

**UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN**

**Facultad de Ingeniería**

Escuela Profesional de Ingeniería en Informática y Sistemas

**ANÁLISIS COMPARATIVO DEL DESEMPEÑO DE LOS  
SISTEMAS GESTORES DE BASES DE DATOS:  
POSTGRESQL Y MYSQL CON LOS DATOS  
ACADÉMICOS DE LA UNIVERSIDAD  
NACIONAL JORGE BASADRE  
GROHMANN – 2019**

**TESIS**

Presentada por:

Bach. Milca Saray Gines Colana

Para optar el Título Profesional de:

**INGENIERO EN INFORMÁTICA Y SISTEMAS**

TACNA – PERÚ

2022

# UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN

## FACULTAD DE INGENIERÍA

### JURADO CALIFICADOR Y CALIFICACIÓN DE LA SUSTENTACIÓN

#### DE TESIS

TESIS N° \_\_\_\_\_

TÍTULO PROFESIONAL DE:

Ingeniero en Informática y Sistemas

La Secretaría Académica de la Facultad de Ingeniería, por Resolución de Facultad N° 06966-2022-FAIN/UNJBG, designó jurado para la sustentación oral de la Tesis titulada: "ANÁLISIS COMPARATIVO DEL DESEMPEÑO DE LOS SISTEMAS GESTORES DE BASE DE DATOS: POSTGRESQL Y MYSQL CON LOS DATOS ACADÉMICOS DE LA UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN – 2019 “.

#### El mismo que está conformado por:

Presidente : Mag. Gianfranco Alexey Málaga Tejada

Secretario : Dr. Edwin Antonio Hinojosa Ramos

Vocal : Dr. Erbert Francisco Osco Mamani

Para calificar la sustentación virtual de la Tesis en acto público, el día 09 de febrero de 2022. Presentado por la Bachiller Milca Saray Gines Colana, de la Escuela Profesional de Ingeniería en Informática y Sistemas.

El Jurado Calificador en forma secreta e individual emitió su opinión sobre el tema de la tesis expuesta y procedió a obtener el promedio que arrojó el calificativo de aprobado con la nota de catorce (14).

Para ratificar lo detallado firman:



Mag. Gianfranco Alexey Málaga Tejada  
Presidente







Dr. Edwin Antonio Hinojosa Ramos  
Secretario



Dr. Erbert Francisco Osco Mamani  
Vocal

**UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN**  
**FACULTAD DE INGENIERÍA**  
**ESCUELA PROFESIONAL DE INGENIERÍA EN INFORMÁTICA Y SISTEMAS**  
  
**“ANÁLISIS COMPARATIVO DEL DESEMPEÑO DE LOS SISTEMAS GESTORES**  
**DE BASE DE DATOS: POSTGRESQL Y MYSQL CON LOS DATOS**  
**ACADÉMICOS DE LA UNIVERSIDAD NACIONAL**  
**JORGE BASADRE GROHMANN – 2019”**

TESIS PRESENTADA Y APROBADA EL 09 DE FEBRERO DE 2022  
ESTANDO EL JURADO CALIFICADOR INTEGRADO POR:

Presidente	:	 _____ Mag. Gianfranco Alexey Málaga Tejada
Secretario	:	 _____ Dr. Edwin Antonio Hinojosa Ramos
Vocal	:	 _____ Dr. Erbert Francisco Osco Mamani
Asesor	:	 _____ Dr. Edgar Aurelio Taya Acosta

## **DEDICATORIA**

*Dedico esta tesis a mis padres y a  
mis hermanas, quienes son el  
incentivo de mi vida y que con su  
apoyo y comprensión incondicional  
estuvieron siempre a mi lado y se  
encargaron de mi formación tanto  
profesional como personal.*

## **AGRADECIMIENTOS**

*A Dios, mis padres y mis hermanas por ser mi fortaleza y mi refugio en el trayecto de mi vida.*

*Mi gratitud también, a mi casa de estudios, la Universidad Nacional Jorge Basadre Grohmann, a la Escuela Profesional de Ingeniería de Informática y Sistemas, al asesor de mi tesis, Dr. Edgar Taya Acosta y a cada docente quienes con su apoyo y enseñanzas constituyen la base de mi vida profesional.*

*Finalmente, al personal administrativo de la Dirección de Actividades y Servicios Académicos de la Universidad Nacional Jorge Basadre Grohmann, por brindarme el apoyo para la recolección de datos.*

## CONTENIDO

DEDICATORIA	iv
AGRADECIMIENTOS	v
CONTENIDO	vi
ÍNDICE DE TABLAS	viii
ÍNDICE DE FIGURAS	ix
RESUMEN	1
ABSTRACT	3
INTRODUCCIÓN	5
CAPÍTULO I PLANTEAMIENTO DEL PROBLEMA	7
1.1. DESCRIPCIÓN DEL PROBLEMA	7
1.1.1. Antecedentes del problema	9
1.1.2. Problemática de la investigación	13
1.2. FORMULACIÓN DEL PROBLEMA	14
1.3. JUSTIFICACIÓN E IMPORTANCIA	15
1.4. ALCANCES Y LIMITACIONES	16
1.5. OBJETIVOS	17
1.5.1. Objetivo general	17
1.5.2. Objetivos específicos	17
1.6. HIPÓTESIS	18
1.6.1. Hipótesis general	18
1.6.2. Subhipótesis	18
CAPÍTULO II MARCO TEÓRICO	20
2.1. ANTECEDENTES DEL ESTUDIO	20
2.2. BASES TEÓRICAS	24
2.2.1. Historia de las bases de datos	24
2.2.2. Base de datos	24
2.2.4. Modelo relacional	28

2.2.5. Niveles de abstracción de base de datos	28
2.2.6. Lenguajes de base de datos	29
2.3. DEFINICIÓN DE TÉRMINOS	30
2.3.1. Desempeño	30
2.3.2. Pruebas de desempeño	30
2.3.3. Indicadores de desempeño	31
2.3.4. Tiempo de respuesta	32
2.3.5. Uso de memoria	32
2.3.6. Memoria de acceso aleatorio	32
2.3.7. JMeter	32
2.3.8. Conexión JDBC	34
2.3.9. Sistema gestor de base de datos MySQL	35
2.3.10. Sistema gestor de base de datos PostgreSQL	36
CAPÍTULO III MARCO METODOLÓGICO	37
3.1. TIPO Y DISEÑO DE LA INVESTIGACIÓN	37
3.2. POBLACIÓN Y MUESTRA	38
3.3. OPERACIONALIZACIÓN DE VARIABLES	39
3.4. TÉCNICAS E INSTRUMENTOS PARA RECOLECCIÓN DE DATOS	41
3.5. PROCESAMIENTO Y ANÁLISIS DE DATOS	44
CAPÍTULO IV RESULTADOS Y DISCUSIÓN	61
4.1. RESULTADOS	61
4.1.1. Resultados a nivel descriptivo	61
4.1.2. Resultados a nivel inferencial	68
4.2. DISCUSIÓN	78
CONCLUSIONES	81
RECOMENDACIONES	83
REFERENCIAS BIBLIOGRÁFICAS	84
ANEXOS	90

## ÍNDICE DE TABLAS

<b>Tabla 1.</b> Operacionalización de variables	40
<b>Tabla 2.</b> Especificaciones del hardware	46
<b>Tabla 3.</b> Especificaciones del software	47
<b>Tabla 4.</b> Resultados del tiempo de respuesta para PostgreSQL	61
<b>Tabla 5.</b> Resultados del tiempo de respuesta para MySQL	62
<b>Tabla 6.</b> Resultados del uso de memoria para PostgreSQL	65
<b>Tabla 7.</b> Resultados del uso de memoria para MySQL	66

## ÍNDICE DE FIGURAS

<b>Figura 1.</b> Ranking de sistemas gestores de base de datos por modelo	7
<b>Figura 2.</b> Tendencia histórica de popularidad por tipo de licencia	8
<b>Figura 3.</b> Ranking de SGBD con licencia de código abierto	14
<b>Figura 4.</b> Ranking de sistemas gestores de bases de datos	36
<b>Figura 5.</b> Informe “uso de disco por tabla”	48
<b>Figura 6.</b> Esquema relacional de la base de datos	49
<b>Figura 7.</b> Configuración de la base de datos origen	50
<b>Figura 8.</b> Configuración de la base de datos destino PostgreSQL	51
<b>Figura 9.</b> Configuración de la base de datos destino MySQL	51
<b>Figura 10.</b> Migración de SQL Server a PostgreSQL	52
<b>Figura 11.</b> Migración de SQL Server a MySQL	52
<b>Figura 12.</b> Configuración de la conexión con PostgreSQL	53
<b>Figura 13.</b> Configuración de la conexión con MySQL	54
<b>Figura 14.</b> Configuración del grupo de hilos para el plan de pruebas	56
<b>Figura 15.</b> Reporte del costo de plan de ejecución	57
<b>Figura 16.</b> Costo de plan de ejecución de las consultas SQL	58
<b>Figura 17.</b> Configuración de la petición JDBC para PostgreSQL	59
<b>Figura 18.</b> Configuración de la petición JDBC para MySQL	59
<b>Figura 19.</b> Gráfico comparativo con respecto al tiempo de respuesta	63
<b>Figura 20.</b> Gráfico comparativo de tiempo de respuesta con datos normalizados	64
<b>Figura 21.</b> Gráfico comparativo con respecto al uso de memoria	67
<b>Figura 22.</b> Distribución del tiempo de respuesta para PostgreSQL	69
<b>Figura 23.</b> Distribución del tiempo de respuesta para MySQL	70
<b>Figura 24.</b> Prueba de normalidad para el tiempo de respuesta	71
<b>Figura 25.</b> Distribución del uso de memoria para PostgreSQL	72
<b>Figura 26.</b> Distribución del uso de memoria para MySQL	73
<b>Figura 27.</b> Prueba de normalidad para el uso de memoria	74
<b>Figura 28.</b> Prueba U de Mann-Whitney para el tiempo de respuesta	75
<b>Figura 29.</b> Prueba U de Mann-Whitney para el uso de memoria	77

## **RESUMEN**

La presente tesis se enfoca en comparar el desempeño de los sistemas gestores de base de datos PostgreSQL y MySQL en cuanto al tiempo de respuesta y uso de memoria, utilizando el conjunto de datos académicos proporcionados por la Dirección de Actividades y Servicios Académicos de la Universidad Nacional Jorge Basadre Grohmann – Tacna. Para lo cual se utilizó la metodología de benchmarking que consiste en comparar características de rendimiento de diferentes tecnologías, en nuestro caso los sistemas gestores de base de datos PostgreSQL y MySQL; para la obtención de resultados se utilizó el software JMeter, con los cuales se pudo hacer un análisis estadístico e inferencial. Finalmente se verificó que el sistema gestor de base de datos PostgreSQL ejecutó 200 consultas a nuestra base de datos de 325 536 registros en 12 tablas con un tiempo promedio de 198 milisegundos, mientras que el sistema gestor de base de datos MySQL lo hizo en 3 388 milisegundos bajo las mismas condiciones; sin embargo, el uso de memoria utilizada por el sistema gestor de base de datos PostgreSQL es 1 385 436 bytes y de MySQL es 1 275 725 bytes; es decir, PostgreSQL presenta menor tiempo de respuesta pero mayor uso de memoria, lo contrario a MySQL, que presenta mayor tiempo de respuesta pero menor uso de memoria; comportamiento que coincide total y parcialmente con los resultados de otras

investigaciones que utilizan diferente cantidad de registros, es así que se concluye en que ambos sistemas gestores de base de datos muestran un desempeño eficiente, por un lado, PostgreSQL muestra mayor velocidad al atender las consultas y por otro lado MySQL muestra un mejor manejo en el uso de memoria al atender las consultas, queda en la empresa priorizar el tiempo de respuesta o el uso de memoria y en base a ello poder decidir por uno o por otro.

**Palabras Clave**

Desempeño, bases de datos, PostgreSQL, MySQL, tiempo de respuesta, uso de memoria.

## **ABSTRACT**

This thesis focuses on comparing the performance of the PostgreSQL and MySQL database management systems in terms of response time and memory use, using the set of academic data provided by the Directorate of Activities and Academic Services of the National University Jorge Basadre Grohmann – Tacna. For which the benchmarking methodology was used, which consists of comparing performance characteristics of different technologies, in our case the PostgreSQL and MySQL database management systems; To obtain results, the JMeter software was used, with which a statistical and inferential analysis could be carried out. Finally, it was verified that the PostgreSQL database management system executed 200 queries to our database of 325,536 records in 12 tables with an average time of 198 milliseconds, while the MySQL database management system did so in 3,388. milliseconds under the same conditions; however, the memory usage used by PostgreSQL database management system is 1,385,436 bytes and MySQL is 1,275,725 bytes; that is, PostgreSQL has a shorter response time but higher memory usage, the opposite of MySQL, which has a longer response time but lower memory usage; behavior that fully and partially coincides with the results of other investigations that use different amounts of records, so it is concluded that both database management systems show

efficient performance, on the one hand, PostgreSQL shows greater speed when serving queries and on the other hand, MySQL shows a better management in the use of memory when attending the queries, it is up to the company to prioritize the response time or the use of memory and based on this, be able to decide for one or the other.

**Keywords**

Performance, database, PostgreSQL, MySQL, response time, memory usage.

## **INTRODUCCIÓN**

Hueso Ibáñez (2015) afirma que “la información es un recurso cada vez más crítico e importante para las empresas y organizaciones. Por ello es crucial disponer de herramientas que permitan y faciliten su administración”. (p. 7)

En este contexto, aparecen los sistemas gestores de base de datos que han pasado de ser simples repositorios a verdaderas herramientas inteligentes que gestionan la información de manera eficiente; es por eso que, las entidades deben buscar un sistema gestor de base de datos que tenga un buen desempeño, que les convenga y a su vez que cumpla todos los requerimientos necesarios para una eficiente gestión de la información.

La presente investigación se enfoca en desarrollar un análisis crítico y comparativo que nos ayude en la selección de un determinado SGBD en cuanto al tiempo de respuesta y uso de memoria, cuando es necesario procesar información referente al ámbito académico de nivel superior universitario. La investigación comprende toda esta temática en 5 capítulos, además de las conclusiones, las recomendaciones, la referencia bibliográfica y los anexos, que se encuentran estructurados de la siguiente manera

En el capítulo I elaboramos el planteamiento del problema, su formulación, descripción de la realidad problemática de la investigación.

El capítulo II muestra una recopilación detallada de antecedentes desarrollados en el ámbito de la investigación.

En el capítulo III desarrollamos detalladamente las labores experimentales de nuestro trabajo.

El capítulo IV muestra los resultados, a través del análisis descriptivo estadístico e inferencial.

El capítulo V presenta las discusiones, conclusiones, recomendaciones, referencia bibliográfica y finalmente los anexos.

## CAPÍTULO I

### PLANTEAMIENTO DEL PROBLEMA

#### 1.1. DESCRIPCIÓN DEL PROBLEMA

Las bases de datos relacionales se han proyectado a ser una de las herramientas más utilizadas. No obstante, han tenido transcurrir en consecuencia mucho tiempo para que estas bases de datos evolucionen y sean ampliamente utilizadas en diferentes ámbitos y contextos. En la figura 1, BD-engines nos muestra la popularidad de los sistemas gestores de base de datos desglosada por modelo de base de datos.

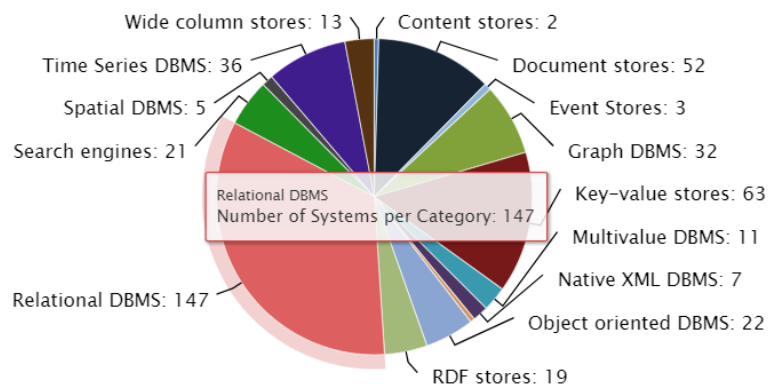


Figura 1. Ranking de sistemas gestores de base de datos por modelo

Fuente: BD-Engines consultado el 02 de Julio del 2021

Con el pasar de los años, el software libre se ha consolidado como alternativa frente al software propietario, ofreciendo los mismos servicios a un costo reducido. (López Herrera, 2016)

Vemos en figura 2 la tendencia histórica de como los sistemas gestores de base de datos con licencia de código abierto aumenta en porcentaje de popularidad hasta superar a los sistemas gestores de base de datos con licencia comercial.

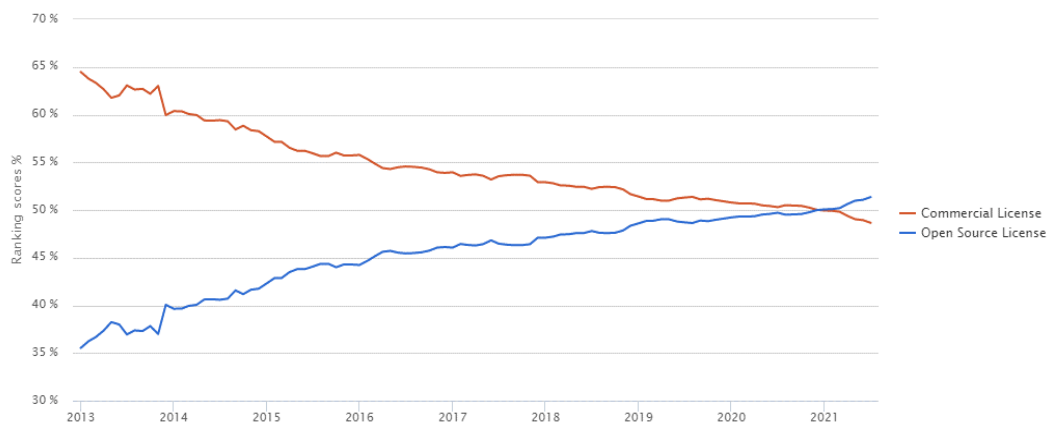


Figura 2. Tendencia histórica de popularidad por tipo de licencia

Fuente: BD-Engines consultado el 02 de Julio del 2021

Existen diferentes sistemas gestores de base de datos, que almacenan y recuperan la información de las bases de datos de una forma práctica y eficiente; por lo tanto, es necesario conocer qué sistema gestor de bases de datos tiene un mejor desempeño, adaptándose y desenvolviéndose mejor a cada necesidad de una entidad, según sus características y funcionalidades.

### **1.1.1. Antecedentes del problema**

Para mayor comprensión sobre el problema, se presenta una colección de investigaciones relacionadas con la realidad problemática de la investigación.

López Herrera (2016) afirma que, la mayoría de los datos sensibles del mundo están almacenados en sistemas gestores de bases de datos, debido a ello, las organizaciones públicas buscan un sistema gestor de base de datos que cumpla con un buen desempeño; asimismo indica que mientras más sencillo sea acceder a la información, será mejor valorado por los usuarios.

Palaquibay & Azucena (2010) realizan un análisis comparativo del funcionamiento y rendimiento de dos SGBD, se estudian los servicios y características principales de los SGBD, también se definen diferentes escenarios y mediante un test de rendimiento, se analiza el comportamiento de cada sistema gestor de base de datos. Para finalizar, sobre los resultados de rendimiento, nos muestra un tiempo de respuesta utilizando consultas de complejidad alta para 100 000 filas de 9 576 de promedio con Oracle, 18 887 de promedio con SQL Server y 2 767 de promedio con MySQL; por lo que se determina que MySQL ofrece el mejor desempeño, seguido por SQL Server y por último Oracle; asimismo se indica que la decisión de utilizar un SGBD u otro depende de las necesidades de cada empresa.

Por otro lado, Cambi Alvarado & Zúñiga García (2006) indican que los dos sistemas gestores de base de datos más populares son sin duda MySQL y PostgreSQL, considerados por muchos, de los mejores que existen en el mercado. Con la finalidad de probar las potencialidades que ofrece cada gestor, se realizan tests de rendimiento en cuanto se refiere a migración de datos, velocidad en búsqueda de datos, backups, etc., utilizando una base de datos con 9 000 registros en 21 tablas; entre los resultados del test de rendimiento se puede observar que PostgreSQL tiene un tiempo de respuesta de 110 ms con memoria utilizada de 210,148 MB, mientras que MySQL tiene un tiempo de respuesta de 356,2 ms con memoria utilizada de 200,844 MB para velocidad en búsqueda y ordenamiento de datos. Finalmente se concluye que ambos son magníficos gestores de base de datos, quedando la decisión por parte de la persona encargada de decidir cuál es el indicado para manejar sus datos.

También, Soberón León (2020) realiza un análisis comparativo de dos SGBD, al igual que la presente investigación, MySQL y PostgreSQL sobre el rendimiento respecto a la manipulación de registros en los procesos CRUD. Se observó que los SGBD tienen ventajas para el manejo de la información, las pruebas se realizaron con una variación en la cantidad de registros para los SGBD, en los resultados se puede evidenciar que, en el tiempo de respuesta para 100 000 registros, PostgreSQL obtuvo 0,067 segundos en promedio con un consumo de memoria de 55 086 359,27 bytes,

mientras que MySQL obtuvo 0,435 segundos en promedio con un consumo de memoria de 808 366 080 bytes, es así que concluye que, MySQL demanda un mayor consumo con respecto a PostgreSQL.

A nivel internacional, Yang (2011) explora las diferentes funciones y características de los sistemas gestores de base de datos MySQL y PostgreSQL, tanto a nivel profesional proveniente del punto de vista técnico, como la arquitectura, funciones principales, consultas, transacciones, etc.; como también, a nivel de exposición de mercado, como la opinión de usuarios, información básica, usabilidad, licencia, etc. De acuerdo con la investigación realizada por Xiaojie, PostgreSQL sin duda tiene las funciones de todo horizonte como un sistema gestor de base de datos. El gestor de base de datos MySQL es muy sencillo de utilizar y en el ámbito científico se le considera mucho más potente que PostgreSQL entre el público, lo que de alguna manera puede compensar las carencias en las funciones también hace que la gente crea que MySQL tiene la capacidad potencial de mejorar a un nivel superior. Su buen desempeño hace que los usuarios consideren a MySQL como un gestor potente que les permita gestionar mejor sus datos.

Otra investigación comparativa nos ofrece Bassil (2012), que presenta un estudio comparativo del rendimiento de los sistemas gestores de base de datos disponibles en el mercado, como MS Access 2010, MS SQL Server 2008, MySQL 5.5,

IBM DB2 y Oracle 11g. El estudio estuvo destinado a la realización de consultas SQL variando su nivel de complejidad en los diferentes sistemas gestores de base de datos bajo prueba, para los cuales, se utilizó una base de datos con 1 000 000 registros. Es así que mediante este estudio se realiza la comparación del tiempo de ejecución, el uso de CPU, uso de memoria y recuento de subprocesos de cada sistema gestor de base de datos. Los gráficos mostraron claramente que IBM DB2 es el sistema gestor de base de datos más rápido con 1 032,3 de promedio, sin embargo, MS Access tiene una menor utilización de la CPU que otros sistemas gestores de base de datos teniendo un 22 % e IBM DB2 es el sistema gestor de base de datos que consume mayor cantidad de memoria primaria con 22,6 MB; en cuanto a MySQL se muestra que el tiempo promedio de ejecución es 1 222,3 con un consumo de memoria de 12,5 MB.

Mientras que (Asiminidis, Kokkonis, & Kontogiannis, 2018), se enfoca en la comparación de bases de datos relacionales como MySQL y PostgreSQL y no relacionales como MongoDB, en la cual los autores concluyen que para consultas de inserción con gran cantidad de datos, PostgreSQL presenta el menor tiempo de ejecución, sin embargo, para una cantidad pequeña de datos MongoDB obtuvo el menor tiempo; para una función de agregación aplicada en una gran cantidad de registros de IoT, MySQL presenta los mejores resultados de desempeño en términos de tiempo de ejecución. También se puede observar una comparación de desempeño

entre MySQL y PostgreSQL para 10MB de tamaño por consultas select un promedio de tiempo de 200 ms y 400 ms respectivamente.

### **1.1.2. Problemática de la investigación**

Como ya se mencionó anteriormente, muchas entidades trabajan con grandes cantidades de datos, y es de vital importancia para ellas gestionar su información de manera eficiente, las herramientas que ayudan y facilitan esa gestión son los sistemas gestores de base de datos.

Existen varios sistemas gestores de base de datos, pero no todos son iguales y elegir entre alguno de ellos es complicado, debido a que no sabemos si ese gestor es la mejor opción para que satisfaga nuestras necesidades de manera eficiente.

Ya que, para algunas entidades es prioridad la reducción de costos y el desempeño del sistema gestor de base de datos de su elección, se tomará en consideración dos de ellos que tienen licencia de código abierto, con la finalidad de definir el SGBD que proporciona mejor manejo de la información con respecto al tiempo de respuesta y uso de memoria utilizando como caso de estudio los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019. Los sistemas gestores de bases de datos son MySQL y PostgreSQL, debido a que, son los más populares en el ranking de sistemas gestores de base de datos con licencia de código abierto que nos ofrece BD-engines.

Rango	Sistema	Puntaje	Clasificación general
1.	MySQL	1228	2.
2.	PostgreSQL	577	4.
3.	MongoDB	496	5.
4.	Redis	168	6.
5.	Elasticsearch	156	8.

Figura 3. Ranking de SGBD con licencia de código abierto

Fuente: BD-Engines consultado el 02 de Julio del 2021

## 1.2. FORMULACIÓN DEL PROBLEMA

Una vez descrita la realidad problemática, estamos en capacidad de formular nuestro problema general de investigación

### Problema general

¿Cómo es el desempeño de los sistemas gestores de base de datos PostgreSQL y MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019?

### Problemas específicos

Para poder responder el problema general, se formularon los siguientes problemas específicos:

- ¿Cómo es el desempeño en cuanto al tiempo de respuesta de los sistemas gestores de base de datos PostgreSQL y MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019?
- ¿Cómo es el desempeño en cuanto al uso de memoria de los sistemas gestores de base de datos PostgreSQL y MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019?

### **1.3. JUSTIFICACIÓN E IMPORTANCIA**

“Hoy los volúmenes de información crecen a un ritmo sin precedentes (...) las empresas no solo desean almacenar esta información, sino quieren sacarle el mayor provecho; los usuarios piden cada vez más velocidad en las consultas” (Romero, Sanabria, & Cuervo, 2012, p.23).

En ese sentido, es necesario contar con un sistema gestor de base de datos que simplifique las tareas las labores de gestión y extracción de datos en el momento preciso, con la certeza de que el SGBD no fallará generando confianza en los procesos; es decir, las empresas se están enfocando en el desempeño que les brinda un sistema gestor de base de datos.

Por otro lado, buscan también la reducción de costos para administrar sus datos; en los últimos años, el software de código abierto llegó a tener mayor interés tanto para la industria del software como para la teoría económica. Algunas compañías han

comenzado a darse cuenta que el software de código abierto puede también ser altamente rentable (Khan, 2006).

La presente investigación se enfoca específicamente a dos sistemas gestores de base de datos de código abierto MySQL y PostgreSQL, con el fin de identificar cuál de estos dos gestores tiene el mejor desempeño en cuanto al tiempo de respuesta y uso de memoria, utilizando como caso de estudio, una base de datos perteneciente al ámbito académico de nivel superior universitario, con la finalidad de realizar las pruebas en un ambiente real con gran cantidad de información. En consecuencia, esta investigación servirá como apoyo cuando se necesite elegir algún sistema gestor de base de datos para el manejo de su información considerando las características y necesidades de cada empresa.

#### **1.4. ALCANCES Y LIMITACIONES**

##### **Alcances**

El alcance de la presente investigación se ciñe a la comparativa del desempeño entre el sistema gestor de base de datos PostgreSQL y el sistema gestor de base de datos MySQL, teniendo como caso de estudio la base de datos relacional del Sistema de Registro de estudiantes matriculados de la Dirección de Actividades y Servicios Académicos de la Universidad Nacional Jorge Basadre Grohmann; dicha comparativa

tomará como criterio una prueba de desempeño que se ha establecido, la cual tiene como indicadores el tiempo de respuesta y uso de memoria.

### **Limitaciones**

No se identificaron limitaciones para la presente investigación.

## **1.5. OBJETIVOS**

En esta sección se define el propósito de la presente investigación y los objetivos específicos para alcanzar dicho propósito.

### **1.5.1. Objetivo general**

Comparar el desempeño de los sistemas gestores de base de datos PostgreSQL y MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019.

### **1.5.2. Objetivos específicos**

- Comparar el tiempo de respuesta de los sistemas gestores de base de datos PostgreSQL y MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019.
- Comparar el uso de memoria de los sistemas gestores de base de datos PostgreSQL y MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019.

## **1.6. HIPÓTESIS**

En esta sección se han planteado hipótesis como propuestas de respuesta a nuestros problemas específicos y general.

### **1.6.1. Hipótesis general**

**Ho:** El desempeño del sistema gestor de base de datos PostgreSQL no es significativamente más eficiente que el desempeño del sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann – 2019.

**Hi:** El desempeño del sistema gestor de base de datos PostgreSQL es significativamente más eficiente que el desempeño del sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019.

### **1.6.2. Subhipótesis**

#### **Hipótesis específica 1**

**Ho:** El tiempo de respuesta del sistema gestor de base de datos PostgreSQL es mayor o igual que el sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann – 2019.

**Hi:** El tiempo de respuesta del sistema gestor de base de datos PostgreSQL es menor que el sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019.

**Hipótesis específica 2**

**Ho:** El uso de memoria del sistema gestor de base de datos PostgreSQL es mayor o igual que el sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019.

**Hi:** El uso de memoria del sistema gestor de base de datos PostgreSQL es menor que el sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019

## **CAPÍTULO II**

### **MARCO TEÓRICO**

En este capítulo se definirán algunos términos y las bases teóricas que forman parte del sustento teórico de la investigación.

#### **2.1. ANTECEDENTES DEL ESTUDIO**

Para mayor comprensión sobre el problema, se presenta una colección de investigaciones relacionadas con el estudio de la presente investigación.

(Klimek, B., & Skublewska-Paszowska, M., 2021) realizan un estudio comparativo sobre el rendimiento de dos sistemas de gestión de bases de datos relacionales, PostgreSQL y MySQL, para lo cual realizan pruebas de rendimiento basado en consultas SQL básicas evaluando solo el tiempo en milisegundos, afirmando en base a sus resultados que PostgreSQL es más eficiente aplicando 1 000 consultas en cantidad de datos pequeñas.

(Praba, A. D., & Safitri, M., 2020) en su estudio, determina el rendimiento del tiempo de respuesta de MySQL y PostgreSQL, en la investigación se realizó el experimento en la base de datos con el número de registros de 50 mil, 100 mil y 1 millón, respectivamente, para dichos experimentos se llevó a cabo con tres consultas: Select\*from, select count () y join table. Los resultados del experimento muestran que PostgreSql tarda más tiempo que Mysql en el procesamiento de consultas.

También, (Wodyk, R., & Skublewska-Paszkowska, M., 2020), realizan una comparación de rendimiento de bases de datos relacionales SQL Server, MySQL y PostgreSQL usando una aplicación web, en esta investigación se evaluó el tiempo de ejecución de consultas, tanto simples como utilizando concatenación de columnas y tablas. Finalmente, los autores concluyen que en el caso de bases de datos en las que el número de registros no es demasiado grande (hasta 1000 registros) y los parámetros técnicos del dispositivo en el que se ejecuta la base de datos son de clase baja o media, MySQL funciona muy bien.

Asimismo, (Asiminidis, Kokkonis & Kontogiannis, 2018) se enfoca en la comparación de bases de datos relacionales (MySQL y PostgreSQL) y no relacionales (MongoDB), en la cual los autores concluyen que para consultas de inserción con gran cantidad de datos, PostgreSQL presenta el menor tiempo de ejecución, sin embargo, para una cantidad pequeña de datos MongoDB obtuvo el menor tiempo; para una función de agregación aplicada en una gran cantidad de registros de IoT, MySQL presenta los mejores resultados de desempeño en términos de tiempo de ejecución.

(Kolonko, 2018), presenta una investigación sobre los métodos de medición del desempeño de las bases de datos, características de las bases de datos y popularidad de los motores de base de datos relacionales y no relacionales con el fin de comparar dos motores de bases de datos representantes de cada familia, asimismo utilizó YCSB

Benchmark como un método de comparación de rendimiento en un experimento para comparar OracleDB y MongoDB. La comparación de desempeño de los motores de base de datos demostró que MongoDB es más eficiente en términos de latencia de operación promedio y rendimiento para cada una de las cargas de trabajo medidas, en cuanto a OracleDB, presentó resultados estables y consistentes en cada una de las categorías dejando la elección final de la base de datos para las especificaciones del proyecto de ingeniería de software.

Por otro lado, (Suri y Sharma, 2011), analizan en detalle dos modelos de bases de datos existentes, la base de datos relacional y la base de datos orientada a objetos, hace un análisis de los beneficios de las bases de datos orientadas a objetos sobre la convencional base de datos relacional en almacenamiento de datos y afirma que para datos complejos el sistema gestor de base de datos orientado a objetos es más rápido que el sistema gestor de bases de datos relacional, el objetivo del sistema gestor de base de datos orientado a objetos es dirigir la manipulación de objetos por parte del software de la aplicación.

(Díaz, Tzancoff, Rodríguez, & Soria, 2008), destacan la utilización de la herramienta Jmeter para el proceso de pruebas con la que simplifica el análisis para la etapa de testeo de rendimiento del servidor web del sitio de la Universidad Nacional de La Plata, siendo de gran ayuda para realizar estas pruebas, gracias a su flexibilidad de

configuración y claridad de visualización de los resultados de las pruebas, lo que ayudó a su interpretación.

(Espinoza & Sarango, 2013), afirman que, en la actualidad, las bases de datos relacionales se han proyectado a ser una de las herramientas más difundidas en nuestra sociedad de la información, las cuales permiten almacenar, manipular y recuperar información de diversos campos. No obstante, han tenido que pasar algunos años para que, estas bases de datos puedan alcanzar el desarrollo que hoy conocemos, comenzando con el almacenamiento de información con tarjetas perforadas y cintas magnéticas, uso de discos que dieron inicio a las bases de datos, bases de datos de red y bases de datos jerárquicas. Surge un modelo relacional, que en un principio consistió en una serie de reglas para la evaluación de administradores de sistemas de datos relacionales.

También, (Quizhpe, 2009), efectúa un análisis comparativo que sirva como referencia para la selección de un determinado Sistema de Gestión de Bases de Datos, haciendo énfasis fundamentalmente en dos puntos de vista: el funcionamiento y el rendimiento de los sistemas; se estudian los servicios más relevantes y características de los sistemas, también se determinan escenarios muy bien diferenciados y mediante un test de rendimiento, observa el comportamiento de cada SGDB. Para finalizar, después de un meticuloso análisis basado en funcionalidad, soporte técnico, costos,

herramientas de administración y rendimiento, se determina que cada sistema tiene características, ventajas e inconvenientes, por tal razón la elección de uno u otro sistema para gestionar una base de datos vendrá definida por las necesidades de la empresa.

## **2.2. BASES TEÓRICAS**

### **2.2.1. Historia de las bases de datos**

Las aplicaciones informáticas de los años setenta se daban por lotes para una tarea muy específica, para esto se utilizaban ficheros, cinta magnéticas y discos. A medida que se introdujeron las líneas de comunicación se fueron creando programas y actualizando los datos en los ficheros. Posteriormente se tuvo que interrelacionar los ficheros y diseñar un modo que fuese de fácil acceso.

A este conjunto de ficheros interrelacionados en un inicio se le denominó Data Banks y posteriormente Data bases. Es así el surgimiento del nombre bases de datos (BD).

### **2.2.2. Base de datos**

Una base de datos es una colección de datos relacionados, se construyen siguiendo un diseño y se almacenan datos para realizar acciones específicas. Los datos que se almacenan en una base de datos tienen un origen y pertenecen o llevan relación

con un evento en específico de la vida real, asimismo el contenido de las bases de datos es de interés de un grupo de usuarios activos. (Elmasri, Díaz Martín, & Navathe, 2011)

### **2.2.3 Sistema gestor de bases de datos**

Un sistema gestor de base de datos es una herramienta que sirve para crear BDs y mantenerlas, debido a que nos facilita su administración. Esta herramienta nos ofrece una interfaz mediante la cual el usuario puede interactuar con la BD. Con los SGBD tenemos una colección de datos relacionados y almacenados y un software para la gestión de dichos datos que nos proporciona herramientas y funciones. (Hueso Ibáñez, 2015)

### **Funciones de los sistemas gestores de bases de datos**

Una de las funciones principales de un SGBD es permitir la ejecución de consultas fundamentales para los usuarios como la creación, actualización, eliminación y edición, garantizando su eficiencia y coherencia. (Hueso Ibáñez, 2015)

Las características y funciones de los SGBD son:

- **Catálogo:** también conocido como diccionario de datos, es donde se almacena la información que describe a los datos, contiene datos como nombre, tipo, tamaño, sus relaciones y condiciones de integridad, así como los usuarios que tienen permiso para gestionar las bases de datos (Hueso Ibáñez, 2015).

- Integridad: otra función es garantizar la integridad de los datos mediante un mecanismo que gestione las transacciones y se ejecuten todas las que necesitan hacerlo o bien que no, para que cambien el contenido correctamente (Hueso Ibáñez, 2015).
- Actualizaciones: Gestionar para que los usuarios accedan concurrentemente es fácil cuando se quiere leer datos, pero cuando los accesos concurrentes requieren actualización, pueden producirse interferencias que generen inconsistencias y fallas en la base de datos (Hueso Ibáñez, 2015).
- Recuperación: Cuando ocurra alguna falla en una transacción que dañe la base de datos, se debe poder reestablecer la base de datos a un estado estable sin errores (Hueso Ibáñez, 2015).
- Integración: otra función es la integración con un software de comunicación como por ejemplo terminales, mediante las cuales se puede conectar a una máquina en la que funciona el SGBD, estos terminales utilizan la comunicación a través de la red desde lugares remotos. En estas comunicaciones el SGBD recibe peticiones y responde de la misma manera (Hueso Ibáñez, 2015).
- Restricciones: El SGBD debe proporcionar las restricciones necesarias para garantizar la consistencia e integración de los datos al realizarse transacciones

y tareas siguiendo ciertas reglas; con el fin de proteger la base de datos y su calidad (Hueso Ibáñez, 2015).

- Administración: la administración efectiva de una base de datos implica un diseño correcto que garantice la disponibilidad y control de los datos. (Hueso Ibáñez, 2015)

### **Utilidades del Sistema Gestor de Base de Datos**

Las bases de datos también tienen utilidades que ayudan al administrador de base de datos a gestionar el sistema de bases de datos. (Elmasri et al., 2011)

Las utilidades de los SGBD son las siguientes:

- Carga: Permite cargar los archivos de datos en la base de datos, especificando el formato (origen) y la estructura (destino) deseada. La transferencia de datos de un SGBD a otro es cada vez más común en las empresas con la proliferación. (Elmasri et al., 2011).
- Copia de seguridad: Esta utilidad crea un backup de la base de datos, mediante la descarga de la base de datos entera en un almacenamiento externo. También se suelen utilizar las copias de seguridad incrementales, donde solo se hace el respaldo de los cambios desde la última copia. La copia de seguridad incremental es más compleja, pero ahorra espacio (Elmasri et al., 2011).

- Reorganización: La reorganización de los archivos de la BD se realiza con el objetivo de optimizar el rendimiento. (Elmasri et al., 2011)
- Monitorización: Una utilidad que ayuda bastante a un administrador de bases de datos (DBA) son las estadísticas para el monitoreo de la BD, que se utilizan para la toma de decisiones con el fin de mejorar el rendimiento (Elmasri et al., 2011).

#### **2.2.4. Modelo relacional**

El modelo relacional es una representación de la BD mediante un grupo de relaciones, cada relación es una tabla que contiene valores y cada fila de esta tabla es un grupo de valores relacionados que representa un hecho que se corresponde con una relación o entidad.

Para la identificación e interpretación del significado de los valores de las filas se le da un nombre a las tablas y columnas (Elmasri et al., 2011).

#### **2.2.5. Niveles de abstracción de base de datos**

Los datos en los sistemas gestores de base de datos se describen en tres niveles de abstracción:

Esquema Conceptual

El esquema conceptual o lógico describe las relaciones de las bases de datos en un RDMS, en uno no relacional, describe los datos en base al modelo de datos (Ramakrishnan & Gehrke, 2003).

#### Esquema Físico

Este esquema detalla otras características de almacenamiento, define como se almacenan las relaciones del esquema lógico en dispositivos externos (Ramakrishnan & Gehrke, 2003).

#### Esquema Externo

Son esquemas personalizados, adicionales a los otros dos ya mencionados, que se crean según las necesidades de un grupo de usuarios en concreto (Ramakrishnan & Gehrke, 2003).

### **2.2.6. Lenguajes de base de datos**

Los SGBD incluyen dos lenguajes: LDD Y LMD, uno es para el esquema y el otro para las consultas; un lenguaje ampliamente usado que incluye estos dos lenguajes es el lenguaje SQL.

#### Lenguaje de definición de datos

El esquema de una BD se detalla mediante un grupo de definiciones o instrucciones, por ejemplo, que cree una entidad llamada cuenta, en el lenguaje SQL sería:

create table cuenta (número-cuenta char (10), saldo integer)

Al ejecutar la instrucción LDD se crea la tabla cuenta (Silberschatz, Korth, & Sudarshan, 2002).

Lenguaje de manipulación de datos

Este lenguaje permite la manipulación de los datos relacionados mediante un modelo apropiado brindando el acceso para los usuarios a dichos datos. Existen dos tipos en este lenguaje:

Procedimentales. El usuario tiene que especificar qué datos desea obtener y cómo hacerlo.

Declarativos o no procedimentales. El usuario solo tiene que especificar qué datos desea obtener. (Silberschatz et al., 2002)

## **2.3. DEFINICIÓN DE TÉRMINOS**

### **2.3.1. Desempeño**

“Es el número de unidades de trabajo que se pueden gestionar por unidad de tiempo; por ejemplo, peticiones por segundo, llamadas por día, visitas por segundo, informes anuales”. (Medina, 2014, p.14)

### **2.3.2. Pruebas de desempeño**

Son una colección de pruebas que sirven para saber si un servicio IT ofrece calidad a sus usuarios con respecto a unos parámetros (disponibilidad, tiempo de

respuesta) ante unos escenarios determinados (tipo de peticiones, número de usuarios). (Medina, 2014)

Tipos de pruebas de desempeño:

- Pruebas de carga: Estas pruebas evalúan el comportamiento de un SGBD al atender muchas peticiones por segundo, dichas peticiones, en un nivel básico de esta prueba, pueden ser la misma cantidad de usuarios concurrentes esperados para el escenario de prueba con unos márgenes (Medina, 2014).
- Pruebas de stress: El objetivo principal de estas pruebas es hallar el punto de ruptura, es lo que la diferencia de las pruebas de carga; asimismo determinar las causas que son producto de problemas que ocurren al probar al SGBD con mucha carga, estos pueden ser el agotamiento de capacidad, mala escalabilidad, condiciones de carrera, fugas de memoria, entre otros (Medina, 2014).

### **2.3.3. Indicadores de desempeño**

Podemos definir los indicadores de desempeño, como una proposición cuantificada que refleja las necesidades de los usuarios y que sirve para la evaluación y comparación del rendimiento (Poll, Boekhorst, & Abad Hiraldo, 1998).

#### **2.3.4. Tiempo de respuesta**

“El tiempo total que tarda un sistema de información en atender una petición de usuario, incluye el tiempo de transmisión, el tiempo de latencia y el tiempo de procesamiento” (Medina, 2014, p.15).

#### **2.3.5. Uso de memoria**

Se define la utilización como el “porcentaje de tiempo que un recurso está ocupado atendiendo las peticiones provenientes del usuario” (Medina, 2014, p.15); en la presente investigación el recurso corresponde a la memoria de acceso aleatorio.

#### **2.3.6. Memoria de acceso aleatorio**

Random Access Memory (RAM), es la memoria de una computadora que contiene instrucciones para que realice determinadas funciones, ésta es volátil, esto significa que los datos se retienen en la RAM mientras la computadora esté encendida, pero se pierden cuando se apaga. Cuando se reinicia la computadora, el sistema operativo y otros archivos se vuelven a cargar en la RAM, generalmente desde un disco duro (HDD) o unidad de estado sólido (SSD).

#### **2.3.7. JMeter**

JMeter ha llegado a ser un referente para la realización de pruebas de rendimiento, es multiplataforma y software libre, su facilidad de uso te permite aprender rápidamente a utilizar esta herramienta para pruebas de rendimiento

multiprotocolo, asimismo ofrece una interfaz gráfica de usuario que evitará el uso de código para la realización de las pruebas, además cuenta con herramientas para la generación de gráficos, ejecución en línea de comandos y ejecución distribuida. (Medina, 2014)

Los dos receptores o listeners que se utilizarán en la presente investigación, son los siguientes:

**a) Ver resultados en árbol**

Sirve para las pruebas funcionales en el proceso de depuración y validación del plan de pruebas, ya que muestra un árbol de todas las respuestas, lo que permite ver la respuesta de cualquier muestra, además de mostrar el tiempo que tardó en obtener esta respuesta.

**b) Reporte Resumen**

Este reporte emite un informe en una tabla, en la cual cada fila es una solicitud con su nombre correspondiente definido en la prueba; el rendimiento intelectual se calcula desde el punto de vista del objetivo del muestreador.

- Etiqueta: sirve para diferenciar los grupos de hilos mediante estas etiquetas.
- # Muestras: cantidad de muestras
- Promedio: tiempo promedio de un grupo de resultados
- Mín: tiempo mínimo de un grupo de resultados

- Max: tiempo máximo de un grupo de resultados
- Std. Dev.: desviación estándar del tiempo
- Error%: porcentaje de error
- Kb / seg: rendimiento en kilobytes por segundo
- Promedio Bytes: tamaño medio de la respuesta.

### **2.3.8. Conexión JDBC**

La API Java Database Connectivity (JDBC) nos permitirá acceder a bases de datos desde nuestro programa Java. Los drivers para poder acceder a cada sistema gestor de base de datos se tienen que descargar aparte, estos drivers son un traductor entre las llamadas de JDBC y los APIs específicos del sistema gestor de base de datos, normalmente un driver es creado por el propio desarrollador del sistema gestor de base de datos (Vanegas, 2005).

Un programa Java que utilice JDBC primero debe establecer una conexión con el sistema gestor de base de datos mediante el uso de un driver específico para cada sistema gestor de base de datos que estemos utilizando. Una vez establecida la conexión, se podrá realizar consultas a la base de datos con cualquier comando SQL mediante peticiones JDBC.

### **2.3.9. Sistema gestor de base de datos MySQL**

MySQL es un sistema gestor de bases de datos relacional (RDBMS) con gran capacidad de almacenamiento, ideal para organizar grandes volúmenes de datos de empresas grandes o pequeñas. La gran fama de MySQL le permite competir con RDBMS conocidos como Oracle y SQL Server.

MySQL tiene todas las herramientas para su instalación, administrar los accesos de usuarios, la información, así como, proteger y hacer volcados de datos. También es compatible con la mayor parte de los lenguajes de programación y ejecutarlos en casi todos los sistemas operativos para el desarrollo de aplicaciones. Las consultas en MySQL se realizan con el lenguaje SQL, el cual lo utilizan todas las RDBMS y mediante este lenguaje se pueden crear BDs, agregar, editar, recuperar y eliminar datos según los criterios requeridos (Gilfillan, 2003).

MySQL fue publicado bajo licencia dual: Licencia pública general/Licencia comercial por Oracle Corporation y está considerada como la base de datos de código abierto más popular del mundo. Por un lado, se ofrece bajo la licencia GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso.

### 2.3.10. Sistema gestor de base de datos PostgreSQL

PostgreSQL es el sistema gestor que tiene herramientas útiles para administrar bases de datos. Existen diferentes ediciones de PostgreSQL adecuadas a diferentes requisitos de rendimiento, tiempo de ejecución y precio. Por otro lado, los componentes de PostgreSQL a instalar también dependen de las necesidades del usuario (Lardière, 2019).

PostgreSQL es completamente software libre y con una licencia BSD, compatible con cualquier uso, ya sea personal o comercial. Este sistema gestor de base de datos es uno de los más populares, lo podemos comparar con el sistema gestor de base de datos MySQL, ya que, ambos se encuentran en el ranking general de popularidad que nos ofrece BD-Engines, obteniendo MySQL el segundo puesto y PostgreSQL el cuarto puesto como se observa en la figura 4.

371 systems in ranking, June 2021

Rank			DBMS	Database Model	Score		
Jun 2021	May 2021	Jun 2020			Jun 2021	May 2021	Jun 2020
1.	1.	1.	Oracle	Relational, Multi-model	1270.94	+1.00	-72.65
2.	2.	2.	MySQL	Relational, Multi-model	1227.86	-8.52	-50.03
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	991.07	-1.59	-76.24
4.	4.	4.	PostgreSQL	Relational, Multi-model	568.51	+9.26	+45.53
5.	5.	5.	MongoDB	Document, Multi-model	488.22	+7.20	+51.14
6.	6.	6.	IBM Db2	Relational, Multi-model	167.03	+0.37	+5.23
7.	7.	8.	Redis	Key-value, Multi-model	165.25	+3.08	+19.61
8.	8.	7.	Elasticsearch	Search engine, Multi-model	154.71	-0.65	+5.02
9.	9.	9.	SQLite	Relational	130.54	+3.84	+5.72
10.	10.	11.	Microsoft Access	Relational	114.94	-0.46	-2.24

Figura 4. Ranking de sistemas gestores de bases de datos

Fuente: BD-Engines

## **CAPÍTULO III**

### **MARCO METODOLÓGICO**

#### **3.1. TIPO Y DISEÑO DE LA INVESTIGACIÓN**

Según Horna (2012) el objetivo de la investigación descriptivo-comparativa es identificar las diferencias o semejanzas de un evento en dos o más grupos.

La presente investigación es del tipo descriptivo-comparativa; ya que, se comparan características de los dos sistemas gestores de bases de datos, en este caso, PostgreSQL y MySQL, se analizan la forma como varían, logrando establecer una relación de comparación considerando la variable desempeño y los indicadores respectivos.

Por otro lado; para Hernández Sampieri et al. (2014), si en una investigación no se manipulan las variables y solo se observa el evento para su respectivo análisis es una investigación no experimental. El diseño de la presente investigación es no experimental, pues se trabajó sobre una situación existente y no provocada.

Finalmente, según la dimensión temporal, esta investigación es transeccional o transversal, “su propósito es describir variables y analizar su incidencia e interrelación en un momento dado” (Hernández Sampieri et al., 2014, p.154).

## **3.2. POBLACIÓN Y MUESTRA**

### **Población**

Según Hernández Sampieri et al., (2014) “una población es el conjunto de todos los casos que concuerdan con una serie de especificaciones”. (p.174)

En consecuencia, la población para la presente investigación comprende los 3 431 970 registros de 253 230 estudiantes matriculados en la Universidad Nacional Jorge Basadre Grohmann en el periodo 1972 -2019.

### **Muestra**

Según Hernández Sampieri et al., (2014) “la muestra es, en esencia, un subgrupo de la población” (p.175); asimismo, nos indica que “en las muestras no probabilísticas, la elección de los elementos no depende de la probabilidad, sino de causas relacionadas con las características de la investigación o los propósitos del investigador”. (p.176)

La investigación trabajó con los 325 536 registros de 8 608 estudiantes matriculados en la Universidad Nacional Jorge Basadre Grohmann en el periodo 2019, debido a que, para el presente trabajo se utilizó el muestreo no probabilístico por conveniencia, con el objetivo de poder trabajar con datos más actuales.

### **3.3. OPERACIONALIZACIÓN DE VARIABLES**

*“Una variable es una propiedad que puede fluctuar y cuya variación es susceptible de medirse u observarse”* (Hernández Sampieri, Fernández Collado, & Pilar Baptista Lucio, 2014, p.105). La única variable identificada en esta investigación es el desempeño de los sistemas gestores de base de datos, por lo cual no existe variable independiente ni dependiente, por lo que es univariable.

#### **Definición de variables**

Variable de estudio: Desempeño de los sistemas gestores de base de datos

- Definición conceptual: Se refiere al indicador medible de la velocidad con que se procesan los datos y tareas de un sistema gestor de base de datos.
- Definición operacional: Evalúa el tiempo de respuesta y uso de memoria al realizar consultas a la base de datos utilizando los diferentes sistemas gestores de base de datos.

#### **Operacionalización de variables**

Según el grado de complejidad, nuestra variable es simple. *“Las variables simples son las que se manifiestan directamente a través de un indicador o unidad de medida. No se descomponen en dimensiones”* (Arias, 2012, p. 59); es por ello que nuestra variable desempeño tiene dos indicadores.

**Tabla 1.**  
*Operacionalización de variables*

<b>VARIABLE</b>	<b>INDICADORES</b>	<b>UNIDAD</b>	<b>ESCALA</b>
<b>Desempeño de los SGBD</b>	-Tiempo de respuesta	ms	Decimal
	-Uso de memoria	bytes	Decimal

Fuente: Indicadores seleccionados para la investigación.

### **Clasificación de variables**

A continuación, se clasificará nuestra variable por su naturaleza y su escala de medición.

Variable de estudio: Desempeño de los sistemas gestores de base de datos

- Por su naturaleza: Cuantitativa

Las variables cuantitativas “son aquellas que se expresan en valores o datos numéricos” (Arias, 2012, p.58).

- Por su escala de medición: Intervalo

La escala de medición por intervalos se utiliza para variables cuantitativas y en la misma no existe un cero absoluto que indique ausencia total de la variable. (Arias, 2012)

### **3.4. TÉCNICAS E INSTRUMENTOS PARA RECOLECCIÓN DE DATOS**

Para Hernández Sampieri et al. (2014) “recolectar los datos implica elaborar un plan detallado de procedimientos que nos conduzcan a reunir datos con un propósito específico”. (p.198)

#### **Técnica**

Según Horna (2012) “las técnicas son herramientas auxiliares que usan los diseños como medios de apoyo; son específicas y tienen un carácter instrumental”. (p. 202).

La técnica que se utilizará para la recolección de los datos en la presente investigación, será la observación, al comparar el desempeño entre dos sistemas gestores de bases de datos.

#### **Instrumento**

Según Horna (2012) “todo instrumento se construye para medir o registrar una variable o conjunto de variables a través de un número de preguntas, afirmaciones o indicadores (llamados ítems)”. (p.245)

El instrumento que se utilizará para medir la única variable en la presente investigación, será una guía de observación, en la cual irán las observaciones tanto del sistema gestor de base de datos PostgreSQL, como del sistema gestor de base de datos MySQL, esto se encuentra en el anexo 2.

La validación de nuestro instrumento se realizará mediante el juicio de expertos que, “*consiste, básicamente, en solicitar a una serie de personas la demanda de un juicio hacia un objeto, un instrumento, un material de enseñanza, o su opinión respecto a un aspecto concreto*”. (Cabero Almenara & Llorente Cejudo, 2013, p.14)

La validez de un instrumento es “el grado en que un instrumento de medida mide aquello que realmente pretende medir o sirve para el propósito para el que ha sido construido”. (Arribas, 2004, p.27) Para el cálculo de la validez de nuestro instrumento utilizaremos el coeficiente de validez de contenido (CVC) de (Hernández Nieto, 2002); que permite valorar el grado de acuerdo de los expertos respecto a cada uno de los diferentes ítems y al instrumento en general, el autor recomienda la participación de entre tres y cinco expertos, en este caso, se consideró la participación de 3 expertos.

Tras la aplicación de la escala tipo Likert de las alternativas de valoración de nuestro instrumento, las cuales fueron malo, regular y bueno; se asignó valores numéricos en el rango del 1 al 3 respectivamente; los resultados de la evaluación emitida por los expertos se encuentran en el Anexo 3. Para obtener el coeficiente de validez, primero calcularemos el CVC para cada ítem mediante la ecuación 1, para nuestro caso solo se realizará una vez, debido a que se evalúa el instrumento en general.

$$CVC_i = \frac{M_x}{V_{máx}} \quad [1]$$

Donde:

- $M_x$  : Promedio de puntuación de expertos
- $V_{máx}$  : Puntaje máxima del ítem

Aplicación de la ecuación 1:

$$CVC_i = \frac{M_x}{V_{máx}} = \frac{3}{3} = 1$$

Seguidamente se calculó el error asignado a cada ítem mediante la ecuación 2, para nuestro caso solo se realizará una vez, debido a que se evalúa el instrumento en general.

$$P_e = \left(\frac{1}{j}\right)^j \quad [2]$$

Donde:

- $j$  : el número de expertos participantes

Aplicación de la ecuación 2:

$$P_e = \left(\frac{1}{j}\right)^j = \left(\frac{1}{3}\right)^3 = 0.037$$

Finalmente, el CVC se calculó aplicando la ecuación 3.

$$CVC = CVC_i - P_e \quad [3]$$

Reemplazando los resultados obtenidos en la ecuación 1 y 2 y obtenemos nuestro coeficiente de validez de contenido.

$$CVC = CVC_i - P_e = 1 - 0.037 = 0.963$$

Respecto a la interpretación de nuestro valor obtenido, el referido autor indica que un CVC:

- menor que 0.60 tiene una validez inaceptable
- menor o igual a 0.70, tiene una validez deficiente
- menor o igual a 0.80, tiene una validez aceptable
- menor o igual a 0.90, tiene una validez buena
- mayor que 0.90, tiene una validez excelente

Por lo que concluimos que nuestro instrumento tiene una validez excelente, debido a que nuestro CVC obtenido, que es 0.963, es mayor que 0.90.

### **3.5. PROCESAMIENTO Y ANÁLISIS DE DATOS**

#### **Análisis de datos**

Para Horna (2012) “las técnicas de análisis de datos son herramientas útiles para organizar los datos recogidos con los instrumentos de investigación” (p.331).

Para la presente investigación, se empleó el software Apache Jmeter, “la navaja suiza” como le llama Medina (2014), para evaluar el desempeño de los sistemas gestores de base de datos, mediante el cual se pudo recabar los datos con nuestro instrumento. La selección de Apache JMeter fue hecha bajo el criterio de presentar un entorno amigable y gran capacidad de procesamiento.

Posteriormente, se usó del software Microsoft Excel y el software SPSS para organizar nuestros datos en cuadros y gráficos estadísticos.

### **Selección de pruebas estadísticas**

#### Análisis descriptivo de la variable de estudio

Para el tratamiento de los datos recabados con el instrumento, en este caso, la guía de observación, se utilizó el software estadístico SPSS que nos proporciona medidas de tendencia central y gráficos para observar el comportamiento de nuestros datos, así como también el software Microsoft Excel para realizar gráficos estadísticos en donde se agruparan los datos obtenidos por cada sistema gestor de base de datos para facilitar su comprensión.

#### Análisis inferencial de la variable de estudio

Al no cumplir las condiciones exigidas para la aplicación de pruebas paramétricas, para la realización de las pruebas de hipótesis se utilizó la prueba U de Mann-Whitney.

Esta prueba compara las medianas de dos muestras independientes que provienen de dos poblaciones iguales o de una misma población, cuando no se cumple el supuesto de normalidad (Ríos & Peña, 2020); esta prueba consiste en combinar las dos muestras, ordenando de manera ascendente y realizando una comparación de los promedios de las muestras en los datos de las muestras combinadas.

## Definición del ambiente de prueba

Todas las pruebas se realizaron bajo las mismas condiciones de hardware y disponibilidad de recursos, con la finalidad de no proporcionar ninguna clase de ventaja o desventaja a cualquiera de los sistemas gestores de base de datos sometidos a prueba.

Las pruebas se realizaron con las siguientes especificaciones técnicas del hardware y software para la simulación de las consultas a la base de datos por parte de los usuarios, de igual manera para el procedimiento de migración de la base de datos se utilizó el mismo software para ambos sistemas gestores de base de datos.

**Tabla 2**  
*Especificaciones del hardware*

	<b>Laptop</b>
<b>Modelo</b>	Dell G5 15 Gaming (5590) /RTX 2060
<b>Procesador</b>	Intel® Core™ i7-9750H 2.6GHz hasta 4.5GHz, 12MB de cache, 6 núcleos
<b>Memoria RAM</b>	16 GB DDR4 2400MHz
<b>Almacenamiento</b>	Sólido SSD 512GB PCIe M.2 (Unidad de estado sólido)
<b>Tarjeta de video</b>	NVIDIA® GeForce® RTX 2060, memoria de vídeo GDDR6 de 6 GB dedicados

Fuente: Especificaciones de laptop Dell G5

**Tabla 3**  
*Especificaciones del software*

<b>Software</b>	<b>Versión</b>	<b>Licencia</b>
<b>Sistema Operativo</b>	Windows 10 Home Single Language © 2020 Microsoft Corporation (64 bits)	Licencia commercial Microsoft CLUF (EULA) OEM
<b>Full Convert Enterprise</b>	Versión 19.01.1523	Licencia para estudiante
<b>Apache JMeter</b>	Versión 5.4.1	Licencia gratuita Apache license 2.0
<b>MySQL</b>	Versión 8.0.23	Licencia publica general/ Licencia commercial GNU GPL
<b>PostgreSQL</b>	Versión 13.2	Licencia gratuita BSD
<b>IBM SPSS software</b>	Versión 23.0	Propietaria
<b>Office 365</b>	Versión Education	Licencia para estudiante

Fuente: Características de los softwares utilizados

### **Caso de estudio**

Se trabajó con la base de datos del Sistema de registro de estudiantes matriculados en la UNJBG en el periodo 2019 conformada por 167 tablas con 3 431 970 registros; por consiguiente, se seleccionaron algunas tablas principales para el propósito de estudio. Para dicha selección se consideró el informe de utilización del espacio en disco por tablas de nuestra base de datos que nos proporciona el sistema gestor de bases de datos SQL Server 2012, a fin de aprovechar las tablas que consumen más espacio en disco.

Nombre de la tabla	Nº de registros	Reservado (KB)	Datos (KB)	Índices (KB)	No utilizado (KB)
dbo.ENC_EncuestaDetalle	1.589.035	86.920	86.552	352	16
dbo.OPE_Entidad	63.459	26.344	26.184	120	40
dbo.ENC_EncuestaIndividualDetalle	413.573	22.664	22.512	96	56
dbo.ACT_FichaDetalle_ReglaEvaluacion	379.481	20.808	20.672	104	32
dbo.MAT_Matricula	58.893	14.768	14.664	88	16
dbo.ACT_FichaDetalle	147.806	11.496	11.224	152	120
dbo.AST_AsistenciaDetalle	200.636	10.312	10.160	48	104
dbo.ADS_CentroEstudio	25.666	7.904	7.760	80	64
dbo.MAT_Ficha	23.884	7.464	7.288	128	48
dbo.ACT_FichaDetalle_Materia	26.619	5.608	5.488	120	0
dbo.DOC_Registro	33.765	5.464	5.392	80	0
dbo.ORH_AsistenciaDetalle	67.988	4.616	4.544	16	56
dbo.ENC_Encuesta	79.822	4.224	4.080	48	96
dbo.RES_Resolucion	30.956	3.384	3.280	48	56
dbo.MAT_Matricula_Resolucion	65.575	3.144	3.056	16	72
dbo.ACT_Acta	8.556	3.168	2.816	104	248
dbo.ADS_Materia	15.107	2.736	2.656	80	0
dbo.ACT_Firma	16.913	2.656	2.088	80	488
dbo.ACT_ReglaEvaluacion	20.525	2.120	2.072	16	32
dbo.PIT_CargaLectivaHorario	21.102	2.120	1.944	32	144
dbo.PIT_CargaLectiva	10.964	2.504	1.872	16	616
dbo.DOC_Firma	25.592	1.792	1.736	16	40
dbo.GTL_Graduado	5.657	1.792	1.736	16	40
dbo.PIT_CargaNoLectiva	5.154	1.544	1.336	16	192
dbo.ACT_Aula	7.046	1.568	1.312	72	184
dbo.PIT_Docente	4.975	1.376	1.248	80	48
dbo.AST_Asistencia	7.547	1.224	1.152	16	56
dbo.INV_ProyectoInvestigacion	922	1.104	1.040	16	48
dbo.GTL_Proyecto	1.331	1.096	1.024	16	56
dbo.ORH_Asistencia	4.163	840	776	16	48

Figura 5. Informe “uso de disco por tabla”

Fuente: Microsoft SQL Server 2012

Entre las tablas que conforman la base de datos existen tablas encargadas de solo almacenar documentos, como por ejemplo encuestas, se las puede identificar porque llevan las siglas ENC en el nombre de la tabla, así como también resoluciones, debido a ello no se consideraron ese tipo de tablas; luego se analizó en base a cantidad de registro y cantidad de datos (KB), por ejemplo, vemos que la tabla ACT\_FichaDetalle\_ReglaEvaluacion tiene 379 481 registros utilizando 20,6 KB mientras que la tabla MAT\_Matricula, con tan solo 58 893 registros utiliza 14,6 KB,

asimismo podemos ver que la tabla ACT\_ReglaEvaluacion tiene 20 525 registros y utiliza 2,0 KB mientras que la tabla ACT\_Acta con 8 556 registros utiliza 2,8 KB; es así que se seleccionó por uso de disco por tabla, debido a que no siempre la tabla con más registros es la más pesada o la que utiliza más espacio en disco.

A continuación, se muestra el esquema de la base de datos que se utilizó como caso de estudio, el cual, cuenta con 325 536 registros en 12 tablas.

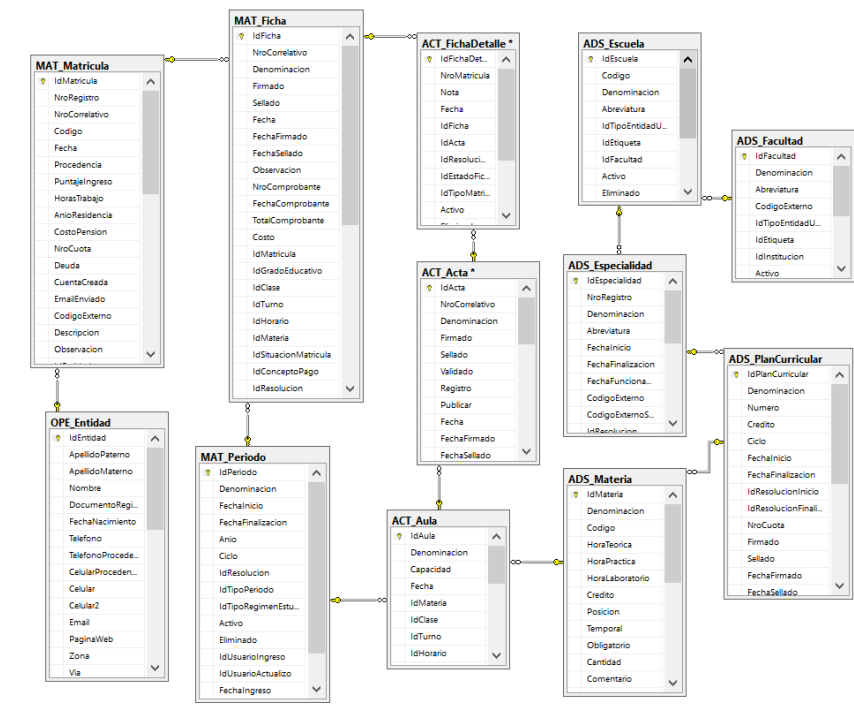


Figura 6. Esquema relacional de la base de datos

Fuente: DASA-UNJBG

## Migración de la base de datos

Se utilizó el software Full Convert Enterprise en su versión 19.01.1523 para realizar la migración de la base de datos de SQLSERVER a PostgreSQL y a MySQL, por ser un software externo y no uno propio de cada sistema gestor de base de datos a prueba.

Primero se seleccionó la base de datos de estudiantes matriculados que está en SQLServer, como base de datos de origen. El nombre de la base de datos es “SYSA\_UNJBG\_R4\_OLD”.

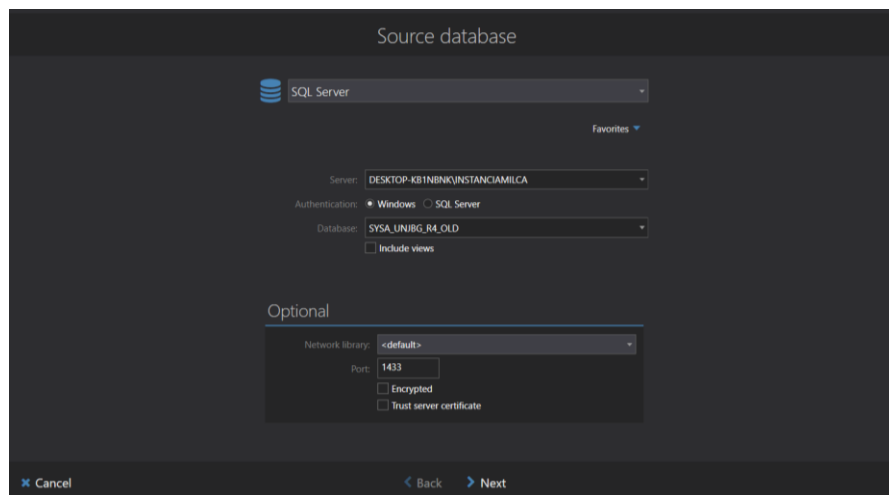


Figura 7. Configuración de la base de datos origen

Fuente: Elaboración propia

Luego se seleccionó la base de datos que se creó previamente como destino “BD\_UNJBG\_2019” en PostgreSQL y en MySQL.

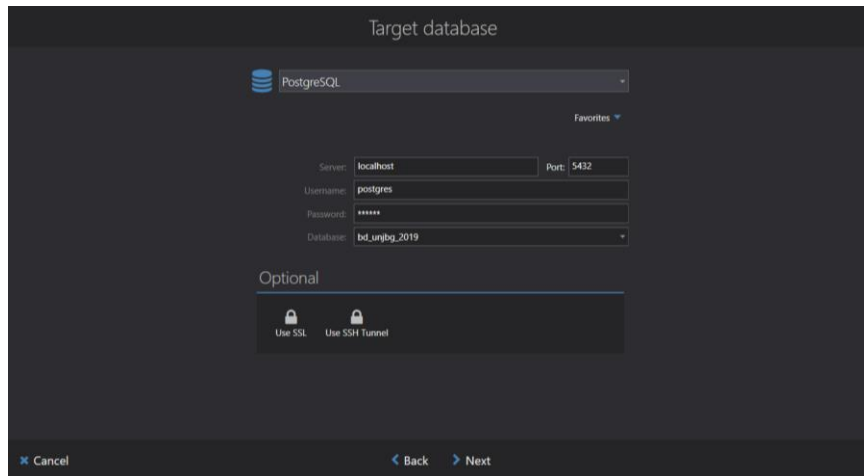


Figura 8. Configuración de la base de datos destino PostgreSQL

Fuente: Elaboración propia

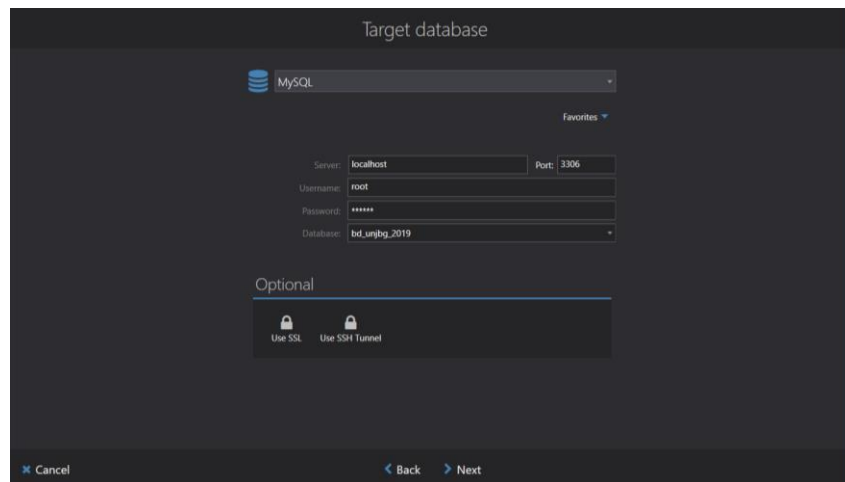


Figura 9. Configuración de la base de datos destino MySQL

Fuente: Elaboración propia

El software se encargó de la migración permitiendo seleccionar el conjunto de tablas referente a matriculas, con el cuál se trabajó, para ambos sistemas gestores.

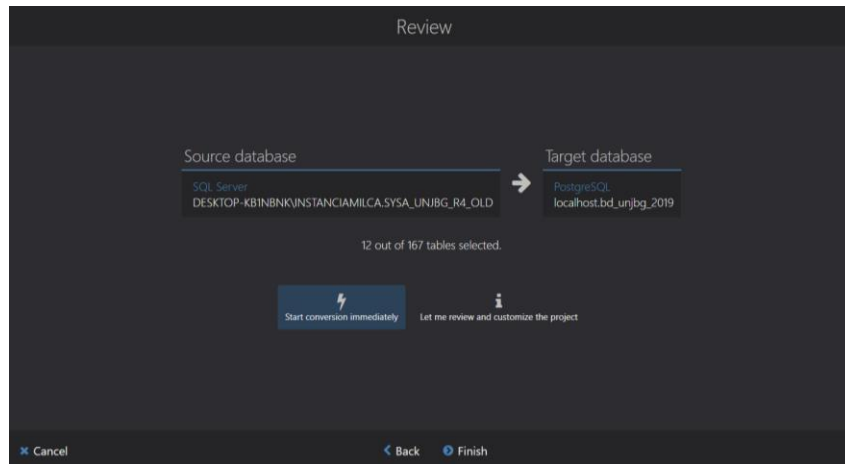


Figura 10. Migración de SQL Server a PostgreSQL

Fuente: Elaboración propia

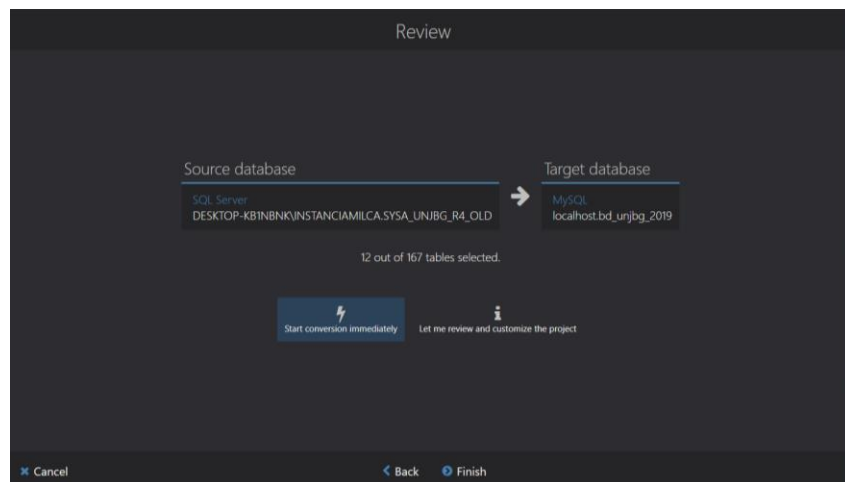


Figura 11. Migración de SQL Server a MySQL

Fuente: Elaboración propia

## Conexión de JMeter con los sistemas gestores de base de datos

Se utilizó la herramienta Apache JMeter en la versión 5.4.1 para realizar el plan de pruebas, para los distintos sistemas gestores de base de datos.

Se configuraron las conexiones JDBC para ambos sistemas gestores de base de datos por separado; se le asignó un nombre de variable para la conexión, se indicó la URL de la base de datos, se seleccionó la clase de driver JDBC correspondiente al sistema gestor de bases de datos y se indicó el usuario y contraseña de acceso a la base de datos.

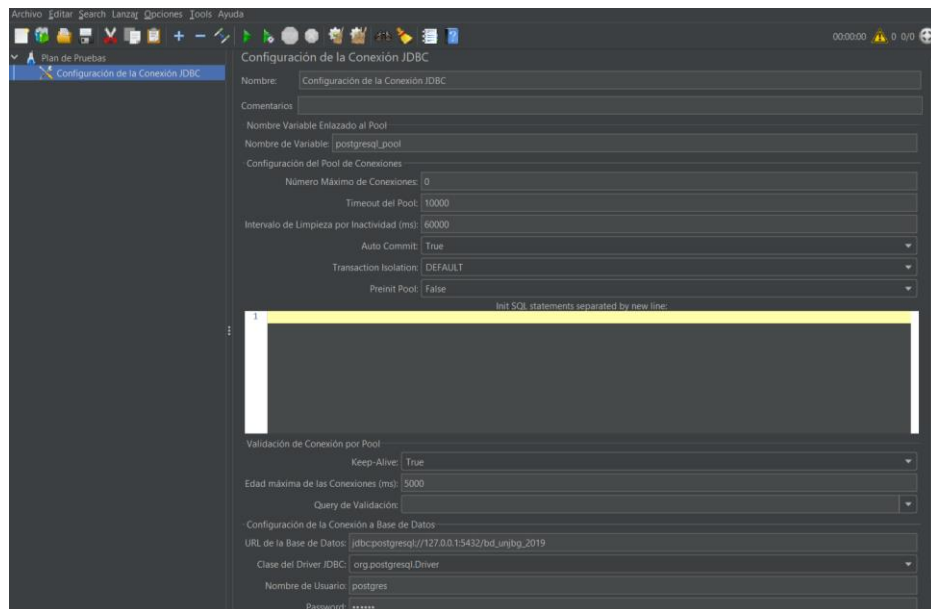


Figura 12. Configuración de la conexión con PostgreSQL

Fuente: Elaboración propia

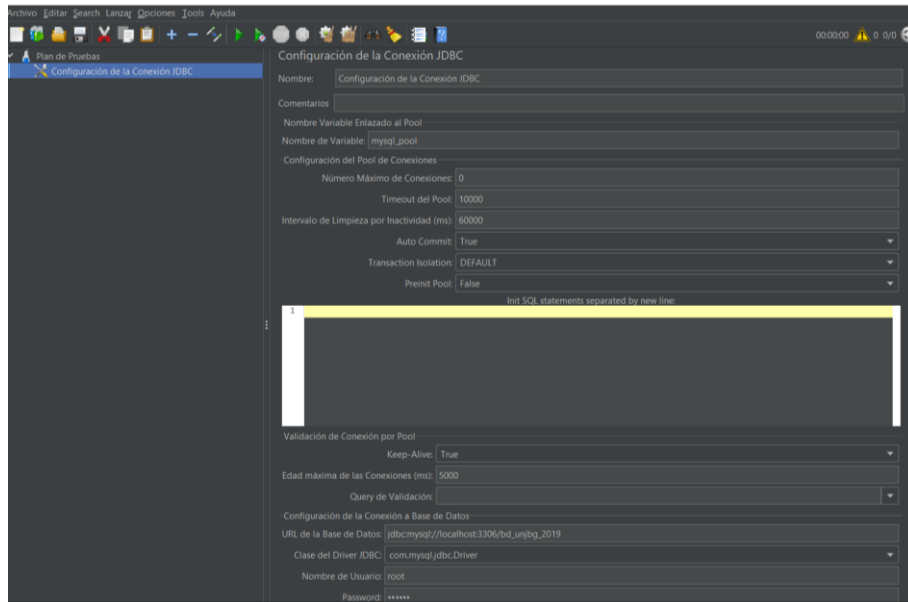


Figura 13. Configuración de la conexión con MySQL

Fuente: Elaboración propia

### Ejecución del plan de pruebas

Se realizó una prueba de carga, la cual consistió en elaborar 1 test que consta de 200 accesos a cada sistema gestor de base de datos, para ello se configuraron 10 threads, los cuales simularon 10 accesos de usuarios respectivamente que realizan 20 consultas diferentes a la base de datos de forma aleatoria con un periodo de aceleración de 2 segundos y 0,2 segundos de retraso entre los usuarios iniciales, esto quiere decir que, cada 0,2 segundos saldrá una consulta y al terminar los 2 segundos deberán acceder los 10 usuarios con 20 consultas cada uno logrando un total de 200 accesos a

cada sistema gestor de base de datos. El retraso entre usuarios se calcula mediante la ecuación 1:

$$\text{Retraso} = \text{aceleración} / \text{cantidad de usuarios} \quad [4]$$

Donde:

- Retraso: Es el tiempo que transcurre entre los usuarios cuando acceden a la base de datos, expresado en segundos.
- Aceleración: Es el tiempo en el que deben acceder todos los usuarios, expresado en segundos.
- Cantidad: Es el total de usuarios.

Esta prueba garantiza la estabilidad de ambos sistemas gestores, lo que quiere decir que ambos ejecutarán todas las peticiones y podremos obtener nuestros resultados sin presentar errores, ya que una prueba de carga no busca encontrar el punto de ruptura de un sistema, lo que si busca es evaluar el comportamiento de ambos sistemas gestores, que coincide con nuestro objetivo. A partir de ello, se recolectó los datos necesarios para luego ser interpretados.

Figura 14. Configuración del grupo de hilos para el plan de pruebas

Fuente: Elaboración propia

Muchos autores al momento de seleccionar las consultas en pruebas de rendimiento se basan en el nivel de complejidad, indicando que una consulta SQL simple es de nivel bajo, una consulta SQL con inner join es de nivel medio y una consulta SQL con, por ejemplo, order by o group by es de nivel alto. Considerando que se pretende hacer las pruebas en un ambiente real con datos académicos, las consultas que se realizan pertenecen a un nivel de complejidad alto, en ese sentido, evaluar consultas inferiores a ese nivel pierde sentido.

Para la selección de consultas en esta investigación nos basamos en el costo de plan de ejecución de las consultas; las consultas se diferencian en el costo debido a como fueron estructuradas, es así, que incluso podemos tener consultas que muestren

los mismos resultados, pero con diferente costo, para ello utilizamos el reporte que nos brinda SQLserver 2012, mediante el cual se seleccionaron consultas con diferente costo en el plan de ejecución.

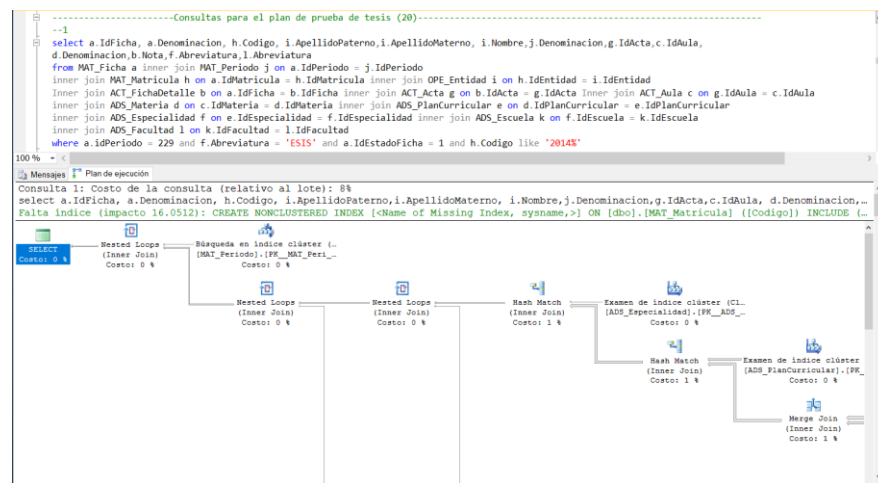


Figura 15. Reporte del costo de plan de ejecución

Fuente: SQL Server 2012

En la figura 15 podemos ver el costo de plan de ejecución de la consulta 1, la cual muestra la información de los alumnos matriculados en la escuela de Ingeniería en Informática y sistemas en el periodo 2019-I y que hayan ingresado en el 2014, este costo se da por porcentaje considerando las 20 consultas.

A continuación, se muestra un gráfico en el que se puede observar el costo por consulta, asimismo las 20 consultas SQL seleccionadas se encuentran en el anexo 4.

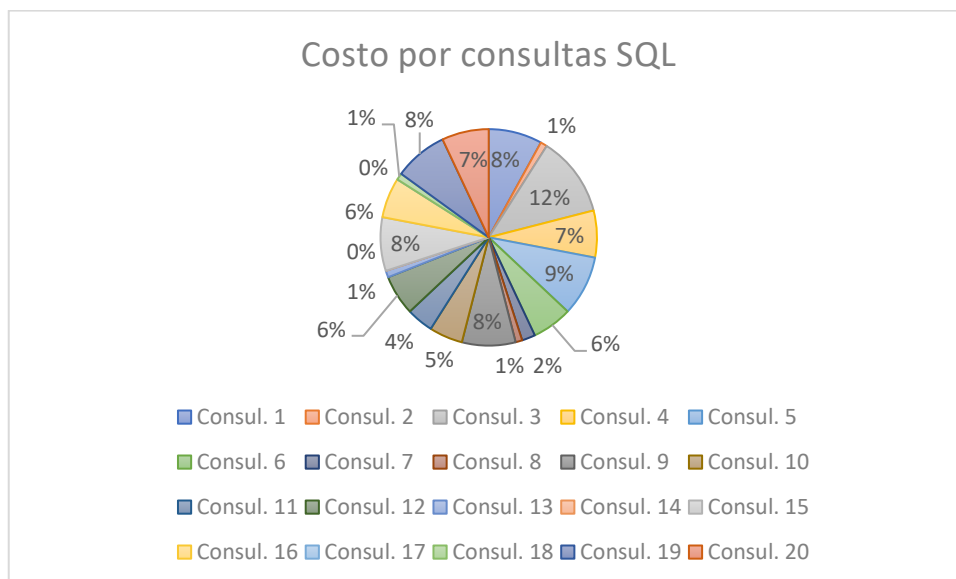


Figura 16. Costo de plan de ejecución de las consultas SQL

Fuente: Elaboración propia

Una vez definidas las 20 consultas se realizó la configuración de las peticiones JDBC para cada consulta, éstas se conectan con la “conexión JDBC”, que se configuró previamente, mediante una variable llamada “postgresql\_pool” y “mysql\_pool” para los diferentes Sistemas Gestores de Base de Datos. En las figuras 17 y 18 se muestra la configuración de la consulta 1 para ambos sistemas gestores.

Petición JDBC

Nombre: Petición JDBC para postgresql

Comentarios

Nombre de Variable Ligada al Pool

Nombre de Variable: postgresql\_pool

Query SQL

Solo Query: Select Statement

```

Query:
1 select a.IdFicha, a.Denominacion, h.Codigo, i.ApellidoPaterno, i.ApellidoMaterno, i.Nombre, j.Denominacion, g.IdActa, c.IdAula,
2 d.Denominacion, b.Nota, f.Abreviatura, l.Abreviatura
3 from MAT_Ficha a inner join MAT_Periodo j on a.IdPeriodo = j.IdPeriodo
4 inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula inner join OPE_Entidad i on h.IdEntidad = i.IdEntidad
5 inner join ACT_FichaDetalle b on a.IdFicha = b.IdFicha inner join ACT_Acta g on b.IdActa = g.IdActa
6 inner join ACT_Aula c on g.IdAula = c.IdAula
7 inner join ADS_Materia d on c.IdMateria = d.IdMateria inner join ADS_PlanCurricular e on d.IdPlanCurricular = e.IdPlanCurricular
8 inner join ADS_Especialidad f on e.IdEspecialidad = f.IdEspecialidad inner join ADS_Escuela k on f.IdEscuela = k.IdEscuela
9 inner join ADS_Facultad l on k.IdFacultad = l.IdFacultad
10 where a.IdPeriodo = 229 and f.Abreviatura = 'ESIS' and a.IdEstadoFicha = 1 and h.Codigo like '2014%'

```

Figura 17. Configuración de la petición JDBC para PostgreSQL

Fuente: Elaboración propia

Petición JDBC

Nombre: Petición JDBC para mysql

Comentarios

Nombre de Variable Ligada al Pool

Nombre de Variable: mysql\_pool

Query SQL

Solo Query: Select Statement

```

Query:
1 select a.IdFicha, a.Denominacion, h.Codigo, i.ApellidoPaterno, i.ApellidoMaterno, i.Nombre, j.Denominacion, g.IdActa, c.IdAula,
2 d.Denominacion, b.Nota, f.Abreviatura, l.Abreviatura
3 from bd_unjbg_2019.MAT_Ficha a inner join bd_unjbg_2019.MAT_Periodo j on a.IdPeriodo = j.IdPeriodo
4 inner join bd_unjbg_2019.MAT_Matricula h on a.IdMatricula = h.IdMatricula
5 inner join bd_unjbg_2019.OPE_Entidad i on h.IdEntidad = i.IdEntidad
6 inner join bd_unjbg_2019.ACT_FichaDetalle b on a.IdFicha = b.IdFicha inner join bd_unjbg_2019.ACT_Acta g on b.IdActa = g.IdActa
7 inner join bd_unjbg_2019.ACT_Aula c on g.IdAula = c.IdAula
8 inner join bd_unjbg_2019.ADS_Materia d on c.IdMateria = d.IdMateria
9 inner join bd_unjbg_2019.ADS_PlanCurricular e on d.IdPlanCurricular = e.IdPlanCurricular
10 inner join bd_unjbg_2019.ADS_Especialidad f on e.IdEspecialidad = f.IdEspecialidad
11 inner join bd_unjbg_2019.ADS_Escuela k on f.IdEscuela = k.IdEscuela
12 inner join bd_unjbg_2019.ADS_Facultad l on k.IdFacultad = l.IdFacultad
13 where a.IdPeriodo = 229 and f.Abreviatura = 'ESIS' and a.IdEstadoFicha = 1 and h.Codigo like '2014%'

```

Figura 18. Configuración de la petición JDBC para MySQL

Fuente: Elaboración propia

Finalmente, para poder visualizar los resultados y poder interpretarlos, necesitamos dos receptores o listeners que nos proporciona JMeter: **Reporte resumen**, para visualizar el uso de memoria promedio y media de los tiempos de respuesta para cada petición JDBC realizada y **Ver resultados en árbol**, para poder ver a detalle cada acceso, tanto el tiempo de respuesta como el uso de memoria.

## CAPÍTULO IV

### RESULTADOS Y DISCUSIÓN

#### 4.1. RESULTADOS

##### 4.1.1. Resultados a nivel descriptivo

A continuación, se presentan los resultados obtenidos para el desempeño de los sistemas gestores de base de datos, por cada uno de los indicadores: tiempo de respuesta y uso de la memoria.

##### **Resultados para el indicador 1: Tiempo de respuesta**

Con respecto al indicador tiempo de respuesta los resultados fueron los siguientes:

**Tabla 4**  
*Resultados del tiempo de respuesta para PostgreSQL*

<b>Accesos a la base de datos</b>	<b># Muestras</b>	<b>Media (ms)</b>	<b>Mín (ms)</b>	<b>Max (ms)</b>	<b>Desv. Estandar</b>	<b>% Error</b>
<b>Total</b>	200	198	1	1 204	266,38	0,00 %

Fuente: Resultados en JMeter de las pruebas ejecutadas

**Tabla 5**  
*Resultados del tiempo de respuesta para MySQL*

<b>Accesos a la base de datos</b>	<b># Muestras</b>	<b>Media (ms)</b>	<b>Mín (ms)</b>	<b>Max (ms)</b>	<b>Desv. Estandar</b>	<b>% Error</b>
<b>Total</b>	200	3 388	277	10 010	2 309,78	0,00 %

Fuente: Resultados en JMeter de las pruebas ejecutadas

En las tablas 4 y 5 se observa que el tiempo de respuesta promedio de los 200 accesos al sistema gestor de base de datos PostgreSQL es 198 ms y al sistema gestor de base de datos MySQL es 3 388 ms, asimismo observamos que el menor tiempo de respuesta de PostgreSQL es 1 ms y de MySQL es 277 ms obteniendo una diferencia de 276 ms, el mayor tiempo de respuesta de PostgreSQL es 1 204 ms y de MySQL es 10 010 ms obteniendo una diferencia de 8 806 ms; en promedio, la diferencia es de 3 190 ms, por lo que encontramos preliminarmente una diferencia significativa en los tiempos de respuesta a favor de PostgreSQL.

Podemos apreciar a detalle el comportamiento del tiempo de respuesta de todas las consultas realizadas a cada sistema gestor de bases de datos en la figura 19.

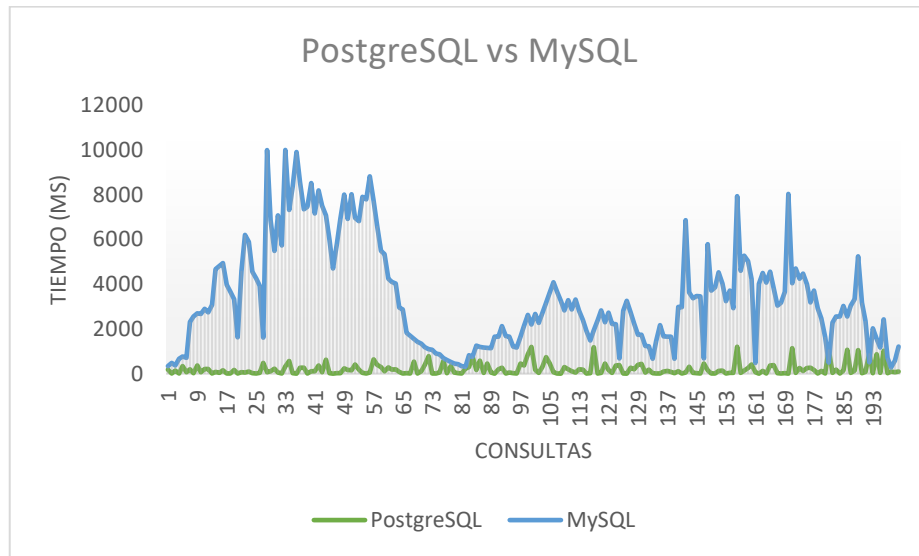


Figura 19. Gráfico comparativo con respecto al tiempo de respuesta

Fuente: Elaboración propia

Se puede apreciar en la figura 19 que los tiempos de respuesta por consulta expresados en milisegundos para ambos sistemas gestores de base de datos presentan picos, pero los picos más altos son los correspondientes al sistema gestor de base de datos MySQL, que presenta picos de hasta 10 009 ms. Sin embargo, si se normalizan los datos podemos ver un comportamiento favorable para MySQL.

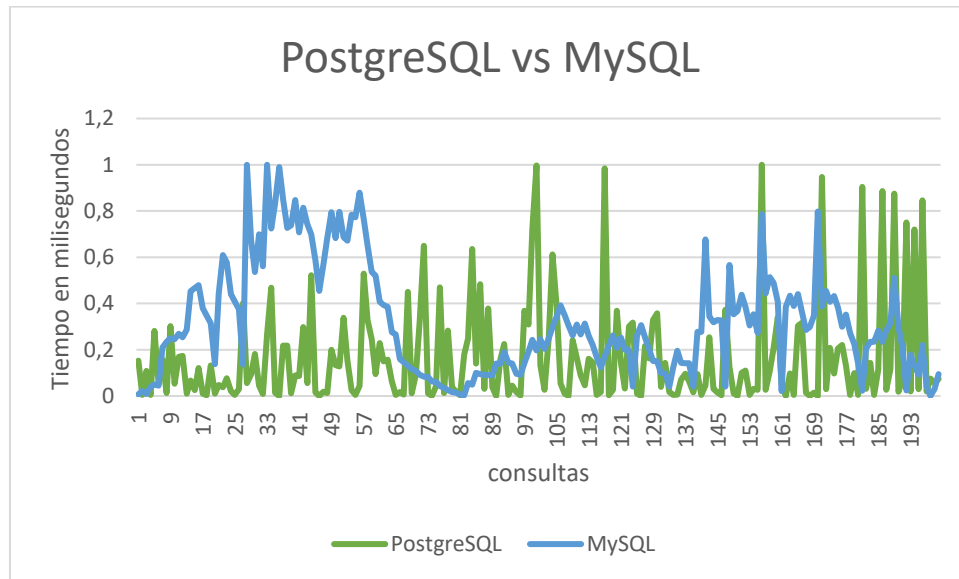


Figura 20. Gráfico comparativo del tiempo de respuesta con datos normalizados

Fuente: Elaboración propia

La figura 20 muestra los tiempos de respuesta normalizados; en la figura 19 se pudo ver que los tiempos de respuesta de PostgreSQL eran menores que los de MySQL, pero al normalizar los datos, se evidencia que PostgreSQL tiene muchos picos a comparación que MySQL, por lo que podemos decir que MySQL es más constante que PostgreSQL en cuanto al tiempo de respuesta por consulta.

Para la normalización de los datos se aplicó la ecuación 5 perteneciente a un método conocido como normalización mínima-máxima, esto con la finalidad de analizar el comportamiento de los datos sin importar su valor.

$$Z_i = (X_i - \min(X)) / (\max(X) - \min(X)) \quad [5]$$

Donde:

- $Z_i$ : El  $i$ -ésimo valor normalizado en el conjunto de datos
- $X_i$ : el  $i$ -ésimo valor en el conjunto de datos
- $\min(X)$ : el valor mínimo en el conjunto de datos
- $\max(X)$ : el valor máximo en el conjunto de datos

La normalización mínima – máxima es de los métodos más comunes de normalización, se escalan los datos al rango de 0 a 1, geoméricamente hablando, la transformación aplasta los datos  $n$ -dimensionales en un hipercubo unitario  $n$ -dimensional. Una vez los datos escalados podemos resaltar el comportamiento restándole importancia al valor de tiempo de respuesta que muestra cada SGBD.

### Resultados para el indicador 2: Uso de memoria

Con respecto al indicador uso de memoria los resultados fueron los siguientes:

**Tabla 6**  
*Resultados del uso de memoria para PostgreSQL*

Accesos a la base de datos	# Muestras	Memoria (bytes)
<b>Total</b>	200	1 385 436

Fuente: Resultados en JMeter de las pruebas ejecutadas

**Tabla 7**  
*Resultados del uso de memoria para MySQL*

<b>Accesos a la base de datos</b>	<b># Muestras</b>	<b>Memoria (bytes)</b>
<b>Total</b>	200	1 275 725,3

Fuente: Resultados en JMeter de las pruebas ejecutadas

En las tablas 6 y 7 se observa que el uso de memoria promedio de los 200 accesos al sistema gestor de base de datos PostgreSQL es 1 385 436 bytes y al sistema gestor de base de datos MySQL es 1 275 725,3 bytes, obteniendo una diferencia de 109 710,7 bytes, por lo que encontramos preliminarmente una diferencia significativa del uso de memoria favor de MySQL.

Podemos apreciar a detalle el comportamiento del uso de memoria de todas las consultas realizadas a cada sistema gestor de bases de datos en la figura 21.

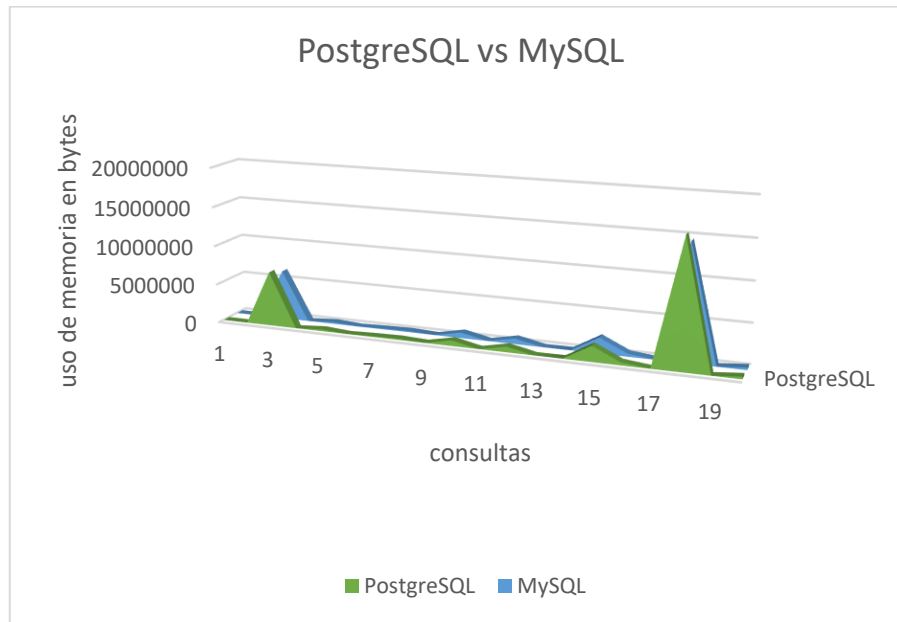


Figura 21. Gráfico comparativo con respecto al uso de memoria

Fuente: Elaboración propia

Se puede apreciar en la figura 21 que el comportamiento del uso de memoria expresado en bytes para el sistema gestor de bases de datos MySQL, es aparentemente el mismo que para el sistema gestor de bases de datos PostgreSQL, ambos presentan tres picos notorios, pero para el caso de PostgreSQL estos alcanzan un gran uso de memoria, llegando a utilizar 1 471 269 bytes más que MySQL en el pico más alto, en el pico mediano, la diferencia de uso de memoria es de 696 548 bytes y en el pico pequeño, la diferencia es de 0 bytes; si bien en la figura 21 aparentemente no hay una

diferencia significativa en el comportamiento, existe una diferencia en bytes a favor de MySQL.

#### **4.1.2. Resultados a nivel inferencial**

##### **Prueba de normalidad**

Se analizaron los datos con un nivel de confianza de 95 % y significancia de 5 % y se plantearon las siguientes hipótesis:

Ho: Los datos provienen de una distribución normal.

Hi: Los datos no provienen de una distribución normal.

Según Guisande González, Vaamonde Liste, & Barreiro Felpeto (2013) el test de Kolmogorov-Smirnov “es la prueba adecuada para testar la normalidad de una muestra si el número de datos es grande ( $n > 30$ ), aunque se puede usar tanto para muestras grandes como pequeñas”. (p.108)

En el caso del indicador tiempo de respuesta se utilizó la prueba de Kolmogorov-Smirnov, por ser la cantidad de valores mayor a 30.

- Si  $P_{valor} \geq \alpha$ , entonces se acepta Ho
- Si  $P_{valor} < \alpha$ , entonces se acepta Hi

Se aplicó la prueba y los resultados fueron los siguientes:

En las figuras 22 y 23 podemos observar la distribución de nuestros datos para el indicador tiempo de respuesta en un gráfico Q-Q normal para cada sistema gestor de base de datos.

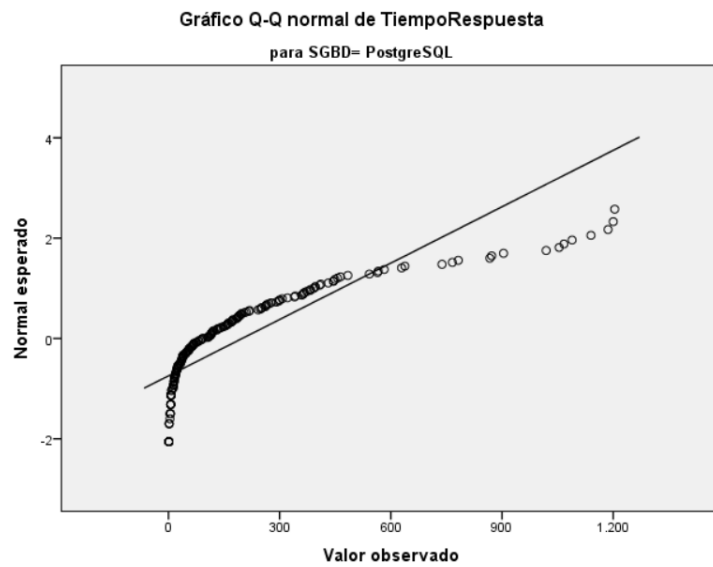


Figura 22. Distribución del tiempo de respuesta para PostgreSQL

Fuente: Elaboración propia

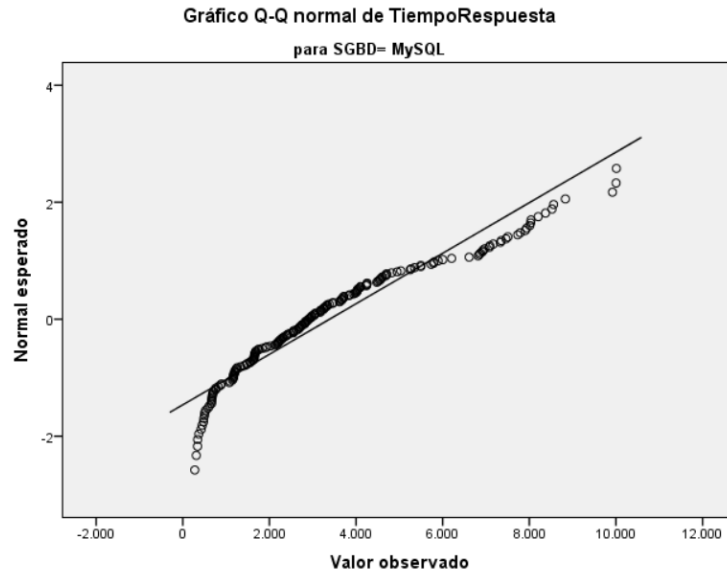


Figura 23. Distribución del tiempo de respuesta para MySQL

Fuente: Elaboración propia

En los gráficos 22 y 23 podemos observar que ambos sistemas gestores de base de datos no siguen la tendencia de normalidad esperada, debido a que muchos de los datos no se encuentran pegados a la tendencia normal que se encuentra representada con una línea diagonal; sin embargo, para el caso de MySQL es donde encontramos mayor dispersión de datos, asimismo en las tablas 4 y 5, anteriormente mostradas, obtuvimos la desviación estándar de cada sistema gestor, la cual apoya este comportamiento; ya que, PostgreSQL obtuvo 266,38 de desviación estándar y MySQL

2 309,78; como la desviación estándar de PostgreSQL es menor entonces la dispersión de los datos será menor.

Seguidamente obtenemos nuestro P-valor mediante el siguiente cuadro que nos proporciona SPSS aplicando la prueba de Kolmogorov-Smirnov.

Pruebas de normalidad				
SGBD		Kolmogorov-Smirnov <sup>a</sup>		
		Estadístico	gl	Sig.
TiempoRespuesta	PostgreSQL	,229	200	,000
	MySQL	,109	200	,000

a. Corrección de significación de Lilliefors

Figura 24. Prueba de normalidad para el tiempo de respuesta

Fuente: Elaboración propia

Mediante los resultados obtenidos en la prueba de normalidad de Kolmogorov, se concluye que al ser nuestro P-Valor = 0,000 <  $\alpha$  = 0,05, entonces se acepta  $H_1$ , entonces los datos no provienen de una distribución normal.

Según Guisande González et al. (2013) el test de Shapiro-Wilk “es la prueba más recomendable para testar la normalidad de una muestra, sobre todo si se trabaja con un número pequeño de datos ( $n < 30$ )”. (p.108)

En el caso del indicador uso de memoria se utilizó la prueba de Shapiro-Wilks, por ser la cantidad de valores menor a 30.

- Si  $P_{valor} \geq \alpha$ , entonces se acepta  $H_0$
- Si  $P_{valor} < \alpha$ , entonces se acepta  $H_1$

Se aplicó la prueba y los resultados fueron los siguientes:

Para poder observar la distribución de nuestros datos para el indicador uso de memoria a continuación se muestra un gráfico Q-Q normal para cada sistema gestor de base de datos.

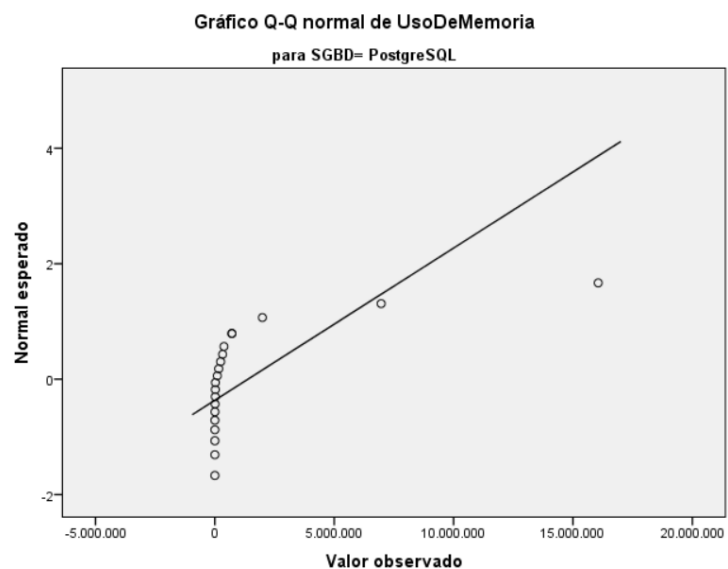


Figura 25. Distribución del uso de memoria para PostgreSQL

Fuente: Elaboración propia

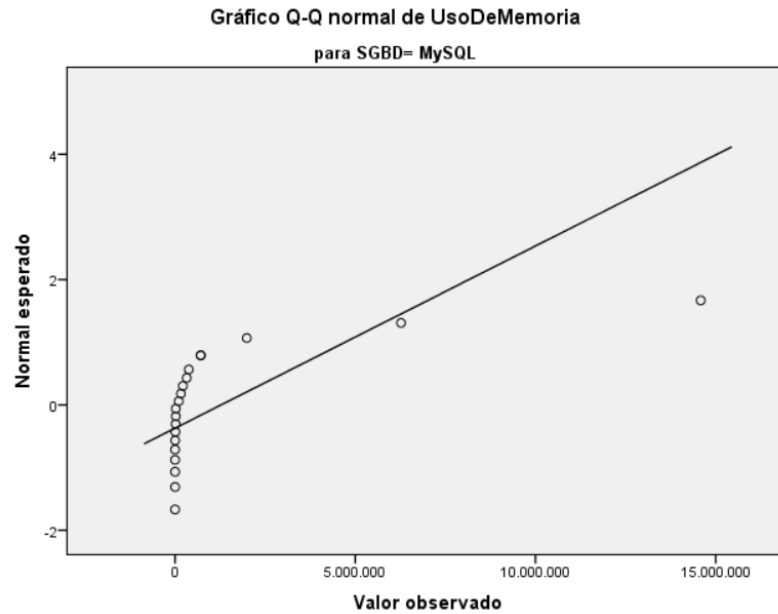


Figura 26. Distribución del uso de memoria para MySQL

Fuente: Elaboración propia

En las figuras 25 y 26 podemos observar que ambos sistemas gestores de base de datos no siguen la tendencia de normalidad esperada, debido a que muchos de los datos no se encuentran pegados a la tendencia normal que se encuentra representada con una línea diagonal; aparentemente las figuras son iguales porque el uso de memoria de ambos sistemas gestores es similar encontrando una diferencia mínima de bytes, para el indicador uso de memoria la dispersión de datos será similar.

Seguidamente obtenemos nuestro P-valor mediante el siguiente cuadro que nos proporciona SPSS aplicando la prueba de Shapiro-Wilks.

Pruebas de normalidad				
SGBD		Shapiro-Wilk		
		Estadístico	gl	Sig.
UsoDeMemoria	PostgreSQL	,421	20	,000
	MySQL	,427	20	,000

a. Corrección de significación de Lilliefors

Figura 27. Prueba de normalidad para el uso de memoria

Fuente: Elaboración propia

Mediante los resultados obtenidos en la prueba de normalidad de Shapiro, se concluye que al ser nuestro P-Valor = 0,000 <  $\alpha$  = 0,05, entonces se acepta  $H_1$ , entonces los datos no provienen de una distribución normal.

### Prueba de hipótesis específica 1

**$H_0$ :** El tiempo de respuesta del sistema gestor de base de datos PostgreSQL es mayor o igual que el sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann – 2019.

**$H_1$ :** El tiempo de respuesta del sistema gestor de base de datos PostgreSQL es menor que el sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019.

Para esta prueba se estableció un nivel de significancia  $\alpha = 0,05$  y como nuestros datos no siguen una distribución normal se utilizó el estadístico de prueba U de Mann-Whitney para comparar medianas, que ya fue definida en la parte de selección de pruebas estadísticas de la presente investigación.

Aplicamos la prueba y obtenemos nuestro P-valor mediante el siguiente cuadro que nos proporciona SPSS.

**Estadísticos de prueba<sup>a</sup>**

	TiempoResp uesta
U de Mann-Whitney	584,000
W de Wilcoxon	20684,000
Z	-16,794
Sig. asintótica (bilateral)	,000

a. Variable de agrupación: SGBD

Figura 28. Prueba U de Mann-Whitney para el tiempo de respuesta

Fuente: Elaboración propia

Según los resultados obtenidos en la prueba U de Mann-Whitney, se concluye que al ser nuestro P-Valor =  $0,000 < \alpha = 0,05$ , entonces se acepta  $H_1$ , por lo tanto, el tiempo de respuesta del sistema gestor de base de datos PostgreSQL es menor que el

sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019.

### **Prueba de hipótesis específica 2**

**Ho:** El uso de memoria del sistema gestor de base de datos PostgreSQL es mayor o igual que el sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019.

**Hi:** El uso de memoria del sistema gestor de base de datos PostgreSQL es menor que el sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019.

Para esta prueba se estableció un nivel de significancia  $\alpha = 0,05$  y como nuestros datos no siguen una distribución normal se utilizó el estadístico de prueba U de Mann-Whitney para comparar medianas, que ya fue definida en la parte de selección de pruebas estadísticas de la presente investigación.

Aplicamos la prueba y obtenemos nuestro P-valor mediante el siguiente cuadro que nos proporciona SPSS.

**Estadísticos de prueba<sup>a</sup>**

	UsoDeMemoria
U de Mann-Whitney	198,000
W de Wilcoxon	408,000
Z	-,054
Sig. asintótica (bilateral)	,957

a. Variable de agrupación: SGBD

Figura 29. Prueba U de Mann-Whitney para el uso de memoria

Fuente: Elaboración propia

Según los resultados obtenidos en la prueba U de Mann-Whitney, se concluye que al ser nuestro P-Valor =  $0,957 > \alpha = 0,05$ , entonces se acepta  $H_0$ , por lo tanto, el uso de memoria del sistema gestor de base de datos PostgreSQL es mayor o igual que el sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019.

## 4.2. DISCUSIÓN

Según los resultados obtenidos en la presente investigación el sistema gestor de base de datos PostgreSQL ejecutó 200 accesos a nuestra base de datos de 325 536 registros en 12 tablas obteniendo un tiempo de respuesta promedio de 198 milisegundos, mientras que MySQL realizó la misma transacción bajo las mismas condiciones obteniendo un tiempo de respuesta promedio de 3 388 milisegundos; sin embargo, el uso de memoria utilizada por PostgreSQL es de 1 385 436 bytes y de MySQL es de 1 275 725,3 bytes; podemos determinar que si bien, PostgreSQL nos ofrece menor tiempo de respuesta, realiza mayor uso de memoria a diferencia que MySQL, que nos ofrece menor uso de memoria pero mayor tiempo de respuesta.

Encontramos un comportamiento similar en la investigación comparativa realizada por Cambi & Zuñiga (2006), en la cual observamos que PostgreSQL muestra un tiempo de respuesta de 110 milisegundos con uso de memoria de 210 148 000 bytes, mientras que MySQL tiene un tiempo de respuesta de 356,2 milisegundos y uso de memoria de 200 844 000 bytes para 9000 registros; de igual manera, PostgreSQL tiene menor tiempo de respuesta, pero mayor uso de memoria y MySQL menor uso de memoria, pero mayor tiempo de respuesta. Vemos que ambos SGBDs en nuestro estudio realizaron menor uso de memoria a comparación del estudio realizado por Cambi & Zuñiga, y también que mostraron mayor tiempo de respuesta, uno de los

factores que intervienen es la cantidad de registros en que nuestros SGBDs a prueba tienen que buscar por ello ocupan mayor tiempo de respuesta.

No obstante, nuestros resultados difieren parcialmente con los resultados de la investigación realizada por León (2020), en la cual se puede evidenciar que, en el tiempo de respuesta para 100 000 registros, PostgreSQL obtuvo 67 milisegundos con un consumo de memoria de 55 086 359,27 bytes, mientras que MySQL obtuvo 435 milisegundos con un consumo de memoria de 808 366 080 bytes, en esta investigación PostgreSQL ofrece un menor tiempo de respuesta y menor uso de memoria respecto a MySQL. En este estudio a comparación del nuestro, vemos que sus resultados muestran menor tiempo de respuesta que nuestros resultados y también que de los tres estudios comparados, el nuestro es el que muestra menor uso de memoria por parte de ambos sistemas gestores de base de datos, otro factor importante que interviene en este tipo de investigaciones es el hardware que se utiliza para la ejecución de las pruebas realizadas, ya que, si utilizamos un equipo mucho más eficiente en cuanto a aceleración de procesos y manejo de recursos, obtendremos mejores resultados.

Por otra parte, podemos contrastar parte de nuestros resultados con más investigaciones recientes, como la investigación realizada por Klimek, B., & Skublewska-Paszkowska, M. (2021), donde podemos comprobar que el comportamiento de ambos SGBD es el mismo en cuanto al tiempo de respuesta, ya que

en esta investigación los autores concluyen que PostgreSQL es más eficiente; esta investigación está enfocada en evaluar el rendimiento en base a solo la velocidad de respuesta con 1000 consultas pero con pequeñas cantidades de registros; de igual manera otra investigación que solo considera el indicador tiempo de respuesta para evaluar el desempeño de los SGBD es la realizada por Praba, A. D., & Safitri, M. (2020), la cual está orientada a realizar pruebas en base a tres consultas `select*from`, `select count()` y `join table`, si bien nuestra investigación consideró más tipos de consultas y un indicador adicional para evaluar el desempeño, se puede realizar un contraste con nuestros resultados, MySQL resultó ser más eficiente para este tipo de consultas básicas, sin embargo en nuestra investigación se consideró consultas con mayor costo de ejecución para lo cual resultó PostgreSQL más eficiente.

Esto demuestra que no hay evidencia suficiente para rechazar nuestra hipótesis nula, el desempeño del SGBD PostgreSQL no es significativamente más eficiente que el desempeño del SGBD MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann – 2019 en cuanto al tiempo de respuesta y uso de memoria.

## CONCLUSIONES

### **Primero**

Se determinó que el tiempo de respuesta de consultas realizadas a la base de datos de matrículas de la Universidad Nacional Jorge Basadre Grohmann - 2019 utilizando el sistema gestor de base de datos PostgreSQL es menor que utilizando el sistema gestor de base de datos MySQL. PostgreSQL ejecutó 200 accesos a nuestra base de datos de 325 536 registros en 12 tablas obteniendo un tiempo de respuesta promedio de 198 milisegundos, mientras que MySQL realizó la misma transacción bajo las mismas condiciones obteniendo un tiempo de respuesta promedio de 3 388 milisegundos, 17 veces más el tiempo que hizo PostgreSQL, evidenciando que para este indicador PostgreSQL tiene mejor desempeño que MySQL gracias a su velocidad de búsqueda.

### **Segundo**

Se pudo comprobar que el uso de memoria al realizar las consultas a la base de datos de matrículas de la Universidad Nacional Jorge Basadre Grohmann - 2019 utilizando el sistema gestor de base de datos PostgreSQL es mayor que utilizando el sistema gestor de base de datos MySQL. PostgreSQL ejecutó 200 accesos a nuestra base de datos de 325 536 registros en 12 tablas utilizando 1 385 436 bytes de memoria

y MySQL realizó la misma transacción bajo las mismas condiciones utilizando 1 275 725,3 bytes de memoria, si bien la diferencia no es ni el doble del uso que realiza MySQL como en el caso del primer indicador que la diferencia era de 17 veces más, en este caso existe una diferencia de 109 710,7 bytes a favor de MySQL por lo que para este indicador se evidencia que MySQL tiene un mejor desempeño que PostgreSQL gracias a su manejo de recursos, en este caso, memoria.

### **Tercero**

A nivel inferencial se concluye que el desempeño del sistema gestor de base de datos PostgreSQL no es significativamente más eficiente que el desempeño del sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann – 2019 en cuanto al tiempo de respuesta y uso de memoria.

Al no ser significativamente más eficiente el desempeño de uno u otro, podemos decir que ambos sistemas gestores de base de datos muestran un desempeño eficiente, por un lado, PostgreSQL muestra mayor velocidad al atender las consultas y por otro lado MySQL muestra un mejor manejo en el uso de memoria al atender las consultas, queda en la empresa priorizar el tiempo de respuesta o el uso de memoria, y en base a ello poder decidir por uno u otro.

## **RECOMENDACIONES**

En futuros trabajos de investigación se sugiere comparar el desempeño de los dos sistemas gestores de base de datos estudiados en el presente trabajo de investigación, considerando otros indicadores, para una mejor referencia sobre cuál de estos dos sistemas gestores de base de datos es la mejor elección para las necesidades de alguna empresa. Cabe señalar que el tiempo de respuesta y uso de memoria dependen de muchos factores como el hardware que se esté utilizando, el tipo de información, el modelo relacional, etc. Es por ello que en la presente investigación se detallan las condiciones del plan de prueba y caso de estudio.

Finalmente en futuras investigaciones se pueden realizar las pruebas con consultas que impliquen actualización, que dependen de la configuración del sistema gestor de base de datos, algunos sistemas gestores de base de datos permiten la lectura de tablas en las que se está realizando una actualización y en otras no, por ello las consultas que implican actualización, como inserción, eliminación y edición, para un análisis profundo, requieren de otro plan de prueba que permita controlar los accesos a una tabla específica.

## REFERENCIAS BIBLIOGRÁFICAS

- Arias, F. G. (2012). *El Proyecto de Investigación. Introducción a la Metodología Científica. 6ta. Edición.* Fidas G. Arias Odón.
- Arribas, M. (2004). *Diseño y validación de cuestionarios.* 5(2004), 7.
- Asiminidis, C., Kokkonis, G., & Kontogiannis, S. (2018). *Database Systems Performance Evaluation for IoT Applications* (SSRN Scholarly Paper N.º ID 3360886). Rochester, NY: Social Science Research Network. <https://doi.org/10.2139/ssrn.3360886>
- Bassil, Y. (2012). A Comparative Study on the Performance of the Top DBMS Systems. *arXiv:1205.2889 [cs]*. Recuperado de <http://arxiv.org/abs/1205.2889>
- Cabero Almenara, J., & Llorente Cejudo, M. del C. (2013). *La Aplicación del Juicio de Experto como Técnica de Evaluación de las Tecnologías de la Información y Comunicación (TIC).* 12.
- Cambi Alvarado, J. D., & Zúñiga García, J. A. (2006). *Comparación entre MySQL vs. PostgreSQL.* Recuperado de <http://dspace.uazuay.edu.ec/handle/datos/2169>
- Díaz, J., Tzancoff Banchoff, C., Rodríguez, A., y Soria, V. (2008). *Usando Jmeter para pruebas de rendimiento.* La Plata. Repositorio Institucional de la Universidad de la Plata. <http://sedici.unlp.edu.ar/handle/10915/21907>

- Elmasri, R., Díaz Martín, J. M., & Navathe, S. B. (2011). *Fundamentos de sistemas de bases de datos*. Madrid [etc.: Pearson Educación.
- Espinoza, R. C., y Sarango, B. C. (2013). Análisis comparativo entre bases de datos relacionales con bases de datos no relacionales. Repositorio Institucional de la Universidad Politécnica Salesiana. <https://dspace.ups.edu.ec/handle/123456789/6977>
- Gilfillan, I. (2003). *La Biblia de MySQL*. Madrid: Anaya Multimedia.
- Guisande González, C., Vaamonde Liste, A., & Barreiro Felpeto, A. (2013). *Tratamiento de datos con R, STATISTICA y SPSS*. España: Díaz de Santos.
- Hernández Nieto, H. (2002). *Contributions to statistical analysis: Sensibility (estability and consistency) of different coefficients of relative variance and the coefficient of proportional variance (Cvp) : the coefficient of content validity (Ccv) and the Kappa coefficient (K), to determine content validity according to the technique of experts panel*. Mérida, Venezuela: Universidad de Los Andes.
- Hernández Sampieri, R., Fernández Collado, C., & Pilar Baptista Lucio, M. (2014). *Metodología de la investigación*. México: McGraw-Hill.
- Horna, A. A. V. (2012). *Siete pasos para una tesis exitosa: Desde la idea inicial hasta la sustentación*. Recuperado de

[https://www.academia.edu/18635649/Siete\\_pasos\\_para\\_una\\_tesis\\_exitosa\\_Desde\\_la\\_idea\\_inicial\\_hasta\\_la\\_sustentacion](https://www.academia.edu/18635649/Siete_pasos_para_una_tesis_exitosa_Desde_la_idea_inicial_hasta_la_sustentacion)

Hueso Ibáñez, L. (2015). *Administración de sistemas gestores de bases de datos (2a.ed.)*. Madrid: RA-MA Editorial. Recuperado de <https://public.ebookcentral.proquest.com/choice/publicfullrecord.aspx?p=5758898>

Khan, Z. (2006). *COMPARISON OF SELECTION CRITERIA OF OPEN SOURCE DBMS AGAINST PROPRIETARY DBMS*. Recuperado de <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.121.5476&rep=rep1&type=pdf>

Klimek, B., & Skublewska-Paszkowska, M. (2021). *Comparison of the performance of relational databases PostgreSQL and MySQL for desktop application*. Journal of Computer Sciences Institute, 18, 61-66.

Kolonko, K. (2018). Performance comparison of the most popular relational and non-relational database management systems. *Digitala Vetenskapliga Arkivet* <https://www.divaportal.org/smash/record.jsf?pid=diva2%3A1199667&dsid=4694>

- Lardière, S. (2019). *PostgreSQL. Administración y explotación de sus bases de datos*. Cornellà de Llobregat (Barcelona).
- López Herrera, P. (2016). *Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL*. Recuperado de <http://repositorioslatinoamericanos.uchile.cl/handle/2250/991352>
- Medina, J. (2014). *Pruebas de Rendimiento TIC*. Lulu.com.
- Palaquibay, Q., & Azucena, P. (2010). *Diseño e Implementación de la Arquitectura de Datos Basada en Comparativas de Rendimiento entre Sistemas de Gestion de Bases de Datos*. Recuperado de <http://dspace.esPOCH.edu.ec/handle/123456789/97>
- Poll, R., Boekhorst, P. te, & Abad Hiraldo, R. (1998). *Medición de calidad: Directrices para la medición del rendimiento en las bibliotecas universitarias*. Madrid: ANABAD.
- Praba, A. D., & Safitri, M. (2020). *Studi Perbandingan Performansi Antara Mysql Dan Postgresql*. *Jurnal Khatulistiwa Informatika*, 8(2).
- Quizhpe Palaquibay, P. A. (2010). *Diseño e Implementación de la Arquitectura de Datos Basada en Comparativas de Rendimiento entre Sistemas de Gestión de Bases de Datos (Bachelor's thesis)*. Repositorio Institucional de la Escuela

Superior                      Politécnica                      de                      Chimborazi.

<http://dspace.esPOCH.edu.ec/handle/123456789/97>

Ramakrishnan, R., & Gehrke, J. (2003). *Database management systems* (3rd ed). Boston: McGraw-Hill.

Ríos, A. R., & Peña, A. M. P. (2020). Estadística inferencial. Elección de una prueba estadística no paramétrica en investigación científica. *Horizonte de la Ciencia*, 10(19), 191-208. <https://doi.org/10.26490/uncp.horizonteciencia.2020.19.597>

Robles Garrote, P., & Rojas, M. D. C. (2015). La validación por juicio de expertos: Dos investigaciones cualitativas en Lingüística aplicada. Recuperado 28 de agosto de 2021, de <https://www.nebrija.com/revista-linguistica/la-validacion-por-juicio-de-expertos-dos-investigaciones-cualitativas-en-linguistica-aplicada.html>

Romero, A. C., Sanabria, J. S. G., & Cuervo, M. C. (2012). Utilidad y funcionamiento de las bases de datos NoSQL. *Facultad de Ingeniería*, 21(33), 21-32.

Silberschatz, A., Korth, H. F., & Sudarshan, S. (2002). *Database system concepts* (4th ed). Boston: McGraw-Hill.

- Soberón León, J. J. (2020). Análisis comparativo de sistemas gestores de bases de datos postgresql y mysql en procesos crud. *Repositorio Institucional - USS*. Recuperado de <http://repositorio.uss.edu.pe//handle/20.500.12802/7012>
- Suri P. y Sharma M, (2011). A comparative study between the Performance of relational and Object oriented database in data warehousing. *International Journal of Database Management Systems*. <https://d1wqtxts1xzle7.cloudfront.net/38399000>
- Vanegas, C. A. (2005). JDBC: EL PUENTE ENTRE JAVA Y LAS BASES DE DATOS. *Revista vínculos*, 2(1), 89-104. <https://doi.org/10.14483/2322939X.4066>
- Wodyk, R., & Skublewska-Paszkowska, M. (2020). *Performance comparison of relational databases SQL Server, MySQL and PostgreSQL using a web application and the Laravel framework*. Journal of Computer Sciences Institute, 17, 358-364.
- Yang, X. (2011). Analysis of DBMS: MySQL Vs PostgreSQL [Fi=AMK-opinnäytetyö|sv=YH-examensarbete|en=Bachelor's thesis]. Recuperado 7 de julio de 2021, de <http://www.theseus.fi/handle/10024/27471>

# **ANEXOS**

## ANEXO I MATRIZ DE CONSISTENCIA

<b>TITULO: ANÁLISIS COMPARATIVO DEL DESEMPEÑO DE LOS SISTEMAS GESTORES DE BASES DE DATOS: POSTGRESQL Y MYSQL CON LOS DATOS ACADÉMICOS DE LA UNIVERSIDAD NACIONAL JORGE BASADRE GROHMANN – 2019</b>			
<b>PROBLEMA</b>	<b>OBJETIVOS</b>	<b>HIPOTESIS GENERAL</b>	<b>VARIABLES E INDICADORES</b>
<p><b>PROBLEMA GENERAL:</b></p> <p>¿Cómo es el desempeño de los sistemas gestores de base de datos PostgreSQL y MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019?</p> <p><b>PROBLEMAS ESPECIFICOS:</b></p> <p>¿Cómo es el desempeño en cuanto al tiempo de respuesta de los sistemas gestores de base de datos PostgreSQL y MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019?</p> <p>¿Cómo es el desempeño en cuanto al uso de memoria RAM de los sistemas gestores de base de datos PostgreSQL y MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019?</p>	<p><b>OBJETIVO GENERAL:</b></p> <p>Comparar el desempeño de los sistemas gestores de base de datos PostgreSQL y MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019</p> <p><b>OBJETIVOS ESPECIFICOS:</b></p> <p>Comparar el tiempo de respuesta de los sistemas gestores de base de datos PostgreSQL y MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019</p> <p>Comparar el uso de memoria de los sistemas gestores de base de datos PostgreSQL y MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019.</p>	<p><b>HIPOTESIS GENERAL:</b></p> <p>El desempeño del sistema gestor de base de datos PostgreSQL es significativamente más eficiente que el desempeño del sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann – 2019.</p> <p><b>HIPOTESIS ESPECIFICAS:</b></p> <p>El tiempo de respuesta del sistema gestor de base de datos PostgreSQL es menor que el sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann – 2019.</p> <p>El uso de memoria del sistema gestor de base de datos PostgreSQL es menor que el sistema gestor de base de datos MySQL utilizando los datos académicos de la Universidad Nacional Jorge Basadre Grohmann - 2019.</p>	<p><b>VARIABLE:</b></p> <p>El desempeño de los sistemas gestores de base de datos PostgreSQL y MySQL.</p> <p><b>INDICADORES:</b></p> <p>Tiempo de respuesta. Uso de memoria.</p>
<b>METODO Y DISEÑO</b>	<b>POBLACION Y MUESTRA</b>	<b>TECNICAS E INSTRUMENTOS</b>	
<p><b>METODO:</b></p> <p>Descriptivo-comparativa</p> <p><b>DISEÑO:</b></p> <p>No Experimental</p>	<p><b>POBLACION:</b></p> <p>3 431 970 registros de 253 230 estudiantes matriculados en la Universidad Nacional Jorge Basadre Grohmann en el periodo 1972 -2019.</p> <p><b>MUESTRA:</b></p> <p>325 536 registros de 8 608 estudiantes matriculados en la Universidad Nacional Jorge Basadre Grohmann en el periodo 2019.</p>	<p><b>TECNICAS:</b></p> <p>Observación</p> <p><b>TRATAMIENTO ESTADISTICO:</b></p> <p>Prueba U de Mann Whitney</p>	<p><b>INSTRUMENTO:</b></p> <p>Guia de observación</p>

## ANEXO II GUIA DE OBSERVACIÓN

En la presente guía de observación se registró los tiempos de respuesta y uso de memoria de los accesos a la base de datos de matrículas en PostgreSQL y MySQL. Estos accesos corresponden a las diferentes consultas SQL seleccionadas para nuestro plan de prueba, las cuales son 20 y se ejecutarán 10 veces cada una de manera aleatoria, obteniendo así, 200 accesos a la base de datos.

Accesos a la base de datos	SGBD PostgreSQL		SGBD MySQL	
	Tiempo de respuesta (ms)	Uso de memoria (bytes)	Tiempo de respuesta (ms)	Uso de memoria (bytes)
Acceso 1	185	18252	344	18252
Acceso 2	7	57	484	18252
Acceso 3	132	18252	369	72
Acceso 4	6	57	662	18252
Acceso 5	340	6965544	761	18252
Acceso 6	63	380	711	72
Acceso 7	191	18252	2319	6965544
Acceso 8	16	57	2562	18252
Acceso 9	365	6965544	2702	18252
Acceso 10	65	380	2666	72
Acceso 11	205	240724	2897	18252
Acceso 12	210	18252	2750	72
Acceso 13	13	57	3071	18252
Acceso 14	82	185	4681	6965544
Acceso 15	31	102492	4819	18252
Acceso 16	147	240724	4948	18252
Acceso 17	14	160675	3973	380
Acceso 18	5	1773	3668	72
Acceso 19	159	18252	3337	72
Acceso 20	14	57	1621	6965544
Acceso 21	59	185	4610	72
Acceso 22	47	102492	6206	6965544
Acceso 23	93	711494	5894	72
Acceso 24	27	160675	4562	380

Acceso 25	7	1773	4255	72
Acceso 26	37	53	3938	72
Acceso 27	484	6965544	1607	6965544
Acceso 28	69	380	10003	67
Acceso 29	113	711494	6847	6965544
Acceso 30	219	18252	5493	380
Acceso 31	55	53	7080	6965544
Acceso 32	12	57	5734	380
Acceso 33	321	711494	10010	67
Acceso 34	564	6965544	7332	6965544
Acceso 35	18	33	8374	240724
Acceso 36	1	21738	9919	6965544
Acceso 37	263	240724	8561	380
Acceso 38	263	18252	7354	380
Acceso 39	15	57	7462	380
Acceso 40	109	380	8524	240724
Acceso 41	105	185	7166	380
Acceso 42	359	711494	8205	240724
Acceso 43	68	102492	7505	185
Acceso 44	629	6965544	7089	380
Acceso 45	20	33	6000	185
Acceso 46	1	21738	4704	380
Acceso 47	24	160675	5800	240724
Acceso 48	16	1773	6918	240724
Acceso 49	242	1989625	8012	240724
Acceso 50	162	380	6929	185
Acceso 51	155	711494	8033	240724
Acceso 52	408	18252	6969	185
Acceso 53	186	380598	6818	102492
Acceso 54	31	57	7906	240724
Acceso 55	7	13947	7792	102492
Acceso 56	54	53	8833	240724
Acceso 57	638	6965544	7731	185
Acceso 58	396	240724	6610	185
Acceso 59	297	1989625	5496	185
Acceso 60	115	185	5352	102492
Acceso 61	277	240724	4244	185
Acceso 62	182	380	4103	102492

Acceso 63	189	380598	4026	160675
Acceso 64	76	102492	2965	185
Acceso 65	7	13947	2873	160675
Acceso 66	22	160675	1832	185
Acceso 67	9	1773	1692	102492
Acceso 68	542	18252	1552	102492
Acceso 69	15	57	1420	102492
Acceso 70	114	711494	1338	160675
Acceso 71	392	711494	1176	102492
Acceso 72	782	6965544	1093	160675
Acceso 73	15	33	1074	1773
Acceso 74	2	21738	892	102492
Acceso 75	51	53	874	1773
Acceso 76	566	18252	695	102492
Acceso 77	16	57	616	160675
Acceso 78	342	185	530	160675
Acceso 79	46	102492	447	160675
Acceso 80	24	160675	423	1773
Acceso 81	5	1773	340	160675
Acceso 82	217	380	311	1773
Acceso 83	300	1989625	819	711494
Acceso 84	766	6965544	759	160675
Acceso 85	169	711494	1256	711494
Acceso 86	582	240724	1203	160675
Acceso 87	39	53	1174	1773
Acceso 88	456	711494	1160	1773
Acceso 89	46	33	1145	1773
Acceso 90	1	21738	1648	711494
Acceso 91	190	185	1648	1773
Acceso 92	271	380598	2126	711494
Acceso 93	6	13947	1684	53
Acceso 94	55	102492	1649	1773
Acceso 95	21	160675	1205	53
Acceso 96	4	1773	1173	1773
Acceso 97	445	240724	1660	711494
Acceso 98	373	380	2146	711494
Acceso 99	872	6965544	2633	711494
Acceso 100	1200	16057172	2198	53

Acceso 101	161	711494	2671	711494
Acceso 102	34	9366	2264	53
Acceso 103	300	1989625	2700	711494
Acceso 104	738	6965544	3176	711494
Acceso 105	464	711494	3618	711494
Acceso 106	67	53	4096	711494
Acceso 107	14	33	3676	53
Acceso 108	1	21738	3252	53
Acceso 109	291	185	2824	53
Acceso 110	198	380	3282	711494
Acceso 111	108	380	2868	53
Acceso 112	56	102492	3321	711494
Acceso 113	195	323118	2820	33
Acceso 114	174	380598	2404	53
Acceso 115	7	13947	1907	33
Acceso 116	22	160675	1485	53
Acceso 117	1186	16057172	1940	711494
Acceso 118	5	1773	2389	711494
Acceso 119	33	9366	2828	711494
Acceso 120	445	240724	2306	33
Acceso 121	171	711494	2734	711494
Acceso 122	40	53	2221	33
Acceso 123	363	1989625	2210	19451
Acceso 124	383	711494	678	1989625
Acceso 125	14	33	2811	19451
Acceso 126	1	21738	3264	711494
Acceso 127	261	323118	2750	33
Acceso 128	196	185	2242	33
Acceso 129	396	240724	1747	33
Acceso 130	431	240724	1737	19451
Acceso 131	49	102492	1241	33
Acceso 132	173	380598	1230	19451
Acceso 133	22	160675	667	1989625
Acceso 134	6	1773	1474	380598
Acceso 135	8	13947	2176	1989625
Acceso 136	87	185	1673	33
Acceso 137	116	185	1660	19451
Acceso 138	76	102492	1657	19451

Acceso 139	20	160675	668	1989625
Acceso 140	115	711494	2980	1989625
Acceso 141	7	1773	2977	19451
Acceso 142	56	102492	6869	711494
Acceso 143	307	1989625	3624	380598
Acceso 144	39	53	3380	13947
Acceso 145	20	160675	3470	380598
Acceso 146	5	1773	3456	19451
Acceso 147	449	711494	679	1989625
Acceso 148	152	711494	5788	19451
Acceso 149	15	33	3719	380598
Acceso 150	2	21738	3855	380598
Acceso 151	122	711494	4536	1989625
Acceso 152	133	380598	4027	33
Acceso 153	6	13947	3245	13947
Acceso 154	39	53	3715	14585903
Acceso 155	34	53	2931	13947
Acceso 156	1204	16057172	7938	1989625
Acceso 157	34	9366	4599	380598
Acceso 158	131	323118	5273	1989625
Acceso 159	252	1989625	5034	13947
Acceso 160	411	711494	4237	13947
Acceso 161	62	33	480	14585903
Acceso 162	1	21738	4022	19451
Acceso 163	119	380598	4503	14585903
Acceso 164	6	13947	4069	9306
Acceso 165	367	711494	4568	14585903
Acceso 166	382	711494	3834	380598
Acceso 167	18	33	3051	13947
Acceso 168	2	21738	3174	380598
Acceso 169	19	33	3664	14585903
Acceso 170	1	21738	8040	380598
Acceso 171	1140	16057172	4044	9306
Acceso 172	37	9366	4702	1989625
Acceso 173	251	1989625	4247	9306
Acceso 174	118	323118	4480	323118
Acceso 175	248	1989625	4004	9306
Acceso 176	266	1989625	3193	13947

Acceso 177	159	380598	3710	14585903
Acceso 178	6	13947	2922	13947
Acceso 179	120	380598	2467	9306
Acceso 180	6	13947	1645	13947
Acceso 181	1089	16057172	501	14585903
Acceso 182	38	9366	2278	380598
Acceso 183	173	380598	2555	323118
Acceso 184	7	13947	2552	323118
Acceso 185	139	323118	3036	14585903
Acceso 186	1067	16057172	2555	9306
Acceso 187	34	9366	3045	14585903
Acceso 188	142	323118	3334	323118
Acceso 189	1054	16057172	5249	323118
Acceso 190	24	9366	3173	9306
Acceso 191	81	323118	2350	13947
Acceso 192	904	16057172	505	14585903
Acceso 193	25	9366	2025	323118
Acceso 194	867	16057172	1559	9306
Acceso 195	36	9366	1167	323118
Acceso 196	1019	16057172	2427	9306
Acceso 197	25	9366	692	9306
Acceso 198	91	323118	277	323118
Acceso 199	68	323118	585	323118
Acceso 200	90	323118	1204	323118

**ANEXO III**  
**MATRIZ DE VALIDACIÓN DEL INSTRUMENTO**

**NOMBRE DEL INSTRUMENTO:** “GUIA DE OBSERVACIÓN PARA EVALUACIÓN DE LOS ACCESOS REALIZADOS A LOS SGBD POSTGRESQL Y MYSQL”

**OBJETIVO:** Registrar el tiempo de respuesta y el uso de memoria de los accesos a la base de datos de matrículas en los SGBD PostgreSQL Y MySQL.

**DIRIGIDO A:** Accesos realizados a la Base de datos de matrículas en los SGBD PostgreSQL Y MySQL.

**APELLIDOS Y NOMBRES DEL EVALUADOR:** *Nombres y apellidos del experto*

**GRADO ACADÉMICO DEL EVALUADOR:** *Grado Académico del experto*

**VALORACIÓN:**

<b>MALO = 1</b>	<b>REGULAR = 2</b>	<b>BUENO = 3</b>
-----------------	--------------------	------------------

---

**FIRMA DEL EVALUADOR**

**EVALUACIÓN DE EXPERTOS  
MATRIZ DE VALIDACIÓN DEL INSTRUMENTO**

**NOMBRE DEL INSTRUMENTO:** “GUIA DE OBSERVACIÓN PARA EVALUACIÓN DE LOS ACCESOS REALIZADOS A LOS SGBD POSTGRESQL Y MYSQL”

**OBJETIVO:** Registrar el tiempo de respuesta y el uso de memoria de los accesos a la base de datos de matrículas en los SGBD PostgreSQL Y MySQL.

**DIRIGIDO A:** Accesos realizados a la Base de datos de matrículas en los SGBD PostgreSQL Y MySQL.

**APELLIDOS Y NOMBRES DEL EVALUADOR:** BARRAZA VIZCARRA, HUGO MANUEL

**GRADO ACADÉMICO DEL EVALUADOR:** MAESTRO EN INGENIERÍA DE SISTEMAS E INFORMÁTICA

**VALORACIÓN:**

MALO = 1	REGULAR = 2	<del>BUENO = 3</del>
----------	-------------	----------------------

  
\_\_\_\_\_  
**FIRMA DEL EVALUADOR**  
.....  
MSc. HUGO BARRAZA VIZCARRA  
Ingeniero en Informática y Sistemas  
CIP: 156107

**EVALUACION DE EXPERTOS  
MATRIZ DE VALIDACIÓN DEL INSTRUMENTO**

**NOMBRE DEL INSTRUMENTO:** “GUIA DE OBSERVACIÓN PARA EVALUACIÓN DE LOS ACCESOS REALIZADOS A LOS SGBD POSTGRESQL Y MYSQL”

**OBJETIVO:** Registrar el tiempo de respuesta y el uso de memoria de los accesos a la base de datos de matrículas en los SGBD PostgreSQL Y MySQL.

**DIRIGIDO A:** Accesos realizados a la Base de datos de matrículas en los SGBD PostgreSQL Y MySQL.

**APELLIDOS Y NOMBRES DEL EVALUADOR:** CHOQUE CHACOLLA, EDUARDO SIMEON

**GRADO ACADÉMICO DEL EVALUADOR:** INGENIERO EN INFORMÁTICA Y SISTEMAS

**VALORACIÓN:**

<b>MALO = 1</b>	<b>REGULAR = 2</b>	<del><b>BUENO = 3</b></del>
-----------------	--------------------	-----------------------------



EDUARDO S. CHOQUE CHACOLLA  
ING. EN INFORMÁTICA Y SISTEMAS  
CIP 101550

**FIRMA DEL EVALUADOR**

**EVALUACION DE EXPERTOS  
MATRIZ DE VALIDACIÓN DEL INSTRUMENTO**

**NOMBRE DEL INSTRUMENTO:** “GUIA DE OBSERVACIÓN PARA EVALUACIÓN DE LOS ACCESOS REALIZADOS A LOS SGBD POSTGRESQL Y MYSQL”

**OBJETIVO:** Registrar el tiempo de respuesta y el uso de memoria de los accesos a la base de datos de matrículas en los SGBD PostgreSQL Y MySQL.

**DIRIGIDO A:** Accesos realizados a la Base de datos de matrículas en los SGBD PostgreSQL Y MySQL.

**APELLIDOS Y NOMBRES DEL EVALUADOR:** CONDORI SILES, ENRIQUE WALDO

**GRADO ACADÉMICO DEL EVALUADOR:** INGENIERO EN INFORMÁTICA Y SISTEMAS

**VALORACIÓN:**

MALO = 1	REGULAR = 2	<del>BUENO = 3</del>
----------	-------------	----------------------



\_\_\_\_\_  
**FIRMA DEL EVALUADOR**

## ANEXO IV CONSULTAS SQL

Las consultas seleccionadas para el plan de pruebas son las siguientes:

### QUERY 1

```
select a.IdFicha, a.Denominacion, h.Codigo, i.ApellidoPaterno,i.ApellidoMaterno,
i.Nombre,j.Denominacion,g.IdActa,c.IdAula,
d.Denominacion,b.Nota,f.Abreviatura,l.Abreviatura
from MAT_Ficha a inner join MAT_Periodo j on a.IdPeriodo = j.IdPeriodo
inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula
inner join OPE_Entidad i on h.IdEntidad = i.IdEntidad
Inner join ACT_FichaDetalle b on a.IdFicha = b.IdFicha
inner join ACT_Acta g on b.IdActa = g.IdActa
Inner join ACT_Aula c on g.IdAula = c.IdAula
inner join ADS_Materia d on c.IdMateria = d.IdMateria
inner join ADS_PlanCurricular e on d.IdPlanCurricular = e.IdPlanCurricular
inner join ADS_Especialidad f on e.IdEspecialidad = f.IdEspecialidad
inner join ADS_Escuela k on f.IdEscuela = k.IdEscuela
inner join ADS_Facultad l on k.IdFacultad = l.IdFacultad
where a.idPeriodo = 229 and f.Abreviatura = 'ESIS' and a.IdEstadoFicha = 1
and h.Codigo like '2014%'
```

### QUERY 2

```
select count(IdEstadoFicha),IdEstadoFicha
from MAT_Ficha
group by IdEstadoFicha
```

### QUERY 3

```
select a.IdFicha, a.Denominacion, h.Codigo, i.ApellidoPaterno,i.ApellidoMaterno,
i.Nombre,j.Denominacion,g.IdActa,c.IdAula,
d.Denominacion,b.Nota,f.Abreviatura,l.Abreviatura
from MAT_Ficha a inner join MAT_Periodo j on a.IdPeriodo = j.IdPeriodo
inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula
inner join OPE_Entidad i on h.IdEntidad = i.IdEntidad
Inner join ACT_FichaDetalle b on a.IdFicha = b.IdFicha
```

```

inner join ADS_Materia d on c.IdMateria = d.IdMateria
inner join ADS_PlanCurricular e on d.IdPlanCurricular = e.IdPlanCurricular
inner join ADS_Especialidad f on e.IdEspecialidad = f.IdEspecialidad
inner join ADS_Escuela k on f.IdEscuela = k.IdEscuela
inner join ADS_Facultad l on k.IdFacultad = l.IdFacultad
where a.idPeriodo = 229 and a.IdEstadoFicha = 1

```

#### QUERY 4

```

select a.IdFicha, a.Denominacion, h.Codigo, i.ApellidoPaterno,
i.ApellidoMaterno, i.Nombre, j.Denominacion, g.IdActa, c.IdAula,
d.Denominacion, b.Nota, f.Abreviatura, l.Abreviatura
from MAT_Ficha a inner join MAT_Periodo j on a.IdPeriodo = j.IdPeriodo
inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula
inner join OPE_Entidad i on h.IdEntidad = i.IdEntidad
Inner join ACT_FichaDetalle b on a.IdFicha = b.IdFicha
inner join ACT_Acta g on b.IdActa = g.IdActa
Inner join ACT_Aula c on g.IdAula = c.IdAula
inner join ADS_Materia d on c.IdMateria = d.IdMateria
inner join ADS_PlanCurricular e on d.IdPlanCurricular = e.IdPlanCurricular
inner join ADS_Especialidad f on e.IdEspecialidad = f.IdEspecialidad
inner join ADS_Escuela k on f.IdEscuela = k.IdEscuela
inner join ADS_Facultad l on k.IdFacultad = l.IdFacultad
where a.idPeriodo = 229 and f.Abreviatura = 'ESIS' and a.IdEstadoFicha = 1
and i.ApellidoPaterno = 'GINES'

```

#### QUERY 5

```

select a.IdFicha, a.Denominacion, h.Codigo, i.ApellidoPaterno,
i.ApellidoMaterno, i.Nombre, j.Denominacion, g.IdActa, c.IdAula,
d.Denominacion, b.Nota, f.Abreviatura, l.Abreviatura
from MAT_Ficha a inner join MAT_Periodo j on a.IdPeriodo = j.IdPeriodo
inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula
inner join OPE_Entidad i on h.IdEntidad = i.IdEntidad
Inner join ACT_FichaDetalle b on a.IdFicha = b.IdFicha
inner join ACT_Acta g on b.IdActa = g.IdActa
Inner join ACT_Aula c on g.IdAula = c.IdAula
inner join ADS_Materia d on c.IdMateria = d.IdMateria
inner join ADS_PlanCurricular e on d.IdPlanCurricular = e.IdPlanCurricular
inner join ADS_Especialidad f on e.IdEspecialidad = f.IdEspecialidad

```

```

inner join ADS_Escuela k on f.IdEscuela = k.IdEscuela
inner join ADS_Facultad l on k.IdFacultad = l.IdFacultad
where a.idPeriodo = 229 and f.idEspecialidad = 21 and a.IdEstadoFicha = 1

```

#### QUERY 6

```

select h.Codigo, i.ApellidoPaterno,i.Nombre,a.IdMatricula,a.IdFicha,
b.IdFichaDetalle, a.IdPeriodo from MAT_Ficha a
inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula
inner join OPE_Entidad i on h.IdEntidad = i.IdEntidad
Inner join ACT_FichaDetalle b on a.IdFicha = b.IdFicha
where a.idPeriodo = 229 and i.ApellidoPaterno = 'GINES'

```

#### QUERY 7

```

select h.Codigo from MAT_Ficha a
inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula
where a.idPeriodo = 229 and a.IdEstadoFicha = 1
group by (h.Codigo)

```

#### QUERY 8

```

select a.IdFicha, a.IdEstadoFicha, j.Denominacion
from MAT_Ficha a inner join MAT_Periodo j on a.IdPeriodo = j.IdPeriodo
where a.idPeriodo = 229 and a.IdEstadoFicha = 1

```

#### QUERY 9

```

select top 10 percent a.IdFicha, a.Denominacion, h.Codigo, i.ApellidoPaterno,
i.ApellidoMaterno, i.Nombre,j.Denominacion,g.IdActa,c.IdAula, d.Denominacion
from MAT_Ficha a inner join MAT_Periodo j on a.IdPeriodo = j.IdPeriodo
inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula
inner join OPE_Entidad i on h.IdEntidad = i.IdEntidad
Inner join ACT_FichaDetalle b on a.IdFicha = b.IdFicha
inner join ACT_Acta g on b.IdActa = g.IdActa
Inner join ACT_Aula c on g.IdAula = c.IdAula
inner join ADS_Materia d on c.IdMateria = d.IdMateria
where a.idPeriodo = 229 and h.Codigo like '2014%'

```

#### QUERY 10

```
select h.Codigo, i.ApellidoPaterno,a.IdMatricula,a.IdFicha from MAT_Ficha a
inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula
inner join OPE_Entidad i on h.IdEntidad = i.IdEntidad
where a.IdEstadoFicha = 1
```

#### QUERY 11

```
select a.IdFicha,h.Codigo, i.Nombre
from MAT_Ficha a inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula
inner join OPE_Entidad i on h.IdEntidad = i.IdEntidad
where i.Nombre = 'MILCA SARAY'
```

#### QUERY 12

```
select h.Codigo, i.ApellidoPaterno,a.IdMatricula,a.IdFicha from MAT_Ficha a
inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula
inner join OPE_Entidad i on h.IdEntidad = i.IdEntidad
where a.IdEstadoFicha = 1 order by h.Codigo DESC
```

#### QUERY 13

```
select count(IdPeriodo) as 'Periodo'
from MAT_Ficha
group by IdPeriodo
```

#### QUERY 14

```
select * from MAT_Periodo
```

#### QUERY 15

```
select a.IdFicha, a.Denominacion, h.Codigo, i.ApellidoPaterno,
i.ApellidoMaterno, i.Nombre,j.Denominacion,b.Nota
from MAT_Ficha a inner join MAT_Periodo j on a.IdPeriodo = j.IdPeriodo
inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula
inner join OPE_Entidad i on h.IdEntidad = i.IdEntidad
```

```
Inner join ACT_FichaDetalle b on a.IdFicha = b.IdFicha
where b.Nota <10
```

#### QUERY 16

```
select a.IdFicha, a.Denominacion, h.Codigo, i.ApellidoPaterno,i.ApellidoMaterno,
i.Nombre,j.Denominacion,b.Nota
from MAT_Ficha a inner join MAT_Periodo j on a.IdPeriodo = j.IdPeriodo
inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula
inner join OPE_Entidad i on h.IdEntidad = i.IdEntidad
Inner join ACT_FichaDetalle b on a.IdFicha = b.IdFicha
where b.Nota >16 and a.IdPeriodo =229
```

#### QUERY 17

```
select d.Denominacion,f.Abreviatura,l.Abreviatura
from ADS_Materia d inner join ADS_PlanCurricular e on
d.IdPlanCurricular = e.IdPlanCurricular
inner join ADS_Especialidad f on e.IdEspecialidad = f.IdEspecialidad
inner join ADS_Escuela k on f.IdEscuela = k.IdEscuela
inner join ADS_Facultad l on k.IdFacultad = l.IdFacultad
where f.Abreviatura = 'ESIS'
```

#### QUERY 18

```
select * from ACT_FichaDetalle
```

#### QUERY 19

```
select top 50 percent a.IdFicha, a.Denominacion, h.Codigo, i.ApellidoPaterno,
i.ApellidoMaterno, i.Nombre,j.Denominacion,g.IdActa,c.IdAula,
d.Denominacion,b.Nota,f.Abreviatura,l.Abreviatura
from MAT_Ficha a inner join MAT_Periodo j on a.IdPeriodo = j.IdPeriodo
inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula
inner join OPE_Entidad i on h.IdEntidad = i.IdEntidad
```

```

Inner join ACT_FichaDetalle b on a.IdFicha = b.IdFicha
inner join ACT_Acta g on b.IdActa = g.IdActa
Inner join ACT_Aula c on g.IdAula = c.IdAula
inner join ADS_Materia d on c.IdMateria = d.IdMateria
inner join ADS_PlanCurricular e on d.IdPlanCurricular = e.IdPlanCurricular
inner join ADS_Especialidad f on e.IdEspecialidad = f.IdEspecialidad
inner join ADS_Escuela k on f.IdEscuela = k.IdEscuela
inner join ADS_Facultad l on k.IdFacultad = l.IdFacultad
where a.idPeriodo = 229 and a.IdEstadoFicha = 1 and h.Codigo like '2014%'

```

## QUERY 20

```

select a.IdFicha, a.Denominacion, h.Codigo, i.ApellidoPaterno,i.ApellidoMaterno,
i.Nombre,j.Denominacion,g.IdActa,c.IdAula
from MAT_Ficha a inner join MAT_Periodo j on a.IdPeriodo = j.IdPeriodo
inner join MAT_Matricula h on a.IdMatricula = h.IdMatricula
inner join OPE_Entidad i on h.IdEntidad = i.IdEntidad
Inner join ACT_FichaDetalle b on a.IdFicha = b.IdFicha
inner join ACT_Acta g on b.IdActa = g.IdActa
Inner join ACT_Aula c on g.IdAula = c.IdAula
where a.idPeriodo = 229 and a.IdEstadoFicha = 1 and h.Codigo like '2014%'

```