UNIVERSIDAD SEÑOR DE SIPÁN

FACULTAD DE INGENIERIA, ARQUITECTURA Y URBANISMO

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS

TESIS

ANÁLISIS COMPARATIVO DE SISTEMAS GESTORES DE BASES DE DATOS POSTGRESQL Y MYSQL EN PROCESOS CRUD

PARA OPTAR EL TITULO PROFESIONAL DE INGENIERO DE SISTEMAS

Autor(a):
Bach. León Soberón Jenner Jesús

Asesor(a):
Mg. Tuesta Monteza Victor Alexis

Línea de Investigación:
Infraestructura, Tecnología y Medio Ambiente

Pimentel – Perú
2020

Dedicatoria

Este trabajo está dedicado a mi esposa María del Carmen que con mucho cariño y apoyo moral insistió en seguir mi objetivo académico, de concluir mis estudios y que además ha hecho mucho sacrificio de su parte para que nuestro proyecto de vida se pueda concluir y tener muchas más oportunidades.

A mis hijos Jenner Leonardo y María Alejandra, por la alegría que me brindan en el día a día y que me motiva para esforzarme en lo que me proponga.

Agradecimiento

Un agradecimiento especial a mi hermano Limbert, por su gran apoyo en todo, desde el inicio de mis estudios hasta la actualidad, proporcionándome guía y consejo académico. Se mantuvo pendiente de mis objetivos personales y nunca me pediste algo a cambio.

A mi asesor académico Ingeniero Victor Tuesta Monteza por brindarme sus conocimientos y sus orientaciones y que confió en mi persona en este proyecto.

A mi madre, por sus consejos y de sus preocupaciones, por su amor en mi desarrollo académico, personal, familiar y especialmente agradecer a Dios.

Gracias.

RESUMEN

El presente trabajo de investigación se centra en un análisis comparativo de los sistemas gestores de base de datos MySql y PostgreSql en la manipulación de registros demostrando el rendimiento en iguales escenarios en los procesos CRUD. Los gestores de bases de datos de código libre se han convertido en una gran alternativa en la gestión y administración de registros, entre ellos MySQL y PostgreSQL, ambos gestores de bases de datos tienen características especiales para el manejo de la información, siendo uno de ellos el de mejor rendimiento.

El objetivo es comparar los sistemas gestores de bases de datos PostgreSql y MySql en procesos CRUD, los cuales serán contabilizados con aplicaciones y hojas de cálculo. Los escenarios están relacionados de acuerdo a la cantidad de los procesos entre 10000, 100000 y 1001000 registros y a la estructura física del entorno de red. Se utilizó aplicación NET para el proceso de carga en cada terminal y almacenar los tiempos de respuesta y recopiladores de datos de Windows en el servidor para contabilizar el consumo de memoria y CPU en cada proceso CRUD. Los resultados de cuadros con los promedios en cada escenario, se aprecia un alto consumo de recursos y considerable tiempo de respuesta por parte de MySQL. Se concluye que PostgreSQL tiene buen rendimiento, pero MySQL manipula con mayor tiempo, los registros con imágenes, siendo una desventaja para PostgreSQL en estos procesos.

Palabras clave: Base de datos, MySql, PostgreSql, rendimiento, SQL, DBMS.

ABSTRACT

The present research work focuses on a comparative analysis of the database management systems MySql and PostgreSql in the manipulation of records demonstrating performance in the same scenarios in CRUD processes. The free code database managers have become a great alternative in the management and administration of records, including MySQL and PostgreSQL, both database managers have special features for the handling of information, one of them being the best performance.

The objective is to compare the management systems of PostgreSql and MySql databases in CRUD processes, which will be accounted for with applications and spreadsheets. The scenarios are related according to the number of processes between 10000, 100000 and 1001000 records and the physical structure of the network environment. The NET application was used for the load process in each terminal and store the response times and Windows data collectors in the server to post the consumption of memory and CPU in each CRUD process. In the charts results with the averages in each scenario show a high consumption of resources and considerable response time by MySQL. It is concluded that PostgreSQL has good performance in the CRUD processes while MySQL manipulates with longer time the records with images in consultation and eliminations, being this a disadvantage for PostgreSQL.

KEYWORDS: Database, MySql, PostgreSql, performance, SQL, DBMS..

ÍNDICE

INDICE DE TABLAS

INDICE DE FIGURAS

I. INTRODUCCIÓN

1.1. Planteamiento del Problema.

Debido al progresivo uso de la información en centros hospitalarios por la alta concurrencia en consultorio externo, que a su vez implican el manejo de otras acciones como historias clínicas, diagnóstico por imágenes y laboratorio, los cuales manipulan datos de tipo cadena como también imágenes que los hace cada vez más compleja en el manejo de la información y que afecta el rendimiento de los SGBD, es de suma importancia observar el tiempo de respuesta y consumo de recursos como CPU y memoria en procesos CRUD de los gestores de bases de datos de código libre en centros hospitalarios donde se manipulan grandes volúmenes de información como resultado de las transacciones generadas en sus operaciones y que permita identificar propiedades y características de los SGBD. En su investigación (Tongkaw & Tongkaw, 2016), se basa en brindar información con instrumentación y tareas básicas con pruebas de carga en bases de datos MariaDB y MySQL para probar el rendimiento en el uso de recursos de memoria y uso de CPU en un entorno de prueba con máquinas virtuales usando OLTP-Simple y OLTP- Seats, cuyos resultados del experimento muestran que MySQL tiene un rendimiento significativamente mejor que MariaDB.

Según (Portilla & Bernal, 2018) en su investigación sobre los gestores de bases de datos Oracle y MySql sobre el rendimiento en los comandos Insert, Select y Delete para probar el tiempo de repuesta, en el cual se implementó un esquema de base de datos y se incluyeron varios registros. Se realizaron un recorrido por diferentes sentencias de Lenguaje Estructurado de Consultas que fueron ejecutados en ambos gestores de bases de datos, se procedió a ejecutar las sentencias Select, Insert y Delete con una cantidad de datos y que posteriormente fueron tabulados. En la consulta simple, y concatenada simple, Inserción y Borrado, MySql es más eficiente que Oracle, sin embargo, en sentencias de consulta complejas Oracle supera a MySql. Concluyen que cuando se trabaja con una base de datos con muchas tablas y grandes cantidades de datos y que implican muchos procesos, Oracle es mejor en el uso de consultas de registros, sin embargo, MySql es más rápido para agregar registros, borrar y consultas pequeñas simples.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

La investigación (Huang, Mozafari, Schoenebeck, & T., 2016) le ponen mayor atención a la previsibilidad de rendimiento de ejecución transacciones y la escalabilidad de transacciones en los sistemas gestores de bases de datos. Las principales fuentes de variación en las latencias de transacción hacia bases de datos MySql y PostgreSql en el rendimiento de estado crítico que son los principales puntos significativos de esta investigación y además se centra en la reducción de la latencia y aumentar la escalabilidad. Se procedió a identificar las fuentes de variación de latencia y rendimiento con un marco de trabajo. En segundo lugar, Analizar MySql y PostgreSql y sus causas de variación. Se procedió a utilizar el marco de trabajo Vprofiler en el código fuente de MySql y PostgreSql para identificar principales varianzas de las latencias. Además, la aplicación de un algoritmo VATS para reducir la espera varianza, media, y latencia de transacciones. Se concluye que, usando nuestra herramienta, analizamos las bases de códigos de dos DBMS populares, lo que nos lleva a soluciones tanto genéricas como específicas de DBMS para reducir la varianza y la latencia.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

En la tabla I se tiene un ranking donde se aprecia las bases de datos relacionales de mayor uso y de mayor popularidad.

Tabla 1
Base de datos más populares

Rank			DBMS	Database Model	Score		
Apr 2019	Mar 2019	Apr 2018			Apr 2019	Mar 2019	Apr 2018
1.	1.	1.	Oracle	Relational, Multi-model	1279.94	+0.80	-9.85
2.	2.	2.	MySQL	Relational, Multi-model	1215.14	+16.89	-11.26
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1059.96	+12.11	-35.55
4.	4.	4.	PostgreSQL	Relational, Multi-model	478.72	+8.91	+83.25
5.	5.	5.	MongoDB	Document	401.98	+0.64	+60.57
6.	6.	6.	IBM Db2	Relational, Multi-model	176.05	-1.15	-12.89
7.	↑ 8.	↑ 9.	Redis	Key-value, Multi-model	146.38	+0.25	+16.27
8.	↑ 9.	8.	Elasticsearch	Search engine, Multi-model	146.00	+3.21	+14.64
9.	↓ 7.	↓ 7.	Microsoft Access	Relational	144.65	-1.55	+12.43
10.	10.	↑ 11.	SQLite	Relational	124.21	-0.66	+8.23

Nota: Recuperado de https://db-engines.com/en/ranking

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

1.2. Antecedentes de estudio

(Truica, Radulescu, Alexandru, & Ion, 2015) En su investigación "La Evaluación del rendimiento de las operaciones CRUD en una base de datos orientada a documentos replicados de forma asíncrona" centran su atención en el rendimiento de las bases de datos documentales frente a base de datos relacionales. Proceden a utilizar comandos CRUD para calcular el tiempo de ejecución bajo las mismas características del modelo de datos para DBMS de instancia única y para DBMS distribuidos. Los experimentos lo ejecutaron con máquinas virtuales con la misma configuración, las pruebas se realizaron con procesamiento por lotes. Se comparó el rendimiento en una única instancia y luego en entorno distribuido. Los resultados obtenidos en una sola instancia de proceso de actualización, las bases de datos documentales son mejores; en cuanto a las bases de datos relacionales, PostgreSql es mejor que MySql al utilizar Insert, Delete y Select. Las conclusiones entre las bases de datos orientadas a documentos, CouchDB tiene buen desempeño, pero se queda atrás cuando se trata de modelado de datos. En Ambos entornos el rendimiento de Couchbase y PostgreSql son muy similares.

(Poljak, Poščić, & Jakšić, 2017) En su investigación "Análisis comparativo de los sistemas de gestión de bases de datos relacionales seleccionados" realizan un estudio sobre la problemática de los tipos de datos de cada RDBMS Oracle, MySql y PostgreSql, sus formatos de cadenas, sintaxis y rendimiento. Realizando una comparación básica de las estructuras de cada RDBMS. Se diseñó una tabla comparando el soporte, lenguajes, plataformas, licencias, los datos, tabla para comparar su sintaxis, tabla para comparar los tipos de datos y tabla para comparar el rendimiento en cuanto la velocidad de consultas. El resultado relevante muestra que Oracle está diseñado para sistemas complejos. PostgreSql soporta más sistemas operativos considerando además de ser libre. MySql destinado a usuarios libres, la educación y los negocios. Oracle y PostgreSql tiene la sintaxis con similares características. Con respecto a tipos de datos dependerá del trabajo en particular de cada diseño. En cuanto a rendimiento Oracle es mejor en sistemas grandes; y después de ver todas las comparaciones, Oracle sería la mejor opción para RDBMS, si se trata de utilizarlo en sistemas grandes y complejos y estaría dispuesto a pagar, mientras que MySql es nuestra recomendación para las versiones de código abierto.

(Figueroa, Rollo, & Murthy, 2017) Realizan una breve comparación de los sistemas de base datos PostgreSql y MsSql, muy utilizados en la clase empresarial, la investigación se basó en encontrar similitudes y diferencias en los sistemas de base de datos MsSql y PostgreSql. Se trabajó principalmente en Identificar campos en ambos gestores de DB y sus equivalencias, columnas calculadas, índices de diseño, vistas y la ejecución de consultas DML. El trabajo consistió el diseñó de tablas con campos de datos relacionados, ejecución de sentencias DLL involucrando procedimientos y funciones; en ambos gestores se apreciaron compatibilidad en los estándares, pero aún tienen muchas diferencias en aplicaciones empresariales. En sus conclusiones encontraron que tanto MsSql y PostgreSql incluyen características competitivas aplicados a sistemas empresariales. MsSql adopta un enfoque de implementación amigable, PostgreSql es bastante adecuado para aplicaciones empresariales, pero requiere un esfuerzo adicional por la falta de ciertas características.

Para (López, 2016) En su tesis "*Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL*" se centran en ver la seguridad y desempeño de los SGBD y realizan un análisis comparativo a través de estudios estructurales de cada gestor y dar a conocer cuál es el que presenta mejores propiedades y así obtener información para la toma de decisiones, se procedió a identificar las características generales de cada gestor de bases de datos relacionales. En el trabajo de investigación se detallaron las características importantes de MySql y PostgreSql en cuanto al tipo de datos, procedimientos, funciones, seguridad, requerimientos, arquitectura, interfaces con lenguajes de programación y sus aplicaciones. La comparación entre los SGBD MySql y PostgreSql se observa características relevantes de cada uno de los SGBD, pero por su complejidad no se pueden generalizar. Concluye lo siguiente: Ambas aplicaciones tiene característica que las identifican, ya sea en sus propiedades o esquemas, en el caso del SGBD MySQL ha sido un gestor de datos muy ventajoso, ya que manipula una gran cantidad de datos, caracterizándolo como un gestor de buen rendimiento. Su conectividad, velocidad y seguridad lo hacen apropiado en la actividad empresarial. La característica técnica de PostgreSQL cuenta con estabilidad y potencia haciendo una base de datos potente y robusta. Funciona muy

bien con grandes cantidades de datos y una alta concurrencia de usuarios. Pero consume más recursos que MySQL. Es más fácil encontrar documentación de soporte para MySQL que para PostgreSQL.

(Almonacid, 2016) En su tesis "*Comparación entre gestores de bases de datos relacionales*" identifican la problemática del rendimiento entre gestores de Bases de Datos Relacionales a través de sentencias de manipulación de datos utilizando el lenguaje SQL; se procede a documentarse sobre cada SGBD Sql Server, PostgreSql y MySql, se realizó modificaciones en las configuraciones de Tuning de cada gestor. En la investigación del proyecto se ejecutarán las sentencias de transacciones de inserción y consulta con las configuraciones predeterminadas de cada gestor, luego se realizará lo mismo, pero realizando cambios en la configuración de cada gestor. En los resultados se podía observar que PostgreSQL era el gestor de bases de datos que presentaba los menores tiempos de respuesta en los distintos escenarios. La conclusión con respecto a los resultados obtenidos en los escenarios con distinta complejidad de consultas, se puede determinar que los gestores MySQL y SQL Server 2012 presentan comportamientos similares durante las pruebas de inserción y de ejecución de consultas. Por último, se pudo concluir, por estadística descriptiva, que el gestor de bases de datos que presentó los menores tiempos de respuesta, ya sea en la inserción de datos y/o en la ejecución de consultas fue PostgreSQL.

Según (Andrade & Parra, 214) en su tesis "*Análisis de rendimiento entre* Postgresql *y Sql Server usando Hammerdb y Manage Engine aplicado al Sistema Académico de Conduespoch*" identifican el problema de estudio en el rendimiento para el desarrollo de aplicaciones web, la investigación consiste en utilizar herramientas para medir el rendimiento de los SGBD y que interactúan con el servidor, además de una aplicación web. Se procede a crear bases de datos con estructuras iguales para ambos gestores. Para las pruebas de tiempos de respuesta y Tiempo del aseguramiento de la integridad de datos se realizará una aplicación web sencilla, mediante la cual se enviará un número determinado de transacciones a los SGBD. En las pruebas se observó que el uso de CPU, SQL Server consumió menor porcentaje que PostgrSql; en memoria RAM PostgreSql consumió menor

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

porcentaje; en uso de red, SQL Server procesa más paquetes. En inserción y consulta de registros, SQL Server supera a PostgreSql. La investigación concluye: SQL Server Express tiene un mejor rendimiento en comparación al SGBD PostgreSQL en la mayoría de acciones. Sin embargo, PostgresSQL fue más veloz comparación con el SGBD SQL Server en las operaciones de consulta de registros.

(Cevallos, 2014) en su tesis "Análisis Comparativo de Respaldo y Recuperación de Base de Datos Licenciada (Oracle Utilizando RMAN) VS Open Source (MYSQL Utilizando MYSQL Administrator)" se centran en los errores del software o hardware que generen perdida de información por posibles causas de desastres naturales o error humano involuntario. Se procedió a documentarse acerca de las características e información de los SGBD, se preparó una encuesta a profesionales de informática. Se utilizó la técnica de la encuesta a especialistas como son ingenieros y técnicos en el área de sistemas sobre sus experiencias y conocimiento en respaldo y recuperación de bases de datos licenciada y Opensource. Se trabajó en dos Laptops para ver los procesos de respaldo y recuperación. Los resultados fueron los siguientes: Al realizar el respaldo en una base pequeña Oracle tardó 70 sg y MySQL 1, en una base mediana Oracle tardó 117 sg y MySQL 107 sg, en una base grande Oracle tardó 382 sg y MySQL 525 sg. En la recuperación, en una base pequeña Oracle tardó 0 sg y MySQL 6 sg, en una base mediana Oracle tardó 118 sg y MySQL 739, en una base grande Oracle tardó 241 sg y MySQL 523 sg. En una base pequeña, mediana y grande, MySQL usa más memoria. Concluye lo siguiente, Oracle RMAN es una herramienta muy confiable y segura, ofrece una base completa para hacer respaldos y recuperación.

Según (Secco, Da Silva, Maracci, & Pazoti, 2016) Señalan que la elección de base de datos debe tener cuidado para almacenar información relevante, ya que los bancos deben proporcionar una alta confiabilidad, las prueba se realizó con la base de datos Casandra y PostgreSQL, por lo que el rendimiento se comprobó en dos organizaciones empresariales, utilizando el mismo modelo de base de datos, instalando un solo servidor con un solo núcleo para ambos gestores de bases de datos, posteriormente crearon clases de conexión tanto para Casandra y PostgreSQL y los cuales realizaron las operaciones CRUD. Concluyen que

PostgreSQL en la administración es fácil y dinámica, además de ser gratuito, pero lento en la búsqueda de información debido a sus reglas relacionales, a diferencia de Cassandra es una alternativa flexible y escalable cuando se trata de una tabla grande de datos.

1.3. Teorías relacionadas el tema

En el siguiente punto iniciamos describiendo teorías relacionados a nuestro tema de investigación.

1.3.1. Base de Datos

Una base de datos es un conjunto formado por varias Tablas con alguna afinidad temática. (Arias M. , 2017, pág. 135)

Se define una base de datos como un conjunto de datos organizados y relacionados entre sí. (Tejada, 2017)

Una base de datos es una colección o depósito de datos, donde estos se encuentran lógicamente relacionados entre sí. Se toma un modelo del mundo real para poder trabajar con esos datos a través de aplicaciones y programas. (Jiménez, 2015)

Llamamos base de datos (o bases de datos) a un conjunto de datos dispuestos con el objetivo de proporcionar información a los usuarios y permitir transacciones como inserción, eliminación y actualización de datos. (Benítez & Arias, 2015, pág. 4)

Una base de datos es un conjunto de informaciones organizadas que son administradas utilizando un programa especial denominado gestor de bases de datos. (García, 2012, pág. 167)

Una base de datos (BD) es un conjunto de datos relacionados entre sí, organizados y estructurados con información referente a algo (Ramos & Ramos, 2007, pág. 2)

Es una colección interrelacionada de datos, almacenados en un conjunto sin redundancias innecesarias cuya finalidad es la de servir a una o más aplicaciones de la manera más eficiente. (Nevado, 2010, pág. 22)

1.3.2. Transacción

Una transacción es una unidad lógica de procesamiento en una base de datos, incluyendo dos o más operaciones en una base de datos. (Arias Á. , Fundamentos de Programación y Bases de Datos, 2016, pág. 229)

Una transacción es una secuencia de operaciones de bases de datos que tienen acceso a ésta (Coronel, Morris, & Rob, 2011, pág. 440)

Una transacción consiste en una secuencia de instrucciones de consulta o actualizaciones (Silberschatz, Korth, & Sudarshan, 2006, pág. 90)

Una transacción es una unidad lógica de procesamiento en una base de datos, incluyendo dos o más operaciones en una base de datos que obligatoriamente deben ocurrir para que toda la unidad tenga sus modificaciones establecidas de forma permanente. (Benítez & Arias, 2015, pág. 111)

Una transacción es un conjunto de operaciones dependientes las unas de las otras que se ejecutan en una base de datos. (Bernabé, 2015)

1.3.3. SQL

El lenguaje estructurado para consulta (SQL, por sus siglas en inglés; Structure Query Languaje) es un lenguaje basado en el desarrollo de programas que especifican o declaran un conjunto de condiciones, proposiciones, afirmaciones, restricciones, ecuaciones o transformaciones. (Baca, 2016, pág. 158)

SQL (Lenguaje de consulta estructurado) es el lenguaje de acceso a bases de datos relacionales más extendido. Con este sistema, el cliente especifica las instrucciones para crear, borrar o dotar de contenido las tablas de la base de datos. (García, 2012, pág. 172)

SQL es un lenguaje de acceso a base de datos que explota la flexibilidad y potencia de los sistemas relacionales, permitiendo gran variedad de operaciones sobre los mismos. (Tejada, 2017)

SQL es el nombre de un lenguaje desarrollado para la formulación de búsquedas en bases de datos. (Arias Á. , Fundamentos de Programación y Bases de Datos, 2016, pág. 115)

1.3.4. Lenguaje de definición de datos (LDD o DDL)

DDL (Data Definition Language) es el nombre dado a los comandos SQL que se utilizan para crear y modificar tablas. (Arias Á. , Fundamentos de Programación y Bases de Datos, 2016, pág. 119)

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Los esquemas de las bases de datos se especifican mediante un conjunto de definiciones expresadas mediante un lenguaje especial denominado lenguaje de definición de datos (LDD). (Silberschatz, Korth, & Sudarshan, 2006, pág. 8)

Permite definir la representación lógica de los datos que integran la base de datos. (Nevado, 2010, pág. 36)

Lenguaje que se utiliza para escribir específicamente cómo serán los datos a almacenar en la base de datos y las condiciones que deben cumplir para ser aceptados. (Bernabé, 2015)

Lenguaje que ayudará a definir la estructura y los componentes, tablas, atributos y restricciones de la base de datos. (Jiménez, 2015)

1.3.5. Lenguaje de manipulación de datos (LMD o DML)

DML (Data Manipulation Languaje) Es el conjunto de comandos que manipulan los datos de una base de datos. (Arias Á. , Fundamentos de Programación y Bases de Datos, 2016, pág. 127)

Un lenguaje de manipulación de datos (LMD) es un lenguaje que permite a los usuarios tener acceso a los datos organizados mediante el modelo de datos correspondiente o manipularlos. (Silberschatz, Korth, & Sudarshan, 2006, pág. 7)

Permite realizar dos funciones en la gestión de los datos: Definición del nivel externo o de usuario de los datos. La manipulación de los datos (inserción, borrado, modificación y recuperación). (Nevado, 2010, pág. 34)

Lenguaje utilizado para manipular esos datos (leerlos, añadir datos, modificarlos, borrarlos). (Bernabé, 2015)

Lenguaje que ayudará a la manipulación de los datos, pudiendo ser utilizado para realizar consultas y modificaciones en la base de datos. (Jiménez, 2015)

1.3.6. Consulta

Las consultas son los objetos de una base de datos que permiten recuperar datos de una tabla atendiendo a unos criterios, hacer modificaciones sobre ellos y almacenar los resultados en otra tabla. (Tejada, 2017)

Una consulta es una sentencia mediante la cual se solicita información de la base de datos a través de un lenguaje como DML. (Jiménez, 2015)

Una consulta es una instrucción que solicita que se recupere información. (Silberschatz, Korth, & Sudarshan, 2006, pág. 8)

Un requerimiento de consulta es llamado query y expresa un criterio de búsqueda en una expresión de consulta. (Cisneros, 1998, pág. 65)

Las consultas nos van a mostrar los datos que cumplan los criterios especificados en su diseño. (Ramos & Ramos, 2007, pág. 64)

Es una petición específica hecha al DBMS para manipulación de datos. (Coronel, Morris, & Rob, 2011, pág. 8)

1.3.7. Modelo relacional

El modelo relacional usa una colección de tablas para representar tanto los datos como sus relaciones. Cada tabla tiene varias columnas, y cada columna tiene un nombre único. (Silberschatz, Korth, & Sudarshan, 2006, pág. 6)

El modelo relacional es un modelo de datos utilizado para representar datos interrelacionados. (Bernabé, 2015)

Una BD relacional representa al mundo real mediante tablas o relaciones que contienen la información ordenada de un modo organizado. Las tablas se relacionan entre sí por columnas comunes. (Tejada, 2017)

El modelo relacional de datos se implementa por medio de un complejo sistema de administración de base de datos relacional (RDBMS, por sus siglas en ingles). (Coronel, Morris, & Rob, 2011, pág. 36)

Tiene este nombre debido a que organiza los datos en tablas y establece relaciones entre las tablas. Este es el modelo más popular. (Arias Á. , 2016, pág. 93)

1.3.8. Rendimiento

El rendimiento de los sistemas de bases de datos es un aspecto crítico de la mayor parte de los sistemas informáticos empresariales. El rendimiento no sólo tiene que ver con el uso eficiente del hardware de cálculo y de almacenamiento que se usa, sino también con la eficiencia de las personas que interactúan con el sistema y de los procesos que dependen de los datos de las bases de datos. (Silberschatz, Korth, & Sudarshan, 2006, pág. 208)

El rendimiento en una base de datos es un apartado fundamental. Si el acceso a los datos es lento, se estará creando un cuello de botella que lastrará a todas las aplicaciones que utilicen la base de datos. (Bernabé, 2015)

Número total de consultas en un intervalo de tiempo dado. (Talledo, 2016, pág. 75)

1.3.9. Tiempo de respuesta

Es el tiempo que tarda una sola transacción desde el comienzo hasta el final en promedio o en el peor de los casos. (Silberschatz, Korth, & Sudarshan, 2006, pág. 208)

El tiempo de respuesta se define como el tiempo y que implica enviar un mensaje desde la localidad A a la B. (Cobo, pág. 94)

1.3.10. MySql

Mysql es un sistema de base de datos libre, de código abierto, rico en funciones para los usuarios novatos, y más sencillo que otros sistemas con características similares como PostgreSql. (Arias Á. , Bases de Datos con MySQL, 2014, pág. 19)

MySql es un SGBDR extremadamente expandido y popular en los servidores de Internet. Su éxito viene por un lado de su facilidad de implementación y por otro de su carácter original Open Source. (Deléglise, 2013, pág. 28)

MySQL fue desarrollado originalmente para manejar grandes bases de datos mucho más rápido que las soluciones existentes y se ha utilizado con éxito en entornos de producción muy exigentes desde hace varios años. (Giacomo, 2005, pág. 11)

Sistema libre y gratuito para uso no comercial. Ampliamente utilizado en programación web. Utilizado por millones de webs en todo el mundo. (Bernabé, 2015)

Mysql es una base de datos relacional que utiliza el lenguaje SQL (Structured Query Language Lenguaje de consulta Estructurado) Se trata de un SBD de código abierto, lanzado en 1995. (Arias Á. , 2014, pág. 39)

1.3.11. PostgreSql

PostgreSql es un sistema de gestión de base de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de

gestión de base de datos de código abierto más potente del mercado (Zea, Molina, & Redrován, 2017, pág. 12)

Potente sistema de base de datos objeto-relacional de código abierto (Henriquez, Iglesias, Amaris, & Ropain, 2013)

Sistema libre y gratuito. Su principal cometido es ser una alternativa a MySql. (Bernabé, 2015)

Es un motor de bases de datos orientadas a objetos para Linux. Soporta lenguajes de programación como C++, Java y PHP. Soporta bases de datos de más de 200 GB, es de distribución gratuita y se consigue en Internet. (Gamboa, 2004, pág. 20)

1.3.12. Memoria virtual, memoria RAM

La memoria virtual es el mecanismo que emplea el computador para sustituir la memoria secundaria por la primaria. Se usa cuando al estimar que la capacidad de la memoria principal no es suficientemente grande para albergar toda la información necesaria durante la ejecución de programas. (Rubio, Gómez, Letón, Rodrigo, & Chaos, 2017, pág. 39)

Las memorias de acceso aleatorio, RAM, son memorias de tipo semiconductor. Las memorias RAM son volátiles, ya que al suspender la alimentación de la memoria la información es eliminada. (Rubio, Gómez, Letón, Rodrigo, & Chaos, 2017, pág. 43)

Memoria de acceso aleatorio, RAM (Random Access Memory): es un tipo de acceso muy empleado en las memorias primarias. Requiere el mismo consumo de tiempo de acceso, independientemente de la posición de memoria a la que se desee acceder. (Rubio, Gómez, Letón, Rodrigo, & Chaos, 2017, pág. 41)

La memoria RAM puede ser escrita y leída cuantas veces sea necesario y su característica principal es su volatilidad, es decir, la pérdida de la información almacenada cuando deja de estar alimentada por corriente eléctrica. Realmente esta es la parte de la memoria que más utiliza el ordenador como mesa de trabajo para tomar datos, instrucciones y volcar resultados. (Heredero, 2004, pág. 66)

1.3.13. CPU

La CPU, unidad de proceso central, también llamado microprocesador. En la CPU se realiza los cálculos, se controlan y se coordinan los componentes del

computador. Este componente permite interpretar y realizar las operaciones que se solicitan al ejecutar un programa de software. (Rubio, Gómez, Letón, Rodrigo, & Chaos, 2017, pág. 28)

La unidad central de proceso o CPU también suele recibir el nombre de procesador y es la parte más importante de la computadora. En ella se trabaja con los datos de entrada y las instrucciones de programa correspondientes y también se controla el resto de los elementos o componentes. Se enarga por tanto de realizar, controlar y coordinar las operaciones de todo el sistema u ordenador. (Heredero, 2004, pág. 64)

1.3.14. Tipo de consultas

1.3.14.1. Consultas selectivas

Donde se localizan registros que cumplen una determinada condición, según un criterio de selección. (Nevado, 2010, pág. 33)

Consultas que extraen aquellos datos de una tabla que cumplen unos criterios especificados. Una vez obtenido el resultado, los datos se podrán consultar para ser modificados o no según el tipo de consulta de selección. (Tejada, 2017)

Permite visualizar determinados registros bajo condiciones que se establecen en la sentencia, extrayendo los resultados necesarios a consultar. (Zea, Molina, & Redrován, 2017, pág. 32)

Permite dar respuesta a las preguntas que se puedan plantear sobre los datos introducidos en tablas. Básicamente son aquellas que extraen o muestran datos de una o varias tablas que cumplen una serie de condiciones establecidas. (Carmona, 2013)

1.3.14.2. Consulta de acción

Consulta de acción: permiten realizar acciones automáticas como modificación y eliminación de registros que cumplan determinados criterios. Existen varios tipos de consultas de acción: de eliminación, de actualización, de datos anexados y de creación de tablas. (Tejada, 2017)

Son aquellas que permitan realizar operaciones con los datos de una tabla de modo automatizado: añadir registros procedentes de otras tablas, actualizar u operar con los datos de un campo o tabla, etc. (Carmona, 2013)

1.3.14.3. Consulta sobre la totalidad de los datos

Donde se recuperan todos los datos de la BD o todos los de un determinado tipo. (Nevado, 2010, pág. 33)

Consultas que se definen en lenguaje SQL (lenguaje de consulta estructurado o Structured Query Language). (Tejada, 2017)

1.3.15. Triggers

El trigger se ejecuta al ejecutar la instrucción SQL que lo ha activado, incluso si están implicados varios registros de datos. (Gabillaud, 2015, pág. 315)

Un trigger es un conjunto con un nombre propio de instrucciones SQL vinculados a una tabla y que se desencadena cuando interviene una inserción, una supresión o una modificación en dicha tabla. (Thibaud, 2006, pág. 186)

Un trigger (disparador o desencadenador) es un bloque de código que se ejecuta automáticamente cuando ocurre algún evento (como inserción, actualización o borrado) sobre una determinada tabla (o vista) (Zea, Molina, & Redrován, 2017, pág. 46)

1.3.16. Procedimientos Almacenados

Los procedimientos almacenados (Stored Procedures) son objetos que corresponde a un conjunto de instrucciones LMD que se pueden ejecutar mediante una simple llamada por su nombre o con la instrucción EXECUTE. (Gabillaud, 2015, pág. 292)

1.3.17. Definición de términos básicos

a) Información: Es el resultado de procesar datos sin elaborar para dejar ver su significado. (Coronel, Morris, & Rob, 2011, pág. 5)

b) Sistemas de información: Un sistema de información (SI) es un conjunto formal de procesos, que recopilan, elaboran y distribuyen la información o parte de ella, para poder realizar operaciones propias de una empresa, sus actividades de control y dirección según su estrategia de negocio. (Nevado, 2010, pág. 17)

c) Datos: Se trata de la información relevante que almacena y gestiona el sistema de información. Los datos por sí solos no aportan conocimiento, es necesario procesarlos y transformarlos. (Nevado, 2010, pág. 18)

d) Sistema de administración de base de datos (BDMS): Es un conjunto de programas que maneja la estructura de la base de datos y controla el acceso a los datos guardados en ella. (Coronel, Morris, & Rob, 2011, pág. 7)

e) Calidad de datos: Es un método completo para promover la precisión, validez y puntualidad de los datos. (Coronel, Morris, & Rob, 2011, pág. 9)

f) Diseño de base de datos: Se refiere a las actividades que se concentran en el diseño de la estructura de la base de datos que se usará para guardar y administrar datos del usuario final. (Coronel, Morris, & Rob, 2011, pág. 10)

g) Integridad de los datos: Se define como la condición en la que todos los datos de la base de datos son consistentes con los hechos y condiciones del mundo real. En otras palabras, los datos son precisos, no hay inconsistencia de datos. Los datos son verificables, los datos siempre darán resultados consistentes. (Coronel, Morris, & Rob, 2011, pág. 16)

h) Usuarios finales: Este término se refiere a todos los usuarios que no tienen que ver directamente con la gestión de la base de datos. (Arias Á. , Fundamentos de Programación y Bases de Datos, 2016, pág. 95)

i) Relación y Tupla: Es el elemento básico del modelo relacional y se puede presentar como una tabla. En ella se distingue un conjunto de columnas, denominadas atributos, que representan las propiedades de la misma y un conjunto de filas llamadas tuplas que son las ocurrencias de la relación. (Cobo, pág. 44)

j) ACID (Atomicity, Consistency, Isolation and Durability, en español Atomicidad, Consistencia, Aislamiento y Durabilidad): Permite clasificar las transacciones de los sistemas de gestión de base de datos. Se dice que es ACID compliant cuando éste permite realizar transacciones. (Talledo, 2015, pág. 237)

k) Integridad referencial: La integridad referencial es un sistema de reglas utilizadas por diversos gestores de bases de datos relacionales. El motivo es asegurarse que los registros de las tablas relacionadas sean válidos y que no se borren o modifiquen los datos relacionados de forma accidental generando datos corruptos con respecto a su integridad. (Talledo, 2015, pág. 108)

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

l) Concurrencia de transacciones: La concurrencia de transacciones es el nombre que se da cuando se producen dos o más operaciones en paralelo en un sistema de base de datos. Como las transacciones pueden entrar en conflicto mediante la manipulación de un sólo elemento de una base de datos, debemos utilizar técnicas que eviten este tipo de conflicto. (Benítez & Arias, 2015, pág. 115)

1.4. Formulación del problema.

¿Cuál de los gestores de base de datos MySql y PostgreSql tendrá mejor rendimiento en los procesos CRUD?

1.5. Justificación e importancia del estudio.

En el presente documento se pretende obtener información suficiente para evaluar los dos gestores de bases de datos de código libre, observando los procesos básicos de bases de datos como insertar, leer, editar y eliminar registros y con la intensión de brindar información acerca del rendimiento en dichos procesos, conocidos también como procesos CRUD en aplicaciones de escritorio.

Con esta investigación podemos tener un alcance más claro sobre estos procesos en estos gestores de bases de datos independiente de cada proceso como también en su conjunto y su resultados brindaran información como un aporte más para futuras investigaciones y un apoyo en la toma de decisiones en proyectos de sistemas informáticos, además será un antecedente para tomar en cuenta sobre algunas configuraciones de estos gestores de bases de datos; sin desestimar la potencialidad de MySQL y PostgreSQL.

El presente proyecto reúne las características y condiciones técnicas que aseguran el cumplimiento de los objetivos que trata de consolidar los estudios previos a este trabajo, llevados a cabo por experiencias técnicas, con aspiraciones a brindar apoyo de información necesaria acerca de MySQL y PostgreSQL. Así también tendremos un alcance significativo académico, sobre el uso de los resultados que se puedan obtener al término del presente trabajo.

1.6. Hipótesis.

Hipótesis nulas (H0). - El gestor de base de datos MySql tiene mejor rendimiento en comparación a PostgreSql.

Hipótesis alterna (Ha). - El gestor de base de datos PostgreSql tiene mejor rendimiento en comparación a MySql.

1.7. Objetivos.

1.7.1. Objetivo general.

Comparar sistemas gestores de bases de datos PostgreSql y MySql en procesos CRUD

1.7.2. Objetivos específicos.

a) Seleccionar SGBD.
b) Establecer métricas para la evaluación de los gestores de base de datos.
c) Ejecutar los procesos CRUD en el caso de estudio.
d) Analizar resultados.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

II. MATERIAL Y MÉTODO

2.1. Tipo y diseño de la investigación.

2.1.1. Tipo de investigación.

Para el estudio del trabajo de investigación en la comparación de gestores de base de datos MySql y PostgreSql y de acuerdo a la naturaleza, el tipo de investigación es Cuantitativa ya que se llevará a cabo con el análisis de los resultados numéricos de los sistemas gestores de base de datos MySql y PstgreSql en la ejecución de los procesos CRUD.

2.1.2. Diseño de la investigación.

El diseño de la investigación del estudio corresponde al tipo de diseño Cuasi Experimental, aplicadas a dos reconocidos gestores de Base de datos, MySql y PostgreSql y analizar los resultados de nuestras pruebas en cuanto al rendimiento de los procesos CRUD.

2.2. Población y muestra.

La población a considerar para nuestras pruebas se basa a tres escenarios con 10000, 100000 y 1001000 registros.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

2.3. Variables.

2.3.1. Variable Independiente

Gestores de base de datos MySQL y PostgreSQL.

2.3.2. Variable dependiente

Rendimiento de los gestores de base de datos.

Tabla 2
Variable Dependiente

Variable	Dimensión	Indicadores	Formula	Técnicas e instrumentos de recolección de datos
Rendimiento de los gestores de base de datos	Consumo de recursos	Consumo de CPU.	$R = \frac{F_{Reloj} \times N_{bits}}{C_{Pro}}$	Técnica de observación. Fichas de observación
		Consumo de Memoria RAM.	Cantidad de bytes	
	Velocidad de respuesta	Tiempo de respuesta	Tiempo actual de ejecución del proceso –Tiempo de inicio del proceso (en milisegundos)	

Nota:

$$R = \frac{F\_Reloj \times N\_bits}{C\_Pro}$$

Dónde:

R : Rendimiento

F_Reloj : Frecuencia del reloj

N_bits : Número de bits

C_Pro : Ciclos de procesamiento

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

$$Throughput = \frac{Frecuencia\ del\ reloj\ \times Número\ de\ bits}{Ciclos\ de\ procesamiento}$$

(Piñal M., Álvarez G., & Sánchez G., 2009) Cantidad de datos procesados por unidad de tiempo

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

Para obtener la información se hizo uso de la observación.

2.4.1. Observación

Lo realizaremos de la siguiente manera:

Para el tiempo de velocidad y respuesta: Se procederá a recolectar la información a través de la aplicación NET que estarán instalados en cada terminal, con los formularios diseñados para ejecución de los procesos, por cada registro ingresado (Create), consultado (Read), actualizado (Update) y eliminado (Delete), que en determinado tiempo en milisegundos irá recolectando la información de esta unidad de medida para éste indicador y que se ejecutarán paralelamente en los tres terminales que realizarán la carga.

Al término de cada proceso CRUD, el formulario enviara la información recolectada a una hoja de cálculo Excel.

Esta acción se ejecutará por igual en los procesos CRUD, tanto en MySQL como en PostgreSQL.

Para consumo de CPU y consumo de memoria RAM: Se utilizará el Conjunto de recopiladores de datos (Herramienta de Windows) que se ejecutará en el servidor y que recopilará el consumo de CPU cada 30 segundos. Para estos dos indicadores se debe tener en cuenta que en los tres terminales se están ejecutando los formularios, ya sea para ingresar, consultar, actualizar y eliminar registros.

En cada proceso CRUD y cada 30 segundos se irá registrando en una hoja de cálculo asignado por el Conjunto de recopiladores de datos la información recolectada, tanto para MySQL como para PostgreSQL.

2.5. Procedimiento de análisis de datos.

2.5.1. Observación

Para el análisis de los datos se procedió a realizar lo siguiente:

a) Para medir el tiempo de velocidad y respuesta: Haremos uso de la hoja de cálculo Excel, el cual ha registrado valores numéricos en milisegundos de los procesos CRUD, con dicha información se procederá a promediar la información y diseñar cuadros estadísticos en columna. Estos procedimientos se realizarán en cada terminal que contiene la información por cada proceso CRUD.

b) Para medir el consumo de CPU y la memoria RAM: Haremos uso de la hoja de cálculo de Excel a través del conjunto de recopiladores de datos de Windows, el cual ha almacena valores numéricos de los procesos CRUD, con dicha información se procederá a promediar la información y diseñar cuadros estadísticos en columna. Este procedimiento se realizará en el equipo que se utilizó como servidor por cada proceso CRUD.

2.6. Criterios éticos

Los criterios a tomar, son los siguientes:

2.6.1. Ambiente

El presente trabajo de investigación no genera actividades que deterioren el medio ambiente ya sea institucional o fuera de ella; pero se debe tomar en consideración los efectos del hardware desfasado con varios años y la forma cómo los componentes electrónicos al deteriorarse afectan al ambiente, orientando de manera correcta su desecho en caso de desuso.

2.6.2. Confidencialidad

El trabajo de investigación preserva la identidad de los registros.

2.6.3. Objetividad

El presente proyecto se basa en información que pretende mostrar la realidad desde diferentes puntos de vista y la evaluación de investigaciones que demuestren sus conclusiones, es así también, el reporte de resultados que el mismo demuestre.

2.6.4. Originalidad

Toda la información que se incluya en el presente trabajo, será citada oportunamente para demostrar la originalidad del mismo y demostrar la ocurrencia y citaciones de investigaciones de otros proyectos.

2.6.5. Veracidad

La información plasmada en el presente trabajo tendrá registrado todos los procesos en cada acción en archivos digitalizados, así también en cuanto al contenido para su posterior auditoria y demostrar la veracidad de los datos.

2.7. Criterios de rigor científico.

2.7.1. Confiabilidad

Se realizará los registros de los datos, ya sea ingreso, edición, y listado calculando y demostrando los criterios ya establecidos.

2.7.2. Validación

El presente proyecto será validado por los expertos de proyectos y su juicio pertinente, a su vez también será entregado los resultados del mismo con la estructura de base de datos y la aplicación.

III. RESULTADOS

3.1. Resultados en tablas y gráficos.

3.1.1. Tiempo de velocidad y respuesta (ml)

a) Escenario sobre 10000 Registros

Tabla 3
Ingreso de 10000 registros en velocidad y respuesta

Ingreso de 10000 registros			
Terminal	Registros	MySQL	PostgreSQL
		Promedio mlsg	Promedio mlss
Pc1	3334	0.0551	0.0087
Pc2	3333	0.0559	0.0097
Pc3	3333	0.0621	0.0094

Fuente: Elaboración propia

El análisis de la tabla para el ingreso de 10000 registros, observamos que el promedio en milisegundos en cada uno de tres equipos, MySQL demanda mayor tiempo de velocidad y respuesta.

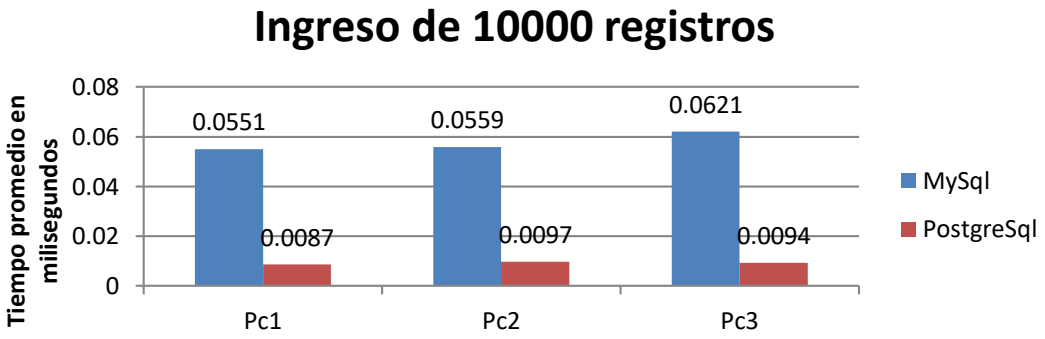

Figura 1 Ingreso de 10000 registros

Observamos en la gráfica que el promedio entre los tres equipos en milisegundos, MySQL demanda mayor tiempo en el registro de los mismos a diferencia de PostgreSQL cuya velocidad de respuesta es muy corta.

Tabla 4
Consulta sobre 10000 registros en velocidadny respuesta

Consulta de Registros			
Terminal	Registros	MySQL	PostgreSQL
		Promedio mlsg	Promedio mlsg
Pc1	1000	0.0351	0.0137
Pc2	1000	0.0398	0.0136
Pc3	1000	0.0315	0.0134

Fuente: Elaboración propia

La consulta de 1000 registros en cada equipo muestra a MySQL con mayor tiempo de velocidad y respuesta para este proceso a diferencia de PostgreSQL.

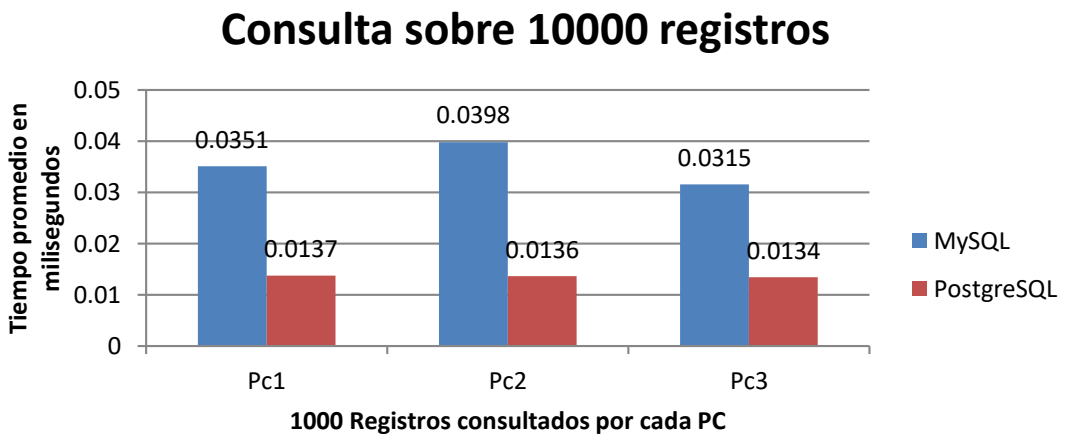

Figura 2 Consulta sobre 10000 registros

Se puede observar en la gráfica que MySQL en la consulta de registros demanda mayor tiempo de velocidad y respuesta para este proceso a diferencia de PostgreSQL.

Tabla 5
Actualización sobre 10000 registros en velocidad y respuesta

Actualización de Registros			
Terminal	Registros	MySQL	PostgreSQL
		Promedio mlsg	Promedio mlsg
Pc1	1000	0.0757	0.0118
Pc2	1000	0.0751	0.0108
Pc3	1000	0.0757	0.0119

Fuente: Elaboración propia

En el proceso de actualización, el análisis muestra a MySQL con mayor tiempo de velocidad y respuesta a diferencia de PostgreSQL con menor tiempo.

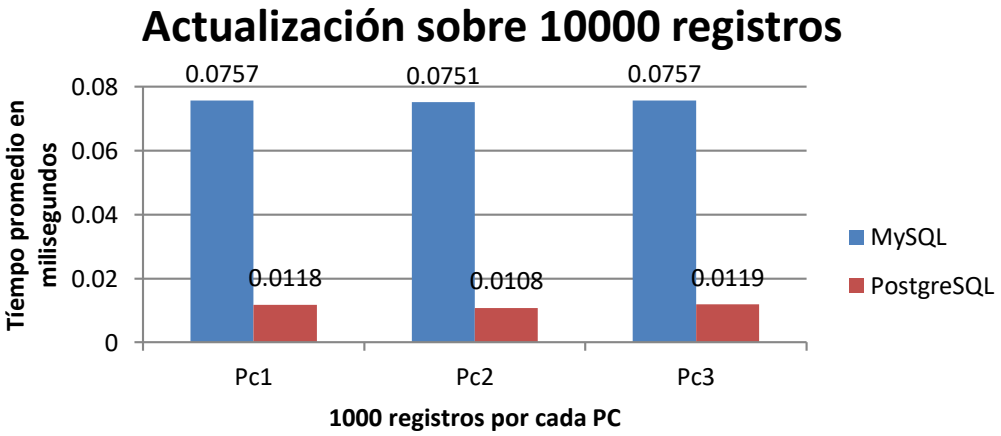

Figura 3 Actualización sobre 10000 registros

Se observa en la gráfica que MySQL demanda mayor tiempo de velocidad y respuesta en la actualización de registros a diferencia de PostgreSQL.

Tabla 6
Eliminación sobre 10000 registros en velocidad y respuesta

Eliminación de Registros			
Terminal	Registros	MySQL	PostgreSQL
		Promedio mlsg	Promedio mlsg
Pc1	1000	0.0742	0.0113
Pc2	1000	0.0743	0.0105
Pc3	1000	0.0729	0.0109

Fuente: Elaboración propia

En el proceso de eliminación de 1000 registros en cada equipo, MySQL demanda mayor tiempo de velocidad y respuesta a diferencia de PostgreSQL que tiene menor tiempo de respuesta.

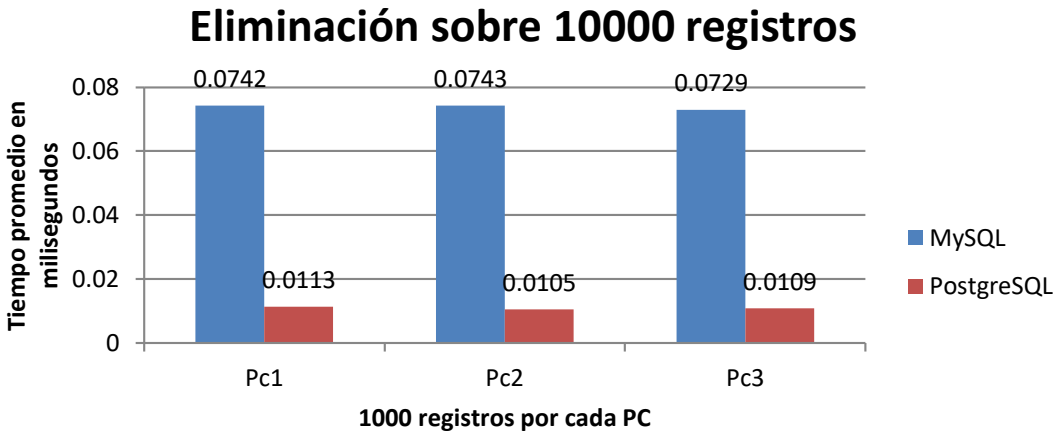

Figura 4 Eliminación sobre 10000 registros

Observamos en la gráfica, que el proceso de eliminación de 1000 registros en cada equipo, MySQL demanda mayor tiempo de velocidad y respuesta a diferencia de PostgreSQL.

b) Escenario sobre 100000 Registros

Tabla 7
Ingreso de 100000 registros en velocidad y respuesta

Ingreso de 100000 Registros			
Terminal	Registros	MySQL	PostgreSQL
		Promedio mlsg	Promedio mlsg
Pc1	33334	0.0618	0.0084
Pc2	33333	0.0561	0.0077
Pc3	33333	0.0560	0.0079

Fuente: Elaboración propia

Para el ingreso de 100000 registros, el análisis muestra que, el promedio en milisegundos en cada uno de los tres equipos, MySQL demanda mayor tiempo en el proceso de guardar registros.

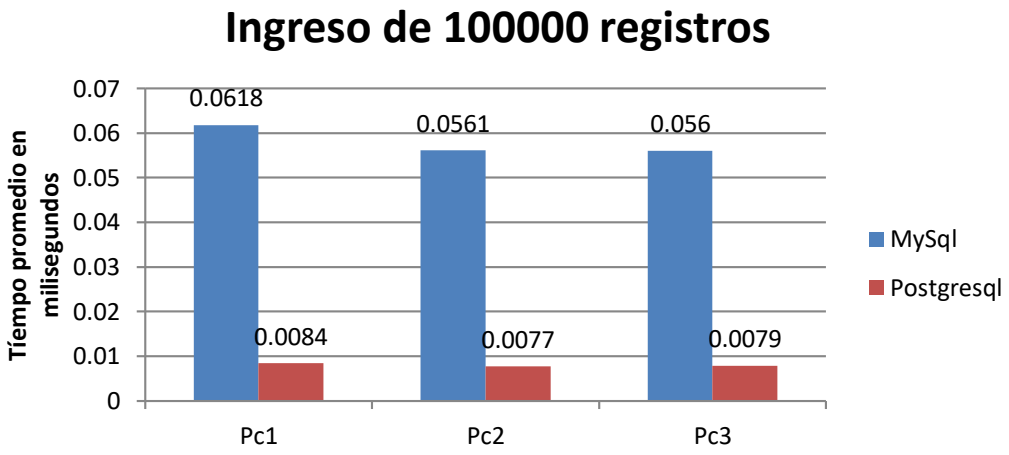

Figura 5 Ingreso de 100000 registros

Se puede observar que, para el ingreso de 100000 registros, el promedio en milisegundos en cada uno de tres equipos, MySQL demanda mayor tiempo en el proceso de grabar registros.

Tabla 8
Consulta sobre 100000 registros en velocidad y respuesta

Consulta de registros			
Terminal	Registros	MySQL	PostgreSQL
		Promedio mlsg	Promedio mlsg
Pc1	1000	0.1453	0.0212
Pc2	1000	0.1472	0.0237
Pc3	1000	0.1425	0.0221

Fuente: Elaboración propia

En la consulta de 1000 registros por cada equipo, la tabla muestra a MySQL con mayor tiempo de velocidad y respuesta para este proceso a diferencia de PostgreSQL.

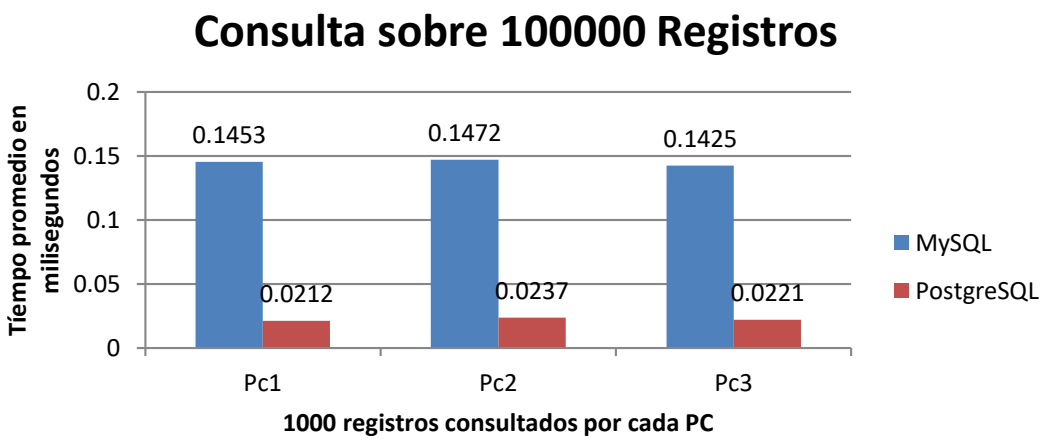

Figura 6 Consulta sobre 100000 registros

Observamos que, en la consulta de 1000 registros por cada equipo, MySQL demanda mayor tiempo de velocidad y respuesta para este proceso.

Tabla 9
Actualización sobre 100000 registros en velocidad y respuesta

Actualización de registros			
Terminal	Registros	MySQL	PostgreSQL
		Promedio mlsg	Promedio mlsg
Pc1	1000	0.1346	0.0318
Pc2	1000	0.1413	0.0388
Pc3	1000	0.1394	0.0352

Fuente: Elaboración propia

En actualización de 1000 registros en cada equipo, la tabla muestra a MySQL con mayor tiempo de velocidad y respuesta a diferencia de PostgreSQL.

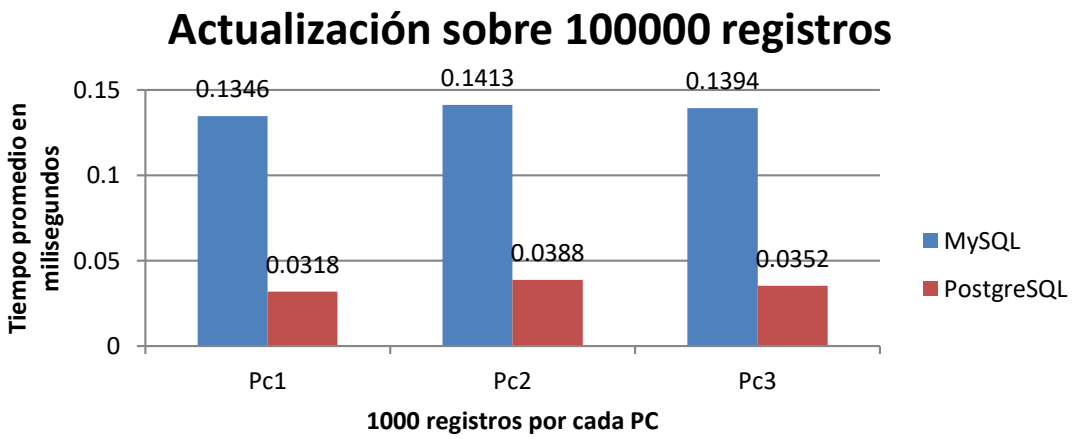

Figura 7 Actualización sobre 100000 registros

Observamos en la gráfica que, en la actualización de 1000 registros en cada equipo, MySQL demanda mayor tiempo de velocidad y respuesta a diferencia de PostgreSQL.

Tabla 10
Eliminación sobre 100000 registros en velocidad y respuesta

Eliminación de registros			
Terminal	Registros	MySQL	PostgreSQL
		Promedio mlsg	Promedio mlsg
Pc1	1000	0.1209	0.0229
Pc2	1000	0.1162	0.0256
Pc3	1000	0.1248	0.0231

Fuente: Elaboración propia

En el proceso de eliminación de 1000 registros en cada equipo, MySQL demanda mayor tiempo de velocidad y respuesta a diferencia, siendo PostgreSQL el que tiene menor tiempo de respuesta.

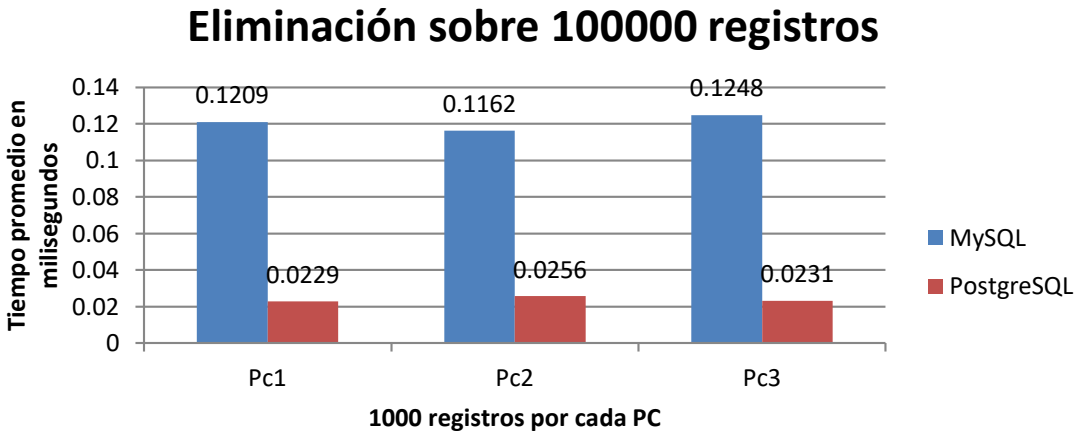

Figura 8 Eliminación sobre 100000 registros

Observamos que, para el proceso de eliminación de 1000 registros en cada equipo, MySQL demanda mayor tiempo de velocidad y respuesta a diferencia de PostgreSQL.

c) Escenario sobre 1001000 Registros

Tabla 11
Ingreso de 1001000 registros en velocidad y respuesta

Ingreso de 1001000 registros			
Terminal	Registros	MySQL	PostgreSQL
		Promedio mlsg	Promedio mlsg
Pc1	333715	0.0626	0.0075
Pc2	333726	0.0589	0.0068
Pc3	333559	0.0645	0.0074

Fuente: Elaboración propia

Para el ingreso de 1001000 registros, el análisis muestra que, el promedio en milisegundos en cada uno de los tres equipos, MySQL demanda mayor tiempo en el proceso de guardar registros.

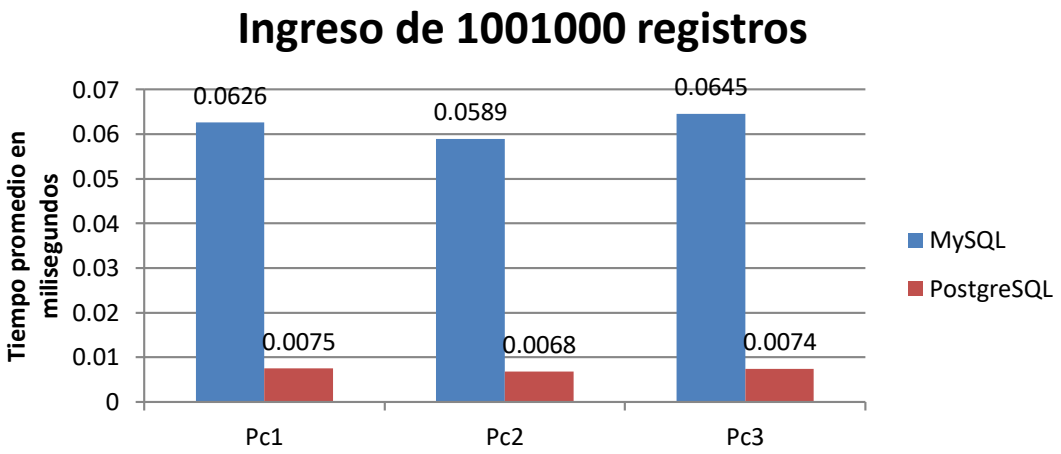

Figura 9 Ingreso de 1001000 registros

Se puede observar que, para el ingreso de 1001000 registros, el promedio en milisegundos en cada uno de tres equipos, MySQL demanda mayor tiempo en el proceso de grabar registros.

Tabla 12
Consulta sobre 1001000 registros en velocidad y respuesta

Consulta de registros			
Terminal	Registros	MySQL	PostgreSQL
		Promedio mlsg	Promedio mlsg
Pc1	1000	1.3915	0.1040
Pc2	1000	1.3925	0.1039
Pc3	1000	1.3862	0.1027

Fuente: Elaboración propia

En la consulta de 1000 registros por cada equipo, la tabla muestra a MySQL con mayor tiempo de velocidad y respuesta para este proceso a diferencia de PostgreSQL.

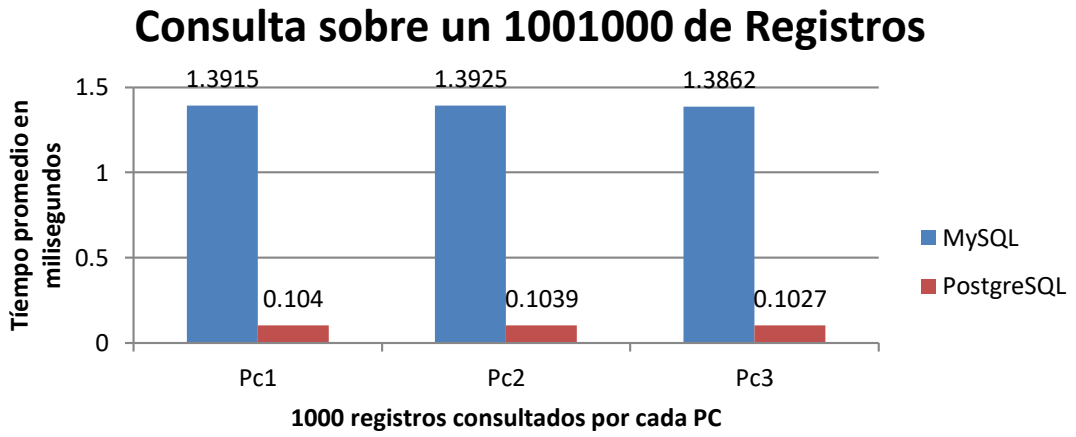

Figura 10 Consulta sobre 1001000 registros

Observamos que, en la consulta de 1000 registros por cada equipo, MySQL demanda mayor tiempo de velocidad y respuesta para este proceso.

Tabla 13
Actualización sobre 1001000 registros en velocidad y respuesta

Actualización de registros			
Terminal	Registros	MySQL	PostgreSQL
		Promedio mlsg	Promedio mlsg
Pc1	1000	0.7810	0.1209
Pc2	1000	0.7914	0.1471
Pc3	1000	0.7608	0.1340

Fuente: Elaboración propia

En actualización de 1000 registros en cada equipo, la tabla muestra a MySQL con mayor tiempo de velocidad y respuesta a diferencia de PostgreSQL.

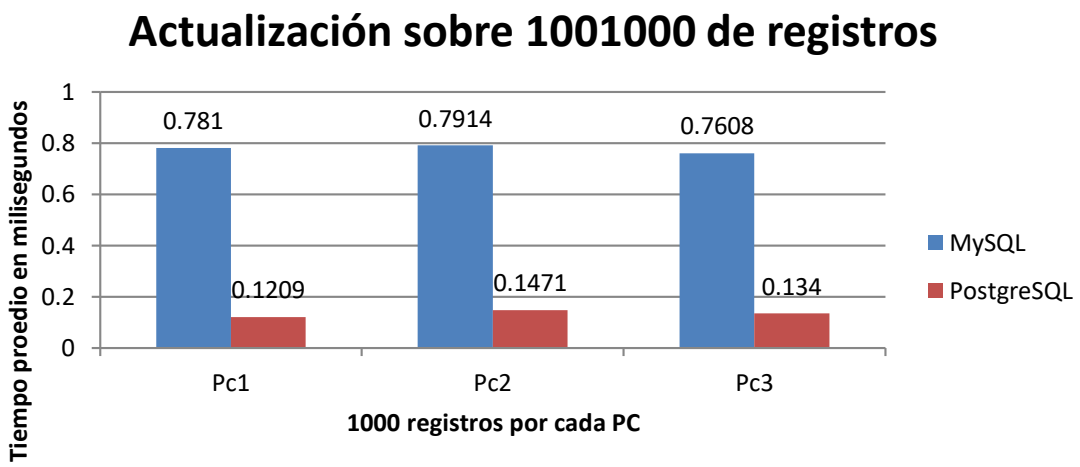

Figura 11 Actualización sobre 1001000 registros

Observamos en la gráfica que, en la actualización de 1000 registros en cada equipo, MySQL demanda mayor tiempo de velocidad y respuesta a diferencia de PostgreSQL.

Tabla 14
Eliminación sobre 1001000 registros en velocidad y respuesta

Eliminación de registros			
Terminal	Registros	MySQL	PostgreSQL
		Promedio mlsg	Promedio mlsg
Pc1	1000	0.8139	0.1053
Pc2	1000	0.8173	0.1068
Pc3	1000	0.8355	0.1063

Fuente: Elaboración propia

En el proceso de eliminación de 1000 registros en cada equipo, la tabla muestra a MySQL con mayor tiempo de velocidad y respuesta a diferencia de PostgreSQL.

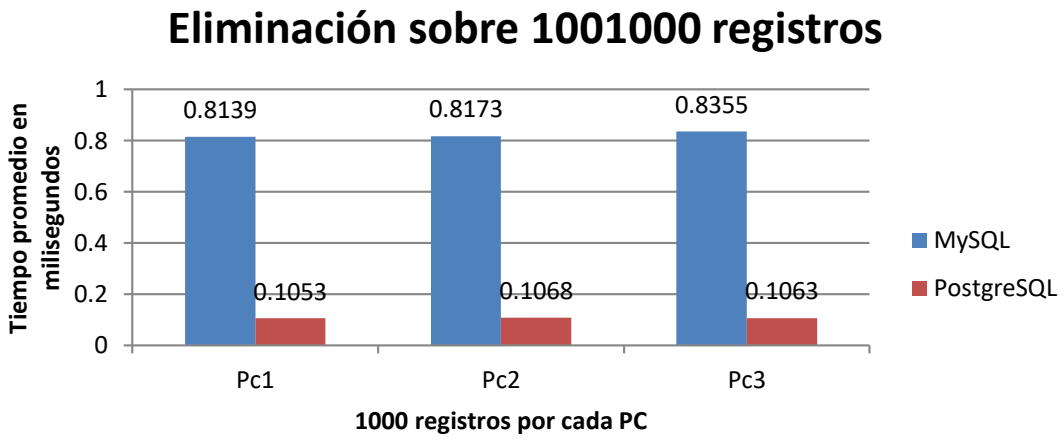

Figura 12 Eliminación sobre 1001000 registros

Observamos que, para el proceso de eliminación de 1000 registros en cada equipo, MySQL demanda mayor tiempo de velocidad y respuesta a diferencia de PostgreSQL.

3.1.1.1.1. Consumo de Memoria en Bytes

Tabla 15
Consumo de memoria en ingreso de registros

Promedio de Bytes privados en memoria (Ingreso de registros)		
Total registros	MySQL	PostgreSQL
10000 Reg	793333248	38262528
100000 Reg	791999605.9	38675420.69
1001000 Reg	793204896.7	48618331.62

Fuente: Elaboración propia

El análisis de la tabla nos muestra que, en el proceso de ingreso de registros en los tres escenarios, muestra a MySQL con mayor consumo de memoria.

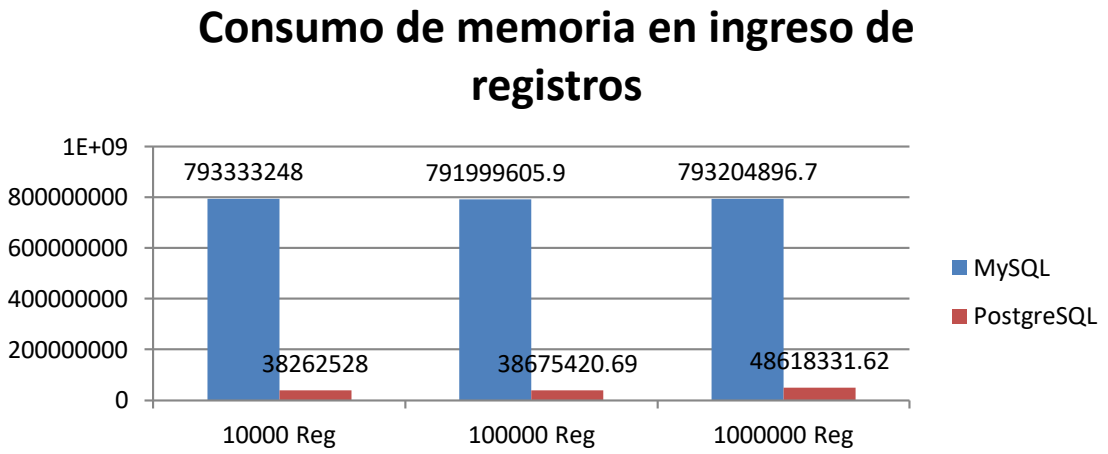

Figura 13 Consumo de memoria en ingreso de registros

Se puede observar en la gráfica que, en el ingreso de registros en los tres escenarios, MySQL demanda mayor consumo de memoria.

Tabla 16
Consumo de memoria en consulta de registros

Promedio de Bytes privados en memoria (Consulta de registros)		
Total registros	MySQL	PostgreSQL
10000 Reg	807302485.3	53659477.33
100000 Reg	808366080	55086359.27
1001000 Reg	818280408.4	54473356.1

Fuente: Elaboración propia

El análisis de la tabla nos muestra que, en el proceso de consulta de registros en los tres escenarios, muestra a MySQL con mayor consumo de memoria.

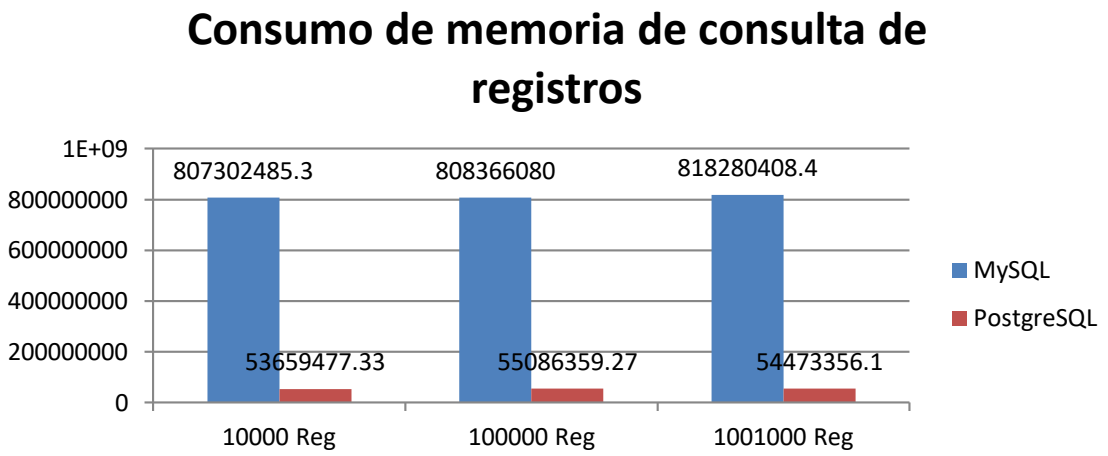

Figura 14 Consumo de memoria en consultas

Se puede observar en la gráfica que, en consulta de registros en los tres escenarios, MySQL demanda mayor consumo de memoria.

Tabla 17
Consumo de memoria en actualización de registros

Promedio de Bytes privados en memoria (Actualización de registros)		
Total registros	MySQL	PostgreSQL
10000 Reg	793756672	53731696.42
100000 Reg	790488300.3	55344911.06
1001000 Reg	784728518.7	43342496.84

Fuente: Elaboración propia

El análisis de la tabla nos muestra que, en el proceso de actualización de registros en los tres escenarios, nos muestra a MySQL con mayor consumo de memoria.

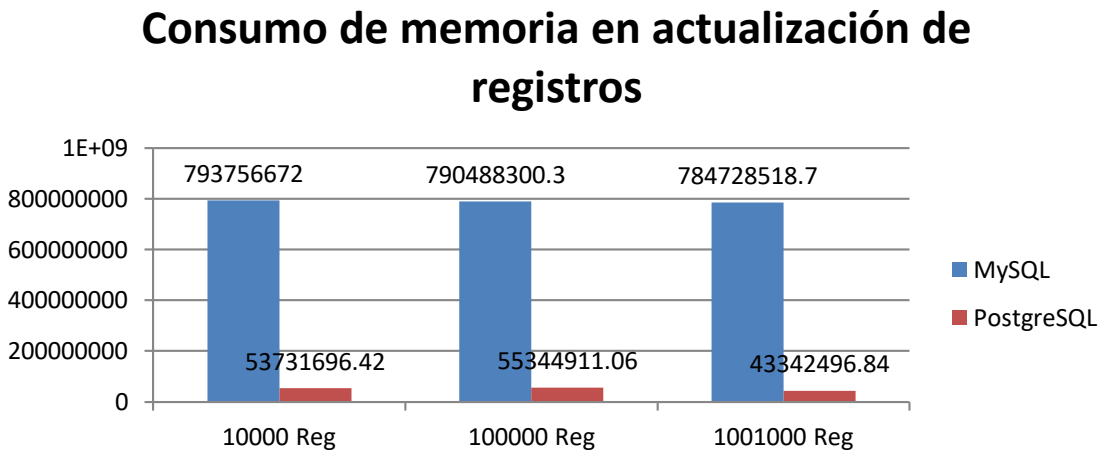

Figura 15 Consumo de memoria en actualizaciones

Se puede observar en la gráfica que, en la actualización de registros en los tres escenarios, MySQL demanda mayor consumo de memoria.

Tabla 18
Consumo de memoria en eliminación de registros

Promedio de Bytes privados en memoria (Eliminación de registros)		
Total registros	MySQL	PostgreSQL
10000 Reg	820602993.8	53966537.7
100000 Reg	818649842.5	55840494.93
1001000 Reg	811839405.1	54494303.05

Fuente: Elaboración propia

El análisis de la tabla nos muestra que, en el proceso de eliminación de registros en los tres escenarios, nos muestra a MySQL con mayor consumo de memoria.

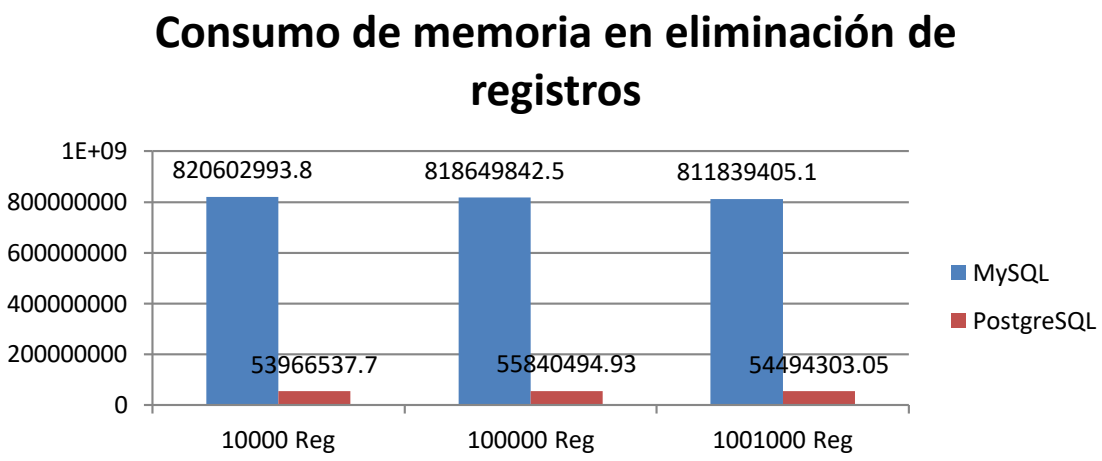

Figura 16 Consumo de memoria en eliminación

Se puede observar en la gráfica que, en la eliminación de registros en los tres escenarios, MySQL demanda mayor consumo de memoria.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.1.1.1.2. Consumo de CPU

Tabla 19
Consumo de CPU en ingreso de registros

Promedio de % de tiempo del procesador (Ingreso)		
Total registros	MySQL	PostgreSQL
10000 Reg	17.06	0.44
100000 Reg	42.94	0.04
1001000 Reg	119.07	0.03

Fuente: Elaboración propia

El análisis de la tabla nos muestra que, en el proceso de ingreso de registros en los tres escenarios, muestra a MySQL con mayor consumo de CPU.

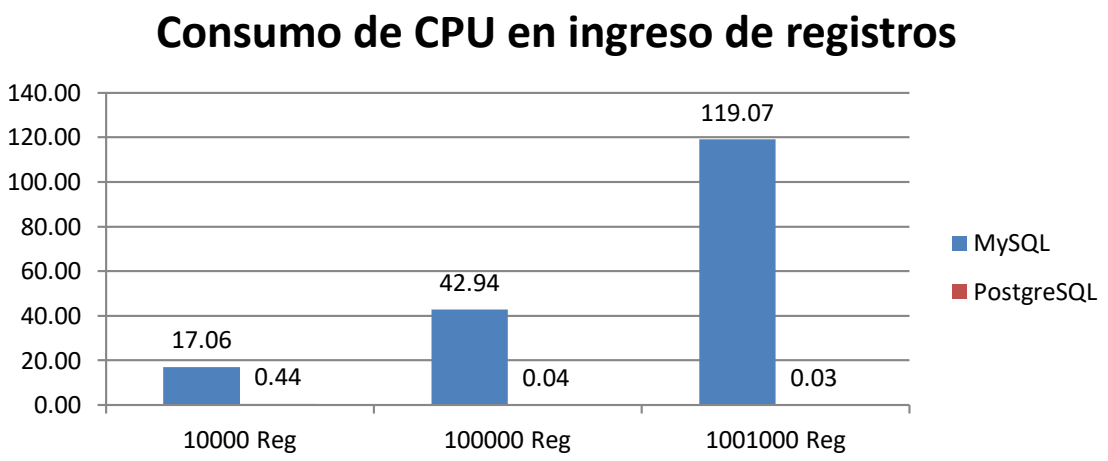

Figura 17 Consumo de CPU en ingreso de registros

Se puede observar en la gráfica que, en el ingreso de registros en los tres escenarios, MySQL demanda mayor consumo de CPU.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Tabla 20
Consumo de CPU en consulta de registros

Promedio de % de tiempo del procesador (Consulta)		
Total registros	MySQL	PostgreSQL
10000 Reg	39.86	11.13
100000 Reg	126.20	32.99
1001000 Reg	262.88	100.60

Fuente: Elaboración propia

El análisis de la tabla nos muestra que, en el proceso de consulta de registros en los tres escenarios, muestra a MySQL con mayor consumo de CPU.

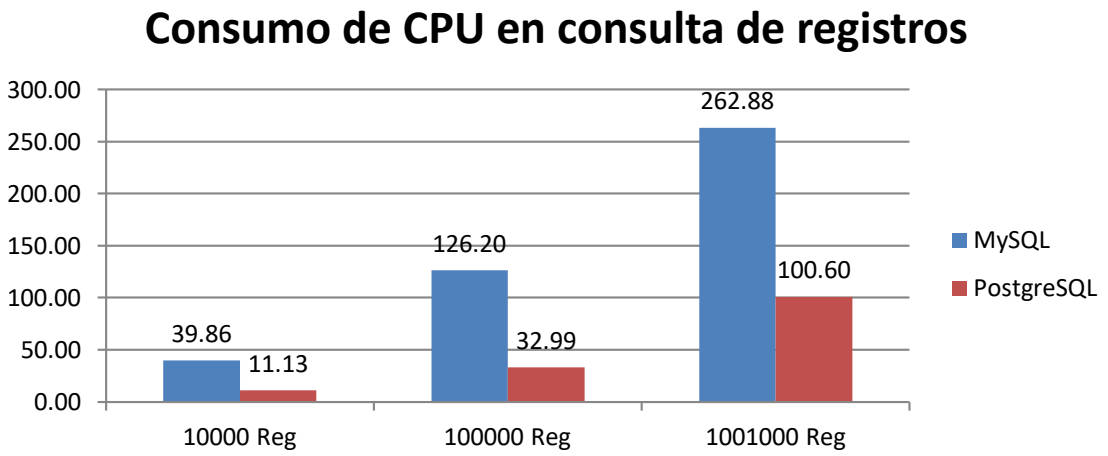

Figura 18 Consumo de CPU en Consulta de registros

Se observa en la gráfica que, en la consulta de registros en los tres escenarios, MySQL demanda mayor consumo de CPU.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Tabla 21
Consumo de CPU en actualización de registros

Promedio de % de tiempo del procesador (Actualización)		
Total registros	MySQL	PostgreSQL
10000 Reg	24.70	5.13
100000 Reg	70.35	16.22
1001000 Reg	206.66	82.64

Fuente: Elaboración propia

El análisis de la tabla nos muestra que, en el proceso de actualización de registros en los tres escenarios, nos muestra a MySQL con mayor consumo de CPU.

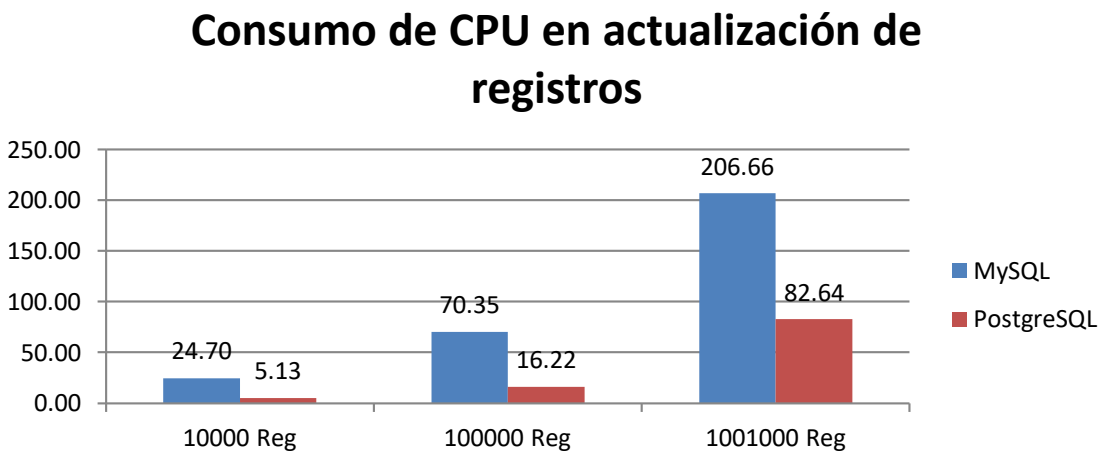

Figura 19 Consumo de CPU en actualizaciones de registros

Se puede observar en la gráfica que, en la actualización de registros en los tres escenarios, MySQL demanda mayor consumo de CPU.

Tabla 22
Consumo de CPU en eliminación de registros

Promedio de % de tiempo del procesador (Eliminación)		
Total registros	MySQL	PostgreSQL
10000 Reg	21.07	5.96
100000 Reg	71.02	22.34
1001000 Reg	147.88	74.65

Fuente: Elaboración propia

El análisis de la tabla nos muestra que, en el proceso de eliminación de registros en los tres escenarios, nos muestra a MySQL con mayor consumo de CPU.

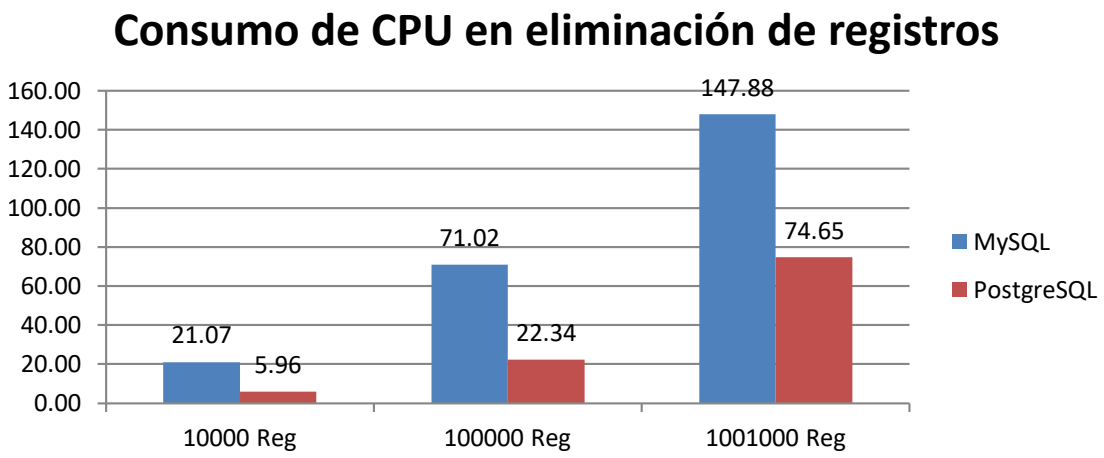

Figura 20 Consumo de CPU en eliminación de registros

Se puede observar en la gráfica que, en la eliminación de registros en los tres escenarios, MySQL demanda mayor consumo de CPU.

d) Escenario de pruebas con imágenes

En el siguiente escenario tendremos los resultados de pruebas realizados con imágenes obtenidos del estándar DICOM y convertidos a formato JPG y PNG, con los cuales se almacenará en MySQL y PostgreSQL.

Tiempo de velocidad y respuesta Ingreso de registros con imágenes

Tabla 23
Velocidad y respuesta en ingreso de imágenes

	MySQL	PostgreSQL
Pc1	0.1080	0.0575
Pc2	0.1134	0.0573
Pc3	0.1275	0.0912

Fuente: Elaboración propia

El análisis de la tabla nos muestra que, en el proceso de ingresar datos con imágenes con los tres equipos, nos muestra a MySQL con mayor tiempo de velocidad y respuesta.

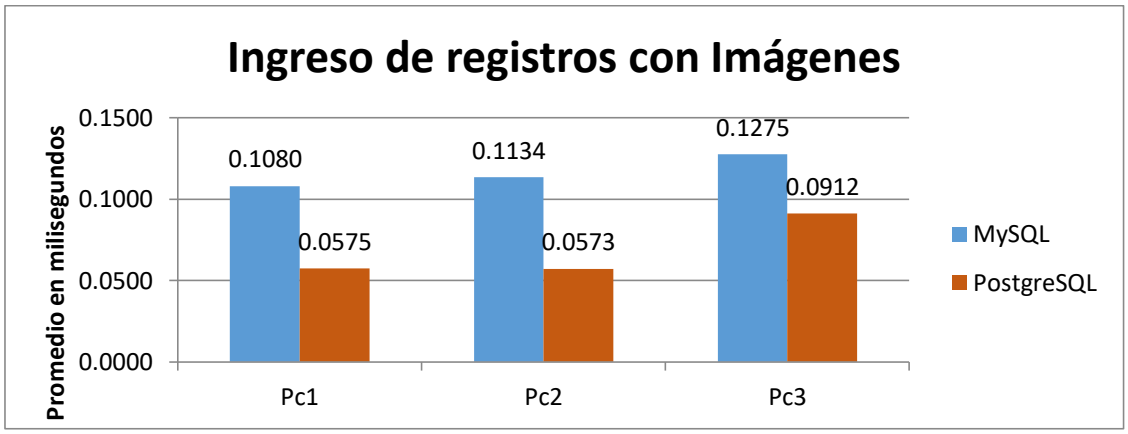

Figura 21 Promedio en ingreso de imágenes en MySQL y PostgreSQL

Se puede observar en la gráfica que, en el ingreso de registros con imágenes en los tres equipos, nos muestra a MySQL con mayor tiempo de velocidad y respuesta.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Tiempo de velocidad y respuesta consulta de registros con imágenes

Tabla 24
Consulta de imágenes MySQL y PostgreSQL

	MySQL	PostgreSQL
Pc1	200 reg. 4.3913	18 reg. 0.1568
Pc2	200 reg. 4.3985	9 reg. 0.13
Pc3	200 reg. 3.6313	18 reg. 0.1601

Fuente: Elaboración propia

El análisis de la tabla nos muestra que en el proceso de consultar datos con imágenes con los tres equipos evidenciamos que MySQL ha realizado 200 consultas con un promedio alto en milisegundos en velocidad de respuesta, pero PostgreSQL solamente ha realizado entre 9 a 18 consultas.

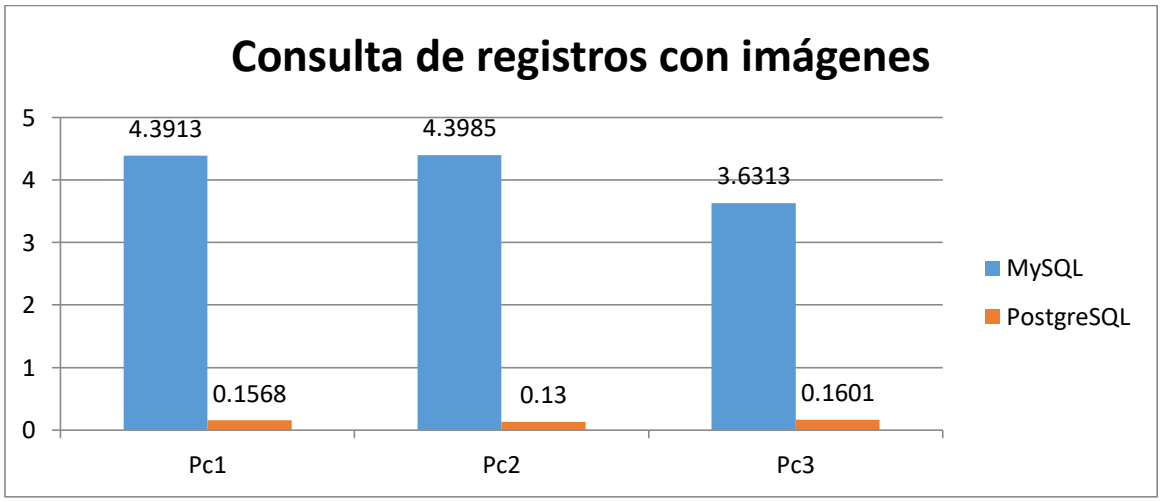

Figura 22 Velocidad y respuesta en consulta de imágenes

Observamos un mayor tiempo de velocidad en respuesta por parte de MySQL y menor tiempo para PostgreSQL, pero según nuestra tabla de información, MySQL realiza mayor cantidad de consultas y PostgreSQL no cumplió con los requerimientos. La observación consiste en que las pruebas de consulta para PostgreSQL se restringieron debido a la saturación y perdía en el tiempo de espera al obtener una conexión con la base de datos, en cambio MySQL continuaba realizando consultas con prolongado tiempo de velocidad y respuesta.

Tiempo de velocidad y respuesta eliminación de registros con imágenes

Tabla 25
Eliminación de imágenes en MySQL y PostgreSQL

	MySQL	PostgreSQL
Pc1	50 reg. 0.0144	8 reg. 0.046
Pc2	50 reg. 0.0128	9 reg. 0.04825
Pc3	50 reg. 0.0085	8 reg. 0.0441

Fuente: Elaboración propia

El análisis de la tabla nos muestra que en el proceso de eliminar datos con imágenes con los tres equipos evidenciamos que MySQL ha realizado 50 eliminaciones con un promedio alto en milisegundos en velocidad de respuesta, pero PostgreSQL solamente ha realizado entre 8 a 9 eliminaciones.

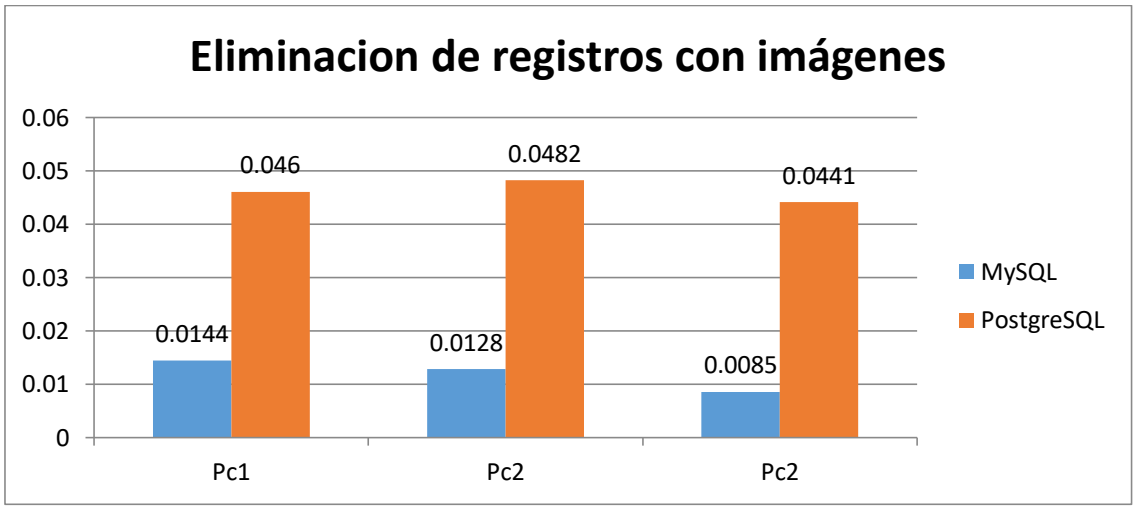

Figura 23 Velocidad y respuesta en eliminar imágenes

Observamos un mayor tiempo de velocidad en respuesta por parte de MySQL y menor tiempo para PostgreSQL, pero según nuestra tabla de información, MySQL realiza mayor cantidad de eliminaciones y PostgreSQL no cumplió con los requerimientos.

Consumo de CPU en ingreso de registros

Tabla 26
Consumo de CPU en ingreso de imágenes

Ingreso de imágenes con 3000 registros		
	MySQL	PostgreSQL
En los 3 terminales	141.3957	0.1094

Fuente: Elaboración propia

En nuestra tabla nos muestra que MySQL tiene un mayor consumo de CPU en ingreso de registros con imágenes con 3000 registros.

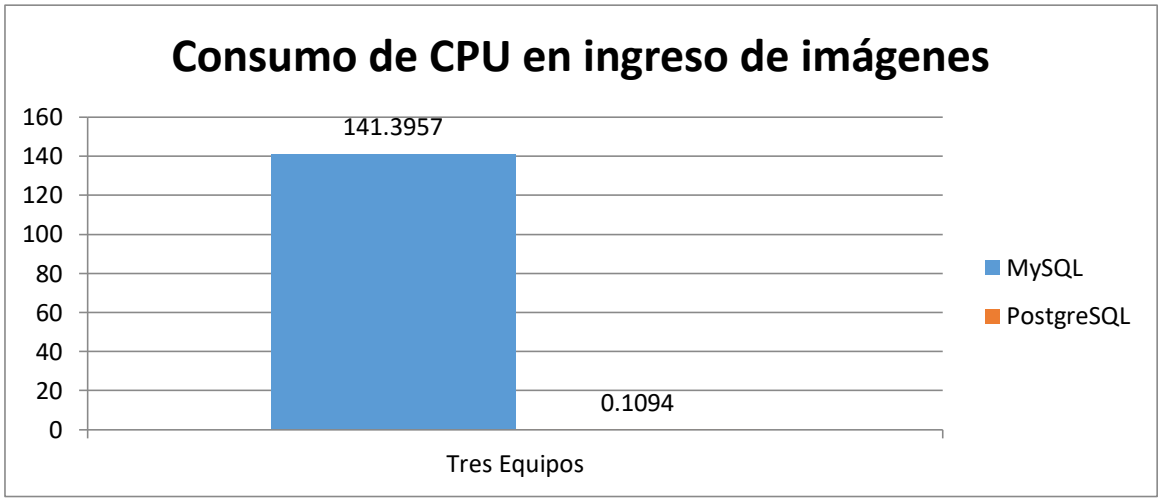

Figura 24 Consumo de CPU en ingreso con imágenes

Observamos que MySQL consume gran porcentaje de CPU en el ingreso de registros con imágenes.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Consumo de memoria en Bytes para ingreso de registros con imágenes

Tabla 27
Consumo de memoria en ingreso de imágenes

Ingreso de imágenes con 3000 registros		
	MySQL	PostgreSQL
En los 3 terminales	811412969.7	41166848

Fuente: Elaboración propia

En nuestra tabla nos muestra que MySQL tiene un mayor consumo de memoria en ingreso de registros con imágenes con 3000 registros.

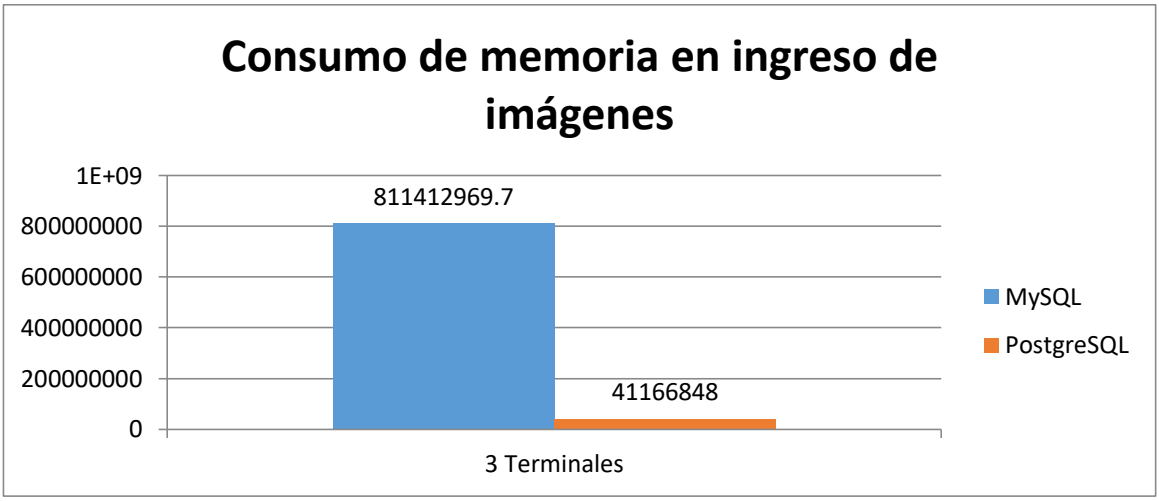

Figura 25 Consumo de memoria en ingreso de imágenes

Observamos que MySQL consume gran cantidad de memoria en el ingreso de registros con imágenes.

Consumo memoria Bytes para Consulta y eliminación: no se llevó a cabo por inconsistencia de los procesos por parte de PostgreSQL, lo cual se observa en la Tabla 25 y 26.

Consumo de CPU para Consulta y eliminación: no se llevó a cabo por inconsistencia de los procesos por parte de PostgreSQL, lo cual se observa en la Tablas 25 y 26.

3.2. Discusión de resultados

Con los resultados obtenidos inicialmente, producto de los procesos realizados y que tienen diferentes unidades de medición:

Tiempo de Velocidad y respuesta; expresado en milisegundos (ms)

Consumo de memoria RAM; expresado en bytes (B)

Consumo de CPU; expresado en (%) porcentaje de tiempo

Tomando en cuenta investigaciones previas a este trabajo, (Portilla & Bernal, 2018) que realizaron pruebas con un esquema, incluyendo varios registros, realizando el recorrido por diferentes sentencias de lenguaje estructurado de consultas, concluyen que MySql es rápido en base de datos con pequeñas estructuras, pero lento con grandes cantidades de datos a diferencia de Oracle. Para nuestro trabajo hemos considerado la utilización de una estructura grande de base de datos con uso de 10000, 100000 y 1001000 registros con pruebas de carga con procesos CRUD, en los gestores de bases de datos MySQL y PostgreSQL, observándose que este último tiene un mayor rendimiento en cuanto a tiempo de velocidad y respuesta, así también en el consumo de CPU y memoria RAM. Observamos que para el manejo de imágenes, los dos gestores de permiten ingresar estos objetos, pero al realizar consultas y eliminaciones, PostgreSQL deja de obtener dicha información ya que perdía conexión por el tiempo de espera para sus procesos a diferencia de MySQL que si continuaba realizando sus consultas y eliminaciones.

3.3. Aporte práctico

En la siguiente gráfica exponemos nuestro aporte práctico.

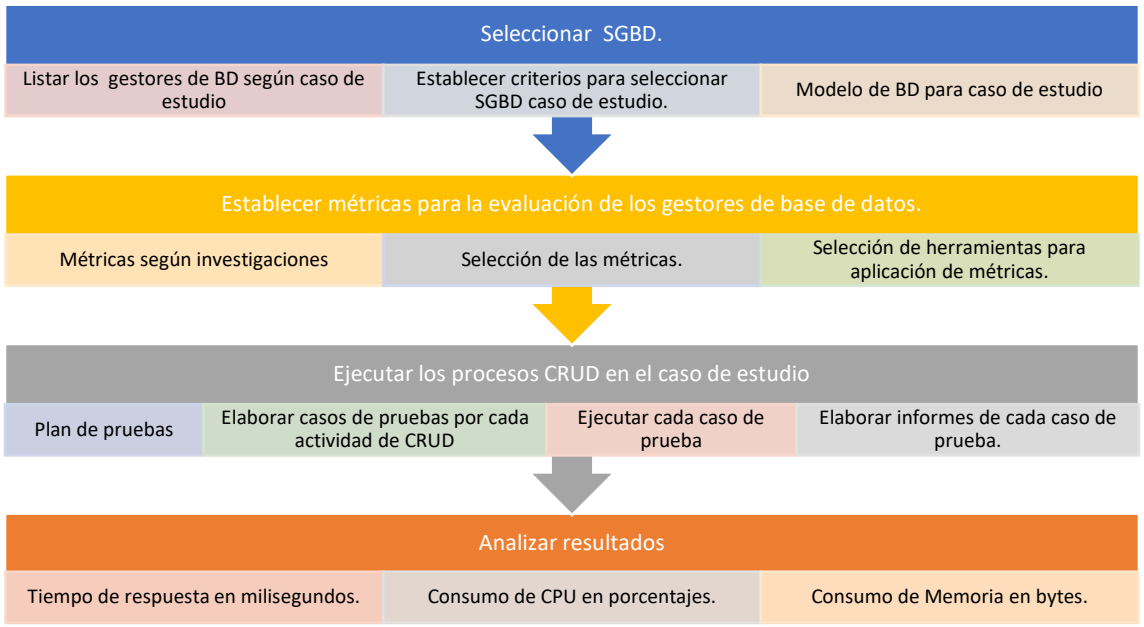

Figura 26 Aporte práctico

3.3.1. Seleccionar SGBD

3.3.1.1. Listar gestores de BD según caso de estudio

A continuación, presentamos en la Tabla XXVIII, una lista del ranking de las bases de datos con mayor popularidad.

Tabla 28
SGBD con mayor popularidad

Rank			DBMS	Database Model	Score		
Apr 2019	Mar 2019	Apr 2018			Apr 2019	Mar 2019	Apr 2018
1.	1.	1.	Oracle	Relational, Multi-model	1279.94	+0.80	-9.85
2.	2.	2.	MySQL	Relational, Multi-model	1215.14	+16.89	-11.26
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	1059.96	+12.11	-35.55
4.	4.	4.	PostgreSQL	Relational, Multi-model	478.72	+8.91	+83.25
5.	5.	5.	MongoDB	Document	401.98	+0.64	+60.57
6.	6.	6.	IBM Db2	Relational, Multi-model	176.05	-1.15	-12.89
7.	↑ 8.	↑ 9.	Redis	Key-value, Multi-model	146.38	+0.25	+16.27
8.	↑ 9.	8.	Elasticsearch	Search engine, Multi-model	146.00	+3.21	+14.64
9.	↓ 7.	↓ 7.	Microsoft Access	Relational	144.65	-1.55	+12.43
10.	10.	↑ 11.	SQLite	Relational	124.21	-0.66	+8.23

Nota: Recuperado de https://db-engines.com/en/ranking

MySQL y PostgreSQL se encuentran dentro del ranking de los primeros gestores de bases de datos según fuente DB-Engines.com.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.3.1.2. Establecer criterios para seleccionar SGBD caso de estudio.

A continuación, se presenta criterios de información sobre los gestores de bases de datos identificando otras características para el presente trabajo.

Tabla 29
Criterios para seleccionar los SGBD

	Oracle	MySQL	Ms SQL	PostgreSQL	MongoDB
Libre		X		X	
Solo en la nube					
Esquema de datos	X	X	X	X	X
Soporte XML	X	X	X	X	X
Índices secundarios	X	X	X	X	X
SQL	X	X	X	X	
Llaves extranjeras	X	X	X	X	
Concurrencia	X	X	X	X	X
Total	6	7	6	7	4

Fuente elaboración propia

Nota:

Obtenido y recuperado de https://db-engines.com/en/ranking

Tabla 30
Servidores para SGBD

	Oracle	MySQL	Ms SQL	PostgreSQL	MongoDB
Free BSD		X		X	
HP- UX	X			X	
Linux	X	X	X	X	X
Net BSD				X	
OS X	X	X		X	X
Solaris	X	X		X	X
Unix				X	
Windows	X	X	X	X	X
Total	5	5	2	8	4

Fuente elaboración propia

Nota:

Obtenido y recuperado de https://db-engines.com/en/ranking

En base a los resultados de las dos tablas anteriormente expuestas, se puede determinar el criterio para poder realizar el estudio sobre los dos gestores de bases de datos MySQL y PostgreSQL.

3.3.1.3. Modelo de bases de datos para caso de estudio.

Para nuestro caso de estudio se planteó utilizar un modelo de base de datos aplicado a un hospital, en donde podemos detallar los procesos a realizar. El modelo a tomar en cuenta de acuerdo al siguiente flujograma:

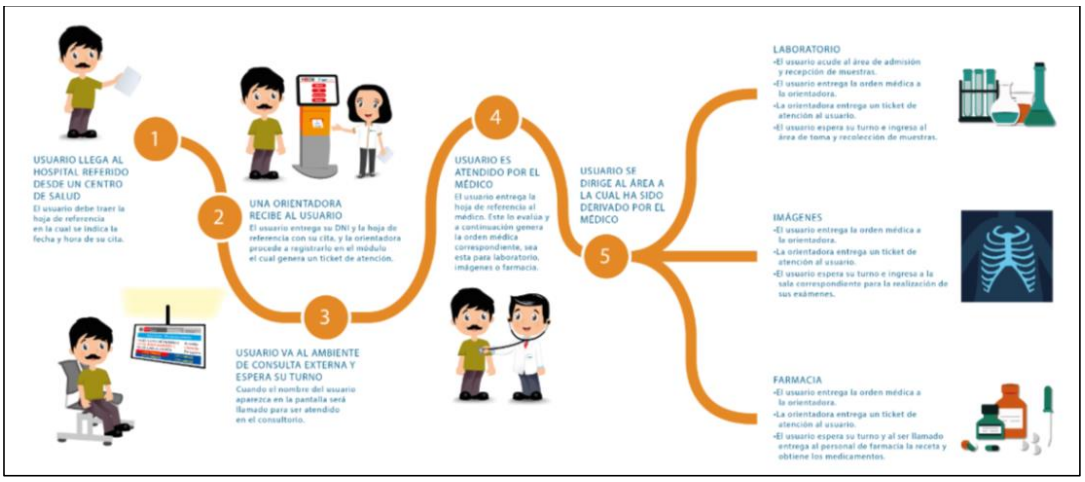

Figura 27 Flujograma de la asistencia en un hospital

Nota: Recuperado de https://www.heves.gob.pe/portal/como-me-atiendo/

Para el diseño de la base de datos se ha tomado en cuenta un modelo del cual tendrá las mismas características en ambos gestores de bases de datos (SGBD), tomando en cuenta el flujograma antes expuesto.

3.3.2. Establecer métricas para la evaluación de los gestores de base de datos

3.3.2.1. Métricas según investigaciones

Según los artículos previos a este trabajo presentamos métricas usadas por investigaciones para evaluar nuestro caso de estudio.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Tabla 31
Métricas según investigaciones

Articulo	Métrica	Autores
1. Una Comparación del rendimiento de base de datos de MariaDB y MySql con la carga de trabajo OLTP	Consumo de CPU Consumo de RAM	S. Tongkaw A. Tongkaw
2. Comparación Del Rendimiento De Los Comandos Insert, Select Y Delete En Los Sistemas Gestores De Bases De Datos Oracle Y Mysql	Tiempo de respuesta	J. Portilla Tovar, M. Bernal Gómez
3. Identificación de las principales fuentes de variación en las latencias de transacción: hacia bases de datos más predecibles	Variación de latencia	J. Huang, B. Mozafari, G. Schoenebeck y W. T.
4. La Evaluación del rendimiento de las operaciones CRUD en una base de datos orientada a documentos replicados de forma asíncrona.	Tiempo de ejecución y respuesta	C. Truica, F. Radulescu, Boicea A., and Bucur I.
5. Análisis comparativo de los sistemas de gestión de bases de datos relacionales seleccionados.	Tiempo de velocidad de consultas	Poljak, R.; Poščić, P., Jakšić, D.
6. Una breve comparación de dos sistemas RDBMS de clase empresarial.	Estándares de código SQL Tipos de campos Columnas calculadas Índices de diseño DML disparadores	Figueroa, A. and Rollo, S. and Murthy, S.

Fuente: Elaboración propia

Tabla 32
Calificación de métricas

Métrica	Investigaciones						
	1	2	3	4	5	6	Total
Consumo de CPU	X						1
Consumo de RAM	X						1
Tiempo de respuesta		X		X	X		3
Estándares de código SQL						X	1
Tipos de campos						X	1
Columnas calculadas						X	1
Índices de diseño						X	1
DML disparadores						X	1
Variación de latencia			X				1

Fuente: Elaboración propia

Con la información obtenida en la tabla se observa que tiempo de respuesta tienen mayor puntuación, considerando además que consumo de CPU y RAM tienen igual puntuación que el resto de métricas.

3.3.2.2. Selección de métricas

Para nuestro caso de estudio tomaremos en cuenta el tiempo de velocidad y respuesta además de Consumo de CPU y RAM como métricas, apropiadas para el trabajo de investigación.

3.3.2.3. Selección de herramientas para aplicación de métricas

3.3.2.3.1. Lenguajes de programación

Tabla 33
Lenguajes de programación más populares

Apr 2019	Apr 2018	Lenguajes de programación	Ratings	Change
1	1	Java	15.035%	-0.74%
2	2	C	14.076%	+0.49%
3	3	C++	8.838%	+1.62%
4	4	Python	8.166%	+2.36%
5	6	Visual Basic .NET	5.795%	+0.85%
6	5	C#	3.515%	-1.75%
7	8	JavaScript	2.507%	-0.99%
8	9	SQL	2.272%	-0.38%
9	7	PHP	2.239%	-1.98%
10	14	Assembly language	1.710%	+0.05%

Nota: Recuperado de https://www.tiobe.com/tiobe-index/

Para nuestro trabajo se ha creído conveniente utilizar el lenguaje de programación Visual Basic Net para el diseño de nuestra aplicación, por estar en los 5 primeros lugares entre los más populares y que se usara para diseñar la aplicación y realizar las pruebas de carga y medir el tiempo de velocidad y respuesta.

3.3.2.3.2. Herramienta para Consumo de recursos

Windows cuenta con herramientas que permiten observar y recopilar información de los procesos y consumo del sistema, entre ellos tenemos Rendimiento, Conjunto de recopiladores de datos, entre otras opciones gráficas.

Monitor de rendimiento de Windows

"El Monitor de rendimiento es una herramienta de visualización simple pero potente para ver datos de rendimiento, tanto en tiempo real como desde archivos de registro. Con él puede examinar los datos de rendimiento en un gráfico, histograma o informe".

Conjuntos de recopiladores de datos

Esta opción es un complemento del Monitor de rendimiento y nos permite recopilar información de los sucesos, pero que también puede ser configurado para trabajar independientemente y obtener datos de la performance del sistema.

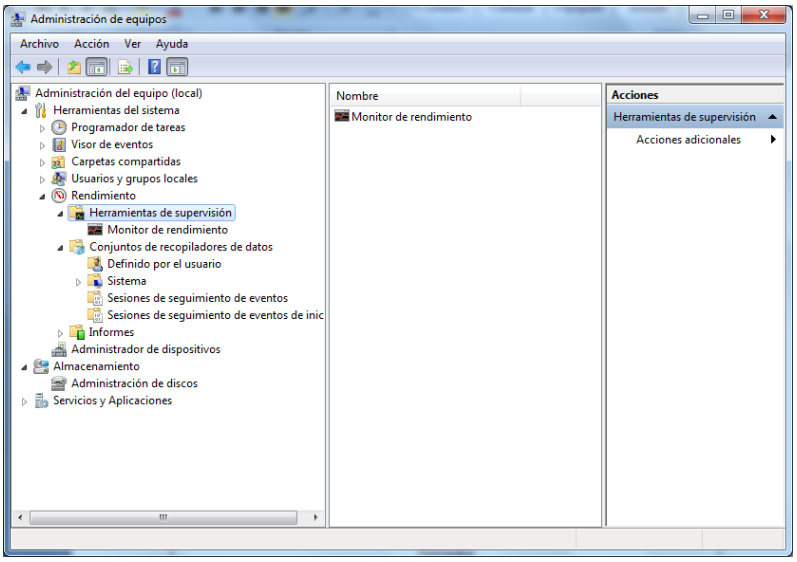

Figura 28 Conjunto de recopilador de datos

Fuente: Microsoft documentación https://bit.ly/3ckMNvZ

3.3.3. Ejecutar los procesos CRUD en el caso de estudio

3.3.3.1. Plan de pruebas

3.3.3.1.1. Determinación de los escenarios físicos

Las pruebas se realizarán en escenarios iguales para ambos SGBS, bajo una arquitectura de red distribuida que se requiere de un servidor, la base de datos llevara de nombre **rendimiento** para ambos SGBD MySql y PostgreSQL, dichas pruebas se ejecutarán con tres terminales clientes conectados mediante un switch al servidor, las cuales se detallan a continuación:

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Tabla 34
Características del servidor

Servidor	
Tipo	Escritorio
Procesador	Intel (R) Core (TM) i7-4790
Potencia	3.60 GHz
Memoria	8 GB
Disco Duro	1 TB
Sistema Operativo	Windows 7 Professional

Fuente: Elaboración propia

Tabla 35
Características del terminal1

Terminal1	
Tipo	Laptop
Procesador	Intel (R) Core (TM) i5 4210u
Potencia	2.70 GHz
Memoria	6 GB
Disco Duro	1 TB
Sistema Operativo	Windows 8.1

Fuente: Elaboración propia

Tabla 36
Características del terminal2

Terminal2	
Tipo	Escritorio
Procesador	Intel (R) Core (TM) i7 2600
Potencia	3.40 GHz
Memoria	4 GB
Disco Duro	500 GB
Sistema Operativo	Windows 7 Ultimate

Fuente: Elaboración propia

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Tabla 37
Características del terminal3

Terminal3	
Tipo	Escritorio
Procesador	Intel (R) Pentium (R) Dual
Potencia	2.00 GHz
Memoria	4 GB
Disco Duro	160 GB
Sistema Operativo	Windows 7 Ultimate

Fuente: Elaboración propia

Diseño de la red

El diseño de nuestra red se aprecia en la siguiente figura.

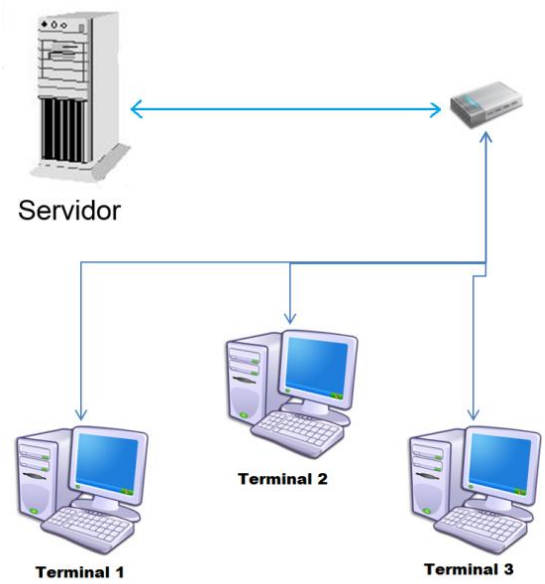

Figura 29 Esquema de red utilizado para la realización de las pruebas

3.3.3.1.2. Esquema de la base de datos

Para las pruebas se va a utilizar una base de datos con la misma estructura, cantidad de tablas, tipos de datos, llaves, tanto en MySQL como PostgreSQL, así también se ingresarán el mismo volumen de registros para ambos gestores.

Cabe aclarar que el escenario que se van a plantear está dirigido hacía un entorno asistencial de salud, donde se manipula grandes volúmenes de datos. El diseño de la base de datos se muestra en la siguiente figura.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

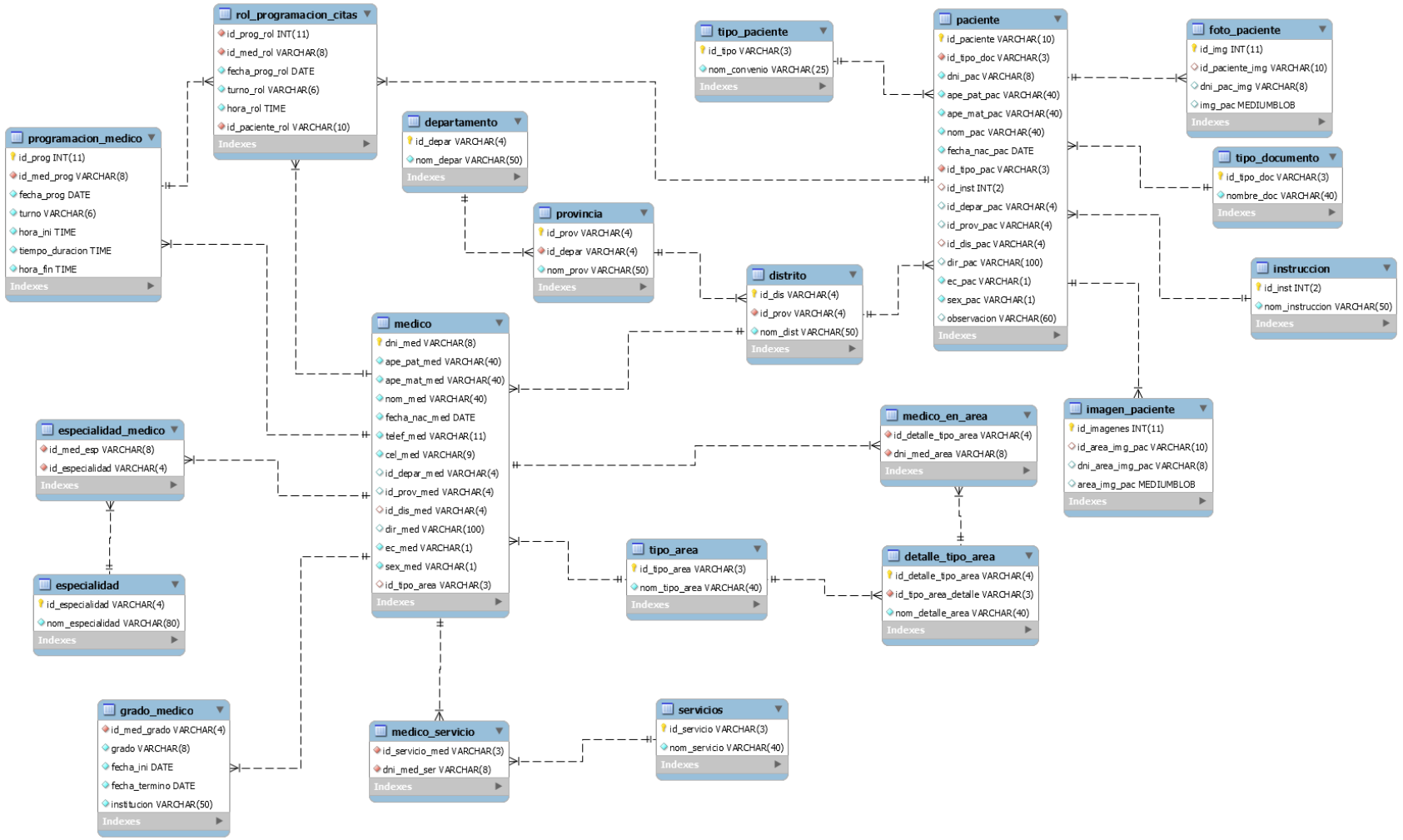

Figura 30 Modelos de base de datos

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Tabla 38
Volumen de registros en la Base de Datos

MySQL y PostgreSQL	
Tabla	Registros
Medico	20
Grado_medico	4
Especialidad	20
Especialidad_medico	30
Programación_medico	0
Rol_programacion_citas	0
Tipo_area	2
Detalle_tipo_area	45
Médicos_en_area	40
Medico_servicio	30
Paciente	Inicialmente 0
Tipo_paciente	6
Tipo_documento	5
Departamento	24
Provincia	194
Distrito	1825
Instrucción	7
servicios	12
Imagen_paciente	0
Foto_paciente	0

Fuente: Elaboración propia

3.3.3.2. Elaborar casos de pruebas por cada actividad CRUD

3.3.3.2.1. Determinación de pruebas

Para las pruebas de tiempo de velocidad y respuesta se usará los equipos terminales que generarán las transacciones mediante la aplicación Visual Basic NET. el mismo que tendrá opciones para realizar los procesos CRUD y que se ejecutaran independientemente uno del otro.

Así se generará carga de trabajo para cada proceso en ambos SGBD y que se describe de la siguiente manera:

a) Ingreso. - La aplicación Net se encargará del ingreso de registros, los mismos que estarán virtualizados y que se ingresarán automáticamente a los cuadros de textos de los formularios para posteriormente ser guardados a la base de datos.
b) Consulta. - Se tendrá un formulario encargado de realizar las consultas, éste seleccionará al azar un "id" aleatoriamente para su posterior consulta a la base de datos.
c) Edición o actualización. - Contaremos con formulario para realizar las actualizaciones, estas consultarán al azar un registro para realizarán una modificación y posteriormente ser actualizado.
d) Eliminación. - Además se tendrá un formulario para la eliminación de registros, este extraerá un registro al azar y lo eliminará.

Los tres terminales se ejecutarán paralelamente en el mismo tiempo realizando la carga de trabajo hacia el servidor con la base de datos.

Para el consumo de recursos utilizaremos el **Conjunto de recopiladores de datos** en el servidor por cada escenario, configurado de tal manera que recopile los siguientes datos:

a) Bytes privados. - Recopila el tamaño actual, en bytes, de la memoria que el proceso tiene asignada y que no puede compartirse con otros procesos.
b) Porcentaje de tiempo del procesador. - Recopila el porcentaje de tiempo transcurrido durante las transacciones que usaron el procesador para la ejecución de sus instrucciones.

Para el consumo de CPU tomaremos porcentajes de tiempo del procesador y para el consumo de la memoria RAM bytes privados usados por los procesos.

Las muestras a tomar en cuenta para cada proceso y SGBD, estarán enfocadas a promedios de resultados.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.3.3.2.2. Determinar tiempo de velocidad y respuesta

Se utilizará la aplicación en Visual Net, este se ejecutará en los terminales haciendo las peticiones al servidor de base de datos, la misma que permitirá ingresar, consulta, editar y eliminar, registrando el tiempo de velocidad y respuesta en milisegundos, el mismo que contiene código para contabilizar el tiempo de los procesos ejecutados y enviados a una hoja de cálculo para su posterior análisis.

La aplicación Net realizará las pruebas de carga y tendrá como objetivo medir el tiempo de velocidad y respuesta del servidor de base de datos los cuales se ejecutarán en tres escenarios de tiempos: 10000, 100000 y 1001000 registros.

Por cada escenario se realizará los procesos CRUD que inicialmente tendrán cero registros. Para la ejecución de las pruebas se utilizarán procedimientos almacenados.

Los Scripts de los procedimientos se encuentran en el **Anexo N° 1 y Anexo N° 2**

3.3.3.2.3. Determinación del consumo de recursos

En el servidor tendremos al "Conjuntos de recopiladores de datos" configurado para registrar el consumo de recursos en bytes de memoria RAM y el porcentaje de tiempo del procesador, en donde se encuentran los SGBD MySQL y PostgreSQL y que se ejecutara al recibir las instrucciones de carga ejecutados por los tres terminales.

Esta prueba tendrá como objetivo medir el consumo de los recursos en el servidor de los gestores de bases de datos durante la ejecución de las pruebas de carga, el valor de medida en esta prueba para CPU y memoria RAM será expresado en porcentajes y bytes respectivamente.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Administrador de Equipos. - En el encontramos herramientas de rendimiento, conjunto de recopiladores de rendimiento y otras herramientas.

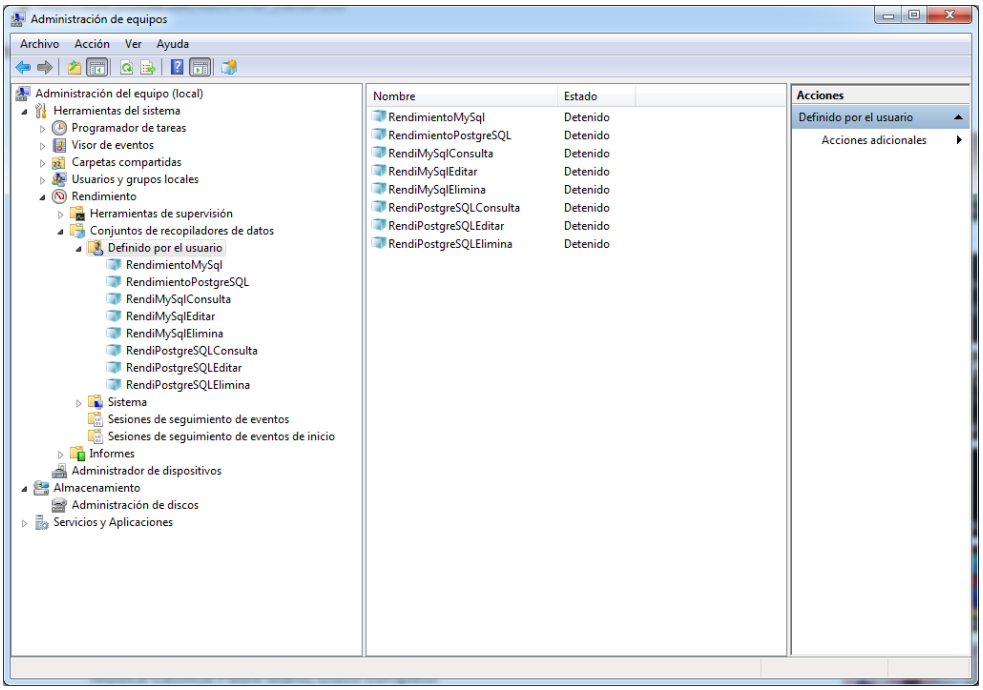

Figura 31 Configuración del conjunto de recopiladores de datos

Cuadro propiedades del Contador de MySQL (DataCollector01)

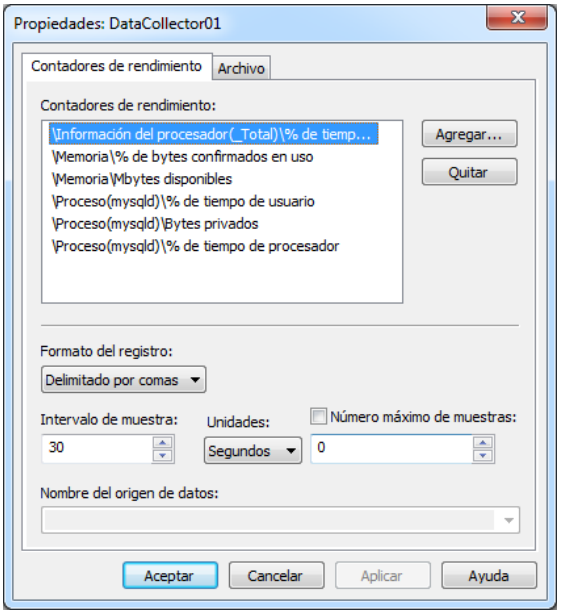

Figura 32 Propiedades del conjunto de recopiladores para MySQL

Cuadro propiedades del Contador de PostgreSQL (DataCollector01)

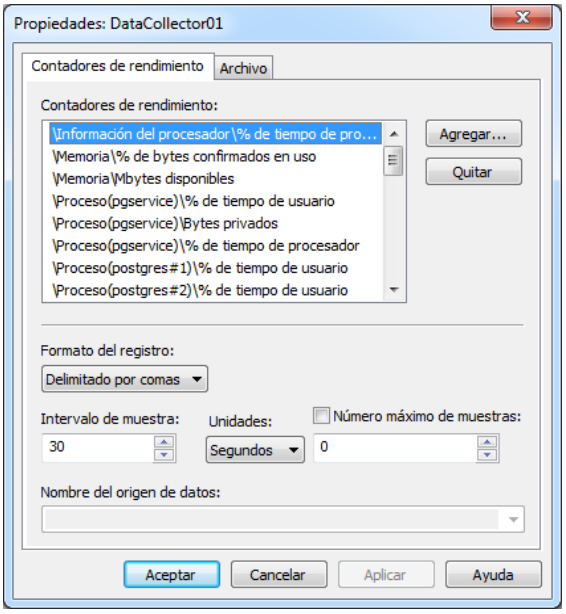

Figura 33 Propiedades del conjunto de recopiladores para PostgreSQL

Por defecto los nombres de los contadores son DataCollector01

Estos cuadros contienen las características de lo que se desea contabilizar, de la aplicación o aplicaciones que deseamos obtener información.

3.3.3.3. Ejecutar cada caso de prueba

3.3.3.3.1. Tiempo de velocidad y respuesta en MySQL

Las pruebas para medir el tiempo de velocidad y respuesta de los gestores de bases de datos MySQL y PostgreSQL, se realizaron con los 4 tipos de operaciones CRUD: inserción, consulta, modificación y eliminación. Los tres terminales ejecutaran uno de los procesos CRUD con una carga de registros. Por ejemplo, se ejecutará el ingreso de 10000 registros, a cada terminal se le asignara una lista de registros y se ejecutaran paralelamente para llenar la base de datos.

En el Anexo N° 3 se aprecia el Menú de la ventana principal MySQL

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Ingreso de Registros en MySQL

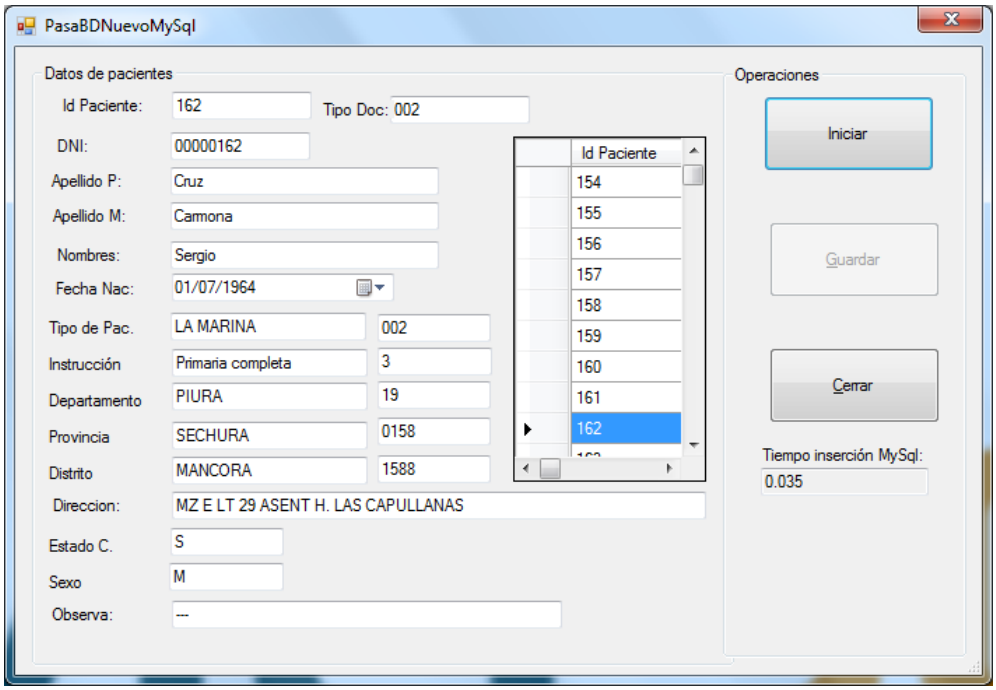

Figura 34 Formulario para ingresar registros en MySQL

El formulario toma un registro virtual de la PC, lo pasa a los cuadros de textos, posteriormente ejecuta la orden de guardar para almacenarlo en la base de datos MySQL, así también el formulario registrara el tiempo de velocidad y respuesta que transcurre en la ejecución de la instrucción para guardar y lo llevara a una hoja de cálculo al termino del proceso. Este formulario se estará ejecutando de forma similar en los tres equipos, realizando la carga de datos.

Consulta de Registros en MySQL

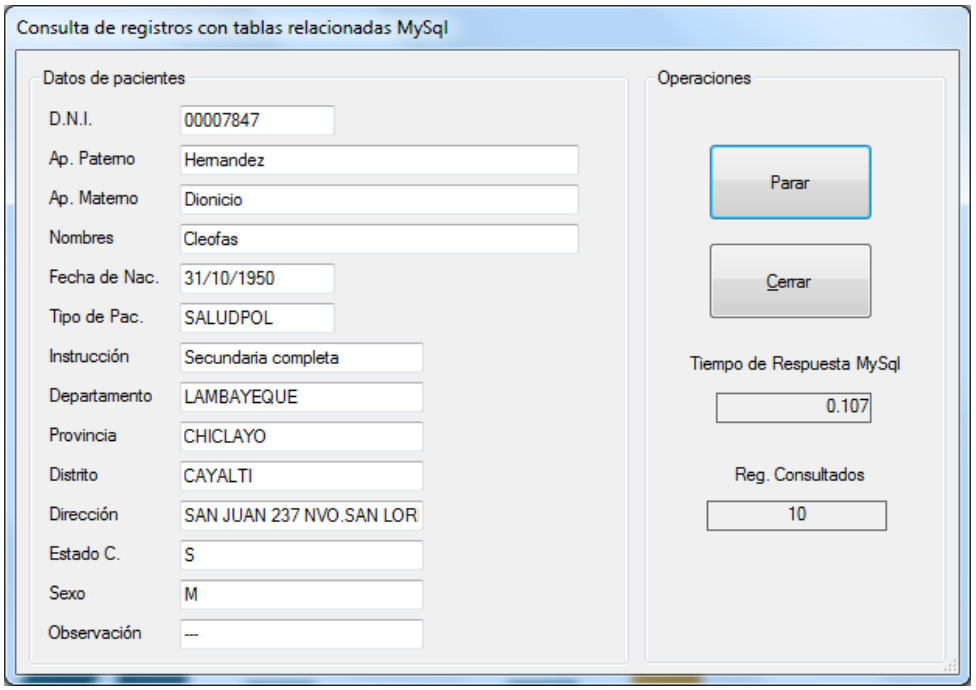

Figura 35 Formulario para consultar registros en MySQL

Este formulario seleccionará al azar un "id" de forma aleatoria, ejecutará la consulta y visualizará dicho registro correspondiente al "id" de búsqueda, así también registrará el tiempo de velocidad y respuesta de dicha consulta para llevarlo a una hoja de cálculo; este formulario también nos permite contabilizar cuantos registros a consultado.

Este formulario se estará ejecutando de forma similar en los tres equipos, realizando la carga de consulta de datos.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Actualizar Registros en MySQL

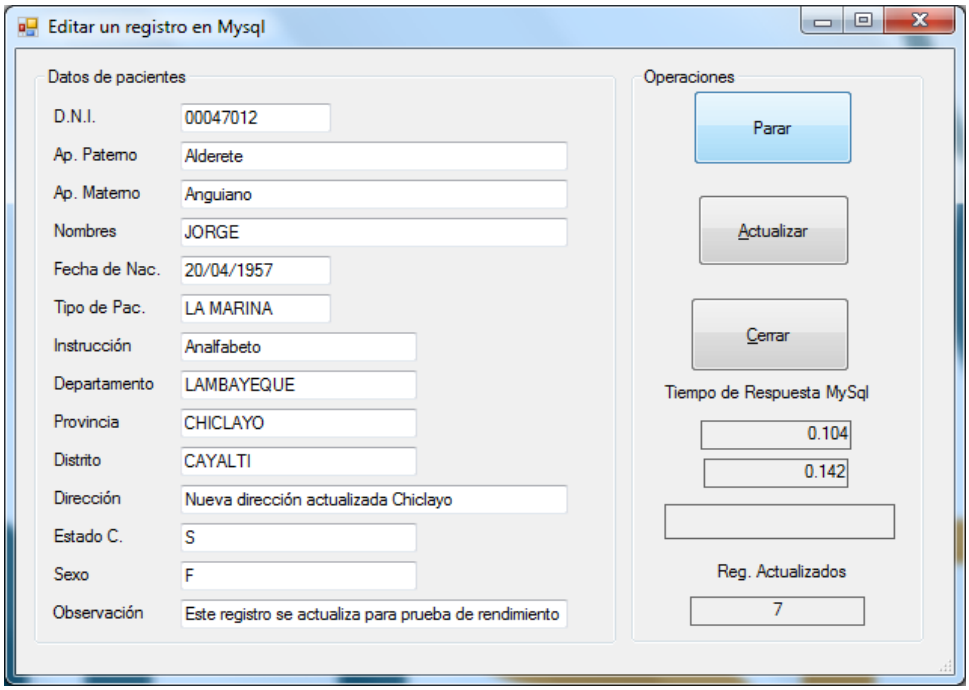

Figura 36 Formulario para actualizar registros en MySQL

Este formulario seleccionará al azar un "id" de forma aleatoria, ejecutara la consulta, visualiza dicho registro correspondiente al "id" de búsqueda, posteriormente realizará los cambios correspondientes y luego lo actualizara, registrara el tiempo de velocidad y respuesta de dicha actualización para llevarlo a una hoja de cálculo; este formulario también nos permite contabilizar cuantos registros han sido actualizados.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Eliminación de Registros en MySQL

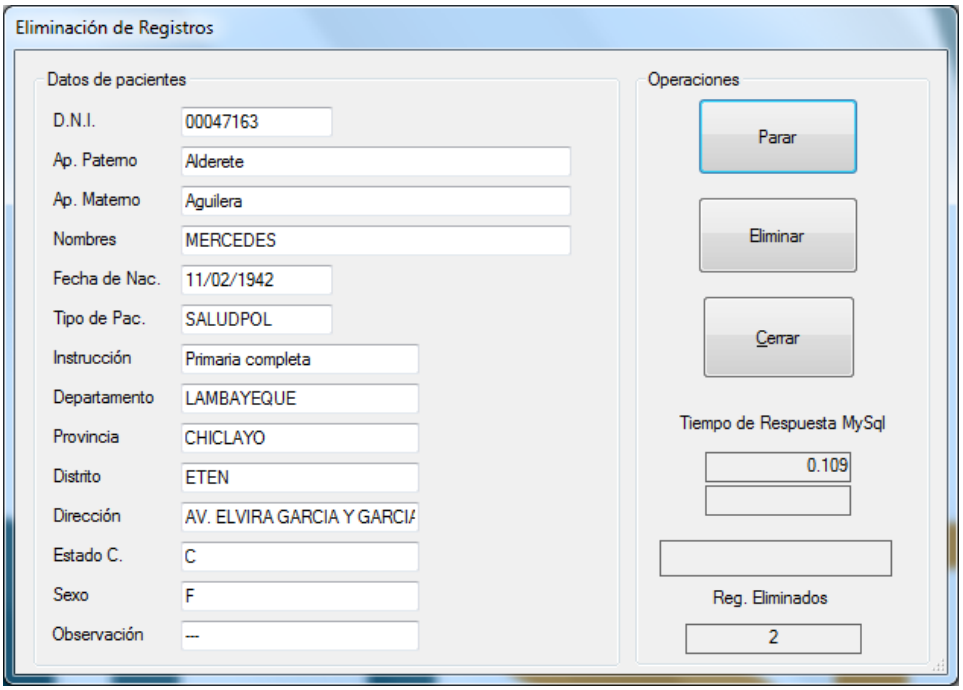

Figura 37 Formulario para eliminar registros en MySQL

Este formulario seleccionará al azar un "id" de forma aleatoria, ejecutara la consulta, visualiza dicho registro correspondiente al "id" de búsqueda, posteriormente ejecutara la orden de eliminar, así también registrara el tiempo de velocidad y respuesta de eliminar, para llevarlo a una hoja de cálculo; este formulario también nos permite contabilizar cuantos registros han sido eliminados.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.3.3.3.2. Tiempo de velocidad y respuesta en PostgreSQL

Ingreso de Registros en PostgreSQL

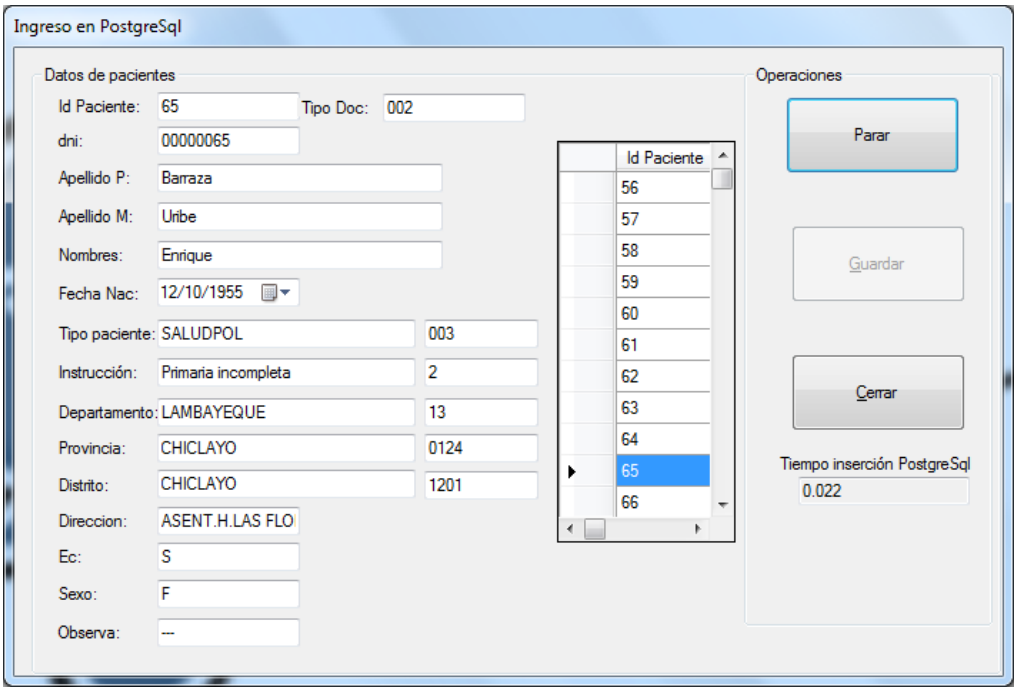

Figura 38 Formulario para ingresar registros en PostgreSQL

El formulario toma un registro virtual de la PC, lo pasa a los cuadros de textos, posteriormente ejecuta la orden de guardar para almacenarlo en la base de datos PostgreSQL, así también el formulario registrara el tiempo de velocidad y respuesta que transcurre en la ejecución de la instrucción para guardar y lo llevara a una hoja de cálculo al termino del proceso. Este formulario se estará ejecutando de forma similar en los tres equipos, realizando la carga de datos

En el Anexo N° 4 se aprecia el Menú de la ventana principal MySQL

Consulta de registros en PostgreSQL

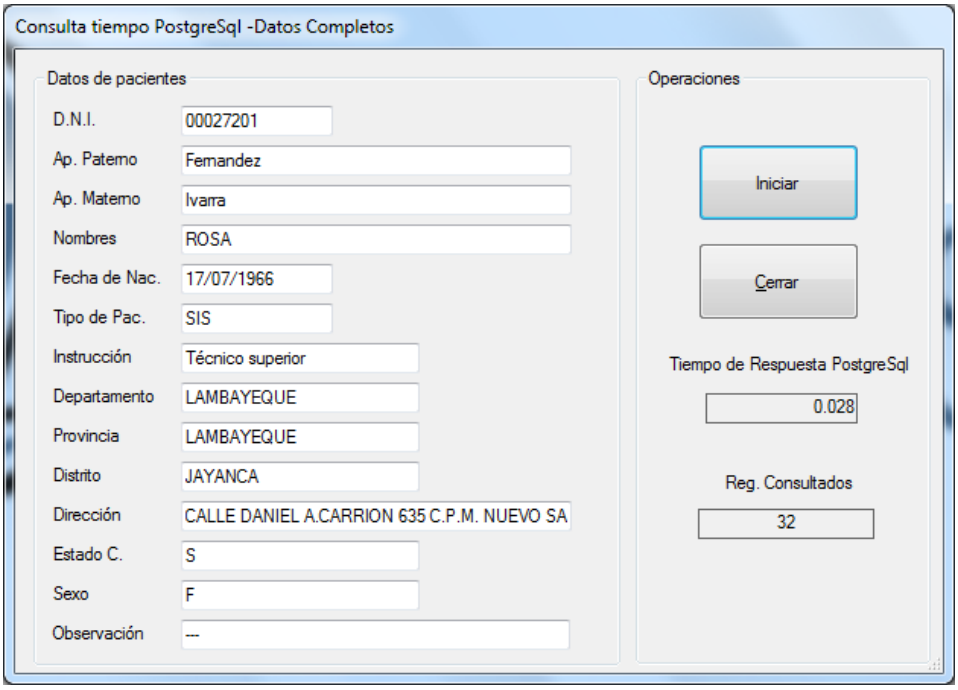

Figura 39 Formulario para consultar registros en PostgreSQL

Este formulario seleccionará al azar un "id" de forma aleatoria, ejecutará la consulta y visualizará dicho registro correspondiente al "id" de búsqueda, así también registrará el tiempo de velocidad y respuesta de dicha consulta para llevarlo a una hoja de cálculo; este formulario también nos permite contabilizar cuantos registros a consultado.

Este formulario se estará ejecutando de forma similar en los tres equipos, realizando la carga de consulta de datos.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Actualizar un registro en PostgreSQL

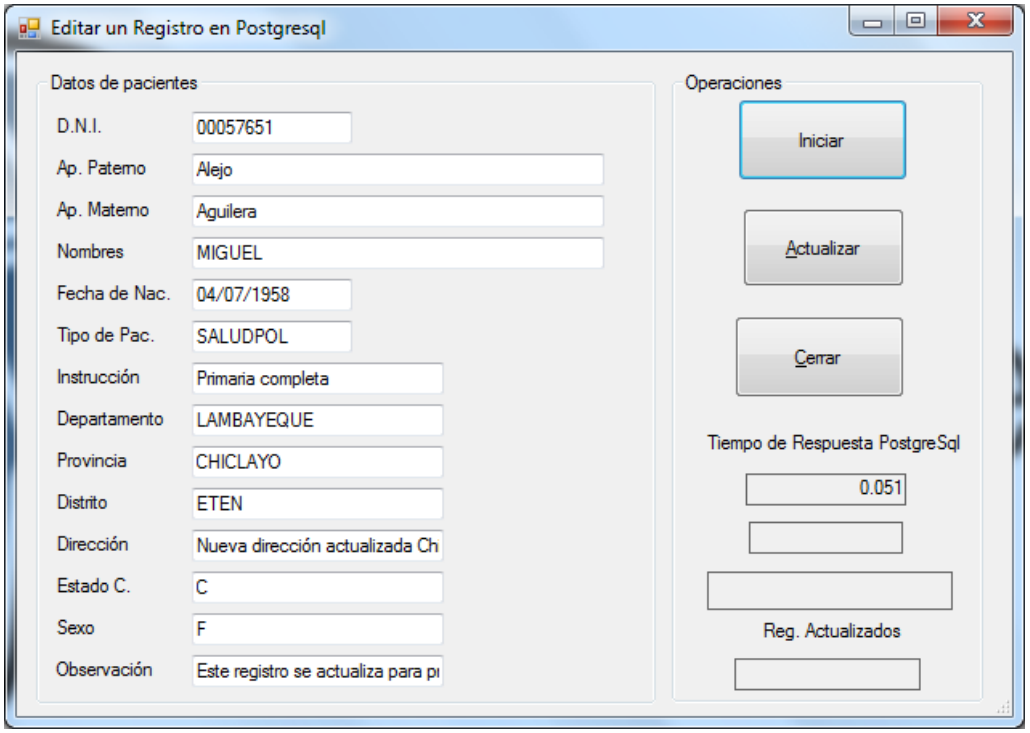

Figura 40 Formulario para actualizar registros en PostgreSQL

Este formulario seleccionará al azar un "id" de forma aleatoria, ejecutara la consulta, visualiza dicho registro correspondiente al "id" de búsqueda, posteriormente realizará los cambios correspondientes y luego lo actualizara, registrara el tiempo de velocidad y respuesta de dicha actualización para llevarlo a una hoja de cálculo; este formulario también nos permite contabilizar cuantos registros han sido actualizados.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Eliminación de registros en PostgreSQL

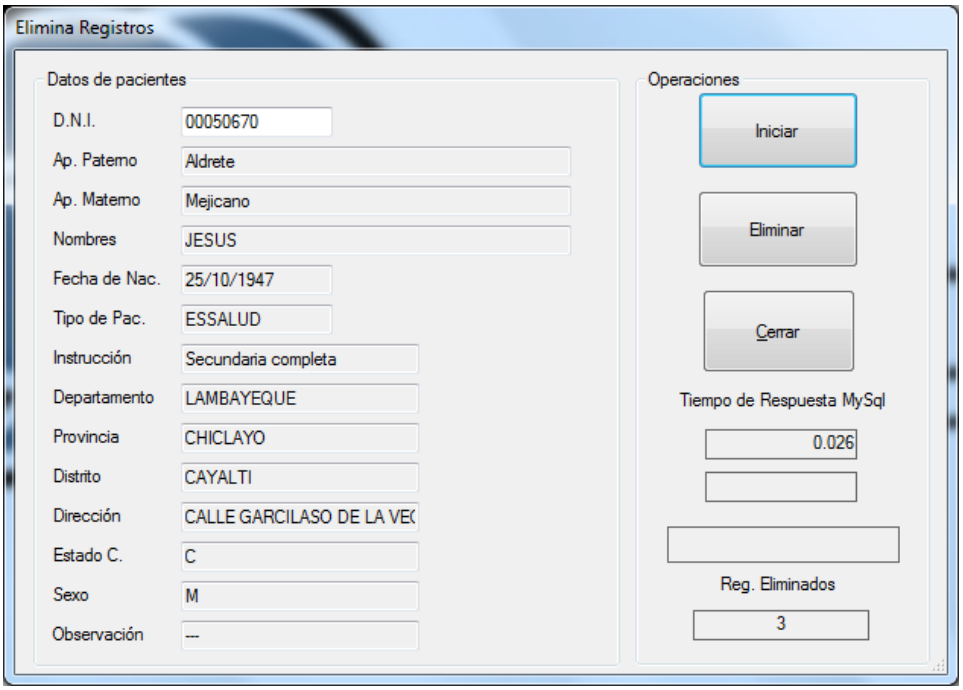

Figura 41 Formulario para eliminar registros en PostgreSQL

Este formulario seleccionará al azar un "id" de forma aleatoria, ejecutara la consulta, visualiza dicho registro correspondiente al "id" de búsqueda, posteriormente ejecutara la orden de eliminar, así también registrara el tiempo de velocidad y respuesta de eliminar, para llevarlo a una hoja de cálculo; este formulario también nos permite contabilizar cuantos registros han sido eliminados.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.3.3.3.3. Tiempo de velocidad y respuesta en MySQL con imágenes

Ingreso de imágenes

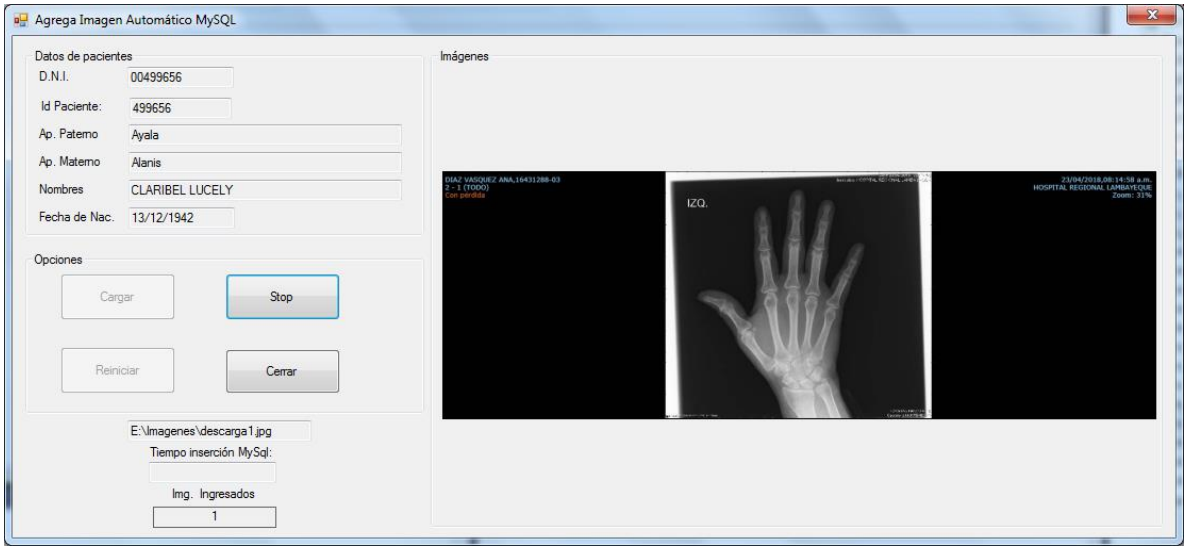

Figura 42 Formulario para ingreso de imágenes en MySQL

Consultas de imágenes

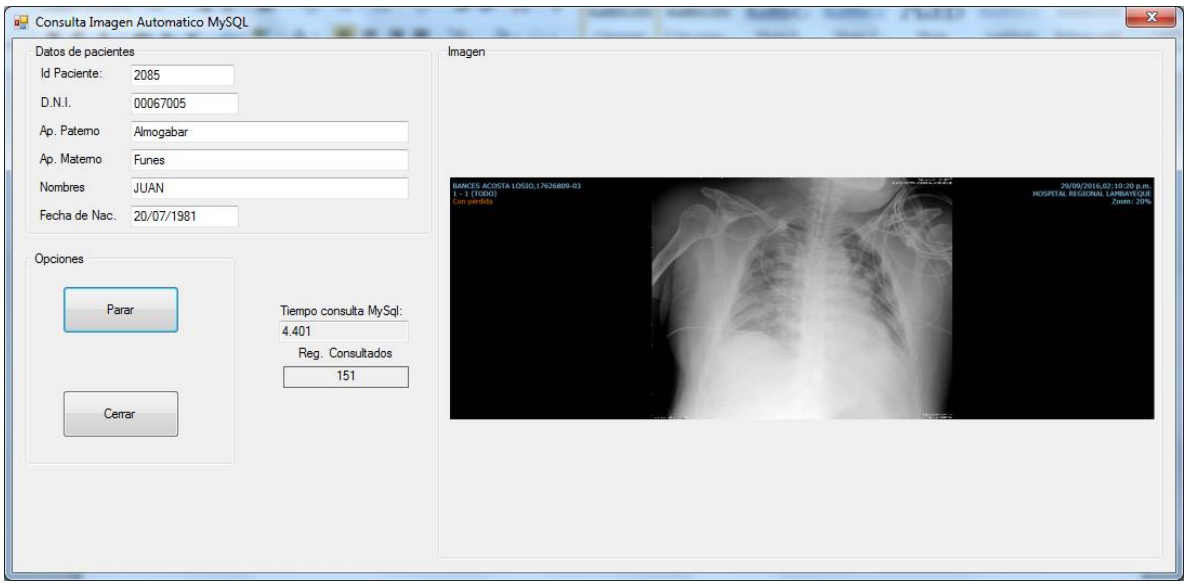

Figura 43 Formulario para consulta de Imágenes en MySQL

Con MySQL, los tres equipos realizan consultas de imágenes con tiempo de respuesta prolongados al inicio, pero luego se reduce el tiempo.

Eliminación de imágenes

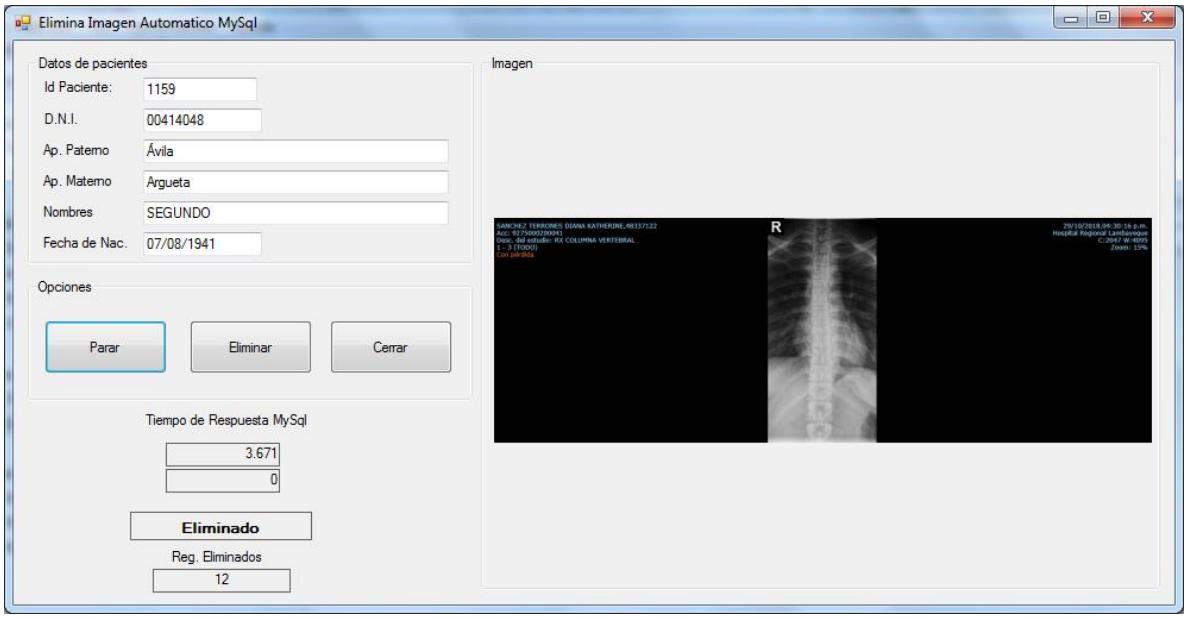

Figura 44 Formulario para eliminación de imágenes en MySQL

Con MySQL, los tres equipos realizan consultas y eliminación de imágenes, con tiempo de respuesta prolongada, pero no deja de realizar sus procesos.

3.3.3.3.4. Tiempo de velocidad y respuesta en PostgreSQL con imágenes

Ingreso de Imágenes

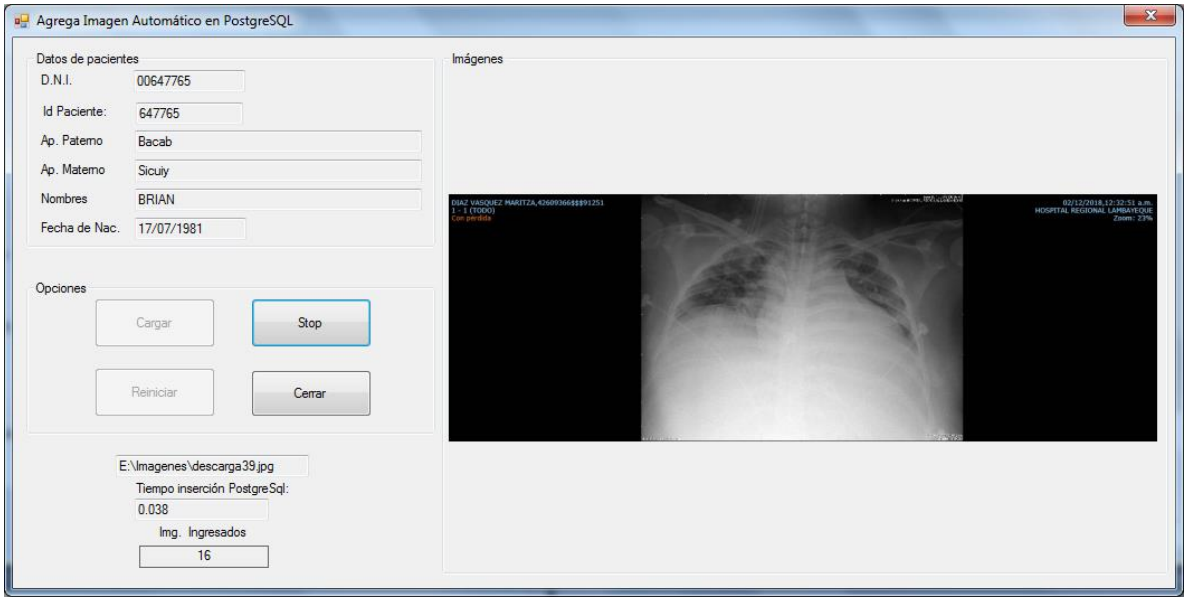

Figura 45 Formulario para ingreso de imágenes en PostgreSQL

Consulta de imágenes

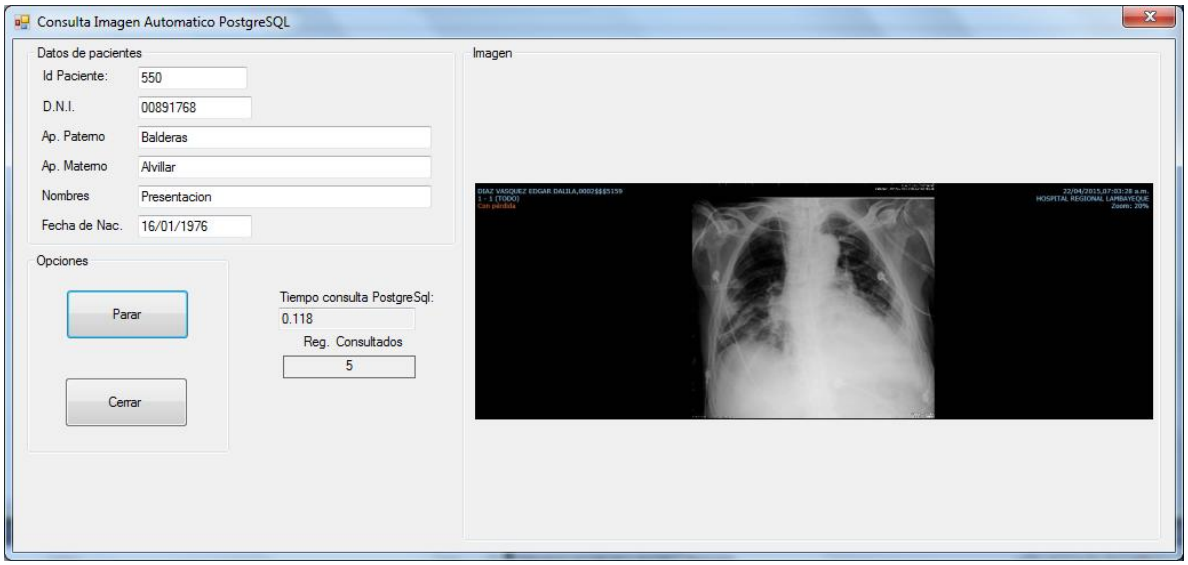

Figura 46 Formulario para consulta de imágenes en PostgreSQL

Eliminación de imágenes

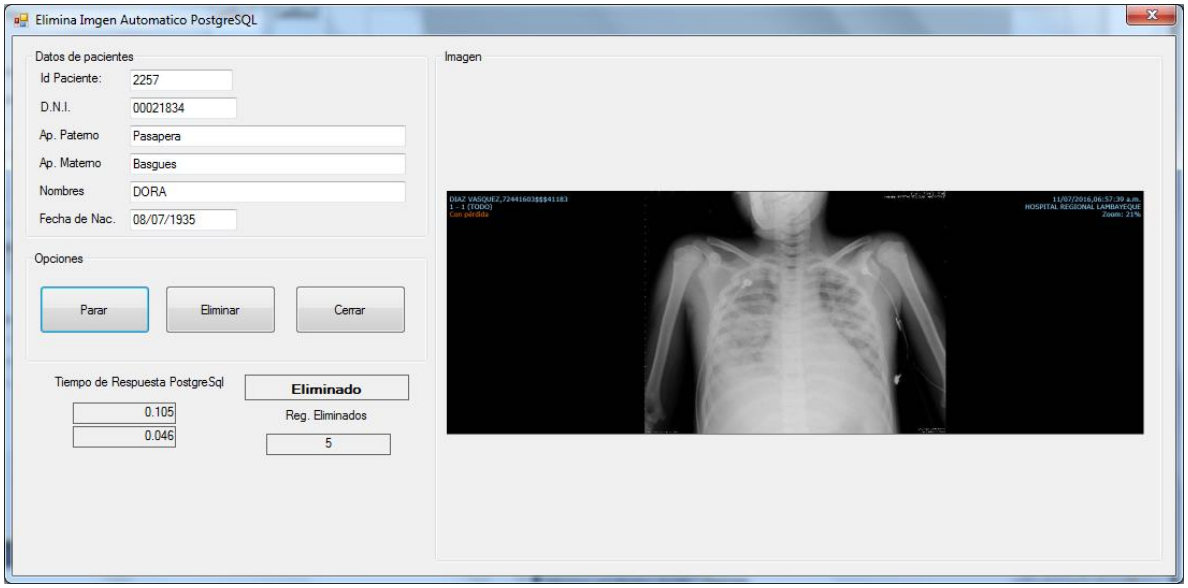

Figura 47 Formulario para eliminación de imágenes en PostgreSQL

Con PostgreSQL, los tres equipos realizan las consultas y eliminación con tiempos cortos de respuesta, pero en el lapso de un promedio de 9 consultas, deja de obtener respuesta con el servidor.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.3.3.3.5. Consumo de memoria y CPU

Para recolectar los datos de consumo de memoria y CPU, primero se ha establecido las opciones a recolectar, tanto para MySQL como para PotgreSQL. Esto se realizará en el Servidor.

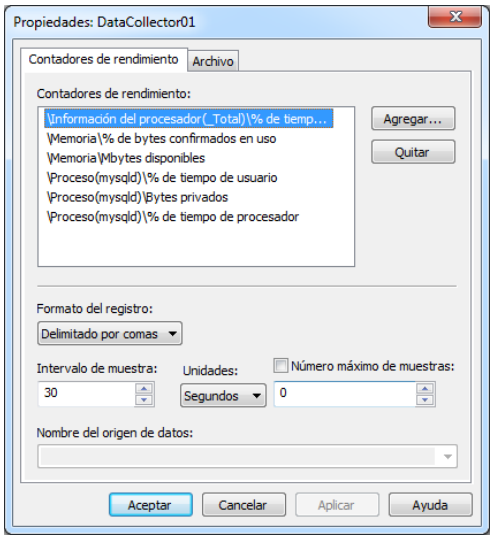

Figura 48 Propiedades del consumo de MySQL

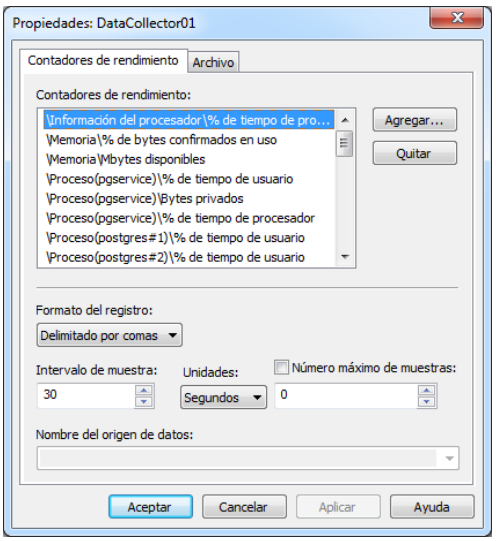

Figura 49 Propiedades del consumo de PostgreSQL

Posteriormente se debe iniciar la ejecución de los recolectores al momento de iniciar los procesos de los formularios en cada equipo. El recolector mostrara una pequeña flecha que indica que se ha iniciado el proceso de recolección.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

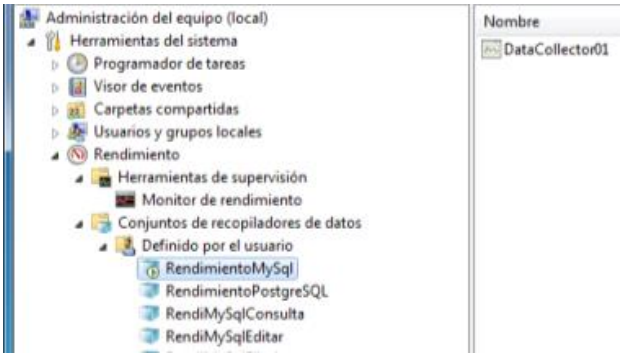

Figura 50 Ejecución del recopilador de datos

3.3.3.4. Elaborar informes de cada caso de pruebas

Pruebas de Tiempo de velocidad y respuesta

En el ingreso de registros se utilizó tres equipos con características diferentes, los cuales se le asignó una cantidad de datos virtuales los cuales estuvieron divididos en tres escenarios de tiempo con 10000, 100000 y 1001000 de registros.

Tabla 39
Informe de tiempo de velocidad y respuesta

Tiempo para el ingreso de registros		
Registros	MySQL en min	PostgreSQL en min
10000	16.0	16.0
100000	195.5	145.0
1001000	1816.5	1778.5

Fuente: Elaboración propia

En las consultas, modificación y eliminación de registros, se planifico realizar las pruebas sobre 1000 registros por cada equipo para cada escenario.

Tabla 40
Tiempo en minutos para ingresar registros

Tiempo para la consulta de registros		
Registros consultados	MySQL en min	PostgreSQL en min
1000 de 10000 reg.	4.5	4.5
1000 de 100000 reg.	10.5	5.5
1000 de 1001000 reg.	90.0	30.5

Fuente: Elaboración propia

Tabla 41
Tiempo en minutos para actualizar registros

Tiempo para la actualización de registros		
Registros actualizados	MySQL en min	PostgreSQL en min
1000 de 10000 reg.	14.0	14.0
1000 de 100000 reg.	19.5	17.0
1000 de 1001000 reg.	110.0	45.5

Fuente: Elaboración propia

Tabla 42
Tiempo en minutos para eliminar registros

Tiempo para la eliminación de registros		
Registros eliminados	MySQL en min	PostgreSQL en min
1000 de 10000 reg.	18.0	17.0
1000 de 100000 reg.	19.0	15.0
1000 de 1001000 reg.	123.0	62.0

Fuente: Elaboración propia

En los tres escenarios con 10000, 100000 y 1001000 registros para los procesos CRUD se observa un menor tiempo en la ejecución de las transacciones para PostgreSQL. De igual manera en la velocidad de repuesta de las operaciones se aprecia gran diferencia, observándose que MySQL tiene mayor tiempo de respuesta que PostgreSQL.

Prueba de consumo de CPU en porcentaje utilizado

En él % de tiempo del procesador en los tres escenarios, se aprecia el consumo del servicio de MySQL en relación al porcentaje de tiempo, es más elevado que el servicio de PostgreSQL, pero se observa además que PostgreSQL activa subprocesos, dividiendo las acciones del servicio haciéndolas más flexibles (postgres, postgres#1, postgres#2,…)

Prueba de consumo de Memoria en bytes utilizado

Los bytes privados de los procesos de cada servicio usados en los tres escenarios, se observa que PostgreSQL ocupa menos bytes en relación a MySQL.

3.3.4. Analizar resultados

3.3.4.1. Tiempo de velocidad y respuesta

Ingreso de registros en milisegundos

A continuación, se presenta un resumen sobre los tiempos de velocidad y respuesta en milisegundos, en la cual se expone los resultados finales de estas pruebas sobre el ingreso de 10000, 100000 y 1001000 registros, cada cual tomada en cuenta con los resultados de promedios en cada caso.

Tabla 43
Promedio de tiempo de velocidad y respuesta en tres escenarios en ingreso de registros

	MySQL	PostgreSQL
10000 Reg	0.0577	0.0093
100000 Reg	0.0580	0.0080
1001000 Reg	0.0620	0.0072
Promedio	0.0592	0.0082

Fuente: Elaboración propia

En análisis de la tabla demuestra que, en los tres escenarios, MySQL tiene un mayor tiempo de velocidad y respuesta en ingreso de registros.

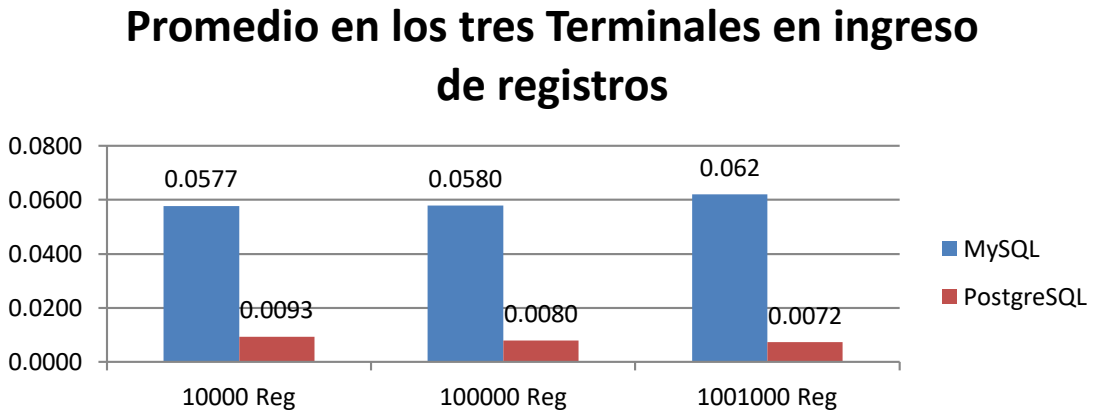

Figura 51 Promedio de tiempo de velocidad y respuesta en ingreso de registros

Observamos que en los tres escenarios con mayor tiempo de velocidad y respuesta en los procesos a MySQL.

Tabla 44
Suma total en tiempo de velocidad y respuesta en los tres escenarios en ingreso de registros

Total en los tres escenarios con los procesos CRUD en su conjunto		
	MySQL	PostgreSQL
Promedio total	0.0592	0.0082

Fuente: Elaboración propia

La tabla demuestra la suma total de los promedios en los tres escenarios que, MySQL tiene un mayor tiempo de velocidad y respuesta en ingreso de registros.

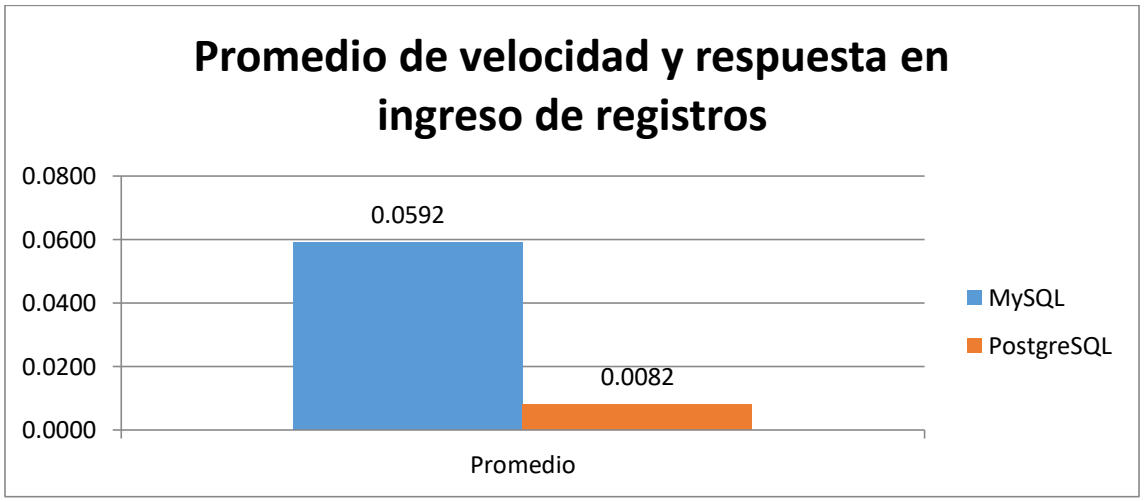

Figura 52 Promedio total en tiempo de velocidad y respuesta en consulta de registros

En la gráfica se observa que PostgreSQL tiene menor tiempo de velocidad y respuesta en ingreso de registros.

Consulta de registros en milisegundos

Resumen sobre los tiempos de velocidad y respuesta en milisegundos, en la cual se expone los resultados finales de estas pruebas sobre la consulta de 10000, 100000 y 1001000 registros, cada cual tomada en cuenta con los resultados de promedios en cada caso.

Tabla 45
Promedio de velocidad y respuesta en tres escenarios en consulta de registros

Terminal	MySQL	PostgreSQL
10000 Reg	0.0355	0.0136
100000 Reg	0.145	0.0223
1001000 Reg	1.3901	0.1035
Promedio	0.5235	0.0465

Fuente: Elaboración propia

El análisis de la tabla muestra el total de los promedios en los tres escenarios que, en donde MySQL tiene un mayor tiempo de velocidad y respuesta en consulta de registros.

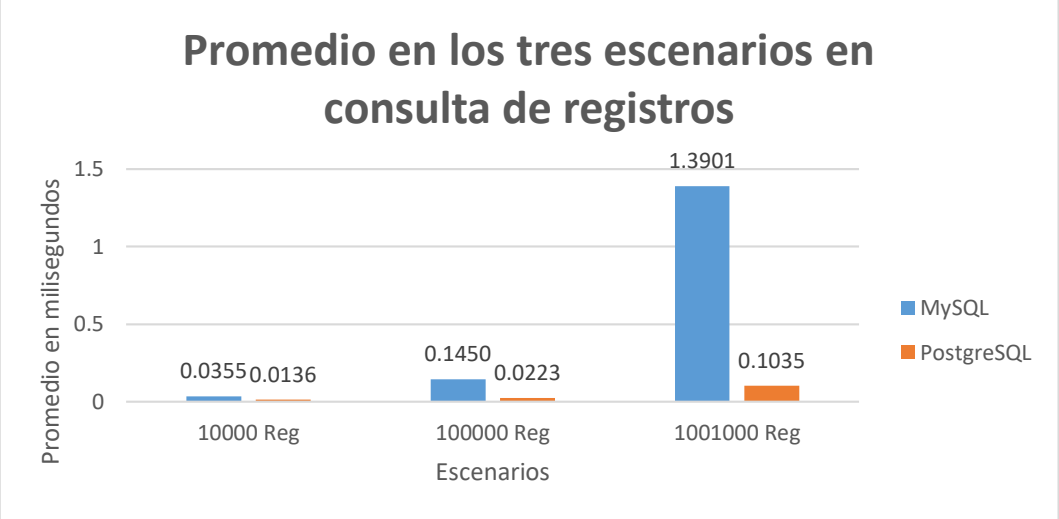

Figura 53 Promedio en tiempo de velocidad y respuesta en consulta de registros

Observamos que PostgreSQL tiene un menor tiempo de velocidad y respuesta en consulta de registros.

Tabla 46
Promedio en tiempo de velocidad y respuesta en tres escenarios en consulta de registros

Terminal	MySQL	PostgreSQL
Promedio total	0.5235	0.0465

Fuente: Elaboración propia

La suma total de los promedios en los tres escenarios muestra que MySQL tiene un mayor tiempo de velocidad y respuesta en consulta de registros.

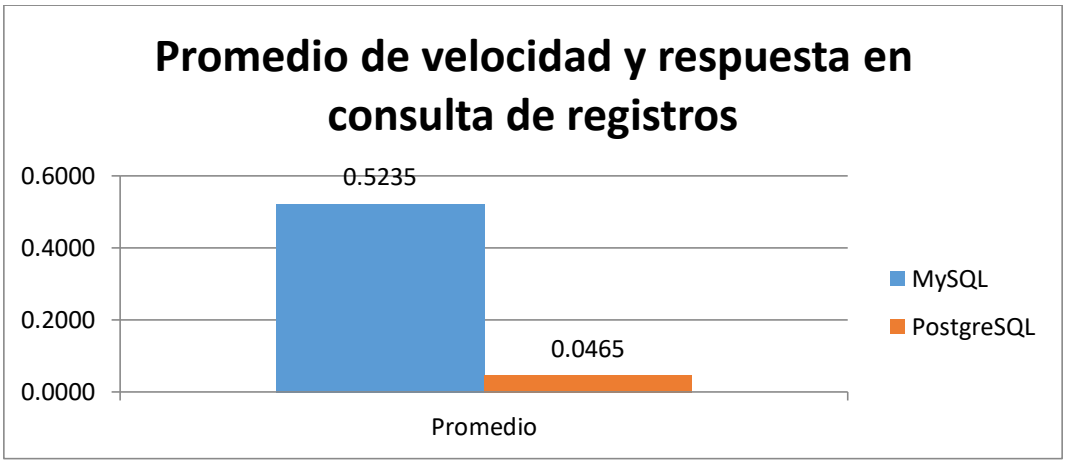

Figura 54 Promedio total de tiempo de velocidad y respuesta en consulta de registros

Observamos que PostgreSQL tiene un menor tiempo de velocidad y respuesta en consulta de registros.

Actualización de registros en milisegundos

Tabla 47
Promedio en tiempos de velocidad y respuesta en tres escenarios en actualización de registros

Terminal	MySQL	PostgreSQL
10000 Reg	0.0755	0.0115
100000 Reg	0.1384	0.0353
1001000 Reg	0.7777	0.1340
Promedio	0.3306	0.0603

Fuente: Elaboración propia

El total de los promedios en cada escenario muestra que MySQL tiene un mayor tiempo de velocidad y respuesta en actualización de registros.

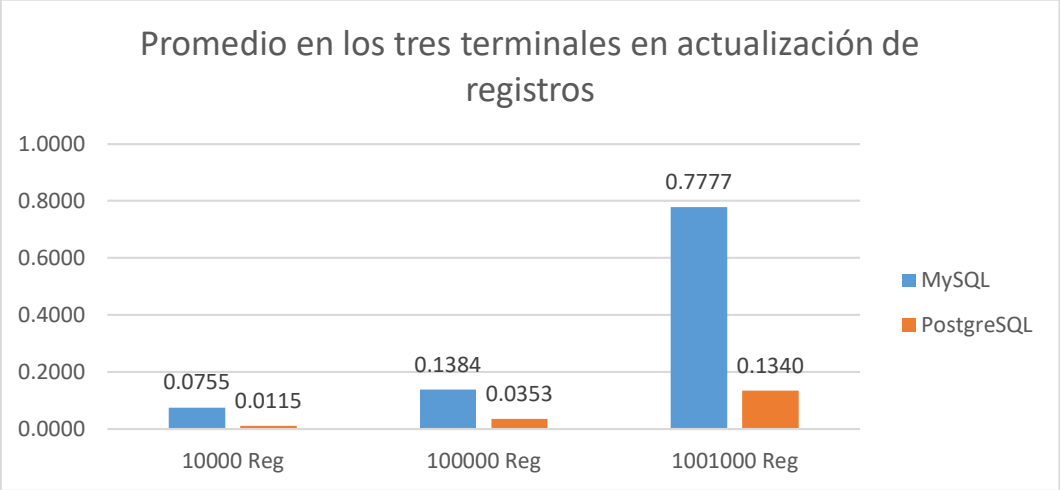

Figura 55 Promedio en tiempo de velocidad y respuesta en actualización de registros

Observamos que PostgreSQL tiene un menor tiempo de velocidad y respuesta en actualización de registros.

Tabla 48
Suma total en tiempo de velocidad y respuesta de los tres escenarios en actualización de registros

Terminal	MySQL	PostgreSQL
Promedio	0.3306	0.0603

Fuente: Elaboración propia

La suma total de los promedios en los tres escenarios muestra que MySQL tiene un mayor tiempo de velocidad y respuesta en actualización de registros.

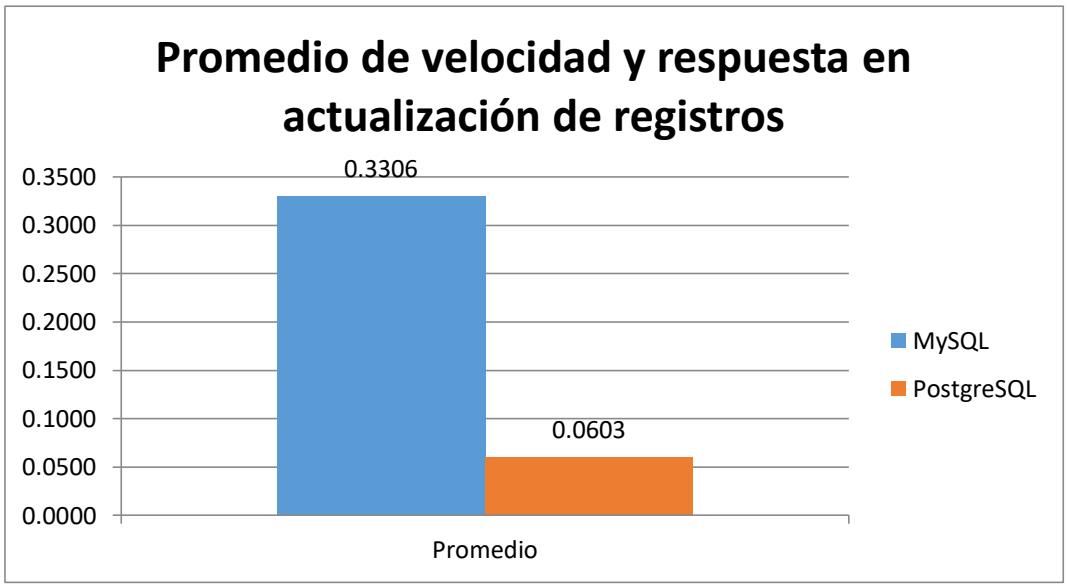

Figura 56 Promedio total de tiempo de velocidad y respuesta en actualización de registros

Observamos que PostgreSQL tiene un menor tiempo de velocidad y respuesta en actualización de registros.

Eliminación de registros en milisegundos

Tabla 49
Promedio en tiempo de velocidad y respuesta en tres escenarios en eliminación de registros

Terminal	MySQL	PostgreSQL
10000 Reg	0.0738	0.0109
100000 Reg	0.1206	0.0239
1001000 Reg	0.8222	0.1061
Promedio	0.3389	0.0470

Fuente: Elaboración propia

El total de los promedios en cada escenario muestra que MySQL tiene un mayor tiempo de velocidad y respuesta en eliminación de registros.

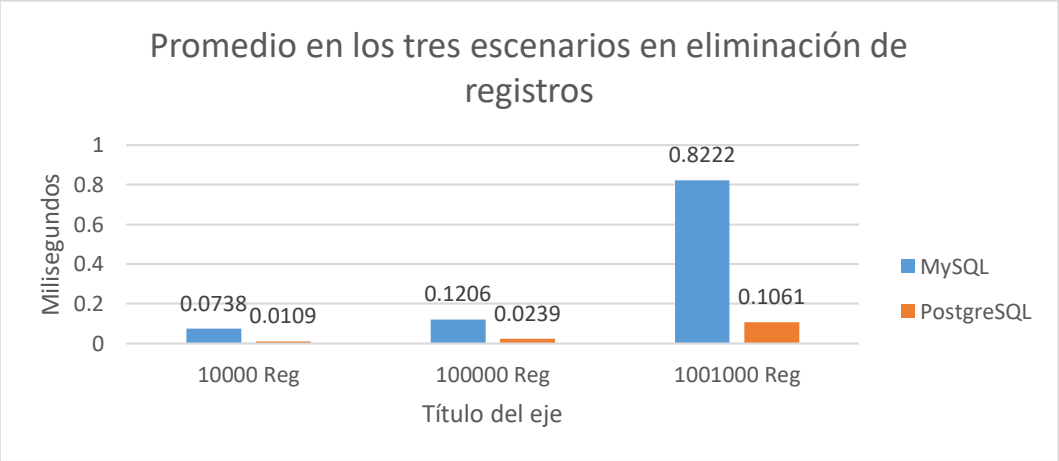

Figura 57 Promedio en tiempo de velocidad y respuesta en eliminación de registros

Observamos que PostgreSQL tiene un menor tiempo de velocidad y respuesta en eliminación de registros.

Tabla 50
Promedio total en tiempo de velocidad y respuesta en tres escenarios en eliminación de registros

Terminal	MySQL	PostgreSQL
Promedio	0.3389	0.0470

Fuente: Elaboración propia

La suma total de los promedios en los tres escenarios muestra que MySQL tiene un mayor tiempo de velocidad y respuesta en eliminación de registros.

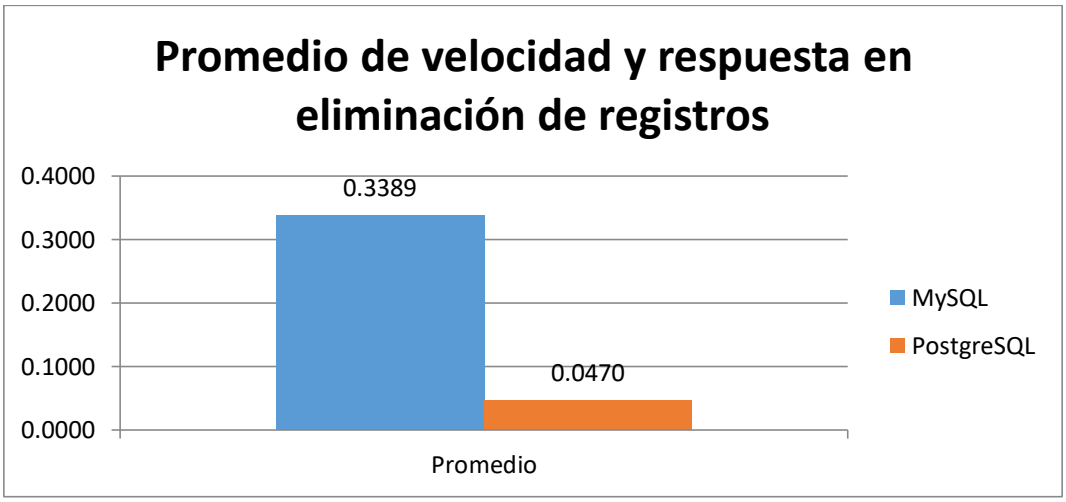

Figura 58 Promedio total de tiempo de velocidad y respuesta en eliminación de registros

Observamos que PostgreSQL tiene un menor tiempo de velocidad y respuesta en eliminación de registros.

3.3.4.2. Consumo de CPU en porcentaje

En la siguiente tabla se presenta la información resumen sobre el consumo de CPU en porcentajes, en la cual se expone los resultados finales de los procesos CRUD, tomando en cuenta los resultados de promedios en cada caso.

Tabla 51
Promedio total de consumo del CPU en procesos CRUD

	MySQL	PostgreSQL
Create	59.69	0.17
Read	142.98	48.24
Update	100.57	34.66
Delete	79.99	34.32
Promedio	95.81	29.35

Fuente: Elaboración propia

El total de los promedios en los procesos CRUD muestra que MySQL tiene mayor consumo de CPU en cada proceso.

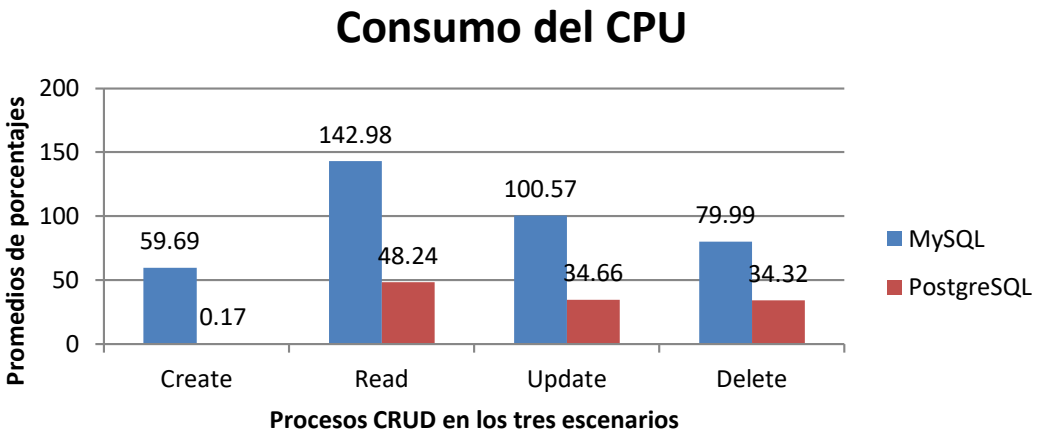

Figura 59 Promedio en consumo de CPU

En los tres escenarios realizados con procesos CRUD se observa que el consumo de porcentaje de CPU utilizado por MySQL es mayor al de PostgreSQL.

Tabla 52
Promedio total del consumo de CPU en procesos CRUD

Total en los tres escenarios con los procesos CRUD en relación a consumo del CPU		
	MySQL	PostgreSQL
Promedio	95.81	29.35

Fuente: Elaboración propia

Los promedios en los procesos CRUD demuestra que MySQL tiene mayor consumo de CPU en cada proceso.

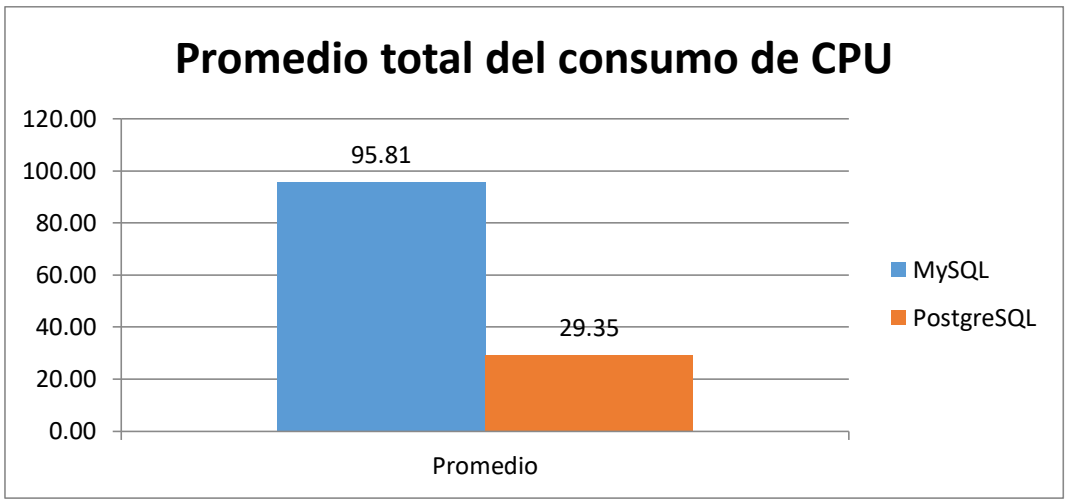

Figura 60 Promedio total en

Observamos notoriamente que MySQL consume gran porcentaje de CPU en relación a PostgreSQL.

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

3.3.4.3. Consumo de Memoria en bytes utilizada

En la siguiente tabla se presenta la información resumen sobre el consumo de bytes privados de memoria de los procesos CRUD, en la cual se expone los resultados, tomando en cuenta los resultados de promedios en cada caso.

Tabla 53
Promedio total del consumo de memoria en procesos CRUD

Escenarios: 10000, 100000 y 1001000	MySQL	PostgreSQL
Create	792845916.9	41852093.44
Read	811316324.6	54406397.57
Update	789657830.3	50806368.11
Delete	817030747.1	54767111.89
Promedio	802712704.7	50457992.75

Fuente: Elaboración propia

El total de los promedios en los procesos CRUD demuestra que MySQL tiene mayor consumo de memoria en cada proceso.

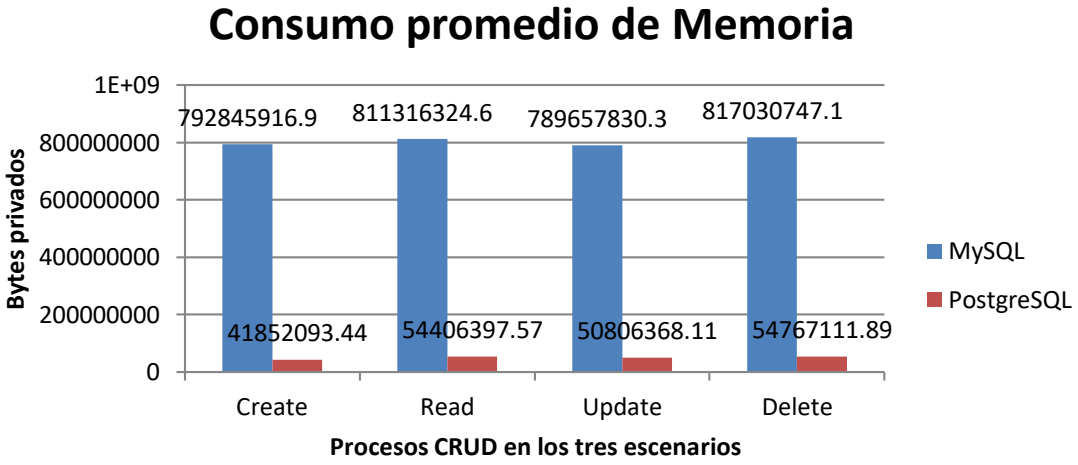

Figura 61 Promedio en consumo de memoria

En los procesos CRUD se observa que el consumo de memoria utilizado por MySQL es mayor al de PostgreSQL.

Tabla 54
Promedio de consumo de memoria en procesos CRUD

Escenarios: 10000, 100000 y 100100	MySQL	PostgreSQL
Promedio	802712704.7	50457992.75

Fuente: Elaboración propia

Los promedios en los procesos CRUD demuestra que MySQL tiene mayor consumo de memoria en cada proceso.

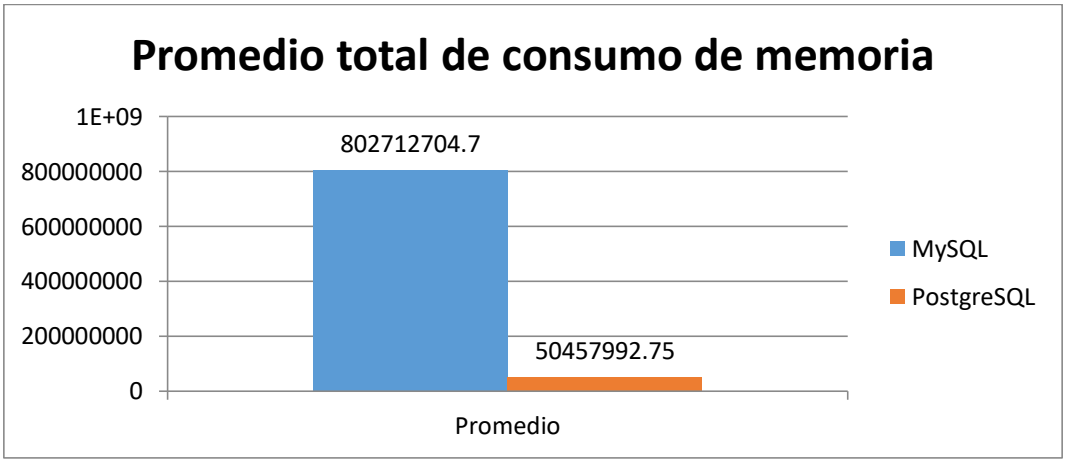

Figura 62 Promedio total de consumo de memoria

Con este grafico visualizamos la diferencia entre MySQL y PostgreSQL, en donde MySQL consume mayor cantidad de memoria bytes en los procesos CRUD.

3.3.4.4. Tiempo de velocidad y respuesta con imágenes

En el siguiente escenario se muestra información de los resultados con imágenes

Tabla 55
Promedio de velocidad y respuesta en ingreso de registros con imágenes

	MySQL	PostgreSQL
Terminal 1	0.108	0.058
Terminal 2	0.113	0.057
Terminal 3	0.128	0.091
Promedio	0.116	0.069

Fuente: Elaboración propia

El promedio en los tres terminales para el ingreso de registros con imágenes se observa que MySQL 0.116 mlsg y PostgreSQL 0.069 mlsg.

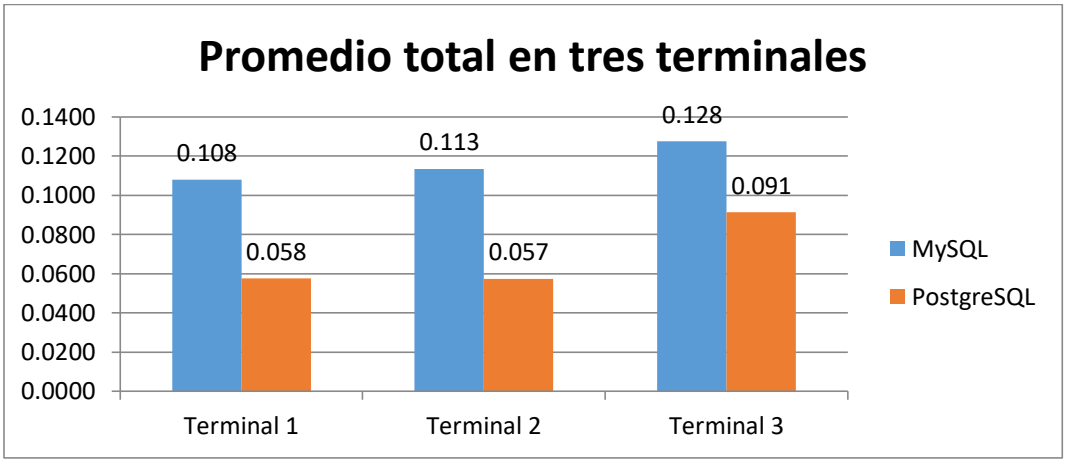

Figura 63 Promedio en terminales de velocidad y respuesta en ingreso de registros con imágenes

Tabla 56
Promedio de velocidad y respuesta en ingreso de registros con imágenes

	MySQL	PostgreSQL
Promedio	0.116	0.069

Fuente: Elaboración propia

Se observa que MySQL 0.116 mlsg y PostgreSQL 0.069 mlsg

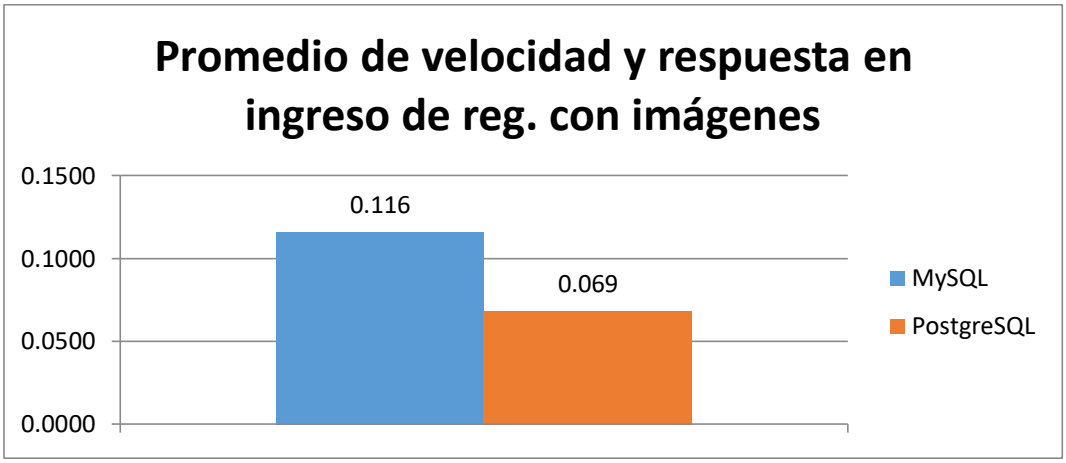

Figura 64 Promedio total de velocidad y respuesta en ingreso de registros con imágenes

IV. CONCLUSIONES Y RECOMENDACIONES

4.1. Conclusiones

En las pruebas desarrollados en este trabajo se puede evidenciar que en el tiempo de respuesta en las operaciones CRUD para PostgreSQL se observa con un mejor rendimiento, logrando un resultado de 0.0405 mlsg (40.5 µs) respecto a MySQL que obtuvo 0.3131 mlsg (313.1 µs), en el empleo de recursos, PostgreSQL consumió menor cantidad de CPU y memoria siendo una opción para la gestión de sistemas con tipos de datos básicos en el manejo de grandes cantidades de datos; sin embargo en los procesos de consultas y eliminaciones con imágenes MySQL ejecutó ilimitados procesos de consultas llegando a superar a PostgreSQL ya que este se limita en la ejecución de estos procesos de manera continua, siendo esto una ventaja para MySQL. Se concluye que MySQL es la mejor opción como SGBD en un entorno hospitalario en procesos CRUD por su variado manejo en diferentes escenarios con imágenes.

4.2. Recomendaciones

En la información de este documento se observa gran diferencia entre los dos gestores de bases de datos en el manejo de registros, aunque en la práctica, la realización de los procesos no es perceptible por el ojo humano en la ejecución de instrucciones ya que los dos gestores son aparentemente rápidos por igual; pero encontramos una deficiencia en la consulta y eliminación de imágenes para PostgreSQL lo cual constituye una desventaja, sin embargo se recomienda realizar otros estudios de estas y otras características de los sistemas gestores de bases de datos y proporcionar más información con el objetivo de brindar mayor apoyo a las investigaciones futuras en la mejora de productividad de los mismos.

REFERENCIAS

Almonacid, D. J. (2016). Comparación entre gestores de bases de datos relacionales. Obtenido de http://repositoriodigital.ucsc.cl/handle/25022009/1092

Andrade, M., & Parra, J. (214). Análisis de rendimiento entre Postgresql y Sql Server usando Hammerdb y Manage Engine aplicado al Sistema Académico de Conduespoch. Obtenido de http://dspace.espoch.edu.ec/handle/123456789/3740

Arias, Á. (2014). *Bases de Datos con MySQL* (2ª Edición ed.). Obtenido de https://cutt.ly/OjJXMO

Arias, Á. (2016). *Fundamentos de Programación y Bases de Datos.* (I. C. Academy, Ed.) Obtenido de https://cutt.ly/zjJVkW

Arias, M. (2017). *Aprende Programación Web con PHP y MySQL: 2ª Edición* (2ª Edición ed.). (I. C. Academy, Ed.) CreateSpace Independent Publishing Platform. Obtenido de https://cutt.ly/DjJVF3

Baca, G. (2016). *Introducción a la seguridad informática.* (G. E. Patria, Ed.) Obtenido de https://cutt.ly/1jBEMY

Benítez, M., & Arias, Á. (2015). *Curso de Introducción a la Administración de Bases de Datos.* (I. C. Academy, Ed.) Obtenido de https://cutt.ly/6jX97M

Bernabé, A. (2015). *Acceso a datos en aplicaciones web del entorno servidor.* (I. Editorial, Ed.) Obtenido de https://cutt.ly/AjLAbR

Carmona, G. (2013). *Aplicaciones informáticas de bases de datos relacionales.* (I. Editorial, Ed.) Obtenido de https://cutt.ly/hjKA6H

Cevallos, I. (2014). Análisis Comparativo de Respaldo y Recuperación de Base de Datos Licenciada (Oracle Utilizando RMAN) VS Open Source (MYSQL Utilizando MYSQL Administrator). Obtenido de http://repositorio.ug.edu.ec/handle/redug/6540

Cisneros, J. (1998). *Panorama sobre base de datos. Un enfoque práctico* (L. Medina Torres ed.). Obtenido de https://cutt.ly/OjXgpP

Cobo, A. (s.f.). *Diseño y programación de bases de datos.* (V. Libros, Ed.) Madrid, España: Visión Libros. Obtenido de https://cutt.ly/sjKMfa

Coronel, C., Morris, S., & Rob, P. (2011). *Bases de Datos, Diseño, Implementacion y Administracion.* (C. L. Editores, Ed.) Obtenido de https://cutt.ly/zjJM2D

DB-Engines. (Junio de 2019). Obtenido de DB-Engines: https://db-engines.com/en/ranking

Deléglise, D. (2013). *MySQL 5 (versiones 5.1 a 5.6): Guía de referencia del desarrollador.* (E. ENI, Ed.) Obtenido de https://cutt.ly/hjLRwW

Figueroa, A., Rollo, S., & Murthy, S. (Octubre de 2017). A Brief Comparison of Two Enterprise-Class RDBMSs. *ads*. Obtenido de https://cutt.ly/PjJi2q

Gabillaud, J. (2015). *SQL Server 2014: SQL, Transact SQL, diseño y creación de una base de datos (con ejercicios prácticos corregidos).* (E. ENI, Ed.) Obtenido de https://cutt.ly/cjZyUu

Gamboa, S. (2004). *Creatividad y entornos virtuales de aprendizaje* (Ilustrada ed.). (U. P. Nacional, Ed.) Obtenido de https://cutt.ly/cjZaPa

García, E. (2012). *Fundamentos de informática en entornos bioinformáticos.* (E. UOC, Ed.) Obtenido de https://cutt.ly/qjZjN6

Giacomo, M. D. (Mayo-Junio de 2005). MySQL: lessons learned on a digital library. *in IEEE Software*. Obtenido de https://ieeexplore.ieee.org/document/1438321

Henriquez, N., Iglesias, A., Amaris, L., & Ropain, Y. (2013). Postgresql una alternativa efectiva en las empresas. *REVISTA I+D EN TIC, IV*(1). Obtenido de https://cutt.ly/ejCRKg

Heredero, C. (2004). *Informática y comunicaciones en la empresa.* (E. Editorial, Ed.) Obtenido de https://cutt.ly/wjZZ5f

Hospital de Emergencia Villa el Salvador. (s.f.). Obtenido de Hospital de Emergencia Villa el Salvador: https://www.heves.gob.pe/portal/como-me-atiendo/

Huang, J., Mozafari, B., Schoenebeck, G., & T., W. (Febrero de 2016). Identifying the Major Sources of Variance in Transaction Latencies: Towards More Predictable Databases. *Databases*. Obtenido de https://arxiv.org/abs/1602.01871

Jiménez, M. (2015). *Bases de datos relacionales y modelado de datos.* (I. Editorial, Ed.) Obtenido de https://cutt.ly/VjKbDZ

López, P. (2016). Comparación del desempeño de los Sistemas Gestores de Bases de Datos MySQL y PostgreSQL. Obtenido de http://ri.uaemex.mx/handle/20.500.11799/62548

Mahajan, D., Kyung, J., Sacks, J., Ardalan, A., Kumar, A., & H., E. (2018). In-RDBMS Hardware Acceleration of Advanced Analytics. *DIGITAL LIBRARY, 11*, 1317-1331.

Microsoft. (s.f.). *Microsoft documentación*. Obtenido de Creación de conjuntos de recopiladores de datos: https://bit.ly/3ckMNvZ

Microsoft. (s.f.). *Microsoft Documentación*. Obtenido de Uso de Monitor de rendimiento: https://bit.ly/3aflHol

Nevado, V. (2010). *Introduccion a las Bases de Datos relacionales.* (E. V. Libros, Ed.) Obtenido de https://cutt.ly/WjKgjZ

Piñal M., J. F., Álvarez G., R., & Sánchez G., A. M. (2009). IMPLEMENTACIÓN EN HARDWARE DEL ESTÁNDAR DE ENCRIPTACIÓN AVANZADO (AES), EN UNA PLATAFORMA FPGA, EMPLEANDO EL MICROCONTROLADOR PICOBLAZETM. 1-8. Obtenido de http://www.redalyc.org/articulo.oa?id=73012215005

Poljak, R., Poščić, P., & Jakšić, D. (Mayo de 2017). Comparative analysis of the selected relational database management systems. *International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 1496-1500. doi:10.23919/MIPRO.2017.7973658

Portilla, J., & Bernal, M. (05 de 2018). Comparación del rendimiento de los comandos Insert, Select y Delet en los sistemas gestores de bases de datos Oracle y MYSQL.

Revistas Académicas INVENTUM, 12(23), 10-21. Obtenido de https://cutt.ly/ZjCHMi

Ramos, A., & Ramos, M. (2007). *Operaciones con bases de datos ofimáticas y corporativas.* (E. Paraninfo, Ed.) Obtenido de https://cutt.ly/ajJqgn

Rubio, M., Gómez, S., Letón, E., Rodrigo, S., & Chaos, D. (2017). *INTRODUCCIÓN A LA INFORMÁTICA BÁSICA.* (E. UNED, Ed.) Obtenido de https://cutt.ly/ejCqRz

Secco, R., Da Silva, F., Maracci, F., & Pazoti, M. (Noviembre de 2016). Análise Comparativa Entre O Banco De Dados Cassandra (Modelo Nosql) E O Postgresql (Modelo Relacional) Em Duas Diferentes Organizações Empresariais. *Colloquium Exactarum, 8*(2), 39-56. Obtenido de https://cutt.ly/pzoZpg

Silberschatz, A., Korth, H., & Sudarshan, S. (2006). *Fundamentos de bases de datos* (Quinta edición ed.). (A. G. F. SÁENZ PÉREZ, Trad.) Madrid, España. Obtenido de https://cutt.ly/fjJwei

Talledo, J. (2015). *Acceso a datos en aplicaciones web del entorno servidor.* (S. Ediciones Paraninfo, Ed.) Obtenido de https://cutt.ly/cKeWaH

Talledo, J. (2016). *Administración y monitorización de los SGBD.* (S. Ediciones Paraninfo, Ed.) Obtenido de https://cutt.ly/rKwMO2

Tejada, E. C. (2017). *Utilización de las bases de datos relacionales en el sistema de gestión y almacenamiento de datos.* (I. Editorial, Ed.) IC Editorial. Obtenido de https://cutt.ly/5KY5sC

Thibaud, C. (2006). *Recursos Informáticos MYSQL 5.* (E. ENI, Ed.) Obtenido de https://cutt.ly/yzalJ5

TIOBE. (2019). Obtenido de https://www.tiobe.com/tiobe-index/

Tongkaw, S., & Tongkaw, A. (2016). A Comparison of Database Performance of MariaDB and MySQL with OLTP Workload. *Conference on Open Systems (ICOS)*, 117-119. doi:10.1109/ICOS.2016.7881999

Truica, C., Radulescu, F., Alexandru, B., & Ion, B. (Agosto de 2015). Performance Evaluation for CRUD Operations in Asynchronously Replicated Document

Oriented Database. *International Conference on Control Systems and Computer Science, Bucharest*, 191-196. doi:10.1109/CSCS.2015.32

Zea, M., Molina, J., & Redrován, F. (2017). *ADMINISTRACIÓN DE BASES DE DATOS CON POSTGRESQL.* (3Ciencias, Ed.) Obtenido de https://cutt.ly/EjCdPf

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

ANEXOS

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Anexo 1: Scripts de procedimientos en MySQL

```
delimiter $$
create procedure rendimiento.muestra_departamento(
in widdepar varchar(4)
)
begin
select id_depar,nom_depar from departamento where id_depar like widdepar;
end
$$


delimiter $$
create procedure rendimiento.muestra_provincia(
in widprovin varchar(4)
)
begin
select id_prov,nom_prov from provincia where id_prov like widprovin;
end
$$

delimiter $$
create procedure rendimiento.muestra_distrito(
in widdistri varchar(4)
)
begin
select id_dis,nom_dist from distrito where id_dis like widdistri;
end
$$
```


Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
delimiter $$
create procedure rendimiento.muestra_convenio(
in widtipo varchar(3)
)
begin
select id_tipo,nom_convenio from tipo_paciente where id_tipo like widtipo;
end
$$

delimiter $$
create procedure rendimiento.muestra_instruccion(
in widinst varchar(3)
)
begin
select id_inst,nom_instruccion from instruccion where id_inst like widinst;
end
$$

delimiter $$
create procedure rendimiento.muestra_paciente_dos(
in wdni varchar(8)
)
begin
select P.dni_pac, P.ape_pat_pac, P.ape_mat_pac, P.nom_pac, P.fecha_nac_pac,
T.nom_convenio, I.nom_instruccion, DE.nom_depar, PR.nom_prov, DI.nom_dist,
P.dir_pac, P.ec_pac, P.sex_pac, P.observacion
from paciente P join departamento DE on P.id_depar_pac=DE.id_depar join provincia PR
on P.id_prov_pac=PR.id_prov join distrito DI on P.id_dis_pac=DI.id_dis join
tipo_paciente T on P.id_tipo_pac=T.id_tipo join instruccion I on P.id_inst=I.id_inst
where P.dni_pac like wdni;
end
$$
delimiter $$
```


Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
CREATE PROCEDURE rendimiento.actualiza_paciente(
      IN dni_pac_ CHAR(8),
      IN ape_pat_pac_ VARCHAR(40),
      IN ape_mat_pac_ VARCHAR(40),
      IN nom_pac_ VARCHAR(40),
      IN dir_pac_ VARCHAR(100),
      IN ec_pac_ VARCHAR(1),
      IN sex_pac_ VARCHAR(1),
      IN observacion_ VARCHAR(60)
   )
   NOT DETERMINISTIC
   CONTAINS SQL
   SQL SECURITY DEFINER
   COMMENT ''
BEGIN
UPDATE paciente SET dni_pac=dni_pac_, ape_pat_pac=ape_pat_pac_,
ape_mat_pac=ape_mat_pac_, nom_pac=nom_pac_, dir_pac=dir_pac_, ec_pac=ec_pac_,
sex_pac=sex_pac_, observacion=observacion_
WHERE dni_pac=dni_pac_;
END
$$

delimiter $$
CREATE PROCEDURE rendimiento.elimina_paciente(
      IN dni_pac_ CHAR(8)
)
BEGIN
Delete From paciente Where dni_pac=dni_pac_;
END
$$
```


Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Anexo 2: Scripts de Procedimientos en PostgreSQL

```
--Creamos el contenedor
CREATE TYPE datos_departamento AS
(
   id_depar VARCHAR(4),
   nom_depar VARCHAR(50)
);

--Ahora creamos la función
CREATE OR REPLACE FUNCTION muestra_departamento(id_depar_ VARCHAR)
RETURNS setof datos_departamento AS
$BODY$
DECLARE
   r record;
   sql text;
BEGIN
sql := 'SELECT id_depar,nom_depar FROM departamento WHERE 0=0 '; -- Para
-- simplificar ejecutamos un texto.
-- Lógica de filtrado
IF id_depar_ IS NOT NULL THEN sql := sql || ' AND
id_depar='||quote_literal(id_depar_); END IF;
for r in execute sql loop
   return next r;
end loop;
end;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
```


Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
--Creamos el contenedor provincia
CREATE TYPE datos_provincia AS
(
   id_prov VARCHAR(4),
   id_depar VARCHAR(4),
   nom_prov VARCHAR(50)
);

--Ahora creamos la función
CREATE OR REPLACE FUNCTION muestra_provincia(id_prov_ VARCHAR)
RETURNS setof datos_provincia AS
$BODY$
DECLARE
   rprov record;
   sql text;
BEGIN
sql := 'SELECT id_prov,id_depar,nom_prov FROM provincia WHERE 0=0 '; -- Para
IF id_prov_ IS NOT NULL THEN sql := sql || ' AND id_prov='||quote_literal(id_prov_);
END IF;
for rprov in execute sql loop
   return next rprov;
end loop;
end;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
```


Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
---Contenedor distrito
CREATE TYPE datos_distrito AS
(
   id_dis VARCHAR(4),
   id_prov VARCHAR(4),
   nom_dist VARCHAR(50)
);

--Función muestra distrito
CREATE OR REPLACE FUNCTION muestra_distrito(id_dis_ VARCHAR) RETURNS
setof datos_distrito AS
$BODY$
DECLARE
   rdis record;
   sql text;
BEGIN
sql := 'SELECT id_dis,id_prov,nom_dist FROM distrito WHERE 0=0 '; -- Para
IF id_dis_ IS NOT NULL THEN sql := sql || ' AND id_dis='||quote_literal(id_dis_); END
IF;
for rdis in execute sql loop
   return next rdis;
end loop;
end;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
```


Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
---Contenedor
CREATE TYPE datos_convenio AS
(
   id_tipo VARCHAR(3),
   nom_convenio VARCHAR(25)
);

--Función muestra convenio
CREATE OR REPLACE FUNCTION muestra_convenio(id_tipo_ VARCHAR)
RETURNS setof datos_convenio AS
$BODY$
DECLARE
   rtipo record;
   sql text;
BEGIN
sql := 'SELECT id_tipo,nom_convenio FROM tipo_paciente WHERE 0=0 '; -- Para
IF id_tipo_ IS NOT NULL THEN sql := sql || ' AND id_tipo='||quote_literal(id_tipo_);
END IF;
for rtipo in execute sql loop
   return next rtipo;
end loop;
end;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
```


Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
---Contenedor
CREATE TYPE datos_instruccion AS
(
   id_inst INT,
   nom_instruccion VARCHAR(50)
);

--Función
CREATE OR REPLACE FUNCTION muestra_instruccion(id_inst_ VARCHAR)
RETURNS setof datos_instruccion AS
$BODY$
DECLARE
   rins record;
   sql text;
BEGIN
sql := 'SELECT id_inst,nom_instruccion FROM instruccion WHERE 0=0 '; -- Para
IF id_inst_ IS NOT NULL THEN sql := sql || ' AND id_inst='||quote_literal(id_inst_);
END IF;
for rins in execute sql loop
   return next rins;
end loop;
end;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;
```


Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
--Contenedor
CREATE TYPE datos_contenedor_completo AS
(
   dni_pac VARCHAR(8),
   ape_pat_pac VARCHAR(40),
   ape_mat_pac VARCHAR(40),
   nom_pac VARCHAR(40),
   fecha_nac_pac DATE,
   tipo_pac VARCHAR(25),
   inst_pac VARCHAR(50),
   depar_pac VARCHAR(50),
   prov_pac VARCHAR(50),
   dis_pac VARCHAR(50),
   dir_pac VARCHAR(100),
   ec_pac VARCHAR(1),
   sex_pac VARCHAR(1),
   observacion VARCHAR(60)
);


--Función
CREATE OR REPLACE FUNCTION muestra_paciente_completa(dni_pac_ VARCHAR)
RETURNS setof datos_contenedor_completo AS
$BODY$
DECLARE
   r record;
   sql text;
BEGIN
sql := 'select P.dni_pac, P.ape_pat_pac, P.ape_mat_pac, P.nom_pac, P.fecha_nac_pac,
T.nom_convenio, I.nom_instruccion, DE.nom_depar, PR.nom_prov, DI.nom_dist,
P.dir_pac, P.ec_pac, P.sex_pac, P.observacion
```


Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
from paciente P join departamento DE on P.id_depar_pac=DE.id_depar join provincia PR
on P.id_prov_pac=PR.id_prov join distrito DI on P.id_dis_pac=DI.id_dis join
tipo_paciente T on P.id_tipo_pac=T.id_tipo join instruccion I on P.id_inst=I.id_inst
where 0 = 0'; -- Para
IF dni_pac_ IS NOT NULL THEN sql := sql || ' AND dni_pac='||quote_literal(dni_pac_);
END IF;
for r in execute sql loop
   return next r;
end loop;
end;
$BODY$
LANGUAGE plpgsql VOLATILE
COST 100;

--Contenedor
CREATE TYPE datos_contenedor_edita AS
(
   dni_pac_ VARCHAR(9),
   ape_pat_pac_ VARCHAR(40),
   ape_mat_pac_ VARCHAR(40),
   nom_pac_ VARCHAR(40),
   dir_pac_ VARCHAR(100),
   ec_pac_ VARCHAR(1),
   sex_pac_ VARCHAR(1),
   observacion_ VARCHAR(60)
);
```


Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
--Función
CREATE OR REPLACE FUNCTION actualiza_paciente(dni_pac_ VARCHAR,
   ape_pat_pac_ VARCHAR,
   ape_mat_pac_ VARCHAR,
   nom_pac_ VARCHAR,
   dir_pac_ VARCHAR,
   ec_pac_ VARCHAR,
   sex_pac_ VARCHAR,
   observacion_ VARCHAR)
   RETURNS setof datos_contenedor_edita AS
$$
declare begin
UPDATE paciente
SET ape_pat_pac=ape_pat_pac_,
   ape_mat_pac=ape_mat_pac_,
   nom_pac=nom_pac_,
   dir_pac=dir_pac_,
   ec_pac=ec_pac_,
   sex_pac=sex_pac_,
   observacion=observacion_
   WHERE dni_pac=dni_pac_;
end;
$$
LANGUAGE plpgsql VOLATILE;

CREATE TYPE datos_contenedor_elimina AS
(
   dni_pac VARCHAR(9),
   ape_pat_pac VARCHAR(40),
   ape_mat_pac VARCHAR(40),
   nom_pac VARCHAR(40),
   fecha_nac_pac DATE,
   tipo_pac VARCHAR(25),
```


Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

```
    inst_pac VARCHAR(50),
    depar_pac VARCHAR(50),
    prov_pac VARCHAR(50),
    dis_pac VARCHAR(50),
    dir_pac VARCHAR(100),
    ec_pac VARCHAR(1),
    sex_pac VARCHAR(1),
    observacion VARCHAR(60)
);

CREATE OR REPLACE FUNCTION elimina_paciente(
        dni_pac_ VARCHAR) returns setof datos_contenedor_elimina as
$$
BEGIN
Delete From paciente Where dni_pac=dni_pac_;
END;
$$
LANGUAGE plpgsql VOLATILE;
```


Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe

Anexo N° 3

Anexo N° 4

Campus Universitario Km. 5 Carretera a Pimentel - Chiclayo - Perú
Teléf: (+51)(74) 481610 / 481620 - Fax: 203861
www.uss.edu.pe