

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN

Facultad de Ingeniería

Escuela Profesional de Ingeniería en Informática y Sistemas

**“RENDIMIENTO DE LA BASE DE DATOS DEL SISTEMA WEB DE
REGISTRO DE POSTULANTES UTILIZANDO EL MODELO
RELACIONAL Y NO RELACIONAL EN LA OFICINA
DE ADMISIÓN DE LA UNIVERSIDAD NACIONAL
JORGE BASADRE GROHMANN,
TACNA – 2015”**

TESIS

Presentada por:

Bach. Ayrtonn Frankho Meléndez Quispe

Para optar el Título Profesional de:

INGENIERO EN INFORMÁTICA Y SISTEMAS

**TACNA – PERÚ
2021**

**UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN
FACULTAD DE INGENIERÍA**

**JURADO CALIFICADOR Y CALIFICACIÓN DE LA SUSTENTACIÓN DE
TESIS**

TESIS N°: _____

TÍTULO PROFESIONAL DE:
Ingeniero en Informática y Sistemas

La Secretaria Académica de la Facultad de Ingeniería, por resolución de Facultad N° 03238-2015-FAIN/UNJBG, designó jurado para la sustentación de la Tesis titulada: "RENDIMIENTO DE LA BASE DE DATOS DEL SISTEMA WEB DE REGISTRO DE POSTULANTES UTILIZANDO EL MODELO RELACIONAL Y NO RELACIONAL EN LA OFICINA DE ADMISION DE LA UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN, TACNA – 2015".

El mismo que está conformado por:

Presidente : Dr. Erbert Francisco Osco Mamani

Secretario : Mgr. Gianfranco Alexey Málaga Tejada

Vocal : Msc. Edgar Aurelio Taya Acosta

Asesor(a) : Dra. Karin Yanet Supo Gavancho

Para calificar la sustentación de la Tesis en acto público el día 10 de diciembre del 2015, presentado por el bachiller Ayrtonn Frankho Meléndez Quispe, de la Escuela Profesional de Ingeniería en Informática y Sistemas.

El jurado calificador en forma secreta e individual, emitió su opinión sobre el tema de la tesis expuesta y procedió a obtener el promedio que arrojó el calificativo de aprobado con nota de dieciséis (16).


Para ratificar lo dictado firman:



Mgr. Erbert Francisco Osco Mamani
Presidente



Ing. Gianfranco Alexey Málaga Tejada
Secretario



MSc. Edgar Aurelio Taya Acosta
Vocal



Dra. Karin Yanet Supo Gavancho
Asesor(a)

UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN

FACULTAD DE INGENIERÍA

ESCUELA PROFESIONAL DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS

**“RENDIMIENTO DE LA BASE DE DATOS DEL SISTEMA WEB DE REGISTRO
DE POSTULANTES UTILIZANDO EL MODELO RELACIONAL Y NO
RELACIONAL EN LA OFICINA DE ADMISION DE LA UNIVERSIDAD
NACIONAL JORGE BASADRE GROHMANN, TACNA – 2015”**

TESIS SUSTENTADA Y APROBADA EL 10 DE DICIEMBRE DEL 2015
ESTANDO EL JURADO CALIFICADOR INTEGRADO POR:

Presidente


Mgr. Erben Francisco Osco Mamani


Secretario


Ing. Gianfranco Alexey Málaga Tejada

Vocal


MSc. Edga Aurelio Taya Acosta

Asesor(a)


Dra. Karin Yanet Supo Gavancho
Asesor(a)

AGRADECIMIENTOS

Agradezco a la Universidad Nacional Jorge Basadre Grohmann, la cual me abrió las puertas para poder estudiar mi carrera, así como también a los diferentes docentes que me brindaron sus conocimientos y su apoyo para poder llegar a este momento.

El presente trabajo está dedicado a mis padres, que sin ellos y su apoyo incondicional no habría llegado hasta aquí.

Y a todos aquellos que de una u otra manera me extendieron la mano, muchas gracias.

DEDICATORIA

A Dios.

Por darme la vida, salud y las fuerzas necesarias para no desmayar y seguir adelante frente a los problemas que se me presentaron, permitiéndome llegar a este momento tan importante de mi formación académica.

A mis Padres.

Por todo su amor, apoyo y comprensión incondicional estuvieron siempre a mi lado, ellos que siempre me dieron una palabra de aliento en los momentos más difíciles y que han son el incentivo de mi vida.

CONTENIDO

DEDICATORIA	v
RESUMEN	xiii
ABSTRACT	xiv
INTRODUCCIÓN	1
CAPÍTULO I PLANTEAMIENTO DE LA INVESTIGACIÓN	4
1.1.Descripción del problema	4
1.1.1. Antecedentes del problema	4
1.1.2. Problemática de la investigación.....	6
1.2.Formulación del problema	7
1.3.Justificación.....	7
1.4.Alcances y limitaciones	8
1.1.1. Alcances	8
1.1.2. Limitaciones	9
1.5.Objetivos	9
1.1.1. Objetivo General	9
1.1.2. Objetivos Específicos	9
1.6.Hipótesis.....	10
1.1.1. Hipótesis global.....	10
1.1.2. Subhipótesis.....	10

1.7. Variables	11
1.1.1. Identificación de variables	11
1.1.2. Definición de las variables.....	11
1.1.3. Operacionalización de variables	11
1.1.4. Clasificación de las variables	12
1.8. Diseño de la investigación.....	12
1.1.1. Diseño no experimental	12
1.1.2. Población y muestra.....	13
1.1.3. Técnicas e instrumentos para recolección de datos	13
1.1.4. Análisis de datos.....	13
1.1.5. Selección de pruebas estadísticas.....	14
CAPÍTULO II MARCO TEÓRICO.....	16
2.1. Marco referencial.....	16
2.1.1. Historia del Modelo relacional	16
2.1.2. NoSQL	17
2.1.3. JMeter	19
2.1.4. MongoDB	19
2.1.5. JSON – JavaScript Object Notation	20
2.1.6. ISO 9126.....	21

2.2. Bases Teóricas respecto al problema	21
2.2.1. Base de datos relacional.....	21
2.2.2. Base de datos no relacional.....	23
2.2.3. Tipo de bases de datos no relacionales.....	23
2.2.4. Atomicidad de una Base de Datos Relacional	27
2.2.5. Características NoSQL	28
2.2. Rendimiento de bases de datos	29
2.2.1. Prueba de Rendimiento	29
2.2.2. Para que sirve una prueba de rendimiento	29
CAPÍTULO III DESARROLLO.....	31
3.1 Análisis comparativo entre el modelo de datos relacional y no relacional	31
3.1.1. Estructura.....	31
3.1.2. Restricciones.....	34
3.1.3. Operaciones.....	35
3.2 Procedimiento de migración de base de datos.....	36
3.2.1. Migración de base de datos	37
3.2.1. Obtención de datos	46
4.1. Plan de pruebas con Jmeter.....	48
4.2. Plan de prueba de rendimiento para SQLSever.....	49
4.3. Procedimiento de evaluación de rendimiento con Jmeter	52

4.4. Plan de prueba de rendimiento MongoDB	56
CAPÍTULO IV RESULTADOS.....	60
5.1. Resultados SQLServer usando modelo relacional	60
5.2. Resultados MongoDB usando modelo no relacional	61
5.3. SQLServer vs MongoDB	62
5.3.1. Tiempo de respuesta de consulta	62
5.3.1. Consumo de KB/s	62
5.3.2. Consumo de almacenamiento en disco duro	64
5.4. Contratación de hipótesis	65
5.4.1. Hipótesis General.....	65
5.4.2. Prueba T de Student para dos muestras Independientes utilizando el software SPSS.....	67
CAPÍTULO V DISCUSIONES.....	73
CAPÍTULO VI CONCLUSIONES Y RECOMENDACIONES.....	75
6.1. Conclusiones	75
6.2. Recomendaciones.....	76
CAPÍTULO VII REFERENCIAS BIBLIOGRÁFICAS.....	77
ANEXOS.....	82
ANEXO 01 MATRIZ DE CONSISTENCIA	83
ANEXO 02 SISTEMA WEB DE REGISTRO DE POSTULANTE	86
ANEXO 03 TIEMPOS DE CONSULTA.....	89

ÍNDICE DE FIGURAS

Figura 1 <i>Ranking de base de datos</i>	20
Figura 2 <i>Equivalencia entre tabla y colección</i>	32
Figura 3 <i>Equivalencia fila y documento</i>	33
Figura 4 <i>Equivalencia entre columna y campo</i>	33
Figura 5 <i>Equivalencia clave primaria</i>	34
Figura 6 <i>Modelo Relacional</i>	37
Figura 7 <i>Estructura de un documento</i>	38
Figura 8 <i>Migración del modelo relacional al no relacional</i>	39
Figura 9 <i>Modelo entidad-relación del sistema de registro de postulantes</i>	41
Figura 12 <i>Consultas SQL para copiar datos entre tablas</i>	43
Figura 18 <i>Creación de Grupos de Hilos usando Jmeter</i>	52
Figura 19 <i>Configuración grupos de hilos – SQLServer</i>	53
Figura 20 <i>Configuración de la Conexión JDBC en Jmeter</i>	54
Figura 21 <i>Creación de petición – SQLServer</i>	55
Figura 22 <i>Configuración grupo de hilos – MongoDB</i>	57
Figura 23 <i>Configuración de la conexión – MongoDB</i>	58
Figura 24 <i>Consulta NoSQL en Jmeter</i>	59
Figura 25 <i>Tiempo total de consulta-SQLServer</i>	61
Figura 26 <i>Tiempo total de consulta-MondoDB</i>	61

Figura 27 <i>SQLServer vs Mongo en relación del tiempo por usuarios</i>	62
Figura 28 <i>Consumo de KB/Sec para SQLServer y MongoDB</i>	63
Figura 29 <i>Megabytes ocupados por SQLServer</i>	64
Figura 30. <i>Megabytes ocupados por MongoDB</i>	64
Figura 31 <i>Uso de disco duro SQLServer y MongoDB</i>	65
Figura 32 <i>Ingreso de datos al software SPSS</i>	68
Figura 33 <i>Resultado de prueba de normalidad</i>	69
Figura 34 <i>Prueba de Levene</i>	70
Figura 35 <i>Resultado de la prueba T de Student</i>	71

ÍNDICE DE TABLAS

Tabla 1 <i>Descripción de la variable</i>	11
Tabla 2 <i>Comparación de estructura del modelo relación y no relacional</i>	32
Tabla 3 <i>Comparación de restricciones entre el modelo relacional y no relacional</i>	34
Tabla 4 <i>Restricciones semánticas del modelo relacional</i>	35
Tabla 5 <i>Comparación de operación de consulta a las bases de datos relacional y no relacional</i>	36
Tabla 6 <i>Plan de prueba para evaluar el rendimiento</i>	50
Tabla 7 <i>Resumen de resultados de la prueba de rendimiento al modelo relacional</i>	60
Tabla 8 <i>Resumen de resultados de la prueba de rendimiento al modelo no relacional</i>	61
Tabla 9 <i>Consumo de KB/sec de la base de datos relacional y no relacional</i>	63
Tabla 10 <i>Comparación de almacenamiento SQLServer y MongoDB</i>	64
Tabla 11 <i>Resultado de la prueba de normalidad</i>	69
Tabla 12 <i>Resumen de prueba de Levene</i>	70
Tabla 13 <i>Decisión estadística</i>	71

RESUMEN

El presente trabajo de investigación, tiene como principal objetivo comparar el rendimiento de la base de datos del Sistema Web de Registro de Postulantes utilizando el modelo relacional y no relacional en el proceso de consulta de datos en la Oficina de Admisión de la Universidad Nacional Jorge Basadre Grohmann, para esto se analizará primero la base de datos relacional, luego se estudiará las bases de datos que utilizan el modelo no relacional. Una vez comprendido ambos modelos se procedió a migrar la base de datos relacional hacia el modelo no relacional.

El diseño de la investigación es no experimental, descriptivo-comparativo, ya que se pretende examinar la diferencia en el rendimiento de una base de datos utilizando el modelo relacional y no relacional. Para la obtención de datos se utilizó el software Jmeter que permitió determinar el rendimiento de ambos modelos. Los resultados obtenidos fueron procesados con el software SPSS, demostrando que una base de datos no relacional tiene mejor rendimiento que una base de datos relacional.

Palabras claves: Modelo relacional y no relacional, Base de datos, SqlServer, MongoDB.

ABSTRACT

The main objective of this research work is to compare the performance of the database of the Applicant Registration Web System using the relational and non-relational model in the data query process in the Admission Office of the Jorge Basadre National University Grohmann, for this the relational database will be analyzed first, then the databases that use the non-relational model will be studied. Once both models were understood, the relational database was migrated to the non-relational model.

The research design is non-experimental, descriptive-comparative, since it is intended to examine the difference in the performance of a database using the relational and non-relational model. To obtain data, the Jmeter software was used, which made it possible to determine the performance of both models. The results obtained were processed with the SPSS software, showing that a non-relational database has better performance than a relational database.

Keywords: Relational and non-relational model, Database, SqlServer, MongoDB.

INTRODUCCIÓN

En la actualidad existe una gran polémica alrededor del tema relacionado con la gestión de la información digital que se necesita almacenar para su posterior recuperación y análisis por parte de los diseñadores, arquitectos y desarrolladores de aplicaciones informáticas de cualquier tipo. Por un lado están los tradicionales sistemas de gestión de base de datos relacionales (RDBMS, por sus siglas en inglés) y por otro los prometedores sistemas de bases de datos no relacionales conocidos como NoSQL (*Notonly SQL*).

Los primeros, dueños de la mayor parte del mercado del almacenamiento de datos, con una robustez innegable y años de explotación en múltiples entornos de gestión de información.

Los segundos, emergentes y novedosos, ofrecen sin embargo una nueva forma de pensar en el desarrollo de aplicaciones web orientadas y centradas en el usuario. Se hace necesario entonces hacer unos análisis conscientes de ambas variantes, con el objetivo de escoger la solución óptima al momento de escoger un sistema gestor de base de datos

Las bases de datos NoSQL surgieron como consecuencia de los rendimientos de los motores basados en SQL, ya que no cumplían las expectativas de tiempo de respuesta.

Esta siguiente generación de sistemas de gestión de bases de datos contiene unas características particulares, ya que es no relacional-NoSQL, distribuida, de código abierto y horizontalmente escalable.

A continuación, se detalla la estructura de la tesis:

En el capítulo 1 abarca el planteamiento de la investigación, los cuales incluyen: la descripción del problema, formulación del problema, justificación, alcances y limitaciones, objetivos, hipótesis y variables.

En el capítulo 2 comprende el marco teórico que es el fundamento de la investigación, en donde se realiza una recopilación bibliográfica, en la cual se detallan las bases teóricas para el desarrollo esta investigación.

En el capítulo 3 se realiza el desarrollo, a través de un análisis comparativo entre el modelo relacional y no relacional, y finalmente la obtención de datos.

En el capítulo 4 se muestra los resultados obtenidos, para luego someterlos a un análisis estadístico descriptivo e inferencial para la contrastación de hipótesis

En el capítulo 5 se presentan las discusiones, aquí se comparan los datos obtenidos con los de otros investigadores que han trabajado el mismo tema.

En el capítulo 6 están las conclusiones y recomendaciones, obtenidas después de haber realizado la investigación.

En el capítulo 7 las referencias bibliográficas, sobre la cual se basa la presente investigación.

CAPÍTULO I

PLANTEAMIENTO DE LA INVESTIGACIÓN

1.1. Descripción del problema

1.1.1. Antecedentes del problema

Por más de veinte años el escenario de almacenamiento de la información se ha dominado por las bases de datos relacionales. Un novedoso concepto de almacenamiento de datos gana espacio, las llamadas bases de datos NoSQL, especialmente debido a la escalabilidad y velocidad de sus tiempos de respuestas, superiores a las de los sistemas relacionales. (Vasquez y Sotolongo, 2013).

Las bases de datos NoSQL han experimentado un importante incremento en su aplicación en los últimos tiempos. La gran flexibilidad que ofrecen y las posibilidades que brindan desde el punto de vista de la optimización en sus diseños de acuerdo al problema a resolver las convierten en una atractiva variante a tener en cuenta para los desarrolladores de aplicaciones de gestión de información (Gracia y Yanes, 2012).

El desarrollo y crecimiento masivo de las redes de computadoras y medios de almacenamiento, a lo largo de los últimos años, ha motivado la

aparición de un creciente interés por los sistemas de clasificación automática de documentos. Las bases de datos tradicionales tienen limitaciones a la hora de indexar y realizar búsquedas en tiempo real sobre grandes volúmenes de datos. (Ramírez, Cuevas y Martínez, 2011).

Desde el momento en que los humanos han requerido almacenar datos que lleven a la obtención de información oportuna y fiable también se ha buscado la manera más eficiente de manipular dichos datos. Con el tiempo la creciente cantidad de información que se intenta manejar supera por mucho la infraestructura existente, esto es, se ha llegado a la necesidad de buscar soluciones que permitan almacenar cada vez más información incluso de la que se podría imaginar (Manjarrez, Martínez y Cuevas, 2014).

La necesidad de almacenar y procesar grandes volúmenes de datos ha dado origen al término Big Data. Estos sistemas manejan información obtenida desde diversas fuentes y formatos, como es el caso de páginas web, redes sociales, el análisis del genoma humano, la física de partículas, entre otros. Estos almacenes de datos presentan dificultades que no pueden ser resueltas mediante el uso de sistemas de gestión de bases de datos tradicionales (Jaramillo, 2014).

1.1.2. Problemática de la investigación

El rápido crecimiento de la información, presentan hoy nuevos desafíos que los sistemas web deben satisfacer, a nivel de base de datos, exigen, tales como soportar data a gran escala, soportar estresantes frecuencias de lectura y escritura, soportar cambios de esquema frecuentes.

Para la presente investigación se ha tomado como caso de estudio la base del Sistema Web de Registro de Postulantes de la Oficina de Admisión de la Universidad Nacional Jorge Basadre Grohmann que tiene como finalidad promover las carreras profesionales, además de planificar, organizar, dirigir y controlar el proceso de admisión de los postulantes a la universidad.

Actualmente el Sistema de Registro de Postulantes posee una base de datos con el modelo relacional que utiliza como Sistema Gestor de bases de datos a SQL Server 2008.

Es por ello que se propone, a pesar de no ser muy difundidas, el uso de una base de datos orientada al modelo no relacional o NoSQL, ya que

estas tienen como principales características, la flexibilidad de esquema, escalabilidad horizontal y tiempo de respuesta óptimo.

1.2. Formulación del problema

¿Cuál es el rendimiento de la base de datos del Sistema Web de Registro de Postulantes utilizando el modelo relacional y no relacional en la Oficina de Admisión de la Universidad Nacional Jorge Basadre Grohmann, Tacna-2015?

1.3. Justificación

NoSQL hace referencia al conjunto de tecnologías en base de datos que buscan alternativas al sistema de base de datos relacional, en un contexto donde priman la velocidad, el manejo de grandes volúmenes de datos y la posibilidad de tener un sistema distribuido.

Las tres grandes causas del “bullicio” que está causando NoSQL son: los cambios en el tamaño y la cantidad (e incluso estructura) de los datos, la necesidad de velocidad por parte de los usuarios y la falta de innovación en el área.

Las características que plantean las bases de datos NoSQL responden a las necesidades actuales de las diferentes organizaciones, por lo que se convierten en una alternativa a las tradicionales bases de datos, debido a su capacidad y a la velocidad, que son dos principios que priman hoy en el mundo de los sistemas (Romero y Gonzáles, 2007).

Al movimiento NoSQL, le queda mucho camino por delante; las tecnologías tienen que madurar, y para ello se hace necesaria la participación de la comunidad académica; por esto el presente trabajo es un punto de partida para conocer del tema y comenzar a vincularse y tener en cuenta este movimiento.

1.4. Alcances y limitaciones

1.1.1. Alcances

El trabajo de investigación tiene como alcance:

- Realizar un diagnóstico de la base de datos relacional actual.
- Realizar la migración de la base de datos utilizando el modelo no relacional.
- Realizar la comparación del rendimiento entre las bases de datos utilizando el modelo relacional y no relacional del sistema.

1.1.2. Limitaciones

- El trabajo de investigación se limitará solo al uso de MongoDB como base de datos no relacional y SQL Server para el modelo relacional.
- Las pruebas de rendimiento se realizarán en un mismo computador.
- Para evaluar el rendimiento de ambas bases de datos se usará la herramienta Jmeter.

1.5. Objetivos

1.1.1. Objetivo General

- O:** Determinar el rendimiento de la base de datos del Sistema Web de Registro de Postulantes utilizando el modelo relacional y no relacional en la Oficina de Admisión de la Universidad Nacional Jorge Basadre Grohmann, Tacna 2015.

1.1.2. Objetivos Específicos

- O₁:** Determinar el uso de memoria de una base de datos relacional y no relacional.
- O₂:** Determinar el uso de disco duro de una base de datos relacional y no relacional.

1.6. Hipótesis

1.1.1. Hipótesis global

H₀: El rendimiento de la base de datos utilizando el modelo relacional no es mejor que el modelo no relacional, en el Sistema Web de Registro de Postulantes de la Universidad Nacional Jorge Basadre Grohmann, Tacna 2015.

H₁: El rendimiento de la base de datos utilizando el modelo relacional es mejor que el modelo no relacional, en el Sistema Web de Registro de Postulantes de la Universidad Nacional Jorge Basadre Grohmann, Tacna 2015.

1.1.2. Subhipótesis

a) El uso de memoria respecto a las consultas realizadas utilizando una base de datos no relacional es menor que usando una base de datos relacional.

b) El uso de disco duro respecto al almacenamiento de una base de datos relacional es menor que una base de datos usando el modelo relacional.

1.7. Variables

1.1.1. Identificación de variables

Es univariable: Rendimiento.

1.1.2. Definición de las variables

a) Rendimiento

Se refiere a la medida o cuantificación de la velocidad con que se realiza una tarea o proceso.

1.1.3. Operacionalización de variables

Tabla 1

Descripción de la variable

Variable	Indicador	Definición
Rendimiento	Tiempo de respuesta	Lapso de tiempo que transcurre entre que un usuario hace una petición a la base de datos.
	Uso de disco duro	Cantidad de megabytes ocupados por la base de datos.
	Uso de memoria ram.	Cantidad de kilobyte de memoria ram utilizados por consulta a la base de datos.

Fuente: Elaboración propia

1.1.4. Clasificación de las variables

A continuación, se muestra la clasificación de las variables por la función que cumple en la hipótesis, por su naturaleza y escala de medición:

➤ Variable: Rendimiento

Por su función : independiente

Por su naturaleza : Cuantitativa

Escala de medición: intervalo

1.8. Diseño de la investigación

1.1.1. Diseño no experimental

La investigación es del tipo Descriptivo-Comparativo.

Según Sánchez y Reyes (1996), la investigación descriptiva comparativa consiste en recolectar en dos o más muestras con el propósito de observar el comportamiento de una variable, tratando de controlar estadísticamente otras variables que se considera pueden afectar la variable estudiada.

1.1.2. Población y muestra

Para esta investigación se tomará como:

- Población: la cantidad de registros de postulantes de la base de datos del Sistema Web de Registro de Postulantes.

La cual está compuesta de 10 445 registros de postulantes.

- Muestra: Se escogió un total de 2000 registros para realizar las distintas pruebas de esfuerzo a la base de datos relacional y no relacional.

1.1.3. Técnicas e instrumentos para recolección de datos

Los datos se han obtenido a través del software Jmeter, el cual simuló varias peticiones a la base de datos con el modelo relacional y no relacional.

Se utilizará como instrumentos las Tablas Comparativas con los datos obtenidos.

1.1.4. Análisis de datos

Los datos recopilados fueron procesados y analizados con el software estadístico SPSS (*Statiscal Package for Social Sciences*) en su versión 23.0. Para la elaboración de tablas y/o gráficos estadísticos se usó Microsoft Excel 2010.

1.1.5. Selección de pruebas estadísticas

Para comprobar la hipótesis se aplicó la prueba estadística T de *Student* para dos Muestra Independientes.

Rubio y Berlanga (2012) dicen que las pruebas paramétricas exigen ciertos requisitos previos para su aplicación: la distribución Normal de la variable cuantitativa en los grupos que se comparan, la homogeneidad de varianzas en las poblaciones de las que proceden los grupos y una n muestral no inferior a 30. Su incumplimiento conlleva la necesidad de recurrir a pruebas estadísticas no paramétricas. Las pruebas paramétricas se clasifican en dos: prueba t (para una muestra o para dos muestras relacionadas o independientes) y prueba de ANOVA (para más de dos muestras independientes).

Prueba T para dos muestras independientes, esta opción debe utilizarse cuando la comparación se realice entre las medias de dos poblaciones independientes (los individuos de una de las poblaciones son distintos a los individuos de la otra) como, por ejemplo, en el caso de la comparación de las poblaciones de hombres y mujeres. Por lo tanto, compara las medias de una variable para dos grupos de casos.

La matriz de datos debe tener una columna para los datos de la variable de interés y una segunda columna con los códigos que definen las poblaciones objeto de comparación. La prueba calcula estadísticos descriptivos para cada grupo además de la prueba de Levene para igualdad de varianzas, así como los valores de t para varianzas iguales y desiguales y el intervalo de confianza del 95% para la diferencia de medias.

CAPÍTULO II

MARCO TEÓRICO

2.1. Marco referencial

2.1.1. Historia del Modelo relacional

Cuando en el año 1970 el Dr. E. F. Codd propone un nuevo modelo de datos, los SGBD imperantes en el mercado, de tipo Codasyl y Jerárquico, no habrían logrado superar el grave inconveniente que suponía la dependencia de las aplicaciones desarrolladas en ellos respecto a las estructuras de los datos.

A diferencia de estos modelos de datos basados en punteros físicos por los que tenía que navegar el programador a fin de recuperar y actualizar los datos, el Modelo Relacional (MR) se propone, como principal objetivo aislar al usuario de las estructuras físicas de los datos, consiguiendo así la independencia de las aplicaciones respecto de los datos, finalidad perseguida desde los inicios de las bases de datos.

En el nuevo modelo, basado en la teoría matemática de las relaciones, los datos se estructuran lógicamente en forma de relaciones – tablas. Esta formalización matemática convirtió rápidamente al modelo en una fuente fundamental de la investigación en bases de datos.

Los avances más importantes que el modelo de datos relacional incorpora respecto a los modelos anteriores son:

- Sencillez y uniformidad.
- Solida fundamenta teórica.
- Independencia de la interfaz de usuario.

A pesar de que, desde su introducción, el modelo relacional se convirtió en uno de los principales temas de investigación en bases de datos, los primeros sistemas relacionales tardaron en aparecer, sobre todo por dificultades de implementación, y en general los sistemas de gestión de bases de datos existentes en el mercado no suelen recoger, incluso en la actualidad, algunas de las características y propiedades del modelo relacional teórico propuesto por Codd.

2.1.2. NoSQL

Según Graterol (2014) NoSQL o “No solamente SQL” (*Not Only SQL*) es un término acuñado por Carlo Strozzi en 1988 y nuevamente retomado por Eric Evans en 2009 y se refiere a un conjunto de bases de datos que se diferencian en gran parte de las bases de datos convencionales, en características tanto de uso como implementación; estos tipo de base de datos no usan SQL o al menos no como lenguaje predeterminado para realizar las consultas.

Las bases de datos NoSQL, no soportan totalmente ACID, esto lo explica el Teorema del profesor Eric Brewer, Teorema CAP, donde menciona:

Es imposible para un sistema distribuido garantizar simultáneamente las siguientes tres características:

- *Consistency* (Consistencia): todos los nodos ven la misma data al mismo tiempo.
- *Availability* (Disponibilidad): una garantía de que todos los requerimientos recibirán una respuesta de que el requerimiento fue exitoso o fallido.
- *Partition Tolerance* (Tolerancia a la Partición): el sistema continua operando a pesar de la perdida arbitraria de mensajes, o la falla de parte del sistema.

SQL no es un lenguaje predominante entre los distintos tipo de bases de datos no relacionales, por lo general cada motor tiene su propio lenguaje de consultas. Cabe destacar que la información no se almacena con un esquema que el DBA o el desarrollador propone con anterioridad de manera virtual, es decir, no se crea en el motor antes de utilizar la base de datos sino al almacenar el primer valor.

2.1.3. JMeter

Jmeter es una herramienta open source muy completa, implementada en Java que permite realizar test de comportamiento funcional y medir el rendimiento. También se puede utilizar para realizar pruebas de estrés, por ejemplo, en un servidor, y poner a prueba su rendimiento (Diaz, Tzancoff, Rodríguez y Soria, 1900).

2.1.4. MongoDB

MongoDB es un sistema de bases de datos no relacionales, multiplataforma e inspirada en el tipo de bases de datos documental y clave/valor, su nombre proviene del término en inglés *humongous*. Esta liberada bajo licencia de software libre, específicamente GNU AGPL 3.0. MongoDB usa el formato BSON (JSON Compilado) para guardar la información, dando la libertad de manejar un esquema libre. Este motor de bases de datos es uno de los más conocidos y usados, pudiéndolo comparar en popularidad con MySQL en el caso de las bases de datos relacionales (Graterol, 2014).

Figura 1
Ranking de base de datos

289 systems in ranking, December 2015

Rank			DBMS	Database Model	Score		
Dec 2015	Nov 2015	Dec 2014			Dec 2015	Nov 2015	Dec 2014
1.	1.	1.	Oracle	Relational DBMS	1497.55	+16.61	+37.76
2.	2.	2.	MySQL	Relational DBMS	1298.54	+11.70	+29.96
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1123.16	+0.83	-76.89
4.	4.	↑5.	MongoDB +	Document store	301.39	-3.22	+54.87
5.	5.	↓4.	PostgreSQL	Relational DBMS	280.09	-5.60	+26.09
6.	6.	6.	DB2	Relational DBMS	196.13	-6.40	-14.13
7.	7.	7.	Microsoft Access	Relational DBMS	140.21	-0.75	+0.31
8.	8.	↑9.	Cassandra +	Wide column store	130.84	-2.08	+36.78
9.	9.	↓8.	SQLite	Relational DBMS	100.85	-2.60	+6.15
10.	10.	10.	Redis +	Key-value store	100.54	-1.87	+12.66

Fuente: <http://db-engines.com/en/ranking>

2.1.5. JSON – JavaScript Object Notation

JSON es un formato independiente del lenguaje, aunque su uso extendido hasta hace poco era en el lenguaje Javascript. Actualmente se usa JSON en grandes cantidades de sistemas para intercambiar información por su simplicidad en comparación con XML.

Este formato soporta gran cantidad de tipos de datos, lo que lo hace atractivo para un uso generalizado, y cada vez más lenguajes de programación dan soporte a este formato (Graterol, 2014).

2.1.6. ISO 9126

Echeverría y Abella (2014), mencionan que la norma ISO 9126 describe un conjunto de características para la calidad interna y externa, que son además divididas en sub-características que se manifiestan externamente cuando el software se usa como una parte del sistema computarizado, y son un resultado de los atributos internos del software.

La presente investigación está asociada a la característica de eficiencia (rendimiento), definida como: la capacidad del producto de software para proporcionar una ejecución o desempeño apropiado, en relación con la cantidad de recursos utilizados usados, bajo condiciones establecidas.

2.2. Bases Teóricas respecto al problema

2.2.1. Base de datos relacional

Es una colección de datos relacionados. Por datos entendemos hechos conocidos que pueden registrarse y que tienen un significado implícito.

Una base de datos tiene las siguientes propiedades implícitas:

- Una base de datos representa algún aspecto del mundo real, en ocasiones llamado minimundo o universo de discurso. Los cambios introducidos en el minimundo se reflejan en la base de datos.
- Una base de datos es un conjunto de datos lógicamente coherente, con cierto significado inherente. Una colección aleatoria de datos no puede considerarse propiamente una base de datos.
- Toda base de datos se diseña, construye y puebla con datos para un propósito específico. Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a dichos usuarios.

Elmasri y Navathe (2007), mencionan que una base de datos tiene una fuente de la cual se derivan los datos, cierto grado de interacción con los acontecimientos del mundo real y un público que esta activamente interesado en el contenido de la base de datos.

Las bases de datos pueden ser de cualquier tamaño y tener diversos grados de complejidad.

2.2.2. Base de datos no relacional

Las bases de datos NoSQL son sistemas de almacenamiento de información que no cumplen con el esquema entidad-relación. Mientras que las tradicionales bases de datos relacionales basan su funcionamiento en talas, joins y transacciones ACID. Las bases de datos NoSQL no imponen una estructura de datos en forma de tablas y relaciones entre ellas en ese sentido son más flexibles, ya que suelen permitir almacenar información en otros formatos como clave-valor similar a tablas Hash, Mapeo de Columnas, Documentos o Grafos. (Martín, Chávez, Rodríguez, Valenzuela y Murazzo, 2013).

2.2.3. Tipo de bases de datos no relacionales

Córdova y Cuzco (2013) clasifican las bases de datos no relacionales de la forma siguiente:

a) Clave-Valor

Cada uno de los elementos se identifica por una llave por medio del cual podemos recuperar la información de forma rápida, este valor por lo general es almacenado como un objeto BLOB (nombre que se da a datos que son objetos binarios de gran tamaño ej. Imágenes, música,

etc.), ya que para la base de datos lo importante no es su contenido sino la clave y el valor.

Una característica importante es que se puede escalar fácilmente haciendo particiones de los valores de acuerdo a su clave, si una clave está entre 1 y 1000 va a un servidor, los de 1001 y 2000 a otro y así sucesivamente. Dentro de estas bases de datos podemos encontrar:

- ✓ BigTable de Google
- ✓ SimpleDB de Amazon
- ✓ Cassandra
- ✓ Riak
- ✓ Voldemort
- ✓ MemcacheDB

b) Orientadas a Columnas

Esta base de datos guarda cada uno de sus valores en columnas y no en filas, con este cambio se puede ganar más velocidad al momento de realizar las lecturas y mucho más si la consulta es un número reducido de columnas, pero no es eficiente para realizar escrituras es por esto que este tipo de soluciones se usa en aplicaciones con un número bajo de escrituras.

c) Basada en documentos

Estas bases de datos almacenan su información como un documento con una estructura simple como JSON o XML con una llave única, siendo similar a las bases de datos Clave-Valor con la diferencia que su valor es un archivo, cuando el servidor entiende cada uno de los datos este puede realizar las operaciones sobre ellos.

Las implementaciones con esta base de datos permiten que las consultas sean muy avanzadas, estableciendo las relaciones entre ellos, la única desventaja es que no permiten realizar JOINS (combinar registros de dos o más tablas), encontramos entre las más importantes a MongoDB y CouchDB.

Este tipo de bases de datos son simplemente un almacén clave-valor con la excepción de que el valor no se guarda solo como un campo binario, sino con un formato definido de forma tal que el servidor pueda entenderlo. Esto no quiere decir que tengan un esquema, siguen siendo schema-less, seguimos teniendo solo 2 campos y uno de ellos binario.

La diferencia es que el campo binario puede ser entendido por la base de datos. Dicho formato a menudo es JSON, pero puede ser XML o cualquier otra cosa. Si el servidor entiende los datos, puede hacer operaciones con ellos. De hecho varias de las implementaciones de este tipo de bases de datos permiten consultas muy avanzadas sobre los datos, e incluso establecer relaciones entre ellos, aunque siguen sin permitir joins por cuestiones de rendimiento. Todo esto sin perder de vista que sigue siendo un clave-valor con todas las ventajas que nos dan estas bases de datos.

Por ello este tipo de bases de datos son la gran medida los responsables del auge actual de NoSQL. Ofrecen una gran performance y escalabilidad sin perder todo el beneficio del modelo relacional. Entre las bases de datos de este tipo, están los más famosos ejemplos de la familia NoSQL: MongoDB y CouchDB.

d) Orientadas a grafos

La información que almacena esta base de datos lo hace en forma de grafo, dando la importancia no solo a los datos sino también a cada una de las relaciones que existen, estas relaciones pueden tener

atributos y realizar consultas directas a las mismas y al estar almacenadas de esta manera es más eficiente y fácil de navegar.

Este tipo de base de datos son aprovechadas si la información es representada fácilmente como en las redes sociales o sistemas que recomienden algún producto, la ventaja que tienen estas bases de datos es que se pueden aplicar algoritmos estadísticos para recomendaciones y tienen que recorrer cada uno de los grafos.

Entre las implementaciones más usadas esta Neo4J, Hyperbase-DB e InfoGrid.

2.2.4. Atomicidad de una Base de Datos Relacional

El término transacción hace referencia a un conjunto de operaciones que forman una única unidad lógica de trabajo. Por ejemplo, la transferencia de dinero de una cuenta a otra es una transacción que consta de dos actualizaciones, una para cada cuenta.

Resulta importante que, o bien se ejecuten completamente todas las acciones de una transacción, o bien, en caso de fallo, se deshagan los efectos parciales de la transacción. Esta propiedad se denomina atomicidad.

2.2.5. Características NoSQL

Castro, González y Callejas (2012), mencionan las propiedades comunes para un conjunto de tecnologías, se proponen seis características específicas para poder encasillar a las bases de datos NoSQL:

- Escalabilidad horizontal: refiriéndose a la facilidad añadir, eliminar o realizar operaciones con elementos (hardware) del sistema, sin afectar el rendimiento.
- Habilidad de distribución: tiene que ver con la escalabilidad horizontal, pero haciendo énfasis en su soporte; para ello se tienen cuenta la habilidad de replicar y distribuir los datos sobre los servidores.
- Uso eficiente de recursos: aprovecha las nuevas tecnologías, como los discos en estado sólido, el uso eficiente de recursos como la RAM y los sistemas distribuidos en general.
- Libertad de esquema: al no tener un esquema rígido se permite mayor libertad para modelar los datos; además facilita la integración con los lenguajes de programación orientados a objetos, lo que evita el proceso de mapeado.
- Modelo concurrencias débil: no implementa ACID (*Atomicity, Consistency, Isolation and Durability*), que reúne las características necesarias para que una serie de instrucciones puedan ser

consideradas una transacción, sin embargo, si se tienen en cuenta algunas consideraciones para asegurar estos aspectos, pero no son tan estrictas.

- Consultas simples: las consultas requieren menos operaciones y son más naturales, por lo tanto, se gana en simplicidad y eficiencias.

2.2. Rendimiento de bases de datos

2.2.1. Prueba de Rendimiento

Las pruebas de rendimiento realizadas sobre computadoras, redes, software u otros dispositivos, son utilizados para determinar la velocidad y eficiencia de los mismos. Este procedimiento puede involucrar tanto test cuantitativos, por ejemplo, medir tiempos de respuesta o cantidad en millones de líneas de código, como test cualitativos, en los cuales se evalúa fiabilidad, escalabilidad e interoperabilidad. Estas pruebas de rendimiento pueden ser realizadas a través de herramientas que proveen pruebas de estrés, que permiten determinar la estabilidad del sistema (Diaz, Tzancoff, Rodríguez y Soria, 1900).

2.2.2. Para que sirve una prueba de rendimiento

Según Medina, J. (2014) una prueba de rendimiento, dependiendo del tipo de prueba, y del punto del ciclo de vida del servicio en los que se realicen pruebas puede tener diversas utilidades.

➤ Fase de desarrollo

Detección de defectos de rendimiento tempranos.

➤ Fase de preproducción

Predecir el funcionamiento de la aplicación en el entorno de producción en base a unas estimaciones de uso y escalado.

Detección de errores o defectos de rendimiento en el servicio (software o hardware).

Comparar el rendimiento del servicio en preproducción con el de otros servicios evaluados.

➤ Fase de producción

Evaluar si un servicio se adecua correctamente a las necesidades de calidad de servicio del usuario (principalmente en parámetros de tiempo de respuesta).

Evaluar la estabilidad y la degradación del servicio.

Evaluar como cambios en la infraestructura afecta al rendimiento del mismo.

Detección de defectos y mejoras de eficiencia: cuellos de botella, comportamiento ante niveles de carga.

CAPÍTULO III

DESARROLLO

3.1 Análisis comparativo entre el modelo de datos relacional y no relacional

Se realizó el análisis comparativo entre el modelo de datos relacional y no relacional, donde determinamos equivalencias y diferencias entre ambos modelos a partir de las propiedades estándares en los modelos de datos:

- Estructura
- Restricciones
- Operaciones

3.1.1. Estructura

Se hace referencia a las relaciones y/o elementos que deben mantenerse para dar forma a los datos.

A continuación, se muestra tales elementos por modelo.

Tabla 2

Comparación de estructura del modelo relación y no relacional

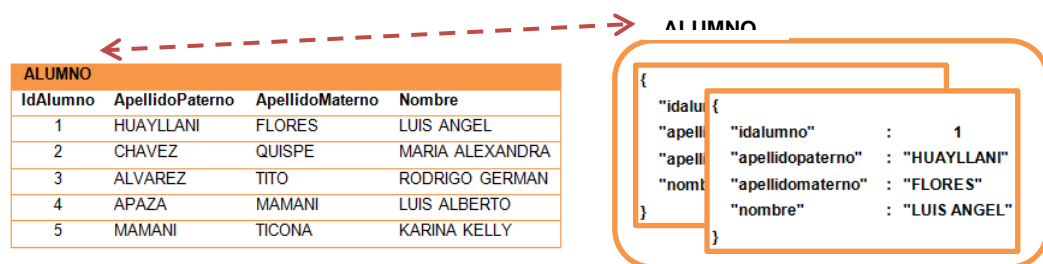
MODELO RELACIONAL	MODELO NO RELACIONAL
Tabla	Colección
Fila	Documento
Columna	Campo

Fuente: Elaboración propia

➤ Después de realizar una analogía, a nivel teórico, entre ambos modelos, se pudo ver la equivalencia entre tabla (modelo relacional) y colección (modelo no relacional), ya que por un lado la tabla esta denotada por una cardinalidad que varía acorde al conjunto de tuplas, y la colección, por el conjunto de documentos con esquemas no impuestos en cada uno de ellos.

Figura 2

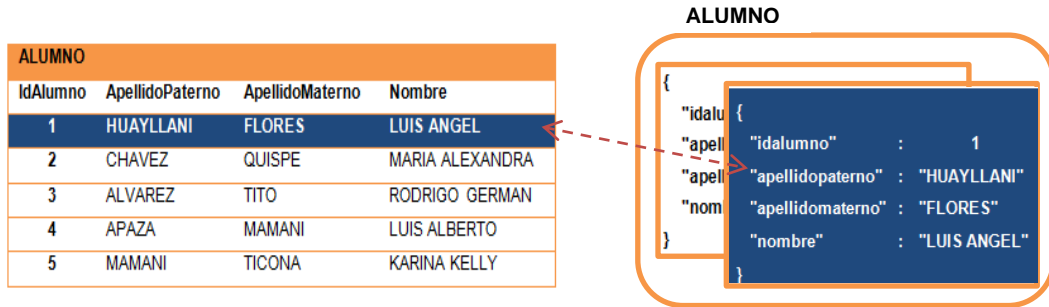
Equivalencia entre tabla y colección



Fuente: Elaboración Propia

➤ Segundo lugar, la equivalencia entre fila y documento

Figura 3
Equivalencia fila y documento



Fuente: Elaboración propia

- Tercer lugar, la equivalencia entre columna y campo.

Figura 4
Equivalencia entre columna y campo



Fuente: Elaboración propia

3.1.2. Restricciones

En cuanto a las reglas que permiten representar una base de datos, a nivel de los modelos relacional y no relacional, solo existe una equivalencia a nivel de restricción inherente.

Tabla 3

Comparación de restricciones entre el modelo relacional y no relacional

	MODELO RELACIONAL	MODELO NO RELACIONAL
INHERENTES	X	X
SEMÁNTICAS	X	

Fuente: Elaboración propia

Si bien el modelo de datos documental impone exista un código único (<ObjectId>) autogenerado para cada documento en las colecciones existentes de una base de datos no relacional, este sería el equivalente a una clave primaria en el modelo relacional.

Figura 5

Equivalencia clave primaria



Fuente: Elaboración propia

Por otro lado, de las diferencias, el modelo de datos documental no soporta restricciones semánticas o restricciones que deben expresarse directamente en los esquemas del modelo de datos.

Tabla 4
Restricciones semánticas del modelo relacional

RESTRICCION	TIPO
Semánticas	Clave Primaria (PRIMARY KEY)
	Unicidad (UNIQUE)
	Obligatoriedad (NOT NULL)
	Integridad Referencial (FOREIGN KEY)
	Verificación (CHECK)
	Asertos (ASSERTION)
	Disparadores (TRIGGER)

Fuente: Elaboración propia

3.1.3. Operaciones

En cuanto a las operaciones en ambos modelos de tos, existe equivalencia a nivel de recuperación y actualización, a pesar de ser diferentes los modelos de datos, se mantiene el CRUD (*Create, Read, Update, Delete*), ya sea que poseen diferente nominación a nivel de implementación, se cumple funcionalmente con el propósito de Consulta, Inserción, Borrado y Modificación.

Tabla 5

Comparación de operación de consulta a las bases de datos relacional y no relacional

		IMPLEMENTACION	IMPLEMENTACION
	OPERACIÓN	MODELO RELACIONAL	MODELO NO RELACIONAL
Recuperación	Consulta	Select	Find
	Inserción	Insert	Insert
Actualización	Borrado	Delete	Remove
	Modificación	Update	Update

Fuente: Elaboración propia

3.2 Procedimiento de migración de base de datos

A partir del análisis comparativo se procede a la migración del modelo de datos relacional hacia el no relacional.

Para esto proceso se hizo uso de una pequeña aplicación, para poder lograr la comunicación de un modelo de datos relacional que utiliza como SGBD SQLServer, hacia el no relacional, en este caso MongoDB.

3.2.1. Migración de base de datos

Gran parte de los datos que usamos hoy en día cuenta con estructuras complejas que pueden ser modeladas y representadas de manera más eficiente usando JSON (JavaScript Object Notation) documentos, en lugar de tablas.

Por ejemplos, los datos que pertenecen a una relación padre-hijo en dos tablas RDBMS comúnmente serian embebidos en un solo documentos en MongoDB.

Figura 6
Modelo Relacional



Fuente: <https://www.mongodb.com/collateral/rdbms-mongodb-migration-guide>

Modelando la misma data en MongoDB nos permite crear un esquema en el que integramos una serie de subdocumentos de cada vehículo directamente en el documento persona.

Figura 7

Estructura de un documento

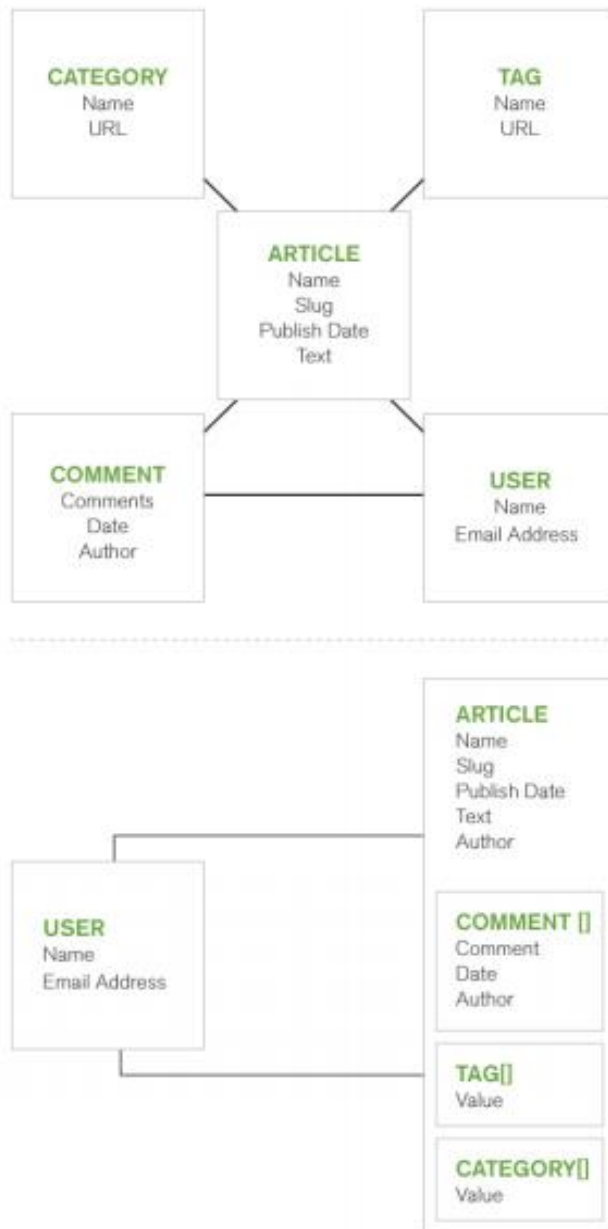
```
{
  first_name: "Paul",
  surname: "Miller",
  city: "London",
  location: [45.123,47.232],
  cars: [
    { model: "Bentley",
      year: 1973,
      value: 100000, ...},
    { model: "Rolls Royce",
      year: 1965,
      value: 330000, ...},
  ]
}
```

Fuente: <https://www.mongodb.com/collateral/rdbms-mongodb-migration-guide>

En este ejemplo, el modelo relacional consta solo de dos tablas.

A continuación, veremos un ejemplo con 5 tablas.

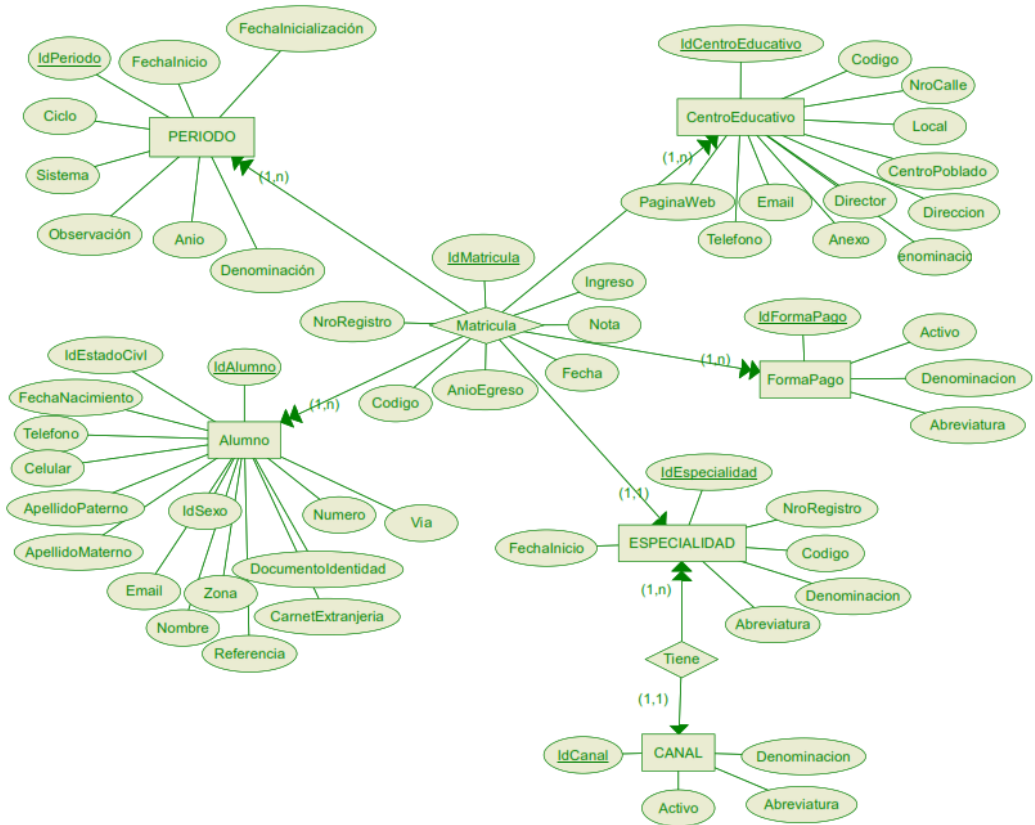
Figura 8
Migración del modelo relacional al no relacional



Fuente: <https://www.mongodb.com/collateral/rdbms-mongodb-migration-guide>

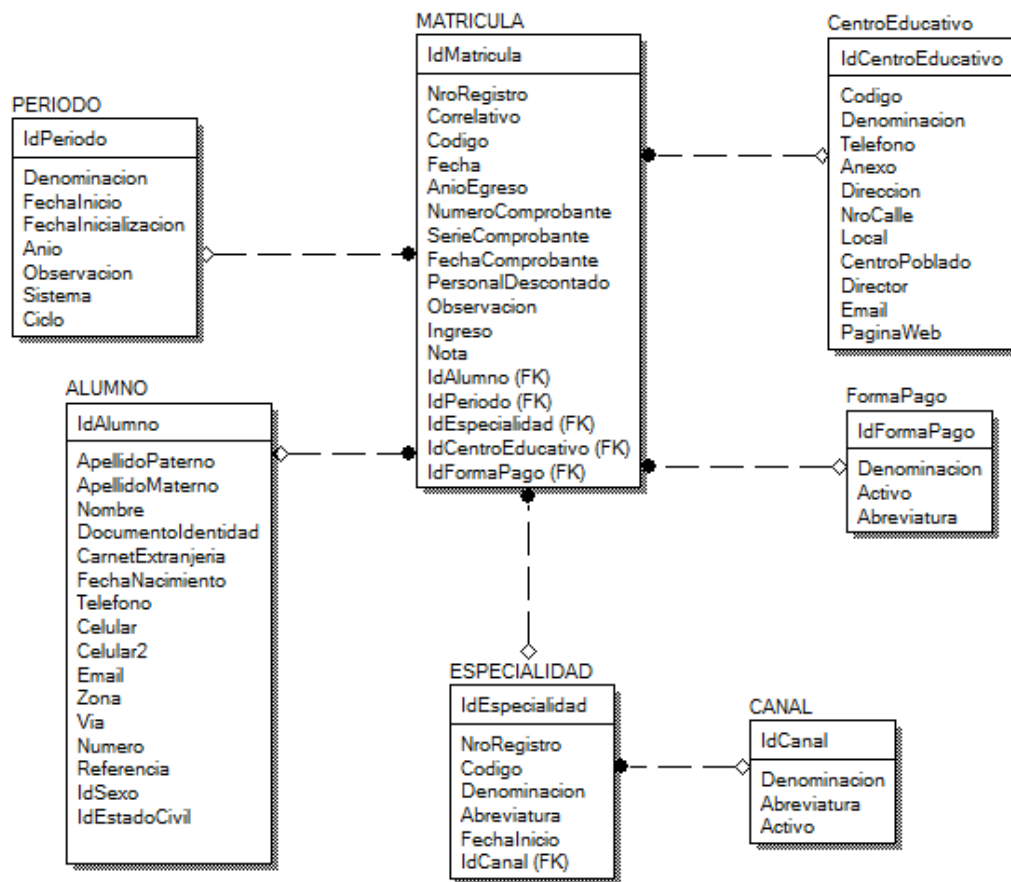
- La base de datos actual cuenta con 28 tablas, donde la mayoría son tablas flotantes sin ninguna relación. Para poder realizar la migración de datos hacia el modelo no relacional se tuvo que adaptar el modelo actual, eliminando las tablas flotantes que no tienen alguna relación (cardinalidad), reduciéndolo a 7 tablas.

Figura 9
Modelo entidad-relación del sistema de registro de postulantes



Fuente: Elaboración propia

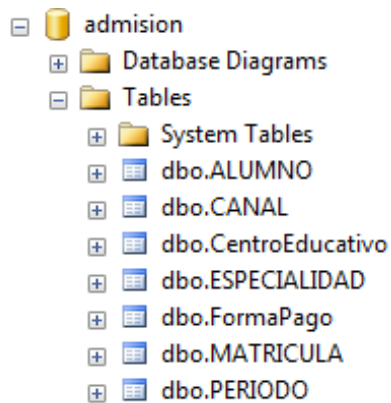
Figura 10
Modelo relacional del sistema de registro de postulantes



Fuente: Elaboración propia

Inmediatamente se procedió a generar la nueva base de datos en SQLServer a partir del nuevo modelo relacional, la cual se denominará “admisión”.

Figura 11
Base de datos en SQLServer



Fuente: Elaboración propia

Luego se procedió a poblar la base de datos, pasando los datos de la base de datos actual al modelo propuesto, para esto se utilizó consulta en SQLServer.

Figura 10
Consultas SQL para copiar datos entre tablas

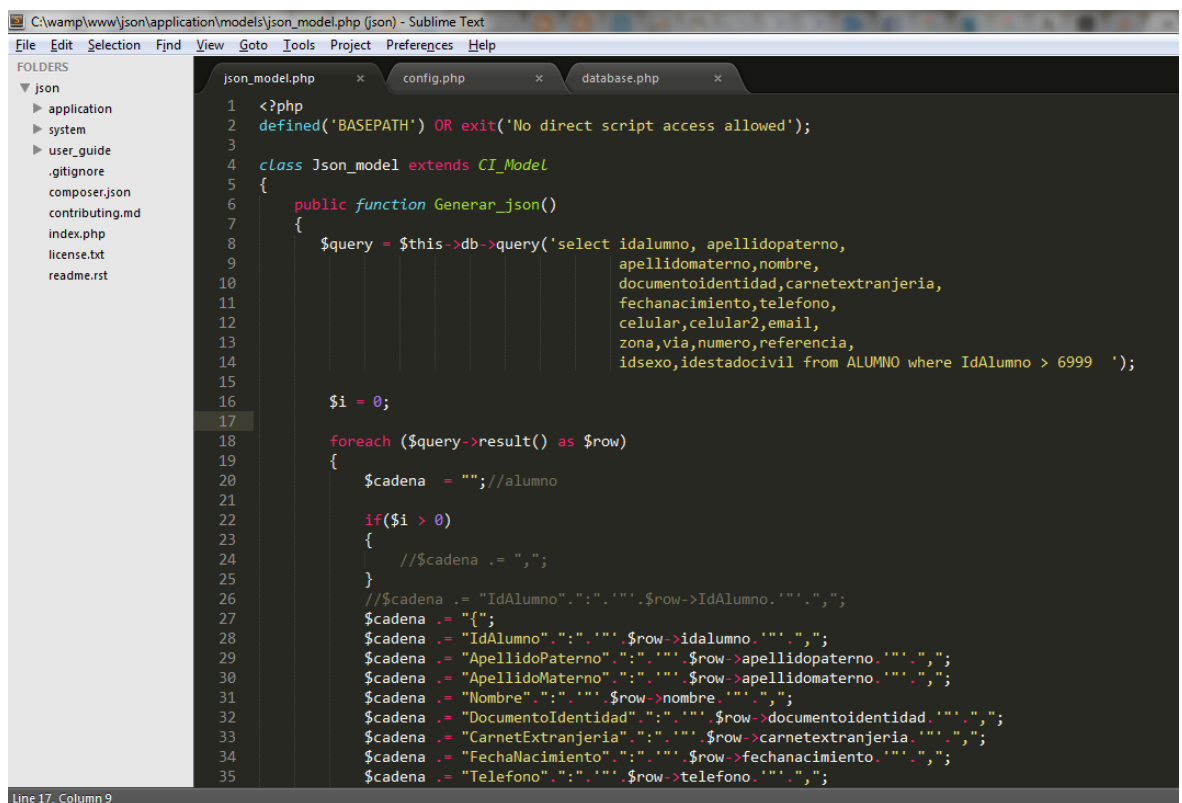
```
SQLQuery1.sql - lo...admission (sa (53))*
insert into admission.dbo.ALUMNO (IdAlumno,ApellidoMaterno,ApellidoMaterno,Nombre,
    DocumentoIdentidad,CarnetExtranjeria,FechaNacimiento,Telefono,Celular,
    Celular2,Email,Zona,Via,Numero,Referencia,IdSexo,IdEstadoCivil)
select IdAlumno,ApellidoMaterno,ApellidoMaterno,Nombre,DocumentoIdentidad,
    CarnetExtranjeria,FechaNacimiento,Telefono,Celular,Celular2,Email,Zona,
    Via,Numero,Referencia,IdSexo,IdEstadoCivil
from SYSA_ADMI.dbo.OPE_Alumno
```

Fuente: Elaboración propia

MongoDB utiliza el formato BSON para almacenar documentos, para esto se desarrolló una pequeña aplicación en el lenguaje de programación PHP, la cual se conecta con la base de datos en SQLServer, y genera la estructura de datos que requiere MongoDB.

Figura 13

Código de la aplicación para migrar datos al modelo no relacional



```
1 <?php
2 defined('BASEPATH') OR exit('No direct script access allowed');
3
4 class Json_model extends CI_Model
5 {
6     public function Generar_json()
7     {
8         $query = $this->db->query('select idalumno, apellidopaterno,
9             apellidomaterno,nombre,
10            documentoidentidad,carnetextranjeria,
11            fechanacimiento,telefono,
12            celular,celular2,email,
13            zona,via,numero,referencia,
14            idsexo,idestadocivil from ALUMNO where IdAlumno > 6999 ');
15
16         $i = 0;
17
18         foreach ($query->result() as $row)
19         {
20             $cadena = ""; //alumno
21
22             if($i > 0)
23             {
24                 // $cadena .= ",";
25             }
26             // $cadena .= "IdAlumno"." ":" ". $row->IdAlumno. " " ;
27             $cadena .= "{";
28             $cadena .= "IdAlumno"." ":" ". $row->idalumno. " " ;
29             $cadena .= "ApellidoPaterno"." ":" ". $row->apellidopaterno. " " ;
30             $cadena .= "ApellidoMaterno"." ":" ". $row->apellidomaterno. " " ;
31             $cadena .= "Nombre"." ":" ". $row->nombre. " " ;
32             $cadena .= "DocumentoIdentidad"." ":" ". $row->documentoidentidad. " " ;
33             $cadena .= "CarnetExtranjeria"." ":" ". $row->carnetextranjeria. " " ;
34             $cadena .= "FechaNacimiento"." ":" ". $row->fechanacimiento. " " ;
35             $cadena .= "Telefono"." ":" ". $row->telefono. " " ;
```

Fuente: Elaboración propia

Una vez que la aplicación genera los BSON, estos son importados a la base de datos en MongoDB.

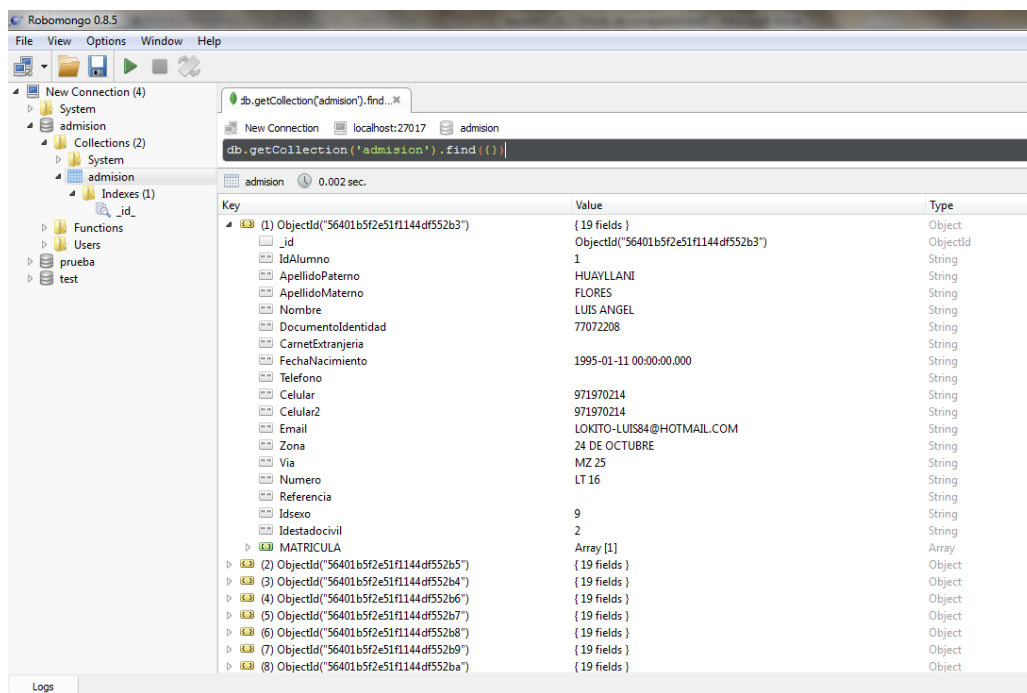
Figura 14
Código de aplicación para generar Json

```
{[IdAlumno:"7000",ApellidoPaterno:"ACERO",ApellidoMaterno:"ANCACHI",Nombre:"YANET",DocumentoIdentidad:"74728314",CarnetExtranjeria:"",FechaNacimiento:"1995-02-28",Telefono:"00-00-00-0000",Celular:"950055715",Celular2:"",Email:"YANETACERO@HOTMAIL.COM",Zona:"Via:TALACA PAMPA SITANA",Numero:"MZ-15, LT-16",Referencia:"A 2 CUADRAS DEL IE JUVENAL ORDÓÑEZ",IdEstadocivil:"2",MATRICULA:[{IdMatricula:"9068",NroRegistro:"1108",Correlativo:"1085",Codigo:"241085",Fecha:"2014-03-21 13:26:40.177",AnioEgreso:"2011",NumeroComprobante:"133291",SerieComprobante:"4",FechaComprobante:"2014-03-20"}],PersonalDescuento:"",Observacion:"",Ingreso:"",Nota:"",IdAlumno:"7000",ESPECIALIDAD:[{IdEspecialidad:"30",NroRegistro:"29",Codigo:"24",Denominacion:"CIENCIAS CONTABLES Y FINANCIERAS",Abreviatura:"ESCF",FechaInicio:"",CANAL:[{IdCanal:"4",Denominacion:"CIENCIAS ADMINISTRATIVAS, CONTABLE Y",Abreviatura:"CANAL 4",Activo:"1"}]},CentroEducativo:[{IdCentroEducativo:"1",Codigo:"0000000",Denominacion:"NO ESPECIFICADO",Telefono:"",Anexo:"",Direccion:"",NroCalle:"",Local:"",CentroPoblado:"",Director:"",Email:"",PaginaWeb:""}]},FormaPago:[{IdFormaPago:"2",Denominacion:"Deposito Banco",Activo:"1",Abreviatura:""}],Periodo:[{IdPeriodo:"13",Denominacion:"2014 -> FASE II",FechaInicio:"",FechaFinalizacion:"",Anio:"2014",Observacion:"",Ciclo:"sistema:1"}]}],{IdAlumno:"7001",ApellidoPaterno:"ZULOETA",ApellidoMaterno:"NEYRA",Nombre:"ROSA LILIA",DocumentoIdentidad:"75505681",CarnetExtranjeria:"",FechaNacimiento:"1997-08-18",Telefono:"00-00-00-0000",Celular:"952622773",Celular2:"",Email:"rosa_zuloeta_neyra@hotmail.com",Zona:"LAS AMERICAS",Via:"S/N",Numero:"MZ-U, LT-20",Referencia:"A 3 CUADRAS DEL PARADERO DE LA 2B",IdEstadocivil:"2",MATRICULA:[{IdMatricula:"9071",NroRegistro:"1111",Correlativo:"1093",Codigo:"201093",Fecha:"2014-03-21 13:29:39.680",AnioEgreso:"2013",NumeroComprobante:"179053",SerieComprobante:"4",FechaComprobante:"2014-03-21"}],PersonalDescuento:"",Observacion:"",Ingreso:"",Nota:"",IdAlumno:"7001",ESPECIALIDAD:[{IdEspecialidad:"21",NroRegistro:"20",Codigo:"20",Denominacion:"DERECHO Y CIENCIAS POLITICAS",Abreviatura:"ESDE",FechaInicio:"",CANAL:[{IdCanal:"3",Denominacion:"CIENCIAS SOCIALES",Abreviatura:"CANAL 3",Activo:"1"}]},CentroEducativo:[{IdCentroEducativo:"21449",Codigo:"0568618",Denominacion:"42003 GREGORIO ALBARRACIN",Telefono:"242664",Anexo:"0",Direccion:"CUZCO 444",NroCalle:"Local:486643",CentroPoblado:"TACNA",Director:"FEDERICO MAMANI FLORES",Email:"g_albaracin@yahoo.es",PaginaWeb:""}]},FormaPago:[{IdFormaPago:"2",Denominacion:"Deposito Banco",Activo:"1",Abreviatura:""}],Periodo:[{IdPeriodo:"13",Denominacion:"2014 -> FASE II",FechaInicio:"",FechaFinalizacion:"",Anio:"2014",Observacion:"",Ciclo:"sistema:1"}]},[IdMatricula:"13707",NroRegistro:"622",Correlativo:"",Codigo:"",Fecha:"2015-03-25 12:49:08.583",AnioEgreso:"2013",NumeroComprobante:"950989",SerieComprobante:"",FechaComprobante:"2015-03-25 00:00:00.000",PersonalDescuento:"",Observacion:"",Ingreso:"",Nota:"",IdAlumno:"7001",ESPECIALIDAD:[{IdEspecialidad:"21",NroRegistro:"20",Codigo:"20",Denominacion:"DERECHO Y CIENCIAS POLITICAS",Abreviatura:"ESDE",FechaInicio:"",CANAL:[{IdCanal:"3",Denominacion:"CIENCIAS SOCIALES",Abreviatura:"CANAL 3",Activo:"1"}]},CentroEducativo:[{IdCentroEducativo:"21449",Codigo:"0568618",Denominacion:"42003 GREGORIO ALBARRACIN",Telefono:"242664",Anexo:"0",Direccion:"CUZCO 444",NroCalle:"Local:486643",CentroPoblado:"TACNA",Director:"FEDERICO MAMANI FLORES",Email:"g_albaracin@yahoo.es",PaginaWeb:""}]},FormaPago:[{IdFormaPago:"2",Denominacion:"Deposito Banco",Activo:"1",Abreviatura:""}],Periodo:[{IdPeriodo:"21",Denominacion:"2015 -> FASE II",FechaInicio:"",FechaFinalizacion:"",Anio:"2015",Observacion:"",Ciclo:"sistema:1"}]}],{IdMatricula:"13717",NroRegistro:"632",Correlativo:"1152",Codigo:"261152",Fecha:"2015-03-25 13:06:30.960",AnioEgreso:"2013",NumeroComprobante:"1800384",SerieComprobante:"",FechaComprobante:"2015-03-24"}],PersonalDescuento:"",Observacion:"",Ingreso:"",Nota:"",IdAlumno:"7001",ESPECIALIDAD:[{IdEspecialidad:"32",NroRegistro:"31",Codigo:"26",Denominacion:"INGENIERIA COMERCIAL",Abreviatura:"ESCO",FechaInicio:"",CANAL:[{IdCanal:"4",Denominacion:"CIENCIAS ADMINISTRATIVAS, CONTABLE Y",Abreviatura:"CANAL 4",Activo:"1"}]},CentroEducativo:[{IdCentroEducativo:"21484",Codigo:"0716886",Denominacion:"42218 MARISCAL CACERES",Telefono:"310155",Anexo:"0",Direccion:"INTERNACIONAL 1249",NroCalle:"Local:487826",CentroPoblado:"CIUDAD NUEVA",Director:"BENEDICTO CABE DURAND",Email:"mariscalcaceres82@yahoo.es",PaginaWeb:""}]},FormaPago:[{IdFormaPago:"2",Denominacion:"Deposito Banco",Activo:"1",Abreviatura:""}],Periodo:[{IdPeriodo:"21",Denominacion:"2015 -> FASE II",FechaInicio:"",FechaFinalizacion:"",Anio:"2015",Observacion:"",Ciclo:"sistema:1"}]}],{IdAlumno:"7002",ApellidoPaterno:"LLANQUE",Nombre:"LUZ DELIA",DocumentoIdentidad:"76604126",CarnetExtranjeria:"",FechaNacimiento:"1996-01-04",Telefono:"00-00-00-0000",Celular:"952080848",Celular2:"",Email:"SON_DE_SC@HOTMAIL.COM",Zona:"24 de junio",Via:"s/n",Numero:"MZ, LT, 03",Referencia:"mas abajo de colegio gerardo arias"}
```

Fuentes: Elaboración propia

Para importar los datos en formato BSON, generados por la aplicación se utilizó la herramienta llamada **RoboMongo**, la cual facilita la gestión de bases de datos que utilizan MongoDB.

Figura 15
Herramienta para administrar bases de datos no relacionales



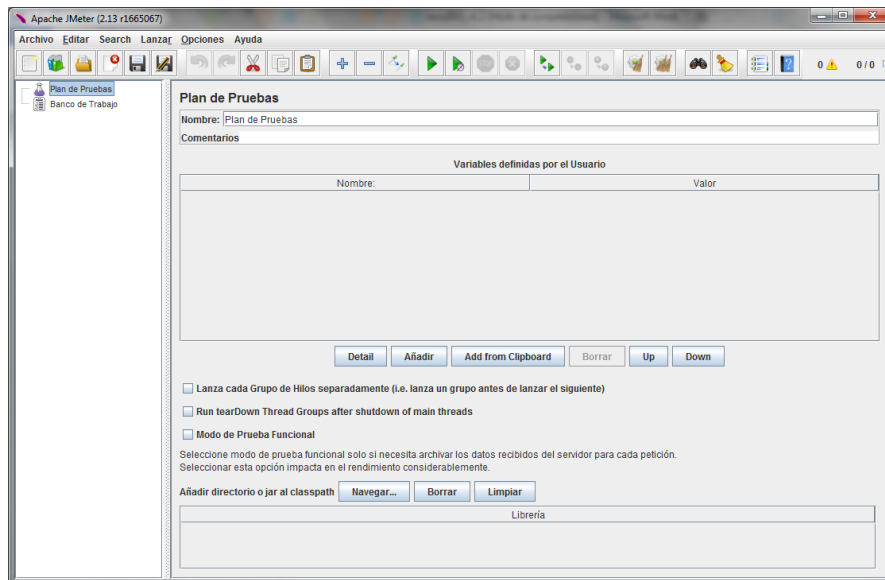
Fuente: Elaboración propia

3.2.1. Obtención de datos

Para analizar el tiempo de respuesta de las bases de datos que utilizan el modelo relación en SQLServer y el no relacional en MongoDB, se utilizó la herramienta Jmeter.

Jmeter es una herramienta de software libre muy completa, implementada en Java que permite realizar que permite medir el rendimiento, realizar pruebas de estrés.

Figura 16
Creación Plan de prueba usando la herramienta Jmeter



Fuente: Elaboración propia

Para poder evaluar los resultados del test de rendimiento se utilizaron 2 componentes provistos por la herramienta Jmeter.

➤ Summary Report: permite visualizar los resultados del test realizado.

Los datos que presenta son:

- Label: etiqueta de la muestra.
- #Muestras: cantidad de thread (hilos)
- Media: tiempo promedio en milisegundos (ms) para un conjuntos de resultados.
- Min: tiempo mínimo transcurrido en acceder a la base de datos.
- Max: tiempo máximo transcurrido en acceder a la base de datos.

- Rendimiento: rendimiento medido en base a peticiones por segundo/minuto/hora.
- Kb/sec: rendimiento medido en Kilobytes por segundo.
- Avg. Bytes: tamaño medio de respuesta del servidor (en bytes).

4.1. Plan de pruebas con Jmeter

Las pruebas de rendimiento se realizan en una laptop marca HP modelo Pavilion dv5 con las siguientes características:

- Sistema Operativo: Windows 7 Home Premium.
- Tipo de Sistema: 64 bits.
- Procesador: Intel Core I5 M480 2.67 GHZ.
- Memoria Ram: 8GB.

Figura 17
Especificaciones técnicas del computador

Ver información básica acerca del equipo

Edición de Windows

Windows 7 Home Premium
Copyright © 2009 Microsoft Corporation. Reservados todos los derechos.
[Obtener más características con una nueva edición de Windows 7](#)

Sistema

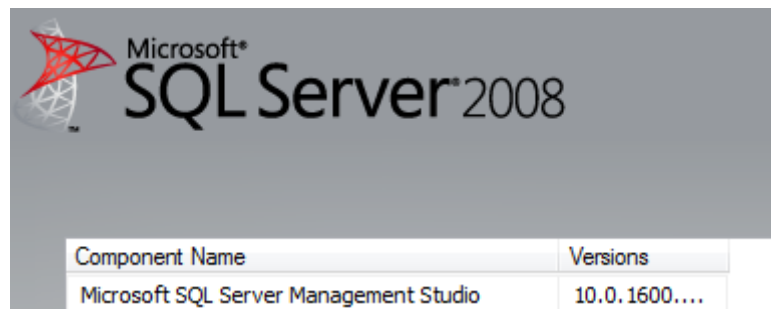
Fabricante:	Hewlett-Packard
Modelo:	HP Pavilion dv5 Notebook PC
Evaluación:	Evaluación de la experiencia en Windows
Procesador:	Intel(R) Core(TM) i5 CPU M 480 @ 2.67GHz 2.67 GHz
Memoria instalada (RAM):	8.00 GB (7.80 GB utilizable)
Tipo de sistema:	Sistema operativo de 64 bits
Lápiz y entrada táctil:	La entrada táctil o manuscrita no está disponible para esta pantalla

Fuente: Elaboración propia

También se usó SQLServer 2008 10.016, para el modelo relacional.

Figura 18

SQLServer 2008, usado para el modelo relacional

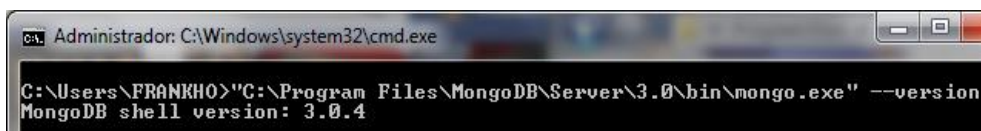


Fuente: Elaboración propia

Y MongoDB V. 3.0.4, utilizado para el modelo no relacional.

Figura 19

Versión de MongoDB en modo consola



Fuente: Elaboración propia

4.2. Plan de prueba de rendimiento para SQLSever

La prueba realizada consistió en realizar un test de 100 threads cada uno, los cuales simulan 2000 accesos de usuarios respectivamente, con un periodo de subido de 400 segundos.

Tabla 6*Plan de prueba para evaluar el rendimiento*

PRUEBA N°	N° USUARIOS	PERIODO	REPETICIONES
1	2000	400	1

Fuente: Elaboración propia

Para el test de rendimiento se usaron la misma consulta cada una con su propia sintaxis tanto para SQLServer y MongoDB, la cual muestra el listado de todos los postulantes del canal 2 y que sean de la escuela ESIS.

➤ SQLServer

Select a.IdAlumno, a.ApellidoPaterno, a.ApellidoMaterno, a.Nombre, a.DocumentoIdentidad, a.CarnetExtranjeria, a.FechaNacimiento, a.Telefono, a.Celular, a.Celular2, a.Email, a.Zona, a.Via, a.Numero, a.Referencia, a.IdSexo, a.IdEstadoCivil, b.IdMatricula, b.NroRegistro, b.Correlativo, b.Codigo, b.Fecha, b.AnioEgreso, b.NumeroComprobante, b.SerieComprobante, b.FechaComprobante, b.PersonalDescontado, b.Observacion, b.Ingreso, b.Nota, b.IdAlumno, b.IdPeriodo, b.IdEspecialidad, b.IdCentroEducativo, b.IdFormaPago, c.IdEspecialidad, c.NroRegistro, c.Codigo, c.Denominacion, c.Abreviatura, c.FechaInicio, d.IdCanal, d.Denominacion, d.Abreviatura, d.Activo, e.IdPeriodo, e.Denominacion, e.FechaInicio,

e.FechaFinalizacion, e.Anio, e.Observacion, e.Ciclo, e.sistema,
f.IdCentroEducativo, f.Codigo, f.Denominacion, f.Telefono, f.Anexo,
f.Direccion, f.NroCalle, f.Local, f.CentroPoblado, f.Director, f.Email,
f.PaginaWeb, g.IdFormaPago, g.Denominacion, g.Activo,
g.Abreviatura

From ALUMNO a

Inner Join MATRICULA b on a.IdAlumno = b.IdAlumno

Inner Join ESPECIALIDAD c on b.IdEspecialidad = c.IdEspecialidad

Inner Join CANAL d on c.IdCanal = d.IdCanal

Inner Join PERIODO e on b.IdPeriodo = e.IdPeriodo

Inner Join CentroEducativo

f on b.IdCentroEducativo = f.IdCentroEducativo

Inner Join FormaPago g on b.IdFormaPago = g.IdFormaPago

Where d.Abreviatura = 'CANAL 2' **And** c.Abreviatura='ESIS'

Order By a.ApellidoPaterno asc

➤ MongoDB

```
db.getCollection('admission').find($and:[  
{"MATRICULA.ESPECIALIDAD.CANAL.Abreviatura":"CANAL 2"},  
{"MATRICULA.ESPECIALIDAD.Abreviatura":"ESIS"}]  
}).sort({"ApellidoPaterno":1})
```

4.3. Procedimiento de evaluación de rendimiento con Jmeter

Se realizará una prueba de carga básica sobre una base de datos SQLServer y posteriormente se realizará un análisis de los resultados.

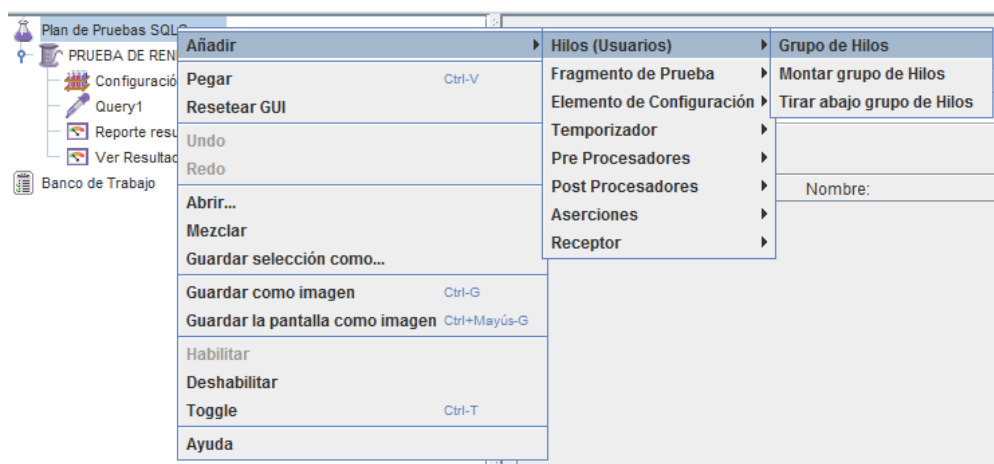
La prueba consiste en una simulación de conexiones concurrentes que realizaran una consulta a la base de datos.

- Crear un grupo de hilos(usuarios)

El grupo de hilos actúa como si fuera un usuario y permite controlar:

- El número de usuarios concurrentes que acceden a la base de datos.
- La frecuencia de accesos.
- El número de solicitudes.

Figura 11
Creación de Grupos de Hilos usando Jmeter



Fuente: Elaboración propia

- Configurar el grupo de hilos.

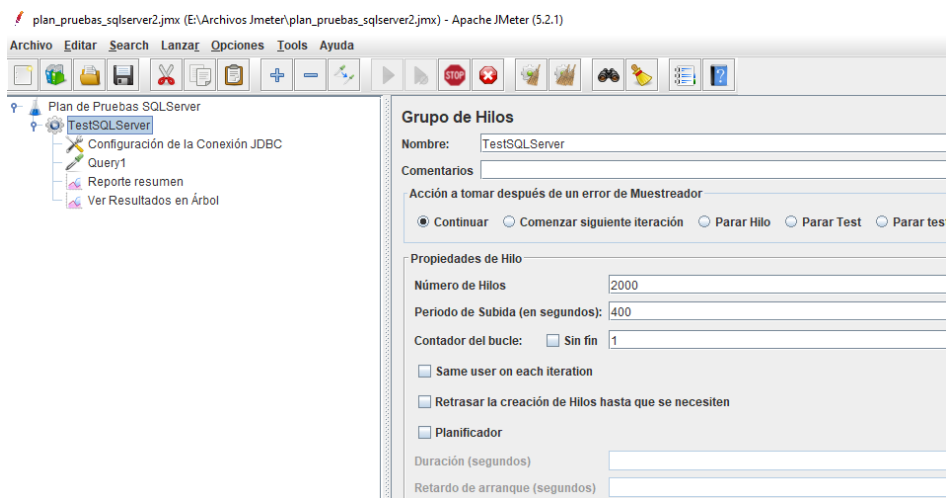
Para este test de rendimiento se configuro el grupo de hilos con los siguientes datos:

- Nombre: TestSQLServer
- Acción a tomar después de un error de muestreador: por defecto que es Continuar
- Numero de hilos (usuarios): 2000
- Periodo de subida (en segundos): 400
- Contador de bucle: 1

El periodo de subida se calcula de la siguiente manera:

- Periodo de carga = periodo de subida/número de hilos.

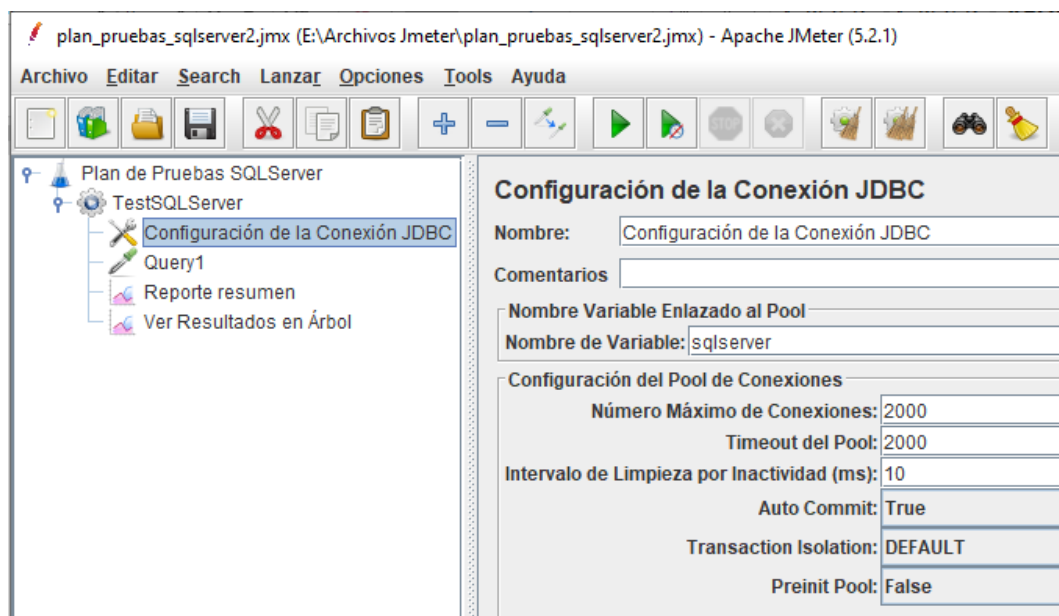
Figura 12
Configuración grupos de hilos – SQLServer



Fuente: Elaboración propia

- Configuración de conexión JDBC
 - Nombre: Configuración de la Conexión JDBC
 - Nombre de Variable: sqlserver
 - URL de la base de datos: jdbc:sqlserver://FRANKHO-HP:1433;databaseName=admisión
 - Clase del driver JDBC:
 - Usuario: sa
 - Password: *****

Figura 13
Configuración de la Conexión JDBC en Jmeter



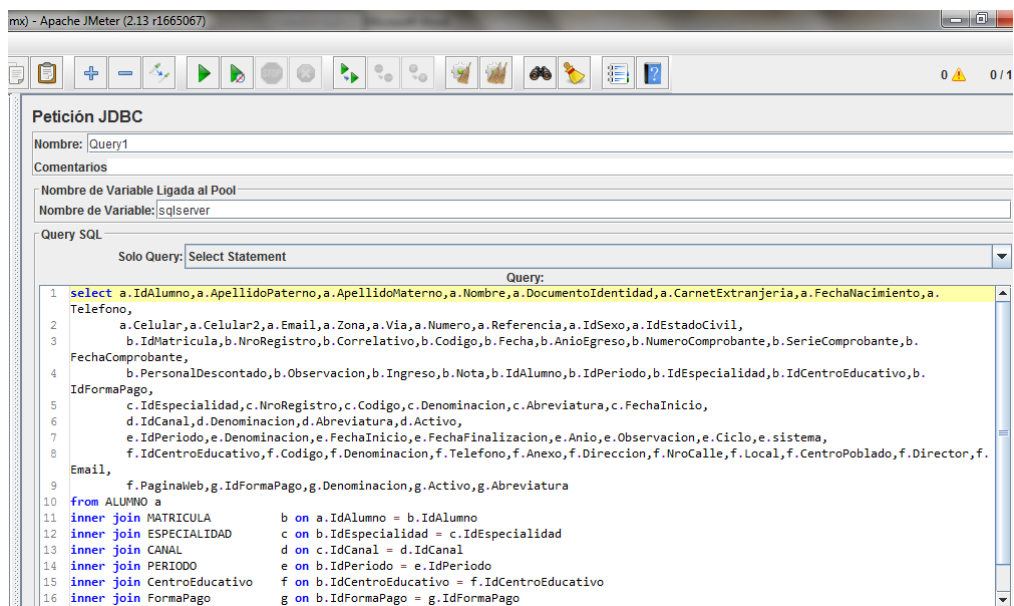
Fuente: Elaboración propia

➤ Crear petición JDBC

Aquí se configuran las sentencias o consultas SQL a ejecutar en la base de datos, y los configuramos con los siguientes datos:

- Nombre: Query1
- Nombre de Variable: sqlserver

Figura 14
Creación de petición – SQLServer



Fuente: Elaboración propia

➤ Muestra de resultados

Para poder visualizar/capturar e interpretar los datos, se necesitan los receptores o listeners.

Alguno de ellos es:

- Reporte resumen
- Ver resultados en árbol

4.4. Plan de prueba de rendimiento MongoDB

Se realizó una prueba de carga básica sobre una base de datos MongoDB y posteriormente se realizará un análisis de los resultados.

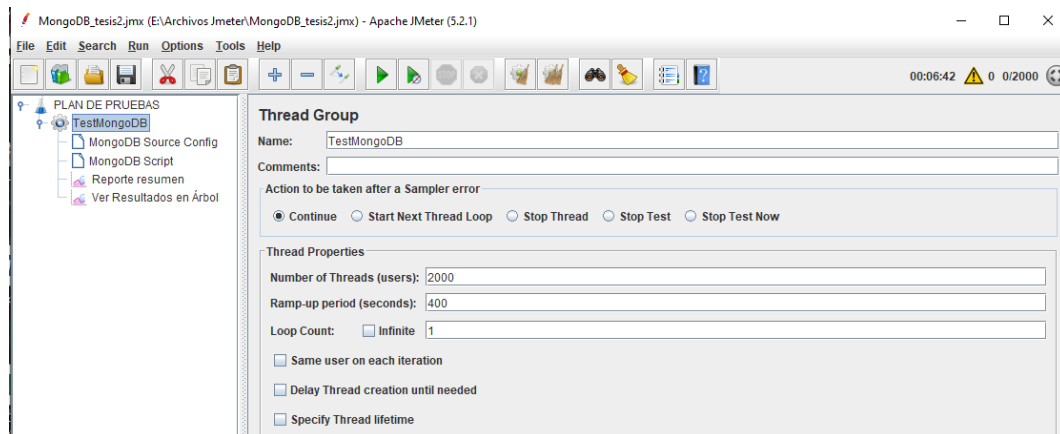
La prueba consiste en una simulación de conexiones concurrentes que realizaran una consulta a la base de datos.

➤ Configurar el grupo de hilos.

Para este test de rendimiento se configuro el grupo de hilos con los siguientes datos:

- Nombre: TestMongoDB
- Acción a tomar después de un error de muestreador: por defecto que es Continuar
- Numero de hilos (usuarios): 50
- Periodo de subida (en segundos): 5
- Contador de bucle: 1

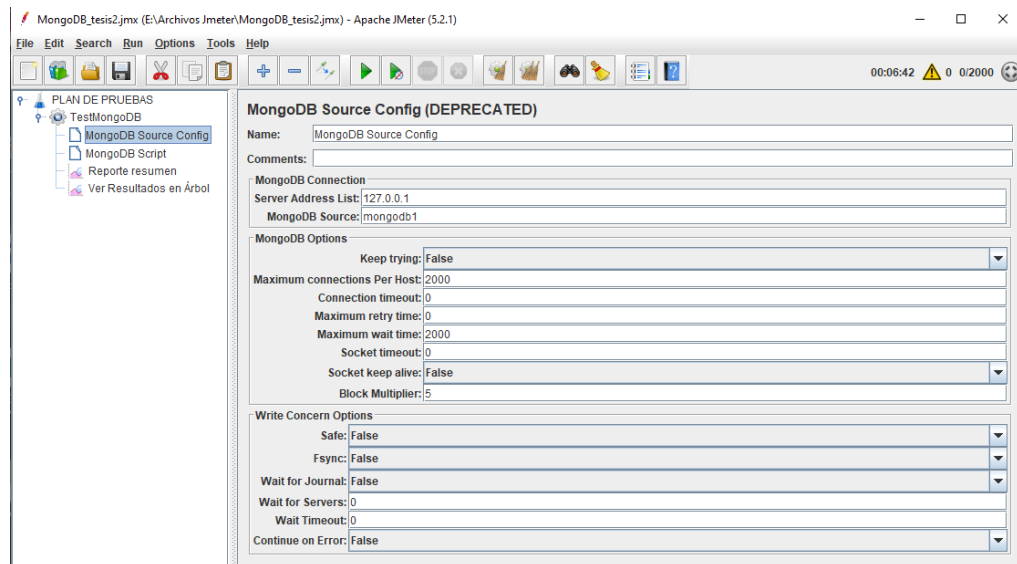
Figura 15
Configuración grupo de hilos – MongoDB



Fuente: Elaboración propia

- Configuración de conexión JDBC
 - Nombre: MongoDB Source Config
 - Server Address List: 127.0.0.1
 - MongoDB Source: mongodb1

Figura 16
Configuración de la conexión – MongoDB



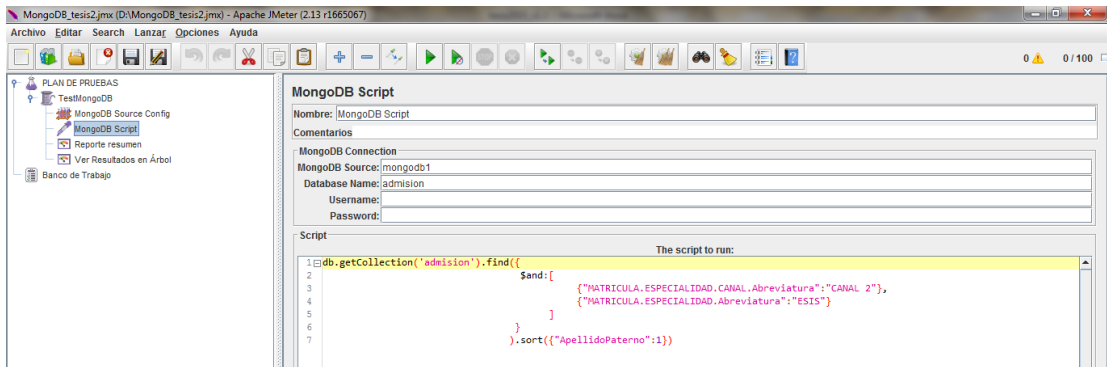
Fuente: Elaboración propia

➤ **Crear MongoDB Script**

Aquí se configuran las sentencias o consultas NoSql a ejecutar en la base de datos, y los configuramos con los siguientes datos:

- Nombre: MongoDB Script
- MongoDB Source: mongodb1
- Database Name: admisión

Figura 17
Consulta NoSQL en Jmeter



Fuente: Elaboración propia

CAPÍTULO IV

RESULTADOS

5.1. Resultados SQLServer usando modelo relacional

Luego de realizar el test de rendimiento con la herramienta Jmeter a la base de datos SQLServer, se obtuvo los siguientes resultados:

Tabla 7

Resumen de resultados de la prueba de rendimiento al modelo relacional

PRUEBA DE RENDIMIENTO	# Muestras	Media	Min	Max	Desv. Estándar	% Error	Rendimiento	Kb/sec	Media de Bytes
1	2000	2956	29	48496	9274.57	0.00%	5.00023	1888.63	386775
Total	2000	2956	29	48496	9274.57	0.00%	5.00023	1888.63	386775

Fuente: Elaboración propia

Para la primera prueba se configuraron 2000 usuarios con un periodo de subida de 400 segundos.

El tiempo total utilizado para los 2000 usuarios se puede calcular con la siguiente formula:

Figura 18*Tiempo total de consulta-SQLServer*

Tiempo Total = #Muestras * Media = 2000 * 2956 = 5912000 milisegundos = 98533.33 segundos
--

Fuente: Elaboración propia

5.2. Resultados MongoDB usando modelo no relacional

Luego de realizar el test de rendimiento con la herramienta Jmeter

a la base de datos MongoDB, se obtuvo los siguientes resultados:

Tabla 8*Resumen de resultados de la prueba de rendimiento al modelo no relacional*

MongoDB Script	#Muestras	Media	Min	Max	Desv. Estandar	Error %	Rendimiento	KB/sec	Media de Bytes
	2000	74	57	2083	70.31	0.00%	5.00161	247.68	50708
TOTAL	2000	74	57	2083	70.31	0.00%	5.00161	247.68	50708

Fuente: Elaboración propia

Figura 19*Tiempo total de consulta-MondoDB*

Tiempo Total = #Muestras * Media = 2000 * 74 = 148 000 milisegundos = 2466.66 segundos

Fuente: Elaboración propia

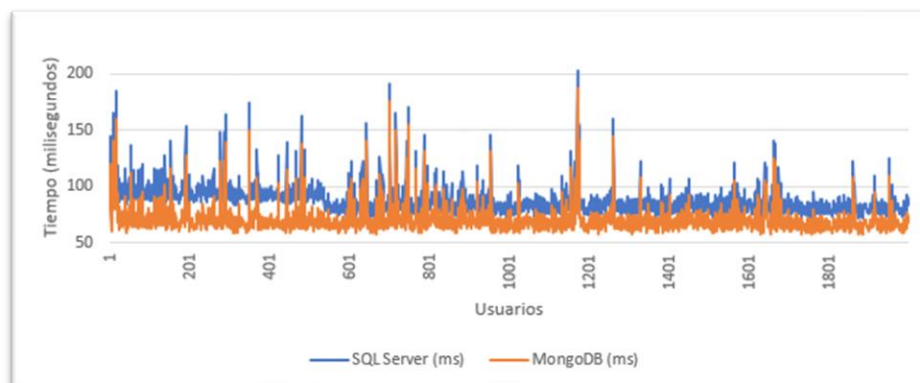
5.3. SQLServer vs MongoDB

5.3.1. Tiempo de respuesta de consulta

Se puede apreciar la comparación de tiempo de consulta a las bases de datos SQLServer y MongoDB. Se puede apreciar que el tiempo es más constante en relación a tiempo por usuario para la base de datos MongoDB, y para la base de datos que usa el modelo relacional tiene picos de hasta 140 milisegundos.

Figura 20

SQLServer vs Mongo en relación del tiempo por usuarios



Fuente: Elaboración propia

5.3.1. Consumo de KB/s

El consumo de KB/s para la base de datos SQLServer y la base de datos MongoDB, obtenidos del reporte resumen usando la herramienta Jmeter. Se puede apreciar que el consumo de KB/sec es menor en la base de datos Mongo.

Tabla 9

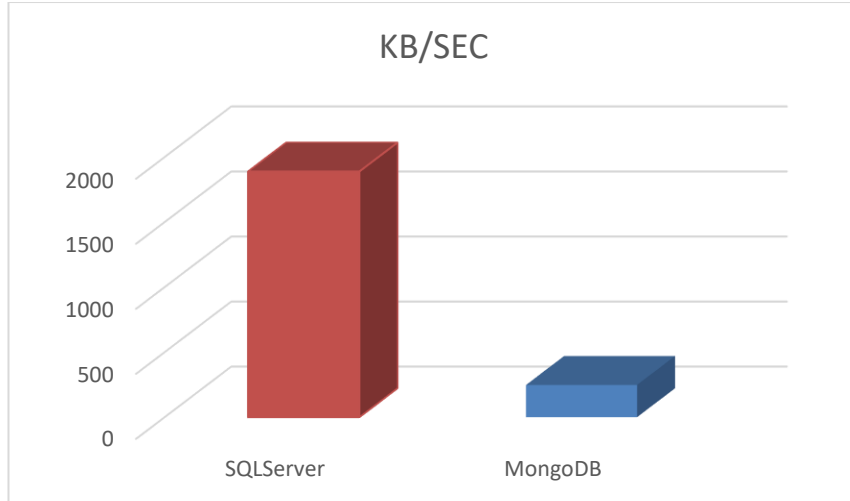
Consumo de KB/sec de la base de datos relacional y no relacional

KB/SEC	
SQLServer	1888.63
MongoDB	247.68

Fuente: Elaboración propia

Figura 21

Consumo de KB/Sec para SQLServer y MongoDB



Fuente: Elaboración propia

5.3.2. Consumo de almacenamiento en disco duro

El consumo de megabytes en disco duro por ambos modelos relacional y no relacional se detalla a continuación:

Figura 22
Megabytes ocupados por SQLServer

Nombre	Fecha de modifica...	Tipo	Tamaño
admisión	02/11/2015 10:21 a...	SQL Server Database Primary Data File	8,192 KB
admisión_log	02/11/2015 10:21 a...	SQL Server Database Transaction Log File	8,384 KB

Fuente: Elaboración propia

Figura 23.
Megabytes ocupados por MongoDB

Nombre	Fecha de modifica...	Tipo	Tamaño
admisión.0	01/11/2015 11:34 ...	Archivo 0	65,536 KB
admisión.ns	01/11/2015 11:34 ...	Archivo NS	16,384 KB

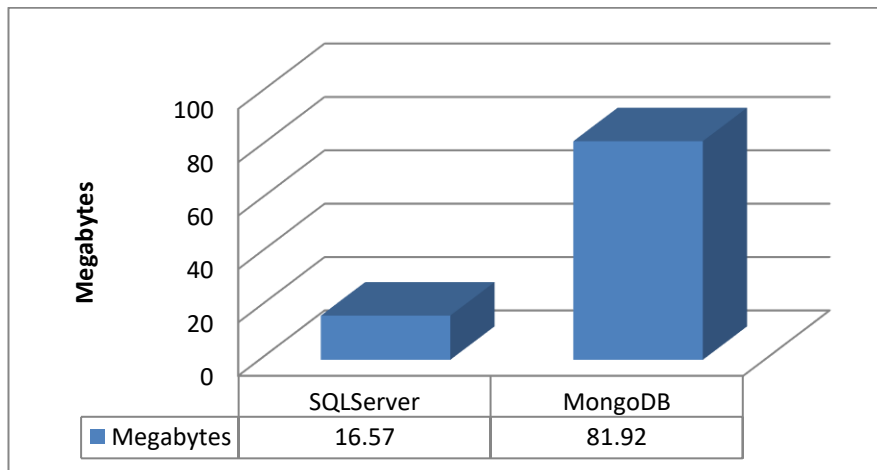
Fuente: Elaboración propia

Tabla 10
Comparación de almacenamiento SQLServer y MongoDB

	Megabytes
SQLServer	16.57
MongoDB	81.92

Fuente: Elaboración propia

Figura 24
Uso de disco duro SQLServer y MongoDB



Fuente: Elaboración propia

Se puede apreciar que el uso de disco duro es menor para la base de datos SQLServer con 16.57 megabytes en comparación con la base de datos MongoDB que ocupa 81.92 megabytes.

5.4. Contrastación de hipótesis

5.4.1. Hipótesis General

El rendimiento de la base de datos utilizando el modelo relacional es mejor que el modelo no relacional, en el Sistema Web de Registro de Postulantes de la Universidad Nacional Jorge Basadre Grohmann, Tacna 2015.

➤ Definición de variables

TA: Tiempo promedio para consultar a la base de datos que utiliza el modelo relacional (segundos)

TD: Tiempo promedio para consultar a la base de datos que utiliza el modelo no relacional (segundos)

➤ Hipótesis estadísticas

H₀: El tiempo promedio para consultar la base de datos que utiliza el modelo relacional, no es menor o igual que el tiempo promedio para consultar la base de datos que utiliza el modelo no relacional.

$$H_0: T_A \leq T_D$$

H₁: El tiempo promedio para consultar la base de datos que utiliza el modelo relacional, es mayor que el tiempo promedio para consultar la base de datos que utiliza el modelo no relacional.

$$H_0: T_A > T_D$$

➤ Prueba Estadística

Prueba T de Student para dos muestras independientes, ya que los datos no siguen una distribución normal.

- Nivel de significancia

El nivel de significancia (α) escogido para la prueba de hipótesis será del 5%, Por lo tanto el nivel de confianza ($1 - \alpha = 0.95$) será del 95%.

5.4.2. Prueba T de Student para dos muestras Independientes

utilizando el software SPSS

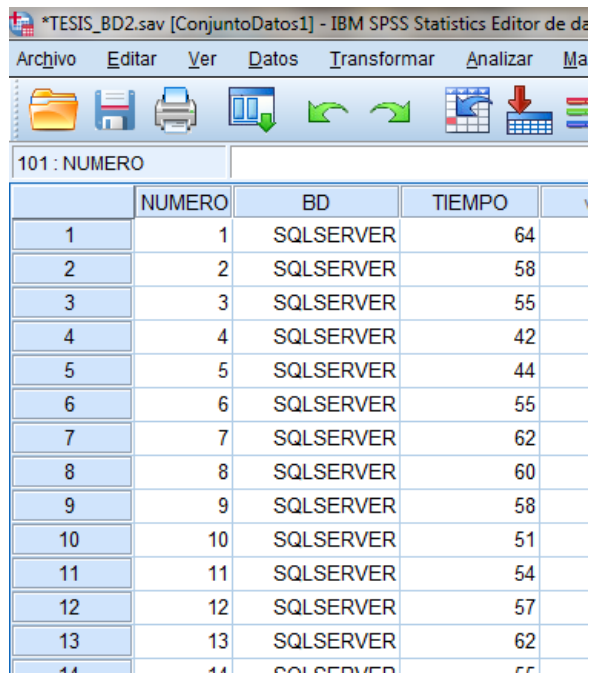
Para aplicar la prueba T de Student para dos muestras independientes se ha utilizado el software estadístico SPSS v. 23.

Dicha prueba debe cumplir las siguientes condiciones:

- Ingreso de datos

Se ha ingresado los datos luego de la simulación de consulta de datos realizada por 50 usuarios para cada base de datos (SQLSever y MongoDB).

Figura 25
Ingreso de datos al software SPSS



	NUMERO	BD	TIEMPO
1	1	SQLSERVER	64
2	2	SQLSERVER	58
3	3	SQLSERVER	55
4	4	SQLSERVER	42
5	5	SQLSERVER	44
6	6	SQLSERVER	55
7	7	SQLSERVER	62
8	8	SQLSERVER	60
9	9	SQLSERVER	58
10	10	SQLSERVER	51
11	11	SQLSERVER	54
12	12	SQLSERVER	57
13	13	SQLSERVER	62
14	14	SQLSERVER	55

Fuente: Elaboración propia

➤ Normalidad

Se usa esta prueba cuando los datos no siguen una distribución normal.

Para este caso se usó la prueba de Kolmogorov-Smirnov, ya que la muestra es mayor a 30.

El criterio para determinar si la (VA) se distribuye normalmente es necesario:

- a. P-valor $\geq \alpha$ Aceptar H0 = Los datos provienen de una distribución normal.
- b. P-valor $< \alpha$ Aceptar H1 = Los datos no provienen de una distribución normal.

Figura 26

Resultado de prueba de normalidad

Pruebas de normalidad				
BD		Kolmogorov-Smirnov ^a		
		Estadístico	gl	Sig.
TIEMPO	SQLSERVER	,124	50	,055
	MONGODB	,101	50	,200*

Fuente: Elaboración propia

Tabla 11

Resultado de la prueba de normalidad

NORMALIDAD		
P-Valor (SQLServer) = 0.055	>	$\alpha = 0.05$
P-Valor (MongoDB) = 0.200	>	$\alpha = 0.05$
Conclusión: La variable tiempo en ambos grupos se comporta normalmente.		

Fuente: Elaboración propia

- Igualdad de varianzas: se usa la prueba de Levene para saber si existen igual de varianzas

El criterio para determinar si la (VA) posee igual de varianza:

- P-valor $\geq \alpha$ Aceptar H0 = Las varianzas son iguales
- P-valor $< \alpha$ Aceptar H1 = Existe diferencia significativa entre las varianzas

Figura 27
Prueba de Levene

		Prueba de Levene de igualdad de varianzas	
		F	Sig.
TIEMPO	Se asumen varianzas iguales	,031	,860
	No se asumen varianzas iguales		

Fuente: Elaboración propia

Tabla 12
Resumen de prueba de Levene

IGUALDAD DE VARIANZA		
P-Valor (SQLServer) = 0.860	>	$\alpha = 0.05$
<u>Conclusión:</u> Existe igual de varianzas.		

Fuente: Elaboración propia

➤ Decisión estadística

Ya que se cumplieron los 2 requisitos: normalidad e igualdad de varianza, se procede a realizar la prueba t de student.

Figura 28
Resultado de la prueba T de Student

prueba t para la igualdad de medias						
t	gl	Sig. (bilateral)	Diferencia de medias	Diferencia de error estándar	95% de intervalo de confianza de la diferencia	
					Inferior	Superior
15,733	98	,000	15,280	,971	13,353	17,207

Fuente: Elaboración propia

El criterio para decidir es:

Si la probabilidad obtenida P-valor $\leq \alpha$, se rechaza H0 (Se acepta H1).

Si la probabilidad obtenida P-valor $> \alpha$, no se rechaza H1 (Se acepta H0).

Tabla 13
Decisión estadística

P-Valor = 0	<	$\alpha = 0.05$
<u>Conclusión:</u>		
Se rechaza H0, y se acepta H1		

Fuente: Elaboración propia

Como el valor de p no supera el nivel de significancia, se rechaza H_0 y se acepta H_1 , por lo tanto el rendimiento de una base de datos usando el modelo relacional no es mejor que una base de datos usando el modelo no relacional.

CAPÍTULO V

DISCUSIONES

El objetivo general de la presente investigación fue comparar el rendimiento de la base de datos del Sistema Web de Registro de Postulantes utilizando el modelo relacional y no relacional en la Oficina de Admisión de la Universidad Nacional Jorge Basadre Grohmann, para lo cual se tuvo que analizar la base de datos que se usa actualmente, y después de un análisis sobre las bases de datos que no usan el modelo relacional, se procedió a realizar la migración hacia dicho modelo. Luego ya teniendo las dos bases de datos una con el modelo relacional y otras con el no relacional, se realizó un test de rendimiento con la herramienta Jmeter.

Los datos de la presente investigación, fueron obtenidos a través de la herramienta Jmeter, la cual simulo consultas a las bases de datos SQLServer y MongoDB.

Según los resultados obtenidos la base de datos SQLServer y MongoDB se obtuvieron los siguientes resultados, un tiempo total de 4.1s y 3.2s.

Usando el modelo relacional demoro un tiempo total de 2.75 segundos, un tiempo promedio de 2 segundos con un consumo de 7460.2 KB/sec para 50 usuarios al momento de realizar consultas a la base de datos.

Para la base de datos MongoDB y SQLServer utilizan 2 segundos y 2.75 segundos, con un consumo de 14.73 KB/sec y 7460.2 KB/sec, respectivamente. Esto demuestra que el uso de una base de datos no relacional tiene un mejor rendimiento que una base de datos relacional.

CAPÍTULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Primera

Se determinó que el tiempo de respuesta a una consulta a la base de datos usando un modelo relacional es menor que usando una base de datos usando el modelo no relacional.

Segunda

El principal beneficio de usar una base de datos no relacional es la flexibilidad de almacenar datos, ya que estos no siguen esquemas, por lo que si se requiere agregar un campo extra no es necesario cambiar la estructura de nuestra base de datos.

Tercera

Se pudo comprobar que el tamaño de las bases de datos no relacionales es mucho más grande que una base de datos relacional, pero esto no es inconveniente en los tiempos de ejecución, ya que esto es más eficiente.

6.2. Recomendaciones

Primera

Para optar por una base de datos no relacional, se debe realizar un análisis minucioso, es vital determinar cada uno de los requerimientos y características que requiere, para de esta manera poder satisfacer las necesidades requeridas.

Segunda

Al momento de elegir una base de datos no relacional se debe tener en cuenta que se está sacrificando rendimiento por seguridad. En comparación con la base de datos relacional que poseen procedimientos, vistas, triggers, las bases de datos no relacionales no las poseen.

CAPÍTULO VII

REFERENCIAS BIBLIOGRÁFICAS

Libros

Elmasri, R., Navathe, S. (2007). Bases de datos y usuarios de bases de datos, (5ª ed.), *Fundamentos de Sistemas de Bases de Datos* (pp.4), Madrid: Pearson Educacion S.A

Graterol, Y. (2014). Introducción a NoSQL, *MongoDb en Español* (pp. 6)

Medina, J (2014), Para qué sirve una prueba de rendimiento, (2ª ed.), *Pruebas de Rendimiento TIC* (pp.21)

Piattini Esperanza, M. (2000).Modelo de datos relacional, *Diseño de Bases de Datos Relacionales* (pp. 93-94), Madrid: RA-MA

Sánchez, H., Reyes, C. (2006), *Metodología y diseño de la investigación científica*, Lima: Editorial Visión Universitaria

Artículos

Castro Romero, A; González Sanabria, J; Callejas Cuervo, M. (2012), *Utilidad y funcionamiento de las bases de datos NoSQL*, 21(33), 24-30, Recuperado el 28 de noviembre de 2015, de <http://dialnet.unirioja.es/descarga/articulo/5029469.pdf>

Díaz, J; Tzancoff, C; Rodríguez, A; Soria, V. (1900), Usando Jmeter para pruebas de rendimiento, Recuperado el 28 de noviembre de 2015 de http://www.linti.unlp.edu.ar/uploads/docs/usando_jmeter_para_pruebas_de_rendimiento.pdf

Echeverría, D; Abella, A. (2014), *Testing como Práctica para Evaluar la Eficiencia en Aplicaciones Web*, 2(5), 307, Recuperado el 28 de noviembre de 2015 de <http://sistemas.unla.edu.ar/sistemas/redisla/ReLAIS/relais-v2-n5-307-309.pdf>

Gracia del Busto, H.; Yanes Enríquez, O. (2012). *Bases de datos NoSQL*, 11(3), 21, Recuperado el 28 de noviembre de 2015, de <http://revistatelematica.cujae.edu.cu/index.php/tele/article/viewFile/74/74>

Jaramillo Valbuena, S; Mario Londoño, J. (2014), *Sistemas para almacenar grandes volúmenes de datos*, 13(37), 18, Recuperado el 28 de noviembre de 2015, de <http://revistas.uis.edu.co/index.php/revistagti/article/view/4689/4881>

Manjarrez Antaño, A; Martínez Castro, J; Cuevas Valencia, R. (2014), *Migración de Bases de Datos SQL a NoSQL*, Recuperado el 28 de noviembre de 2015, de http://www.researchgate.net/publication/272482056_Migracin_de_Bases_de_Datos_SQL_a_NoSQL

Martín, A; Chávez, S; Rodríguez, N; Valenzuela, A; Murazzo, M. (2013), *Bases de Datos NoSQL en Cloud Computing*, Recuperado el 28 de noviembre de 2015 de <http://sedici.unlp.edu.ar/bitstream/handle/10915/27121/Bases+de+Datos+NoSql+en+Cloud+Computing.pdf?sequence=1>

Ramírez Valenzo, M; Cuevas Valencia, R; Martínez Castro, J. (2011), *Integración de búsquedas de texto completo en Bases de Datos NoSQL*, 8(1), 81, Recuperado el 28 de noviembre de 2015, de <http://revistas.udistrital.edu.co/ojs/index.php/vinculos/article/view/4177/5837>

Rubio, M; Berlanga, V. (2012), Cómo aplicar las pruebas paramétricas bivariadas t de Student y ANOVA en SPSS, Recuperado el 28 de noviembre de 2015 de <http://diposit.ub.edu/dspace/bitstream/2445/45286/1/612532.pdf>

Vasquez Ortiz, Y.; Sotolongo León, A. (2013). *Mirada a bases de datos NoSQL de código abierto orientadas a documentos*, 6(9), 61, Recuperado el 28 de noviembre de 2015, de <http://publicaciones.uci.cu/index.php/SC/article/view/1382/720>

Tesis

Córdova, R. y Cuzco, B. (2013). *Análisis Comparativo Entre Bases De Datos Relacionales Con Bases De Datos No Relacionales*. Recuperada el 28 de noviembre de 2015 de <http://dspace.ups.edu.ec/bitstream/123456789/6977/1/UPS-CT003639.pdf>

Websites

MongoDB. (2015). RDBMS to MongoDB Migration Guide: *Considerations and Best Practices*. Recuperado el 28 de noviembre de 2015 de <https://www.mongodb.com/collateral/rdbms-mongodb-migration-guide>

ANEXOS

ANEXO 01

MATRIZ DE CONSISTENCIA

RENDIMIENTO DE LA BASE DE DATOS DEL SISTEMA WEB DE REGISTRO DE REGISTRO DE POSTULANTES UTILIZANDO EL MODELO RELACIONAL Y NO RELACIONAL EN LA OFICINA DE ADMISION DE LA UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN, TACNA - 2015

PROBLEMAS	OBJETIVOS	HIPOTESIS	UNIVARIABLE
<p>Problema General</p> <p>¿Cuál es el rendimiento de la base de datos del Sistema Web de Registro de Postulantes utilizando el modelo relacional y no relacional en la Oficina de Admisión de la Universidad Nacional Jorge Basadre Grohmann, Tacna-2015?</p>	<p>Objetivo General</p> <p>Comparar el rendimiento de la base de datos del Sistema Web de Registro de Postulantes utilizando el modelo relacional y no relacional en la Oficina de Admisión de la Universidad Nacional Jorge Basadre Grohmann, Tacna 2015.</p>	<p>Hipótesis General</p> <p>H₀: El rendimiento de la base de datos utilizando el modelo relacional es mejor que el modelo no relacional, en el Sistema Web de Registro de Postulantes de la Universidad Nacional Jorge Basadre Grohmann, Tacna 2015.</p> <p>H₁: El rendimiento de la base de datos utilizando el modelo relacional no es mejor que el modelo no relacional, en el Sistema Web de Registro de Postulantes de la Universidad Nacional Jorge Basadre Grohmann,</p>	<p>Variable dependiente</p> <p>Tiempo respuesta</p>


		Tacna 2015.	
<p>Problema Específico 1</p> <p>¿El uso de memoria ram es mayor usando una base de datos relacional que una base de datos no relacional?</p>	<p>Objetivo Específico 1</p> <p>Determinar y comparar el uso de memoria de una base de datos relacional y no relacional.</p>	<p>Subhipótesis 1</p> <p>H₀: El uso de memoria ram utilizando una base de datos relacional es mayor que utilizando un base de datos no relacional</p> <p>H₁: El uso de memoria ram utilizando una base de datos relacional es menor que utilizando un base de datos no relacional</p>	<p>Variable</p> <p>Cantidad de memoria ram (kb)</p>
<p>Problema Específico 2</p> <p>¿El uso de almacenamiento en disco duro de una base de datos relacional es mayor que una base de datos no relacional?</p>	<p>Objetivo Específico 2</p> <p>Determinar y comparar el uso de disco duro de una base de datos relacional y no relacional</p>	<p>Subhipótesis 2</p> <p>H₀: El uso de disco duro utilizando una base de datos relacional es mayor que utilizando una base de datos no relacional.</p> <p>H₁: El uso de disco duro utilizando una base de datos relacional es menor que utilizando una base de datos no relacional.</p>	<p>Variable</p> <p>Espacio utilizado en disco duro (MB)</p>

ANEXO 02

SISTEMA WEB DE REGISTRO DE

POSTULANTE


Administración



Acceso al Sistema de Registro de Postulantes. !

Usar su Número de Documento de Identidad válido para poder tener acceso a la Administración..

Inscripción en Línea



N° Documento(D.N.I.):

Modalidad
INSCRIPCIÓN CERRADA ▼

Actualizado el 06/01/2014 a Horas : 17:11 PM

[Vista Previa de la Guía de Inscripción](#)
[Vista Previa de las Vacantes 2014](#)
[Vista Previa de Cronograma y Requisitos](#)

SyadWeb © es un software de Inscripciones de postulantes.
Copyright © 2012 - ADMISION-UNJBG. Versión 1.0.0.0

Captura de Pantalla del Sistema web de Registros de postulantes

para el acceso a postulantes, disponible en

http://www.oasa.unjbg.edu.pe/WbfSYAM_Default.aspx

Universidad Nacional Jorge Basadre Grohmann					
Oficina de Admisión					
Datos Personales					
Número D. N. I.	12345678	Código Web		Código Insc.	
Apellido Paterno	ABC	Apellido Materno	DEF	Nombres	OPQ
Sexo	MASCULINO ▼	Fecha Nacimiento	12/12/1970	Estado Civil	SOLTERO ▼
Departamento Nacimiento	TACNA ▼	Provincia Nacimiento	TACNA ▼	Distrito Nacimiento	TACNA ▼
Departamento Procedencia	TACNA ▼	Provincia Procedencia	TACNA ▼	Distrito Procedencia	TACNA ▼
Lugar Residencia	TACNA ▼	Tipo de Zona	URBANIZACIÓN ▼	Zona	123
Tipo de Vía	AVENIDA ▼	Vía	123	Número	123
Referencia	CERCA DE 123			Teléfono	
Celular #	123456789	Celular 2		Nacionalidad	PERU ▼
E-mail #	ASD@HOTMAIL.COM	Lengua Materna	ESPAÑOL O CASTELLANO ▼	Segunda Lengua	INGLES ▼
Departamento Colegio	TACNA ▼	Provincia Colegio	TACNA ▼	Distrito Colegio	TACNA ▼
Colegio	FRANCISCO ANTONIO DE ZELA - Secundaria de Adultos -> 0499970 ▼			Tipo Colegio	ESTATAL ▼
Grado Instrucción	EDUCACIÓN SECUNDARIA ▼	Preparación	NO ESPECIFICADO ▼	Año Egreso	2013 ▼
Canal	NO ESPECIFICADO ▼	Carrera Profesional			
Forma Pago	No Especificado ▼	Serie Comprobante		Número Comprobante	
Fecha Comprobante		Modalidad Ingreso	No Especificado ▼		

SyadWeb ® es un software de Inscripciones de postulantes.
 Copyright © 2012 - ADMISION-UNJBG. Versión 1.0.0.0

Captura de pantalla de módulo de registro de postulantes

ANEXO 03

TIEMPOS DE CONSULTA

#	SQLServer (ms)	MongoDB(ms)
1	64	46
2	58	48
3	55	42
4	42	38
5	44	40
6	55	39
7	62	38
8	60	42
9	58	39
10	51	43
11	54	42
12	57	39
13	62	42
14	55	47
15	52	38
16	58	41
17	58	43
18	54	43
19	54	46

20	57	45
21	53	38
22	55	34
23	60	45
24	57	42
25	56	35
26	56	38
27	52	33
28	52	34
29	45	36
30	54	35
31	61	38
32	58	47
33	68	51
34	62	41
35	53	33
36	58	35
37	44	33
38	47	46
39	54	38
40	59	43

41	65	32
42	55	38
43	53	44
44	52	38
45	55	46
46	58	40
47	58	40
48	57	38
49	56	40
50	58	45