

# UNIVERSIDAD TÉCNICA DEL NORTE



FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

CARRERA DE INGENIERÍA EN SISTEMAS  
COMPUTACIONALES

**“BENCHMARKING DE BASE DE DATOS NOSQL: PROTOTIPO DE REGISTRO DE  
SERVICIOS HOTELEROS DEL HOTEL “SIERRA NORTE”**

TRABAJO DE GRADO PREVIO A LA OBTENCIÓN DE TÍTULO DE  
INGENIERA EN SISTEMAS COMPUTACIONES

AUTOR:

JOSE DAVID ORTIZ YEPEZ

DIRECTOR:

ING. DIEGO JAVIER TREJO ESPAÑA

IBARRA-ECUADOR

2023



# UNIVERSIDAD TÉCNICA DEL NORTE

## BIBLIOTECA UNIVERSITARIA

### AUTORIZACIÓN DE USO Y PUBLICACIÓN

#### A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

#### 1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

DATOS DEL AUTOR	
CÉDULA DE IDENTIDAD	1002549846
APELLIDOS Y NOMBRES	JOSE DAVID ORTIZ YEPEZ
DIRECCIÓN	Tobías Mena 5-11 y Calixto Miranda
E-MAIL	sevenortiz2012@gmail.com
TELÉFONO MÓVIL	0960550565

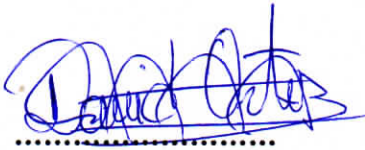
DATOS DE LA OBRA	
TÍTULO	<b>“BENCHMARKING DE BASE DE DATOS NOSQL: PROTOTIPO DE REGISTRO DE SERVICIOS HOTELEROS DEL HOTEL “SIERRA NORTE”</b>
AUTOR	JOSE DAVID ORTIZ YEPEZ
FECHA	06-09-2023
PROGRAMA	PREGRADO
TÍTULO POR EL QUE OPTA	INGENIERÍA EN SISTEMAS COMPUTACIONALES
DIRECTOR	ING.DIEGO JAVIER TREJO ESPAÑA

## · 2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 06 días del mes de septiembre de 2023

### **EL AUTOR:**

A handwritten signature in blue ink, appearing to read 'José David Ortiz Yépez', written over a dotted line.

Firma

Nombre: José David Ortiz Yépez



UNIVERSIDAD TECNICA DEL NORTE

FACULTAD DE INGENIERIA EN CIENCIAS APLICADAS

### CERTIFICACION DIRECTOR DE TESIS

Por medio del presente yo Ing. Diego Trejo, certifico que el Sr. José David Ortiz Yopez, portador de la cédula de identidad Nro. 1002549846., ha trabajado en el desarrollo del proyecto de tesis **“BENCHMARKING DE BASE DE DATOS NOSQL: PROTOTIPO DE REGISTRO DE SERVICIOS HOTELEROS DEL HOTEL “SIERRA NORTE”**, previo a la obtención del título de ingeniería en sistemas computacionales, lo cual ha realizado en su totalidad con responsabilidad.

Es todo cuanto puedo certificar en honor a la verdad.

ING. DIEGO JAVIER TREJO ESPAÑA

DIRECTOR DE TRABAJO DE GRADO

## **DEDICATORIA**

Dedico mi tesis en primer lugar a Dios, ya que fue mi fortaleza para culminar este largo proyecto en una etapa de mi vida.

Dedico a mis padres que gracias a su apoyo en todo sentido han sido motivación para culminar con éxito.

Y finalmente a mi mismo que tuve todo el valor y fortaleza para nunca rendirme a pesar de las adversidades que se me cruzaron en mi camino.

## **AGRADECIMIENTO**

A mi tutor, Msc. Diego Trejo, que principalmente a tenido la paciencia para poderme apoyar y guiar a culminar mi trabajo de grado, agradezco con sus amplios conocimientos pudo ser ese empuje a pesar de mi cambio de vida profesional, me motivo a culminar una gran etapa de mi vida.

A mis docentes, que tuve muchos, que cada uno supo enseñarme todos los valores éticos y profesionales que me han permitido llegar a ser una gran persona tanto en lo moral como en lo profesional.

A mis padres que sin ellos no seria nadie, mi padre ex Docente de la FICA, que gracias a su ejemplo de constancia y trabaja, fue un pilar para poder culminar mi carrera, mi madre que con su constante apoyo en todos los aspectos fue otra pieza fundamental para mi culminación.

A mi amiga y compañera la Ing. Veronica Angamarca, que me supo ayudar y dar unas guías para poder realizar mi trabajo de grado,

## Índice de Contenido

UNIVERSIDAD TÉCNICA DEL NORTE.....	i
<b>AUTORIZACIÓN DE USO Y PUBLICACIÓN .....</b>	<b>ii</b>
<b>A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE .....</b>	<b>ii</b>
<b>CONSTANCIAS .....</b>	<b>iii</b>
<b>CERTIFICACIÓN DIRECTOR DE TESIS .....</b>	<b>iv</b>
<b>DEDICATORIA.....</b>	<b>v</b>
<b>AGRADECIMIENTO .....</b>	<b>vi</b>
Índice de Contenido .....	vii
Índice de Figuras .....	x
ÍNDICE DE TABLAS.....	xii
RESUMEN.....	xiv
ABSTRACT .....	xv
INTRODUCCION .....	16
Antecedentes.....	16
Situación actual.....	16
Prospectiva .....	16
Planeamiento del Problema.....	17
Objetivos .....	17
Objetivo general .....	17
Objetivo Especifico .....	17
Alcance.....	18
Justificación.....	18
Económico.....	19
Social.....	19
CAPITULO I .....	20
MARCO TEORICO.....	20
INTRODUCCIÓN .....	20
1.1 Bases de Datos NoSQL y Bases de Datos SQL .....	20
¿Qué son las Bases de Datos NoSQL?.....	20
¿Qué son las Bases de Datos SQL?.....	22
1.1.1 Comparación entre las Bases de Datos NoSQL y SQL .....	22
1.1.2 Beneficios de las Bases de Datos NoSQL .....	23

1.1.3.- La consistencia de los datos en las Bases de Datos NoSQL .....	24
1.2 Clasificación de Base de datos NOSQL .....	25
1.3.- Bases de datos NOSQL a Investigar.....	29
1.3.1 MongoDB. - .....	29
1.3.2.- Apache Cassandra.....	30
1.3.3.- CouchDB.....	31
1.3.4.- Redis .....	32
1.3.5.- Neo4J .....	32
1.4.- Comparación de las Base de datos NOSQL Seleccionadas.....	33
1.5.- Razones para usar MongoDB .....	34
CAPITULO II.....	37
2..1. ANÁLISIS COMPARATIVO DE LAS BASES DE DATOS NOSQL.....	37
2.1.1 Introducción .....	37
2.1.2 Aplicación de la Encuesta. ....	38
2.1.3 Encuesta aplicada.....	38
2.1.4 Resultado de la encuesta aplicada a la población. ....	49
2.2.- REQUISITOS FUNCIONALES Y NO FUNCIONALES .....	50
2.2.1.- REQUISITOS FUNCIONALES .....	50
2.2.2 REQUISITOS NO FUNCIONALES .....	52
2.3 DESARROLLO .....	2
2.3.1 Análisis de Requerimientos .....	2
2.3.1.1 Roles del Sistema .....	2
2.3.1.2 Historias de Usuario .....	2
2.3.1.3 Product Backlog .....	7
2.3.1.4.- Roles del Develop team .....	8
2.3.1.5 Creación de los Sprints.....	8
2.3.1.5.1 Sprint 0 .....	8
2.3.1.5.2.- Sprint 1 .....	9
2.3.1.5.3 Sprint 2 .....	12
2.3.1.5.4 Sprint 3 .....	13
2.4 Diseño .....	16
2.4.1 Arquitectura del Sistema Web .....	17
2.4.2 Diagrama del Modelado de la Base de datos .....	18
2.5 Desarrollo del prototipo del Registro de Servicios Hoteleros. ....	18
2.5.1 Sprint 1 .....	18
2.5.2 Sprint 2 .....	20



2.5.3 Sprint 3 .....	23
2.5.4 Sprint 4 .....	26
2.6 Pruebas.....	31
CAPITULO III.....	34
VALIDACION DE RESULTADOS .....	34
<b>3.1.- Pruebas de Funcionamiento</b> .....	34
3.2 Evaluación de la velocidad de respuesta de los datos .....	35
3.3 Análisis e Interpretación de resultados.....	39
3.4 Validación de Resultados .....	41
CONCLUSIONES Y RECOMENDACIONES .....	44
CONCLUSIONES .....	44
RECOMENDACIONES .....	45
<b>BIBLIOGRAFIA</b> .....	46

## Índice de Figuras

<b>Figura 1.-</b> Planeamiento del Problema.....	17
<b>Figura 2.-</b> Estructura de la Base de Datos NOSQL. ....	21
<b>Figura 3.-</b> Clave-Valor.....	25
<b>Figura 4.-</b> Ejemplo Clave –Valor.....	26
<b>Figura 5.-</b> Ejemplo de base de datos documental. ....	27
<b>Figura 6.-</b> Base Datos de Grafos.....	28
<b>Figura 7.-</b> Ejemplo de base de datos en grafos. ....	28
<b>Figura 8.</b> Ejemplo de base de datos columnares.....	29
<b>Figura 9.-</b> Logo de la base de datos MongoDB.....	30
<b>Figura 10 .-</b> Logo base de Datos Cassandra.....	30
<b>Figura 11.-</b> Arquitectura de la base de tos Apache Cassandra.....	31
<b>Figura 12.-</b> Logo de las bases de datos CouchDB.....	31
<b>Figura 13.-</b> Logo de la Base de datos Redis. ....	32
<b>Figura 14.-</b> Logo de la base de Datos Neo4j.....	32
Figura 15.- Tipos de Escala de Likert-.....	38
<b>Figura 16.-</b> Resultados Pregunta 1.....	39
<b>Figura 17.-</b> Resultados de la Pregunta 2. ....	40
<b>Figura 18.-</b> Resultados de la Pregunta 3. ....	42
<b>Figura 19.-</b> Resultados de la Pregunta 4. ....	43
<b>Figura 20.-</b> <i>Resultados de la Pregunta 5.</i> ....	44
Figura 21.- Resultados de la Pregunta 6. ....	45
<b>Figura 22.-</b> Resultados de la Pregunta 7. ....	47
Figura 23.-Resultados de la Pregunta 8. ....	48
Figura 24.- Arquitectura del Prototipo de Registro de Servicios Hoteleros. ....	17
Figura 25.- Diseño de la base de datos. ....	18
Figura 26.- Configuración del Servidor.....	20
Figura 27.- Bienvenido a Sistema Web. ....	21
<b>Figura 28.-</b> . Inicio de Sesión de Clientes.....	21
Figura 29.- Menú de clientes .....	22
<b>Figura 30.-</b> Modificar Información del Cliente. ....	22
<b>Figura 31..</b> Ingreso como Administrador.....	23

Figura 32.- Menú Como Administrador. ....	23
<b>Figura 33.- Vista como</b> Usuario Administrador.....	24
Figura 34.- Vista como usuario clientes.....	25
Figura 35.- Registro de los Servicios Hoteleros. ....	25
Figura 36.- Uso de Servicios Hoteleros. ....	26
Figura 37.- Inicio para generar pagos. ....	26
Figura 38.- Genera al valor a pagar por el cliente.....	27
<b>Figura 39.-</b> Llenado de datos para guardar en pago.....	27
Figura 40.- Lista de Pagos. ....	28
<b>Figura 41.-</b> Historial de Reservasiones.....	28
Figura 42.- Ingreso y Despliegue de usuario cliente.....	29
Figura 43.- Ingreso y Despliegue de Administrador.....	30
<b>Figura 44.-</b> Rendimiento de la base de datos MongoDB.....	37
<b>Figura 45.-</b> Muestra de evaluación de velocidades en MongoDB con mejor Rendimiento en registros.....	39
Figura 46.- Comparativa entre las variables Muestras, Media y Desviación Estándar en MongoDB. ....	40
Figura 47.- Comparación entre las variables Muestras, Rendimiento y Error en MongoDB..	41
Figura 48.- Matriz de Correlación de Pearson de la Base de Datos MongoDB. ....	43
Figura 49.- Resultado de Correlación Muestra-Rendimiento.....	43
Figura 50.- Resultado de Correlación Rendimiento-Error.....	44

## ÍNDICE DE TABLAS

<b>Tabla 1.-</b> Comparación entre NOSQL y SQL	23
<b>Tabla 2.-</b> Benéficos de las bases de datos NOSQL.	23
<b>Tabla 3.</b> Consistencia de los datos en las Bases de Datos NoSQL.	24
<b>Tabla 4.</b> Comparación de las Bases de Datos seleccionadas.	33
<b>Tabla 5.-</b> Elección entre Base de datos Cassandra y MongoDB,	34
<b>Tabla 6.-</b> Términos que utilizados en MongoDB.	34
<b>Tabla 7.-</b> Razones el uso de MongoDB.	35
<b>Tabla 8.-</b> Resultados de la Pregunta 1	39
<b>Tabla 9.-</b> Resultados Resumen de Pregunta 1.	40
Tabla 10.-Resultados de la Pregunta 2.	40
Tabla 11.- Resultados de Resumen de la Pregunta 2.	41
Tabla 12.-Resultados de la Pregunta 3.	41
Tabla 13.-Resultado Resumen de la Pregunta 3.	42
Tabla 14.- Resultados de la Pregunta 4.	43
Tabla 15.- Resultados Resumen de la pregunta 4.	43
Tabla 16.- Resultado de Pregunta 5.	44
Tabla 17.- Resultado Resumen de la Pregunta 5.	45
Tabla 18.-Resultado de la Pregunta 6.	45
Tabla 19.-Resultado Resumen de la Pregunta 6.	46
Tabla 20.- Resultados de la Pregunta 7.	46
Tabla 21.-Resultados de Resumen de la Pregunta 7.	47
Tabla 22.- Resultados de la Pregunta 8.	48
Tabla 23.-Resultados Resumen de la Respuesta 8.	48
Tabla 24.- Tabla de Requisitos Funcionales.	50
Tabla 25.- Tabla de requisitos no funcionales.	52
<b>Tabla 26.-</b> Roles del Sistema.	2
<b>Tabla 27.-</b> Historia de Usuario RSB-001.	2
<b>Tabla 28.-</b> Historias de Usuario RSB-005.	3
<b>Tabla 29.-</b> Historias de Usuario RSB-006.	4
<b>Tabla 30.-</b> Historia de Usuario RSB-007.	4
<b>Tabla 31.-</b> Historia de Usuario RSB-008.	5
<b>Tabla 32.-</b> Historia de Usuario RSB-006.	5
<b>Tabla 33.-</b> Historia de Usuario RSB-010.	6

<b>Tabla 34.-</b> Historial de Usuario RSB-08	6
<b>Tabla 35.-</b> Historial Usuario RSB-09	6
<b>Tabla 36.-</b> Product Backlog (Pila de Producto).	7
<b>Tabla 37.-</b> Roles del Develop team.	8
<b>Tabla 38.-</b> Sprint Backlog Sprint 1.	9
<b>Tabla 39.-</b> Retrospectiva Sprint 1.	9
Tabla 40.- Backlog Sprint 1	10
Tabla 41.- Retrospectiva Sprint 2.	11
Tabla 42.- Sprint Backlog Sprint 2.	12
Tabla 43.- Retrospectiva Sprint 2.	13
Tabla 44.- Sprint Backlog Sprint 3.	14
Tabla 45.- Retrospectiva Sprint 3.	16
<b>Tabla 46.-</b> Sprint 1	18
<b>Tabla 47.-</b> Software especificado para el desarrollo.	19
<b>Tabla 48.</b> Sprint 2.	20
<b>Tabla 49.-</b> Sprint 3.	23
<b>Tabla 50.-</b> Sprint 4.	26
Tabla 51.- Prueba 1.	31
Tabla 52.- Prueba 2.	31
Tabla 53.- Prueba 3.	32
Tabla 54.- Prueba 4.	33
Tabla 55.- Demostración de valores de registro de usuarios	35
Tabla 56.- Selección de variables para la evaluación de velocidad de datos con MongoDB.	36
Tabla 57.- Rendimiento de la base de datos MongoDB	36
Tabla 58.- Selección de variables para la demostración de la velocidad de datos con MongoDB.	37
Tabla 59.- Demostración Rendimiento de la Bases de Datos NOSQL, para los registros.	38
Tabla 60.- Selección de variables para el análisis de los datos en MongoDB,	40
Tabla 61.- P-Value De Las Tres Bases De Datos.	42

## RESUMEN

El presente benchmarking o estudio comparativo entre las nuevas tendencias de las bases de datos NoSQL, existe un gran impacto de las nuevas tecnologías orientadas a la web, nos ayuda a aclarar sobre el uso de las BDD<sup>1</sup> de NoSQL, que nos permite optimizan recursos y hacen que su funcionalidad en el prototipo de registro de servicios hoteleros en el Hotel “Sierra Norte”, sea más rápido, y el uso de pocos recursos, para así tener mejores beneficios al implementar el sistema de registro se servicios hoteleros.

**Introducción**, se detalla el planteamiento de problema, situación actual, prospectiva, objetivos, alcance y justificación para el inicio del proyecto de la tesis.

**Capítulo I**, se da conocer el Marco Teórico, y se describe a cada una de las bases de datos NoSQL que son más utilizadas y son: Apache Casandra, MongoDB, Redis, CouchDB y Neo4J y da conocer sus conceptos, funcionalidades, razones y requerimientos. De la misma manera se describe los Roles, eventos y herramientas de la metodología Scrum.

**Capítulo II** .- se procede al desarrollo, haciendo uso de la metodólogo Scrum, donde se determina el análisis de requerimientos del sistema , análisis y selección de la base de datos, mediante una encuesta online, donde se elaboró nueve preguntas para poder seleccionar entre las cinco BDD NOSQL, mencionadas en la **plataforma QuestionPro**, a su vez también se da conocer el diagrama de modelado, de la misma manera comienza con el desarrollo del sistema aplicando los Sprints que son cuatro, con sus respectivas historias de usuarios, el cual se determina el tiempo que se demora en ejecutar cada historia de usuario y las pruebas necesarias para ver el uso cada uno de los Sprint.

**Capítulo III**.- se da a conocer la validación de resultados obtenidos una vez implementado el prototipo de registro de servicios hoteleros, a su vez se presentan las conclusiones, donde podemos indicar que la base de datos NoSQL seleccionada MongoDB, es la más apta y nos ayudan a optimizar recursos en su almacenamiento

**Capítulo IV**. Se completa con las conclusiones, recomendaciones y referencias bibliográficas.

---

<sup>1</sup> BDD: Base de Datos

## ABSTRACT

This NoSQL database benchmarking, among the new trends of NoSQL databases, due to the great impact of new web-oriented technologies, helps us clarify the use of NoSQL BDD, which allows us to optimize resources and They make its functionality in the prototype of registration of hotel services in the Hotel "Sierra Norte" faster, and the use of few resources, in order to have better benefits when implementing the registration system for hotel services.

Introduction, the problem statement, current situation, prospective, objectives, scope and justification for the start of the thesis project.

Chapter I, the Theoretical Framework is disclosed, each of the NoSQL databases that are most used is described and they are: Apache Cassandra, MongoDB, Redis, CouchDB and Neo4J and it discloses their concepts, functionalities, reasons and requirements. In the same way, the Roles, events and tools of the Scrum methodology are described.

Chapter II .- development is carried out, making use of the Scrum methodologist, where the analysis of system requirements, analysis and selection of the database is determined, through an online survey, where nine questions were distributed to be able to select between the five NoSQL databases mentioned, in turn the modeling diagram is also disclosed, in the same way it begins with the development of the system applying the Sprints that are four, with their respective user stories, which determines the time that it takes to execute each user story and the necessary tests to see the use of each of the Sprints.

Chapter III.- the validation of results obtained once the hotel services registration prototype has been implemented, in turn the conclusions are presented, where we can indicate that the selected NoSQL database MongoDB is the most suitable. and help us to optimize resources in your storage

Chapter IV. Finally, conclusions, recommendations and bibliographical references are added.

## INTRODUCCION

### Antecedentes

Los principales almacenamientos de las bases de datos existen dos tipos de gestores, el uno es SQL/BD relacionales y el otro es NoSQL/Bd no relacionales, ambos son modelos que han venido contribuyendo de manera primordial con el desarrollo tecnológico, dichas bases de datos son utilizadas por casi todos los usuarios, porque se presentan muy prácticas, y además ofrecen beneficios al momento de almacenar registros. Finalmente establecen unas estrategias sobre la gestión de los datos distantes al modelo tradicional, motivadas por la aparición de la Web 2.0, el Big Data y las aplicaciones web en tiempo real. El contexto en el que nace esta tecnología es de gran relevancia, ya que justifica las estrategias tomadas para cada tipo de base de datos NoSQL.(Tejero Gómez, 2022).

### Situación actual

En la Hostería Sierra Norte en los últimos años se ha logrado identificar un gran número de clientes, las misma demandan el uso de más almacenamiento para resguardar la información esto ha provocado que las Bases de Datos (BD) con las cuales trabajan , se lleguen a tener dificultades debido a que no soportan controlar grandes volúmenes de datos, porque su estructura de almacenamiento llega a tener conflictos cuando se presenta aglomeración de la información, es por esto se busca emplear otras herramientas para mejorar la calidad del servicio de información y gestión, al realizar esto ayudaría a solucionar los problemas pero con el inconveniente, que consumirían más recursos económicos de lo planificado para ese sector.

### Prospectiva

Desde el punto de vista, la selección de la tecnología de almacenamiento adecuado implica la consideración de numerosos aspectos, aunque el rendimiento suele ser el factor más importante, es necesario considerar ciertos aspectos como la funcionalidad, la facilidad de operación, sencillez de uso, disponibilidad de recursos para el trabajo continuo, los cuales son requerimientos que se necesita para solucionar las necesidades de distribución de información, gestión y procesamiento de la información (Shuib Basri, Rohiza Ahmad, Junzu Watada, Aparicio, 2018).

Es por ello que se ha visto la necesidad de un estudio comparativo sobre las BDD NoSQL, con el fin de investigar y comprobar si en la actualidad este sector de la hostería se ve conforme con la manera de trabajar de las bases de datos relacionales y si en algún momento han considerado optar por la utilización de otras herramientas tecnológicas, que los puedan ayudar a mejorar su calidad de servicio, al realizar esta investigación se podrá recolectar información necesaria para determinar si es viable el uso de Bases de Datos NoSQL, en los sistemas actuales y dicha información plasmarla en el estudio que se desarrollará, en donde se dará a conocer los resultados



de la investigación como también se optará por informar sobre las ventajas y oportunidades que se tiene al implementar este tipo de Bases de Datos (Cachimuel Loyo, 2020).

### Planeamiento del Problema

La falta de conocimiento sobre las bases de datos NoSQL, puede traer inconvenientes como: los desarrolladores al no conocer sobre este tipo de bases de datos, prefieren utilizar las tradicionales bases de datos relacionales y descartar otras opciones que podrían servirles de mejor manera, y les traería beneficios en sus proyectos a realizarse y con menos uso de recursos económicos para la hostería donde trabajen.

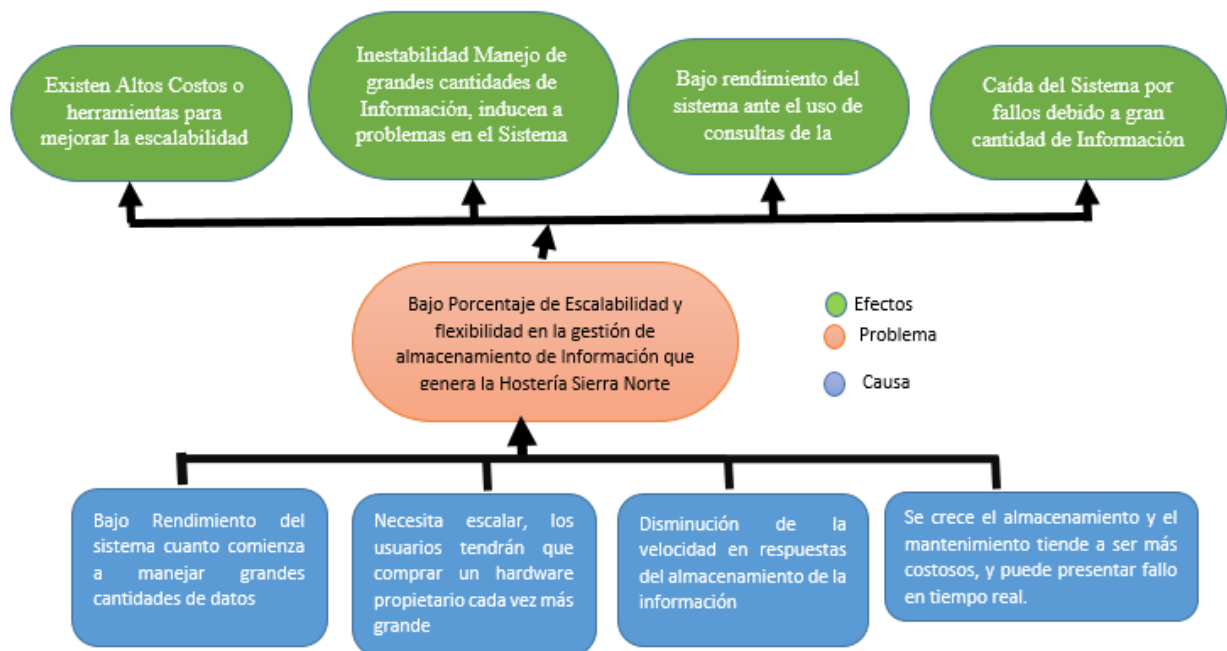


Figura 1.- Planeamiento del Problema.

### Objetivos

#### Objetivo general

- ✓ Realizar el “Benchmarking de Base De Datos NOSQL: Prototipo Registro de Servicios Hoteleros del Hotel “Sierra Norte”

#### Objetivo Especifico

- ✓ Estructurar el marco teórico basado en la investigación de las Bases de Datos NoSQL: MongoDB, Apache Cassandra, CouchDB, Redis y Neo4J
- ✓ Desarrollar un prototipo Registro de Servicios Hoteleros del Hotel “Sierra Norte” que permita almacenar el registro de los mismos.
- ✓ Validar los resultados.

## **Alcance**

Se desarrollará este proyecto en un prototipo de Registro de Servicios Hoteleros del Hotel “Sierra Norte” que contará con la implementación , en una de las Bases de Datos NoSQL, a las cuales se las evaluará mediante una encuesta a personas profesionales que desarrollan sistemas y se verificara la velocidad de respuesta de los datos, y determinar la capacidad de las características de los datos de tener el potencial intrínseco para satisfacer la necesidades explicitas o implícitas del Prototipo Registro de Servicios Hoteleros del Hotel “Sierra Norte”.

El objetivo principal de este estudio es establecer un estudio comparativo entre las bases de datos NoSQL para poder identificar la más apropiada que cuente con las características necesarias para poder utilizarla como una herramienta de comunicación con el proceso de registro de los servicios hoteleros, el fin de comprobar que las bases de datos NoSQL cumplen un mejor desempeño en lo que respecta a control de la gestión de la información.

## **Justificación**

La intención de implementar este proyecto es para investigar y recolectar información, para otorgar un procedimiento que revele el impacto de las bases de datos NOSQL en el sector de los servicios hoteleros, como también dar a conocer la importancia de estas bases de datos en el ámbito de desarrollo de sistemas modernos, en donde se dará a conocer las ventajas que brindan estas bases de datos como son: el control y manipulación de datos debido a sus estructura, performance, escalabilidad, flexibilidad, entre otras , todas estas ventajas hacen notar que este tipo de bases de datos sean las indicadas para trabajar en proyectos que controlen gran cantidad de información.

El actual proyecto tiene una guía hacia los siguientes objetivos de desarrollo sostenible:

- ✓ Construir bases resistentes, comenzar el desarrollo inclusivamente y sostenible y fomentar la innovación.
- ✓ Apoyar el desarrollo de tecnologías, la investigación e innovaciones nacionales en los países en desarrollo, incluso garantizando un entorno normativo propicio a la diversificación industrial y la adición de valor a los productos básicos, entre otras cosas (Maté et al., 2021).

Este proyecto está justificado por la falta de investigaciones e implementación de estudios sobre las Bases de Datos NoSQL, de acuerdo a lo averiguado en la localidad se ha realizado un estudio comparativo y es aplicado en un prototipo en una red social universitaria de la universidad técnica del norte y es tomada como base para investigación, de la misma manera si existen compañías en el extranjero que ya han utilizado estas bases para el uso de grandes cantidades de información y en algunos casos estas empresas han desarrollado sus propias bases de datos no relacionales como son: Amazon, Facebook, Twitter entre otras compañías, que han optado por estas bases de datos debido a su escalabilidad y flexibilidad con los datos que trabajan.

### **Económico**

El impacto a nivel económico implica disminuir gastos que se pueden presentar al utilizar las Bases de Datos tradicionales ya que las bases de datos NoSQL son en algunos casos open source lo que ayudaría a eliminar algunos procesos económicos innecesarios dentro del sector de los servicios hoteleros que decida cambiar de proceso de almacenamiento.

### **Social**

Al realizar el estudio comparativo se va contar con un impacto en la parte social, de tal forma dará a conocer a los programadores, todos los beneficios que tendrían al trabajar con estas bases de datos NOSQL y podrían implementarlas en sus proyectos.

## CAPITULO I

### MARCO TEORICO

#### INTRODUCCIÓN

En la actualidad los volúmenes de datos que se generan y consumen, se destacan por un crecimiento acelerado, lo que implica que los repositorios que contienen una colección de datos, no solamente sean considerados para realizar las consultas tradicionales, sino también como repositorios a partir de los cuales se puede obtener información relevante que sea útil para la toma de decisiones.

El modelo tradicional de base de datos relacional es consistente, sus ventajas y desventajas son bien conocidas. NOSQL<sup>2</sup> corresponde a una estrategia de persistencia que no sigue el modelo de datos relacional, y no utiliza SQL como lenguaje de consulta, en otras palabras, no están supeditadas a una estructura de datos en forma de tablas y relaciones entre ellas, permitiendo a los usuarios almacenar información en formatos diferentes a los tradicionales (Raquel Cáceres, Nélica; Tolada, Ana Carolina; Pérez, 2018).

Su principal ventaja es que, a diferencia de las bases de datos relacionales, manejan datos no estructurados como archivos de procesamiento de texto, correo electrónico, multimedia, y las redes sociales de manera eficiente (Morales, 2021). Así mismo son más fáciles de trabajar para los desarrolladores que no están familiarizados con el lenguaje de consulta estructurado. SQL es el lenguaje de programación más utilizado para consultar y actualizar datos relacionales bases de datos. Algunas BDD NoSQL pueden funcionar en un entorno distribuido de los usuarios, podría así escalar una sola base de datos, ejecutándolo a través de máquinas con costos adicionales en lugar de tener en una sola maquina más poderosa y costosa (Kiran Fahd, Samuel Kaspi &, 2016).

#### 1.1 Bases de Datos NoSQL y Bases de Datos SQL<sup>3</sup>

##### ¿Qué son las Bases de Datos NoSQL?

En la actualidad los sistemas web y aplicaciones han cambiado drásticamente en el mundo del internet de las cosas, y debido a ello en los últimos años se han realizado cambios en las empresas porque han tenido la obligación de automatizar todos sus procesos y reestructurar el manejo de su información e infraestructura tecnológica, es por ello que en la última década se ha venido mencionando lo que es Big Data, Big Users y Cloud Computing, dichas tecnologías controlan grandes cantidades información y por esta razón las mismas están teniendo como opción trasladar toda su información hacia las llamadas bases de datos NoSQL porque ellas poseen herramientas y estructuras adecuadas para simplificar operaciones a gran escala. A demás se puede enfatizar que muchos desarrolladores

---

<sup>2</sup> NOSQL: Not Only SQL

<sup>3</sup> SQL: lenguaje de consulta estructurada.

opinan que es factible utilizar el tipo de modelo de datos que es sin esquema porque es óptimo para controlar la gran variedad de datos que se procesan hoy en día dentro de las empresas (Ahmed et al., 2018)

Esto indica que, base de datos NoSQL (Not Only SQL), corresponde a una estrategia de persistencia que no sigue el modelo de datos relacional, no utiliza SQL como lenguaje de consulta; en otras palabras, no están supeditadas a una estructura de datos en forma de tablas y relaciones entre ellas, permitiendo a los usuarios almacenar información en formatos diferentes a los tradicionales, usando estrategias como clave-valor, mapeo de columnas, documentos o gráficos (Cachimuel Loyo, 2020).

En la siguiente Fig. 2, se muestra la estructura que manejan las bases de datos NoSQL.

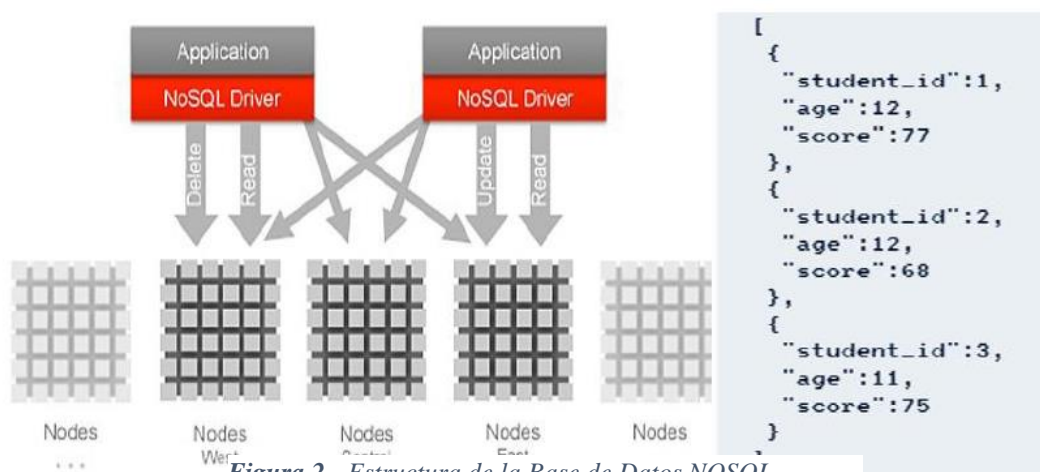


Figura 2.- Estructura de la Base de Datos NOSQL.

Fuente: (Cachimuel Loyo, 2020).

De acuerdo con su estructura y modo de operación con la información las bases de datos NoSQL no cumplen en su totalidad con las características de atomicidad, consistencia, aislamiento y durabilidad (ACID) y debido a ello escalan horizontalmente, por lo que proporcionan de esta manera eficiencia para ser utilizadas en entornos distribuidos (Sarasa, 2016) & (Cachimuel Loyo, 2020).

Según (Jaime & Arroyo Lerma, 2018), las ventajas son las siguientes:

**Ventajas:**

- ✓ **Por lo general emplean pocos recursos:** Estos a diferencia de los gestores SQL pueden montarse y funcionar perfectamente sobre hardware de pocos recursos.
- ✓ **Escalabilidad Horizontal:** Agregar nodos al sistema mejora su rendimiento debido a la propiedad que posee de escalamiento horizontal.

- ✓ **Manejo de grandes cantidades de datos:** Los gestores NoSQL prácticamente nacieron para cubrir esta necesidad, no es de extrañarse que se desarrollen mejor en este campo que los SQL, esto es posible gracias a que maneja una estructura distribuida.
- ✓ **Uso eficiente de los recursos:** aprovecha bien tecnologías como los discos sólidos, la memoria RAM y los sistemas distribuidos en general.
- ✓ **Esquema de datos flexible:** Esta propiedad nos otorga mayor libertad para modelar los datos, al no hacer uso de esquemas rígidos como las tablas en el caso de los gestores relacionales.
- ✓ **Modelo de concurrencia débil:** No implementa ACID, lo que quiere decir que reúne las características necesarias para que una serie de instrucciones sean consideradas una transacción.
- ✓ **Simplicidad en las consultas:** las consultas son más naturales por lo tanto se gana en simplicidad y eficiencia.

### ¿Qué son las Bases de Datos SQL?

Según (Joyanes Aguilar, 2019), los Sistemas de Gestión de Bases de Datos Relacionales (SGBDR), se apoyan en datos relacionales, donde se constituyen hoy en día el corazón de la mayoría de las plataformas distribuidas. De esta manera tiene el modelo de datos: entidad/relación.

El proceso inicial en el diseño de una base de datos es la creación de un modelo de datos, siendo la representación en escala de la realidad, además refleja la estructura de negocio de la organización, por medio de datos y relación, un modelo de datos entidad-relación (E-R) es útil para hacer corresponder los significados e interacciones de las empresas del mundo real con su esquema conceptual

El modelo entidad relación utiliza la notación tiene en cuenta conceptos como:

- Entidad: Objeto del mundo real distinguible de otros objetos. Una entidad se describe usando un conjunto de atributos.
- Conjunto de entidades: Una colección de entidades similares.
- Relación: Asociación entre dos o más entidades.
- Atributos: Son las propiedades que caracterizan un conjunto de entidades.
- Conjunto de relaciones: colección de relaciones similares.

#### 1.1.1 Comparación entre las Bases de Datos NoSQL y SQL

*Tabla 1.- Comparación entre NOSQL y SQL*

En la tabla 1, se observa la comparación entre las bases de datos NOSQL y SQL

PROPIEDAD	BASE DE DATOS NOSQL	BASE DE DATOS SQL
<b>Forma de almacenar los datos</b>	Documentos, clave -valor	Tablas
<b>Organización de los datos</b>	Estructura flexible o dinámico.	Estructura bien definida.
<b>Escalabilidad</b>	Horizontal.	Vertical
<b>Lenguaje de consulta</b>	Propio lenguaje de consulta	SQL estándar
<b>Relación de los datos</b>	Documentos anidados	Llaves Foráneas
<b>Seguridad</b>	No Existe	Transacciones, consistencia, aislamiento

Fuente: (Daniela & Karina, 2021)

### 1.1.2 Beneficios de las Bases de Datos NoSQL

*Tabla 2.- Benéficos de las bases de datos NOSQL.*

<b>Escalabilidad:</b>	<ul style="list-style-type: none"> <li>✓ Está orientado hacia el diseño de estructuras distribuidas.</li> <li>✓ Los nodos se crean de acuerdo con la disponibilidad en que los usuarios necesiten utilizar el almacenamiento en la base de datos,</li> <li>✓ Permite eliminar los costos y la complejidad de la fragmentación manual.</li> </ul>
<b>Alto Rendimiento:</b>	<ul style="list-style-type: none"> <li>✓ Optimización el uso de modelos de datos.</li> </ul> <p>Se enfocan al control de estructuras documentales, clave-valor y gráficos los cuales mediante patrones dan un acceso confiable y un rendimiento proporcional hacia la información que se esté almacenando.</p>
<b>Control de datos con flexibilidad:</b>	<ul style="list-style-type: none"> <li>✓ Ofrece libertad para los administradores de bases de datos en el ámbito de velocidad y flexibilidad en el cambio de esquemas y consultas</li> </ul>
<b>Desarrollo con agilidad:</b>	<ul style="list-style-type: none"> <li>✓ Soluciones a situaciones imprevistas ya que las bases de datos NoSQL</li> </ul>

	<ul style="list-style-type: none"> <li>✓ satisfacen de una manera rápida y ágil a las consultas efectuadas</li> <li>✓ obtener información en grandes cantidades.</li> </ul>
--	---

Fuente:(Cachimuel Loyo, 2020)& (Sarasa, 2016).

### 1.1.3.- La consistencia de los datos en las Bases de Datos NoSQL

Según (Sarasa, 2016), la consistencia está en trabajar bajo un perspectiva, en donde se plantea diferentes problemas que se han registrado en la seguridad de los datos, es por ello que se da a conocer cuatro tipos de consistencias que se presentan en las bases de datos NoSQL y son las siguientes:

Según (Sarasa, 2016)& (Cachimuel Loyo, 2020), consistencia

**Tabla 3.** *Consistencia de los datos en las Bases de Datos NoSQL.*

<b>Consistencia en las Escrituras:</b>	Cuando varios usuarios realizan proceso de actualización de un mismo dato al mismo tiempo, esto quiere decir que al momento que los usuarios actualiza un dato al mismo tiempo se le actualiza al usuario que actualizo primero y al siguiente usuario se le pierde la actualización debido a la última que se aplica se le sobrescribe en la primera. La solución planteada es tener bloqueos de escritura para que cuando un usuario esté actualizando, a los demás se le bloque hasta cuando culmine el usuario que se encuentre actualizando.
<b>Consistencia de lectura</b>	Existe el problema de lectura-escritura, cuando un usuario realiza una lectura de un dato y al mismo tiempo otro usuario desea hacer una escritura, donde se encuentra leyendo, por esto se ha planteado que la base de datos relacionales, se apliquen transacciones, donde el usuario solo aplique lecturas y luego escritura. Por concepto la BDD NOSQL no soporta transacciones incorrectas, por lo que son datos orientados a grafos, y llegan a soportar cierto tipo de transacciones como también las Bases de datos orientadas hacia agregados que puede soportar actualizaciones atómicas sobre un agregado
<b>Consistencia de sesión:</b>	Puede llegar a un problema de las ventanas de inconsistencia, ya que casi siempre se llega a mantener una carga que se encuentra



	<p>ejecutándose en un clúster que mantenga la carga de peticiones a los diferentes nodos donde sea solicitada una ventana, para controlar esta clase de situaciones es necesario implementar un tipo de consistencia de tipo leer lo que se escribe, lo que implica aplicar el concepto de actualización en donde se pueda verificar que solamente se pueda ver la actualización ejecutada por el usuario.</p>
--	--

Fuente:(Cachimuel Loyo, 2020)& (Sarasa, 2016).

## 1.2 Clasificación de Base de datos NOSQL

Según (ISRAEL, 2018), Las bases de datos no relacionales suelen clasificarse según la forma en que almacenan su información en:

- a) **Clave-Valor:** (Redis, Aerospike, Riak, etc.): Se observa en Fig.3, Como almacenan los datos en una estructura “clave-valor”, conocida como “diccionario”, cuyos valores no requieren un esquema fijo. En este tipo de base de datos no existe el concepto de relaciones, están pensadas para almacenar información básica y que pueda ser consultada de forma muy rápida. En este caso, cada registro almacenado está estructurado por dos componentes, una clave y un valor (José Manuel Reche, 2016).

La novedad con este tipo de bases de datos NoSQL es la posibilidad de tener los datos en un ambiente distribuido. Se recomienda su uso en casos donde se necesita velocidad en las consultas o se tienen muchos datos con estructura simple que requieren ser procesados una y otra vez y tienen valores cambiantes.

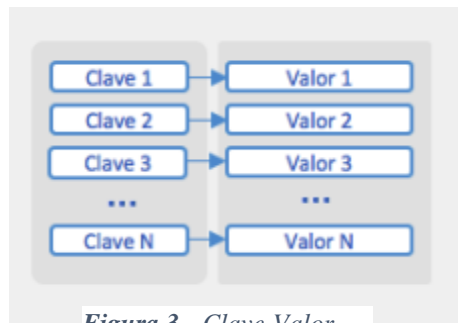


Figura 3.- Clave-Valor

Fuente:(Calvo, 2017)

Para comprender como está estructurada se presenta en la Fig. 5 un ejemplo donde se puede apreciar las respectiva clave y valor que se maneja dentro de estas bases de datos

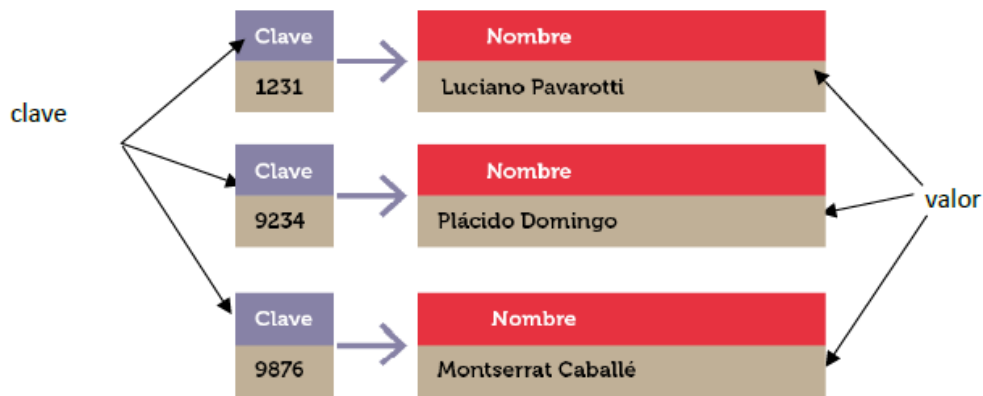


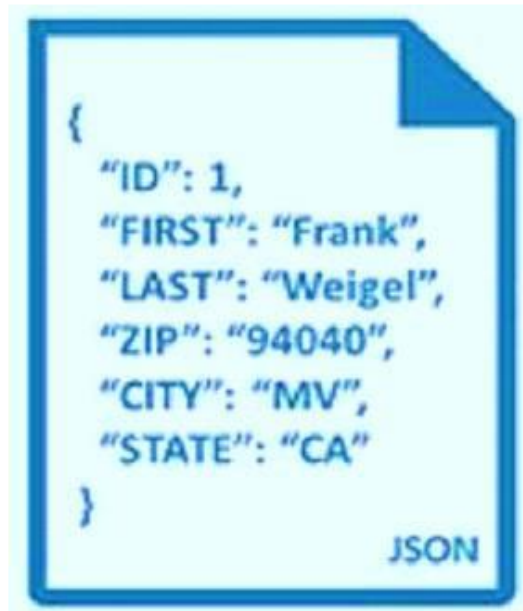
Figura 4.- Ejemplo Clave –Valor.

Fuente:(Cachimuel Loyo, 2020)& (Sarasa, 2016).

- b) **Base de datos Documental (MongoDb, de 10gen, y CouchDB, de Apache).** A diferencia de las filas y las columnas, las bases de datos NoSQL mantienen dos tipos de datos en documento. Cabe destacar que no debe confundirse el término documento con ficheros Excel, Word, etc. El término documento se refiere a una estructura de datos (clave, valor) representada en formato XML, JSON o BSON, que puede llegar a tener una estructura de datos anidados, por el cual estos documentos siguen un mínimo de reglas estándares para que la base de datos pueda entenderlo durante el post-procesamiento este es un documento más compacto y más legibles (Zuleta Sevilla, 2018).

La ventaja principal de JSON reside en su facilidad de uso, tanto para las personas como por los ordenadores. El formato JSON tiene tres fundamentos básicos:   
 } Pares de valores clave o atributos: JSON es almacenado en valores clave, a estos se les llama a veces atributos, las claves son cadenas simples y los valores pueden ser de cualquier tipo.   
 } Incrustación de objetos JSON: los valores clave también pueden ser objetos, lo que permite crear una jerarquía de objetos, colocar un objeto JSON dentro de otro se denomina modelo de datos incrustados, en base de datos documentales.   
 } Matrices: son un lenguaje de programación natural en todos los lenguajes de programación y estructura de datos. El formato JSON también admite almacenamiento de matrices como valores contra una clave (KYOCERA, 2017)

En la Fig. 5, se observa un ejemplo como se presenta un documento JSON con su respectiva clave y valores.

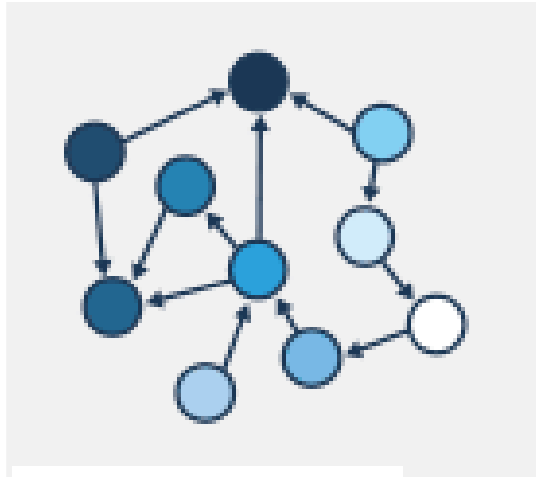


*Figura 5.- Ejemplo de base de datos documental.*

*Fuente:(Calvo, 2017).*

- c) **Modelo orientado a grafos (Neo4j, OrientDB, GraphDB):** los datos se almacenan en formato de grafos, son excelentes para la gestión de datos interconectados y superan el concepto de “relación” de las RDMBS. Su principal ventaja es que, a diferencia de las RDBMS, los grafos persisten tal cual (en modo binario) en la base de datos, con sus propiedades y relaciones, con lo cual las consultas son mucho más rápidas comparadas con una RDBMS (José Manuel, 2016).

Esto indica que gozan de un excepcional rendimiento para responder de forma eficiente al análisis y consulta de volúmenes gigantescos de datos. Ofrece también una diversidad de métodos analíticos y de consulta que la convierten en una de las opciones más flexibles en NoSQL. Es bastante frecuente conseguir la implementación de este modelo de base de datos en estructuras web de blogs. Twitter es uno de los casos más relevantes donde una base de datos de Grafos está relacionada, también otras funcionalidades que no te puedes perder. Entre las marcas más reconocidas está nuestra ya conocida Neo4j, pero esta no es la única (*Graph Everywhere, 2018*).

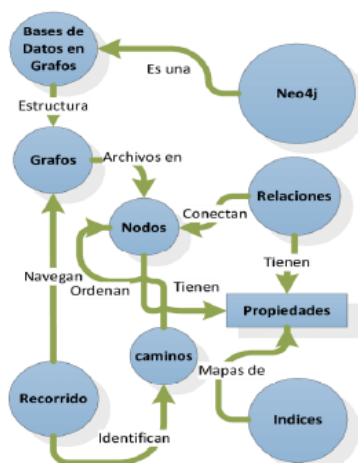


*Figura 6.- Base Datos de Grafos.*

**Fuente:**(Calvo, 2017)

Para la utilización de estas bases de datos es necesario que el modelado esté correctamente normalizado para que cada tabla pueda poseer una columna y cada relación adquiera solamente dos, lo que ocasionara que los cambios que se realicen en la estructura de la información solo tengan efectos a nivel local y no afecte los procesos de modelamiento. En cuanto al diseño y consulta, están orientados específicamente a datos en donde las relaciones están representadas con formas de grafos, lo que significa que la información representa a los datos que se encuentran interconectados con una cantidad no determinada de relaciones entre ellos. Mientras que las consultas a efectuarse deben ser mediante índices, ya que los sistemas que trabajan con este tipo de datos proporcionan una adyacencia libre de índices(Cachimuel Loyo, 2020).

En la Fig. 7, se muestra un ejemplo del recorrido que se realiza por los nodos mediante la utilización de una teoría de grafos.



*Figura 7.- Ejemplo de base de datos en grafos.*


**Fuente:**(Calvo, 2017).

**Bases de Datos de Columnares.** – Según (Cassandra, ScyllaDB, etc.), se almacenan los datos en forma de columnas (familias), a diferencia de las RDBMS, se almacenan los datos en forma de fila. Es decir, cada registro (fila) puede contener una o más columnas y cada columna puede contener distintas estructuras de datos además de forma versionada. Este tipo de bases de datos están pensadas para la consulta de datos de tipo histórico (logs, eventos, etc.), con el objetivo de obtener información agregada.

Posteriormente que para el rendimiento los sistemas columnares pueden llegar a superar a los sistemas que se afianzan con las bases de datos relacionales, pero dicha superioridad en muchos casos puede variar en varios aspectos, como son las consultas en las que incluyen los cálculos o accesos individuales a los registros los cuales pueden llegar a ser lentos o rápidos en comparación con los sistemas relacionales que poseen una indexación adecuada (Cachimuel Loyo, 2020).

En la Fig. 8 se muestra un ejemplo de cómo las columnas se encargan de agrupar los datos para poder realizar consultas.

clave fila	Datos personales		Datos profesional	
empid	Nombre	ciudad	designación	salario
1	raju	hyderabad	gerente	50,000
2	ravi	chennai	sr. ingeniero	30,000
3	rajesh	delhi	jr. ingeniero	25,000


  
 agrupación de datos

**Figura 8.** Ejemplo de base de datos columnares.

**Fuente:**(Lennin & Loyo, 2020).

### 1.3.- Bases de datos NOSQL a Investigar

Para realizar el benchmarking o estudio se han seleccionado, cinco de Bases de Datos NoSQL, las cuales fueron seleccionadas de acuerdo con la clasificación en donde informan los líderes y retadores en lo que respecta a sistema de gestión de bases de datos, es así como se hizo la respectiva investigación.

#### 1.3.1 MongoDB. -



*Figura 9.- Logo de la base de datos MongoDB.*

**Fuente:**(KATHERINE GEOMARA, 2019)

Son Bases de datos documentales una de las favoritas de los desarrolladores en MongoDB de la décima generación. Sus inicios son a finales de 2007, pero su liberación en licencia Open-source, fue en el año 2009. Este importante gestor de datos NOSQL almacena documentos en un formato muy parecido al JSON a alta velocidad. Construido en C++ tiene un rendimiento increíble que le permite ser muy flexible y potente. Es ideal para proyectos en los que se requiera alto nivel de escalabilidad. Tiene capacidad para recibir miles de lecturas por segundo sin pestañear. Múltiples gestores de contenido y juegos online se benefician de sus propiedades (Graph Everywhere, 2018).

En la fig. 9, se muestra el logo de la base de datos MongoDB.

### **1.3.2.- Apache Cassandra**



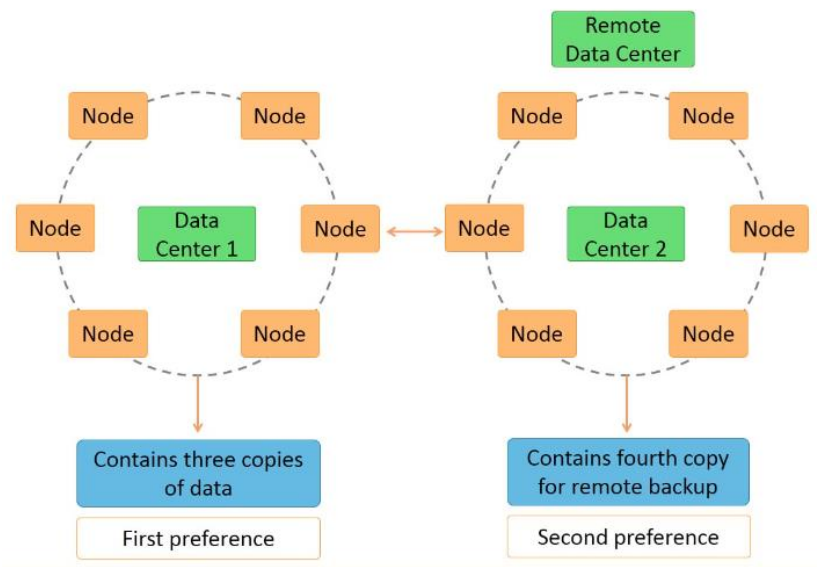
*Figura 10 .- Logo base de Datos Cassandra.*

**Fuente:**(KATHERINE GEOMARA, 2019)

Apache Cassandra nace desde como un proyecto de Google. Varios procesos han transcurrido desde esa época, hasta que en el año 2010, obtuvo su graduación como proyecto de alto nivel en Apache Incubador, de esta manera la base de datos, es de tipo clave-valor. Cassandra es una base de datos NoSQL que está diseñada para almacenar cantidades gigantescas de datos y realizar distribuciones a través de varios nodos. Esto permite que el almacenamiento de datos pueda

estar repartido entre diversos servidores sin un solo punto de fallo. La gran mayoría de nosotros tiene una cercanía con esta base de datos ya que es una de las herramientas esenciales de Facebook (Graph Everywhere, 2018).

En la Fig. 11, se muestra arquitectura de la base de datos Apache Cassandra.



**Figura 11.-** Arquitectura de la base de datos Apache Cassandra

**Fuente:**(lonami.dev, 2020)

### 1.3.3.- CouchDB



**Figura 12.-** Logo de las bases de datos CouchDB.

**Fuente:**(Zuleta Sevilla, 2018)

CouchDB es una base de datos NoSQL, que nace con aspiraciones bastante altas. Su principal desarrollador anhelaba a que CouchDB, se convirtiera en la base de datos más usada en internet. En 2008 el proyecto pasa a formar parte de Apache Incubator, llega a nosotros con la intención de facilitar la accesibilidad y compatibilidad web con diferentes tipos de dispositivos. Su primera versión estable llegó al público en el año 2010. En esta base de datos se incluyen en formato JSON. La forma en la que los datos son organizados es según pares de valor -clave (Zuleta Sevilla, 2018).

### 1.3.4.- Redis



*Figura 13.- Logo de la Base de datos Redis.*

*Fuente:(Calvo, 2017)*

Redis es la base de datos NOSQL de tipo clave-valor, es creada por Salvatore Sanfilippo y Pieter Noordhuis, está apoyado por VMWare, es de código abierto es patrocinado y desarrollado por RedisLabs. Su diseño principal está basado en el almacenamiento de tablas de hashes, aunque no es restrictiva solamente hacia este modelo. Se puede imaginar como un array gigante en memoria para almacenar datos, que pueden ser cadenas, conjuntos de datos o listas. Tiene la ventaja de que sus operaciones son atómicas y persistentes. Por ponerle un ejemplo, Redis no permite realizar consultas, únicamente se puede insertar y obtener datos, además de las operaciones comunes sobre conjuntos (diferencia, unión e inserción). Creado en ANSI C, por lo tanto, es compatible y funciona sin problemas en sistemas Unix, Linux y sus derivados, Solaris, OS/X, sin embargo, no existe soporte oficial para plataformas Windows (*José Manuel, 2016*).

### 1.3.5.- Neo4J



*Figura 14.- Logo de la base de Datos Neo4j.*

*Fuente:(Calvo, 2017).*

La base de datos Neo4j es desarrollada en software libre está orientada a grafos, construida con lenguaje Java, tiene un funcionamiento transaccional que almacena en estructura de grafos. Su primera versión fue lanzada en 2010. Esta BDD con diferentes tipos de grafos, tiene las características principales que sus desarrolladores es un alto desempeño, si las consultas empiezan a crecer de forma exponencial el rendimiento de la base datos no disminuye. Es Ágil, flexible y altamente escalable (Graph Everywhere, 2018).



#### 1.4.- Comparación de las Base de datos NOSQL Seleccionadas

Con la información alcanzada y detallada anteriormente en los subtemas se pudo conseguir establecer un cuadro comparativo con las principales características que diferencian a cada una de las bases de datos NoSQL que se van a utilizar para el estudio a realizar y en la Tabla 4, se puede comprobar dichas diferencias y se las puede representar de la siguiente manera.

**Tabla 4.** Comparación de las Bases de Datos seleccionadas.

Descripción	Apache Cassandra	Redis	MongoDB
Modelo de Bases de Datos Primaria	Su columna ancha de almacenamiento es basada en la estructura BigTable y DinamoDB	Su modelo de bases de datos permite almacenar los datos mediante tablas hashes.	Su modelo de BDD es documental y es una de las más populares.
Desarrollador	Es desarrollada por Apache Software Fundación.	Es desarrollada por RedisLabs.	Es desarrollada por MongoDB, Inc
Lenguaje de Programación	Java	ANSI C	C++
Servidor de Sistemas Operativos	Linux, OS X, Windws	Linux, * BSC, OS Windows	Linux, OS X, Solaris, Windws
Lenguajes de Programación soportados	C++, Java, JavaScript, PHP, Python, Haskell	C, C#, C++, Java, PHP, Python, Erlang, Go, Haskell, Javascript, Perl, Ruby, Scala	C#, Delphi, Haskell, Java, JavaScript, Matlab, Perl, PHP, Python
Conceptos de Usuario	Existe un acceso de los usuarios por objeto.	Existe una lista de control para acceso de los usuarios (ACL)	Existe un acceso para usuarios y roles.

**Fuente:** Adaptado de (Zaforas, Paramio & Gómez, 2017) &(Cachimuel Loyo, 2020)

Posteriormente vamos a saber qué las BDD NOSQL es la adecuada para mi aplicación. En la siguiente tabla 5.

**Tabla 5.- Elección entre Base de datos Cassandra y MongoDB,**

	<b>Cassandra</b>	<b>MongoDB</b>
<b>Caso de Uso</b>	Aplicaciones en tiempo real, comercio electrónico, sistemas de detección de fraude, internet de Las cosas, streaming	Analítica en tiempo real, catálogo de productos, Content Management System, Mobile, IoT almacenamiento de logs, Sistemas de Inventarios.
<b>Almacenamiento de datos</b>	Columnas de ancho flexible, la información se almacena en objetos de BDD, similares a una tabla de una base de datos relacional.	Todos se almacena en documentos.
<b>Replicación</b>	Replicación incorporada, fácil de configurar	Replicación incorporada, pero necesitas algún soporte para configurarlo.
<b>Sintaxis de consultas</b>	Si usas SQL, CQL es fácil de configurar	Requiere aprender un nuevo lenguaje de consultas
<b>Escalabilidad</b>	Ofrece escalabilidad y alta disponibilidad con un mínimo de administración	Es una alternativa si tienes datos no estructurados sin una clara definición.

**Fuente:**(Guerrero, 2018).

### 1.5.- Razones para usar MongoDB

Según (Ramirez, 2018), MongoDB guarda los datos en estructuras parecidas a JSON, para familiarizarse , observamos en la tabla 6, los términos utilizados en MongoDB:

**Tabla 6.-Términos que utilizados en MongoDB.**

<b>Colecciones</b>	Semejante a una tabla en el mundo de bases de datos relacionales y sirven para agrupar documentos.
<b>Documentos</b>	La unidad fundamental en que se guarda información dentro de MongoDB, son estructuras parecidas a un JSON. Semejante a una fila en una tabla de una base de datos relacional.

<b>Drivers</b>	Son las librerías empleadas por nuestras aplicaciones para comunicarse con MongoDB.
----------------	---

*Fuente: (Ramirez, 2018)*

¿Estamos listo para conocer las razones por las cuales recomiendo usar MongoDB?, Observe en la tabla 7.

*Tabla 7.- Razones el uso de MongoDB.*

<b>1.- Modelamiento de datos</b>	Si estás pensando en emplear NodeJS como tu backend, es posible que hayas escuchado hablar de arquitecturas MERN (Mongo Express React Node.js) o MEAN (Mongo Express datos se almacenan en estructuras parecidas a un JSON, esto hace que el flujo de datos dentro de la aplicación no tenga mayores cambios en la estructura de datos. Si no estás usando un stack basado en JavaScript, no hay problema ya todos los drivers de MongoDB vienen con serializadores que convierten las estructuras de datos de tu lenguaje de programación en documentos que MongoDB puede entender.
<b>2. Escalabilidad</b>	Si estás esperando un gran flujo de usuarios es ideal que la base de datos que elijas pueda escalar con la demanda, MongoDB al ser una base de datos distribuida puede escalar no solamente de forma vertical (CPU y RAM) sino que también de forma horizontal (creando más nodos).
<b>3. Flexibilidad</b>	El usar una base de datos basado en documentos y que sea Schema. Less como lo es MongoDB hace que tu base de datos crezca con tu aplicación sin tener que ejecutar scripts que crean campos con valores por defecto cada vez que quieras agregar un campo nuevo en tus registros. Dentro de MongoDB es normal que los documentos dentro de una colección no tengan exactamente los mismos campos.

<p><b>4. Alta disponibilidad</b></p>	<p>Si tienes usuarios en muchas partes del mundo y quieres que las aplicaciones se encuentren distribuidas regionalmente, MongoDB te permite tener clúster distribuidos. Esto mejora la velocidad de consulta al disminuir la latencia que existe entre el cluster de base de datos y el servicio que ejecuta la query. Además de la ventaja adicional de que si una región no se encuentra disponible, las otras pueden mantener la aplicación disponible.</p>
<p><b>5. Transacciones</b></p>	<p>Si vienes del mundo SQL y habías estado prevenido de usar MongoDB debido a que no ofrecía transacciones, te cuento que en su versión 4.0 MongoDB nos trae transacciones ACID (Atomicity, Consistency, Isolation, Durability) entre múltiples documentos.</p>
<p><b>6. Poderosa sintaxis para hacer consultas</b></p>	<p>En MongoDB tenemos múltiples operadores que nos permiten crear consultas poderosas con poco código, además que contamos con las Agregaciones que son un mecanismo que nos permite realizar operaciones entre múltiples colecciones.</p>
<p><b>7. Es de código abierto</b></p>	<p>Esto es una ventaja mayúscula ya que no tienes que pagar licencias para usar MongoDB dentro de tu proyecto de forma comercial, además de que existen muchos proveedores que ofrecen MongoDB como SaaS como es el caso de MongoDB Atlas, MLab entre otros.</p>

*Fuente: (Ramirez, 2018).*

## CAPITULO II

### 2..1. ANÁLISIS COMPARATIVO DE LAS BASES DE DATOS NOSQL

#### 2.1.1 Introducción

En la actualidad existen varias bases de datos NoSQL, como también se pueden encontrar algunas que apuntan a resolver un solo tipo de problemas, de la misma forma hay otras que buscan abarcar mucho más. Para realizar este estudio de comparación entre las bases de datos NoSQL, MongoDB, Redis, Apache Cassandra, CouchDB y Neo4J, presentamos criterios apropiados que permitan distinguir las características únicas y distintas del resto de las bases de datos NoSQL. Además, se presentan criterios específicos que permiten hacer la diferenciación entre ellas que persiguen objetivos comunes.

En base a varios criterios es necesario realizar la aplicación de una encuesta a un determinado grupo de personas, los cuales nos permitirán obtener datos estadísticos que ayudará en la selección de la base de datos más adecuada para el desarrollo de aplicaciones web.

#### Estudio Comparativo de Herramientas

Una correcta comparación respecto a las características más importantes en cuanto a la base de datos NoSQL MongoDB, Redis, Apache Cassandra, CouchDB y Neo4J, es el eje fundamental para seleccionar cuál de las herramientas será la mejor para el desarrollo del aplicativo, y la que genere una mejor solución al problema planteado. Estos puntos deben ser analizados de manera conveniente para obtener resultados positivos.

Los puntos que se toman en cuenta deberán ser valorados por una escala la cual deberá asumir los parámetros que son tomados de acuerdo a la necesidad que se ha planteado, la cual se utilizara, la escala de Likert, es una escala de calificación que se utiliza para cuestionar a varias personas dependiendo del **tipo de escala**, puede utilizar los resultados de las preguntas formuladas.

Acuerdo	Frecuencia
<ul style="list-style-type: none"> <li>• Totalmente de acuerdo</li> <li>• De acuerdo</li> <li>• Indeciso</li> <li>• En desacuerdo</li> <li>• Totalmente en desacuerdo</li> </ul>	<ul style="list-style-type: none"> <li>• Muy frecuentemente</li> <li>• Frecuentemente</li> <li>• Ocasionalmente</li> <li>• Raramente</li> <li>• Nunca</li> </ul>
Importancia	Probabilidad
<ul style="list-style-type: none"> <li>• Muy importante</li> <li>• Importante</li> <li>• Moderadamente importante</li> <li>• De poca importancia</li> <li>• Sin importancia</li> </ul>	<ul style="list-style-type: none"> <li>• Casi siempre verdad</li> <li>• Usualmente verdad</li> <li>• Ocasionalmente verdad</li> <li>• Usualmente no verdad</li> <li>• Casi nunca verdad</li> </ul>

Figura 15.- Tipos de Escala de Likert-

**Fuente:** (QuestionPro, 2021).

Bien para realizar la comparación se utilizará la escala de LIKERT como se ha dicho antes, la cual mide o valora las opciones o ítems que se propone para realizar un trabajo, en dicha escala se mide el criterio de las personas o de la información que se obtiene en una investigación, la cual puede manejar con frecuencia, importancia y conocimiento, todo esto en función de lo que estime conveniente el usuario. Se tomará la escala de Frecuencia, Conocimiento e Importancia. Aplicando una encuesta.

### 2.1.2 Aplicación de la Encuesta.

Es una herramienta para obtener información sobre una población determinada, mediante la elaboración de un cuestionario que se aplica a los individuos que conforman dicha población. La información requerida puede referirse a los atributos de los sujetos que componen esa población, así como también su comportamiento actitudes, opiniones, necesidades, características entre otras. Para contribuir con la investigación de las cinco bases de datos NoSQL, se realizó una encuesta, la misma que contiene preguntas que serán respondidas de manera online en la plataforma “<https://www.questionpro.com/>” y personas que trabajen en empresas de desarrollo de software.

### Población

Corresponde al conjunto de referencia sobre el cual se va a desarrollar la investigación, en este caso se tomó a personal capacitado en el desarrollo de aplicaciones informáticas.

### 2.1.3 Encuesta aplicada

La encuesta se realizó a 12.000, personas entre ellas a personas profesionales que trabajen en

desarrollo de software.

## Cuestionario

### Pregunta 1

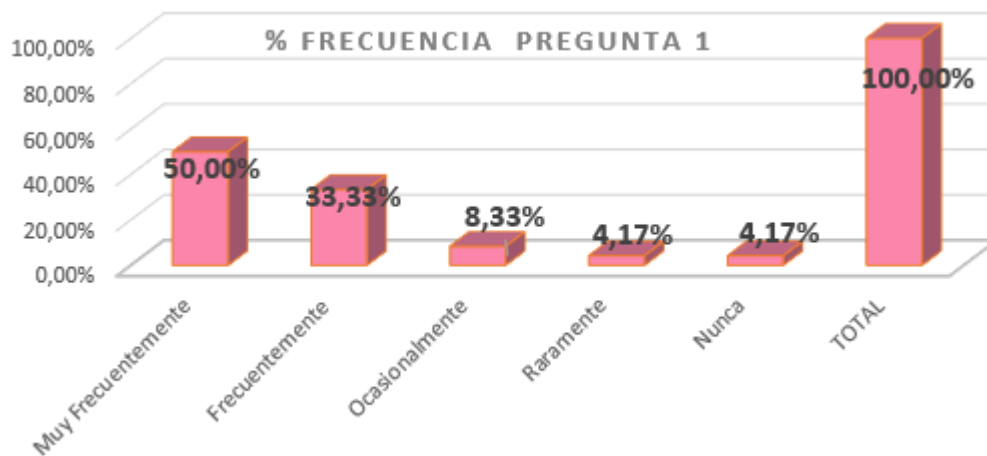
¿Dentro del estudio de las bases de datos NOSQL, con qué frecuencia busca información usted?

*Tabla 8.- Resultados de la Pregunta 1*

RESPUESTAS	PERSONAS	% FRECUENCIA
Muy Frecuentemente	6000	50,00%
Frecuentemente	4000	33,33%
Ocasionalmente	1000	8,33%
Raramente	500	4,17%
Nunca	500	4,17%
<b>TOTAL</b>	<b>12000</b>	<b>100,00%</b>

**Fuente:** Propia.

Representación Grafica



*Figura 16.- Resultados Pregunta 1.*

**Fuente:** Propia.

## Resultados

De las personas realizada la encuesta, un 50,00% de los encuestados buscan información sobre las BDD NOSQL, el 33,33%, es frecuentemente, el 8,33%, es ocasionalmente y otro es el 4,17%, es raramente y nunca.

Los resultados resumen pueden reflejarse en la siguiente tabla

*Tabla 9.- Resultados Resumen de Pregunta 1.*

<b>RESPUESTA</b>	<b>VALOR PORCENTUAL</b>
Muy Frecuentemente	50,00%
Frecuentemente	33,33%
Ocasionalmente	8,33%
Raramente	4,17%
Nunca	4,17%

*Fuente: Propia.*

## Pregunta 2

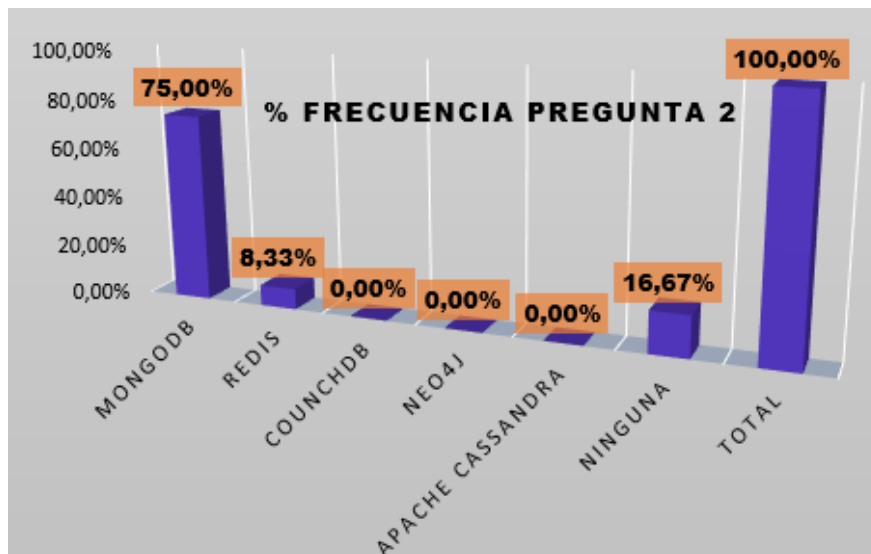
¿Cuál de estas bases de datos NOSQL tiene más información?

*Tabla 10.-Resultados de la Pregunta 2.*

<b>RESPUESTAS</b>	<b>PERSONAS</b>	<b>% FRECUENCIA</b>
MongoDB	9000	75,00%
Redis	1000	8,33%
CouchDB	0	0,00%
Neo4J	0	0,00%
Apache Cassandra	0	0,00%
Ninguna	2000	16,67%
<b>Total</b>	<b>12000</b>	<b>100,00%</b>

*Fuente: Propia.*

## Representación Grafica



*Figura 17.- Resultados de la Pregunta 2.*

*Fuente: Propia.*



### Respuesta

Los resultados de la encuesta, ha demostrado que un 75,00% de las personas tienen información y conocimiento de la BDD MongoDB, el 8,33 % es de Redis, y el otro 16,67% no tiene conocimiento de ninguna, el otro 0%, es de CouchDB, Apache Cassandra y Neo4J.

*Tabla 11.- Resultados de Resumen de la Pregunta 2.*

<b>RESPUESTA</b>	<b>VALOR PORCENTUAL</b>
MongoDB	75,00%
Redis	8,33%
CouchDB	0,00%
Neo4J	0,00%
Apache Cassandra	0,00%
Ninguna	16,67%

*Fuente: Propia.*

### Pregunta 3

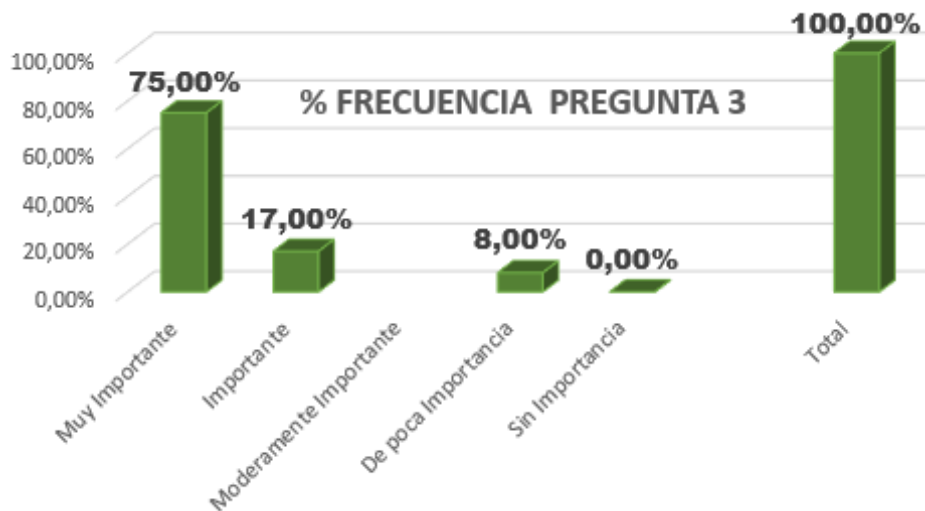
¿Qué importancia tiene las bases de datos NOSQL sea de alto rendimiento, flexibilidad, escala y grandes volúmenes de datos?

*Tabla 12.-Resultados de la Pregunta 3.*

<b>RESPUESTAS</b>	<b>PERSONAS</b>	<b>% FRECUENCIA</b>
Muy Importante	9000	75,00%
Importante	2040	17,00%
Moderadamente Importante		
De poca Importancia	960	8,00%
Sin Importancia	0	0,00%

*Fuente: Propia.*

Representación grafica



*Figura 18.- Resultados de la Pregunta 3.*

*Fuente: Propia.*

## Resultados

De las personas encuestada un 75,00%, nos da conocer, que las BDD NOSQL, es muy importante para usar por su alto rendimiento, flexibilidad y escala de grandes volúmenes de datos, un 17,00%, es importante y el 8,00%, es de poca importancia.

*Tabla 13.-Resultado Resumen de la Pregunta 3.*

RESPUESTAS	VALOR PORCENTUAL
Muy Importante	75,00%
Importante	17,00%
Moderadamente Importante	
De poca importancia	8,00%
Sin importancia	0,00%

*Fuente: Propia.*

## Pregunta 4

**¿Está de Acuerdo usted que la base de datos MongoDB, es excelente en el desarrollo de una aplicación, porque dispone de mucha información y documentación?**

Tabla 14.- Resultados de la Pregunta 4.

RESPUESTA	PERSONAS	% FRECUENCIA
Totalmente de Acuerdo	9400	78,33%
De Acuerdo	1000	8,33%
Indeciso	800	6,67%
En desacuerdo	800	6,67%
Totalmente en Desacuerdo		
<b>Total</b>	<b>12000</b>	<b>100%</b>

Fuente: Propia.

Representación grafica



Figura 19.-Resultados de la Pregunta 4.

Fuente: Propia.

## Resultado

De las personas encuestada un 78,33% considera que la base de datos MongoDB, es totalmente de acuerdo porque dispone de mucha información y documentación, un 8,33%, está de acuerdo, un 6,67%, están en Indecisos y en desacuerdo, dando un total de 12000 personas encuestadas equivale un 100%.

Tabla 15.- Resultados Resumen de la pregunta 4.

RESPUESTA	VALOR PORCENTUAL
Totalmente de Acuerdo	78,33%
De Acuerdo	8,33%
Indeciso	6,67%

En desacuerdo	6,67%
Totalmente en Desacuerdo	

*Fuente: Propia.*

## Pregunta 5

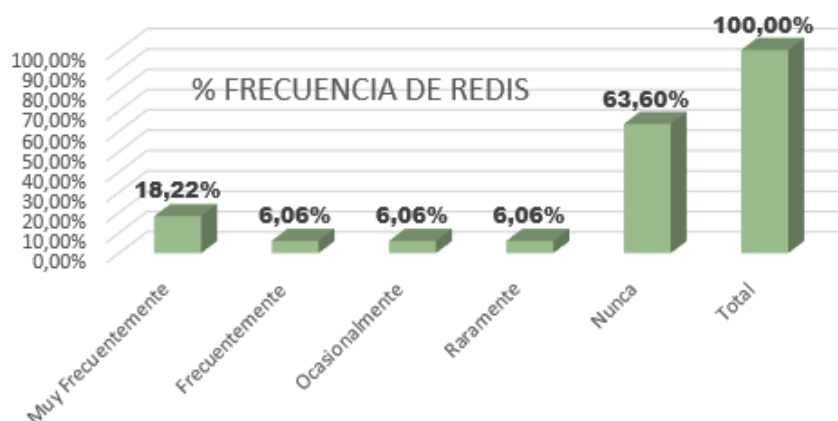
¿Con que frecuencia utiliza usted la BDD Redis en el desarrollo de un sistema?

*Tabla 16.- Resultado de Pregunta 5.*

RESPUESTAS	PERSONAS	% FRECUENCIA
Muy Frecuentemente	2000	18,22%
Frecuentemente	1000	6,06%
Ocasionalmente	1000	6,06%
Raramente	1000	6,06%
Nunca	7000	63,60%
<b>Total</b>	<b>12000</b>	<b>100%</b>

*Fuente: Propia*

Representación grafica



**Figura 20.- Resultados de la Pregunta 5.**

*Fuente: Propia.*

## Resultados

De las personas encuestada un 63,60%, dicen que la base de datos Redis nunca han utilizado en un desarrollo, un 6,06 % han utilizado, raramente, ocasionalmente, frecuentemente han usado y el otro

18,22%, dicen que han usado muy frecuentemente, con total de 12000 personas así dando el 100%.

*Tabla 17.- Resultado Resumen de la Pregunta 5.*

RESPUESTA	VALOR PORCENTUAL
Muy Frecuentemente	18,22%
Frecuentemente	6,06%
Ocasionalmente	6,06%
Raramente	6,06%
Nunca	63,60%

*Fuente: Propia.*

### Pregunta 6

¿Con que frecuencia utiliza la base de datos Apache Cassandra en el desarrollo de un sistema?

*Tabla 18.-Resultado de la Pregunta 6.*

RESPUESTAS	PERSONAS	% FRECUENCIA
Muy Frecuentemente		
Frecuentemente		
Ocasionalmente	2000	18,20%
Raramente	2000	18,20%
Nunca	8000	63,60%
<b>Total</b>	<b>12000</b>	<b>100%</b>

*Fuente: Propia.*

Representación grafica



*Figura 21.- Resultados de la Pregunta 6.*

*Fuente: Propia.*

## Resultado

De las personas encuestadas un 63,60%, que la base de datos Apache Cassandra nunca lo han utilizado para el desarrollo de una aplicación, un 18,20%, la usan Raramente y Ocasionalmente, con un total de 12000 personas que da un total de 100%.

*Tabla 19.-Resultado Resumen de la Pregunta 6.*

<b>RESPUESTAS</b>	<b>VALOR PORCENTUAL</b>
Muy Frecuentemente	
Frecuentemente	
Ocasionalmente	18,20%
Raramente	18,20%
Nunca	63,60%

*Fuente: Propia*

## Pregunta 7

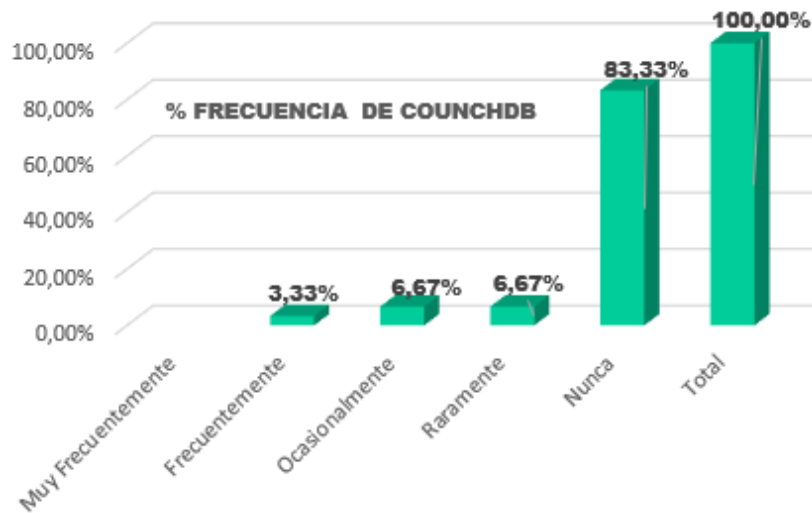
**¿Con que frecuencia utiliza la base de datos CouchDB en el desarrollo de un sistema?**

*Tabla 20.- Resultados de la Pregunta 7.*

<b>RESPUESTA</b>	<b>PERSONAS</b>	<b>% FRECUENCIA</b>
Excelente		
Satisfactorio	400	3,33%
Aceptable	800	6,67%
Poco Satisfactorio	800	6,67%
No lo usado	10000	83,33%

*Fuente: Propia*

## Representación Grafica



*Figura 22.- Resultados de la Pregunta 7.*

*Fuente: Propia*

## Resultado

De las personas encuestada un 83,33%, que la base de datos CouchDB, nunca lo han utilizado para el desarrollo de una aplicación, un 6,67%, la usa, Ocasionalmente y Raramente, un 3,33% es frecuentemente, un total de 12000 personas encuestadas dando un 100%.

*Tabla 21.-Resultados de Resumen de la Pregunta 7.*

RESPUESTA	VALOR PORCENTUAL
Muy Frecuentemente	
Frecuentemente	3,33%
Ocasionalmente	6,67%
Raramente	6,67%
Nunca	83,33%

*Fuente: Propia*

## Pregunta 8

¿Cómo le calificaría la base de datos Neo4J en el desarrollo de un sistema?

Tabla 22.- Resultados de la Pregunta 8.

RESPUESTAS	PERSONAS	% FRECUENCIA
Muy Frecuentemente		
Frecuentemente	2000	16,67%
Ocasionalmente	1000	8,33%
Raramente	9000	75,00%
Nunca		
<b>Total</b>	<b>12000</b>	<b>100%</b>

Fuente: Propia.

Representación Grafica

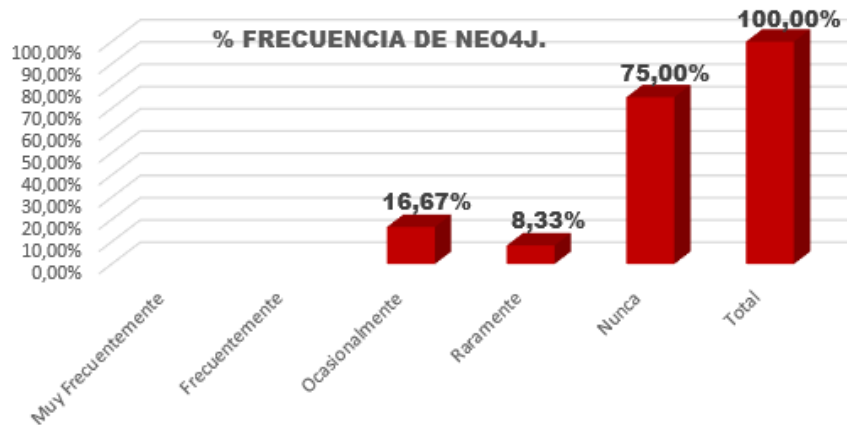


Figura 23.-Resultados de la Pregunta 8.

Fuente: Propia.

### Resultado

De las personas encuestada un 75,00%, que la base de datos Neo4J nunca lo han utilizado, un 8,33 %, es raramente, un 16,67%, es ocasionalmente han utilizado dentro de un sistema, así dando un total de 12000 personas encuestadas, dando un 100%.

Tabla 23.-Resultados Resumen de la Respuesta 8.



<b>RESPUESTA</b>	<b>VALOR PORCENTUAL</b>
Muy Frecuentemente	
Frecuentemente	
Ocasionalmente	16,67%
Raramente	8,33%
Nunca	75,00%

*Fuente: Propia.*

#### **2.1.4 Resultado de la encuesta aplicada a la población.**

Según el resultado de la encuesta realizada, se puede argumentar de acuerdo a lo investigado, dicen que tiene un amplio conocimiento e información sobre las bases de datos NOSQL, y de la misma manera la base datos MongoDB, dispone de bastante documentación e información y es excelente para el desarrollo de un sistema, porque tiene un alto rendimiento, flexibilidad y escala de grandes volúmenes de datos.

## 2.2.- REQUISITOS FUNCIONALES Y NO FUNCIONALES

### 2.2.1.- REQUISITOS FUNCIONALES

Tabla 24.- Tabla de Requisitos Funcionales.

<b>ID REQUISITOS</b>	<b>NOMBRE DEL REQUISITOS</b>	<b>DESCRIPCION DEL REQUISITOS</b>	<b>USUARIO</b>	<b>MEDIO</b>
RF - 001	Ingreso e Registro de Usuarios	El sistema permite el registro e ingreso de los datos de los Usuarios	Cliente Administrador	Pantalla
RF - 002	Registro del Historial de Reservasiones	El Sistema permitirá que el usuario Cliente puede ver el historial de las reservasiones	Cliente	Pantalla
RF - 003	Registro de uso de servicios y generación de pagos	El Sistema permitirá que el usuario Cliente puede registrar el uso de los servicios del hotel y una vez confirmado la reservación puede autogenerar el valor total de uso de servicios del hotel y de las habitaciones.	Cliente	Pantalla
RF - 004	Registro de Reservasiones	<ul style="list-style-type: none"> <li>✓ El sistema permitirá que el usuario Cliente debe registrar (tipo de habitación, descripción y precio)</li> <li>✓ El sistema permitirá que el usuario cliente genera su reservación desde fecha actual y fecha final, el tipo de habitación que más desee.</li> <li>✓ El sistema puede cancelar las reservasiones en caso de que ya desee.</li> <li>✓ El sistema permite que el Usuario Cliente, puede agregar y quitar las reservasiones, dependiendo del estado que se</li> </ul>	Cliente	Pantalla

		encuentre.		
RF- 005	Registro de Cliente	El Sistema permite que el usuario cliente puede registrar sus datos y una vez registrado en pantalla le saldrá un mensaje con el nombre del usuario y su clave que registro en caso de ser cliente nuevo, en casos de ser ya cliente solo debe ingresar con su Nombre de usuario y la contraseña.	Cliente	Pantalla
RF - 006	Ingreso usuario Administrador	✓ El sistema permitirá el ingreso con su contraseña registrada o brindada.	Administrador	Pantalla
RF - 007	Visualizaciones Información del usuario	✓ El sistema permitirá de visualizar los clientes registrados y filtrar la buscar por su identificación, nombre y estado. ✓ El sistema permitirá ver los pagos generados por los clientes.	Administrador	Pantalla
RF - 008	Visualización de búsqueda	El sistema permitirá filtrar la búsqueda de las reservaciones por el estado, el pago realizado por cliente que genero la reservación.	Administrador	Pantalla
RF - 009	Inactivas usos de servicios.	El sistema permitirá que el usuario puede inactivar los usos de servicios	Administrador	Pantalla
RF - 010	Registro de uso de servicios	El sistema permitirá pueda registrar los usos del Servicios del hotel	Administrador	Pantalla

Fuente: Propia.

## 2.2.2 REQUISITOS NO FUNCIONALES

Tabla 25.- Tabla de requisitos no funcionales.

<b>ID REQUISITOS</b>	<b>DESCRIPCION DEL REQUISITOS</b>
RNF - 001	El sistema permitirá que la interfaz es implementada como un sistema web
RNF - 002	Cada usuario que desee reservar, debe ingresar al sistema a la página de inicio de sesión, debe ingresar con su Nombre de usuario y contraseña, si cliente es nuevo debe primero registrarse y llenar los datos para su ingreso.
RNF - 003	El sistema debe tener una interfaz gráfica sencilla, amigable.
RNF - 004	El sistema debe ser desarrollado por bajo software libre y el lenguaje de programación JavaScript, para el prototipo, durante el entorno del desarrollo se ha visto necesario bajo dos partes la Backend que utiliza el Express.js, con métodos de consulta de y servicios para generar un servicio rest. Desde la FrontEnd, bajo eventos en la página web.
RNF -005	El sistema permite utilizar los servicios de red interna.

Fuente: Propia.

**Cálculo de la muestra.** Considerando que la población es finita ya es de 1200.000 personas, se ha aplicado una fórmula, considerando:

- $n$  = es lo que buscamos, tamaño de la muestra.
- $Z$  es el valor de la distribución normal tipificada correspondiente al nivel de confianza elegido, en este caso planteamos la fórmula con un nivel de confianza del 95%, por lo tanto,  $Z = 1.96$ .
- $E$  es el error de la muestra asumido para nuestra muestra y que suele ser inferior al 5% ( $E < 0.05$ ) para el caso de variables categóricas. En el caso de variables métricas,  $E$  es el margen de error que podemos cometer por encima y por debajo de la media, medido en la escala original de la variable, en este caso es del 0,05.
- $p$ , es la probabilidad asociada a la ocurrencia de una categoría concreta de la variable categórica, y  $q = 1-p$ . En el caso de que el valor de  $p$  sea desconocido, se asume la máxima varianza, es decir:  $p = q = 0,5$ .

Siendo la fórmula la siguiente:

$$n = \frac{Z^2 p q N}{E^2 (N - 1) + Z^2 p q}$$

Dónde:

$n = ?$

$N =$  Población (1200)

$Z^2 = 1.96$  (coeficiente de confianza) nivel de confianza 95%

$p = 0.50$  proporcionalidad del evento de estudio

9,604

$q = 0.50$  complemento de  $p$ .

$E = 0.05$  (5%) error absoluto.

Reemplazando la fórmula:

$$n = \frac{1.96^2 * 0.50 * 0.50 * 1200}{0,05^2 (1200 - 1) + 1.96^2 * 0.50 * 0.50} = \frac{1152}{3950} = 4,55 \cong 5$$

Por lo tanto, se trabajará con 1200 elementos; es decir se debe aplicar el instrumento de medición a 1200 personas

## 2.3 DESARROLLO

### 2.3.1 Análisis de Requerimientos

El aplicativo web a desarrollarse se orienta en la creación de un prototipo de Registro de Servicios Hoteleros del Hotel “Sierra Norte”, en donde se pueda organizar y verificar el proceso de almacenamiento y control de los registros, en la misma forma en la utilización de nuevas tecnologías que modernicen los aplicativos webs y optimicen sus procesos en cuanto a funcionalidad, fiabilidad y eficiencia respecta.

#### 2.3.1.1 Roles del Sistema

En la metodología Scrum, el equipo se concentra en construir del aplicativo de calidad, para lo cual el trabajo en equipo es esencial; cada integrante debe comprometerse a una colaboración efectiva por lo que deben tener a su conocimiento sus roles y funciones en el aplicativo.

El equipo de trabajo para el desarrollo de la aplicación es presentado a continuación:

*Tabla 26.- Roles del Sistema.*

<b>Rol</b>	<b>Nombre</b>	<b>Cargo</b>
Jefe de Proyecto SCRUM Master	David Ortiz	Programadora del desarrollo del sistema.
Propietario del Producto Product Owner	Clientes, Administración	Entidad que auspicia el desarrollo del sistema
Externos Interesados Stake Holders	Ing. Diego Trejo	Asesor de trabajo de grado 2

**Fuente:** Propia.

#### 2.3.1.2 Historias de Usuario

Por lo tanto, conforme a las valoraciones que sea podido establecer se presentan las siguientes historias de usuario de acuerdo con los requerimientos siguientes:

*Tabla 27.- Historia de Usuario RSB-001.*

<b>Historias de Usuario</b>	
<b>Código:</b> RSB-001	<b>Usuario:</b> Desarrollador
<b>Nombre:</b> Investigación de herramientas para el desarrollo del prototipo de Registro de Servicios	

Hoteleros.		
<b>Prioridad:</b> alta	<b>Riesgo:</b> Alto	<b>Estimación:</b> 6 horas
<b>Descripción:</b> Búsqueda de tutoriales y manuales para que el desarrollador se capacite para manejar el lenguaje de programación y el software con el cual se proceda a desarrollar el prototipo de Registro del Servicios Hoteleros, de la misma manera se debe investigar las versiones del software para no tener inconvenientes al momento de realizar las conexiones desde el framework de desarrollo hacia las bases de datos.		
<b>Pruebas de aceptación:</b>		
<ul style="list-style-type: none"> <li>✓ Revisión del proceso de capacitación hacia el desarrollador para probar su habilidad en el manejo del lenguaje de programación JavaScript.</li> <li>✓ Observación de las versiones con las cuales son compatibles cada software seleccionado para el desarrollo.</li> <li>✓ Efectuar pruebas de compatibilidad entre las versiones de software instaladas para probar que no exista problemas de conexión.</li> </ul>		

**Fuente:** Propia.

*Tabla 28.- Historias de Usuario RSB-005.*

<b>Historias de Usuario</b>		
<b>Código:</b> RSB-002	<b>Usuario:</b> Desarrollador	
<b>Nombre:</b> Inicio del desarrollo del Prototipo de Registro de Servicios Hoteleros.		
<b>Prioridad:</b> alta	<b>Riesgo:</b> alto	<b>Estimación:</b> 4 horas
<b>Descripción:</b> Descarga del framework y las dependencias necesarias para efectuar el desarrollo del prototipo de Registro de Servicios Hoteleros, una vez instaladas las dependencias se efectúa la creación y configuración del servidor para la ejecución del aplicativo web, como también se realiza la conexión hacia las bases de datos para culminar con la configuración del servidor en node.js.		
<b>Pruebas de aceptación:</b>		
<ul style="list-style-type: none"> <li>✓ Ejecutar el proyecto una vez se haya terminado de realizar la configuración del servidor.</li> <li>✓ Comprobar la conexión de las bases de datos hacia el prototipo Registro de Servicios Hoteleros, mediante mensajes por consola que dirán si existen errores de configuración o conexión.</li> <li>✓ Verificación de la configuración de servidor</li> </ul>		

**Fuente:** Propia.

*Tabla 29.- Historias de Usuario RSB-006.*

Historias de Usuario		
<b>Código:</b> RSB-003	<b>Usuario:</b> Desarrollador	
<b>Nombre:</b> Creación de Registro y Login		
<b>Prioridad:</b> alta	<b>Riesgo:</b> alto	<b>Estimación:</b> 10 horas
<p><b>Descripción:</b></p> <p>Crear un esquema de datos donde especifiquen los atributos necesarios para registrar usuarios, una vez realizado el proceso permite crear las vistas para hacer el llamado al modelo y realizar los registros, como a su vez se debe crear los métodos para autenticación para que puedan habilitar los usuarios registrados, este método será llamado por una dirección de login que se creara para ingresar en la cuenta creada por el usuario.</p> <p>Solicitar las observaciones apropiadas para controlar los errores que den en el momento de realizar el Login y registro de nuevos usuarios.</p>		
<p><b>Pruebas de aceptación:</b></p> <ul style="list-style-type: none"> <li>✓ Controlar que los atributos creados en el modelo de datos cumplan con los atributos de la entidad creada en el modelo de datos en DbSchema.</li> <li>✓ Controlar la dirección especificada para el registro y login sean redireccionadas hacia las vistas apropiadas.</li> <li>✓ Verificar que los métodos de autenticación estén controlando el ingreso de los usuarios registrados.</li> <li>✓ Los usuarios al registrarse deberán ingresar una contraseña que debe ser mayor a cuatro caracteres, en caso de que el usuario solo ingrese cuatro caracteres como contraseña le aparecerá un mensaje pidiéndole que ingrese una contraseña segura.</li> <li>✓ Verificar que los métodos creados cuenten con los atributos y condiciones necesarias para cumplir con la consulta (leer, eliminar), que vaya a realizar el usuario.</li> </ul>		

**Fuente:** Propia.

*Tabla 30.- Historia de Usuario RSB-007.*

Historias de Usuario		
<b>Código:</b> RSB-004	<b>Usuario:</b> Desarrollador	
<b>Nombre:</b> Registro de Clientes		
<b>Prioridad:</b> alta	<b>Riesgo:</b> alto	<b>Estimación:</b> 3 horas
<p><b>Descripción:</b></p> <p>El Administrador, Recepcionista podrá registrar a los clientes con los siguientes datos;</p> <ul style="list-style-type: none"> <li>✓ Nombres y Apellidos,</li> </ul>		



<ul style="list-style-type: none"> <li>✓ CI, Teléfono</li> <li>✓ Email y Dirección</li> </ul> <p>Con la información ingresada tiene acceso el cliente a todos los servicios que ofrece al Hotel.</p>
<p><b>Pruebas de aceptación:</b></p> <ul style="list-style-type: none"> <li>✓ Verificar que al momento de que el cliente ingrese sus datos.</li> <li>✓ Mensaje de error de registro y caracteres.</li> </ul>

**Fuente:** Propia.

*Tabla 31.- Historia de Usuario RSB-008.*

Historias de Usuario		
<b>Código:</b> RSB-05	<b>Usuario:</b> Desarrollador	
<b>Nombre:</b> Creación de vistas principales que tendrá el usuario logeado		
<b>Prioridad:</b> alta	<b>Riesgo:</b> alto	<b>Estimación:</b> 4 horas
<p><b>Descripción:</b></p> <p>El usuario una vez logeado se le presenta como vista principal su perfil con sus datos como nombre, correo y el tiempo cuando creo su cuenta en el prototipo de servicios hoteleros, como también podrá encontrar con varias opciones de registro de servicios hoteleros, uso de servicios hoteleros y generación de pagos totales e Historial de reservaciones.</p>		
<p><b>Pruebas de aceptación:</b></p> <ul style="list-style-type: none"> <li>✓ Verificar que al momento de que el usuario ingrese a su cuenta se muestre su perfil para poder ingresar diferente tipo de información.</li> <li>✓ Verificar que las opciones se redirijan hacia los servicios hoteleros específicos.</li> </ul>		

**Fuente:** Propia.

*Tabla 32.- Historia de Usuario RSB-006.*

Historias de Usuario		
<b>Código:</b> RSB-006	<b>Usuario:</b> Desarrollador, Administrador	
<b>Nombre:</b> Registro de Servicios Hoteleros		
<b>Prioridad:</b> alta	<b>Riesgo:</b> alto	<b>Estimación:</b> 4 horas
<p><b>Descripción:</b> El administrador, Recepcionista podrá registrar los servicios hoteleros de dos categorías</p> <ul style="list-style-type: none"> <li>✓ Habitaciones</li> <li>✓ Otros</li> </ul> <p>El registro de servicios se debe realizar el Tipo de habitaciones, Nombre, características y Precio.</p>		

**Pruebas de aceptación:**

- ✓ Verificar el ingreso de datos de los servicios hoteleros.
- ✓ Mensaje de error.

**Fuente:** Propia.

*Tabla 33.- Historia de Usuario RSB-010.*

<b>Historias de Usuario</b>		
<b>Código:</b> RSB-007	<b>Usuario:</b> Desarrollador, Administrador	
<b>Nombre:</b> Uso de Servicios Hoteleros y generación de pagos		
<b>Prioridad:</b> alta	<b>Riesgo:</b> alto	<b>Estimación:</b> 4 horas
<b>Descripción:</b> El administrador, Recepcionista puede registrar el uso de servicios hoteleros y generación del pago total.		
<b>Pruebas de aceptación:</b>		
<ul style="list-style-type: none"> <li>✓ Verificar el ingreso de datos de los servicios hoteleros.</li> <li>✓ Mensaje de error.</li> <li>✓ Verificación de Registro de Pago total</li> </ul>		

**Fuente:** Propia.

*Tabla 34.- Historial de Usuario RSB-08*

<b>Historias de Usuario</b>		
<b>Código:</b> RSB-08	<b>Usuario:</b> Desarrollador, Administrador	
<b>Nombre:</b> Historial de Reservaciones		
<b>Prioridad:</b> alta	<b>Riesgo:</b> alto	<b>Estimación:</b> 4 horas
<b>Descripción:</b> El administrador, Recepcionista, deberá generar el Historial del Reservaciones		
<b>Pruebas de aceptación:</b>		
<ul style="list-style-type: none"> <li>✓ Verificar el ingreso de datos de los servicios hoteleros.</li> <li>✓ Mensaje de error.</li> <li>✓ Verificación de Registro de Pago total</li> </ul>		

**Fuente:** Propia.

*Tabla 35.- Historial Usuario RSB-09*

<b>Historias de Usuario</b>	
<b>Código:</b> RSB-09	<b>Usuario:</b> Desarrollador, Administrador
<b>Nombre:</b> Ingreso y Despliegue de información de usuarios	

<b>Prioridad:</b> alta	<b>Riesgo:</b> medio	<b>Estimación:</b> 5 horas
<b>Descripción:</b> El usuario en el aso desea ingresar la información personal podrá hacerlo desplegando una vista, donde podrá ingresar sus datos y después será desplegada su información		
<b>Pruebas de aceptación:</b>		
<ul style="list-style-type: none"> <li>✓ Controlar la dirección especificado para el ingreso de información, este adecuadamente definida.</li> <li>✓ Realizar las pruebas necesarias del método de ingreso de información, para asegurar que se almacenen en la base de datos.</li> <li>✓ Revisar las opciones de textos seleccionados, estén concretamente los campos, para que los usuarios puedan seleccionar correctamente.</li> </ul>		

**Fuente:** Propia.

### 2.3.1.3 Product Backlog

A continuación, se detalla una lista de las funcionalidades que tendrá el prototipo de registro de servicios hoteleros

*Tabla 36.- Product Backlog (Pila de Producto).*

<b>Código</b>	<b>Historias de Usuario</b>	<b>Prioridad</b>	<b>Riesgo</b>	<b>Estimación (en horas)</b>
RSB-001	Investigación de herramientas para el desarrollo del prototipo de registro de servicios Hoteleros	alta	alto	6 horas
RSB-002	Inicio del desarrollo del Prototipo de registro de servicios Hoteleros	alta	alto	4 horas
RSB-003	Creación de Registro y Login	alta	alto	10 horas
RSB-004	Registro de Cliente	alta	alto	3 horas
RSB-005	Creación de vistas principales que tendrá el usuario logeado	alta	alto	4 horas
RSB-006	Registro de Servicios Hoteleros	alta	alto	4 horas
RSB-007	Uso de Servicios Hoteleros y generación de pagos	alta	alto	4 horas
RSB-008	Historial de Reservaciones	alta	media	4 horas

RSB-09	Ingreso y Despliegue de información de usuarios	Alto	media	5 horas
--------	---	------	-------	---------

Fuente: Propia.

### 2.3.1.4.- Roles del Develop team

Dentro del desarrollo se ha visto necesario mencionar las funciones que cumple las personas implicadas y describe en la cada una de ellas en la tabla 24.

*Tabla 37.- Roles del Develop team.*

Rol	Descripción
<b>Diseñador</b>	Es el encargado de crear el diseño arquitectónico mediante los requerimientos establecidos y de acuerdo con ello generar prototipos de la arquitectura del sistema web.
<b>Analista</b>	Es el encargado de realizar el análisis respectivo de los requerimientos, estructura del sistema y los diagramas de arquitectura del sistema web.
<b>Desarrollo</b>	Es el encargado de explorar los diferentes ambientes, herramientas y lenguajes de programación en la que el prototipo podría ser desarrollado.
<b>Usuario</b>	Es necesario que el usuario este familiarizado con el sistema para que conozca las funcionalidades y las pueda evaluar.
<b>Testers</b>	Son los encargados de llevar a cabo pruebas planificadas para comprobar el funcionamiento correcto del sistema web y verificar posibles errores y en caso de a verlos se comunican el personal encargado del desarrollo del sistema.

### 2.3.1.5 Creación de los Sprints

Según las Historias de Usuario se proceder a crear los Sprints respectivos para obtener una mejor planificación para el desarrollo del prototipo.

#### 2.3.1.5.1 Sprint 0

- Planificación del Sprint 0

Para continuar con cumplimiento del Sprint 0, se necesita especificar el orden de la investigación y analizar tanto las funcionalidades que tendrá el aplicativo, para iniciar con el desarrollo, de esta manera se realizar los avances ordenados y planificados.

- a) Spring Backlog

**Tabla 38.- Sprint Backlog Sprint 1.**

Historia de Usuario	Fases de Desarrollo	Actividad	Número de Horas
<b>RSB-001</b>	Investigación	Buscar las herramientas necesarias para poder iniciar la fase de desarrollo	6
<b>Eventos</b>	Planificación	Detallar las tareas a realizar en el Sprint actual	4
	Revisión	Revisión de los resultados obtenidos con el Scrum Master	2
	Retrospectiva	Analizar los resultados del Sprint	2
	Total		34

**Fuente:** Propia.

- Revision del Sprint 0

Al terminar el sprint 0 se puede determinar cuáles serían las mejores funcionalidades con las cuales podrá contar el prototipo de registro de servicios hoteleros. Como también se ha hecho una investigación del software apropiado para realizar el modelado y el desarrollo del prototipo. El proceso de tratamiento de los atributos dentro de cada base de datos se ha visto que existen múltiples diferencias en cada una de ellas.

#### *Retrospectiva del Sprint 0*

**Tabla 39.-Retrospectiva Sprint 1.**

<b>RETROSPECTIVA</b>	
<b>Aciertos (¿Qué salió bien del Sprint?)</b>	- La investigación realizada ayudó al equipo de desarrollo a alejar ciertas dudas y encontrar la información necesaria para conocer el funcionamiento de la base datos NOSQL
<b>Errores (¿Qué no salió bien del Sprint?)</b>	- Existió contratiempos en las búsquedas de cracks para el software del modelado de la base de datos y debido a eso se usó más de tiempo de acuerdo a lo determinado.
<b>Mejoras (¿Qué mejoras se implementará?)</b>	- Las fuentes de investigación son verdaderas en algunas cosas las fuentes contienen información ya pasada, y no sirve para el desarrollo del prototipo.

**Fuente:** Propia.

#### **2.3.1.5.2.- Sprint 1**

- Planificación del Sprint 0

Para el cumplimiento del Sprint 0, es necesario especificar un orden de investigación y analizar tanto las librerías necesarias como los registros adecuados que tendrá el prototipo de registro de servicios hoteleros. Como también será necesario codificar métodos para interactuar con las bases de datos, MongoDB Se ha especificado otras tareas adecuadas para el registro de usuarios, y los servicios hoteleros.

a) Spring Backlog

*Tabla 40.- Backlog Sprint 1*

Historia de Usuario	Fases de Desarrollo	Actividad	Número de Horas
<b>RSB-002</b>	Análisis	Según los videos, manuales investigados de debería seleccionar las dependencias que ayude con el desarrollo del prototipo.	2
	Inicio del Desarrollo	Se descarga el framework Express y las dependencias para comenzar con la configuración del servidor node.js y realizar las conexiones hacia la base de datos.	3
<b>RSB-003</b>	Análisis	Especificar el proceso de funcionamiento de registro y login de cada usuario hacia el prototipo de registro de servicios hoteleros., de esta forma podrá controlar a cada usuario con su respectiva cuenta que va utilizar.	5
	Desarrollo	Se desarrollará para el registro y login los adecuados registros para poder acceder a la cuenta de cada usuario, esto debe comprobar que cada método de autenticación esté controlando las redirecciones de los usuarios hacia la vista correcta.	5
	Control	Se desarrollará un método que permita autenticara los usuarios porque el mismo pueda ingresar a su cuenta con las debida seguridad y no exista jaqueo de información o acceso a la cuenta por usuarios no autorizados.	5

<b>RSB-004</b>	Análisis	Analizar los registros necesarios de los clientes para poder tener un prototipo de registro de servicios hoteleros excelente.	2	
	Registro y especificaciones	Se integrará el script necesario para el desarrollo de registros de clientes que tendrá el prototipo de registro de Servicios hotel.	2	
	<b>Eventos</b>	Planificación	Detallar las tareas a realizar en el Sprint actual	5
		Revisión	Revisar los resultados obtenidos del Sprint con el Scrum Master.	3
	Retrospectiva	Analizar los resultados del Sprints.	3	
	<b>Total horas</b>			<b>35</b>

Fuente: Propia

- Revision del Sprint 1

Al terminar el sprint 1, se comprobó que la estimación de tiempo a cada tarea tiene algunas variantes ya que algunas actividades es necesario emplear más tiempo de los previsto antes de culminar, ya que se ha encontrado una cantidad de errores al momento de ejecutar el sistema, y como se tenía la necesidad de verificar el redireccionamiento de los registros para el desplazamiento de usuarios para los diferentes usuarios.

- Retrospectiva del Sprint 1

Tabla 41.- Retrospectiva Sprint 2.

<b>RETROSPECTIVA</b>	
<b>Aciertos (¿Qué salió bien del Sprint?)</b>	- Las investigaciones ayudaron a entender de una manera adecuada como es el funcionamiento de la base de datos, y con ello modelar dentro del software.
<b>Errores (¿Qué no salió bien del Sprint?)</b>	- En su principio no se investigó en la totalidad las base de datos y por ende se presentaron problemas, por ellos se utilizó más tiempo para hacer la investigación más fondo para entender y el trato de atributos dentro de las base de datos.
<b>Mejoras (¿Qué mejoras se implementará?)</b>	- La investigación para cada una de las áreas a utilizarse será analizada debidamente e investigada adecuadamente para que no haya ningún problema en el desarrollo del sistema y uso de tiempos de lo requerido para cada actividad.

Fuente: Propia

### 2.3.1.5.3 Sprint 2

- Planificación del Sprint 2

Para determinar el cumplimiento del sprint 3 es necesario investigar cómo realizar el registro y login de los usuarios, de esta forma podrán actuar con la base de datos y así obtener la autorización correspondiente en sus cuentas respectivas, y podrán acceder a sus vistas principales creadas y manipuladas por los usuarios logeados, de esta manera se realizará un avance ordenado y planificado.

#### a) Spring Backlog

*Tabla 42.- Sprint Backlog Sprint 2.*

Historia de Usuario	Fases de Desarrollo	Actividad	Número de Horas
<b>RSB-005</b>	Análisis	Detallar como se aplicará el proceso de login alas vista principal que tendrá el usuario, una vez que se haya ingresado a sus cuentas.	2
	Creación	Se creará la vista correspondiente donde el usuario podrá acceder a su información, mediante un redireccionamiento a las vistas principales, en donde encontraran una vez que haya realizado el usuario logeado.	3
	Control	Para verificar el funcionamiento se revisara cada botón o opción del menú principal seleccionado hacia las vista correspondientes,	2
<b>RSB-006</b>	Análisis	Analizar los registros necesarios de los servicios hoteleros	5
	Registro y especificaciones	Se integrará el script necesario para el desarrollo de registros de los servicios hoteleros que tendrá el prototipo.	5
<b>Eventos</b>	Planificación	Detallar las tareas a realizar en el Sprint actual	2
	Revisión	Revisar los resultados obtenidos del Sprint con el Scrum Master	2
	Retrospectiva	Analizar los resultados del Sprints.	3
	<b>Total horas</b>		24

Fuente: Propia

- Revision del Sprint 2



Al concluir el sprint 2, se puede observar que el proceso de los registros al aplicativo, tenga una manera dinámica de registro y login ordenada de los usuarios, a su vez se puede ver que se debe tener unas reglas necesarias de autenticación donde permita el usuario tener una seguridad al momento de ingreso de información e ingresar a sus cuentas.

Para poder comprobar y controlar del registro y login, se encuentran codificados adecuadamente y realizo algunas pruebas para la verificación de los mensajes de error que estén funcionando de manera correcta, de esta manera se describe los errores que se controlan dentro del registro y login

- El usuario al registrarse deberá ingresar su contraseña y reconfirmar en caso que no coincida deberá presentar un mensaje de error en pantalla.
- El usuario al registrar la contraseña deberá ingresar un número mayor de cuatro caracteres, para tener mayor seguridad, en caso que desee ingresar mejor a cuatro caracteres la contraseña deberá presentar en pantalla un mensaje de error.
- El usuario una vez registrado podrá acceder a su cuenta, en caso que ingrese una contraseña errónea, no podrá ingresar, se le presentara un mensaje en pantalla de contraseña incorrecta.

Los registros y creación de vistas es un aspecto necesario para el proceso de ingresos de usuarios, ya que con ello se pueden comprobar, que los registros son creados perfectamente, desarrollados y están redirigiéndose a vistas apropiadas.

- Retrospectiva del Sprint 2

*Tabla 43.- Retrospectiva Sprint 2.*

<b>RETROSPECTIVA</b>	
<b>Aciertos (¿Qué salió bien del Sprint?)</b>	- La investigación ayuda a especificar bien los parámetros para el registro y login de los usuarios, es por ello al realizar las pruebas necesarias de registro y login fueron exitosas, ya que los métodos de programación son utilizados son de acuerdo a las especificaciones realizadas e investigadas.
<b>Errores (¿Qué no salió bien del Sprint?)</b>	- En un principio presentaron mensajes de errores de los usuarios sobre el mal ingreso de contraseñas al registrases, pero después de un largo análisis hacia la codificación los errores se resolvieron sin problema.
<b>Mejoras (¿Qué mejoras se implementará?)</b>	- Es necesario reducir el tiempo en cada actividad y verificación que estén cumpliendo los plazos establecidos.

Fuente: Propia.

#### 2.3.1.5.4 Sprint 3

- Planificación del Sprint 3

Para determinar el cumplimiento del sprint 4 es necesario analizar todos los atributos necesarios para especificar los modelos correspondientes que serán almacenados en cada una de las bases de datos NoSQL a utilizarse, como también ir especificando los métodos y promesas que serán importantes para la importación de los modelos hacia las bases de datos.

También será necesario establecer un área donde los usuarios puedan ingresar su información, este proceso será analizado cuidadosamente para que los usuarios autoricen que su información pueda ser ingresada como no ingresada, en caso de ser ingresada esta información podrá ser tratada con los debidos niveles de seguridad establecidos.

a) Spring Backlog

*Tabla 44.- Sprint Backlog Sprint 3.*

<b>Historia de Usuario</b>	<b>Fases de Desarrollo</b>	<b>Actividad</b>	<b>Número de Horas</b>
<b>RSB-007</b>	Análisis	Considerar los métodos necesarios para poder extraer la información desde la base de datos el Uso de los servicios hoteleros y la generación total de pago	3
	Desarrollo	Implementar un método para extraer los datos del cliente y desplegar el valor total de pago.	4
	Control	Verificar que los datos pedidos dentro del método sean desplegados dentro una tabla y controlar que no exista dentro de la consulta.	2
RSB-008	Análisis	Considerar los métodos necesarios para poder extraer la información desde la base de datos del historial de las reservaciones que están aprobadas o rechazadas	3
	Desarrollo	Implementar un método para extraer los datos del historial de las reservaciones y ver si están aprobadas o rechazadas	4
	Control	Verificar que los datos pedidos dentro del método sean desplegados dentro una tabla y controlar que no exista dentro de la consulta.	2
RSB - 009	Análisis	Considerar los atributos del método de la base de datos creada y programarlo en el archivo	3

		correspondiente para crear un esquema y llamarlo dentro de la clase del proyecto en desarrollo	
	Desarrollo	Crear métodos adecuados para ingresar y despegar la información dentro de la vista del prototipo de registro de servicios hoteleros.	5
	Control	Verificar mediante pruebas que los métodos estén cumpliendo las consultad correspondientes hacia la base de datos, y dentro de la vista validar los respectivos campos, para que sean comprensibles a los usuarios.	4
<b>Eventos</b>	Planificación	Detallar las tareas a realizar en el Sprint actual	5
	Revisión	Revisar los resultados obtenidos del Sprint con el Scrum Master	2
	Retrospectiva	Analizar los resultados del Sprints.	3
	<b>Total horas</b>		<b>40</b>

*Fuente: Propia.*

- Revision del Sprint 3

Durante los procesos de desarrollo de los Sprint observaron y analizaron como los usuarios dentro de sus cuentas podían ver que los clientes están registrados dentro del prototipo de registro de servicios hoteleros, así como se creó un método para obtener información mediante una consulta a la base de datos y desplegar los usuarios registrados, y ver la información obtenida de esta manera a su vez se pudo crear un modelo para el ingreso de información para los usuarios, este modelo se obtiene mediante la verificación de datos creado en base de datos, y realizado este procedimiento se creara un método para guardar la información y desplegar en una vista para probar este proceso se realizara lo siguiente.

- Validación de campos de textos, como para el ingreso de correos, fecha, cedula, donde exista el límite y máximo.
- Se ingresó la información requerida dentro de los campos y se envía la petición para almacenar la información ingresada.
- El control de despliegue de la información con para usuarios y como ingreso de información.

En cuanto al uso de servicios hoteleros y generación de pagos se crea un método para obtener la información desde la base de datos con los registros de clientes que están haciendo uso de los

servicios del hotel y los estén disponibles y si el cliente desea generar el pago generara de la misma manera el total de pago de uso de los servicios hoteleros y podrá registrar si es forma efectiva o transferencia, y para probar el proceso si está correctamente de realizar los siguientes procedimientos

- Ingreso información de campos de textos y envió de información para almacenarla.
- Se realiza el control de despliegue de información del pago total por el uso de los servicios hoteleros y registrar si fue de forma efectiva o transferencia, si bajo transferencia se debe registrar el número comprobante de la transferencia.
- En cuanto al historial de Reservaciones se creará un método para obtener la información desde la BDD de los clientes, quien está aprobado o rechazado, para verificar y probar el proceso se realizará los siguientes procedimientos.
- El ingreso de la información requerida en campos de texto y envió de guardar información ingresada.
- Se realizó el control de despliegue de información de historial de reservaciones.

Al término del cumplimiento de ingreso de información de historial de reservaciones, uso de reservaciones y generación de pagos, se resalta establecer un modelo de datos adecuados que cumpla los atributos y parámetros necesarios en cada historia de usuario

- Retrospectiva del Sprint 3

*Tabla 45.- Retrospectiva Sprint 3.*

<b>RETROSPECTIVA</b>	
<b>Aciertos (¿Qué salió bien del Sprint?)</b>	- Mediante consultas puedo obtener información deseada y se puede desplegar en las vistas el historial de reservaciones y generación del pago total del uso de los servicios hoteleros.
<b>Errores (¿Qué no salió bien del Sprint?)</b>	- Por cuestiones de redireccionamiento hubo inconvenientes en el despliegue e información de las vistas y debido a eso se toma más tiempo de lo planificado.
<b>Mejoras (¿Qué mejoras se implementará?)</b>	- Poner más atención a los errores que se muestran en la consola y verificar que los registros que se pidan vayan de acuerdo a las descritas,

Fuente: Propia.

## 2.4 Diseño

Es una fase del desarrollo del software, que se encarga de estructurar la arquitectura del software mediante procesos de calidad para cumplir con los requerimientos del cliente que han sido remarcados en las historias de usuario. Los procesos del diseño permiten a los diseñadores interpretar todos los aspectos necesarios de los sistemas web a construir. (Mero, 2020).

Durante el diseño del software se evalúa la calidad del desarrollo del proyecto y para ello se realizará procedimientos de revisión técnica como los siguientes:

- a) Los diseñadores deben efectuar todos los requerimientos explícitos y acumular los requerimientos implícitos ya que al implementarlos en las guías correspondientes deben ser claros y concisos para aquellas personas que se encarguen de desarrollar el código y los encargados de realizar las pruebas mantengan la integridad del código desarrollado. (Mero, 2020).
- b) Para que el diseño sea efectivo este debe tener una idea íntegra de lo que va a realizar el sistema web y con ello contemplar en la totalidad la funcionalidad, comportamiento y dominio de los datos es decir que debe cumplir con todos los estándares de calidad que espera el cliente. (Mero, 2020).

#### 2.4.1 Arquitectura del Sistema Web

El sistema web se desarrollará en el lenguaje de programación JavaScript, para el prototipo de Registro de servicios hoteleros, y durante el entorno del desarrollo se ha visto necesario el uso de node.js, con el framework Express y los base de datos NOSQL son cinco que fueron seleccionadas: MongoDB, Apache Cassandra, CouchDB, Redis y Neo4J, dentro de las cuales MongoDB es la seleccionada según el estudio y la investigación.

Para entender de la mejor manera el proceso se diseñó una arquitectura que contempla como estará el funcionamiento del prototipo, como puede observar en la siguiente Fig. 24

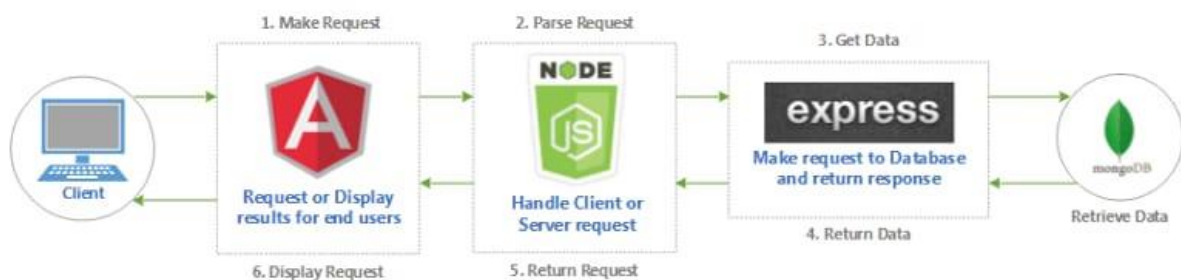


Figura 24.- Arquitectura del Prototipo de Registro de Servicios Hoteleros.

## 2.4.2 Diagrama del Modelado de la Base de datos

En la siguiente Fig. 25. se identifica la elaboración del modelo de datos de MongoDB y las herramientas.

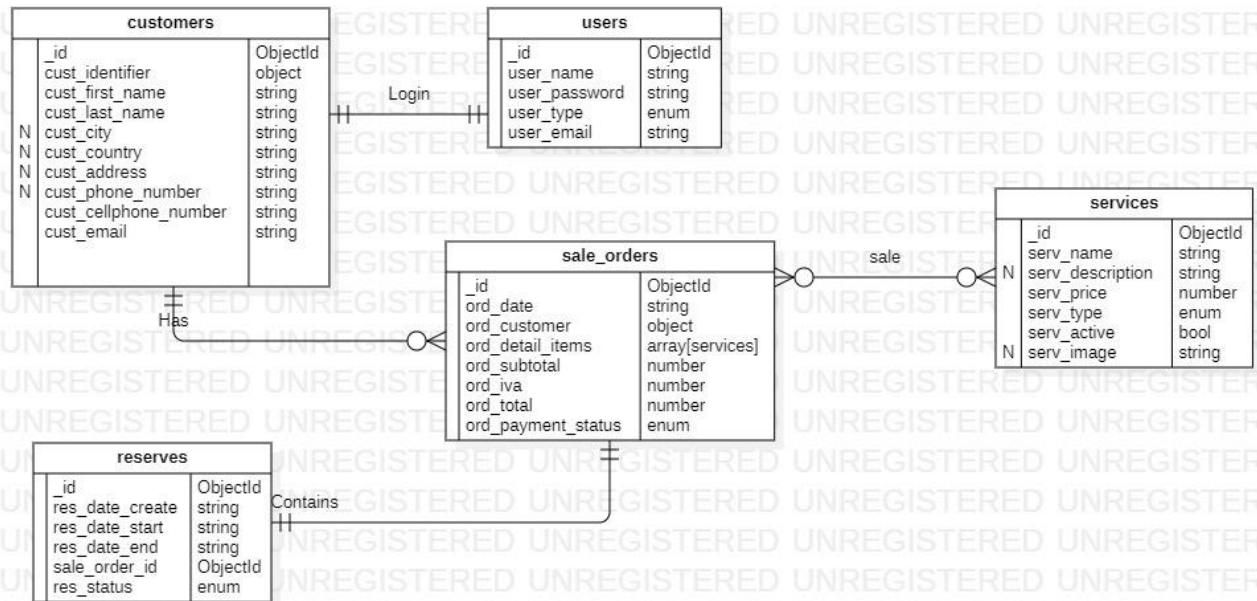


Figura 25.- Diseño de la base de datos.

Fuente: Propia.

## 2.5 Desarrollo del prototipo del Registro de Servicios Hoteleros.

Durante el proceso de desarrollo y una vez terminado cada uno de los sprint se presentan los resultados obtenidos.

### 2.5.1 Sprint 1

Tabla 46.- Sprint 1

Historias de Usuario	Descripción
RSB-001	Investigación de herramientas para el desarrollo del sistema web

Fuente: Propia.

Con las tareas elegidas en el Sprint 1, en el cual podemos ver en la tabla 48 y de acuerdo con ella se pudo establecer las funcionalidades que va a tener el prototipo de registro de servicios hoteleros en donde se concretó que prestará diferentes servicios, de registro de servicios del hotel, uso de los

servicios y generación de pagos, registro de clientes, cada uno tendrá un documento y tabla dentro de la base de datos MongoDB.

Se permaneció con la realización de una investigación donde se definían las herramientas necesarias que se va a utilizar para el desarrollo del sistema web, tal como se puede observar en la tabla 47.

**Tabla 47.- Software especificado para el desarrollo.**

SOFTWARE	DESCRIPCION
	<p>Se lo reconoce como un sistema de bases de datos multiplataforma que se orienta a documentos, por lo que al realizar sus registros con esquemas de datos diferentes y con atributos definidos hace que el mismo no se repita y no haya redundancia en los datos ingresados.</p>
	<p>Es un software de código abierto multiplataforma que posee herramientas adecuadas para desarrollar cualquier tipo de proyecto, este software está basado en el lenguaje de programación ECMAScript que fortalece la sintaxis y características que brinda javascript.</p>
	<p>Es el framework más conocido de node.js, cuenta con varias características que lo hacen un framework robusto, rápido y flexible, además de contar con once middleware poderosos y session-handler que lo hacen único en su clase.</p>
	<p>AngularJS, es un framework de JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página.</p>
<pre data-bbox="217 1581 467 1948"> }, "keywords": [], "author": "", "license": "ISC", "dependencies": {   "bcryptjs": "^2.4.3",   "connect-flash": "^0.1.1",   "errorhandler": "^1.5.1",   "express": "^4.17.1",   "express-handlebars": "^3.1.0",   "express-session": "^1.16.2",   "fs-extra": "^8.1.0",   "md5": "^2.2.1",   "method-override": "^3.0.0",   "mongoose": "^5.6.2",   "morgan": "^1.9.1",   "multer": "^1.4.1",   "passport": "^0.4.0",   "passport-local": "^1.0.0",   "timeago.js": "^4.0.0-beta.2",   "uuid": "^3.3.2" }, </pre>	<p>Librerías o módulos de node.js que se especializan cada uno en cumplir una función respectiva dentro del desarrollo del prototipo de registro de servicios hoteleros.</p> <p>Cada librería será invocada en las clases respectivas para poder usar sus características y poder ir estructurando los servicios que proporcionará los registros de los servicios hoteleros.</p>

Fuente:(Sarasa, 2016)&(Rubio et al., 2020)&(Cachimuel Loyo, 2020).

## 2.5.2 Sprint 2

Tabla 48. Sprint 2.

Historias de Usuario	Descripción
RSB-002	Inicio del desarrollo del sistema web
RSB-003	Creación de Registro y Login
RSB-004	Registro de Clientes

Fuente: Propia.

Con las tareas asignadas en el Sprint 2 se procedió a dar el inicio al desarrollo del prototipo de registro de Servicios hoteleros, de tal forma se realizó como primer paso fue la configuración del servidor y la conexión de node.js con las bases de datos de MongoDB, en la Fig. 26, se muestra los mensajes por consola que muestran que el servidor y la conexión hacia la base de datos se encuentran en perfecto estado.

```
26 //Settings
27 app.set('port',process.env.PORT ||3003);
28 app.set('views',path.join(__dirname,'../views'));
29 app.engine('.hbs',exphbs({
30   defaultLayout:'main',
31   layoutsDir:path.join(app.get('views'),'layouts'),
32   partialsDir:path.join(app.get('views'),'partials'),
33   extname:'.hbs',
34   helpers:require('./helpers')
35 }));
36 app.set('view engine','.hbs');
37
38 //middlewares
39 app.use(morgan('dev'));
40 app.use(multer({dest:path.join(__dirname,'../public/upload/temp')}}).single('image'));
41 app.use(express.urlencoded({extended:false}));
42 app.use(express.json());
43
44 //Middlewares del otro proyecto
45 app.use(express.urlencoded({extended:false}));
46 app.use(methodOverride('_method'));
47 app.use(session({
48   secret:'secret',
49   resave:true,
50   saveUninitialized:true
51 }));
52 app.use(passport.initialize());
53 app.use(passport.session());
54 app.use(flash());
```

Figura 26.- Configuración del Servidor

Fuente: Propia.

En la siguiente Fig. 27, observamos la bienvenida al prototipo de Registro de Servicios Hoteleros



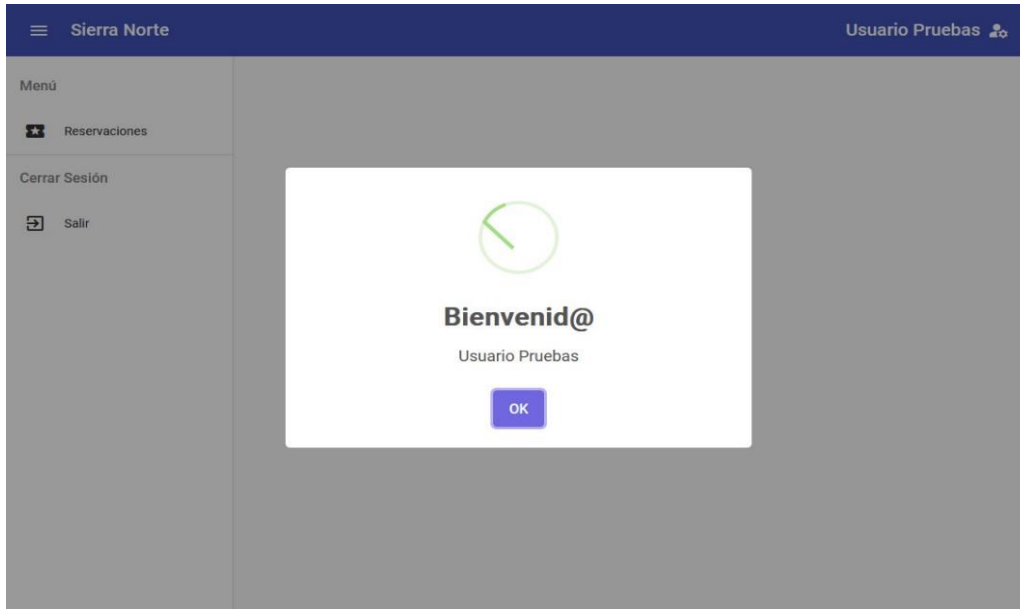
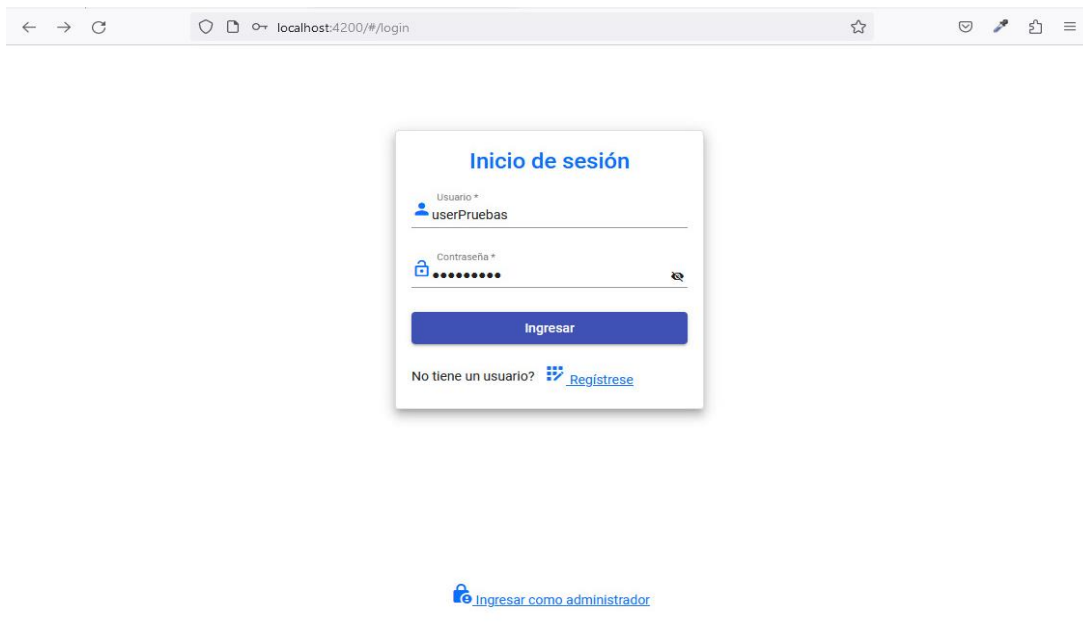


Figura 27.- Bienvenido a Sistema Web.

Fuente: Propia.

En la fig.28, se observa el inicio de Sesión de clientes



Fuente: Propia.

Figura 28.-. Inicio de Sesión de Clientes.

Se observa en la Fig. 29 y 30, el menú de Clientes. donde puede ver, realizar o modificar las reservaciones

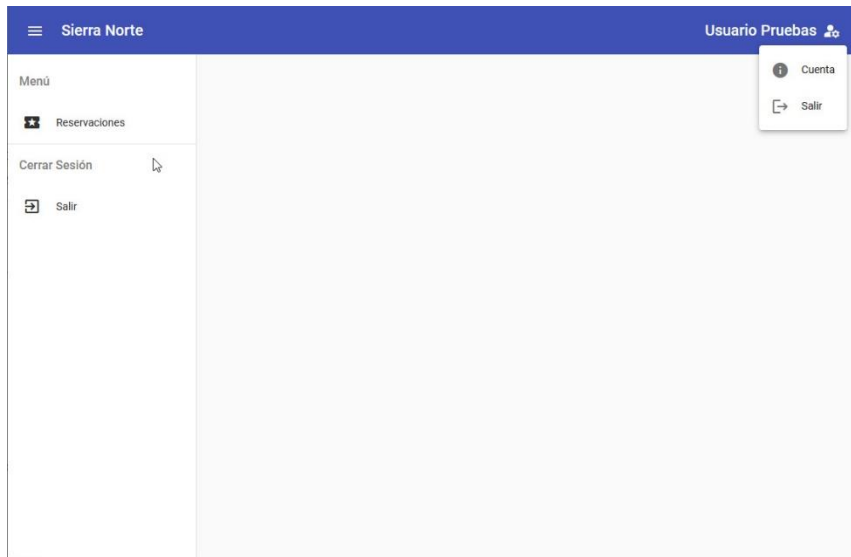


Figura 29.- Menú de clientes

Fuente: Propia.

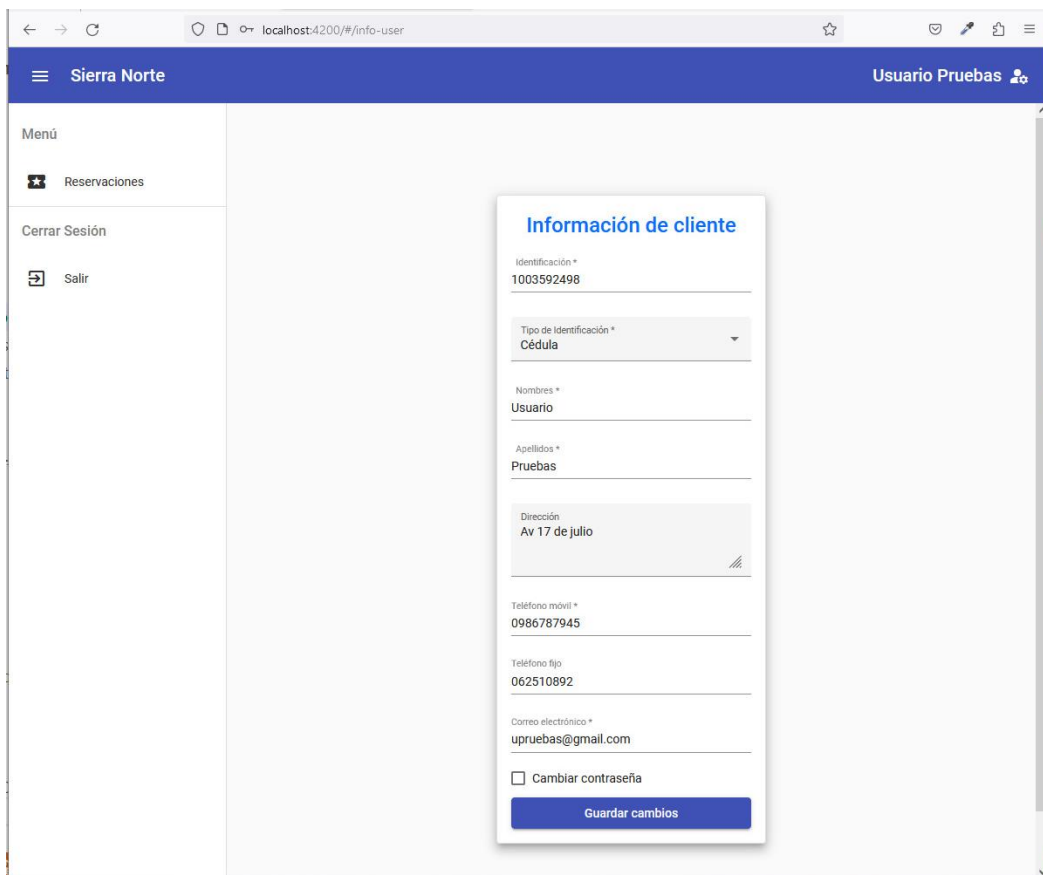
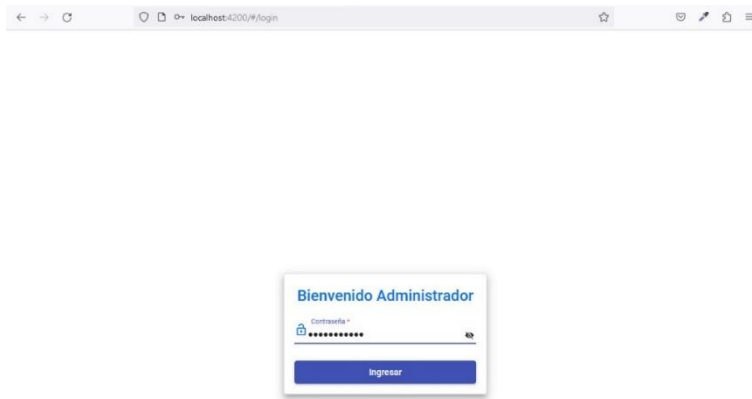


Figura 30.- Modificar Información del Cliente.

Fuente: Propia.

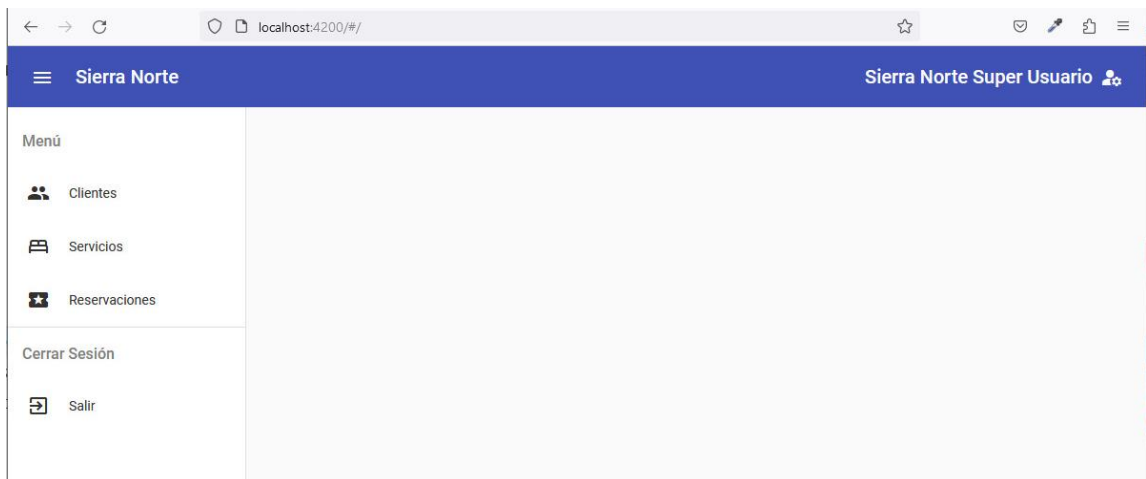
En la Fig. 31.- Ingreso como administrador con su Usuario y Contraseña



**Figura 31..** Ingreso como Administrador.

*Fuente: Propia.*

En la fig. 32.- Se observa el Menú principal cuando se Ingresa como Usuario Administrador



**Figura 32.-** Menú Como Administrador.

*Fuente: Propia.*

### 2.5.3 Sprint 3

**Tabla 49.-** Sprint 3.

Historias de Usuario	Descripción
RSB-005	Creación de Vistas principales que tendrán el usuario logeado.

*Fuente: Propia.*

En la fig. 33, 34 y 35, se observa la vista como usuario administrador, vista como clientes y registro de servicios hoteleros.

The screenshot displays a web application interface. At the top, a blue header bar contains the text 'Sierra Norte' on the left and 'luis fernando gomez' with a user icon on the right. Below the header, the page is divided into three main sections. On the left is a sidebar menu with the title 'Menú' and two items: 'reservas' with a star icon and 'Cerrar Sesión' with a door icon and the text 'salir'. The central section is titled 'informacion de cliente' and contains a form with the following fields: 'Identificación \*' with the value '1002122990', 'tipo de Identificación\*' with a dropdown menu showing 'Cédula', 'nombres \*' with the value 'Luis Fernando', 'Apellidos \*' with the value 'Gomez', 'direccion' with the value 'Ibarra', 'telefono movil \*' with the value '0994711178', and 'telefono fijo' with the value '062510627'.

**Figura 33.- Vista como Usuario Administrador.**

*Fuente: Propia.*

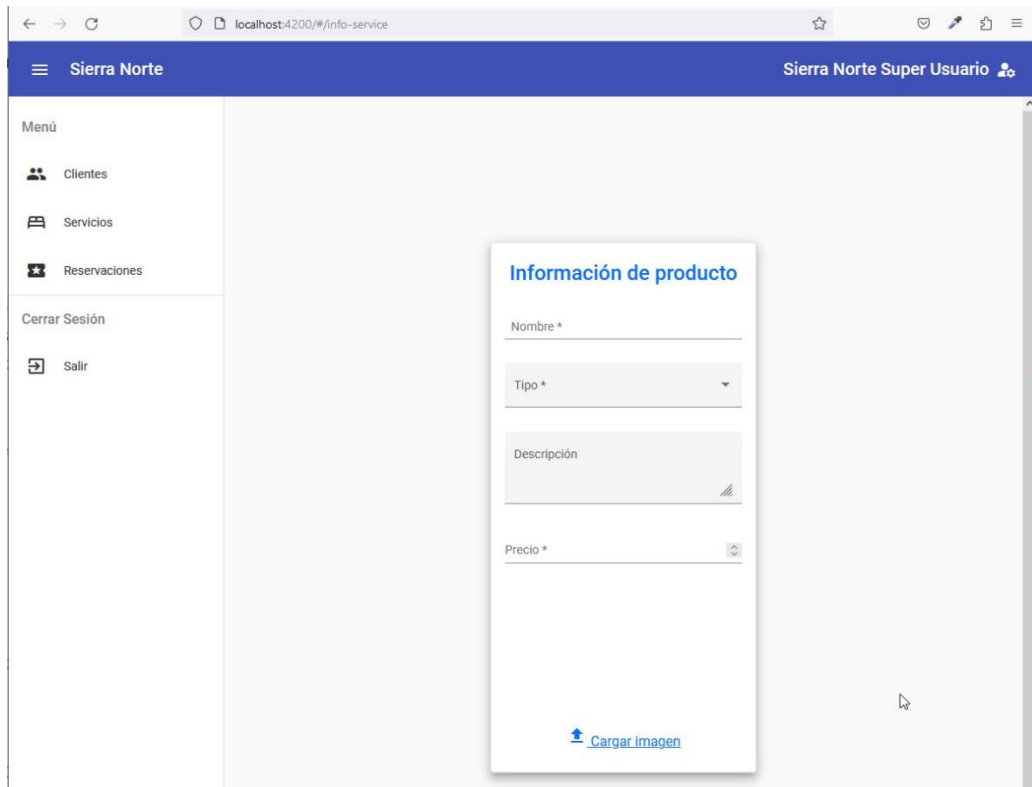


Figura 34., Vista como usuario clientes

Fuente: Propia

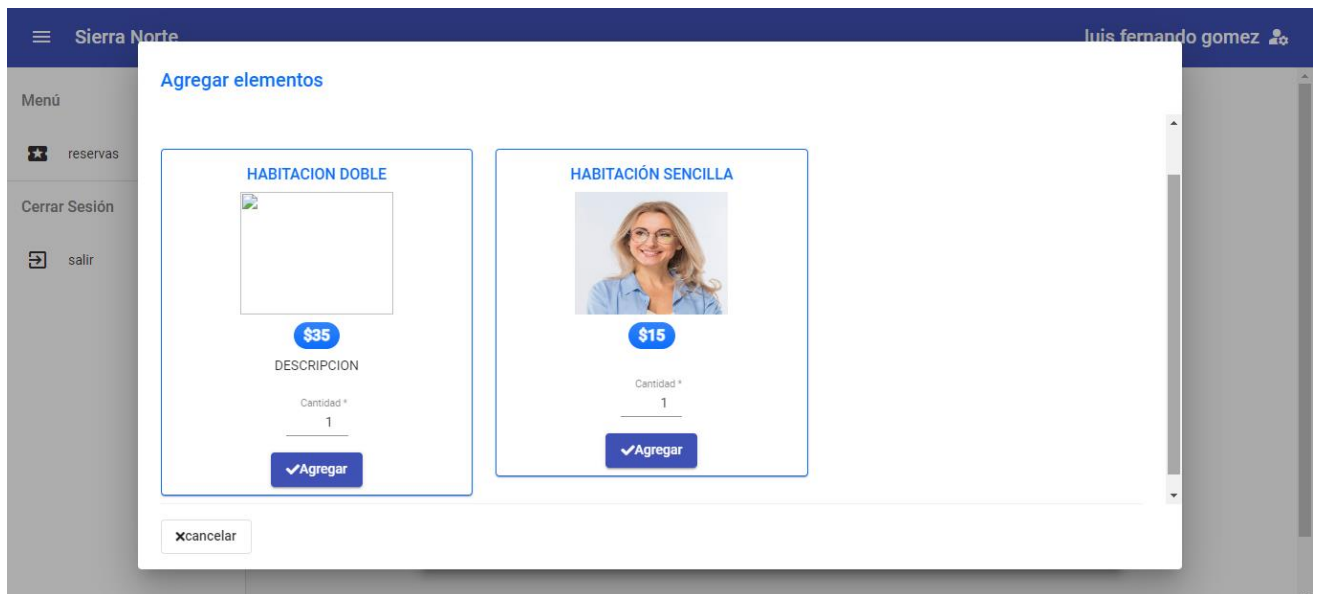


Figura 35.- Registro de los Servicios Hoteleros.

Fuente: Propia.

## 2.5.4 Sprint 4

Tabla 50.- Sprint 4.

Historias de Usuario	Descripción
RSB-007	Uso de Servicios Hoteleros y Generación de Pagos
RSB-008	Historial de Reservaciones
RSB-009	Ingreso y Despliegue de Información de usuarios.

Fuente: Propia.

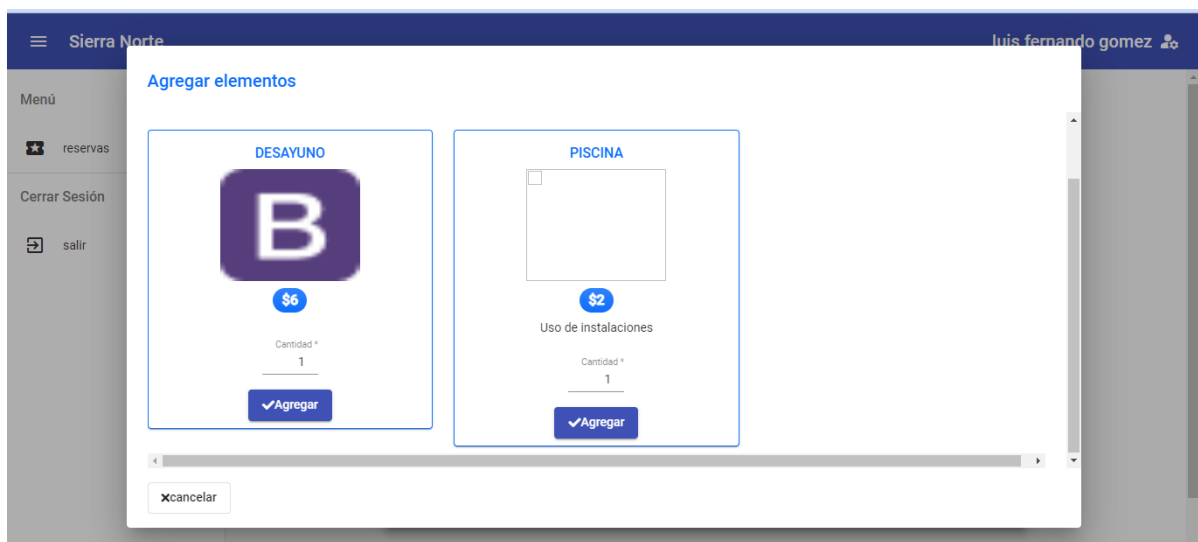


Figura 36.- Uso de Servicios Hoteleros.

Fuente: Propia

En la fig. 36, para generar el pago debe ir a ver reservas y se le genera el valor a pagar te muestra.

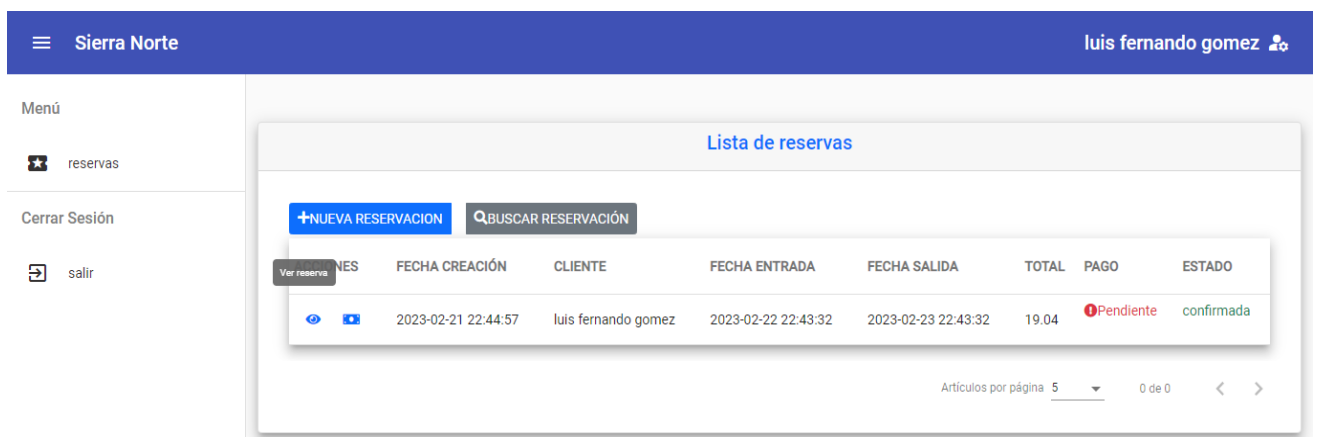


Figura 37.- Inicio para generar pagos.

Fuente: Propia.

Una vez generado el valor a pagar, se observa en la Fig. 37, debe ir al botón Pagar, se genera de qué manera a realizar el pago sea efecto, transferencia, tarjeta, se observa en la Fig. 38

Sierra Norte luis fernando gomez

Menú

- reservas

Cerrar Sesión

- salir

### Información de reserva

confirmada Pendiente de pago

\$Pagar

fecha de inicio: 2/22/2023, 22:43:32      fecha de fin: 2/23/2023, 22:43:32

CANTIDAD	NOMBRE	UNIDAD V.	TOTAL PARCIAL	IVA	TOTAL
1	PISCINA	2	2	0.24	2.24
1	HABITACIÓN SENCILLA	15	15	1.8	16.8

Artículos por página: 5      1 - 2 de 2

TOTAL PARCIAL	17
IVA	2.04
<b>TOTAL</b>	<b>19.04</b>

Figura 38.- Genera al valor a pagar por el cliente.

Fuente: Propia.

Sierra Norte luis fernando gomez

Menú

- reservas

Cerrar Sesión

- salir

--

- depósito
- transferir
- tarjeta de credito
- Tarjeta de Débito
- Efectivo

Nro transacciones \*  
12345566

Nro de cuenta bancaria/ tarjeta / identificación \*  
10003113907

Nombre de titular cuenta/tarjeta \*  
Veronica Angamarca

monto \*  
0

guardar cambios

Figura 39.- Llenado de datos para guardar en pago.

Fuente: Propia

Sierra Norte luis fernando gomez

Menú

- reservas

Cerrar Sesión

- salir

lista de pagos

ACCIONES	FECHA CREACIÓN	MONTO	TIPO	NRO	ENTIDAD FINANCIERA	CUENTA
<a href="#">👁</a>	2023-02-21 22:54:11	19.04	Efectivo	12345566		10003113907

Artículos por página 5 1 - 1 de 1 < >

*Figura 40.- Lista de Pagos.*

*Fuente: Propia*

Sierra Norte luis fernando gomez

Menú

- reservas

Cerrar Sesión

- salir

Lista de reservas

Ver reserva	NES	FECHA CREACIÓN	CLIENTE	FECHA ENTRADA	FECHA SALIDA	TOTAL	PAGO	ESTADO
<a href="#">👁</a>		2023-02-21 22:44:57	luis fernando gomez	2023-02-22 22:43:32	2023-02-23 22:43:32	19.04	🚫 Pendiente	confirmada

Artículos por página 5 1 - 1 de 1 < >

*Figura 41.- Historial de Reservas.*

*Fuente: Propia*



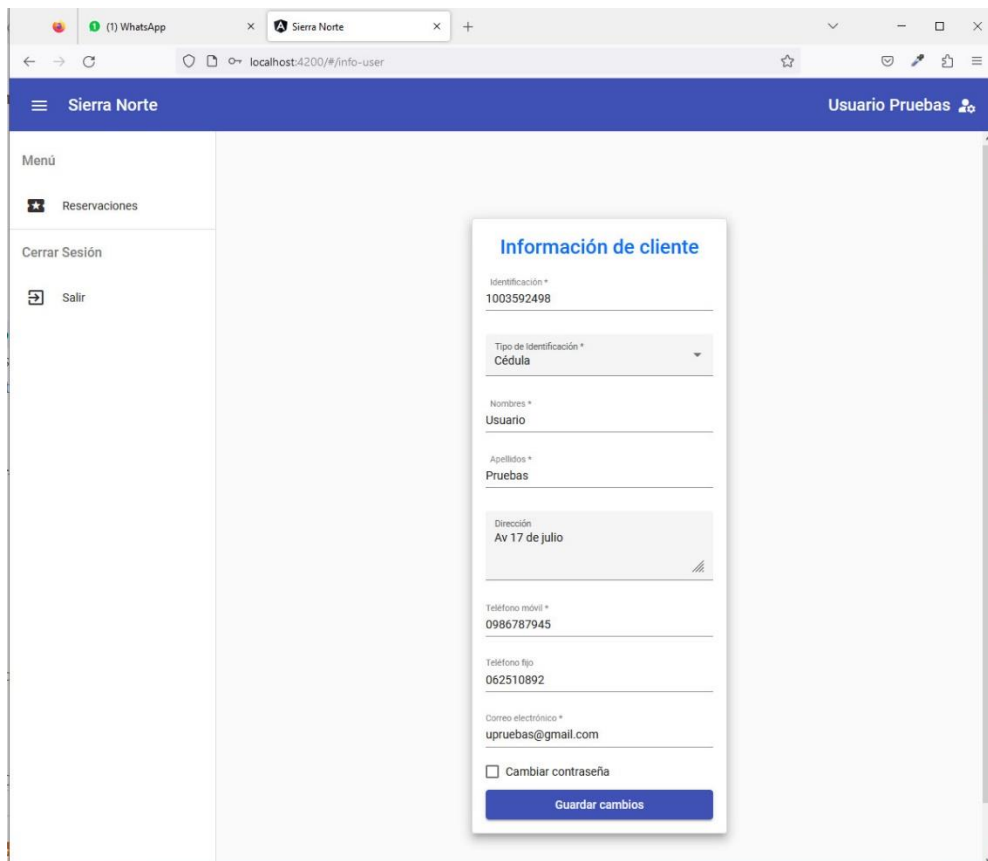
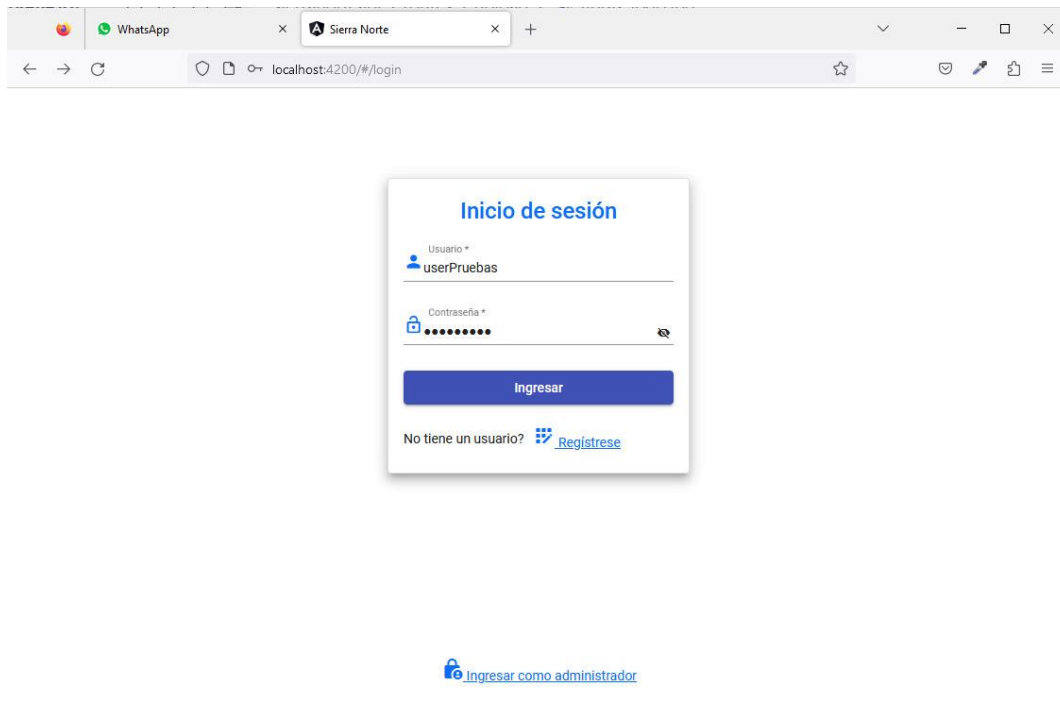


Figura 42.- Ingreso y Despliegue de usuario cliente

Fuente: Propia

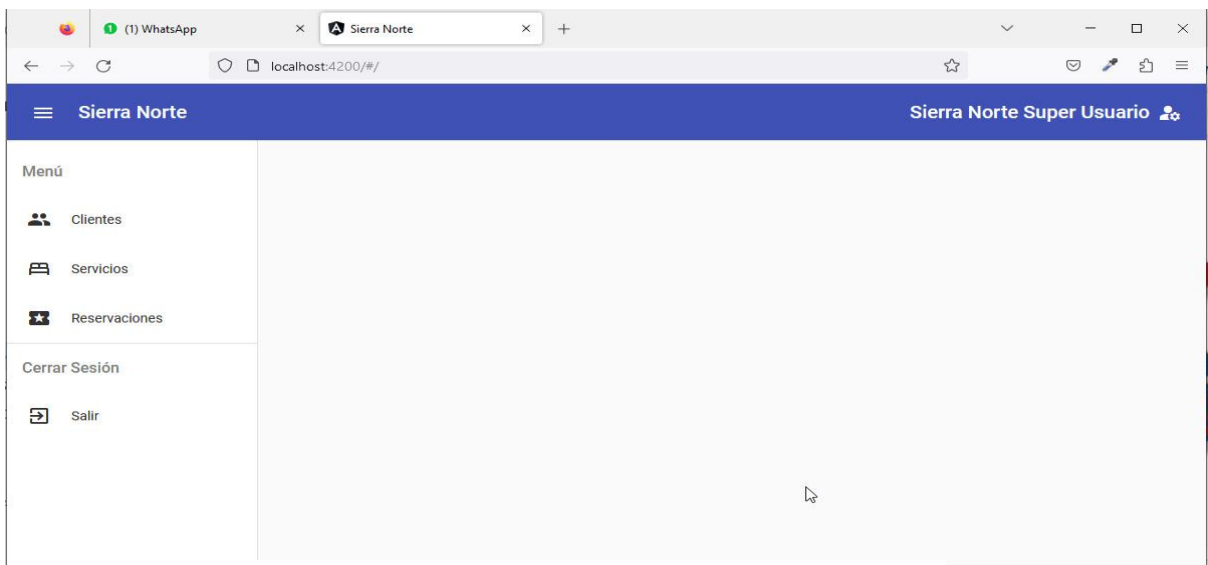
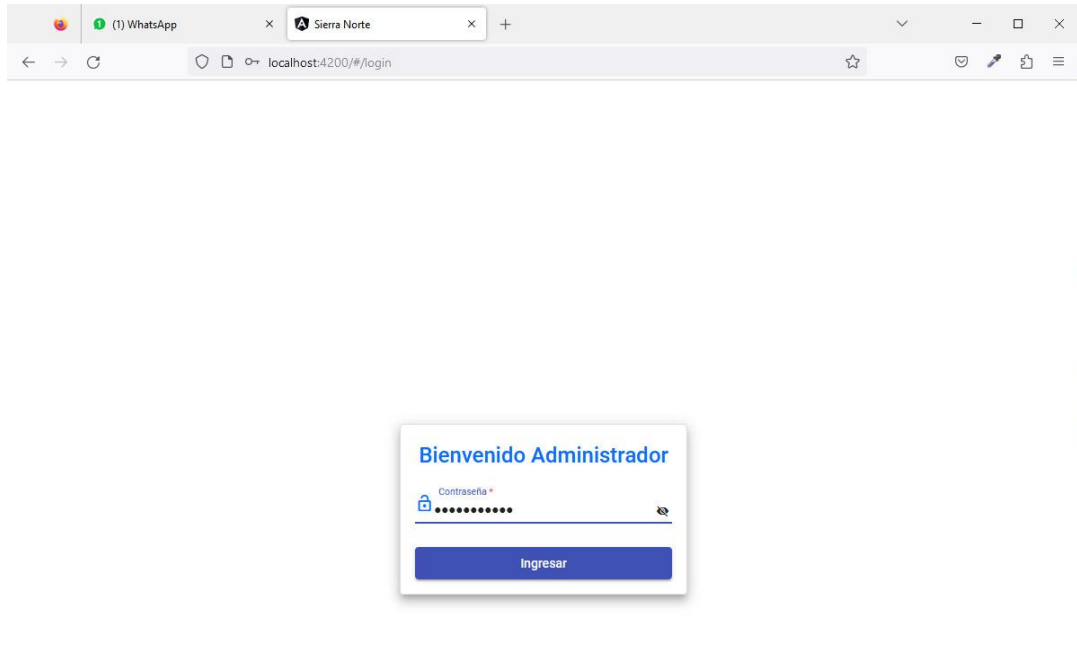


Figura 43.- Ingreso y Despliegue de Administrador.

Fuente: Propia.

## 2.6 Pruebas

Tabla 51.- Prueba 1.

<i>Proceso</i>	<i>Actividad</i>	<i>Porcentaje de Aceptación</i>
<i>Modelado de base de datos</i>	Modelado del base de datos MongoDB	100%
<i>Pruebas de Compatibilidad</i>	Realizar las pruebas de compatibilidad entre las versiones	100%

Tabla 52.- Prueba 2.

<i>Proceso</i>	<i>Actividad</i>	<i>Porcentaje de Aceptación</i>
<i>Configuraciones</i>	Configuración del servidor Nod.js	100%
<i>Utilización de dependencias</i>	Descargar y utilización de dependencias para el cumplimiento de cada clase creada en el prototipo de registro de servicios hoteleros.	100%
<i>Modelo Node.js</i>	Conexión de un modelo donde especifique los atributos necesarios para migrar la información hacia base de datos.	100%
<i>Creación de Métodos</i>	Creación de un método especificación para el almacenamiento	100%
<i>Conexión</i>	Conexión desde Node.js y la base de datos MongoDB,	100%

Fuente: Propia

Tabla 53.- Prueba 3.

<b>Proceso</b>	<b>Actividad</b>	<b>Porcentaje de Aceptación</b>
<i>Modelo Node.js</i>	Creación de un modelo donde especifique los atributos necesarios para migrar la base de datos	100%
<i>Uso de Dependencias</i>	Utilización de dependencias para el despliegue de alertas por pantalla en caso de presentar errores en registro y Login.	100%
<i>Modelo Node.js</i>	Conexión de un modelo donde especifique los atributos necesarios para migrar la información hacia base de datos.	100%
<i>Creación de Métodos</i>	Creación de un método especificación para el almacenamiento	100%
<i>Creación de vistas</i>	Creación de vistas adecuadas para redirigir desde el índice , hasta las vista de registro y Login	100%
<i>Creación de métodos</i>	Creación un método de autenticación para el control de registro y Login	100%
<i>Control de Contraseñas</i>	Definir las cantidades de caracteres necesarios para el registro de usuarios.	100%
<i>Creación de registros</i>	Creación de registros necesarios para mostrar la vista principal del usuario logeado.	100%
<i>Control de registro</i>	Verificar los registros creados y cuenten con autenticación de usuarios, para que exista un control de seguridad adecuada.	100%

Fuente: Propia.

Tabla 54.- Prueba 4.

<b>Proceso</b>	<b>Actividad</b>	<b>Porcentaje de Aceptación</b>
<i>Modelo Node.js</i>	Creación de un modelo donde especifique los atributos necesarios para migrar la base de datos	100%
<i>Uso de Dependencias</i>	Utilización de dependencias para el despliegue de alertas por pantalla en caso de presentar errores en registro y Login.	100%
<i>Modelo Node.js</i>	Conexión de un modelo donde especifique los atributos necesarios para migrar la información hacia base de datos.	100%
<i>Creación de Métodos</i>	Creación de un método especificación para el almacenamiento	100%
<i>Creación de vistas</i>	Creación de vistas adecuadas para redirigir desde el índice , hasta las vista de registro y Login	100%
<i>Creación de métodos</i>	Creación un método de autenticación para el control de registro y Login	80%
<i>Control de Contraseñas</i>	Definir las cantidades de caracteres necesarios para el registro de usuarios.	100%
<i>Creación de registros</i>	Creación de registros necesarios para mostrar la vista principal del usuario logeado.	100%
<i>Control de registro</i>	Verificar los registros creados y cuenten con autenticación de usuarios, para que exista un control de seguridad adecuada.	100%

*Fuente: Propia.*

## CAPITULO III

### VALIDACION DE RESULTADOS

El prototipo de registro de servicios hoteleros aprobado mediante la aplicación de la norma ISO/IEC 25012, la cual se ha elegido, la característica principal para evaluar lo que es la eficiencia, la que se ajusta adecuadamente para realizar la validación de los resultados que se espera obtener en el prototipo de Registro de Servicios Hoteleros conjuntamente con la BDD NoSQL, que se está haciendo uso la base de datos NOSQL MongoDB.

Para verificar la validación de resultados, durante el uso del prototipo de registro de servicios hoteleros, se utiliza el Software JMeter, ya que cuenta con las herramientas necesarias para poder realizar las pruebas, durante las cuales se verifica que los sistemas web desarrollados cuentan con un alto rendimiento y eficiencia al momento de efectuar un número alto de peticiones y respuestas hacia el servidor. (Apache JMeter, 20022)

De esta manera al momento de implementar las pruebas se efectúa una simulación de un ambiente de producción con un alto índice de ayudas por parte de los usuarios hacia los sistemas web.

#### 3.1.- Pruebas de Funcionamiento

Las pruebas de funcionamiento se ha logrado mediante el uso de la herramienta JMeter ,donde se ha logrado obtener la velocidad de respuesta de los registros de usuarios , Clientes, registro servicios hoteleros, uso de servicios y generación de pagos, en el prototipo, de esta manera que durante el proceso de pruebas se ha logrado conseguir diferentes tipos de resultados para las variables que Jmeter, a evalúa los resultados como : la media, mediana, mínimo, máximo, desviación estándar, porcentaje de error, rendimiento y el promedio total en bytes de todas las muestras o registros (avg. Bytes). Es decir que todas las variables antes mencionadas serán las encargadas de dar a conocer los valores de NOSQL, y así logramos determinar que es mejor dentro del manejo de grandes cantidades de información.

A continuación, en la tabla 58, se describen los valores consignados por JMeter durante la fase de prueba con las bases de datos NoSQL haciendo uso MongoDB.

Tabla 55.- Demostración de valores de registro de usuarios

BASE DE DATOS MONGODB									
Muestras	Media	Mínimo	Máximo	Desviación	Error %	Rendimiento (bit/sec)	Recibidos (kb/sec)	Enviados (kb/sec)	Avg Bytes
100,0	12321,0	12093,0	12702,0	142,4	0,0	7,6	56,9	9,2	7663,6
200,0	23653,0	21753,0	23933,0	181,3	0,0	8,1	60,7	9,8	7663,6
300,0	34715,0	32646,0	35084,0	313,1	0,0	8,4	62,5	10,1	7663,6
400,0	30958,0	4002,0	41461,0	15541,2	29,5	9,4	58,2	9,1	7663,6
500,0	27195,0	4000,0	41829,0	17410,5	38,6	11,8	67,1	9,7	7663,6
600,0	23461,0	4000,0	43474,0	18963,5	51,8	13,5	69,2	9,3	7663,6
700,0	20914,0	4000,0	39675,0	17539,0	51,9	17,4	86,9	10,1	7663,6
800,0	18450,0	4000,0	42103,0	17940,0	62,5	18,7	84,8	9,5	7663,6
900,0	16791,0	4000,0	41324,0	17450,1	66,0	21,4	92,9	9,7	7663,6
1000,0	16879,0	4000,0	44556,0	18391,6	68,2	22,2	94,3	9,3	7663,6
<b>TOTAL</b>	<b>225337,0</b>	<b>94494,0</b>	<b>366141,0</b>	<b>123872,7</b>	<b>368,5</b>	<b>138,5</b>	<b>733,5</b>	<b>95,7</b>	<b>76636,0</b>

Fuente: Propia

### 3.2 Evaluación de la velocidad de respuesta de los datos

Con los valores obtenidos y explicados en las tablas 55, se ha logrado extraer las variables independientes con sus respectivos valores cuantificables, para poder demostrar que la base de datos posee la velocidad de respuesta adecuada que permitirá concluir que la base de datos MongoDB, puede manejar grandes cantidades de información de una manera eficiente de acuerdo con la característica escogida de la norma ISO/IEC 25012.

Con lo mencionado anteriormente se procede a dar inicio a la fase de evaluación de las herramientas, en donde se ha ingresado en el sistema web conectado hacia la base de datos MongoDB, un usuario el cual sirve como base para poder efectuar el análisis dentro del software Jmeter, una vez analizado el usuario se procede de la siguiente manera y se da inicio con el ingreso de 100 usuarios y se va aumentando de 100 en 100 hasta llegar a 100 registros de usuarios conforme se ha establecido el incremento gradual de las muestras o registros, es así como en las tablas 55 y 56, se puede identificar los valores alcanzados para las variables dependientes de las muestras o registros, cabe mencionar que dentro de estas tablas se encuentra el valor total de cada una de las variables y en especial se identifica como principal variable de análisis lo que es el rendimiento en donde se obtiene como valor total 138.5 bit/sec, este valor al final servirá para comprobar los valores totales para determinar el estudio.

Tabla 56.- Selección de variables para la evaluación de velocidad de datos con MongoDB.

<b>Base de Datos MongoDB</b>					
Muestras	Desviación Estándar	Rendimiento (bit/sec)	Recibidos (kb/sec)	Enviados (kb/sec)	Error %
100	142,38	7,6	56,91	9,16	0
200	181,25	8,1	60,67	9,77	0
300	313,14	8,4	62,52	10,07	0
400	15541,23	9,4	58,18	9,05	29,5
500	17410,54	11,8	67,09	9,72	38,6
600	18963,45	13,5	69,24	9,28	51,83
700	17539,01	17,4	86,93	10,12	51,86
800	17940,04	18,7	84,79	9,53	62,5
900	17450,09	21,4	92,9	9,67	66
1000	18391,56	22,2	94,29	9,31	68,2
<b>TOTAL</b>	<b>123872,69</b>	<b>138,5</b>	<b>733,52</b>	<b>95,68</b>	<b>368,49</b>

Fuente: Propia.

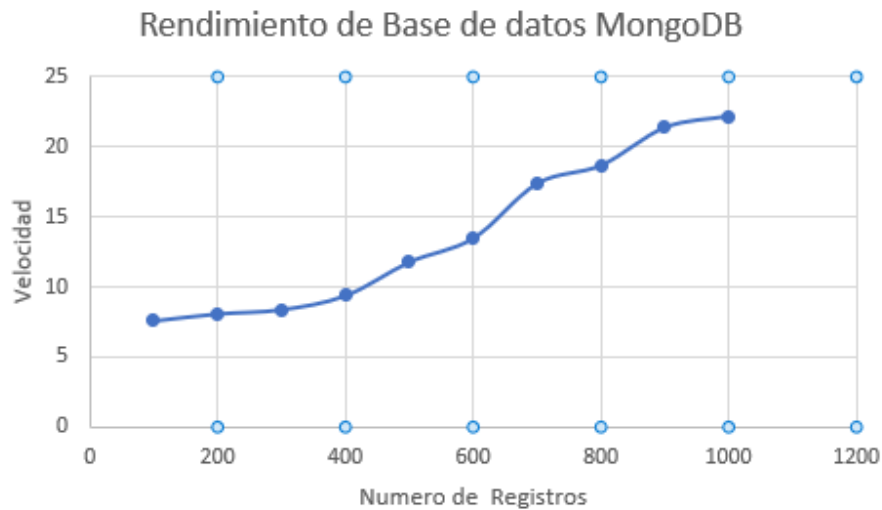
Tabla 57.- Rendimiento de la base de datos MongoDB

<b>Rendimiento de la Base de Datos NoSQL MongoDB (bit/sec)</b>	
<b>Muestras</b>	<b>MongoDB</b>
100	7,6
200	8,1
300	8,4
400	9,4
500	11,8
600	13,5
700	17,4
800	18,7
900	21,4
1000	22,2
<b>TOTAL</b>	<b>138,5</b>

Fuente: Propia



Con la información mostrada en la tabla 57, se puede establecer en la Fig. 44, se puede observar la gráfica, donde se da a conocer las dudas que se presentan en el rendimiento de la base de datos NoSQL y de acuerdo con la misma se puede decir que MongoDB posee una ligera diferencia en el rendimiento.



*Figura 44.- Rendimiento de la base de datos MongoDB.*

*Fuente: Propia*

Para complementar la fase de evaluación de las herramientas se realizó las mismas en los registros de Clientes y uso de servicios y generación de pagos para comprobar el rendimiento obtenido al momento de los registro respectivos, de esta misma manera se procedió a hacer el registros los cuales fueron aumentando progresivamente de 100 en 100 hasta llegar a 1000 registros para la verificación de la velocidad de respuesta en el prototipo de registro de servicios hoteleros conectados hacia las bases de datos NoSQL.

Las pruebas efectuadas hacia el sistema web conectado a la base de datos MongoDB se especifican en la tabla 58, donde se identifica los valores conseguidos para las variables dependientes de la muestras o registros, cabe indicar que dentro de esta tabla se encuentra el valor total de la variable, en especial la variable de análisis rendimiento que posee un valor total de 2237,4 bit/sec.

*Tabla 58.- Selección de variables para la demostración de la velocidad de datos con MongoDB.*

Muestras	Desviación Estándar	Rendimiento (bit/sec)	Recibidos (kb/sec)	Enviados (kb/sec)	Error %
100	4,42	100,4	28,73	50,69	0
200	12,65	186,2	53,28	94,75	0

300	71,02	239,2	68,45	121,72	0
400	69,06	155,8	44,59	79,28	29,21
500	134,67	163	46,65	82,95	36,1
600	245,64	213,3	61,03	108,52	51,75
700	212,89	277,8	79,48	141,33	51,24
800	280,24	249,8	71,49	127,12	58,9
900	363,49	315,2	90,2	160,39	69,1
1000	200,06	336,7	96,34	171,31	68,05
<b>Total</b>	<b>1594,14</b>	<b>2237,4</b>	<b>640,24</b>	<b>1138,06</b>	<b>364,35</b>

*Fuente: Propia*

Para concluir el proceso de pruebas para la sección de registro de los servicios hoteleros, se ha podido realizar un ordenamiento de información el mismo que se da a conocer en la tabla 59 , en donde se puede observar cómo los valores de rendimiento cambian de una forma mínima en cada una de las muestras ingresadas dentro del software JMeter, es de esta manera como se ha podido conseguir un resultado final de la comparación realizada de las tres bases de datos NoSQL y por ende se ha llegado a la conclusión de que la base de datos con mejor rendimiento dentro del análisis realizado es: MongoDB ya que la misma cuenta con un rendimiento (velocidad de respuesta) de 2237,4 bit/sec.

*Tabla 59.- Demostración Rendimiento de la Bases de Datos NOSQL, para los registros.*

Rendimiento de la Base de Datos NoSQL MongoDB (bit/sec)	
<b>Muestras</b>	<b>MongoDB</b>
100	100,4
200	186,2
300	239,2
400	155,8
500	163
600	213,3
700	277,8
800	249,8
900	315,2
1000	336,7
<b>Total</b>	<b>2237,4</b>

*Fuente: Propia*

Con la información descrita en la tabla 59 se puede manifestar que en la Fig.45, existe en la gráfica que demuestra las dudas que se presentan en el rendimiento de la base de datos NoSQL y de acuerdo con la misma se puede decir que MongoDB cuenta con una ligera diferencia de rendimiento.



*Figura 45.- Muestra de evaluación de velocidades en MongoDB con mejor Rendimiento en registros*

*Fuente: Propia*

### 3.3 Análisis e Interpretación de resultados

Considerando los valores descritos en las tablas 56, 57 ,58 ,59 y 60, se ha procedido a seleccionar las variables dependientes a analizar para poder comprobar la característica seleccionada de la norma ISO/IEC 25012 que se refiere a la eficiencia, de esta manera se puede comprobar que el rendimiento no se ve alterado drásticamente cuando se produce un incremento gradual de acuerdo al número de peticiones que se realice al servidor por parte del registro de usuarios ,dentro del prototipo de registro de servicios hoteleros.

Examinando detenidamente y considerando los componentes necesarios para realizar la correcta interpretación de los resultados se ha establecido en la tabla 60, tomar en cuenta las variables de muestra o registros, media, desviación estándar, rendimiento y error porcentual ya que las mismas identifican los valores adecuados para verificar la eficiencia que presenta el sistema web conectado hacia la base de datos MongoDB.

Tabla 60.- Selección de variables para el análisis de los datos en MongoDB,

Muestras	Media	Desviación estandar	Rendimiento	Error %
			(bit/sec)	
100	12321	142,38	7,6	0
200	23653	181,25	8,1	0
300	34715	313,14	8,4	0
400	30958	15541,23	9,4	17,5
500	27195	17410,54	11,8	38,6
600	23461	18963,45	13,5	51,5
700	20914	17539,01	17,4	56,29
800	18450	17940,04	18,7	62,5
900	16791	17450,09	21,4	66
1000	16879	18391,56	22,2	68,2

Fuente: Propia

De acuerdo con la tabla 60, se obtiene la Fig. 46, en la cual se establece una comparativa entre las variables, en donde se confirma que la media da a conocer un tiempo promedio en milisegundos los mismos que van de acuerdo con el incremento que se estableció en la muestra, como también se comprueba como la desviación estándar indica la cantidad de variación existente entre los tiempos de respuesta que se vayan obteniendo conforme la muestra aumente.

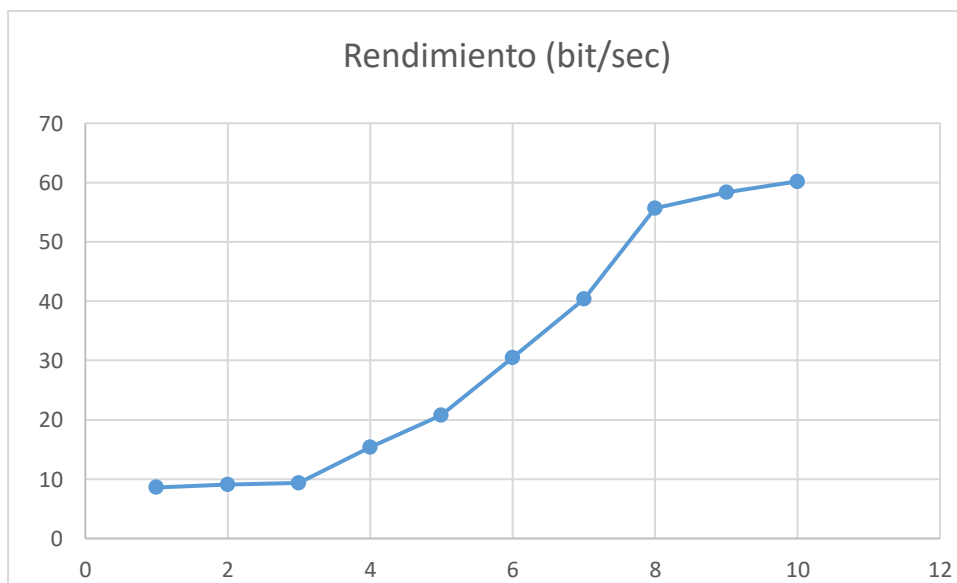
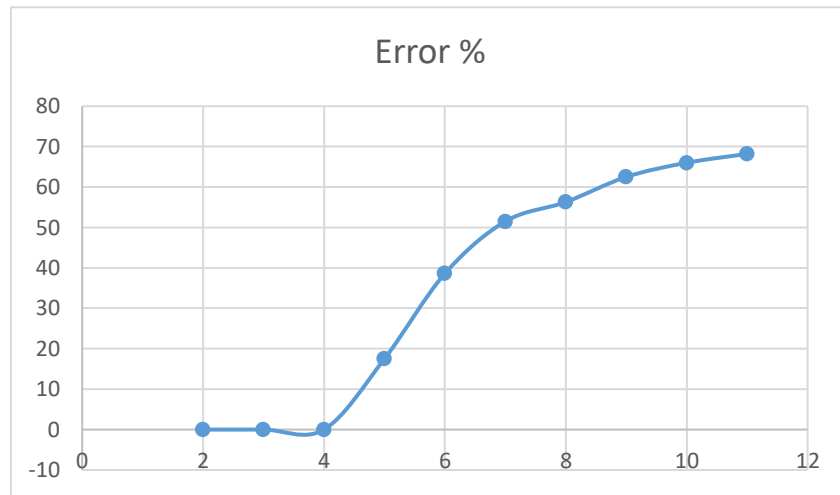


Figura 46.- Comparativa entre las variables Muestras, Media y Desviación Estándar en MongoDB.

*Fuente: Propia*

Con respecto a la tabla 6 es factible establecer una comparativa que se demuestra en la Fig. 52, en donde se puede apreciar el porcentaje de error que se puede encontrar conforme se van efectuando los ingresos de las muestras, como también se identifica el rendimiento que consiste en representar el número de transacciones por minuto que es capaz de soportar el sistema web conforme al incremento de las muestras o registros.



*Figura 47.- Comparación entre las variables Muestras, Rendimiento y Error en MongoDB.*

*Fuente: Propia*

### 3.4 Validación de Resultados

Los datos analizados en las secciones anteriores nos sirvieron para identificar la eficiencia obtenida en sistema web conectada hacia la base de dato MongoDB y mediante ese proceso se demostró es la mejor opción para trabajar con grandes cantidades de información, pero para esta sección de validación ha sido necesario comprobar la relación existente entre los valores y para realizar esta interpretación, se ha utilizado lo que es la regresión lineal con el objetivo de analizar las variables como media, desviación estándar, error porcentual, rendimiento y kb por segundo recibidos y enviados.

Para el análisis de los resultados es necesario verificar el tipo de regresión lineal oportuno a utilizarse para los valores obtenidos de la base de datos, por lo tanto, para probar este punto se ha optado por manejar las pruebas de Shapiro Wilk y para ello se ha utilizado el software estadístico R que contiene herramientas para hacer cálculos estadístico, además posee una gran variedad de técnicas estadísticas como modelos lineales y no lineales, pruebas estadísticas y análisis, clasificación y agrupamiento de

series temporales para el cálculo estadístico de variables de estudio (La Fundación R-project, 2020).

Con la descripción dada anteriormente se pudo ingresar los datos obtenidos de JMeter dentro del análisis que realiza R y se pudo conseguir mediante cálculos los siguientes datos que se encuentran descritos en la tabla 64. En donde se definió que el p-value debe ser menor o igual a 0,05 para que los datos presenten una distribución normal y en caso de que no se cumpla con esta regla los datos presentados no tendrán una distribución normal.

*Tabla 61.- P-Value De Las Tres Bases De Datos.*

Base de datos	Mongo DB	Cumple con p-value
Media	0,08479	Si
Desviación Estándar	0,00896	Si
Error %	0,01183	Si
Rendimiento (bit/sec)	0,2272	Si
Recibidos (kb/sec)	0,2139	Si
Enviados (kb/sec)	0,7196	Si

Con los valores obtenidos en la tabla 61 se puede decir que las variables si cumplen con la propiedad de p-value ya que si tienen una distribución normal, por lo tanto cumplen también con la regla de la normalidad de datos, esto quiere decir que de acuerdo con los valores consignados se puede aplicar las pruebas de correlación de Pearson la misma que ayudará a verificar si existe una covariación lineal entre dos variables cuantitativas, y para designar las variables a analizarse hay que ver su grado relación existen para lo cual es necesario utilizar la siguiente formula:

$$r_{xy} = \frac{\sum xy}{\sqrt{\sum x^2} \sqrt{\sum y^2}}$$

Para verificar el método de correlación de Pearson es necesario aplicar la siguiente regla:

$$1 \geq r_{xy} \geq -1 \text{ donde } r_{xy} = \text{Coeficiente de correccion}$$

Con la regla anterior puntualizada se procede a realizar el análisis entre las variables y se obtiene la figura 53, donde se muestra la matriz generada en RStudio de la comparación de las variables cuantitativas del sistema web conectado a la base de datos MongoDB

```

Muestras      Media      Desviación estandar Rendimiento      Error %
Min.   : 100   Min.   :12321   Min.   : 142.4   Length:11        Min.   : 0.000
1st Qu.: 325   1st Qu.:17272   1st Qu.: 4120.2   Class :character  1st Qu.: 4.375
Median : 550   Median :22188   Median :17430.3   Mode  :character  Median :45.050
Mean   : 550   Mean   :22534   Mean   :12387.3                    Mean   :36.059
3rd Qu.: 775   3rd Qu.:26310   3rd Qu.:17839.8                    3rd Qu.:60.947
Max.   :1000   Max.   :34715   Max.   :18963.5                    Max.   :68.200
NA's   :1     NA's   :1     NA's   :1     NA's   :1
> |

```

Figura 48.- Matriz de Correlación de Pearson de la Base de Datos MongoDB.

Con las matrices de Correlación de Pearson conseguidas de forma general, se procede a verificar la correlación que existe entre dos variables que están fuertemente relacionadas, en este caso para las tres bases de datos sea elegido realizar las pruebas de la siguiente manera: muestra-rendimiento y rendimiento-error.

A continuación, se da a conocer mediante figuras cuales fueron los resultados conseguidos de la correlación entre dos variables fuertemente relacionadas.

Los resultados obtenidos para la base de datos MongoDB se detallan en la Fig. 49, donde se especifica que existen una correlación de 0.99 entre las variables muestra y rendimiento lo que indica que las dos variables tienen una relación fuerte alta positiva, por lo tanto, se determina que a mayor número de usuarios registrados mayor será el rendimiento.

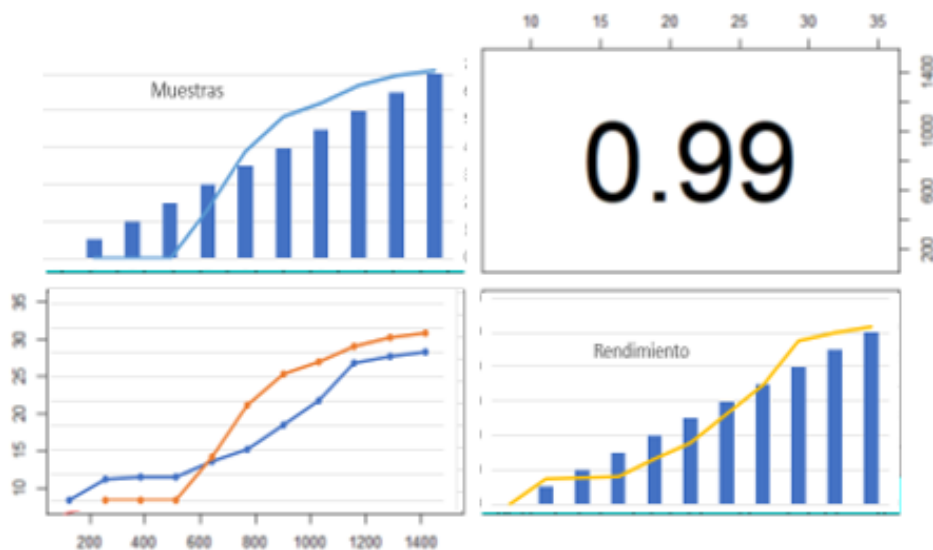


Figura 49.- Resultado de Correlación Muestra-Rendimiento.

Fuente: Propia

Mientras que el resultado obtenido para el análisis de correlación entre las variables rendimiento y

porcentaje de error es de 0.91 el mismo que es definido en la Fig. 55, indica que existe una fuerte relación alta positiva, por lo cual se concluye que a mayor rendimiento obtenido mediante las muestras el porcentaje de error aumentará.



Figura 50.- Resultado de Correlación Rendimiento-Error.

Fuente: Propia

## CONCLUSIONES Y RECOMENDACIONES

### CONCLUSIONES

- De acuerdo con el alcance establecido en el anteproyecto se investigó las funcionalidades con las que cuenta los prototipos de registro de servicios hoteleros y de acuerdo con su complejidad se las repartió en las historias de usuario y con ello se pudo establecer el orden apropiado para desarrollar los prototipos con las funcionalidades establecidas para que los usuarios registrados puedan interactuar entre sí.
- Con los resultados obtenidos de las pruebas realizadas con el software JMeter se pudo conseguir los valores para las variables: muestras, media, desviación estándar, error porcentual y rendimiento con las cuales se pudo realizar el respectivo análisis y aplicando la característica de eficiencia de la Norma ISO/IEC 25012 se pudo con llegar a la conclusión de que MongoDB es la base de datos NoSQL que controla mejor lo que es la velocidad de respuesta de los datos, ya que se pudo comprobar que al momento de registrar los datos MongoDB tuvo un total de 138,5 bit/sec , datos fueron obtenidos y se pudo comprobar mediante estas pruebas a la mejor base de datos NoSQL dentro de este marco de estudio.



- Es recomendable que de acuerdo con la norma ISO/IEC 25012 y a la característica de eficiencia se establece, que el aspecto se debe regir la fase de pruebas para encontrar el resultado de la velocidad de respuesta de los datos.
- El software R, cuenta con herramientas diferentes y nos ayudó a calcular los valores estadísticos necesarios para determinar el tipo de correlación existente entre las variables cuantitativas y de acuerdo con ello se consiguió establecer el grado de correlación de Pearson para el grupo de variables que se encuentran fuertemente relacionadas.

### **RECOMENDACIONES**

- Es recomendable que en la carrera de ingeniería es indispensable que los profesionales y estudiantes tengan conocimiento sobre lo que es la estadística ya que con ella se puede dar a conocer a las personas que los productos que se desarrollan tienen la valía y la calidad adecuada ya que cuentan con un respaldo teórico donde se prueba que mediante métodos, técnicas o modelos estadísticos el software desarrollado es completamente aceptable o tiene cierto grado de falencias a ser corregidos.
- Es recomendable durante el desarrollo del marco teórico es necesario realizar las investigaciones dentro de fuentes bibliográficas que cuenten con investigaciones de alta veracidad, ya que es ahí donde se puede encontrar ya sea artículos o libros científicos que pueden ayudar con las investigaciones.
- Es recomendable verificar el alcance establecido en el anteproyecto para poder realizar un levantamiento de requisitos y poder con ello adecuar las historias de usuario de una forma ordenada y con los respectivos tiempos para que el desarrollo de los prototipos de registro de servicios hoteleros, esté con el tiempo estipulado y con las respectivas funcionalidades establecidas por el producto owner.

## BIBLIOGRAFIA

- Ahmed, M. R., Khatun, A., Ali, A., & Sundaraj, K. (2018). A literature review on NoSQL database for big data processing. *International Journal of Engineering and Technology(UAE)*, 7(2), 1–6. <https://doi.org/10.14419/ijet.v7i2.12113>
- Apache JMeter. (20022). *Apache JMeter - Apache JMeter™*.
- Cachimuel Loyo, B. L. (2020). *Universidad técnica del norte*. Universidad tecnica del Norte.
- Calvo, D. (2017). *Comparativa de Bases de Datos Relacionales con NoSQL*.
- Daniela, D. E. A., & Karina, P. C. V. (2021). *Análisis comparativo de rendimiento en operaciones de escritura para bases de datos SQL y NoSQL*.
- Graph Everywhere. (2018). *Bases de Datos NoSQL \_ Qué son, marcas, tipos y ventajas*. <https://www.grapheverywhere.com/bases-de-datos-nosql-marcas-tipos-ventajas/>
- Guerrero, N. (2018). *¿Cuándo usar Cassandra y Cuando MongoDB\_ – Programa en Línea*. <https://www.programaenlinea.net/cuando-usar-cassandra-cuando-mongodb/#:~:text=Si necesita una base de,fácil y rápido que MongoDB.>
- ISRAEL, A. L. D. (2018). *Pontificia Universidad Católica del Testing Entre Bases de Datos SQL y NoSQL*.
- Jaime, S. H., & Arroyo Lerma, D. (2018). *Testing Entre Bases de Datos SQL y NoSQL*.
- KATHERINE GEOMARA, V. V. (2019). *ANÁLISIS COMPARATIVO DEL DESEMPEÑO COMPUTACIONAL ENTRE DOS GESTORES DE BASES DE DATOS EN CONSULTAS SOBRE LA BASE DE DATOS UNIPROT/SWISS-PROT*. <https://www.grapheverywhere.com/bases-de-datos-nosql-marcas-tipos-ventajas/>
- Kiran Fahd, Samuel Kaspi &, S. V. & R. V. (2016). SQL Versus NoSQL Movement with Big Data Analytics. *International Journal of Information Technology and Computer Science*, 8(12), 59–66. <https://doi.org/10.5815/ijitcs.2016.12.07>
- La Fundación R-project. (2020). *R: El Proyecto R para Computación Estadística*. In *R-project*. <https://www.r-project.org/>
- Lennin, B., & Loyo, C. (2020). *Universidad técnica del norte*.
- lonami.dev. (2020). *Cassandra\_ Introducción*. <https://lonami.dev/blog/mdad/cassandra-introduccion/>
- Maté, A., Peral, J., Trujillo, J., Blanco, C., García-Saiz, D., & Fernández-Medina, E. (2021). Improving security in NoSQL document databases through model-driven modernization. *Knowledge and Information Systems*, 63(8), 2209–2230. <https://doi.org/10.1007/s10115-021-01589-x>
- Morales, S. S. M. (2021). *DE DATOS NO*.
- OTÁLORA LEGUIZAMÓN, J. S. O., & BRAVO MORA, C. A. (2018). *PROTOTIPO DE GESTOR DE MEDICACIÓN UTILIZANDO BASES DE DATOS NoSQL JUAN [UNIVERSIDAD*

DISTRITAL “FRANCISCO JOSÉ DE CALDAS”].

<https://medium.com/@arifwicaksanaa/pengertian-use-case-a7e576e1b6bf>

QuestionPro. (2021). Escala de Likert: Qué es y cómo utilizarla en tus encuestas. In *QuestionPro* (pp. 1–17). [https://www.questionpro.com/blog/es/que-es-la-escala-de-likert-y-como-utilizarla/%0Ahttps://www.questionpro.com/blog/es/que-es-la-escala-de-likert-y-como-utilizarla/#Ejemplos\\_de\\_preguntas\\_de\\_escala\\_de\\_likert%0Ahttps://www.questionpro.com/blog/es/que-es-la](https://www.questionpro.com/blog/es/que-es-la-escala-de-likert-y-como-utilizarla/%0Ahttps://www.questionpro.com/blog/es/que-es-la-escala-de-likert-y-como-utilizarla/#Ejemplos_de_preguntas_de_escala_de_likert%0Ahttps://www.questionpro.com/blog/es/que-es-la)

Ramirez, A. (2018). *¿Qué es MongoDB y razones para usarlo\_ - Platzi*. <https://Platzi.Com/Blog/7-Razones-Mongodb>.

[https://www.googleadservices.com/pagead/aclk?sa=L&ai=DChcSEwinooiyhev7AhUM24YKHVR3BrQYABAAGgJ2dQ&ohost=www.google.com&cid=CAESauD2jhj2X7yDtyxFVFD9uO5f\\_ril2nLxorFQEzcezomN17Zlg2omsxaQFJuj3ZdOzmfDILMKUBDHScffphPhDhX-QN0TRal6QcdjBo58Zg\\_utRgJScUJU\\_Xc2kfc8l67x](https://www.googleadservices.com/pagead/aclk?sa=L&ai=DChcSEwinooiyhev7AhUM24YKHVR3BrQYABAAGgJ2dQ&ohost=www.google.com&cid=CAESauD2jhj2X7yDtyxFVFD9uO5f_ril2nLxorFQEzcezomN17Zlg2omsxaQFJuj3ZdOzmfDILMKUBDHScffphPhDhX-QN0TRal6QcdjBo58Zg_utRgJScUJU_Xc2kfc8l67x)

Raquel Cáceres, Nélica; Tolada, Ana Carolina; Pérez, R. D. (2018). Benchmarking de Bases de Datos NoSQL para el almacenamiento de Modelos Semánticos. *Workshop de Investigadores En Ciencias de La Computación*, 238–241.

[http://sedici.unlp.edu.ar/bitstream/handle/10915/67263/Documento\\_completo.pdf-PDFA.pdf?sequence=1](http://sedici.unlp.edu.ar/bitstream/handle/10915/67263/Documento_completo.pdf-PDFA.pdf?sequence=1)

Rubio, F., . P. V., & Reyes Ch, R. P. (2020). NoSQL contra SQL en la Administración de datos masivos: Un estudio empírico. *KnE Engineering*, 2020, 40–49.

<https://doi.org/10.18502/keg.v5i1.5917>

Sarasa, A. (2016). *Introducción a las bases de datos NoSQL usando MongoDB*.

Shuib Basri, Rohiza Ahmad, Junzu Watada, Aparicio, A. A. I. y M. T. G. (2018). *Modelo de de esquema automático para bases de de de documentos NoSQL*.

Tejero Gómez, E. (2022). *Introducción a las bases de datos NoSQL. Sistemas de bases de datos orientados a grafos*.

Zuleta Sevilla, N. E. (2018). *“ESTUDIO DE LA INTEGRACIÓN DEL LENGUAJE DE PROGRAMACIÓN PHP CON LA BASE DE DATOS NOSQL APACHE COUCHDB*.