drop-In: /lib/systemd/system/apache2.service.d
            └─apache2-systemd.conf
   Active: active (running) since Tue 2018-04-24 20:14:39 UTC; 9min ago
 Main PID: 2583 (apache2)
    Tasks: 55 (limit: 1153)
   CGroup: /system.slice/apache2.service
           └─2583 /usr/sbin/apache2 -k start
           └─585 /usr/sbin/apache2 -k start
           └─2586 /usr/sbin/apache2 -k start
```

Como se puede ver en esta salida, el servicio se ha iniciado exitosamente. Sin embargo, el mejor test para realizar esta comprobación es el de solicitar una página al servidor Apache.

Puedes acceder a la página por defecto de Apache para confirmar que éste se encuentra en correcta ejecución a través de tu dirección IP. Si no conoces la dirección IP de tu servidor, puedes obtenerla de diferentes maneras desde la línea de comandos.

Prueba digitando los siguiente en la línea de comandos de tu servidor:

```
hostname -I
```

Se te retornará algunas direcciones separadas por espacios. Pruébalas todas en tu navegador web para asegurar su funcionamiento.

Alternativamente, puedes digitar el siguiente comando, el cual te debería retornar la dirección IP pública de la manera que es percibida desde un lugar externo en internet:

```
curl -4 icanhazip.com
```


Cuando tengas la dirección IP de tu servidor, ingrésala en la barra de direcciones de tu navegador:

```
curl -4 icanhazip.com
```

Cuando tengas la dirección IP de tu servidor, ingrésala en la barra de direcciones de tu navegador:

```
http://ip_de_tu_servidor
```

A continuación, deberías ver la página web predeterminada de Ubuntu 18.04:



## Apache2 Ubuntu Default Page

**It works!**

This is the default welcome page used to test the correct operation of the Apache2 server after installation on Ubuntu systems. It is based on the equivalent page on Debian, from which the Ubuntu Apache packaging is derived. If you can read this page, it means that the Apache HTTP server installed at this site is working properly. You should **replace this file** (located at `/var/www/html/index.html`) before continuing to operate your HTTP server.

If you are a normal user of this web site and don't know what this page is about, this probably means that the site is currently unavailable due to maintenance. If the problem persists, please contact the site's administrator.

### Configuration Overview

Ubuntu's Apache2 default configuration is different from the upstream default configuration, and split into several files optimized for interaction with Ubuntu tools. The configuration system is **fully documented in `/usr/share/doc/apache2/README.Debian.gz`**. Refer to this for the full documentation. Documentation for the web server itself can be found by accessing the **manual** if the `apache2-doc` package was installed on this server.

The configuration layout for an Apache2 web server installation on Ubuntu systems is as follows:

```
/etc/apache2/
|-- apache2.conf
|   |-- ports.conf
|-- mods-enabled
|   |-- *.load
|   |-- *.conf
|-- conf-enabled
|   |-- *.conf
|-- sites-enabled
|   |-- *.conf
```

- `apache2.conf` is the main configuration file. It puts the pieces together by including all remaining configuration files when starting up the web server.
- `ports.conf` is always included from the main configuration file. It is used to determine the listening ports for incoming connections, and this file can be customized anytime.
- Configuration files in the `mods-enabled/`, `conf-enabled/` and `sites-enabled/` directories contain particular configuration snippets which manage modules, global configuration fragments, or virtual host configurations, respectively.
- They are activated by symlinking available configuration files from their respective `*-available/` counterparts. These should be managed by using our helpers **`a2enmod`**, **`a2dismod`**, **`a2ensite`**, **`a2dissite`**, and **`a2enconf`**, **`a2disconf`**. See their respective man pages for detailed information.
- The binary is called `apache2`. Due to the use of environment variables, in the default configuration, `apache2` needs to be started/stopped with `/etc/init.d/apache2` or `apache2ctl`. **Calling `/usr/bin/apache2` directly will not work** with the default configuration.

### Document Roots

By default, Ubuntu does not allow access through the web browser to *any* file apart of those located in `/var/www`, **`public_html`** directories (when enabled) and `/usr/share` (for web applications). If your site is using a web document root located elsewhere (such as in `/srv`) you may need to whitelist your document root directory in `/etc/apache2/apache2.conf`.

The default Ubuntu document root is `/var/www/html`. You can make your own virtual hosts under `/var/www`. This is different to previous releases which provides better security out of the box.

### Reporting Problems

Please use the `ubuntu-bug` tool to report bugs in the Apache2 package with Ubuntu. However, check **existing bug reports** before reporting a new bug.

Please report bugs specific to modules (such as PHP and others) to respective packages, not to the web server itself.

#### **Anexo 4:** Instrumento de recolección de datos.

### FICHA DE REGISTRO DE DATOS

<b>PROYECTO:</b> _____ _____ _____	<b>ENCARGADO:</b> _____ <b>FECHA:</b> _____ _____
--	---

**TIPO DE OPERACIÓN:** ( )VOLCADO  
 ( )LISTAR  
 ( )ACTUALIZAR  
 ( )ELIMINAR

<b>COLECCIÓN DE BD</b> _____ <b>Nº DOCUMENTOS:</b> _____  <b>SISTEMA GESTOR</b> <b>DE BASE DE DATOS</b> _____ <b>NOSQL:</b> _____  <b>SOFTWARE</b> <b>UTILIZADO:</b> _____	<b>HORA INICIO:</b> _____ <b>HORA FIN:</b> _____
--	---

**DESCRIPCION DEL PROCESO:** \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_  
 \_\_\_\_\_

**OBSERVACIONES:**

- 1.LA FICHA TIENE QUE SER LLENADA CON LETRA LEGIBLE.
- 2.ES NECESARIO QUE SE REGISTRE EL ENCARGADO QUE EJECUTA CADA PROCESO.
- 3.EL TIPO DE OPERACIÓN TIENE QUE MARCARSE CON UNA X O UN CHECK.
- 4.ES NECESARIO QUE SE DETALLE LA DESCRIPCION DEL PROCESO

### **Anexo 5: Informe de Base de Datos utilizado en las pruebas**

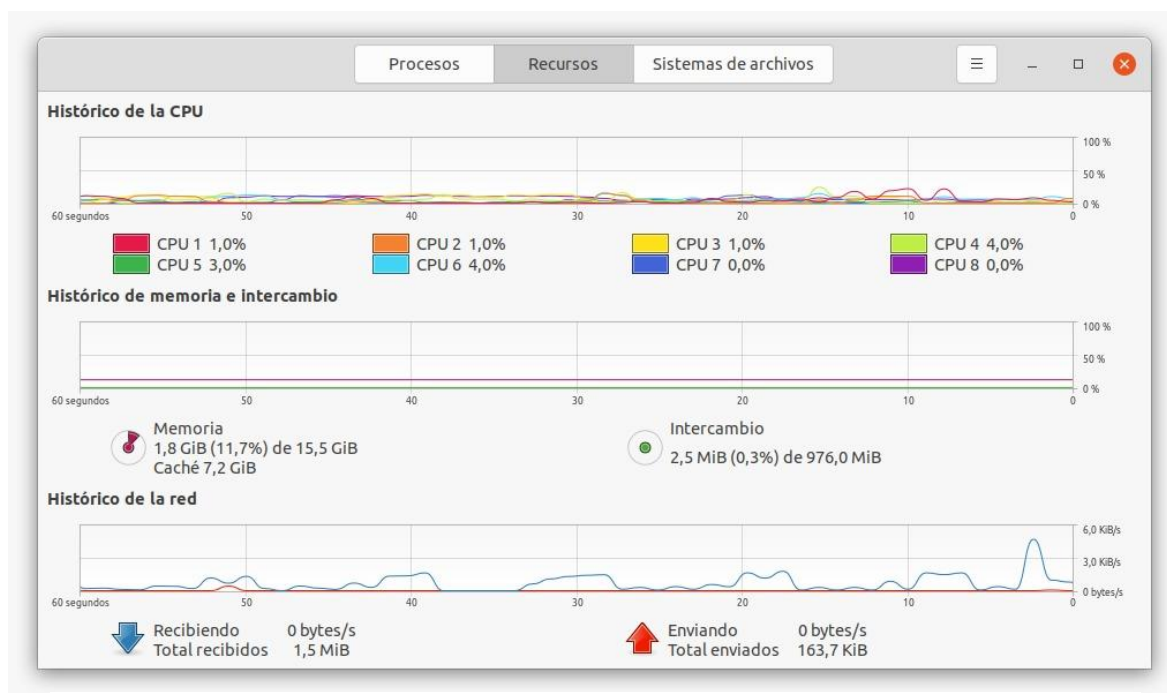
Los datos utilizados, pertenecen a la base de datos del ministerio de salud de Chile la cual viene siendo publicada desde el año 2001 en la página del departamento de estadísticas información de salud - DEIS (Chile) en un apartado como datos libres. Los datos utilizados "establecimientos" y "egresos hospitalarios" es información perteneciente a propiedad del estado de Chile. A continuación, se detalla aspectos generales de la base datos.

<b>Idioma</b>	Español
<b>Licencia</b>	Creative Commons Non-Commercial (Any) <a href="https://datos.gob.cl/dataset/12680">https://datos.gob.cl/dataset/12680</a>
<b>Autor</b>	Departamento de Estadísticas e información de Salud (DEIS) - Chile
<b>Tipo de Datos</b>	Texto Plano
<b>Repositorio</b>	Plataforma de datos abierto DEIS Chile
<b>Url Repositorio</b>	<a href="https://deis.minsal.cl/">https://deis.minsal.cl/</a>

## Anexo 6: Explicación sobre el porcentaje del uso de CPU en los Resultados

De acuerdo a las tablas presentadas, en la sección Resultados sobre el porcentaje de uso de CPU, donde los resultados de algunas tablas de las distintas pruebas de rendimiento (Volcado, Listado, Actualización y Eliminación) superan el 100%; partiremos explicando desde las características físicas (Hardware) de nuestro entorno de pruebas, se utilizó un laptop con un procesador Core i7 de Octava Generación con un procesador de 8 núcleos. Asimismo, se explica cuando un procesador tiene 8 núcleos favorece el rendimiento de sistemas que trabajan en multi-tarea. Un ejemplo es el uso de antivirus. Un antivirus es capaz de consumir el 100% de un núcleo, parando la ejecución de otras aplicaciones hasta que termine su trabajo. En estos casos disponer de más núcleos mejora la respuesta de todo el sistema.

Asimismo, cuando se realizaron las pruebas de rendimiento el gestor de base de datos Apache CouchDB consumió el 100% de un núcleo, tomando lo que le hacía falta del siguiente núcleo para poder realizar las operaciones de rendimiento, es por eso que en algunas tablas el porcentaje del uso del CPU supera el 100%, ya que el comando TOP (comando que se usó para medir el uso del CPU) suma el porcentaje de uso de cada uno de sus núcleos del procesador.



**Figura 81-** *Histórico del CPU mostrado sus 8 núcleos en porcentaje*