

Universidad de las Ciencias Informáticas
FACULTAD 6



Título: Base de Datos de alta disponibilidad para la plataforma VideoWeb 2.0.

Trabajo de Diploma para optar por el título de
Ingeniero en Ciencias Informáticas.

Autor: Isleivys Rodríguez Pupo

Tutor: Ing. Ángel Dayán Marín Abreu

La Habana, Junio 2013
“Año 55 de la Revolución.”

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconocer a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los ____ días del mes de _____ del año _____.

Isleivys Rodríguez Pupo

Firma del Autor

Ángel Dayán Marín Abreu

Firma del Tutor

DATOS DE CONTACTO

Tutor: Ing. Ángel Dayán Marín Abreu.

Especialidad de graduación: Ingeniero en Ciencias Informáticas.

Categoría docente: Instructor.

Años de experiencia en el tema: 7 años.

Año de graduado: 2009.

Correo electrónico: admarin@uci.cu

Agradecimientos

A mi mami por ser mi inspiración, la persona en quien pienso siempre, por darme la seguridad de que siempre podré contar con ella, por su amor y preocupación.

A mi papá por su amor, porque de una forma u otra siempre ha estado presente y se ha preocupado por mí.

A Arbey por convertirse en mi otro papá y apoyarme siempre, por su amor y dedicación y su preocupación.

A mi hermana por su amor, apoyo, preocupación y ser mi guía.

A mis abuelos por su amor incondicional y preocupación.

A Yoel el novio de mi hermana por darme ánimos para que aprendiera mucho.

A toda mi familia de forma general por estar siempre pendientes de mis estudios.

A Osiel por ser mi novio, mi amigo, por apoyarme siempre desde 1er año, por darme ánimos para seguir adelante y por estar ahí siempre.

A mi amigo Andris por sus consejos y ayudarme siempre desde primer año.

A Frank Alberto por ayudarme y por el valioso tiempo que me dedico.

A Maralis y Deymis por brindarme su amistad y por sus consejos.

A las buenas amistades que he encontrado en el trascurso de la carrera como Darinca, Elaine, Doina, Oricel, Yunet, Leydis, Breisy, Tahimi, Yudi, Adrian, Yorbeny, Ernesto, Yoandy y Yudiel.

A mis compañeros de aula, de apartamento y a todos los que han estado conmigo en estos cinco años.

A mi tutor porque supo guiarme y apoyarme en todo momento, por su sacrificio y entrega.

Al tribunal y oponente gracias por su tiempo, por sus sugerencias y consejos.

Dedicatoria

A mi mamá por dedicar su vida a mi hermana y a mí, por su apoyo, por hacerme quien soy, por guiarme y creer en mí.

Resumen

Debido al constante desarrollo de la informática y el auge que han alcanzado los contenidos audiovisuales han propiciado que las instituciones que se dediquen a proveer servicios de transmisión de contenidos audiovisuales. Con el surgimiento de estas tecnologías surge la necesidad de utilizar las bases de datos y los clústeres de alta disponibilidad para evitar la pérdida y gestión de la información referente a los contenidos audiovisuales y brindar servicios continuamente ante una gran concurrencia de usuarios. En la Universidad de las Ciencias Informáticas (UCI) en la facultad 6, se encuentra en desarrollo la Plataforma VideoWeb 2.0 que tiene como propósito la gestión, publicación y transmisión de contenidos audiovisuales a través de la red, para ello cuenta con diferentes módulos que generan un gran volumen de información, que deben ser accedidos por varios usuarios al mismo tiempo y en todo momento. La investigación tiene como objetivo solucionar el problema de la gestión, manipulación y la alta disponibilidad de los datos asociados a los contenidos audiovisuales, esto fue posible mediante un correcto diseño de una base de datos y la implementación de un clúster de alta disponibilidad para bases de datos. Se obtuvo como resultado de la investigación un diseño de la base de datos que responde a las necesidades planteadas por el proyecto y la implementación del clúster que ayuda a disminuir los tiempos de respuesta de las peticiones realizadas por una alta concurrencia de usuario.

Palabras claves: balanceo de carga, bases de datos, clúster, modelo, réplica, sistema gestor de bases de datos.

ÍNDICE DE CONTENIDOS

Introducción	1
Capítulo 1 Fundamentación teórica de la investigación	5
1.1 Introducción	5
1.2 Conceptos asociados al dominio del problema	5
1.2.1 Introducción a las bases de datos	5
1.2.2 Clasificación de las bases de datos	5
1.2.3 Modelo de Datos.....	5
1.2.4 Sistema Gestor de base de datos	7
1.2.5 Diseño de base de datos	11
1.2.6 Introducción a la tecnología clúster.....	12
1.2.7 Clasificación de los clústeres	12
1.3 Análisis del objeto de estudio	14
1.3.1 Descripción General.....	14
1.3.2 Situación problemática.....	16
1.4 Definición de tecnologías de desarrollo	17
1.4.1 Metodología de desarrollo de software.....	17
1.4.2 Lenguaje de modelado.....	18
1.4.3 Herramienta CASE.....	19
1.4.4 Sistema Gestor de base de Datos.....	20
1.4.5 Herramienta para balancear las cargas en un clúster de alta disponibilidad	20
1.4.6 Herramienta para la alta disponibilidad en el clúster	22
1.5 Pruebas para medir el rendimiento del clúster y comprobar su funcionamiento	23
1.5.1 Pruebas de configuración	23
1.5.2 Pruebas de rendimiento	23
1.5.3 Pruebas de carga.....	23
1.5.4 Pruebas de benchmark	23
1.6 Conclusiones Parciales	24
Capítulo 2 Descripción y análisis de la propuesta de solución.	25
2.1 Introducción	25

2.2	Requisitos funcionales y no funcionales	25
2.3	Diseño de la Base de Datos	28
2.3.1	Nomenclatura de la base de datos.....	28
2.3.2	Patrones de diseño	28
2.3.3	Descripción de las tablas	29
2.4	Propuesta de diseño de la solución de clúster	33
2.4.1	Diseño lógico	34
2.4.2	Diseño físico	35
2.5	Configuraciones	37
2.5.1	Configuraciones realizadas en PostgreSQL.....	38
2.5.2	Configuraciones realizadas en Pgpool-II	38
2.5.3	Configuraciones realizadas en el Heartbeat.....	38
2.6	Políticas de seguridad para la base de datos de la Plataforma VideoWeb 2.0	40
2.7	Conclusiones Parciales	40
Capítulo 3 Validación de la Base de Datos de Alta Disponibilidad		42
3.1	Introducción	42
3.2	Validación teórica del diseño de la base de datos	42
3.2.1	Integridad de los datos.....	42
3.2.2	Normalización de la base de la base de datos	44
3.2.3	Análisis de la redundancia de la información	47
3.3	Validación funcional del diseño del clúster obtenido	48
3.3.1	Pruebas de carga y benchmark	48
3.3.2	Pruebas de configuración del clúster	48
3.3.3	Pruebas al diseño genérico de la propuesta de solución.....	49
3.4	Análisis de los resultados obtenidos	51
3.5	Conclusiones Parciales	53
Conclusiones generales		54
Recomendaciones		55
Referencias Bibliográficas		¡Error! Marcador no definido.
Bibliografía		59
Glosario de términos		¡Error! Marcador no definido.

Anexos 61

ÍNDICE DE FIGURAS

Figura 1. Fases de diseño 11
Figura 2. Arquitectura del clúster.....34
Figura 3. Alta disponibilidad en el clúster35

Figura 4. Representación lógica del clúster.....	35
Figura 5. Representación del clúster cuando funciona normalmente.....	36
Figura 6. Representación física cuando falla un nodo.	36
Figura 7. Representación física cuando falla el balanceador primario.	37
Figura 8. Diseño general del clúster.....	37
Figura 9. Formato de authkeys.....	39
Figura 10. Integridad Referencial.	43
Figura 11. Integridad de Dominio.	44
Figura 12. Integridad de Entidad.	44
Figura 13. Fases de normalización.....	46
Figura 14. Análisis del % del CPU en las cuatro iteraciones.....	51
Figura 15. Análisis del % de la RAM en las cuatro iteraciones.	52
Figura 16. Análisis de las consultas efectuadas en cada uno de los nodos.	52
Figura 17. Análisis de las transacciones por segundo realizadas.	53
Figura 18. Análisis de los recursos cuando funciona un solo nodo.....	53

ÍNDICE DE TABLAS

Tabla 1: archivo_multimedia.....	30
Tabla 2: almacenamiento.....	30
Tabla 3: almacen_local.....	30
Tabla 4: tipologia_data.....	31
Tabla 5: publicacion.....	31
Tabla 6: solicitud.....	31
Tabla 7: naturaleza.....	32
Tabla 8: empresa_externa.....	32
Tabla 9: cliente.....	32
Tabla 10. role.....	32
Tabla 11: role_permission.....	33
Tabla 12: streaming.....	33
Tabla 13: credito.....	33
Tabla 14. Pruebas de diseño genérico.....	50
Tabla 15. Consumo de recursos.....	50
Tabla 16: rango_ip.....	62
Tabla 17. rango_horario.....	62

Introducción

En la actualidad el desarrollo de las tecnologías y las comunicaciones han sido el motor impulsor para que la sociedad utilice con menos frecuencia los medios tradicionales de comunicación masiva, dando paso a que se inclinen por la utilización de servicios de transmisión de contenido audiovisuales a través de sus ordenadores. Las Tecnologías de Información y Comunicación (TIC) han evolucionado, convirtiéndose en un eslabón fundamental para la gestión de la información.

En sus inicios los datos se almacenaban manualmente, lo que lo hacía un proceso engorroso y lento. Posteriormente con el surgimiento de las computadoras y las necesidades de automatizar los procesos y actividades que se realizaban diariamente en las empresas, surgen las bases de datos. Desde el punto de vista informático, una base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten guardar información para posteriormente ser usada (1).

Actualmente las bases de datos ocupan un lugar determinante en cualquier área comercial o tecnológica. Han solucionado varios problemas de almacenamiento y gestión de la información, pues con el avance de las nuevas tecnologías se han encaminado a automatizar el manejo de grandes volúmenes de información en numerosos centros.

Las bases de datos y el auge de internet han propiciado el intercambio de información desde cualquier parte del mundo, ampliando el alcance de las aplicaciones de bases de datos en todas las organizaciones. Para lograr que los datos sean accesibles en todo momento por los usuarios, es importante contar con un sistema de alta disponibilidad.

La alta disponibilidad está basada en la replicación de elementos de una base de datos y tiene el objetivo de proporcionar máxima disponibilidad de los servicios que se ofrecen, de tal manera que estos se brinden ininterrumpidamente. Se consigue alta disponibilidad haciendo redundantes los sistemas y cuando se produce la caída de un nodo que brinda el servicio, un nodo que cuenta con el mismo sistema de almacenamiento, es decir con la misma base de datos toma el control y continúa brindando el servicio (2).

Con el acelerado avance de la informática y las bases de datos, han evolucionado también los contenidos audiovisuales, siendo muy utilizados por empresas que se dediquen a la transmisión de contenidos audiovisuales a través de la red. Estos sistemas a su vez necesitan de un sistema de almacenamiento que les permita una correcta y rápida gestión de los datos asociados a los contenidos audiovisuales. Estas bases de

datos se encargan de almacenar información tales como el título del material, la duración, el autor, el formato, entre otros elementos. Con la correcta gestión y manipulación de este tipo de información se logra que las búsquedas de los materiales audiovisuales sean más rápidas y eficientes.

Actualmente en el departamento de Señales Digitales perteneciente a la facultad 6, de la Universidad de las Ciencias Informáticas (UCI), se está desarrollando la Plataforma VideoWeb 2.0. El desarrollo de dicha plataforma se encuentra liderado por el proyecto Catalogación y Publicación de Medias, que tiene como principal objetivo desarrollar un sistema informático para la transmisión de contenidos audiovisuales a través de una infraestructura de red. La plataforma debe permitir la gestión y administración de los contenidos audiovisuales a publicar y la solicitud de los materiales previamente almacenados, para ello cuenta con diferentes módulos que generan gran cantidad de información. La información debe ser accedida por varios usuarios al mismo tiempo y en todo momento. El acceso concurrente a la información por parte de los usuarios podría causar la sobrecarga del servidor de base de datos, provocando que éste no pueda atender todas las solicitudes, haciendo que el sistema se vuelva inestable.

La base de datos con que cuenta la Plataforma VideoWeb fue diseñada para la versión 6 de Drupal, al surgir la versión 7, el proyecto de Catalogación y Publicación de Medias toma la decisión de migrar hacia esa versión e implementar una nueva versión de VideoWeb. A esta nueva versión se le agregan nuevas funcionalidades al sistema como la catalogación, solicitud y descarga de los materiales audiovisuales, por lo que la base de datos con que cuenta VideoWeb en su versión 1.0 no satisface los requisitos de la nueva versión que se desea implementar.

Para darle solución a la problemática planteada se define como **problema a resolver**: ¿Cómo lograr la gestión, manipulación y alta disponibilidad de los datos utilizados en la plataforma VideoWeb 2.0?

Para darle solución a lo antes planteado, se propone como **objetivo general** diseñar una base de datos de alta disponibilidad para la plataforma VideoWeb 2.0.

Surgiendo como **objeto de estudio** el diseño de bases de datos relacionales de alta disponibilidad para la gestión de información asociada a los contenidos multimedia, enmarcado en el **campo de acción** correspondiente al diseño de la base de datos de alta disponibilidad para la plataforma VideoWeb 2.0.

La **idea a defender** sobre la cual se sustenta la investigación es con el diseño de la base de datos de alta disponibilidad para la plataforma VideoWeb 2.0 se pretende garantizar una correcta organización, gestión y alta disponibilidad de los datos referentes a los contenidos multimedia.

Para dar cumplimiento al objetivo general del trabajo se delimitaron las siguientes tareas:

- Caracterizar los procesos de gestión y manipulación de datos en la plataforma VideoWeb 2.0.
- Caracterizar el gestor de bases de datos a utilizar.
- Realizar el diseño de la base de datos de la plataforma VideoWeb 2.0.
- Implementar el clúster de alta disponibilidad de bases de datos para la plataforma VideoWeb 2.0.
- Realizar pruebas para validar el correcto funcionamiento de la base de datos.

Con el correcto cumplimiento de las tareas se espera obtener como **posibles resultados**:

- Documentación que recoja el resultado de la investigación realizada.
- Artefactos resultantes del diseño de la base de datos para la plataforma VideoWeb 2.0.
- Manual de configuración del clúster de alta disponibilidad de base de datos con el gestor de base de datos utilizado
- Documentación sobre las pruebas realizadas a la base de datos de la Plataforma VideoWeb 2.0.

Para lograr un mejor entendimiento y desarrollo de la investigación se emplean métodos de investigación teóricos, dentro de los cuales se encuentra:

- **Analítico – sintético**, a través de este método se logró llegar a comprender cómo es el proceso de almacenamiento de una base de datos de alta disponibilidad, la estructura que debe tener la información y cómo implementar un clúster para bases de datos.
- **Histórico – lógico**, se utilizó para analizar el surgimiento de las bases de datos y su evolución a lo largo de la historia y para caracterizar el gestor de bases de datos a utilizar.

Este documento se estructura en tres capítulos, los cuales recogen toda la información necesaria para el entendimiento del mismo, además contará con las conclusiones, recomendaciones, referencias bibliográficas, bibliografía, anexos y glosario de términos.

En el **Capítulo I**: Fundamentación teórica de la investigación: se realiza un estudio acerca de las bases de datos y del clúster de alta disponibilidad para bases de datos, se analizan el objeto de estudio, la situación problemática, las herramientas que existen para el diseño de base de datos, las características del gestor de base de datos a utilizar, las herramientas más usadas para el diseño de las bases de datos y herramientas para lograr la alta disponibilidad, el balanceo de carga y réplica de los datos.

Introducción

En el **Capítulo II**: Descripción y análisis de la solución propuesta: se describe el modelo de datos, las configuraciones, la seguridad de la base de datos y la implementación del clúster de alta disponibilidad.

En el **Capítulo III**: Validación de la base de datos de alta disponibilidad: se describen los resultados alcanzados luego de realizadas las pruebas de validación necesarias para comprobar el correcto funcionamiento tanto de la base de datos, así como del clúster de alta disponibilidad.

Capítulo 1

Fundamentación teórica de la investigación

1.1 Introducción

En el presente capítulo se definen un conjunto de conceptos que posibilitarán un mejor entendimiento de la investigación, se analiza el objeto de estudio y la situación problemática de la misma. Se realiza un estudio sobre las bases de datos y las principales tecnologías para la configuración de un clúster de alta disponibilidad para bases de datos, además de un breve enfoque a los sistemas gestores de bases de datos, especialmente PostgreSQL, las herramientas de modelado visual y herramientas a utilizar para balancear las cargas de un clúster de bases de datos de alta disponibilidad y réplica de los datos.

1.2 Conceptos asociados al dominio del problema

1.2.1 Introducción a las bases de datos

El término de bases de datos fue percibido por primera vez en 1963, en California, Estados Unidos (3). Inicialmente una base de datos podía ser una de las antiguas bibliotecas o cualquier tipo de información que fue conservada en papel y seguidamente almacenada en un estante. En la actualidad y debido al desarrollo tecnológico de campos como la informática, se han convertido en una de las herramientas más difundidas en el mundo de la información, se encuentran en formato digital lo que ofrece un amplio rango de soluciones al problema de almacenar datos, siendo utilizadas como fuentes de almacenamiento y recuperación de información en todos los campos a nivel social, científico, económico y político.

El concepto de bases de datos ha sido definido por varios autores, a continuación se exponen algunos conceptos:

- Conjunto de datos interrelacionados entre sí, almacenados con carácter más o menos permanente en la computadora. O sea, que una base de datos puede considerarse una colección de datos variables en el tiempo (4).
- Serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los distintos sistemas de información (5).
- Una base de datos es un conjunto de datos persistentes que es utilizado por los sistemas de aplicaciones de alguna organización dada (6).

Capítulo 1 Fundamentación teórica de la investigación

Las bases de datos básicamente son un almacén o sistema formado por un conjunto de datos pertenecientes a un mismo contexto, almacenados de forma organizada y relacionados entre sí, permitiendo guardar información para ser usada posteriormente.

1.2.2 Clasificación de las bases de datos

Las bases de datos pueden ser clasificadas en diversas formas teniendo en cuenta los siguientes criterios:

Según la forma en que cambia la información almacenada, es decir variabilidad:

Bases de datos estáticas: Son bases de datos en las que no se puede modificar la información que está guardada, solo se puede consultar, es decir son bases de datos de solo lectura (7).

Bases de datos dinámicas: Son bases de datos en las que la información almacenada está en constante cambio, ya que sobre ellas se pueden realizar operaciones tales como: adicionar, actualizar y eliminar información (7).

Según la información que almacenan:

Bases de datos bibliográficas: Son bases de datos que almacenan la información necesaria para poder localizar la fuente primaria. Puede contener un fragmento de la publicación original, pero nunca el texto completo (8).

Bases de datos de texto completo: Almacenan las fuentes primarias, es decir, guardan toda la información íntegra de las fuentes primarias, no fragmentos, ni referencias (8).

Directorios: Son los directorios y guías telefónicas en formatos electrónicos (8), se puede mencionar como ejemplo, los directorios digitales como los que publica la Empresa de Telecomunicaciones de Cuba (ETECSA) con los números telefónicos de sus usuarios.

1.2.3 Modelo de Datos

Un modelo de datos (MD) es un conjunto de conceptos y reglas que permiten definir un mundo abstracto y teórico, tal que las conclusiones que se puedan deducir de él coincidan con las manifestaciones aparentes del mundo real (9).

Modelos de datos jerárquicos

Los modelos de datos jerárquicos almacenan la información de forma jerárquica, donde los datos se van almacenando de forma similar a un árbol. Este árbol se utiliza para la representación lógica de los datos, el

Capítulo 1 Fundamentación teórica de la investigación

cual está compuesto por elementos llamados nodos. El nivel más alto del árbol se denomina raíz. Cada nodo representa un registro con sus correspondientes campos (10).

Este modelo es especialmente útil en aplicaciones que manejan grandes volúmenes de información y datos compartidos, permitiendo crear estructuras estables y de gran rendimiento.

La poca flexibilidad de este modelo puede obligar a la introducción de redundancias cuando es preciso instrumentar, mediante el modelo jerárquico, situaciones del mundo real que no responden a una jerarquía (11).

Modelo de datos Orientado a Objetos

Las bases de datos orientadas a objetos se crearon para tratar de satisfacer las necesidades de estas nuevas aplicaciones. La orientación a objetos ofrece flexibilidad para manejar algunos de estos requisitos y no está limitada por los tipos de datos y los lenguajes de consulta de los sistemas de bases de datos tradicionales. Una característica clave de las bases de datos orientadas a objetos es la potencia que proporcionan al diseñador al permitirle especificar tanto la estructura de objetos complejos, como las operaciones que se pueden aplicar sobre dichos objetos (12).

Una base orientada a objetos es una base de datos que incorpora todos los conceptos importantes del paradigma de objetos:

- Encapsulación: Propiedad que permite ocultar información al resto de los objetos, impidiendo así accesos incorrectos o conflictos.
- Herencia: Propiedad a través de la cual los objetos heredan comportamientos dentro de una jerarquía de clases.
- Polimorfismo: Propiedad de una operación mediante la cual puede ser aplicada a distintos tipos de objetos.

Modelo de datos de Red

Una base de datos en red consiste en un conjunto de registros conectados entre sí mediante punteros. Los registros son en muchos aspectos parecidos a las entidades del modelo entidad-relación (E-R). Cada registro es un conjunto de campos (atributos), cada uno de los cuales sólo contiene un valor. Los punteros son asociaciones entre exactamente dos registros.

Capítulo 1 Fundamentación teórica de la investigación

A diferencia del modelo jerárquico, en este modelo, un hijo puede tener varios padres. Ofrece una solución eficiente al problema de la redundancia de datos. Este modelo de datos permite representar relaciones N: M (mucho a mucho) sin embargo es poco utilizado (10).

Modelo de datos Relacional

En 1970 Edgar Frank Codd introdujo el modelo relacional, donde todos los datos están estructurados a nivel lógico como tablas formadas por filas y columnas, aunque a nivel físico pueden tener una estructura completamente distinta. Un punto fuerte del modelo relacional es la sencillez de su estructura lógica (13).

Algunas de las ventajas del modelo relacional son (13):

- Define un álgebra, llamada álgebra relacional a partir de la cual se realizan todas las manipulaciones posibles sobre las relaciones, que se obtienen mediante el uso de operadores.
- Garantía de independencia de los datos.
- Ofrece la normalización por ser más comprensible y aplicable.
- Garantiza la integridad referencial, así al eliminar un registro elimina todos los registros relacionados dependientes.
- Garantiza herramientas para evitar la duplicación de registros, a través de campos claves o llaves.

Algunas de sus principales características son (10):

- Puede ser entendido y usado por cualquier usuario.
- Permite ampliar el esquema conceptual sin modificar las aplicaciones de gestión.
- Los usuarios no necesitan saber dónde se encuentran los datos físicamente.

1.2.4 Sistema Gestor de base de datos

Los sistemas gestores de bases de datos (SGBD) son aplicaciones que permiten a los usuarios definir, crear y mantener la base de datos y proporcionar un acceso controlado de la misma. Un SGBD debe prestar los siguientes servicios (14):

- Definición y creación de la base de datos.
- Manipulación de los datos realizando consultas, inserciones y actualizaciones.
- Acceso controlado a los datos mediante mecanismos de seguridad.
- Mantener la integridad de los datos.
- Controlar la concurrencia de los datos.

Capítulo 1 Fundamentación teórica de la investigación

- Mecanismos de copias de respaldo de los datos.

En general un SGBD es un conjunto de programas, procedimientos, lenguajes, etc., que facilita tanto a los usuarios no informáticos como a los programadores, analistas o al administrador de la BD de los medios necesarios para describir, recuperar, manipular datos almacenados en una base de datos, manteniendo su integridad, confidencialidad y seguridad.

Los SGBD proveen facilidades para la manipulación de grandes volúmenes de datos (14):

- Simplifican la programación de equipos de consistencia.
- Manejando las políticas de respaldo adecuadas, garantizan que los cambios de la base de datos serán siempre consistentes.
- Organizan los datos con un impacto mínimo en el código de los programas.
- Bajan drásticamente los tiempos de desarrollo y aumentan la calidad del sistema desarrollado, si son bien explotados por los desarrolladores.
- Usualmente, proveen interfaces y lenguajes de consulta que simplifican la recuperación de los datos.

Actualmente existen varios sistemas gestores de bases de datos, algunos son:

Oracle

Sistema de gestión de base de datos objeto-relacional (ORDBMS por el acrónimo en inglés de *Object-Relational Data Base Management System*), desarrollado por Oracle Corporation¹. Es considerado uno de los sistemas de bases de datos más completos, destacando su soporte de transacciones, estabilidad, escalabilidad y soporte multiplataforma. Oracle es básicamente una herramienta cliente/servidor para la gestión de Bases de Datos.

Oracle es el sistema gestor de base de datos para aplicaciones y entornos empresariales con mayor adopción en el mundo. Es un sistema de base de datos comercial líder a nivel mundial. El sistema puede funcionar tanto en plataforma Microsoft Windows como en Linux (15).

Microsoft SQL Server

Sistema para la gestión de bases de datos producido por Microsoft² basado en el modelo relacional y el más extendido para entornos Microsoft. Dispone de todas las funcionalidades de un gestor de base de datos,

¹Oracle Corporation es una de las mayores compañías de software del mundo. Sus productos van desde bases de datos hasta sistemas de gestión, fundada en 1987.

²Microsoft es una empresa de origen estadounidense, fundada el 4 de abril de 1975 por Bill Gates y Paul Allen.

Capítulo 1 Fundamentación teórica de la investigación

índices, desencadenadores y procedimientos almacenados que hacen un sistema gestor de base de datos completo. Por su propia naturaleza no puede ser utilizado en servidores Linux.

Microsoft SQL Server cuenta con soportes para transacciones, incluye también un entorno gráfico de administración, que permite el uso de comandos DDL (lenguaje de definición de datos) y DML (lenguaje de manipulación de datos) gráficamente, permite trabajar en modo cliente servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información. Además permite administrar información de otros servidores de datos. Se encuentra distribuido bajo licencia propietaria. Algunas de sus desventajas son que no maneja compresión de datos por tanto ocupa mucho espacio en disco y no cuenta con soporte para tablas HEAP, son tablas que se almacenan de forma temporal en memoria. Se les denomina HEAP debido a que los registros se almacenan desordenadamente insertándose en el primer espacio disponible dentro de las páginas pertenecientes a la tabla, este tipo de tablas son muy rápidas para ingresos de información (16).

MySQL

Es un sistema de gestión de bases de datos relacionales, multi-hilo y multiusuario con más de seis millones de instalaciones. Surgió alrededor de la década de los 90, creado por Michael Widenis, licenciado bajo la Licencia Pública General de la GNU (más conocida como *GNU GPL*) (17).

Las principales características de este gestor de base de datos son las siguientes (17):

- Aprovecha la potencia de sistemas multiprocesadores gracias a su implementación multi-hilo.
- Soporta gran cantidad de tipos de datos para las columnas.
- Gran portabilidad entre sistemas
- Soporta hasta 32 índices por tablas.
- Gestión de usuario y contraseña, manteniendo un buen nivel de seguridad en los datos.

Muchas de las organizaciones más grandes y de más rápido crecimiento del mundo, como Facebook, Google, Adobe, Alcatel Lucent y Zappos se basan en MySQL para ahorrar tiempo y dinero en sitios web con grandes volúmenes de datos, sistemas críticos de negocio y software empaquetado. MySQL se ejecuta en más de 20 plataformas, incluyendo Linux, Windows, Mac OS, Solaris, AIX de IBM. Además ofrece una amplia gama de herramientas de bases de datos, servicios de soporte, formación y consultoría (18).

PostgreSQL

Es un gestor de base de datos objeto-relacional (*ORDBMS*) basado en el proyecto POSTGRES, de la

Capítulo 1 Fundamentación teórica de la investigación

universidad de Berkeley, aunque no es un sistema de gestión de bases de datos puramente orientado a objetos, es una derivación libre (*OpenSource*) de este proyecto y utiliza el lenguaje SQL92/SQL99. Está escrito en C++ usando la librería gráfica multiplataforma wxWidgets, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows y se encuentra distribuido bajo la licencia BSD (*Berkeley Software Distribution*) (19).

La seguridad en PostgreSQL se materializa en tres aspectos (20):

- Seguridad en la manipulación de los ficheros de PostgreSQL.
- Seguridad en los accesos de los clientes.
- Definición de los privilegios a los usuarios PostgreSQL para acceder a los objetos de la base de datos.

A continuación se enumeran las principales características (20):

- Implementación del estándar SQL92/SQL99.
- Soporta distintos tipos de datos como datos de tipo fecha, elementos gráficos, datos de sobre redes y cadenas de bits.
- Incorpora una estructura de datos array.
- Incorpora funciones de diverso índole: manejo de fechas, geométricas y orientadas a operaciones con redes.
- Permite la declaración de funciones propias, así como la definición de disparadores.
- Soporta el uso de índices, reglas y vistas.
- Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.

PostgreSQL está considerado como uno de los gestores de bases de datos de código abierto más avanzados del mundo. Proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como Oracle. Se considera como uno de los sistemas de bases de datos más completos destacando su soporte de transacciones, estabilidad, escalabilidad, además es multiplataforma. PostgreSQL aproxima los datos a un modelo objeto-relacional y es capaz de manejar complejas reglas. Las características que más se han tenido en cuenta durante su desarrollo son: estabilidad, potencialidad, robustez, facilidad de administración e implementación de estándares.

Capítulo 1 Fundamentación teórica de la investigación

1.2.5 Diseño de base de datos

El diseño de una base de datos se descompone en tres subprocesos o tres fases fundamentales: Diseño Conceptual, Diseño Lógico y Diseño Físico:

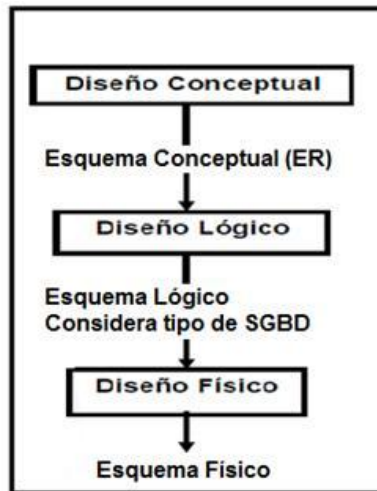


Figura 1. Fases de diseño.

Diseño Conceptual:

Es una representación de todo el contenido de la información de la base de datos, constituye la primera fase de diseño, que es obtener una buena representación de los recursos de información. La primera fase cuenta de dos momentos: análisis de requisitos, donde se centra el trabajo en definir qué es lo que se va a representar, y la conceptualización, donde se piensa en cómo se va a proceder para representar lo antes definido (21).

Diseño Lógico:

El diseño lógico parte del esquema conceptual y da como resultado un esquema lógico. Un esquema lógico es una descripción de la estructura de la base de datos en términos de las estructuras de datos que puede procesar un tipo de Sistema Gestor de Base de Datos (SGBD). Además es una fuente de información para el diseño físico, juega un papel importante para la etapa de mantenimiento del sistema (21).

Diseño Físico:

El diseño físico parte del diseño lógico y da como resultado un esquema físico. Un esquema físico es una descripción de la implementación de una base de datos en memoria secundaria: las estructuras de

Capítulo 1 Fundamentación teórica de la investigación

almacenamiento y los métodos utilizados para tener un acceso eficiente a los datos. Por ello, el diseño físico depende del SGBD concreto y el esquema físico se expresa mediante su lenguaje de definición de datos (21).

1.2.6 Introducción a la tecnología clúster

La tecnología de clústeres ha evolucionado en apoyo de actividades que van desde aplicaciones de supercómputo y software de misiones críticas, servidores web y comercio electrónico, hasta bases de datos de alto rendimiento, entre otros usos.

Un clúster es un conjunto de computadoras, a menudo con semejantes componentes de hardware, que se interconectan entre sí a través de un sistema de red de alta velocidad y son capaces de elevar la eficiencia para realizar determinadas tareas que individualmente no podrían realizar debido a la creciente necesidad de potencia computacional que demandan algunas aplicaciones (22).

Simplemente, un clúster es un sistema paralelo o distribuido que posee una arquitectura formada por un grupo de equipos independientes denominados nodos, que se conectan entre sí, ejecutan acciones de manera conjunta y aparecen como un único recurso computacional ante aplicaciones y clientes, cumple además con los requerimientos de un sistema tolerante a fallos.

1.2.7 Clasificación de los clústeres

De acuerdo a la función específica para la que han sido diseñados, los clústeres se catalogan como implementados para lograr alto desempeño o alto rendimiento, mientras que otros se utilizan específicamente para lograr alta disponibilidad y balancear las cargas.

Clúster de Alto Rendimiento (HP):

En un clúster de alto rendimiento se ejecutan tareas que requieren de gran capacidad computacional, grandes cantidades de memoria, o ambos a la vez. Este tipo de clúster divide las tareas en tareas más pequeñas y las reparte entre los nodos que lo conforman para que sean calculadas en ellos y así agilizar el procesamiento de los datos. Estos sistemas se implementan en un ambiente de programación paralela utilizando algoritmos que hacen uso de recursos compartidos tales como CPU, memoria, datos y servicios. Entre las aplicaciones más comunes de clústeres de alto rendimiento se encuentra el pronóstico numérico del estado del tiempo, astronomía, investigación en criptografía y análisis de imágenes (23).

Clúster de Balanceo de Carga:

El clúster de balanceo de carga se encarga de colocar en paralelo varios servidores (servidores reales) capaces de brindar el mismo servicio y de alguna forma repartir el trabajo entre ellos, tal que los clientes vean

Capítulo 1 Fundamentación teórica de la investigación

servidas sus solicitudes en tiempos menores y aceptables. Los clientes deben ver al conjunto de servidores como si fuera uno solo (servidor virtual). Un clúster de balanceo de carga tiene peculiaridades de clúster de alta disponibilidad y alto rendimiento (24).

Clúster de Alta Disponibilidad (HA):

Un clúster de alta disponibilidad es un conjunto de dos o más servidores, que se caracteriza por compartir el sistema de almacenamiento y estar constantemente monitorizándose entre sí. Si se produce un fallo del hardware o de los servicios de alguno de los nodos que forman el clúster, el software de alta disponibilidad es capaz de rearrancar automáticamente los servicios que han fallado en cualquiera de los otros equipos del clúster. Y cuando el servidor que ha fallado se recupera, los servicios se migran de nuevo a la máquina original (25).

En general las razones para implementar un clúster de alta disponibilidad son que aumentan la disponibilidad, mejoran el rendimiento, son tolerantes ante fallos, permiten la recuperación ante fallos en tiempo aceptable.

Clúster para bases de datos

Un clúster de bases de datos está compuesto por servidores dedicados al almacenamiento de datos usando un SGBD para ello, garantizando la comunicación entre los servidores que propicie la replicación de los datos.

Un clúster de base de datos es una arquitectura en la que se tienen varios equipos con parte de los datos del usuario trabajando al unísono como un solo sistema. La arquitectura de un clúster de base de datos viene definida por la manera en que se almacenan los datos en cada nodo (26).

Soluciones de clúster

En el mundo existe una tendencia al aumento del uso de este tipo de tecnología para servidores de bases de datos. Un ejemplo de esto es el clúster de aplicaciones de la compañía Google el cual cuenta con más de 60 000 servidores repartidos en diferentes localidades del mundo (Norteamérica, Asia y Europa) (27).

Cada uno de los servidores de datos de Google tiene instalado un sistema operativo Linux y sobre ellos se realizan técnicas de balanceo de carga y replicación. El uso de un sistema distribuido de almacenamiento de datos no centralizado garantiza un menor costo en la implementación del clúster y reduce las posibilidades de ocurrencia de fallos, por otra parte aumenta la escalabilidad y el rendimiento del sistema lo cual hace posible que puedan ser atendidas hasta 40 millones de búsquedas por día (27).

Soluciones de clúster en Cuba

Capítulo 1 Fundamentación teórica de la investigación

En Cuba no está difundido el uso de la tecnología clúster para servidores de bases de datos, aunque se emplean técnicas de clúster con otros fines, como por ejemplo ETECSA, que utiliza el software Vyatta Core³ para la implementación de routers de bajo costo en la infraestructura de redes en la Red Nacional, este software a su vez utiliza técnicas de clusterización de alta disponibilidad y de balanceo de carga en la red.

Otro ejemplo del uso de la tecnología de clúster fuera del ámbito de las bases de datos, es el Centro de Ingeniería Genética y Biotecnología que cuenta con un clúster o conjunto de computadoras construidas mediante la utilización de hardwares comunes, que se comportan como si fuesen una única computadora. En él se almacenan y actualizan datos de alto flujo, tanto de los buscados en Internet, como los generados por proyectos propios, con los que se realizan cálculos y comparaciones complejas (28).

1.3 Análisis del objeto de estudio

A continuación se describe el objeto de estudio, el cual estará centrado en diseño de bases de datos relacionales de alta disponibilidad para la gestión de información asociadas a contenidos multimedia.

1.3.1 Descripción General

En la actualidad con el acelerado desarrollo de la informática y con ella el de las bases de datos, han evolucionado también los contenidos audiovisuales, siendo muy utilizados en la actualidad por empresas televisoras o empresas que se dediquen a proveer servicios de transmisión de materiales audiovisuales, dando paso a que se utilicen las bases de datos para gestionar información de contenidos multimedia.

Las bases de datos de contenidos audiovisuales son un conjunto de información referentes a los contenidos audiovisuales que se encuentra sistemáticamente organizada y que pueden ser recuperados y manipulados. La información almacenada podría tratarse de datos tales como el autor y título de un material audiovisual.

Debido al gran volumen de información que generan los sistemas informáticos que manipulan contenidos audiovisuales y los grandes requerimientos de hardware necesarios para lograr el correcto almacenamiento y acceso concurrente a la misma por numerosos usuarios es necesario realizar el diseño de una base de datos que garantice la alta disponibilidad. La alta disponibilidad se puede lograr con la implementación de un clúster para bases de datos, posibilitando la máxima disponibilidad y confiabilidad de los servicios que ofrece el sistema, de tal manera que estos se brinden ininterrumpidamente, aumenten la velocidad de procesamiento e incrementen el número de transacciones y disminuya el tiempo de respuesta.

³Vyatta Core es un software de código abierto para enrutamiento y seguridad en una red.

Capítulo 1 Fundamentación teórica de la investigación

Las bases de datos relacionales son una elección adecuada para realizar el diseño de las bases de datos que gestionan contenidos audiovisuales, debido a que desde los años 80 es el modelo más utilizado para modelar problemas reales y administrar los datos de forma dinámica, permitiendo además una mayor eficacia, flexibilidad y confianza en el tratamiento de los datos.

Una base de datos relacional es una base de datos en donde todos los datos visibles al usuario están organizados estrictamente como tablas de valores y donde todas las operaciones de la base de datos operan sobre dichas tablas. Las bases de datos son percibidas por los usuarios como una colección de relaciones normalizadas de diversos grados que varían con el tiempo.

La mayor parte de las bases de datos y sistemas de información actuales se basan en el modelo relacional, debido a que es fácil de entender y de utilizar para un usuario, cuenta con herramientas para evitar la duplicidad de registros, favorece la normalización por ser más comprensible y aplicable y la información puede ser recuperada o almacenada mediante consultas.

RUP plantea que el propósito de diseñar una base de datos es asegurar que los datos persistentes son almacenados consistentemente y de una forma eficiente y definir el comportamiento que debe ser implementado en la base de datos. Para obtener dicho resultado se debe mapear el diseño de clases persistente al modelo de datos, elaborando el modelo de datos a partir de las clases persistentes encontradas (29).

Según la metodología RUP el diseño de una base de datos debe desarrollar una serie de artefactos y actividades garantizando así su cumplimiento. Entre estos artefactos y actividades se pueden encontrar:

- Identificar las clases persistentes.
- Elaborar el diagrama de clases persistente.
- Crear claves primarias y restricciones de integridad.
- Definir las reglas de integridad referencial y de la información.
- Normalizar el diseño de la base de datos para su optimización.

En los últimos años se ha presenciado un incremento paulatino en el uso de los contenidos audiovisuales en la web. Gracias a la incesante evolución de las tecnologías de la información y de la comunicación, todo tipo de organizaciones y particulares son ahora capaces de crear, editar, publicar y consumir este tipo de contenidos de una forma rápida y sencilla (30).

Capítulo 1 Fundamentación teórica de la investigación

Existen varias aplicaciones que se dedican a gestionar contenidos audiovisuales en el ámbito internacional, entre las que se encuentra Youtube. Youtube es una plataforma que permite la publicación de video online, fundada en febrero del 2005, en la cual los usuarios de cualquier parte del mundo pueden subir y compartir videos, con su título, descripción o resumen, calidad visual y duración, los cuales son publicados en dependencia de la categoría a la que pertenezcan y pueden ser visualizados en la interfaz del sitio mediante un reproductor flash embebido en el navegador. Los datos asociados a los videos que son publicados por los usuarios son almacenados en una base de datos, permitiendo posteriormente a los usuarios realizar búsquedas de los materiales audiovisuales almacenados, a través de etiquetas de metadato, títulos y descripciones que los usuarios asignan a sus videos.

En la Universidad de las Ciencias Informática, en la facultad 6, en el departamento de Señales Digitales, existen aplicaciones en las cuales de una u otra forma se manipulan contenidos audiovisuales. Estas tienen como particularidad que son aplicaciones o plataformas que almacenan los datos asociados a los contenidos audiovisuales, provocando que el volumen de información a almacenar sea de gran magnitud. Estas plataformas pueden ser usadas para crear, editar, gestionar y publicar contenidos multimedia en diversos formatos, facilitan el acceso a las publicaciones de los contenidos a un rango mayor de usuarios, estas aplicaciones permiten comprimir y convertir el video y audio a un determinado formato digital, manteniendo la calidad del video a un nivel muy aceptable.

Estas plataformas son sistemas flexibles, capaces de adaptarse a varios entornos y/o dar respuesta a varios tipos de necesidades, son integrables con múltiples infraestructuras de bases de datos (MySQL, PostgreSQL). Permiten una gestión adaptable a las necesidades del usuario, realizan búsquedas por múltiples criterios, incluyendo tipo, fechas, categorías o incluso parámetros o metadatos relacionados con los materiales audiovisuales.

1.3.2 Situación problemática.

Actualmente en el departamento de Señales Digitales, como se explica anteriormente, existen diversas aplicaciones donde se manipulan contenidos audiovisuales. Estas aplicaciones tienen la peculiaridad de que pueden ser accedidas por cierta cantidad de usuarios. La plataforma VideoWeb no está exenta de lo antes mencionado debido a que maneja una gran cantidad de contenido audiovisual que pueden ser accedidos por un gran número de usuarios.

Capítulo 1 Fundamentación teórica de la investigación

Actualmente el volumen de información que se maneja en la plataforma VideoWeb crece a gran escala, por lo que se requiere de un sistema capaz de manejar el alto volumen de información apoyado en un hardware con los requerimientos necesarios para el manejo de las peticiones concurrentes de los usuarios.

Cuando muchos usuarios acceden de manera concurrente a una aplicación, está necesita ser capaz de procesar todas las peticiones y dar respuestas en tiempos aceptables para el usuario final. Un sistema como el que se describió anteriormente está limitado por los recursos de hardware, lo que trae consigo un fuerte problema, ya que con el paso del tiempo el acceso de los usuarios de manera concurrente asciende grandemente y los recursos son insuficientes, como solución se puede aumentar la capacidad de procesamiento con nuevos componentes de hardware o adquirir una nueva máquina con mejores condiciones, pero esto tiene un alto valor que incurre como un costo adicional al del software y los propios del mantenimiento del mismo.

Existen otras formas de mejorar el rendimiento de los servidores, entre ellas se encuentra la implementación de un clúster para base de datos de alta disponibilidad, constituyendo una solución económicamente más rentable que el uso de una supercomputadora y brinda las mismas funcionalidades con una mayor flexibilidad. Logrando así la gestión, manipulación y la alta disponibilidad de los datos utilizados en la plataforma VideoWeb.

1.4 Definición de tecnologías de desarrollo

1.4.1 Metodología de desarrollo de software

Las metodologías de desarrollo de software surgen ante la necesidad de utilizar una serie de procedimientos, técnicas, herramientas y soporte documental a la hora de desarrollar un software.

Una metodología de desarrollo de software no es más que un marco de trabajo usado para estructurar, planificar y controlar el proceso de desarrollo en sistemas de información (31).

Dichas metodologías pretenden guiar a los desarrolladores al crear un nuevo software, pero los requisitos de un software a otro son muy variados y cambiantes, dando lugar a que exista una gran variedad de metodologías para la creación de software.

RUP

El Proceso Unificado de Software (*Rational Unified Process* en inglés, y sus siglas RUP) es un conjunto de metodologías adaptables al contexto y necesidades de cada organización, donde el software es organizado

Capítulo 1 Fundamentación teórica de la investigación

como una colección de unidades atómicas llamados objetos, constituidos por datos y funciones, que interactúan entre sí (32).

A continuación se mencionan sus principales características (33):

- Forma disciplinada de asignar tareas y responsabilidades (quién hace qué, cuándo y cómo).
- Es iterativo e incremental.
- Uso de arquitectura basado en componentes.
- Administración de requisitos.
- Busca detectar defectos en las fases iniciales.
- Intenta reducir al número de cambios tanto como sea posible.
- Realiza el análisis y diseño, tan completo como sea posible.

Para realizar un correcto diseño de la base de datos es preciso apoyarse en una metodología de desarrollo, por lo cual se escoge a RUP. RUP describe que para lograr un correcto diseño de una base de datos es preciso desarrollar una serie de artefactos y actividades dentro de los que se encuentra, identificar las clases persistentes, crear claves primarias y restricciones de integridad, realizar los diagramas lógico y físicos y por último normalizar el diseño obtenido.

1.4.2 Lenguaje de modelado

El lenguaje de modelado es la notación (principalmente gráfica) que usan los métodos para expresar un diseño. El proceso indica los pasos que se deben seguir para llegar a un diseño (34). Los lenguajes de modelado, son modelos que permiten la representación del comportamiento y la estructura de los sistemas del mundo real.

UML

El Lenguaje de Modelado Unificado (UML - *Unified Modeling Language*) es un lenguaje que se utiliza para especificar, visualizar, construir y documentar los artefactos que modelan un sistemas de software (35). UML es gratuito, accesible a todos y conforma la colección de las mejores técnicas de ingeniería que han probado ser un éxito en la modelación de sistemas grandes y complejos.

Para la realización del presente trabajo se utilizó como lenguaje de modelado UML en su versión 2.0, debido a que cuenta con un módulo modelador de esquemas de bases de datos relacionales. Este módulo permite la edición gráfica de los esquemas de bases de datos relacionales, con soporte a los niveles lógico y físico, generación de código DDL (*Lenguaje de Definición de Datos*) e ingeniería inversa a partir de esquemas

Capítulo 1 Fundamentación teórica de la investigación

Postgresql 8.X, MySql 5, Oracle 8i y Oracle 10, además permite modelar y gestionar bases de datos visualmente (36).

1.4.3 Herramienta CASE

Las herramientas CASE (*Computer Aided Software Engineering*) proponen una nueva filosofía del concepto de ciclo de vida basado en la automatización, para lo cual proporcionan un conjunto de herramientas bien integradas, que enmarcadas dentro de una determinada metodología, permiten automatizar las fases del ciclo de vida de un sistema de software. En general las herramientas CASE son un conjunto de métodos, utilidades y técnicas que facilitan la automatización del ciclo de vida del desarrollo de sistemas de información (37).

A continuación se mencionan algunas características de las herramientas CASE:

- Verificar el uso de todos los elementos en el sistema diseñado.
- Automatizar el dibujo de diagramas.
- Ayudar en la documentación del sistema.
- Ayudar en la creación de relaciones en la Base de Datos.
- Generar estructuras de código.
- Proporcionar topologías de aplicación flexibles.
- Control de Versión.
- Crear código compilado en el servidor.
- Soporte multiusuario.
- Seguridad.

Visual Paradigm

Constituye una herramienta CASE que utiliza UML como lenguaje de modelado que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientado a objetos, construcción, pruebas y despliegue, permite la realización de la ingeniería tanto directa como inversa. Puede crear todo tipo de diagrama de clases, genera la documentación correspondiente de cada diagrama y permite además realizar el diseño de bases de datos (38).

Por las múltiples ventajas con que cuenta, que es muy fácil de instalar y actualizar, que cuenta con una comunidad que brinda soporte técnico, y que presenta una versión limitada con licencia libre, además de que

Capítulo 1 Fundamentación teórica de la investigación

la UCI adquirió una licencia comercial para la versión 5.0 profesional, se selecciona Visual Paradigm for UML 8.0 debido que incluye nuevas características como son (39) :

- Confirma los cambios parciales al servidor.
- Dibuja diagramas de datos.
- Protege con contraseña los diagramas.
- Examina los diagramas dependientes del proyecto.
- Genera bases de datos con datos de ejemplos.

1.4.4 Sistema Gestor de base de Datos.

PostgreSQL

Como sistema gestor de bases de datos se seleccionó PostgreSQL debido a que es considerado uno de los gestores de bases de datos de código abierto más avanzado del mundo, ya que cuenta con características que solo se podían encontrar en bases de datos comerciales tales como Oracle. Algunas de estas características son que cuenta con soporte para transacciones, estabilidad y escalabilidad. Además ha sido utilizado en la versión 1.0 de la Plataforma VideoWeb con buenos resultados, así como en muchos otros productos desarrollados dentro del departamento de Señales Digitales, esto unido a sus potencialidades para utilizarlo en varios sistemas operativos, su licencia libre y su robustez lo convierten en el SGBD factible para una aplicación que requiera manipular información persistente en entornos de software libre.

1.4.5 Herramienta para balancear las cargas en un clúster de alta disponibilidad

Los clúster que se implementan para obtener una alta disponibilidad usan varias tecnologías para ganar un nivel extra de fiabilidad. A continuación se describe alguna de las herramientas que se utilizan para balancear las peticiones de los clientes hacia los distintos nodos de bases de datos.

CyberCluster

Herramienta multimaster que presenta una replicación sincrónica basada en PostgreSQL 8.2. Proporciona balanceo de carga, asegurándose de que todos los nodos de la base de datos se puedan utilizar durante su funcionamiento. CyberCluster en su arquitectura presenta un balanceador de carga, un servidor de base de datos, y un servidor de replicación (40).

CyberCluster contiene su propio equilibrador de carga que puede ser utilizado para distribuir la carga dentro del clúster. La carga en el sistema está determinada por el número de consultas activas. La máquina con el menor número de consultas activas es elegida para realizar una nueva solicitud. Si el balanceador de carga

Capítulo 1 Fundamentación teórica de la investigación

detecta un problema en un nodo de la base de datos activa, automáticamente separa la base de datos del mismo.

Pero sin embargo esta herramienta es lenta a la hora de distribuir las peticiones y requiere de una versión modificada del servidor. Otro problema que presenta es que cuando se replican objetos grandes deben ser colocados en un directorio que pueda ser leído por todos los nodos de la BD.

Pgpool-II

Pgpool-II es un software mediador que trabaja entre servidores y clientes de base de datos PostgreSQL. Pgpool-II funciona sobre Linux, Solaris, FreeBSD y la mayoría de las arquitecturas UNIX (Para Windows no está soportado). Las versiones de PostgreSQL soportadas son de la 6.4 en adelante. Para usar la paralelización de consultas es necesario la versión 7.4 o superior.

Pgpool-II proporciona las siguientes características (22):

- Limita el excedente de conexiones: PostgreSQL soporta un cierto número de conexiones concurrentes y rechaza las que superen dicha cifra a no ser que el límite máximo de conexiones sea aumentado, lo que provoca un incremento en el consumo de recursos y afecta al rendimiento del sistema. Por su parte Pgpool también tiene un límite máximo de conexiones pero resuelve el problema manteniendo las conexiones extras en una cola de espera en lugar de devolver un error.
- Pool de conexiones: Pgpool mantiene abiertas las conexiones a los servidores PostgreSQL y las reutiliza siempre que se solicita una nueva conexión con las mismas propiedades (nombre de usuario, base de datos y versión del protocolo). Ello reduce la sobrecarga en las conexiones y mejora la productividad global del sistema.
- Replicación: Pgpool a través de esta función permite crear una copia en dos o más discos físicos, de modo que el servicio puede continuar sin parar los servidores en caso de fallo en algún disco.
- Balanceo de carga: Pgpool se aprovecha de la característica de replicación para reducir la carga en cada uno de los servidores PostgreSQL distribuyendo las consultas SELECT entre los múltiples servidores, mejorando así la productividad global del sistema. En el mejor caso, el rendimiento mejora proporcionalmente al número de servidores PostgreSQL. El balanceo de carga funciona mejor en la situación en la cual hay muchos usuarios ejecutando muchas consultas al mismo tiempo.
- Paralelización de consultas: Al usar la función de paralelización de consultas, los datos pueden dividirse entre varios servidores, de modo que la consulta puede ejecutarse en todos los servidores de manera

Capítulo 1 Fundamentación teórica de la investigación

concurrente para reducir el tiempo total de ejecución. La paralelización de consultas es una solución adecuada para búsquedas de datos a gran escala.

Pgpool-II se escoge como herramienta a utilizar para balancear las cargas en un clúster de alta disponibilidad, ya que es una de las mejores y más completas herramientas para brindar soluciones de alta disponibilidad y balanceo de carga en el manejador de bases de datos libres PostgreSQL y es además una herramienta de software libre.

1.4.6 Herramienta para la alta disponibilidad en el clúster

En la actualidad las organizaciones dependen cada vez más de sus sistemas de información, por lo que se desea que estos sean seguros y permanezcan disponibles el mayor tiempo posible.

El proyecto Linux-HA (Alta Disponibilidad) tiene como objetivo proporcionar una solución de alta disponibilidad para Linux que promueva la fiabilidad y disponibilidad de sistemas a través de su comunidad de desarrolladores.

Linux-HA (Alta Disponibilidad) se utiliza ampliamente y como una parte muy importante, en muchas soluciones de Alta Disponibilidad. Desde que comenzó en el año 1999 a la actualidad, sigue siendo una de las mejores soluciones de software HA para muchas plataformas (25).

A continuación se describe alguna de las herramientas que se utilizan para lograr la alta disponibilidad en diversos sistemas.

HAProxy

es una solución libre que ofrece servicios de alta disponibilidad, balanceo de carga y proxy para aplicaciones basadas en el protocolo HTTP. Se diseñó originalmente para servir de proxy a aplicaciones web que requieren persistencia a nivel de capa de aplicación del modelo OSI. En versiones posteriores se le añadió la facilidad de balanceo de carga a nivel de capa aplicación. Actualmente HAProxy no soporta SSL (*Secure Socket Layer*) (41).

Heartbeat

Heartbeat es un software fácilmente portable, corre en todos los Linux conocidos, así como en FreeBSD y Solaris. Heartbeat es una de las implementaciones principales del estándar Open Clúster Framework (OCF). Puede llevar a cabo la detección de la caída de nodos, las comunicaciones y la gestión del clúster en un solo proceso. Actualmente soporta un modelo de dependencias muy sofisticado para clústeres de N nodos, y es muy útil y estable (22).

Capítulo 1 Fundamentación teórica de la investigación

Permite implementar clústeres de control descentralizado, es estable, flexible y eficiente, requiere de dos máquinas como mínimo para su implementación, trabaja enviando latidos (ping) verificando que el servidor principal esté activo, estos envíos constantes de ping requieren de una respuesta por parte del servidor, cuando transcurre un tiempo y el servidor principal no responde, Heartbeat determina que el mismo está fuera de servicio o está inactivo, automáticamente este activa el servidor secundario de forma que asuma todas las peticiones del cliente sin afectar la aplicación.

Debido a que la herramienta Heartbeat permite implementar clústeres de control descentralizado, la misma ha demostrado ser muy estable, flexible y eficiente, además de que permite configurar clústeres de 2 o más nodos, se escoge como herramienta a utilizar para lograr la alta disponibilidad del clúster.

1.5 Pruebas para medir el rendimiento del clúster y comprobar su funcionamiento

Para medir el rendimiento de un clúster así como comprobar su correcto funcionamiento, se realizan un grupo de pruebas definidas como:

1.5.1 Pruebas de configuración

Las pruebas de configuración se realizan con el objetivo de comprobar el correcto funcionamiento del clúster sobre diferentes configuraciones de hardware y/o software. Durante su realización tiende a ocurrir un ciclo de perfeccionamiento en el diseño de la solución. Las pruebas de configuración constituyen la base para validar una nueva arquitectura de despliegue.

1.5.2 Pruebas de rendimiento

Este tipo de pruebas se realiza para determinar o validar la velocidad, escalabilidad y la estabilidad de las aplicaciones bajo prueba. El rendimiento se refiere principalmente a la información relativa a los tiempos de respuestas y la utilización de recursos.

1.5.3 Pruebas de carga

Este tipo de prueba es una subcategoría de las pruebas de rendimiento, que tienen como objetivo comprender el comportamiento de una aplicación ante una carga determinada. La carga puede estar determinada por la conexión de un número de usuarios conectados de manera concurrente al sistema en un tiempo determinado.

1.5.4 Pruebas de benchmark

Este tipo de prueba tiene el objetivo de comparar el comportamiento de los sistemas o las aplicaciones sobre diferentes configuraciones de software y hardware. La realización de este tipo de prueba consiste en medir el

Capítulo 1 Fundamentación teórica de la investigación

rendimiento de las aplicaciones sobre determinada configuración de software o hardware y comparar los resultados obtenidos con configuraciones similares.

1.6 Conclusiones Parciales

Durante el desarrollo del presente capítulo se han abordado conceptos y definiciones que son importantes para el proceso de diseño de la base de datos de alta disponibilidad para la Plataforma VideoWeb 2.0. Se realizó un estudio acerca de los sistemas de bases de datos y la tecnología clúster, definiendo así las diferentes herramientas a utilizar para lograr el diseño de la bases de datos de alta disponibilidad para la plataforma.

Durante el transcurso del capítulo se pudo comprender que la necesidad fundamental de crear un clúster de alta disponibilidad para la base de datos de materiales multimedia es que debido a que la Plataforma VideoWeb se dedica a proveer servicios de transmisión de materiales audiovisuales, necesita obtener un sistema tolerante a fallos, evitando que fallas tanto del hardware como del software afecten el servicio que se brinda al usuario.

Capítulo 2

Descripción y análisis de la propuesta de solución.

2.1 Introducción

El presente capítulo está centrado en el diseño de una solución genérica para la base de datos, así como una propuesta de diseño para el clúster que brinda la alta disponibilidad de los datos de la Plataforma VideoWeb 2.0, además se describen los patrones de diseño utilizados en el diseño de la base de datos.

2.2 Requisitos funcionales y no funcionales

Los requisitos se han convertido en un punto clave en el desarrollo de aplicaciones informáticas. Un gran número de proyectos de software actualmente fracasan debido a una mala definición, especificación y administración de requisitos.

Roger S. Pressman expresa que para que un esfuerzo de desarrollo de software tenga éxito, es esencial comprender perfectamente los requisitos del software. Independientemente de lo bien diseñado o codificado que esté un programa, si se ha analizado y especificado pobremente, decepcionará al usuario y desprestigiará al que lo ha desarrollado. La parte más difícil en la construcción de sistemas software es decidir precisamente qué construir. Ninguna otra parte del trabajo conceptual es tan ardua como establecer los requisitos técnicos detallados, incluyendo todas las interfaces con humanos, máquinas y otros sistemas. Ninguna otra parte del trabajo puede perjudicar tanto el resultado final si se realiza de forma errónea. Ninguna otra parte es tan difícil de rectificar posteriormente (42).

A continuación se listan los principales requisitos funcionales y no funcionales de la Plataforma VideoWeb 2.0 que la base de datos debe ser capaz de gestionar.

Requisitos funcionales

Los requisitos funcionales son capacidades o condiciones que debe cumplir o tener el sistema. A continuación se presentan los requisitos funcionales principales del sistema:

RF1: Adicionar tipología: el sistema debe permitir adicionar una nueva tipología de archivo multimedia.

RF2: Editar tipología: el sistema debe permitir editar los datos de una tipología ya existente.

RF3: Eliminar tipología: el sistema debe permitir eliminar una tipología de archivo multimedia.

Capítulo 2 Descripción y análisis de la solución propuesta

RF4: Adicionar campos a una tipología existente: el sistema debe permitir adicionar campos a las tipologías existentes.

RF5: Adicionar archivos multimedia: el sistema debe permitir almacenar los archivos multimedia en el servidor.

RF6: Editar datos asociados a los archivos multimedia: el sistema debe permitir modificar los datos asociados a los archivos multimedia.

RF7: Eliminar archivos multimedia: el sistema debe permitir eliminar los archivos multimedia con los datos asociados a estos.

RF8: Crear publicación de archivo multimedia: el sistema debe permitir crear publicaciones multimedia a partir de los archivos multimedia previamente almacenados.

RF9: Editar publicación de archivo multimedia: el sistema debe permitir modificar los datos de la publicación multimedia.

RF10: Activar publicación de los archivos multimedia: el sistema debe permitir activar las publicaciones multimedia previamente elaboradas, creando un enlace simbólico al archivo en el punto de publicación.

RF11: Terminar publicación de los archivos multimedia: el sistema debe permitir desactivar las publicaciones multimedia previamente publicadas, eliminando el enlace simbólico del punto de publicación.

RF12: Realizar búsqueda de publicaciones de archivos multimedia: el sistema debe permitir realizar búsquedas de publicaciones de archivos multimedia.

RF13: Eliminar publicación de archivos multimedia: el sistema debe permitir eliminar las publicaciones multimedia.

RF14: Realizar búsqueda básica de materiales audiovisuales a solicitar: el sistema debe permitir realizar búsquedas básicas de los materiales que se encuentran almacenados en el servidor.

RF16: Realizar búsqueda avanzada de materiales audiovisuales a solicitar: el sistema debe permitir realizar búsquedas definiendo filtros avanzados relativos a la fecha o a la descripción del material, lo que posibilitará la búsqueda más específica de materiales atendiendo a las fechas almacenadas y a las descripciones asociadas a los mismos.

RF17: Insertar solicitud: el sistema debe permitir adicionar solicitudes de materiales audiovisuales.

RF18: Editar solicitud: el sistema debe permitir modificar los datos asociados a las solicitudes de los materiales audiovisuales que se han realizado.

Capítulo 2 Descripción y análisis de la solución propuesta

RF19: Eliminar solicitud: el sistema debe permitir eliminar los datos de algunas solicitudes de materiales audiovisuales que se han realizado.

RF20: Buscar solicitud: el sistema debe permitir acceder a los datos de las solicitudes realizadas por los usuarios.

Requisitos no funcionales

Los requisitos no funcionales son cualidades o propiedades que debe cumplir el sistema, estos ni describen información a guardar, ni funciones a realizar, a través de los mismos se debe obtener un producto atractivo, usable, rápido y confiable. A continuación se presentan los requisitos no funcionales del sistema para la gestión y almacenamiento de datos:

Confiabilidad.

- El SGBD escogido debe presentar facilidades de administración de roles y usuarios restringiendo el acceso a los datos.

Rendimiento.

- Al realizar cualquier gestión de la información en la base de datos, ya sea mediante acciones de inserción, búsqueda, eliminación o modificación de datos, el tiempo de respuesta a las peticiones del usuario debe ser aceptable.

Eficiencia.

- El sistema debe estar disponible las 24 horas.
- El servidor de base de datos debe mantener buenas prestaciones ante la demanda concurrente de peticiones.

Software.

- El SGBD a utilizar debe ser: PostgreSQL 9.1.
- Herramienta a utilizar para balancear las cargas: Pgpool-II 3.1.1.
- Herramienta a utilizar para brindar la alta disponibilidad: Heartbeat 3.0.5.

Hardware.

Los requerimientos mínimos para que el servidor de base de datos funcione correctamente son:

- Arquitectura 64 bits (x64).

Capítulo 2 Descripción y análisis de la solución propuesta

- Procesador: 1 QuadCore.
- Memoria RAM: 4 GB.

2.3 Diseño de la Base de Datos

El diseño propuesto para la bases de datos de la plataforma VideoWeb 2.0 contiene 139 tablas, dentro de las que se encuentra las tablas generadas por el CMS Drupal 7, siendo utilizadas por los módulos, administración, configuración, gestión de contenido, catalogación, presentación de contenido y recuperación y préstamo, haciendo uso principal de las tablas, archivo_multimedia, almacenamiento, almacen_local, streaming, publicacion, tipologia, tipologia_data, solicitud, naturaleza, cliente, empresa_externa, users, role, role_permission y credito. Los modelos lógicos y físicos del diseño de la base de datos obtenida se pueden encontrar en la documentación del proyecto.

2.3.1 Nomenclatura de la base de datos

En el diseño de la base de datos se utilizó una nomenclatura sencilla que permite una mejor comprensión y organización de los elementos que componen el modelo. Las nomenclaturas utilizadas son:

- Se determina que el nombre de la base de datos se escriba con letras minúsculas y sin acentos. Ejemplo videoweb2.0.
- Los nombres de las tablas se escriben igualmente con letras minúsculas, sin acentos y en caso de ser un nombre compuesto se separan por “_”. Ejemplo almacen_local.
- Los nombres de los campos se escriben igualmente con letras minúsculas, sin acentos y las palabras separadas por “_”. Ejemplo punto_almacenamiento.
- Los identificadores de las tablas deben contener el vocablo id. Ejemplo id_solicitud.
- Los identificadores de llave foránea debe estar constituido por id y el nombre de la tabla de la cual se recibe la llave. Ejemplo: almacenamiento_id.
- Los nombres de las tablas que se crean debido a las relaciones de mucho a mucho, estarán conformados por el nombre de las tablas que se relacionan separados por “_”. Ejemplo: solicitud_cliente.

2.3.2 Patrones de diseño

En el diseño de bases de datos y en la modelación de datos en general se presentan elementos repetitivos en disímiles modelos, los que correctamente identificados suelen pasar a patrones de diseño. Un patrón de diseño es una solución a un problema determinado que surge durante el proceso de desarrollo de software, el cual está probado y documentado, además se define como una solución común a un problema común de un determinado contexto (43).

Capítulo 2 Descripción y análisis de la solución propuesta

Durante el diseño de la base de datos para la plataforma VideoWeb 2.0 se utilizaron diferentes patrones de diseño como los patrones de asociación y el patrón de control de acceso basado en roles.

Los patrones de asociación representan las relaciones entre entidades del modelo, estos fueron utilizados para representar asociaciones de tipo muchos a muchos. Las bases de datos relacionales no soportan relaciones directas del tipo muchos a muchos. Este problema puede ser resuelto añadiendo al modelo una tabla de asociación cuya funcionalidad es representar este tipo de relación entre varias entidades. La tabla resultante estará compuesta por las llaves primarias de las entidades a relacionar, las que a su vez formarán parte de la clave primaria y compuesta de la entidad resultante de la relación. En el modelo de base de datos que se propone, este patrón se pone de manifiesto a la hora de relacionar la tabla solicitud con la tabla cliente, pues una solicitud puede tener uno a varios clientes mientras que un cliente puede realizar una o varias solicitudes.

Para mantener la seguridad de la base de datos de la plataforma VideoWeb 2.0 se utilizó el patrón control de acceso basado en roles (RBAC). A través de este modelo, los permisos para ejecutar ciertas operaciones son asignados a roles específicos y estos roles son asignados a los usuarios del sistema. El manejo de los permisos de cada usuario está determinado por la asignación del rol apropiado, ya que a los usuarios no se les asignan permisos directamente, sino que los adquiere a través de los roles que le son asignados. Este patrón se pone de manifiesto entre las entidades usuario y roles donde un usuario puede poseer un conjunto de roles y las funcionalidades que puede realizar dicho usuario en el sistema son asignadas al rol en específico.

2.3.3 Descripción de las tablas

A continuación se describen de forma general las tablas principales de la base de datos de la Plataforma VideoWeb 2.0, las restantes descripciones de las tablas se encuentran en el anexo 1.

Nombre: <code>archivo_multimedia</code>	
Descripción: Tabla que guarda los datos asociados a los materiales audiovisuales.	
Atributo	Tipo
<code>id_am</code>	Integer
<code>title</code>	Varchar
<code>created</code>	Integer
<code>changed</code>	Integer
<code>u_id</code>	Bigint
<code>nombre_fisico</code>	Varchar

Capítulo 2 Descripción y análisis de la solución propuesta

duracion	Integer
punto_almacenamiento_id	Integer
disponible	Boolean
cataloged	Integer

Tabla 1: archivo_multimedia.

Nombre: almacenamiento	
Descripción: Tabla que guarda los datos asociados a los tipos de almacenamientos de los materiales audiovisuales.	
Atributo	Tipo
id	Integer
nombre	Varchar
tipo	Varchar
descripcion	Varchar

Tabla 2: almacenamiento.

Nombre: almacen_local	
Descripción: Tabla que guarda los datos de un tipo de almacenamiento.	
Atributo	Tipo
id	Integer
almacen_id	Bigint
ruta	Text

Tabla 3: almacen_local.

Nombre: tipologia_data	
Descripción: Tabla que guarda los datos asociados a las tipologías.	
Atributo	Tipo
id_td	Integer
title	Varchar
type	Varchar

Capítulo 2 Descripción y análisis de la solución propuesta

tiempo_inicio	Integer
tiempo_fin	Integer
restriccion	Varchar
descripcion	Varchar
am_id	Integer
palabras_claves	Varchar

Tabla 4: tipologia_data.

Nombre: publicacion_am	
Descripción: Tabla que guarda los datos de las publicaciones realizadas.	
Atributo	Tipo
id	Integer
nodo_id	Integer
streaming_id	Integer
am_id	Integer
costo	Float

Tabla 5: publicacion.

Nombre: solicitud	
Descripción: Tabla que guarda los datos asociados a las solicitudes realizadas por los usuarios.	
Atributo	Tipo
id_solicitud	Integer
tiempo_inicio	Bigint
tiempo_fin	Bigint
fecha	Varchar
formato	Varchar
estado	Varchar
usuario_id	Bigint
material_id	Bigint
naturaleza_id	Bigint

Tabla 6: solicitud.

Capítulo 2 Descripción y análisis de la solución propuesta

Nombre: naturaleza	
Descripción: Tabla que guarda los tipos de solicitudes que puede realizar un cliente.	
Atributo	Tipo
id_naturaleza	Bigint
tipo_naturaleza	Varchar

Tabla 7: naturaleza.

Nombre: empresa_externa	
Descripción: Tabla que guarda los datos asociados a la institución a la cual pertenece el cliente.	
Atributo	Tipo
id_empresa	Integer
nombre	Varchar

Tabla 8: empresa_externa.

Nombre: cliente	
Descripción: Tabla que guarda los datos de los clientes que realizan una solicitud.	
Atributo	Tipo
id_cliente	Integer
ci	Bigint
nombre	Varchar
apellidos	Varchar
empresa_id	Bigint

Tabla 9: cliente.

Nombre: role	
Descripción: Tabla que guarda los datos asociados a los roles.	
Atributo	Tipo
id_r	Integer
name	Varchar
weigth	Integer

Tabla 10. role.

Capítulo 2 Descripción y análisis de la solución propuesta

Nombre: role_permission	
Descripción: Tabla que guarda los datos asociados a los permisos que pueden tener los roles.	
Atributo	Tipo
id_permission	Integer
r_id	Integer
module	Varchar

Tabla 11: role_permission.

Nombre: streaming	
Descripción: Tabla que guarda los datos asociados a un tipo de almacenamiento.	
Atributo	Tipo
id_s	Integer
almacenamiento_id	Bigint
puerto	Bigint
protocolo	Varchar
ip	Varchar

Tabla 12: streaming.

Nombre: credito	
Descripción: Tabla que guarda los datos asociados al presupuesto con que cuentan los clientes.	
Atributo	Tipo
id_c	Integer
users_id	Integer
saldo	Integer

Tabla 13: credito.

2.4 Propuesta de diseño de la solución de clúster

Se confeccionó una propuesta de solución para la puesta en marcha de un clúster de servidores de bases de datos. Los nodos utilizados para conformar el clúster pueden ser computadoras personales con disímiles características de hardware.

A continuación se describen los diferentes tipos de diseños realizados.

Capítulo 2 Descripción y análisis de la solución propuesta

2.4.1 Diseño lógico

Para la implementación de un clúster de alta disponibilidad, se propone que el mismo sea implementado sobre el sistema operativo Ubuntu 12.04, utilizando como sistema de gestión de bases de datos PostgreSQL en su versión 9.1, además utilizar la herramienta Pgpool-II con la versión 3.1.1 para el balanceo de carga y Heartbeat con la versión 3.0.5 como herramienta para implementar la alta disponibilidad.

El clúster está compuesto por la siguiente arquitectura:

- Un nodo Pgpool-II primario.
- Un nodo Pgpool-II secundario de respaldo.
- Dos nodos de datos PostgreSQL.

Durante el funcionamiento normal del clúster todas las sentencias SQL que son enviadas por el cliente hacia el clúster pasan por el nodo Pgpool, en el cual son analizadas y repartidas entre cada uno de los nodos postgresql, disminuyendo así la carga entre los mismos.

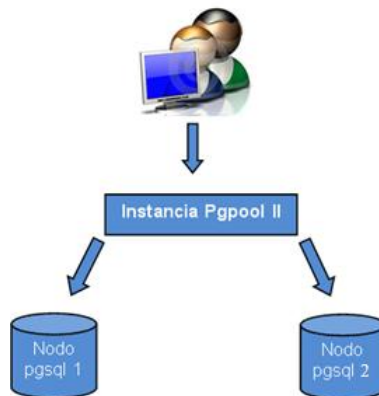


Figura 2. Arquitectura del clúster

En la arquitectura presentada si el nodo Pgpool primario queda fuera de servicio, el clúster en general quedará fuera de servicio también, convirtiéndose en un punto de fallas en el nodo primario. Para evitar este tipo de falla, es preciso configurar un nodo de respaldo, implementando así la alta disponibilidad. En el caso de caída del nodo primario, el de respaldo tomaría el control, siendo este proceso transparente al cliente.

Capítulo 2 Descripción y análisis de la solución propuesta

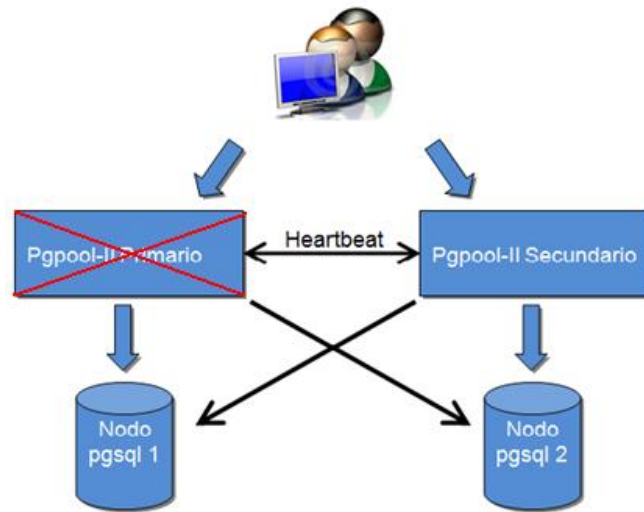


Figura 3. Alta disponibilidad en el clúster

Una vez restablecido el nodo primario, toma nuevamente el control, quedando el nodo de respaldo como reserva, volviendo a su normalidad el clúster. Pero si uno de los nodos pgsq es detenido, todas las consultas que se le son enviadas las atenderá el otro nodo, hasta que el mismo sea restablecido. Obteniéndose el siguiente modelo lógico.

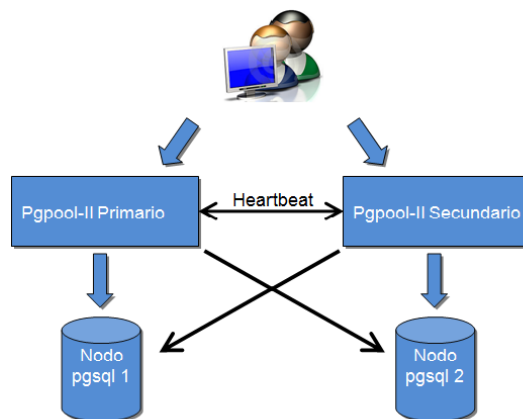


Figura 4. Representación lógica del clúster.

2.4.2 Diseño físico

La arquitectura propuesta para el clúster está compuesta por dos nodos, encargados de balancear las peticiones entre los servidores, por lo que es necesario instalar la herramienta Pgpool-II-3.1.1, encargada de proporcionar el balanceo de carga e implementar un mecanismo de recuperación ante fallas.

Capítulo 2 Descripción y análisis de la solución propuesta

El diseño físico de la solución en el caso de que el nodo Pgpool-II se encuentre funcionando normalmente sería:

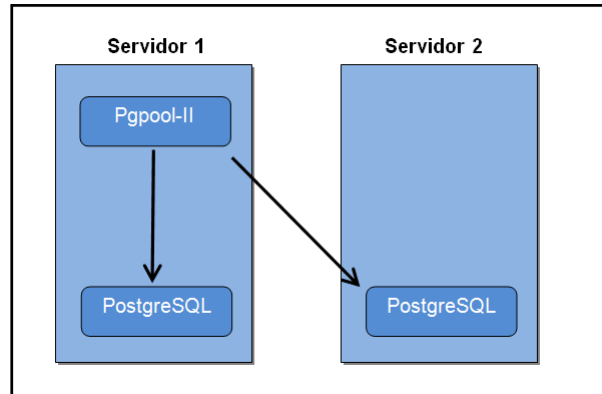


Figura 5. Representación del clúster cuando funciona normalmente.

La ventaja que ofrece este diseño es que si la instancia del nodo SQL del servidor 1 dejara de funcionar en un momento determinado, esto no afectará en nada a la solución, pues las peticiones son direccionadas hacia el otro nodo SQL (ver Fig. 8).

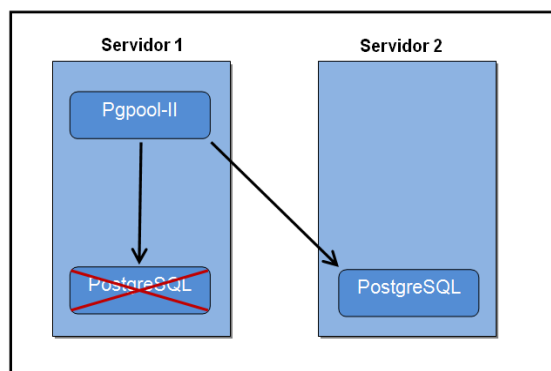


Figura 6. Representación física cuando falla un nodo.

Sin embargo este diseño presenta un punto de fallas, ya que el nodo Pgpool-II puede en un momento determinado dejar de prestar el servicio, es decir dejar de funcionar e impedir la entrada de sentencias SQL provenientes del cliente al servidor. Debido a este fallo, es preciso que el nodo de respaldo cuente con la implementación de alta disponibilidad con Pgpool-II, la cual posibilita que el servidor secundario que contiene el nodo de respaldo, ejecute las acciones del servidor principal y continúa prestando el servicio sin afectar al cliente (ver Fig. 9).

Capítulo 2 Descripción y análisis de la solución propuesta

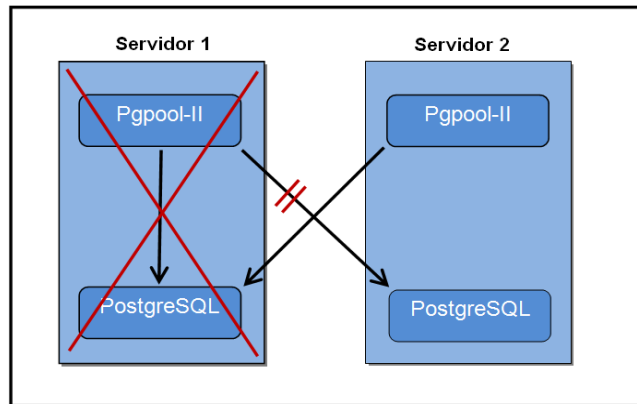


Figura 7. Representación física cuando falla el balanceador primario.

Pgpool-II ofrece la posibilidad de continuar brindando el servicio tras el fallo de un servidor de PostgreSQL en el clúster, para lo cual utiliza la herramienta Heartbeat, que permite detectar la caída de un nodo. Obteniéndose el siguiente modelo físico:

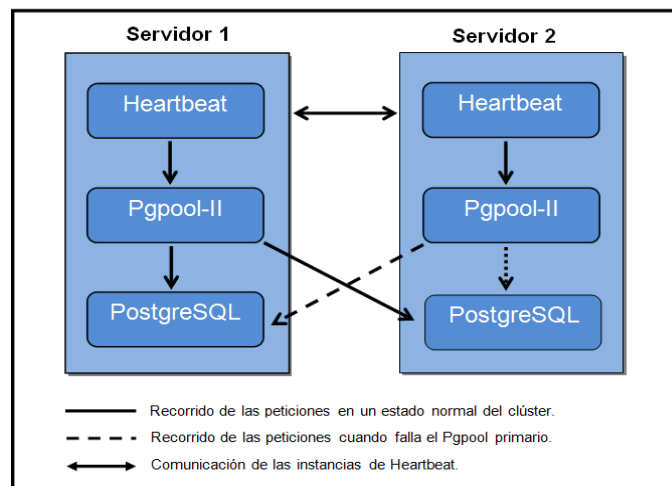


Figura 8. Diseño general del clúster

2.5 Configuraciones

Para lograr el correcto funcionamiento del diseño obtenido es preciso realizar varias configuraciones en las herramientas utilizadas para lograr la alta disponibilidad en los servidores de bases de datos, en el anexo 2 se puede observar detalladamente los cambios ocurridos en los archivos de configuración de las herramientas PostgreSQL, Pgpool-II y Heartbeat.

Capítulo 2 Descripción y análisis de la solución propuesta

2.5.1 Configuraciones realizadas en PostgreSQL

A continuación se describen los ficheros que se utilizaron en las configuraciones realizadas en Postgresql:

pg_hba.conf: Es un fichero que se utiliza para definir cómo y de dónde un usuario puede utilizar el clúster de base de datos. El formato general del fichero *pg_hba.conf* es un conjunto de registros, uno por línea. Cada registro especifica un tipo de conexión, rango de direcciones IP de cliente (si es relevante para el tipo de conexión), un nombre de base de datos, un nombre de usuario, y el método de autenticación que se utiliza para las conexiones que coincidan con estos parámetros.

postgresql.conf: El archivo *postgresql.conf* se utiliza para establecer varios parámetros que afectan al clúster de la base de datos, como por ejemplo cómo y a través de qué puerto se realizan las conexiones a PostgreSQL. En este fichero se modificó los parámetros *listen_addresses* para indicar en qué direcciones IP PostgreSQL debería escuchar para detectar aplicaciones cliente, *log_connections* y *log_disconnections*.

2.5.2 Configuraciones realizadas en Pgpool-II

pcp.conf: Pgpool-II proporciona una interfaz de gestión desde la cual el administrador puede recoger el estado de Pgpool-II y finalizar los procesos de Pgpool-II a través de la red. El fichero *pcp.conf* es un fichero de nombres de usuario y contraseñas usado para autenticarse con la interfaz. Al mismo se le agregó como usuario Pgpool2 y para generar la contraseña se utilizó la utilidad *pg_md5*, lo que significa que las contraseñas se cifran en formato hash MD5, MD5 es un algoritmo de reducción criptográfico.

pgpool.conf: a través de este fichero se realizan todas las configuraciones necesarias para que el Pgpool funcione correctamente. Primeramente se configura la directiva *listen_addresses* para indicar que el Pgpool escuche en todas las interfaces, luego se configura para que todos los procesos que forman parte del Pgpool se guarden en el directorio por defecto */var/run/postgresql*, posteriormente se configura para que la comprobación de estado de los servidores PostgreSQL se realice con el usuario *pgpool2* cada 60 segundos y con un tiempo máximo de espera de 20 minutos y por último se dejan todos los límites de conexiones, número de procesos, tiempos de espera y similares a sus valores por defecto. Los cambios que se realicen en este fichero afectarán a todas las bases de datos que se tenga definida en el clúster de bases de datos.

2.5.3 Configuraciones realizadas en el Heartbeat

ha.cf: el fichero *ha.cf* especifica las opciones de configuración de Heartbeat, indicando:

- Dónde se quiere guardar los logs.

Capítulo 2 Descripción y análisis de la solución propuesta

- En *keepalive* indica que los ping se enviarán cada 2 segundos.
- En *deadtime* se especifica que si un nodo no responde en 30 segundos está caído.
- En *wartime* se indica que si no responde en 30 segundos se lanzará una alerta.
- Se usará el puerto UDP 649 para la comunicación entre las máquinas.
- El parámetro *failback* en encendido indica el comportamiento en caso de recuperación de un nodo caído.
- Se indica que nodos debe monitorizar.

haresources: En este archivo se definen los recursos que son gestionados por Heartbeat, en el mismo se le indica que nodo será el primario y que los recursos que debe gestionar serán la dirección IP de servicio y el servicio Pgpool. La secuencia que sigue Heartbeat a la hora de levantar el servicio es la siguiente: primero comprueba que el nodo donde se debe arrancar el servicio está encendido, una vez comprobado, levanta un alias con la dirección del servicio que se le indique y finalmente establece el servicio Pgpool.

authkeys: En este archivo se define el nivel de seguridad con la que se envía la información a los nodos del clúster Heartbeat. Sólo debe ser legible por root y rige por el siguiente formato.

```
auth <num>  
<num> <algorithm> <secret>
```

Figura 9. Formato de authkeys

num: es un índice de clave simple, comenzando con 1.

algorithm: es el algoritmo de firma que se utiliza.

secret: es la clave de autenticación actual.

Una vez terminadas las configuraciones en el Heartbeat es preciso darle permisos a los archivos de configuración, por ejemplo al archivo *authkeys* es obligatorio que sea accesible sólo por el usuario root y que tenga permisos 600, lo que significa que el propietario tiene permisos de lectura y escritura sobre el archivo. Los restantes archivos *haresources* y *ha.cf* ubicados en el directorio */etc/ha.d* solamente tendrán permisos 640, lo que significa que el usuario propietario puede leer y escribir, el grupo puede leer el archivo y otros no pueden hacer nada.

Otro de los archivos que se configuró fue el fichero *hosts*, el cual es un fichero sin extensión que se encuentra en el directorio */etc*, que permite que se le añadan tantas resoluciones de dominios como se desee, para lograr

Capítulo 2 Descripción y análisis de la solución propuesta

que el Heartbeat funcione se le agregó a este fichero las direcciones IP y nombres de dominio de los nodos que conformaran el clúster.

2.6 Políticas de seguridad para la base de datos de la Plataforma VideoWeb 2.0

En la actualidad uno de los puntos más críticos en las base de datos es la seguridad, pues muchos sistemas están expuestos a ataques, y en la mayoría de los casos no cuentan con una infraestructura de protección. La información almacenada en una base de datos está en constante riesgo de sufrir ataques que puedan provocar su modificación o pérdida, por ello es de vital importancia velar la seguridad de la misma, protegiéndola contra accesos no autorizados o cualquier acción que puedan violar la integridad y confidencialidad de los datos. La integridad radica en asegurar que los datos almacenados sean correctos, es decir verídicos. La disponibilidad de los datos requiere que la información esté en disposición de aquellos que deban acceder a ella.

A continuación se enumeran las medidas de seguridad para la base de datos de la Plataforma VideoWeb:

- Si se desea eliminar o modificar un elemento de la base de datos que está siendo utilizado por otro elemento que depende de él, el sistema no debe permitir que este elemento sea eliminado, salvo las excepciones conciliadas previamente.
- Si dos usuarios acceden a un recurso e intentan modificar la información simultáneamente, el sistema solo debe permitir que modifique uno de ellos.
- Debe garantizarse la recuperación de los datos, en caso de algún fallo, a través de copias de respaldo realizadas.
- La información almacenada en la base de datos deberá estar protegida de acceso no autorizado evitando modificaciones no deseadas en los datos.
- Controlar el acceso al servidor de base de datos.

2.7 Conclusiones Parciales

El diseño de la base de datos que se obtuvo responde a las necesidades planteadas, con el mismo se logra reducir la redundancia evitando la existencia de datos repetidos innecesariamente aumentando la integridad de los datos, evitando así que se creen anomalías en las consultas realizadas para extraer e insertar datos en el sistema.

El diseño de la solución del clúster se realizó teniendo en cuenta las características de los posibles entornos en los cuales se va a desplegar dicho clúster. Además con el correcto diseño de la base de datos y la

Capítulo 2 Descripción y análisis de la solución propuesta

configuración e implementación del clúster se pretende adquirir un sistema tolerante a fallas, logrando así que el servicio de transmisión de materiales audiovisuales a través de la red esté disponible las 24 horas.

Capítulo 3

Validación de la Base de Datos de Alta Disponibilidad.

3.1 Introducción

En el presente capítulo se describe el proceso de validación del diseño de la base de datos de alta disponibilidad de la Plataforma VideoWeb 2.0, a través de una validación teórica y funcional. Se abordan aspectos de gran importancia dentro de los que destacan: la integridad y normalización de la base de datos. Además se realizan pruebas de carga al clúster y a la bases de datos para comprobar su correcto funcionamiento.

3.2 Validación teórica del diseño de la base de datos

El diseño de una base de datos es una de las etapas más importantes en el desarrollo del sistema que se desea implementar, por lo que se debe tener en cuenta una serie de aspectos importantes que permiten garantizar un mejor diseño. Estos aspectos son la integridad y la redundancia de los datos, asegurando el acceso y la protección de estos contra situaciones no deseadas.

3.2.1 Integridad de los datos

El concepto de integridad, proviene del término latino *integrítas*, se refiere a la cualidad de íntegro (44). La integridad en una base de datos se refiere a la corrección y exactitud de la información contenida. Una base de datos determinada podría estar sujeta a cualquier cantidad de restricciones de integridad de una complejidad arbitraria (45). Se puede definir la integridad en término de base de datos con el conjunto de seguridades que son utilizadas para mantener los datos correctos.

Al modificar el contenido de la base de datos con sentencias INSERT, DELETE o UPDATE, la integridad de los datos puede verse afectada añadiendo datos no válidos, modificando datos existentes tomando un valor incorrecto o eliminando datos que al hacerlo violan alguna regla (44).

La integridad de los datos es uno de los factores más importantes a la hora de realizar el diseño de una base de datos, se distribuye en los siguientes tipos:

Integridad Referencial

Capítulo 3 Validación de la Base de Datos de Alta Disponibilidad

La integridad referencial es un sistema de reglas que utilizan la mayoría de las bases de datos relacionales para asegurar que los registros de las tablas relacionadas son válidos y que no se borren o cambien datos relacionados de forma accidental produciendo errores de integridad. La integridad referencial garantiza que la relación entre dos tablas permanezca sincronizada durante las operaciones de actualización y eliminación. Un ejemplo de la integridad referencial es en las tablas `role` y `role_permission`, la integridad referencial requiere que estas dos tablas estén sincronizadas. Es decir que el `id` de la tabla `role` por la relación que existe entre dichas tablas se debe encontrar en la tabla `role_permission`. El sistema no puede eliminar la fila `id_role` de la tabla `role` porque la fila `role_id_role` de la tabla `role_permission` se quedaría sin una referencia. Sin embargo, se podría permitir la eliminación de la fila `role_id_role` de la tabla `role_permission` y eliminar también todas las filas de la tabla `role` que tengan la misma identificación, manteniéndose la integridad referencial en ambas tablas.

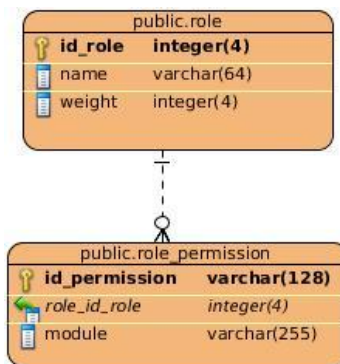


Figura 10. Integridad Referencial.

Integridad de Dominio

La integridad de dominio viene dada por el conjunto de valores posibles para un atributo determinado. Restringe los valores que puede tomar un atributo respecto a su dominio, por ejemplo como se puede observar en la figura 11 en la tabla `naturaleza` el atributo `tipo_naturaleza` solo puede tomar los valores de naturaleza externa o interna.

Capítulo 3 Validación de la Base de Datos de Alta Disponibilidad

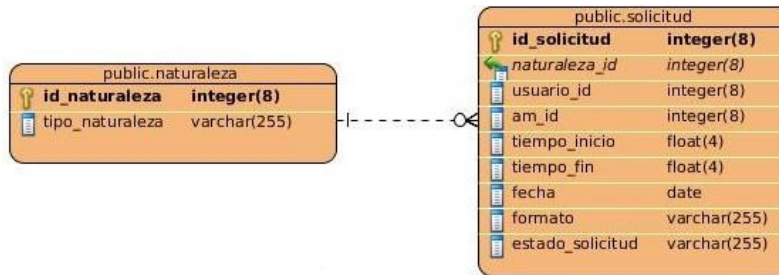


Figura 11. Integridad de Dominio.

Integridad de Entidad

Las restricciones de entidades aseguran la integridad de las entidades que son modeladas por el sistema. Un caso de este tipo de restricción es el de las llaves primarias para cada tabla. En el caso de la base de datos para la Plataforma VideoWeb cada entidad tiene definida su llave primaria, que no puede ser nula ni se puede repetir. Definir correctamente la llave primaria, es de vital importancia, ya que representa la principal herramienta que utiliza el servidor de BD para seleccionar la información que se necesite, evitando la duplicidad de registros y el lanzamiento de excepciones por errores en los registros. Para garantizar la integridad de entidad de los datos de la BD, se definió una llave primaria para cada tabla, las mismas son de tipo entero y no permiten valores nulos (NOT NULL) como se puede observar en la figura 12.



Figura 12. Integridad de Entidad.

3.2.2 Normalización de la base de la base de datos

El concepto de normalización fue introducido por E.D. Codd y fue pensado para aplicarse a sistemas relacionales. Sin embargo, tiene aplicaciones más amplias. La normalización es la expresión formal del modo de realizar un buen diseño. Provee los medios necesarios para describir la estructura lógica de los datos en un sistema de información (46).

Ventajas de la Normalización:

Capítulo 3 Validación de la Base de Datos de Alta Disponibilidad

- Evitar anomalías en la actualización.
- Mejorar la independencia de los datos, permitiendo realizar extensiones de la base de datos afectando muy poco, o nada, a los programas de aplicación existentes que acceden a la misma.
- Mínima redundancia, la información no está duplicada innecesariamente.
- Máximo rendimiento de las aplicaciones, sólo se trata aquella información que va a servir de utilidad a cada aplicación.
- Menos oportunidades para generar incoherencias.
- Facilidad de uso y claridad, los datos están agrupados en tablas que identifican claramente una entidad o relación. Su representación es clara y sencilla de entender incluso para usuarios finales.
- Flexibilidad y facilidad de gestión, la información que necesitan los usuarios se puede obtener de las tablas relacionales o relaciones mediante operaciones del álgebra y cálculo relacionales.
- Precisión, las interrelaciones entre las tablas consiguen mantener información diferente relacionada con toda exactitud.

En la normalización de una base de datos se involucran varias fases que se realizan en orden. La realización de la 2da fase supone que se ha concluido la 1ra y así sucesivamente. Tras completar cada fase se dice que la relación está en: primera forma normal (1ra FN), segunda forma normal (2da FN), tercera forma normal (3ra FN), forma normal de Boyce-Codd (FNBC), cuarta forma normal (4ta FN) y quinta forma normal (5ta FN).

Las relaciones en 1ra FN son un subconjunto del universo de todas las relaciones posibles. Las relaciones en 2da FN son un subconjunto de las que están en 1ra FN y así sucesivamente, como se muestra en la siguiente figura:

Capítulo 3 Validación de la Base de Datos de Alta Disponibilidad

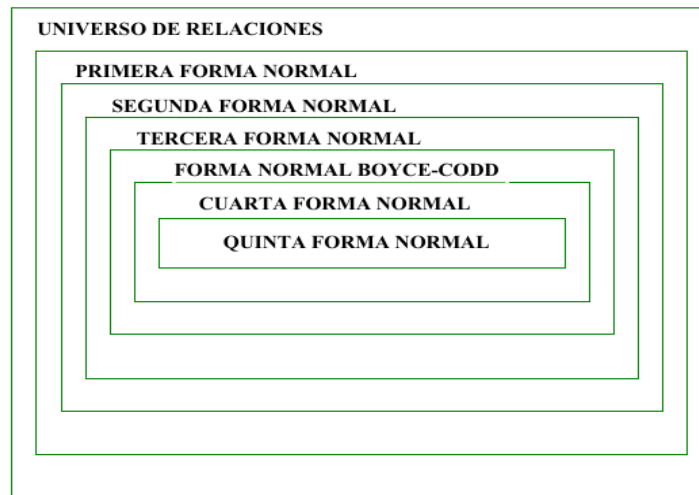


Figura 13. Fases de normalización.

Primera Forma Normal (1ra FN)

Para que un esquema relacional este en 1ra Forma Normal se tiene que asegurar que cada tupla contiene solamente atributos con valores, es decir, no existan campos multievaluados y además no existen campos compuestos ni derivados. Lo que se persigue en realidad al normalizar a este nivel es: eliminar los grupos repetitivos de las tablas individuales, crear una tabla separada por cada grupo de datos relacionados e identificar cada grupo de datos relacionados con una clave primaria.

Segunda Forma Normal (2da FN)

La segunda forma normal plantea que todos los atributos que no forman parte de la llave primaria, deben depender de manera total de los atributos que forman la llave primaria. Se dice que una entidad está en 2da FN si está en 1ra FN y si sus atributos no llaves (ni primarias ni candidatas) son funcional y completamente dependientes de la llave primaria. La 1ra FN asegura que todas las columnas que no son llave sean completamente dependientes de la llave primaria, es decir, establece que todas las dependencias parciales se deben eliminar y separar dentro de sus propias tablas.

Tercera Forma Normal (3ra FN)

Una relación está en tercera forma normal sí, y sólo sí, está en 2da FN y además, cada atributo que no está en la clave primaria no depende transitivamente de la clave primaria. En la 2da forma normal se elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de otras columnas que tampoco son llave. Una tabla está normalizada en la 2da FN si todas las

Capítulo 3 Validación de la Base de Datos de Alta Disponibilidad

columnas que no son llave son funcionalmente dependientes por completo de la llave primaria y no hay dependencias transitivas.

Cuarta Forma Normal (4ta FN)

Para que un esquema se encuentre en 4ta FN debe cumplir que no existan dos o más atributos independientes con dependencia respecto a un conjunto de atributos, formando parte de la llave junto a dicho conjunto, es decir, que no existan dependencia entre atributos llaves.

Quinta Forma Normal (5ta FN)

La 5ta forma normal está diseñada para reducir la redundancia en relaciones que almacenan hechos multievaluados a través del aislamiento de relaciones semánticamente relacionadas. Una relación está en 5ta FN cuando su contenido no puede ser reconstruido a partir de un conjunto de tablas, se excluye la posibilidad de tablas con los mismos atributos llaves.

El esquema de relación de la base de datos de la Plataforma VideoWeb 2.0 está en 1ra FN ya que cada tupla contiene exactamente un valor para cada atributo de las tablas de la base de datos, o sea, no existen campos multievaluados. También se cumple que el esquema de relación está en 2da FN, pues primeramente se encuentran en 1ra FN, y además todos los atributos que no son claves en las tablas, dependen totalmente de la llave primaria. Además el esquema de relación se encuentra en 3ra FN, pues se encuentra en 2da FN, y no existen dependencias transitivas de atributos no primos. No se logró normalizar la base de datos hasta la 4ta y 5ta forma normal porque en la base de datos diseñada no se utilizan hechos multievaluados y cada atributo depende de la llave primaria de la tabla a la cual pertenece.

La normalización tiene como objetivo evitar la redundancia de los datos, evitar problemas de actualización de los datos en las tablas y proteger la integridad de los datos. Según Edgar F. Codd el creador de las tres primeras formas normales, plantea que las primeras tres formas normales son suficientes para lograr los objetivos antes mencionados.

3.2.3 Análisis de la redundancia de la información

La redundancia de datos es aquella información duplicada o almacenada varias veces en la misma base de datos. Esto dificulta la tarea de modificación de datos y es el motivo más frecuente de inconsistencia de los mismos. La redundancia siempre debe evitarse, aunque en proyectos grandes es imposible evitarla al 100%. Con un buen diseño de una base de datos se logra evitar la aparición de información redundante.

Capítulo 3 Validación de la Base de Datos de Alta Disponibilidad

Después de llevar la base de datos 3ra FN, es decir, al normalizarla, queda libre de redundancias, ya que se elimina completamente la presencia de datos repetidos innecesariamente. La normalización es una técnica que elimina en gran medida la redundancia de los datos.

3.3 Validación funcional del diseño del clúster obtenido

Para comprobar el rendimiento y correcto funcionamiento de la base de datos y el clúster es necesario realizar determinadas pruebas. Las pruebas persiguen el objetivo de simular una carga de producción real y observar cómo se comporta la BD y el clúster ante una determinada carga.

3.3.1 Pruebas de carga y benchmark

Las pruebas de carga fueron utilizadas para validar y evaluar la aceptabilidad de los límites operacionales de un sistema bajo distintos volúmenes de trabajo mientras el clúster se mantiene constante y la otra variante se le aplicó la misma prueba a un solo servidor PostgreSQL. Las pruebas de benchmark se utilizaron para medir el rendimiento del sistema sobre diferentes configuraciones de software y hardware. Este tipo de pruebas de se realiza para encontrar el entorno idóneo de despliegue del sistema.

En el caso particular de la realización de pruebas a los nodos del clúster de bases de datos se han establecido distintas métricas a recolectar como son: el tiempo de respuesta de las consultas, cantidad de transacciones, por ciento de uso de CPU (Unidad Central de Procesamiento) y por ciento de uso de memoria RAM (Memoria de Acceso Aleatorio), pero prestando mayor atención y tomando como métrica principal el tiempo de respuesta de las consultas.

Para la realización de este tipo de prueba se utilizó la herramienta Pgbench para generar carga mediante el acceso simultáneo de un determinado número de peticiones en los nodos que conforman el clúster de bases de datos. Para medir el consumo de recurso cuando se realizan las peticiones a los servidores, se utilizó la herramienta Sysstat.

Se debe tener en cuenta que estas pruebas se realizarán sobre computadoras personales con diferente nivel de recursos que los servidores reales en los que serán desplegadas las aplicaciones. Esto trae consigo que los resultados obtenidos no sean exactos pero ayudan a entender como mejora o se deteriora el funcionamiento de las aplicaciones desplegadas sobre el clúster.

3.3.2 Pruebas de configuración del clúster

El clúster incluye las siguientes funcionalidades:

Capítulo 3 Validación de la Base de Datos de Alta Disponibilidad

- Envía las consultas de lectura (SELECT), a todos los nodos SQL, balanceando las mismas entre aquellos que presenten mayor disponibilidad.
- Envía todas las consultas de modificación (INSERT, UPDATE), hacia todos los nodos SQL.
- Alta disponibilidad para el servicio de balanceo de carga.

Funcionalidades a probar

- Balanceo de carga.
- Replicación.
- Alta disponibilidad en el balanceador de carga.
- Recuperación ante fallos de los servidores de bases de datos.
- Correcto funcionamiento de la aplicación.

Comportamiento esperado de las funcionalidades del clúster a probar

- Balanceo de carga: Se espera que al llegar las consultas de lectura (SELECT) al balanceador de carga las distribuya entre los servidores de bases de datos de la forma más equitativa posible.
- Replicación: Se espera que cuando se realice alguna operación de modificación, la misma se realice sobre todos los servidores de bases de datos mediante el proceso de replicación.
- Alta disponibilidad en el balanceador de carga: Se espera que si el balanceador de carga primario deja de prestar sus servicios el balanceador de carga de respaldo sea capaz de recuperar el sistema automáticamente y sin la intervención del administrador.
- Recuperación ante fallos de los servidores de bases de datos: Se espera que si uno de los servidores de bases de datos deja de prestar sus servicios el balanceador de carga lo detecte y deje de reenviarle consultas para que esto no afecte el funcionamiento del sistema.

3.3.3 Pruebas al diseño genérico de la propuesta de solución

Para validar que el diseño de forma general del clúster obtenido y medir el rendimiento del mismo se realizaron pruebas de configuración y carga a los nodos que conforman el clúster de bases de datos para la Plataforma VideoWeb 2.0. El listado de las pruebas realizadas y el propósito de cada una de ellas se describen en la siguiente tabla.

Capítulo 3 Validación de la Base de Datos de Alta Disponibilidad

Prueba	Tipo	Descripción
Prueba 1	Carga	Comprobar que el diseño para la propuesta de solución balancea de las cargas entre los nodos que conforman el clúster de bases de datos, realizándose todas las consultas UPDATE e INSERT en todos los nodos.
Prueba 2	Configuración	Comprobar que el diseño para la propuesta de solución realiza correctamente la replicación de datos.
Prueba 3	Alta disponibilidad	Comprobar que el diseño para la propuesta de solución se recupera ante algún fallo de algún nodo.

Tabla 14. Pruebas de diseño genérico.

Prueba 1

Para analizar como el balanceador de carga distribuye las conexiones entre los nodos de bases de datos, se utiliza la herramienta Pgbench para generar cierta cantidad de consultas. Para comprobar la cantidad de consultas que se realiza en cada nodo, se revisa los logs de acceso de los nodos PostgreSQL.

Se realizaron cuatro iteraciones en la prueba 1, en una primera iteración se conectaron de manera concurrente 100 clientes realizando 500 consultas simultáneamente en los nodos que conforman el clúster. En una segunda iteración se conectaron 100 clientes realizando 100 consulta de manera concurrente. En la tercera iteración se conectaron 1000 usuarios realizando 1000 consulta y en una cuarta iteración se conectaron los mismo 1000 clientes realizando 10000 consultas, obteniéndose el siguiente resultado (ver tabla 15).

Recursos	1ra iteración	2da Iteración	3ra Iteración	4ta Iteración
% de uso de CPU	35.12%	42.8%	63.62%	79.17%
% de consumo de memoria RAM	16.54%	11.15%	49.18%	60.21
Consultas atendidas en el 1er nodo	257	52	506	5011
Consultas atendidas en el 2do nodo	243	48	494	4989
Transacciones por segundos	26.48	19.26	93.37	135.89

Tabla 15. Consumo de recursos.

Prueba 2

Capítulo 3 Validación de la Base de Datos de Alta Disponibilidad

Para comprobar cómo se comporta el proceso de replicación, se realizan consultas de modificación y posteriormente se analiza si las consultas se realizaron a su vez en todos los nodos de bases de datos. Para verificar que la modificación se realizó en ambos nodos, se examinan los logs de acceso de PostgreSQL en cada uno de los nodos.

Prueba 3

Para simular la caída de uno de los nodos, se desconectó la red, observando el comportamiento del balanceador de carga cuando detecta que uno de los nodos no se encuentra brindando el servicio. Como respuesta a esto se configuró la herramienta Heartbeat, la cual a través de una dirección IP (*Internet Protocol*) de servicio monitorea constantemente los nodos balanceadores de carga, al ver que un balanceador de carga falla emite varios mensajes de errores indicando que hay un nodo fuera de servicio y reenvía todas las peticiones que realizan los usuarios al otro balanceador de carga, siendo este proceso transparente para el usuario.

3.4 Análisis de los resultados obtenidos

Con la correcta realización de las pruebas se pudo comprobar que el clúster de bases de datos para la Plataforma VideoWeb 2.0 funciona correctamente. En la figura 14 se pueda observar el porcentaje del CPU en cada una de las iteraciones realizadas, comprobándose que a mayor cantidad de usuarios y mayor cantidad de consultas realizadas por los mismos, mayor será el porcentaje del CPU.

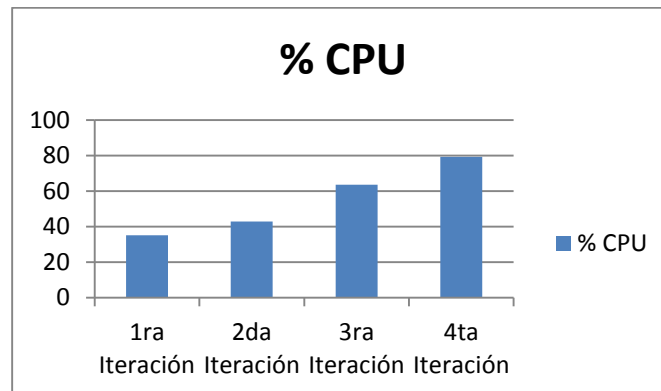


Figura 14. Análisis del % del CPU en las cuatro iteraciones.

En la figura 15 se puede observar el porcentaje de la RAM, que a medida que se incrementan las consultas y los clientes conectados concurrentemente al clúster de la base de datos, el mismo va aumentando.

Capítulo 3 Validación de la Base de Datos de Alta Disponibilidad

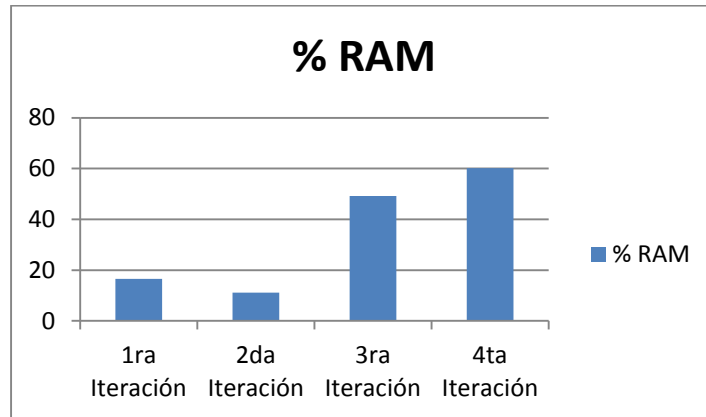


Figura 15. Análisis del % de la RAM en las cuatro iteraciones.

En la siguiente figura se puede observar la cantidad de consultas que atiende por cada nodo en las cuatro iteraciones realizadas.

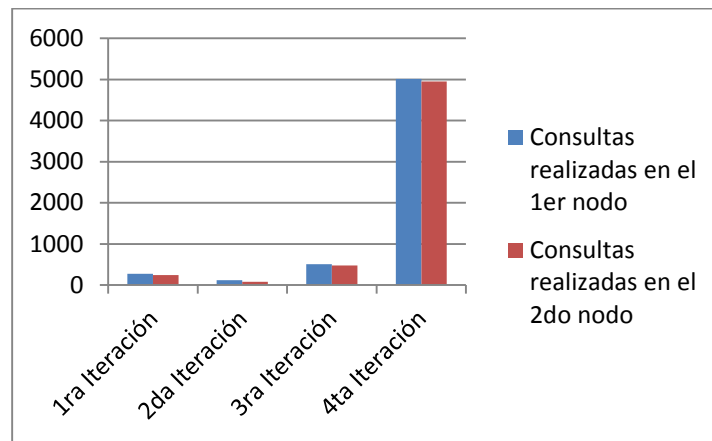


Figura 16. Análisis de las consultas efectuadas en cada uno de los nodos.

En la figura 17 se evidencia el tiempo que demora el clúster en atender todas las peticiones realizadas por los clientes. El mismo va aumentando en dependencia de la cantidad de clientes conectados y la cantidad de consultas que realizan.

Capítulo 3 Validación de la Base de Datos de Alta Disponibilidad

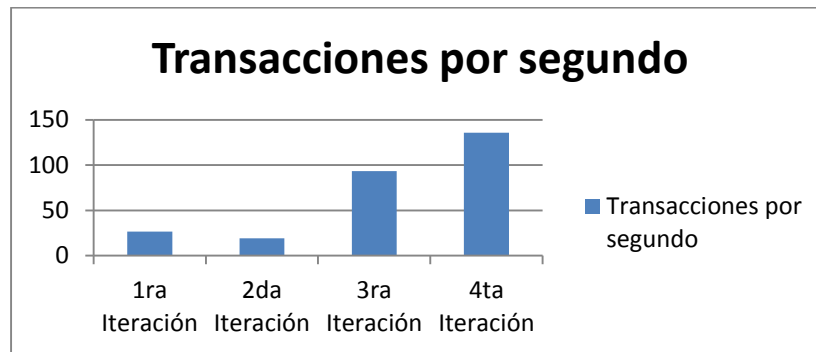


Figura 17. Análisis de las transacciones por segundo realizadas.

A continuación se observará el comportamiento de un solo nodo atendiendo todas las peticiones. Los parámetros a tener en cuenta son, el tiempo que demora un solo nodo en atender todas las peticiones de manera concurrente realizadas en las cuatro iteraciones, así como el % de la memoria RAM y el % del CPU utilizado.

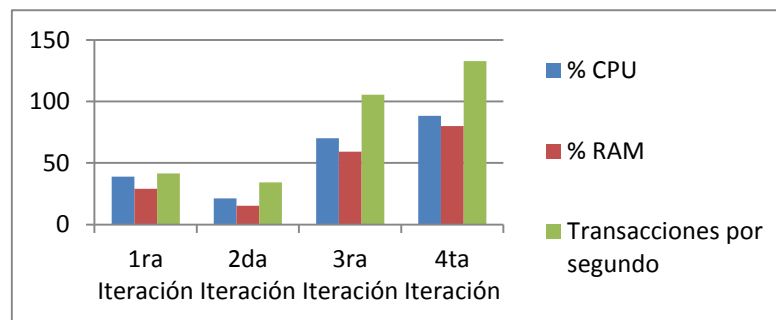


Figura 18. Análisis de los recursos cuando funciona un solo nodo.

3.5 Conclusiones Parciales

Con la validación teórica del diseño realizado se chequearon una serie de parámetros, aspectos y consideraciones importantes a tener en cuenta para realizar un correcto diseño de una BD. Se definieron restricciones de integridad para garantizar la validez de los datos introducidos o modificados en la misma, además como el proceso de normalización, se garantizó la eliminación total de la redundancia de información.

Con la correcta realización de las pruebas descritas en el capítulo recién concluido se logra que la Plataforma VideoWeb brinde un servicio ininterrumpido, permitiendo agilizar los procesos que se realizan en la plataforma, ya sean de inserción, modificación, eliminación o búsqueda de información referente a los contenidos audiovisuales.

Conclusiones generales

A partir de la investigación realizada para obtener el diseño de una propuesta de solución que permita gestionar y manipular los datos referentes a los contenidos audiovisuales y obtener tiempos de respuestas aceptables ante una gran concurrencia de usuario, se concluye que se obtuvieron los resultados enumerados a continuación:

- Se seleccionó el modelo de datos y el SGBD más factible a utilizar en la gestión y manipulación de grandes volúmenes de datos.
- Se realizó el diseño de la BD, el cual cumple con las necesidades de la plataforma.
- Se obtuvieron los modelos lógicos y físicos en el diseño de la base de datos.
- Con las pruebas realizadas a la base de datos se minimizó la redundancia de información en la misma.
- Con la correcta implementación y configuración del clúster de alta disponibilidad se logró que el tiempo de respuesta de los servicios de transmisión y publicación de los contenidos audiovisuales sea aceptable.

De manera general se obtuvo el diseño de una BD que gestiona un gran volumen de información y que el tiempo de respuesta de los servicios que ofrece la Plataforma VideoWeb 2.0 sea aceptable

Recomendaciones

Una vez cumplido el objetivo general, se recomienda

- Continuar el estudio de la solución propuesta y mejorarla mediante la implementación con la versión 3.2 de la herramienta Pgpool-II.

Referencias Bibliográficas

1. **Navarrete, Navarrete.** *Introducción a las bases de datos.* 2004.
2. **paredes, Juan Pedro.** *Alta Disponibilidad para Linux.* 2001.
3. **Florida, Orlando.** Historia de las bases de datos. [En línea] [Citado el: 4 de Marzo de 2013.] <http://www.preceden.com/timelines/48777-historia-de-las-bases-de-datos>.
4. **Silberschatz, Abraham.** *Fundamentos de Base de Datos.* 2002.
5. **Mato, Rosa María.** *Sistemas de Bases de datos.* La Habana : Pueblo y Educación, 2005.
6. **Longman, Addison Wesley.** *Introducción a los sistemas de Base de Datos.* 2000.
7. **Barajas, Matilde Moreno.** El servicio de consulta dinámica. [En línea] [Citado el: 2012 de 11 de 27.] <http://www.gestiopolis.com/administracion-estrategia/servicio-de-consulta-olap-bases-de-datos.htm>.
8. **Santos, Juan C.** Bases de datos y sistemas de gestión de bases relacionadas y procedimiento software. [En línea] 2009. [Citado el: 28 de 11 de 2012.] <http://www.slideshare.net/luisitops/bases-de-datos-1497047>.
9. **Castro Elena, García Sonia.** *Fundamentos de Bases de datos.* Univerisdad Carlos III Madrid. : s.n., 2009.
10. **Alvarez, Sara.** Modelos de bases de datos. [En línea] 2007. [Citado el: 27 de 11 de 20012.] <http://www.desarrolloweb.com/articulos/modelos-base-datos.html>.
11. **Moraga, M. Ángeles.** *El Modelo de Datos.* Universidad de Castilla - La Mancha : s.n., 2001.
12. **Marqués, Merche.** Tema 2. Bases de datos orientadas a objetos. [En línea] 2002. [Citado el: 28 de 11 de 2012.] <http://www3.uji.es/~mmarques/e16/teoria/cap2.pdf...>
13. **Ezequiel, Rozic Sergio.** *Bases de dato.* Buenos Aires, Argentina : MP Ediciones, 2004.
14. **Ramírez, Raquel Zambrano.** *Sistemas gestores de base de datos.* Cordoba : Pueblo Nuevo, 2008.
15. **ICM.** Bases de dato. [En línea] [Citado el: 28 de 11 de 2012.] <http://www.icm.es/tecnologias/bases-de-datos>.
16. **Microsoft.** Microsoft. [En línea] [Citado el: 30 de 11 de 2012.] <http://www.microsoft.com/sqlserver/en/us/product-info/top-twelve.aspx>.
17. **Casillas Santillán Luis Alberto, Ginestá Marc Gibert, Pérez Mora Óscar.** Bases de Datos MYSQL. [En línea] 2007. [Citado el: 5 de 10 de 2012.] <http://www.uoc.edu>.
18. **MYSQL.** MYSQL. [En línea] [Citado el: 30 de 11 de 2012.] <http://www.mysql.com/why-mysql>.
19. **Martínez, Rafael.** PostgreSQL. [En línea] 2010. [Citado el: 21 de 11 de 2012.] www.postgresql.org.es/sobre_postgresql.

Referencias Bibliográficas

20. **Matos, Loan Joa.** *Manual de Administración de servidores de Bases de datos PostgreSQL.* 2011.
21. **Christopher, J Date.** *Introducción a los Sistemas de Bases de Datos.* Ciudad de La Habana : Editorial Félix, 2003.
22. **Peña, Montero Adrian Misael.** *Guía para la instalación de un clúster de Bases de Datos.* La Habana : Centro de Tecnologías de Gestión de Datos DATEC, 2010.
23. **Narayan, Aditya.** Introducción a los conceptos básicos de la informática de alto rendimiento con tecnología cluster Linux. [En línea] 2005. [Citado el: 30 de 11 de 2012.] <http://www.ibm.com/developerworks/linux/library/l-cluster1>.
24. **MacVittie, Lori.** Clustering versus load-balancing. [En línea] 2007. [Citado el: 30 de 11 de 2012.] <https://devcentral.f5.com/blogs/us/clustering-versus-load-balancing>.
25. **Clavijo, Paulo.** Clusters de Alta Disponibilidad. [En línea] [Citado el: 2 de 12 de 2012.] www.lintips.com..
26. **ITC Group.** Clusters de Bases de Datos y Aplicaciones. [En línea] 2012. [Citado el: 30 de 11 de 2012.] <http://www.itcgroup.us/index.php/soluciones/mision-critica/lave-en-mano/clusters-de-bases-de-datos-y-aplicaciones>.
27. **Barroso, Luiz André.** *Web search for a planet: The Google cluster.* s.l. : IEEE, 2012.
28. **Juventud Rebelde.** La «cibernética» de la biotecnología. [En línea] 15 de 3 de 2012. [Citado el: 5 de 12 de 2012.] http://www.cigb.edu.cu/index.php?option=com_content&view=article&id=295.
29. **Vélez, Antonio.** *RUP: Etapas Análisis y Diseño.* Universidad del Balle, Palmaria : s.n., 2010.
30. **Merayo, Ramon.** El contenido audiovisual: otro reto para la accesibilidad web. [En línea] [Citado el: 10 de 12 de 2012.] <http://www.ub.edu/bid/21/voces2.htm>.. 21.
31. **CMS.** Selecting a developmet approach. [En línea] 27 de Marzo de 2008. [Citado el: 1 de Marzo de 2013] <https://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>.
32. **IBM.** IBM Rational Unified Process (RUP). [En línea] [Citado el: 10 de 12 de 2012.] <http://www-01.ibm.com/software/awdtools/rup/>..
33. **Floras, Díaz Mirian Milagros.** Metodología XP Vs. Metodología Rup. [En línea] [Citado el: 1 de Marzo de 2013.] <http://www.usmp.edu.pe/publicaciones/boletin/fia/info49/articulos/RUP%20vs.%20XP.pdf>.
34. **Cornejo, José Enrique González.** El lenguaje de Modelado Unificado UML. [En línea] [Citado el: 1 de Marzo de 2013.] <http://www.docirs.cl/uml.htm>.

Referencias Bibliográficas

35. **Fuentes Lidia, Vallecillo Antonio.** Una Introducción a los Perfiles UML. Lenguajes y Ciencia de la Computación. [En línea] [Citado el: 19 de Enero de 2013.] http://www.intramed.net/UserFiles/Files/3_endnote.pdf..
36. **Prodevelop.** Documentation Case UML 2.0. [En línea] <https://confluence.prodevelop.es/display/caseuml2/Documentation>.
37. **González López Pascual, González López Ana Amelia , Gallud Lázaro José Antonio.** *Herramientas Case. ¿Cómo incorporarlas con éxito en nuestra organización?* Universidad de Castilla - La Mancha : s.n., 2011.
38. **HERNANDIS, J. A.** Visual Paradigm para UML. [En línea] 2005. [Citado el: 11 de 12 de 2012.] <http://www.versioncero.com/noticia/210/visual-paradigm-for-uml>.
39. **Visual Paradigm.** Visual Paradigm for UML 8.0 Release Notes. [En línea] [Citado el: 1 de Marzo de 2013.] <http://www.visual-paradigm.com/aboutus/newsreleases/vpuml80.jsp>.
40. **Neustadt, Wiener.** CyberCluster 1.0 Technical Documentation. [En línea] [Citado el: 10 de 05 de 2013.] http://www.cybertec.at/download/dokumentation/documentation_cybercluster.pdf..
41. **Tarreau, W.** HAProxy. The Reliable, High Performance TCP/HTTP Load Balancer. [En línea] [Citado el: 10 de 05 de 2013.] <http://haproxy.1wt.eu>.
42. **Pressman, Roger S.** *Software Engineering a Practitioner's Approach.* . New York San Francisco : s.n., 2010. 7 ma Edición.
43. **Jacobson Ivar, Booch Grady, Rumbaugh James.** *El proceso unificado del software.* Madrid : Pearson educacion SA, 2000.
44. **Definición.** Definición de. [En línea] [Citado el: 20 de 03 de 2013.] <http://definicion.de/integridad/>.
45. **Savedra, Juan.** *Integridad Informática enfocada en las bases de datos.* 2009.
46. **Garcias, Matos Rosa Maria.** *Diseño de Bases de Datos.* La Habana : Pueblo y Educación, Octubre 1999.
47. **Schneier, Bruce.** Cryptanalysis of SHA-1. [En línea] [Citado el: 30 de marzo de 2013.] http://www.schneier.com/blog/archives/2005/02/cryptanalysis_o.html.

Bibliografía

Aguilar Roselló V.J. Clustering de alta disponibilidad bajo GNU / LINUX. In Departamento de Lenguajes y Sistemas Informáticos. Universidad de Alicante. 2001.

Barajas Matilde Moreno. El servicio de consulta dinámica. [En línea] [Citado el: 2012 de 11 de 27.] <http://www.gestiopolis.com/administracion-estrategia/servicio-de-consulta-olap-bases-de-datos.htm>.

Bases de Datos. PostgreSQL & Pgpool-II. [En línea] 2011. [Citado el: 04 de 12 de 2012.] <http://basesdedatosues.blogspot.com/2010/06/bdd-postgresql-Pgpool-ii.html>.

Bayona Valentín Maribel, González Horta Ulises. Implementación de un clúster de alta disponibilidad utilizando herramientas libres. [En línea] 2009. [Citado el: 06 de 01 de 2013.] <http://aslcuba.wordpress.com/2009/07/16/implementacion-de-un-cluster-de-alta-disponibilidad-utilizando-herramientas-libres/>.

Date Christopher J. Introducción a los Sistemas de Bases de Datos. Ciudad de La Habana: Editorial Félix, 2003.

Elmasri Ramez. Fundamentos de sistemas de bases de datos, Madrid. Pearson Educación. 2002.

Florida Orlando. Historia de las bases de datos. [En línea] [Citado el: 4 de Marzo de 2013.] <http://www.preceden.com/timelines/48777-historia-de-las-bases-de-datos>.

Gómez Ballester Eva, Martínez Barco Patricio, Moreda Pozo Paloma, Suárez Cueto Armando, Montoyo Guijarro Andrés, Saquete Boro Estela. Introducción a las Bases de Datos 1. 2007. Universidad de Alicante España.

Hernández Lenin. Alta Disponibilidad de Pgpool-II. [En línea] 2012. [Citado el: 01 de 01 de 2013.] <http://tubasededatoslibre.org/site/wp-content/uploads/2012/05/PonenciaLHernandez.pdf>.

Ibañez Luna, Denisse Haidee. Implementación de un clúster de alta disponibilidad de bases de datos para la GEDGAPA. 2012.

Iglesias Aparicio Abraham. A High Availability platform design using Heartbeat and integration in a production environment. Editorial Universidad Politécnica de Catalunya. 2011.

Longman, Addison Wesley. Introducción a los sistemas de Base de Datos. 2000.

MacVittie, Lori. Clustering versus load-balancing. [En línea] 2007. [Citado el: 30 de 11 de 2012.] <https://devcentral.f5.com/blogs/us/clustering-versus-load-balancing>.

Martínez Paloma, Castro Elena, Cavero José, Dolores Cuadra Dolores, Ana Iglesias Ana, Nieto Carlos. Diseño de Bases de Datos Problemas Resueltos. 2005. Editorial *Alfaomega Grupo Editor*.

Martinez Rafael. Replicación y Alta Disponibilidad con Pgpool-II. [En línea] 2009. [Citado el: 07 de 12 de 2012.] <http://www.postgresql.org.es/node/313>.

Moraga, M. Ángeles. El Modelo de Datos. Universidad de Castilla - La Mancha: s.n., 2001.

Narayan, Aditya. Introducción a los conceptos básicos de la informática de alto rendimiento con tecnología clúster Linux. [En línea] 2005. [Citado el: 30 de 11 de 2012.] <http://www.ibm.com/developerworks/linux/library/l-cluster1>.

Nevado Cabello María Victoria. Introducción a las Bases de Datos Relacionales. 2010.

Paredes Juan Pedro. Alta disponibilidad para Linux. [En línea] 2001. [Citado el: 01 de 01 de 2013.] <ftp://ftp.inf.utfsm.cl/pub/Linux/Docs/LuCaS/Presentaciones/200103hisपालinux/paredes/pdf/LinuxHA.pdf>.

Ramírez, Raquel Zambrano. Sistemas gestores de base de datos. Cordoba: Pueblo Nuevo, 2008.

Romesburg H. Charles, Cluster analysis for researchers. 2004. North Carolina.

Sabater Jaume. Replicación y alta disponibilidad de PostgreSQL con Pgpool-II. [En línea] 2008. [Citado el: 30 de 11 de 2012.] <http://linuxsilo.net/articles/postgresql-Pgpool.html>.

Silberschatz Abraham. Fundamentos de Base de Datos. 2002.

Anexos

Anexo #1: Descripción de las tablas restantes tablas de la Plataforma VideoWeb 2.0.

Nombre: tipologia

Descripción: Tabla que guarda los tipos de tipologías existentes.

Atributo	Tipo
id_t	int4
type	varchar
title	varchar
title_label	varchar
created	int4
changed	int4
help	text
auxiliar	int2
weight	int4
disabled	int2
u_id	int8

Tabla 16: tipologia

Nombre: relacion_am_data_tipologia

Descripción: Tabla que surge de la relación de mucho a mucho de las tablas tipologia_data y archivo_multimedia, guardando los datos comunes entre dichas tablas.

Atributo	Tipo
archivo_multimedia_id_am	int4
tipologia_data_id_td	int4
tiempo_inicio	int4
tiempo_fin	int4

Tabla 17: relacion_am_data_tipologia

Nombre: rango_ip

Descripción: Tabla que guarda las restricciones por ip que pueden realizar sobre la aplicación.

Atributo	Tipo
id_rango	int4
nombre_rango_ip	varchar
rango_ip	varchar
media_id	varchar
tiene_acceso	int4
version_ip	varchar

Tabla 16: rango_ip

Nombre: rango_horario

Descripción: Tabla que guarda las restricciones por horario que se puede acceder a la aplicación.

Atributo	Tipo
id_rango_horario	int4
hora_inicio	time
hora_fin	time
media_id	varchar
tiene_acceso	int4

Tabla 17. rango_horario

Anexo # 2: Manual de Configuración de una Clúster de Alta Disponibilidad para la Base de Datos de la Plataforma VideoWeb 2.0.

Glosario de términos

Álgebra Relacional: es un conjunto de operaciones simples sobre tablas relacionales, a partir de las cuales se definen operaciones más complejas mediante composición.

Balanceador de carga: es un concepto usado en informática que se refiere a la técnica usada para compartir el trabajo a realizar entre varios ordenadores.

CASE: Ingeniería de Software Asistida por Computadora, aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software.

Clúster: conjunto de computadoras, que se interconectan entre sí a través de un sistema de red de alta velocidad y son capaces de elevar la eficiencia para realizar determinadas tareas que individualmente no podrían realizar.

DDL: Lenguaje de definición de datos (Data Definition Language) es un lenguaje para definir y describir los objetos de la base de datos, su estructura, relaciones y restricciones. En la práctica puede consistir en un subconjunto de instrucciones de otro lenguaje informático

DML: Lenguaje de Manipulación de Datos (Data Manipulation Language, DML) es un lenguaje proporcionado por el sistema de gestión de base de datos que permite a los usuarios de la misma llevar a cabo las tareas de consulta o manipulación de los datos, organizados por el modelo de datos adecuado.

Drupal: es un CMS o sistema de gestión de contenidos que se utiliza para crear sitios web dinámicos y con gran variedad de funcionalidades. Drupal es un software libre, escrito en PHP, que cuenta con una amplia y activa comunidad de usuarios y desarrolladores.

GNU GPL: La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente sus siglas del inglés GNU GPL, es una licencia creada por la Free Software Foundation en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

Licencia BSD: es una licencia de software otorgada principalmente para los sistemas BSD (Berkeley Software Distribution), es una licencia de software libre permisiva que permite el uso del código fuente en software no libre.

MD5: MD5 (*Message-Digest Algorithm 5* o *Algoritmo de Resumen del Mensaje 5*) es un Algoritmo de reducción criptográfico de 128 Bits ampliamente usado. El código MD5 fue diseñado por Ronald Rivest en 1991.

Nodo: En la terminología de clúster, un nodo es cualquier máquina física que juega un determinado papel dentro del clúster.

OCF (Open Clúster Framework): Es un estándar mediante el cual el CRM (software que provee del control sobre los recursos desplegados en el clúster) interactúa con los servicios que corren en cada máquina. Mediante los estándares OCF (realmente son scripts que interactúan con otros scripts) se manejan los scripts típicos de "/etc/init.d" para obtener el estado de cada servicio, pararlos o arrancarlos.

RBAC: Patrón Control de Acceso Basado en Roles.

Réplica: es el proceso de copiar y mantener actualizados los datos en varios nodos de bases de datos.

SHA1: SHA1 (Secure Hash Algorithm, Algoritmo de Hash Seguro) es un sistema de funciones hash criptográficas relacionadas de la Agencia Nacional de Seguridad estadounidense.

SQL/92: es un estándar desarrollado en 1992, diseñado para los sistemas de bases de bases de datos (RDBMS).

SQL/99: es un estándar actual para bases de datos objeto-relacionales, desarrollado en 1999.

SSL: SSL del inglés *Secure Socket Layer* o Capa de Enchufe Seguro. Proporciona servicios de seguridad cifrando los datos intercambiados entre el servidor y el cliente con un algoritmo de cifrado simétrico.

Tablas HEAP: son tablas que se almacenan de forma temporal en memoria.