

UNIVERSIDAD TÉCNICA DE AMBATO



FACULTAD DE INGENIERÍA EN SISTEMAS ELECTRÓNICA E INDUSTRIAL

MAESTRÍA EN GESTIÓN DE BASES DE DATOS III VERSIÓN

Tema: "EVALUACIÓN DEL IMPACTO DE LOS SISTEMAS DE GESTIÓN NOSQL EN LOS TIEMPOS DE RESPUESTA DE CONSULTA DE CAUSAS EN LA FUNCIÓN JUDICIAL DEL ECUADOR."

Trabajo de Titulación Previo a la obtención del Grado Académico de
Magíster en Gestión de Bases de Datos

Autora: Ing. Ana Virginia Flores Pimbo

Director: Ing. Mg. Klever Renato Urvina Barrionuevo.

Ambato – Ecuador

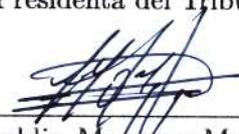
2017

A la Unidad Académica de Titulación de la Facultad de Ingeniería en Sistemas, Electrónica e Industrial.

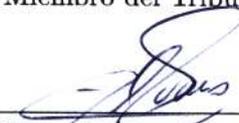
El Tribunal receptor del Trabajo de Investigación presidido por la Ingeniera Elsa Pilar Urrutia Urrutia Magíster, e integrado por los señores Ingeniero Franklin Mayorga Mayorga Magister, Ingeniero Edison Alvarez Mayorga Magister, Ingeniero Carlos Nuñez Miranda Magister, designados por la Unidad Académica de Titulación de la Universidad Técnica de Ambato, para receptor el Trabajo de Investigación con el tema: “EVALUACIÓN DEL IMPACTO DE LOS SISTEMAS DE GESTIÓN NoSQL EN LOS TIEMPOS DE RESPUESTA DE CONSULTA DE CAUSAS EN LA FUNCIÓN JUDICIAL DEL ECUADOR”, elaborado y presentado por la señorita Ingeniera Ana Virginia Flores Pimbo, para optar por el Grado Académico de Magister en Gestión de Bases de Datos; una vez escuchada la defensa oral del Trabajo de Investigación el Tribunal aprueba y remite el trabajo para uso y custodia en las bibliotecas de la UTA.



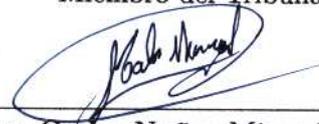
Ing. Elsa Pilar Urrutia Urrutia, Mg.
Presidenta del Tribunal



Ing. Franklin Mayorga Mayorga, Mg.
Miembro del Tribunal



Ing. Edison Alvarez Mayorga, Mg.
Miembro del Tribunal



Ing. Carlos Nuñez Miranda, Mg.
Miembro del Tribunal

AUTORÍA

La responsabilidad de las opiniones, comentarios y críticas emitidas en el Trabajo de Investigación presentado con el tema: “EVALUACIÓN DEL IMPACTO DE LOS SISTEMAS DE GESTIÓN NoSQL EN LOS TIEMPOS DE RESPUESTA DE CONSULTA DE CAUSAS EN LA FUNCIÓN JUDICIAL DEL ECUADOR”, le corresponde exclusivamente a: Ingeniera Ana Virginia Flores Pimbo, Autora bajo la Dirección del Ingeniero Kléver Renato Urvina Barrionuevo Magister, Director del Trabajo de Investigación; y el patrimonio intelectual a la Universidad Técnica de Ambato



Ing. Ana Virginia Flores Pimbo

CC: 1804192837

AUTORA



Ing. Kléver Renato Urvina Barrionuevo, Mg.

CC: 1802667970

DIRECTOR

DERECHOS DEL AUTOR

Autorizo a la Universidad Técnica de Ambato, para que el Trabajo de Investigación, sirva como documento disponible para su lectura, consulta y procesos de investigación, según las normas de la Institución.

Cedo los Derechos de mi trabajo, con fines de difusión pública, además apruebo la reproducción de este, dentro de las regulaciones de la Universidad.



Ing. Ana Virginia Flores Pimbo

CC: 1804192837

ÍNDICE

Portada	i
A la Unidad académica de titulación	ii
Autoría	iii
Derechos del autor	iv
Agradecimiento	xii
Dedicatoria	xiii
Introducción	1
CAPÍTULO I EL PROBLEMA DE INVESTIGACIÓN	2
1.1 Tema de Investigación	2
1.2 Planteamiento del problema	2
1.2.1 Contextualización	2
1.2.2 Análisis crítico	4
1.2.3 Prognosis	5
1.2.4 Formulación del problema	5
1.2.5 Interrogantes	6
1.2.6 Delimitación del objeto de investigación	6
1.3 Justificación	6
1.4 Objetivos	7
1.4.1 General	7
1.4.2 Específicos	7
CAPÍTULO II MARCO TEÓRICO	8
2.1 Antecedentes Investigativos	8
2.2 Fundamentación filosófica	9
2.3 Fundamentación legal	9

2.4	Categorías Fundamentales	10
2.4.1	Red de inclusión	10
2.4.2	Constelación de ideas de la variable independiente	10
2.4.3	Constelación de ideas de la variable dependiente	11
2.5	Hipótesis	15
2.6	Señalamiento de variables	15
CAPÍTULO III METODOLOGÍA		16
3.1	Enfoque	16
3.2	Modalidad Básica de la investigación	16
3.3	Nivel o tipo de investigación	16
3.4	Población y muestra	16
3.5	Operacionalización de variables	17
3.5.1	Variable independiente	17
3.5.2	Variable dependiente	18
3.6	Recolección de información	18
3.7	Procesamiento de la información	18
CAPÍTULO IV ANÁLISIS E INTERPRETACIÓN DE RESULTADOS		20
4.1	Análisis e interpretación de los resultados	20
4.1.1	Encuesta	20
4.1.2	Verificación de hipótesis	30
4.1.2.1	Modelo lógico	30
4.1.2.2	Modelo estadístico	30
CAPÍTULO V CONCLUSIONES Y RECOMENDACIONES		33
5.1	Conclusiones	33
5.2	Recomendaciones	34
CAPÍTULO VI PROPUESTA		35
6.1	Datos informativos	35
6.2	Antecedentes de la propuesta	35
6.3	Justificación	36
6.4	Objetivos	36
6.4.1	Objetivo general	36
6.4.2	Objetivos específicos	37
6.5	Análisis de factibilidad	37
6.5.1	Tecnológica	37
6.5.2	Operativa	37

6.5.3	Económica	37
6.6	Fundamentación	37
6.6.1	Científico Técnica	37
6.6.1.1	Arquitectura actual del sistema de consulta de causas en la Función Judicial	37
6.6.1.2	Análisis y selección del sistema de gestión de datos NoSQL	43
6.7	Metodología	48
6.7.1	Preparación de ambientes de prueba	48
6.7.2	Extracción, transformación y carga de datos desde Microsoft SQL Server 2012 a MongoDB 3.4	53
6.7.3	Optimizar base de datos	62
6.7.4	Recolección de datos	63
6.7.5	Análisis de resultados	76
6.7.6	Conclusiones	81
6.7.7	Recomendaciones	82
	Bibliografía	82
	Referencias	82
	ANEXOS	87

ÍNDICE DE TABLAS

2.1	Sistemas de gestión SQL vs NoSQL	13
3.1	Población a ser investigada	17
3.2	Operacionalización de la variable independiente	17
3.3	Operacionalización de la variable dependiente	18
3.4	Plan de recolección de la información	18
4.1	Encuestas - Administradores de bases de datos	20
4.2	Nivel de conocimiento de las Bases de datos NoSQL	20
4.3	Velocidad de procesamiento en los sistemas gestores de datos	21
4.4	Sistemas NoSQL una opción de implementación	22
4.5	Integridad referencial en las aplicaciones	23
4.6	Esquema libre en las aplicaciones	24
4.7	Des-normalización de datos	25
4.8	Escalamiento horizontal	26
4.9	Lenguaje NoSQL	27
4.10	Tipos de sistemas NoSQL	28
4.11	Sistemas NoSQL más conocidos	29
4.12	Frecuencia observadas	31
4.13	Cálculo de Chi-Cuadrado	32
6.1	Usuarios por hora - Consulta de Causas	42
6.2	Software para entornos de prueba	48
6.3	Hardware para los entornos de prueba	49
6.4	Características tabla ActuacionesJudiciales - SQL Server	63
6.5	MongoDB - Configuración JMeter - Grupo de Hilos	70
6.6	Test de rendimiento 1	72
6.7	Test de rendimiento 2	73
6.8	Test de rendimiento 2	74
6.9	Resultados Test1	75
6.10	Resultados Test2	75
6.11	Resultados Test3	76

6.12 Promedio - Test1	77
6.13 Promedio - Test 2	77
6.14 Promedio - Test3	78
6.15 Datos necesarios para contrastar la hipótesis - Test 1	80
6.16 Datos necesarios para contrastar la hipótesis - Test 2	80
6.17 Datos necesarios para contrastar la hipótesis - Test 3	81

ÍNDICE DE FIGURAS

1.1	Árbol de problemas	5
2.1	Categorías fundamentales	10
2.2	Subcategorías de la variable independiente	10
2.3	Subcategoría de la variable dependiente	11
4.1	Nivel de conocimiento de las Bases de datos NoSQL	21
4.2	Velocidad de procesamiento en los sistemas gestores de datos	22
4.3	Velocidad de procesamiento en los sistemas gestores de datos	23
4.4	Integridad referencial en las aplicaciones	24
4.5	Esquema libre en las aplicaciones	25
4.6	Des-normalización de datos	26
4.7	Escalamiento horizontal	27
4.8	Lenguaje NoSQL	28
4.9	Tipos de sistemas NoSQL	29
4.10	Sistemas NoSQL más conocidos	30
6.1	Arquitectura aplicación web - Consulta de Causas	38
6.2	Arquitectura de despliegue - Consultas de Causas	39
6.3	Arquitectura de base de datos - Consulta de Causas	40
6.4	Proceso de búsqueda - Consulta de Causas	41
6.5	Horas de mayor demanda - Consulta de Causas	43
6.6	DB-Engines - Ranking of Document Stores	46
6.7	Relación de términos entre una base de datos Relacional y una Documental: MongoDB	47
6.8	Arquitectura del escenario de pruebas	49
6.9	MongoDB starting	51
6.10	Conexión a Robomongo	51
6.11	Creación base de datos XELADAB	51
6.12	Verificación de creación de Base de datos y colección	52
6.13	Sql Server Data Tools	53
6.14	Integration Services Project	54

6.15	Creación de paquete en SSIS	54
6.16	1.- Extracción de datos desde Microsoft SQL Server	55
6.17	2.- Extracción de datos desde Microsoft SQL Server	56
6.18	3.- Extracción de datos desde Microsoft SQL Server	56
6.19	4.- Extracción de datos desde Microsoft SQL Server	57
6.20	Script Inserta datos en MongoDB	58
6.21	Agregar DLL a namespaces	58
6.22	Ejecución del paquete en SSIS	61
6.23	Datos migrados a mongoDB	61
6.24	SQL Server 2012 - Indices creados sobre la tabla ActuacionesJudiciales	62
6.25	MongoDB - Indices creados sobre la colección ActuacionesJudiciales	62
6.26	MongoDB - Indices proceso de creación	63
6.27	SQL Server - Configuración JMeter - Grupo de Hilos	69
6.28	SQL Server - Configuración JMeter - Conexión JDBC	69
6.29	SQL Server - Configuración JMeter - Peticiones JDBC	70
6.30	MongoDB - Configuración JMeter - Conexión al servidor	71
6.31	MongoDB- Configuración JMeter - Peticiones al servidor	71
6.32	Test1 - SQLServer vs MongoDB	77
6.33	Test2 - SQLServer vs MongoDB	78
6.34	Test3 - SQLServer vs MongoDB	79

AGRADECIMIENTO

Rey de majestad tremenda a quienes salves será por tu gracia, ¡sálvame, fuente de piedad!

Piadoso no permitas que mi alma sufra hay un agujero enorme en ella. Ven y dame la mano, ¡sálvame, fuente de piedad!

Divina gracia, llevo demasiada vida corriendo por mis venas, pensando que el amor parará el dolor, diciendo que el amor matará al miedo, ¿lo creo? ¿debo creerlo?, ¡sálvame, fuente de piedad!

Dueño de nuestras almas, juzgador de nuestros pecados, no permitas que este cuerpo se desanime en la lucha y búsqueda del sentido de su existencia, ¿sobreviviré? debo sobrevivir, me levantaré y caminaré hacia la luz, ¡sálvame, fuente de piedad!

DEDICATORIA

Para tí madre, gracias por ser nuestra guía. Para tí abuelita, gracias por todo tu cariño. Para ustedes hermanos y sobrinos por su apoyo y amor constante. Para mi padre.

Ana Virginia Flores Pimbo

**UNIVERSIDAD TÉCNICA DE AMBATO FACULTAD DE
INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL**

**DIRECCIÓN DE POSGRADO MAESTRÍA EN GESTIÓN DE BASE
DE DATOS**

TEMA: “EVALUACIÓN DEL IMPACTO DE LOS SISTEMAS DE GESTIÓN NoSQL EN LOS TIEMPOS DE RESPUESTA DE CONSULTA DE CAUSAS EN LA FUNCIÓN JUDICIAL DEL ECUADOR.”

AUTORA: Ing. Ana Virginia Flores Pimbo

DIRECTOR: Ing. Kléver Renato Urvina Barrionuevo

FECHA: 07 de Junio del 2017

RESUMEN

La presente investigación busca comprobar cuan eficientes son los sistemas NoSQL con respecto a los relacionales. El caso de estudio es la evaluación de tiempos de respuesta en las consultas utilizando el sistema MongoDB y Microsoft SQL Server, para observar cual es el comportamiento de cada uno en un entorno de prueba.

La evaluación de tiempos se realizó sobre los datos de las actuaciones judiciales de los procesos judiciales que maneja la Función Judicial del Ecuador. Para esto, ha sido necesario ejecutar varios pasos, de entre ellos se puede destacar la migración de datos desde Microsoft SQL Server a MongoDB. Para luego proceder con las pruebas basados en una muestra.

De esta manera, se recopiló información generada por cada uno de los sistemas de gestión de datos. Posteriormente se generó un análisis de la información obtenida, para finalmente emitir las conclusiones y recomendaciones.

Palabras clave: NoSQL, MongoDB, Microsoft SQL Server, Tiempos de respuesta, Base de datos, No estructuradas, Gestión de datos, ETL, Escalabilidad, Jmeter.

UNIVERSIDAD TÉCNICA DE AMBATO FACULTAD DE
INGENIERÍA EN SISTEMAS, ELECTRÓNICA E INDUSTRIAL

DIRECCIÓN DE POSGRADO MAESTRÍA EN GESTIÓN DE BASE
DE DATOS

THEME: “EVALUATION OF THE IMPACT CONCERNING NOSQL AND QUERY RESPONSE TIME OF THE CAUSES IN THE ECUADORIAN JUDICIARY FUNCTION”

AUTHOR: Ing. Ana Virginia Flores Pimbo

DIRECTED BY: Ing. Kléver Renato Urvina Barrionuevo

DATE: 07 de Junio del 2017

ABSTRACT

The present research seeks to verify how efficient could be NoSQL systems respect to relationals. The case study is the comparison and evaluation of the behaviors between MongoDB system and Microsoft SQL Server system in a test environment.

The evaluation and comparison of times was made on the data of the judicial trials handle by the Judicial Function of the government of the Republic of Ecuador. For this reason, it was necessary to carry out several steps since the migration of data from Microsoft SQL Server to MongoDB. Later, both systems were check in tests with a random sample.

So, the information generated in test by each system was stock up. At last, the researcher make an analysis of the results, giving conclusions and suggestions.

Keyword: NoSQL, MongoDB, SQL Server, Response times, Data Base, Unstructured, Data management, Scalability, ETL, Jmeter.

INTRODUCCIÓN

Si bien los avances tecnológicos en cuanto a Sistemas de Gestión de Bases de Datos (SGBD) traen consigo múltiples beneficios en diferentes aspectos como velocidad de procesamiento, escalabilidad y otros, no significa que reemplacen en su totalidad a los sistemas que han venido funcionando hasta ahora. Pero es necesario conocer que tan ventajosas son estas nuevas tecnologías para con nuestro entorno de trabajo.

Es por ello que el presente trabajo denominado “EVALUACIÓN DEL IMPACTO DE LOS SISTEMAS DE GESTIÓN NoSQL EN LOS TIEMPOS DE RESPUESTA DE CONSULTA DE CAUSAS EN LA FUNCIÓN JUDICIAL DEL ECUADOR.”, se orienta a evidenciar la importancia de considerar nuevas herramientas para la gestión de datos.

La investigación contempla un análisis sobre los tiempos de respuesta en las consultas entre un sistema NoSQL y otro relacional, demostrando la superioridad de los sistemas NoSQL cuando de procesar grandes cantidades de datos se trata.

Este documento es una guía de fácil interpretación que consolida información del proceso que se llevó a cabo para demostrar cuán ventajoso es un sistema con respecto al otro. Analizando cada uno de los resultados arrojados en las pruebas, referencia que ha permitido y ha sido base fundamental para emitir conclusiones y recomendaciones.

CAPÍTULO I

EL PROBLEMA DE INVESTIGACIÓN

1.1. Tema de Investigación

Evaluación de impacto de los sistemas de gestión NoSQL en los tiempos de respuesta de consulta de causas en la Función Judicial del Ecuador.

1.2. Planteamiento del problema

1.2.1. Contextualización

Las exigencias de la sociedad por lograr gobiernos democráticos y transparentes, ha llevado a que los Estados establezcan políticas públicas, planes estratégicos y operativos que otorguen a los ciudadanos el derecho del acceso a la Información Pública. Si bien, no todos cuentan con una Política de Acceso a la Información Pública; es oportuno destacar algunos casos tales como: Chile, Colombia, Uruguay, Perú entre otros en América Latina.

Chile con la Ley 20.285¹ y Colombia con la Ley 1712², regularizan el derecho de Acceso a la Información Pública. Dichas leyes convergen en el punto de transparentar la función pública. Chile ha desarrollado el sitio web denominado Portal de Transparencia³ como cumplimiento de la ley. En cuanto a Colombia en el Artículo 17, señala que se debe garantizar que los sistemas de información electrónica sean una herramienta para promover el acceso a la información pública, por tanto deben estar alineados con la estrategia de gobierno en línea. Colombia, “busca construir un Estado más eficiente, más transparente y más participativo por medio de las Tecnologías de Información y Comunicación (TIC).” (Ministerio de Tecnologías de

¹Ley 20285, Sobre Acceso a la Información Pública. Publicada el 20 Agosto de 2008. Vigente desde el 05 de Enero de 2016.

²Ley 1721, Ley de Transparencia y del Derecho de Acceso a la Información Pública Nacional. Publicada el 6 de Marzo de 2014”

³Ver en: Portal Transparencia Chile. URL. <https://www.portaltransparencia.cl/PortalPdT/>

la Información y las Comunicaciones de la República de Colombia, s.f.)

Al igual que Chile y Colombia, Ecuador forma parte de los países que fomentan el derecho al acceso a la información pública. Dispuesto así en la Ley Orgánica de Transparencia y Acceso a la Información Pública (LOTAIP). En este contexto, Ecuador ha desarrollado el Plan Nacional de Gobierno Electrónico.

El plan antes mencionado cuenta con varios principios entre ellos el, “**Principio de interrelación por medios digitales:** Garantiza que no sea obligatorio que una persona asista a una dependencia gubernamental para realizar un trámite, salvo lo dispuesto en la normativa vigente, y que siempre habrá la opción de usar medios digitales y acciones en línea para realizar cualquier trámite.” (Secretaría Nacional de la Administración Pública de la República del Ecuador. 2014)

El Plan de Gobierno Electrónico contempla tres objetivos que son: “Gobierno Cercano, Gobierno Abierto y Gobierno Eficiente y Eficaz. Los objetivos referidos están asociados con incrementar la provisión y calidad de servicios en línea; incrementar el acceso y la transparencia a la información pública como medio para fomentar la participación y colaboración ciudadana en el quehacer del gobierno; incrementar la eficiencia, eficacia y desempeño de las entidades públicas.” (Secretaría Nacional de la Administración Pública de la República del Ecuador. 2014)

En el caso particular de la Función Judicial del Ecuador, han sido implementados diferentes servicios alineados con los objetivos del Plan de Gobierno Electrónico entre los cuales tenemos: el Sistema Automatizado de Trámites Judiciales del Ecuador (SATJE), el Sistema Único de Pensiones Alimenticias (SUPA), el Sistema Informático de Remates Judiciales, Consulta de Causas Web entre otros. Cada uno de estos servicios benefician a la ciudadanía ahorrando tiempo y dinero ya que se encuentran en línea, facilitando el acceso y transparentando la información.

El enfoque orientado hacia la excelencia en la atención a la ciudadanía promulgado en el Plan de Gobierno Electrónico ha llevado a la Función Judicial a consolidar una gran infraestructura tecnológica que es el pilar fundamental y soporte de los sistemas antes mencionados con la intención de mejorar los servicios del ciudadano.

El presente estudio se enfoca directamente al sistema de Consulta de Causas Web, sistema que permite la consulta de actuaciones o diligencias judiciales, dando cumpli-

miento así a lo establecido en la Constitución de la República del Ecuador. Capítulo segundo, sección tercera, artículo 18 (...) 2 sobre “Acceder libremente a la información generada en entidades públicas ...” (Asamblea Constituyente de la República del Ecuador, 2008), al Código Orgánico General de Procesos. Libro I, título I, artículo 8 sobre la, “Transparencia y publicidad de los procesos judiciales” (Asamblea Nacional de la República del Ecuador, 2015) y al Código Orgánico de la Función Judicial, capítulo II, artículo 13 del “Principio de publicidad.” (Función Judicial de la República del Ecuador, 2009)

El sistema de Consulta de Causas Web, se ha convertido en la piedra angular entre la Función Judicial y la ciudadanía, el uso de Internet es común entre la sociedad y por ende el servicio del sistema ya mencionado debe responder con rapidez a las consultas del ciudadano.

El sistema de consulta tiene un retardo de una hora en la actualización de los datos para la visualización al público cuando son procesos nuevos, esto, en vista de la gran cantidad de datos que se manejan en la Función Judicial, una vez que los procesos judiciales son visibles al público sobre estos se puede realizar consultas sobre sus litigantes, dependencias jurisdiccionales donde se encuentra y actuaciones judiciales. El retardo que se ha detectado actualmente es al momento de obtener la respuesta de las actuaciones judiciales, el texto de la actuación judicial que se realizan sobre algunos procesos superan las cinco páginas esto sumado al número de actuaciones registradas, al creciente número de usuarios que hacen uso del sistema y las diferentes fuentes de datos que se utilizan para visualizar el reporte en el sistema, ha provocado que en ciertas ocasiones se presenten problemas de lentitud.

1.2.2. Análisis crítico

La Figura 1.1 refleja las causas y los posibles efectos del problema. El sistemas de consultas web tiene como propósito la visualización de información referente a la procesos judiciales y su historia. Los datos presentados provienen de otros subsistemas que operan en la intranet de la Función Judicial tales sorteos, escritos y trámite. La integración de toda esta información incide directamente en los tiempos de respuesta que se tiene al realizar consultas especialmente al consultar las actuaciones judiciales.

Adicionalmente se debe indicar que al encontrarse el servicio disponible en Internet la concurrencia de usuarios es alta pues se estima que diariamente más de 2000

usuarios acceden al servicio de tal manera que presentar la información en el tiempo adecuado conlleva un tratamiento especial de los datos por parte de la Función Judicial.

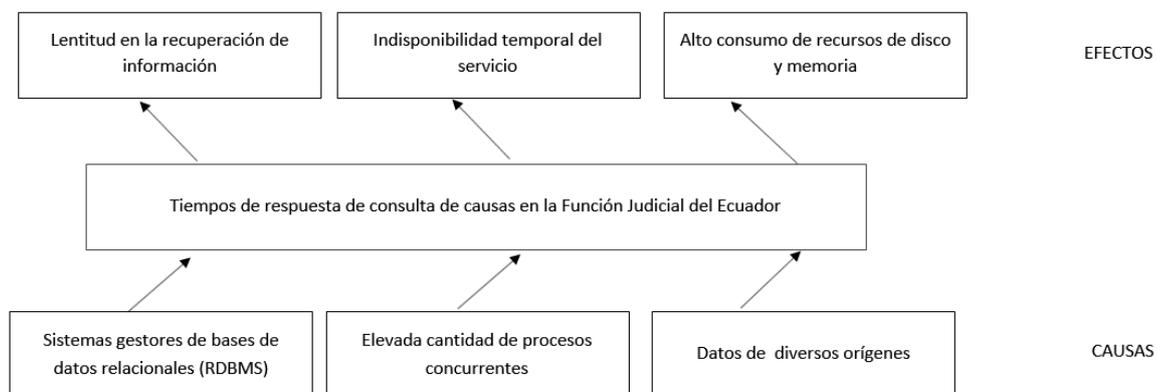


Figura 1.1: Árbol de problemas

1.2.3. Prognosis

El explosivo incremento de información a obligado a la institución a buscar métodos, técnicas y herramientas que permitan la correcta administración de los datos.

Si no se establece un sistema adecuado para mejorar los tiempos de respuesta y rendimiento del servicio, se puede llegar a provocar la no disponibilidad total dificultando el acceso a la aplicación.

En el caso de no aplicar un proceso óptimo para el sistema de consulta, cada vez más se dificultará su administración.

De continuar con el proceso actual en un corto periodo de tiempo sería necesario re-potenciar la infraestructura que soporta el servicio, incrementando los gastos y tiempo del personal técnico.

1.2.4. Formulación del problema

¿Incide los sistemas de gestión NoSQL en los tiempos de respuesta de consulta de causas en la Función Judicial del Ecuador?

1.2.5. Interrogantes

¿Cuál es el proceso utilizado actualmente para extracción y presentación de la información ?

¿Qué tipos de Gestores de Base de Datos (SGBD) existen en la actualidad y cuáles son los más utilizados por las aplicaciones web?

¿Qué características debe poseer la herramienta para lograr que las consultas se ejecuten en menor tiempo?

¿Qué técnicas y diseños de datos son utilizados para el almacenamiento y análisis de grandes cantidades de información?

1.2.6. Delimitación del objeto de investigación

Delimitación del contenido:

- Campo: Base de Datos.
- Área: Sistemas gestores de bases de datos.
- Aspecto: Bases de datos documentales para la recuperación y acceso a la información.

Delimitación espacial:

- Empresa: Funcion Judicial del Ecuador departamento de Tecnologías de la Información y Comunicación planta central.

Delimitación temporal:

- La investigación se efectuará desde marzo 2017 a junio 2017.

1.3. Justificación

El uso de medios digitales es inminente respecto a la prestación de servicios públicos y privados. Y garantizar la disponibilidad de estos obliga a desarrollar herramientas basadas en tecnologías y estándares actualizados.

Sin duda los servicios electrónicos generan gran cantidad de datos, lo cual ha llevado a buscar nuevas formas de almacenamiento y tratamiento de la información. La tendencia hacia el uso de base de datos no estructuradas o el uso de nuevos tipos de datos en base de datos relacionales (RDBMS) ofrece una serie de beneficios en cuanto a rendimiento y almacenamiento, estos beneficios pueden ser utilizados por la Función Judicial del Ecuador, para mejorar los tiempos de respuesta en su servicio de consulta de causas.

La presente investigación se encuentra ligada con los artículos establecidos en el

Decreto 149⁴ sobre la implementación del Gobierno Electrónico en la Administración Pública, específicamente con los Artículos 1, 4 y 5.

Desde luego la presente investigación es factible y viable, se cuenta con un amplio marco teórico para su aplicación en la Función Judicial, siendo posible plantear una solución al problema a investigar.

1.4. Objetivos

1.4.1. General

Determinar la incidencia de los sistemas de gestión NoSQL en los tiempos de consulta de causas en la Función Judicial del Ecuador.

1.4.2. Específicos

- Analizar el proceso actual establecido por la Función Judicial para extraer y presentar la información a través del sistema de Consulta de Causas.
- Determinar los tiempos de respuesta promedio de las consultas a la base de datos realizadas por el sistema de Consultas de Causas para obtener las actuaciones judiciales.
- Determinar una solución factible y viable aplicando métodos, técnicas y herramientas que permitan mejorar los tiempos de respuesta al obtener las actuaciones judiciales en el sistema de Consultas de Causas.

⁴Decreto 149, Implementación del Gobierno Electrónico en la Administración Pública. Registro Oficial Suplemento 146 de 18-dic.-2013”

CAPÍTULO II

MARCO TEÓRICO

2.1. Antecedentes Investigativos

Luego de realizada la búsqueda en los repositorios de la Facultad de Ingeniería en Sistemas Electrónica e Industrial de la Universidad Técnica de Ambato, no se ha encontrado trabajos con temáticas similares a la planteada.

Sin embargo luego de la búsqueda en repositorios de universidades e institutos externos se ha encontrado temas como:

“Integración de búsquedas de texto completo en bases de datos NOSQL.” En este trabajo se hace una investigación de diferentes bases de datos NoSQL y métodos de indexación. Concluyendo que: “Si pretendemos desarrollar una aplicación que requiera la lectura/escritura de cantidades de datos y pueda dar servicio a millones de usuarios sin perder rendimiento, entonces debemos plantearnos el uso de una base de datos NoSQL.

Se puede utilizar una base de datos NoSQL para almacenar toda la información de una aplicación, aunque en la mayoría de los casos se recurre a sistemas mixtos que combinan los clásicos sistemas relacionales (fácilmente manipulables e interrogables con el lenguaje SQL) con soluciones NoSQL para aquellas funcionalidades que requieren millones de consultas en tiempo real.” (Valenzo et al, 2013)

“Grandes Datos y Algoritmos Eficientes para Búsquedas de Escala Web. El objetivo principal en este proyecto es estudiar el problema de las búsquedas a gran escala e incorporar el análisis de datos masivos para mejorar las prestaciones de los sistemas de búsqueda.” (Tolosa et al, 2016)

“Rendimiento de tecnologías NoSQL sobre cantidades masivas de datos.” En es-

ta investigación se evalúa las “características de carga de datos y consultas de los mismos, tanto en bases de datos tradicionales las cuales utilizan lenguaje de consulta SQL, como en bases de datos NoSQL, frente a distintas estructuras y tipos de datos, y propone una alternativa para realizar un diseño de una arquitectura de persistencia adaptada a las distintas necesidades.” Concluyendo entre otras que “considerando las limitaciones de las bases de datos relacionales, el movimiento NoSQL que está surgiendo parece adecuado para resolver una gran cantidad de problemas adjudicados al fenómeno de Big Data, resultando tipos de datos poco influyentes, en este caso, con la performance de los motores. ”(Florencia et al, 2014)

“Búsquedas en grandes volúmenes de datos. El presente trajo de investigación se enfoca al diseño de índices eficientes que sirvan de apoyo a sistemas de recuperación de información orientados a conjuntos masivos de datos multimedia”. (Britos et al, 2016)

Una vez revisado diferentes investigaciones de distintas universidades y artículos científicos de varias revistas que se relacionan con el tema de investigación se ha llegado a concluir que, es de vital importancia realizar un estudio sobre las características de los sistemas gestores de base de datos, las fortalezas y debilidades que presentan cada uno de ellos en cuanto a las aplicaciones que las utilizan. Las pruebas de rendimiento no se quedan de lado, esto permitirá comparar la situación actual del procesos que se utiliza para la extracción de la información con la alternativa que se pretende proponer.

2.2. Fundamentación filosófica

La presente investigación se enmarca en el paradigma Critico Propositivo, es crítico porque realiza un Análisis Crítico del Problema, y es Propositivo porque busca proponer una solución factible al problema.

2.3. Fundamentación legal

El presente estudio se basa en documentación publicada en Internet bajo la licencia Creative Commons (CC) y software distribuido bajo la Licencia Pública General de GNU.

El Código Orgánico General de Procesos. Libro I, título I, artículo 8 sobre la, “Transparencia y publicidad de los procesos judiciales” (Asamblea Nacional de la República del Ecuador, 2015) y al Código Orgánico de la Función Judicial, capítulo

II, artículo 13 del “Principio de publicidad.” (Función Judicial de la República del Ecuador, 2009)

2.4. Categorías Fundamentales

2.4.1. Red de inclusión

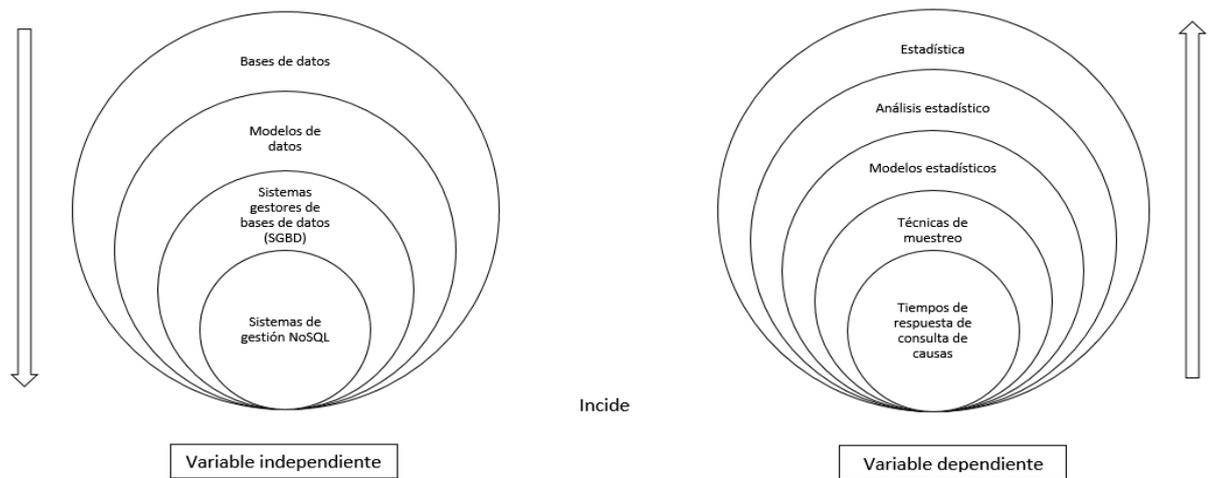


Figura 2.1: Categorías fundamentales

2.4.2. Constelación de ideas de la variable independiente

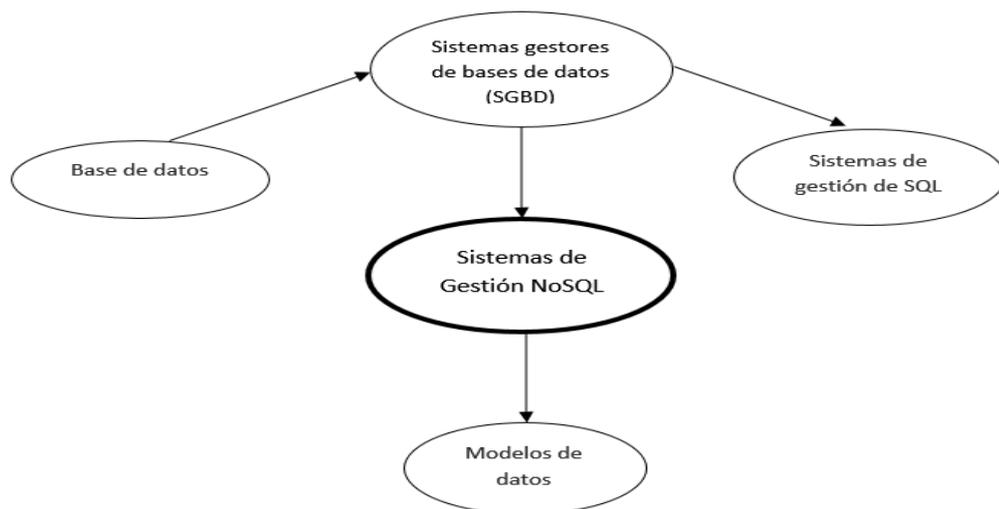


Figura 2.2: Subcategorías de la variable independiente

2.4.3. Constelación de ideas de la variable dependiente

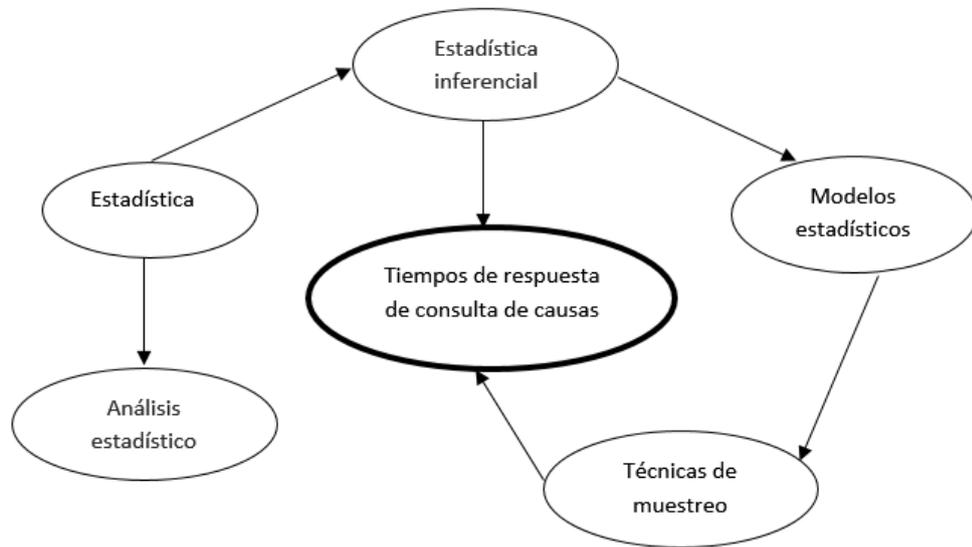


Figura 2.3: Subcategoría de la variable dependiente

Base de datos

“Conjunto de datos no redundantes, almacenados en un soporte informático, organizado de forma independiente de su utilización y accesible simultáneamente por distintos usuarios y aplicaciones. Los datos deben estar estructurados y almacenados de forma totalmente independiente de las aplicaciones que la utilizan” (Cobo, 2007). La diferencia de una Base de Datos respecto a otro sistema de almacenamiento de datos es que estos se almacenan de forma que no cumplan con tres requisitos básicos: no redundancia, independencia y concurrencia

Sistemas gestores de base de datos (SGBD)

“Un sistema de gestión de bases de datos es un software o conjunto de programas que permite crear y mantener una bases de datos. El SGDB actúa como interfaz entre los programas de aplicación (Usuarios) y el sistema operativo. El objetivo principal de un SGDB es proporcionar un entorno eficiente a la hora de almacenar y recuperar la información de la base de datos”. (Cobo, 2007)

Un SGBD se compone de un lenguaje de definición de datos, un lenguaje de manipulación de datos y de un lenguaje de consulta.

Entre algunos de los sistemas de gestión de base de datos libres tenemos Post-

greSQL, MySQL, MongoDB y con licencias Oracle y Microsoft SQL Server como los más conocidos.

Sistemas de gestión SQL (Sistemas de gestión de bases de datos relacionales)

Los sistemas de gestión de base de datos relacionales (RDBMS) se basan en una arquitectura relacional, utilizan SQL como lenguaje estándar para el modelado, la consulta y manipulación de los datos. Los datos se almacenan en tablas, las tablas se componen de filas y columnas. Cada columna almacena información sobre una propiedad determinada de la tabla o relación (representa un atributo) y cada fila (tupla) representa una ocurrencia de la relación. Las bases de datos relacionales poseen un esquema (estructura) relativamente estático ya que pocas veces se alteran (Alcalde, 2016). La normalización es un proceso característico del modelo relacional garantiza la integridad de los datos, eliminar la redundancia y las dependencias incoherentes, logrando una base de datos óptima (Microsoft, 2017). La mayor parte de los RDBMS cumplen con las características ACID para el manejo de transacciones. El lenguaje SQL (Lenguaje estructurado de consultas) se ha convertido en el estándar para las bases de datos relacionales. Sin embargo sobre esta base común se ha desarrollado diversas versiones como las de Oracle o la de Microsoft SQL Server. (Sarria, s.f.)

Sistemas de gestión NoSQL

NoSQL (“No Sólo SQL”), es una base de datos no relacional, maneja datos no estructurados, la información no se almacena en tablas como tal se almacenan de forma distinta, generalmente como clave-valor, posee un esquema dinámico por tanto se puede cambiar la estructura de la información sin tener que rediseñar todo nuevamente. NoSQL permite almacenar y recuperar gran cantidad de información en un mínimo de tiempo. No requiere operaciones join y tienen muy poco rendimiento en cuanto a la arquitectura ACID (Atomicity, Consistency, Isolation and Durability). NoSQL posee un esquema de datos dinámico la des-normalización de los datos evita el uso de operaciones join disminuyendo el costo de leer los datos. La normalización de los datos en el modelo relacional hace que las consultas avanzadas involucren unir varias tablas lo cual es costoso en términos de recursos de cómputo. Al poseer un esquema dinámico el rediseño de una base de datos NoSQL es mucho más fácil comparado con el modelo relacional. (Kollapur, 2012)

Sistemas de gestión de datos SQL vs NoSql

SQL	NoSql
Escalabilidad Baja	Escalabilidad Alta
Rendimiento bajo	Rendimiento alto
La fiabilidad es alta	La fiabilidad es baja
Utiliza propiedades ACID	Utiliza propiedades BASE
Implementación costosa	Costo de implementación moderados
Alta seguridad	Muy baja seguridad
Procesamiento de datos lento	Procesamiento de datos veloz

Tabla 2.1: Sistemas de gestión SQL vs NoSQL

Fuente: <http://www.kasperu.com/courses/BG011/II/default.htm> y Salazar, J. (2014)

Modelo de datos

Un modelo de datos es un mecanismo formal que determina la estructura lógica de una base de datos para representar, almacenar, organizar y manipular los datos de manera general y sistemática.

Es una “colección de conceptos bien definidos matemáticamente que ayudan a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo. Conceptualmente, una aplicación puede ser caracterizada por:

- Propiedades estáticas: entidades (u objetos), propiedades (o atributos) de esas entidades, y relaciones entre esas entidades.
- Propiedades dinámicas: operaciones sobre entidades, sobre propiedades o relaciones entre operaciones.
- Reglas de integridad sobre las entidades y las operaciones (por ejemplo, transacciones).

Así, un modelo de datos se distingue de otro por el tratamiento que da a estas tres categorías.” (Moreno, 2013)

Estadística

“La Estadística se ocupa de los métodos y procedimientos para recoger, clasificar, resumir, hallar regularidades y analizar los datos, siempre y cuando la variabilidad e incertidumbre sea una causa intrínseca de los mismos; así como de realizar inferencias a partir de ellos, con la finalidad de ayudar a la toma de decisiones y en su caso formular predicciones.

La estadística podemos clasificarla en estadística descriptiva y estadística inferencial.” (Rodríguez, s.f.).

Estadística inferencial

“La estadística inferencial está formada por procedimientos empleados para hacer inferencias acerca de características poblacionales, a partir de información contenida en una muestra sacada de esta población. El objetivo de la estadística inferencial es hacer inferencias (es decir, sacar conclusiones, hacer predicciones, tomar decisiones) acerca de las características de una población a partir de información contenida en una muestra.” (Mendenhall et al, 2010)

- “Población: conjunto de personas, entidades u objetos del cual se quiere saber algo que nos interesa para tomar una determinación acertada.
- Muestra: conjunto de medidas u observaciones tomadas a partir de una población dada; es un subconjunto de la población.”(García y Mateus, s.f.)

Modelos estadísticos

Un modelo estadístico es una ecuación matemática que reproduce los fenómenos que observamos de la forma más exacta posible. Para ello tiene en cuenta los datos suministrados y la influencia que el azar tiene en estas observaciones.

“Modelos discretos: Los modelos discretos, son modelos de probabilidad de variable aleatoria discreta. Los más importante son los modelos de BERNOUILLI (especialmente la distribución binomial y la distribución de Poisson).

Modelos continuos: Los modelos continuos, son modelos de probabilidad de variable aleatoria continua. Los más importante son , entre otros :

El modelo Uniforme. El modelo Exponencial . El modelo o distribución Normal.”
(Universidad de Valencia, s.f.)

Técnicas de muestreo

Para que se cumpla el principio de representatividad, debe prestarse atención al proceso de selección de los sujetos, utilizando una técnica de muestreo adecuada que aumente la probabilidad de obtener una muestra representativa.

“Muestreo aleatorio simple: Si una muestra de n elementos se selecciona de entre una población de N elementos, usando un plan muestral en el que cada una de las posibles muestras tiene la misma probabilidad de selección, entonces se dice que el muestreo es aleatorio y la muestra resultante es una muestra aleatoria simple

Muestra aleatoria estratificada: Un muestreo aleatorio estratificado comprende seleccionar una muestra aleatoria simple de cada uno de un número dado de subpoblaciones o estratos.

Muestra de conglomerados: es una simple muestra aleatoria tomada de los conglomerados disponibles en la población.

Muestra aleatoria sistemática 1 en k: Involucra la selección aleatoria de uno de los primeros k elementos de una población ordenada y luego la selección sistemática de cada k-ésimo elemento de ahí en adelante.” (Mendenhall et al, 2010)

Análisis estadístico

El análisis de datos es un proceso de inspeccionar, limpiar y transformar datos con el objetivo de resaltar información útil, lo que sugiere conclusiones, y apoyo a la toma de decisiones. El análisis de datos tiene múltiples facetas y enfoques, que abarca diversas técnicas en una variedad de nombres, en diferentes negocios, la ciencia, y los dominios de las ciencias sociales. Se colecciona data y se analiza para preguntar cuestiones, probar conjeturas o probar la invalidez de teorías. (Judd y McClelland, 1989)

2.5. Hipótesis

¿Los sistemas de gestión NoSQL inciden en los tiempos de respuesta de consulta de causas en la Función Judicial del Ecuador.?

2.6. Señalamiento de variables

Variable Independiente: Sistemas de gestión NoSQL

Variable Dependiente: Tiempos de respuesta de consulta de causas

Conector: Incide

CAPÍTULO III

METODOLOGÍA

3.1. Enfoque

La presente investigación se enmarca en un enfoque cuantitativo, debido a que se valorará los tiempos de rendimiento en el procesamiento de datos.

3.2. Modalidad Básica de la investigación

En esta investigación se utilizará las siguientes formas de investigación: bibliográfica y de campo. La primera permitirá la recolección de datos científicos que se encuentran en libros y documentos publicados en Internet, mientras que la investigación de campo permitirá obtener resultados sobre los tiempos de respuesta de los SGBD con procesamiento de consultas.

3.3. Nivel o tipo de investigación

La presente investigación se encuentra en los siguientes niveles: experimental y explicativa(deductivo)

Experimental ya que se llevarán a cabo experimentos en el laboratorio sobre diferentes escenarios y condiciones para así obtener los datos necesarios que permitirán medir la relación entre las variables del problema.

Explicativa permitirá buscar el por que de los hechos mediante el establecimiento de relaciones causa-efecto con los resultados obtenidos del experimento.

3.4. Población y muestra

Población

La población de la presente investigación estará formada por:

Descripción	Población	%
Administradores de base de datos	10	100 %
Total	10	100 %

Tabla 3.1: Población a ser investigada

Muestra

En virtud de que las población a ser investigada no pasa de los 100 elementos se trabajará con la totalidad del universo sin que sea necesario sacar muestras representativas.

3.5. Operacionalización de variables

3.5.1. Variable independiente

CONCEPTUALIZACIÓN	DIMENSIONES	INDICADORES	ITEMs BÁSICOS	TÉCNICAS E INSTRUMENTOS
Sistemas que facilitan la gestión de grandes cantidades de datos, que no cumplen con los esquemas entidad/relación es decir que no se imponen una estructura de datos en forma de tablas y relaciones entre ellas, por lo tanto estos sistemas son más flexibles y escalables.	Gestión de grandes cantidades de datos	Uso de sistemas de gestión NoSQL	¿Beneficios de los sistemas de gestión NoSQL para el procesamiento de datos masivos.?	Encuestas
	Estructura de datos	Rendimiento de los sistemas de gestión NoSQL	¿Es posible pasar los datos de un sistema de gestión relacional a uno no relacional.?	
		Adaptación de los sistemas de gestión NoSQL	¿Se puede integrar los sistemas de gestión NoSql con otras aplicaciones.?	
	Escalabilidad	Distribución y almacenamiento	¿Es importante un almacenamiento distribuido y replicado de los datos?	

Tabla 3.2: Operacionalización de la variable independiente

3.5.2. Variable dependiente

CONCEPTUALIZACIÓN	DIMENSIONES	INDICADORES	ITEMs BÁSICOS	TÉCNICAS E INSTRUMENTOS
Período de tiempo entre el cual el usuario ingresa una consulta y, el tiempo de procesamiento hasta obtener una respuesta.	Consultas Procesamiento de datos	Tipo de información Tiempo de procesamiento	¿Qué tipo de información se obtiene de las consultas? ¿Es posible optimizar las consultas que actualmente se realizan? ¿Es preferible tener respuestas rápidas en las consultas que preocuparse por el espacio de almacenamiento?	Encuestas

Tabla 3.3: Operacionalización de la variable dependiente

3.6. Recolección de información

Para la presente investigación se utilizará el siguiente plan de recolección de información Cuadro:3.4.

Preguntas básicas	Explicación
¿Para qué?	Para alcanzar los objetivos de la investigación
¿De qué personas u objetos?	Sujetos: administradores y expertos de bases de datos ... Objetos: sistema, base de datos.
¿Sobre qué aspectos?	Indicadores (matriz de operacionalización de variables)
¿Quién, Quiénes?	Ana Flores Pimbo
¿Cuándo?	Tiempo de desarrollo de la tesis
¿Dónde?	Función Judicial del Ecuador, departamento de Tecnologías de Información y Comunicación, planta central.
¿Cuántas veces?	Una para la obtención de datos para la Investigación
¿Qué técnicas de recolección?	Encuesta, entrevista, observación
¿Con qué?	Guías de preguntas a consultar
¿En qué situación?	Durante jornadas de elaboración de tesis

Tabla 3.4: Plan de recolección de la información

Elaborado por: Investigadora.

3.7. Procesamiento de la información

Los datos recogidos se transforman siguiendo ciertos procedimientos.

- Revisión crítica de la información recogida; es decir, limpieza de la información contradictoria, incompleta, no pertinente.
- Clasificar los datos recolectados según sus indicadores.
- Procesamiento de los datos para su tabulación.

Análisis e Interpretación de Resultados

- Análisis de los resultados estadísticos, destacando tendencias o relaciones fundamentales de acuerdo con los objetivos e hipótesis.
- Interpretación de los resultados, con apoyo del marco teórico, en el aspecto pertinente.
- Comprobación de hipótesis para la verificación estadística.
- Establecimiento de conclusiones y recomendaciones

CAPÍTULO IV

ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

4.1. Análisis e interpretación de los resultados

4.1.1. Encuesta

Se encuestó a personal que ejerce las funciones de administradores de base de datos y funciones de desarrollo de software.

En la siguiente Tabla 4.1 se muestra el número de encuestas realizadas.

Tipo	Número	Porcentaje
Administradores de bases de datos	10	100 %
Total	10	100 %

Tabla 4.1: Encuestas - Administradores de bases de datos

A continuación, las preguntas de la encuesta y sus respectivas respuestas tabuladas, analizadas e interpretadas:

1. ¿Cuál es su nivel de conocimiento acerca de las Bases de datos NoSQL?

Alternativo	Frecuencia	Porcentaje
Excelente	0	0 %
Bueno	2	20 %
Normal	2	20 %
Regular	2	20 %
Malo	4	40 %
TOTAL	10	100 %

Tabla 4.2: Nivel de conocimiento de las Bases de datos NoSQL

Análisis e Interpretación

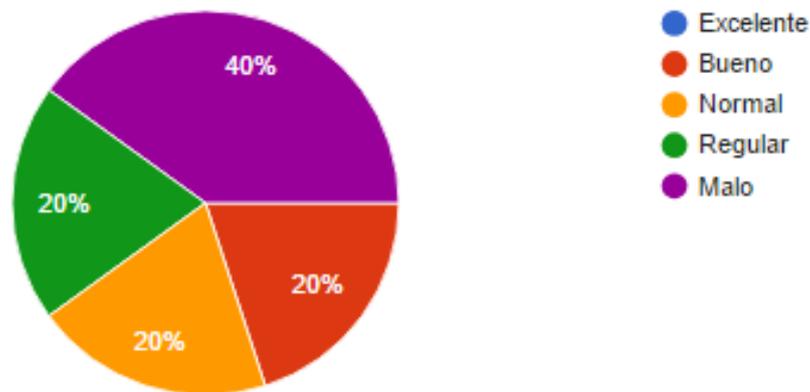


Figura 4.1: Nivel de conocimiento de las Bases de datos NoSQL

Las encuestas realizadas demuestran que referente al nivel de conocimiento sobre las bases de datos NoSQL; el 40 % no tiene conocimiento; el 60 % restante divide su conocimiento entre normal, bueno y regular. Del resultado obtenido se puede constatar que la mayoría de los encuestados tiene un conocimiento mínimo sobre las bases NoSQL.

- ¿Considera que la velocidad de procesamiento en los sistemas gestores de datos es una característica?

Alternativo	Frecuencia	Porcentaje
Muy importante	9	90 %
Importante	1	10 %
Relevante	0	0 %
Indiferente	0	0 %
Sin importancia	0	0 %
TOTAL	10	100 %

Tabla 4.3: Velocidad de procesamiento en los sistemas gestores de datos

Análisis e Interpretación

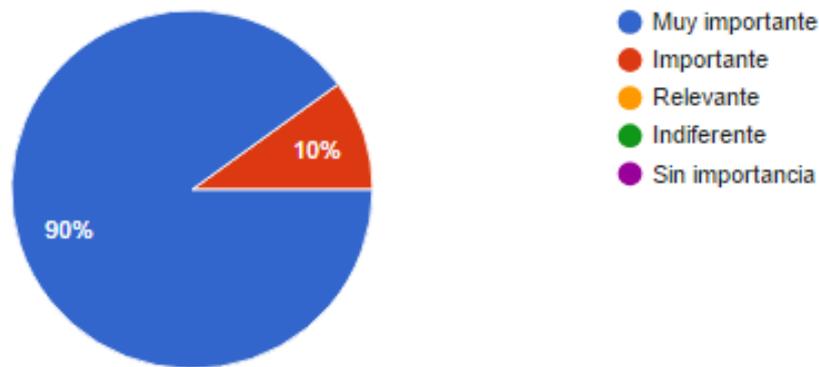


Figura 4.2: Velocidad de procesamiento en los sistemas gestores de datos

Las encuestas realizadas demuestran que referente a la velocidad de procesamiento de los sistemas gestores de datos; el 90 % lo define como muy importante; el 10 % lo considera importante. Del resultado obtenido se puede constatar que la mayoría de los encuestados ven esta característica como muy importante para el procesamiento de datos.

3. Se conoce que los sistemas NoSQL presentan un mejor rendimiento al procesar grandes cantidades de datos. Basados en este hecho. ¿Considera Usted que en su ámbito laboral los sistemas NoSQL son una opción?

Alternativo	Frecuencia	Porcentaje
Muy importante	2	20 %
Importante	1	10 %
Relevante	4	40 %
Indiferente	3	30 %
Sin importancia	0	0 %
TOTAL	10	100 %

Tabla 4.4: Sistemas NoSQL una opción de implementación

Análisis e Interpretación

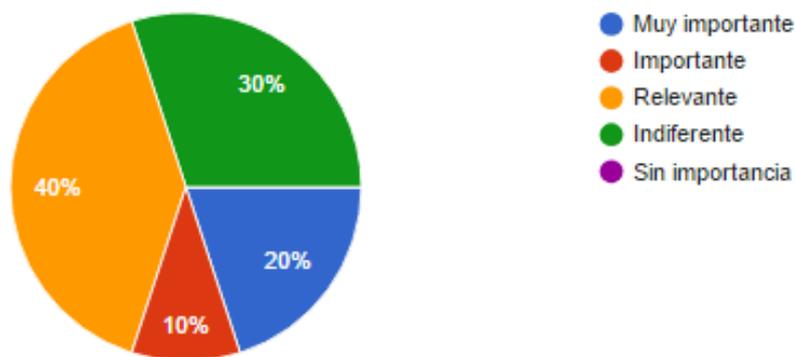


Figura 4.3: Velocidad de procesamiento en los sistemas gestores de datos

Las encuestas realizadas demuestran que referente a considerar a los sistemas NoSQL como una opción de implementación; el 40 % lo define como relevante; el 30 % indiferente; el 20 % muy importante; el 10 % como importante. Del resultado obtenido se puede constatar que la mayoría de los encuestados consideran la implementación de un sistemas NoSQL en su empresa.

4. Se conoce que los sistemas NoSQL no garantizan la integridad referencial de los datos. ¿Qué tan importante es este aspecto para las aplicaciones con las que usted trabaja?

Alternativo	Frecuencia	Porcentaje
Muy importante	6	60 %
Importante	2	20 %
Relevante	2	20 %
Indiferente	0	0 %
Sin importancia	0	0 %
TOTAL	10	100 %

Tabla 4.5: Integridad referencial en las aplicaciones

Análisis e Interpretación

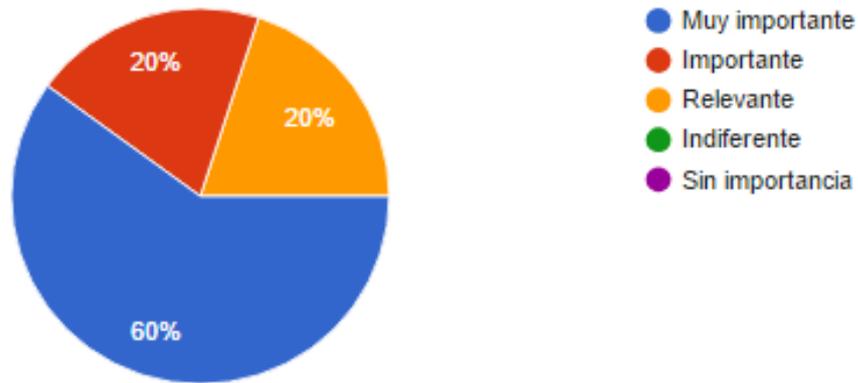


Figura 4.4: Integridad referencial en las aplicaciones

Las encuestas realizadas demuestran que referente a considerar a la integridad referencial como aspecto importante en las aplicaciones; el 60 % lo define como muy importante; el 20 % importante; el 20 % relevante. Del resultado obtenido se puede constatar que la mayoría de los encuestados consideran muy importante este aspecto en las aplicaciones.

5. Los sistemas NoSQL permiten un esquema libre en el registro de datos. Es decir no todos los registros comparten los mismos atributos. ¿Qué tan importante es este aspecto para las aplicaciones con las que usted trabaja?

Alternativo	Frecuencia	Porcentaje
Muy importante	4	40 %
Importante	2	20 %
Relevante	3	30 %
Indiferente	1	10 %
Sin importancia	0	0 %
TOTAL	10	100 %

Tabla 4.6: Esquema libre en las aplicaciones

Análisis e Interpretación

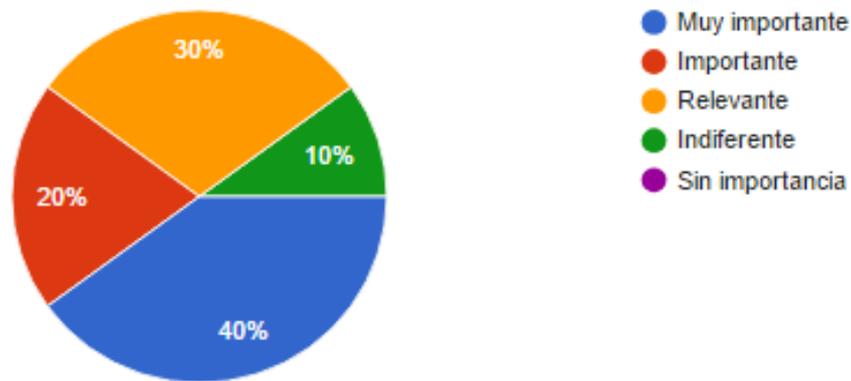


Figura 4.5: Esquema libre en las aplicaciones

Las encuestas realizadas demuestran que referente a considerar un esquema libre en el registro de datos; el 40% lo define como muy importante; el 30% como relevante; el 20% importante y el 10% indiferente. Del resultado obtenido se puede constatar que la mayoría de los encuestados consideran muy importante este aspecto en las aplicaciones.

- Uno de los pasos previos para utilizar un sistema NoSQL es la des-normalización de datos. ¿Cómo catalogaría esta tarea basado en la estructura actual de datos que usted maneja?

Alternativo	Frecuencia	Porcentaje
Muy Fácil	3	30 %
Fácil	2	20 %
Medio	2	20 %
Difícil	2	20 %
Muy Difícil	1	10 %
TOTAL	10	100 %

Tabla 4.7: Des-normalización de datos

Análisis e Interpretación

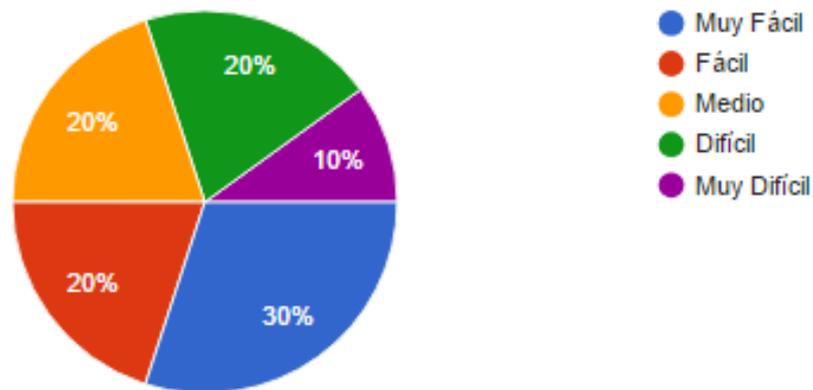


Figura 4.6: Des-normalización de datos

Las encuestas realizadas demuestran que referente a des-normalizar los datos; el 30 % lo considera muy fácil; el 20 % fácil; el 20 % medio; el 20 % difícil; y el 10 % muy difícil. Del resultado obtenido se puede constatar que la mayoría de los encuestados consideran muy fácil este aspecto en el paso de datos de una base relacional a una no relacional.

7. Una característica importante de los sistemas NoSQL es el escalamiento horizontal, esto involucra usar más servidores de forma paralela. ¿Qué tan importante es este aspecto para las aplicaciones con las que usted trabaja?

Alternativo	Frecuencia	Porcentaje
Muy importante	4	40 %
Importante	3	30 %
Relevante	1	10 %
Indiferente	2	20 %
Sin importancia	0	0 %
TOTAL	10	100 %

Tabla 4.8: Escalamiento horizontal

Análisis e Interpretación

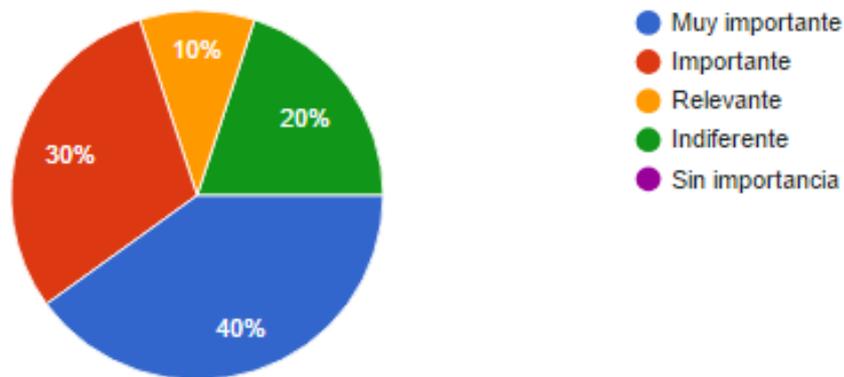


Figura 4.7: Escalamiento horizontal

Las encuestas realizadas demuestran que referente al escalamiento horizontal de los sistemas NoSQL; el 40 % lo considera muy importante; el 30 % importante; el 20 % indiferente; y el 10 % relevante. Del resultado obtenido se puede constatar que la mayoría de los encuestados consideran muy importante para las aplicaciones.

8. ¿Considera que aprender un lenguaje NoSQL es?

Alternativo	Frecuencia	Porcentaje
Muy importante	3	30 %
Importante	4	40 %
Relevante	2	20 %
Indiferente	1	10 %
Sin importancia	0	0 %
TOTAL	10	100 %

Tabla 4.9: Lenguaje NoSQL

Análisis e Interpretación

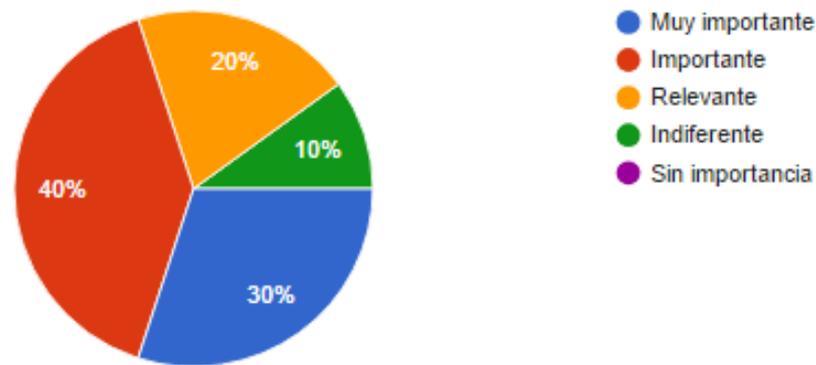


Figura 4.8: Lenguaje NoSQL

Las encuestas realizadas demuestran que referente a aprender un lenguaje NoSQL; el 40 % lo considera importante; el 30 % muy importante; el 20 % relevante; y el 10 % indiferente. Del resultado obtenido se puede constatar que la mayoría de los encuestados consideran importante el aprender un lenguaje NoSQL.

9. De los sistemas gestores de datos NoSQL por la forma de almacenamiento de la información cuál de las siguientes considera que es el más utilizado:

Alternativo	Frecuencia	Porcentaje
Bases de datos clave/valor	4	40 %
Bases de datos orientadas a columnas	1	10 %
Bases de datos documentales	3	30 %
Bases de datos en grafo	1	10 %
Ninguno	1	10 %
Otros	0	0 %

Tabla 4.10: Tipos de sistemas NoSQL

Análisis e Interpretación

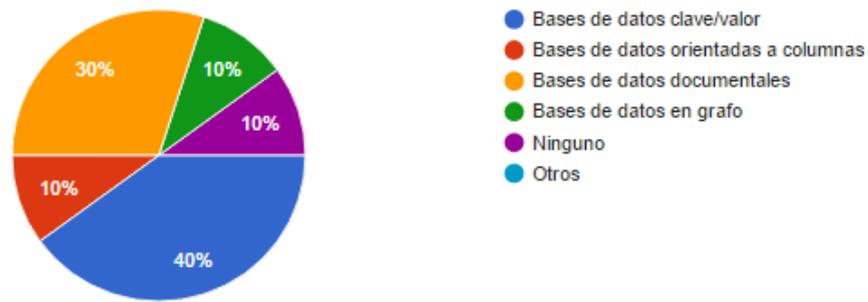


Figura 4.9: Tipos de sistemas NoSQL

Las encuestas realizadas demuestran que referente a los tipos de sistemas NoSQL más utilizados; el 40 % lo considera a las bases clave/valor; el 30 % documentales; el 10 % orientadas a columnas; el 10 % orientadas a grafos; y el 10 % ninguno. Del resultado obtenido se puede constatar que la mayoría de los encuestados consideran las bases de datos documentales como las más utilizadas.

10. De los siguientes sistemas gestores de datos NoSQL de cual usted ha escuchado hablar más:

Alternativo	Frecuencia	Porcentaje
Redis	0	0 %
Cassandra	1	10 %
MongoDB	8	80 %
Neo4j	1	10 %
Otro	0	0 %

Tabla 4.11: Sistemas NoSQL más conocidos

Análisis e Interpretación

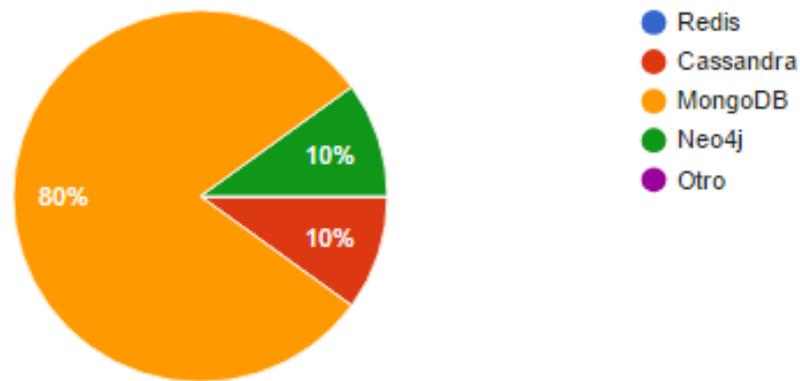


Figura 4.10: Sistemas NoSQL más conocidos

Las encuestas realizadas demuestran que referente a los sistemas NoSQL más utilizados; el 80 % considera que MongoDB es el más utilizado; el 10 % Cassandra; y el 10 % Neo4j. Del resultado obtenido se puede constatar que la mayoría de los encuestados consideran a MongoDB como las más utilizadas.

4.1.2. Verificación de hipótesis

4.1.2.1. Modelo lógico

Formulación de la hipótesis:

Hipótesis Nula(H_0).-¿Los sistemas de gestión NoSQL **NO** inciden en los tiempos de respuesta de consulta de causas en la Función Judicial del Ecuador.?

Hipótesis Alternativa(H_1).-¿Los sistemas de gestión NoSQL **SI** inciden en los tiempos de respuesta de consulta de causas en la Función Judicial del Ecuador.?

4.1.2.2. Modelo estadístico

Para la verificación de la hipótesis se escogió la prueba de Chi Cuadrado, por tanto se define la siguiente fórmula.

$$X^2 = \sum \left[\frac{(f_0 - f_e)^2}{f_e} \right]$$

X^2 = Chi cuadrado

f_0 = Frecuencia observada

f_e = Frecuencia esperada

Parámetros:

- Para realizar la matriz de tabulación cruzada se tomará en cuenta 4 preguntas de la encuesta realizada.
- Nivel de significación: $\alpha = 0.05$
- Grados de libertad: $gl = 12$

$$gl = (c-1) * (f-1)$$

$$gl = (5 - 1) * (4 - 1)$$

$$gl = 12$$

Frecuencias observadas

Preguntas	MUY IMPORTANTE	IMPORTANTE	RELEVANTE	INDIFERENTE	SIN IMPORTANCIA	TOTAL
¿Considera que la velocidad de procesamiento en los sistemas gestores de datos es una característica?	9	1	0	0	0	10
Se conoce que los sistemas NoSQL presentan un mejor rendimiento al procesar grandes cantidades de datos. Basados en este hecho. ¿Considera Usted que en su ámbito laboral los sistemas NoSQL son una opción?	2	1	4	3	0	10
Una característica importante de los sistemas NoSQL es el escalamiento horizontal, esto involucra usar más servidores de forma paralela. ¿Qué tan importante es este aspecto para las aplicaciones con las que usted trabaja?	4	3	1	2	0	10
¿Considera que aprender un lenguaje NoSQL es?	3	4	2	1	0	10
TOTAL	18	9	7	6	0	40

Table 4.12: Frecuencia observadas

Cálculo del Chi-Cuadrado

N_o	f_o	f_e	$(f_o - f_e)^2$	$(f_o - f_e)^2/f_e$
1	9	4.5	20.25	4.50
2	1	4.5	12.25	2.72
3	0	4.5	20.25	4.50
4	0	4.5	20.25	4.50
5	0	4.5	20.25	4.50
6	2	2.25	0.06	0.03
7	1	2.25	1.56	0.69
8	4	2.25	3.06	1.36
9	3	2.25	0.56	0.25
10	0	2.25	5.06	2.25
11	4	1.75	5.06	2.89
12	3	1.75	1.56	0.89
13	1	1.75	0.56	0.32
14	2	1.75	0.06	0.04
15	0	1.75	3.06	1.75
16	3	1.5	2.25	1.50
17	4	1.5	6.25	4.17
18	2	1.5	0.25	0.17
19	1	1.5	0.25	0.17
20	0	1.5	2.25	1.50
			X_c^2	38.70

Table 4.13: Cálculo de Chi-Cuadrado

Desición

$$X_c^2 = 38.70$$

$$X_t^2 = 21,0261$$

Con el grado de libertad 12 y 95% de confiabilidad se tiene que $X_c^2 > X_t^2$, lo que implica que se rechaza la Hipótesis Nula (H_0) y se acepta la Hipótesis Alternativa (H_1), es decir: “Los sistemas de gestión NoSQL **SI** inciden en los tiempos de respuesta de consulta de causas en la Función Judicial del Ecuador.”

Capítulo V

CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

De los resultados obtenidos de las encuestas realizadas se puede denotar que en nuestro ámbito laboral no es muy común el uso de sistemas NoSQL y el nivel de conocimiento con respecto a este es mínimo. Se puede entender este particular pues aún muchas empresas no han visto la necesidad de usar esta tecnología debido a que los sistemas relacionales aún satisfacen sus necesidades en cuanto almacenamiento y procesamiento de información.

La velocidad de procesamiento de los sistemas gestores de datos es elemental para el desempeño óptimo de los programas desarrollados por las empresas. Y el determinar que tipo de gestor utilizar es fundamental, se entiende que esta decisión depende del tipo de negocio, cantidad de datos que se estima almacenar, infraestructura y presupuesto disponible. Plantear las características que debe poseer el sistema a elegir es responsabilidad del administrador de base de datos conjuntamente con el equipo de desarrollo de software.

Los sistemas NoSQL presentan un mejor rendimiento al procesar grandes cantidades de datos pero dejan de lado algunas características como la integridad referencial y consistencia, se debe valorar los escenarios en los cuales plantear su uso exclusivo o considerar una arquitectura híbrida con sistemas relacionales.

Por otro lado los sistemas NoSQL solucionan los problemas de escalabilidad al distribuir los datos de forma horizontal; este almacenamiento distribuido descentraliza los datos distribuyendolos en varios nodos garantizando la disponibilidad. Este sistema de distribución aprovecha además todos los recursos hardware disponibles.

5.2. Recomendaciones

Es claro que los sistemas NoSQL a diferencia de los sistemas relacionales tienen poco tiempo en el mercado tecnológico, sin embargo es recomendable conocer y entender el manejo de soluciones innovadoras que respondan a necesidades actuales y futuras en las empresas e instituciones.

Debido a que los sistemas NoSQL dejan de lado algunas características con respecto a los sistemas relacionales es recomendable realizar un análisis de cuán ventajoso y en qué escenarios es positiva su implementación.

Los retos que conlleva la implementación de un sistema NoSQL puede depender del contexto donde se desea aplicar. El diseño de una estructura no relacional o la migración de una estructura relacional a una no relacional puede ser tan fácil o complicada como el negocio sea. Es recomendable seleccionar un sistema gestor de datos que se adapte a las necesidades de la empresa, para lo cual definir las características que este debe cumplir es muy importante.

Los RDBMS se basan en una arquitectura relacional, utilizan SQL como lenguaje estándar para el modelado, la consulta y manipulación de los datos. El cambio a una base de datos NoSQL conlleva utilizar un lenguaje diferente a SQL. Es recomendable según la experiencia del administrador de base de datos y desarrolladores de software estimar la curva de aprendizaje y asegurar un correcto funcionamiento y administración del sistema de gestión.

Capítulo VI

PROPUESTA

6.1. Datos informativos

Tema

Los sistema de gestión NoSQL y su aplicación para la mejora de los tiempos de respuesta de las consultas de causas en la Función Judicial del Ecuador.

Institución ejecutora

Consejo de la Judicatura - Función Judicial del Ecuador

Beneficiarios

Equipo técnico responsable del desarrollo del sistema de Consulta de Causas.

Ubicación

Provincia: Pichincha

Ciudad: Quito

Dirección: Edificio Plaza 2000, Av. Francisco Salazar y 12 de Octubre

Equipo técnico responsable:

- Investigadora: Ana Virginia Flores Pimbo

Tiempo estimado para la ejecución

Inicio: 1 de Abril de 2017

Fin: 8 de Mayo de 2017

6.2. Antecedentes de la propuesta

El sistema de consulta de causas de la Función Judicial es el encargado de presentar a la ciudadanía toda la información referente a un proceso judicial; por ende tanto la disponibilidad y agilidad en los tiempos de respuesta del sistema es trascendental.

A partir de la investigación realizada, conocemos que en la actualidad existen múltiples opciones en cuanto a sistemas gestores de datos que garantizan tiempos de respuesta mínimos en el proceso de grandes consultas, estos sistemas gestores de datos son denominados NoSQL.

NoSQL está orientado a almacenar datos no estructurados y no utilizan modelos de tablas como las bases de datos relacionales haciéndolas más flexibles y optimas al consultar grandes cantidades de datos. Sin embargo las bases de datos NoSQL no necesariamente reemplazan a las SQL cada una de ellas tiene un uso específico y es posible crear sistemas híbridos.

6.3. Justificación

La relevancia de esta investigación radica en verificar si un sistema de gestión NoSQL mejora los tiempos de respuesta de las consultas de causas en la Función Judicial del Ecuador.

Al mismo tiempo que permitirá ampliar el conocimiento sobre otro tipo de sistemas que no son modelos relacionales los cuales manipulan grandes volúmenes de datos, donde su principal característica es la rapidez de almacenamiento y recuperación de grandes cantidades de información. Enfocándonos así en buscar nuevas alternativas para solventar a los diferentes contartiempos de almacenamiento y procesamiento de información que se presentan en la actualidad.

Finalmente se debe indicar que el presente estudio contemplará un análisis entre dos bases de datos una relacional y no relacional el cual permitirá medir el rendimiento de cada uno de los sistemas, con estos resultados es posible conocer cuan óptimas son las bases de datos.

6.4. Objetivos

6.4.1. Objetivo general

Demostrar que un sistema de gestión NoSQL mejora los tiempos de respuesta de las consultas de causas en la Función Judicial del Ecuador.

6.4.2. Objetivos específicos

- Analizar el proceso actual establecido por la Función Judicial para extraer y presentar la información a través del sistema de Consulta de Causas.
- Seleccionar un sistema de gestión de datos NoSQL en base a las necesidades del aplicativo de consulta de causas y de la Función Judicial.
- Realizar un análisis comparativo entre el rendimiento proporcionado por un sistema de gestión de datos NoSQL y el sistema utilizado para la consulta de causas en la Función Judicial.

6.5. Análisis de factibilidad

6.5.1. Tecnológica

La propuesta tecnológica es factible ya que se cuenta con los elementos necesarios para la preparación de los entornos de prueba.

6.5.2. Operativa

El contar con resultados acerca del rendimiento de los sistemas de gestión NoSQL y el sistema de base de datos actual que utiliza la aplicación de consulta de causas, es un aporte importante para la institución pues se cuenta con datos validos que permitirán a las autoridades tomar las mejores decisiones con respecto a la solución tecnológica para mejorar el rendimiento de la aplicación.

6.5.3. Económica

Las herramientas a utilizar en los entornos de prueba son libres y además la institución cuenta con licencias de Windows Server 2008, Microsoft SQL Server 2008 y Microsoft SQL Server 2012.

6.6. Fundamentación

6.6.1. Científico Técnica

6.6.1.1. Arquitectura actual del sistema de consulta de causas en la Función Judicial

Es importante considerar un análisis de la arquitectura actual tanto de desarrollo, despliegue y base de datos del sistema de consulta de causas para comprender su funcionamiento y la interacción entre los elementos que la componen.

Arquitectura de desarrollo

El sistema de consulta de causas es una aplicación web, su desarrollo está dividido en capas, actualmente se tienen las siguientes:

- Capa de presentación
- Capa de lógica de negocios
- Capa de acceso a datos

La Figura 6.6.1.1 representa la arquitectura utilizada para el desarrollo de aplicaciones web y aplicada para el desarrollo de consulta de causas. Para el desarrollo del aplicativo en mención se utiliza las siguientes herramientas:

- Interfaz de usuario : Java Server Faces (JSF) y Primefaces
- Lógica de negocios y acceso a datos: JAVA
- Base de datos: Microsoft SQL Server 2008 R2
- Servidor de aplicaciones: JBOSS

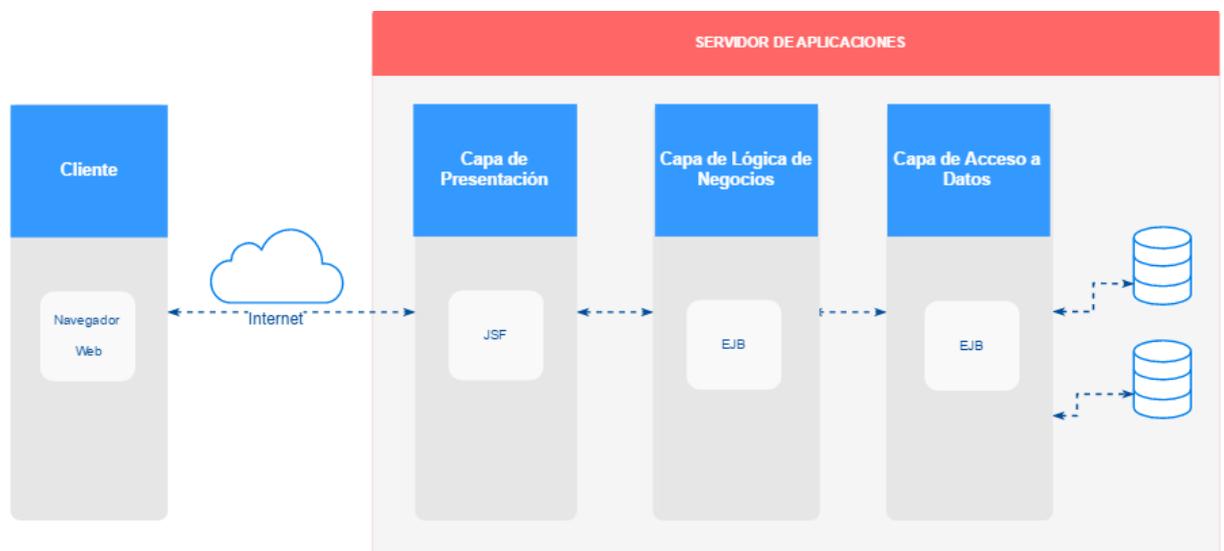


Figura 6.1: Arquitectura aplicación web - Consulta de Causas

Despliegue del aplicativo

El despliegue del aplicativo de consulta de causas se lo realiza sobre un servidor de aplicaciones JBOSS, debido a la alta concurrencia de usuarios que tiene el aplicativo la institución ha optado por balancear la carga a dos nodos donde se encuentran

desplegados los archivos .war que contiene los archivos del aplicativo.

Cada nodo tiene configurado el acceso a datos (dataSource) a una instancia distinta de base de datos. La Figura 6.6.1.1 representa la forma de despliegue del aplicativo y su interacción con la base de datos.

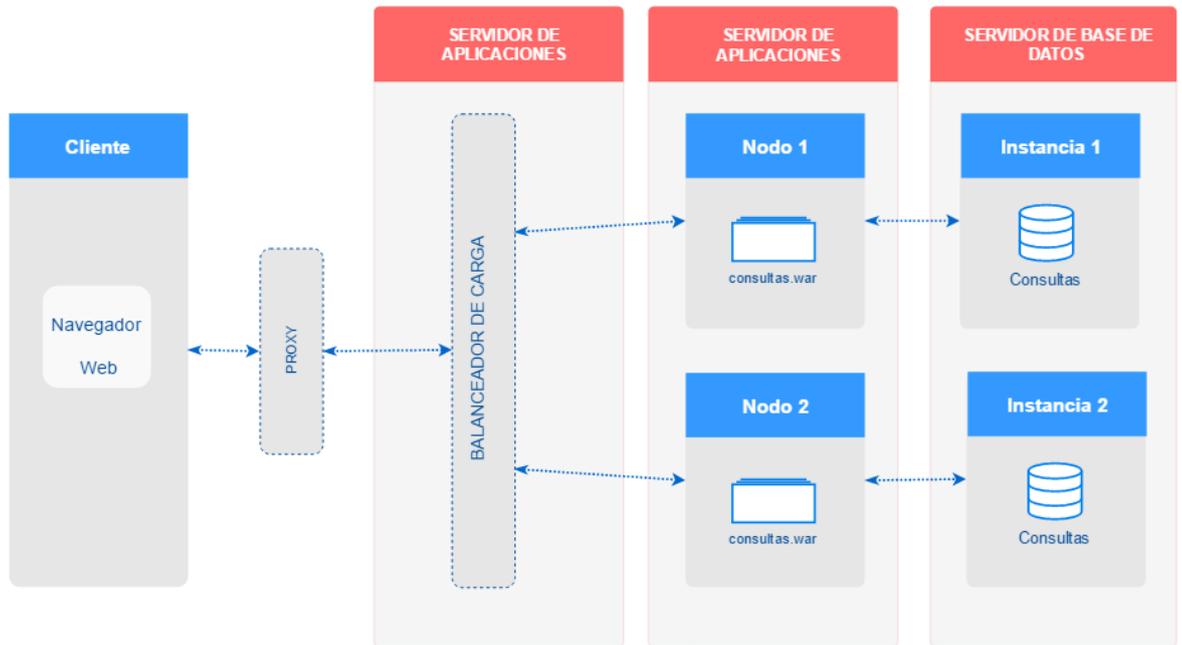


Figura 6.2: Arquitectura de despliegue - Consultas de Causas

En vista de que el aplicativo web solo envi a peticiones de lectura al servidor de base de datos y por su alta concurrencia; para este aplicativo en particular se ha implementado una arquitectura diferente. Se trabaja sobre una replica en l nea de la base de ProcesosJudiciales ubicada en dos instancias de base de datos distinta.

Base de datos

En cada una de las instancias existe la base de datos llamada Consultas que contiene dos tablas planas TablaActor y TablaDemandado donde se almacena los depurados de las causas seg n el tipo de litigante, el llenado de estas tablas se lo hace mediante consultas a la base de replica (por medio de s nimos) y la ejecuci n de jobs de cargas incrementales que procesan y limpian la data a ser almacenada, la consulta de los datos de litigantes, dependencias jurisdiccionales y actuaciones judiciales del proceso se las consulta directamente por medio de s nimos a la base de replica ProcesosJudiciales. La Figura 6.6.1.1 representa la arquitectura de base de datos utilizada.

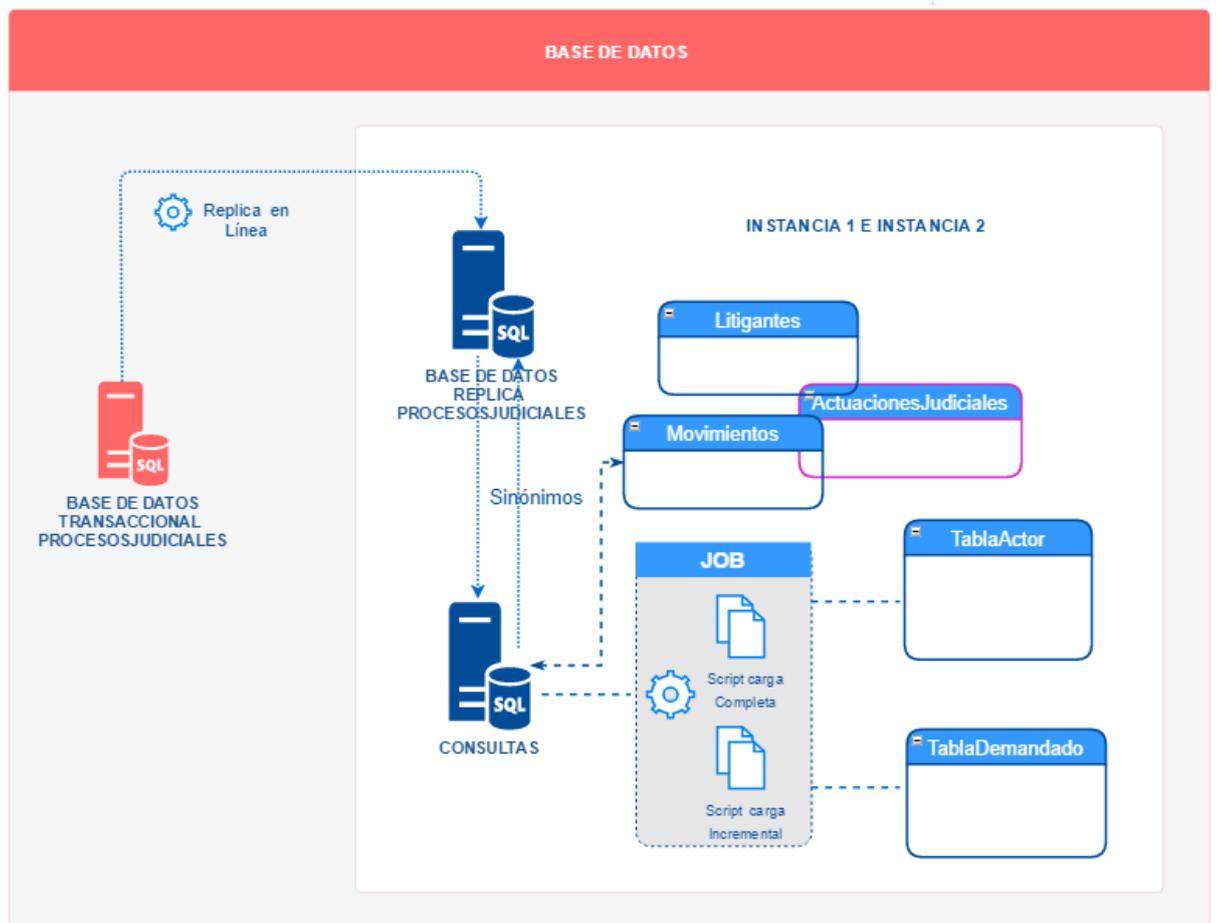


Figura 6.3: Arquitectura de base de datos - Consulta de Causas

Acceso y uso del aplicativo

Para acceder al aplicativo se realiza a través de la página web de la Función Judicial o por medio de la url `consultas.funcionjudicial.gob.ec/informacionjudicial/public/informacion.jsf`

El aplicativo visualiza la información en tres secciones:

1. Búsqueda de procesos: por actor y/o demandado, número de proceso, se puede utilizar más filtros.
2. Movimientos del proceso: muestra los movimientos de la causa por las diferentes dependencias jurisdiccionales.
3. Detalle del proceso: Información de las actuaciones judiciales realizadas en las dependencias jurisdiccionales.

La Figura 6.4 visualiza el proceso de búsqueda en el sistemas de consulta de causas. En general el proceso de consulta se realiza con normalidad, los tiempos de respuesta pueden variar dependiendo de la cantidad de coincidencias que tenga el parámetro ingresado. Sin embargo y particularmente en el proceso de la obtención de las actuaciones judiciales se ha observado que lleva mayor tiempo en procesar la consulta, no en todos los casos pero sí en aquellos procesos judiciales que tienen muchas actuaciones judiciales con un texto muy amplio.

Figura 6.4: Proceso de búsqueda - Consulta de Causas

Número de usuario que acceden al aplicativo Consulta de Causas

Alrededor de 11.434 usuarios cada día las 24 horas interactúan con la página de consulta entre ellos nuevos usuarios y usuarios recurrentes. Para inspeccionar la concurrencia de usuarios y las horas en las que el aplicativo es más utilizado, se ha

tomado al azar el 20 de Marzo del 2017 como fecha de análisis.

Por medio de la herramienta Google Analytics¹ se obtuvieron los siguientes datos
Tabla 6.1:

Hora	Número de usuario
00:00:00	67
01:00:00	30
02:00:00	18
03:00:00	14
04:00:00	17
05:00:00	28
06:00:00	76
07:00:00	208
08:00:00	647
09:00:00	929
10:00:00	1019
11:00:00	1038
12:00:00	926
13:00:00	714
14:00:00	815
15:00:00	893
16:00:00	813
17:00:00	702
18:00:00	537
19:00:00	505
20:00:00	461
21:00:00	451
22:00:00	332
23:00:00	194
Total:	11434

Tabla 6.1: Usuarios por hora - Consulta de Causas

En la Figura 6.5 se puede identificar las horas en las cuales hay más demanda. El análisis indica que entre las 9:00 AM y 12:00 PM y entre las 14:00 AM y 16:00 PM existe mayor concurrencia de usuarios. Siendo 1038 el valor más alto a las 11:00AM. Cabe recalcar que estos valores varían constantemente dependiendo del año, mes, día y hora.

¹Google Analytics. URL: https://es.wikipedia.org/wiki/Google_Analytics Visitado el 27/03/2017

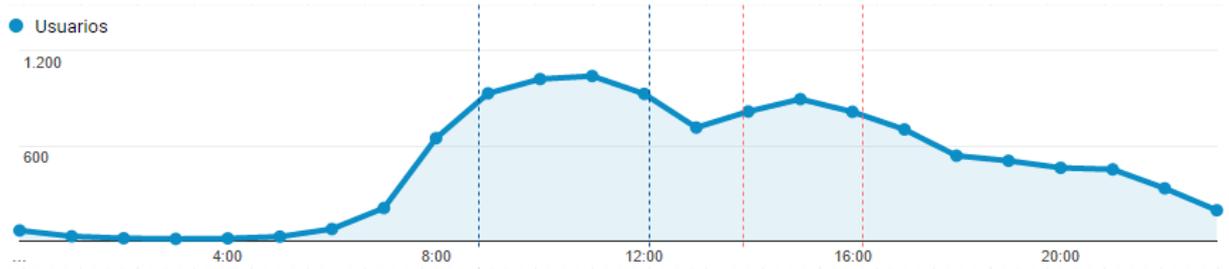


Figura 6.5: Horas de mayor demanda - Consulta de Causas

6.6.1.2. Análisis y selección del sistema de gestión de datos NoSQL

Las bases de datos NoSQL son sistemas distribuidos de almacenamiento, creadas para manejar grandes cantidades de datos de manera rápida y eficiente. Tienen un alto rendimiento, gran capacidad de almacenamiento, alta escalabilidad (horizontal) y alta disponibilidad. Utilizan un modelo (estructura) de datos simple, un lenguaje de consulta simple y una mínima consistencia e integridad de los datos. NoSQL, maneja datos no estructurados, la información no se almacena en tablas como tal se almacenan de forma distinta, generalmente como clave-valor, posee un esquema dinámico por tanto se puede cambiar la estructura de la información sin tener que re-diseñar todo nuevamente.

Los sistemas NoSQL en general se clasifican en los siguientes grupos cada una diseñada para manejar diferentes tipos de información como son: las documentales, clave-valor, multivalor, orientadas a columnas y a grafos, nativas en XML y objetos, tipo tabla y bases híbridas. El uso de un tipo u otro de los sistemas NoSQL depende del problema a resolver ya que cada tipo está optimizado para una aplicación específica. Como los más mencionados según su modelo de datos podemos hablar de los siguientes (Kaur y Rani, 2013) (Kumar et al, 2014) (Grolinger et al, 2013):

Clave-Valor (Key-Value Stores).- Estos sistemas tiene un modelo simple de almacenamiento basado en pares de clave y valor. Una clave única identifica el valor dentro del almacén de datos. Debido a que poseen un esquema libre se puede almacenar tipos de datos enteros, cadenas, un array o un objeto, son muy eficientes en el almacenamiento de datos distribuidos pero no son adecuados cuando se requiere relaciones o estructuras. No soporta verificación intrínseca de la integridad de datos, claves foráneas y disparadores. Delegando a las aplicaciones cliente la validación de datos como integridad y referencia cruzada. En ciertos sistemas NoSQL nos permiten almacenar claves-valor duplicados, en otros no, o se pueden configurar para que exista duplicado. La desventaja del manejo sencillo de datos es que no puede

hacer consultas de rango (consultas que cumplan alguna condición) ya que no son directas. A pesar de su coste de uso, este tipo de bases de datos son populares por los sistemas basados en la implementación del paradigma Cloud Computing como Amazon SimpleDB. Entre los sistemas clave-valor tenemos a: Redis, Memcached, Voldemort, Riak KV y otros.

Almacenamiento de Columnas (Column stores).- En esta estructura de almacenamiento cada fila está compuesta por una familia de columnas y diferentes filas puede tener columnas. De forma similar a los sistemas clave-valor, la clave de la fila se asemeja a la clave y el conjunto de columnas se asemeja al valor, representado por la clave de la fila. Sin embargo, cada familia de columnas actúa como clave para una o más columnas que la contiene, donde cada columna contiene un par de nombre-valor. Estos sistemas permiten la indización de columnas y las consultas son mucho más rápidas que los sistemas clave-valor. Son utilizados en sistemas de almacenes de datos e inteligencia de negocios, sistemas de gestión de relaciones de clientes, catálogos de bibliotecas, sistemas de consulta ad-hoc. Entre los sistemas orientados a columnas tenemos a: Cassandra, HBase, Hypertable, Google Cloud Bigtable y otros.

Documentales (Document stores).- Está diseñada para gestionar datos semi-estructurados o no estructurados apropiadas para aplicaciones web. Su estructura interna tiene una definición abstracta de un documento, que es equivalente a una tabla en una base de datos relacional. Estos sistemas gestionan datos que pueden representarse como documentos, que son estructuras jerárquicas autodescriptivas que pueden contener objetos anidados y atributos de lista y además no requieren adherencia a un esquema fijo. El almacenar y recuperar todos los datos como una sola unidad ayuda en el rendimiento y la escalabilidad, la consulta se basa en el contenido del documento diferenciando así de los sistemas clave-valor donde los valores pasan a un segundo plano, facilitando así las operaciones complejas. Los datos se almacenan en formatos de estructuras como JSON (JavaScript Object Notation), XML o derivados y ofrecen varias API para varios lenguajes. Este tipo de almacenamiento permite indizar las claves primarias de los documentos así como a su contenido. Existen aplicaciones que proporcionan servicios utilizando interfaces REST basadas en representaciones JSON que pueden ser mapeadas directamente al almacén de datos. Entre los sistemas documentales podemos mencionar a: MongoDB, CouchDB, Amazon DynamoDB, RavenDB y otros.

Grafos (Graph databases).- Están diseñadas para representar la información como

nodos de un grafo y sus relaciones con las aristas del mismo, de manera que se pueda usar teoría de grafos para recorrer la base de datos. La información en un grafo se modela utilizando tres elementos básicos: el nodo (vértice), la relación (arista) con dirección y tipo (etiquetado y dirigido) y la propiedad (atributo). Es un modelo completamente diferente a los sistemas clave-valor, columnar o documental, las bases de datos tipo grafo permiten almacenar eficientemente las relaciones entre los diferentes nodos de datos por tanto son muy eficientes en atravesar relaciones entre diferentes entidades. Son utilizados en aplicaciones de análisis de patrones, búsqueda de rutas, análisis de dependencias, redes sociales. Algunas bases de datos tipo grafo como Neo4J son totalmente compatibles con ACID. Sin embargo, no son tan eficientes como otros sistemas NoSQL en escenarios distintos a la manipulación de gráficos y relaciones. Entre las bases de datos más utilizadas son: Neo4J, OrientDB, Titan, ArangoDB y otros.

Selección de la herramienta

Actualmente el sistema de consulta de causas se encuentra soportado sobre una base de datos relacional siendo esta Microsoft SQL Server 2008 R2; basados en la teoría y con conocimiento de como se encuentra estructurada la data que sería migrada, nos enfocaremos en los sistemas **NoSQL documentales**, con los siguientes argumentos:

- Permite consultas complejas al conjunto de datos.
- Son estructuras jerárquicas y proporciona medios para recuperar registros en función de su contenido real.
- Permite anidar estructuras más complejas.
- Posee una estructura semántica y su uso es simple.

Luego de haber definido la categoría del sistema NoSQL a usar, se procedió a seleccionar el sistema gestor de datos siendo este **MongoDB 3.4** basado en los siguientes argumentos:

1. Posee una segmentación marcada en el mercado tecnológico: MongoDB posee una alta popularidad entre los sistemas de su categoría la lista que ofrece DB-Engines² en el mes de abril de 2017 lo ubica entre los tres primeros, la Figura 6.6 refleja este hecho.

²DB-Engines. Ranking of Document Stores. URL: <http://db-engines.com/en/ranking/document+store>

2. Posee una versión Community Edition, licencia Open Source: Solución que encaja con los artículos del Decreto Ejecutivo 1014³ y el Código Ingenios, que establece el uso de programas que permitan su acceso a los códigos fuentes y que sus aplicaciones puedan ser mejoradas.
3. Soporte de la comunidad y acceso a documentación en línea.
4. Soporte de empresas nacionales o de latinoamérica: este aspecto es relevante pues aún que se cuente con el soporte de la comunidad es necesario el asesoramiento y experiencia de personal certificado.

La Figura 6.6 refleja la tendencia de popularidad de cada uno de los sistemas en la categoría de NoSQL documentales.

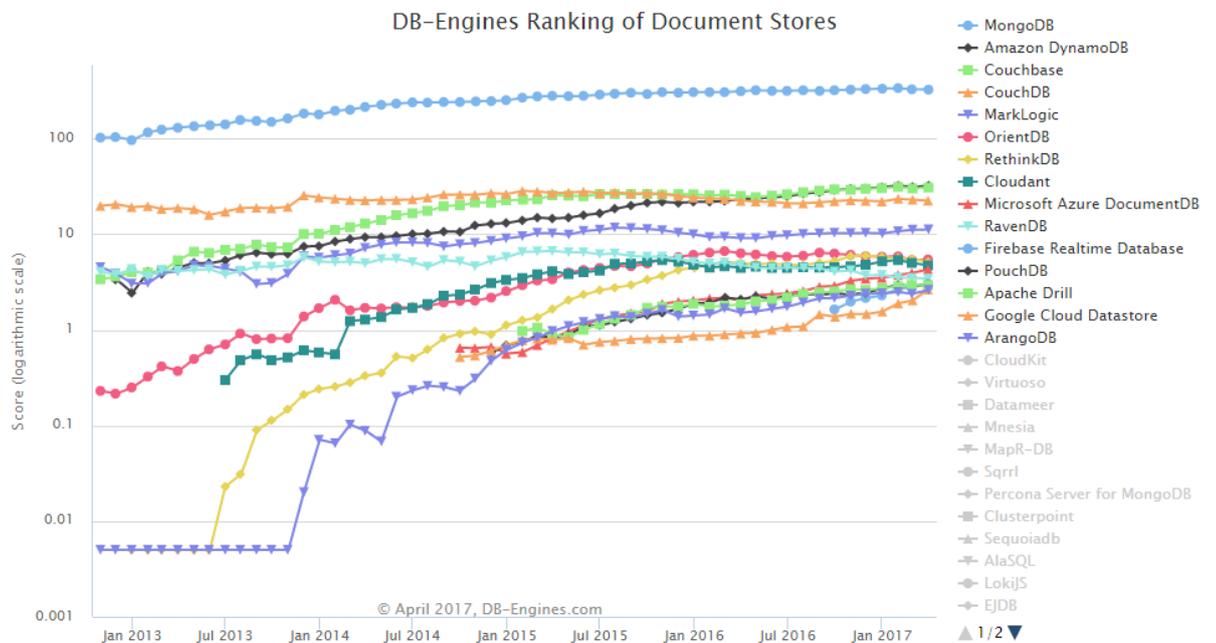


Figura 6.6: DB-Engines - Ranking of Document Stores
Fuente: http://db-engines.com/en/ranking_trend/document+store

MongoDB

Es un proyecto open source desarrollado por 10gen⁴ en octubre de 2007. En la

³Decreto Ejecutivo 1014. URL: <https://softwarelibre.conocimiento.gob.ec/documentos-tecnicos/decreto-1014/>

⁴10gen - MongoDB, Inc. URL: <https://www.mongodb.com/press/10gen-announces-company-name-change-mongodb-inc>

actualidad es una de las bases más utilizadas según DB-Engine, Database Trends & Applications⁵ y Gartner's⁶. Está implementado bajo el lenguaje de programación C++. No posee el concepto de transacciones ni de triggers. Almacena los datos en documentos. Estos documentos son almacenados en BSON, que es una representación binaria de JSON. La principal diferencia con respecto a las bases clave-valor es que los valores en MongoDB pueden ser otros documentos, arrays o incluso arrays de documentos. Los documentos JSON de una misma colección no necesariamente se apegan al mismo esquema.

En MongoDB los documentos JSON se agrupan en una colección. Una colección es el equivalente a una relación en un SGBD relacional. Un documento equivale a una fila de una relación y una clave de un documento equivale a un atributo de una relación. En la Figura 6.7 se establece un paralelo entre los términos de una base de datos relacional y una base de datos documental MongoDB. (Moreno et al, 2016) (Grolinger et al, 2013) (Herranz, 2014)

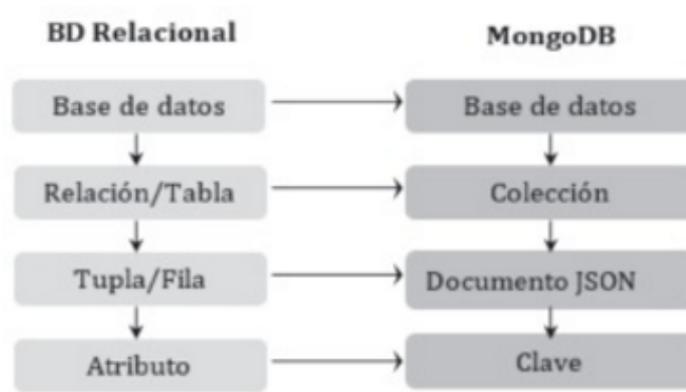


Figura 6.7: Relación de términos entre una base de datos Relacional y una Documental: MongoDB

Fuente: (Moreno et al, 2016)

Generalidades:

- "Indexado. MongoDB soporta índices secundarios, permitiendo una buena aceleración en muchas queries, pudiendo indexar cualquier campo de la base de datos.
- Agregación. MongoDB viene con muchas funciones predefinidas que facilitan en gran medida el tratamiento de las colecciones almacenadas.

⁵Database Trends & Applications. URL: <http://www.dbta.com>

⁶Gartner's. URL: www.gartner.com

- Replicación y Balanceo de Carga. MongoDB utiliza un sistema de replicación empleando la arquitectura maestro – esclavo con el resto de máquinas. Además, hace uso de un tipo de particionado horizontal llamado sharding. Estas dos herramientas de escalado hacen de mongo una herramienta poderosa cuando se trata de crecer mucho en poco tiempo." (Herranz, 2014)

6.7. Metodología

Para el desarrollo de la propuesta se considerará los siguientes pasos:

1. Preparación de ambientes de pruebas
2. Extracción, transformación y carga de datos a MongoDB 3.4
3. Optimizar de base de datos
4. Recolección de datos
5. Análisis de resultados
6. Conclusiones
7. Recomendaciones

6.7.1. Preparación de ambientes de prueba

Hardware y software a utilizar

Para los entornos de pruebas y posterior evaluación de las herramientas se ha utilizado el siguiente software ver Tabla 6.2.

Sistema Operativo	Microsoft Windows Server 2008 R2 Enterprise 64-bits
	CentOS 7 64-bits
Sistemas Gestores de Datos	Microsoft SQL Server 2012 (SP3)
	MongoDB Community Edition 3.4
Herramienta de evaluación	JMeter 2.13

Tabla 6.2: Software para entornos de prueba

Proxmox Virtual Environment, fué utilizado como entorno de virtualización para gestionar las máquinas virtuales que servirán para realizar las pruebas de carga. Cada uno de los sistemas gestores de datos se instalaron en una máquina virtual con las siguientes características Tabla 6.3. Además se creó una estación cliente desde la cual se realizará las peticiones de consulta para validar el rendimiento y comportamiento

de los sistemas gestores de datos, para esto se utilizará la herramienta *JMeter*⁷ la Figura 6.8 representa la arquitectura utilizada para realizar las pruebas.

Hardware	Descripción
Procesador	Intel Core 2 DUO 8 Cores
Memoria RAM	12 GB
Disco duro	800 GB

Tabla 6.3: Hardware para los entornos de prueba

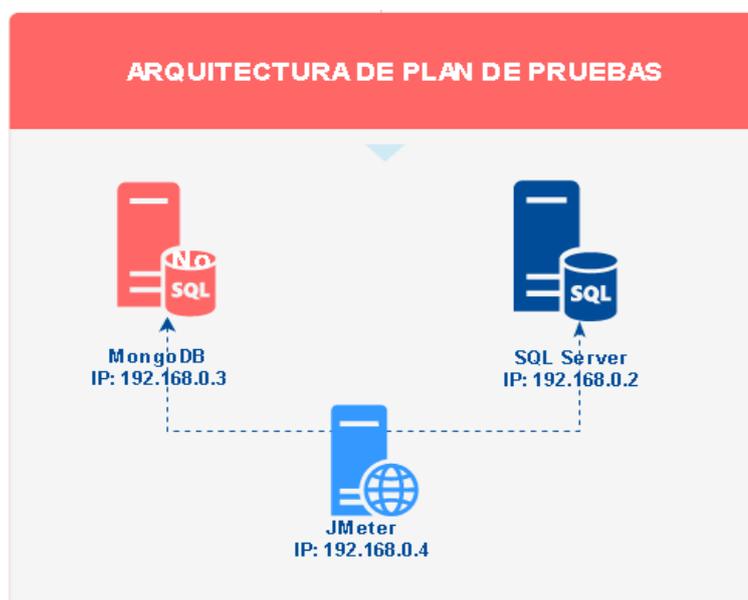


Figura 6.8: Arquitectura del escenario de pruebas

Instalación de herramientas

Se procedió a preparar 2 entornos de evaluación:

- Microsoft SQL Server 2012 y MongoDB 3.4 sobre Microsoft Windows Server 2008 R2 Enterprise 64-bits.
- MongoDB 3.4 sobre CentOS7 64-bits.

Instalación Microsoft SQL Server 2012 y MongoDB 3.4 sobre Microsoft Windows Server 2008 R2 Enterprise 64-bits.

⁷Apache JMeter. URL:http://jmeter.apache.org/usermanual/component_reference.html Visitado el 27/05/2017

Microsoft SQL Server 2012

Para el proceso de instalación se procedió a seguir los pasos descritos en la documentación de Microsoft. Sección, Instalar SQL Server 2012⁸. Los componentes que debemos instalar son:

- Database Engine Services
- Integration Services
- Management Tools-Complete
- Client Tools SDK
- SQL Client Connectivity SDK

MongoDB Community Edition 3.4

Para el proceso de instalación se siguió los pasos descritos en la documentación de MongoDB⁹ de su sitio oficial. Sección, Install MongoDB Community Edition on Windows. Una vez instalado MongoDB, se procedió a levantar el servicio por medio del PowerShell de Windows ejecutando el siguiente comando:

```
"C:\Program Files\MongoDB\Server\3.4\bin\mongod.exe"
```

Una vez instalado el sistema mongoDB procedemos a crear las base de datos y colección.

1. Verificar que el servicio de mongoDB esté iniciado, ver Figura 6.7.1.
2. Acceder a la consola de mongoDB, para esto se utilizó de la herramienta Robomongo¹⁰ la cual permite administrar gráficamente las bases de datos y se creó una nueva conexión hacia el servidor de base de datos, ver Figura 6.10. Crear la base de datos sobre la cual se migró la data desde SQL Server 2012 y realizó las pruebas, en Robomongo sobre la conexión creada clic derecho y seleccionar Create Database, ver Figura 6.11.
3. Crear la colección donde se almacenará los documentos JSON, utilizando el siguiente comando
> `db.createCollection("actuacionesjudiciales")`

⁸ Microsoft. Instalar SQL Server 2012. URL:[https://technet.microsoft.com/es-es/library/bb500395\(v=sql.110\).aspx](https://technet.microsoft.com/es-es/library/bb500395(v=sql.110).aspx)

⁹ mongoDB. MONGODB MANUAL 3.4 URL: <https://docs.mongodb.com/manual/introduction/>

¹⁰ Robomongo is Robo 3T URL:<https://robomongo.org/>

La Figura 6.12 visualiza la creación de la base de datos y la colección.

```
PS C:\Program Files\MongoDB\Server\3.4\bin> .\mongod.exe
2017-07-17T20:46:31.580-0700 I CONTROL [initandlisten] MongoDB starting : pid=3308 port=27017 dbpath=C:\data\db\ 64-bit
host=WIN-CAGMML1UE0S
```

Figura 6.9: MongoDB starting

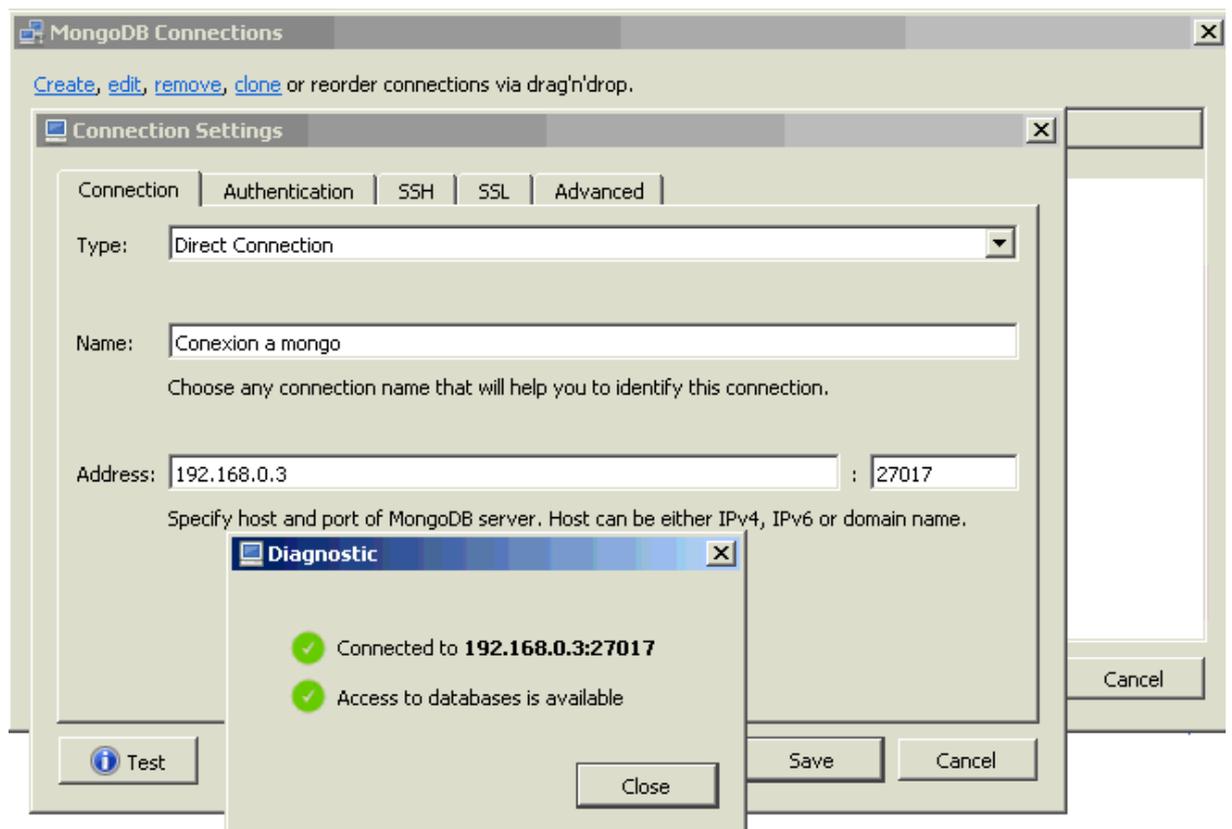


Figura 6.10: Conexión a Robomongo

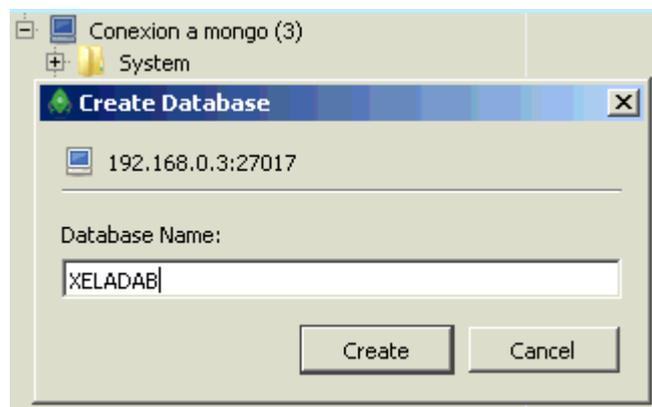


Figura 6.11: Creación base de datos XELADAB



Figura 6.12: Verificación de creación de Base de datos y colección

Instalación de MongoDB 3.4 sobre CentOS7 64-bits

Para el proceso de instalación se siguió los pasos descritos en la documentación de MongoDB¹¹ de su sitio oficial. Sección, Install MongoDB Community Edition on Red Hat Enterprise or CentOS Linux. A continuación se detallan los comandos utilizados como usuario administrador:

1. Configuramos el administrador de paquetes (yum). Creamos el archivo `/etc/yum.repos.d/mongodb-org-3.4.repo` que se utilizará para instalar directamente utilizando yum. Aquí pegamos las siguientes líneas:


```

1 [mongodb-org-3.4]
2 name=MongoDB Repository
3 baseurl=https://repo.mongodb.org/yum/redhat/$releasever/mongodb-
  org/3.4/x86_64/ gpgcheck=1 enabled=1 gpgkey=https://www.
  mongodb.org/static/pgp/server-3.4.asc
      
```
2. Ejecutamos el siguiente comando `yum install -y mongodb-org`
3. En el archivo `/etc/selinux/config` verificamos la configuración de nuestro SELinux y le configuramos con el estado `enforcing`.
4. Permitimos se escuche el puerto 27017 que es con el que por defecto trabaja mongoDB ejecutando la siguiente línea `semanage port -a -t mongod_port_t -p tcp 27017`
5. Iniciamos el servicio de mongoDB con la siguiente línea `service mongod start`
6. Para asegurarnos que mongoDB se inicie luego de reiniciar el sistema ejecutamos la siguiente línea `chkconfig mongod on`

¹¹mongoDB. MONGODB MANUAL 3.4 URL: <https://docs.mongodb.com/manual/introduction/>

7. Reiniciamos el sistema con `reboot`
8. Con el comando `service mongod status` podemos ver el estado actual del servicio de mongoDB.

Una vez instalado el sistema mongoDB procedemos a crear las base y colección.

1. Verificamos que el servicio de mongoDB esté iniciado
2. Accedemos a la consola de MongoShell con el comando
`$ mongo`
3. Creamos la base de datos sobre la cual vamos a trabajar
`> use XELADAB`
4. Creamos la colección donde se almacenará los documentos JSON
`> db.createCollection("actuacionesjudiciales")`

6.7.2. Extracción, transformación y carga de datos desde Microsoft SQL Server 2012 a MongoDB 3.4

Una vez creada la base de datos en mongoDB procedió a exportar los datos desde SQL Server. Para esto haremos uso de la herramienta *Microsoft SQL Server Integration Services Designer (SSIS)*.

Como paso previo se debe obtener las dlls MongoDB.Bson.dll y MongoDB.Driver.dll para C#. Los drivers se encuentran disponibles en <https://github.com/mongodb/mongo-csharp-driver/releases/download/v1.8.3.9/CSharpDriver-1.8.3.zip> y deben ser instalados en C:\Windows\assembly se puede utilizar el siguiente comando para acceder directamente a la carpeta (`%windir%/assembly`)

1. En nuestros programas buscamos Sql Server Data Tools.



Figura 6.13: Sql Server Data Tools

2. Abrir el programa y creamos un nuevo proyecto: File-> New -> Project ver Figura 6.14

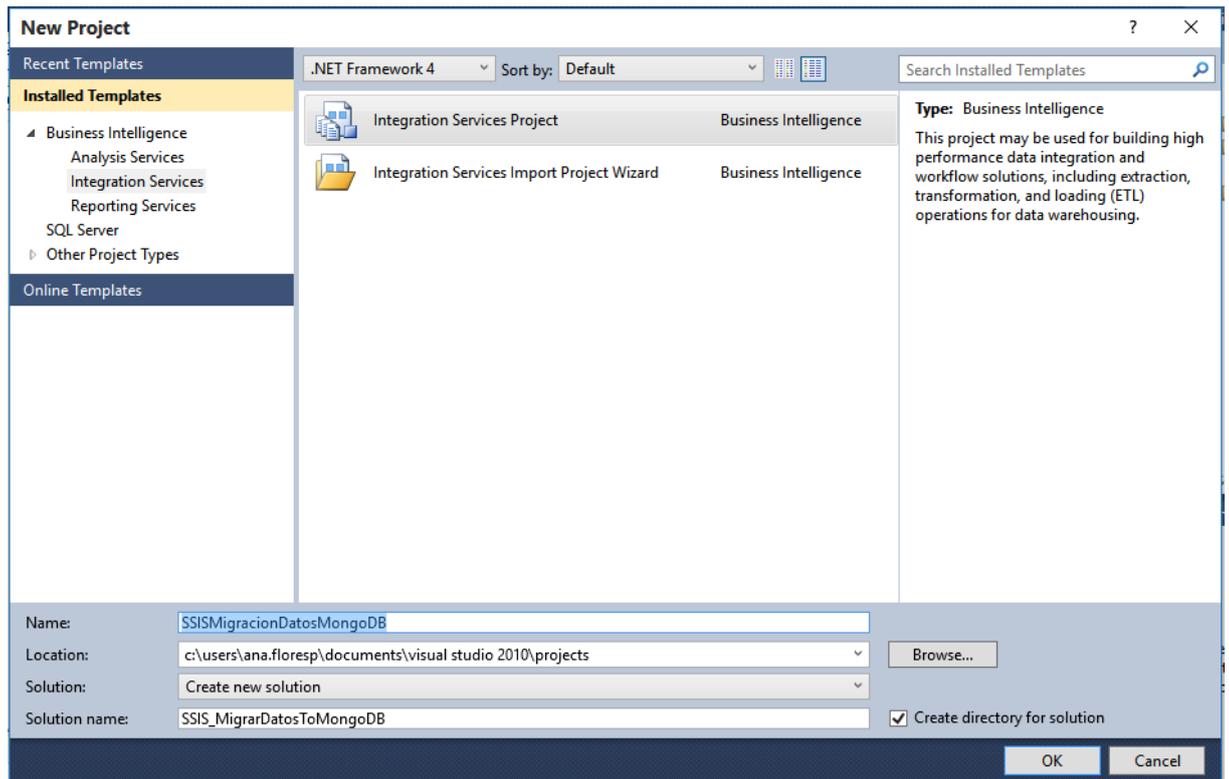


Figura 6.14: Integration Services Project

3. Crear un nuevo paquete Package.dtsx y agregamos los siguientes componentes OLE DB Source y Script Component luego conectamos los 2 componentes ver Figura 6.15.

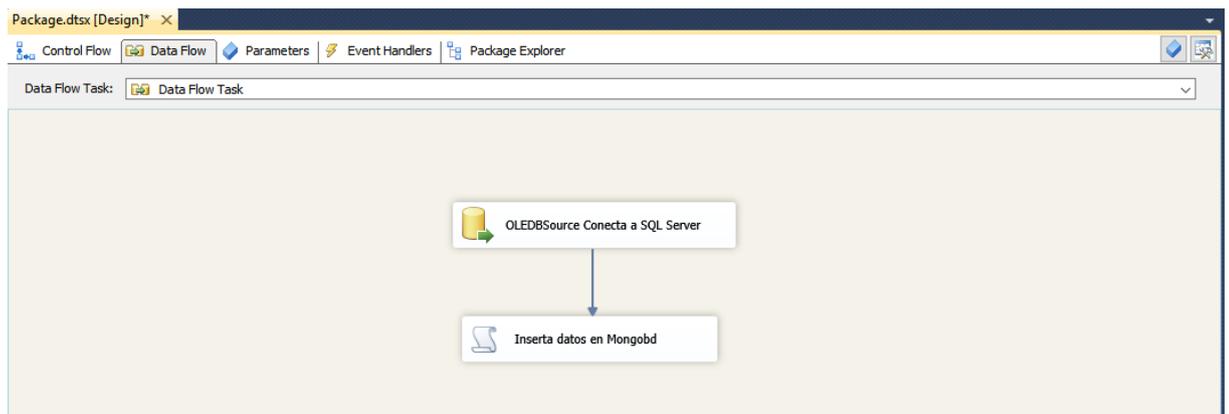


Figura 6.15: Creación de paquete en SSIS

4. Dar clic en el componente OLEDBSource Conecta a SQL Server y configuramos la conexión a nuestra base de datos SQL Server, para extraer la información.

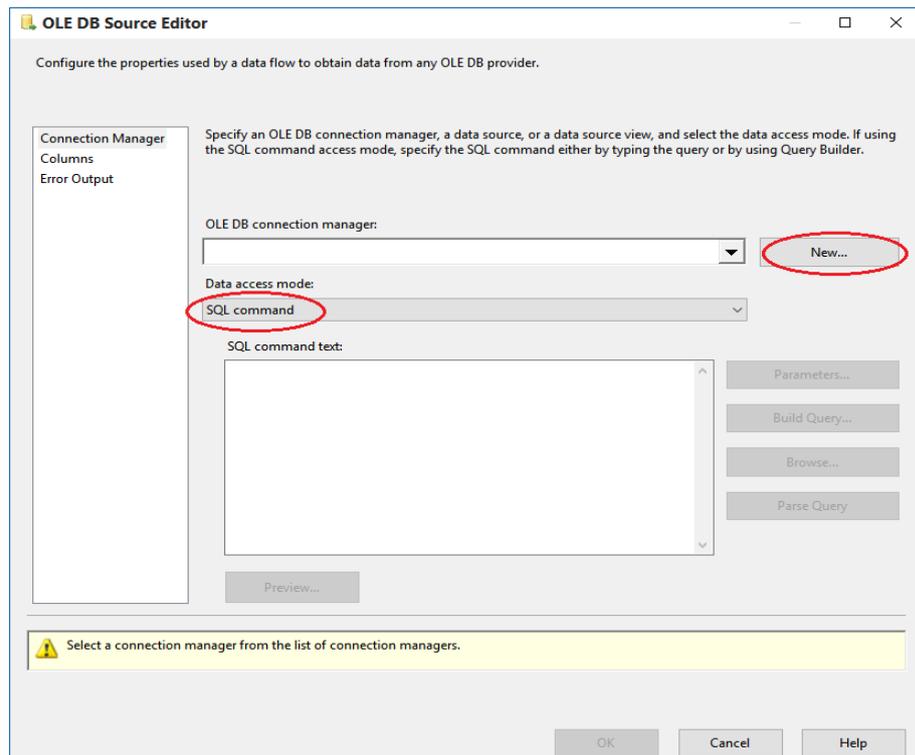


Figura 6.16: 1.- Extracción de datos desde Microsoft SQL Server

En la pantalla OLE DB Source Editor ver Figura 6.16 damos clic en New y en la pantalla de Configure OLE DB Connection Manager ver Figura 6.17 damos clic en New en la pantalla de Connection Manager ver Figura 6.18 especificamos el nombre del servidor de base de datos y la base de datos de donde se extraerá la información, probamos la conexión dando clic en Test Connection si la conexión es exitosa obtendremos el siguiente mensaje Test connection succeeded, damos clic en OK. Regresamos a la pantalla de OLE DB Source Editor seleccionando la conexión hecha, aquí seleccionamos el modo de acceso a datos el cual será SQL Command. Escribimos el código con el cual se obtiene los datos para la transformación y carga ver Figura 6.19.

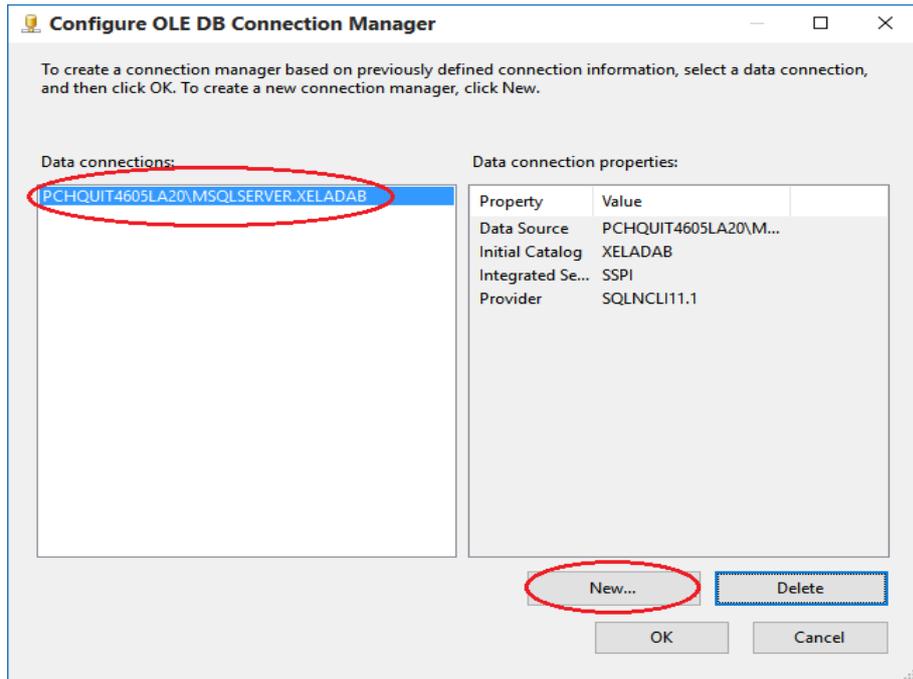


Figura 6.17: 2.- Extracción de datos desde Microsoft SQL Server

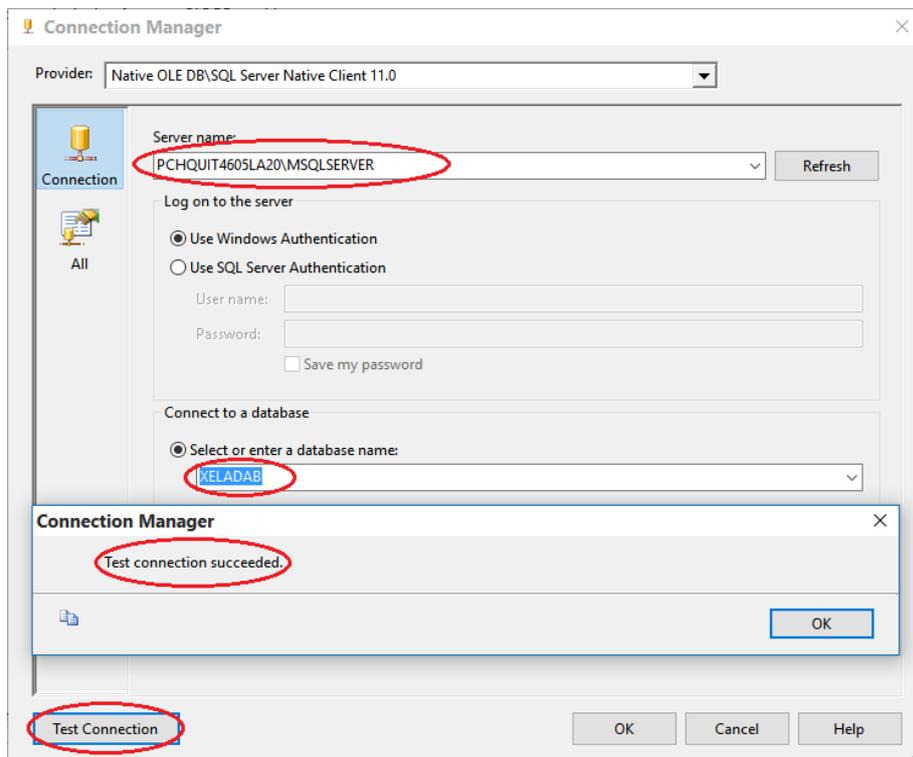


Figura 6.18: 3.- Extracción de datos desde Microsoft SQL Server

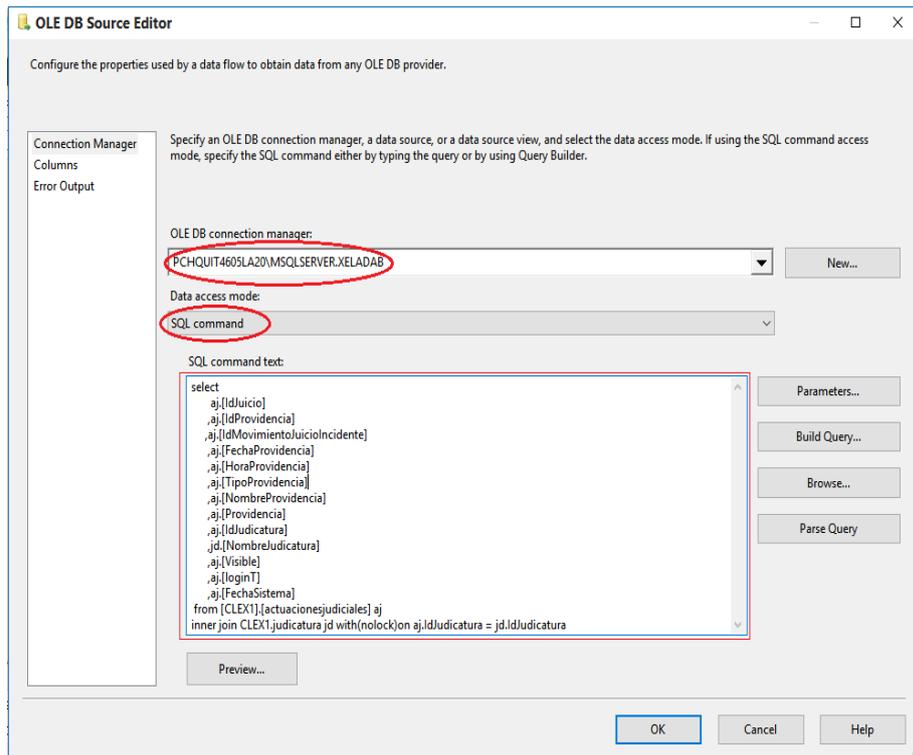


Figura 6.19: 4.- Extracción de datos desde Microsoft SQL Server

5. Dar click en el componente Inserta datos en MongoDB y damos clic en Edit Script ver Figura 6.20.

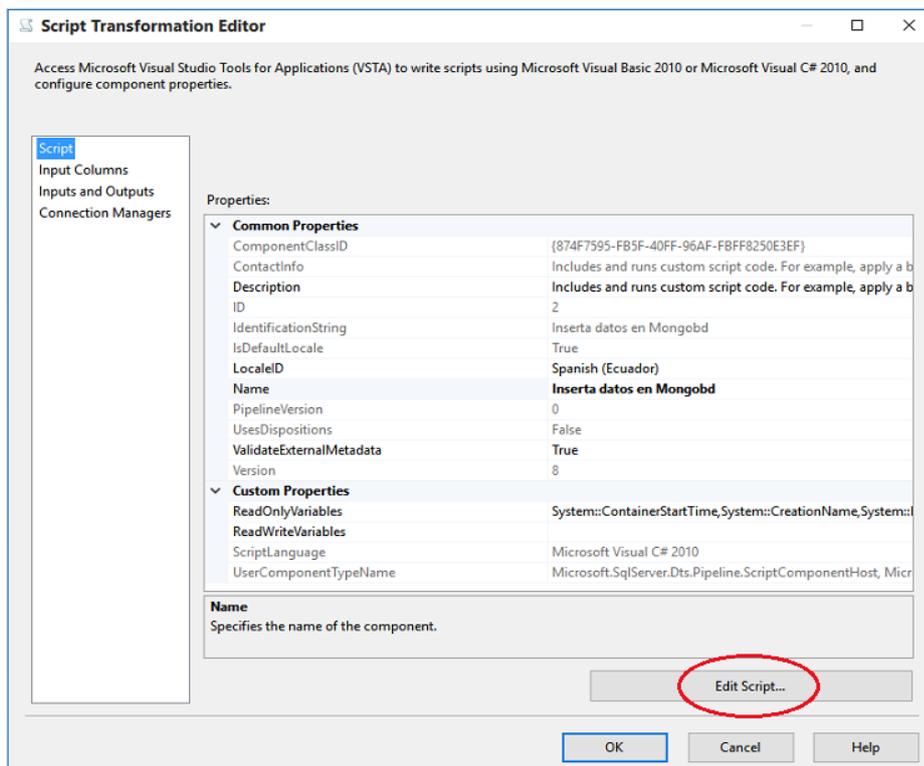


Figura 6.20: Script Inserta datos en MongoDB

6. En esta sección agregar las dll instaladas y declaramos en el Namespaces para su posterior uso ver Figura 6.21

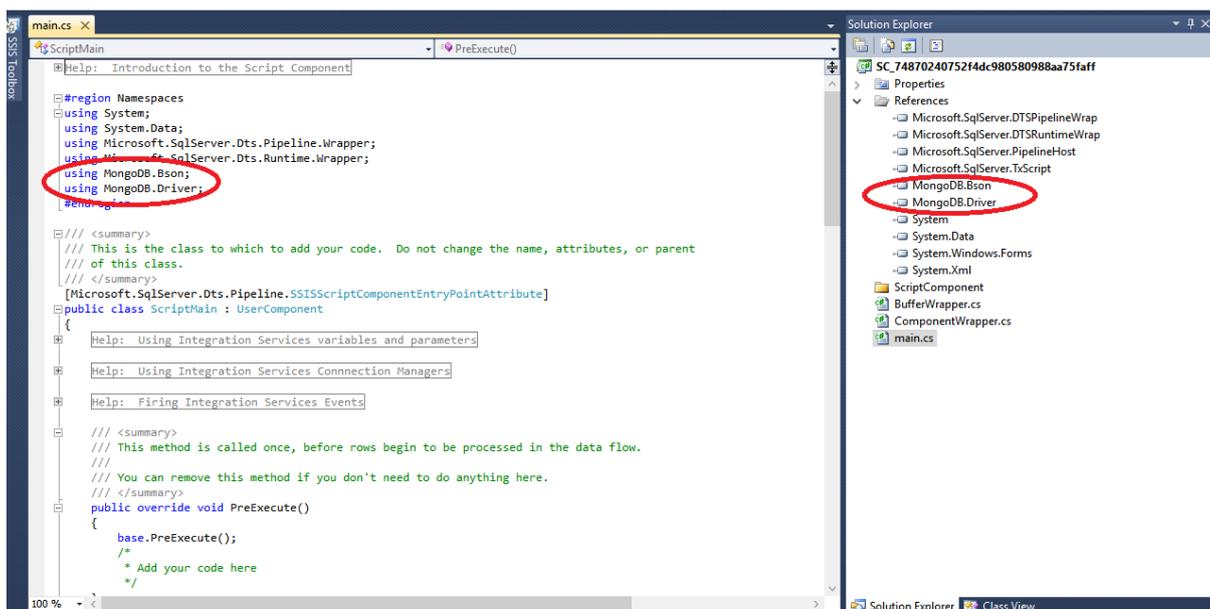


Figura 6.21: Agregar DLL a namespaces

7. En la sección main.cs se escribió el siguiente código el mismo que nos permitirá hacer la transformación y carga de los datos hacia mongoDB.

```
1 #region Namespaces
2 using System;
3 using System.Data;
4 using System.Collections.Generic;
5 using Microsoft.SqlServer.Dts.Pipeline.Wrapper;
6 using Microsoft.SqlServer.Dts.Runtime.Wrapper;
7 using MongoDB.Bson;
8 using MongoDB.Bson.Serialization;
9 using MongoDB.Driver;
10 using MongoDB.Driver.Builders;
11 using MongoDB.Driver.Communication;
12 using MongoDB.Driver.Internal;
13 using MongoDB.Driver.Linq;
14 using MongoDB.Driver.Wrappers;
15 #endregion
16 public override void Input0_ProcessInputRow(Input0Buffer Row)
17     {
18         var mongoConnString = "mongodb://192.168.4.115";
19         var mongoClient = new MongoClient(mongoConnString);
20         var mongoServer = mongoClient.GetServer();
21         var mongoDB = mongoServer.GetDatabase("XELADAB");
22         MongoClient <BsonDocument>
23             collectionActuacionesjudicialesYaInsertados = mongoDB.
24             GetCollection <BsonDocument>("actuacionesjudiciales");
25
26         IMongoQuery queryJuicio= Query.EQ("_id", Row.IdJuicio);
27         var existeJuicio =
28             collectionActuacionesjudicialesYaInsertados.Find(
29                 queryJuicio).Count();
30
31         BsonDocument providenciasDocumento = new BsonDocument { {
32             "_id" , Row.IdActuacion },
33             { "fechaActuacion", Row.FechaActuacion },
34             { "horaActuacion", Row.HoraActuacion },
35             { "tipoActuacion", Row.TipoActuacion.Trim() },
36             { "nombreActuacion", Row.NombreActuacion.Trim() },
37             { "Actuacion", Row.Actuacion.ToString() },
38             { "Judicatura", Row.Judicatura.Trim() },
39             { "Visible", Row.Visible.Trim() },
40             { "LoginT", Row.loginT.Trim() },
41             { "FechaSistema", Row.FechaSistema }
42         };
43     }
```

```

37     BsonArray providencias = new BsonArray {
38         providenciasDocumento };
39
40     Documento = new BsonDocument { { "_id" , Row.IdMovimiento
41         },
42         { "providencias", providencias}
43     };
44     BsonArray incidentes = new BsonArray { incidentesDocumento
45         };
46
47     if (existeJuicio == 0) {
48         BsonDocument procesos = new BsonDocument { { "_id"
49             , Row.IdProceso.Trim() },
50         { "incidentes", incidentes }
51     };
52     Actuacionesjudiciales.Insert(procesos);
53 }
54 else {
55     var cantidadIncidenteEncontrado = BsonSerializer.
56     Deserialize<BsonDocument[]>(
57     collectionActuacionesjudicialesYaInsertados.
58     Find(Query.EQ("_id", Row.IdJuicio)).SetFields(
59     Fields.ElemMatch("incidentes", Query.EQ("_id",
60     Row.IdMovimiento))).ToJson())[0].ElementCount
61     ;
62
63     if (cantidadIncidenteEncontrado == 1) {
64     Actuacionesjudiciales.Update(Query.EQ("_id", Row.IdProceso),
65     Update.Push("incidentes", incidentesDocumento));
66     }else {
67         BsonDocument queryCondicion =
68             MongoDB.Bson.Serialization.
69             BsonSerializer.Deserialize<
70             BsonDocument>("{ 'incidentes ': { '
71             $elemMatch' : { '_id': " + Row.
72             IdMovimiento+ " } } }");
73     QueryDocument queryDocCondicion = new
74     QueryDocument(queryCondicion);
75     Actuacionesjudiciales.Update(queryDocCondicion
76     , Update.Push("incidentes.$.providencias",
77     providenciasDocumento));
78 }
79 }
80 }

```

62 }

8. Verificar que el proceso se ha ejecutado con éxito Figura 6.22 . Los datos cargados en mongoDB se pueden verificar ejecutando `db.actuacionesjudiciales.find()` en la consola de Robomongo ver Figura 6.23.

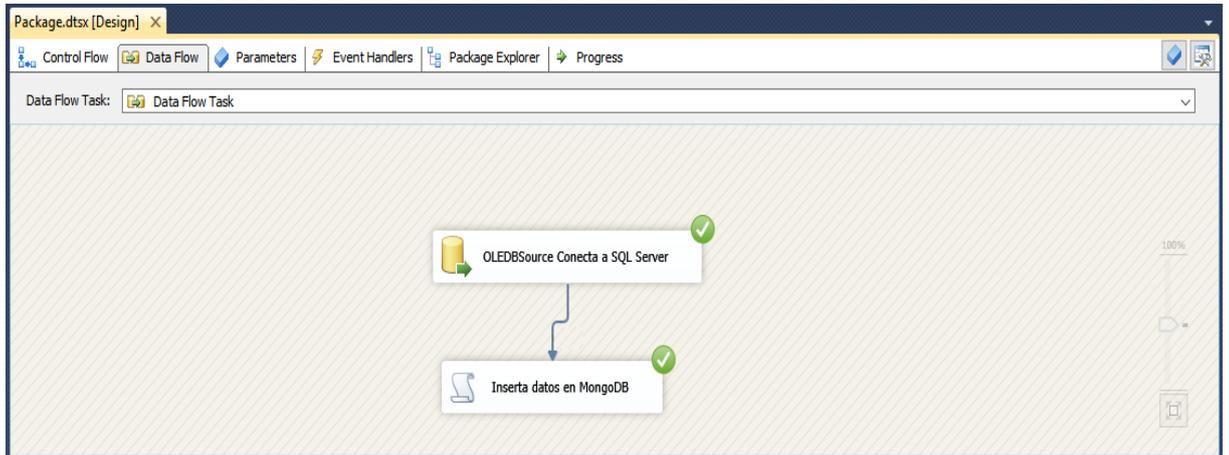


Figura 6.22: Ejecución del paquete en SSIS

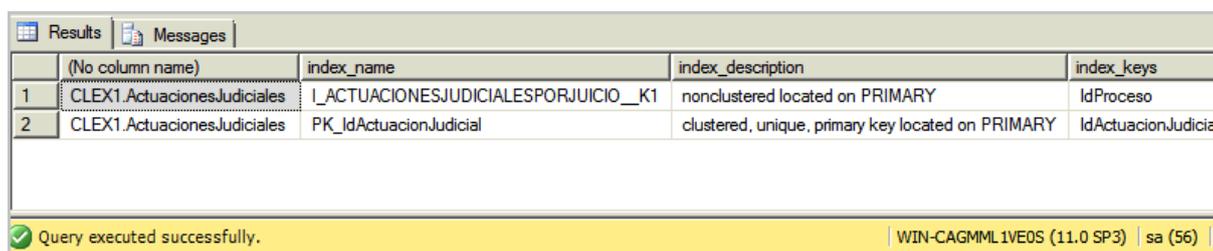
The screenshot shows the Robomongo console with the command `db.actuacionesjudiciales.find()` executed. The results are displayed in a table with columns 'Key', 'Value', and 'Type'. The 'Key' column contains a range of numbers from 1 to 29, and the 'Value' column contains objects with 4 fields. The 'Type' column is 'Object'.

Key	Value	Type
(1) 57	{ 4 fields }	Object
(2) 58	{ 4 fields }	Object
(3) 59	{ 4 fields }	Object
(4) 60	{ 4 fields }	Object
(5) 61	{ 4 fields }	Object
(6) 62	{ 4 fields }	Object
(7) 63	{ 4 fields }	Object
(8) 64	{ 4 fields }	Object
(9) 65	{ 4 fields }	Object
(10) 66	{ 4 fields }	Object
(11) 67	{ 4 fields }	Object
(12) 68	{ 4 fields }	Object
(13) 4163857	{ 4 fields }	Object
(14) 129	{ 4 fields }	Object
(15) 130	{ 4 fields }	Object
(16) 131	{ 4 fields }	Object
(17) 132	{ 4 fields }	Object
(18) 133	{ 4 fields }	Object
(19) 134	{ 4 fields }	Object
(20) 135	{ 4 fields }	Object
(21) 136	{ 4 fields }	Object
(22) 137	{ 4 fields }	Object
(23) 138	{ 4 fields }	Object
(24) 139	{ 4 fields }	Object
(25) 140	{ 4 fields }	Object
(26) 5330646	{ 4 fields }	Object
(27) 120124	{ 4 fields }	Object
(28) 120125	{ 4 fields }	Object
(29) 120126	{ 4 fields }	Object

Figura 6.23: Datos migrados a mongoDB

6.7.3. Optimizar base de datos

Antes de proceder con la obtención de datos se procedió a optimizar los motores de base de datos, para obtener que índices son necesarios crear en SQL Server 2012 se utilizó la herramienta Database Engine Tuning Advisor. En vista de que la tabla cuenta con una clave primaria se tiene un índice clusterizado por tanto la herramienta nos recomienda crear un índice no clusterizado sobre el campo IdProceso ordenado de forma ascendente, la Figura 6.24 visualiza los índices creados sobre la tabla ActuacionesJudiciales.



	(No column name)	index_name	index_description	index_keys
1	CLEX1.ActuacionesJudiciales	I_ACTUACIONESJUDICIALESPORJUICIO__K1	nonclustered located on PRIMARY	IdProceso
2	CLEX1.ActuacionesJudiciales	PK_IdActuacionJudicial	clustered, unique, primary key located on PRIMARY	IdActuacionJudicial

Query executed successfully. WIN-CAGMML1VE0S (11.0 SP3) sa (56)

Figura 6.24: SQL Server 2012 - Índices creados sobre la tabla ActuacionesJudiciales

Para MongoDB 3.4 se procedió a crear los mismos índices que en SQL Server 2012, en MongoDB al crear la colección se crea por defecto el `_id` Index equivalente a un índice clusterizado de SQL Server. La Figura 6.25 visualiza los índices creados sobre la colección ActuacionesJudiciales. La Figura 6.26 el proceso de creación del índice simple sobre el campo IdJuicio.



Figura 6.25: MongoDB - Índices creados sobre la colección ActuacionesJudiciales

```

Administrator: Windows PowerShell
2017-07-18T03:45:10.107-0700 I - [conn8] Index Build: 17762600/29105559 61%
2017-07-18T03:45:13.001-0700 I - [conn8] Index Build: 18093900/29105559 62%
2017-07-18T03:45:16.007-0700 I - [conn8] Index Build: 18258700/29105559 62%
2017-07-18T03:45:19.000-0700 I - [conn8] Index Build: 18631600/29105559 64%
2017-07-18T03:45:22.022-0700 I - [conn8] Index Build: 19151400/29105559 65%
2017-07-18T03:45:25.000-0700 I - [conn8] Index Build: 19845600/29105559 68%
2017-07-18T03:45:28.001-0700 I - [conn8] Index Build: 20230300/29105559 69%
2017-07-18T03:45:31.000-0700 I - [conn8] Index Build: 20467500/29105559 70%
2017-07-18T03:45:34.000-0700 I - [conn8] Index Build: 20832900/29105559 71%
2017-07-18T03:45:37.000-0700 I - [conn8] Index Build: 21133000/29105559 72%
2017-07-18T03:45:53.493-0700 I - [conn8] Index Build: 21315900/29105559 73%
2017-07-18T03:45:56.000-0700 I - [conn8] Index Build: 21481600/29105559 73%
2017-07-18T03:45:59.000-0700 I - [conn8] Index Build: 21759100/29105559 74%
2017-07-18T03:46:02.000-0700 I - [conn8] Index Build: 22087400/29105559 75%
2017-07-18T03:46:05.024-0700 I - [conn8] Index Build: 22393900/29105559 76%
2017-07-18T03:46:08.010-0700 I - [conn8] Index Build: 22595500/29105559 77%
2017-07-18T03:46:11.000-0700 I - [conn8] Index Build: 22882700/29105559 78%
2017-07-18T03:46:14.000-0700 I - [conn8] Index Build: 23217600/29105559 79%
2017-07-18T03:46:56.711-0700 I - [conn8] Index Build: 23217700/29105559 79%
2017-07-18T03:46:59.000-0700 I - [conn8] Index Build: 23469700/29105559 80%
2017-07-18T03:47:02.022-0700 I - [conn8] Index Build: 23768600/29105559 81%
2017-07-18T03:47:05.000-0700 I - [conn8] Index Build: 24062300/29105559 82%
2017-07-18T03:47:32.268-0700 I - [conn8] Index Build: 24062400/29105559 82%
2017-07-18T03:47:35.021-0700 I - [conn8] Index Build: 24323400/29105559 83%
2017-07-18T03:47:38.000-0700 I - [conn8] Index Build: 24594600/29105559 84%
2017-07-18T03:47:41.000-0700 I - [conn8] Index Build: 24814800/29105559 85%
2017-07-18T03:47:44.000-0700 I - [conn8] Index Build: 25185400/29105559 86%
2017-07-18T03:47:47.000-0700 I - [conn8] Index Build: 25474400/29105559 87%
2017-07-18T03:47:50.000-0700 I - [conn8] Index Build: 25812700/29105559 88%
2017-07-18T03:47:53.000-0700 I - [conn8] Index Build: 26087200/29105559 89%
2017-07-18T03:47:56.000-0700 I - [conn8] Index Build: 26345500/29105559 90%
2017-07-18T03:47:59.000-0700 I - [conn8] Index Build: 26717400/29105559 91%
2017-07-18T03:48:02.000-0700 I - [conn8] Index Build: 27054800/29105559 92%
2017-07-18T03:48:05.010-0700 I - [conn8] Index Build: 27392800/29105559 94%
2017-07-18T03:48:08.000-0700 I - [conn8] Index Build: 27696100/29105559 95%
2017-07-18T03:48:11.000-0700 I - [conn8] Index Build: 28073800/29105559 96%
2017-07-18T03:48:14.000-0700 I - [conn8] Index Build: 28399300/29105559 97%
2017-07-18T03:48:17.000-0700 I - [conn8] Index Build: 28709500/29105559 98%
2017-07-18T03:48:20.000-0700 I - [conn8] Index Build: 29041500/29105559 99%
2017-07-18T03:48:49.000-0700 I - [conn8] Index: (2/3) BTree Bottom Up Progress: 14710000/29105559 50%
2017-07-18T03:48:57.542-0700 I INDEX [conn8] done building bottom layer, going to commit
2017-07-18T03:48:58.843-0700 I INDEX [conn8] build index done. scanned 29105559 total records. 405 secs
2017-07-18T03:48:58.872-0700 I COMMAND [conn8] command XELADAB.$cmd command: createIndexes < createIndexes: "actuacione
sjudicialesplana", indexes: [ < ns: "XELADAB.actuacionesjudicialesplana", key: < idJuicio: 1.0 >, name: "idJuicio_1" > ]
> numYields:0 resLen:98 locks:< Global: < acquireCount: < r: 1, w: 1 > >, Database: < acquireCount: < W: 1 > >, Collect
ion: < acquireCount: < w: 1 > > > protocol:op_command 405752ms
2017-07-18T03:48:58.880-0700 I COMMAND [conn16] command admin.$cmd command: listDatabases < listDatabases: 1 > numYield
s:0 resLen:215 locks:< Global: < acquireCount: < r: 8 > >, Database: < acquireCount: < r: 3 > >, acquireWaitCount: < r: 1
>, timeAcquiringMicros: < r: 234567116 > > protocol:op_command 234568ms

```

Figura 6.26: MongoDB - Indices proceso de creación

6.7.4. Recolección de datos

Para la presente investigación se trabajó con la tabla *ActuacionesJudiciales* esta contiene la historia de los procesos judiciales. En la Tabla 6.4 se establece las características de la tabla al momento de realizar las pruebas. El funcionamiento de la tabla es la siguiente: dado un número de proceso judicial se obtendrá todas las actuaciones judiciales realizadas entorno a este.

Característica	Detalle
Número de registros	29410990
Tamaño de la tabla	59314 MB
Tipos de datos que almacena	char, varchar, text, int, datetime

Tabla 6.4: Características tabla ActuacionesJudiciales - SQL Server

La tabla *ActuacionesJudiciales* tiene el campo llamado **actuacion** de tipo Text en SQL Server y almacena el texto plano de la actuación judicial. Por tanto

cabe recalcar que se pretende analizar el efecto que produce la variación en las consultas con respecto al tamaño y cantidad de las actuaciones judiciales en los tiempos de respuesta de los sistemas gestores de datos en estudio.

Determinación de los escenarios de prueba

Características generales:

- Los dos sistemas gestores de datos tendrán como número máximo de 500 conexiones simultáneas a la base de datos
- El número de usuarios concurrentes será 99 el cual se ha obtenido en base a los datos de la Tabla 6.1 y la siguiente fórmula: $Te = ((To) + (4 * Tp) + (Tr))/6$ donde:

Te = Cantidad o Tiempo Esperado

To = Tiempo o Cantidad Optimista

Tp = Tiempo o Cantidad Promedio

Tr = Tiempo o Cantidad Pesimista

$$Te = (1038 + (4 * 476) + 14)/6$$

$$Te = 492,6$$

$$Te = 493$$

Al valor obtenido en Te le dividimos para 5 que son el número de horas más concurridas al aplicativo según la Tabla 6.1 con esto obtendremos el número de usuarios concurrentes Nu para evaluar el rendimiento de las bases de datos.

$$Nu = Te/5$$

$$Nu = 493/5$$

$$Nu = 98,6$$

$$Nu = 99$$

Al valor obtenido en Nu lo dividimos para 60 segundos y obtendremos el lapso de tiempo en segundos entre consulta y consulta ejecutada Tx (*periodo de*

subida) .

$$Tx = Nu/60$$

$$Tx = 99/60$$

$$Tx = 1,65$$

$$Tx = 2$$

- El grupo de datos con los cuales se realizará las pruebas son 3: todos los procesos judiciales registrados, los procesos judiciales con la mayor cantidad de actuaciones judiciales registradas y los procesos judiciales con mayor peso obtenido de la suma de la cantidad en bytes del campo *Actuacion* de la tabla *ActuacionesJudiciales*. Para el grupo de los de mayor cantidad y mayor peso se utilizó el siguiente proceso y la formula de distribución normal o “Campana de Gauss”.

Grupo de mayor cantidad

Se buscará el 2.5 de los casos que registren mayor cantidad de actuaciones, haciendo uso de la Distribución normal. Para lo cual se siguió el siguiente proceso: De los datos registrados en la tabla *ActuacionesJudiciales* se agrupado y contado las actuaciones judiciales registradas por cada proceso judicial. Luego de obtener este resultado se ha ordenado de forma descendente por la cantidad de actuaciones judiciales ver el Anexo 2. La formula utilizada es la siguiente :

$$Pr = \mu + \sigma + \sigma$$

Donde:

Pr = Distribucion normal

μ = Media de la población

σ = Desviación típica

Cálculo de la Media poblacional

$$\mu = \sum_N^X$$

Donde:

Σ =sumatoria
 X =valores o datos
 N =número de elementos

$$\mu = 29410990 / 4134199$$

$$\mu = 7$$

Cálculo de la varianza poblacional

$$\sigma^2 = \frac{\sum(X - \mu)^2}{N}$$

$$\sigma^2 = \frac{567205665}{4134199}$$

$$\sigma^2 = 137$$

Cálculo de la desviación típica o estándar

$$\sigma = \sqrt{\sigma^2}$$

$$\sigma = \sqrt{137}$$

$$\sigma = 12$$

Por tanto

$$Pr = 7 + 12 + 12$$

$$Pr = 31$$

Es decir **31** será la cantidad mínima de actuaciones judiciales que debe tener un proceso judicial para formar parte de la muestra para el análisis de tiempo. Como el listado de la población fue ordenado de forma descendente por la cantidad de actuaciones judiciales, se tomarán todos aquellos que tengan 31 o más, en la cantidad de actuaciones judiciales. Es decir tendremos **161544** registros de los cuales de forma aleatoria se seleccionará 99 para las pruebas.

Grupo de mayor peso

Se buscará el 2.5 de los casos que registren mayor peso de la suma en bytes

de las actuaciones judiciales asociadas, haciendo uso de la formula de Distribución normal (estandar). Para lo cual se utilizó el siguiente proceso: De los datos registrados en la tabla *ActuacionesJudiciales* se agrupado y sumado la cantidad en bytes del campo *actuacion* registrado por cada proceso. Luego de obtener este resultado se ha ordenado de forma descendente por el peso de las actuaciones judiciales, ver el Anexo 2. La fórmula es la misma que se utilizó para obtener el grupo de mayor cantidad. Dando como resultado los siguientes valores:

Cálculo de la Media poblacional

$$\mu = \frac{\sum X}{N}$$

$$\mu = 35837212587 / 4134199$$

$$\mu = 8668$$

Cálculo de la varianza poblacional

$$\sigma^2 = \frac{\sum (X - \mu)^2}{N}$$

$$\sigma^2 = \frac{1985215304151537}{4134199}$$

$$\sigma^2 = 480193456$$

Cálculo de la desviación típica o estándar

$$\sigma = \sqrt{\sigma^2}$$

$$\sigma = \sqrt{480193456}$$

$$\sigma = 21913$$

Por tanto

$$Pr = 8668 + 21913 + 21913$$

$$Pr = 52494$$

Es decir **52494** será el peso mínimo que debe tener un proceso judicial para formar parte de la muestra para el análisis de tiempo. Como el listado de la población fue ordenado de forma descendente por la suma del peso de actuaciones judiciales asociadas, se tomarán todos aquellos que tengan **52494 bytes** o más en la suma de sus actuaciones judiciales. Es decir tendremos es **109462** registros de los cuales de forma aleatoria se seleccionará 99 para las pruebas.

Para la ejecución del plan de pruebas se se usó JMeter herramienta que permitió validar el rendimiento y comportamiento del manejo de concurrencia de un servidor de base de datos. Tanto para SQL Server 2012 y MongoDB 3.4 se ha configurado los siguientes escenarios.

JMeter plan de pruebas

Thead Group (Grupo de Hilos)

- Number of Threads (user) : 99
- Ramp-Up Period (periodo de subida): 2
- Loop Count (número de repeticiones): 1

Plan para SQL Server 2012

Gupo de hilos

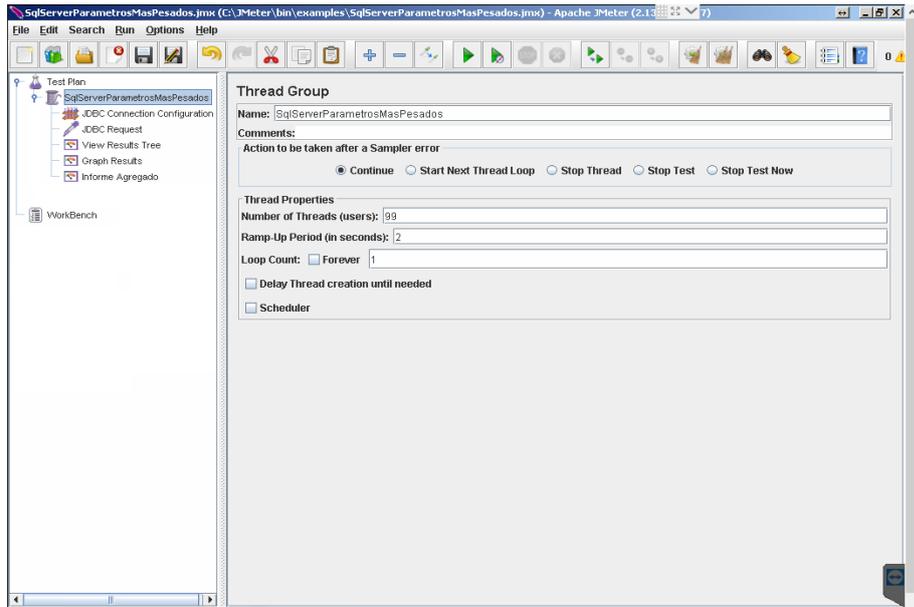


Figura 6.27: SQL Server - Configuración JMeter - Grupo de Hilos

Conexión JDBC al servidor

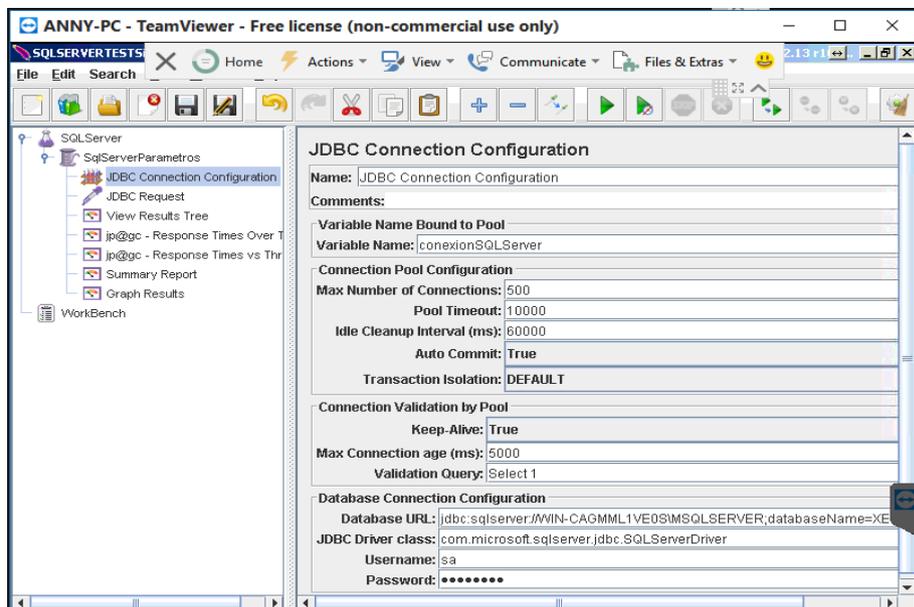


Figura 6.28: SQL Server - Configuración JMeter - Conexión JDBC

Peticiones JDBC

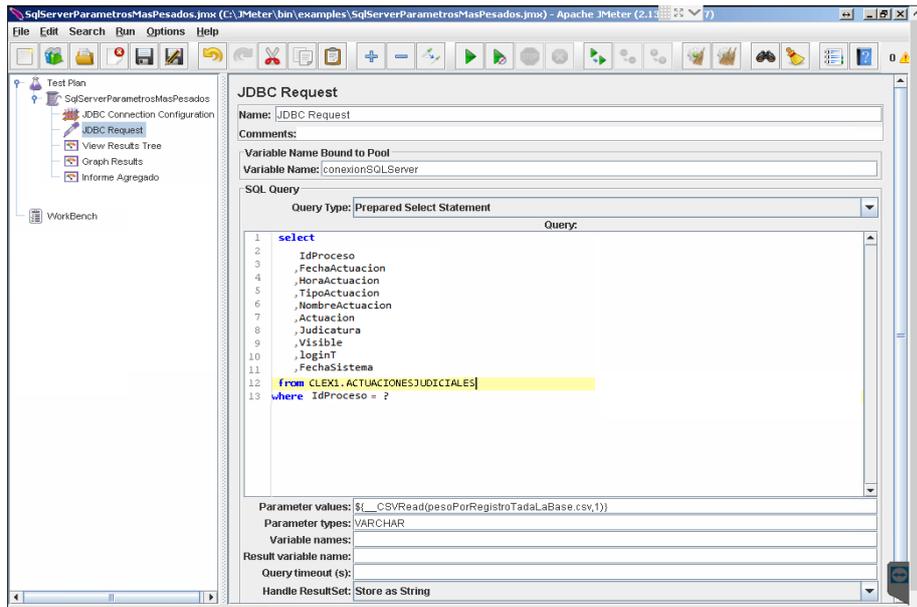


Figura 6.29: SQL Server - Configuración JMeter - Peticiones JDBC

Plan para MongoDB

Gupo de hilos

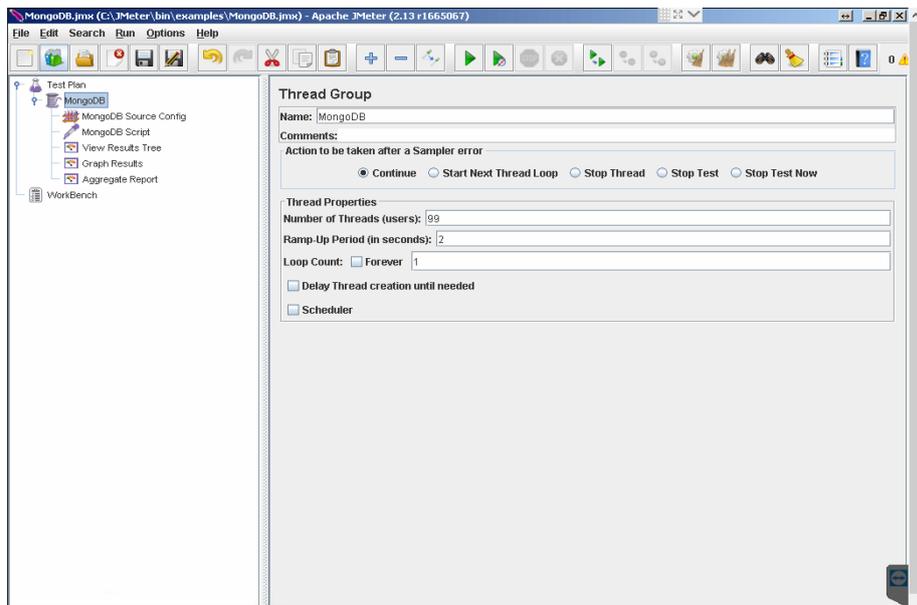


Tabla 6.5: MongoDB - Configuración JMeter - Grupo de Hilos

Conexión al servidor (DataSource)

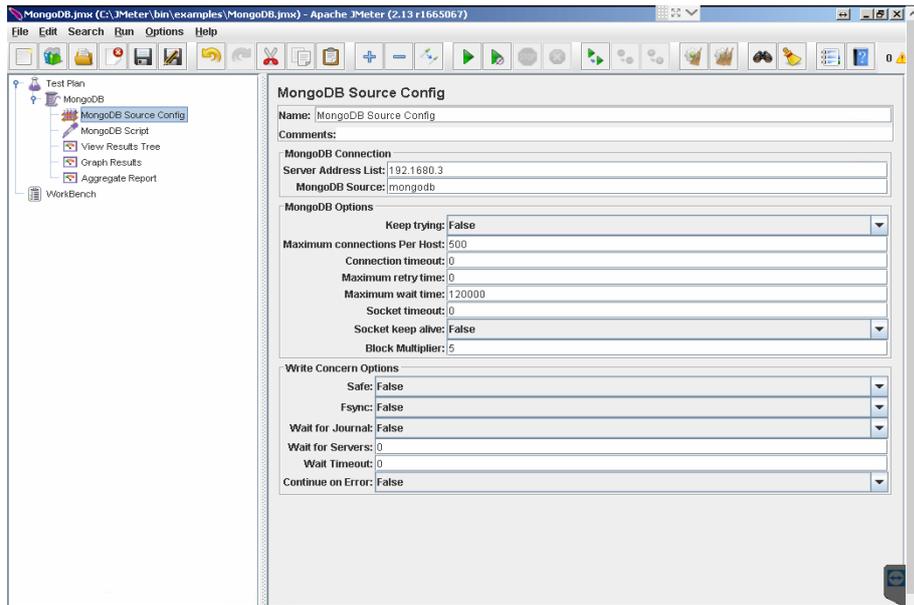


Figura 6.30: MongoDB - Configuración JMeter - Conexión al servidor

Peticiones al servidor (Script)

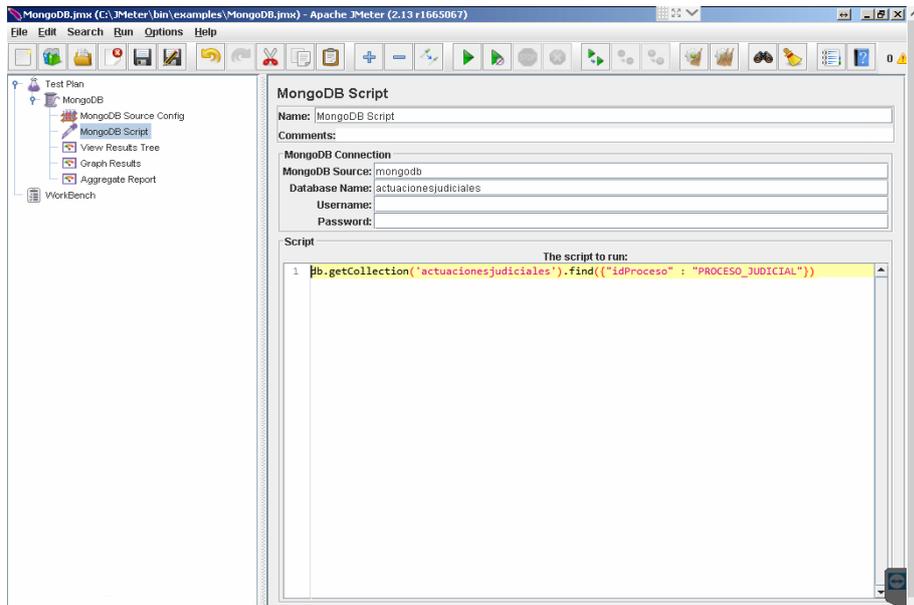


Figura 6.31: MongoDB- Configuración JMeter - Peticiones al servidor

Para los planes de prueba diseñados tanto para SQL Server 2012 y MongoDB 3.4 se aplicará los siguientes escenarios Tabla 6.6, 6.7 y 6.8.

Test 1.- Todos los procesos judiciales registrados en la base de datos sin importar cantidad o peso de las actuaciones judiciales asociadas	
Operación	Selección
Objetivo	Obtener los tiempos de respuesta promedio de SQL Server y MongoDB utilizado para completar el proceso de búsqueda con 99 números de usuarios y X proceso.
Proceso	JMeter se encargará de ejecutar el Plan de pruebas para obtener de forma aleatoria 99 procesos judiciales, que luego servirán para que cada uno de los 99 usuarios dispere una consulta a la base de datos cada 2 segundos con un número de proceso judicial.
Descripción	Se ejecutará 5 veces cada prueba y el promedio de los valores obtenidos serán utilizados para realizar el análisis de resultados
Procedimiento	
1.- Generar un listado de 99 procesos judiciales seleccionados de forma aleatoria. 2.- Implementar la consulta correspondiente. 3.- Ejecutar el plan de pruebas programado en JMeter con la consulta correspondiente a cada usuario. 4.- Repetir el proceso del punto 3.	
Código SQL Server	
SELECT IdProceso, FechaActuacion, HoraActuacion ,TipoActuacion, NombreActuacion, Actuacion ,Judicatura, Visible, loginT, FechaSistema FROM CLEX1.ActuacionesJudiciales WHERE IdProceso = ?	
Código MongoDB	
db.getCollection('actuacionesjudiciales').find({"IdProceso" : "ProcesoJudicial"})	

Tabla 6.6: Test de rendimiento 1

Test 2.- Procesos judiciales con <i>mayor peso</i> de actuaciones judiciales asociadas	
Operación	Selección
Objetivo	Obtener los tiempos de respuesta promedio de SQL Server y MongoDB utilizado para completar el proceso de búsqueda con 99 números de usuarios y X proceso.
Proceso	JMeter se encargará de ejecutar el Plan de pruebas para obtener de forma aleatoria 99 procesos judiciales del grupo de los de mayor peso, que luego servirán para que cada uno de los 99 usuarios dispare una consulta a la base de datos cada 2 segundos con un número de proceso judicial
Descripción	Se ejecutará 5 veces cada prueba y el promedio de los valores obtenidos serán utilizados para realizar el análisis de resultados
Procedimiento	
1.- Generar un listado de 99 procesos judiciales seleccionados de forma aleatoria. 2.- Implementar la consulta correspondiente. 3.- Ejecutar el plan de pruebas programado en JMeter con la consulta correspondiente a cada usuario. 4.- Repetir el proceso del punto 3.	
Código SQL Server	
<pre> SELECT IdProceso, FechaActuacion, HoraActuacion ,TipoActuacion, NombreActuacion, Actuacion ,Judicatura, Visible, loginT, FechaSistema FROM CLEX1.ActuacionesJudiciales WHERE IdProceso = ? </pre>	
Código MongoDB	
<pre> db.getCollection('actuacionesjudiciales').find({"IdProceso" : "ProcesoJudicial"}) </pre>	

Tabla 6.7: Test de rendimiento 2

Test 3.- Procesos judiciales con <i>mayor cantidad</i> de actuaciones judiciales asociadas	
Operación	Selección
Objetivo	Obtener los tiempos de respuesta promedio de SQL Server y MongoDB utilizado para completar el proceso de búsqueda con 99 números de usuarios y X proceso.
Proceso	JMeter se encargará de ejecutar el Plan de pruebas para obtener de forma aleatoria 99 procesos judiciales del grupo de los de mayor cantidad, que luego servirán para que cada uno de los 99 usuarios dispare una consulta a la base de datos cada 2 segundos con un número de proceso judicial
Descripción	Se ejecutará 5 veces cada prueba y el promedio de los valores obtenidos serán utilizados para realizar el análisis de resultados
Procedimiento	
1.- Generar un listado de 99 procesos judiciales seleccionados de forma aleatoria. 2.- Implementar la consulta correspondiente. 3.- Ejecutar el plan de pruebas programado en JMeter con la consulta correspondiente a cada usuario. 4.- Repetir el proceso del punto 3.	
Código SQL Server	
<pre>SELECT IdProceso, FechaActuacion, HoraActuacion ,TipoActuacion, NombreActuacion, Actuacion ,Judicatura, Visible, loginT, FechaSistema FROM CLEX1.ActuacionesJudiciales WHERE IdProceso = ?</pre>	
Código MongoDB	
<pre>db.getCollection('actuacionesjudiciales').find({"IdProceso" : "ProcesoJudicial"})</pre>	

Tabla 6.8: Test de rendimiento 2

Resultados de la ejecución de los planes de prueba

Los valores totales obtenidos por el componente «Aggregate Graph» de JMeter se muestran en las siguientes tablas. La descripción de cada uno de los datos se presentan en el Anexo 3.

Test 1.- Todos los procesos judiciales registrados en la base de datos sin importar cantidad o peso de las actuaciones judiciales asociadas

El listado de los procesos judiciales obtenidos de forma aleatoria para el test se presentan en el Anexo 4.

SQL Server sobre Windows									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
SQL Server Request1	99	1168	1320	1351	39	1862	0,00%	51,8	72174,9
SQL Server Request2	99	953	1029	1101	51	1181	0,00%	83,3	116123,2
SQL Server Request3	99	882	1007	1046	23	1075	0,00%	91,4	127488,9
SQL Server Request4	99	858	901	918	67	980	0,00%	100,5	140173,1
SQL Server Request5	99	642	690	700	18	715	0,00%	138,5	4594,3

MongoDB sobre Windows									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
MongoDB Request1	99	1222	1236	2160	66	2419	0,00%	40,9	28,6
MongoDB Request2	99	1155	1168	2117	38	2358	0,00%	42	29,3
MongoDB Request3	99	1127	1104	2074	32	2316	0,00%	42,7	29,8
MongoDB Request4	99	1141	1135	2027	34	2269	0,00%	43,4	30,3
MongoDB Request5	99	1200	1195	2116	38	2345	0,00%	42,2	29,5

MongoDB sobre Linux									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
MongoDB Request1	99	919	892	1670	37	1883	0,00%	52,4	36,6
MongoDB Request2	99	902	902	1673	22	1859	0,00%	53,2	37,2
MongoDB Request3	99	876	864	1630	23	1844	0,00%	53,7	37,5
MongoDB Request4	99	898	895	1632	22	1835	0,00%	53,9	37,7
MongoDB Request5	99	888	891	1642	25	1848	0,00%	53,5	37,4

Tabla 6.9: Resultados Test1

Test 2.- Procesos judiciales con mayor peso de actuaciones judiciales asociadas

El listado de los procesos judiciales obtenidos de forma aleatoria para el test se presentan en el Anexo 5.

SQL Server sobre Windows									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
SQL Server Request1	99	1183	1446	1462	217	1510	0,00%	64,1	91145,7
SQL Server Request2	99	1048	1229	1250	172	1330	0,00%	73,7	102731,1
SQL Server Request3	99	920	1129	1192	120	1226	0,00%	79,4	110722,2
SQL Server Request4	99	835	1059	1080	250	1167	0,00%	83,7	116712,2
SQL Server Request5	99	792	863	909	70	937	0,00%	104,5	145797,8

MongoDB sobre Windows									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
MongoDB Request1	99	1319	1344	2291	73	2531	0,00%	39,1	27,3
MongoDB Request2	99	1186	1194	2152	34	2422	0,00%	40,8	28,5
MongoDB Request3	99	1149	1142	2079	39	2329	0,00%	42,5	29,7
MongoDB Request4	99	1157	1156	2038	38	2290	0,00%	43,2	30,1
MongoDB Request5	99	1173	1164	2103	38	2381	0,00%	41,5	29

MongoDB sobre Linux									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
MongoDB Request1	99	958	957	1722	23	1905	0,00%	52	36,3
MongoDB Request2	99	940	929	1671	20	1889	0,00%	52,3	36,5
MongoDB Request3	99	920	912	1681	22	1908	0,00%	51,9	36,2
MongoDB Request4	99	909	904	1657	24	1861	0,00%	53,2	37,2
MongoDB Request5	99	908	900	1657	21	1858	0,00%	53,3	37,2

Tabla 6.10: Resultados Test2

Test 3.- Procesos judiciales con mayor cantidad de actuaciones judiciales asociadas

El listado de los procesos judiciales obtenidos de forma aleatoria para el test se presentan en el Anexo 5.

SQL Server sobre Windows									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
SQL Server Request1	99	8057	7715	9945	4687	11140	0	7	5621,5
SQL Server Request2	99	922	1190	1205	189	1234	0	80,2	64519,3
SQL Server Request3	99	833	1115	1129	112	1143	0	86,2	69292,3
SQL Server Request4	99	774	858	864	50	907	0	109	87683,7
SQL Server Request5	99	713	877	882	26	889	0	110,5	88858

MongoDB sobre Windows									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
MongoDB Request1	99	1303	1285	2248	52	2476	0	39,4	27,5
MongoDB Request2	99	1192	1190	2184	39	2410	0	41,1	28,7
MongoDB Request3	99	1155	1144	2130	39	2372	0	41,7	29,1
MongoDB Request4	99	1180	1198	2087	44	2353	0	42	29,3
MongoDB Request5	99	1160	1178	2047	43	2302	0	43	30

MongoDB sobre Linux									
Label	# Samples	Average	Median	90% Line	Min	Max	Error %	Throughput	KB/sec
MongoDB Request1	99	943	933	1719	28	1913	0,00%	51,7	36,1
MongoDB Request2	99	887	882	1615	22	1804	0,00%	54,9	38,3
MongoDB Request3	99	897	888	1621	26	1827	0,00%	54,2	37,8
MongoDB Request4	99	913	915	1639	33	1832	0,00%	54	37,7
MongoDB Request5	99	894	889	1606	23	1790	0,00%	55,3	38,6

Tabla 6.11: Resultados Test3

6.7.5. Análisis de resultados

Para el análisis de los datos obtenidos de los resultados de los test realizados a SQL Server y MongoDB, se procederá a trabajar con la información del campo media (Average) el cual, representa los tiempos promedios en mili segundos(ms) de la ejecución de cada uno de los test para responder a las peticiones de 99 usuarios concurrentes consultando cada uno un proceso judicial diferente, de estos resultados se tomó 5 muestras.

Test 1.- Todos los procesos judiciales registrados en la base de datos sin importar cantidad o peso de las actuaciones judiciales asociadas

SQL Server sobre Windows	
Label	Average
SQL Server Request1	1168
SQL Server Request2	953
SQL Server Request3	882
SQL Server Request4	858
SQL Server Request5	642
Promedio: 900,6	

MongoDB sobre Windows	
Label	Average
MongoDB Request1	1222
MongoDB Request2	1155
MongoDB Request3	1127
MongoDB Request4	1141
MongoDB Request5	1200
Promedio: 1169	

MongoDB sobre Linux	
Label	Average
MongoDB Request1	919
MongoDB Request2	902
MongoDB Request3	876
MongoDB Request4	898
MongoDB Request5	888
Promedio: 896,6	

Tabla 6.12: Promedio - Test1

Para la elaboración de las gráficas se ha calculado el valor promedio de los tiempos de respuesta de todas las muestras tomadas y estos resultados se aproximado al valor superior. La Figura 6.32 indica el tiempo de respuesta en mili segundos utilizados por los sistemas de bases de datos SQL Server y MongoDB para, responder 99 peticiones concurrentes a una tabla o colección respectivamente que contiene 29410990 registros o documentos. Los datos utilizados para la consulta varían entre cantidad y peso de las actuaciones judiciales asociadas. Se puede observar en la gráfica que existe una variación en los tiempos de respuesta, siendo favorable para SQL Server sobre Windows y MongoDB sobre Linux.

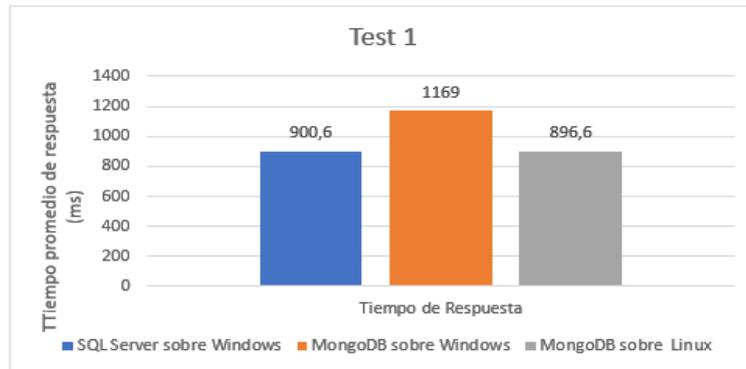


Figura 6.32: Test1 - SQLServer vs MongoDB

Test 2.- Procesos judiciales con mayor peso de actuaciones judiciales asociadas

SQL Server sobre Windows	
Label	Average
SQL Server Request1	1183
SQL Server Request2	1048
SQL Server Request3	920
SQL Server Request4	835
SQL Server Request5	792
Promedio: 955,6	

MongoDB sobre Windows	
Label	Average
MongoDB Request1	1319
MongoDB Request2	1186
MongoDB Request3	1149
MongoDB Request4	1157
MongoDB Request5	1173
Promedio: 1196,8	

MongoDB sobre Linux	
Label	Average
MongoDB Request1	958
MongoDB Request2	940
MongoDB Request3	920
MongoDB Request4	909
MongoDB Request5	908
Promedio: 927	

Tabla 6.13: Promedio - Test 2

Para la elaboración de las gráficas al igual que en el Test 1 se calculó el valor promedio de los tiempos de respuesta. En este caso las consultas realizadas son utilizando 99 juicios que están dentro de la muestra de mayor peso. Se puede observar en la gráfica que existe una variación en los tiempos de respuesta, siendo nuevamente favorable para SQL Server sobre Windows y MongoDB sobre Linux.

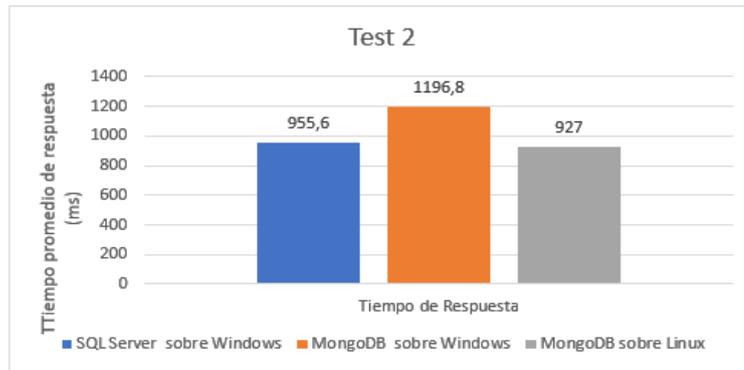


Figura 6.33: Test2 - SQLServer vs MongoDB

Test3.- Procesos judiciales con mayor cantidad de actuaciones judiciales asociadas

SQL Server sobre Windows		MongoDB sobre Windows		MongoDB sobre Linux	
Label	Average	Label	Average	Label	Average
SQL Server Request1	8057	MongoDB Request1	1303	MongoDB Request1	943
SQL Server Request2	922	MongoDB Request2	1192	MongoDB Request2	887
SQL Server Request3	833	MongoDB Request3	1155	MongoDB Request3	897
SQL Server Request4	774	MongoDB Request4	1180	MongoDB Request4	913
SQL Server Request5	713	MongoDB Request5	1160	MongoDB Request5	894
Promedio: 2259,8		Promedio: 1198		Promedio: 906,8	

Tabla 6.14: Promedio - Test3

Para la elaboración de las gráficas al igual que en el Test 1 y Test 2 se obtuvo el valor promedio de los tiempos de respuesta. En este caso las consultas realizadas son utilizando 99 juicios que están dentro de la muestra de mayor cantidad de actuaciones judiciales. Se puede observar en la gráfica 6.34 que existe una variación en los tiempos de respuesta, siendo nuevamente favorable para MongoDB en Windows y Linux..

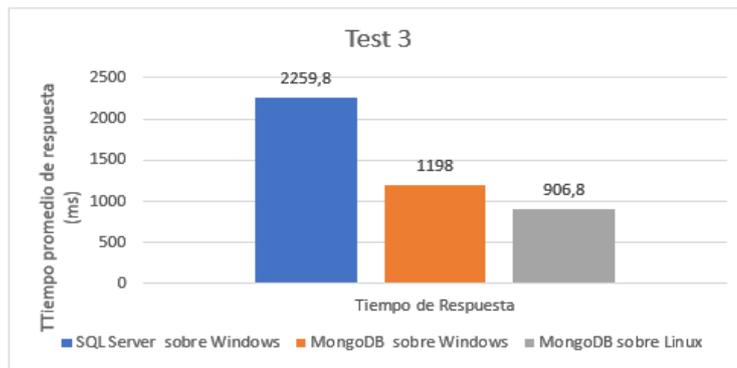


Figura 6.34: Test3 - SQLServer vs MongoDB

Como puede verse en las Figuras 6.32, 6.33 y 6.34, la diferencia entre SQL Server y MongoDB es alta. Los tiempos promedios para obtener los resultados de las consultas son constantes en MongoDB y el tiempo utilizado para resolver las primeras consultas es menor que SQL Server.

MongoDB ofrece mayor rendimiento sobre el sistemas operativo Linux. Mientras que SQL Server tiende a incrementar los tiempos de respuesta cuando a cantidad se refiere. Cabe recalcar que las consultas no tienen complejidad alguna, pues el análisis es con respecto a la cantidad de registros que tiene la tabla en cuestión y el peso y cantidad de actuaciones judiciales asociadas al proceso judicial.

Contraste de Hipótesis

Hipótesis Nula (H_0).-¿Los sistemas de gestión NoSQL **NO** inciden en los tiempos de respuesta de consulta de causas en la Función Judicial del Ecuador.?

Hipótesis Alternativa (H_1).-¿Los sistemas de gestión NoSQL **SI** inciden en los tiempos de respuesta de consulta de causas en la Función Judicial del Ecuador.?

La comprobación de hipótesis se realizó basado en los valores obtenidos de los tiempos de respuesta de los sistemas gestores de datos sobre el sistema operativo Windows. Para su comprobación se desarrolló un contraste de igualdad de varianzas, las hipótesis se han planteado como un contraste unilateral de la siguiente manera:

H_0 : La varianza de X es igual que la de Y

H_1 : La varianza de X es diferente que la de Y

$$\left. \begin{array}{l} H_0 : \sigma_x^2 = \sigma_y^2 \rightarrow H_0 : \sigma_x^2 / \sigma_y^2 = 1 \\ H_1 : \sigma_x^2 \neq \sigma_y^2 \rightarrow H_1 : \sigma_x^2 / \sigma_y^2 \neq 1 \end{array} \right\} \rightarrow \alpha = 0.05$$

Desarrollando con el método simplificado del contraste, tenemos que:

$$\frac{S_{mayor}^2}{S_{menor}^2} > F_{n_{num}-1, n_{denom}-1, \alpha/2} \Rightarrow SE RECHAZA H_0$$

Comprobación Test 1.- Para contrastar las varianzas se ha utilizado los siguientes datos:

Sobre Windows		
	SQL SERVER	MONGODB
μ	900,600	1169,000
S	169,283	36,094
S^2	28656,640	1302,800
n	5	5

Tabla 6.15: Datos necesarios para contrastar la hipótesis - Test 1

$$\frac{28656,640}{1302,800} > F_{4,4,0,025}$$

$$21,9962 > 9,605 \Rightarrow SE RECHAZA H_0$$

Comprobación Test 2.- Para contrastar las varianzas se ha utilizado los siguientes datos:

Sobre Windows		
	SQL SERVER	MONGODB
μ	955,600	1196,800
S	143,405	62,426
S^2	20565,040	3896,960
n	5	5

Tabla 6.16: Datos necesarios para contrastar la hipótesis - Test 2

$$\frac{20565,040}{3896,960} > F_{4,4,0,025}$$

$$5,2772 < 9,605 \Rightarrow NO SE RECHAZA H_0$$

Comprobación Test 3.- Para contrastar las varianzas se ha utilizado los siguientes datos:

Sobre Windows		
	SQL SERVER	MONGODB
μ	2259,800	1198,000
S	2899,420	54,181
S^2	8406637,360	2935,600
n	5	5

Tabla 6.17: Datos necesarios para contrastar la hipótesis - Test 3

$$\frac{8406637,360}{2935,600} > F_{4,4,0,025}$$

$$2863,6862 > 9,605 \Rightarrow SE RECHAZA H_0$$

De los resultados analizados se puede ver que los tiempos promedios presentados por SQL Server en los en test 1 y el test 2 son menores a los presentados por MongoDB, sin embargo, al obtener la desviación estandar del test 1 y test 2 en SQL Server se puede notar que hay un dispersión de datos muy amplia con respecto al promedio siendo estos poco fiables. Mientras que los resultados de MongoDB son más fiables ya que se encuentran más agrupados. En el test 3 por lo contrario SQL Server presenta una promedio más elevado en comparación con MongoDB pero de igual forma que en el test 1 y 2 los datos se encuentran dispersos.

6.7.6. Conclusiones

Se observa en los resultados de las pruebas realizadas que los tiempos de respuesta de SQL Server en las primeras consultas es muy elevado sin embargo luego de varias consultas tienden a disminuir esto se debe a que los datos ya no son procesados en disco si no memoria, sin embargo, MongoDB con respecto a SQL Server toma menos tiempo en resolver las primeras consultas manteniendo un tiempo de respuesta poco variable que tiende a disminuir.

Uno de los aspectos más importantes que se puede destacar es la variación en los tiempos de respuesta que presenta el sistemas gestor de datos NoSQL MongoDB con respecto al sistema operativo donde se encuentra instalado, notando un mejor desempeño en sistemas operativos Linux esto a pesar de que MongoDB es un sistema multiplataforma.

SQL Server posee una estructura de tablas para almacenar los datos y MongoDB utiliza el formato de documentos JSON, para lograr migrar la información contenida en SQL Server a MongoDB fue necesario un procesos de extracción, transformación y carga (ETL) dentro del cual se estructuró los documentos JSON que posteriormen-

te fueron almacenados dentro del motor no relacional. Para el proceso de migración se consideró un análisis previo de la data a migrar concluyendo que para este caso de estudio la estructuración de los documentos JSON presentaban una complejidad media pues el desarrollador se está iniciando en esta tecnología.

6.7.7. Recomendaciones

El optar por sistemas gestores de datos NoSQL conlleva mejoras en cuanto a rendimiento y escalabilidad sin embargo es necesario establecer los requerimientos mínimos en cuanto a hardware y software sobre el cual estaría soportado el sistema gestor de datos, en base a los resultados obtenidos del caso de estudio es recomendable la instalación de MongoDB sobre una plataforma Linux.

La implementación de nuevas tecnologías en este caso un sistema gestor de datos NoSQL en entornos de producción conlleva tareas de administración por lo que es recomendable se disponga de personal capacitado dentro de la institución, también es importante contar con el soporte de empresas certificadas quienes garanticen el correcto funcionamiento y respondan a posibles eventualidades con el sistema.

Para este caso de estudio la migración de datos tuvo una complejidad media, sin embargo es recomendable hacer un análisis profundo sobre las particularidades en cuanto a la data que procesa cada una de las instituciones, esto en vista de que el proceso de migración de datos de una base relacional a una no relacional, puede llegar a ser una tarea sencilla o compleja que conlleve mayor esfuerzo y tiempo. Se recomienda también se analice la estructura de los documentos que almacenarán la data a migrar buscando la mejora en los tiempos de respuesta.

Referencias

Alcalde, A. (2016). Diseño de Bases de Datos (I) - Conceptos del Modelo Relacional. Descargado de <https://elbauldelprogramador.com/disenio-de-bases-de-datos-i-conceptos/>

Asamblea Nacional de la República del Ecuador. (2015). Código orgánico general de procesos. Descargado de: <http://www.funcionjudicial.gob.ec/pdf/CODIGO%20ORGANICO%20GENERAL%20DE%20PROCESOS.pdf>

Asamblea Constituyente de la República del Ecuador. (2008). Constitución del Ecuador. Descargado de <http://www.asambleanacional.gov.ec/documentos/constituciondebolsillo.pdf>

Britos, L., Di Gennaro, M., Gil Costa G., Kasián, F., Lobos, J., Ludueña, V., Molina, R., Printista, A., Reyes, N., Roggero, P., Trabes, G. (2016). Búsquedas en grandes volúmenes de datos. En XVIII workshop de investigadores en ciencias de la computación (wicc 2016, entre ríos, argentina).

Función Judicial de la República del Ecuador. (2009). Código orgánico de la función judicial. Descargado de <https://www.oas.org/juridico/mla/sp/ecu/specu-int-text-cofj.pdf>

García, H., Matus, J. (s.f.). ESTADÍSTICA DESCRIPTIVA E INFERENCIAL I, FASCÍCULO 1. ANTECEDENTES HISTÓRICOS DE LA ESTADÍSTICA Y SUS FUNCIONES. Descargado de http://www.conevyt.org.mx/bachillerato/material_bachilleres/cb6/5sempdf/edin1/edin1_f1.pdf

Grolinger, K., Higashino, W., Tiwari, A. y AM Capretz, M. (2013). Data management in cloud environments: NoSQL and NewSQL data stores. Descargado de <http://journalofcloudcomputing.springeropen.com/articles/10.1186/2192-113X-2-22>

Herranz Gómez Raúl (2014). BASES DE DATOS NOSQL: ARQUITECTURA Y EJEMPLOS DE APLICACIÓN. Descargado de: https://e-archivo.uc3m.es/bitstream/handle/10016/22895/PFC_raul_herranz_gomez_2014.pdf?sequence=1&isAllowed=y

Judd, Charles y, McClelland, Gary (1989). Data Analysis. Harcourt Brace Jovanovich. ISBN 0-15-516765-0.

Kaur, K. y Rani, R. "Modeling and querying data in NoSQL databases," in Big Data, 2013 IEEE International Conference on, 2013, pp. 1-7.

Kollapur, P. (2011, 12). Migration of relational data structure to cassandra (no sql) data structure. Descargado del <https://www.codeproject.com/articles/279947/migration-of-relational-data-structure-to-cassandr>

Kumar, R., Parashar, B., Gupta, S. y Gupta, N. (2014). Apache Hadoop, NoSQL and NewSQL Solutions of Big Data. Descargado de https://www.researchgate.net/publication/268449070_Apache_Hadoop_NoSQL_and_NewSQL_Solutions_of_Big_Data

María Florencia Pollo-Cattaneo y Marcelo López Nocera y Giovanni Daián Rotoli. (2014). Rendimiento de tecnologías nosql sobre cantidades masivas de datos. Cuaderno Activa, 6 , 11-17. Descargado de <http://ojs.tdea.edu.co/index.php/cuadernoactiva/article/view/197/199>

Mendenhall, W., Beaver, R. y Beaver, B. (2010). Introducción a la probabilidad y estadística. Descargado de http://investigadores.cide.edu/aparicio/data/refs/Mendenhall_Prob_Estadistica_13.pdf

Ministerio de Tecnologías de la Información y las Comunicaciones de la República de Colombia. (s.f.). Estrategia de gobierno en línea. Disponible en <http://estrategia.gobiernoenlinea.gov.co/623/w3-propertyvalue-7650.html> Última visita: Marzo 2017.

Microsoft (2017). Fundamentos de la normalización de bases de datos. Descargado de <https://support.microsoft.com/es-cl/help/283878/description-of-the-database-normalization-basics>

Moreno, A. (2013). Bases de datos: Modelos de datos. Descargado de <http://elies.rediris.es/elies9/4-2.htm>

Moreno, F., Quintero, J. y Rueda. R. (2016). UNA COMPARACIÓN DE RENDIMIENTO ENTRE ORACLE Y MONGODB. Descargado de: http://www.scielo.org.co/scielo.php?pid=S0124-81702016000100007&script=sci_arttext&tlng=es

Okman, Lior; Gal-Oz, Nurit; Gonen, Yaron; Gudes, Ehud; Abramov, Jenny; , "Security Issues in NoSQL Databases," Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on , vol., no., pp.541-547, 16-18 Nov. 2011 doi : 10.1109/TrustCom.2011.70

Pablos Heredero, C. (2004). Ilustraciones de la aplicación de las tecnologías de información en la empresa española. ESIC Editorial. Descargado de <https://books.google.com.ec/books?id=cswZknbMz0MC>

Rodríguez, F. (s.f). Elementos de estadística. Descargado de http://ingenieria.udea.edu.co/eventos/enim/estadistica/Documento_soporte_estad%EDstica_b%E1sico.pdf

Salazar, J. (2014). ANÁLISIS COMPARATIVO DE DOS BASES DE DATOS SQL Y DOS BASES DE DATOS NO SQL. Descargado de <http://repositorio.utp.edu.co/dspace/bitstream/handle/11059/5119/0057565S161.pdf?sequence=1>

Sarria, F. (s.f). Programación en SQL con PostgreSQL. Descargado de <http://www.um.es/geograf/sigmur/sigpdf/postgresql.pdf>

Secretaría Nacional de la Administración Pública de la República del Ecuador. (2014). Plan nacional de gobierno electrónico 2014 - 2017. Descargado de <http://ec.okfn.org/files/2014/12/PlanGobiernoElectronicoV1.pdf>

Tolosa, G. H., Bancharo, S., Rissola, E. A., Delvechio, T., y Feuerstein, E. (2016). Grandes datos y algoritmos eficientes para búsquedas de escala web. En XVIII workshop de investigadores en ciencias de la computación (WICC 2016, entre ríos, argentina).

Universidad de Valencia (s.f). Modelos de probabilidad. Descargado de <https://www.uv.es/ceaces/base/modelos%20de%20probabilidad/MODEPR1.htm>

Valenzo, M., Valencia, R. y Castro, J. (2013). Integración de búsquedas de texto completo en bases de datos nosql. Revista Vínculos, 8 (1), 80– 92. Descargado de <http://revistas.udistrital.edu.co/ojs/index.php/vinculos/article/view/4177>

ANEXOS

Anexo 1

Procesos judiciales mayor cantidad de actuaciones judiciales asociadas. Se adjunta solo una muestra de los datos

Cantidad	NoProceso	Cantidad	NoProceso	Cantidad	NoProceso
2989	10*4**00900*0	334	0**0**01374*1	289	13*41*0110038
2774	10*54*008094*	334	0**0**01310719	289	09911*0090*90
2388	07**1*014000*	334	09*8**0130111HP	288	0**0**013****
1508	07*51*014000*E	334	09*8**013134*	285	1**8**0140111
1392	07*53*0140005E	331	0933**014*87*7	285	09*81*013*4**
1305	07*58*01401*7	330	0**0**01305*9	284	09901*01*013*
1168	07*55*01*0187	328	0995**0101311	284	09908*01*01*4
826	09*8**0130030	328	0**0**013539*	284	0**0**0130915
545	09*8**0131498	326	09907*0100144	284	09911*0100*47
451	09901*01001*4	326	09903*01*01*3	283	09903*0090041
449	14111*01*0*78	325	099*1*01*0179	282	09*8**01405*5
444	09911*0110180	324	0**0**013*870	282	09907*0110189
443	09*8**0130439	321	099*1*010131*	282	13*41*01000*8
441	11100*001000*	320	099***0100358	281	09*74*000*51*
437	09911*011013*	320	131***0111*19	281	0**0**013878*
432	09*8**0131**0	319	09951*013118*	281	09*8**01304*1
432	0**0**013*9*9	319	0**0**0138339	280	0991**0100*1*
430	09903*0110031	318	09907*01*00*0	279	09*85*01315444
416	13*41*0090044	316	13*41*0100015	278	0**0**0138*18
412	13454*01*0003	315	09907*0090*7*	278	01*13*0140373
402	09*8**0130*79	314	09**3*011087*	278	0**0**0138539
401	0**8**014*110	312	09*8**0133*501	278	0**0**013*531
400	09907*01*0198	311	13*41*010007*	276	099*4*009**5*
389	13*41*01100*3	311	0933**014*8807	276	09*8**013033*
388	09907*0090348	310	13*41*0090048	275	09*8**0139*7*
387	07*55*0130001R	307	11*03*0140784	274	0**0**013144**
376	0**0**0138599	306	0**0**0138*57	273	09*81*013*505
376	13*41*01300*0	306	07953*0130079	273	0*501*00*0118
373	0990**0130*08	305	0**0**0138*03	271	09*8**013*3*5
373	0**0**013*505	304	131***0110911	271	09903*01*0017
371	09*8**0131438	302	0991**01*00*8	271	09907*01*0044
364	09903*01*0**4	300	0933**0143403	270	0**0**0133084
360	0**0**01315**	300	099*1*0110479	270	09957*0091*9*D
358	0**0**0131533	300	0**0**0131*3*	269	09907*0110070
352	07951*0130*19	299	099*4*009370*	268	0**0**01311517
352	0**0**013*89*	297	0**0**0131*3*	268	09****0110919
350	09907*01*00**	297	09911*01*0084	267	09905*0130188
349	07*4**01*0*51	295	07954*0100333	267	0**0**0131471*
347	09903*003039*	295	0995**0110594	266	09903*0110078
346	13*41*0130011	294	0**0**0131718	266	09*8**0131094
346	11903*01301**	294	5*5**011100000	266	0*30**0110001
345	09*85*0130118	293	07951*0090978	265	0933**014*94*5
344	0**0**0131*13	292	099*4*0101090	265	11*03*0139*33
344	0990**01*0071	292	09901*01*011*	265	11*03*01378*9
342	099*4*010151*	291	09907*0100***	264	099*0*0090980
334	09*8**013378*1	289	09905*0130148	264	10*41*00801*7
264	0933**014*508	248	0**0**0139*17	233	09957*009**00
263	09*8**0131155	248	099*4*009*774	233	11*03*0139*99

263	11*03*013*1**	248	0**0**0134559	232	09911*013018*
263	099*5*01*1491	247	0995**009*94*	232	13177*01400**
263	0**0**0135743	247	0795**0100881	232	09305*0070507
263	09*85*01370*9	247	09959*009*344D	231	099*4*0090984
262	09*8**013**541	247	099*4*0100*11	231	09*8**01339*3*
262	0995**0130090	246	11*03*0134*0*	231	0**0**013080*
262	0**0**01313*8	246	0**0**0139591	231	09903*0130151
262	09957*011089*	246	099*1*00931*5C	231	0**0**0133***
262	0933**0147**3	246	0**0**0135757	231	099*1*009498*
262	09907*0100003	245	09*8**0130058	230	0933**0141*8*0
261	0933**014*5*7*	245	099*4*0111308	230	09904*0130180
260	10303*0030189	244	09*8**013**0*	230	099*4*010038*
260	09*81*01319*7	244	0**0**013**87	230	05100*01*0013
260	0933**01459*49	244	0**0**0138481	230	09911*010019*
260	14*5**0100009T	243	0995**010137*	230	0**0**0131573
260	09*8**0130718	243	099*1*0101099	229	09910*0110145
259	09*8**0130*08	243	0933**014*9150	229	09903*01*0138
259	11*03*013954*	243	9953*0130000000000	229	0**0**01319**
259	0**0**0131134	242	0933**01411078	228	0**0**01313*7*
258	09904*01*001*	242	13454*01*0030	228	0995**0094591D
257	0**0**013*9*0	242	07953*01*0**1	228	09907*01*0199
257	09*8**0130584	241	0**0**013184*	228	09*85*013*885
256	01*04*0139034	240	0**0**013*09*	227	1**41*0130039
256	0**0**013*510	240	0991**01100**	227	0995**0030190D
255	0**0**0131511	240	13*83*014047*	227	933**01444
255	0**0**0138508	239	0933**01415745	227	05*81*013**54
255	0991**0130198	239	09907*0100108	227	07953*01307*0
255	09*8**01309*8	239	0995**0110**5	226	0933**0141530*
255	099*4*0090858	238	**0**013100000	226	09901*00908*9
254	0**0**0131*97	238	08*01*013193*7	226	0995**009*199
254	0990**013017*	238	09903*0090739	225	05*41*01*0007
254	09907*0110090	237	0995**0130840	225	099*1*00915*9
254	0933**0143183	237	0930**0040*18	225	07954*00*0199
253	0**0**013157*	236	09*8**013033*	225	13*41*008001*
253	13*41*0080011	236	09907*01101*8	225	099*4*011131*
252	0995**0140341	235	09907*01*0058	225	13454*0110033
252	0**0**0131*33	235	09*8**0130807	224	0**0**0131*90
252	099*4*01014**	235	093*9*0030573	224	09907*01*003*
252	0**8**0141739	235	0990**011018*	224	0**0**0137793
251	0933**01455774	234	0**0**013**04	224	0**0**013*313
251	09*8**0130189	234	099*0*0110*91	223	099*1*0111371
251	09908*0110*55	234	0933**01459440	223	0990**0100133
250	0**0**013*3**	234	0**0**013**97	223	099*4*010070*
249	09903*01*00*5	234	0933**01419*5*	223	0**0**0131093
249	0**0**0130500	233	11803*0130475	223	14*01*0135945
223	099*4*01*0430	213	099*4*009*11*	207	11903*01*0078
223	0931119990584	212	09*8**0130*1*	207	09403*01*0058
223	09333*0130751	212	0**0**0131*39	207	0933**014*9*98
222	0**0**013110*	212	09907*0130194	207	11*03*0137*89
222	09911*01*0170	212	09*85*01318**	206	09959*0111*8*
222	09901*0100140	212	0**0**013*80*	206	0933**0144993*
222	09309*0000401	212	0**0**013*714	206	0**01*01*0093
221	09901*0110179	212	0**0**0130908	206	0**0**01304*9
220	09907*01*0130	212	0**0**0130*09	206	**0**0131500000000
220	0991**0130*11	212	09951*01310*8	206	0**0**0131789

220	0**0**0130**8	211	09907*01101*7	206	0**0**013349*
220	13454*0100055	211	0933**0144911	205	11*03*0137834
220	09903*0100*54	211	0**0**0130898	205	0**0**0131834
220	09903*01400*8	211	07953*00*07*7	205	09*74*0010080
220	09907*01301*0	211	09*8**0131378	205	11*59*0140017
219	099*1*0094*05	211	07953*00901*0	205	09908*0140007
219	0995**0100183	211	093*9*0040*48	204	9.33**0143E+6*
218	0**0**013841*	211	09*8**0130*3*	204	0**0**0135407
218	099*1*01*0485	211	11903*01*0089	204	0**0**0134549
218	09333*013071*	210	099*0*0100490	204	07951*0130*57
218	09307*00*0349	210	09907*01*0178	204	0**0**0138344
217	11801*0130078	210	07*54*01400*4	204	0**0**01354*5
217	09903*01*0173	210	09907*0110194	204	09901*01*0090
216	11*03*014081*	210	13*41*0130039	204	09903*0110033
216	0**0**0135*50	210	0**0**013*934	204	0**0**0131151
216	09*8**013*05*	210	099*4*0091474	204	0933**01453774
216	09910*0130173	210	0**0**013*9**	204	0995**01400**
216	11*03*0138851	210	093*9*00*05*7	204	0**0**0130555
216	0795**01103*9	210	0**0**0131848	204	07*4**01303*5
215	099***01*013*	210	0995**009347*	204	09*8**013*890*
215	14305*0110174	209	0**0**0130407	204	09*85*01310047
215	099*1*0110954	208	0**0**0131***8	203	0**0**013*5*9
215	13*41*0070019	208	09*85*0131348	203	0**0**0130*81
215	0**0**013*5*4	208	0**0**013133*5	203	07951*00804**
214	11*03*0138951	208	0**0**013*535	203	0933**01439803
214	0**0**0138*78	208	099*5*01*0*38	203	0**8**0130531
214	11*03*0139157	208	09****01*007*	203	099*1*0090095
214	09959*0100148	208	0**0**0131559	202	0**0**0138478
214	0**0**0138*37	208	09*8**0133053	202	09903*01*0075
214	0**0**0131*750	208	09309*0000*95	202	099*1*01*0081
214	11*03*013*0*4	208	0**0**0131*447	202	0**0**01311147
213	0**0**0130**0	208	0**0**013*18*	202	09905*0130173
213	0**0**01385*8	208	099*4*0110830	202	0933**014*0*40
213	09301*00*015*	207	05*41*01101*7	202	0995**0111100
213	09959*01*0410	207	0990**010018*	202	0933**014*48**
213	13*41*01*0083	207	09957*01*0150	202	099*4*009**54
213	11*03*014000000	207	0795**01109*7	202	09*84*0141403
201	09911*0110185	196	0**0**0135901		
201	0**0**0130*77	196	0*95**00910*8		
201	05*41*01*0045	196	09*8**01301*0		
201	099*4*009000*	196	0**0**01397**		
201	09311*00*00*9	196	0**01*01*011*		
201	0**0**01314055	196	0933**014*0484		
201	01351*0050048	196	09903*0100155		
201	11*03*013*31*	195	07951*01*0709		
201	099*5*01*0784	195	0933**01459*5		
201	0933**014*1014	195	09951*0140455		
200	0991**01303**	195	0995**009*43*		
200	13*41*00900*0	195	0995**011133*		
200	0995**00949**	195	13*41*01*005*		
200	0**0**013149*9	195	0**0**0131953		
200	0933**01431*97	195	0**0**0131*35		
200	0**0**0137049	195	0933**0141404*		
200	09957*009*81*	195	099*0*0095438D		
200	0933**0148001	195	08*01*013358*		

199	099*1*01*0170	195	01351*00501*7
199	09**0*01*0150	194	0**8**013050*
199	0933**0143470*	194	0933**0147081
199	0**0**0131*50	194	9,33**014E+*8
199	0933**01411*38	194	099*5*0110930
198	0**0**013300*	194	099*4*0110411
198	0**0**0131107	194	1**41*013003*
198	09*85*0131*359	194	099*4*0101*88
198	0995**0091388	194	0933**01431484
197	099*1*011144*	194	0**0**01315*1
197	0*30**0080100	194	0**0**0137971
197	09*8**0130009	194	0**0**013*940
197	0**0**013***3	193	0933**014584*
197	05*81*01313405	193	0**0**0131957
197	0**0**01311771	193	09907*0130078
197	0995**0111975		...
197	099*4*009*59*		
197	0**0**0138570		
197	0**0**013*854		
196	11*03*0141**9		
196	0**0**01317*4		
196	11*03*013945*		
196	0**0**01317*9		
196	13*41*01*0013		
196	0**0**0131*455		
196	0995**009*8*0		
196	09*85*01338*7		
196	099*4*0110198		
196	0**0**013**90		

Anexo 2

Procesos judiciales mayor peso de la suma de actuaciones judiciales asociadas. Se adjunta solo una muestra de los datos

Peso (Bytes)	NoProceso	Peso (Bytes)	NoProceso
7498043	06*0**0140074	1392931	1190**01100*4
3950116	07*61*0140006	1382707	0991**01400*5
3540877	10*54*008094*	1376091	13334*0141370
3255855	09*86*0130611	1363580	0990**0130197
2932016	14111*01*0*78	1362700	09910*01*00*3
2902138	0*10**0130087	1359109	09904*0130180
2893249	14111*0130479	1346659	18461*0140633G
2760311	07*58*01401*7	1337981	9910*0130173
2622943	17951*008*035	1326394	0991**0130*33
2491607	111*1*01*0*33	1324041	09*86*01319616
2393404	1**44*014011*	1316648	09*86*0131498
2361544	09*86*0130030	1314433	09904*01401*0
2233986	09*8**0130110HP	1273176	09*86*013*68**
2196938	09956*01100*4	1272757	09910*01*0083
2172848	0**0**0138478	1248350	13*41*0060045
2098560	07*51*014000*E	1238815	09*81*0133510
2089865	09907*01*0198	1236223	05*41*0110183
2041537	09*86*01336773	1221867	10*43*0130064
1996740	07*53*0140005E	1218312	17*81*014116*
1987732	13*41*0080050	1216861	131***0110911
1927018	13*41*01300*0	1209178	09*63*01*0*38
1865814	09903*0140068	1201777	09901*01*013*
1839454	09*86*0131*154	1176913	09907*00901*1
1745950	09*86*0140541	1170278	10*4**0130051
1740152	09904*0110134	1160910	05*41*0110187
1720203	1190**01*0087	1153102	09*81*0140138
1708603	13177*0140059	1149561	09*86*0130036
1706735	09910*0110103	1148265	09910*0110158
1701864	0**41*0100034	1146519	09907*0130160
1698310	10*81*0140*97	1144502	09*8**0130111HP
1679277	07*55*01*0187	1141596	17*47*0140*78
1659109	10*4**0090060	1133670	09*86*01333670
1658756	1190**0110099	1126492	9911*0130186
1636292	09*86*0133*814	1126326	09*86*0130189
1626126	13*41*0090044	1123663	09*81*0131751
1564792	13177*0140006	1123104	01*83*0141*71
1543049	07951*0090978	1122707	1190**01*0064
1482281	09910*0130166	1116889	09*83*0130065
1482262	0*30**0110001	1113336	13303*01*0354

1473074	09904*0140136	1112063	9*81*0130084
1467401	131***0110957	1109958	1**44*01404*3
1405711	0**41*0070009	1103316	9*61*01400*9
1404686	01*8**01*1808	1102598	17*68*0141516
1399865	09*01*01*8066	1098481	09*86*0133500*
1396287	11903*01*0078	1091171	13301*0130*06
1084724	11903*01301**	883660	05*0**013431*
1079760	08*4**01*00*9	883213	13301*0100469
1077646	09*81*013*505	878726	09906*01*0071
1071901	10*43*0130184	878577	09911*0110136
1063919	09905*0130067	874011	09910*0110178
1062729	17*56*0131489	873324	09907*0130*15
1062116	0**56*01301*3	872308	08*43*0110108
1050317	09914*0140050	871864	0**0**01411*1
1042273	1**53*01*0108	868364	13*41*01*0049
1039255	13303*0130516	867970	09911*0110180
1030333	09907*0100015	864831	09956*0111809
1027968	09905*0130188	862551	13*41*01*0056
1023058	11903*0130007	859619	09*86*0131438
1015017	0**81*0130468	858537	05*41*01*0070
1014114	09*86*0130867	857590	17*6**0141805
997814	09*81*013*4**	857500	0991**0140100
990101	09907*0130194	857242	9*69*0091869
989428	11901*01300*9	855476	09*86*01316085
981340	11801*0130078	851365	09901*0110179
974011	10*01*0131508	848277	131***01*0668
964893	08*41*01*001*	846326	09907*0110070
964256	17*56*0140155	842150	09903*0130094
958363	09*86*01339*3*	839995	131***0130130
951502	13*41*0130039	837426	13*41*01*00*6
946073	08*41*01300*0	832052	09907*0130146
945768	09906*0130*08	827682	09907*0090348
945285	10*43*0140109	824458	13801*0140030
939944	09907*0100144	824073	09910*0100107
938137	08*43*01*00*5	817917	09901*0130*07
935266	09*85*0130570	817403	17*57*0140379
934159	17309*0131083	817101	0**0**01318356
934140	11903*01100*8	815995	09903*0130174
933246	13177*01400*8	815358	09*86*0130439
932190	06953*01300*9	813198	11903*01*0089
930061	09504*0110055	811566	*1*41*010007*
929146	05*41*0110001	806956	933**01438837
927383	131***0111619	805392	1190**0100073
922796	091*1*0130369	804772	09907*0090*76
913594	09*86*0137144	802006	**0**01318393
905411	1033**0130340	801913	0510**01*0058
905035	*0331*0131339	801384	09914*0130*09

897866	08*5**0100387	798796	17*4**0140170
897726	09905*0140077	798518	**41*0130001
897134	09*86*0131*60	795708	09*86*0141665
896460	10*81*013161*	794959	0**51*011041*
884487	06*8**014*110	793870	09*86*013531*
883793	09*85*0138407	785523	09907*0110090
783268	0**0**01317367	714395	0990**01*0189
782555	09456*01*07*6	714141	09903*0090041
779243	0991**01*0058	710936	1**41*010009*
779003	10*43*0140061	710508	0*60**0090184
776517	09*86*01311416	710463	08*53*0111107
776450	11*57*0140*03	709765	0991**014009*
776151	13*41*0110063	708536	09907*0130084
773214	08951*01*0846	707614	09*85*01393*5
773177	093*0*0140081	706968	09904*013017*
772950	09*85*0130791	706099	09907*0130075
772204	0**56*01*0156	706000	1190**01100*4A
772122	0**0**0131*53	703639	09914*0130014
770730	131***01*0577	702893	1**41*01*0147
770502	*1101*003000*	700571	09914*0130*10
768119	09904*0110177	699722	10*43*01300*9
765840	131***011060*	699288	09910*01*0*05
761962	09910*0090305	699198	09*86*0130771
758748	17*56*0140598	695041	16*41*0130039
755611	09907*0100003	694886	09907*0100*6*
753393	17*43*014018*	691924	1**44*01400*7
752110	13176*01401*0	691737	07*5**0140069
750224	9910*0110114	691395	1190**0070013
746463	09*86*0133053	690991	9*86*0130584
745984	17*47*0130403	690743	09907*0110034
744361	09903*01*0138	690664	09907*01*0060
743268	09907*01*0199	690477	05*41*01*004*
739998	9910*0110145	688916	9*81*013169*
738342	05*81*0131380	688092	1*03*014057*
732344	8*4**0140005	687780	09910*0130111
729419	1190**01*0063	687324	13*45*0140033
729024	17*68*0141315	687087	10*81*0130304
727861	9903*0130117	686008	0190**0140071
727416	17*56*0140*77	683697	09907*0140074
724765	17*43*0140164	683298	09*81*0140763
722321	9904*0140098	680966	0933**01440*41
720479	09901*0130193	680962	09*86*014*966
719566	17*6**0141481	679872	10*4**0130057
719055	09910*01*003*	679493	09903*01*0*56
718632	1190**011007*	679232	09903*0110031
718560	*1*5**01*0134	678900	0991**0140113
717943	13176*0140104	678791	1**53*01*0067

717230	*3*81*014309*	676120	09904*0130071
716292	09910*01*0057	675972	06*8**0130531
716137	06331*0130179	675751	09904*0130091
716128	9901*01*0116	675084	9907*0110135
715817	09909*0100010	673536	09910*0110054
714606	991**0140089	673109	0991**0130*48
673032	0991**0130*39	630647	4*51*01400*6
672989	13176*0140109	629143	08*4**01301*1
671501	09907*01101*7	628218	1190**0130033
670203	09906*0130176	628008	09907*0130155
669299	09*86*01336548	627218	09905*0140034
667503	09906*013016*	626786	10*54*01*0811
666746	0930**01*0*15	625963	*1*41*0130173
666530	09907*0130071	625272	09903*01*0195
666399	09907*01*00*4	625247	*1*5**0130041
666387	05*53*0100645	623180	09*69*01*0045
665398	09903*01*0**5	623167	1**8**0140111
663440	0190**0140088	623155	08*43*0130064
663235	09903*0110146	622477	7710*0140157
662970	09904*01*0*18	622467	09904*01*0016
662469	09914*0130*78	622400	09*85*0134137
660874	09907*013019*	621211	09907*0130046
657687	09904*0130188	620687	1190**0130037
656826	13*41*0100068	620352	09907*0110038
656453	09*86*013179*	619303	09*85*01411857
656223	1**57*0140058	618983	09903*0130151
655329	09901*01*0**4	617345	09*86*013*8906
654975	09901*0110146	615192	09*8**01405*5
653699	05*81*0135696	615097	09*86*0136118
653515	09*85*01316359	614484	173*0*0140967
652327	11901*01*0056	614430	0991**0140019
651633	09907*01*00*6	614430	13177*0140131
651418	09901*01*0183	614414	13*41*01*0083
651065	09*86*013134*	614338	13*41*0080046
650716	09*85*013380*	614042	09*85*01300*3
648848	1033**0140190	613582	0645**0130377*
648431	09*6**01*0076	612870	1190**0080030
646280	13461*01*0011	612395	09907*01301*3
646248	09907*0130150	611700	05*41*01101*7
644323	131***01*0576	611453	09*86*0131558
642156	09*8**0130*65HP	611314	09914*014006*
641892	1033**014037*	610605	06*8**0141739
640471	07951*0130389	610542	1190**00700*5
639711	131*1*01*01**	610351	16*41*0110075
638229	09907*0110131	610038	09*63*01*0*5*
637511	17371*0144533	607314	09*86*013*453
636704	09904*0130164	607291	09903*01*00*5

636510	09*86*0133*501	607223	0**41*01*0063
635819	13956*01400*9	606377	05*41*01*0016
635354	09*86*0141470	605080	09*85*0130716
634364	0995**01*0418	604909	17*81*0143541
634213	*3*81*0143398	603900	13334*0140933
632292	09*85*0144755	603609	09905*01*003*
602262	5*81*013870*	575909	10*43*0140*71
601700	17313*0140494	575877	01*8**0130183
601596	06*8**0145185	575788	09503*0110140
601560	09906*0130164	575710	11905*0140006B
600730	09*86*0130394	574208	09*86*013*9955
600271	09903*013005*	571941	07951*0131586
599837	9911*01*007*	571914	09*86*013*505
599778	09907*0130018	571628	09904*01*0195
598874	09904*0140064	570257	08*4**0130099
598058	9907*01*0014	570077	111*1*0130157
596790	09911*0100145	568775	17*6**0141690
595901	11903*01*0093	568695	9*86*0130653
594650	13*65*0130181	568646	09*86*013033*
594423	13*41*0080003	568415	9907*0110189
594062	09901*0100164	568406	13*41*0090048
592931	05*81*0130*11	568346	9901*01*0044
592563	1195**01007*9	567929	09*86*0130118
592339	8*43*0130013	567320	1**44*01401*7
590708	09*86*0131161	566591	990**0110030
590512	09*86*01339635	565599	11901*0130008
590479	09907*01*0180	565311	13177*0140057
590444	09*86*0130160	565294	1033**0140633
589800	09903*0110078	564539	13*41*01*0043
589200	133*0*0140050G	563748	13314*0140061G
588834	05*41*0070115	563247	07951*0130*19
588405	09911*01*0147	562718	09*86*0130356
586823	9905*0110197	562540	09*63*0101751
586662	9907*01*0130	562343	10*41*0130044
586498	09907*01*0037	560249	0991**01*0053
586136	09*86*01434*0	560121	09*86*01300*7
585508	09901*0110115	559780	09*66*0140083
585410	10*4**01*0110	559756	*1331*0130141
585173	11905*0140006	559536	09*86*0138601
583989	09908*0110**8	559007	10*03*0140093G
582710	09904*0080550	558850	5*41*01*0067
581186	9907*01*0044	558547	11901*0130004
580995	09*6**0110454	558424	08*51*01*0403
580380	10*4**01*0039	558387	13176*01401*8
580270	09*66*0130**1	557802	08*43*01*0051
579897	09903*01*0**4	557783	09914*01303*4
579089	09*66*0130339	557249	0**0**0130407

578989	1190**01*0048	556609	09504*0087878
578989	09*85*0141843	555943	09*8**0140146
578926	09910*0090573	555309	18171*0130500
578545	*1*51*0130077	554916	0**0**01354*5
577149	09905*007036*	554503	05*81*0131084
576488	1190**0110046	554349	09914*0140186
554119	09910*01*0*10	546296	09914*0130*46
553688	11901*01*00*0	545268	09*81*014196*
553676	09914*0130314	545010	05*41*01*0*00
553482	13*41*0110080	544950	1190**01*0055
553217	09*61*01301*8FP	544057	17*01*0140011G
553178	09*63*01*0*63	543725	9*85*0131735
552890	09903*0110*39	542782	09*86*0130875
551156	09907*013016*	542196	17*6**0141941
550903	09907*0130154	541944	13*41*01*0013
550741	9907*0140016	541721	09914*0140367
550630	07*4**01*0*51	541364	13*41*01*00*1
549526	1033**01410*6	540997	09910*0110005
548645	91*1*0140068	540964	13454*01*0003
548468	0990**01*00*7	539601	0991**0130358
548468	09*63*0110665	538804	1190**0110058
547704	13*63*0140105	538748	09906*0140100
547327	13*41*01*0004	538693	1190**010008*

Anexo 3

Aggregate Graph: Esta componente permite obtener resultados más precisos de las peticiones realizadas al servidor de pruebas. Los datos que se presentan son:

- Etiqueta (Label) : etiqueta de la muestra
- #Muestras (Samples): cantidad de Thread utilizados para la prueba.
- Media
(Average): tiempo promedio en milisegundos para un conjunto de resultados.
- Mediana (Median): valor en tiempo del percentil 50.
- Línea de 90% (90%
line): máximo tiempo utilizado por el 90% de la muestra, al resto de la misma le llevo más tiempo.
- Min: tiempo mínimo de la muestra de una determinada URL.
- Max: tiempo máximo de la muestra de una determinada URL.
- %Error: porcentaje de requerimientos con errores.
- Rendimiento
(Throughput): rendimiento medido en los requerimientos por segundo / minuto / hora.
- KB/sec: rendimiento medido en Kbytes por segundo.

Anexo 4

Test 1 .- 99 Procesos judiciales seleccionados de forma aleatoria sin importar cantidad o pesos de las actuaciones judiciales asociadas.

Peso (Bytes)	Cantidad	NoProceso	Peso (Bytes)	Cantidad	NoProceso
7726	19	01*04*01313*5	20508	19	09331*009077*
1723	2	01*81*01*3083C	24834	19	0933**014100*8
12377	19	01401*0130084	15599	19	0933**0141*997
6837	19	01606*01*0361	6561	19	0933**0141340
1320	1	0**0**0146376	14969	19	0933**01415918
14304	19	0**81*0130951	6305	19	0933**014395*
17104	19	0*30**0100038	8610	19	0933**01440950
25434	19	03*8**0150045G	14250	19	0933**0145395
9615	19	05*0**013*158*	9184	19	0933**0146**53
15839	19	05*0**0137568	23669	19	0935**01*0936
38632	19	05*51*00403*3	47280	19	09357*0110514
2548	2	05*5**0111*63	733	2	09403*0040193
2079	2	0530**010011*P	981	2	09404*0100074
23209	19	05304*0080056	20714	19	0957**013*694
393	2	06*0**0130*18	39505	19	09904*01*0055
40949	19	06571*0140599	35398	19	09907*0090409
20569	19	06571*0141539	17823	19	09951*009*06*
10197	2	06571*0141806	34187	19	09955*0140166
15142	19	07*01*0140769	8458	19	09956*0093647
457	2	07*55*0010041	9460	19	09959*01*0194
18869	19	07315*0140083	19093	19	09963*0110404
1686	2	07333*015079*G	17173	19	09966*0140*19
26889	19	07571*014*317	122	1	09968*0110181HA
33836	19	0795**013117*	9560	19	09969*01*13*5
16092	19	07953*010085*	26031	19	09969*014003*
23600	19	07953*01306*6	19117	19	10*03*0140781
31506	19	08953*0100009	68393	19	10571*0140001
16514	19	09*01*0140083G	222092	441	11100*001000*
20952	19	09*01*0146767	21936	19	11151*0140*88G
17343	19	09*08*0148087	15809	19	11*03*01310*66
25166	19	09*60*00906*7	19257	19	11*03*013*343
536	2	09*69*0100859	19030	19	11*03*0150481
125	1	09*73*0090198PF	20304	19	11371*0140101
15581	19	09*84*013075*	573	2	11401*011108*C
7016	19	09*84*01311910	1618	2	1145**011007*T
24599	19	09*84*01411041	21643	19	1**03*01*1397
815358	443	09*86*0130439	35290	19	13151*0130719
33376	19	09*86*0130905	4381	2	13316*0070*65
14457	19	09*86*0140665	1829	2	13451*01*03*3E
6936	19	0930**0030636	21238	19	14*54*0140059
357	2	09303*0070093P	14217	19	16*01*014**75

7289	19	09305*0070196	19954	19	16*81*0140348
7719	19	09307*0080115	18809	19	16331*0143113
17028	19	09308*0090*13	68998	19	17*01*0141188
14006	19	09309*0040*37	1009	2	17*30*0150595
648	19	09310*0010505	25996	19	17986*014093*
89065	19	093*6*0130448PF	23999	19	18*0**013*01*4
6661	8	093*8*0030*67	33654	19	18571*014198*
2459	2	*3*81*0150566G			
2027	1	*3331*013733*			
48186	19	*3571*0150*50			

Anexo 5

Test 2.- 99 Procesos judiciales seleccionados de forma aleatoria del grupo de mayor peso de la suma de los bytes de las actuaciones judiciales asociadas.

Peso (Bytes)	NoProceso	Peso (Bytes)	NoProceso
1404686	01*8**01*1808	1148265	09910*0110158
1123104	01*83*0141*71	1362700	09910*01*00*3
2902138	0*10**0130087	716292	09910*01*0057
783268	0**0**01317367	1272757	09910*01*0083
2172848	0**0**0138478	1482281	09910*0130166
1405711	0**41*0070009	1337981	09910*0130173
1701864	0**41*0100034	1126492	09911*0130186
1062116	0**56*01301*3	1326394	0991**0130*33
1482262	0*30**0110001	1382707	0991**01400*5
1236223	05*41*0110183	1050317	09914*0140050
738342	05*81*0131380	53599	09953*0096891
1543049	07951*0090978	2196938	09956*01100*4
1079760	08*4**01*00*9	1170278	10*4**0130051
1399865	09*01*01*8066	1221867	10*43*0130064
1103316	09*61*01400*9	1071901	10*43*0130184
1209178	09*63*01*0*38	3540877	10*54*008094*
1112063	09*81*0130084	1698310	10*81*0140*97
1123663	09*81*0131751	905411	1033**0130340
1238815	09*81*0133510	2491607	111*1*01*0*33
1153102	09*81*0140138	1392931	1190**01100*4
2233986	09*8**0130110HP	1658756	1190**0110099
1116889	09*83*0130065	1122707	1190**01*0064
1149561	09*86*0130036	1720203	1190**01*0087
1126326	09*86*0130189	1396287	11903*01*0078
3255855	09*86*0130611	1084724	11903*01301**
1839454	09*86*0131*154	2393404	1**44*014011*
53600	09*86*0131*773	1109958	1**44*01404*3
1316648	09*86*0131498	1042273	1**53*01*0108
1324041	09*86*01319616	1216861	131***0110911
1273176	09*86*013*68**	1467401	131***0110957
1636292	09*86*0133*814	1564792	13177*0140006
1133670	09*86*01333670	1708603	13177*0140059
1098481	09*86*0133500*	1248350	13*41*0060045
2041537	09*86*01336773	1987732	13*41*0080050
1745950	09*86*0140541	1626126	13*41*0090044
76674	09571*0148*87	1927018	13*41*01300*0
568346	09901*01*0044	687324	13*45*0140033
1201777	09901*01*013*	1091171	13301*0130*06
1363580	0990**0130197	1113336	13303*01*0354
1740152	09904*0110134	1039255	13303*0130516
1359109	09904*0130180	1376091	13334*0141370

1314433	09904*01401*0	2932016	14111*01*0*78
1473074	09904*0140136	2893249	14111*0130479
1063919	09905*0130067	1141596	17*47*0140*78
1176913	09907*00901*1	1062729	17*56*0131489
2089865	09907*01*0198	1102598	17*68*0141516
1146519	09907*0130160	1218312	17*81*014116*
1706735	09910*0110103	53598	17453*0140**4
2622943	17951*008*035		
109016	18461*0131606*		
1346659	18461*0140633G		

Anexo 6

Test 3 .- 99 Procesos judiciales seleccionados de forma aleatoria del grupo de mayor cantidad de actuaciones judiciales asociadas.

Peso (Bytes)	NoProceso	Peso (Bytes)	NoProceso
330	*22*22**3*529	300	*93322**434*3
284	*22*22**3*9*5	451	*99**2*****64
334	*22*22**3**7*9	292	*99**2**2***6
344	*22*22**3*2*3	284	*99**2**2**32
360	*22*22**3*526	347	*99*32**3*396
358	*22*22**3*533	430	*99*32*****3*
300	*22*22**3*632	326	*99*32**2**23
297	*22*22**3*636	364	*99*32**2*264
294	*22*22**3*7*8	289	*99*52**3**48
288	*22*22**32266	344	*99*62**2**7*
373	*22*22**325*5	373	*99*62**3*2*8
324	*22*22**3287*	315	*99*72**9*276
352	*22*22**32896	388	*99*72**9*348
328	*22*22**35396	326	*99*72*****44
432	*22*22**36929	291	*99*72****262
334	*22*22**3746*	350	*99*72**2**26
306	*22*22**38257	318	*99*72**2**6*
319	*22*22**38339	400	*99*72**2**98
376	*22*22**38599	284	*99*82**2**24
305	*22*22**386*3	289	*99**2**9*29*
294	*52522****665	437	*99**2*****36
401	*62822**42***	444	*99**2*****8*
349	*72422**2*25*	297	*99**2**2**84
1508	*725*2**4***2E	302	*99*22**2**28
1392	*72532**4***5E	319	*995*2**3**82
1168	*72552**2**87	328	*99522*****3**
387	*72552**3****R	295	*99562****594
1305	*72582**4**27	321	*996*2****3*2
2388	*726*2**4***6	300	*996*2****479
293	*795*2**9*978	325	*996*2**2**79
352	*795*2**3*2*9	320	*99622****358
306	*79532**3**79	299	*99642**937*2
295	*79542****333	292	*99642*****9*
314	*92632****872	342	*99642****5*2
285	*928*2**32422	2774	**2542**8*942
334	*92822**3****HP	441	*****2*****2
345	*92852**3***8	307	**2*32**4*784
826	*92862**3**3*	346	**9*32**3**22
443	*92862**3*439	285	*22822**4****
402	*92862**3*679	304	*3*222****9**
432	*92862**3*26*	320	*3*222****6*9

334	*92862**3*342	416	*324*2**9**44
371	*92862**3*438	310	*324*2**9**48
545	*92862**3*498	316	*324*2*****5
312	*92862**3325**	311	*324*2*****72
334	*92862**33782*	289	*324*2*****38
331	*93322**428727	389	*324*2*****63
311	*93322**4288*7	346	*324*2**3****
376	*324*2**3**2*		
412	*34542**2***3		
449	*4***2**2*278		