UACM
Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA DE SOFTWARE

Implementación de una Base de Datos distribuida con MariaDB Galera

T E S I S

PARA OBTENER EL TÍTULO DE

LICENCIADO EN INGENIERÍA DE SOFTWARE

P R E S E N T A :

SERGIO GALICIA GALICIA

D I R E C T O R

DR. JUAN CARLOS CRUZ ROMERO

Ciudad de México, marzo de 2023.

.

UACM
Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno

CA Coordinación Académica

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN



UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO
COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS©

La presente obra y cada uno de sus elementos está protegido por la Ley Federal del Derecho de Autor; por la Ley de la Universidad Autónoma de la Ciudad de México, así como lo dispuesto por el Estatuto General Orgánico de la Universidad Autónoma de la Ciudad de México; del mismo modo por lo establecido en el Acuerdo por el cual se aprueba la Norma mediante la que se Modifican, Adicionan y Derogan Diversas Disposiciones del Estatuto Orgánico de la Universidad de la Ciudad de México, aprobado por el Consejo de Gobierno el 29 de enero de 2002, con el objeto de definir las atribuciones de las diferentes unidades que forman la estructura de la Universidad Autónoma de la Ciudad de México como organismo público autónomo y lo establecido en el Reglamento de Titulación de la Universidad Autónoma de la Ciudad de México.

Por lo que el uso de su contenido, así como cada una de las partes que lo integran y que están bajo la tutela de la Ley Federal de Derecho de Autor, obliga a quien haga uso de la presente obra a considerar que solo lo realizará si es para fines educativos, académicos, de investigación o informativos y se compromete a citar esta fuente, así como a su autor ó autores. Por lo tanto, queda prohibida su reproducción total o parcial y cualquier uso diferente a los ya mencionados, los cuales serán reclamados por el titular de los derechos y sancionados conforme a la legislación aplicable.

Dedicatoria.

Quiero dedicar esta tesis a mis padres María Magdalena Galicia Galindo, Sergio Galicia Pineda por haberme forjado como la persona que soy en la actualidad, muchos de mis logros se los debo a ustedes entre los que se incluye este. Me formaron con reglas y algunas libertades, pero al final de cuentas, me motivaron constantemente para alcanzar mis anhelos.

La recompensa de nuestro trabajo no es lo que obtenemos, sino en lo que nos CONVERTIMOS.

Paulo Coelho.

Agradecimientos

A mi director de tesis Dr. Juan Carlos Cruz Romero, por aceptarme para realizar esta tesis bajo su dirección. Su apoyo y confianza en mi trabajo y su capacidad de guiar mis ideas han sido un aporte invaluable.

A la Universidad Autónoma de la Ciudad de México (UACM), por haberme permitido formarme en ella, sobre todo por haber obtenido la carrera.

Para los miembros del Laboratorio de Cómputo para la Enseñanza de las Ciencias (LACECI), por abrirme las puertas para desarrollarme profesionalmente a lo largo de mi carrera y permitirme participar en distintos proyectos, así como la convivencia con los integrantes de este gran equipo.

A los lectores Lic. Mishel De la Mora García, M. en C. Enrique Cruz Martínez y M. en I. Otoniel Manuel Ortiz Ruiz.

Para mis amigos y compañeros que han compartido los “ires y venires” en el plano escolar, profesional y personal durante la estancia en la UACM y en mi trabajo actual, con los que he compartido grandes recuerdos, no puedo dejar de recordar cuantas tardes y horas de trabajo nos juntamos a lo largo de nuestra formación, les agradezco por su apoyo y constancia, al estar en las horas más difíciles, por compartir horas de estudio. Gracias por estar siempre allí.

Índice General

Introducción 1
Problemática 2
Objetivo 6
1. Bases de Datos 7
1.1. ¿Qué son las bases de datos? 7
1.2. ¿Qué es un sistema manejador de base de datos DBMS? 8
1.3. Motores de bases de datos existentes 10
1.4. DB2 11
1.5. SQL Server 11
1.6. Oracle 13
1.7. MySQL 13
1.8. MariaDB 14
1.9. PostgreSQL 15
2. Bases de datos distribuidas 16
2.1. ¿Qué son las bases de datos distribuidas? 16
2.2. Sistema de gestión de bases de datos distribuidas 18
2.2.1. Procesamiento distribuido 20
2.3. Tipos de bases de datos distribuidas 20
2.4. Replicación 22
2.4.1. Modelo síncrono 22
2.4.2. Modelo asíncrono 23
2.5. Ventajas y desventajas de un sistema de gestión de base de datos distribuido 24
2.5.1. Ventajas 24
2.5.1.1. Estructura organizada 24
2.5.1.2. Mejora de la capacidad de compartir y la autonomía local 24
2.5.1.3. Disponibilidad real 24
2.5.1.4. Fiabilidad mejorada 24
2.5.1.5. Desempeño mejorado 25
2.5.1.6. Economía 25
2.5.1.7. Crecimiento modular 25
2.6.1. Desventajas 25
2.6.1.1. Complejidad 25
2.6.1.2. Costo 26
2.6.1.3. Control de integridad más difícil 26

2.6.1.4. Falta de estándares 26
2.6.1.5. Falta de experiencia 26
2.6.1.6. Diseño de base de datos más complejo 26
2.6. Bases de datos centralizadas versus Base de datos distribuidas 27
3. Sistema Clúster 28
3.1. ¿Qué es un clúster? 28
3.2. Tipos de Clúster 30
3.3. Aplicaciones e Implementación 33
3.4. Balanceadores 34
3.5. Tipos de Balanceadores 35
3.6. Software de Balanceo 36
3.7. Arquitectura de bases de datos distribuida 39
3.8. Algoritmos de replicación 40
3.8.1. Líderes y seguidores. 40
3.8.2. Replicación síncrona. 40
3.8.3. Replicación asíncrona. 40
4. Implementación del Clúster 42
4.1. Construcción del clúster 42
4.1.1. Requerimientos 42
4.1.1.1. Requerimientos de sistema 43
4.1.1.2. Requerimientos de negocio 44
4.2. Software de Clustering para bases de datos 45
4.3. Por qué usar MariaDB Galera Clúster. 48
4.4. Instalación y configuración del MariaDB Galera Clúster 50
4.5. Instalación y configuración MaxScale. 72
4.6. Pruebas 82
5. Conclusiones 94
Glosario 96
Bibliografía 108
Índice de Figuras 113

Introducción

Para la realización de este proyecto se pretende analizar las problemáticas de las instituciones educativas de educación superior públicas las cuales ofrecen educación a distancia en plataformas virtuales, dado que en la pandemia de COVID-19 ha tenido gran impacto en este tipo de educación en todo el mundo.

El aumento de la demanda, la mayoría de las instituciones de educación superior tuvieron que cerrar temporalmente sus instalaciones físicas lo que llevo en un aumento de la demanda de la educación a distancia. Otras tuvieron que adaptarse rápidamente dado que los contagios por esta nueva enfermedad iban en aumento.

Para hacer frente a este problema y dado que no se sabía cuándo se retomarían las clases y el creciente aumento de estudiantes en las instituciones a distancia han tenido que buscar nuevas formas de brindar educación virtual, incluyendo el uso de plataformas en línea, video conferencias y herramientas de colaboración en línea.

Aunque con la educación a distancia ha permitido a muchos estudiantes seguir aprendiendo durante la pandemia y después de la misma (nueva normalidad), se sigue adoptando el modelo a distancia el cual se siguen explorando opciones de educación en línea para el futuro.

Dado que se ha tenido un crecimiento constante en la matrícula estudiantil, pasando de poco más de 5 mil estudiantes en 2013 a más de 116 mil estudiantes en la actualidad. Además, la institución[1] ha establecido convenios de colaboración con diversas instituciones nacionales e internacionales para fomentar la movilidad estudiantil y el intercambio académico.

Actualmente se cuenta con una plataforma Moodle la cual es una plataforma de aprendizaje digital de código abierto (también conocida como sistema de gestión de aprendizaje), ampliamente adoptada por distintas instituciones para crear y administrar cursos en línea, para grandes volúmenes de estudiantes dentro de una institución, esta plataforma es lo suficientemente robusta para servir a miles de estudiantes, administradores, creadores de contenido e instructores simultáneamente.

[1]Se omite el uso del nombre por temas de confidencialidad y de acuerdo de no divulgación.

Problemática

En el semestre 2022-2 se tomó un grupo de aproximadamente 41,516 alumnos inscritos, los cuales iniciaron el semestre el día 11 de Julio y concluyeron 4 de diciembre. Dado lo anterior se obtuvo una medición del tráfico hacia la base de datos, mediante la herramienta de Zabbix, el comportamiento es el de la figura 3 (página 4).

En el gráfico se muestra una alta demanda a principios y mediados del semestre concretamente, a inicios del mes de septiembre es cuando empieza a elevarse la carga, posteriormente hay un decaimiento a finales del mes vuelve a subir (peticiones a la base de datos), para finales del semestre hay un aumento considerable, los picos altos representan la fluctuación en las transacciones dentro de la base de datos, la arquitectura que se tuvo en ese periodo es la siguiente figura 1:

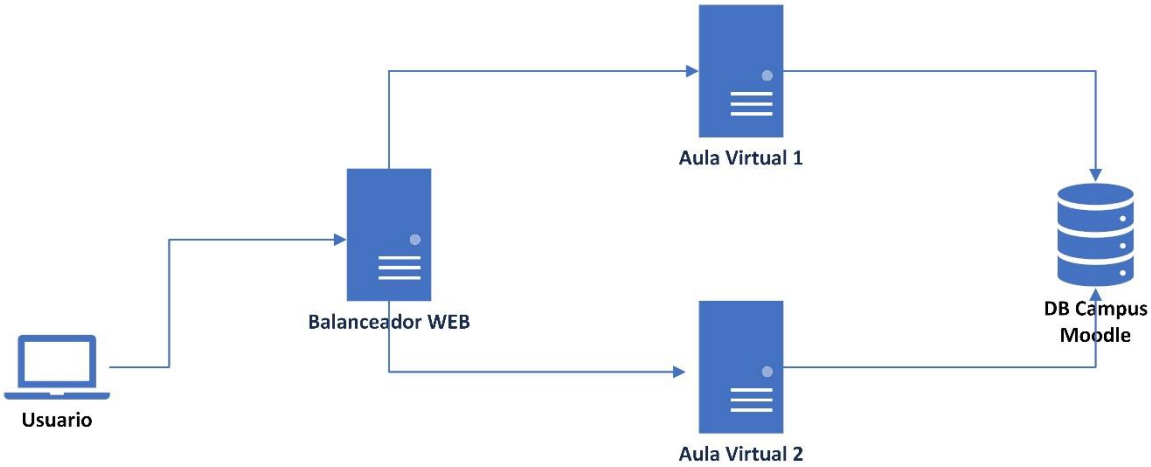


Figura 1 Arquitectura Semestre 2022-2

En el semestre en curso (2023-1) se tomó un grupo aproximadamente de 58,319 alumnos inscritos. El semestre inició el día 23 de enero, la cantidad de transacciones a la base de datos durante un día se muestran en el siguiente gráfico figura 4 (página 5).

Del gráfico anterior se muestra la cantidad de peticiones elevada en las primeras horas, a las 4:00 AM disminuyen, posteriormente hay un aumento considerable iniciando a las 7:30 AM, este aumento se eleva un poco el segundo aumento se da las 2:00 PM y aumenta considerablemente de 5:00 PM a 6:00 PM, este aumento es el más alto a lo largo del día, se tiene un aumento, pero no mayor conforme va finalizado el día, la arquitectura que se tiene es la que se muestra en la siguiente imagen figura 2.



Figura 2 Arquitectura Semestre 2023-1

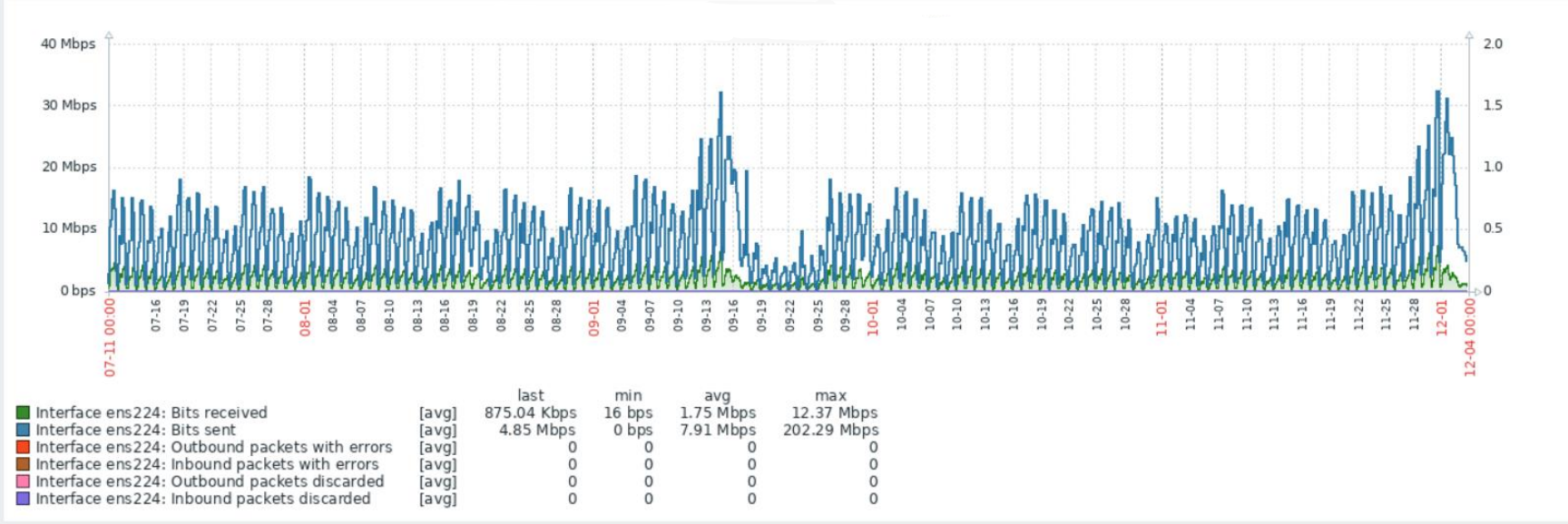


Figura 3 Comportamiento del Semestre 2022-2

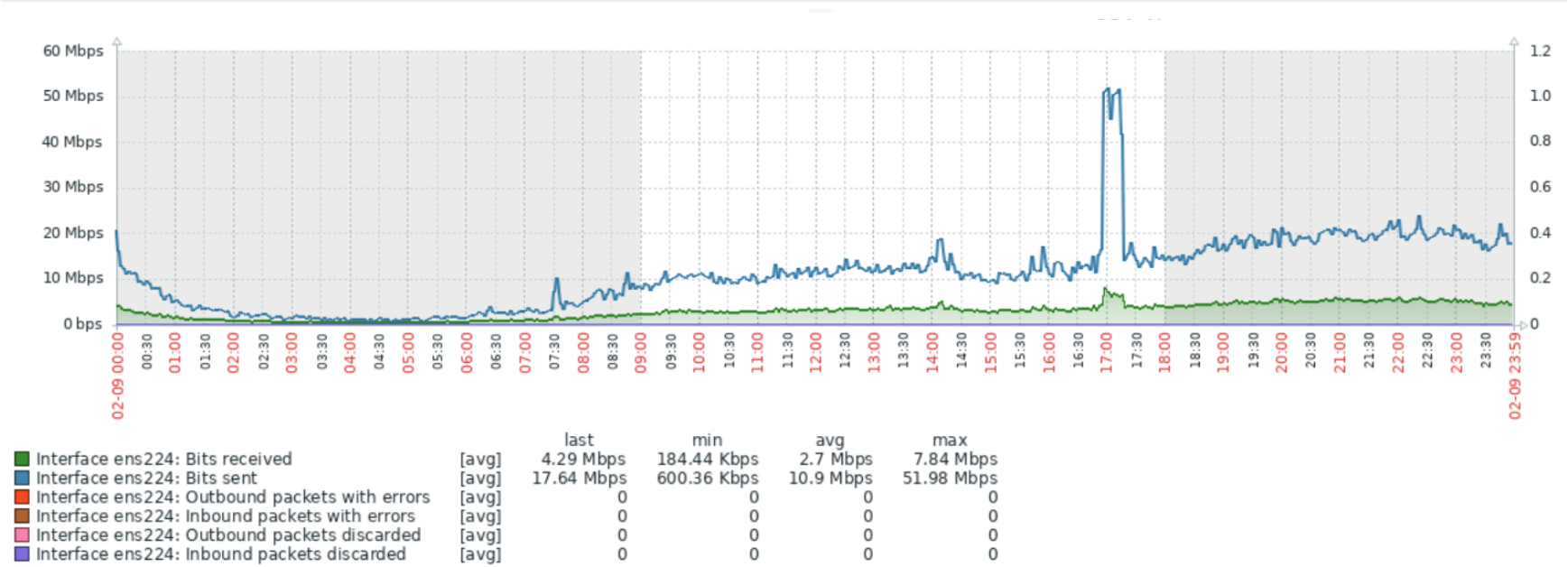


Figura 4 Comportamiento del Semestre 2023-1

Objetivo

Para esta institución educativa es primordial maximizar el tiempo de operación continua de los servicios que entrega la infraestructura de tecnologías de la información (TI), debido a la alta demanda del servicio de aulas virtuales (campus) y al crecimiento constante de la matrícula estudiantil, por lo anterior, se busca proveer de una herramienta que ayude a disminuir el tiempo de peticiones en la base de datos, el cual es insuficiente para satisfacer la alta demanda en el consumo de los servicios por parte de los usuarios (estudiantes), considerando que la plantilla estudiantil se incrementa considerablemente (del 2022-2 al 2023-1 se tuvo un incremento del 40% aproximadamente) y que los accesos fluctúan fuertemente en horas y días específicos, debido a eventos como fechas límite para entrega de tareas, exámenes, o similares; ocasionando un incremento en las peticiones al sitio y por tanto una caída del rendimiento del aplicativo (Moodle).

Se pretende el uso de una solución moderna de gestión de bases de datos que cumpla con las siguientes características:

- Robusto.
- Alta disponibilidad.
- Escalabilidad.
- Soporte de múltiples conexiones
- Replicación.
- Compatible con Moodle.

Ahora se dará una breve introducción de las herramientas actuales: Los sistemas modernos que proveen las características antes mencionadas se conocen como clústeres de base de datos, que son un conjunto de servidores (nodos) interconectados que administran la misma base de datos desde distintos puntos con la finalidad de proporcionar la disponibilidad en caso de falla de alguno de ellos. Como complemento y con fines de administrar correctamente el funcionamiento del clúster nos apoyaremos en un servidor extra configurado como balanceador de carga. Es una especie de front-end invisible al usuario que provee los servicios a través suyo y cuenta con la funcionalidad de monitor de recursos, su principal característica es distribuir las peticiones a la base de datos de la manera más eficiente entre los nodos del clúster.

Para la realizar este proyecto se propone un análisis del software antes mencionado para solucionar esta problemática, dada la ley de austeridad del gobierno en curso se propone el uso de software libre, el cual debe ser compatible con la plataforma de gestión del aprendizaje (Moodle), así como la compatibilidad de servidores de esta institución superior, que cuente con sistema de replicación y contar con un plan de recuperación ante desastres DRP por sus siglas en inglés “Disaster Recovery Plan”[2], asi mismo debe ser escalable.

[2] Un DRP es un plan que una organización o empresa desarrolla para recuperarse de eventos que puedan interrumpir sus operaciones normales, como desastres naturales, fallos en sistemas de tecnología de la información, ataques cibernéticos, entre otros (Hiles, 2014).

1. Bases de Datos

1.1. ¿Qué son las bases de datos?

Vivimos en un mundo donde se recopilan grandes cantidades de información a diario, "Estamos viviendo en la era de la información" es un dicho popular, sin embargo, en realidad vivimos en la era de los datos. Grandes cantidades de petabytes de datos circulan en nuestras redes locales e internet, donde varios dispositivos generan datos todos los días a través de diversas aplicaciones desde el uso de correo electrónico hasta redes socio digitales y chats de diversa índole.

Una base de datos es un conjunto de datos ordenados o comúnmente llamada una colección de archivos interrelacionados entre sí, típicamente almacenados electrónicamente en un sistema de computadora. El contenido de esta engloba a la información (almacenadas en archivos), de tal manera que los datos estén disponibles para los usuarios, con la finalidad de eliminar la redundancia o al menos minimizarla, todo esto es controlado por un sistema de gestión de base de datos DBMS (Data Base Management System, por sus siglas en inglés) (Cap. 1.2).

Los tres componentes principales de un sistema de base de datos son el hardware, el software DBMS y los datos a manejar, así como el personal encargado del manejo del sistema, a este conjunto con las aplicaciones que están asociados con ellos, se conocen como un sistema de base de datos, que a menudo se reducen a solo base de datos. Estos se modelan típicamente en filas y columnas en una serie de tablas para que el procesamiento y la consulta de datos sean eficientes. Luego se puede acceder, administrar, modificar, actualizar, controlar y organizar fácilmente los datos. La mayoría de las bases de datos utilizan lenguaje de consulta estructurado Structured Query Language (SQL)[3] para escribir y consultar datos.

SQL se utiliza para lo siguiente:

- Modificar estructuras de índices y tablas de datos.
- Agregar, actualizar y eliminar.
- Recuperación de subconjuntos de información desde dentro de los sistemas de administración de bases de datos (DBMS), esta información se usa para procesar transacciones y aplicaciones de análisis de datos.

Las consultas SQL y otras operaciones toman la forma de comandos escritos como declaraciones.

[3] SQL es un lenguaje de dominio específico estandarizado, que se utiliza para administrar las bases de datos y realizar varias operaciones con los datos que contienen, Este lenguaje fue creado en la década de los años 1970, es utilizado por administradores de base de datos Data Base Administrator (DBA) como también por desarrolladores que escriben scripts de integración de datos y análisis de datos (Weinberg, 2009)

1.2. ¿Qué es un sistema manejador de base de datos DBMS?

Una base de datos generalmente requiere de un software integral llamado Sistema de Administración de bases de datos (DBMS), el cual sirve como una interfaz entre la base de datos y sus usuarios o programas finales, lo que permite a los usuarios recuperar, actualizar y administrar cómo se organiza y optimiza la información, facilita la supervisión y el control de esta, lo que permite una variedad de operaciones administrativas, como la supervisión del rendimiento, el ajuste, las copias de seguridad y la recuperación.

Este motor (software) permite acceder, bloquear y modificar los datos, el esquema de base de datos define la estructura de estos, los tres elementos fundamentales ayudan a proporcionar concurrencia, seguridad e integridad de los datos y procedimientos, uniformes de administración de datos,

El DBMS admite muchas tareas típicas de la administración de la base de datos, incluyendo la gestión de cambios, la supervisión y el ajuste del rendimiento, así como la seguridad, copias de seguridad y recuperación, proporciona una vista centralizada de los datos a los que pueden acceder múltiples usuarios desde múltiples ubicaciones de manera controlada, así mismo el software puede limitar los datos que puede visualizar los usuarios finales, proporcionando muchas vistas de un solo esquema de base de datos, los usuarios ni el software no tienen manera de saber dónde se encuentran físicamente los datos o qué tipo de almacenamiento residen porque el DBMS maneja todas las solicitudes.

Componentes de un DBMS

Motor de almacenamiento

Este elemento básico es usado por el DBMS se utiliza para almacenar los datos, con el cual el software debe interactuar con el sistema de archivos al nivel sistema operativo, puede usar componentes adicionales para almacenar datos o interactuar con los datos reales.

Catálogo de metadatos

A veces llamado catálogo del sistema o diccionario de la base datos (diccionario de datos), este funciona como un depósito para todos los objetos de la base de datos que se han creado. Es decir, cuando se crea toda la estructura y los objetos de esta, el DBMS registra automáticamente la información sobre ellos, verifica las solicitudes de cada uno de los usuarios y a su vez estos puedan consultar el catálogo, este puede incluir información sobre los objetos de base de datos, esquemas, programas, seguridad, rendimiento, comunicación y otros detalles de administración.

Idioma de acceso

El DBMS debe proporcionar una API "*Interfaz de Programación de Aplicaciones*" (es un mecanismo que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos) para acceder a los datos, generalmente en forma de un lenguaje de acceso, esto para acceder y modificar datos dentro de la propia base de datos, pero también puede usarse para crear objetos dentro de la misma (asegurar y automatizar), SQL es un ejemplo del lenguaje de acceso a la base de datos y abarca varios conjuntos de instrucciones, incluido el lenguaje de control de datos para autorizar el acceso a los mismos.

Motor de optimización

Un DBMS puede proporcionar un motor de optimización el cual se utiliza para analizar las solicitudes de idioma de acceso a la base de datos y convertirlas en instrucciones procesables para acceder y modificar los datos.

Procesamiento de consultas

Después de optimizar una consulta, el DBMS debe proporcionar un medio para ejecutar la consulta y devolver los resultados correspondientes.

Administrador de bloqueo

Es un componente crucial del DBMS gestiona el acceso simultáneo a los mismos datos, los bloqueos son necesarios para garantizar que varios usuarios no intenten modificar los mismos datos simultáneamente.

Administrador de registros

El DBMS registra todos los cambios realizados en los datos administrados por el DBMS, el registro, y el componente "administrador de registros" se utiliza para garantizar que los registros se realicen de manera eficiente y precisa, por otra parte, se utiliza el administrador de registros durante el apagado y el inicio para garantizar la integridad de los datos, e interactúa con las utilidades de la base de datos para crear copias de seguridad y ejecutar recuperaciones.

Utilidades de datos

El software (DBMS) también proporciona un conjunto de utilidades para administrar y controlar las actividades de la base de datos por ejemplo se incluyen reorganización, actualizar estadísticas (runstats), respaldos y copias, recuperación, verificaciones de integridad, carga de datos, descarga de datos y recuperación de la base de datos[4].

[4] (Abraham Silberschatz, 2014) (Mullins, 2022)

1.3. Motores de bases de datos existentes

Actualmente es casi imposible concebir un sistema de información que no maneje una o más bases de datos y es una pieza fundamental que se encarga de ello, comúnmente le llamamos motor de base de datos (antiguamente llamado sistema gestor o sistema de administración de base de datos).

Un motor de base de datos es un software que se encarga de gestionar y almacenar los datos de una base de datos, así como de proporcionar una serie de herramientas y funcionalidades para acceder y manipular los datos de forma eficiente. Estos son un componente esencial en cualquier sistema de gestión de base de datos y su elección puede tener un impacto significativo en el rendimiento y la escalabilidad del sistema.

Existen diferentes tipos de motores de base de datos, como los motores de base de datos relacionales, los motores de base de datos NoSQL y los motores de base de datos orientados a grafos. Cada uno de estos tipos de motores se adapta mejor a determinados tipos de aplicaciones y usos de la base de datos. Los motores de base de datos son esenciales en la mayoría de las aplicaciones informáticas modernas, ya que proporcionan un lugar centralizado para almacenar y gestionar los datos de la aplicación. Gracias a ellos, es posible realizar consultas y actualizaciones de datos de manera eficiente y rápida, están compuesto principalmente por tres lenguajes, siendo estos DDL[5] (Lenguaje de definición de datos), DML[6] (Lenguaje de manipulación de datos) y SQL (Lenguaje de consulta).

Existen varios parámetros que deben cumplir los motores de base de datos para ser considerados eficientes y adecuados para su uso en una aplicación determinada. Algunos de estos parámetros son:

- Escalabilidad: la capacidad del motor de base de datos para manejar una cantidad creciente de datos y de usuarios sin disminuir el rendimiento.

- Disponibilidad: la capacidad del motor de base de datos para estar disponible y operativo en todo momento, incluso en caso de fallos o problemas.

- Rendimiento: la velocidad a la que el motor de base de datos puede procesar y devolver las consultas y transacciones realizadas por los usuarios.

- Fiabilidad: la capacidad del motor de base de datos para mantener la integridad y la consistencia de los datos almacenados, incluso en caso de fallos o problemas.

- Seguridad: la capacidad del motor de base de datos para proteger los datos almacenados y controlar el acceso a ellos.

- Eficiencia: la capacidad del motor de base de datos para utilizar de manera eficiente los recursos informáticos disponibles, como la memoria, el disco y la CPU[7].

[5] Data Definition Language
[6] Data Manipulation Language
[7] (Hector Garcia-Molina, 2008) (Tow, 2005) (Fowler, 2015)

Algunos motores de base de datos son:

1.4. DB2

Original estaba disponible exclusivamente en mainframes de IBM, pero en la década de 1990, se ha migrado a una variedad de otras plataformas, incluidas (Linux, Unix, Windows, IBM i, VSE, VM, y z/OS).

z/OS es el sistema operativo principal para la plataforma de hardware más potente de IBM, IBMZ, DB2 para z/OS es el servidor de datos de la empresa de para IBM Z, gestiona los datos empresariales en una empresa y da soporte a aplicaciones empresariales.

DB2 es una familia de productos (sistema manejador de base de datos DBMS) de IBM, es utilizado por diferentes organizaciones de todos los tamaños, la cual proporciona una plataforma de datos para operaciones transaccionales y analíticas, así como la disponibilidad asi como para mantener flujos de trabajo transaccionales y el análisis funcionando de manera eficiente.

Además de ser un DBMS relacional, ofrece soporte integrado para varias capacidades NoSQL, incluyendo XML, almacenamiento de gráficos y notación de objetos JavaScript o JSON[8].

1.5. SQL Server

En 1988 Microsoft publicó su primera versión de SQL server, diseñada para sistemas OS/2 y fue un desarrollo conjunto con Sybase. A partir de la década de los 90's, Microsoft empezó a desarrollar su versión para plataformas NT, en continua evolución con la adopción del lenguaje Transact SQL o T-SQL, originalmente de IBM, se doto al lenguaje ciertas mejoras para permitir a los usuarios la definición de funciones propias, entre otras mejoras.

SQL Server es un sistema de administración de base de datos relacionales -Relational Data Base Management Systems- (RDMS) que admite una gran amplia variedad de aplicaciones de procesamiento de transacciones, inteligencia comercial y análisis en entornos de TI corporativos, Microsoft SQL Server es una de las tres tecnologías líderes de base de datos en el mercado, junto con Oracle y DB2 de IBM.

Al igual que los demás softwares RDMS, SQL server se basa en el lenguaje SQL el cual es el lenguaje de programación estandarizado que los administradores de base datos (DBA) entre otros usuarios que puedan administrar la base de datos y consultar los datos que contienen, SQL Server está vinculado a Transact-SQL (lenguaje de búsquedas estructurado por sus siglas en ingles), una implementación que Microsoft agrega un conjunto de extensiones de programación propietarias al lenguaje estándar.

Al igual que otras tecnologías RDBMS, este sistema manejador de base de datos se basa principalmente en una estructura de tabla basada en filas que conecta elementos de datos relacionados en diferentes tablas entre sí, evitando la necesidad de almacenar datos de forma redundante en varios lugares dentro de la base de datos, el modelo relacional también proporciona integridad referencial y otras restricciones de integridad para mantener la precisión de los datos.

[8] (IBM, 2022) (IBM, 2022)

Estas verificaciones son parte de una adhesión más amplia a los principios de atomicidad, consistencia, aislamiento y durabilidad, conocidos colectivamente como propiedades ACID, y están diseñadas para garantizar que las transacciones de la base de datos se procesen de manera confiable[9].

Las operaciones ACID son un conjunto de propiedades que se utilizan para garantizar la integridad y la consistencia de los datos en una base de datos. Estas propiedades son:

- **Atomicidad:** Esta propiedad garantiza que todas las operaciones de una transacción se completarán o se revertirán por completo. Esto significa que, si una transacción incluye varias operaciones, todas ellas se realizarán o ninguna de ellas. Esto es importante para evitar que los datos queden en un estado incógnito o inconsistente.

- **Consistencia:** Esta propiedad asegura que todas las operaciones de una transacción dejarán la base de datos en un estado válido. Por ejemplo, si una transacción incluye la creación de una nueva tabla y la inserción de datos en ella, la base de datos debe estar en un estado válido tanto antes como después de la transacción.

- **Aislamiento:** Esta propiedad garantiza que las transacciones se ejecutan de manera aislada una de otra. Esto significa que, si dos transacciones se están ejecutando al mismo tiempo, no se afectarán mutuamente. Esto es importante para evitar conflictos entre transacciones y garantizar la integridad de los datos.

- **Durabilidad:** Esta propiedad asegura que una vez que se ha completado una transacción, los cambios realizados en la base de datos son permanentes y no pueden ser revertidos[10].

[9] (Banescu, 2018) (Souza, 2018) (Microsoft, s.f.)
[10] (Snodgrass, 2000) (Gray, 1993)

1.6. Oracle

Es presentado 1977 con el nombre SDL (Sofware Development Laboratories por sus siglas en inglés), en 1979 SDL cambia su nombre a Relational Software, Inc. (RSI) La función de SDL fue movida principalmente por un estudio sobre los SGBD (Sistemas Gestores de Base de Datos), en la actualidad Oracle (Nasdaq: ORCL), esta tecnología se encuentra casi en todas las industrias alrededor del mundo y en las oficinas de 98 de cada 100 empresas

Oracle Database es un tipo de sistema de gestión de base de datos de tipo objeto-relacional (ORDBMS Object Relational Data Base Management System), desarrollado por Oracle Corporation. Este software es considerado como uno de los sistemas manejadores de base de datos más completos sus principales características son: Soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma.

Es la base de datos número uno en el mercado ya que posee la capacidad de soportar todos los datos relacionales estándares, así como también los oriundos como XML, XQuery y WebDAV, JDBC, SQLJ ODBC.Net, OLE.Net, ODP.Net, los procedimientos pueden estar escritos en java PL SQL o utilizando .Net CLR

Es la primera base de datos diseñada para la computación del tipo grid empresarial, que es la forma más flexible y rentable de administrar información y aplicaciones. La computación en red empresarial crea grandes grupos de servidores y almacenamiento modular estándar de la industria, con esta arquitectura, cada nuevo sistema se puede aprovisionar rápidamente desde el grupo de componentes. No hay necesidad de cargas de trabajo máximas, porque la capacidad se puede agregar o reasignar fácilmente desde los grupos de recursos según sea necesario. Tiene estructuras lógicas y estructuras físicas, debido a que las estructuras físicas y lógicas están separadas, el almacenamiento físico de datos se puede administrar sin afectar el acceso a las estructuras de almacenamiento lógicas[11].

1.7. MySQL

Es un software de gestión de bases de datos que cuenta con una doble licencia, por una parte, es de código abierto, pero por la otra cuenta con una versión comercial gestionada por la compañía Oracle. En la actualidad es la base de datos de código abierto más popular y utilizada en el mundo entero. Al igual que Microsoft SQL Server todas tienen la finalidad y se utilizan en el mismo entorno y su uso actual es la de dar forma y facilitar la comunicación entre las páginas web y servidores. Tras su difusión a comienzos de la década de 2000, MySQL se dedicaba principalmente a las aplicaciones personales o profesionales de gama baja. En los últimos años se han caracterizado por la adhesión de los grandes protagonistas de la web. Hoy en día, la inmensa mayoría de los sitios de gran densidad de tráfico, como las redes sociales o muchos portales comunitarios los usan, ha alcanzado su nivel de rendimiento mediante el uso intensivo de este manejador de base de datos.

Una de las ventajas más importantes de MySQL es que trabaja con bases de datos relacionales, es decir, utiliza múltiples tablas que se interconectan entre sí para almacenar la

[11] (Oracle, 2022) (Forma talent Bussines School, 2022)

información y organizarla correctamente, a pesar de su finalidad y su entorno en el que se utiliza, cabe destacar que es un sistema desarrollado originalmente en C y C++.

Si bien es un sistema que sirve para almacenar datos de manera relacional (base de datos relacional), como también para administrar todos estos datos sin apenas complicaciones gracias a su interfaz visual y a todas las opciones y herramientas de las que dispone, para las organizaciones u compañías usar esta herramienta es vital por las facilidades que plantea y las utilidades que tiene[12].

1.8. MariaDB

Es un Sistema Gestor de Dase de Datos (SGBD), es decir un conjunto de programas que permiten modificar, almacenar y extraer información. MariaDB surge a raíz de la compra de la compañía desarrolladora de otro (SGBD) llamado MySQL por la empresa Sun Microsystems. El desarrollador original, decide tomar el código fuente de MySQL para posteriormente modificarlo para convertirlo en MariaDB, permitiendo así la existencia de una versión de este producto con licencia GPL (General Public License). Es parte de la mayoría de las ofertas en la nube y es el predeterminado en la mayoría de las distribuciones de Linux. Esta base de datos (Software) se utiliza para diversos fines, como el almacenamiento de datos, comercio electrónico funciones al nivel empresarial y aplicaciones de registro.

Al ser de código abierto, lo que significa que se puede utilizar de manera gratuita, el alojamiento de este sistema manejador de base de datos puede ser muy económico y tiene mucha documentación en internet. Como se comentó anteriormente es un sistema de gestión de contenido, también conocidos como CMS. UN CMS es un sistema de publicación lo que permite a los usuarios publicar y gestionar grandes cantidades de contenido dentro de un sitio web, sus principales usos son la rapidez, escalabilidad y robustez, las últimas versiones de sistema incluyen funciones GIS y JSON[13].

[12] (neoattack, 2020) (Oracle, 2022)
[13] (Mariadb, 2022) (Galvis, 2020)

1.9. PostgreSQL

PostgreSQL es un sistema de gestión de bases de datos relacionales de código abierto que ha ganado mucha popularidad en los últimos años debido a su robustez, seguridad y capacidad de escalabilidad, desarrollado originalmente en la Universidad de California en Berkeley. Es uno de los sistemas de bases de datos más populares del mundo y se utiliza ampliamente en aplicaciones empresariales, de ciencia de datos y web.[14]

Una de las características más destacadas es su capacidad para manejar grandes cantidades de datos, puede manejar conjuntos de datos masivos con facilidad, y sus características de escalabilidad lo convierten en una opción popular para aplicaciones empresariales y científicas. Además, cuenta con soporte para múltiples plataformas, lo que lo hace accesible para una amplia variedad de usuarios.

Sobre todo, lo más importante es su compatibilidad con SQL, esto significa que los desarrolladores y administradores de bases de datos pueden utilizar el lenguaje SQL estándar para interactuar con las bases de datos de PostgreSQL, lo que facilita la integración con otras aplicaciones y sistemas además es altamente extensible. Los usuarios pueden integrar fácilmente código personalizado y desarrollar funciones de la base de datos personalizadas para satisfacer sus necesidades específicas. Esto lo hace una opción popular para aplicaciones web, científicas y empresariales que requieren un alto grado de personalización.

Al ser un software de código abierto, tiene una comunidad de desarrolladores activa que ha creado una amplia variedad de herramientas y recursos para ayudar a los usuarios a aprovechar al máximo sus capacidades. La documentación oficial de PostgreSQL es extensa y se actualiza regularmente para mantenerse al día con las últimas[15] características y funcionalidades del sistema.

[14] (The PostgreSQL Global Development Group, 2023)
[15] (PostgreSQL 15.2, 2023)

2. Bases de datos distribuidas

2.1. ¿Qué son las bases de datos distribuidas?

Las bases de datos distribuidas son aquellas que están divididas en dos o más componentes, que pueden estar situados en máquinas diferentes o en diferentes ubicaciones geográficas. Estos componentes pueden funcionar de forma independiente, pero se encuentran interconectados de manera que pueden compartir información y trabajar juntos para proporcionar servicios de base de datos, se utilizan a menudo cuando es necesario proporcionar acceso a los datos desde diferentes lugares geográficos o cuando se necesita alta disponibilidad y redundancia de los datos. También pueden ser útiles para mejorar el rendimiento y la escalabilidad de una base de datos.

Existen diferentes tipos de bases de datos distribuidas, como las bases de datos distribuidas centralizadas, en las que una única máquina actúa como controlador central y coordina el acceso a los datos almacenados en otras máquinas. Una característica principal la transparencia de ubicación, lo que significa que las aplicaciones no necesitan conocer la ubicación exacta del sitio donde se almacenan los datos. Cuando se ejecuta una consulta en una base de datos distribuida, un grupo colectivo de sitios en diferentes centros de datos trabajan en conjunto para responder la pregunta.

También existen las bases de datos distribuidas descentralizadas, en las que cada máquina tiene una copia de todos los datos y puede funcionar de forma independiente sin depender de un controlador central.

Un sistema centralizado de administración de bases de datos distribuidas -Distributed Data Base Management System- (DDBMS) integra los datos de manera lógica para que puedan administrarse como si estuvieran almacenados en la misma ubicación. El DDBMS sincroniza todos los datos periódicamente y garantiza que las actualizaciones y eliminaciones de datos realizadas en una ubicación se reflejen automáticamente en los datos almacenados en otro lugar. Por el contrario, de una base de datos centralizada consta de un único archivo de base de datos ubicado en un sitio que utiliza una única red.

Uno de los principales problemas a los que se enfrentan las bases de datos distribuidas es la gestión de la consistencia de los datos. Es importante garantizar que todas las copias de los datos estén sincronizadas y que no haya discrepancias entre ellas. Para ello, se utilizan diferentes técnicas de control de concurrencia y de resolución de conflictos.

Otro problema que se presenta en las bases de datos distribuidas es la comunicación entre los componentes. Es importante garantizar una comunicación rápida y eficiente entre los componentes para poder proporcionar un servicio de base de datos óptimo.

Una de las principales ventajas de las bases de datos distribuidas es la posibilidad de distribuir la carga de trabajo entre diferentes componentes, lo que permite aumentar la capacidad y el rendimiento del sistema. También pueden proporcionar alta disponibilidad, ya que, si un componente falla, otros componentes pueden seguir proporcionando servicios de base de datos.

Sin embargo, las bases de datos distribuidas también presentan desafíos. Uno de ellos es la complejidad añadida en la gestión y el diseño del sistema, ya que se deben tener en cuenta aspectos como la sincronización de datos y la resolución de conflictos. Además, pueden ser más difíciles de programar y depurar debido a la naturaleza distribuida del sistema[16].

[16] (Moore, s.f.) (Fauna, 2020)

2.2. Sistema de gestión de bases de datos distribuidas

El sistema de software que permite la gestión de la base de datos distribuida y hace que la distribución sea transparente para los usuarios, un sistema de administración de base de datos distribuidas (DDBMS) consta de una sola base de datos lógica que se divide en varios fragmentos. Cada fragmento se almacena en una o más computadoras mediante el control por separado de un DBMS, con las computadoras conectadas por una red de comunicaciones, cada sitio es capaz de procesar de forma independiente las solicitudes hechas por los usuarios los cuales requieren acceso a los datos locales, es decir cada sitio tiene un grado de autonomía local y también es capaz de procesar datos almacenados en otras computadoras en red.

Los usuarios acceden a la base de datos distribuida a través de aplicaciones, estas se clasifican en aquellas que no requieren datos de otros sitios (aplicaciones locales) y aquellas que si requieran datos de otros sitios (aplicaciones globales), por lo que un DDBMS debe tener al menos una aplicación global.

Un DDBMS tiene las siguientes características:

- Colección de datos compartidos lógicamente relacionados.
- Los datos se dividen en varios fragmentos.
- Los fragmentos se pueden replicar.
- Los fragmentos (réplicas), se asignan a los sitios.
- Los sitios están vinculados por una red de comunicaciones.
- Los datos de cada sitio están vinculados por una red de comunicaciones.
- Los datos de cada sitio están bajo control de un DBMS.
- El DBMS en cada sitio puede manejar aplicaciones locales de forma autónoma.
- Cada DBMS participa en al menos una aplicación local.

No es necesario que cada sitio en el sistema tenga su propia base de datos local, como lo ilustra en la topología del DDBMS la cual es muestra en la siguiente imagen[17] (Figura 5).

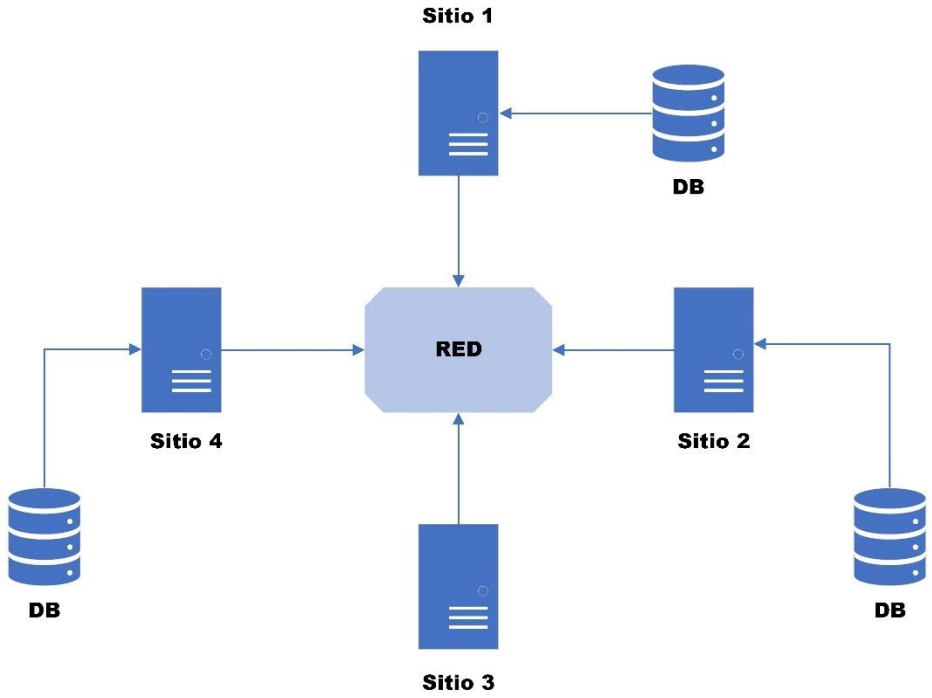


Figura 5 Topología de un DDBMS

[17] (Computer Notes, 2022)

2.2.1. Procesamiento distribuido

Es importante hacer una distinción entre un DBMS distribuido y el procesamiento distribuido.

Procesamiento distribuido es una base de datos centralizada a la que se puede acceder a través de una red informática. Sin embargo, el punto clave con la definición de un DBMS distribuido es el sistema que consta de datos que se distribuyen físicamente en varios sitios de la red, los datos están centralizados a pesar de que otros usuarios pueden acceder a los datos a través de la red, no consideramos que esto sea un DBMS distribuido, simplemente un procesamiento distribuido[18].

2.3. Tipos de bases de datos distribuidas

Existen dos tipos de arquitecturas de bases de datos las cuales son las siguientes:

Heterogéneas: Está conformado por distintos productos DBMS, que no necesariamente tiene que estar basados en el mismo modelo de datos subyacente, es decir, el sistema puede estar compuesto por distintos DBMS relacionales, en red, jerárquicos y orientados a objetos, generalmente esto resulta cuando los sitios han implementado su propia base de datos y la integración se considera en una etapa posterior.

En un sistema heterogéneo, se requieren traducciones para permitir la comunicación entre diferentes DBMS, para proporcionar transparencia los usuarios, deben poder realizar solicitudes en el idioma del DBMS en su sitio local, por lo que este tiene la tarea de localizar los datos y realizar cualquier traducción necesaria, es posible que se requieran datos de otro sitio que pueda tener:

- Hardware diferente.
- Diferentes productos DBMS.
- Diferente Hardware y diferentes productos DBMS.

Si el hardware es diferente, pero los productos DBSM son los mismos, la traducción es sencilla, lo que implica el cambio de código y longitudes de palabras. Si los productos DBMS son diferentes, la traducción es complicada, ya que involucra el mapeo de la estructura de datos en un modelo de datos a las estructuras de datos equivalentes en otro modelo de datos.

Por ejemplo, las relaciones en el modelo de datos relacionales se asignan a registros y conjuntos en el modelo de red, es necesario traducir el lenguaje de consulta utilizado, las sentencias SQL SELECT las cuales se asignan a las sentencias FIND y GET de la red, si tanto el hardware como el software son diferentes, se requieren estos dos tipos de traducción, esto hace que el procesamiento sea extremadamente complejo.

Una solución típica que utilizan algunos sistemas relacionales que forman parte de los DDBMS heterogéneos es utilizar puertas de enlace, que convierten el lenguaje y modelo de cada DBMS diferente en el lenguaje y el modelo del sistema relacional.

[18] (Computer Notes, 2022)

Homogéneas: Todos los sitios utilizan el miso producto DBMS, son más fáciles de diseñar y administrar, lo cual nos proporciona un crecimiento incremental, lo que nos facilita agregar nuevos sitios al DDBMS (Distribuides Data Base Management System), lo que permite mayor rendimiento al explotar la capacidad de procesamiento paralelo en múltiples sitios[19].

[19] (Computer Notes, 2022)

2.4. Replicación

La replicación de datos es el proceso de copiar y mantener los datos en varias bases de datos o servidores de bases de datos. En una base de datos distribuida, los datos se dividen y se almacenan en múltiples servidores de bases de datos, es una técnica que se utiliza para mantener una copia exacta en los mismos (figura 6), lo cual mejorar el rendimiento y la disponibilidad de los datos. La replicación de datos se utiliza para asegurar que todas las copias de los datos estén sincronizadas y actualizadas, lo que permite a los usuarios acceder a la misma información desde cualquier servidor de base de datos, lo que es especialmente útil en entornos empresariales en los que es necesario acceder a la información de manera rápida y fiable. La replicación en entornos distribuidos también puede utilizarse para mejorar la seguridad de la información almacenada, ya que, si uno de los servidores se daña o se vuelve inaccesible, las copias de la base de datos en otros servidores aún estarán disponibles.

Hay varios tipos de replicación de bases de datos distribuidas, incluyendo la replicación síncrona y la replicación asíncrona.

2.4.1. Modelo síncrono

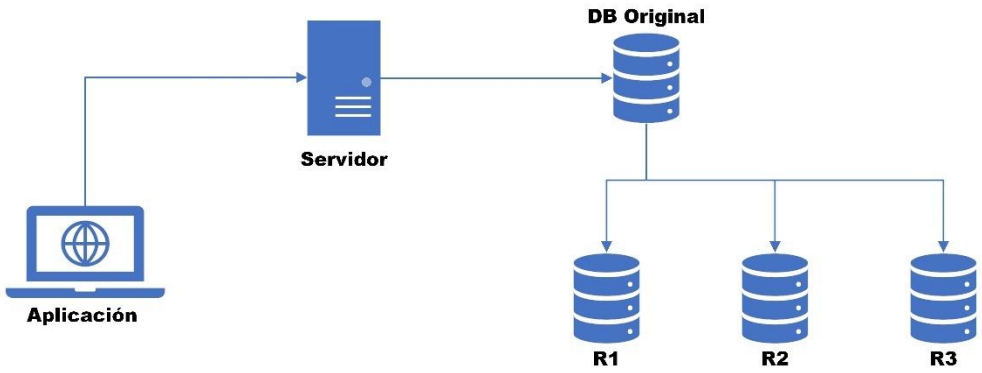


Figura 6 Diagrama de Replicas

Es un enfoque utilizado para mantener la consistencia y la disponibilidad de los datos en una base de datos distribuida. Este modelo se basa en la idea de que todas las réplicas de la base de datos deben estar sincronizadas y alineadas en todo momento. Cuando se realiza una transacción en una de las réplicas, se bloquea la escritura en todas las demás réplicas hasta que la transacción se haya completado y sincronizado en todas las réplicas. Esto garantiza que todas las réplicas tengan la misma información en todo momento y evita problemas de inconsistencias de datos.

Sin embargo, esta sincronización constante tiene un costo en términos de rendimiento, al bloquear la escritura en todas las réplicas durante la transacción, se reduce la capacidad del sistema para manejar grandes cantidades de tráfico de escritura simultánea. Además, si una de las réplicas falla durante la transacción, el proceso de sincronización se retrasará y puede afectar la disponibilidad del sistema.

A pesar de estos inconvenientes, el modelo sincrónico de replicación es ampliamente utilizado en sistemas que requieren alta disponibilidad y consistencia de datos, como sistemas financieros o de salud. En estos casos, la precisión de los datos es más importante que el rendimiento y se considera un precio aceptable pagar por la sincronización constante.

El modelo sincrónico de replicación es utilizado para mantener la consistencia y la disponibilidad de los datos en una base de datos distribuida, aunque tiene un costo en términos de rendimiento, como sistemas financieros o de salud, aunque puede no ser la opción más adecuada en entornos con altos requisitos de rendimiento.

2.4.2. Modelo asíncrono

Es un modelo utilizado para mantener la disponibilidad de los datos en una base de datos distribuida. En este, las réplicas de la base de datos no están sincronizadas en tiempo real y pueden tener versiones ligeramente diferentes de los datos en cualquier momento. En lugar de bloquear la escritura en todas las réplicas durante una transacción, el modelo asincrónico permite la escritura simultánea en todas las réplicas. Esto aumenta la capacidad del sistema para manejar grandes cantidades de tráfico de escritura y reduce los retrasos causados por la sincronización.

Sin embargo, la falta de sincronización en tiempo real puede llevar a problemas de consistencia de datos. Si se realizan varias escrituras en diferentes réplicas sin sincronizar, es posible que algunas réplicas tengan versiones conflictivas de los datos. Para resolver estos conflictos, se pueden utilizar técnicas como el uso de *timestamps* (registros de tiempo) o la resolución de conflictos mediante la elección de la versión más reciente.

A pesar de estos problemas de consistencia, el modelo asincrónico de replicación es adecuado para entornos con altos requisitos de rendimiento y disponibilidad. Es comúnmente utilizado en aplicaciones en tiempo real, como sistemas de mensajería instantánea y redes sociales, donde es importante permitir la escritura simultánea y evitar retrasos en la sincronización.

El modelo asincrónico de replicación es utilizado para mantener la disponibilidad de los datos en una base de datos distribuida, con un mayor rendimiento y capacidad de manejo de tráfico de escritura que el modelo sincrónico. Sin embargo, tiene un mayor riesgo de inconsistencias de datos y puede no ser adecuado en entornos donde la consistencia de los datos es crítica[20].

[20] (Mohan, 1990) (Ozsu, 2011)

2.5. Ventajas y desventajas de un sistema de gestión de base de datos distribuido

La distribución de datos y aplicaciones tiene ventajas potenciales sobre los sistemas de base de datos centralizados tradicionales, desafortunadamente hay desventajas, en este apartado revisaremos las ventajas y desventajas de DDBMS.

2.5.1. Ventajas

2.5.1.1. Estructura organizada

Muchas organizaciones están distribuidas en varios lugares, por ejemplo, en un banco tiene muchas oficinas en diferentes ciudades, es común que las bases de datos utilizadas en una aplicación de este tipo se distribuyan en estas ubicaciones, un banco puede mantener una base de datos en cada sucursal que contenga detalles como el personal que trabaja en ese lugar, la información de la cuenta de los clientes entre otros datos, sin embargo el personal de una sucursal hará consultas locales de la base de datos, La sede de la empresa puede desear realizar consultas globales que involucren el acceso a datos en todas o varias sucursales.

2.5.1.2. Mejora de la capacidad de compartir y la autonomía local

La distribución geográfica de una organización se puede reflejar en la distribución de los datos, los usuarios de un sitio pueden acceder a los datos almacenados en otros sitios, estos se pueden colocar en el sitio cerca de los usuarios que normalmente usan esos datos. De esta manera, los usuarios tienen control local de los datos y en consecuencia, pueden establecer y hacer cumplir políticas locales con respecto al uso de estos datos, un administrador de base de datos global (DBA) es responsable de todo el sistema, generalmente, es parte de esta responsabilidad se asigna al nivel local, de modo que el DBA local pueda administrar el DBMS local.

2.5.1.3. Disponibilidad real

En un DBMS centralizado, una falla en la computadora finaliza las aplicaciones del DBMS, sin embargo, una falla en un sitio de un DDBMS, o una falla en un enlace de comunicación que hace que algunos sitios sean inaccesibles, no hace que todo el sistema funcione, los DBMS distribuidos están diseñados para continuar funcionando a pesar de tales fallas, si falla un solo nodo, el sistema puede redirigir las solicitudes del nodo fallido a otro sitio.

2.5.1.4. Fiabilidad mejorada

Como los datos se pueden replicar de modo que existan en más de un sitio, la falla de un nodo o de un enlace de comunicación no hace necesariamente que los datos sean inaccesibles.

2.5.1.5. Desempeño mejorado

Como los datos se encuentran cerca del sitio de mayor demanda y dado el paralelismo inherente de los DBMS distribuidos, la velocidad de acceso a la base de datos puede ser mejor que la que se puede lograr desde una base de datos centralizada remota. Además, dado que cada sitio maneja solo una parte de la base de datos completa, es posible que no haya la misma competencia para los servicios de E/S (Entrada/Salida) y la unidad central de proceso -Central Processing Unit- (CPU)[21] que caracteriza a un DBMS centralizado.

2.5.1.6. Economía

Generalmente cuesta mucho menos crear un sistema de computadoras más pequeñas con la potencia equivalente a una sola computadora grande, ya que es más rentable para las divisiones y departamentos corporativos obtener computadoras separadas. Por otro lado, es mucho más rentable agregar estaciones de trabajo a una red que actualizar un sistema central. El segundo ahorro de costos potencial ocurre cuando la base de datos está geográficamente remota y las aplicaciones requieren acceso a datos distribuidos.

En tales casos, debido al costo relativo de la transmisión de datos a través de la red en comparación con el costo del acceso local, puede ser mucho más económico dividir la aplicación y realizar el procesamiento localmente en cada sitio.

2.5.1.7. Crecimiento modular

En un entorno distribuido, es mucho más fácil manejar la expansión, se pueden agregar nuevos sitios a la red sin afectar las operaciones de otros sitios. Esta flexibilidad permite que una organización se expanda con relativa facilidad.

Agregar potencia de procesamiento y almacenamiento a la red generalmente puede manejar el aumento en el tamaño de la base de datos, en un DBMS centralizado, el crecimiento puede implicar cambios tanto en el hardware (la adquisición de un sistema más potente) como en el software (la adquisición de un DBMS más potente o configurable).

2.6.1. Desventajas

2.6.1.1. Complejidad

Un DBMS distribuido que oculta la naturaleza distribuida del usuario y proporciona un nivel aceptable de rendimiento, confiabilidad y disponibilidad es inherentemente más complejo que un DBMS centralizado. El hecho de que los datos se puedan replicar también agrega un nivel adicional de complejidad al DBMS distribuido.

Si el software no maneja adecuadamente la replicación de datos, habrá una degradación en la disponibilidad, confiabilidad y rendimiento en comparación con el sistema centralizado, y las ventajas que citamos anteriormente se convertirán en desventajas.

[21] Central Processing Unit (unidad central de proceso)

2.6.1.2. Costo

La mayor complejidad significa que podemos esperar que los costos de adquisición y mantenimiento de un DDBMS sean más altos que los de un DBMS centralizado. Además, una distribución.

DBMS requiere hardware adicional para establecer una red entre sitios. Hay costos continuos de comunicación incurridos con el uso de esta red. También hay costos de mano de obra adicionales para administrar y mantener los DBMS locales y la red subyacente.

2.6.1.3. Control de integridad más difícil

La integridad de la base de datos se refiere a la validez y consistencia de los datos almacenados. La integridad generalmente se expresa en términos de restricciones, que son reglas de consistencia que la base de datos no puede violar. Hacer cumplir las restricciones de integridad generalmente requiere acceso a una gran cantidad de datos que definen las restricciones.

En un DBMS distribuido, los costos de comunicación y procesamiento que se requieren para hacer cumplir las restricciones de integridad son altos en comparación con un sistema centralizado.

2.6.1.4. Falta de estándares

Aunque los DBMS distribuidos dependen de una comunicación efectiva, solo ahora estamos comenzando a ver la aparición de protocolos estándar de comunicación y acceso a datos, esta falta de estándares ha limitado significativamente el potencial de los DBMS distribuidos, tampoco existen herramientas o metodologías para ayudar a los usuarios a convertir un DBMS centralizado en un DBMS distribuido.

2.6.1.5. Falta de experiencia

Los DBMS distribuidos de propósito general no han sido ampliamente aceptados, aunque muchos de los protocolos y problemas se entienden bien. En consecuencia, todavía no tenemos el mismo nivel de experiencia en la industria que tenemos con los DBMS centralizados. Para un posible adoptante de esta tecnología, esto puede ser un impedimento importante.

2.6.1.6. Diseño de base de datos más complejo

Además de las dificultades normales de diseñar una base de datos centralizada, el diseño de una base de datos distribuida debe tener en cuenta la fragmentación de datos, la asignación de fragmentación a sitios específicos y la replicación de datos[22].

[22] (Computer Notes., 2022)

2.6. Bases de datos centralizadas versus Base de datos distribuidas

Las bases de datos son sistemas que se utilizan para almacenar, organizar y recuperar información. Existen dos tipos principales de bases de datos: las centralizadas y las distribuidas.

Bases de datos centralizadas.

Son aquellas que se encuentra en un único lugar y es accedida por todos los usuarios a través de una red. Este tipo de base de datos es más sencilla de configurar y administrar, ya que toda la información se encuentra en un solo lugar. Sin embargo, esto también puede ser una desventaja, ya que todas las operaciones dependen de un único servidor, lo que puede aumentar el riesgo de fallos. Además, una base de datos centralizada puede ser menos escalable que una distribuida, ya que no es tan fácil añadir más servidores para manejar un aumento en la carga de trabajo.

Sin embargo, las bases de datos centralizadas también tienen algunas desventajas. Una de ellas es que son más propensas a fallos debido a que todas las operaciones dependen de un único servidor. Si este servidor falla, toda la base de datos se verá afectada y no podrá ser accedida por los usuarios, debido a que toda la información se encuentra en un único lugar.

Bases de datos distribuidas.

Son aquellas que están divididas en varias partes y se encuentra en múltiples ubicaciones. Este tipo de base de datos es más escalable y resistente a fallos, ya que la carga de trabajo se distribuye entre varios servidores. Además, si un servidor falla, la base de datos aún puede seguir funcionando gracias a los demás servidores.

Sin embargo, las bases de datos distribuidas también tienen algunas desventajas. Una de ellas es que pueden ser más complicadas de configurar y administrar debido a que la información se encuentra en varias ubicaciones y puede requerir más esfuerzo para mantenerla sincronizada. Además, las bases de datos distribuidas pueden ser menos eficientes debido a que las operaciones pueden requerir más tiempo para completarse debido a la necesidad de acceder a varias ubicaciones.

Por lo tanto, las bases de datos centralizadas son más sencillas de configurar y administrar, pero pueden ser menos escalables y resistentes a fallos que las distribuidas. Por otro lado, las bases de datos distribuidas son más escalables y resistentes a fallos, pero pueden ser más complicadas de configurar y administrar, es importante evaluar cuál es el mejor tipo de base de datos para cada situación particular[23].

[23] (Burgess, 2019) (Kumar, 2018)

3. Sistema Clúster

3.1. ¿Qué es un clúster?

Un clúster es un conjunto de computadoras conectadas entre sí llamadas nodos, a este conjunto se le conoce de otra forma como grupo o racimo haciendo referencia a conjuntos o conglomerados de computadoras construidas mediante el uso de hardware en común y que se comportan como si fuera una sola la cual es mucho más poderosa mediante el uso de software conocido como middleware que hace que se vea y administre como un solo sistema.

A diferencia de las computadoras Grid[24], donde cada nodo realiza una tarea diferente, el clúster (nodo maestro) asigna la misma tarea a cada nodo. Cada uno de estos suelen estar conectados entre sí a través de una red local de alta velocidad donde cada nodo ejecuta su propia instancia de un sistema operativo.

Para construir un clúster, los nodos individuales deben estar conectados en una red de alta velocidad a través tarjetas de red, cables de red y uno o más switches (Figura 7), para permitir la comunicación entre ellos, el software que se debe de utilizar para unir a los nodos y formar el clúster, se puede optar por tener un dispositivo de almacenamiento compartido y / o almacenamiento local en cada uno de los nodos.

Normalmente al menos un nodo se designa como nodo líder (master) y actúa como punto de entrada (login) al clúster, el master puede ser responsable de delegar el trabajo entrante a los otros nodos, en otras palabras, es el administrador de tareas a través del middleware.

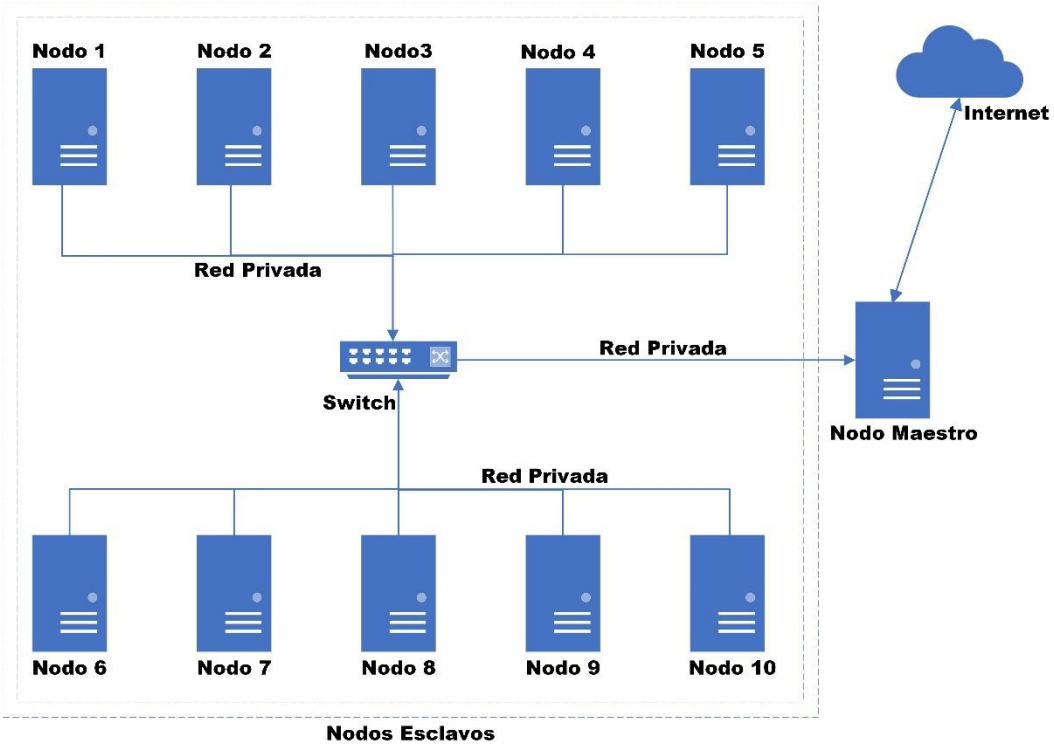


Figura 7 Modelo de un clúster

[24] Grid: Es una técnica de computación distribuida que permite unir recursos de computación de diferentes lugares para trabajar juntos como una sola entidad, con el objetivo de resolver problemas complejos o realizar tareas de gran escala.

Idealmente, un clúster funciona como un solo sistema, el usuario que accede a este no necesita saber si el sistema es un clúster o maquina individual, además este debe diseñarse para minimizar la latencia y evitar cuellos de botella en la comunicación de nodo a nodo.

El uso de este tipo de arquitectura varía debido al tipo de necesidades como las aplicaciones de supercómputo, servidores web, comercio electrónico, bases de datos de alto rendimiento por mencionar algunas.

Los clústers puede variar desde un sistema simple de dos nodos que conecta dos computadoras personales a una supercomputadora con la misma arquitectura específica del grupo. El clúster comúnmente se utiliza para computación de alto rendimiento -High Performance Computing- (HPC[25], por sus siglas en inglés) rentable y de alta disponibilidad -High Availability- (HA[26], por sus siglas en inglés) por empresas, centros de investigación o instituciones de gobierno. En este tipo de hardware si un componente falla (también conocido como tolerancia a fallas) los otros nodos continúan proporcionando el servicio de tal forma que se vea ininterrumpido para el usuario, todo esto es el resultado de la convergencia de varias tendencias tecnológicas actuales, entre las que se pueden destacar:

- Microprocesadores de alto rendimiento.
- Redes de alta velocidad.
- Software para el cómputo distribuido de alto rendimiento.
- Crecientes necesidades de potencia computacional.

La agrupación de este tipo de conjunto de computadoras se basa en un software de administración centralizado el cual verifica que los nodos estén disponibles como servidores orquestados, el sistema operativo adecuado puede evitar el tiempo de inactividad de las aplicaciones, lo que gestiona la réplica de los datos en varios de los nodos proporciona un servicio de computación por error a cualquier distancia con distinta agrupación geográfica[27].

Los servicios que proporciona un clúster principalmente son:

- Alto rendimiento.
- Alta disponibilidad.
- Escalabilidad.
- Balanceo de carga.

[25] High Perfomance Computing
[26] High Avialability
[27] (Open SUSE, 2022) (Capitalone, s.f.) (WatElectronics, s.f.)

3.2. Tipos de Clúster

Principalmente hay dos tipos de clústers:

Clúster asimétrico.

Es una configuración de clúster en la que uno o varios nodos tienen una capacidad de procesamiento y almacenamiento superior a los demás nodos en el clúster. Esta característica permite que algunos nodos actúen como "maestros" y otros actúen como "esclavos" en términos de tareas y responsabilidades.

En un entorno de clúster asimétrico, los nodos maestros suelen ser responsables de tareas de gestión de recursos y coordinación, como el equilibrio de carga, el control de acceso y la asignación de tareas. Por otro lado, los nodos esclavos realizan tareas de procesamiento de datos, como el análisis de datos, el procesamiento de transacciones y la generación de informes (figura 8).

Algunos desafíos a considerar de este tipo de clúster, son la configuración y la administración de los diferentes niveles de capacidad de los nodos. Es importante asegurarse de que los nodos maestros tengan suficiente capacidad para manejar las tareas de gestión y coordinación, mientras que los nodos esclavos tengan suficiente capacidad para manejar las tareas de procesamiento de datos.

Otro desafío es garantizar la seguridad de los datos en un entorno de clúster asimétrico. Es importante asegurar que los datos sensibles solo estén almacenados en nodos seguros y que se implementen medidas de seguridad adecuadas para proteger los datos en tránsito y en reposo.

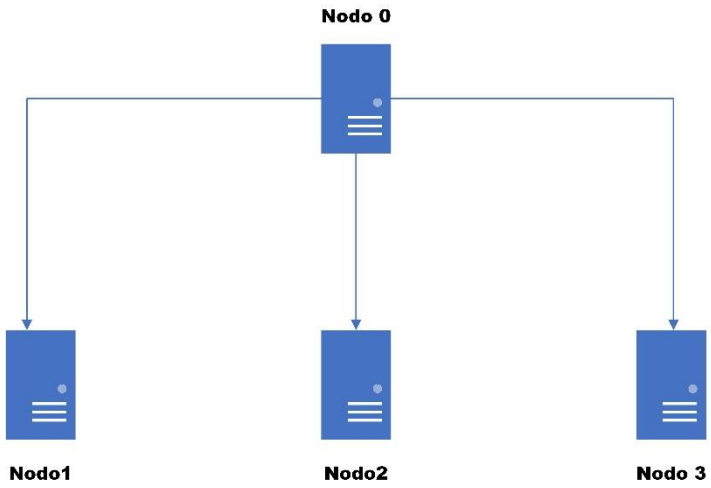


Figura 8 Clúster Asimétrico

Clúster simétrico

En este tipo de clúster, los servidores están organizados en una red de computadoras. La idea es agrupar múltiples servidores en un solo grupo, con el objetivo de mejorar la disponibilidad, el rendimiento y la escalabilidad.

En un clúster simétrico, todos los servidores tienen las mismas capacidades y funciones. Cada servidor puede tomar el control de las tareas de otro servidor si este falla. Esto permite que el sistema continúe funcionando sin interrupciones, ya que los usuarios no notarán la diferencia entre los servidores. Además, los clústers simétricos también permiten una mayor escalabilidad. A medida que se necesita más capacidad, se pueden agregar más servidores al grupo. Esto permite que el sistema crezca de manera eficiente y escalable (figura 9).

Sin embargo, los clústers simétricos también tienen algunos desafíos. Uno de ellos es la configuración y la administración. Es importante asegurarse de que todos los servidores estén configurados de manera consistente y que se realice una buena administración de los servidores para garantizar un buen rendimiento.

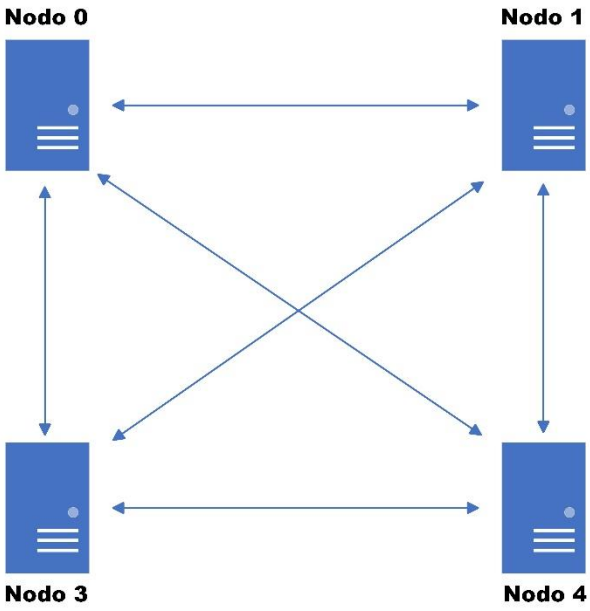


Figura 9 Clúster simétrico

Los tipos de cómputo en clúster son:

Equilibrio de carga del clúster

Los clústeres de equilibrio de carga se emplean en situaciones de utilización de Internet y redes aumentadas; y funcionan como el factor fundamental. Este tipo de agrupación ofrece los beneficios de una mayor capacidad de red y un rendimiento mejorado.

Aquí, todos los nodos permanecen tan cohesivos con toda la instancia donde los objetos de nodo completo están completamente atentos a las solicitudes que están presentes en la red. Todos los nodos no operarán en un solo proceso, mientras que re-dirigen las solicitudes individualmente a medida que llegan dependiendo del algoritmo del programador. El otro elemento crucial en la técnica de equilibrio de carga es la escalabilidad, donde esta característica se logra cuando todos los servidores están totalmente empleados.

Junto con los servidores en equilibrio de carga que tienen una capacidad similar en la reacción del cliente, se crean muchas de las preocupaciones ya que bastantes solicitudes pueden ser atendidas por estos servidores que pueden estar en confusión.

Por lo tanto, el equilibrio de carga crea transparencia para los usuarios, los servidores incluso mantienen la carga estable de solicitudes entrantes o tráfico para sus recursos desde las computadoras conectadas que operan programas similares y que se desarrollan como un clúster. Cuando se produce una falla en un nodo, las solicitudes se asignan nuevamente a los nodos disponibles. Este enfoque se implementa en servidores web Linux.

Clústers de alta disponibilidad

Estos también se denominan clústers de conmutación por error, las computadoras a menudo enfrentan problemas de fallas de diversos tipos. Por lo tanto, la alta disponibilidad está en consonancia con la creciente dependencia de las computadoras ya que estas tienen una responsabilidad crucial en muchas de las organizaciones y aplicaciones.

En este modelo, se utilizan sistemas informáticos redundantes en la situación de mal funcionamiento de cualquier componente, cuando hay un mal funcionamiento de un solo punto, el sistema parece ser completamente confiable ya que la red tiene elementos de clúster redundantes. A través de la implementación de clústers de alta disponibilidad, los sistemas pueden ir con funcionalidad extendida y proporcionan servicios informáticos consistentes como bases de datos complicadas, actividades comerciales, servicios al cliente como sitios web electrónicos y distribución de archivos de red.

Clústers de alto rendimiento

En este modelo se utiliza supercomputadoras para resolver problemas computacionales complejos, junto con la gestión de aplicaciones dependientes de operaciones de Entrada /Salida (IO) como los servicios web, los clústers de alto rendimiento se emplean en modelos computacionales de anomalías climáticas y en diseño, fatiga de materiales y choques de vehículos, se desarrollan clústers de computadoras más estrechamente conectados para trabajos que podrían considerar "supercomputación"[28].

[28] (WatElectronics, s.f.) (geeksforgeeks, 2021)

3.3. Aplicaciones e Implementación

Recapitulando: Un clúster es un grupo de computadoras interconectadas que trabajan juntas para realizar tareas que requieren una gran cantidad de procesamiento de datos, estos sistemas se utilizan a menudo en centros de datos y en entornos empresariales para mejorar la eficiencia y el rendimiento del procesamiento de datos.

Una de las principales aplicaciones de los clústers es la ejecución de aplicaciones de alto rendimiento, como simulaciones científicas y financieras, análisis de datos masivos y procesamiento de imágenes. Muchas de las tareas que realizan los científicos requieren el procesamiento de grandes cantidades de datos, lo que puede ser muy costoso y tardado si se realiza en un solo ordenador. Los clústers permiten dividir el trabajo entre varios nodos, lo que permite acelerar el procesamiento y obtener resultados más rápidamente. Además, los clústers también son útiles en el procesamiento de transacciones en línea y en la ejecución de tareas de simulación y modelado.

Hay varias formas de implementar un clúster, dependiendo del propósito y la escala del sistema. Una opción es utilizar un sistema de clúster de failover, que consiste en un grupo de servidores que se monitorizan entre sí y que pueden tomar el control de las tareas del servidor principal en caso de fallo. Otro enfoque es el balanceo de carga, que distribuye las tareas entre los nodos del clúster para aprovechar al máximo el rendimiento y minimizar el tiempo de inactividad. En cuanto al hardware, es necesario elegir los ordenadores que conformarán el clúster, así como el tipo de interconexión entre ellos, existen diferentes opciones de interconexión, como el uso de redes LAN (Local Area Network) o WAN (Wide Area Network), o la utilización de enrutadores y switches. También es importante tener en cuenta el sistema operativo que se utilizará en cada nodo de este y el software de gestión y distribución de tareas.

Una vez seleccionado y configurado el hardware y el software adecuados, es necesario definir la estructura y el funcionamiento. Esto incluye la asignación de tareas a cada nodo y la definición de mecanismos de fallo y recuperación en caso de averías en alguno de los nodos.

Existen numerosas aplicaciones de computación en clúster algunas de ellas son:

- La computación en clúster se puede implementar en el modelado meteorológico.
- Soporta simulaciones nucleares y averías en vehículos.
- Se utiliza en el procesamiento de imágenes y también en electromagnetismo.
- Perfecto para ser utilizado en aplicaciones de astrofísica, aerodinámica y minería de datos.
- Ayudar a resolver problemas computacionales complejos.
- Tiene la flexibilidad de asignar la carga de trabajo como pequeñas porciones de datos, lo que se denomina computación en cuadrícula.
- La computación en clúster tiene la capacidad de funcionar en muchas aplicaciones web, como seguridad, motores de búsqueda, servidores de bases de datos, servidores web, proxy y correo electrónico.
- Desastres naturales como aplicaciones para salvar vidas a través de este enfoque, ya que pueden pronosticar la ocurrencia de terremotos o tornados[29].

[29] (WatElectronics, s.f.) (Red Had, 2022) (nginx, s.f.)

3.4. Balanceadores

El tráfico en la web ha aumentado constantemente de forma exponencial en la última década, los usuarios de distintas plataformas son cada vez más exigentes en materia de rapidez, acceso y seguridad por lo que los servidores deben hacer frente a una mayor demanda.

Un balanceador (load balancer) es una herramienta de red que distribuye el tráfico entre varios servidores en un clúster, con el objetivo de mejorar el rendimiento y la disponibilidad de un sistema[30]. Esto se logra a través del uso de algoritmos que determinan cómo distribuir la carga entre los servidores, teniendo en cuenta factores como la capacidad de procesamiento y la carga actual de cada servidor (figura 10).

Uno de los beneficios principales de utilizar un balanceador en un clúster es la alta disponibilidad del sistema. Si uno de los servidores falla, este puede reenviar automáticamente el tráfico a los servidores restantes, garantizando que el sistema siga funcionando sin interrupciones. Esto es especialmente importante en sistemas críticos, como plataformas de e-commerce (comercio electrónico) o sistemas de atención al cliente, donde una interrupción del servicio podría tener consecuencias graves.

Otro beneficio de utilizar un balanceador es la mejora del rendimiento del sistema. Si un servidor se encuentra sobrecargado, éste puede redistribuir la carga a otros servidores para equilibrar la carga y asegurar un rendimiento óptimo. Además, al distribuir el tráfico entre varios servidores, se reduce la carga en cada uno de ellos, lo que permite que el sistema procese más solicitudes en menos tiempo.

Existen diferentes algoritmos que pueden utilizarse para determinar cómo distribuir la carga entre los servidores. Algunos de los más comunes son el Round Robin[31], el equilibrado de carga basado en el tiempo de respuesta y el equilibrado de carga basado en la capacidad del servidor. Cada algoritmo tiene sus propias ventajas y desventajas, y es importante elegir el adecuado para las necesidades específicas del sistema. Al distribuir la carga entre varios servidores, se garantiza que el sistema siga funcionando incluso si un servidor falla y se mejora el rendimiento al equilibrar la carga entre los servidores[32].

[30] (Fisher, 2015)
[31] Es un algoritmo de balanceo de carga utilizado en redes y sistemas para distribuir la carga de trabajo entre varios dispositivos o servidores. El objetivo de Round Robin es asegurar que ningún dispositivo o servidor se sobrecargue mientras se distribuye de manera equitativa la carga de trabajo entre todos los dispositivos o servidores disponibles
[32] (nginx, s.f.) (Digital Ocean, 2022) (Foram F Kherani, 2014)

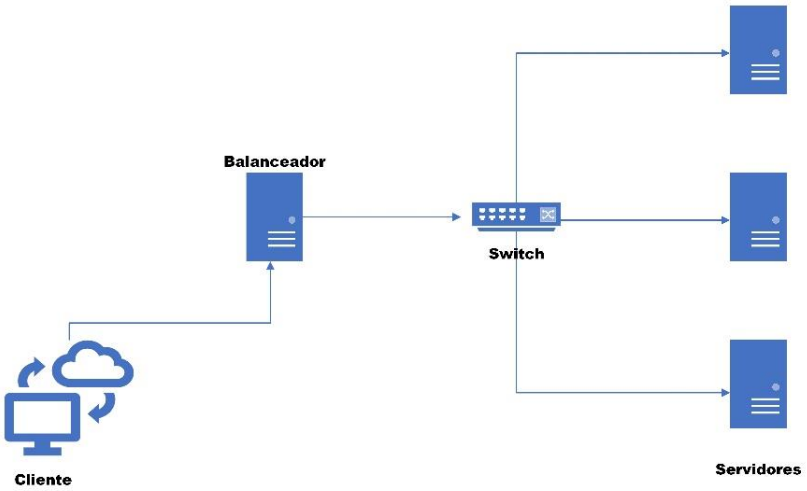


Figura 10 Diagrama de un balanceador.

3.5. Tipos de Balanceadores

Como se comentó anteriormente, un balanceador es una herramienta que se utiliza para distribuir la carga de trabajo en una red de servidores, con el objetivo de mejorar el rendimiento, la escalabilidad y la disponibilidad de un sistema. Existen diferentes tipos cada uno con sus propias características y aplicaciones específicas.

A continuación, se presentan y se describen brevemente tres de los tipos más comunes:

1. **Load balancer de capa 4:** Este tipo distribuye la carga de trabajo basándose en la dirección IP y el puerto de destino de cada paquete de datos. Es decir, se encarga de equilibrar la carga a nivel de red, sin tener en cuenta el contenido de las solicitudes ni las respuestas. Los load balancer de capa 4 son rápidos y sencillos, pero no pueden realizar tareas avanzadas de filtrado ni de optimización de la carga.

2. **Load balancer de capa 7:** Este tipo distribuye la carga de trabajo basándose en el contenido de las solicitudes y las respuestas a nivel de aplicación. Es decir, puede redirigir las solicitudes a un servidor específico en función del idioma o la geolocalización del usuario, o puede filtrar las solicitudes basándose en ciertos patrones o palabras clave. Los load balancer de capa 7 son más flexibles y potentes que los de capa 4, pero también son más complejos y consumen más recursos.

3. **Load balancer global:** Este tipo distribuye la carga de trabajo entre servidores ubicados en diferentes regiones geográficas, con el fin de reducir la latencia y mejorar la disponibilidad de la aplicación. Los load balancer globales suelen utilizar DNS (Domain Name System) para redirigir las solicitudes a un servidor cercano al usuario, o bien utilizan una red privada virtual -Virtual Private Network- (VPN) para conectar servidores en diferentes regiones.

4. **Load balancer inverso:** Este tipo de load balancer distribuye la carga de trabajo entre servidores web o de aplicaciones, con el fin de reducir la sobrecarga en el servidor de origen y mejorar el rendimiento de la aplicación. Los load balancer inverso suelen utilizar técnicas de caché y de optimización de la conexión para acelerar la entrega de contenidos estáticos y dinámicos.

5. **Load balancers por hardware:** Estos utilizan una placa de red dedicada para distribuir el tráfico entre los servidores. Son muy rápidos y resistentes, pero también son costosos y requieren de mantenimiento físico.

6. **Load balancers por software:** Estos se ejecutan en una máquina normal y utilizan el software para distribuir el tráfico. Son más económicos y flexibles que los load balancers por hardware, pero también son menos resistentes y pueden tener un rendimiento ligeramente inferior.

Existen diferentes tipos de load balancer que se pueden utilizar en función de las necesidades y requerimientos de un sistema. Los load balancer de capa 4 son rápidos y eficientes, pero no pueden tomar en cuenta el estado de la aplicación ni realizar reenvíos basados en contenido. Los balanceadores de capa 7 son más flexibles y pueden ofrecer un mayor control sobre el tráfico, pero también son más costosos y menos eficientes. Los load balancer de software son flexibles y versátiles, pero requieren de una mayor configuración y mantenimiento[33].

3.6. Software de Balanceo

Recapitulando un balanceador (load balancer) es una herramienta de software o hardware que distribuye la carga de trabajo entre varios servidores. Esto se puede hacer para aumentar la disponibilidad y el rendimiento de un sistema.

Existen diferentes maneras de implementar un load balancer, y se pueden utilizar diferentes algoritmos para determinar cómo se distribuye la carga de trabajo entre los servidores. Algunos algoritmos comunes son:

Round Robin: En este algoritmo, cada solicitud se envía al siguiente servidor en una lista circular.

Least Connections: En este algoritmo, la carga se envía al servidor con el menor número de conexiones activas.

Weighted Round Robin: Este algoritmo es similar al Round Robin, pero cada servidor tiene un peso asignado que determina cuántas solicitudes se envían a él[34].

Existen varias opciones de software para implementar un balanceador. Algunas opciones populares son:

HAProxy: Es un software de balanceo de carga de red de código abierto muy utilizado en la industria. Desde su lanzamiento en 2000, ha ganado una gran cantidad de usuarios y se ha convertido en una opción viable para una amplia variedad de entornos de redes.

[33] (Foram F Kherani, 2014) (Digital Ocean, 2022) (Load Balancing 101: The Essentials of Load Balancers, s.f.)
[34] (Load Balancing Algorithms and Techniques, s.f.) (Load Balancer Types: What You Need to Know, s.f.)

Una de las principales ventajas es su rendimiento, es capaz de manejar miles de conexiones simultáneas de manera eficiente, lo que lo hace ideal para entornos de alta demanda. Además, es relativamente fácil de configurar y utilizar, lo que lo hace atractivo para desarrolladores y administradores de sistemas, también es versátil y soporta una amplia variedad de protocolos de red, como HTTP (Hypertext Transfer Protocol), HTTPS (Hypertext Transfer Protocol Security), TCP (Transfer Protocol Control) y FTP (File Transfer Protocol). Esto lo convierte en una opción atractiva para entornos donde se necesita más que un simple balanceador de carga. Además, proporciona una amplia variedad de opciones de configuración y de monitorización, lo que lo hace adecuado para una gran cantidad de entornos de redes.

Una de las principales desventajas, es que no proporciona soporte para aplicaciones basadas en Java. Si se quiere utilizar Java con HAProxy, es necesario utilizar un módulo de terceros, como mod_cluster. Aunque esto no es un problema para muchos usuarios, puede ser un impedimento para algunos desarrolladores que buscan una solución completa y fácil de utilizar[35].

NGINX: Es un software de servidor web de código abierto muy popular en la industria de la tecnología de la información. Desde su lanzamiento en 2004, ha ganado una gran cantidad de usuarios y se ha convertido en una opción viable para una amplia variedad de entornos de redes.

Una de las principales ventajas es su rendimiento. Capaz de manejar miles de conexiones simultáneas de manera eficiente, lo que lo hace ideal para entornos de alta demanda. Además, es relativamente fácil de configurar y utilizar, lo que lo hace atractivo para desarrolladores y administradores de sistemas, también es versátil y se puede utilizar como servidor web, balanceador de carga, proxy inverso y como servidor de correo electrónico. Esto lo convierte en una opción atractiva para entornos donde se necesita más que un simple servidor web. Además, proporciona una amplia variedad de opciones de configuración y de monitorización, lo que lo hace adecuado para una gran cantidad de entornos de redes.

Una de las principales desventajas de NGINX es que no proporciona soporte para PHP (Hypertext Preprocesor) nativo. Si se quiere utilizar PHP con NGINX, es necesario utilizar un módulo de terceros, como PHP-FPM (Hypertext Prepocesor FastCGI Process Manager). Aunque esto no es un problema para muchos usuarios, puede ser un impedimento para algunos desarrolladores que buscan una solución completa y fácil de utilizar.

En resumen, NGINX es una opción de servidor web de código abierto muy atractiva gracias a su rendimiento, facilidad de uso y versatilidad. Si bien no proporciona soporte nativo para PHP, es una opción viable para una amplia variedad de entornos de redes[36].

F5 BIG-IP: Es una plataforma de balanceo de carga y de gestión de tráfico de red de alto rendimiento y de alta disponibilidad. Desde su lanzamiento en 1996, ha ganado una gran cantidad de usuarios y se ha convertido en una opción viable para una amplia variedad de entornos de redes.

Una de las principales ventajas es su rendimiento. Es capaz de manejar miles de conexiones simultáneas de manera eficiente, lo que lo hace ideal para entornos de alta demanda. Además, proporciona una amplia variedad de opciones de configuración y de monitorización, lo que lo

[35] (Haproxy, 2022) (Digital Ocean, 2022) (nginx, s.f.) (What Is Server and Application Load Balancing? Types, Configuration Methods, and Best Tools, s.f.)
[36] (nginx, 2022) (Digital Ocean, 2022) (udemy, 2022)

hace adecuado para una gran cantidad de entornos de redes, también se puede utilizar como balanceador de carga, proxy inverso y como servidor de seguridad. Esto lo convierte en una opción atractiva para entornos donde se necesitan más que un simple balanceador de carga. Además, cuenta con integración con herramientas de monitorización como Zabbix y Nagios, lo que facilita la supervisión y el diagnóstico de problemas.

Una de las principales desventajas es su precio. Es una solución de pago y puede ser costosa para algunos usuarios. Además, puede ser un poco compleja de configurar y utilizar, lo que puede hacer que no sea la mejor opción para usuarios con poca experiencia en el área de redes[37].

MaxScale: Es un software de balanceo de carga de base de datos de código abierto desarrollado por MariaDB Corporation. Desde su lanzamiento en 2014, ha ganado una gran cantidad de usuarios y se ha convertido en una opción viable para una amplia variedad de entornos de bases de datos.

Una de las principales ventajas es su rendimiento. Es capaz de manejar miles de conexiones simultáneas de manera eficiente, lo que lo hace ideal para entornos de alta demanda. Además, es relativamente fácil de configurar y utilizar, lo que lo hace atractivo para desarrolladores y administradores de sistemas, puede utilizar como balanceador de carga, proxy inverso y como servidor de seguridad para bases de datos. Esto lo convierte en una opción atractiva para entornos donde se necesitan más que un simple balanceador de carga. Además, cuenta con integración con herramientas de monitorización como Zabbix y Nagios, lo que facilita la supervisión y el diagnóstico de problemas.

Una de las principales desventajas es que solo es compatible con bases de datos basadas en el motor MariaDB. Si se quiere utilizar otro tipo de bases de datos, es necesario utilizar otras soluciones de balanceo de carga. Además, aunque es fácil de configurar y utilizar, puede ser un poco limitado en cuanto a las opciones de configuración avanzadas. Es una opción de balanceador de carga de base de datos de código abierto muy atractiva gracias a su rendimiento, facilidad de uso y versatilidad. Es ideal para entornos de alta demanda y proporciona una amplia variedad de opciones de configuración y de monitorización. Sin embargo, solo es compatible con bases de datos basadas en el motor MariaDB y puede ser un poco limitado en cuanto a las opciones de configuración avanzadas[38].

[37] (F5 BIG-IP Load Balancer, 2022) (F5 BIG-IP: A Comprehensive Guide, 2022) (F5 BIG-IP Load Balancer: Configuration and Best Practices , 2022)
[38] (mariadb, 2022) (Digital Ocean, 2022)

3.7. Arquitectura de bases de datos distribuida

La arquitectura de base de datos distribuidas -Distributed Data Base- (DBD) es una forma de diseñar y organizar bases de datos que están almacenadas en varios servidores y/o ubicaciones geográficas diferentes, pero que aparecen como una sola base de datos al usuario final. Esta tiene como objetivo mejorar la disponibilidad, escalabilidad y rendimiento de la base de datos, ya que permite el acceso simultáneo a los datos por parte de varios usuarios y aplicaciones. Se utiliza a menudo en entornos empresariales y de investigación científica donde se requiere un alto rendimiento y escalabilidad para procesar y analizar grandes cantidades de datos.

Una base de datos distribuida se compone de varios nodos, cada uno de los cuales puede tener su propio almacenamiento y procesamiento de datos, estos nodos se comunican entre sí y trabajan juntos para procesar y almacenar los datos de la base de datos. Al utilizar múltiples nodos, una base de datos distribuida puede proporcionar un rendimiento y capacidad de almacenamiento mucho mayores que una base de datos tradicional en un único servidor.

Una de las principales ventajas de las bases de datos distribuidas es su capacidad para gestionar grandes cantidades de datos. Al utilizar múltiples nodos, una base de datos distribuida puede almacenar y procesar grandes cantidades de datos de manera eficiente. Además, al distribuir los datos entre varios nodos, se puede reducir la carga en cualquier nodo individual, lo que mejora el rendimiento y la escalabilidad del sistema.

Otra ventaja de las bases de datos distribuidas es su alta disponibilidad. Al utilizar múltiples nodos, una base de datos distribuida puede continuar funcionando incluso si un nodo individual falla. Esto es especialmente importante en entornos empresariales donde se requiere un alto nivel de disponibilidad para garantizar la continuidad del negocio.

Otro desafío es el rendimiento y la escalabilidad. A medida que aumenta el número de servidores y ubicaciones en la arquitectura de base de datos distribuidas, puede resultar más difícil gestionar y optimizar el rendimiento de la base de datos. Además, es necesario contar con una infraestructura adecuada y una buena planificación para asegurar la escalabilidad y la disponibilidad de la base de datos.

La arquitectura de base de datos distribuidas es una forma eficaz de mejorar la disponibilidad, escalabilidad y rendimiento de la base de datos. Sin embargo, su implementación conlleva ciertos desafíos, como la gestión y sincronización de datos en diferentes servidores[39].

[39] (Molina, 2009) (Valduriez, Arquitecturas de bases de datos distribuidas, 2011) (Valduriez, Arquitecturas de bases de datos distribuidas: conceptos y diseño, 2005)

3.8. Algoritmos de replicación

La replicación de datos es un método común que se utiliza para mejorar el rendimiento del acceso a los datos en sistemas distribuidos.

La replicación significa mantener una copia de los mismos datos en varias máquinas que están conectadas a través de una red, si los datos que está replicando no cambian con el tiempo, la replicación es fácil: simplemente copie los datos en cada nodo. Toda la dificultad de la replicación radica en el manejo de cambios en los datos replicados.

3.8.1. Líderes y seguidores.

Cada escritura en la base de datos debe ser procesada por cada réplica, de lo contrario, las réplicas no contendrán los mismos datos, una solución más común para esto se llama replicación basada en líderes (también conocida como replicación maestro-esclavo).

Todas las escrituras u otras operaciones que cambian los datos se envían a la réplica designada como líder (maestro), después de realizar los cambios localmente, el líder notifica a los seguidores sobre los cambios en los datos, en tanto los seguidores (esclavos), en respuesta, actualizan su propia copia local de los datos para mantenerse al día con el líder, se puede leer desde cualquier réplica, pero solo el líder aceptará escrituras.

3.8.2. Replicación síncrona.

En este método de replicación, una vez que el maestro recibe una solicitud de escritura de la aplicación primero, escribe datos en la base de datos y luego envía estos cambios a todas las réplicas. Posteriormente, el maestro espera hasta que todas las réplicas escriben los cambios en su base de datos y los reconocen y finalmente el maestro responde con un mensaje de éxito a la solicitud de escritura de la aplicación.

3.8.3. Replicación asíncrona.

En el método de replicación asincrónica, el maestro envía la confirmación a la aplicación tan pronto como recibe el mensaje y se escribe correctamente en la base de datos, luego envía la solicitud de replicación a todas las réplicas.

La replicación tiene tres algoritmos de mayor uso:

1. Replicación de un solo líder (Single-leader replication).

En esta configuración, los clientes siempre envían escrituras (en el caso de bases de datos, consultas INSERT, UPDATE y DELETE) al maestro y nunca a un esclavo, sin embargo, estos pueden responder consultas de lectura.

El principal beneficio de tener un solo líder(maestro) es que evitamos los conflictos causados por escrituras simultáneas, todos los clientes escriben en el mismo servidor, por lo que la coordinación es más sencilla.

El problema es que debemos asegurarnos de que solo un nodo (esclavo) pueda manejar todas las escrituras, aunque podemos dividir el trabajo de lectura en todo el clúster, todas las escrituras van a un solo servidor, y si su aplicación es muy intensiva en escritura, eso podría ser un problema, sin embargo, debemos tomar en cuenta que la mayoría de las aplicaciones leen muchos más datos de los que escriben, por lo que debe de analizar si eso es realmente un problema.

2. Replicación de varios líderes (Multi-leader replication).

La razón principal para considerar un enfoque de múltiples maestros es que resuelve algunos de los problemas que enfrentamos cuando solo tenemos un nodo líder. Es decir, tenemos más de un nodo que maneja escrituras, y estas escrituras pueden ser realizadas por bases de datos más cercanas a los clientes, si la aplicación necesita manejar una gran cantidad de escrituras, podría tener sentido dividir ese trabajo entre varios líderes, además el precio de latencia para escribir en una base de datos que está muy lejos es demasiado alto, por lo que se podría tener un líder en cada ubicación (por ejemplo, uno en Norteamérica, uno en Europa y otro en Asia).

Otro caso práctico es cuando necesita admitir clientes fuera de línea, que podrían estar escribiendo en su propia base de datos (líder), y estas escrituras deben sincronizarse con el resto de las bases de datos una vez que este cliente se vuelva a conectar.

El principal problema al que se enfrentará cuando varios líderes acepten escribir por lo que se necesita alguna forma de resolver los conflictos. Por ejemplo, supongamos que se tiene una restricción de base de datos para garantizar que los correos electrónicos de usuarios sean únicos, si dos clientes escriben a dos líderes diferentes que aún no están sincronizados, ambas escrituras tendrán éxito en sus respectivos líderes, pero tendremos problemas cuando intentemos replicar esos datos.

3. Replicación sin líderes (Leaderless replication).

La idea básica es que los clientes enviarán escrituras no solo a una réplica, sino a varias (o, en algunos casos, a todas). La ventaja que se tiene aquí es que podemos tolerar las fallas de los nodos más fácilmente. Por ejemplo, lo que sucedería en un escenario en el que tuviéramos que enviar un escrito a un solo líder y, por alguna razón, ese líder no respondió, la escritura fallaría y tendríamos que iniciar un proceso de conmutación por error para elegir un nuevo líder que pudiera comenzar a recibir escrituras nuevamente.

¿Qué sucede si la escritura tiene éxito en dos réplicas, pero falla en una? Ahora tiene dos réplicas con el nuevo valor y una con el valor anterior. Recordemos que estas réplicas no se comunican entre sí, no hay ningún líder manejando ningún tipo de sincronización.

Sin embargo, todavía tenemos un problema, una de las réplicas todavía tiene el valor anterior y necesitamos sincronizarla de alguna manera con el resto de las réplicas (después de todo, la réplica es el proceso de mantener los mismos datos en varios lugares).

Por lo general, hay dos formas de hacerlo: podemos hacer que el cliente sea responsable de esta actualización, o podemos tener otro proceso que se encargue solo de encontrar diferencias en los datos y corregirlas.

Hacer que el cliente lo arregle es conceptualmente simple, cuando el cliente lee datos de varios nodos y detecta que uno de ellos está obsoleto, envía una solicitud de escritura con el valor correcto. A esto se le suele llamar reparación de lectura[40].

[40] (Pasha Kostohrys, 2020) (Viswanath, 2020)

4. Implementación del Clúster

4.1. Construcción del clúster

En resumen: un clúster de base de datos es un conjunto de servidores que trabajan juntos para proporcionar servicios de base de datos a las aplicaciones de una organización. Los servidores en el clúster se dividen en dos categorías: nodos maestros y nodos esclavos. Los nodos maestros son responsables de la gestión de la base de datos y de la toma de decisiones sobre cómo distribuir el trabajo entre los nodos esclavos. Los nodos esclavos son responsables de realizar el trabajo de procesamiento de datos y de almacenamiento de datos.

Dadas las múltiples ventajas y la problemática mencionada al principio de este trabajo de tesis que es solventar la carga de las plataformas asi como creciente demanda de los usuarios y analizando las distintas soluciones se decide implementar un clúster de base de datos el cual dispone de alta disponibilidad y atención para la creciente demanda dentro de las plataformas.

4.1.1. Requerimientos

Los requerimientos en ingeniería de software son una lista de las necesidades y expectativas del usuario o cliente con respecto a un sistema o aplicación de software. Los requerimientos deben ser especificados de manera clara y concisa para garantizar que el software cumpla con las expectativas del usuario y resuelva los problemas para los que fue diseñado. La identificación y documentación de requerimientos es una parte importante del proceso de ingeniería de software, ya que sirve como base para el diseño y desarrollo del software.

Existen diferentes tipos de requerimientos en ingeniería de software, que se clasifican según su naturaleza y propósito:

- Requerimientos funcionales: son aquellos que especifican las tareas o funciones que el software debe realizar.
- Requerimientos no funcionales: son aquellos que especifican cómo debe ser el software, pero no lo que debe hacer. Estos incluyen requisitos de rendimiento, seguridad, usabilidad.
- Requerimientos de usuario: son aquellos que especifican las necesidades y expectativas del usuario final con respecto al software.
- Requerimientos de sistema: son aquellos que especifican las necesidades y expectativas del sistema en el que el software se ejecutará.
- Requerimientos de negocio: son aquellos que especifican las necesidades y expectativas del negocio o empresa que utilizará el software.

Para la construcción del clúster se tienen los siguientes requerimientos:

4.1.1.1. Requerimientos de sistema

Los Requerimientos de Sistema (RS) para la implementación del clúster son los siguientes:

RS 1: Máquinas servidoras. Cada nodo del clúster necesitará una máquina servidora con suficiente potencia de procesamiento, memoria y almacenamiento para soportar las cargas de trabajo del clúster. Las máquinas pueden ser virtuales o físicas.

RS 2: Sistema operativo compatible. Todas las máquinas servidoras del clúster deben tener un sistema operativo compatible entre sí para garantizar la comunicación y colaboración adecuadas.

RS 3: Red de alta velocidad. El clúster necesitará una red de alta velocidad y de baja latencia para garantizar la comunicación rápida y eficiente entre los nodos.

RS 4: Software de clúster: Se necesitará un software de clúster para gestionar y coordinar la ejecución de aplicaciones en el clúster, el cual tiene que ser libre o de código abierto.

RS 5: Soporte técnico. Es importante contar con personal técnico cualificado para configurar, administrar y mantener el clúster.

RS 6: Documentación. Se deberá mantener una documentación adecuada del clúster, incluyendo su configuración, topología y procedimientos de mantenimiento.

RS 7: Conmutadores de red de alta calidad. Se necesitarán conmutadores de red de alta calidad para conectar los nodos del clúster y permitir la comunicación entre ellos.

RS 8: Alimentación eléctrica redundante. Es importante contar con una alimentación eléctrica redundante para garantizar la disponibilidad del clúster en caso de fallos en la energía.

RS 9: Enfriadores redundantes. Es importante contar con enfriadores redundantes para garantizar la estabilidad térmica del clúster y evitar fallos debidos al sobrecalentamiento.

RFS 10: Espacio de almacenamiento compartido. Se necesitará un espacio de almacenamiento compartido para almacenar los datos y aplicaciones que se ejecutan en el clúster.

4.1.1.2. Requerimientos de negocio

Los Requerimientos de Negocio (RN) para un clúster son las necesidades y expectativas del negocio o empresa que utilizará el clúster en cuanto a su funcionamiento y rendimiento.

Para la implementación de este tenemos los siguientes requerimientos:

RN 1: Disponibilidad. El clúster debe estar disponible en todo momento para garantizar la continuidad del servicio.

RN 2: Escalabilidad. El clúster debe poder escalar para soportar el crecimiento del negocio u organización y la demanda de recursos.

RN 3: Seguridad. El clúster debe ser seguro y proteger los datos del negocio de posibles ataques o accesos no autorizados.

RN 4: Integración. El clúster debe integrarse adecuadamente con otros sistemas y aplicaciones.

RN 5: Rentabilidad. El clúster debe ser rentable y ofrecer un buen rendimiento en relación con su coste.

RN 6: Flexibilidad. El clúster debe ser flexible y adaptarse a los cambios en el negocio y en la demanda de recursos.

RN 7: Soporte. El clúster debe contar con un buen soporte técnico y servicios de mantenimiento para garantizar su correcto funcionamiento.

RN 8: Cumplimiento normativo: el clúster debe cumplir con todas las normativas y regulaciones aplicables al negocio u organización, es decir para este proyecto debe satisfacer la alta demanda de los usuarios dentro de los sistemas (Moodle).

RN 9: Gestión del cambio. El clúster debe ser capaz de adaptarse a los cambios en el negocio y en la demanda de recursos de manera eficiente y sin interrupciones.

RN 10: Gestión de la calidad. El clúster debe cumplir con los estándares de calidad aplicables al negocio.

4.2. Software de Clustering para bases de datos

El software de clustering de base de datos es una herramienta informática que se utiliza para agrupar conjuntos de datos en categorías similares, conocidas como "clústers". Esta técnica de análisis de datos es muy útil en diversas aplicaciones, como la minería de datos, el análisis de tendencias y la segmentación del mercado. Permitiendo a los usuarios agrupar grandes conjuntos de datos en grupos más pequeños y manejables, lo que facilita la visualización y el análisis de los datos.

El proceso de clustering de base de datos se basa en el uso de algoritmos matemáticos que comparan los datos entre sí y los agrupan en clústers según sus similitudes. Estos algoritmos pueden ser de dos tipos: supervisados o no supervisados. Los algoritmos supervisados requieren que se proporcione un conjunto de datos de entrenamiento con etiquetas de clase, mientras que los algoritmos no supervisados no requieren etiquetas y se basan en la propia estructura de los datos para agruparlos.

Existen muchos tipos diferentes de software de clustering de bases de datos, y cada uno tiene sus propias ventajas y desventajas. Uno de los tipos más populares es el algoritmo de clustering jerárquico, que utiliza un enfoque de "divide y conquistar" para dividir los datos en grupos cada vez más pequeños. Otra opción común es el algoritmo de clustering basado en centroides, que utiliza puntos centrales para agrupar los datos.

Este software también puede ser utilizado para detectar patrones y tendencias en los datos. Por ejemplo, si una empresa quiere saber qué productos son más populares entre sus clientes, puede utilizar el software de clustering para agrupar a sus clientes en función de sus compras y encontrar patrones en sus compras. Esto puede ayudar a la empresa a entender mejor a sus clientes y a tomar decisiones de marketing más informadas.

Además, el software de clustering de bases de datos puede ser utilizado para mejorar la eficiencia de la base de datos. Es decir, si una base de datos tiene un gran número de registros, puede ser difícil para los usuarios acceder a la información que necesitan de manera rápida y eficiente. El software de clustering puede agrupar los registros de manera que los usuarios puedan acceder a ellos de manera más rápida y fácil.

Existen varios tipos de software de clúster de base de datos, algunos de los cuales son:

Oracle Real Application Clústers (RAC): Es una opción de Oracle Database que permite la ejecución de una base de datos Oracle en múltiples nodos de un clúster compartiendo un almacenamiento común. Esto proporciona alta disponibilidad y escalabilidad para la base de datos. Cuando se utiliza RAC, varias instancias se ejecutan en diferentes nodos del clúster, y cada instancia tiene acceso a la base de datos completa. Cada instancia puede realizar operaciones de lectura y escritura en la base de datos, y el almacenamiento compartido se sincroniza entre las instancias para mantener la integridad de los datos.

RAC utiliza un conjunto de componentes y tecnologías para permitir el acceso a la base de datos desde múltiples nodos, incluyendo el uso de Oracle Clusterware y Oracle Global Cache Service. La capa de almacenamiento de datos utiliza una base de datos Oracle tradicional y proporciona alta disponibilidad mediante la replicación en tiempo real de los datos. La capa de procesamiento de transacciones maneja las transacciones y asegura la integridad de los

datos. La capa de API proporciona una interfaz para que las aplicaciones accedan a los datos[41].

Microsoft SQL Server Always On: Es una opción de alta disponibilidad y recuperación ante desastres High Availability/Disaster Recovery (HA/DR) de Microsoft SQL Server. Proporciona una alta disponibilidad de bases de datos a través de la configuración de varias réplicas de bases de datos que pueden ser utilizadas en caso de fallo de una base de datos principal.

Algunas de las características clave de Microsoft SQL Server Always On incluyen:

- Réplicas de lectura: las réplicas de lectura permiten la realización de copias de seguridad y la ejecución de consultas en una réplica de base de datos sin afectar al rendimiento de la base de datos principal.

- Failover automático: en caso de fallo de la base de datos principal, Microsoft SQL Server Always On puede realizar un failover automático a una réplica de base de datos disponible, minimizando el tiempo de inactividad.

- Soporte de varios centros de datos: Microsoft SQL Server Always On permite la configuración de réplicas de bases de datos en varios centros de datos, lo que proporciona una mayor protección ante desastres.[42]

MySQL Clúster: Es una solución de base de datos distribuida que utiliza la tecnología de clustering para proporcionar alta disponibilidad y rendimiento en tiempo real. Está diseñada para manejar grandes cantidades de transacciones en línea y consultas de datos, y puede ser utilizada en aplicaciones empresariales críticas para el negocio.

Utiliza una arquitectura de n-capas, que incluye una capa de datos, una capa de transacciones y una capa de API. La capa de datos almacena los datos en memoria y utiliza replicación en tiempo real para asegurar la alta disponibilidad. La capa de transacciones maneja las transacciones y asegura la integridad de los datos. La capa de API proporciona una interfaz para que las aplicaciones accedan a los datos.

MySQL Clúster se ha utilizado en una variedad de aplicaciones, incluyendo plataformas de comercio electrónico, sistemas de gestión de relaciones con clientes y sistemas de gestión de contenidos. Ha demostrado ser una opción confiable y de rendimiento para bases de datos de alta concurrencia y alto tráfico[43].

[41] (Oracle, 2022) (Oracle, 2022)
[42] (Microsoft , 2022) (Microsoft, 2022)
[43] (Mysql, 2022) (Mysql, 2022)

IBM DB2 pureScale: Es una opción de base de datos de IBM que ofrece alta disponibilidad y escalabilidad para aplicaciones empresariales. Utiliza un enfoque basado en clúster para permitir la expansión y el balanceo de carga en entornos de alta concurrencia. Esto permite que las aplicaciones continúen funcionando sin interrupciones incluso en caso de fallos en el sistema.

Algunas características clave de IBM DB2 pureScale incluyen:

- Soporte para transacciones de base de datos en tiempo real y consultas analíticas simultáneas.
- Capacidad de añadir y quitar nodos de forma dinámica sin interrupciones en el servicio.
- Soporte para la replicación asíncrona y sincrónica de datos entre nodos del clúster.
- Integración con IBM WebSphere Application Server y IBM Tivoli System Automation.

IBM DB2 pureScale se utiliza principalmente en entornos empresariales para proporcionar alta disponibilidad y escalabilidad en aplicaciones críticas para el negocio. Se ha utilizado en aplicaciones de comercio electrónico, banca, seguros y otras industrias empresariales[44].

PostgreSQL Streaming Replication: Es una característica de la base de datos PostgreSQL que permite la replicación en tiempo real de datos de una base de datos principal a una o más bases de datos secundarias. Esto significa que cualquier cambio realizado en la base de datos principal se refleja de manera automática en las bases de datos secundarias, lo que permite tener una copia de respaldo en tiempo real de los datos y aumentar la disponibilidad y la tolerancia a fallos.

La replicación se realiza mediante el envío de registros de transacciones desde la base de datos principal a las bases de datos secundarias mediante un flujo de datos en tiempo real. Esto permite que la replicación sea rápida y eficiente, ya que solo se transmiten los cambios realizados en la base de datos en lugar de copiar todos los datos cada vez.

La replicación streaming se puede utilizar en diferentes configuraciones, como la replicación maestro-esclavo, en la que la base de datos principal actúa como el maestro y las bases de datos secundarias actúan como esclavas, o en configuraciones más complejas con múltiples maestros y esclavos[45].

[44] (IBM, 2022) (IBM, s.f.)
[45] (Postgresql, 2022) (severalnines, 2022)

MariaDB Galera Clúster: Es una arquitectura de clustering para bases de datos MariaDB. Es una solución de alta disponibilidad y tolerante a fallos diseñada para garantizar que los datos estén siempre disponibles y sean consistentes en todos los nodos del clúster, utiliza un algoritmo de replicación síncrona para mantener la consistencia de los datos en todos los nodos del clúster. Esto significa que cada transacción debe ser confirmada en todos los nodos del clúster antes de que se considere completa. Si una transacción no puede ser confirmada en todos los nodos, se cancela para evitar la pérdida de datos.

Es una opción popular para la creación de sistemas de bases de datos altamente disponibles y tolerantes a fallos.

Algunas de las ventajas de MariaDB Galera Clúster incluyen:

- Alta disponibilidad: los nodos del clúster pueden fallar o ser reiniciados sin interrumpir el servicio de bases de datos.

- Alta escalabilidad: se pueden agregar más nodos al clúster para aumentar el rendimiento y la capacidad.

- Fácil de usar: MariaDB Galera Clúster se integra con el servidor de bases de datos MariaDB y se puede utilizar con herramientas y aplicaciones existentes[46].

4.3. Por qué usar MariaDB Galera Clúster.

Recapitulando como ya hemos visto la problemática sobre la alta demanda de los usuarios en la plataforma Moodle. MariaDB es un sistema de gestión de bases de datos relacional (RDBMS) que es muy similar a MySQL, que es otro RDBMS muy popular.

MariaDB Galera es una versión del sistema de gestión de base de datos MariaDB que utiliza una configuración de replicación multi-maestro sincrónica. Esto significa que permite configurar un grupo de servidores MariaDB que pueden atender consultas de lectura y escritura, con los cambios realizados en un servidor que se replican automáticamente en los demás. Esto puede proporcionar mejoras en el rendimiento y la tolerancia a fallos, ya que puede distribuir la carga de las consultas de base de datos en varios servidores y seguir atendiendo consultas incluso si uno o más servidores fallan. Está basado en la biblioteca de replicación Galera, que está diseñada para manejar escenarios complejos de replicación multi-maestro con altos niveles de rendimiento y fiabilidad.

La replicación multi-maestro es una técnica de base de datos que permite a varios servidores actúen como maestros, es decir, que puedan realizar cambios en la base de datos y replicarlos a los demás servidores. Esto se diferencia de la replicación maestro-esclavo, en la que solo un servidor maestro puede realizar cambios y los demás servidores esclavos solo pueden replicarlos. La replicación multi-maestro puede proporcionar ventajas en términos de rendimiento y escalabilidad, ya que permite distribuir la carga de trabajo entre varios servidores y puede permitir la realización de cambios más rápidamente al no depender de un único servidor maestro.

[46] (MariDB, 2022) (MariaDB, 2022)

MariaDB Galera utiliza el algoritmo de replicación Galera para sincronizar los cambios entre los servidores del clúster. Este algoritmo utiliza un protocolo de confirmación de escritura para asegurar que todos los servidores del clúster reciban y apliquen los cambios de manera consistente. Además, Galera utiliza un mecanismo de certification-based non-blocking conflict detection (CBCD por sus siglas en ingles) para detectar y resolver conflictos entre los cambios realizados en los servidores del clúster. Esto permite que MariaDB Galera maneje de manera eficiente la replicación multi-maestro en entornos con alta concurrencia y cambios simultáneos.

Las razones por la que se elige usar MariaDB Galera en este trabajo son las siguientes:

1. **Altamente disponible:** Es un motor de replicación de base de datos que permite que varios servidores funcionen como un clúster y proporciona alta disponibilidad y tolerancia a fallos. Esto puede ser útil para Moodle, que debe estar disponible para los estudiantes y profesores en todo momento.

2. **Escalabilidad:** Permite que el clúster se expanda fácilmente a medida que aumenta la carga de trabajo, lo que puede ser útil para Moodle, que puede tener grandes cantidades de usuarios concurrentes en un sitio dado.

3. **Desempeño:** Puede mejorar el rendimiento de la base de datos al distribuir la carga de trabajo entre varios servidores y permitir que las operaciones se ejecuten en paralelo.

4. **Facilidad de uso:** Fácil de instalar y configurar, y ofrece una interfaz de usuario intuitiva para la administración del clúster.

5. **Compatibilidad:** Compatible con MySQL, por lo que es fácilmente intercambiable con él. Esto significa que si ya tienes una instalación de Moodle que utiliza MySQL, puedes cambiar a Galera sin tener que hacer cambios significativos en la configuración de tu sitio Moodle.

6. **Seguridad:** Incluye características de seguridad integradas, como cifrado de red y autenticación de usuario, para proteger la base de datos y los datos de los usuarios.

7. **Características adicionales:** Incluye algunas características adicionales que no están presentes en MySQL, como soporte para grupos de servidores y la capacidad de realizar consultas de JSON.

8. **Comunidad activa:** Cuenta con una comunidad activa de usuarios y desarrolladores que pueden proporcionar soporte y recursos útiles para la implementación y el mantenimiento de un clúster de base de datos.

9. **Licencia de código abierto:** Está disponible bajo la Licencia Pública General de GNU (GNU -GNU's is Not Linux- GPL -General Public License-), lo que significa que es gratis y se puede utilizar y modificar libremente.

10. **Soporte:** Respaldado por MariaDB Corporation, una empresa de software de base de datos de código abierto que proporciona soporte profesional y servicios de asistencia para los usuarios de Galera.

4.4. Instalación y configuración del MariaDB Galera Clúster

Para proceder a hacer la instalación se debe tener previamente los siguientes requisitos de hardware.

- Almacenamiento: 130 GB
- CPU'S: 4
- RAM: 8 GB

Requerimientos de software:

- CentOS 7
- Galera Clúster 10.3

La instalación del software de galera se instalará en los siguientes servidores:

- BD 1, IP: XXX.XXX.XXX.201 (nodo 1)
- BD 2, IP: XXX.XXX.XXX.202 (nodo 2)
- BD 3, IP: XXX.XXX.XXX.203 (nodo 3)

Paso 1: Actualizamos el sistema operativo con la siguiente instrucción (figura 11).

yum -y upgrade.



Figura 11 Actualización sistema operativo

Paso 2: Instalar el dnf con la siguiente instrucción (figura 12): ***yum -y install dnf***

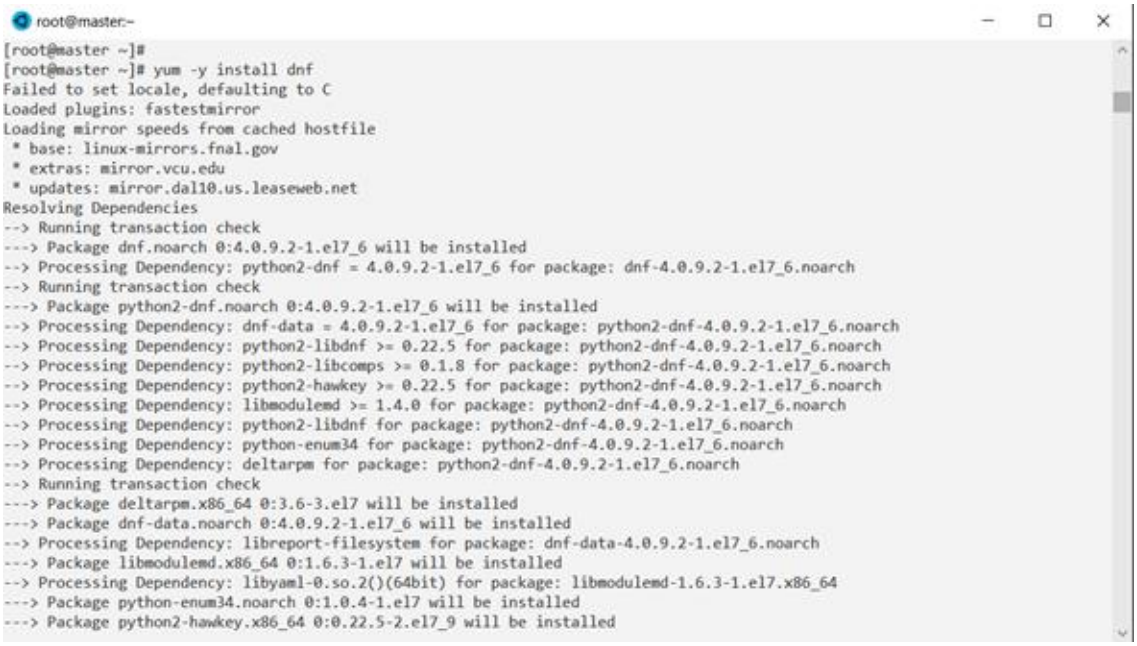


Figura 12 Instalación DNF

Paso 3: Instar el paquete **net-tools** para restablecer la instrucción **ifconfig** de vuelta, ya que no viene instalado por defecto en CentOS (figura 13), con la siguiente instrucción.

dnf install -y net-tools mlocate

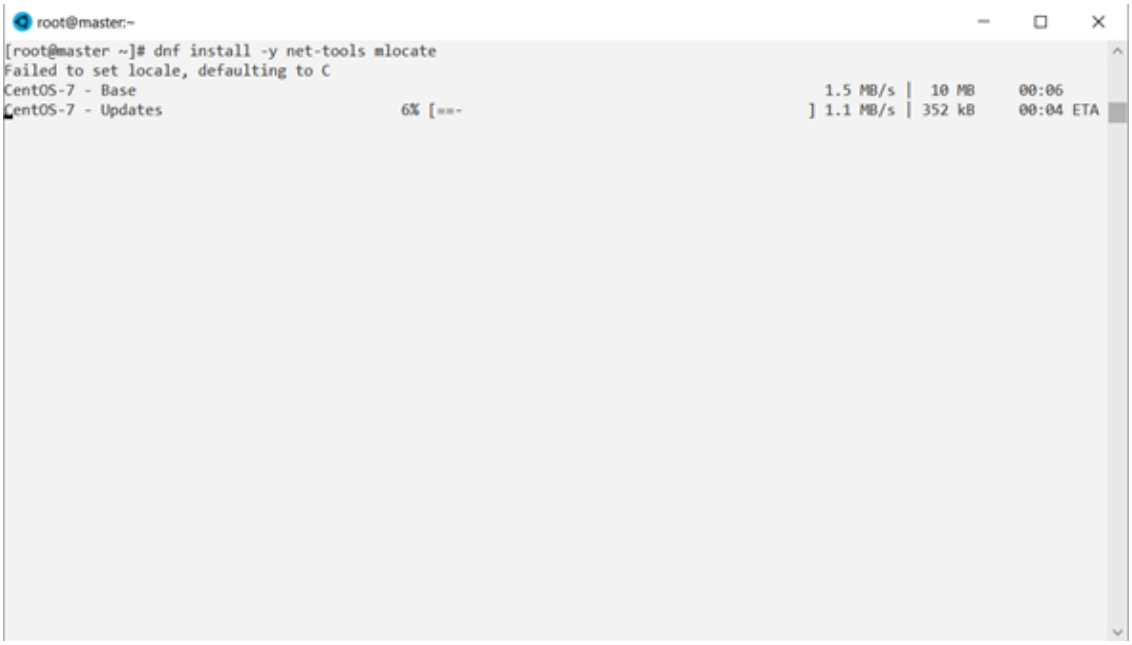


Figura 13 Instalación net-tools

Paso 4: Se instalan varias utilidades **Midnight Commander**, **nmon**, **htop, nano, nmap** y otras (figura 14), con la siguiente instrucción.

dnf -y install mc nano nmap unzip

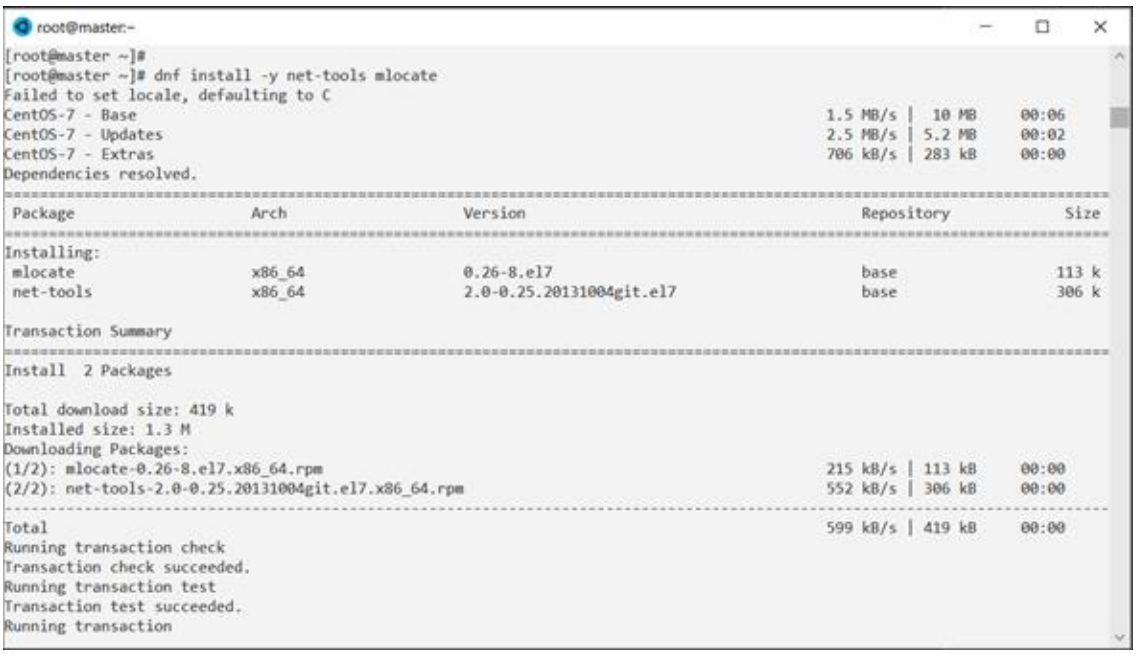


Figura 14 Instalación de utilidades (nmon, htop, nano y nmap)

Paso 5: Para instalar el paquete **nmon** se requieren los repositorios **epel** de CentOS (figura 15).

dnf -y install epel-release
dnf -y install nmon htop

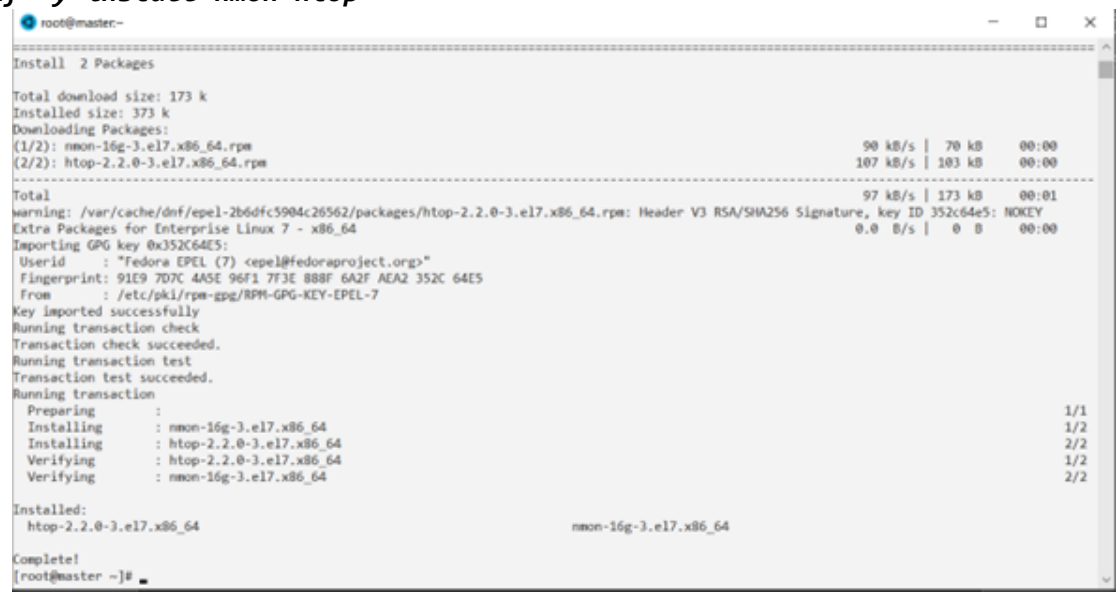


Figura 15 Instalación repositorios epel

Paso 6: Para hacer que funcione la réplica de la base de datos en cada uno de los nodos que se tengan instalados es necesario instalar **rsync (**figura 16).

dnf -y install rsync

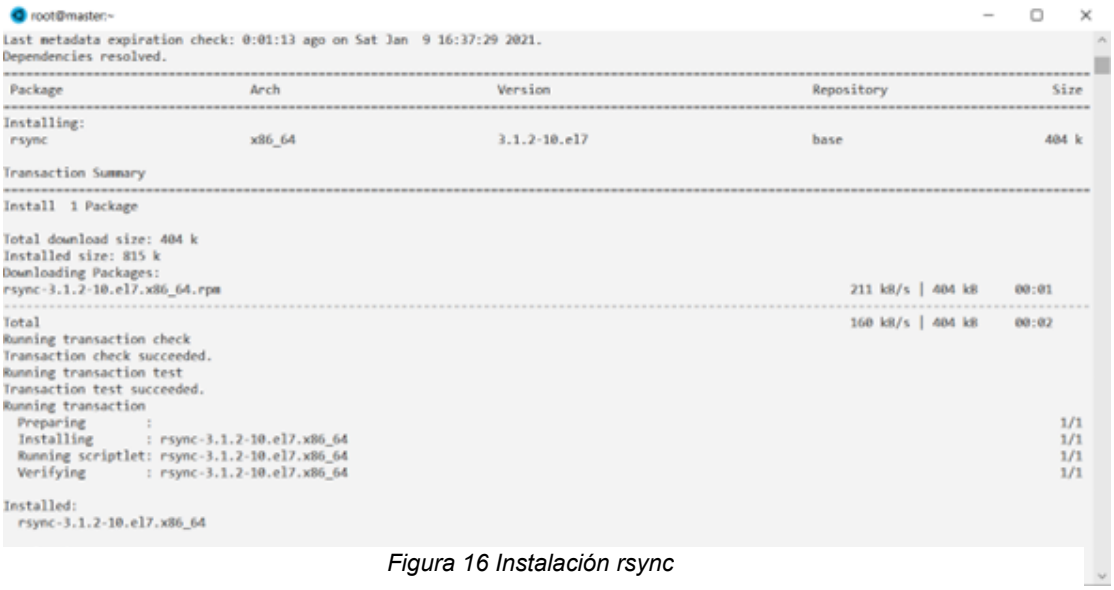


Figura 16 Instalación rsync

Paso 7: Se deben configurar las políticas de SELinux y en cada nodo instalar ***policycoreutils*** (figura 17).

dnf -y install policycoreutils-python

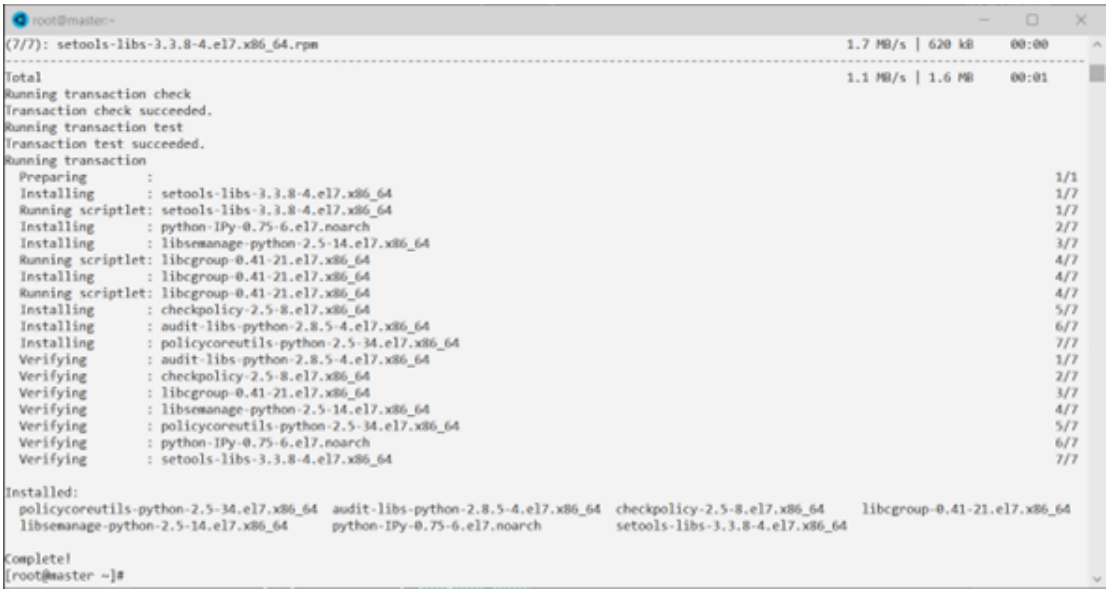


Figura 17 Configuración SELinux.

Posteriormente se instalan algunos paquetes que serán de utilidad (figura 18).

```
dnf -y install dnf-plugins-core
dnf -y copr enable konimex/neofetch
```

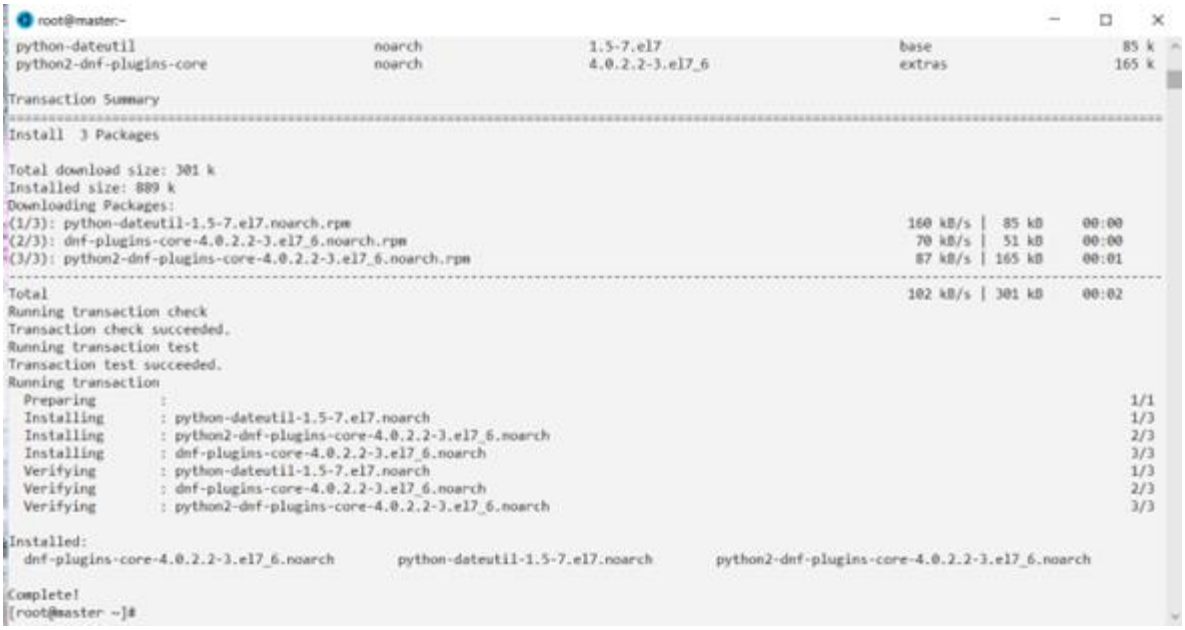


Figura 18 Instalación de paquetes adicionales

Paso 8: Instalación de MariaDB.

Antes de instalar se debe de dar de alta el repositorio el cual cuenta con la versión de MariaDB a instalar, mediante la siguiente instrucción, se crea el archivo (figura 19).

```
vi /etc/yum.repos.d/mariadb.repo
```

En modo de edición ingresamos las siguientes líneas.

```
[mariadb]
name = MariaDB
baseurl = http://yum.mariadb.org/10.3/centos73-amd64/
gpgkey=https://yum.mariadb.org/RPM-GPG-KEY-MariaDB
gpgcheck=1
```



Figura 19 Configuración repositorio MariaDB

Posteriormente se actualizan los repositorios

yum -y update

Ahora se instalará la última versión estable de MariaDB 10.4, junto a las bibliotecas de Galera versión 4 (figura 20).

yum -y install MariaDB-server MariaDB-client galera-4

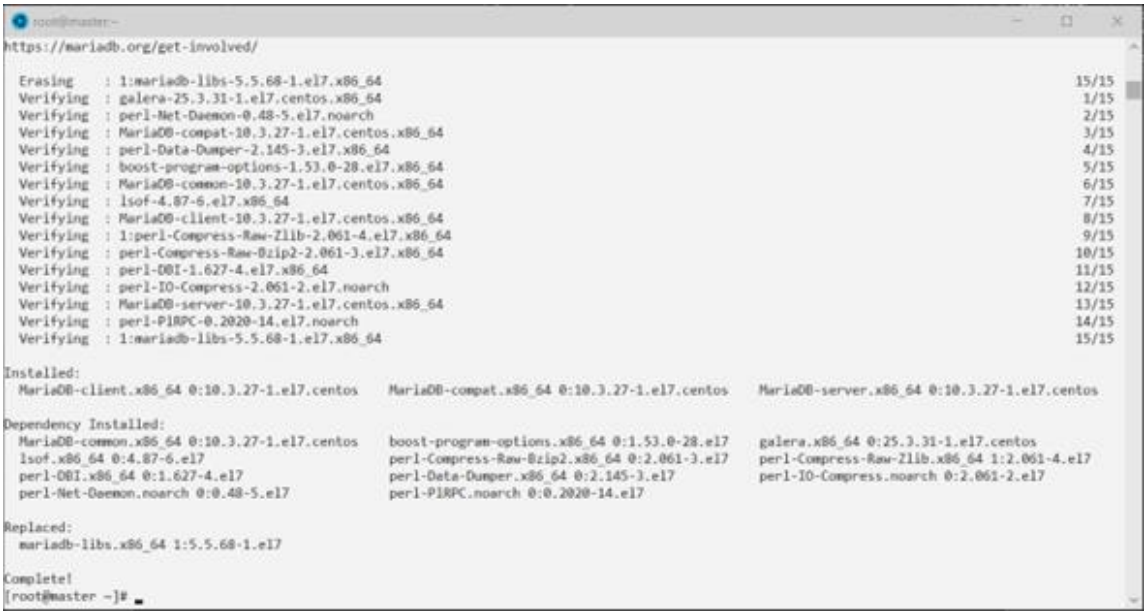


Figura 20 Instalación MariaDB

Al tener una instalación exitosa se ingresan las siguientes instrucciones:

- Se configura MariaDB para que inicie como un servicio del Sistema Operativo (figura 21).

systemctl enable mariadb

```
[root@master ~]# systemctl enable mariadb
Created symlink from /etc/systemd/system/mysql.service to /usr/lib/systemd/system/mariadb.service.
Created symlink from /etc/systemd/system/mysqld.service to /usr/lib/systemd/system/mariadb.service.
Created symlink from /etc/systemd/system/multi-user.target.wants/mariadb.service to /usr/lib/systemd/system/mariadb.service.
```

Figura 21 Inicialización MariaDB

- Se inicia el servicio de MariaDB.

systemctl start mariadb

- Se válida el estado del servicio de MariaDB (figura 22).

systemctl status mariadb

```
[root@master ~]# systemctl status mariadb
● mariadb.service - MariaDB 10.3.27 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; vendor preset: disabled)
  Drop-In: /etc/systemd/system/mariadb.service.d
           └─migrated-from-my.cnf-settings.conf
   Active: active (running) since Sat 2021-01-09 16:56:53 PST; 13s ago
     Docs: man:mysqld(8)
           https://mariadb.com/kb/en/library/systemd/
  Process: 1797 ExecStartPost=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
  Process: 1754 ExecStartPre=/bin/sh -c [ ! -e /usr/bin/galera_recovery ] && VAR= ||   VAR=`cd /usr/bin/..; /usr/bin/galera_recovery`; [ $? -eq 0
]   && systemctl set-environment _WSREP_START_POSITION=$VAR || exit 1 (code=exited, status=0/SUCCESS)
  Process: 1752 ExecStartPre=/bin/sh -c systemctl unset-environment _WSREP_START_POSITION (code=exited, status=0/SUCCESS)
 Main PID: 1765 (mysqld)
   Status: "Taking your SQL requests now..."
   CGroup: /system.slice/mariadb.service
           └─1765 /usr/sbin/mysqld

Jan 09 16:56:53 master mysqld[1765]: 2021-01-09 16:56:53 0 [Note] InnoDB: 10.3.27 started; log sequence number 1625443; transaction id 20
Jan 09 16:56:53 master mysqld[1765]: 2021-01-09 16:56:53 0 [Note] InnoDB: Loading buffer pool(s) from /var/lib/mysql/ib_buffer_pool
Jan 09 16:56:53 master mysqld[1765]: 2021-01-09 16:56:53 0 [Note] InnoDB: Buffer pool(s) load completed at 210109 16:56:53
Jan 09 16:56:53 master mysqld[1765]: 2021-01-09 16:56:53 0 [Note] Plugin 'FEEDBACK' is disabled.
Jan 09 16:56:53 master mysqld[1765]: 2021-01-09 16:56:53 0 [Note] Server socket created on IP: '::'.
Jan 09 16:56:53 master mysqld[1765]: 2021-01-09 16:56:53 0 [Note] Reading of all Master_info entries succeeded
Jan 09 16:56:53 master mysqld[1765]: 2021-01-09 16:56:53 0 [Note] Added new Master_info '' to hash table
Jan 09 16:56:53 master mysqld[1765]: 2021-01-09 16:56:53 0 [Note] /usr/sbin/mysqld: ready for connections.
Jan 09 16:56:53 master mysqld[1765]: Version: '10.3.27-MariaDB'  socket: '/var/lib/mysql/mysql.sock'  port: 3306  MariaDB Server
Jan 09 16:56:53 master systemd[1]: Started MariaDB 10.3.27 database server.
```

Figura 22 Estatus del servicio MariaDB

Posteriormente se fortalece la seguridad de la base de datos con la aplicación destinada para ello se ingresa la siguiente instrucción (figura 23 y figura 24).

mysql_secure_installation

La salida y las respuestas son las siguientes:

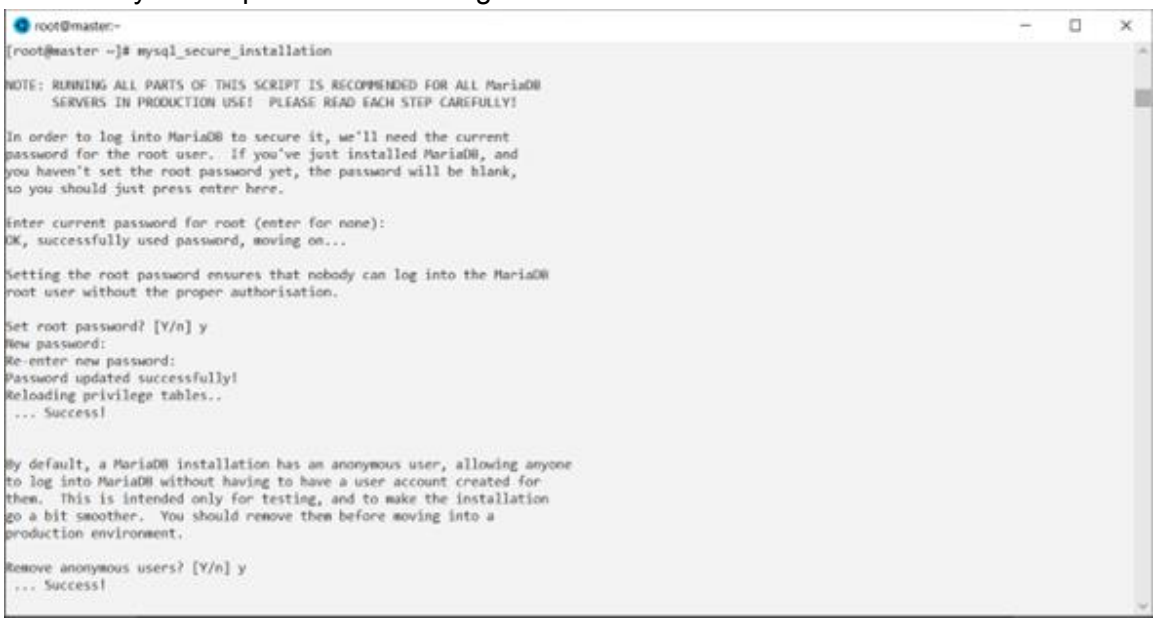


Figura 23 Configuración de Seguridad MariaDB

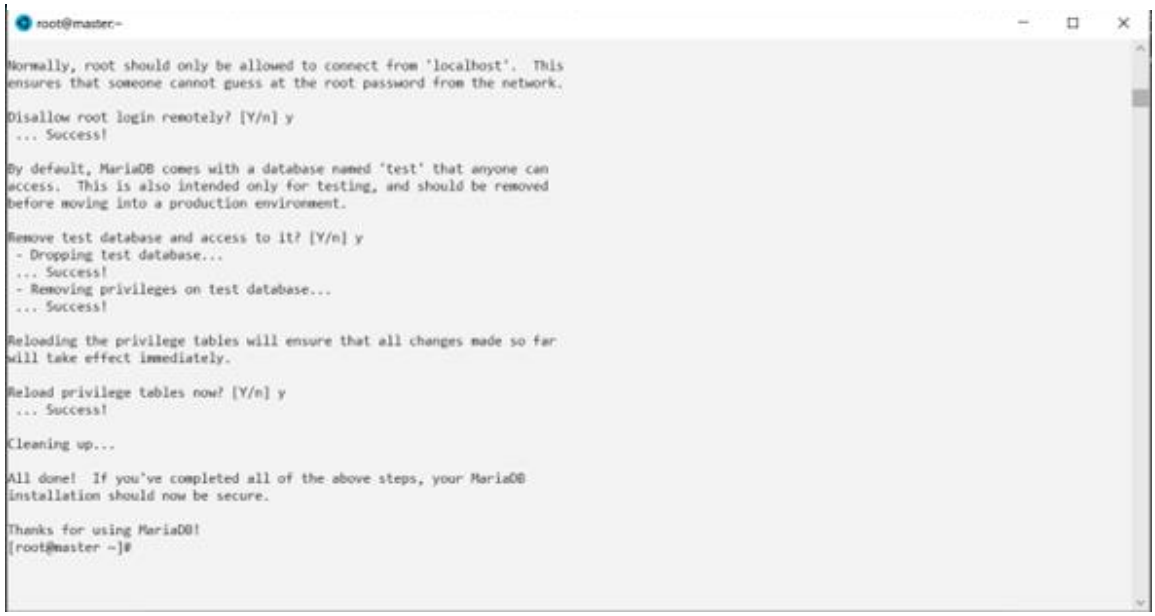


Figura 24 Configuración de seguridad MariaDB

Después de configurar exitosamente, se debe detener el servicio de MariaDB.
systemctl stop mariadb

Nota:

Repetir los pasos del 1 al 8 para todos los nodos

Paso 9: Una vez que se tengan completos los pasos anteriores en cada uno de los nodos, se configuran los nodos en clúster con la siguiente instrucción:

```
vi /etc/my.cnf.d/server.cnf
```

El contenido será el siguiente:

```
#
# * Galera-related settings
#
[galera]
# Mandatory settings
wsrep_on=ON
wsrep_provider=/usr/lib64/galera/libgalera_smm.so
wsrep_cluster_address="gcomm://XXX.XXX.XXX.201,XXX.XXX.XXX.202,
XXX.XXX.XXX.203"
wsrep_cluster_name="Galera_Cluster"
binlog_format=row
default_storage_engine=InnoDB
innodb_autoinc_lock_mode=2
wsrep_node_address="XXX.XXX.XXX.201"
wsrep_node_name="master"
wsrep_sst_method=rsync
#
# Allow server to accept connections on all interfaces.
#
#bind-address=0.0.0.0
#
# Optional setting
#wsrep_slave_threads=1
innodb_flush_log_at_trx_commit=0
```

En la siguiente ruta se ingresa la dirección en donde se encuentra instalado Galera.

```
wsrep_provider=/usr/lib64/galera/libgalera_smm.so
```

Ingresar las direcciones IP de los nodos en los cuales se instalado Galera.

```
wsrep_cluster_address="gcomm://XXX.XXX.XXX.201,XXX.XXX.XXX.202,
XXX.XXX.XXX.203"
```

Se configura el nombre del clúster.

```
wsrep_cluster_name="Galera_Cluster"
```

Nota:

Todo lo anterior se tiene que ingresar en cada nodo.

En estas líneas se pone la información de cada nodo.

Se ingresa la dirección IP del nodo.

```
wsrep_node_address="XXX.XXX.XXX.201"
```

Nombre del nodo.

```
wsrep_node_name="master"
```

Método de sincronización (para todos los nodos).

```
wsrep_sst_method=rsync
```

Esta línea es opcional se recomienda no des comentarla

Nota:

Si el servidor está preconfigurado por un tercero con reglas de cortafuegos local, se debe abrir los puertos necesarios para Galera y María DB en cada nodo para que haya comunicación en el clúster.

```
#bind-address=0.0.0.0
```

Paso 10: Configurando el Firewall de CentOS Linux en todos los nodos.

Reiniciamos el servicio dbus.

```
systemctl restart dbus
```

- Configuramos el servicio firewalld para que inicie con el SO.

```
systemctl enable firewalld
```

- Iniciamos el servicio del firewalld

```
systemctl start firewalld
```

- Verificamos el estado del firewalld

```
systemctl status firewalld
```

CentOS usa la utilidad **firewall-cmd**, las siguientes instrucciones abrirán puertos y acceso a las direcciones IP de la red de replicación para los servicios de MariaDB y de Galera:

```
firewall-cmd --zone=public --add-service=mysql --permanent
firewall-cmd --zone=public --add-port=3306/tcp --permanent
firewall-cmd --zone=public --add-port=4444/tcp --permanent
firewall-cmd --zone=public --add-port=4567/tcp --permanent
firewall-cmd --zone=public --add-port=4567/udp --permanent
firewall-cmd --zone=public --add-port=4568/tcp --permanent
```

Para corroborar que la configuración se aplicó correctamente, se tiene que verificar con la siguiente instrucción.

```
iptables -L
```

Ahora hay que agregar cada nodo a la zona **public** ejecutando las siguientes instrucciones.

```
firewall-cmd --permanent --zone=public --add-source=XXX.XXX.XXX.201/32
firewall-cmd --permanent --zone=public --add-source=XXX.XXX.XXX.202/32
firewall-cmd --permanent --zone=public --add-source=XXX.XXX.XXX.203/32
```

Posteriormente se aplican todas estas reglas de **firewall-cmd**:

```
firewall-cmd --reload
```

Paso 11: Configuración de SELinux

Ahora se crea una política de SELinux que permitirá a todos los nodos en el clúster ser capaces de comunicarse entre sí y realizar las operaciones de replicación.

SELinux es un módulo del kernel que mejora la seguridad del sistema operativo con soporte para control de acceso y políticas de control de acceso prioritario. Viene habilitado por defecto en CentOS 7 y restringe al demonio de MariaDB para realizar muchas actividades necesarias, por lo que crearemos una política realizando una serie de actividades necesarias seteando SELinux en modo permisivo para MariaDB. Luego se crea una política a partir de eventos en los logs, para finalmente volver a dejar a SELinux en modo seguro (enforcing) una vez que la política definitiva se haya instalado exitosamente.

Primero permitiremos acceso a los puertos necesarios para el clúster Galera ejecutando las siguientes instrucciones en cada nodo:

```
semanage port -a -t mysqld_port_t -p tcp 4567
semanage port -a -t mysqld_port_t -p udp 4567
semanage port -a -t mysqld_port_t -p tcp 4568
semanage port -a -t mysqld_port_t -p tcp 4444
```

La instrucción ***"semanage"*** se usa para administrar la política de SELinux. la sub-instrucción ***"port"*** se usa para modificar las asignaciones de tipo de puerto en la política de SELinux. El indicador ***-a*** significa "agregar" y le dice a ***"semanage"*** que agregue una nueva asignación de tipo de puerto. El indicador -t especifica el tipo de puerto y el indicador -p especifica el protocolo (TCP o UDP[47]). Con la instrucción que proporcionó, ***"semanage"*** agregará una nueva asignación para el puerto.

Esta instrucción se usa para permitir que el demonio MySQL (mysqld) se vincule a un puerto que no sea el puerto predeterminado 3306. De manera predeterminada, mysqld solo puede vincularse al puerto 3306, pero puede usar la instrucción para permitir que se vincule a un puerto diferente.

Luego deberá reiniciar el demonio MySQL para que los cambios surtan efecto.

Puertos:

4444, 4567 y 4567 TCP y UDP no están asociados a MySQL por defecto. MySQL generalmente utiliza el puerto 3306 para la conexión de clientes. Si está utilizando una configuración personalizada de MySQL, es posible que haya cambiado el puerto utilizado para conectarse a la base de datos. En ese caso, de los puertos 4567 y 4568 podría estar siendo utilizado por MySQL.

Si está utilizando una distribución de MySQL que incluye un firewall, es posible que se tenga que abrir el puerto 4567 en el firewall para permitir la conexión de clientes a la base de datos a través de ese puerto. Si está utilizando una máquina virtual, es posible que también tenga que configurar la red para permitir el tráfico entrante en los puertos 4444, 4567 y 4568.

Es importante tener en cuenta que, a menos que haya cambiado la configuración predeterminada de MySQL, es poco probable que los puertos 4444, 4567 y 4568 esté siendo utilizado por MySQL. Si está teniendo problemas para conectarse a la base de datos, es más probable que se deba a un problema con la configuración del puerto 3306 o alguna otra configuración relacionada con la conexión.

> **Nota:**
>
> Es posible que salga el error de tipo **ValueError** cuando se de acceso a algunos de estos puertos, debido a que el estatus de ese puerto en SELinux ya esté configurado, sin afectar el desarrollo de todo este proceso.

[47] User Datagram Protocol

Luego se ejecutará la siguiente instrucción en cada nodo para dejar el dominio de SELinux sobre MySQL SELinux en modo permisivo de manera temporal, para poder trabajar y generar registros de actividad de MariaDB y Galera y así notificar a SELinux que dicha actividad es normal en cada servidor.

```
semanage permissive -a mysqld_t
```

Esta instrucción puede que demore un poco en completarse y no arroje salida alguna en la Terminal luego, se debe detener el clúster en todos los nodos para arrancarlo nuevamente, en el nodo primario en modo Bootstrap con las políticas de SELinux compartidas en los demás nodos del clúster.

Para realizar esto, se debe ejecutar la siguiente instrucción en cada nodo:

```
systemctl stop mariadb
```

> **Nota:**
>
> Si no hay errores, no se mostrará ninguna salida al finalizar el inicio de MariaDB en modo Bootstrap

Paso 12: Configuración del clúster.

Posteriormente se inicia el arranque, en el primer nodo del clúster en modo Bootstrap es un proceso utilizado en Mariadb Galera para inicializar una nueva instancia de un clúster de base de datos. Es utilizado cuando se agrega un nuevo nodo a un clúster existente o cuando se establece un clúster desde cero. Durante el proceso de Bootstrap, la nueva instancia se sincroniza con uno de los nodos existentes del clúster y obtiene una copia de la base de datos completa[48]. Para generar eventos de comunicación entre los nodos que serán añadidos a una política segura y permanente de SELinux. En el nodo primario del clúster arrancaremos MariaDB en modo bootstrap ejecutando (figura 25):

galera_new_cluster



Figura 25 Creación del clúster

[48] (Mariadb, 2022) (Mariadb, 2022)

Luego se crea una base de datos en el primer nodo junto a una tabla con el objetivo específico de generar logs de eventos SST de Galera entre los nodos ejecutando la siguiente instrucción en el nodo primario (figura 26):

mysql -u root -p

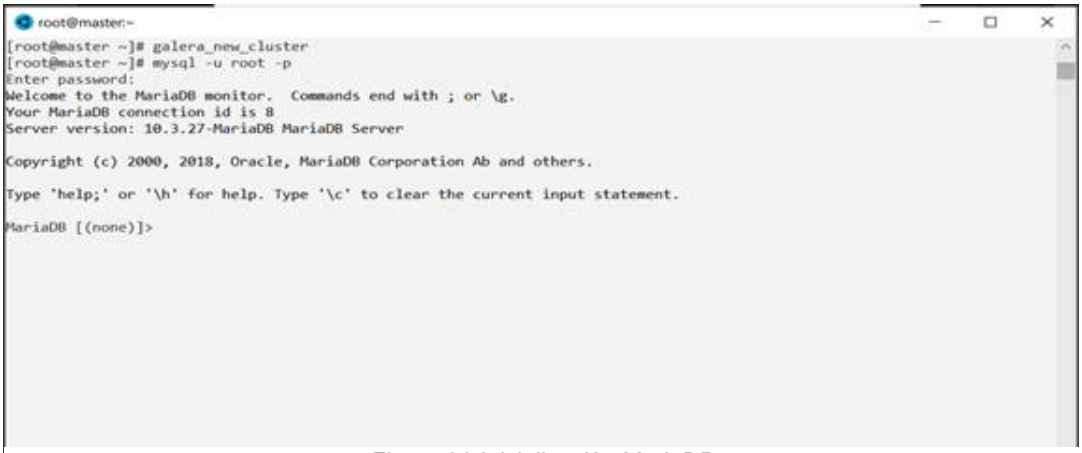


Figura 26 Inicialización MariaDB

Una vez en la consola de mysql escribimos la siguiente *query* compuesta:

MariaDB [(none)]> CREATE DATABASE selinux;
MariaDB [(none)]> CREATE TABLE selinux.selinux_policy (id INT NOT NULL AUTO_INCREMENT,
PRIMARY KEY(id));
MariaDB [(none)]> INSERT INTO selinux.selinux_policy VALUES ();

Después de ejecutada la consulta compuesta hay que salir de la consola de mysql con:

MariaDB [(none)]> quit;

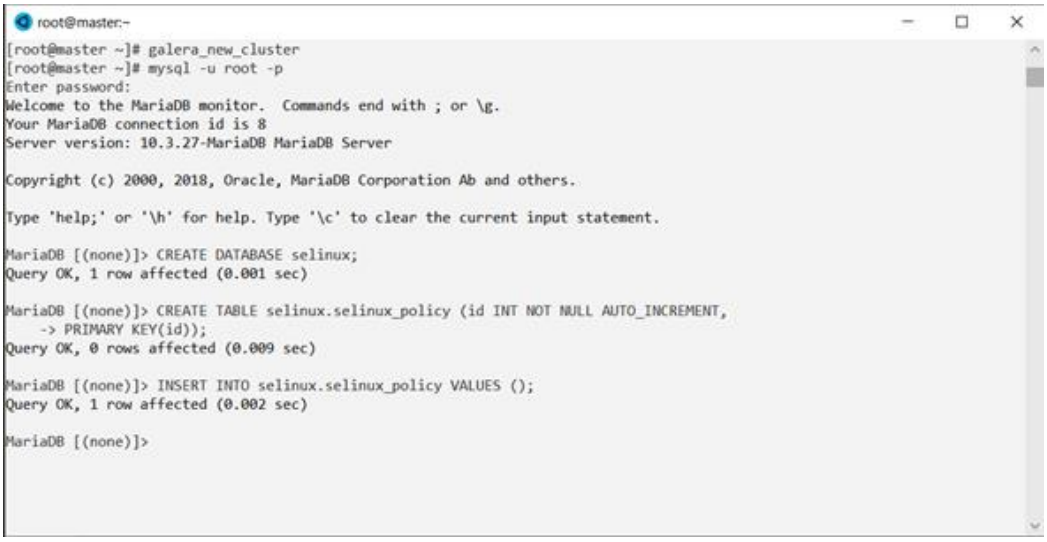


Figura 27 Creación de base de datos inicial

Luego se inicializan los nodos con la instrucción:

systemctl start mariadb

> **Nota:**
>
> En caso de tener errores mariadb.service failed because the control process exited with error code. See "systemctl status mariadb.service" and "journalctl -xe" for details.
>
> Se recomienda reiniciar todos los nodos y repetir con la instrucción galera_new_cluster (master) y systemctl start mariadb (demas nodos).

Con todos los nodos en operación, se debe ejecutar la siguiente instrucción para generar actividades en la base de datos y así generar eventos IST de Galera en cada uno de los nodos (figura 28):

mysql -u root -p -e 'INSERT INTO selinux.selinux_policy VALUES ();'

```
[root@nodo2 ~]# mysql -u root -p -e 'INSERT INTO selinux.selinux_policy VALUES ();'
Enter password:
[root@nodo2 ~]# grep mysql /var/log/audit/audit.log | sudo audit2allow -M Galera
******************* IMPORTANT ***********************
To make this policy package active, execute:

semodule -i Galera.pp
```

Figura 28 Configuración políticas selinux.

Se crea y habilita una política permanente y segura en SELinux ejecutando la siguiente instrucción en los tres nodos:

grep mysql /var/log/audit/audit.log | sudo audit2allow -M Galera

Para verificar su correcto funcionamiento, consultamos la tabla de estado de clúster en MariaDB con la instrucción (figura 29).

mysql -u root -p -e " SHOW STATUS LIKE 'wsrep_cluster_size' "
La instrucción anterior solo se ejecuta en el nodo maestro (master).

```
[root@nodo1 ~]# mysql -u root -p -e " SHOW STATUS LIKE 'wsrep_cluster_size' "
Enter password:
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| wsrep_cluster_size | 1     |
+--------------------+-------+
[root@nodo1 ~]# mysql -u root -p -e " SHOW STATUS LIKE 'wsrep_cluster_size' "
Enter password:
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| wsrep_cluster_size | 2     |
+--------------------+-------+
[root@nodo1 ~]#
```

Figura 29 Estatus de cada nodo del clúster

Ahora se ejecuta en el nodo “n”.

```
[root@nodo2 ~]# mysql -u root -p -e " SHOW STATUS LIKE 'wsrep_cluster_size' "
Enter password:
+--------------------+-------+
| Variable_name      | Value |
+--------------------+-------+
| wsrep_cluster_size | 3     |
+--------------------+-------+
[root@nodo2 ~]#
```

Figura 30 Ejecución en cualquier nodo

En la imagen anterior se aprecia el cambio en la tabla y la columna Value, la cual cambia cuando agregamos nodos al clúster (figura 30).

Solución de problemas:

En caso de que el nodo o nodos no se puedan levantar con la instrucción ***systemctl start mariadb*** y aparezca el siguiente error ***mariadb.service failed because the control process exited with error code. See "systemctl status mariadb.service" and "journalctl -xe" for details***.

Se recomienda usar estas dos posibles soluciones.

Solución 1:

En caso de aparecer el error antes mencionado, se puede aplicar esta solución solo en el caso de la instalación, este error es muy común a la hora de ejecutar la siguiente instrucción ***systemctl start mariadb***, para desplegar más acerca de este error ejecutamos la siguiente instrucción ***systemctl status mariadb*** y saldrá lo siguiente (figura 31).



Figura 31 Error al inicializar el servicio de MariaDB

Para solucionar este problema se recomienda aplicar las siguientes instrucciones en cada nodo.

```
semanage port -a -t mysqld_port_t -p tcp 4567
semanage port -a -t mysqld_port_t -p udp 4567
semanage port -a -t mysqld_port_t -p tcp 4568
semanage port -a -t mysqld_port_t -p tcp 4444
```

```
semanage permissive -a mysqld_t
```

Después de ejecutar las instrucciones anteriores se ejecuta la siguiente instrucción en ambos nodos del clúster (figura 32).

```
systemctl stop mariadb
```

Nota:

Es posible que salga el error de tipo **ValueError** cuando se de acceso a algunos de estos puertos, debido a que el estatus de ese puerto en SELinux ya esté configurado, sin afectar el desarrollo de todo este proceso.

De igual forma aplica cuando se quieran agregar más nodos al clúster teniendo en cuenta editar el archivo de configuración `vi /etc/my.cnf.d/server.cnf` explicado en el paso 9.



Figura 32 Ejecución del modo permisive

Solución 2: En caso de que la solución anterior no sea favorable se recomienda desinstalar el Galera y Mariadb, con la siguiente instrucción.

```
yum -y remove MariaDB-server MariaDB-client galera-4
```

Para garantizar la correcta desinstalación es necesario remover los siguientes directorios de la base de datos.

```
rm -rf /var/lib/mysql/
```

```
rm -rf /etc/my.cnf
```

Con la siguiente instrucción eliminamos todos los paquetes de MariaDB-*

```
yum -y remove MariaDB-*
```

Limpiamos los repositorios restantes.

```
yum -y autoremove
```

4.5. Instalación y configuración MaxScale.

Para configurar el MaxScale se requiere de tener configurado previamente los servidores con el software de Galera MariaDB, los cuales previamente, se modificaron y tienen las siguientes direcciones IP para este clúster.

- ***XXX.XXX.XXX.201***
- ***XXX.XXX.XXX.202***
- ***XXX.XXX.XXX.203***

El servidor donde se instala MaxScale tiene la siguiente dirección IP ***XXX.XXX.XXX.204*** y sus requerimientos de hardware son los siguientes:

- Almacenamiento: 110 GB
- CPU'S: 4
- RAM: 8 GB

Al igual que los nodos de Galera MariaDB usaremos Centos 7.

Antes de empezar a instalar los paquetes correspondientes se debe configurar el siguiente archivo "*host*", dentro del servidor antes mencionado, en el cual agregaremos las direcciones IP de los nodos del clúster (figura 33).

vi /etc/hosts



Figura 33 Configuración de los Hosts

Se guardarán las configuraciones dentro del archivo, posteriormente se deben instalar los siguientes paquetes.

MariaDB y MaxScale desde los repositorios de yum.

curl -sS https://downloads.mariadb.com/MariaDB/mariadb_repo_setup | bash

Se construye el cache de yum.

yum makecache fast

Después se instala MaxScale utilizando la siguiente instrucción.

```
yum install -y maxscale
```

Una vez que se tiene instalado MaxScale se puede conectar al cualquiera de los nodos, esto para crear un usuario el cual se conectará con el servidor de MaxScale, para este caso se conecta al nodo1 el cual tiene la IP ***XXX.XXX.XXX.201.***

Una vez dentro, para acceder a la base de datos usamos la siguiente instrucción.

```
mysql -u root -p
```

Posteriormente para agregar el usuario se usan las siguientes instrucciones.

Primero se crea el usuario en este caso se llama "***maxscale***" con la dirección IP del servidor ***XXX.XXX.XXX.204*** y su contraseña ***"xxxxxxxxxx".***

```
create user 'maxscale'@'XXX.XXX.XXX.204' identified by 'xxxxxxxxxx';
```

Ahora se le otorga los permisos de SELECT al usuario previamente creado.

```
grant select on mysql.user to 'maxscale'@'XXX.XXX.XXX.204';
```

Se Otorgan los permisos de SELECT dentro del esquema de base de datos al usuario previamente creado.

```
grant select on mysql.db to 'maxscale'@'XXX.XXX.XXX.204';
```

Se le dan los permisos sobre ciertas tablas privadas al usuario previamente creado

```
grant select on mysql.tables_priv to 'maxscale'@'XXX.XXX.XXX.204';
```

Se Otorgan los permisos de mostrar las bases de datos existentes dentro del manejador al usuario previamente creado.

```
grant show databases on *.* to 'maxscale'@'XXX.XXX.XXX.204';
```

Se recargan los privilegios.

```
flush privileges;
```

Para verificar los permisos del usuario creado se usa la siguiente instrucción (figura 34).

```
SHOW GRANTS FOR "maxscale"@"XXX.XXX.XXX.204";
```

La instrucción ***"grant"*** es el permiso otorgado a un usuario para realizar ciertas tareas en concreto acciones en una tabla o vista tales como SELECT, INSERT, UPDATE, DELETE, etc. Este permiso puede ser otorgado o revocado por un usuario con privilegios de administrador.

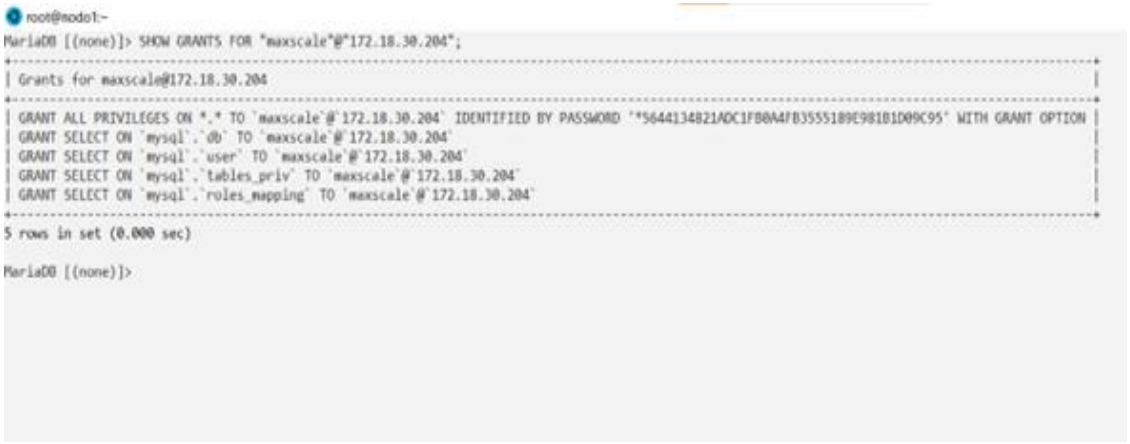


Figura 34 Verificación de permisos

En cada uno de los nodos del clúster (nodo1, nodo2 y nodo3) se deben de abrir los siguientes puertos para que funcione el MaxScale con las siguientes instrucciones.

firewall-cmd --zone=public --add-port=4000/tcp --permanent

firewall-cmd --zone=public --add-port=4408/tcp --permanent

Se agrega la dirección IP del MaxScale.

firewall-cmd --permanent --zone=public --add-source=XXX.XXX.XXX.204/32

Ahora se recargan los servicios del firewall.

firewall-cmd --reload

Una vez que se ha configurado el usuario en la base de datos, se edita el archivo de configuración del MaxScale para conectar el clúster.

Antes de modificar se tiene que crear una copia del archivo, con la siguiente instrucción.

mv /etc/maxscale.cnf /etc/maxscale.cnf.old

Ahora se modifica el archivo con la siguiente instrucción.

vi /etc/maxscale.cnf

En el encabezado del archivo están definidos los hilos del servidor de MaxScale el cual se recomienda en automático.

[maxscale]

threads=auto

Se agregan las direcciones y el nombre de los servidores del clúster de MariaDB.

[nodo1]

type=server

address=XXX.XXX.XXX.201

port=3306

```
protocol=MariaDBBackend
[nodo2]
type=server
address=XXX.XXX.XXX.202
port=3306
protocol=MariaDBBackend
[nodo3]
type=server
address=XXX.XXX.XXX.203
port=3306
protocol=MariaDBBackend
```

Posteriormente se agrega el monitor el cual debe tener el usuario previamente creado en la base de datos, así como el nombre de los servidores del clúster.

```
[MariaDB-Monitor]
type=monitor
module=mariadbmon
servers=nodo1,nodo2,nodo3
user=maxscale
password=xxxxxxxxxx
monitor_interval=2000
```

A continuación, se agregan las definiciones del servidor. es decir, los servicios de lectura y de escritura, el cual debe tener el nombre de los servidores de galera, así como el usuario previamente agregado en la base de datos.

```
[Read-Only-Service]
type=service
router=readconnroute
servers=nodo1,nodo2,nodo3
user=maxscale
```

```
password=xxxxxxxxxx

router_options=slave


[Read-Write-Service]

type=service

router=readwritesplit

servers=nodo1,nodo2,nodo3

user=maxscale

password=xxxxxxxxxx
```

Se tiene que configurar los puertos tanto de escritura como de lectura en este caso tenemos el puerto 4000 y 4006 los cuales están definidos por omisión.

```
[Read-Only-Listener]

type=listener

service=Read-Only-Service

protocol=MariaDBClient

port=4008


[Read-Write-Listener]

type=listener

service=Read-Write-Service

protocol=MariaDBClient

port=4006
```

Ahora se tienen que agregar los puertos al firewall dentro del servidor antes de arrancar MaxScale.

```
firewall-cmd --zone=public --add-port=4000/tcp --permanent
```

```
firewall-cmd --zone=public --add-port=4408/tcp --permanent
```

Se agregan las IP's de los servidores del clúster.

```
firewall-cmd --permanent --zone=public --add-source=XXX.XXX.XXX.201/32
```

```
firewall-cmd --permanent --zone=public --add-source=XXX.XXX.XXX.202/32
```

```
firewall-cmd --permanent --zone=public --add-source=XXX.XXX.XXX.203/32
```

```
firewall-cmd --permanent --zone=public --add-source=XXX.XXX.XXX.204/32
```

Se recargan los servicios del firewall

```
firewall-cmd –reload
```

Una vez que se tiene la configuración previa se debe agregar el servicio de MaxScale para que inicie cuando arranque el sistema operativo el cual se hace con la siguiente instrucción.

```
systemctl enable maxscale.service
```

Para ponerlo en marcha se usar la siguiente instrucción.

```
systemctl start maxscale.service
```

Para verificar su correcto funcionamiento se tiene que usar la siguiente instrucción (figura 35).

```
systemctl status maxscale.service
```



Figura 35 Estatus del servicio maxscale

Ahora para verificar que los nodos estén activos se hace con la siguiente instrucción (figura 36).

maxctrl list servers

root@maxscale:~

```
[root@maxscale ~]# maxctrl list servers
```

Server	Address	Port	Connections	State	GTID
nodo1	172.18.30.201	3306	0	Master, Running	
nodo2	172.18.30.202	3306	0	Running	
nodo3	172.18.30.203	3306	0	Running	

```
[root@maxscale ~]#
```

Figura 36 Lista de servidores disponibles

Para verificar los servicios se hace con la siguiente instrucción (figura 37).

maxctrl list services

root@maxscale:~

```
[root@maxscale ~]# maxctrl list services
```

Service	Router	Connections	Total Connections	Servers
Read-Only-Service	readconnroute	0	0	nodo1, nodo2, nodo3
Read-Write-Service	readwritesplit	0	0	nodo1, nodo2, nodo3

```
[root@maxscale ~]#
```

Figura 37 Lista de servicios

Solución de problemas.

Cuando se inicia el servicio de MaxScale, pero al hacer un ***"status"*** nos damos cuenta de que tiene las siguientes advertencias (warnings), por ejemplo.

Se debe a los permisos del usuario dentro de la misma, en caso de que no funcione se recomienda darle todos los permisos con la siguiente instrucción.

```
grant show databases on *.* to maxadmn@'%';
```

si el problema sigue igual se recomienda editar el archivo de configuración del maxscale ***/etc/maxscale.cnf*** y comentar los servicios de escritura y lectura, en caso de que no arranque, se debe regresar a la configuración anterior y ahora se puede arrancar de manera normal al revisar el ***status*** los errores deberán de desaparecer esto puede llegar a pasar cuando la configuración no se ha cargado correctamente (figura 38).



Figura 38 Advertencias de permisos

4.6. Pruebas

Dada la configuración e instalación de MariaDB Galera Cluster y del balanceador maxscale y probar su correcto funcionamiento, se pretende conectar este a un aplicativo el cual es Moodle con la finalidad de hacer un plan de pruebas, estas se hicieron con el software Jmeter, el cual es una prueba indirecta es decir solo se probará desde el aplicativo y estresará[49] al clúster para hacer una métrica del número de usuarios soportados por la plataforma de manera concurrente, a su vez hacer mejoras del mismo.

En el siguiente diagrama (figura 39) se muestra la estructura completa de los componentes antes mencionados.

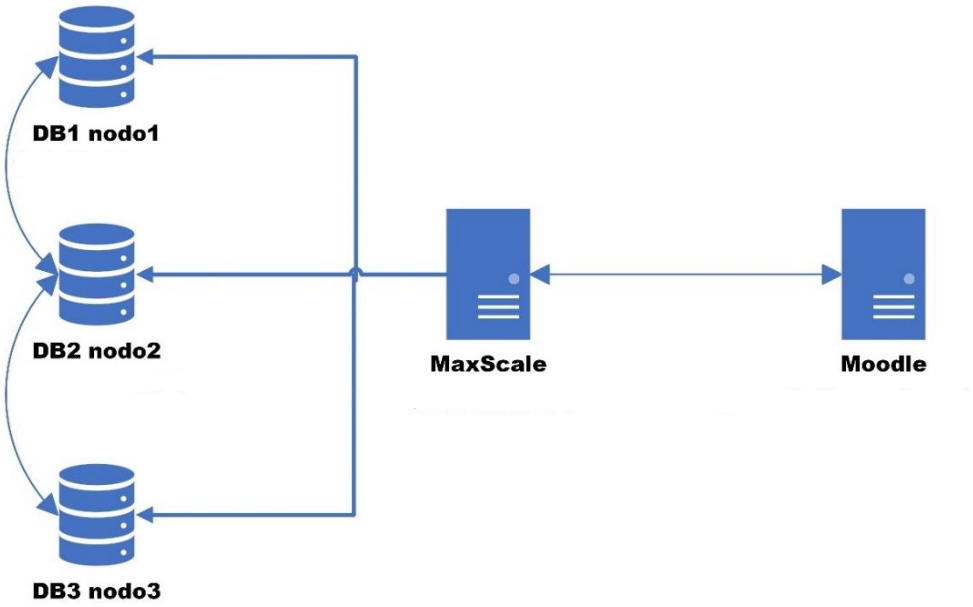


Figura 39 Arquitectura del clúster Moodle

Apache Jmeter: Es una herramienta de software libre que se utiliza para probar el rendimiento de aplicaciones y sistemas. Fue diseñada originalmente para probar aplicaciones basadas en HTTP, pero ahora se puede utilizar para probar muchos tipos diferentes de aplicaciones, incluyendo aplicaciones basadas en web, FTP, JDBC, LDAP y muchas otras.

JMeter[50] se ejecuta como una aplicación de consola y permite a los usuarios crear y configurar scripts de prueba utilizando una interfaz gráfica de usuario. Una vez creados, estos scripts se pueden ejecutar de forma automatizada para simular el tráfico y el uso de la aplicación por parte de los usuarios. Los resultados de la prueba se pueden visualizar en forma de gráficos y tablas para facilitar la interpretación y el análisis.

[49] Estresar común mente se denomina hacer una prueba de estrés, la cuál es una técnica utilizada para evaluar la capacidad de un sistema o entidad para manejar situaciones de carga máxima o condiciones extremas. En términos generales, se realiza sometiendo a un sistema a una carga mayor de lo que normalmente experimentaría en condiciones normales de operación. Esto puede implicar, por ejemplo, aumentar el tráfico en un sitio web, procesar grandes volúmenes de datos en una base de datos, aumentar el número de usuarios que acceden a un sistema en un corto período de tiempo, someter la aplicación a condiciones climáticas extremas

[50] (Apache, 2022)

Moodle: Es una plataforma de gestión del aprendizaje Learning Management System (LMS, por sus siglas en inglés) de código abierto que se utiliza para crear y administrar cursos en línea. Fue desarrollado por el profesor de educación a distancia Martin Dougiamas en el año 1999 y es utilizado por instituciones educativas, empresas y organizaciones en todo el mundo. Ofrece una amplia gama de funcionalidades para ayudar a los docentes a crear y administrar cursos en línea, incluyendo la creación de contenidos multimedia, la comunicación con los estudiantes, el seguimiento del progreso de los estudiantes y la evaluación del aprendizaje.

Permite a los estudiantes participar en discusiones en línea, trabajar en proyectos en equipo y tener acceso a recursos y materiales de estudio en línea, es altamente personalizable y se puede integrar con una amplia variedad de aplicaciones y herramientas de terceros.

Configuración Moodle.

La prueba de estrés se realizará en la plataforma de Moodle (figura 40):

https://moodle-aula-virtual.com/



Figura 40 Plataforma Moodle

Para configurar el Moodle con el clúster, se tuvo que verificar la IP interna del balanceador MaxScale el cual se obtiene con la siguiente instrucción: **`Ifconfig`**, (figura 41) la cual es **`XXX.XXX.XXX.29.`**

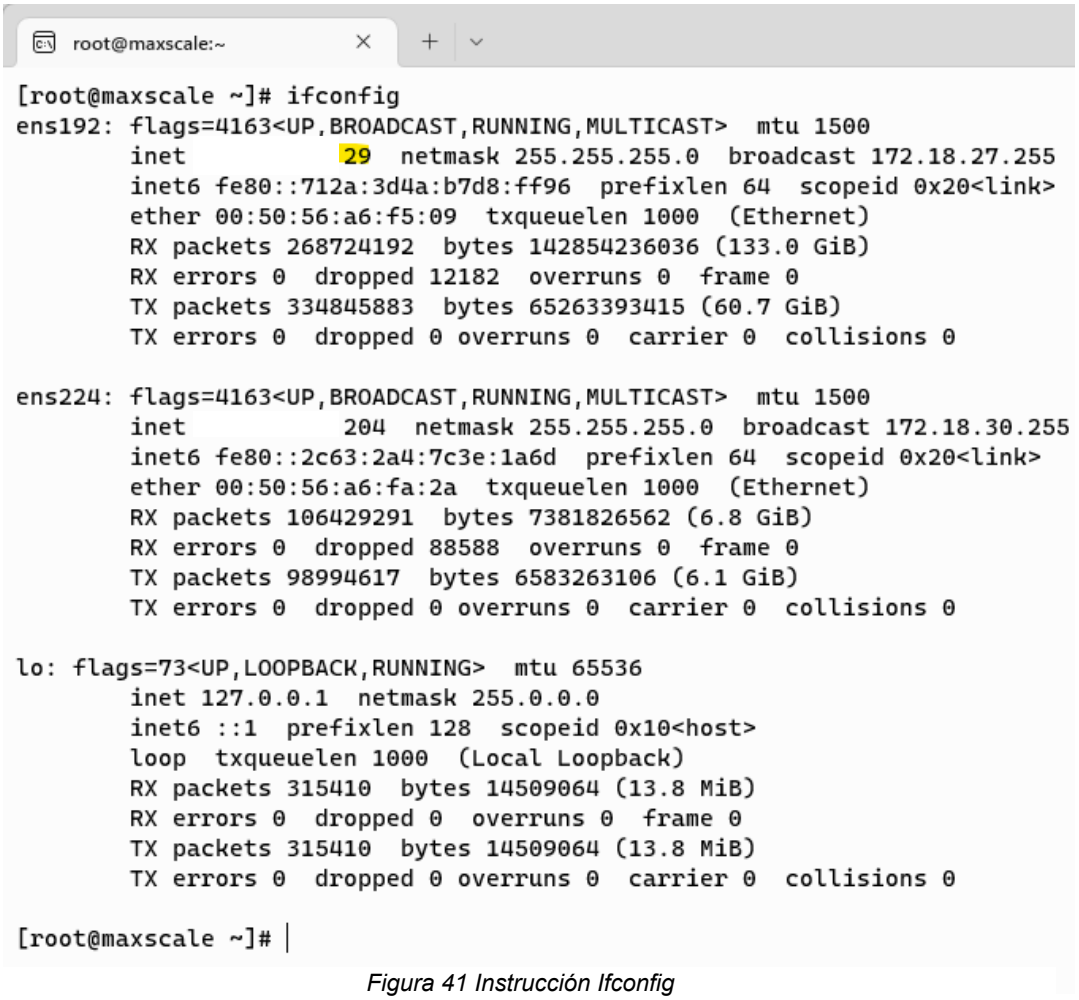


Figura 41 Instrucción Ifconfig

Ahora se tiene que editar el archivo de configuración de Moodle **`config.php`**, a su vez se creó un usuario en la base de datos para la plataforma (figura 42).

Dirección IP del balanceador MaxScale es: **`XXX.XXX.XXX.29`**

Esquema de base de datos es: **`moodle_pruebas`**

Usuario de la base de datos es: **`Moodle_User`**

Password de la base de datos es: **`Prueba-123`**

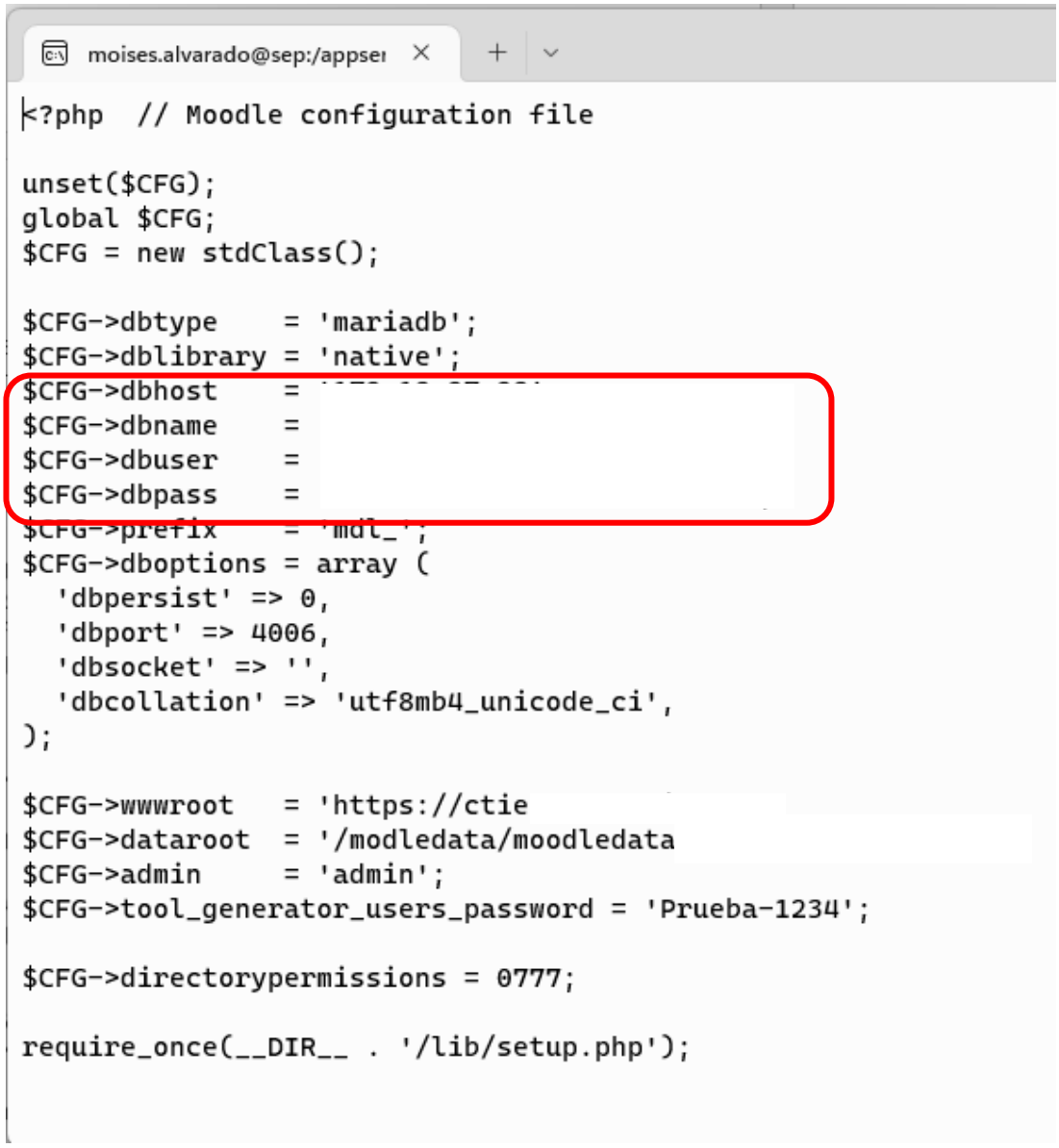


Figura 42 Configuración del Moodle

Se guarda la configuración, se realiza una prueba rápida de benchmark el cual es un plugin nativo dentro de Moodle, el cual se bajó de esta liga.
https://moodle.org/plugins/report_benchmark

Para generar el reporte una vez que se tiene instalado nos dirigimos a **/ Reportes / Benchmark. / Reportes / Benchmark** (figura 43).

Configuración plan de pruebas Jmeter.

Benchmark (Comparativo) del Sistema

Puntuación del Benchmark: 104 puntos

#	Descripción	Tiempo (segundos)	Límite aceptable	Límite crítico
1	Tiempo para cargar Moodle Cargar el archivo de configuración "config.php"	0.011	0.5	0.8
2	Velocidad de procesamiento del procesador Se llama a una función PHP en un bucle para probar la velocidad del procesador	0.106	0.5	0.8
3	Desempeño de lectura de archivos Leer un archivo varias veces para revisar la velocidad de lectura de la carpeta temporal de Moodle	0.014	0.5	0.8
4	Desempeño de escritura de archivos Escribir un archivo varias veces para revisar la velocidad de escritura de la carpeta temporal de Moodle	0.092	1	1.25
5	Desempeño de lectura del curso Leer un curso varias veces para revisar la velocidad de lectura de la base de datos	0.229	0.75	1
6	Desempeño de escritura del curso Escribir un curso varias veces para revisar la velocidad de escritura de la base de datos	0.039	1	1.25
7	Desempeño de Base de Datos (#1) Ejecutar una consulta SQL compleja para probar la velocidad de la Base de datos	0.054	0.5	0.7
8	Desempeño de Base de Datos (#2) Ejecutar una consulta SQL compleja para probar la velocidad de la Base de datos	0.096	0.3	0.5
9	Desempeño del tiempo para conectarse con la cuenta de invitado Revisar el tiempo usado para cargar la página de ingreso de la cuenta de invitado	0.277	0.3	0.8
10	Tiempo para conectarse para una cuenta de usuario falsa Medir el tiempo usado para cargar la página de ingreso de una cuenta falsa de usuario	0.119	0.3	0.8
	Tiempo total	**1.037 s**		
	Puntuación	**104 puntos**		

¡Felicitaciones!
El desempeño de su instalación Moodle parece perfecto.

Figura 43 Resultado del Bechmarck Moodle

Ahora se tiene que generar un plan de pruebas para estresar el sistema mediante un plan de pruebas Jmeter, para esto se necesita agregar las siguientes líneas en el archivo de configuración de la plataforma Moodle (***config.php***) (figura44).

La siguiente línea establece una única contraseña para todos los usuarios.

```
$CFG->tool_generator_users_password = 'Prueba-1234';
```

```
$CFG->noemailever = true;
```

Se tienen que agregan al archivo antes mencionado.



Figura 44 Configuración pruebas Moodle

Posteriormente se habilita la opción de desarrollo en la siguiente ruta del aula.

Administración del sitio / Desarrollo / Depuración (debugging).

En el apartado de mensajes de depuración se tiene que seleccionar Desarrollador (figura 45).

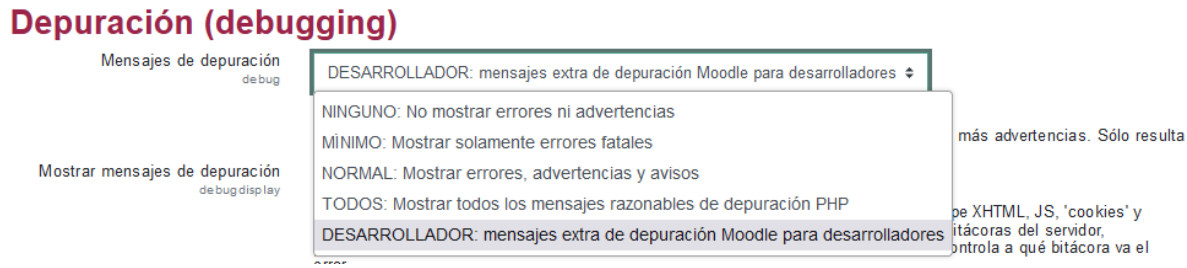


Figura 45 Modo de depuración

Posteriormente se genera un curso de prueba (figura 46) el cual se hará en la siguiente ruta. **/ Desarrollo / Hacer un grupo de prueba. / Desarrollo / Hacer un grupo de prueba.**

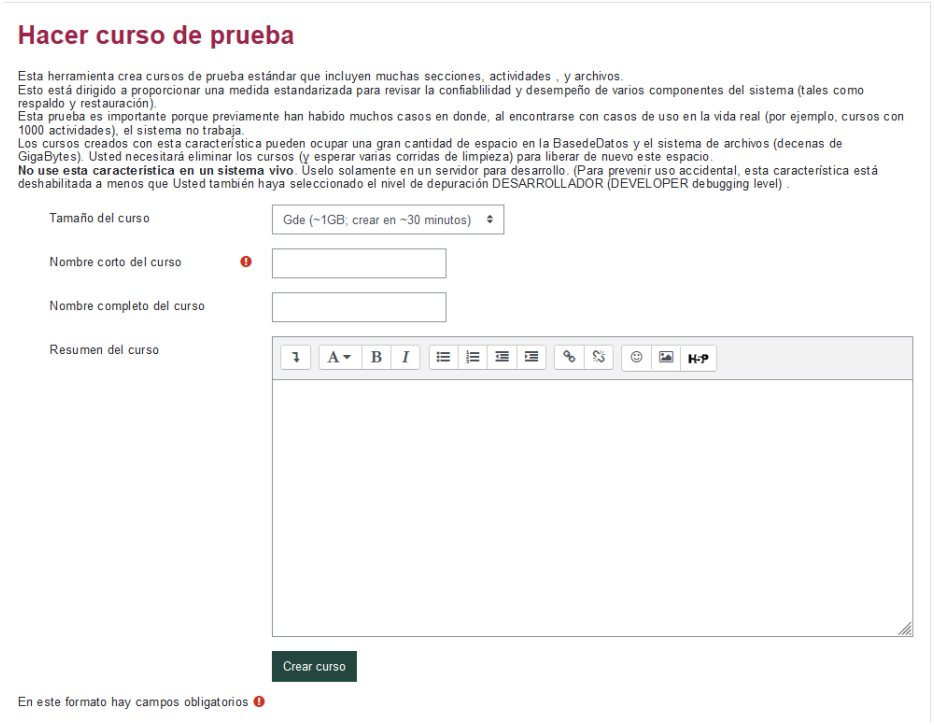


Figura 46 Curso de Prueba

A su vez se seleccionan los siguientes apartados, mensajes de depuración e información del rendimiento y se guarda la configuración (figura 47).

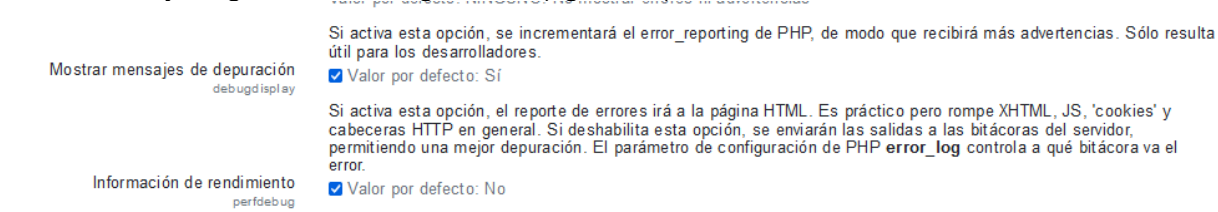


Figura 47 Configuración Mensajes de depuración y rendimiento

Posteriormente de que se generó el curso de prueba, se ahora se genera un curso de prueba M el cual toma aproximadamente 2 minutos (figura 48).

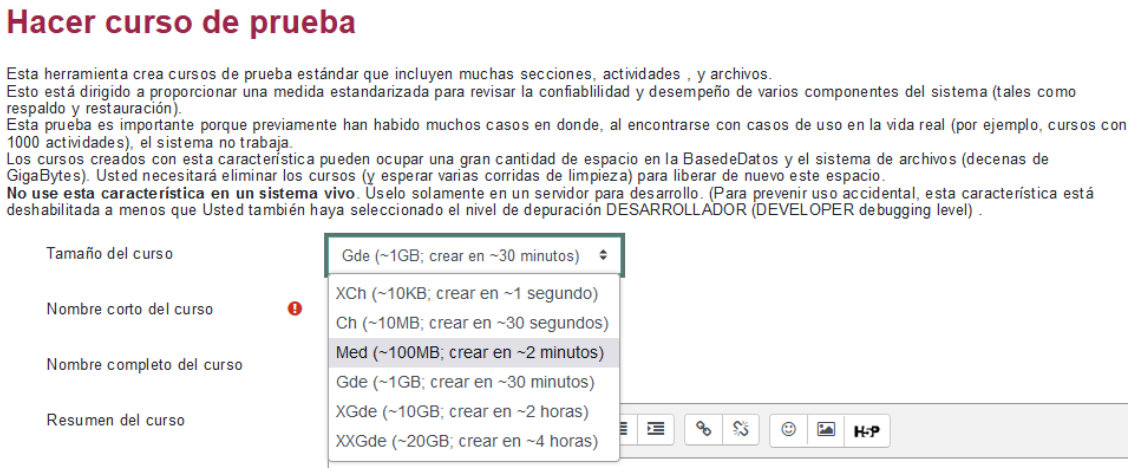


Figura 48 Curso de prueba Moodle

Una vez generado el curso de prueba se procede a generar el plan de pruebas el cual lo realizara en el siguiente apartado (figura 49).

Administración del sitio / Desarrollo / Hacer un plan de prueba Jmeter.

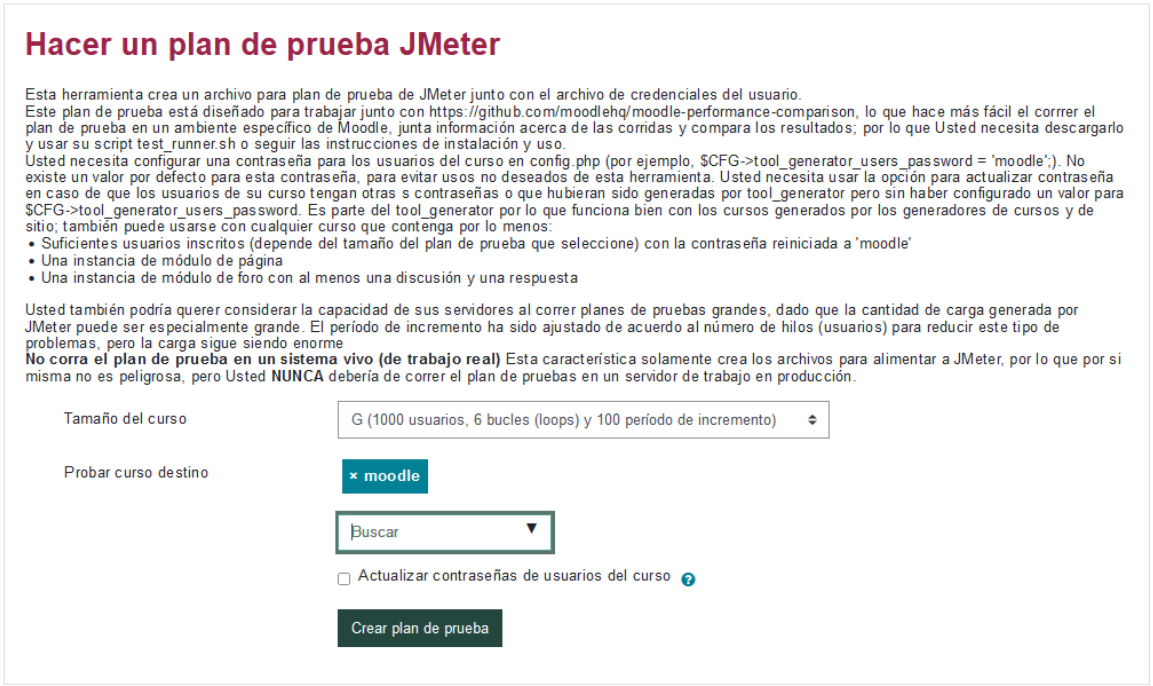


Figura 49 Plan de prueba Jmeter Moodle

De manera opcional se puede seleccionar “Actualizar contraseñas de usuarios del curso”.

Para este ejercicio se decidió hacer una prueba de tamaño XG el cual es de 10G aproximadamente (figura 50).

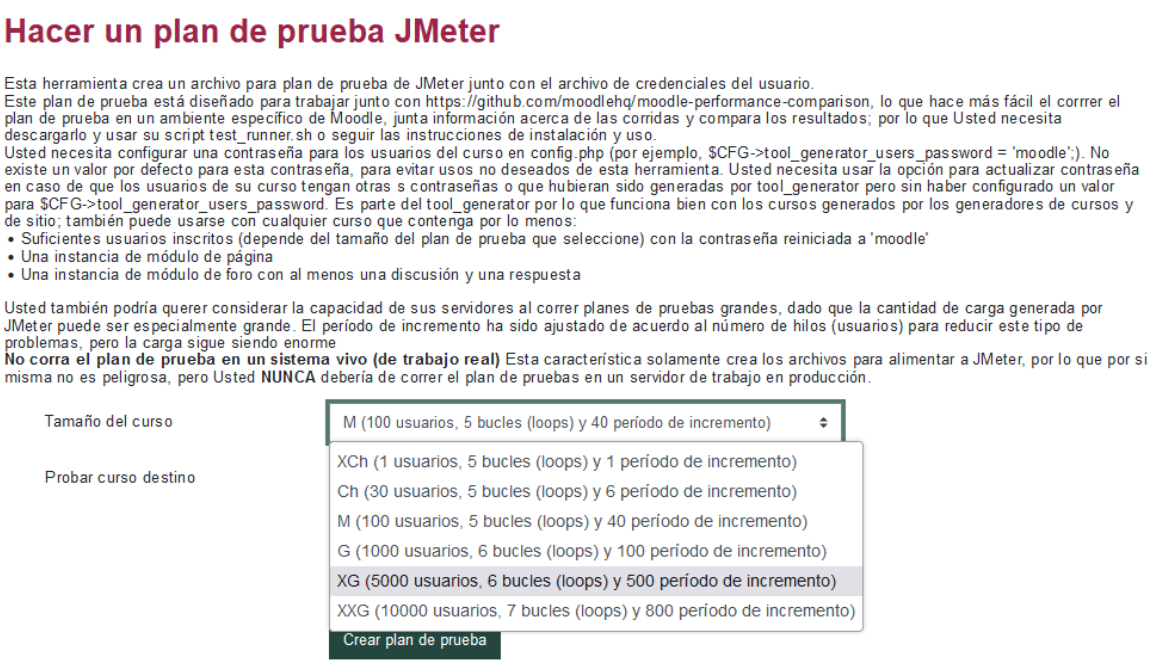


Figura 50 Plan de prueba XG

Damos click en crear plan de prueba y posteriormente bajamos los archivos necesarios para generar el plan en el Jmeter (figura 51).

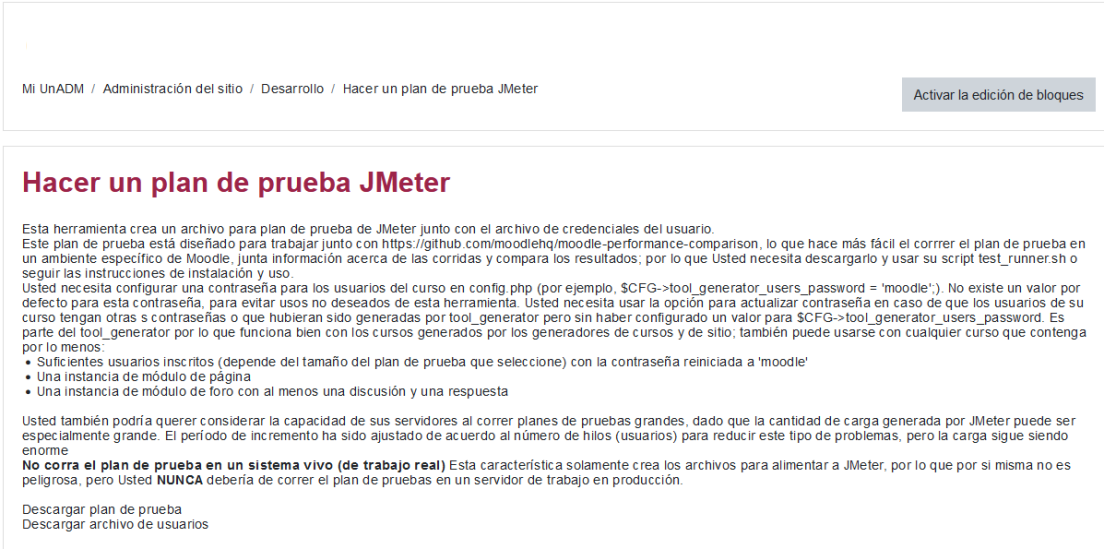


Figura 51 Creación plan de prueba Moodle

Posteriormente tenemos estos dos archivos (figura 52).

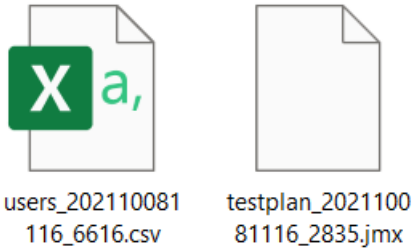


Figura 52 Archivos generados prueba Moodle

Los resultados en la prueba realizada en un equipo personal y fuera de la VPN institucional fueron los siguientes (figura 53):

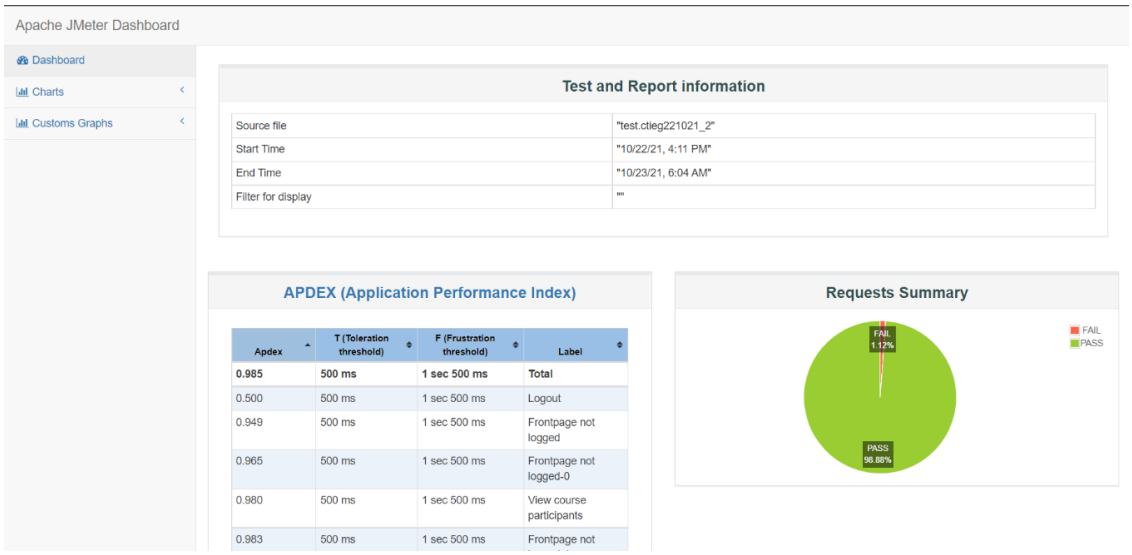


Figura 53 Resultados de la Prueba Jmeter Moodle

Como se puede apreciar la prueba termino satisfactoriamente resolviendo el 97.55% de las peticiones.

Los errores que se obtuvieron en la prueba que fue en el 2.45% de las peticiones al servidor, solo en el texto que se espera cuando se hace ***Login*** en el sitio este es un error en la aserción que configuró el script dentro de Moodle y finalmente en un ***sesskey*** que no fue capturado correctamente (figura 54).

Errores

Tipo de error	Numero de errores	% en errores	% en todas las muestras
Prueba fallida: se espera que el texto contenga / Ha iniciado sesión como /	600	85,71%	2,10%
Código de respuesta no HTTP: java.net.URISyntaxException / Mensaje de respuesta no HTTP: carácter ilegal en la consulta en el índice 52: http://ctie.unadmexico.mx/login/logout.php?sesskey=${SESSION_SESSKEY}	100	14,29%	0,35%

Figura 54 Errores durante la prueba Jmeter

A continuación, se muestran los hits por segundo obtenidos durante la prueba en el siguiente gráfico el máximo fue de 8 hits por segundo (figura 55).

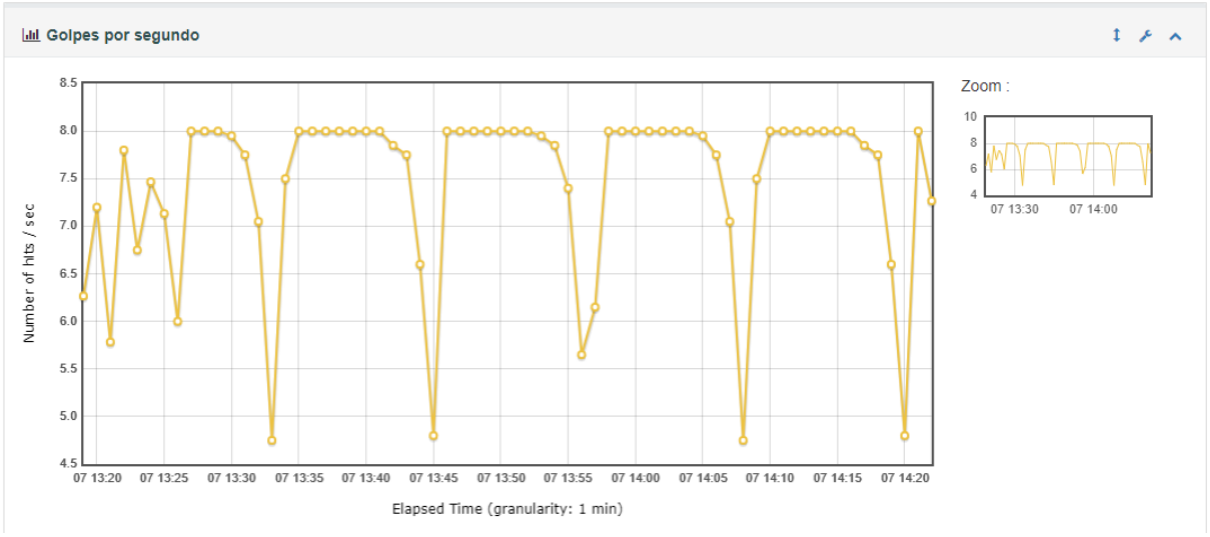


Figura 55 Hits por segundo

El siguiente gráfico muestra el total de transacciones resueltas por segundo durante la prueba el máximo fue de 8 transacciones (figura 56).

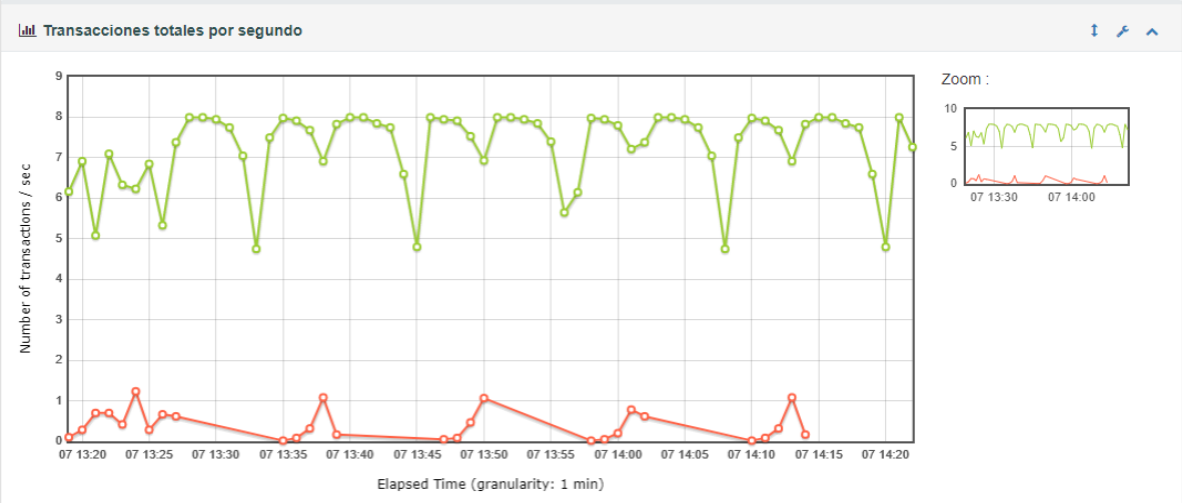


Figura 56 Máximo de transacciones

Este gráfico muestra la media de solicitudes resueltas por segundo llegando a un máximo de 31 solicitudes en 52 milisegundos (figura 57).

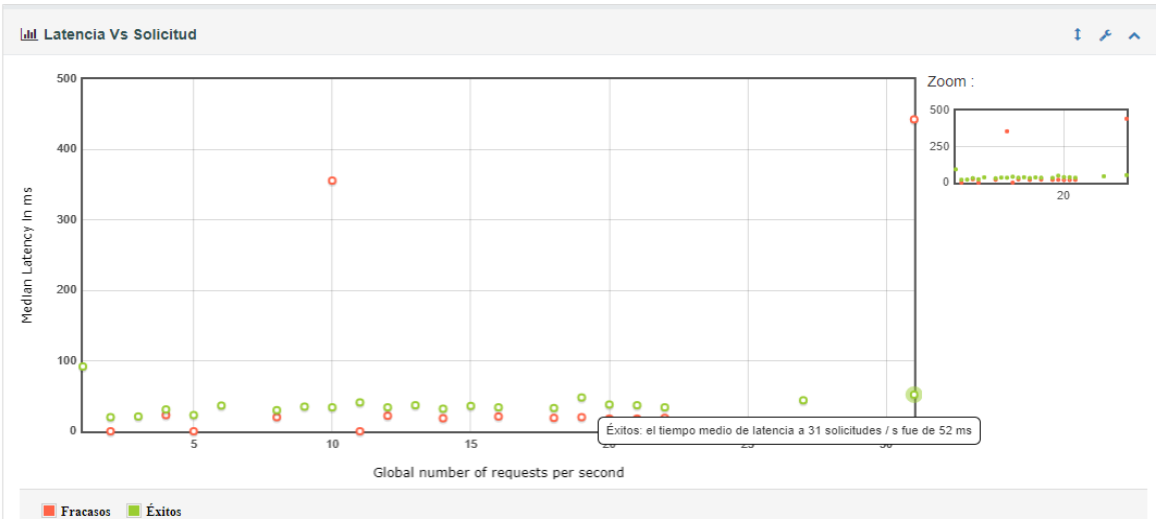


Figura 57 Tiempo máximo de solicitudes

5. Conclusiones

El objetivo principal de este trabajo de tesis se ha cumplido ya que se ha desarrollado un clúster de base de datos para el mejoramiento del servicio de las aulas (Moodle), el cual cumple con las características de alta disponibilidad, replicación síncrona entre todos los nodos del clúster lo que significa que todos los nodos tienen una copia exacta obteniendo una consistencia de los datos (si un nodo falla es redirigido al que esté disponible), lo que favorece a la plataforma al tener un mejor rendimiento en los horarios altamente demandantes, lo que le da mucha agilidad y rapidez en las consultas a la base de datos.

La facilidad de gestión del clúster se da mediante el balanceador MaxScale favoreciendo la distribución de las consultas de manera equilibrada lo cual nos garantiza menor tiempo de espera dentro de la plataforma, lo cual si se hace mediante la distribución en todos los nodos del clúster aumentando la capacidad de procesamiento y reduciendo la sobre carga en un nodo. Una de las características más importantes de este balanceador, la capacidad de detectar cuando un nodo no esté disponible de manera automática es redireccionado hacia otro nodo y mantenido la continuidad en el servicio además de ser un balanceador de carga cuenta con una herramienta de monitoreo en tiempo real lo que nos ayuda a tener una buena administración en el clúster asi como las estadísticas de cada nodo.

Como alternativas a este problema existen muchas, algunas de software privativo las cuales son muy costosas (el coste se eleva por la cantidad de procesamiento) y otras de software libre. Para efectos de este proyecto se eligió una alternativa que fuese de software libre y de fácil implementación y mantenimiento. Una de las principales razones por la cual se eligió esta alternativa fue su funcionamiento lo cual cumple con las necesidades de las instituciones de educación superior a distancia y es altamente compatible con la plataforma de Moodle y su motor INNODB.

Al implementar el clúster surgieron algunos inconvenientes, dado que es software libre dentro de los foros existe poca información de la instalación asi como de los paquetes extra para concluir la configuración, la sincronización de los nodos fue uno de los tropiezos estos se instalaban de manera normal, pero al sincronizar cada uno de éstos con el nodo maestro no era posible. Se tuvo que volver a reinstalar el software he investigar los comandos y configuraciones adecuadas para que se sincronizaran dado que existen múltiples configuraciones dependiendo de la distribución del sistema operativo.

Además, se realizaron varias pruebas para conectar el balanceador estas de compatibilidad en principio se usó Haproxy, pero no se podía tener un control de los nodos; es decir, desde su propia interfaz los nodos no se podían reiniciar o encender. Por tal motivo, se trató de elegir un balanceador compatible con MariaDB Galera.

Este proyecto de mejora no termina aquí. En un futuro se pretenden agregar más nodos y tener varios esquemas de base de datos de otros sistemas. Asi mismo, teniendo un Data Warehouse (centro de datos), para que otras aplicaciones convivan con el clúster como son: desarrollo (QA) o para hacer análisis de datos (big data), Inclusive tener un nodo exclusivo para respaldos es decir configurarlo de tal manera que se hagan respaldos en frio (la copia se hace en horas donde no hay mucha carga en la base de datos) y en caliente (generalmente esta copia se hace cuando hay mucha carga dentro de la base de datos, sin embargo esta copia no es completa dado que a lo largo del día existirán más cambios). Con esta misma estructura podemos implementar un clúster web en conjunto con este clúster de base de datos, lo que como resultado nos dará un sistema robusto, altamente disponible y tolerante a fallos. MariaDB Galera se puede instalar en otras distribuciones de Linux y las configuraciones dependerán de las necesidades del problema.

Partiendo de esta solución, se puede implementar en diversas instituciones educativas u organizaciones ajustando a sus distintas problemáticas. Para el caso de la Universidad Autónoma de la Ciudad de México (UACM), el mejoramiento de un sistema de clúster podría ayudar a agilizar las inscripciones de los estudiantes cada semestre, justamente cuando el sistema se ve más afectado y que causa molestia a la comunidad universitaria; esta propuesta podría ayudar a solventar este problema.

Glosario

.Net CLR (Common Language Runtime): Es el núcleo de la plataforma de desarrollo .NET de Microsoft. Es el entorno de ejecución de tiempo de la plataforma .NET, que se encarga de ejecutar el código compilado en lenguajes compatibles con .NET, como C#, Visual Basic, F#, entre otros.

La CLR es responsable de la administración de la memoria, la verificación de tipos, la seguridad, la gestión de excepciones, la gestión de hilos, entre otras funciones fundamentales para el correcto funcionamiento de las aplicaciones .NET. La CLR proporciona una plataforma común para todos los lenguajes de programación compatibles con .NET, lo que permite que los desarrolladores utilicen el lenguaje de programación que prefieran para desarrollar aplicaciones, se utiliza como una capa intermedia entre el código fuente y el sistema operativo, permitiendo a las aplicaciones .NET ser ejecutadas en diferentes sistemas operativos y plataformas. Es una de las principales características de .NET, ya que proporciona una plataforma de desarrollo de alto nivel para crear aplicaciones seguras, escalables y fáciles de mantener.

C: Es un lenguaje de programación de alto nivel, estandarizado, de uso general y procedural, desarrollado en los años 70 por Dennis Ritchie en Bell Labs. Es uno de los lenguajes de programación más populares y ampliamente utilizado en el mundo, con una gran variedad de aplicaciones, desde sistemas operativos, control de sistemas, aplicaciones de escritorio, dispositivos móviles, videojuegos, entre otros, cuenta con una gran cantidad de bibliotecas estandarizadas y una gran comunidad de desarrolladores, lo que permite una gran cantidad de recursos y documentación en línea. Es un lenguaje de programación muy potente y versátil, con una gran cantidad de aplicaciones en diferentes ámbitos, desde sistemas operativos hasta aplicaciones de escritorio y dispositivos móviles.

C++: Es un lenguaje de programación de alto nivel, estandarizado, de uso general y orientado a objetos, que fue desarrollado en los años 80 por Bjarne Stroustrup en Bell Labs como una extensión del lenguaje C. Es uno de los lenguajes de programación más populares y ampliamente utilizado en el mundo, con una gran variedad de aplicaciones, desde sistemas operativos, control de sistemas, aplicaciones de escritorio, dispositivos móviles, videojuegos, entre otros.

Con JavaScript, los desarrolladores pueden crear funcionalidades como formularios de validación, menús desplegables, sliders de imágenes, mapas interactivos, entre otros. Es un lenguaje de programación versátil y se ha vuelto muy popular entre los desarrolladores web, ya que se puede utilizar tanto en el lado del cliente como en el lado del servidor (con Node.js).

DNS (Domain Name System): es un sistema de nombres de dominio que se utiliza para traducir nombres de dominio legibles por humanos en direcciones IP numéricas que los dispositivos de red pueden entender. Es una especie de ""directorio telefónico"" de internet, que se encarga de asociar nombres de dominio con direcciones IP.

El DNS funciona como una jerarquía de servidores, donde cada nivel de la jerarquía tiene una función específica. Los servidores DNS raíz son los que contienen información sobre los dominios de nivel superior (como .com, .org, .edu, etc.). Los servidores DNS de nivel superior contienen información sobre los dominios de segundo nivel (como ejemplo.com) y los

servidores DNS de segundo nivel contienen información sobre los hosts (como www.ejemplo.com).

Cuando un usuario introduce un nombre de dominio en su navegador web (por ejemplo, www.ejemplo.com), su dispositivo realiza una solicitud al sistema DNS para obtener la dirección IP correspondiente. La solicitud es enviada a través de la jerarquía de servidores DNS hasta que se encuentra la dirección IP correcta. Una vez obtenida la dirección IP, el navegador web puede conectarse al servidor web correspondiente y descargar la página solicitada.

FIND y GET: Son dos sentencias comunes utilizadas en lenguajes de programación para buscar y recuperar información de una estructura de datos.

La sentencia FIND se utiliza para buscar un elemento específico en una estructura de datos, como una lista, un arreglo o una tabla. La sentencia FIND busca el elemento especificado en la estructura de datos y devuelve la posición en la que se encuentra o un valor que indica que el elemento no se ha encontrado. La sintaxis para utilizar la sentencia FIND varía según el lenguaje de programación.

La sentencia GET es utilizada para recuperar un valor o un conjunto de valores de una estructura de datos, como una lista, un arreglo o una tabla. La sentencia GET es similar a la sentencia FIND, pero en lugar de devolver la posición del elemento buscado, devuelve el valor o los valores del elemento. La sintaxis para utilizar la sentencia GET también varía según el lenguaje de programación.

Firewall: Es un sistema de seguridad de red que actúa como una barrera entre una red interna y una red externa, como Internet. Su principal función es controlar el tráfico de red que entra y sale de la red, permitiendo o bloqueado el tráfico según un conjunto de reglas predefinidas.

FTP (File Transfer Protocol): Es un protocolo de red estándar utilizado para transferir archivos entre dispositivos conectados a una red, como computadoras o servidores. FTP permite a los usuarios subir y descargar archivos de un servidor remoto, así como también la gestión de archivos y carpetas en el servidor.

FTP utiliza dos canales de comunicación: un canal de control y un canal de datos. El canal de control se utiliza para establecer la conexión entre el cliente y el servidor y para enviar comandos y respuestas. El canal de datos se utiliza para transmitir los archivos, también tiene dos modos de operación, "modo activo" y "modo pasivo". En el modo activo, el servidor se encarga de establecer la conexión para el canal de datos con el cliente, mientras que, en el modo pasivo, es el cliente el que establece la conexión. El modo pasivo es más seguro y es utilizado en ambientes de red detrás de un firewall.

Existen varias aplicaciones de cliente y servidor de FTP disponibles para diferentes sistemas operativos. Los clientes de FTP populares incluyen FileZilla, WinSCP y CuteFTP.

GNU GPL (General Public License): es una licencia de software libre desarrollada por la Free Software Foundation (FSF) y es también conocida como la Licencia Pública General de GNU. La GPL es una licencia de derecho de autor que se aplica a programas de computadora y otros tipos de obras intelectuales.

La GPL permite a los usuarios utilizar, estudiar, modificar y distribuir el software libremente, siempre y cuando se cumplan ciertas condiciones. Estas condiciones incluyen la obligación de proporcionar el código fuente del software junto con cualquier versión modificada del software, y la obligación de incluir una copia de la GPL con cualquier distribución del software.

Grid (grid computing): Es una técnica de computación distribuida que permite unir recursos de computación de diferentes lugares para trabajar juntos como una sola entidad, con el objetivo de resolver problemas complejos o realizar tareas de gran escala.

La computación en grid se basa en la idea de utilizar recursos de computación desconectados, como computadoras personales, servidores, centros de datos y supercomputadoras, que trabajan juntos para resolver un problema o realizar una tarea. Estos recursos son organizados en una red para trabajar de manera coordinada y pueden ser controlados y administrados de forma centralizada, es utilizada para aplicaciones que requieren un gran rendimiento, como la simulación científica, el procesamiento de imágenes, el análisis financiero, el análisis de datos, la minería de datos, entre otras. Es una técnica de computación distribuida muy utilizada en ciencia e investigación, ya que permite a los investigadores acceder a recursos de computación de gran escala para resolver problemas científicos complejos.

HTTP (Hypertext Transfer Protocol): Es un protocolo de red utilizado para transmitir datos a través de la World Wide Web (WWW). Es el protocolo que permite la comunicación entre un servidor web y un navegador web, está basado en peticiones y respuestas. Cuando un usuario introduce una URL en su navegador web, el navegador envía una petición HTTP al servidor web correspondiente solicitando la página web. El servidor web, a su vez, envía una respuesta HTTP que contiene la página web solicitada.

HTTP es un protocolo sin estado, lo que significa que cada petición y respuesta HTTP son independientes entre sí y no mantienen ninguna información sobre las peticiones anteriores. Sin embargo, existe un protocolo llamado HTTPS (HTTP Secure) que proporciona una capa adicional de seguridad mediante el uso de certificados digitales y encriptación.

HTTPS (HTTP Secure): Es un protocolo de red basado en HTTP (Hypertext Transfer Protocol) que proporciona una capa adicional de seguridad para las comunicaciones a través de la World Wide Web (WWW). Es utilizado para asegurar que la información transmitida entre un navegador web y un servidor web es privada y segura.

La principal diferencia entre HTTP y HTTPS es que HTTPS utiliza un sistema de encriptación para proteger la información transmitida. El protocolo de encriptación utilizado en HTTPS es SSL (Secure Sockets Layer) o TLS (Transport Layer Security). Estos protocolos utilizan un mecanismo de certificado digital para autenticar al servidor web y para garantizar que la información transmitida es privada y segura.

Cuando un usuario accede a un sitio web utilizando HTTPS, el navegador web verifica el certificado digital del servidor web antes de establecer la conexión. Si el certificado es válido, se establece una conexión segura y toda la información transmitida a través de la conexión es encriptada.

IBM i: (anteriormente conocido como AS/400 o i5/OS) es un sistema operativo desarrollado y comercializado por IBM. Es un sistema operativo empresarial diseñado para ser utilizado en servidores IBM Power Systems y es especialmente popular en entornos de negocios, como contabilidad, inventario y gestión de relaciones con los clientes. IBM i ofrece una interfaz de línea de comandos y una interfaz gráfica de usuario, y cuenta con una amplia variedad de

herramientas y aplicaciones para el desarrollo de software y la gestión de bases de datos. Es conocido por su estabilidad y escalabilidad, y es utilizado principalmente en entornos de negocios de tamaño mediano y grande.

IBM Z: Es una familia de mainframes desarrollado y comercializado por IBM. Los mainframes son computadoras de gran capacidad y rendimiento, diseñadas para manejar grandes cargas de trabajo y transacciones en entornos críticos de negocios. IBM Z incluye modelos como IBM z15 y IBM z14, que proporcionan una gran capacidad de procesamiento, almacenamiento y seguridad. Los mainframes IBM Z son utilizados en una variedad de aplicaciones críticas de negocios, como banca, seguros, retail, servicios públicos y ciencia de datos.

Java: Es un lenguaje de programación de alto nivel y orientado a objetos desarrollado por Sun Microsystems (ahora propiedad de Oracle) en los años 90. Es un lenguaje de programación multiplataforma, lo que significa que el código escrito en Java puede ejecutarse en diferentes sistemas operativos y plataformas sin necesidad de ser recompilado, es utilizado en una amplia variedad de aplicaciones, como el desarrollo de aplicaciones de escritorio, aplicaciones web, sistemas empresariales, aplicaciones móviles, videojuegos, entre otros. Es uno de los lenguajes de programación más populares y ampliamente utilizado en el mundo.

Java cuenta con una gran cantidad de herramientas y bibliotecas estandarizadas, como el entorno de desarrollo integrado (IDE) Eclipse o NetBeans, así como también con una gran comunidad de desarrolladores y una amplia documentación y recursos en línea."

JavaScript: Es un lenguaje de programación interpretado y de alto nivel. Es principalmente utilizado para agregar interactividad y dinamismo a las páginas web, pero también se utiliza en aplicaciones de servidor, app móvil, Internet de las cosas y más. JavaScript se ejecuta en el navegador del cliente y puede acceder y manipular elementos del documento HTML o del Document Object Model (DOM) de una página web.

JSON es similar a XML en el sentido de que ambos son utilizados para describir y transportar datos en un formato estructurado. Sin embargo, a diferencia de XML, JSON tiene una sintaxis más simple y legible, y se basa en un subconjunto de las características de JavaScript, lo que lo hace más fácil de leer y escribir tanto para humanos como para máquinas, se utiliza en una variedad de aplicaciones, como el intercambio de datos entre el lado del cliente y el servidor en aplicaciones web, el almacenamiento y el intercambio de datos en bases de datos, y el intercambio de datos entre diferentes aplicaciones y sistemas. Es un formato de datos muy popular debido a su simplicidad, rapidez de procesamiento y compatibilidad con una variedad de lenguajes de programación.

JSON: (JavaScript Object Notation) es un formato de texto ligero para intercambio de datos. Es un estándar abierto y ampliamente utilizado para el intercambio de datos en internet y en sistemas de aplicaciones.

LAN (Local Area Network): es una red de computadoras que se utiliza para conectar dispositivos en un área geográfica limitada, como un edificio, una oficina o un campus universitario. Una LAN es una red privada que se utiliza para compartir recursos, como impresoras, archivos y aplicaciones, entre los usuarios conectados a ella, se construye utilizando dispositivos de red, como switches, routers y hubs, y utiliza protocolos de red estándar, como TCP/IP, para comunicarse. Pueden ser conectadas a otras redes, como WAN (Wide Area Network) o Internet, para permitir la comunicación entre dispositivos en diferentes áreas geográficas.

Las LANs son ampliamente utilizadas en empresas, organizaciones gubernamentales y educativas para compartir recursos y facilitar la comunicación entre los usuarios. Además, también son utilizadas en hogares y pequeñas oficinas para compartir una conexión a Internet y recursos de red, como impresoras y archivos.

LDAP (Lightweight Directory Access Protocol): Es un protocolo de red utilizado para acceder y administrar bases de datos de directorios. Una base de datos de directorio es una colección de información organizada jerárquicamente, como un árbol de carpetas en un sistema de archivos. Las entradas en una base de datos de directorio pueden incluir información como nombres de usuario, contraseñas, direcciones de correo electrónico y grupos de usuarios. LDAP se utiliza principalmente para almacenar información de usuarios y grupos de una red, como nombres de usuario, contraseñas, direcciones de correo electrónico y permisos de acceso. También se utiliza para almacenar información de recursos de red, como servidores y dispositivos de red.

Los clientes LDAP, como los programas de correo electrónico o los servicios de autenticación, pueden utilizar LDAP para buscar y recuperar información de una base de datos de directorio. Los administradores pueden utilizar herramientas LDAP para crear, modificar y eliminar entradas en la base de datos de directorio.

Linux: Es un sistema operativo de código abierto basado en Unix. Fue creado por Linus Torvalds en 1991 y se ha convertido en uno de los sistemas operativos más populares para servidores y dispositivos móviles.

Los firewalls pueden ser configurados para permitir o bloquear ciertos tipos de tráfico de red, como el tráfico HTTP (web), el tráfico de correo electrónico, el tráfico de FTP, entre otros. También pueden ser configurados para permitir o bloquear el tráfico de red proveniente de direcciones IP específicas o de redes específicas.

Los firewalls pueden ser software o hardware. Los firewalls de software se ejecutan en un ordenador y controlan el tráfico de red en ese dispositivo. Los firewalls de hardware son dispositivos independientes que se conectan a la red y controlan el tráfico de red en toda la red.

Nagios: Es un sistema de monitorización de redes y servidores de código abierto, similar a Zabbix. Permite a los administradores de sistemas monitorear la disponibilidad y el rendimiento de sus servidores, redes y aplicaciones, y recibir alertas en caso de problemas, utiliza una arquitectura de cliente-servidor, donde un servidor Nagios se encarga de recolectar y almacenar la información de monitorización, y uno o varios clientes Nagios se encargan de recolectar la información de los dispositivos y servidores que se están monitoreando.

NoSQL: Es un término genérico que se refiere a los sistemas de bases de datos que no se ajustan al modelo relacional de bases de datos. A diferencia de las bases de datos relacionales, como MySQL o Oracle, que utilizan tablas con filas y columnas para almacenar datos, los sistemas NoSQL utilizan estructuras de datos diferentes, como documentos, gráficos o clave-valor, para manejar grandes volúmenes de datos no estructurados y/o en rápido crecimiento.

ODBC.Net: Es una interfaz de programación de aplicaciones (API) desarrollada por Microsoft que permite a las aplicaciones de .NET acceder y manipular datos en una variedad de bases de datos utilizando el estándar de conectividad ODBC (Open Database Connectivity).

ODBC es un estándar de la industria que proporciona una forma estandarizada de acceder a diferentes bases de datos relacionales, independientemente del sistema operativo o la plataforma utilizada. ODBC.NET es una implementación de ODBC para el entorno de desarrollo .NET de Microsoft, que permite a las aplicaciones de .NET acceder a bases de datos relacionales utilizando el lenguaje de programación C# o VB.NET.

ODBC.NET proporciona una interfaz de programación de aplicaciones (API) para acceder y manipular datos en una base de datos utilizando operaciones estándar de ODBC, como la creación de conexiones, la ejecución de consultas y actualizaciones, y el manejo de transacciones. Permite a los desarrolladores utilizar las ventajas de .NET, como la seguridad y la administración de memoria, mientras interactúan con bases de datos.

ODP.Net: Es una interfaz de programación de aplicaciones (API) desarrollada por Oracle que permite a las aplicaciones de .NET acceder y manipular datos en una base de datos Oracle utilizando el lenguaje de programación C# o VB.NET.

ODP.NET es una implementación de Oracle para el entorno de desarrollo .NET de Microsoft. Es una interfaz de programación de aplicaciones (API) para acceder a bases de datos Oracle, proporciona una conexión directa a la base de datos Oracle desde aplicaciones .NET, ODP.NET proporciona una interfaz de programación de aplicaciones (API) para acceder y manipular datos en una base de datos Oracle utilizando operaciones estándar de Oracle, como la creación de conexiones, la ejecución de consultas y actualizaciones, y el manejo de transacciones. Permite a los desarrolladores utilizar las ventajas de .NET, como la seguridad y la administración de memoria, mientras interactúan con bases de datos Oracle.

OLE.Net: Es una tecnología desarrollada por Microsoft que permite la interconexión y la interoperabilidad entre diferentes aplicaciones y sistemas utilizando la tecnología OLE (Object Linking and Embedding).

OLE es un sistema de Microsoft que permite la creación de objetos compuestos, permite la integración de diferentes tipos de objetos en una sola aplicación, como texto, imágenes, gráficos, audio, video y hojas de cálculo. Es una tecnología muy utilizada para la creación de documentos compuestos, como los creados en Microsoft Office.

OLE.NET es una implementación de OLE para el entorno de desarrollo .NET de Microsoft, que permite a las aplicaciones de .NET interactuar con objetos OLE y utilizar los servicios de OLE para crear objetos compuestos y compartirlos entre aplicaciones. Esto permite a los desarrolladores utilizar las ventajas de .NET, como la seguridad y la administración de memoria, mientras interactúan con objetos OLE.

PHP (Hypertext Preprocessor): Es un lenguaje de programación de código abierto, principalmente utilizado para el desarrollo de aplicaciones web. Fue creado originalmente en 1995 por Rasmus Lerdorf y es ampliamente utilizado en la creación de sitios web dinámicos y aplicaciones web.

PHP es un lenguaje de script del lado del servidor, lo que significa que el código PHP se ejecuta en el servidor web y se envía al navegador web en forma de HTML estático. PHP es compatible con una gran variedad de bases de datos y sistemas operativos, y se puede integrar fácilmente con HTML, CSS y JavaScript, es utilizado para crear aplicaciones web complejas, como sistemas de gestión de contenido (CMS), foros, tiendas en línea, aplicaciones de redes sociales, entre otros. PHP también es utilizado para el procesamiento de formularios,

la validación de datos, la autenticación de usuarios y otras funciones comunes en aplicaciones web.

PHP-FPM (PHP FastCGI Process Manager): Es un módulo de PHP que proporciona un mecanismo de gestión de procesos para PHP. FPM es una alternativa al método tradicional de ejecutar PHP como un módulo de Apache o como un CGI.

FPM se utiliza para mejorar el rendimiento y la escalabilidad de aplicaciones PHP al manejar procesos PHP de manera eficiente. FPM crea y administra un conjunto de procesos PHP que manejan las solicitudes de los clientes. Cada proceso PHP es independiente y puede manejar varias solicitudes al mismo tiempo.

FPM también proporciona una serie de características avanzadas, como la capacidad de establecer límites de recursos para los procesos PHP, la detección automática de procesos muertos y la posibilidad de priorizar procesos. Esto ayuda a mejorar la estabilidad y el rendimiento de las aplicaciones PHP y a prevenir problemas como el agotamiento de recursos.

PL SQL: Es un lenguaje de programación desarrollado y utilizado por Oracle para su plataforma de bases de datos. Es un lenguaje de programación específico del sistema (SQL) utilizado para interactuar con bases de datos relacionales y es altamente integrado con el sistema de gestión de bases de datos de Oracle, permite a los desarrolladores escribir bloques de código SQL y procedimientos almacenados, funciones, desencadenadores, entre otros, para realizar tareas de administración de bases de datos, como crear y modificar tablas, índices y procedimientos almacenados. También se pueden realizar operaciones de manipulación de datos como insertar, actualizar y eliminar registros, así como también consultas complejas para recuperar datos de la base de datos.

PL/SQL tiene características de un lenguaje de programación general, como la capacidad de controlar el flujo del programa, la manipulación de variables, excepciones y manejo de transacciones, lo que lo hace muy potente para desarrollar aplicaciones complejas y escalables. Es muy utilizado en entornos empresariales y en aplicaciones de bases de datos de gran escala, debido a su gran rendimiento y escalabilidad.

Plataformas NT: Son un conjunto de sistemas operativos desarrollados y comercializados por Microsoft. El nombre "NT" significa "New Technology" y se refiere a la arquitectura de núcleo de estos sistemas operativos, que es diferente a la arquitectura de los sistemas operativos anteriores de Microsoft, como MS-DOS.

Round-Robin: Es un algoritmo de balanceo de carga utilizado en redes y sistemas para distribuir la carga de trabajo entre varios dispositivos o servidores. El objetivo de Round-Robin es asegurar que ningún dispositivo o servidor se sobrecargue mientras se distribuye de manera equitativa la carga de trabajo entre todos los dispositivos o servidores disponibles.

La forma en que funciona Round-Robin es que se lleva un registro de los dispositivos o servidores disponibles y se les asigna una tarea en un orden rotativo. Por ejemplo, si hay tres servidores disponibles: A, B y C, y hay que asignar una tarea a un servidor, primero se asignaría a A, luego a B, y finalmente a C. A continuación, se vuelve a comenzar con A, y así sucesivamente.

Round-Robin es un algoritmo simple, fácil de implementar y escalable, es utilizado en varios casos, como en DNS, en servidores web, en redes de telecomunicaciones, entre otros. Es una alternativa popular a otros algoritmos de balanceo de carga como el algoritmo de equilibrio de

carga basado en el rendimiento o el algoritmo de equilibrio de carga basado en el número de conexiones.

SELinux (Security-Enhanced Linux): es un sistema de seguridad de código abierto desarrollado por el gobierno de los Estados Unidos y diseñado para proporcionar un mayor nivel de seguridad en sistemas Linux. SELinux proporciona un sistema de control de acceso basado en políticas, que permite restringir los permisos de los procesos y usuarios en el sistema, también establece un conjunto de reglas y políticas que definen qué procesos y usuarios tienen acceso a qué recursos del sistema. Estas reglas y políticas son utilizadas para controlar el acceso a archivos, directorios, servicios de red y otras funciones críticas del sistema, se integra en el núcleo de Linux y trabaja de manera transparente para el usuario, con lo cual no requiere configuraciones adicionales, pero puede ser configurado para adaptarse a las necesidades de seguridad de una organización.

SELinux es una herramienta poderosa para mejorar la seguridad de un sistema Linux, ya que proporciona un control de acceso basado en políticas que ayuda a prevenir ataques y violaciones de seguridad. Sin embargo, algunas veces su configuración y administración puede ser compleja y requiere un conocimiento técnico experto.

SQLJ: Es una tecnología que permite a los desarrolladores escribir código Java que se ejecuta directamente en una base de datos relacional utilizando SQL (Structured Query Language) estándar. Es una especificación de la industria que proporciona una interfaz de programación de aplicaciones (API) para integrar Java con bases de datos relacionales.

SQLJ permite a los desarrolladores escribir código Java que incluye declaraciones SQL directamente en el código fuente, lo que permite que el código Java sea compilado y traducido a código SQL estándar para su ejecución en una base de datos. Esto permite a los desarrolladores utilizar las ventajas de Java, como la portabilidad y la seguridad, mientras interactúan con bases de datos, proporciona una interfaz de programación de aplicaciones (API) para integrar Java con bases de datos relacionales, así como también un conjunto de herramientas para traducir y ejecutar el código SQLJ en una base de datos relacional. Es compatible con muchas bases de datos relacionales populares, como Oracle, DB2, SQL Server, entre otros.

TCP (Transmission Control Protocol): es un protocolo de red de comunicaciones orientado a conexión, que garantiza la confiabilidad de los datos transmitidos a través de una red. Es uno de los protocolos fundamentales de Internet y se utiliza para transmitir datos en la capa de transporte del Modelo OSI (Open Systems Interconnection).

TCP proporciona un servicio de transmisión fiable, lo que significa que se asegura de que los datos transmitidos lleguen correctamente al destinatario y en el orden en el que fueron enviados. Para lograr esto, TCP utiliza mecanismos de control de flujo y detección de errores para asegurar la entrega confiable de los datos, también establece un canal de comunicación entre los dispositivos antes de comenzar a transmitir los datos, lo que se conoce como establecimiento de conexión. Una vez establecida la conexión, los dispositivos pueden comenzar a transmitir datos de manera confiable. Al finalizar la transmisión, se procede a cerrar la conexión.

TCP es ampliamente utilizado en aplicaciones como el correo electrónico, el protocolo FTP, el protocolo SSH y el protocolo Telnet. Es utilizado en conjunto con IP (Internet Protocol) para formar el protocolo TCP/IP, el protocolo de red estándar utilizado en Internet.

Transact SQL o T-SQL: Es un lenguaje de programación desarrollado y utilizado por Microsoft para su plataforma de bases de datos SQL Server. Es un lenguaje de programación específico del sistema (SQL) utilizado para interactuar con bases de datos relacionales. T-SQL es un lenguaje de programación estandarizado para bases de datos relacionales, basado en SQL, pero con extensiones y mejoras específicas para el uso en SQL Server.

Los sistemas operativos de plataforma NT incluyen:

- Windows NT: La primera versión de la plataforma NT, lanzada en 1993. Incluye características como un sistema de archivos protegido, soporte para múltiples usuarios y soporte para una variedad de hardware.
- Windows 2000: Lanzado en 2000, incluye características adicionales para las redes y el soporte para servidores.
- Windows XP: Lanzado en 2001, es una versión de escritorio de la plataforma NT y es uno de los sistemas operativos más populares de Microsoft.
- Windows Server 2003: Lanzado en 2003, es una versión de servidor de la plataforma NT.
- Windows Server 2008: Lanzado en 2008, incluye características adicionales para las redes, seguridad y virtualización.
- Windows Server 2012: Lanzado en 2012, incluye características adicionales para la nube y el soporte para dispositivos móviles.
- Windows Server 2016: Lanzado en 2016, incluye características adicionales para la seguridad y la gestión de datos.
- Windows Server 2019: Lanzado en 2019, incluye características adicionales para la seguridad, la privacidad y el soporte para inteligencia artificial y análisis de datos.

Aunque la plataforma NT ha sido reemplazada por sistemas operativos más modernos como Windows 10, aún sigue siendo utilizado en algunos entornos empresariales y de servidores.

Transact SQL o T-SQL: Es un lenguaje de programación desarrollado y utilizado por Microsoft para su plataforma de bases de datos SQL Server. Es un lenguaje de programación específico del sistema (SQL) utilizado para interactuar con bases de datos relacionales. T-SQL es un lenguaje de programación estandarizado para bases de datos relacionales, basado en SQL, pero con extensiones y mejoras específicas para el uso en SQL Server.

Con T-SQL, los desarrolladores pueden crear procedimientos almacenados, funciones, desencadenadores, vistas y tablas temporales, así como también realizar tareas de administración de bases de datos, como crear y modificar tablas, índices y procedimientos almacenados. También se pueden realizar operaciones de manipulación de datos como insertar, actualizar y eliminar registros, así como también consultas complejas para recuperar datos de la base de datos.

UDP (User Datagram Protocol): Es un protocolo de red sin conexión que se utiliza para transmitir datos a través de una red. Es uno de los protocolos fundamentales de Internet y se utiliza para transmitir datos en la capa de transporte del Modelo OSI (Open Systems Interconnection).

A diferencia del protocolo TCP, UDP no proporciona un servicio de transmisión fiable. Los datagramas UDP son transmitidos sin establecer una conexión previa y sin garantizar la entrega de los datos. Es responsabilidad del programa de aplicación asegurar la confiabilidad

de la transmisión de datos si es necesario. Por lo tanto, UDP es más ligero y rápido que TCP, pero tiene una menor fiabilidad.

UDP es utilizado en aplicaciones en las que la entrega confiable de datos no es crítica, como el protocolo DNS, el protocolo DHCP, el protocolo SNMP y el protocolo de tiempo de protocolo NTP. También es utilizado en aplicaciones en las que la velocidad es más importante que la fiabilidad, como el streaming de video y audio en tiempo real.

Unix: Es un sistema operativo multiusuario, multitarea y de código abierto. Fue desarrollado a finales de los años 60 por Bell Labs y ha sido influyente en la evolución de los sistemas operativos modernos, especialmente en lo que respecta a la filosofía de diseño y la interfaz de línea de comandos. Varias versiones de Unix, como Linux y macOS, son ampliamente utilizadas en servidores, computadoras de escritorio y dispositivos móviles**.**

VM IBM VM (Virtual Machine): es un sistema operativo de máquina virtual desarrollado y comercializado por IBM. Fue introducido en 1972 y se ejecuta en mainframes IBM System/370 y sus sucesores. IBM VM proporciona un entorno de máquina virtual que permite a varios sistemas operativos, conocidos como guest, ejecutarse simultáneamente en una única máquina física, conocida como host. Es utilizado para proporcionar compatibilidad con diferentes sistemas operativos y aplicaciones en un solo sistema físico.

VSE IBM VSE (Virtual Storage Extended): es un sistema operativo desarrollado y comercializado por IBM para su familia de mainframes System/370 y sus sucesores. Fue introducido en 1972 y ha sido utilizado en una variedad de entornos empresariales, incluyendo transacciones, bases de datos y procesamiento de lotes. IBM VSE es conocido por su capacidad para manejar grandes cantidades de datos y transacciones en un entorno seguro y escalable. A diferencia de sistemas operativos más modernos, IBM VSE utiliza una interfaz de línea de comandos y no cuenta con una interfaz gráfica de usuario. Aunque ha sido reemplazado por sistemas operativos más modernos en muchos entornos, todavía es utilizado en algunas aplicaciones críticas de negocios.

WAN (Wide Area Network): es una red de computadoras que se utiliza para conectar dispositivos en áreas geográficas amplias, como una ciudad, un estado o incluso en diferentes países. Una WAN es una red privada o pública que se utiliza para compartir recursos y facilitar la comunicación entre los usuarios conectados a ella, se construye utilizando dispositivos de red, como routers, switches y hubs, y utiliza protocolos de red estándar, como TCP/IP, para comunicarse. Las WANs pueden conectarse a redes LAN (Local Area Network) para permitir la comunicación entre dispositivos en diferentes áreas geográficas.

Las WANs son ampliamente utilizadas en empresas, organizaciones gubernamentales y educativas para conectar sucursales, oficinas remotas y centros de datos, y compartir recursos entre ellos. También son utilizadas para conectar redes LAN en diferentes áreas geográficas, permitiendo a los usuarios acceder a recursos y aplicaciones en diferentes lugares.

WebDAV JDBC: Son un conjunto de especificaciones y tecnologías relacionadas con el acceso y la manipulación de datos en bases de datos a través de protocolos WebDAV.

WebDAV (Web Distributed Authoring and Versioning) es un protocolo estándar que permite la edición y gestión colaborativa de recursos en la web. Es un estándar HTTP que proporciona funciones para crear, editar y eliminar recursos en un servidor web de forma remota.

JDBC (Java Database Connectivity) es una interfaz de programación de aplicaciones (API) de Java que permite a las aplicaciones Java conectarse a bases de datos relacionales y ejecutar operaciones de base de datos, como consultas y actualizaciones.

WebDAV JDBC combina estas dos tecnologías para proporcionar una forma de acceder y manipular datos en bases de datos utilizando WebDAV. Permite a las aplicaciones acceder y manipular los datos en una base de datos utilizando operaciones WebDAV, como PUT, GET, POST, DELETE, etc. A través de JDBC, permitiendo a los desarrolladores utilizar las funciones estandarizadas de WebDAV para interactuar con bases de datos.

Windows: Es un sistema operativo desarrollado y comercializado por Microsoft. Fue lanzado por primera vez en 1985 y ha sido la plataforma predominante en computadoras de escritorio y portátiles desde entonces. Windows ofrece una interfaz gráfica de usuario (GUI) fácil de usar y una amplia variedad de herramientas y aplicaciones preinstaladas. Además, ofrece compatibilidad con una gran cantidad de software y dispositivos, lo que lo hace popular entre los usuarios de computadoras personales y empresas.

XML (eXtensible Markup Language): es un lenguaje de marcas utilizado para describir y transportar datos en un formato estructurado. Es un estándar abierto y ampliamente utilizado para el intercambio de datos en internet y en sistemas de aplicaciones, permite definir etiquetas personalizadas para describir los datos, lo que permite que los datos sean entendidos por aplicaciones que utilizan diferentes sistemas operativos y plataformas. Es similar al lenguaje HTML utilizado para describir la estructura de una página web, pero a diferencia de HTML, XML es utilizado para describir cualquier tipo de datos, no solo para describir la estructura de una página web. XML se utiliza en una variedad de aplicaciones, como la creación y el intercambio de documentos, el almacenamiento y el intercambio de datos en bases de datos, y el intercambio de datos entre diferentes aplicaciones y sistemas.

XQuery: Es un lenguaje de programación para consultar y manipular datos en documentos XML. Es un estándar de la industria, desarrollado por la W3C (World Wide Web Consortium) para proporcionar una forma estandarizada de acceder a los datos en documentos XML.

XQuery se basa en XPath, un lenguaje de consulta para navegar por la estructura de un documento XML, pero proporciona una sintaxis más completa para realizar operaciones como filtrado, ordenamiento y agrupamiento de datos. XQuery también permite la creación de nuevos documentos XML a partir de los datos existentes, así como también la modificación y eliminación de datos en un documento XML existente, se utiliza en una variedad de aplicaciones, como la creación y el intercambio de documentos, el almacenamiento y el intercambio de datos en bases de datos, y el intercambio de datos entre diferentes aplicaciones y sistemas. Es un lenguaje de consulta muy poderoso para trabajar con documentos XML y es ampliamente utilizado en aplicaciones web y móviles.

z/OS: IBM z/OS es un sistema operativo desarrollado y comercializado por IBM para su familia de mainframes IBM Z. Es un sistema operativo empresarial diseñado para manejar grandes cargas de trabajo y transacciones en entornos críticos de negocios. z/OS es conocido por su capacidad para manejar grandes cantidades de datos, alta disponibilidad, seguridad y escalabilidad. z/OS proporciona una interfaz de línea de comandos y herramientas para la gestión de sistemas, incluyendo la administración de recursos, seguridad y monitoreo del sistema. Además, ofrece compatibilidad con una amplia variedad de lenguajes de programación y aplicaciones, así como soporte para el sistema de archivos de IBM's Integrated File System (IFS) y el sistema de archivos Hierarchical File System (HFS). Es utilizado principalmente en entornos de negocios de tamaño mediano y grande, y en aplicaciones críticas de negocios.

Zabbix: Es un sistema de monitorización de redes y servidores de código abierto. Permite a los administradores de sistemas monitorear la disponibilidad y el rendimiento de sus servidores, redes y aplicaciones, y recibir alertas en caso de problemas.

Zabbix utiliza una arquitectura de cliente-servidor, donde un servidor Zabbix se encarga de recolectar y almacenar la información de monitorización, y uno o varios clientes Zabbix se encargan de recolectar la información de los dispositivos y servidores que se están monitoreando, es altamente escalable y se puede utilizar para monitorear desde pequeñas redes hasta grandes entornos de TI. Zabbix soporta una amplia variedad de protocolos de red y dispositivos, y proporciona una gran cantidad de opciones de alertas y notificaciones, además ofrece una interfaz web intuitiva que permite a los administradores de sistemas visualizar y analizar la información de monitorización, crear informes personalizados y configurar alertas. También proporciona una API abierta para integrarse con otras herramientas de gestión de sistemas y automatizar tareas.

Bibliografía

Abraham Silberschatz, H. F. (2014). *Fundamentos de Bases de Datos.* Mc Graw Hill.

Apache. (2022). *Jmeter*. Obtenido de https://jmeter.apache.org/

Banescu, B. (2018). *Pro SQL Server on Linux.* Apress.

Burgess, C. (2019). Choosing between centralized and decentralized databases. *Database Journal.*

Capitalone. (s.f.). *What is a Cluster? An Overview of Clustering in the Cloud*. Obtenido de https://www.capitalone.com/tech/cloud/what-is-a-cluster/

Computer Notes. (2022). *What is Distributed Database? Characteristics of Distributed Database Management System.* Obtenido de https://ecomputernotes.com/database-system/adv-database/distributed-database

Computer Notes. (2022). *What is the difference between Homogeneous and Heterogeneous Distributed DBMSs*. Obtenido de https://ecomputernotes.com/database-system/adv-database/homogeneous-and-heterogeneous-distributed-dbmss

Computer Notes. (2022). *What are the Advantages and Disadvantages of Distributed Database Management System?* Obtenido de https://ecomputernotes.com/database-system/adv-database/advantages-and-disadvantages-of-ddbms

Dancuk, M. (2 de 9 de 2021). *What Is A Database? - Everything You Need to Know | phoenixNAP KB*. Obtenido de https://phoenixnap.com/kb/what-is-a-database#:~:text=Why%20are%20Databases%20Important%3F

DataScientest. (2022). *MongoDB : todo sobre la base de datos NoSQL orientada a documentos*. Obtenido de https://datascientest.com/es/mongodb-todo-sobre-la-base-de-datos-nosql-orientada-a-documentos#:~:text=ingenier%C3%ADa%20de%20datos.-,MongoDB%20es%20una%20base%20de%20datos%20NoSQL%20orientada%20a%20documentos,almacenan%20como%20colecciones%20y%20documento

Digital Ocean. (2022). *HAProxy: The Reliable, High Performance TCP/HTTP Load Balancer*. Obtenido de https://www.digitalocean.com/community/tutorials/haproxy-the-reliable-high-performance-tcp-http-load-balancer

Digital Ocean. (2022). *MaxScale: A High-Performance Database Load Balancer* . Obtenido de https://www.digitalocean.com/community/tutorials/maxscale-a-high-performance-database-load-balancer

Digital Ocean. (2022). *NGINX: A High Performance Web Server*. Obtenido de https://www.digitalocean.com/community/tutorials/nginx-high-performance-web-server

Digital Ocean. (2022). *What is Load Balancing?* Obtenido de https://www.digitalocean.com/community/tutorials/what-is-load-balancing

F5 BIG-IP Load Balancer. (2022). Obtenido de https://www.f5.com/products/big-ip

F5 BIG-IP Load Balancer: Configuration and Best Practices . (2022). Obtenido de https://www.pluralsight.com/courses/f5-big-ip-load-balancer-configuration-best-practices

F5 BIG-IP: A Comprehensive Guide. (2022). Obtenido de https://www.packtpub.com/networking-and-servers/f5-big-ip-comprehensive-guide

Fauna. (20 de Abril de 2020). *The Why and How of Distributed Databases* . Obtenido de https://fauna.com/blog/the-why-and-how-of-distributed-databases

Fisher, M. L. (2015). *The Art of Scalability: Scalable Web Architecture, Processes, and Organizations for the Modern Enterprise.* Addison-Wesley Professional.

Foram F Kherani, P. V. (2014). Load Balancing in cloud computing. *Department of computer engineering, Lok Jagruti Kendra Institute of Technology, India*, 6.

Forma talent Bussines School. (2022). *Formación en Oracle: Historia y Características*. Obtenido de https://formatalent.com/formacion-en-oracle-historia-y-caracteristicas/

Fowler, P. J. (2015). *NoSQL Distilled.* Addison-Wesley.

Galvis, A. (27 de Octubre de 2020). *¿Qué es Maria DB? y un poco de su Historia.* Obtenido de https://mariadbuts.wordpress.com/2020/10/27/example-post-3/#:~:text=MariaDB%20surge%20a%20ra%C3%ADz%20de,a%20los%20que%20llama%20MariaDB.

geeksforgeeks. (3 de Junio de 2021). *https://www.geeksforgeeks.org/clusters-computer-organisation/.* Obtenido de https://www.geeksforgeeks.org/clusters-computer-organisation/

Gray, J. &. (1993). *Transaction processing: concepts and techniques. San Francisco: Morgan Kaufmann.*

Haproxy. (2022). *The HAProxy Handbook: A Complete Guide to Configuring and Using HAProxy*. Obtenido de https://www.haproxy.com/documentation/hapee/1-9r1/haproxy-handbook/

Hector Garcia-Molina, J. D. (2008). *Database Systems: The Complete Book.* Pearson.

Hiles, A. (2014). *The Definitive Handbook of Business Continuity Management.* John Wiley & Sons.

IBM. (16 de 1 de 2022). *Db2 11 - Introducción - ¿Qué es Db2 para z/OS para z/OS?* Obtenido de www.ibm.com: https://www.ibm.com/docs/es/db2-for-zos/11?topic=getting-started-db2-zos

IBM. (2022). *DB2 Pure Scale*. Obtenido de https://www.ibm.com/es-es/products/db2-pure-scale

IBM. (2022). *IBM Db2 – Software de gestión de datos*. Obtenido de www.ibm.com: https://www.ibm.com/es-es/db2

IBM. (2023). *¿Qué es el almacenamiento de datos?* | *IBM*. Obtenido de https://www.ibm.com/mx-es/topics/data-storage

IBM. (s.f.). *IBM DB2*. Obtenido de 2022: https://www.ibm.com/support/knowledgecenter/es/SSEPGG_11.1.0/com.ibm.db2.luw.qb.server.doc/doc/c0008549.html

Kumar, A. (2018). *Centralized vs. decentralized databases: A comparison.* DZone.

Load Balancer Types: What You Need to Know. (s.f.). Obtenido de https://www.digitalocean.com/community/tutorials/load-balancer-types-what-you-need-to-know

Load Balancing 101: The Essentials of Load Balancers. (s.f.). Obtenido de https://www.nginx.com/resources/glossary/load-balancing-101-essentials-load-balancers/

Load Balancing Algorithms and Techniques. (s.f.). Obtenido de https://www.cloudflare.com/learning/ddos/glossary/load-balancing-algorithms-and-techniques/

Mariadb. (2022). *Bootstraping a new node in a Galera Cluster*. Obtenido de https://severalnines.com/blog/bootstrapping-new-node-galera-cluster

Mariadb. (2022). *Mariadb*. Obtenido de https://mariadb.org/

Mariadb. (2022). *Mariadb Galera Cluster documentation*. Obtenido de https://mariadb.com/kb/en/library/galera-cluster-guides/

MariaDB. (2022). *MariaDb Galera Cluster Documentation*. Obtenido de https://mariadb.com/kb/en/library/mariadb-galera-cluster-documentation/

mariadb. (2022). *MaxScale: A Database Load Balancer*. Obtenido de https://mariadb.com/products/technology/maxscale/

MariDB. (2022). *What is Mariadb Galera Cluster*. Obtenido de https://mariadb.com/kb/en/library/what-is-mariadb-galera-cluster/

Microsoft . (2022). *Always On availability groups: a high-availability and disaster-recovery solution*. Obtenido de https://docs.microsoft.com/en-us/sql/database-engine/availability-groups/windows/always-on-availability-groups-sql-server?view=sql-server-ver15Always On availability groups: a high-availability and disaster-recovery solution

Microsoft. (2022). *Sql Server Always On*. Obtenido de https://cloudblogs.microsoft.com/sqlserver/2012/06/12/introducing-alwayson-availability-groups-in-sql-server-2012/

Microsoft. (s.f.). *SQL Server*. Obtenido de SQL Server: https://www.microsoft.com/es-es/sql-server/sql-server-downloads

Mohan, C. (1990). *Data Replication Strategies in Distributed Database Systems.* IEEE Transactions on Knowledge and Data Engineering.

Molina, J. G.-F. (2009). *Base de datos distribuidas: conceptos y arquitecturas.* Universitat Politècnica de València.

Mongodb. (2022). *What is Mongodb*. Obtenido de https://www.mongodb.com/es/what-is-mongodb

Moore, L. (s.f.). *distributed database* . Obtenido de 2022: https://www.techtarget.com/searchoracle/definition/distributed-database

Mullins, C. S. (2022). *database management system (DBMS)*. Obtenido de https://www.techtarget.com/searchdatamanagement/definition/database-management-system

Mysql. (2022). *Mysql Clúster*. Obtenido de https://www.mysql.com/products/cluster/

Mysql. (2022). *Mysql Clúster Architecture*. Obtenido de https://dev.mysql.com/doc/refman/8.0/en/mysql-cluster-architecture.html

neoattack. (27 de Ago de 2020). *Mysql*. Obtenido de https://neoattack.com/neowiki/mysql/

nginx. (2022). *NGINX: From Beginner to Pro* . Obtenido de https://www.nginx.com/resources/library/nginx-from-beginner-to-pro/

nginx. (s.f.). *HAProxy: A High-Performance Load Balancer and Reverse Proxy*. Obtenido de https://www.nginx.com/resources/glossary/haproxy-high-performance-load-balancer-reverse-proxy/

nginx. (s.f.). *Load Balancing Clusters.* Obtenido de https://www.nginx.com/resources/glossary/load-balancing-cluster/

Open SUSE. (2022). *Definition Computer Cluster*|. Obtenido de https://www.suse.com/suse-defines/definition/computer-cluster/

Oracle. (2022). *High Availability Rac*. Obtenido de https://www.oracle.com/database/technologies/high-availability/rac

Oracle. (2022). *Mysql - What is MySQL?* Obtenido de https://dev.mysql.com/doc/refman/8.0/en/what-is-mysql.html

Oracle. (2022). *Oracle*. Obtenido de https://www.oracle.com/corporate/

Oracle. (2022). *RACOV*. Obtenido de https://docs.oracle.com/en/database/oracle/oracle-database/19/racov

Ozsu, M. T. (2011). *Principles of Distributed Database Systems (3rd ed.).* Upper Saddle River, NJ: Pearson Education.

Pasha Kostohrys. (15 de Mayo de 2020). *Database replication — an overview*. Obtenido de https://medium.com/@pkostohrys/database-replication-an-overview-f7ade110477

PostgreSQL 15.2. (2023). *PostgreSQL Docs*. Obtenido de https://www.postgresql.org/docs/

Postgresql. (2022). *Streaming Replication*. Obtenido de https://www.postgresql.org/docs/current/streaming-replication.html

Red Had. (2022). *High-Availability Clusters*. Obtenido de https://www.redhat.com/en/topics/high-availability/what-is-a-high-availability-cluster

severalnines. (2022). *Postgresql Replication Comprehensive Guide*. Obtenido de https://severalnines.com/blog/postgresql-replication-comprehensive-guide

Snodgrass, R. T. (2000). *Developing time-oriented database applications in SQL. New York: Morgan Kaufmann Publishers.*

Souza, J. (2018). *SQL Server 2017 Administrator's Guide.* Packt Publishing.

The PostgreSQL Global Development Group. (2023). *PostgreSQL: The World's Most Advanced Open Source Relational Database*. Obtenido de https://www.postgresql.org/

Tipos de base de datos. (2021). Obtenido de https://www.databases.com/es/tipos-de-bases-de-datos/

Tow, D. (2005). *SQL Tuning.* O'REILLy.

udemy. (2022). *NGINX: The Complete Guide*. Obtenido de https://www.udemy.com/course/nginx-the-complete-guide/)

Valduriez, M. T. (2005). *Arquitecturas de bases de datos distribuidas: conceptos y diseño.* Thomson.

Valduriez, M. T. (2011). *Arquitecturas de bases de datos distribuidas.* Pearson.

Viswanath, S. (9 de Junio de 2020). *Single Leader Replication — a quick overview*. Obtenido de https://medium.com/architectural-tapas/single-leader-replication-a-quick-overview-f7b99861a620

WatElectronics. (s.f.). *Cluster Computing : Architecture & Its Types*. Obtenido de 2021: https://www.watelectronics.com/cluster-computing-architecture-its-types/

Weinberg, J. R. (2009). *The Complete Reference SQL.* Mc Graw Hill.

What Is Server and Application Load Balancing? Types, Configuration Methods, and Best Tools. (s.f.). Obtenido de 2022: https://www.dnsstuff.com/what-is-server-load-balancing

Índice de Figuras

Figura 1 Arquitectura Semestre 2022-2 ..... *2*
Figura 2 Arquitectura Semestre 2023-1 ..... *3*
Figura 3 Comportamiento del Semestre 2022-2 ..... *4*
Figura 4 Comportamiento del Semestre 2023-1 ..... *5*
Figura 5 Topología de un DDBMS *19*
Figura 6 Diagrama de Replicas *22*
Figura 7 Modelo de un clúster *28*
Figura 8 Clúster Asimétrico *30*
Figura 9 Clúster simétrico *31*
Figura 10 Diagrama de un balanceador. *35*
Figura 11 Actualización sistema operativo *50*
Figura 12 Instalación DNF *51*
Figura 13 Instalación net-tools *51*
Figura 14 Instalación de utilidades (nmon, htop, nano y nmap) *52*
Figura 15 Instalación repositorios epel *52*
Figura 16 Instalación rsync *53*
Figura 17 Configuración SELinux. *53*
Figura 18 Instalación de paquetes adicionales *54*
Figura 19 Configuración repositorio MariaDB *55*
Figura 20 Instalación MariaDB *55*
Figura 21 Inicialización MariaDB *56*
Figura 22 Estatus del servicio MariaDB *56*
Figura 23 Configuración de Seguridad MariaDB *57*
Figura 24 Configuración de seguridad MariaDB *57*
Figura 25 Creación del clúster *63*
Figura 26 Inicialización MariaDB *64*
Figura 27 Creación de base de datos inicial *64*
Figura 28 Configuración políticas selinux. *65*
Figura 29 Estatus de cada nodo del clúster *66*
Figura 30 Ejecución en cualquier nodo *66*
Figura 31 Error al inicializar el servicio de MariaDB *68*
Figura 32 Ejecución del modo permisive *70*
Figura 33 Configuración de los Hosts *72*
Figura 34 Verificación de permisos *74*
Figura 35 Estatus del servicio maxscale *78*
Figura 36 Lista de servidores disponibles *79*
Figura 37 Lista de servicios *79*
Figura 38 Advertencias de permisos *81*
Figura 39 Arquitectura del clúster Moodle *82*
Figura 40 Plataforma Moodle *83*
Figura 41 Instrucción Ifconfig *84*
Figura 42 Configuración del Moodle *85*
Figura 43 Resultado del Bechmarck Moodle *86*
Figura 44 Configuración pruebas Moodle *87*

Figura 45 Modo depuración *88*
Figura 46 Curso de prueba Moodle *88*
Figura 47 Configuración Mensajes de depuración y rendimiento *88*
Figura 48 Curso de prueba Moodle *89*
Figura 49 Plan de prueba Jmeter Moodle *89*
Figura 50 Plan de prueba XG *90*
Figura 51 Creación plan de prueba Moodle *90*
Figura 52 Archivos generados prueba Moodle *91*
Figura 53 Resultados de la Prueba Jmeter Moodle *91*
Figura 54 Errores durante la prueba Jmeter *92*
Figura 55 Hits por segundo *92*
Figura 56 Máximo de transacciones *93*
Figura 57 Tiempo máximo de solicitudes *93*