



“2019, Año del caudillo del sur, Emiliano Zapata”.

DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN

OPCIÓN I.- TESIS

TRABAJO PROFESIONAL

“APLICACIÓN DE MÉTODOS DE FRAGMENTACIÓN Y
REPLICACIÓN DE DATOS EN LA NUBE”

QUE PARA OBTENER EL GRADO DE:

**MAESTRO EN
SISTEMAS COMPUTACIONALES**

PRESENTA:

I.S.C FELIPE CASTRO MEDINA

DIRECTOR DE TESIS:

DRA. LISBETH RODRÍGUEZ MAZAHUA

ORIZABA, VER. MÉXICO

JUNIO 2019.





EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MÉXICO

Instituto Tecnológico de Orizaba

"2019, Año del Caudillo del Sur, Emiliano Zapata"

FECHA: 07/10/2019
 DEPENDENCIA: POSGRADO
 ASUNTO: Autorización de Impresión
 OPCIÓN: I

C. FELIPE CASTRO MEDINA
 CANDIDATO A GRADU DE MAESTRO EN:
 SISTEMAS COMPUTACIONALES

De acuerdo con el Reglamento de Titulación vigente de los Centros de Enseñanza Técnica Superior, dependiente de la Dirección General de Institutos Tecnológicos de la Secretaría de Educación Pública y habiendo cumplido con todas las indicaciones que la Comisión Revisora le hizo respecto a su Trabajo Profesional titulado:

"APLICACION DE METODOS DE FRAGMENTACION Y REPLICACION DE DATOS EN LA NUBE".

Comunico a Usted que este Departamento concede su autorización para que proceda a la impresión del mismo.

A T E N T A M E N T E


 MARIO LEÑCIO ARRIJOJA RODRIGUEZ
 JEFE DE LA DIV. DE ESTUDIOS DE POSGRADO



Avenida Oriente 9 Núm. 852, Colonia Emiliano Zapata, C.P. 94320 Orizaba, Veracruz, México

Tel. 01 (272) 7 24 40 96, Fax. 01 (272) 7 25 17 28 e-mail: orizaba@itorizaba.edu.mx

www.orizaba.tecnm.mx





EDUCACIÓN

SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO NACIONAL DE MEXICO

Instituto Tecnológico de Orizaba

"2019, Año del Caudillo del Sur, Emiliano Zapata"

FECHA : 09/09/2019

ASUNTO: Revisión de Trabajo Escrito

C. MARIO LEONCIO ARRIJOJA RODRIGUEZ
JEFE DE LA DIVISION DE ESTUDIOS
DE POSGRADO E INVESTIGACION.
P R E S E N T E

Los que suscriben, miembros del jurado, han realizado la revisión de la Tesis del (la) C. :

FELIPE CASTRO MEDINA

la cual lleva el título de:

"APLICACION DE METODOS DE FRAGMENTACION Y REPLICACION DE DATOS EN LA NUBE".

Y concluyen que se acepta.

A T E N T A M E N T E

PRESIDENTE : DRA. LISBETH RODRIGUEZ MAZAHUA

SECRETARIO : M.C. MARIA ANTONIETA ABUD FIGUEROA

VOCAL : M.C. CELIA ROMERO TORRES

VOCAL SUP. : M.S.C. LUIS ANGEL REYES HERNANDEZ

EGRESADO (A) DE LA MAESTRIA EN SISTEMAS COMPUTACIONALES

OPCION: I Tesis

FIRMA

FIRMA

FIRMA

FIRMA



Avenida Oriente 9 Núm. 852, Colonia Emiliano Zapata, C.P. 94320 Orizaba, Veracruz, México

Tel. 01 (272) 7 24 40 96, Fax. 01 (272) 7 25 17 28 e-mail: orizaba@itorizaba.edu.mx

www.orizaba.tecnm.mx



Agradecimientos

Agradezco el apoyo de mi novia y de mis padres, por haber siempre creído en mí muchas veces más de lo que yo creía en mí mismo, por quererme con tanta fuerza y por amarme en los días difíciles.

Agradezco a mis padres por siempre sorprenderme al resolver los problemas que se presentan y permanecer juntos, por su entusiasmo en la vida y por su gran paciencia, han sido el ejemplo más fuerte de amor. Doy gracias a mi madre por siempre escucharme y por siempre tener los brazos abiertos para mí. Doy gracias a mi padre por mostrarme siempre el camino, por siempre compartir su manera de ver la vida y por el gran amor que lo muestra de muchas formas cada día. Doy las gracias a la familia de mi novia por acogerme siempre, por formar parte de este logro, por siempre compartir su entusiasmo, su sonrisa, su tiempo y su felicidad.

Doy las gracias de una manera especial a Sunny, mi novia, por siempre alentarme a crecer, por el mutuo apoyo para perseguir eso que nos gusta, por siempre hacer de lo complejo algo simple, por creer en mí, por compartir su vida conmigo, por su contagiosa felicidad, por su paciencia y su enorme amor incondicional.

Doy las gracias a la Dra. Lisbeth por siempre apoyarme durante este proyecto, por compartir su experiencia y su enorme conocimiento, por entusiasmarme por este tema y mostrar mucho interés en mi formación. Agradezco también a la Dra. Lisbeth por compartir su experiencia laboral y entusiasmarme a continuar creciendo académicamente.

Agradezco especialmente a la maestra Betty por ser una excelente profesora y brindarme las herramientas y la oportunidad de entrar a este programa de posgrado, por su gran cariño y su entusiasmo. Agradezco también a la maestra Betty por permitirme formar parte en la primer experiencia personal de un proyecto real con la programación Web, sin ese proyecto no habría adquirido muchas herramientas indispensables para la realización de este trabajo.

Agradezco al Consejo Nacional de Ciencia y Tecnología (CONACyT) por el apoyo de la beca de manutención otorgada durante el periodo de estudios y al Tecnológico Nacional de México (TecNM) por dar soporte a esta investigación.

Índice

Índice.....	VI
Índice de figuras.....	VIII
Índice de tablas.....	X
Resumen.....	XI
Abstract.....	XII
Introducción.....	XIII
Capítulo 1. Antecedentes.....	1
1.1. Marco teórico.....	1
1.1.1. Fragmentación de datos.....	1
1.1.1.1. Fragmentación horizontal.....	1
1.1.1.2. Fragmentación vertical.....	1
1.1.1.3. Fragmentación híbrida.....	2
1.1.2. Replicación de datos.....	2
1.1.3. Cómputo en la nube.....	3
1.1.4. Bases de datos como servicio (DaaS).....	3
1.1.5. Software como servicio (SaaS).....	3
1.1.6. Plataforma como servicio (PaaS).....	4
1.1.7. Infraestructura como servicio (IaaS).....	4
1.1.8. Situación tecnológica, económica y operativa de la empresa.....	4
1.2. Planteamiento del problema.....	5
1.3. Objetivo general y específicos.....	6
1.3.1. Objetivo general.....	6
1.3.2. Objetivos específicos.....	6
1.4. Justificación.....	6

Capítulo 2. Estado de la práctica.....	7
2.1. Trabajos relacionados.....	7
2.2. Análisis comparativo.....	18
2.3. Propuesta de solución.....	23
Capítulo 3. Aplicación de la metodología.....	28
3.1. Análisis.....	28
3.2. Selección.....	35
3.3. Implementación.....	36
3.3.1. Análisis de requisitos.....	37
3.3.2. Modelo conceptual.....	39
3.3.3. Modelo de navegación.....	42
3.3.4. Modelo de procesos.....	45
3.3.5. Algoritmo.....	47
3.4. Validación.....	50
Capítulo 4. Resultados.....	51
4.1. Resultado del análisis.....	51
4.2. TPC-E: Esquema y número de tuplas.....	55
4.3. Presentando el sistema.....	58
4.4. Simulador de consultas.....	62
4.5. Caso de estudio en <i>Amazon Web Services</i>	64
5. Conclusiones y recomendaciones.....	73
5.1. Conclusiones.....	73
5.2. Recomendaciones.....	74
Productos académicos.....	75
Referencias.....	76

Índice de figuras

Fig. 2.1. Estructura de la propuesta de solución.....	23
Fig. 3.1. Metodología de búsqueda y evaluación de los trabajos relacionados.....	29
Fig. 3.2. Flujo de trabajo.....	36
Fig. 3.3. Arquitectura para la fragmentación y replicación de datos en la nube.....	37
Fig. 3.4. Diagrama de Casos de uso de la aplicación Web.....	38
Fig. 3.5. Diagrama de actividad del caso de uso “Fragmentar, asignar y replicar”.....	38
Fig. 3.6. Diagrama conceptual de la aplicación.....	39
Fig. 3.7. Diagrama lógico de la aplicación.....	40
Fig. 3.8. Diagrama físico de la aplicación.....	40
Fig. 3.9. Modelo de navegación de la aplicación Web.....	42
Fig 3.10. Página Conexión y archivo de carga del modelo de presentación.....	43
Fig. 3.11. Página Estatus de conexión y análisis de archivo de carga.....	44
Fig. 3.12. Página Selección de tabla a procesar del modelo de presentación.....	44
Fig. 3.13. Página Resultado del modelo de presentación.....	45
Fig. 3.14. Diagrama Fragmentar, asignar y replicar del modelo de procesos.....	46
Fig. 3.15. Fragmentación con traslape.....	49
Fig. 3.16. Resultado del ejemplo.....	50
Fig. 4.1. Cantidad de artículos por editorial.....	52
Fig. 4.2. Proximidad a los tópicos de interés de cada editorial.....	52

Fig. 4.3. Cantidad de artículos por año.....	52
Fig. 4.4. Cantidad de artículos por tema.....	53
Fig. 4.5. Cantidad de artículos por tipo de fragmentación.....	54
Fig. 4.6. Cantidad de artículos por SGBD.....	54
Fig. 4.7. Esquema de la base de datos TPC-E.....	56
Fig. 4.8. Conexión y archivo de carga.....	58
Fig. 4.9. Estatus de conexión y análisis del archivo de carga.....	59
Fig. 4.10. Cuadro de espera y aplicación del método.....	59
Fig. 4.11. Matriz MCRUD.....	60
Fig. 4.12. Tabla ALP y resultados.....	61
Fig. 4.13. Pantalla principal de Simulación CloudSim – TPC-E.....	62
Fig. 4.14. MCRUD de la tabla SETTLEMENT.....	63
Fig. 4.15. Tabla ALP y resultados obtenidos del simulador.....	64
Fig. 4.16. MCRUD de operaciones sobre “CUSTOMER” en Amazon Web Services.....	68
Fig. 4.17. Tabla ALP de operaciones sobre “CUSTOMER” en Amazon Web Services.....	69
Fig. 4.18. Esquema propuesto de asignación y replicación.....	70
Fig. 4.19. Bosquejo de los fragmentos asignados.....	71

Índice de tablas

Tabla 2.1. Análisis comparativo de los trabajos relacionados.....	18
Tabla 2.2. Alternativa de solución.....	24
Tabla 3.1. Comparación de los trabajos relacionados de la editorial ACM.....	30
Tabla 3.2. Comparación de los trabajos relacionados de la editorial IEEE.....	31
Tabla 3.3. Comparación de los trabajos relacionados de la editorial Elsevier.....	32
Tabla 3.4. Comparación de los trabajos relacionados de la editorial Springer.....	33
Tabla 3.5. Comparación de los trabajos relacionados de otras editoriales.....	34
Tabla 3.6. Actores de la aplicación Web.....	37
Tabla 4.1. Artículos por sistema gestor de base de datos.....	55
Tabla 4.2. Número de tuplas de cada tabla en la base de datos TPC-E.....	57
Tabla 4.3. Operaciones obtenidas del simulador.....	63
Tabla 4.4. Operaciones realizadas en Amazon Web Services.....	65
Tabla 4.5. MCRUD de operaciones del atributo “C_ID”.....	69
Tabla 4.6. Comparación de tiempo de ejecución de consultas.....	72

Resumen

La fragmentación y replicación de datos en la nube son temas de gran interés, ya que logran mejorar la seguridad y el desempeño, los cuales son los principales problemas enfrentados por el cómputo en la nube.

El objetivo del presente trabajo fue realizar un análisis comparativo entre las diversas técnicas de fragmentación y replicación de datos aplicadas en la nube para posteriormente implementar una técnica que cumpla con todos los criterios de comparación del análisis: que sea fácil de implementar, que tenga un modelo de costo, que se enfoque en mejorar la recuperación de datos, entre otros. Finalmente, la técnica implementada se validó usando la base de datos estándar o *benchmark* TPC-E.

Las tecnologías que se utilizaron son como marco de trabajo *JavaServer Faces* con la biblioteca *PrimeFaces*, como gestor de bases de datos MySQL y como IDE (*Integrated Development Environment*, entorno de desarrollo integrado) NetBeans, ya que éste es el entorno que mejor se adecua al desarrollo con estas tecnologías.

Abstract

Data fragmentation and replication in the cloud are subjects of great interest, since they manage to improve security and performance, which are the main problems faced by cloud computing.

The objective of this paper was to perform a comparative analysis between the various data fragmentation and replication techniques applied in the cloud to implement a technique that meets all the comparison criteria of the analysis: that is easy to implement, that has a cost model, which focuses on improving data recovery, among others. Finally, the implemented technique was validated using the standard database or TPC-E benchmark.

The technologies that were used are JavaServer Faces as framework with the PrimeFaces library, MySQL as database management system and NetBeans as IDE (Integrated Development Environment), because this is the environment that best adapts to the development with these technologies.

Introducción

La fragmentación es un tema esencial en las bases de datos en la nube, ya que actualmente las demandas de información son enormes y mejorar el desempeño por medio de la fragmentación es una necesidad. La replicación, de igual manera, es un tema crucial para aumentar la disponibilidad de los datos en la nube.

La problemática que observa esta investigación es que hasta ahora no se reporta en la literatura un estudio que compare las diferentes técnicas de fragmentación y replicación de datos existentes aplicadas en la nube. Las personas que buscan una técnica adecuada según sus necesidades, se beneficiarán con este estudio.

Este trabajo resuelve la problemática observada haciendo un análisis y comparando las técnicas de fragmentación y replicación de datos aplicadas en la nube, eligiendo una para su posterior implementación y validación. De esta forma se elige de mejor manera la técnica que más se adecue a las necesidades de cierto escenario.

Este trabajo se organiza en cinco capítulos principales: en el capítulo uno se abarca el marco teórico, planteamiento del problema, objetivo general y específicos y la justificación; el capítulo dos abarca el estado de la práctica y un análisis comparativo; el capítulo tres abarca un análisis profundo de los métodos de fragmentación y replicación aplicados a la nube y el diseño de la aplicación Web para aplicar una técnica seleccionada del análisis; el capítulo cuatro describe los resultados obtenidos por la aplicación Web desarrollada y la descripción de dos casos de estudio para validarla; por último, en el capítulo cinco se presentan las conclusiones y recomendaciones.

Capítulo 1. Antecedentes

1.1. Marco teórico

A continuación, se definen algunos términos relevantes para el trabajo de investigación.

1.1.1. Fragmentación de datos

La fragmentación es una técnica de diseño de bases de datos que consiste en dividir una relación r en un conjunto de fragmentos r_1, r_2, \dots, r_n . Estos fragmentos contienen suficiente información para permitir la reconstrucción de la relación original r [1]. La fragmentación de datos no se fundamenta en la distribución de datos, incluso los sistemas de archivos distribuidos se basan en la distribución de archivos enteros. El problema principal de la fragmentación es la apropiada unidad de distribución. Existen dos estrategias fundamentales para la fragmentación de datos: la horizontal y la vertical. Sin embargo, existe una fragmentación que combina las dos anteriores llamada fragmentación híbrida [2].

1.1.1.1. Fragmentación horizontal

La fragmentación horizontal divide una relación a lo largo de las tuplas. Por ello cada fragmento es un subconjunto de tuplas de esa relación. Hay dos versiones de la fragmentación horizontal: la primaria y la derivada. La primaria se realiza usando predicados que definen una relación. La derivada, por otro lado, es la fragmentación de una relación como resultado de predicados que se definen en otra relación [2].

1.1.1.2. Fragmentación vertical

La fragmentación vertical consiste en dividir una relación en subconjuntos de atributos y en cada fragmento colocar la llave primaria de la relación para permitir su reconstrucción. El objetivo de la fragmentación vertical es dividir una relación en un conjunto de pequeñas relaciones para que la mayor parte de las aplicaciones se ejecuten en un solo fragmento. La fragmentación vertical se investiga en el contexto de sistemas de bases de datos centralizadas y distribuidas, y se desempeña como una herramienta de diseño que permite a las consultas de los usuarios tratar con relaciones más pequeñas [2].

1.1.1.3. Fragmentación híbrida

En muchos casos una fragmentación horizontal o vertical de un esquema de base de datos no es suficiente para satisfacer los requerimientos de las aplicaciones. En estos casos una fragmentación vertical se lleva a cabo después de una horizontal o viceversa, produciendo un árbol estructurado de particiones, a esta técnica se le llama fragmentación híbrida, fragmentación mixta o fragmentación anidada [1].

1.1.2. Replicación de datos

Si la relación r es replicada, una copia de ella se guarda en dos o más sitios. En el caso más extremo se ocupa la replicación total, que consiste en guardar las copias en cada sitio del sistema. Algunas ventajas y desventajas de la replicación se describen a continuación:

- **Disponibilidad:** Si uno de los sitios que contiene la relación r falla, entonces será posible encontrar a la relación r en otro sitio. Así el sistema es capaz de seguir procesando consultas que involucren a r , a pesar de fallas en el sitio.
- **Incremento del paralelismo:** En el caso en donde la mayoría de los accesos a la relación r resultan solo en la lectura de la relación, entonces algunos sitios procesarán consultas que involucren a r en paralelo. Entre más réplicas de r haya, las probabilidades de encontrar los datos necesarios en el sitio donde se ejecutó la transacción aumentan.
- **Mayor sobrecarga en la actualización:** El sistema debe asegurar que todas las réplicas de la relación r sean consistentes, de otro modo habrá errores computacionales. Por ello siempre que r sea actualizada, la actualización debe propagarse por todos los sitios que contengan réplicas de fragmentos de r . El resultado de esto es el aumento de una probable sobrecarga en el sistema [1].

1.1.3. Cómputo en la nube

Cómputo en la nube es un concepto relativamente nuevo en la computación que nació entre los años 1990 y 2000, primero bajo el nombre “Software como servicio”. Inicialmente los vendedores que ofrecían software como servicio proveían aplicaciones personalizables específicas que almacenaban en sus propias máquinas. El concepto de cómputo en la nube

evolucionó a medida que los vendedores comenzaron a ofrecer computadoras genéricas como un servicio sobre las cuales los clientes podían instalar *software* a su elección. Un cliente hace arreglos con el vendedor de cómputo en la nube para obtener cierto número de máquinas con una cierta capacidad o bien una cierta cantidad de almacenamiento. Las aplicaciones Web que necesitan almacenar y recuperar datos para números muy grandes de usuarios son los principales clientes de bases de datos basadas en la nube. Las necesidades de estas aplicaciones difieren de las de aplicaciones de bases de datos tradicionales, ya que valoran la disponibilidad y la escalabilidad sobre la consistencia [1].

1.1.4. Bases de datos como servicio (DaaS)

Las bases de datos como servicio (DaaS, *Database as a Service*) son uno entre los últimos modelos de servicio ofertados por la computación en la nube y son adoptados por muchas organizaciones. La fragmentación de base de datos, la replicación y el almacenamiento son problemas clave que toman la mayoría de los marcos de trabajo orientados a las bases de datos como servicio. En el modelo de servicio DaaS es posible almacenar y manipular enormes bases de datos de manera eficiente y efectiva. El éxito del paradigma DaaS se fundamenta en cómo satisface sus demandas de aplicaciones en términos de fiabilidad y eficiencia de acceso a consultas [3].

1.1.5. Software como servicio (SaaS)

El Software como Servicio (SaaS, *Software as a Service*) es un modelo de entrega del software que usa principalmente el Internet para llegar a cuantas personas sea posible a un costo accesible para cualquiera. El software como servicio consiste en proveer sistemas del mismo modo que los servicios de las compañías de energía eléctrica, de telefonía celular o de los servicios de agua; se enfocan en solventar una necesidad y proveer un servicio especializado por el cual sólo se pagará una cantidad periódica y no un costo total como actualmente se hace al adquirir un programa [4].

1.1.6. Plataforma como servicio (PaaS)

Plataforma como servicio (PaaS, *Platform as a Service*) es un entorno de desarrollo e implementación completo en la nube, con recursos que permiten entregar todo, desde

aplicaciones sencillas basadas en la nube hasta aplicaciones empresariales sofisticadas habilitadas para la nube. Se compran los recursos que se necesita a un proveedor de servicios en la nube, a los que se accede a través de una conexión segura a Internet, pero solo se paga por el uso que se hace de ellos. Al igual que IaaS (*Infrastructure as a Service*), PaaS incluye infraestructura (servidores, almacenamiento y redes), pero también incluye *middleware*, herramientas de desarrollo, servicios de inteligencia empresarial (BI), sistemas de administración de bases de datos, entre otros. PaaS está diseñado para sustentar el ciclo de vida completo de las aplicaciones Web: compilación, pruebas, implementación, administración y actualización [5].

1.1.7. Infraestructura como servicio (IaaS)

La infraestructura como servicio (IaaS, *Infrastructure as a service*) es una infraestructura informática inmediata que se aprovisiona y administra a través de Internet. Permite reducir o escalar verticalmente los recursos con rapidez para ajustarlos a la demanda y se paga por uso.

IaaS evita el gasto y la complejidad que suponen la compra y administración de los servidores físicos e infraestructura de un centro de procesamiento de datos. Cada recurso se ofrece como un componente de servicio independiente, y solo se alquila un recurso concreto durante el tiempo que se necesite. El proveedor de servicios informáticos en la nube administra la infraestructura, mientras que el usuario compra, instala, configura y administra su propio software (sistemas operativos, *middleware* y aplicaciones) [6].

1.2. Situación tecnológica, económica y operativa de la empresa

El Instituto Tecnológico de Orizaba, ubicado en el estado de Veracruz, es uno de los centros educativos mas importantes de la región. Actualmente alberga a mas de 6000 estudiantes distribuidos entre las 8 carreras de ingeniería y 6 postgrados.

Esta casa de estudios pertenece al Tecnológico Nacional de México (TecNM) y es una institución sustentable comprometida con la formación de líderes competentes para responder a los retos y expectativas internacionales.

La misión del instituto se centra en fortalecer los servicios educativos a través de la cobertura, equidad, promoción e inclusión, en la formación integral de los estudiantes impulsando la innovación, ciencia y tecnología; para consolidar la vinculación con pertinencia en los diferentes

sectores estratégicos, modernizando la gestión institucional con transparencia y rendición de cuentas en un ámbito sustentable.

Este trabajo se realizó bajo la dirección de la División de Estudios de Posgrado e Investigación, adjunta al Instituto Tecnológico de Orizaba. Se desarrolló como proyecto de investigación para la Maestría en Sistemas Computacionales, programa ofertado en esta división.

1.3. Planteamiento del problema

La seguridad y el desempeño son dos problemas principales que enfrentan las bases de datos en la nube. Por un lado, la seguridad juega un papel crucial cuando los clientes eligen cómo administrar sus datos, ya que un gran número de ellos no elige un servicio ofrecido por la nube, debido a temores concernientes a la consistencia y la seguridad de los datos. Por otro lado, el desempeño, desde un enfoque relacionado con los tiempos de respuesta y costos de ejecución de las consultas, es otro de los grandes problemas que enfrentan los sistemas basados en la nube, debido a los grandes volúmenes de datos que demandan. Los tiempos de respuesta y los costos de ejecución de las consultas se comprometen cuando las cantidades de datos que necesitan estar en operación son grandes.

En este trabajo se realizó un análisis comparativo de las técnicas de fragmentación y replicación de datos que se aplican en la nube para optimizar el desempeño de las consultas. Para esto, se revisaron los métodos de fragmentación y replicación de datos en la nube desarrollados en los últimos cinco años, por medio de búsquedas en las principales bibliotecas digitales como son ACM digital library, Springer link, Science direct y IEEE Xplore, posteriormente se realizó un análisis comparativo para determinar sus ventajas y desventajas, y después se seleccionó, implementó y validó uno de los métodos anteriormente analizados.

1.4. Objetivos general y específicos

A continuación se muestra el objetivo general y específicos.

1.4.1. Objetivo general

Realizar un análisis comparativo de los métodos de fragmentación y replicación de datos en la nube para conocer sus ventajas y desventajas con respecto a la optimización de las consultas.

1.4.2. Objetivos específicos

- Estudiar y analizar el estado del arte de los métodos de fragmentación y replicación utilizados en la nube, así como de los modelos de costo usados para evaluar esquemas de fragmentación.
- Analizar las tecnologías y estándares para la implementación de los métodos de fragmentación y replicación.
- Realizar un análisis comparativo de los métodos de fragmentación y replicación, así como de los modelos de costo identificados en el estado del arte para conocer sus ventajas y sus desventajas.
- Seleccionar el método que cumpla con más criterios de comparación para su implementación.
- Determinar las tecnologías que se utilizarán para la implementación del método.
- Implementar un método y modelo de costo utilizando las tecnologías seleccionadas.
- Validar el método implementado usando un caso de estudio.

1.5. Justificación

Cada vez más empresas están adoptando el paradigma de cómputo en la nube. Debido a que actualmente se propusieron diversos métodos para la fragmentación y replicación de datos en la nube, es necesario realizar un análisis comparativo de estos métodos para conocer sus fortalezas y debilidades. Dicho análisis beneficiará a las empresas dado que sabrán cuál método se ajusta más a sus necesidades. También, este análisis comparativo sentará las bases para que en un futuro proyecto se desarrolle un método de fragmentación y replicación de datos que tenga las fortalezas de las técnicas actuales, pero que elimine algunas de sus debilidades.

Capítulo 2. Estado de la práctica

Los temas de fragmentación y replicación son altamente abordados por trabajos en los que se busca una mejora de los sistemas en la nube. Para dar un enfoque que contribuya a estos temas es necesario conocer los trabajos relacionados, los cuales se escogieron y analizaron meticulosamente. En este capítulo se describen una serie de trabajos relacionados con la replicación y la fragmentación, posteriormente se hace una síntesis por medio de una tabla y por último se concluye con los hallazgos obtenidos después de haber analizado este número considerable de trabajos.

2.1. Trabajos relacionados

En [7] se mostró una nueva metodología para mejorar el desempeño de la fragmentación, la localización de nodos donde se guardarán los fragmentos en la nube, la replicación y la administración de las réplicas. La seguridad de los datos se compromete cuando surge un ataque a los nodos o a los usuarios en la nube, es por ello que se deben tomar fuertes estrategias de seguridad acompañadas de estrategias que atiendan la optimización de los tiempos de respuesta.

Este trabajo presentó la metodología de división y replicación de datos en la nube para un desempeño y seguridad óptimos (*DROPS División and Replication of Data in Cloud for Optimal Performance and Security methodology*, metodología de división y replicación de datos en la nube para un desempeño y seguridad óptimos). En la metodología DROPS se dividen los archivos en fragmentos y cada fragmento se colocó en un nodo de la nube, cada nodo guarda solo un fragmento de un archivo de datos particular. Los nodos, que guardan los fragmentos de los archivos, tienen cierta distancia entre ellos, la cual se determina por medio del grafo *T-coloring*, el cual es una técnica para la elección de los nodos en donde se guardará un fragmento determinado. De esta manera se limita al atacante a adivinar las ubicaciones de los otros fragmentos. La metodología DROPS no ocupa las técnicas tradicionales de encriptación, y así la metodología se libera de técnicas computacionales costosas. Se muestra que la probabilidad de encontrar y comprometer todos los fragmentos de un archivo, guardados en los nodos de la nube, es extremadamente baja. En cuanto a la replicación, solo se realiza una vez después de

fragmentar el archivo, cada fragmento se replica una sola vez para no comprometer la seguridad de la nube.

Se concluyó que la metodología propuesta es una opción que asegura una mejora en el desempeño de la nube. El trabajo futuro es reducir el tiempo y los recursos utilizados en la descarga, en la actualización y en el guardado del archivo nuevamente, así como estudiar las implicaciones del protocolo TCP (*Transmission Control Protocol*, protocolo de control de transmisión) sobre esta metodología.

En la mayoría de los marcos de trabajo de bases de datos como servicio (DaaS) ocupan la fragmentación, la replicación y la asignación como factores clave para su funcionamiento. En [3] se asumió que la fragmentación ya se realizó y toman los fragmentos de esa base de datos como la unidad en el procedimiento de replicación. Se propuso una nueva estrategia de replicación de bases de datos en un marco de trabajo DaaS, la cual tiene como objetivo mejorar el desempeño del modelo de servicio de base de datos distribuida suministrada por el cómputo en la nube. La replicación propuesta consta de tres pasos: seleccionar los fragmentos más recurridos de la base de datos, guardar esos fragmentos en servidores óptimos y finalmente el mantenimiento de las réplicas.

Las pruebas dieron como resultado que el algoritmo concluyó en un tiempo polinomial y produjo una solución sub óptima. Se estudió la replicación en un contexto DaaS, el cual ofrece servicios NOSQL (por las siglas en inglés "no sólo SQL") que se utilizan renunciando de esta manera a las limitaciones ACID (atomicidad, consistencia, aislamiento, durabilidad) para alcanzar mayor desempeño y escalabilidad. Las pruebas se realizaron usando un simulador OMNeTpp, el cual es un marco de trabajo modular orientado a objetos que simula eventos de red. Como conclusión se obtuvo una eficiente estrategia de replicación para optimizar el desempeño de DaaS en la nube, además, se extendió el trabajo para analizar estrategias para otras aplicaciones distribuidas en la nube.

Los sistemas de almacenamiento en la nube suelen utilizar la fragmentación para que esos datos se distribuyan en diferentes servidores o para que las consultas se ejecuten en paralelo sobre los fragmentos. La respuesta de consultas flexibles se habilita en sistemas de base de datos para

buscar información relacionada con consultas tradicionales que no son contestadas exactamente como se plantean. Una manera para aplicar las consultas flexibles es la generalización de las consultas a nivel de sintaxis. En [8] se estudió la fragmentación basada en taxonomía para el operador de generalización anti-Instanciación que permite encontrar información relacionada en datos distribuidos. El operador anti-instanciación es un operador sintáctico puro, que consiste en cambiar las constantes o variables repetidas en la consulta por variables distintas, las consultas se transforman sintácticamente en expresiones lógicas (utilizando operadores comunes de conjunción \wedge , disyunción \vee , negación \neg , implicación \rightarrow , y universo \forall). Los símbolos, como nombres de tablas o columnas, se toman como constantes y es allí donde aplicaron el operador anti-instanciación. En este trabajo se propuso un enfoque que identifica aquellas tuplas que son buenas candidatas para responder a la consulta anti-instanciada, esas candidatas son puestas en un fragmento para su almacenamiento en un sistema de base de datos distribuida. Este enfoque propuesto de anti-instanciación mejora el almacenamiento con un mecanismo inteligente de respuesta de consultas flexibles. Se utilizó la fragmentación horizontal, ya que la anti-instanciación se aplica sobre un atributo y esto da como resultado un árbol (taxonomía) que divide las tuplas de acuerdo con los valores de ese atributo.

En [9] se presentó una técnica para la fragmentación de bases de datos NOSQL basada en el agrupamiento de los datos de los archivos de registro de la base de datos (*logs*). Las consultas se agruparon por medio de su similitud con otras consultas, y entonces se colocaron en servidores Web locales. Se construyó un sistema prototipo y se comparó el desempeño con una implementación de base de datos basada en la nube, así como con bases de datos fragmentadas por el algoritmo *round robin*. Para la partición se utilizó el algoritmo *k-means* y el nuevo algoritmo basado en el árbol recubridor mínimo (*minimum spanning tree*). Para probar la aplicación se utilizó Amazon Web Services (AWS), Elastic Cloud Computing (EC2). El objetivo del trabajo fue desarrollar un sistema que use el *log* de la base de datos como fuente de información para realizar la fragmentación horizontal de bases de datos en la nube. El trabajo concluyó con un desempeño significativamente mejor que utilizar el algoritmo *round robin* o la base de datos completa.

Moral y Kumar [10] propusieron un sistema que usa la fragmentación y la replicación única (FASR por sus siglas en inglés) para mejorar la seguridad y los tiempos de respuesta en lugar de usar las técnicas tradicionales de encriptación. La selección de los nodos y el almacenamiento está dada por la técnica t-coloring. La contribución principal de FASR fue evitar la pérdida de información en caso de un ataque. La fragmentación se realizó de tal forma que cada uno de los fragmentos de un archivo se almacena en nodos distintos a través de la red. Como ya se mencionó, FASR utiliza el concepto de replicación única, de esta manera se logra aumentar la disponibilidad de datos, la fiabilidad y mejora los tiempos de obtención de datos. Como conclusión se tuvo una aplicación que reduce la información repetida, mejora la seguridad y el desempeño en la nube, a través de la fragmentación y la replicación única.

En [11] se hizo una comparación de métodos para el balanceo de carga de la computación en la nube y se desarrolló un algoritmo de balanceo de carga efectivo, el cual utilizó el poder de diferentes algoritmos existentes colocados juntos para maximizar o minimizar diferentes parámetros como la latencia del rendimiento en diferentes tamaños y topologías de la red. Tres características principales son requeridas en un proveedor de servicios en la nube: publicación de recursos a través de un solo proveedor, habilidad para adquirir recursos transaccionales en demanda y mecanismos para cobrar a los usuarios con base en la utilización de recursos. En este trabajo se presentó una técnica para resolver los problemas de balanceo de carga a través de un alto nivel de fragmentación, el cual en un pequeño *dataset* da una sobrecarga de trabajo, pero en un entorno de alta heterogeneidad en la nube es altamente útil, cuando no se saben las capacidades computacionales de los dispositivos disponibles para usar en la nube.

En [12] se propuso un procedimiento de fragmentación que guarda eficientemente los datos en servidores CSP (por sus siglas en inglés: *Cloud Service Provider*, Proveedor de servicios en la nube) usando la cantidad mínima de encriptación y reduce las expectativas de confianza hacia los proveedores de servicios externos y por lo tanto mejora la privacidad y la confidencialidad. El procedimiento de fragmentación fue aplicado a bases de datos relacionales en la nube donde las tablas fueron tratadas como fragmentos independientes. Para proveer seguridad se crearon fragmentos que son distribuidos en diferentes CSP's para asegurar la desvinculación de los datos en un solo servidor. Para realizar la fragmentación se requirió que las tablas estuvieran en tercera

forma normal, cada fragmento es una tabla, que guarda dentro de sí sus conexiones a otras tablas. Para la encriptación se definieron tres niveles de seguridad para tres niveles de importancia de tablas: tablas de alta confidencialidad, mediana confidencialidad, baja confidencialidad. Se utilizaron tres diferentes algoritmos de encriptación, y cada uno ocupó diferentes recursos computacionales para dar a cada tabla la encriptación que requiere por su nivel de importancia. Como conclusión se obtuvo un algoritmo que mejoró la seguridad mediante técnicas de fragmentación en CSP's.

La computación en la nube con su pago dinámico y características escalables promete computación sobre demanda y ahorro de costo a diferencia de las arquitecturas computacionales tradicionales. En [13] se presentó un enfoque que usa los modelos tradicionales de datos relacionales y distribuye los datos por medio de una fragmentación vertical a través de diferentes proveedores en la nube, de esta manera cada proveedor tendrá solo un pequeño pedazo independiente que es inservible sin las otras partes. Este trabajo propuso reescribir las consultas para paralelizar consultas y *joins* para que de esta manera mejore el desempeño. Además, usó *Hibernet Interceptors* para adaptar las consultas HQL (*Hibernet Query Language*, lenguaje de consultas de Hibernet) para la configuración de la distribución vertical y el marco de trabajo SeDiCo (*A Secure and Distributed Cloud Data store*, un almacén de datos en la nube seguro y distribuido). Se mencionó que el enfoque utilizado no es factible en escenarios prácticos del mundo real, ya que se genera grandes pérdidas de rendimiento. El programa reescritor de la consulta se encargó de dividir la consulta y generar un XML (*extensible markup language*, lenguaje de marcado extensible), el cual le permite definir qué predicado de la consulta pertenece a qué partición. De esta manera se pudo dividir la consulta en subconsultas y ejecutarse en su partición correspondiente. Como conclusión se observaron dos mejoras notables. La primera que los resultados se filtraron más rápido y más fácilmente, ya que el conjunto de resultados estuvo potencialmente limitado al número de tuplas que devuelven las particiones. La segunda fue que la consulta fragmentada se ejecutó en paralelo en cada CPU (*Central Processing Unit*, Unidad Central de Procesamiento) y esto produjo una mejora en el desempeño de la ejecución.

El trabajo que se propuso en [14] se centró en los problemas de desempeño que vienen con el enfoque de distribución de datos del marco de trabajo SeDiCo, el cual distribuye verticalmente

una base de datos a través de nodos en la nube. Se desarrolló una arquitectura de caché de datos y dos procesos de recuperación: la obtención perezosa y paralela. Para medir el desempeño de estas dos estrategias se implementaron en un entorno local, donde instalaron todos los componentes en una máquina física, y también se hicieron las mediciones por medio de un entorno remoto. Se observaron los resultados y se pudo llegar a la conclusión de que la obtención de datos paralela superó en gran medida a la obtención de datos perezosa con el conjunto de datos de TPC-W (punto de referencia transaccional para la Web). También se observó que el enfoque SeDiCo es viable cuando se compara con una base de datos no distribuida. La llegada de arquitecturas NOSQL logra un desempeño óptimo para los volúmenes particionados verticalmente y SeDiCo se verá beneficiado. Se mostró también cómo una arquitectura de la memoria caché es adecuada para los problemas de desempeño. El trabajo concluyó dando nuevos campos de estudio que los autores abordarán, también se mencionó que SeDiCo soporta los sistemas de bases de datos MySQL y Oracle, pero se harán pruebas con otros gestores de base de datos como PostgreSQL y Microsoft SQL Server.

Las aplicaciones que son desplegadas en la nube normalmente trabajan con requerimientos de desempeño dedicado y de alta disponibilidad. Esto se logra por medio de la replicación de datos a través de varios sitios y/o realizando la fragmentación de datos. En [15] se analizaron tres protocolos para la administración de datos distribuidos en la nube: *ReadOne-Write-All-Available* (ROWAA, leer una réplica y escribir en todas las réplicas disponibles), *Majority Quorum* (MQ, la mayor audiencia) y *Data Partitioning* (DP, partición de datos). Presentaron BOWULF, un meta-protocolo basado en un modelo de costos completo que integra los tres protocolos y que selecciona dinámicamente el protocolo con la latencia más baja para una carga de trabajo determinada. Se hicieron dos contribuciones en [15]. Se introdujo el modelo de costos BOWULF que toma en cuenta características de ROWAA, MQ y DP y se analizó la calidad de predicción del modelo de costos comparándolo con los costos de un despliegue real. Se utilizó el *benchmark* TPC-C (*Transaction Processing Performance Council*, Consejo de rendimiento de procesamiento de transacciones) para medir la exactitud del modelo de costos. Como conclusión se obtuvo un meta-protocolo que calcula los costos de los

algoritmos para cierto conjunto de datos y cambia automáticamente el protocolo mejor calificado después de analizar ese conjunto de datos.

Una manera eficiente para mejorar el desempeño de los sistemas de base de datos es el procesamiento distribuido. En [16] se presentó un sistema de base de datos distribuido (DDBS) dinámico sobre un ambiente en la nube, que permite que las decisiones de fragmentación, asignación y replicación (FAR, por sus siglas en inglés, *fragmentation, allocation and replication*) se tomen dinámicamente en tiempo de ejecución. Se presentó una técnica mejorada de asignación y replicación que se aplica en la fase inicial del diseño de bases de datos distribuidas cuando no hay información disponible sobre la ejecución de las consultas. Las decisiones FAR se basan en los patrones de acceso y en la carga de los sitios. La técnica de replicación y asignación mejorada maximiza el número de accesos locales comparado con accesos a sitios remotos, por lo que mejora el desempeño del sistema e incrementa la disponibilidad. La arquitectura que se propuso permite acceder a los usuarios a las bases de datos desde cualquier parte del mundo sin poseer ninguna infraestructura tecnológica a través de un navegador de Internet, una aplicación móvil o una aplicación de escritorio. Se propuso una fragmentación horizontal, esta técnica ayuda a tomar decisiones de fragmentación en la fase inicial del diseño de bases de datos distribuidas. Se presentó el DDBS dinámico sobre ambientes en la nube, que permite tomar decisiones FAR con base en el historial de acceso y carga del sitio, esto mejora el desempeño del sistema e incrementa la disponibilidad.

En [17] se presentó un estudio de la fragmentación basada en agrupamiento para el operador de generalización anti-Instanciación, con el cual se encontró información relacionada con cada consulta en una base de datos distribuida. Se presentó un sistema prototipo que usa la taxonomía médica para determinar las similitudes entre expresiones médicas. Se usó un algoritmo de agrupamiento estándar que obtiene una fragmentación semántica de datos, la cual soporta un mecanismo inteligente de respuestas de consultas flexibles. La respuesta de consultas flexibles ofrece el mecanismo para respuestas inteligentes a consultas de usuarios, para que estas respuestas vayan más allá de la convencional respuesta de consultas exacta. Se presentó una reescritura de consultas detallada y un método de redireccionamiento de consultas que permite el acceso a los fragmentos distribuidos. Se asumió un lenguaje lógico que consiste en un grupo

finito de símbolos predicado, un posible grupo infinito de símbolos de constantes y un grupo infinito de variables. Se usó el mismo mecanismo para el operador anti-instanciación y la reescritura de consultas que en el trabajo [8] ya descrito. Se resolvió el problema de replicación, que consiste en un problema de empaquetamiento (*Bin Packing*), los objetos deben guardarse en el mínimo número de compartimientos de un servidor dado sin exceder su máxima capacidad. Se propuso un enfoque de fragmentación y replicación para sistemas de bases de datos distribuidas, con el cual se mejora el almacenamiento en la nube con la ayuda de un mecanismo de respuestas a consultas flexibles.

Las aplicaciones intensivas de datos son desarrolladas cada día más en ambientes computacionales, estas aplicaciones tienen diferentes requisitos de calidad de servicio (QoS por sus siglas en inglés, *quality of service*). Para dar un continuo soporte al requerimiento QoS se propusieron en [18] dos algoritmos de replicación de datos conscientes de QoS (*QoS-aware data replication*, QADR). El primer algoritmo adoptó la idea intuitiva de la alta calidad de servicio en la primera réplica (HQFR, *high-QoS first-replication*) para realizar la replicación de datos. Sin embargo, este algoritmo codicioso no minimiza el costo de la replicación de datos y la cantidad de réplicas que no cumplen con la QoS. El segundo algoritmo transformó el problema de QADR en el ya conocido problema de “costo mínimo, flujo máximo” (MCMF, *minimum-cost maximum-flow*), para que de esta manera se alcanzaran soluciones óptimas del problema QADR en un tiempo polinomial, sin embargo, este segundo algoritmo ocupó más tiempo computacional que el primero. Se propuso aplicar las técnicas de combinación de nodos para reducir el posible retardo en la replicación de datos. La arquitectura de los algoritmos se basó en la arquitectura del sistema de archivos de Apache Hadoop (HDFS, *Hadoop data file system*). Finalmente, el experimento de simulación demostró la efectividad de los algoritmos propuestos en la replicación y recuperación de datos.

Se propuso en [19] una estrategia dinámica de replicación de datos y un pequeño estudio de la estrategia de replicación adecuada para ambientes de cómputo distribuido. Se analizaron y modelaron las relaciones entre la disponibilidad de sistemas y el número de réplicas. Se evaluaron e identificaron los datos más populares y el desencadenamiento de una operación de replicación cuando la popularidad de los datos pase un límite dinámico. Se calculó un número

adecuado de réplicas para cumplir con el requerimiento de una tasa razonable y efectiva de bytes de sistema y se colocaron réplicas entre los nodos de una manera equilibrada. Se diseñó el algoritmo de replicación dinámica en una nube.

Los autores realizaron un modelo matemático para describir la relación entre la disponibilidad del sistema y el número de réplicas, este modelo contempla el tamaño, el tiempo de acceso y la probabilidad de falla de cada archivo. Un archivo popular fue identificado por la cantidad de veces que fue accedido por los usuarios, cuando la popularidad de un archivo pasaba un límite dinámico se realizaba la operación de replicación. La estrategia propuesta llamada D2RS (*Dynamic Data Replication Strategy*, Estrategia de replicación dinámica de datos) fue implementada usando las herramientas de *CloudSim*. Las pruebas demostraron la eficiencia y la eficacia del sistema que se mejoró por esta estrategia de replicación para la nube. D2RS incrementó la disponibilidad de los datos, mejoró la tasa de ejecución exitosa de las tareas del sistema, y redujo el consumo de banda ancha en la nube.

Se estudió en [20] la replicación en centros de datos alojados en la nube. Este trabajo, diferente a otros enfoques, contempló la eficiencia de la energía y el consumo de banda ancha en el sistema. Este trabajo fue una mejora a la calidad de servicios (QoS) obtenida como resultado de la reducción de retrasos en la comunicación. Los resultados de la evaluación obtenidos de un modelo matemático y simulaciones costosas ayudaron a revelar compensaciones de rendimiento y eficiencia energética, así como guiar el diseño de futuras soluciones de replicación de datos. Se utilizó un simulador computacional llamado *GreenCloud* para realizar las pruebas de la evaluación. La replicación en la nube es una herramienta para proveer mayor disponibilidad, ya que si hay un error en la infraestructura de algún centro de procesamiento de datos, éste fácilmente sigue trabajando con las réplicas de los datos. La disponibilidad es medida por la probabilidad de fallos en los componentes de los centros de datos o en la infraestructura del almacenamiento. En la replicación de este trabajo se asume que todos los datos son permanentemente guardados en los centros de datos y adicionalmente, dependiendo del patrón de acceso, estos son replicados. El resultado que se obtuvo confirmó que replicar los datos más cerca de los consumidores reduce el consumo de energía, el consumo de banda ancha y los

retrasos en la comunicación. Plantearon que el trabajo a futuro se centre en el desarrollo de la implementación de un banco de pruebas sobre la solución que se propuso.

En [21] se analizaron diferentes sistemas de almacenamiento distribuido que usan la fragmentación de datos y la distribución de los mismos como una manera de protección. Las soluciones existentes se clasificaron en dos categorías: a nivel de bits y estructurados. Se describieron los requisitos de alto nivel y los rasgos deseables de la arquitectura de un sistema eficiente de fragmentación de datos, que abordó el rendimiento, la disponibilidad, la capacidad de recuperación y la escalabilidad. Este trabajo se centró en una categoría en particular de soluciones de almacenamiento distribuido: sistemas que usan la fragmentación y la dispersión no sólo para la disponibilidad sino también para proveer seguridad adicional a los datos. Se presentaron soluciones de fragmentación de base de datos para quitar por completo la encriptación. Se plantearon los problemas que se presentan en la fragmentación y sus recomendaciones para abordarlos, uno de ellos es el no saber dónde son guardados los fragmentos que se distribuyen por la nube, ya que algunos desarrollos propusieron esta manera de almacenamiento para incrementar su seguridad, separando los datos incluso en sitios de proveedores distintos, sin embargo, se consideró una solución débil. La solución a este problema es guardar los fragmentos en los sitios de un solo proveedor que garantice la separación física. También se analizaron los sistemas de almacenamiento distribuido existentes para proveer una confidencialidad adicional de los datos. Como trabajo futuro plantearon desarrollar una arquitectura escalable para datos no estructurados que combine seguridad con elasticidad mientras se optimiza el espacio de almacenamiento y los costos. Como conclusión de este trabajo se mencionó que se concentrarán en los problemas del desempeño, abarcando el procesamiento en paralelo y que se proveerá la dispersión de datos sobre niveles de confidencialidad.

Con el siempre actualizado avance en el manejo de la información y los datos, los sistemas de bases de datos distribuidas (DDBS) aún ofrecen las herramientas más demandadas para manejar los enormes volúmenes de datos. Se desarrollaron herramientas para contribuir al desempeño de los DDBS. Sobre estas estrategias, la fragmentación de datos, la asignación de datos, la replicación de datos y el agrupamiento de sitios son las técnicas eficientes más utilizadas como alternativa de los diseños DDBS y cambiarlas sería altamente costoso. Por un lado, una

fragmentación y asignación precisa y bien estructurada de los datos está destinada a mejorar increíblemente su ubicación y promover los rendimientos totales de DDBS. Por otro lado, encontrar un proceso de agrupamiento de sitios prácticos contribuirá notablemente a reducir los costos totales de transmisión. En [22] se desarrolló un enfoque de heurística optimizada para la fragmentación horizontal y la asignación de datos, el cual tuvo por objetivo proponer una fragmentación de datos adecuada, una asignación de datos precisa y diseñar un algoritmo práctico para el agrupamiento de sitios. La fragmentación de datos que se propuso se lleva a cabo por medio de un modelo matemático de asignación y replicación que optimizó notablemente la fragmentación horizontal. El proceso de evaluación buscó manifestar que este trabajo mejoró el desempeño de los DDBS y mejoró los costos de transmisión. Como trabajo futuro se mencionó que se harán más pruebas para desarrollar un marco de trabajo.

2.2. Análisis comparativo

En la Tabla 2.1 se muestra un análisis comparativo de los trabajos anteriormente descritos, para que se observen las diferencias y las similitudes entre ellos de una mejor manera.

Tabla 2.1. Análisis comparativo de los trabajos relacionados

Artículo	Problema	¿Qué fue fragmentado?	Tipo de fragmentación:	Replicación	Tecnologías	Resultados
[7]	Los datos se ven comprometidos cuando atacan a los usuarios y los nodos dentro de la nube sin una estrategia de replicación adecuada.	Se fragmentan archivos, para aumentar la seguridad de los datos en la nube.	Se asemeja a la fragmentación vertical, ya que no se dividen tuplas en los archivos, sino fragmentos de ellas, y cada parte se guarda en un nodo de la nube diferente.	Sí	Máquinas virtuales para hacer pruebas de seguridad sobre los algoritmos.	Se compararon otras metodologías que se utilizan en la nube para la réplica de datos, sin embargo, se obtuvieron mejores resultados con la metodología DROPS y aumentaron los niveles de seguridad acompañado de una ligera caída de rendimiento.
[3]	Se encontraron problemas de diseño por medio de una encuesta debido a las estrategias de replicación, que no son incorrectas, pero que pueden ser mejores.	En este artículo se asume que la fragmentación ya se encuentra realizada, se enfocan solo en la replicación.	Ninguna	Sí	NOSQL, simulador OMNeTpp para las pruebas.	La simulación se realizó con OMNeTpp, se dio una estrategia para mejorar el desempeño en la replicación de fragmentos en bases de datos en la nube. Para las pruebas se fragmentaron de alguna manera los datos (no se estudia en este trabajo la fragmentación, solo se mencionan cuáles existen), se escogieron los sitios más populares (los que tenían el mayor número de visitas) y sobre ellas se aplicó el algoritmo de replicación, el cual mejoró el desempeño de los tiempos de respuesta y disponibilidad de los datos.
[8]	Los tiempos de respuesta largos y las respuestas de consultas vacías originadas por consultas tradicionales a una base de datos distribuida en la nube.	Bases de datos. Se asume que los datos están guardados en tablas relacionales. No se	Horizontal	No	Respuesta flexible de consultas (se habilita como característica en sistemas de bases de datos	Con este enfoque propuesto de anti-instanciación se mejoró el almacenamiento con un mecanismo inteligente de respuesta de consultas flexibles y se evitaron respuestas vacías a consultas tradicionales, ya que proveen más información.

Tabla 2.1. Análisis comparativo de los trabajos relacionados (continuación)

Artículo	Problema	¿Qué fue fragmentado?	Tipo de fragmentación:	Replicación	Tecnologías	Resultados
		especificó qué contenía la base de datos de prueba.			distribuidas).	
[9]	El tiempo de respuesta de consultas de bases de datos tradicionales puede ser mejorado usando un enfoque de partición de base de datos NOSQL.	Bases de datos NOSQL que contenían datos sobre artículos de investigación y autores.	Horizontal	No	NOSQL, MongoDB, AWS, EC2.	El desempeño mejoró significativamente comparado con utilizar bases de datos completas o usar el algoritmo de partición <i>round robin</i> .
[10]	El mayor problema de la computación en la nube son la seguridad y los tiempos de respuesta altos. Un problema importante para la seguridad es prevenir la modificación ilegal y la pérdida de información. En este trabajo se dio una solución a estos problemas.	Archivos, no se mencionó qué contenido tenían.	Vertical	Sí	Máquinas virtuales, DaaS	En este trabajo se mejoró el tiempo de respuesta de la nube debido a la fragmentación, la seguridad y por medio del enfoque de una sola replicación se mejoró también la capacidad de almacenamiento.
[11]	El problema principal del balanceo de carga es el cómo manejar la información que debe mantenerse a través de múltiples peticiones en una sesión de usuario.	Se asume que ya fue fragmentado de alguna manera el conjunto de datos en la nube y se enfocan solo en	Ninguna	No	No se mencionó ninguna herramienta o tecnología más que las mismas herramientas que provee la nube como	El resultado que se obtuvo fue satisfactorio, ya que se logró proponer las soluciones adecuadas para atacar el problema del balanceo de carga y de esta manera mejorar los tiempos de respuesta.

Tabla 2.1. Análisis comparativo de los trabajos relacionados (continuación)

Artículo	Problema	¿Qué fue fragmentado?	Tipo de fragmentación:	Replicación	Tecnologías	Resultados
		el balanceo de carga.			SaaS (<i>Software as a Service</i>).	
[12]	La confidencialidad y la preservación de la seguridad de los datos.	Una base de datos que contenía información sobre algunos centros de investigación (EC2).	Las tablas se tratan como fragmentos.	No	VPN's (<i>Virtual Private Network</i> , red privada virtual)	Se propuso un mecanismo de fragmentación que mejoró la seguridad y la confidencialidad de bases de datos relacionales.
[13]	El desempeño de consultas tradicionales en bases de datos fragmentadas sobre la nube.	Se asume que la fragmentación ya fue realizada verticalmente.	Vertical	No	Hibernet, HQL, SeDiCo	Como resultado se obtuvo una mejora en el desempeño y en el tiempo de respuesta de las consultas.
[14]	El tiempo consumido para unir los datos particionados verticalmente por el marco de trabajo SeDiCo.	Se asume que la fragmentación ya fue realizada verticalmente por el marco de trabajo SeDiCo.	Vertical	No	SeDiCo, MySQL (gestor de base de datos), TPC-W	Se obtuvo una mejora en la obtención de datos en bases de datos fragmentadas verticalmente con el enfoque SeDiCo.
[15]	Las aplicaciones desplegadas en la nube trabajan con requerimientos de desempeño dedicado y alta disponibilidad y los protocolos no son seleccionados de acuerdo al conjunto de	Se asume que la fragmentación ya fue realizada.	Ninguna	Sí	Protocolos ROWAA, MQ y DP. <i>Benchmark</i> TPC-C.	Se obtuvo el meta-protocolo llamado BEOWULF que mejoró la elección entre partición o replicación para cierto conjunto de datos, tomando en cuenta los costos.

Tabla 2.1. Análisis comparativo de los trabajos relacionados (continuación)

Artículo	Problema	¿Qué fue fragmentado?	Tipo de fragmentación:	Replicación	Tecnologías	Resultados
	datos dado.					
[16]	Los costos en las bases de datos distribuidas pueden ser reducidos con la partición de tablas en fragmentos y la replicación.	Una base de datos de cuentas de clientes de bancos.	Horizontal	Sí	No se menciona ninguna tecnología.	Se presentó DDBS, un sistema dinámico de bases de datos distribuidas sobre entornos en la nube, el cual mejoró notablemente los costos en consultas y las técnicas de replicación y asignación, ya que toman en cuenta la ubicación de los sitios.
[17]	Cuando las consultas rígidas no pueden ser contestadas, se devuelve una respuesta vacía, la cual no provee ninguna información útil. Ocupar el mínimo espacio en los compartimentos de cada servidor para no exceder la capacidad de los mismos.	Una base de datos de un hospital.	Horizontal	Sí	Respuesta flexible de consultas (se habilita como característica en sistemas de bases de datos distribuidas).	Se presentó una metodología para la reescritura de consultas y se propuso un enfoque de fragmentación y replicación, el cual mejora el desempeño en los sistemas de bases de datos distribuidas.
[18]	Los costos de replicación y los diferentes requerimientos en cuanto a calidades de servicio en las aplicaciones en la nube.	No se abordó la fragmentación.	Ninguna	Sí	Apache Hadoop, STP (<i>spanning tree Protocol</i> , protocolo de árbol de expansión).	Se presentaron dos algoritmos que ayudaron a reducir los costos de replicación de datos y se propusieron también técnicas de combinación de nodos para mejorar el tiempo de replicación de datos.
[19]	La disponibilidad del sistema, el tiempo de respuesta y el uso de banda ancha en la nube.	No se abordó la fragmentación.	Ninguna	Sí	Herramientas CloudSim	Se obtuvo la estrategia R2RS, la cual se basó en un modelo matemático que contempla tamaño, tiempo de acceso y probabilidad de falla. Esta estrategia identificó datos populares y cuando estos datos populares pasan un número dinámico de acceso se realiza la réplica. Esta estrategia

Tabla 2.1. Análisis comparativo de los trabajos relacionados (continuación)

Artículo	Problema	¿Qué fue fragmentado?	Tipo de fragmentación:	Replicación	Tecnologías	Resultados
						incrementa la disponibilidad del sistema y reduce los tiempos de respuesta.
[20]	La disponibilidad, los tiempos de respuesta y la saturación en los centros de datos en la nube.	No se abordó la fragmentación.	Ninguna	Sí	Green Cloud, DVFS (<i>dynamic voltage and frequency scaling</i>), DPM (<i>dynamic power management</i>).	Los resultados obtenidos confirmaron que replicar los datos más cerca de los consumidores de datos reduce, en las aplicaciones sobre la nube, el consumo de energía, el uso de banda ancha y retrasos en la comunicación.
[21]	Los clientes no optan por migrar sus sistemas a la nube, debido a preocupaciones de disponibilidad, seguridad y privacidad del almacenamiento de datos. Los recursos ocupados para la encriptación pueden ser minimizados.	No se menciona qué fue fragmentado.	Se asemeja a una fragmentación vertical, ya que se optó por una manera de separar los datos bit por bit.	No	Técnica PIR (<i>Private Information Retrieval</i>), Daas.	Se analizaron los sistemas de almacenamiento distribuido y se propusieron métodos de fragmentación para quitar por completo la encriptación.
[22]	Los problemas de fragmentación, en algunos casos, no se atienden de una manera eficaz para los DDDBS. Los costos de comunicación son altos.	Se fragmentó como prueba del funcionamiento una tabla de alumnos.	Horizontal	Sí	No se mencionó ninguna tecnología.	Se sugirió un enfoque optimizado para la fragmentación horizontal. Se propuso sobre todo un agrupamiento de sitios. La asignación de datos se hizo con el modelo de costos propuesto, esto aumentó el espacio de guardado y redujo el acceso remoto.

Se concluye, después de analizar los anteriores artículos, que la mayor parte de estos utilizan las técnicas de fragmentación y replicación para mejorar el desempeño en la nube o para brindar una mayor seguridad. En los trabajos [7], [10], [12] y [21] se atienden problemas de seguridad utilizando la fragmentación para volver ilegibles los datos para un atacante que quiera obtener información. En los trabajos restantes se utilizan las dos técnicas anteriores para resolver problemas de desempeño de maneras diferentes. El volumen de datos en los sistemas de almacenamiento en la nube crece a pasos agigantados, es por eso que aplicar métodos de fragmentación y replicación de datos en la nube es altamente adecuado. Es por esto que en este trabajo se realizó un análisis comparativo de los métodos de fragmentación y replicación de datos aplicados en la nube para conocer sus ventajas y desventajas con respecto a la optimización de consultas y posteriormente tomar el más adecuado según ciertos criterios, implementarlo en un conjunto de datos y evaluar su desempeño.

2.3. Propuesta de solución

Para llegar a un resultado satisfactorio, se proponen una serie de pasos que dividen el trabajo en etapas, las cuales están sujetas a la metodología de desarrollo elegida, como se muestra en la Figura 2.1.

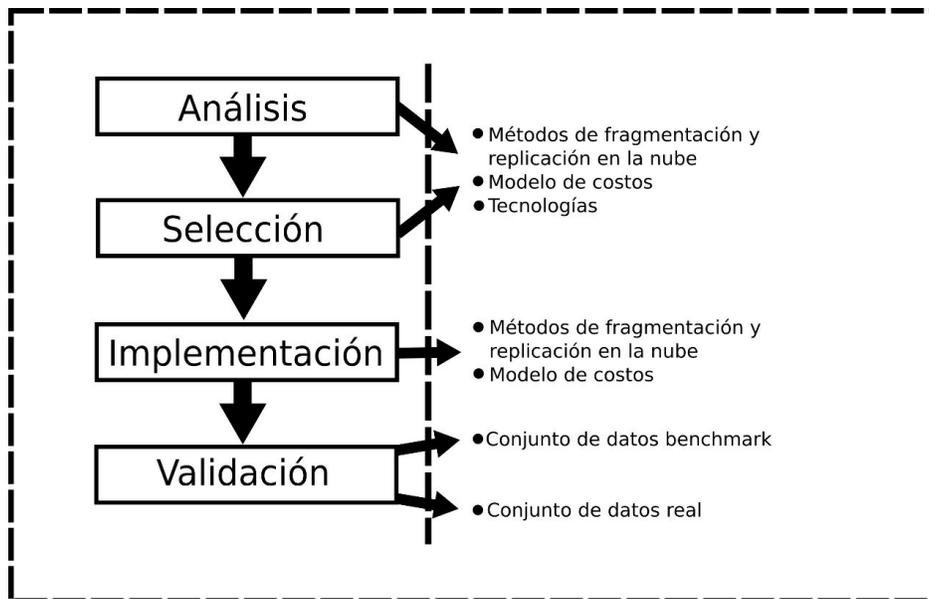


Fig. 2.1 Estructura de la propuesta de solución

En la Figura 2.1 se observan los pasos propuestos para realizar el análisis comparativo, que se describen a continuación.

1. **Análisis:** En esta etapa del trabajo se analizan diferentes artículos científicos por medio de la búsqueda en las principales bibliotecas digitales como son ACM digital library, Springer link, Science direct y IEEE Xplore, donde se encuentren métodos de fragmentación y replicación en la nube, modelos de costo y tecnologías empleadas para su desarrollo.
2. **Selección:** En esta etapa se selecciona un método de fragmentación y replicación, un modelo de costo y tecnologías de los trabajos anteriormente analizados, se eligió el método que cumpla con tener una solución completa, que aborde simultáneamente la fragmentación y la replicación, que contenga un modelo de costos, que esté enfocado en la nube, que tenga facilidad de implementación y por último que su desempeño sea óptimo.
3. **Implementación:** En esta etapa se implementa el método de fragmentación y de replicación y el modelo de costos obtenidos de la fase anterior.
4. **Validación:** En esta etapa se prueba la implementación por medio del *benchmark* TPC-E. De esta manera se comprueba la efectividad de la solución propuesta.

A continuación se muestra la Tabla 2.2, que presenta la alternativa de solución.

Tabla 2.2. Alternativa de solución

Aspecto	Propuesta
Proveedor de cómputo en la nube	AWS
Marco de trabajo	JSF
IDE	NetBeans
Metodología	UWE
SGBD	MySQL
<i>Benchmark</i>	TPC-E
Simulador de cómputo en la nube	CloudSim

Amazon Web Services (AWS) ofrece un amplio conjunto de servicios globales de informática, almacenamiento, bases de datos, análisis, aplicaciones e implementaciones que ayudan a las organizaciones a avanzar con más rapidez, reducir costos de las tecnologías de la información y escalar aplicaciones. Una plataforma de servicios en la nube, como Amazon Web Services, es propietaria y responsable del mantenimiento del hardware conectado en red necesario para dichos servicios de aplicaciones, mientras el usuario se dedica a aprovisionar lo que necesite por medio de una aplicación Web [23].

Diseñado para ser flexible, el marco de trabajo JavaServer Faces aprovecha los conceptos de UI (*User interface*, interfaz de usuario) estándar y niveles de Web existentes sin limitar a los desarrolladores a un lenguaje, protocolo o dispositivo de cliente de marcado en particular. Las clases de componentes de interfaz de usuario incluidas con la tecnología JavaServer Faces encapsulan la funcionalidad del componente, no la presentación específica del cliente, lo que permite que los componentes de la interfaz de usuario de JavaServer Faces se representen en varios dispositivos cliente. Al combinar la funcionalidad del componente de la interfaz de usuario con los visualizadores personalizados, que definen los atributos de representación para un componente de interfaz de usuario específico, los desarrolladores construyen etiquetas personalizadas en un dispositivo cliente particular. La tecnología JavaServer Faces proporciona un renderizador personalizado y una biblioteca de etiquetas personalizadas JSP (*JavaServer Pages*) para renderizar en un cliente HTML, permitiendo a los desarrolladores de aplicaciones *Java Platform, Enterprise Edition* (Java EE), utilizar la tecnología JavaServer Faces en sus aplicaciones [24].

NetBeans IDE es el IDE oficial para Java 8. Con sus editores, analizadores de código y conversores, se actualizan las aplicaciones de forma rápida y sin problemas para usar nuevas construcciones de lenguaje Java 8, como *lambdas*, operaciones funcionales y referencias de métodos. Con su editor Java que mejora constantemente, muchas funciones completas y una amplia gama de herramientas, plantillas y muestras, NetBeans IDE establece el estándar para el desarrollo con tecnologías de vanguardia listas para usar [25].

UWE (*Unified Modeling Language Web Engineering*, Ingeniería Web del Lenguaje Unificado de Modelado) es una metodología que permite especificar de mejor manera una aplicación Web en su proceso de creación, mantiene una notación estándar basada en el uso de UML (*Unified*

Modeling Language) para sus modelos y sus métodos. La metodología define claramente la construcción de cada uno de los elementos del modelo.

En su implementación se deben contemplar las siguientes etapas y modelos:

- **Análisis de requisitos.** Plasma los requisitos funcionales de la aplicación Web mediante un modelo de casos de uso.
- **Modelo de contenido.** Define, mediante un diagrama de clases, los conceptos a detalle involucrados en la aplicación.
- **Modelo de navegación.** Representa la navegación de los objetos dentro de la aplicación y un conjunto de estructuras como son índices, menús y consultas.
- **Modelo de presentación.** Representa las interfaces de usuario por medio de vistas abstractas.
- **Modelo de proceso.** Representa el aspecto que tienen las actividades que se conectan con cada clase de proceso.

Como se hace notar, UWE provee diferentes modelos que permiten describir una aplicación Web desde varios puntos de vista abstractos. Cada uno de estos modelos se representa como paquetes UML, dichos paquetes son procesos relacionados que se refinan en iteraciones sucesivas durante el desarrollo de UWE [26].

MySQL es la base de datos de código abierto más popular del mundo. Con su rendimiento comprobado, fiabilidad y facilidad de uso, MySQL se convirtió en la principal opción de base de datos para aplicaciones basadas en Web, utilizada por propiedades Web de alto perfil, como Facebook, Twitter, YouTube, entre otros. Además, es una opción extremadamente popular como base de datos integrada, distribuida por miles de ISV (*Independent software vendor*, Vendedor Independiente Software) y OEM (*Original equipment manufacturer*, fabricante de equipos originales) [27].

El *benchmark* TPC-E es una carga de trabajo de procesamiento de transacciones en línea (OLTP, *Online Transaction Processing*). Es una mezcla de transacciones intensivas de solo lectura y actualización que simulan las actividades de entornos complejos de aplicaciones OLTP. El esquema de la base de datos, la población de datos, las transacciones y las reglas de implementación han sido diseñados para representar sistemas OLTP modernos [28].

Cloudsim es un marco para modelar y simulación de infraestructuras y servicios de computación en la nube. Originalmente construido en el laboratorio de sistemas de computación y distribución de nubes (CLOUDS, *Cloud Computing and Distributed Systems*). Cloudsim se ha convertido en uno de los simuladores de la nube más populares en la investigación y la academia. CloudSim está completamente escrito en Java [29].

Se seleccionó AWS como proveedor de cómputo en la nube debido a que esta plataforma provee la capa de uso gratuita más grande, integra los servicios necesarios, y tiene la mejor accesibilidad al lenguaje elegido. Se eligió JSF debido a la facilidad de desarrollo con este marco de trabajo, ya que cuenta con una vasta documentación, además de ser gratuito y poseer la facilidad de crear aplicaciones grandes con el lenguaje puro de Java. Se optó por NetBeans como IDE debido a su interfaz intuitiva y a su gran facilidad para desarrollar aplicaciones con el marco de trabajo seleccionado. Se decidió usar UWE como metodología para el desarrollo de software, ya que provee pasos concretos dirigidos a aplicaciones Web y su gran relación con diagramas UML que son considerados estándares en la OMG (*Object Management Group*, grupo de administración de objetos). Se escogió MySQL como sistema gestor de base de datos, debido a la velocidad en sus consultas y al tener una mayor facilidad para realizar la replicación y la fragmentación de datos en AWS. Se seleccionó el *benchmark* TPC-E para validar la implementación de la aplicación, ya que este *benchmark* es adecuado para el tipo de sistemas a los que está enfocada la técnica de fragmentación, asignación y replicación. CloudSim se eligió como marco de trabajo para la simulación de cómputo en la nube, debido a que es fácil de integrar con la aplicación Web y el lenguaje, ya que está escrito en el lenguaje de programación Java.

Capítulo 3. Aplicación de la metodología

Para llevar a cabo la aplicación de la metodología este capítulo se divide en dos partes. La primera realiza una búsqueda y un análisis profundo de los métodos de fragmentación y replicación aplicados a la nube para seleccionar una técnica que cumpla con ciertos criterios. La segunda describe el diseño de la aplicación Web para realizar la ejecución de la técnica seleccionada. En este capítulo se aborda también la especificación de la técnica seleccionada y la manera en la que se desarrolló la fragmentación, replicación y asignación.

3.1 Análisis

Conforme al objetivo general y específicos, se realizó un análisis profundo de los trabajos relacionados con la fragmentación y la replicación de datos en la nube. Se llevó a cabo una búsqueda dentro de este análisis en algunas de las principales bibliotecas digitales de editoriales científicas: ACM, IEEE, Springer y Elsevier.

Como resultado del análisis, bajo algunos criterios principales, se obtuvo el método de fragmentación y replicación más adecuado para aplicar en este trabajo. Para realizar el estudio de los trabajos relacionados se siguió la metodología descrita a continuación en la Figura 3.1.

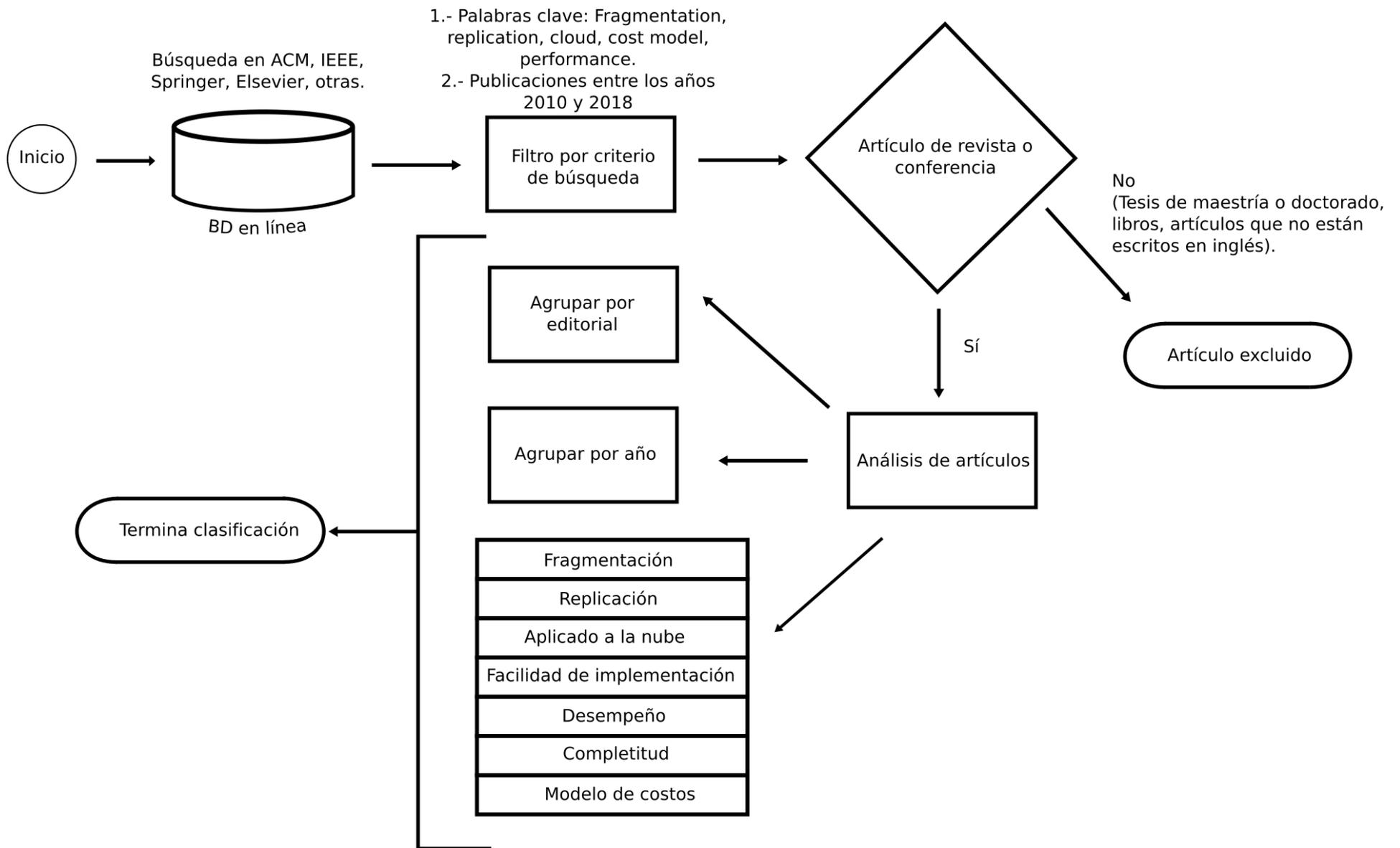


Figura 3.1. Metodología de búsqueda y evaluación de los trabajos relacionados

En la Figura 3.1 se muestran todas las etapas de la metodología propuesta. Como ya se mencionó, se realizó la búsqueda de trabajos en las principales bibliotecas digitales de editoriales científicas, ACM, IEEE, Springer y Elsevier. Los trabajos encontrados que no son publicados por dichas editoriales se categorizan en “Otras”. La búsqueda consiste en encontrar todos los trabajos que contengan las siguientes palabras clave: *Fragmentation* (Fragmentación), *replication* (replicación), *cloud* (nube), *cost model* (modelo de costo) y *performance* (desempeño). Los trabajos deben haberse publicado entre los años 2010 y 2018. Ya obtenidos todos los trabajos, se aplicó un filtro y se descartaron todos los trabajos que sean tesis de maestría o doctorado, así como libros y artículos que no estén escritos en inglés. Los artículos resultantes se clasificaron por editorial y por año. En esta última etapa se analizó cada uno de ellos bajo siete rubros principales: Fragmentación, Replicación, Aplicado a la nube, Facilidad de implementación, Desempeño, Completitud y Modelo de costos. De esta manera, sólo los trabajos que cumplieron con estos siete rubros se seleccionaron.

El proceso de selección dio como resultado 83 artículos, los cuales fueron analizados por las siete características para registrar si cumplieron cada rubro. Las Tablas 3.1-3.5 describen el registro de cada artículo y el análisis de cada característica.

Tabla 3.1. Comparación de los trabajos relacionados de la editorial ACM

Artículo	Fragmentación	Replicación	Aplicado a la nube	Facilidad de implementación	Desempeño	Completitud	Modelo de costos
Abdel et al. [30]	x	x	x	x	x	x	x
Thomson et al. [31]		x			x	x	x
López et al. [32]		x	x		x	x	x
Marcus et al. [33]	x	x	x	x	x		x
Gudadhe et al. [34]		x	x	x	x		
Dhamane et al. [35]		x	x	x	x	x	
Shwe et al. [36]		x	x	x	x		
Bollwein et al. [37]	x		x		x	x	
Xiao et al. [38]		x	x	x	x	x	
Abdul et al. [39]		x	x	x	x	x	x

Tabla 3.2. Comparación de los trabajos relacionados de la editorial IEEE

Artículo	Fragmentación	Replicación	Aplicado a la nube	Facilidad de implementación	Desempeño	Complejidad	Modelo de costos
Nuaimi et al. [40]	x	x	x	x	x		
Stiemer et al. [15]		x	x			x	x
Tziritas et al. [41]		x	x	x	x	x	x
Salunkhe et al. [42]	x	x	x	x			
Ali et al. [7]		x	x		x	x	x
Nuaimi et al. [43]		x	x	x	x	x	
Tseng et al. [44]		x	x	x	x	x	
Zhu et al. [45]		x	x	x		x	
Moral et al. [10]		x	x		x		
Mseddi et al. [46]		x	x	x	x	x	
Thara et al. [3]		x	x	x	x		
Hou et al. [47]		x	x	x	x	x	
Li et al. [48]		x	x		x		
Tripathi et al. [49]		x	x	x	x	x	
Tian et al. [50]		x	x	x		x	
Luo et al. [51]		x	x		x	x	
Xie et al. [52]		x	x	x	x	x	x
Hsu et al. [53]		x	x	x	x	x	

Tabla 3.3. Comparación de los trabajos relacionados de la editorial Elsevier

Artículo	Fragmentación	Replicación	Aplicado a la nube	Facilidad de implementación	Desempeño	Complejidad	Modelo de costos
Alami et al. [54]		x	x	x	x		
Hamrouni et al. [55]		x			x	x	
Kaur et al. [56]		x	x		x	x	x
Mansouri et al. [57]		x	x	x	x	x	x
Amer et al. [22]	x	x		x	x	x	x
Brigit [58]	x	x		x	x	x	x
Montoya et al. [59]		x	x		x		
Mansouri et al. [60]		x	x	x	x	x	x
Rosas et al. [61]	x		x	x	x	x	
Fan et al. [62]	x		x	x	x		
Gkatzikis et al. [63]		x	x	x	x	x	x
Long et al. [64]		x	x		x		x
Sengupta et al. [65]	x	x	x	x		x	x
Zeng et al. [66]		x	x	x	x	x	x
Shabeera et al. [67]			x		x	x	x
Habib et al. [68]				x	x		
Sánchez et al. [69]	x		x	x		x	
Chen et al. [70]		x	x	x	x	x	
Xia et al. [71]			x	x	x	x	x
Romero et al. [72]	x		x	x	x	x	x

Tabla 3.4. Comparación de los trabajos relacionados de la editorial Springer

Artículo	Fragmentación	Replicación	Aplicado a la nube	Facilidad de implementación	Desempeño	Complejidad	Modelo de costos
Benkrid et al. [73]	x	x	x		x	x	x
Mansouri et al. [74]		x	x	x	x	x	x
Guabtni et al. [75]			x	x	x	x	
Darabant et al. [76]		x	x	x	x	x	x
Mansouri [77]		x	x	x	x	x	x
Edwin et al. [78]		x	x	x	x	x	x
Abdel et al. [79]	x	x	x		x		
Wiese [17]	x	x	x	x	x		
Grolinger et al. [80]	x	x	x		x		
Mohammad et al. [81]	x	x	x		x		
Abdel et al. [82]	x	x	x	x	x	x	x
Abdel et al. [16]	x	x	x	x	x		
Kish et al. [83]	x	x	x	x	x		x
Zou et al. [84]	x	x	x	x	x		
Insaf et al. [85]		x	x		x		
Gao et al. [86]	x	x			x	x	
Rehman et al. [87]		x	x	x			
Bellatreche et al. [88]	x		x	x		x	
Ahirrao et al. [89]	x		x	x	x	x	
Hassen et al. [90]	x	x		x	x	x	
Lehner et al. [91]	x	x	x	x	x		

Tabla 3.5. Comparación de los trabajos relacionados de otras editoriales

Artículo	Fragmentación	Replicación	Aplicado a la nube	Facilidad de implementación	Desempeño	Complejidad	Modelo de costos
Phaphoom et al. [92]		x	x	x	x		
Wiese et al. [93]	x	x	x	x	x	x	x
Noraziah et al. [94]	x	x	x	x			
Fauzi et al. [95]	x	x	x	x			
Mrabti et al. [96]	x	x	x	x	x	x	
Khandelwal et al. [97]	x	x	x	x			
Singh et al. [98]		x	x	x			
Subashini y Kavitha [99]	x		x	x		x	x
Scheidgen et al. [100]	x			x	x		x
Hauglid et al. [101]	x	x		x	x	x	x
Das et al. [102]	x	x	x	x	x	x	
Wiese [103]	x	x		x			
Bermbach et al. [104]		x	x	x	x		
Barsoum et al. [105]	x		x	x	x	x	

Se observa en las Tablas 3.1 a 3.5 la comparación de todos los trabajos encontrados en las principales bibliotecas científicas digitales, que abordaron la mayoría de los temas de interés. Cada artículo se evaluó utilizando una metodología para determinar si cada artículo cumple con cada una de las características. De todos los trabajos analizados se observa que pocos artículos cumplen con todas las cualidades deseadas. Los artículos [30], [82] y [93] cumplen con todas las características. En el Capítulo 4 se muestran algunas gráficas que muestran el resultado del análisis y sustentan la selección del trabajo elegido.

3.2. Selección

Los tres artículos ([30], [82] y [93]) que involucran fragmentación y replicación, se aplican a la nube, son fáciles de implementar, se centran en el rendimiento, tienen completitud e incluyen un modelo de costos, se evaluaron y dos de ellos fueron descartados. En [30], es necesario evaluar las relaciones con respecto a los casos de uso que proporciona el mismo usuario, para esto se necesita más información de la se desea que el usuario capture en el sistema propuesto. En un entorno simulado, el trabajo en [30] se desarrollaría correctamente, pero en un entorno real no cumple con lo esperado. En [93] para llevar a cabo la fragmentación, los autores utilizan consultas flexibles, un tema que no se incluye en la delimitación de esta investigación.

El artículo elegido, [82], presenta una metodología deseada y una facilidad de implementación de la técnica para incluir los costos en la asignación y la replicación. La Figura 3.2 muestra el flujo de trabajo propuesto por [82].

La fragmentación y la replicación en [82] se basan en el costo de crear, leer, actualizar y eliminar los predicados incluidos en una matriz llamada MCRUD (*Creation, Reading, Updating and Deletion Matrix*; Matriz de Creación, Lectura, Actualización y Eliminación), esta matriz registra cada operación de un predicado en cada sitio, de esta manera, obteniendo el costo de cada operación, se determina cómo se fragmentará la relación, dónde se asignarán los fragmentos y en qué sitio se replicarán.

La Figura 3.2 muestra el flujo de trabajo que consta de seis pasos principales. El primer paso es el análisis de la información de las consultas extraídas del registro previamente proporcionado por el usuario. En este primer paso, se construye la MCRUD, que se obtiene al colocar las acciones de creación, lectura, actualización y eliminación de cada sitio con respecto al predicado de las consultas. El segundo paso es crear la tabla ALP (Precedencia de Localidad de Atributos) obtenida a través del costo de cada sitio. El costo de cada sitio está determinado por las funciones de costo descritas en [82]. Una vez que se construye la tabla ALP, en el tercer paso, el esquema de fragmentación horizontal se crea de acuerdo al atributo con el costo de operación más alto.

El cuarto paso es obtener una tabla con los resultados finales de asignación y replicación, donde se muestra para cada fragmento, el sitio donde se asignará y el sitio donde se replicará. Esta información se determina con la matriz MCRUD, cada fragmento se asigna al sitio donde tiene el mayor costo de operaciones y se replica en el sitio donde se realizan más operaciones de lectura. El quinto paso es aplicar la fragmentación, la asignación y la replicación de acuerdo con los esquemas descritos. El sexto paso muestra el resultado de este flujo de trabajo, el cual es cada sitio con los fragmentos y las réplicas.

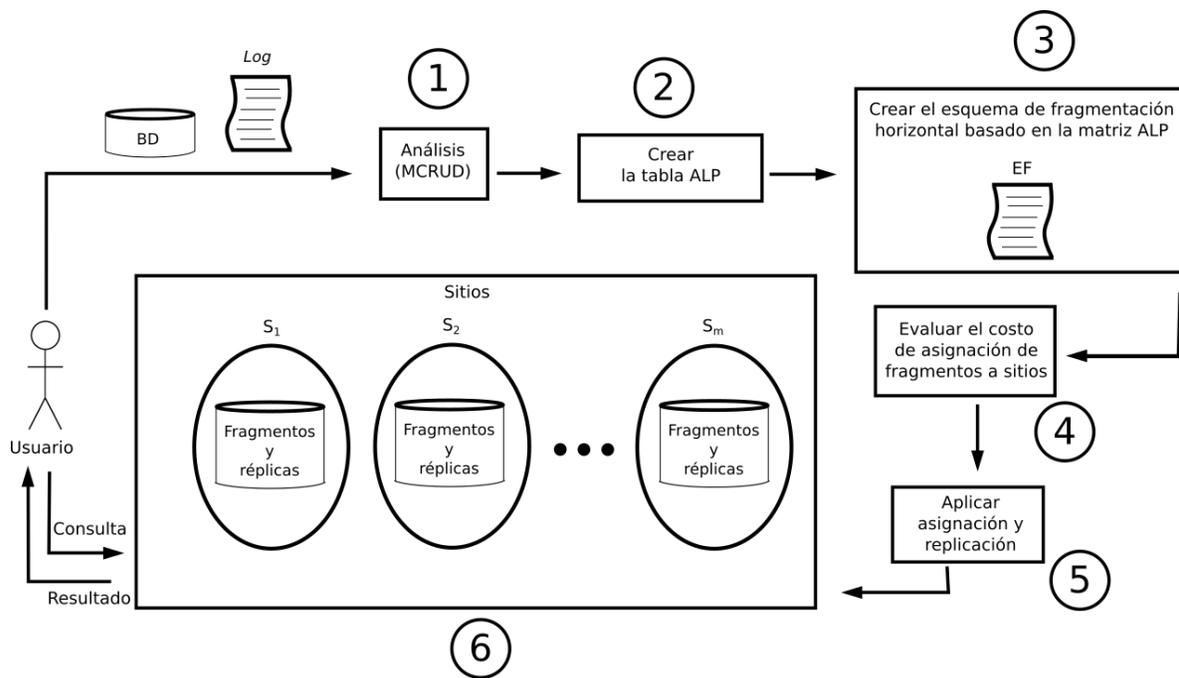


Figura 3.2. Flujo de trabajo

3.3 Implementación

En esta sección se describe el diseño de la aplicación siguiendo la metodología de desarrollo. Se llevan a cabo todos los diagramas y pasos de la metodología elegida para el desarrollo de la aplicación Web.

La Figura 3.3 representa la arquitectura elegida la cual es “Modelo Vista Controlador” (MVC): en la vista están las páginas que interactúan con el usuario para solicitar información sobre la base de datos y cómo acceder a ella. Los *beans* administrados de JSF (Java Server Faces) que

gestionan el flujo de información se encuentran en el controlador. En el modelo, se definen las reglas negocio, que en este escenario particular será la aplicación de la técnica de fragmentación y replicación seleccionada, contemplado el cálculo del modelo de costos de la transformación realizada para una base de datos determinada. Para llevar a cabo la validación de la técnica seleccionada, se construyó una aplicación Web para realizar la fragmentación y la replicación automáticamente.

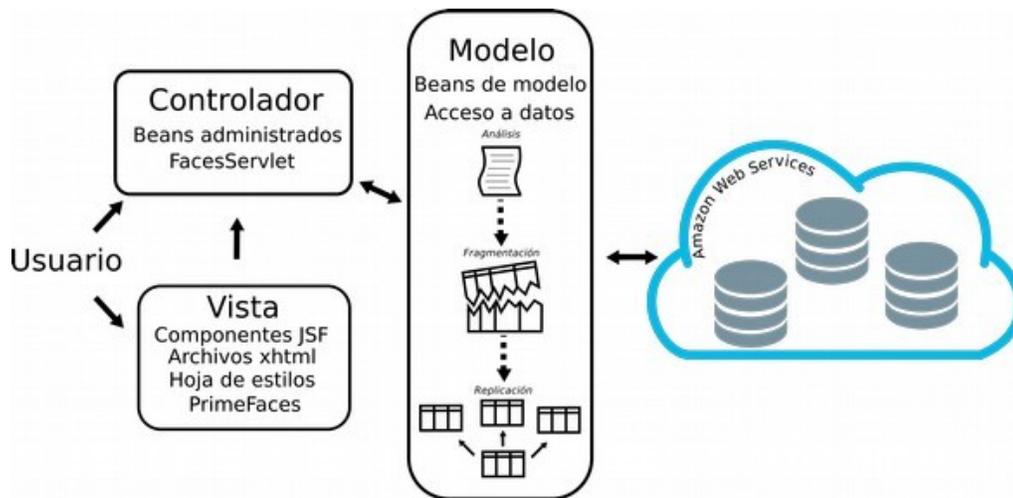


Figura 3.3. Arquitectura para la fragmentación y replicación de datos en la nube

3.3.1 Análisis de requisitos

El objetivo del análisis de requisitos es establecer los requisitos funcionales de la aplicación web, estos requisitos se representan mediante diagramas. El diagrama de casos de uso representa la relación actor-actividades, muestra las actividades a las que está relacionado cada actor.

En la Tabla 3.6 se definen los actores de la aplicación y una breve descripción del papel que desempeñan.

Tabla 3.6. Actores de la aplicación Web

Actor	Descripción
Administrador de la Base de Datos (DBA)	El DBA es el único actor en el sistema y su única función es solicitar la fragmentación proporcionando los archivos correspondientes para su proceso.

Una vez ya definido el único actor en la Tabla 3.6, en la Figura 3.4 se muestra el diagrama de casos de uso, el cual relaciona al actor con sus actividades. Como se observa, solo está relacionado con una actividad.

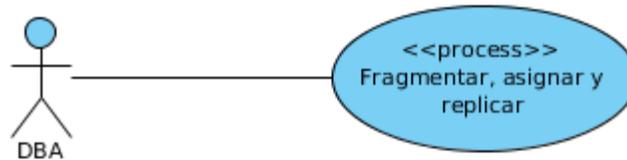


Figura 3.4. Diagrama de Casos de uso de la aplicación Web

En la Figura 3.4 se muestra el caso de uso “Fragmentar, asignar y replicar” con el estereotipo “Process” (Proceso), como lo sugiere la metodología UWE. Este caso de uso se describe con un diagrama de actividades. Se observa tal diagrama en la Figura 3.5.

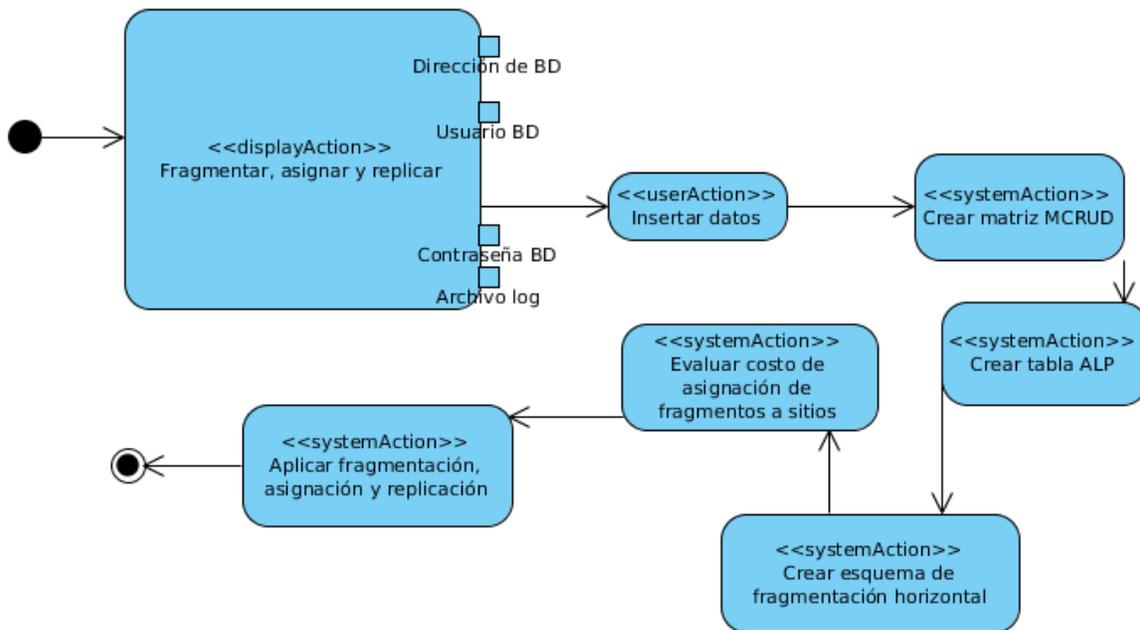


Figura 3.5. Diagrama de actividad del caso de uso “Fragmentar, asignar y replicar”

El diagrama de actividad del caso de uso “fragmentar, asignar y replicar” muestra como cuadro de acción el formulario para capturar la dirección de la base de datos, el usuario, la contraseña y el archivo de carga, ya que el análisis estará basado en este. El segundo cuadro de acción muestra

la actividad del usuario al capturar la información solicitada. En el tercer cuadro de acción se muestra la primera actividad del sistema, crear la matriz de creaciones, lecturas, actualizaciones y eliminaciones (MCRUD). La siguiente actividad es crear la tabla de precedencia de localidad de atributos (ALP), por medio de esta tabla se determina el atributo por el cual se fragmentará cada relación. En el siguiente cuadro de actividad se realiza el esquema de fragmentación horizontal. En la actividad siguiente se evalúa el costo de asignación de fragmentos a sitios, de esta forma se determina en dónde estará cada fragmento y en dónde será replicado. Como última actividad se aplica el esquema de fragmentación hecho previamente, se asignan los fragmentos a los sitios y se replican.

3.3.2 Modelo conceptual

El modelo conceptual, también conocido como modelo de dominio, se encarga de describir cómo se relacionan los requisitos de la aplicación y de esta manera obtener el comportamiento del sistema. El modelo conceptual en este trabajo, se representará por medio de tres diagramas principales: diagrama conceptual, diagrama lógico y diagrama físico de la base de datos. Se observan estos diagramas en las Figuras 3.6, 3.7 y 3.8.

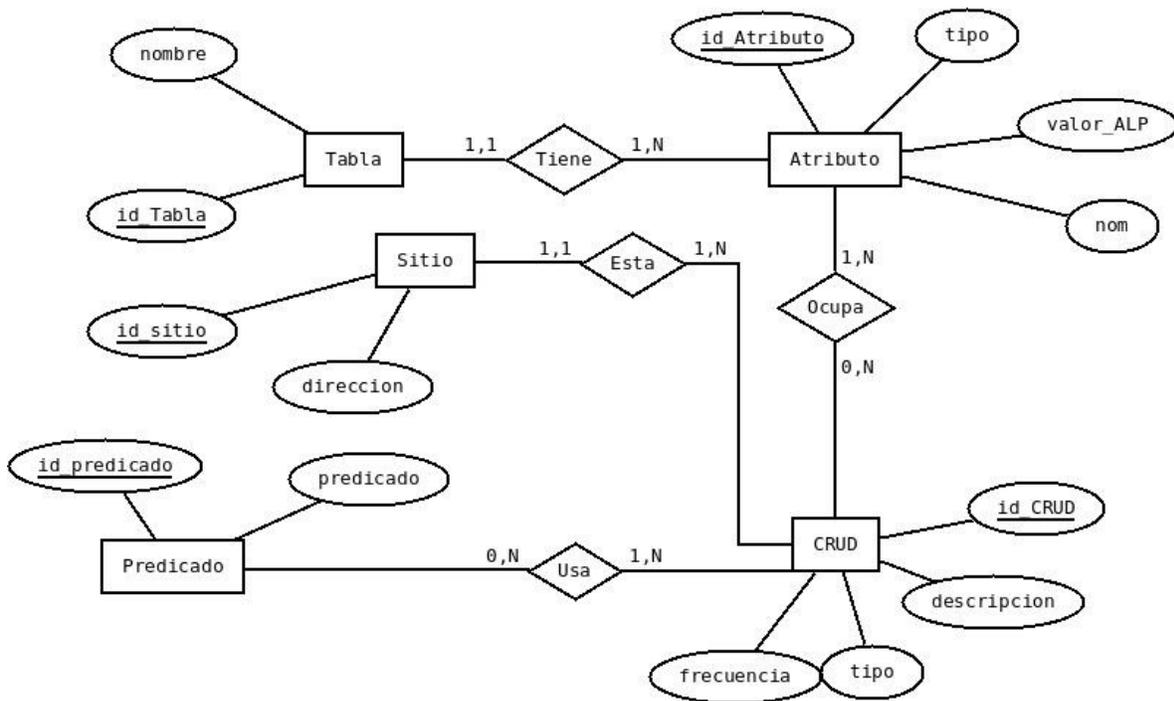


Figura 3.6. Diagrama conceptual de la aplicación

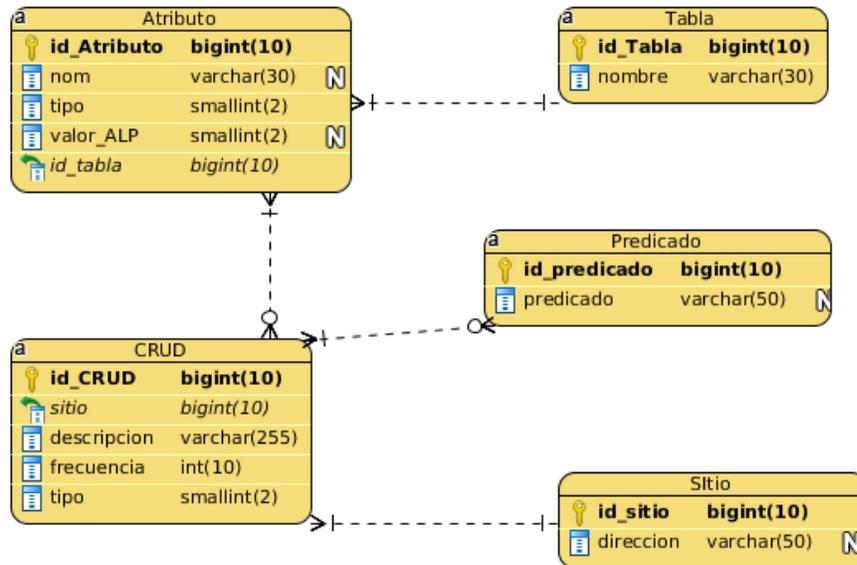


Figura 3.7. Diagrama lógico de la aplicación

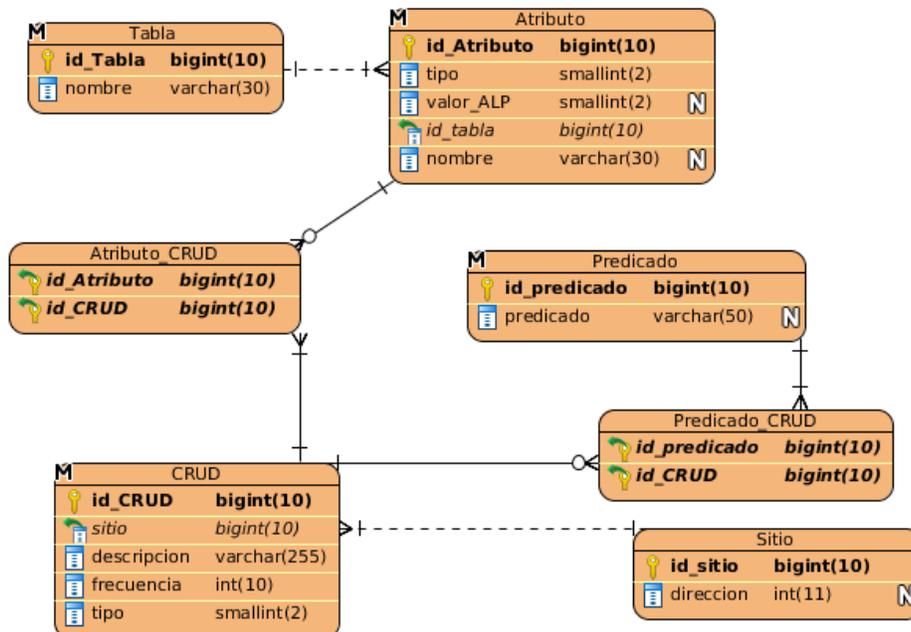


Figura 3.8. Diagrama físico de la aplicación.

Tablas de base de datos

Tabla: La relación *Tabla* es la encargada de almacenar los nombres de todas las tablas obtenidas de la estructura de base de datos proporcionada por el usuario en el archivo de carga.

Atributo: La tabla *Atributo* almacena para cada tabla todos sus atributos, indicando su tipo de dato en la columna *tipo*, su nombre en la columna *nombre* y la cifra resultante en la columna *valor_ALP* después de obtener los valores ALP.

CRUD: La tabla *CRUD* es la encargada de almacenar cualquier operación registrada en el archivo de carga que se relacione con la tabla elegida para realizar el proceso. Se registra en esta tabla cualquier creación, lectura, actualización y eliminación hecha sobre la tabla propuesta por el DBA. En la tabla *CRUD* se almacena la operación en el atributo descripción, se almacena el tipo de operación, su frecuencia de aparición y el sitio en donde se ejecuta.

Atributo_CRUD: La tabla *Atributo_CRUD* relaciona todos los atributos utilizados en cada operación *CRUD*, de esta manera un atributo tiene la oportunidad de estar presente en muchas operaciones *CRUD* y una operación *CRUD* de utilizar muchos atributos.

Predicado: La tabla *Predicado* guarda todos los predicados de cada *CRUD* que se relaciona con la tabla que el DBA seleccionó para ser fragmentada, replicada y asignada. Esta tabla es de mucha importancia para aplicar el método seleccionado, ya que por medio de ella se construirá la matriz *MCRUD*.

Predicado_CRUD: Esta tabla se encarga de mantener la cardinalidad entre las tablas *Predicado* y *CRUD*, ya que un *CRUD* posee la característica de tener muchos predicados y un predicado de estar presente en muchos *CRUD*.

Sitio: La tabla *sitio* se encarga de almacenar todos los sitios a donde se hayan ejecutado *CRUD*'s relacionados con la tabla seleccionada por el DBA.

3.3.3 Modelo de navegación

Mediante el modelo de navegación se conocen todos los caminos posibles dentro de la aplicación para la navegación de los usuarios. Este modelo propone crear mapas de navegación mediante diagramas de clase estereotipados, por medio de estos diagramas se describen todas las rutas posibles por las que se mueve un usuario dentro de un sitio o de una aplicación. En la Figura 3.9 se muestra el modelo de navegación de la aplicación.

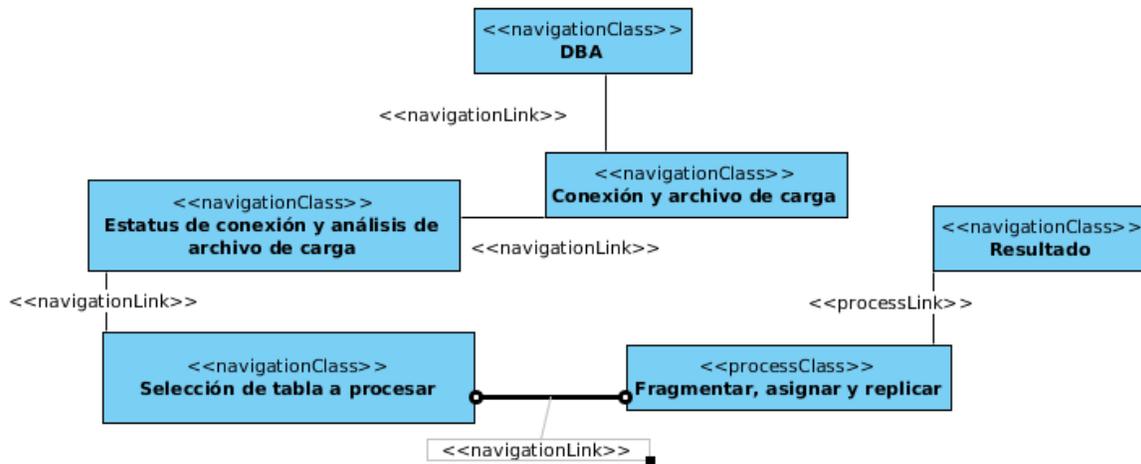


Figura 3.9. Modelo de navegación de la aplicación Web

En la Figura 3.9 se muestra el modelo de navegación del usuario DBA. La primera clase de navegación después del rol DBA es “Conexión y archivo de carga”, la cual es la encargada de mostrar el formulario para la obtención del archivo de carga, la dirección y las credenciales de la base de datos que se desea fragmentar y replicar. Posteriormente en la siguiente clase de navegación se encuentra el estatus de la conexión y el análisis del archivo de carga. Cuando la conexión a la base de datos sea exitosa el siguiente paso es seleccionar la tabla a analizar y así entrar al proceso de fragmentar, asignar y replicar, el cual se muestra en la clase de proceso siguiente. Por último, la clase de navegación “Resultado” muestra el resultado de la fragmentación, asignación y replicación. La seguridad en la aplicación la proveen las bases de datos, ya que se accede a ellas por medio de algún rol descrito dentro de las mismas bases de datos.

En la Figura 3.10 se observa la página de *Conexión y archivo de carga* del modelo de presentación, la cual solo la conforma el formulario inicial para la obtención de la información requerida. En el recuadro con el estereotipo *inputForm* hay cuatro recuadros para la solicitud de información al usuario. El primer recuadro representa el archivo de carga, el cual debe cumplir ciertas características. El segundo recuadro representa la dirección de la base de datos, la cual se conforma por la dirección *IP*, el puerto de conexión y el nombre de la base de datos. Los siguientes dos recuadros son para solicitar el nombre de usuario y la contraseña. Por último, el botón probar conexión es el encargado de continuar la navegación al análisis de la información introducida.

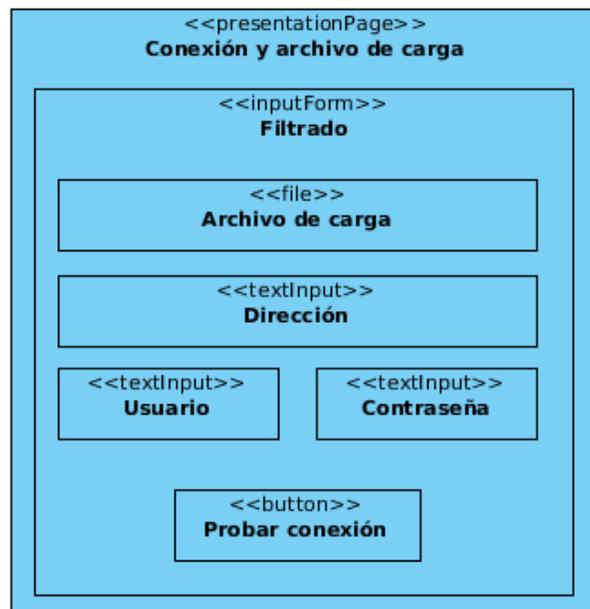


Figura 3.10. Página *Conexión y archivo de carga* del modelo de presentación

En la Figura 3.11 se muestra la página *Estatus de conexión y análisis del archivo de carga* del modelo de presentación. El objetivo de esta página es mostrar el estatus de los datos ingresados, validando la conexión a la base de datos que el usuario indicó y verificar que el archivo de carga posea el formato adecuado para su análisis. Si el estatus de la conexión es exitoso, en el recuadro de texto *Información de la base de datos y el archivo de carga* se desglosará una descripción austera de la base de datos. En este mismo recuadro se mostrará si el archivo de carga fue válido. En la parte inferior hay dos botones, el primer botón es para regresar a la pantalla inicial y el

segundo es para aplicar la fragmentación, la asignación y la replicación. Este último botón permanecerá deshabilitado si el estatus de la conexión no es exitoso o si el archivo de carga no cumple con el formato adecuado.

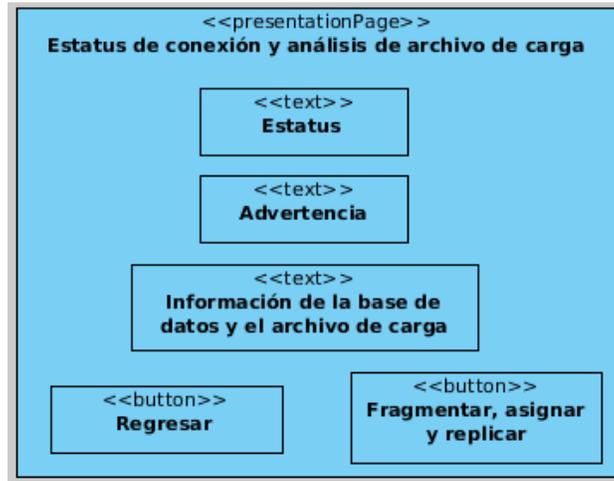


Figura 3.11. Página *Estatus de conexión y análisis de archivo de carga* del modelo de presentación

Se muestra en la Figura 3.12 la página *Selección de tabla a procesar*, la cual se encarga de mostrar todas las tablas creadas por el usuario dentro de la base de datos con la que se hizo conexión. Las tablas se muestran en una lista, y se selecciona una tabla para que se aplique el método de fragmentación, asignación y replicación.

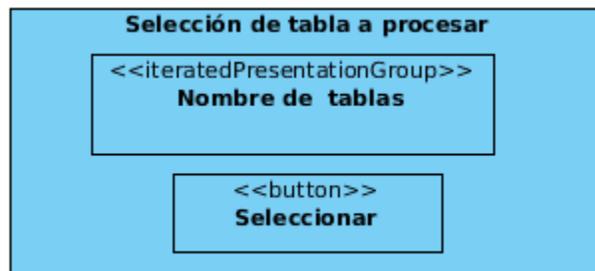


Figura 3.12. Página *Selección de tabla a procesar* del modelo de presentación

En la Figura 3.13 se presenta la página *Resultado* del modelo de presentación. El objetivo de esta página es mostrar el estatus del método de fragmentación, asignación y replicación aplicado a la base de datos indicada. El segundo recuadro, llamado *Directorio*, representa el archivo en donde se almacenará la información de la ubicación de los fragmentos y las réplicas, el cual se descargará si el estatus del proceso fue correcto.

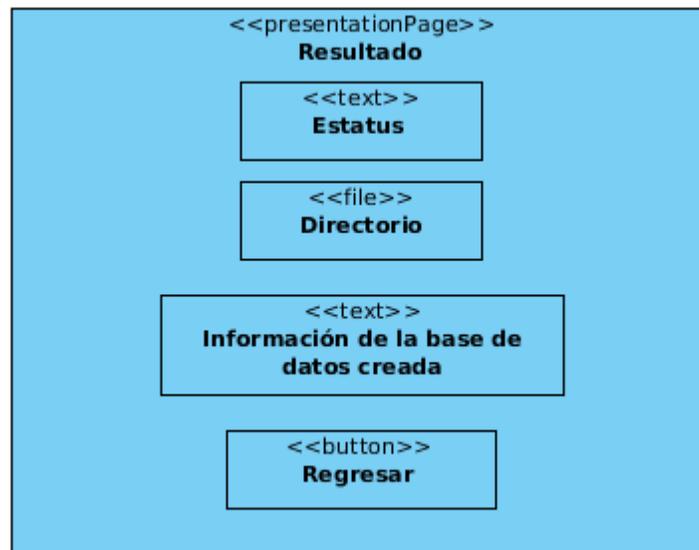


Figura 3.13. Página *Resultado* del modelo de presentación

3.3.4 Modelo de procesos

El diagrama de procesos se basa en el diagrama de actividades, agregando diversos controles que enriquecen la especificación de la aplicación. En la Figura 3.14 se presenta el diagrama de proceso de la aplicación, agregando diversos componentes como los nodos de decisión y la representación de los objetos dentro del flujo de actividades.

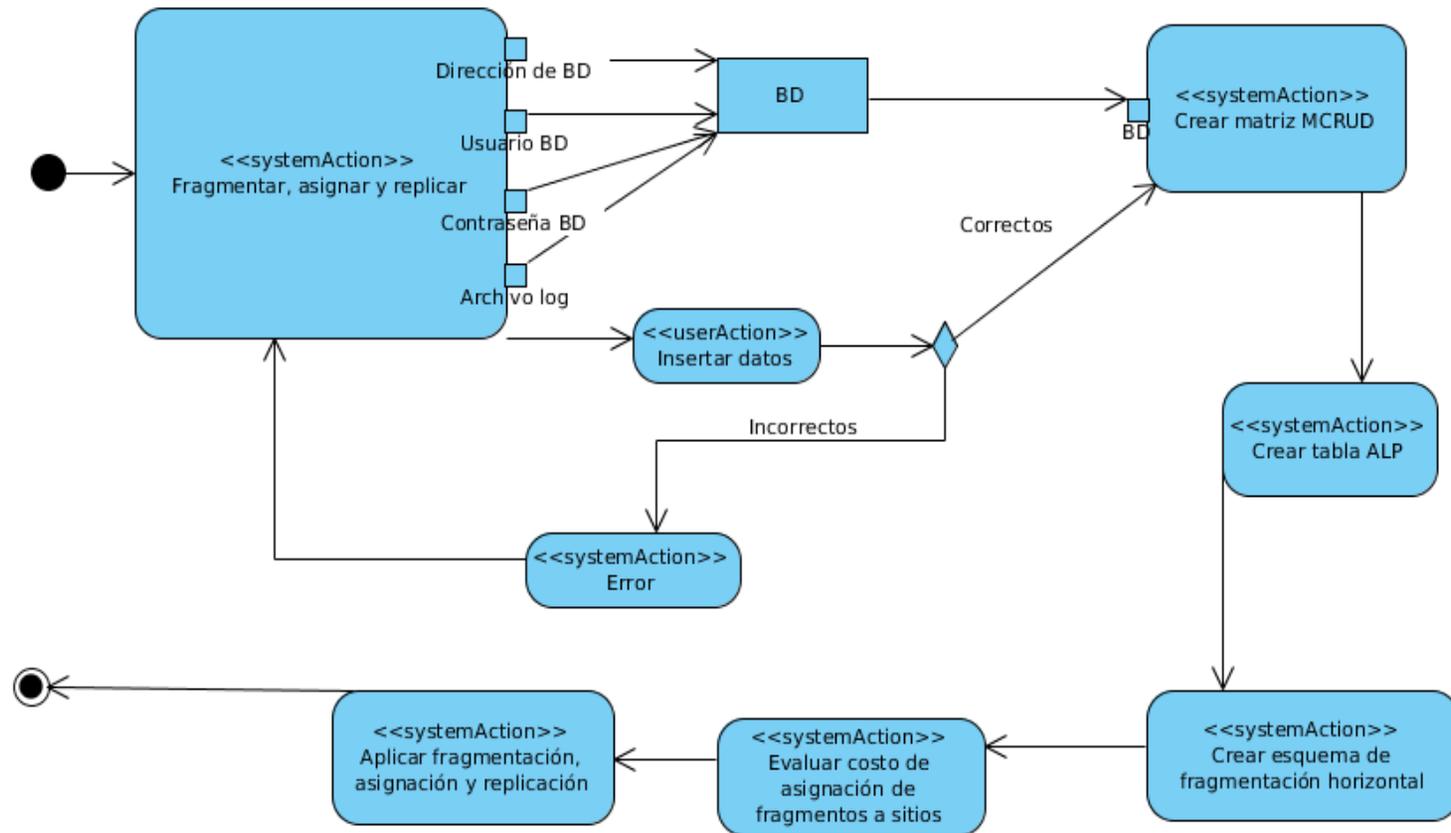


Figura 3.14. Diagrama *Fragmentar, asignar y replicar* del modelo de procesos

El diagrama de procesos *Fragmentar asignar y replicar* comienza desplegando un formulario para la captura de la dirección de base de datos, usuario y contraseña del administrador y el archivo de carga. Se muestra posteriormente en este diagrama la representación del objeto *BD* (Base de datos), el cual se utiliza como parámetro en la acción del sistema *Crear matriz MCRUD*. Dentro del mismo recuadro *Insertar datos* se contempla la selección de la tabla a fragmentar. Esto ocurrirá solo si son correctos los datos insertados en el cuadro de acción de usuario *Insertar datos*. Si los datos son incorrectos, se despliega en pantalla un mensaje de error y el flujo retorna al formulario inicial. En la

acción del sistema *Crear matriz MCRUD* no solo interviene el objeto *BD*, ya que, al tener una conexión exitosa, los demás objetos (*Tabla*, *Atributo* y *CRUD*, entre otros) se utilizan en este cuadro de acción del sistema y en los siguientes cuatro.

Después de crear la matriz MCRUD, la siguiente acción del sistema es *Crear tabla ALP*, la cual determinará el o los atributos que se utilizarán para que las tablas se fragmenten. Posteriormente, la acción del sistema *Crear esquema de fragmentación horizontal* utilizará dichos atributos para llevar a cabo su objetivo. El siguiente cuadro, *Evaluar costo de asignación de fragmentos a sitios*, dará como resultado una tabla que indicará el sitio en el que se colocará cada fragmento y su réplica. Por último, se aplicará la fragmentación, asignación y replicación tomando en cuenta los esquemas y tablas que se obtuvieron.

3.3.5. Algoritmo

Para extender el trabajo de Abdel et. al. [82], en esta sección se presenta un problema común de predicados traslapados, la matriz MCRUD relaciona cada sitio con todos los predicados simples usados por las consultas ejecutadas en esos sitios. Por ejemplo, se tienen dos predicados del mismo atributo, el cual determina el esquema de fragmentación: *año<1996* y *año<2000*. Para obtener cada fragmento a través de los predicados es necesario negar todos los predicados excepto el relacionado al fragmento que se quiere obtener. Si el fragmento que se obtendrá es *año<1996*, negando *año<2000* implícitamente se niega *año<1996*, ya que el predicado más grande contiene al más pequeño y debido a esto se obtendrá un fragmento sin tuplas. Para resolver el traslape de predicados en el proceso de fragmentación se eligió considerar solo los predicados más grandes, es decir, si un predicado contiene otro, solo el fragmento contenedor será contemplado.

La Figura 3.15 representa el pseudocódigo para obtener la matriz de predicados, donde las filas son el conjunto de predicados para obtener cada fragmento. De esta manera, colocando los predicados de cada fila dentro de una consulta, al ser ejecutada se obtendrá cada fragmento con sus correspondientes tuplas.

Algoritmo 1: Fragmentación con traslape

```

1 input: Lista de predicados[], Nombre de la tabla
2 output: Lista de fragmentos
3 for  $x$  from 0 to size of Lista de predicados[] do
4   eliminar todo en línea
5   for  $y$  from 0 to size of Lista de predicados[] do
6     if Lista de predicados[ $y$ ]=Lista de predicados[ $x$ ] then
7       línea[ $y$ ] $\leftarrow$  "(" + Lista de predicados[ $y$ ] + ")"
8     else
9       línea[ $y$ ] $\leftarrow$  "!(" + Lista de predicados[ $y$ ] + ")"
10  predicados de fragmentos[ $x$ ][] $\leftarrow$  línea[]
11  bandera $\leftarrow$  false
12  while bandera=false do
13    encontrado $\leftarrow$  false;
14    línea de encontrado $\leftarrow$  0;
15    for  $i$  from 0 to size of predicados de fragmentos[][] while encontrado=false do
16      consulta  $\leftarrow$  "select * from " + Nombre de la tabla + " where "
17      for  $j$  from 0 to size of predicados de fragmentos [ $i$ ][] do
18        consulta $\leftarrow$  consulta + predicados de fragmentos[ $i$ ][ $j$ ] + " && "
19      consulta $\leftarrow$  consulta - los últimos tres caracteres
20      consulta $\leftarrow$  consulta+" limit 1;"
21      conjunto resultado $\leftarrow$  execute consulta in database
22      if conjunto resultado is empty then
23        línea de encontrado $\leftarrow$   $i$ 
24        encontrado $\leftarrow$  true
25      if encontrado=true then
26        índice $\leftarrow$  0
27        for  $q$  from 0 to size of predicados de fragmentos[línea de encontrado][] do
28          if el primer carácter de predicados de fragmentos[línea de encontrado][ $q$ ] is "!" then
29            índice $\leftarrow$   $q$ ;
30          eliminar línea de encontrado en predicados de fragmentos[][]
31          for  $q$  from 0 to size of predicados de fragmentos[][] do
32            eliminar índice en predicados de fragmentos[ $q$ ][]
33          else
34            bandera $\leftarrow$  true
35      consulta $\leftarrow$  "select * from " + Nombre de la tabla + " where "
36      eliminar todo en línea
37      foreach  $x$  $\leftarrow$  predicado in Lista de predicados[],  $0 \leq i \leq$  size of Lista de predicados[] do
38        consulta $\leftarrow$  consulta + "(" +  $x$  + " ) && "
39      consulta $\leftarrow$  consulta - los últimos tres caracteres
40      consulta $\leftarrow$  consulta+" limit 1;"
41      conjunto resultado $\leftarrow$  execute consulta in database
42      if conjunto resultado is not empty then
43        for  $x$  from 0 to size of Lista de predicados[] do
44          línea[ $x$ ] $\leftarrow$  "(" + Lista de predicados[ $x$ ] + ")"
45        predicados de fragmentos[size of predicados de fragmentos][] $\leftarrow$  línea[]
46      return predicados de fragmentos[][]

```

Figura 3.15. Fragmentación con traslape

En la Figura 3.15, “predicados de fragmentos” es la matriz de predicados. Si la lista de predicados de entrada contiene $id < 30$, $id < 20$, $id = 40$ de una tabla llamada “cuentas”, la cual tiene tuplas con “id” de 1 a 50, como resultado, “predicados de fragmentos” contendrá la matriz mostrada en la Figura 3.16.

En la primera iteración, la consulta “Select * from accounts where $(id < 30) \ \&\& \ ! (id < 20) \ \&\& \ ! (id = 40)$ limit 1” obtendrá una tupla, y es por esto que en esta iteración ningún predicado es eliminado. En la segunda iteración, la consulta “Select * from accounts where $! (id < 30) \ \&\& \ (id < 20) \ \&\& \ ! (id = 40)$ limit 1” obtendrá un conjunto vacío, la iteración corresponde al predicado simple $(id < 20)$ y este se elimina de toda la matriz con sus predicados compuestos. Una vez que un predicado se ha eliminado, todas las iteraciones son ejecutadas nuevamente desde el inicio debido a que todos los predicados compuestos han sido modificados, ya que no contienen al predicado que fue eliminado. Las siguientes dos iteraciones regresan tuplas y por esto son preservadas.

$(id < 30)$	$! (id = 40)$
$! (id < 30)$	$(id = 40)$
$! (id < 30)$	$! (id = 40)$

Figura 3.16. Resultado del ejemplo

3.4 Validación

Como parte de esta investigación se desarrolló una aplicación Web nombrada FRAGMENT, desde la cual se aplica el método elegido de fragmentación, asignación y replicación. La información generada por medio de este método se desglosa en la aplicación y si se desea, el esquema propuesto por la aplicación se aplica físicamente a lo largo de los sitios detectados. Esta aplicación logra mejorar el tiempo de respuesta de las consultas hechas en cada uno de los sitios. La aplicación implementa el método elegido, el cual toma en cuenta la columna más costosa de la tabla que se tratará, teniendo de esta manera un esquema más barato operativamente.

La aplicación permite analizar un archivo de carga proporcionado para extraer la información indispensable para el método, conectarse al servidor remotamente, seleccionar una tabla de todo el esquema encontrado en el gestor de bases de datos, mostrar resultados preliminares del análisis, aplicar el esquema a lo largo de los sitios y mostrar resultados representativos finales. En el capítulo 4 se aborda detalladamente la descripción de la aplicación Web y de igual manera una aplicación Web secundaria dedicada a la simulación de consultas.

Para validar el método implementado se presentan dos casos de estudio en el Capítulo 4. El primer caso de estudio se basa en la simulación de consultas, operaciones y sitios sobre el *benchmark* TPC-E para posteriormente observar el funcionamiento de la aplicación principal. El segundo caso de estudio presenta un ambiente real en la nube sobre *Amazon Web Services*, se realizan una serie de consultas locales y remotas sobre el servidor principal. Por medio de este escenario se prueba nuevamente el método implementado y la generación real de los fragmentos.

Capítulo 4. Resultados

En este capítulo se muestra el resultado del análisis, se describe el funcionamiento de la aplicación Web, así como la especificación de una segunda aplicación dedicada a la simulación de consultas con el marco de trabajo CloudSim para la base de datos del *benchmark* TPC-E y por último se muestran los resultados obtenidos con base en el caso de estudio del mismo *benchmark* para sitios creados y configurados en EC2 de *Amazon Web Services*.

4.1. Resultado del análisis

Para finalizar la etapa del análisis, en esta sección se muestran los resultados mediante gráficas y tablas. De esta manera se observa con mayor claridad otro enfoque de los datos mostrados en las Tablas 3.1 a 3.5.

La Figura 4.1 muestra una gráfica de barras que relaciona el número de artículos con la editorial; la mayoría de los artículos se concentran en la editorial Springer. Los artículos de Springer abordaron principalmente la replicación de datos ([73], [74], [76] - [87], [90], [91]), 18 de 21 artículos. 12 artículos consideraron tanto la fragmentación como la replicación juntas ([73], [79] - [84], [86], [90] y [91]). Las editoriales se evaluaron en relación con la proximidad de los artículos con las características deseadas, Springer ocupó el segundo lugar. ACM es la editorial con la calificación más alta e IEEE la más baja. La Figura 4.2 muestra los resultados de la evaluación de proximidad con los temas de interés de cada editorial.

En la Figura 4.3 se aprecia la distribución de los artículos analizados según el año de publicación. Se observa que en el año 2017 hubo un aumento en el número de artículos. Se visualiza que de los 18 artículos, solo dos pertenecen a Springer ([76] y [78]), la editorial con el mayor número de artículos en la Figura 4.1, siete pertenecen a Elsevier ([22], [59], [60], [63], [67], [69] y [71]), cuatro a ACM ([32], [34], [36] y [37]), cuatro a IEEE ([41], [48], [52] y [53]) y uno a la categoría "otros" ([93]). En 2013, el número de artículos de los tres años anteriores se duplicó, ya que el número de trabajos que consideran la fragmentación y la replicación en conjunto aumentó cuatro veces más ([101] en los tres años anteriores; [80], [91], [94], y [97] en 2013). En 2017, 12 artículos de los 18 abordaron la replicación de datos ([32], [34], [36], [41],

[48], [52], [53], [60], [63], [76], [78], y [93]).

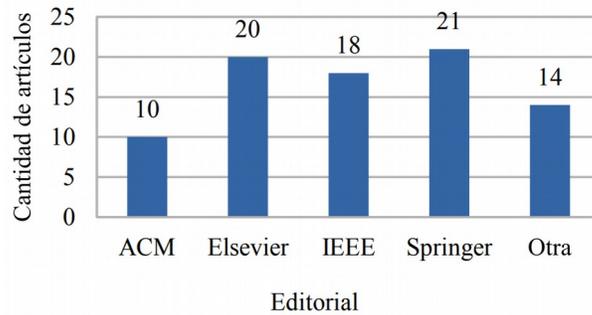


Figura 4.1. Cantidad de artículos por editorial

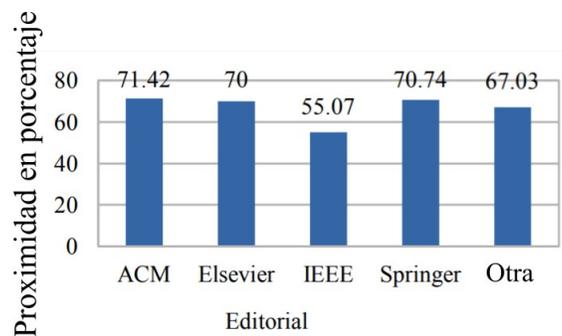


Figura 4.2. Proximidad a los tópicos de interés de cada editorial



Figura 4.3. Cantidad de artículos por año

En la Figura 4.4 se compara la cantidad de artículos que consideran la fragmentación de datos ([30], [33], [37], [40], [42], [22], [58], [61], [62], [65], [69], [72], [73], [79] - [84], [86], [88] - [91], [93] - [97], [99] - [103], y [105]) y el número de trabajos que tienen en cuenta la replicación

de datos ([30] - [36], [38] - [60], [63] - [66], [70], [73], [74], [76] - [87], [90] - [98] y [101] - [104]). Se distribuyen 27 artículos entre los dos temas que contienen tanto a la fragmentación como a la replicación. En la Figura 4.4 se observa que la replicación es un tema más común entre los 83 artículos.

La Figura 4.5 muestra un gráfico de barras que relaciona el número de artículos por tipo de fragmentación. El número de artículos que incluyen fragmentación es 37, como se muestra en la Figura 4.4, sin embargo, hay artículos que incluyen más de un tipo de fragmentación ([80], [88], [90], [91], [93] - [95] y [97]), por lo que en el gráfico de la Figura 4.5 el número total de trabajos no es igual al total de artículos de fragmentación. La mayoría de los artículos que incluyen más de un tipo de fragmentación no entran en la categoría "Híbrido", porque cada tipo lo usa por separado o solo contempla esos tipos en su trabajo para resolver otros problemas ([80], [88], [91], [93] - [95], y [97]). En la categoría "Otros" están todos los artículos que hablan de algún tipo de fragmentación que no es ninguna de las otras categorías ([33], [40], [42], [58], [61], [65], [69], [83], [84], [86], [89], [96], [99], [100] y [105]).

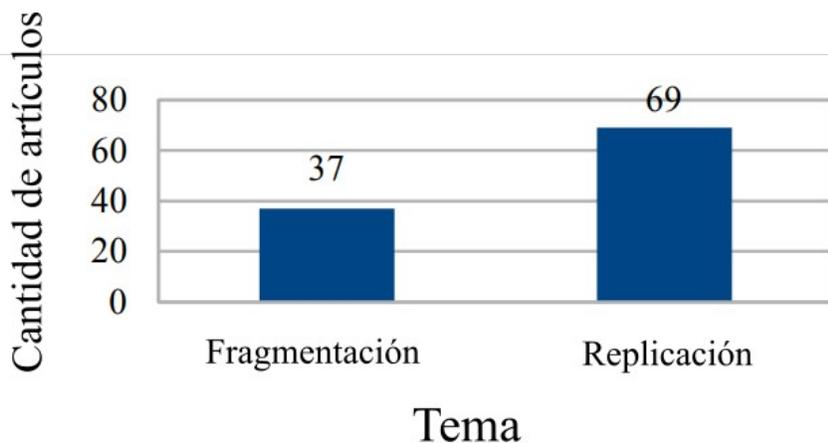


Figura 4.4. Cantidad de artículos por tema

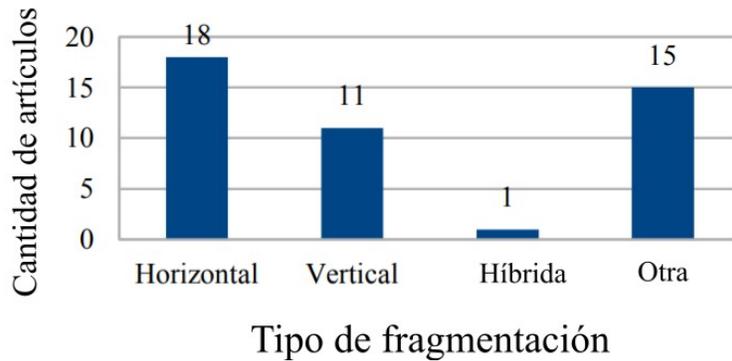


Figura 4.5. Cantidad de artículos por tipo de fragmentación

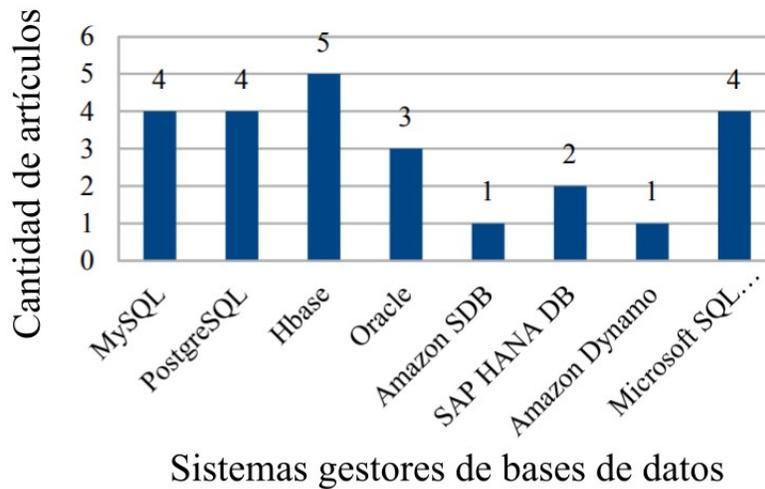


Figura 4.6. Cantidad de artículos por SGBD

En la Figura 4.6 se describe la cantidad de trabajos por sistema gestor de base de datos (SGBD). La mayoría de los artículos no mencionan o no usan SGBD para llevar a cabo su investigación. La Figura 4.6 muestra que los cuatro SGBD más utilizados son PostgreSQL, Hbase, MySQL y Microsoft SQL Server. Algunos artículos incluyen más de un sistema, como [35], porque usa MySQL y PostgreSQL. La Tabla 4.1 muestra cada artículo incluido en la Figura 4.6 y con qué sistema está relacionado.

Tabla 4.1. Artículos por sistema gestor de base de datos

SGBDs	Artículos
Mysql	[35], [75], [78], [84]
PostgreSQL	[17], [33], [35], [37]
Hbase	[38], [68], [72], [89], [99]
Oracle	[47], [88], [90]
Amazon Simple DB	[89]
SAP HANA DB	[93], [102]
Amazon Dynamo	[103]
Microsoft SQL Server	[16], [79], [82], [92]

4.2. *Benchmark* TPC-E

La página oficial de TPC proporciona el *benchmark* TPC-E mediante un *software* nombrado Egen, el cual genera todos los archivos y datos necesarios para la creación del esquema de la base de datos y la población de cada tabla para un SGBD genérico. La aplicación del esquema y la población de las tablas tienen una definición específica, sin embargo, no corresponden específicamente al SGBD utilizado en este proyecto, por esta razón se realizó una adaptación del esquema genérico al adecuado para la creación de la base de datos en MySQL. En la Figura 4.7 se muestra el esquema de la base de datos TPC-E.

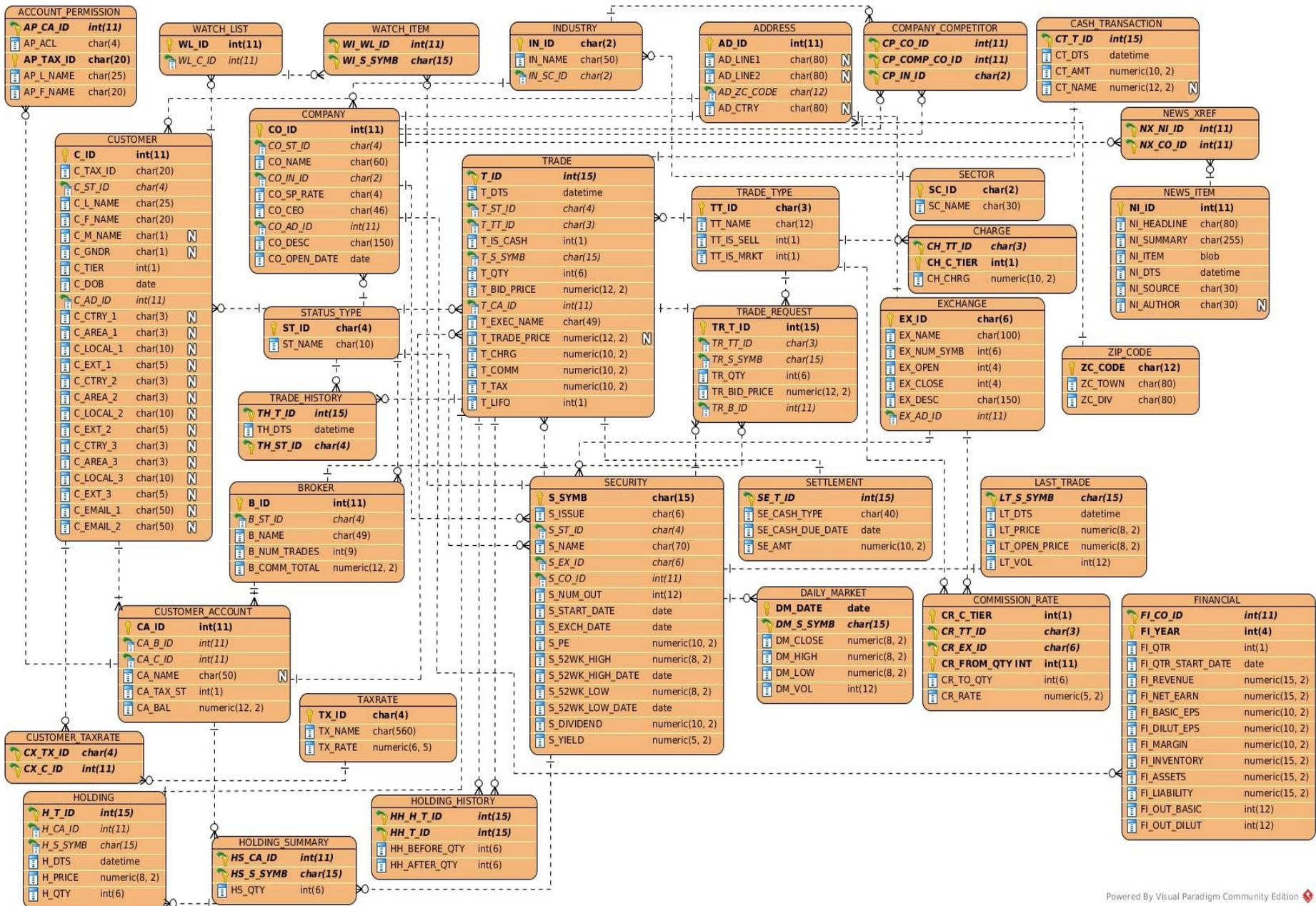


Figura 4.7. Esquema de la base de datos TPC-E

Egen permite configurar diferentes rangos de valores para la creación de la base de datos, entre ellos el número de clientes, el cual afecta al número de tuplas de todas las tablas. Se configuró Egen para producir mil clientes. En la Tabla 4.2 se muestra el nombre de cada tabla y el número de tuplas que contiene.

Tabla 4.2. Número de tuplas de cada tabla en la base de datos TPC-E

Tabla	Número de tuplas
ACCOUNT_PERMISSION	7128
CUSTOMER	1000
CUSTOMER_ACCOUNT	5000
CUSTOMER_TAXRATE	2000
HOLDING	284290
HOLDING_HISTORY	2247265
HOLDING_SUMMARY	49686
WATCH_ITEM	100602
WATCH_LIST	1000
BROKER	10
CASH_TRANSACTION	1590458
CHARGE	15
COMMISSION_RATE	240
SETTLEMENT	1728000
TRADE	1728000
TRADE_HISTORY	4148112
TRADE_REQUEST	0
TRADE_TYPE	5
COMPANY	500
COMPANY_COMPETITOR	1500
DAILY_MARKET	893925
EXCHANGE	4
FINANCIAL	10000
INDUSTRY	102
LAST_TRADE	685
NEWS_ITEM	1000
NEWS_XREF	1000
SECTOR	12
SECURITY	685
ADDRESS	1504
STATUS_TYPE	5
TAXRATE	320
ZIP_CODE	14741

4.3. FRAGMENT

La primer pantalla del sistema muestra el formulario para la captura de la información del administrador de bases de datos, la IP del servidor, el nombre de la base de datos y el archivo de carga. El registro de las operaciones hechas en la base de datos se encuentra en este archivo de carga con el formato propio de MySQL, el cual se debió configurar y habilitar anteriormente. En la Figura 4.8 se muestra la página “Conexión y archivo de carga”.



The image shows a web interface for a system named 'FRAGMENT'. At the top, the word 'FRAGMENT' is displayed in a large, stylized, black font. Below it, there is a rounded rectangular form titled 'CONEXIÓN'. The form contains the following elements:

- A text input field labeled 'Dirección'.
- A text input field labeled 'Nombre de usuario administrador'.
- A text input field labeled 'Clave de usuario administrador'.
- A section labeled 'Archivo general de registro de consultas (General log file)' with a button labeled '+ Elegir'.
- A button labeled 'Conectar' at the bottom center of the form.

Figura 4.8. Conexión y archivo de carga

Ya que el administrador haya ingresado los datos solicitados y presione el botón “Conectar”, la conexión se prueba, el estatus de conexión se muestra en la esquina superior derecha como se observa en la Figura 4.9. Si la conexión fue exitosa, se muestra un cuadro de diálogo en el cual se selecciona una de las tablas que están presentes en el esquema para aplicar el método.



Figura 4.9. Estatus de conexión y análisis del archivo de carga

Al presionar el botón “Fragmentar, asignar y replicar” se despliega un recuadro de espera, el cual demora unos minutos. En este lapso de tiempo se está aplicando el método y produciendo el esquema fragmentado. Se observa este recuadro en la Figura 4.10.



Figura 4.10. Cuadro de espera y aplicación del método

Al finalizar el tiempo de espera se despliega el esquema propuesto como resultado preliminar. Se presentan dos pestañas, MCRUD y Resultado. La primera pestaña muestra la matriz MCRUD, la cual relaciona cada sitio con el tipo de operaciones que se desempeñaron en ellos y el predicado

por el cual fueron hechas dichas operaciones. El ejemplo muestra las *C* como creaciones (*INSERT*), las *R* como lecturas (*SELECT*), las *U* como actualizaciones (*UPDATE*) y las *D* como eliminaciones (*DELETE*). Siguiendo el método elegido, cada operación tiene cierto valor, el cual está ligado a su costo. El valor para *C* es 2, el valor para *R* es 1, el valor para *U* es 3 y el valor para *D* es 2. Los valores entre predicados relacionados con el mismo atributo se suman, los valores dentro de cada ítem de la matriz se suman y, se restan al mayor valor por cada sitio en cada tupla los valores de los otros sitios con menor valor. Las operaciones mencionadas se realizan por cada atributo para determinar su valor ALP. En la Figura 4.11 se observa un ejemplo de la matriz MCRUD.

MCRUD			
Predicados	localhost	sitio3	sitio2
id<20	R		
type='Ind'	U		
id=3	U		
id=19		C	D
name='Dhk'		C	D
type='Cor'		C	
customer_id=5		C	
open_date='2008/01/01'		C	
balance=10000		C	

Figura 4.11. Matriz MCRUD

En la Figura 4.11, para determinar el valor ALP de la columna *id*, se resta del sitio de mayor valor los sitios de menor valor. En la primer tupla la operación es: $(1) - (0) - (0) = 1$, en la tercer tupla la operación es: $(3) - (0) - (0) = 3$, en la cuarta tupla la operación es: $(2) - (2) - (0) = 0$. El valor ALP de *id* es: $1 + 3 + 0 = 4$. Si en algún ítem hubiera más de una operación, por ejemplo *RRU*, se suman sus valores, es decir: $1+1+3=5$, siendo este el valor del ítem.

En la Figura 4.12 se muestra la segunda pestaña, la cual contiene la tabla ALP y la tabla de resultados. Si existe un máximo valor ALP para más de una columna, el administrador elige el

atributo que considere mejor para realizar el esquema. En el ejemplo mostrado en la Figura 4.12 solo existe un atributo con el mayor valor ALP. El atributo *type* es el más costoso y por medio de este se realiza el esquema mostrado en la tabla de resultados.

La tabla de resultados muestra de cada fragmento, a dónde se asignará y a dónde se replicará. Cuando existen fragmentos traslapados, la última columna muestra qué fragmentos son los que se aplicarán marcándolos con un icono verde. Los fragmentos que no serán aplicados no presentan este icono. Cuando los fragmentos encontrados no cubren la tabla por completo se presenta una última tupla nombrada “Sin fragmento”, la cual solo se asigna pero no se replica y representa toda tupla que no esté contenida en los otros fragmentos.

The screenshot shows a web interface with a header 'FRAGMENT' and two tabs: 'MCRUD' and 'Resultado'. The 'Resultado' tab is active and displays two tables.

The first table, titled 'TABLA ALP', lists attributes and their corresponding ALP values:

Atributo	Valor ALP	
id	4	
type	5	<input type="button" value="Seleccionar"/>
customer_id	2	
open_date	2	
balance	2	
name	0	

The second table, titled 'RESULTADO FINAL', shows the final assignment and replication of fragments:

Fragmento	Asignar	Replicar	
type="lnd"	localhost	sitio3	✓
type="Cor"	sitio3	localhost	✓

At the bottom of the interface, there is a button labeled 'Aplicar esquema'.

Figura 4.12. Tabla ALP y resultados

En la posterior presentación del caso de estudio en *Amazon Web Services* se muestra la siguiente pantalla con el resultado final y la representación de los fragmentos debido a que los sitios mostrados en este caso de estudio son simulados y no se aplica la asignación y la replicación sobre ellos.

4.4. Simulador de consultas

Se desarrolló una aplicación Web alterna para integrar la base de datos del *benchmark* TPC-E y el marco de trabajo CloudSim, la cual genera el número de operaciones y sitios indicados en la página principal, enfocados en la base de datos de TPC-E, la cual fue generada con Egen, una herramienta que proporcionan en la pagina oficial de TPC para facilitar el uso del *benchmark*. En la Figura 4.13 se muestra la pantalla principal de la aplicación.

The screenshot shows a web application interface with a light gray background. At the top, the text 'SIMULACIÓN CLOUDSIM - TPCE' is written in a large, bold, black sans-serif font. Below this, there is a white rounded rectangular box with a gray border and a title 'Parámetros' in a smaller font. Inside this box, there are two rows of input fields. The first row has the label 'Número de consultas por sitio' and a text input field containing the number '3'. The second row has the label 'Número de sitios' and a text input field containing the number '2'. Below these input fields, there are two buttons: 'Generar log' and 'Descargar log', both in a light gray color with a white border.

Figura 4.13. Pantalla principal de Simulacion CloudSim – TPC-E

Una vez generado el archivo de carga, el botón de descarga se habilita para la obtención de este archivo. El número de consultas debe ser mayor al numero de sitios, de lo contrario se mostrará un mensaje de error. El archivo que se descarga de este simulador contiene operaciones de lectura, escritura, actualización y eliminación, distribuidas entre sitios simulados. El archivo está en el formato de un *log* común en MySQL.

Para validar el método implementado se generó un archivo *log* con el simulador, configurado para tres sitios con cuatro operaciones cada uno, las cuales se observan en la Tabla 4.3. Este archivo se introdujo en la aplicación Web principal y se seleccionó para aplicar el método a la tabla “SETTLEMENT”. La matriz MCRUD se observa en la Figura 4.14.

Tabla 4.3. Operaciones obtenidas del simulador

Operación	Sitio
DELETE FROM FINANCIAL WHERE FI_OUT_BASIC>=3824575572	192.66.234.124
SELECT SE_CASH_DUE_DATE FROM SETTLEMENT WHERE SE_CASH_DUE_DATE<"2005-01-26"	192.66.234.124
DELETE FROM BROKER WHERE B_ID!=4300000006	192.66.234.124
INSERT INTO BROKER values(4.300000007E9, "ACTV", "Patrick G. Coder", 174202.0, 5131114.3)	192.66.234.124
SELECT WL_C_ID FROM WATCH_LIST WHERE WL_C_ID!=4300000747	220.57.249.11
SELECT SE_T_ID FROM SETTLEMENT WHERE SE_T_ID=200000000159239	220.57.249.11
DELETE FROM WATCH_LIST WHERE WL_C_ID>4300000281	220.57.249.11
INSERT INTO WATCH_LIST values(4.300000438E9, 4.300000141E9)	220.57.249.11
DELETE FROM SETTLEMENT WHERE SE_AMT>=16198.01	8.5.216.212
DELETE FROM COMPANY WHERE CO_CEO!="Amada Lerma"	8.5.216.212
UPDATE HOLDING SET H_DTS=H_DTS WHERE H_DTS<"2005-02-10"	8.5.216.212
INSERT INTO HOLDING values(2.00000001367687E14, 4.3000008454E10, "AP", "2005-02-03", 21.64, 200.0)	8.5.216.212

The screenshot shows the FRAGMENT application interface. At the top is the 'FRAGMENT' logo. Below it is a navigation bar with 'MCRUD' and 'Resultado' tabs. The main area is titled 'MCRUD' and contains a table with three columns representing IP addresses: 192.66.234.124, 8.5.216.212, and 220.57.249.11. On the left side, there is a 'Predicados' (Predicates) panel with three conditions: 'SE_CASH_DUE_DATE < 2005-01-26', 'SE_T_ID = 200000000159239', and 'SE_AMT >= 16198.01'. The table has two rows: the first row is labeled 'R' and the second row is labeled 'D'. At the bottom of the interface is a button labeled 'Aplicar esquema'.

Figura 4.14. MCRUD de la tabla SETTLEMENT

Se observa en la Figura 4.15 la tabla ALP, la cual determina que el atributo más costoso es “SE_AMT”, por el cual se realiza la tabla de resultados. Se observa que serán aplicados dos fragmentos, uno con el predicado que se encontró de este atributo y el otro con todas las tuplas que no se encuentren contenidas en ese predicado.

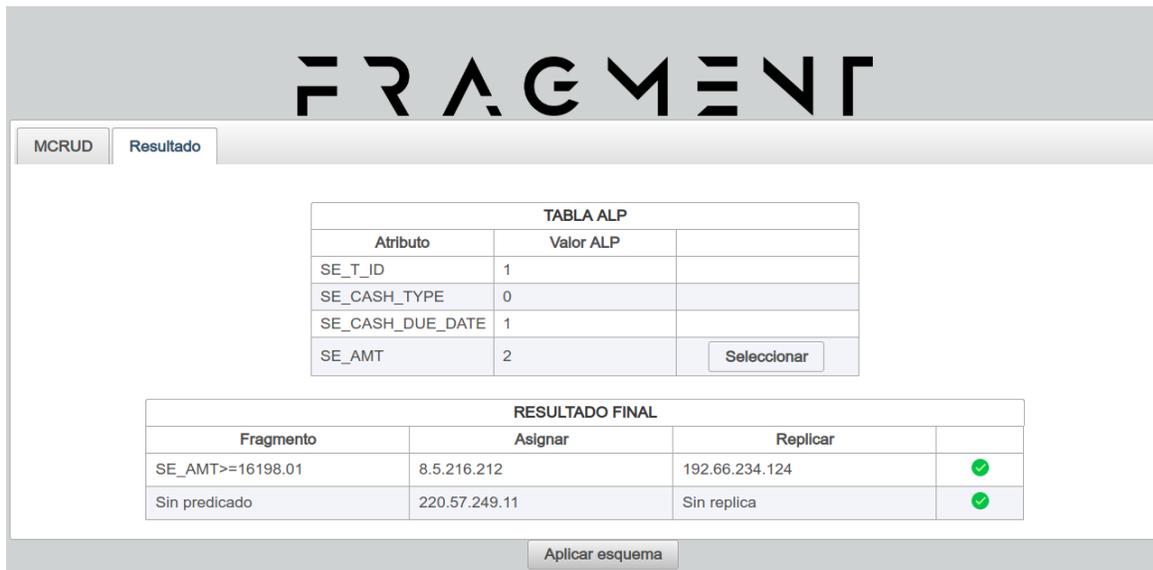


Figura 4.15. Tabla ALP y resultados obtenidos del simulador

4.5. Caso de estudio en *Amazon Web Services*

Para validar la aplicación del método y la implementación real de los fragmentos se utilizó una cuenta en *Amazon Web Services* y se configuraron tres instancias en EC2 de AMI (*Amazon Machine Image*, Imagen de Máquina de Amazon) tipo t2.micro. En una de las instancias se implementó la base de datos del *benchmark* TPC-E y se realizaron las consultas que se muestran en la Tabla 4.4 desde cada uno de los sitios sobre la tabla “CUSTOMER” de la base de datos TPC-E. Cada instancia posee Linux como sistema operativo, 1 GB en memoria RAM, procesador de la familia Intel Xeon con 1 CPU de 2.5 GHz.

Tabla 4.4. Operaciones realizadas en Amazon Web Services

Operación	Sitio	Duración en segundos
select * from CUSTOMER where C_F_NAME="Joshua"	localhost	.00174950
select * from CUSTOMER where C_EMAIL_1="JFowle@msn.com"	localhost	.00169875
update CUSTOMER set C_EMAIL_2="cmgelize@gmail.com" where C_F_NAME="Joshua" and C_EMAIL_1="JFowle@msn.com"	localhost	.00329800
select * from CUSTOMER where C_ID<4300000400	localhost	.00151550
insert into CUSTOMER values(123, "123", "ACTV", "CASTRO", "FELIPE", "M", "M", 2, "1993-11-20", 4300000505, "123", "123", "123", "123", "123", "123", "123", "123", "123", "123", "123", "123", " cmgelize@gmail.com ", "felipecastromedina123@gmail.com");	localhost	.00222075
select * from CUSTOMER where C_ID<4300000300	localhost	.00124100
select * from CUSTOMER where C_ID<4300000300	localhost	.00129675
select * from CUSTOMER where C_ID<4300000300	localhost	.00123400
select * from CUSTOMER where C_ID<4300000300	localhost	.00138400
DELETE from CUSTOMER where C_ID=123	localhost	.00220425
select * from CUSTOMER where C_F_NAME="Magan"	18.188.122.205	.00168775
select * from CUSTOMER where C_EMAIL_1="MLemarr@rr.com"	18.188.122.205	.00169250
update CUSTOMER set C_EMAIL_2="correoinventado@gmail.com" where C_F_NAME="Magan" and C_EMAIL_1=" MLemarr@rr.com "	18.188.122.205	.00326500
select * from CUSTOMER where C_ID>4300000600	18.188.122.205	.00166400

Tabla 4.4. Operaciones realizadas en Amazon Web Services (continuación)

Operación	Sitio	Duración en segundos
UPDATE CUSTOMER set C_EMAIL_1="cmgelize@gmail.com" where C_ID=4300000550	18.223.114.241	.00218200
select * from CUSTOMER where C_ID=4300000450	18.223.114.241	.00029300
UPDATE CUSTOMER set C_EMAIL_1="cmgelize@gmail.com" where C_ID=4300000450	18.223.114.241	.00227075

En la Figura 4.16 se observa la MCRUD de las operaciones anteriores, mostrando cada tipo de operación realizada por cada sitio en cada predicado. En la Figura 4.17 se muestra la tabla ALP, en la cual se presenta el atributo más costoso de la tabla “CUSTOMER”, el cual es “C_ID” con un valor de 38. En la Figura 4.18 se observa el esquema propuesto de asignación y replicación con base en el atributo más costoso.

FRAGMENT

MCRUD
Resultado

MCRUD			
Predicados	localhost	18.188.122.205	18.223.114.241
C_F_NAME='Joshua'	RU		
C_EMAIL_1='JFowle@msr	RU		
C_EMAIL_2='cmgelipe@gr	U		
C_ID<4300000400	R		
C_ID=123	CD		
C_TAX_ID='123'	C	C	
C_ST_ID='ACTV'	C	C	
C_L_NAME='CASTRO'	C	C	
C_F_NAME='FELIPE'	C	C	
C_M_NAME='M'	C	C	
C_GNDR='M'	C	C	
C_TIER=2	C	C	
C_DOB='1993-11-20'	C	C	
C_AD_ID=4300000505	C	C	
C_CTRY_1='123'	C	C	
C_AREA_1='123'	C	C	
C_LOCAL_1='123'	C	C	
C_EXT_1='123'	C	C	
C_CTRY_2='123'	C	C	
C_AREA_2='123'	C	C	
C_LOCAL_2='123'	C	C	
C_EXT_2='123'	C	C	
C_CTRY_3='123'	C	C	
C_AREA_3='123'	C	C	
C_LOCAL_3='123'	C	C	
C_EXT_3='123'	C	C	
C_EMAIL_1='cmgelipe@gr	C	C	UUUUU
C_EMAIL_2='felipecastron	C	C	
C_ID<4300000300	RRRR		
C_F_NAME='Magan'		RU	
C_EMAIL_1='MLemarr@rr.		RU	
C_EMAIL_2='correoinventa		U	
C_ID>4300000600		RRRRR	
C_ID=4300001001		CD	
C_ID=4300000400			R
C_ID=4300000401			U
C_ID=4300000500			RU
C_ID=4300000600			RU
C_ID=4300000550			RU
C_ID=4300000450			RU

Aplicar esquema

Figura 4.16. MCRUD de operaciones sobre “CUSTOMER” en Amazon Web Services

TABLA ALP	
Atributo	Valor ALP
C_ID	38
C_TAX_ID	0
C_ST_ID	0
C_L_NAME	0
C_F_NAME	8
C_M_NAME	0
C_GNDR	0
C_TIER	0
C_DOB	0
C_AD_ID	0
C_CTRY_1	0

Figura 4.17. Tabla ALP de operaciones sobre “CUSTOMER” en *Amazon Web Services*

Para explicar la obtención del valor ALP del atributo “C_ID” se presenta la Tabla 4.5, la cual contiene todas las operaciones relacionadas con este y la operación que se desempeña en cada tupla. Sumando cada subtotal de las operaciones se obtiene el valor ALP del atributo.

Tabla 4.5. MCRUD de operaciones del atributo “C_ID”

Predicado	Localhost	18.188.122.205	18.223.114.241	Operacion
C_ID<4300000400	R			1-0-0=1
C_ID=123	CD			(2+2)-0-0=4
C_ID<4300000300	RRRR			(1+1+1+1)-0-0=4
C_ID>4300000600		RRRRR		(1+1+1+1+1)-0-0=5
C_ID=4300001001		CD		(2+2)-0-0=4
C_ID=4300000400			R	1-0-0=1
C_ID=4300000401			U	3-0-0=3
C_ID=4300000500			RU	(1+3)-0-0=4
C_ID=4300000600			RU	(1+3)-0-0=4
C_ID=4300000550			RU	(1+3)-0-0=4
C_ID=4300000450			RU	(1+3)-0-0=4

RESULTADO FINAL			
Fragmento	Asignar	Replicar	
C_ID<4300000400	localhost	18.188.122.205	✓
C_ID=123	localhost	18.188.122.205	
C_ID<4300000300	localhost	18.188.122.205	
C_ID>4300000600	18.188.122.205	localhost	✓
C_ID=4300001001	18.188.122.205	localhost	
C_ID=4300000400	18.223.114.241	localhost	✓
C_ID=4300000401	18.223.114.241	localhost	✓
C_ID=4300000500	18.223.114.241	localhost	✓
C_ID=4300000600	18.223.114.241	localhost	✓
C_ID=4300000550	18.223.114.241	localhost	✓
C_ID=4300000450	18.223.114.241	localhost	✓
Sin predicado	18.188.122.205	Sin replica	✓

Figura 4.18. Esquema propuesto de asignación y replicación

Algunos fragmentos en la Figura 4.18 se asignan o replican en *localhost*, sin embargo, esta no es una dirección pública. Esto sucede debido a que este sitio es el sitio principal, y algunas operaciones que se realizaron se hicieron de este sitio principal a sí mismo, esto no representa algún problema, ya que la aplicación sustituye automáticamente esta dirección por la pública del sitio principal, la cual se ingresó en la primer pantalla. El fragmento sin predicado se observa en la última tupla de la tabla, debido a que los fragmentos anteriores no cubren toda la tabla original. Este fragmento no se replica, ya que nadie utilizó esta parte de la tabla.

También se observa en la Figura 4.18 que los fragmentos de las tuplas 2, 3 y 5 no se aplicarán, ya que son predicados traslapados. Los predicados de las tuplas 2 y 3 se traslapan con el predicado de la primer tupla, debido a que 123 y 4300000300 están contenidos en C_ID<4300000400, de igual forma el predicado de la tupla 5 se traslapa con el predicado de la cuarta tupla.

Al aplicar el esquema se despliega un bosquejo de los fragmentos asignados, el cual contiene las primeras 10 tuplas y las primeras 5 columnas de cada fragmento. Este bosquejo se observa en la Figura 4.19 mostrando que se asignaron los 9 fragmentos propuestos por el esquema.

FRAGMENT

[Regresar](#)

TABLAS REPRESENTATIVAS DE CADA FRAGMENTO

C_ID	C_TAX_ID	C_ST_ID	C_L_NAME	C_F_NAME
4.30000001E9	078GO5457DB627	ACTV	Fowle	Joshua
4.30000002E9	803MO6290MK444	ACTV	Swigert	Willie
4.30000003E9	329GW9315LX866	ACTV	Labree	Amos
4.30000004E9	047YK9898LQ614	ACTV	Ciciora	Charlotte
4.30000005E9	805QP9293HR611	ACTV	Criscuolo	Franklin
4.30000006E9	079CV0265WT109	ACTV	Rohrscheib	Constance
4.30000007E9	414RB8918PT543	ACTV	Burtchell	Jason
4.30000008E9	857CV4901DE721	ACTV	Kules	Thomas
4.30000009E9	834EZ4964IF367	ACTV	Moe	Richard
4.3000001E9	090EI5140VB829	ACTV	Proud	Lydia

C_ID	C_TAX_ID	C_ST_ID	C_L_NAME	C_F_NAME
4.300000601E9	436DF3716VW130	ACTV	Green	David
4.300000602E9	793PF5333BZ106	ACTV	Wieczorek	Loren
4.300000603E9	783VO0476HL120	ACTV	Suitor	Doris
4.300000604E9	805EM5137JX300	ACTV	Noblin	Marjorie
4.300000605E9	578ZI9698RE699	ACTV	Roberts	Burt
4.300000606E9	127NI5818KS465	ACTV	Nisbet	Renee
4.300000607E9	871AE0115SZ064	ACTV	Niglio	Aaron
4.300000608E9	576ZT4468SX599	ACTV	Montesino	William
4.300000609E9	010BE0002MV918	ACTV	Allum	Laurel
4.30000061E9	577EZ8060VT926	ACTV	Blecha	Tania

C_ID	C_TAX_ID	C_ST_ID	C_L_NAME	C_F_NAME
4.3000004E9	923QR4392RV998	ACTV	Mouw	Patricia

C_ID	C_TAX_ID	C_ST_ID	C_L_NAME	C_F_NAME
4.300000401E9	056ET7428IF606	ACTV	Ottinger	Velia

C_ID	C_TAX_ID	C_ST_ID	C_L_NAME	C_F_NAME
4.3000005E9	909JX4024CH269	ACTV	Tsuzuki	Brett

C_ID	C_TAX_ID	C_ST_ID	C_L_NAME	C_F_NAME
4.3000006E9	264YA9429IU711	ACTV	Kalafut	Henry

C_ID	C_TAX_ID	C_ST_ID	C_L_NAME	C_F_NAME
4.30000055E9	798XA9711IE765	ACTV	Stefanick	Jane

C_ID	C_TAX_ID	C_ST_ID	C_L_NAME	C_F_NAME
4.30000045E9	627GS1214BH773	ACTV	Harmond	Jennifer

C_ID	C_TAX_ID	C_ST_ID	C_L_NAME	C_F_NAME
4.300000402E9	573YD4254VA102	ACTV	Plemons	Cori
4.300000403E9	031ZC0226DB034	ACTV	Dumoulin	Carl
4.300000404E9	379QC1250AJ899	ACTV	Bresette	Sean
4.300000405E9	810MM2091RK196	ACTV	Atwill	Scott
4.300000406E9	163EI5324YW508	ACTV	Rose	Lester
4.300000407E9	697HJ3005XS017	ACTV	Gallati	Yvonne
4.300000408E9	051RK2625FT457	ACTV	Frankovich	John
4.300000409E9	420NF4949LV452	ACTV	Ciesco	Michelle
4.30000041E9	115FQ6277WH610	ACTV	Schedler	Gerard

Figura 4.19. Bosquejo de los fragmentos asignados

Una vez asignados y replicados los fragmentos se realizó una comparación de tiempos de ejecución de las mismas consultas con el esquema fragmentado, esta se observa en la Tabla 4.6, la cual muestra las consultas más significativas.

Tabla 4.6. Comparación de tiempo de ejecución de consultas

Operación	Tiempo sin fragmentación	Tiempo con fragmentación
select * from CUSTOMER where C_ID<4300000400	.00151550	.00031025
select * from CUSTOMER where C_ID>4300000600	.00166400	.00027750
select * from CUSTOMER where C_ID=4300000500	.00029975	.00024425

Capítulo 5. Conclusiones y Recomendaciones

5.1 Conclusiones

La fragmentación y replicación de datos en la nube es un tema de gran interés en la industria y la academia. Se observan en la actualidad una gran cantidad de técnicas para fragmentar y replicar datos, así como las aplicaciones de estas técnicas en ciertos escenarios en los cuales se mejora el desempeño y la disponibilidad e incluso la seguridad de los datos.

Si bien la fragmentación y replicación son temas muy abordados, no se analiza el panorama en su totalidad, ya que sólo se analizan algunas técnicas de fragmentación y replicación. La gran demanda de información de algunos escenarios exige técnicas adecuadas y este trabajo es idóneo para elegir la mejor opción.

Con este trabajo se beneficiarán en gran medida los investigadores que aborden la necesidad de utilizar algún método de fragmentación y replicación, ya que observarán las ventajas y desventajas de cada uno de ellos.

Después del análisis de trabajos relacionados, se implementó el método de fragmentación, asignación y replicación que propone Abdel et al. [82] por medio de una aplicación Web que realiza todas las operaciones requeridas para asignar y replicar fragmentos sobre un conjunto de sitios mediante el esquema que produce el método. Se creó el simulador de operaciones con CloudSim para validar la implementación sobre el *benchmark* TPC-E y por último se utilizó un ambiente real sobre *Amazon Web Services*. La implementación fue exitosa y el desempeño de la base de datos del *benchmark* mejoró, ya que los tiempos de respuesta de las mismas operaciones fueron menores.

Considerar un gran número de aspectos y elegir un método más complejo es una opción, sin embargo, el enfoque del método seleccionado intenta reducir su complejidad para su fácil implementación, conservando los aspectos que se consideran más relevantes en este trabajo.

5.2 Recomendaciones

Al aplicar el método se comprendió el mecanismo que emplean los autores y se observaron ventajas y desventajas. De esta manera, como continuación de esta investigación, sería interesante incluir un factor de mayor importancia en la solución de los predicados traslapados, es decir, que en la incidencia de predicados traslapados no solo se descarten los fragmentos más pequeños y se tomen en cuenta los grandes que los contienen, sino que se realice un análisis de costos sobre ese traslape, para que si los fragmentos mas pequeños tuviesen un mayor costo que los fragmentos grandes, estos se tomen en cuenta en el esquema final.

Además, como trabajo a futuro se sugiere que se tome en cuenta la naturaleza de cada operación, ya que operaciones de eliminación y creación afectan al número de tuplas resultantes de un conjunto de datos origen y no contemplar su naturaleza en la creación del esquema de fragmentación conduciría a la creación de fragmentos con o sin tuplas que no se desean.

La fragmentación estática supone un problema de mantenimiento, ya que el DBA debe saber cuando realizar nuevamente otra fragmentación y esto es una tarea difícil o imposible de realizar adecuadamente. Como trabajo a futuro se propone agregar un modelo de fragmentación dinámica para que esta se base en umbrales de costo para realizar fragmentaciones posteriores a la inicial.

Productos académicos



Felipe Castro Medina, Lisbeth Rodríguez Mazahua, María Antonieta Abud Figueroa, Celia Romero Torres, Luis Ángel Reyes Hernández, Giner Alor Hernández.

Application of data fragmentation and replication methods in the cloud: a review.

2019 International Conference on Electronics, Communications and Computers (ISBN: 978-1-7281-1145-2).

Páginas 47-54

Estado: Publicado.



Felipe Castro Medina, Lisbeth Rodríguez Mazahua, Asdrúbal López Chau, Isaac Machorro Cano, María A. Abud Figueroa.

Design of a horizontal data fragmentation, allocation and replication method in the cloud.

15th IEEE International Conference on Automation Science and Engineering (CASE) 2019.

Estado: Aceptado.



Felipe Castro Medina, Lisbeth Rodríguez Mazahua, Asdrúbal López Chau, María A. Abud Figueroa, Giner Alor Hernández.

Fragment: A Web Application For Database Fragmentation, Allocation And Replication Over A Cloud Environment.

Estado: Enviado

Referencias

- [1] A. Silberschatz, H. F. Korth and S. Sudarshan, "Distributed Databases" in *Database Systems Concepts*, 6th ed. New Haven, CT: Yale University, 2011, pp. 825-847.
- [2] M. T. Özsu and P. Valduriez, "Distributed Databases Design" in *Principles of Distributed Database Systems*, 3th ed. Ontario, CA: University of Waterloo, 2011, pp. 81-113.
- [3] R. J. Thara, S. Shine and C. Sanu, "Optimizing the performance of Database as a Service (DaaS) model — A distributed approach," *Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)*, Tiruchengode, 2013, pp. 1-5.
- [4] R. P. Rafael, "Software como servicio", Fac. de ingeniería en informática y sistemas, Univ. Rafael Landívar, Guatemala, GT. Tech. Rep. Guatemala, GT. Tech. Rep. FI-231-09, 2009.
- [5] "¿Qué es PaaS?". [online]. Available: <https://azure.microsoft.com/es-es/overview/what-is-paas/>. [Accessed: 20/febrero/2018].
- [6] "¿Qué es IaaS?", [online]. Available: <https://azure.microsoft.com/es-es/overview/what-is-iaas/>. [Accessed: 20/febrero/2018].
- [7] M. Ali, K. Bilal, S. Khan, B. Veeravalli, K. Li and A. Zomaya, "DROPS: Division and Replication of Data in the Cloud for Optimal Performance and Security" in *IEEE Transactions on Cloud Computing*, vol. pp, no. 99, pp. 1-14, 2015.
- [8] L. Wiese, "Taxonomy-Based Fragmentation for Anti-instantiation in Distributed Databases", *IEEE/ACM 6th International Conference on Utility and Cloud Computing*, Dresden, 2013, pp. 363-368.
- [9] B. Sauer and W. Hao, "Horizontal cloud database partitioning with data mining techniques" *12th Annual IEEE Consumer Communications and Networking Conference (CCNC)*, Las Vegas, NV, 2015, pp. 796-801.

- [10] W. D. Moral and B. M. Kumar, "Improve the data retrieval time and security through fragmentation and replication in the Cloud", *International Conference on Advanced Communication Control and Computing Technologies (ICACCCT)*, Ramanathapuram, 2016, pp. 539-545.
- [11] V. K. Kaushik, H. K. Sharma, y D. Gopalani, "Load Balancing In Cloud Computing Using High Level Fragmentation Of Dataset", *International Conference on Cloud, Big Data and Trust*, 2013, pp. 118–126.
- [12] A. Hudic, S. Islam, P. Kieseberg, y E. R. Weippl, "Data confidentiality using fragmentation in cloud computing", *Int. J. Communication Networks and Distributed Systems*, vol. 1, no. 3/4, 2012, pp. 325–329.
- [13] J. Kohler, K. Simov, A. Fiech, y T. Specht, "On the Performance of Query Rewriting in Vertically Distributed Cloud Databases", en *Innovative Approaches and Solutions in Advanced Intelligent Systems*, vol. 648, 2016, pp. 59-73.
- [14] J. Kohler y T. Specht, "A performance comparison between parallel and lazy fetching in vertically distributed cloud databases", *International Conference on Cloud Technologies and Applications (CloudTech)*, 2015, pp. 1-7.
- [15] A. Stierner, I. Fetai and H. Schuldt, "Analyzing the performance of data replication and data partitioning in the cloud: The BEOWULF approach", *IEEE International Conference on Big Data (Big Data)*, Washington, DC, 2016, pp. 2837-2846.
- [16] A. E. Abdel, N. L. Badr, and M. F. Tolba, "Dynamic Distributed Database over Cloud Environment", *AMLTA 2014: Advanced Machine Learning Technologies and Applications*, Vol. 488, pp. 67-76, 2014.
- [17] L. Wiese, «Clustering-based fragmentation and data replication for flexible query answering in distributed databases», *Journal of Cloud Computing*, 2014, pp. 1-18.

- [18] J. W. Lin, C. H. Chen and J. M. Chang, "QoS-Aware Data Replication for Data-Intensive Applications in Cloud Computing System", in *IEEE Transactions on Cloud Computing*, vol. 1, no. 1, pp. 101-115, 2013.
- [19] D.-W. Sun, G.-R. Chang, S. Gao, L.-Z. Jin, y X.-W. Wang, «Modeling a Dynamic Data Replication Strategy to Increase System Availability in Cloud Computing Environments», *Journal of Computer Science and Technology*, vol. 27, no. 2, pp. 256-272, 2012.
- [20] D. Boru, D. Kliazovich, F. Granelli, P. Bouvry, y A. Y. Zomaya, «Energy-efficient data replication in cloud computing datacenters», *Cluster Computing*, 2015, pp. 385-402.
- [21] K. Kapusta and G. Memmi, "Data protection by means of fragmentation in distributed storage systems", *International Conference on Protocol Engineering (ICPE) and International Conference on New Technologies of Distributed Systems (NTDS)*, Paris, 2015, pp. 1-8.
- [22] A. Ali, A. A del, M. A. Tacha, "An optimized approach for simultaneous horizontal data fragmentation and allocation in Distributed Database Systems (DDBSs)", en *Heliyon*, vol. 3 no. 12, pp. 1-42, 2017.
- [23] "What is Cloud Computing". [online]. Available: <https://aws.amazon.com>. [Accessed: 17/abril/2019].
- [24] Oracle Corporation (2018, Oct 13). *JavaServer Faces Technology* [Online]. Available: <https://www.oracle.com/technetwork/java/javae/javaserverfaces-139869.html>
- [25] "NetBeans IDE - The Smarter and Faster Way to Code". [online]. Available: <https://netbeans.org/features/index.html>. [Accessed: 16/marzo/2018].
- [26] C. Nieves, J. Ucán y V. Menéndez, "UWE en Sistema de Recomendación de Objetos de Aprendizaje. Aplicando Ingeniería Web: Un Método en Caso de Estudio", *Revista Latinoamericana de Ingeniería de Software*, 2015, vol. 2, pp. 137-143.
- [27] Oracle Corporation (2018, Oct 13). *MySQL* [Online]. Available: <https://www.mysql.com>
- [28] "TPC - Benchmarks". [online]. Available: <http://www.tpc.org/> [Accessed: 17/abril/2019].

- [29] R. Calheiros, R. Ranjan, A. Beloglazov, C. De Rose and R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms", *Software: Practice and Experience*, vol. 41, no. 1, pp. 23-50, 2010.
- [30] A. E. Abdel, N. L. Nadr and M. F. Tolba, "*An Enhanced CRUD For Vertical Fragmentation Allocation And Replication Over The Cloud Environment*", *INFOS '16: Proceedings of the 10th International Conference on Informatics and Systems*, Giza, EG, 2016, pp. 146-152.
- [31] A. Thomson, T. Diamond, S. Weng, K. Ren P. Shao and D. J. Abadi, "*Fast Distributed Transactions and Strongly Consistent Replication for OLTP Database Systems*", *ACM Transactions on Database Systems*, Vol. 39, No. 2, 2014, pp. 11-50.
- [32] C. López, R. Heinsen and E. Huh, "Improving Availability Applying Intelligent Replication in Federated Cloud Storage Based on Log Analysis", *International Conference on Machine Learning and Soft Computing 2017*, Ho Chi Minh City, VN, 2017, pp. 148-153.
- [33] R. Marcus, O. Papaemmanouil, S. Semenova and S. Garber, "NashDB: An End-to-End Economic Method for Elastic Database Fragmentation, Replication, and Provisioning", *Proceedings of the 2018 International Conference on Management of Data*, Houston, TX, USA, 2018, pp. 1253-1267.
- [34] M. B. Gudadhe and A. J. Agrawal, "Performance Analysis Survey of Data Replication Strategies in Cloud Environment", *Proceedings of the 2017 International Conference on Big Data Research*, Osaka, JP, 2017, pp. 38-43.
- [35] R. Dhamane, M. Patiño, V. Vianello and R. Jiménez, "Performance Evaluation of Database Replication Systems", *Proceedings of the 18th International Database Engineering & Applications Symposium*, Porto, PT, 2014, pp. 288-293.
- [36] T. Shwe and M. Aritsugi, "Proactive Re-replication Strategy in HDFS based Cloud Data Center", *Proceedings of the 10th International Conference on Utility and Cloud Computing*, Austin, TX, USA, 2017, pp. 121-130.

- [37] F. Bollwein and L. Wiese, "Separation of Duties for Multiple Relations in Cloud Databases as an Optimization Problem", Proceedings of the 21st International Database Engineering & Applications Symposium, Bristol, UK, 2017, pp. 98-107.
- [38] L. Xiao, K. Ren, Q. Zheng and G. A. Gibson, "ShardFS vs. IndexFS: Replication vs. Caching Strategies for Distributed Metadata Management in Cloud Storage Systems", Proceedings of the Sixth ACM Symposium on Cloud Computing, Kohala Coast, HI, 2015, pp. 236-249.
- [39] M. Abdul, M. Sharfuddin and T. Rangunathan, "Support-based replication algorithm for cloud storage systems", Proceedings of the 7th ACM India Computing Conference, Nagpur, IN, 2014, pp. 1-9.
- [40] K. A. Nuaimi, N. Mohamed, M. A. Nuaimi and J. A. Jaroodi, "A Novel Approach for Dual-Direction Load Balancing and Storage Optimization in Cloud Services," 2014 IEEE 13th International Symposium on Network Computing and Applications, Cambridge, MA, 2014, pp. 105-112.
- [41] N. Tziritas et al., "Data Replication and Virtual Machine Migrations to Mitigate Network Overhead in Edge Computing Systems," in IEEE Transactions on Sustainable Computing, vol. 2, no. 4, pp. 320-332, 1 Oct.-Dec. 2017.
- [42] S. D. Salunkhe and D. Patil, "Division and replication for data with public auditing scheme for cloud storage," 2016 International Conference on Computing Communication Control and automation (ICCUBE), Pune, 2016, pp. 1-5.
- [43] K. Al Nuaimi, N. Mohamed, M. Al Nuaimi and J. Al-Jaroodi, "Dual direction load balancing and partial replication storage of cloud DaaS," 2014 IEEE 3rd International Conference on Cloud Networking (CloudNet), Luxembourg, 2014, pp. 432-437.
- [44] Ching-Yang Tseng, ChangChun Lu and Cheng-Fu Chou, "Efficient privacy-preserving multi-keyword ranked search utilizing document replication and partition," 2015 12th Annual

IEEE Consumer Communications and Networking Conference (CCNC), Las Vegas, NV, 2015, pp. 671-676.

[45] B. Zhu, H. Li, K. W. Shum and S. R. Li, "HFR code: a flexible replication scheme for cloud storage systems," in *IET Communications*, vol. 9, no. 17, pp. 2095-2100, 26 11 2015.

[46] A. Mseddi, M. A. Salahuddin, M. F. Zhani, H. Elbiaze and R. H. Glitho, "On optimizing replica migration in distributed cloud storage systems," 2015 IEEE 4th International Conference on Cloud Networking (CloudNet), Niagara Falls, ON, 2015, pp.

[47] Dezhi Hou, Shidong Zhang and Lanju Kong, "Placement of SaaS cloud data and dynamically access scheduling strategy," 2013 8th International Conference on Computer Science & Education, Colombo, 2013, pp. 834-838.

[48] J. Li and M. Hou, "PMSTCOM: A Novel Replication Consistency Maintenance Strategy in Cloud Storage System," 2017 3rd International Conference on Big Data Computing and Communications (BIGCOM), Chengdu, 2017, pp. 277-283.

[49] A. Tripathi and G. Rajappan, "Scalable Transaction Management for Partially Replicated Data in Cloud Computing Environments," 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), San Francisco, CA, 2016, pp. 260-267.

[50] Yun Tian, Xiao Qin and Yafei Jia, "Secure replica allocation in cloud storage systems with heterogeneous vulnerabilities," 2015 IEEE International Conference on Networking, Architecture and Storage (NAS), Boston, MA, 2015, pp. 205-214.

[51] X. Luo, Y. Wang, Z. Zhang and H. Wang, "Superset: A Non-uniform Replica Placement Strategy towards High-Performance and Cost-Effective Distributed Storage Service," 2013 International Conference on Advanced Cloud and Big Data, Nanjing, 2013, pp. 139-146.

[52] F. Xie, J. Yan and J. Shen, "Towards Cost Reduction in Cloud-Based Workflow Management through Data Replication," 2017 Fifth International Conference on Advanced Cloud and Big Data (CBD), Shanghai, 2017, pp. 94-99.

- [53] C. Hsu, V. W. Freeh and F. Villanustre, "Trilogy: Data placement to improve performance and robustness of cloud computing," 2017 IEEE International Conference on Big Data (Big Data), Boston, MA, 2017, pp. 2442-2451.
- [54] B. Alami, N. Jafari, "A comprehensive review of the data replication techniques in the cloud environments: Major trends and future directions", *Journal of Network and Computer Applications*, Vol. 64, pp. 229-238, 2016.
- [55] T. Hamrouni, S. Slimani and F. Ben, "A data mining correlated patterns-based periodic decentralized replication strategy for data grids", *Journal of Systems and Software*, Vol. 110, pp. 10-27, 2015.
- [56] N. Kaur and S. Singh, "A dynamic, cost-aware, optimized data replication strategy for heterogeneous cloud data centers", *Future Generation Computer Systems*, Vol. 65, pp. 10-32, 2016.
- [57] N. Mansouri and M. M. Javidi, "A new Prefetching-aware Data Replication to decrease access latency in cloud environment", *Journal of Systems and Software*, Vol. 149, pp. 197-215, 2018.
- [58] A. Brigit, "Data allocation optimization for query processing in graph databases using Lucene", *Computers & Electrical Engineering*, Vol. 70, pp. 1019-1033, 2018.
- [59] G. Montoya, H. Skaf-Molli, P. Molli and M. Vidal, "Decomposing federated queries in presence of replicated fragments", *Web Semantics: Science, Services and Agents on the World Wide Web*, Vol. 42, pp. 1-18, 2017.
- [60] N. Mansouri and M. Kuchaki, M.M. Javidi, "DPRS: A dynamic popularity aware replication strategy with parallel download scheme in cloud environments", *Simulation Modelling Practice and Theory*, Vol. 77, pp. 177-196, 2017.
- [61] C. Rosas, A. Sikora, J. Jorba, A. Moreno, A. Espinosa and E. César, "Dynamic tuning of the workload partition factor and the resource utilization in data-intensive applications", *Future Generation Computer Systems*, Vol. 37, pp. 162-177, 2014.

- [62] H. Fan, Z. Ma, D. Wang and J. Liu, "Handling distributed XML queries over large XML data based on MapReduce framework", *Information Sciences*, Vol. 453, pp. 1-20, 2018.
- [63] L. Gkatzikis, V. Sourlas, C. Fischione and I. Koutsopoulos, "Low complexity content replication through clustering in Content-Delivery Networks", *Computer Networks*, Vol. 121, pp. 137-151, 2017.
- [64] S. Long, Y. Zhao and W. Chen, "MORM: A Multi-objective Optimized Replication Management strategy for cloud storage cluster", *Journal of Systems Architecture*, Vol. 60, pp. 234-244, 2014.
- [65] S. Sengupta and K.M. Annervaz, "Multi-site data distribution for disaster recovery—A planning framework", *Future Generation Computer Systems*, Vol. 41, pp. 53-64, 2014.
- [66] Z. Zeng and B. Veeravalli, "Optimal metadata replications and request balancing strategy on cloud data centers", *Journal of Parallel and Distributed Computing*, Vol. 74, pp. 2934-2940, 2014.
- [67] T.P. Shabeera, S.D. Madhu, S. M. Salam and K. Murali, "Optimizing VM allocation and data placement for data-intensive applications in cloud using ACO metaheuristic algorithm", *Engineering Science and Technology, an International Journal*, Vol. 20, pp. 616-628, 2017.
- [68] T. Habib and Z. Ansari, "Partition based clustering of large datasets using MapReduce framework: An analysis of recent themes and directions", *Future Computing and Informatics Journal*, Vol. 20, pp. 1-15, 2018.
- [69] D. Sánchez and M. Batet, "Privacy-preserving data outsourcing in the cloud via semantic data splitting", *Computer Communications*, Vol. 110, pp. 187-201, 2017.
- [70] T. Chen, R. Bahsoon and A. H. Tawil, "Scalable service-oriented replication with flexible consistency guarantee in the cloud", *Information Sciences*, Vol. 264, pp. 349-370, 2014.

- [71] Q. Xia, W. Liang and Z. Xu, “The operational cost minimization in distributed clouds via community-aware user data placements of social networks”, *Computer Networks*, Vol. 112, pp. 263-278, 2017.
- [72] O. Romero, V. Herrero, A. Abelló and J. Ferrarons, “Tuning small analytics on Big Data: Data partitioning and secondary indexes in the Hadoop ecosystem”, *Information Systems*, Vol. 54, pp. 336-356, 2015.
- [73] S. Benkrid, L. Bellatreche, A. Cuzzocrea, “A Global Paradigm for Designing Parallel Relational Data Warehouses in Distributed Environments”, *Transactions on Large-Scale Data- and Knowledge-Centered Systems XV*, Vol. 8920, pp. 64-101, 2014.
- [74] N. Mansouri and M. M. Javidi, “A hybrid data replication strategy with fuzzy-based deletion for heterogeneous cloud data centers”, *The Journal of Supercomputing* (2018), Vol. 74, pp. 1-24, 2018.
- [75] A. Guabtini, R. Ranjan, F. A. Rabhi, “A workload-driven approach to database query processing in the cloud”, *The Journal of Supercomputing*, Vol. 63, pp. 722–736, 2013.
- [76] A. S. Darabant, L. Tambulea, V. Varga, “Access Patterns Optimization in Distributed Databases Using Data Reallocation”, *Database and Expert Systems Applications*, Vol. 10438, pp. 178-186, 2017.
- [77] N. Mansouri, “Adaptive data replication strategy in cloud computing for performance improvement”, *Frontiers of Computer Science*, Vol. 10, pp. 925–935, 2016.
- [78] E. B. Edwin, P. Umamaheswari, M. R. Thanka, “An efficient and improved multi-objective optimized replication management with dynamic and cost aware strategies in cloud computing data center”, *Cluster Computing*, Vol. 21, pp. 1-10, 2017.
- [79] A. E. Abdel, N. L. Badr, M. F. Tolba, “An Enhanced Distributed Database Design Over the Cloud Environment”, *Proceedings of the International Conference on Advanced Intelligent Systems and Informatics 2016*, Vol. 533, pp. 290-298, 2016.

- [80] K. Grolinger, W. A. Higashino, A. Tiwari and M. Capretz, “Data management in cloud environments: NoSQL and NewSQL data stores”, *Journal of Cloud Computing*, Vol. 2, pp. 1-24, 2013.
- [81] J. Mohammad and M. Murshed, “Distributed Database Management Systems: Architectural Design Choices for the Cloud”, *Cloud Computing*, Vol. 7, pp. 23-50, 2014.
- [82] A. E. Abdel, N. L. Badr and M. F. Tolba, “Distributed Database System (DSS) Design Over a Cloud Environment”, *Multimedia Forensics and Security*, Vol. 115, pp. 97-116, 2016.
- [83] A. V. Kish, J. R. Rose, and C. Farkas, “Efficient Partitioning and Allocation of Data for Workload Queries”, *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*, Vol. 313, pp. 549-555, 2014.
- [84] X. Zou, J. Pan, W. Du, and S. Chen, “Elastic Database Replication in the Cloud”, *ICA3PP 2015: Algorithms and Architectures for Parallel Processing*, Vol. 9531, pp. 667-681, 2015.
- [85] E. Insaf and G. Belalem, “Optimization of Tasks Scheduling by an Efficacy Data Placement and Replication in Cloud Computing”, *ICA3PP 2013: Algorithms and Architectures for Parallel Processing*, Vol. 8286, pp. 22-29, 2013.
- [86] Z. Gao, H. Min, X. Li, J. Huang, Y. Jin, A. Lei, S. Bourbonnais, M. Zheng, G. Fuh, “Optimizing Inter-data-center Large-Scale Database Parallel Replication with Workload-Driven Partitioning”, *Transactions on Large-Scale Data- and Knowledge-Centered Systems XXIV*, Vol. 9510, pp. 169-192, 2016.
- [87] S. U. Rehman, S. U. Khan, S. J. Ewen, N. Tziritas, J. Kolodziej, A. Y. Zomaya, S. A. Madani, N. Min-Allah, L. Wang, C. Xu, Q. M. Malluhi, J. E. Pecero, P. Balaji, A. Vishnu, R. Ranjan, S. Zeadally, H. Li, “Performance analysis of data intensive cloud systems based on data management and replication: a survey”, *Distributed and Parallel Databases*, Vol. 34, pp. 179–215, 2016.

- [88] L. Bellatreche, A.Kerkad, S. Breß, D. Geniet, "RouPar: Routinely and Mixed Query-Driven Approach for Data Partitioning", OTM Confederated International Conferences "On the Move to Meaningful Internet Systems", Vol. 8185, pp. 309-326, 2013.
- [89] S. Ahirrao, R. Ingle, "Scalable transactions in cloud data stores", Journal of Cloud Computing, Vol. 4, pp. 1-14, 2015.
- [90] F. Hassen, A. G. Touzi, "Towards a New Architecture for the Description and Manipulation of Large Distributed Data", Big Data in Complex Systems, Vol. 9, pp. 471-497, 2015.
- [91] W. Lehner, K. Sattler, "Transactional Data Management Services for the Cloud", Web-Scale Data Management for the Cloud, Vol. 1, pp. 59-90, 2013.
- [92] N. Phaphoom, X. Wang and P Abrahamsson, "Foundations and Technological Landscape of Cloud Computing", ISRN Software Engineering, vol. 2013, 2013, pp. 1-31.
- [93] L. Wiese, T. Waage and F. Bollwein, "A Replication Scheme for Multiple Fragmentations with Overlapping Fragments," in The Computer Journal, vol. 60, no. 3, pp. 308-328, March 2017.
- [94] A. Noraziah, A. Azila, A. Fairuzullah, N. Aqtar, A. Zinuddin and T. Herawan, "Fault Tolerance On Binary Vote Assignment Cloud Quorum (BVACQ) Replication Technique", Proceeding of the 2013 International Conference on Computer Science and Information Technology (CSIT 2013), Jogjakarta, ID, 2013, pp. 1-6.
- [95] A. A. Che Fauzi et al., "Managing Fragmented Database Replication for Mygrants Using Binary Vote Assignment on Cloud Quorum", Applied Mechanics and Materials, Vols. 490-491, pp. 1342-1346, 2014.
- [96] A. A. El Mrabti, N. Ammari, A. A. El Kalam, A. A. Ouahman and M. De Montfort, "New mechanism for Cloud Computing Storage Security" International Journal of Advanced Computer Science and Applications(IJACSA), Vol. 7, pp. 526-539, 2016.

- [97] V. Khandelwal, "Secure and Efficient Data Storage in Multi-clouds", *International Journal of Information and Computation Technology*, Vol. 3 no. 9, pp. 977-984, 2013.
- [98] S. Singh and R. Sapra, "Secure Replication management in cloud storage", *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, Vol. 3 no. 2, pp. 251-254, 2014.
- [99] S. Subashini and V. Kavitha, "A Metadata Based Storage Model for Securing Data in Cloud Environment," 2011 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, Beijing, 2011, pp. 429-434.
- [100] M. Scheidgen, A. Zubow, J. Fischer, T. H. Kolbe, "Automated and Transparent Model Fragmentation for Persisting Large Models", *International Conference on Model Driven Engineering Languages and Systems*, Vol. 7590, pp. 102-118, 2012.
- [101] J. O. Hauglid, N. H. Ryeng, K. Nørnvåg, "DYFRAM: dynamic fragmentation and replica management in distributed database systems", *Distributed and Parallel Databases*, Vol. 28, pp. 157-185, 2010.
- [102] S. Das, D. Agrawal, A. E. Abbadi, "ElasTraS: An Elastic, Scalable, and Self Managing Transactional Database for the Cloud", *ACM Transactions on Database Systems*, Vol. 38, pp. 1-45, 2013.
- [103] L. Wiese, "Horizontal Fragmentation and Replication for Multiple Relaxation Attributes", *British International Conference on Databases*, Vol. 9147, pp. 157-169, 2015.
- [104] D. Bermbach, M. Klems, S. Tai and M. Menzel, "MetaStorage: A Federated Cloud Storage System to Manage Consistency-Latency Tradeoffs," 2011 IEEE 4th International Conference on Cloud Computing, Washington, DC, 2011, pp. 452-459.
- [105] A. F. Barsoum and M. A. Hasan, "On Verifying Dynamic Multiple Data Copies over Cloud Servers", *International Association for Cryptologic Research Cryptology ePrint Archive*, Vol. 2011, pp. 1-30, 2011.