



FACULTAD DE ARQUITECTURA Y URBANISMO E INGENIERIAS

ESCUELA PROFESIONAL DE INGENIERIA INFORMATICA Y DE SISTEMAS

**COMPARADOR Y SINCRONIZADOR DE BASES DE DATOS RELACIONALES EN
SERVIDORES LOCALES DE LA UNIVERSIDAD PARTICULAR DE CHICLAYO**

**TESIS PARA OPTENER EL TITULO PROFESIONAL DE INGENIERO
INFORMATICO Y DE SISTEMAS**

AUTOR

Cruz Benel Cristian Arturo

Oblitas Jimenez Jose Miguel Orlado

ASESOR

Aguilar Fernandez Luis Guillermo

CHICLAYO – PERU

2017

DEDICATORIA

A mis padres Juan y Teonila, familiares más cercanos por todo su apoyo incondicional durante mis años de estudio y por depositar en mí su confianza para lograr mis objetivos propuestos.

Para mis abuelos que me protegen y velan por mí desde el cielo y que siempre han sido un motor para llevar a cabo mis metas.

A mis padres Miguel y Rosa familiares más cercanos por todo su apoyo incondicional durante mis años de estudio y por depositar en mí su confianza para lograr mis objetivos propuestos.

Para mis abuelos que me protegen y velan por mí desde el cielo y que siempre han sido un motor para llevar a cabo mis metas.

AGRADECIMIENTO

Un agradecimiento especial a los ingenieros. Oscar Castro Yoshida y Jose Aguilar por todos sus consejos y recomendaciones durante el desarrollo de la presente tesis, tanto Como profesor y como asesor y que ha permitido que el proyecto se culmine con éxito.

A todos nuestros profesores que durante mis 4 años de estudios no sólo nos han brindado conocimiento, sino nos han inculcado valores y el deseo de siempre aprender y no sucumbir ante las adversidades.

A todos nuestros amigos y compañeros de clases con los que hemos vivido muchas experiencias tanto en lo académico como lo personal y que han aportado a nuestro crecimiento profesional y como persona.

DECLARACION DE AUTENTICIDAD

Nosotros.....Con DNI
Nº.....y.....
con DNI Nº....., a efecto de cumplir con las disposiciones vigentes
consideradas en el Reglamento de Grados y Títulos de la Universidad Particular de
Chiclayo, Facultad de Arquitectura Urbanismo e Ingeniarías Escuela de Ingeniería
Informática y de Sistemas declaramos bajo juramento que toda la documentación
que acompaño es veraz y auténtica.

Así mismo, declaramos también bajo juramento que todos los datos e información
que se presenta en la presente tesis son auténticos y veraces.

En tal sentido asumimos la responsabilidad que corresponda ante cualquier falsedad,
ocultamiento u omisión tanto de los documentos como de información aportada por lo
cual nos sometemos a lo dispuesto en las normas académicas de la Universidad
Particular de Chiclayo.

Chiclayo,..... del 2017

Nombres y apellidos de los tesistas

PRESENTACIÓN

Señores miembros del Jurado:

En cumplimiento del Reglamento de Grados y Títulos de la Universidad Particular de Chiclayo presentamos ante ustedes la Tesis titulada **“Sincronizador y comparador de bases de datos relacionales en servidores locales de la Universidad Particular de Chiclayo”**, la misma que sometemos a vuestra consideración y esperamos que cumpla con los requisitos de aprobación para obtener el título Profesional de Ingeniero Informático y de Sistemas.

José Miguel Oblitas Jiménez

Cristian Arturo Cruz Benel

INDICE

INDICE DE FIGURAS.....	8
INDICE DE TABLAS.....	11
RESUMEN.....	12
ABSTRACT.....	14
I. INTRODUCCION.....	16
1.1. Realidad problemática.....	17
1.2. Formulación del problema.....	18
1.3. Hipótesis.....	18
1.4. Objetivos.....	19
1.4.1. Objetivo general.....	19
1.4.2. Objetivo específico.....	19
II. BASES TEÓRICAS.....	20
2.1. Antecedentes de la Investigacion.....	21
2.2. Marco teórico.....	27
2.3. Definición de términos.....	33
III. MARCO METODOLÓGICO.....	41
3.1. Variables.....	41
3.1.1. Variable Dependiente.....	41
3.1.2. Variable Independiente.....	41
3.2. Operacionalización de variables.....	42
3.3. Metodología.....	43
3.3.1. Tipo de Estudio.....	43
3.3.2. Diseño de Investigacion.....	44
3.4. Población, Muestra, Muestreo.....	44
3.5. Método de análisis de datos.....	45
3.6. Aspectos éticos.....	45
IV. RESULTADOS.....	47
4.1. Resultados encuesta 01.....	50
4.2. Resultados encuesta 02.....	54

V. DISCUSIÓN.....	57
VI. CONCLUSIONES.....	59
VII. RECOMENDACIONES.....	60
VIII. PROPUESTA TECNOLÓGICA.....	61
8.1. Generalidades.....	61
8.2. Análisis.....	65
8.3. Diseño.....	69
8.3.1. Arquitectura de la red.....	70
8.3.2. Diseño de la tabla a replicar.....	73
8.3.3. Diseño del formulario para las pruebas.....	73
8.4. Implementación.....	76
8.4.1. Virtualización con VMware Workstation.....	76
8.4.2. Sistema Operativo CentOS.....	77
8.4.3. Servidor web Apache.....	81
8.4.4. Lenguaje de programación PHP 5.6.....	83
8.4.5. PostgreSQL 8.0.....	84
8.4.6. Base de datos.....	88
8.4.7. Formularios CRUD.....	93
8.4.8. Replicación con Bucardo.....	104
8.4.8.1. Instalacion de perl.....	104
8.4.8.2. Instalacion de Perl-DBD-Pgpackage.....	104
8.4.8.3. Instalación de Postgresql-plperl.....	105
8.4.8.4. Instalación del módulo DBIx::Safe.....	105
8.4.8.5. Instalación de perl-Time-HiRes.....	106
8.4.8.6. Instalación de perl-boolean.....	106
8.4.8.7. Instalación de Bucardo.....	107
8.5. Presupuesto.....	110
8.5.1. Inversión.....	111
8.5.2. Beneficios.....	112
IX. BIBLIOGRAFÍA.....	114
X. ANEXOS.....	117

INDICE DE FIGURAS

Imagen 1 Ejemplo replicación en diferentes servidores.....	28
Imagen 2 Estructura PostgreSQL.....	30
Imagen 3 Resultados Primera Encuesta.....	49
Imagen 4 Resultados Segunda Encuesta.....	50
Imagen 5 Modelo Conexión desde una aplicación a Postgresql.....	64
Imagen 6 Diseño gráfico del modelo de replicación entre varios servidores.....	66
Imagen 7 Comparativa resultados de encuestas hechas.....	69
Imagen 8 Estructura del diseño actual de la red y su servidor de base de datos.....	71
Imagen 9 Estructura del diseño actual de la red y su servidor de base de datos.....	72
Imagen 10 Estructura de distribución de servidores de bases de datos propuesta en el presente proyecto.....	73
Imagen 11 Estructura de la tabla Alumno que se utilizara para hacer las pruebas de replicación.....	74
Imagen 12 Vista principal del formulario que interactuara con las bases de datos.....	75
Imagen 13 Formulario para Insertar un nuevo registro en la base de datos PostgreSQL.....	75
Imagen 14 Vista para visualizar los detalles de un registro guardado en la base de datos PostgreSQL.....	75
Imagen 15 Formulario para editar un registro guardado en la base de datos PostgreSQL.....	76
Imagen 16 Programa VMWare en el que se virtualizaran los sistemas operativos CentOS 6.3.....	78
Imagen 17 Contenido del archivo hosts del servidor 01.....	79
Imagen 18 Contenido del archivo "hosts" del servidor 02.....	79
Imagen 19 Archivo "network" del servidor01.....	80
Imagen 20 Archivo "network" del servidor02.....	80
Imagen 21 Archivo "ifcfg-eth0" del servidor01.....	81
Imagen 22 Archivo ifcfg-eth0 del servidor02.....	81
Imagen 23 Comandos para configurar IP estática.....	82
Imagen 24 Instalación del servidor web Apache.....	83
Imagen 25 comandos para instalar y ejecutar el servidor web.....	83
Imagen 26 Comandos para agregar los repositorios de donde se descargara PHP 5.6.....	84
Imagen 27 Configuración para descargar desde el repositorio agregado anteriormente.....	84
Imagen 28 Configuración 2 para descargar desde el repositorio agregado anteriormente.....	85
Imagen 29 Comando para registrar los repositorios REMI.....	85
Imagen 30 Comando para iniciar la descarga de PHP 5.6.....	85
Imagen 31 Comandos para descargar e inicializar PostgreSQL.....	86
Imagen 32 Comando para entrar al gestor de bases de datos postgresql.....	86
Imagen 33 Comando para abrir una consola en postgresql.....	87
Imagen 34 Comando para establecer una contraseña al usuario "postgres".....	87
Imagen 35 Configuración del archivo "pg_hba.conf".....	88
Imagen 36 Comando para reiniciar el servicio postgresql.....	89
Imagen 37 Comando para importar un script con código sql.....	92
Imagen 38 Contenido del script que se ejecutara en la base de datos del servidor 01.....	93
Imagen 39 Contenido del script que se ejecutara en la base de datos del servidor 02.....	94
Imagen 40 Estructura de carpetas en un proyecto de Laravel 5.4.....	95
Imagen 41 Contenido de la carpeta "app" en un proyecto de laravel 5.4.....	96
Imagen 42 Contenido de la carpeta "public" en un proyecto de laravel 5.4.....	96
Imagen 43 Contenido de la carpeta "resources" en un proyecto de laravel 5.4.....	97

Imagen 44 Contenido de la carpeta "routes" en un proyecto de laravel 5.4.....	97
Imagen 45 Contenido del archivo ".env" en un proyecto de laravel 5.4.....	98
Imagen 46 Vista principal del formulario que interactuara con las bases de datos.....	99
Imagen 47 Formulario para insertar un nuevo registro en la base de datos.....	99
Imagen 48 Opción de editar un registro de la base de datos.....	100
Imagen 49 Formulario para editar un registro de la base de datos.....	100
Imagen 50 Resultado después de la edición de un registro.....	101
Imagen 51 Opción de eliminar un registro.....	102
Imagen 52 Resultado después de realizar la eliminación de un registro.....	102
Imagen 53 Ubicación de los formularios en el servidor web.....	103
Imagen 54 Comando para editar el archivo "httpd.conf".....	103
Imagen 55 Líneas de código para agregar un nuevo servidor virtual web.....	104
Imagen 56 comando para reiniciar los servicios web apache.....	104
Imagen 57 Asignacion al usuario apache como propietario de la carpeta tesis_crud.....	104
Imagen 58 permisos de lectura y escritura para el usuario propietario de la carpeta "storage".....	104
Imagen 59 Comando para permitir la conexion entre los formularios y la base de datos PostgreSQL.....	104
Imagen 60 Comandos para instalar los paquetes perl.....	105
Imagen 61 Comando para instalar el paquete Perl-DBD-Pgpackage.....	105
Imagen 62 Comando para instalar PostgreSQL-plperl.....	106
Imagen 63 Ubicación de donde se copió el archivo dbix_safe.tar.gz.....	106
Imagen 64 descompresion del archivo dbix_safe.tar.gz.....	107
Imagen 65 Comandos para hacer la instalación del archivo dbix_safe.tar.gz.....	107
Imagen 66 comando para instalar perl-Time-Hires.....	107
Imagen 67 Comando para instalar perl-boolean.....	107
Imagen 68 Creacion de una carpeta para que bucardo guarde los .log que genera.....	107
Imagen 69 Asignación de permisos de lectura y escritura para bucardo en la carpeta /var/log/bucardo.....	108
Imagen 70 Ubicación de donde se copió el instalador de bucardo.....	108
Imagen 71 Comandos para descomprimir e instalar Bucardo 5.4.1.....	109
Imagen 72 Agregación delas bases de datos que se replicaran.....	109
Imagen 73 Asignacion de las tablas que se sincronizaran.....	109
Imagen 74 Asignacion de los herd.....	110
Imagen 75 Creación de los sincronismos en Bucardo.....	110
Imagen 76 Comando para iniciar la replicación.....	110
Imagen 77 Estatus actual de la replicación realizada.....	110
Imagen 78 Comando para editar el archivo crontab.....	111
Imagen 79 Contenido del archivo crontab.....	111
Imagen 80 Reinicio del demonio cron.....	112

INDICE DE TABLAS

<i>Tabla 1 Operacionalizacion de Variables.....</i>	<i>42</i>
<i>Tabla 2 Resultado Grafico a Pregunta 01 de la primera encuesta.....</i>	<i>49</i>
<i>Tabla 3 Resultado Grafico a Pregunta 02 de la primera encuesta.....</i>	<i>50</i>
<i>Tabla 4 Resultado Grafico a Pregunta 03 de la primera encuesta.....</i>	<i>50</i>
<i>Tabla 5 Resultado Grafico a Pregunta 04 de la primera encuesta.....</i>	<i>51</i>
<i>Tabla 6 Resultado Grafico a Pregunta 05 de la primera encuesta.....</i>	<i>51</i>
<i>Tabla 7 Resultado Grafico a Pregunta 06 de la primera encuesta.....</i>	<i>52</i>
<i>Tabla 8 Resultado Grafico a Pregunta 01 de la segunda encuesta.....</i>	<i>53</i>
<i>Tabla 9 Resultado Grafico a Pregunta 02 de la segunda encuesta.....</i>	<i>53</i>
<i>Tabla 10 Resultado Grafico a Pregunta 03 de la segunda encuesta.....</i>	<i>54</i>
<i>Tabla 11 Resultado Grafico a Pregunta 04 de la segunda encuesta.....</i>	<i>54</i>
<i>Tabla 14 Comparativa Fluctuaciones de ancho de banda en universidad particular de Chiclayo</i>	<i>68</i>
<i>Tabla 15 Características principales de VMWare Workstation.....</i>	<i>76</i>
<i>Tabla 16 Cuadro de la Inversion que se realizara.....</i>	<i>110</i>
<i>Tabla 17 Cuadro para calcular en que mes se recuperar la inversión del proyecto.....</i>	<i>112</i>

RESUMEN

En este proyecto de investigación se realizará un análisis, diseño e implementación de un comparador y sincronizador de bases de datos relacionales en el cual se tomará como caso de estudio y aplicación un problema que actualmente está afectando a la adecuada disponibilidad de la información almacenada en su base de datos central de la Universidad Particular de Chiclayo.

En el capítulo I se escribe una introducción de la investigación el cual está conformado por la descripción de la realidad problemática, la formulación del problema, hipótesis, y objetivos de la presente investigación.

En el capítulo II se hace referencia a las bases teóricas en el cual se describen los antecedentes de la investigación, el marco teórico, los términos, definiciones y descripciones de los temas a tratar en todo el proyecto.

En el capítulo III se plantea el marco metodológico que comprende la declaración, definición y operacionalización de las variables intervinientes en esta investigación, También incluye la metodología de investigación, la población y muestra a analizar, las técnicas e instrumentos de recolección de datos. Por último, se determina el método de análisis de datos y el aspecto ético del investigador.

En el capítulo IV y V se establecen los resultados y la discusión de dichos resultados. Para ello establecemos en forma cuantitativa y porcentual los alcances de los datos obtenidos en la aplicación de encuesta a los que interactuaran con el servidor de replicación de bases de datos.

En el capítulo VI y VII se redacta las conclusiones y recomendaciones obtenidas en el desarrollo de la investigación de este informe.

En el capítulo VIII se narra la propuesta tecnológica el cual está conformado por las generalidades del proyecto, el análisis de cómo se encuentra el entorno actual en el que se llevara a cabo la implementación del tema tratado en esta investigación, también se describe como estará diseñado esta propuesta tecnológica mencionando los diseños de arquitectura de red, diseño de la base de datos que se replicara, y en qué ambiente se realizaran las pruebas. En este capítulo es donde se describe paso a paso los procedimientos, configuraciones e instalaciones que se debe hacer para concluir este proyecto satisfactoriamente. Terminando este apartado se considera el costo, presupuesto y beneficios de este proyecto.

Por ultimo en el capítulo IX se escribe las referencias bibliográficas, las cuales están constituido por todas las fuentes bibliografías y linkograficas utilizadas y revisadas desde el inicio hasta el final del proyecto.

ABSTRACT

In this research project an analysis, design and implementation of a comparator and synchronizer of relational databases will be carried out, in which a problem that is currently affecting the adequate availability of the information stored in its case will be taken as a case study and application. Central database of the Private University of Chiclayo.

In chapter I an introduction of the investigation is written, which is conformed by the description of the problematic reality, the formulation of the problem, hypothesis, and objectives of the present investigation.

In chapter II reference is made to the theoretical bases in which the background of the research, the theoretical framework, the terms, definitions and descriptions of the topics to be treated throughout the project are described.

Chapter III considers the methodological framework that includes the declaration, definition and operationalization of the intervening variables in this research. It also includes the research methodology, the population and sample to analyze, the techniques and data collection instruments. Finally, the method of data analysis and the ethical aspect of the researcher are determined.

Chapter IV and V establish the results and the discussion of said results. To do this, we establish in quantitative and percentage terms the scope of the data obtained in the survey application to which they will interact with the database replication server.

In chapter VI and VII the conclusions and recommendations obtained in the development of the investigation of this report are written.

In chapter VIII the technological proposal is narrated which is conformed by the generalities of the project, the analysis of how is the current environment in which the implementation of the topic dealt with in this research is carried out, it is also described how it will be designed this technological proposal mentioning the network architecture designs, design of the database that will be replicated, and in which environment the tests will be carried out. This chapter is where we describe step by step the procedures, configurations and facilities that must be done to successfully complete this project. Finishing this section considers the cost, budget and benefits of this project.

Finally, in chapter IX, the bibliographic references are written, which are constituted by all the bibliography and linkographic sources used and reviewed from the beginning to the end of the project.

I. INTRODUCCION

Las Bases de Datos es una tecnología que se ha desarrollado a pasos agigantados en los últimos años debido al hecho de que las instituciones y empresas consideran de vital importancia a la información, convirtiendo a la misma en el activo principal de las organizaciones, como lo es el caso de la Universidad Particular de Chiclayo, en la cual se hace uso de esta tecnología para almacenar todo tipo de información referente a los estudiantes, como notas, asistencias, pagos, entre otro tipo de datos que se utiliza para poder tomar decisiones o planear estrategias administrativas.

Debido a la importancia que tiene la información que guarda las empresas en sus bases de datos se realizan técnicas de seguridad como medida de prevención ante desastres, la técnica más utilizada es la de realizar backups cada cierto tiempo, sin embargo, existe métodos más óptimos y eficientes para garantizar la disponibilidad y seguridad de la información como lo es la replicación de bases de datos sobre la cual se centra este proyecto de investigación.

En la actualidad los procesos de sincronización suelen realizarse de manera manual, lo que origina errores que pueden perjudicar el desempeño de la base de datos y, por ende, el de los sistemas que dependen de dicha base de datos. Los escenarios más frecuentes en donde se lleva a cabo este proceso son cuando se realizan pases a producción que traen consigo cambios en la estructura de la base de datos y ambientes en donde se utilizan bases de datos distribuidas para el procesamiento y almacenamiento de información en distintas sucursales.

El presente proyecto tiene como objetivo la implementación de servidores de bases de datos en las principales sedes de la Universidad Particular de Chiclayo, luego mediante la técnica de replicación identificar las diferencias y similitudes entre distintos objetos dentro de las bases de datos y a su vez, permita sincronizar la estructura de dichos objetos a una hora determinada.

I.1. Realidad problemática

Debido a que actualmente la Universidad Particular de Chiclayo cuenta con una base de datos centralizada y a la lentitud de la red, cada vez que se quiere hacer la consulta de las notas y/o asistencias de los estudiantes desde una sucursal distinta, muchas veces la información solicitada no es mostrada o llega con bastante tiempo de retraso por ello es necesario la implementación de un comparador sincronizador de bases de datos relacionales como una solución ante este problema.

Al momento de llevar a cabo esta investigación se tuvo que evaluar el tipo de sincronización que se tendrá que realizar y la forma en que se realizará, se llegó a la conclusión de que el tipo de replicación más conveniente es la de Master-to-Master para que las bases de datos trabajen al 100%, debido a que es necesario que se pueda leer y modificar la información en todas las sucursales, por otro lado la forma en que se llevará a cabo la sincronización es asíncronamente ya que es necesario que se realice el intercambio de información entre las bases de datos a una hora del día en que la red sea más rápida con el fin de evitar problemas en la escritura de datos.

Esta nueva alternativa, no convencional, es muy útil también para mantener segura la información de los estudiantes ya que ante cualquier desastre que ocurra en una de las bases de datos su información no será perdida debido a que también se encontrará disponible en las bases de datos de las otras sedes.

Es importante mencionar que las herramientas que se utilizarán para realizar la implementación del comparador sincronizador, tales como el sistema operativo, el gestor de bases de datos, todos ellos son open-

source por lo que no es necesario pagar licencia por alguno de ellos, con lo que se demostrará que no es necesario realizar grandes inversiones económicas para obtener una alta disponibilidad de la información.

Como contribución de esta investigación será la de poder implementarse no solo en los servidores de la universidad particular de Chiclayo sino también en cualquier empresa con sucursales en lugares alejados y que se le complique el acceso a la data de su base de datos centralizada ya por encontrarse muy alejada o por una conexión de red defectuosa, además si se desea contar con una alta disponibilidad de su información, así como tener un respaldo de la data de sus bases de datos este método es una de las mejores opciones a utilizar no solo por su bajo coste sino también porque por medio de esta investigación se brindará un manual muy detallado de cómo se realiza la implementación.

I.2. Formulación del problema

¿Qué tecnologías nos permitirá el diseño e implementación del comparador sincronizador de bases de datos relacionales en servidores de la Universidad Particular de Chiclayo para garantizar una alta disponibilidad de datos de los estudiantes universitarios?

I.3. Hipótesis

La lentitud de la red afecta el tiempo de respuesta de una consulta hecha a una base de datos.

El mayor problema de lentitud de respuesta de la base de datos se encuentra en la Ciudad Universitaria de la Universidad Particular de

Chiclayo ya que se encuentra a gran distancia de la base de datos central ubicada en el local de Miraflores.

Es posible que un comparador sincronizador, tenga la capacidad de evitar que haya tiempo de demora de respuesta al realizar una consulta, y mantener una alta disponibilidad de las asistencias y/notas de los estudiantes en todas las sedes principales de la Universidad Particular de Chiclayo.

I.4. Objetivos

Los objetivos de este trabajo de investigación se pueden dividir en dos grupos: objetivos generales y en objetivos específicos.

I.4.1. Objetivo general

Realizar el diseño e implementación de un comparador y sincronizador de bases de datos relacionales que permita tener una alta disponibilidad de las asistencias y notas de los estudiantes en todas las sedes de la Universidad Particular de Chiclayo.

I.4.2. Objetivo específico

- Instalación del sistema operativo CentOS 6 en los servidores que se utilizarán en las sedes de la Universidad Particular de Chiclayo.
- Analizar e Identificar como está estructurada y configurada la red en los locales de la Universidad Particular de Chiclayo.
- Realizar configuración básica los de servidores web Apache y de bases de datos PostgreSQL 8.0.

- Realizar las pruebas necesarias para confirmar que se realice correctamente la sincronización y que se garantice la disponibilidad de la información de las bases de datos.

II. BASES TEÓRICAS

II.1. Antecedentes de la Investigación

Actualmente existen investigaciones y estudios relacionados con el tema de comparador y sincronizador de bases de datos los cuales serán mencionados a continuación con un breve resumen de la investigación y las personas que realizaron dicho estudio.

Autor : Alejandra Zavala Mendoza
Fecha : Agosto del 2010
Nombre del proyecto : “Replicación y fragmentación de bases de datos distribuidas”
País : México
Lugar : Chihuahua, Coahuila de Zaragoza
Universidad : Universidad Veracruzana

Resumen

El uso y la necesidad de contar con bases de datos va creciendo día con día, ya que en cualquier empresa necesita guardar información ya sea sobre sus ventas, compras, inventarios, clientes, empleados, información financiera, etc. Y al no contar con esta información será muy difícil llevar un buen funcionamiento de cualquier organización.

Las bases de datos facilitan la búsqueda y el almacenamiento de toda la información, garantizando la fiabilidad de la información que esta contiene.

Las bases de datos distribuidas tienen múltiples ventajas. En primer lugar, los datos son localizados en lugar más cercano, por tanto, el acceso es más rápido, el procesamiento es rápido debido a que varios nodos intervienen en el procesamiento de una carga de trabajo, nuevos nodos se pueden agregar fácil y rápidamente. La comunicación entre nodos se mejora, los costos de operación se reducen, son amigables al usuario, la probabilidad de que una

falla en un solo nodo afecte al sistema es baja y existe una autonomía e independencia entre los nodos.

Los almacenamientos de datos son un elemento fundamental para las empresas por ello ha llamado la atención de estas ya que en la actualidad cualquier compañía por pequeña que sea tiene que manejar grandes cantidades de información, para poder tener control sobre todas las áreas de su empresa, es por ello que surge la necesidad de emplear bases de datos dentro de las organizaciones, no importando el giro de las empresas.

Por lo cual, se le denomina “Base de Datos” a la colección de datos aparentes usados por el sistema de aplicaciones de una determinada empresa.

La mayoría de las empresas necesitan contar con bases de datos ya que las bases de datos facilitan la realización de sus actividades, garantiza la seguridad, la confiabilidad de la información, consistencia de los datos, mejora en la disponibilidad de información, guarda información sobre sus clientes, proveedores, información financiera, inventarios, compras ventas etc. Si contamos con un buen control de la información se podrá llevar una empresa de mejor manejar y más organizadamente. La gran cantidad de información que producen grandes empresas geográficamente distribuidas, ha generado una gran dispersión de los datos asociados a sus sistemas de información. Para compartir dichos datos se requiere de tecnología capaz de integrarlos para permitir el acceso concurrente de múltiples usuarios. Una de las tecnologías que se propone alcanzar este tipo de objetivos son las bases de datos distribuidas

Las bases de datos distribuidas mejoran la disponibilidad de los datos pues permite que el sistema pueda seguir operando y los usuarios desde cualquier sitio puedan acceder a la información, exactamente igual como si los datos estuvieran en el mismo sitio del usuario.

Las bases de datos, hoy en día, ocupan un lugar determinante en cualquier área ya que se emplea en diversas empresas no importando el giro de esta, es por ello que deben de tener los conocimientos necesarios para poder utilizar las bases de datos.

Las bases de datos distribuidas se conectan por medio de redes de computadoras, estas pueden tener cientos de usuarios conectados concurrentemente, por lo cual es un problema en muchas empresas que compiten en los mercados mundiales. Debido al alto grado de conectividad que provee Internet a las empresas, se debe tener un buen diseño de replicación y fragmentación.

Este problema consiste en tratar de determinar la mejor forma de dividir las relaciones de la base de datos en fragmentos. La fragmentación mejora la disponibilidad de la información notablemente por que el sistema puede seguir operando mientras, por lo menos, uno de los sitios está activo, con la replicación también se mejora el rendimiento de la recuperación en consultas globales, porque el resultado de semejante consulta se puede obtener localmente en cualquier sitio, esto ayuda a contar con la información necesaria de una forma más fácil y rápida y hoy en día para las empresas contra con la información en un corto tiempo es un factor clave para la toma de decisiones oportunas.

Para contar con una buena base de datos distribuida esta debe de ser replicada y fragmentada. La replicación y fragmentación permite tener disponible la información, reduce costos, mejora el rendimiento de la operación leer, aumenta la flexibilidad, se obtiene la información de una manera más fácil y rápida teniendo un mejor procesamiento de las consultas.

Mediante la replicación de información, las bases de datos distribuidas pueden presentar cierto grado de tolerancia a fallos haciendo que el funcionamiento del sistema no dependa de un solo lugar como en el caso de las bases de datos centralizadas.

Autor	: Milán Franco, Jesús Manuel
Fecha	: 30 de junio 2008
Nombre del proyecto	: “Replicación Autonómica de Bases de Datos”
País	: España
Lugar	: Madrid, C/ Ramiro de Maeztu

Resumen

En los últimos años, la evolución de los sistemas informáticos ha creado sistemas cada vez más complejos que constan de multitud de elementos interconectados a través de redes de comunicaciones rápidas.

La configuración óptima de cada uno de estos elementos para obtener el máximo rendimiento del sistema depende de un gran número de factores como son la carga del sistema, el tipo de carga y el hardware sobre el que se ejecuta, entre otros. Esta diversidad de parámetros de configuración hace que la administración de dichos sistemas sea cada vez más complicada y que requiera administradores de sistemas con mucha experiencia y dedicación a estas tareas.

Dichos sistemas, además, están sujetos a cambios de diversa naturaleza como son variaciones en la carga del sistema, en el tipo de carga, en los recursos disponibles; fallos y recuperaciones de los elementos del sistema, etc. Por esta razón ha surgido la necesidad de crear sistemas que sean capaces de adaptarse de forma automática a los cambios en el entorno que les rodea sin necesidad de intervención humana y que al mismo tiempo exhiban un alto rendimiento y calidad de servicio. Estos sistemas adaptables dinámicamente interaccionan con su entorno para detectar los cambios en el mismo, analizan la información obtenida para generar configuraciones óptimas a las condiciones actuales del entorno y modifican su configuración para adaptarse a estos cambios.

El objetivo de estos sistemas es maximizar el rendimiento del sistema de acuerdo a las métricas de rendimiento de interés. En estos sistemas distribuidos, la replicación de datos juega un papel muy importante como forma de aumentar el rendimiento. Hasta ahora la replicación se implementaba dentro de la propia base de datos, lo que obligaba a realizar modificaciones en la base de datos y estos no siempre eran posibles. Una línea de investigación nueva iniciada en el LSD y que ha sido adoptada por

gran parte de los investigadores es la implementar la lógica de la replicación fuera de la base de datos a nivel de middleware.

Este trabajo de investigación se enmarca dentro de esta línea de investigación y en él se estudia la adaptabilidad dinámica en el contexto de las bases de datos replicadas basadas en middleware. La adaptabilidad del sistema replicado afecta principalmente a dos áreas de trabajo: La adaptación local (el control del nivel de concurrencia) y la adaptación global (el equilibrado de carga). El objetivo de la adaptación local es conseguir el máximo rendimiento de cada réplica individualmente sin llegar a sobrecargarlas, mientras que el objetivo de la adaptación global es conseguir que la carga del sistema esté equilibrada, evitando de esta forma que algunas réplicas estén saturadas, mientras que otras estén ociosas.

El trabajo realizado en esta tesis desarrolla un sistema de replicación adaptable a nivel de middleware que englobe ambas áreas de trabajo desarrollando protocolos y algoritmos para el control de la concurrencia a nivel local y el equilibrado de carga a nivel global y que maximizan de forma automática el rendimiento del sistema bajo distintos tipos de carga, fallos y recuperación de réplicas.

Autor : David Freire V
Fecha : 2010
Nombre del proyecto : “Replicación Síncrona master to master de bases de datos mysql en red hat Enterprise Linux o compatibles”
País : Ecuador
Lugar : Guayaquil, Guayas
Universidad : Universidad Pública de Navarra

Resumen

El objetivo es realizar una replicación de Base de Datos MySQL, que optimice los recursos de la empresa al usar completamente ambas bases de datos, ya que, normalmente en los esquemas de replicación Amo a Esclavo (Master to Slave) se tiene una funcionando y la otra sirve de repositorio de información y de contingencia en caso de que la base principal deje de ser funcional, y no se logra explotar al máximo esta base.

En el esquema actual, Amo a Amo (Master to Master), ambas bases trabajan de igual forma y se logra aprovechar todos los recursos de las misma, no dejando de contener el nivel de contingencia que posee la replicación tradicional Amo a Esclavo (Master to Slave).

También se busca mantener el criterio de Alta disponibilidad, al realizar una replicación Síncrona, es decir, todos los cambios a nivel de inserción, actualización y eliminación de registros que se realicen a cualquiera del base de datos se replicaran automáticamente en la otra base de datos y viceversa.

Autor : Giancarlo Roberto Calderón Garay
Fecha : Abril del 2009
Nombre del proyecto : “Análisis, Diseño e implementación de un comparador y sincronizador de bases de datos relacionales de distintos manejadores”
País : Perú
Lugar : Lima, San Miguel.
Universidad : Pontificia Universidad Católica Del Perú
Resumen

El presente proyecto consiste en el análisis, diseño e implementación de un sincronizador de bases de datos relacionales de distintos manejadores, cuya finalidad es realizar la comparación de objetos entre dos bases de datos y sincronizar dichos objetos de acuerdo a las diferencias encontradas.

Para la gestión del proyecto se ha seguido las prácticas recomendadas por PMI y se han considerado aquellos procesos de gestión necesarios para el desarrollo del proyecto. Para el desarrollo de la herramienta se optado por utilizar la metodología AUP (cuyas siglas en inglés significan Agile Unified Process), cuyas fases y disciplinas se han adaptado mejor al desarrollo de la aplicación.

La arquitectura seleccionada ha permitido que la aplicación pueda trabajar con distintos manejadores de bases de datos relacionales. Para llevar a cabo este objetivo se ha implementado un componente que se encarga de realizar la abstracción de los manejadores e interactúa con las demás capas de manera transparente.

La implementación de esta componente se ha realizado a través de archivos XML que, con una estructura definida, permiten que la aplicación consulte la metadata de la base de datos y construya las sentencias SQL para la sincronización de objetos.

De esta manera, si se desea incorporar otra base de datos sólo se necesita

definir el contenido de la plantilla XML y los parámetros de conexión que van definidos en un archivo de configuración. El proceso de sincronización implementado se puede realizar de dos maneras: mediante una comparación previa de los objetos a sincronizar o mediante un asistente de sincronización, el cual permite al usuario seleccionar de manera más personalizada qué objetos desea sincronizar sin importar qué diferencias existen con los objetos de la base de datos destino.

En ambos escenarios, se genera un archivo de bitácora del proceso de sincronización donde se puede verificar los resultados del proceso.

II.2. Marco teórico

El proceso de Replicación Master-to-Master o Master-to-Slave sobre PostgreSQL, actualmente no está muy propagado por nuestra localidad, ni se encuentran estudios detallados de cómo realizar el mismo, aunque existe un sin número de documentación de procesos de replicación en el internet.

Será necesario plantear algunos parámetros que sirvan de ejes conceptuales sobre los que apoyar la lectura interpretativa de este proyecto de investigación. Para empezar, entenderemos el concepto de replicación en bases de datos relacionales, sus beneficios, las formas en las que se puede dar la sincronización, la herramienta que se utilizará para llevar a cabo este proceso, el gestor de bases de datos con el que se trabajará para la administración de las bases de datos, el sistema operativo sobre el que se llevará a cabo todo el proceso, entre otros conceptos de protocolos y técnicas utilizadas.

Replicación de bases de datos relacionales.

Es el proceso de copiar y mantener objetos de las bases de datos en múltiples bases de datos que forman un sistema de bases de datos distribuido. La replicación permite que los datos de un servidor de bases de datos (el maestro), sean replicados en uno o más servidores de bases de datos (los esclavos).

[CITATION Ale12 \l 10250]

La replicación es la copia sincronizada entre dos servidores de bases de datos de forma que cualquiera de los dos puede entregar los mismos resultados a sus clientes.

[CITATION Par05 \I 10250]

La replicación de datos se refiere al almacenamiento de copias de datos en múltiples sitios conectados mediante una red de ordenadores. Pueden guardarse copias fragmentadas en varios sitios para satisfacer requerimientos de información específicos. Como la existencia de copias de fragmentos puede mejorar la disponibilidad de los datos y el tiempo de respuesta, estas copias reducen los costes de comunicación y de consultas totales

[CITATION Lau10 \I 10250]

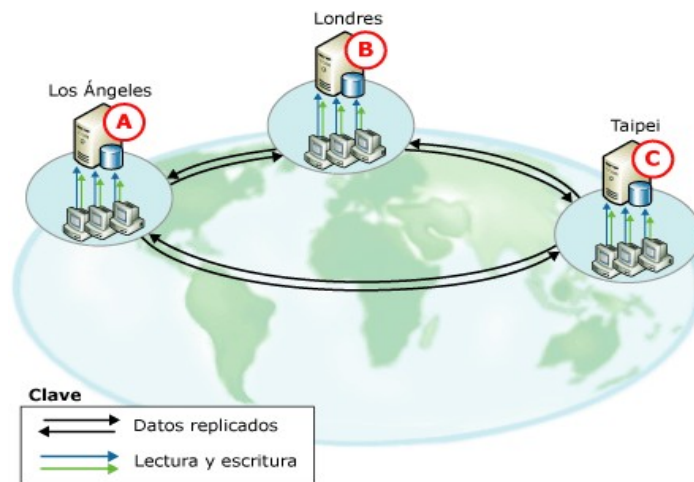


Imagen 1 Ejemplo replicación en diferentes servidores

Herramienta de replicación que se implementará: “Bucardo”.

Bucardo es un sistema de replicación asíncrono que en otras palabras permite "clonar" o "duplicar" registros desde de una Base de Datos (fuentes-objetivos), además cuenta con soporte para manejar múltiples Base de Datos con relaciones maestros-esclavos.

[CITATION And15 \I 10250]

Gestor de bases de datos relacionales PostgreSQL.

PostgreSQL es un sistema de gestión de bases de datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarles a otras bases de datos comerciales.

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando.

[CITATION Jac \I 10250]

Una base de datos es un conjunto de datos relacionados entre sí. Por datos se entienden hechos conocidos que pueden registrarse y que tienen un significado implícito. Por ejemplo, considérense los nombres y números de teléfonos de personas. Tal vez estos datos estén registrados en una libreta de direcciones indexada, o quizá en un disquete, utilizando un ordenador y un software. Se trata de un conjunto de datos relacionados entre sí y que tienen un significado implícito; por tanto, constituyen una base de datos.

[CITATION Lau10 \I 10250]

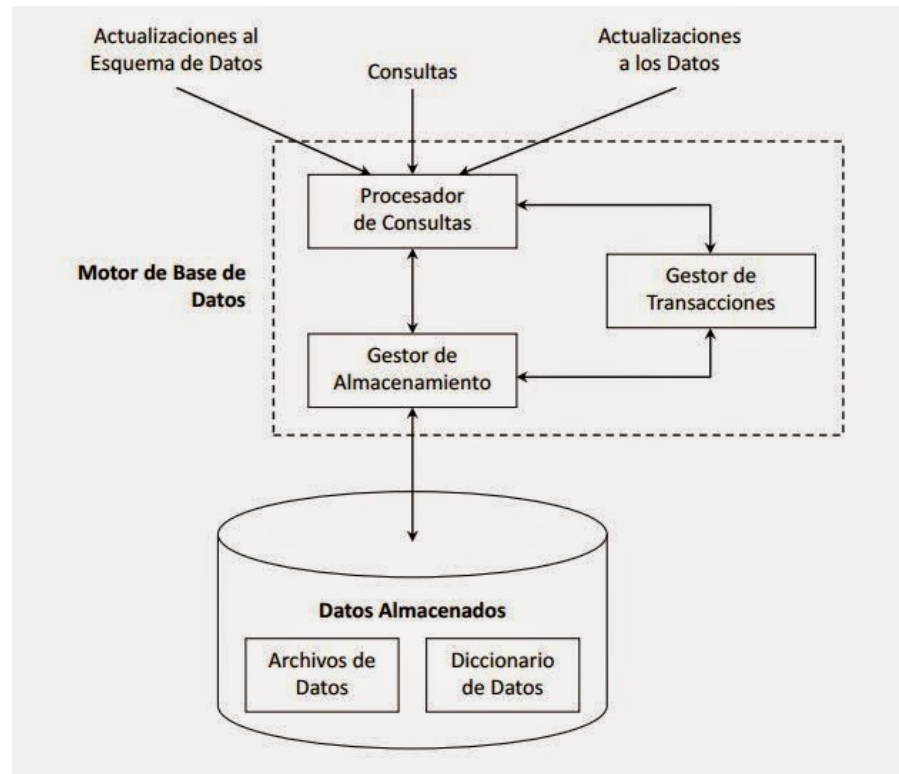


Imagen 2 Estructura PostgreSQL

CentOS.

Es un sistema operativo de código abierto, basado en la distribución Red Hat Enterprise Linux, operándose de manera similar, y cuyo objetivo es ofrecer al usuario un software de "clase empresarial" gratuito. Se define como robusto, estable y fácil de instalar y utilizar.

[CITATION Ber08 \l 10250]

VMWare Workstation

VMWare Workstation es un hipervisor alojado que se ejecuta en versiones x64 de sistemas operativos Windows y Linux (una versión x86 de versiones anteriores estaba disponible); permite a los usuarios configurar máquinas virtuales (VM) en una sola máquina física, y utilizarlas simultáneamente junto con la máquina real. Cada máquina virtual puede ejecutar su propio sistema operativo, incluyendo versiones de Microsoft Windows, Linux, BSD y MS-DOS. VMWare Workstation es desarrollado y vendido por VMWare, Inc., una división de Tecnologías Dell. Hay una versión gratuita, VMWare Workstation Player, para uso no comercial. Una licencia de sistemas operativos es necesaria para utilizar los propietarios como Windows. Las VM de Linux preparadas para diferentes propósitos están disponibles en varias fuentes.

VMWare Workstation soporta la conexión de adaptadores de red de host existentes y el intercambio de unidades de disco físicas y dispositivos USB con una máquina virtual. Puede simular unidades de disco; un archivo de imagen ISO puede montarse como una unidad de disco óptico virtual, y las unidades de disco duro virtuales se implementan como archivos. vmdk.

VMWare Workstation Pro puede guardar el estado de una máquina virtual (una "instantánea") en cualquier instante. Estas instantáneas se pueden restaurar más tarde, efectivamente devolver la máquina virtual al estado guardado, tal como estaba y libre de cualquier daño posterior a la instantánea de la máquina virtual.

VMWare Workstation incluye la capacidad de agrupar varias máquinas virtuales en una carpeta de inventario. Las máquinas de dicha carpeta se pueden encender y apagar como un único objeto, útil para probar complejos entornos cliente-servidor.

[CITATION VMw09 \l 10250]

Rubyrep

es una herramienta hecha en Ruby (como su nombre lo sugiere) para la sincronización de base de datos que se ha venido usando últimamente en este proyecto de investigación, destaca debido a la sencillez de su implementación y la versatilidad de formas en que puede ser usado.

[CITATION Ric11 \l 10250]

II.3. Definición de términos

Máquina virtual. -

Entendamos por una máquina virtual a un Software que simula a una computadora y puede instalar y usar otros sistemas operativos de forma simultanea como si fuese una computadora real sobre nuestro sistema operativo. Una característica esencial de las máquinas virtuales es que los procesos que ejecutan están limitados por los recursos y abstracciones proporcionados por ellas. Estos procesos no pueden escaparse de esta "computadora virtual".

Sistema operativo. -

Es el software que se ejecuta en modo kernel. Parte del problema es que los sistemas operativos realizan dos funciones básicas que no están relacionadas: proporcionar a los programadores de aplicaciones (y a los programas de aplicaciones, naturalmente) un conjunto abstracto de recursos simples, en vez de los complejos conjuntos de hardware; y administrar estos recursos de hardware. Dependiendo de quién se esté hablando, el lector

podría escuchar más acerca de una función o de la otra. Ahora analizaremos ambas.

Linux. -

Estrictamente hablando, Linux es sólo el núcleo (kernel) de un sistema operativo tipo Unix. Fue desarrollado por el finlandés Linus Torvalds en 1991 a partir de Minix, un sistema Unix mínimo, desarrollado con fines didácticos por A. Tanenbaum. El anuncio se hizo en el grupo de noticias comp.os.minix y permitió la participación de numerosos usuarios y programadores de todo el mundo a través de Internet. El kernel alcanzó una estabilidad a nivel de usable en 1994. A partir de entonces Linux se convierte en el complemento ideal del proyecto GNU, al que solo le faltaba un núcleo estable.

CentOS. -

CentOS Linux es una distribución mantenida por la comunidad y derivada de los paquetes fuentes liberados al público por Red Hat para Red Hat Enterprise Linux (RHEL). De tal forma, CentOS Linux está enfocado en ser operacionalmente compatible con RHEL. El Proyecto CentOS principalmente cambia paquetes para eliminar las marcas comerciales y trabajos artísticos de Red Hat. La redistribución de CentOS Linux es libre y no hay que pagarlo.

Open-source. -

es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones éticas y morales las cuales destacan en el llamado software libre.

Base de datos. -

Consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a dichos datos. La colección de datos, normalmente denominada base de datos, contiene información relevante para una empresa. El objetivo principal de un SGBD es proporcionar una forma de almacenar y recuperar la información de una base de datos de manera que sea tanto práctica como eficiente.

Manejador de base de datos. -

Un sistema manejador de bases de datos (SGBD, por sus siglas en inglés) o DataBase Management System (DBMS) es una colección de software muy específico, cuya función es servir de interfaz entre la base de datos, el usuario y las distintas aplicaciones utilizadas.

Como su propio nombre indica, el objetivo de los sistemas manejadores de base de datos es precisamente el de manejar un conjunto de datos para convertirlos en información relevante para la organización, ya sea a nivel operativo o estratégico.

PostgreSQL 8.0.-

PostgreSQL es un Sistema de gestión de bases de datos relacional orientado a objetos y libre, publicado bajo la licencia PostgreSQL, similar a la BSD o la MIT.

Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa o persona, sino que es dirigido por una

comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global Development Group).

Protocolo IP. -

El protocolo de Internet (en inglés Internet Protocol IP es un protocolo de comunicación de datos digitales clasificado funcionalmente en la capa de red según el modelo internacional OSI.

Su función principal es el uso bidireccional en origen o destino de comunicación para transmitir datos mediante un protocolo no orientado a conexión que transfiere paquetes conmutados a través de distintas redes físicas previamente enlazadas según la norma OSI de enlace de datos.

Protocolo HTTP. -

El Protocolo de Transferencia de HiperTexto (Hypertext Transfer Protocol) es un sencillo protocolo cliente-servidor que articula los intercambios de información entre los clientes Web y los servidores HTTP. La especificación completa del protocolo HTTP 1/0 está recogida en el RFC 1945. Fue propuesto por Tim Berners-Lee, atendiendo a las necesidades de un sistema global de distribución de información como el World Wide Web.

Interfaz de red. -

Para ocultar la diversidad de hardware que puede usarse en un entorno de red, TCP/IP define una interfaz abstracta a través de la cual se accede a dicho hardware. Esta interfaz ofrece un conjunto de operaciones que son las mismas para todos los tipos de hardware y básicamente trata con el envío y la recepción de paquetes.

DNS. -

El sistema de nombres de dominio (DNS, por sus siglas en inglés, Domain Name System) es un sistema de nomenclatura jerárquico descentralizado para dispositivos conectados a redes IP como Internet o una red privada. Este sistema asocia información variada con nombre de dominio asignado a cada uno de los participantes. Su función más importante es "traducir" nombres inteligibles para las personas en identificadores binarios asociados con los equipos conectados a la red, esto con el propósito de poder localizar y direccionar estos equipos mundialmente.

Servidor web APACHE. -

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.1 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. Su nombre se debe a que alguien quería que tuviese la connotación de algo que es firme y enérgico, pero no agresivo, y la tribu Apache fue la última en rendirse al que pronto se convertiría en gobierno de EEUU, y en esos momentos la preocupación de su grupo era que llegasen las empresas y "civilizasen" el paisaje que habían creado los primeros ingenieros de internet.

Tabla. -

Tabla en las bases de datos, se refiere al tipo de modelado de datos, donde se guardan los datos recogidos por un programa. Su estructura general se

asemeja a la vista general de un programa de hoja de cálculo.

Una tabla es utilizada para organizar y presentar información. Las tablas se componen de filas y columnas de celdas que se pueden rellenar con textos y gráficos.

Registro o fila en Base de datos. -

En informática, o concretamente en el contexto de una base de datos relacional, un registro (también llamado fila o tupla) representa un objeto único de datos implícitamente estructurados en una tabla. En términos simples, una tabla de una base de datos puede imaginarse formada de filas y columnas o campos. Cada fila de una tabla representa un conjunto de datos relacionados, y todas las filas de la misma tabla tienen la misma estructura.

Sincronización Asíncrona. -

La transmisión asíncrona tiene lugar cuando el proceso de sincronización entre emisor y receptor se realiza en cada palabra de código transmitido. Cada carácter a ser transmitido es delimitado por un bit de información denominado de cabecera o de arranque, y uno o dos bits denominados de terminación o de parada.

Sincronización Síncrona. -

La transmisión síncrona es una técnica que consiste en el enviar una trama de datos (conjunto de caracteres) que configura un bloque de información comenzando con un conjunto de bits de sincronismo (SYN) y terminando con otro conjunto de bits de final de bloque (ETB). En este caso, los bits de sincronismo tienen la función de sincronizar los relojes existentes tanto en el emisor como en el receptor, de tal forma que estos controlan la duración de

cada bit y carácter.

Bucardo. -

Bucardo es una herramienta que sirve para sincronizar y replicar bases de datos PostgreSQL, permitiendo que dicha sincronización se haga en formato Master-Slave, es decir una base de datos donde se actualiza la información y una o varias de sólo lectura que se sincronizan con la primera, o en formato Multi-Master, con dos bases de datos sobre las que se actualiza información y que se sincronizan entre sí. En la propia wiki de PostgreSQL podéis ver una tabla resumen con las distintas herramientas que hay para sincronización y replicación y los tipos que soportan cada una.

Bucardo sync Pushdelta. -

Es un tipo de sincronización que utiliza Bucardo la cual consiste en replicar las filas desde una base de datos (maestro) a una o más bases de datos. Por cada sincronización de este tipo se le debe indicar a Bucardo que tablas y secuencias se desea replicar, y a cuáles bases de datos se le replicaran los cambios.

Bucardo sync fullcopy. -

Es un tipo de sincronización que usa Bucardo el cual consiste en copiar una tabla entera desde una base de datos (la master) a una o más bases de datos esclavos. Cualquier fila almacenada en los esclavos primero son eliminadas, y posteriormente se insertan las filas del maestro mediante el comando COPY del lenguaje SQL. Este método de sincronización solo puede ser usado para tablas que no tienen una llave primaria o única. Por qué siempre copia la tabla entera, lo cual no es eficiente y no debe utilizarse para grandes tablas. Es preferible agregar a la tabla un índice único y luego usar el tipo de

sincronización **pushdelta**.

Bucardo sync Swap. –

Es un tipo de sincronización que utiliza Bucardo en la cual los cambios entre dos bases de datos son intercambiados en ambos sentidos, debido a que las tablas y secuencias son actualizadas en ambos lados, es necesario que haya una forma de resolver conflictos. En cada tabla o secuencia se especifica el tipo de resolución de conflictos que más convenga, puede ser uno estándar el cual viene incluido en el software de bucardo, o también es posible utilizar uno personalizado (debe ser hecho en el lenguaje de programación Perl).

Servidor de Base de datos. -

Servidores de Bases de Datos. También conocidos como RDBMS (acrónimo en inglés de Relational DataBase Management Systems), son programas que permiten organizar datos en una o más tablas relacionadas. Los servidores de Bases de Datos se utilizan en todo el mundo en una amplia variedad de aplicaciones.

PHP. -

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos. El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz

de línea de comandos que puede ser usada en aplicaciones gráficas independientes.

HTML. -

HTML, sigla en inglés de HyperText Markup Language (lenguaje de marcas de hipertexto), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia del software que conecta con la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, videos, juegos, entre otros.

postgresql.conf. -

En este fichero podemos cambiar todos los parámetros de configuración que afectan al funcionamiento y al comportamiento de PostgreSQL en nuestra máquina.

pg_hba.conf. -

Este fichero se utiliza para definir los diferentes tipos de accesos que un usuario tiene en el clúster.

pg_ident.co. -

Este fichero se utiliza para definir la información necesaria en el caso que utilicemos un acceso del tipo ident en pg_hba.conf.

Rpm Package Manager. –

Es una herramienta de administración de paquetes pensada básicamente para GNU/Linux. Es capaz de instalar, actualizar, desinstalar, verificar y solicitar programas.

Firewall Iptables. –

Iptables es un Firewall simple instalado en la mayoría de las distribuciones de Linux. Según la información del propio manual de Linux para iptables, esta es una herramienta de administración para la filtración y NAT de paquetes IPv4, en pocas palabras: es una herramienta para filtrar y bloquear tráfico de Internet o red local.

Virtual Host Apache. –

Un host virtual, o VirtualHost, en Apache nos permite mantener múltiples nombres de host en nuestro servidor.

Comando “chown”. –

Es un comando que viene incluido en todo sistema operativo GNU/Linux el cual permite modificar los usuarios o grupos dueños de un archivo o carpeta en el sistema de archivos.

Comando “chmod”. –

Es un comando que permite cambiar los permisos de acceso a las las carpetas o ficheros dentro de un sistema UNIX.

Perl. –

Es un lenguaje de programación utilizado para hacer scripts en servidores, es capaz de interpretar otros lenguajes de programación y agregarle librerías.

Cron. –

Es un programa que viene incluido en sistemas Linux/Unix el cual permite ejecutar automáticamente comando o scripts (grupo de comandos) a una hora o fecha específica. Es usado normalmente para comandos de tareas administrativas, como respaldos, pero puede ser usado para ejecutar cualquier cosa.

III. MARCO METODOLÓGICO

III.1. Variables

III.1.1. Variable Dependiente

Malestar causado a los alumnos y padres de familia por la falta de disponibilidad de las asistencias, notas, estados de cuenta y otras informaciones de los estudiantes que estén en la base de datos en las sedes de la Universidad de Particular de Chiclayo.

III.1.2. Variable Independiente

- Las prestaciones de la red son muy bajas y toleran poco tráfico debido al poco ancho de banda.
- El cableado de la red desordenado en su mayoría en mal estado o instalado de una forma poco óptima.
- Una base de datos centralizada ubicada en un lugar diferente y a una distancia considerablemente lejana.

III.2. Operacionalización de variables

VARIABLE	CONCEPTO	DIMENSION	INDICADOR
Prestaciones de la red	Velocidad y ancho de banda de la red local de la universidad particular de Chiclayo.	Velocidad de subida de datos Velocidad de bajada de datos-	- Mbps por segundo de subida - Mbps por segundo de bajada
Estado del cableado de red	Estado de deterioro en el que se encuentran las instalaciones físicas de la red LAN de la Universidad particular de Chiclayo.	Definición	- Nro Cables mal ponchados - Nro Cables pelados y sin protección electromagnética. - Escala de antigüedad de los quipos de ruteamiento (0-10)
Ubicación de la base de datos	Distancia a la que se encuentra la base de datos de los sistemas de información que hacen uso de ella.	Distancia desde la sede donde está la base de datos hasta la ciudad universitaria	- Kilómetros
Bases de datos	Son las bases de datos en las que se guarda datos y se accede desde los	Bases de datos centralizadas funcionando	- Nro de bases de datos centralizadas.

	sistemas de información		
--	----------------------------	--	--

Tabla 1 Operacionalización de Variables

III.3. Metodología

La metodológica de estudio será de tipo observacional, descriptiva, transversal y aplicable, en la cual se realizarán múltiples pruebas de simulación para tener un resultado satisfactorio y poder llegar al resultado deseado.

III.3.1. Tipo de Estudio

Por su Método de Investigación Documental y Campo (Teórica y Práctica):

Cuyo método de investigación se concentró en la recopilación de información en forma documental de replicación de bases de datos, respectivamente la recopilación de información se realizó enmarcado dentro del ambiente de redes LAN, uso del sistema operativo CentOS 6 (Linux/fedora), configuración de servidores web, servidores de bases de datos, el funcionamiento, configuración de servidores de replicación de bases de datos, topología de la red en las sedes de la Universidad de Particular de Chiclayo, y el uso de la herramienta de replicación BUCARDO, etc., en el que se presentó un fenómeno o hecho que lo estudiamos los cuales son la utilización del proceso de replicación para copiar y mantener actualizados los datos en varios nodos o servidores de bases de datos en las diferentes sedes de la UDCH

III.3.2. Diseño de Investigación

En este punto a tratar se definirá el estado de las fases a través de las cuales se ira moviendo el presente proyecto de investigación.

Se llegó a la conclusión que, para la realización de este proyecto, se utilizará

un Modelo Incremental, ya que se partirá de un Análisis, Diseño, Configuración y sus respectivas pruebas, constituyendo así el primer avance o incremento, luego de esto se deberá presentar al cliente que hará uso de este proyecto, quien dará sus opiniones, sugerencias lo cual servirá para partir al segundo avance o incremento y mejora de este proyecto. Cabe resaltar que el presente proyecto de investigación queda a disposición para que otros profesionales prosigan una implementación más minuciosa e ir mejorando este proyecto para futuras aplicaciones.

Se plantea una modalidad PROYECTO FACTIBLE, con un 30% Bibliografía, 30% Investigación y 40% propuesta, debido a que la alternativa de solución bosquejada implica investigaciones, pruebas y soluciones con bajo costo, asequibles a empresas de bajos recursos o que recién se encuentran iniciando sus actividades.

III.4. Población, Muestra, Muestreo.

Población

Para esta investigación se tomará como parte de la población a todo aquel cliente que pertenezca a la Universidad Particular De Chiclayo y que quiera realizar una matrícula, consultar estado de notas, de cuenta o cualquier otra información que esta almacenada en la base de datos.

Muestra

Debido a que la población de clientes de la Universidad Particular De Chiclayo que quiera realizar una matrícula, consultar estado de notas, de cuenta o

cualquier otra información es muy extensa, realizaremos el cuestionario a veinte

Clientes que quieran solicitar alguna información, es decir nuestra muestra será de veinte para la primera encuesta, mientras que para la segunda encuesta las muestras se elegirán a treinta clientes que quieran solicitar alguna información.

III.5. Aspectos éticos

La ética informática es considerada como la disciplina que analiza problemas éticos que son creados por la tecnología de los ordenadores o también los que son transformados o agravados por la misma, es decir, por las personas que utilizan los avances de las tecnologías de la información. La ética informática estaría relacionada con los problemas conceptuales y los vacíos en las regulaciones que ha ocasionado la tecnología de la información.

El problema en cuestión es que hay una falta de reglamentación en cómo utilizar estas nuevas tecnologías que posibilitan nuevas actividades para las cuales no hay o no se perciben con nitidez principios de actuación claros. Las personas con responsabilidades en el área de diseño o gestión de sistemas de información cada vez han de tomar más decisiones sobre problemas que no se resuelven con lo legal y lo Cuasi-legal (reglamentos, manuales de procedimiento de las empresas, etc.) sino que rozan lo ético mismo. La tarea de la EI es aportar guías de actuación cuando no hay reglamentación o cuando la existente es obsoleta.

Para que la investigación concluya y se sustente en los principios de la ética, cuando los sujetos de estudio sean personas, se tendrá en cuenta el consentimiento previo de los mismos para participar, tomándose en cuenta

todos los aspectos establecidos al respecto.

Debe tenerse en cuenta en este aspecto si las políticas públicas hacen posible el desarrollo de la investigación, si es factible estudiar el fenómeno en cuestión, si se cuenta con los recursos necesarios para la misma, si los investigadores son competentes para realizar ese tipo de estudio, si es pertinente y luego el consentimiento informado de las personas implicadas en la investigación.

IV. RESULTADOS

Ya obtenida toda la información requerida en la parte anterior, a través de una serie de encuestas se realizó una evaluación de la situación actual a nivel de equipos, configuración y distribución. Estas encuestas fueron realizadas a clientes que usan este servicio en la Universidad Particular de Chiclayo.

En esta etapa de nuestra investigación daremos a conocer la situación actual en que se encuentra la Universidad Particular de Chiclayo, con el fin de encontrar resultados los cuales ayudaran para la elaboración de la propuesta para lograr mejorías en la institución, se realizó dos encuestas a los sectores que inciden de manera directa.

A continuación, se mostrarán los resultados obtenidos en las dos encuestas aplicadas a los clientes que hacen uso de este servicio, con estos resultados llegaremos a conocer la opinión de estos y a su vez comparar un antes y un después de haber realizado implementación del comparador y sincronizados de base de datos relacionales en tiempo real.

PRIMERA ENCUESTA

1era Pregunta

a)	13	ptos
b)	7	ptos
c)	0	ptos

6ta Pregunta

a)	17	ptos
b)	3	ptos

2da Pregunta

a)	3	ptos
b)	2	ptos
c)	15	ptos

3ra Pregunta

a)	18	ptos
b)	2	ptos
c)	0	ptos
d)	0	ptos

4ta Pregunta

a)	16	ptos
b)	3	ptos
c)	1	ptos
d)	0	ptos

5ta Pregunta

a)	5	ptos
b)	8	ptos
c)	3	ptos
d)	2	ptos
e)	2	ptos

Imagen 3 Resultados Primera Encuesta

SEGUNDA ENCUESTA

1era pregunta

a)	0	ptos
b)	1	ptos
c)	19	ptos

2da pregunta

a)	20	ptos
b)	0	ptos
c)	0	ptos

3ra pregunta

a)	20	ptos
b)	0	ptos
c)	0	ptos
d)	0	ptos

4ta pregunta

a)	0	ptos
b)	0	ptos
c)	1	ptos
d)	19	ptos

Imagen 4 Resultados Segunda Encuesta

Como segunda fase para la obtención de los resultados de las encuestas aplicadas, procederemos a tabular las respuestas empleando la herramienta de cálculo Microsoft Excel 2016 con la cual podremos obtener de una forma ordenada y de fácil entendimiento todos los resultados obtenidos en las encuestas. Los resultados serán mostrados en forma de gráficos para que sea de fácil lectura y así poder sacar conclusiones con mayor eficacia.

IV.1. Resultados encuesta 01

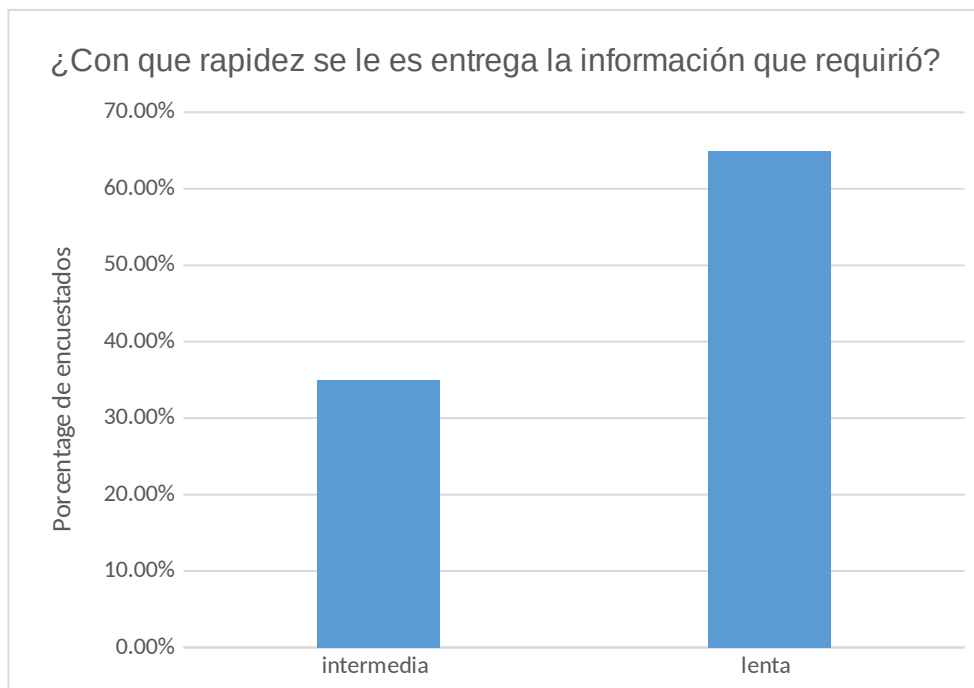


Tabla 2 Resultado Grafico a Pregunta 01 de la primera encuesta

¿Que tan difícil se le es tener que ir a la sede principal para requerir información?

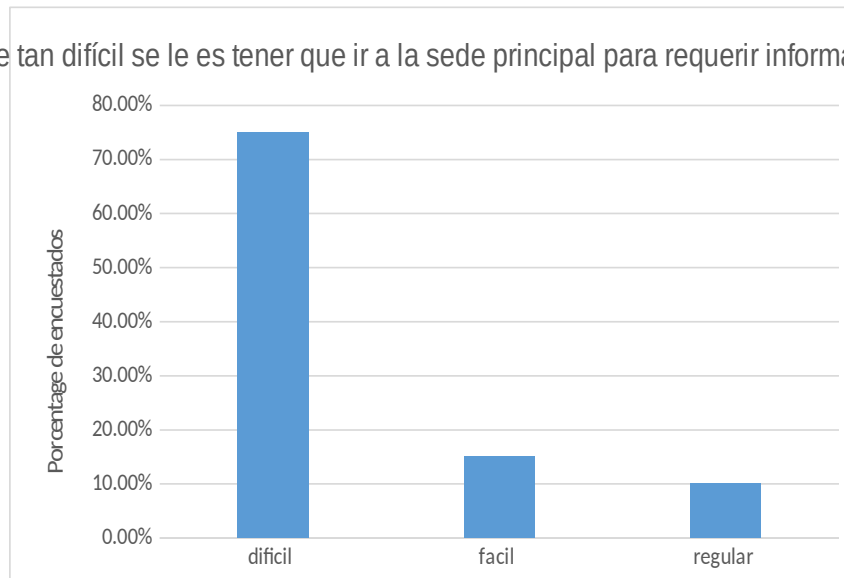


Tabla 3 Resultado Grafico a Pregunta 02 de la primera encuesta

¿Qué tan buena le parece la idea de que haya en todas las sedes de la Universidad Particular de Chiclayo toda la información que usted requiere?

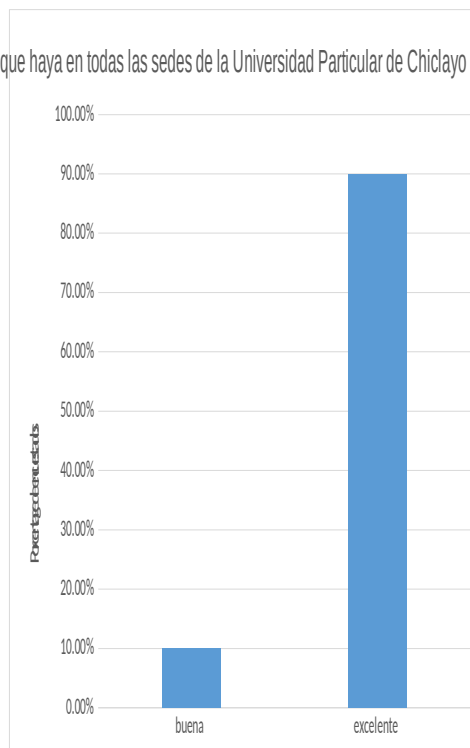


Tabla 4 Resultado Grafico a Pregunta 03 de la primera encuesta

¿En que estado cree usted que se encuentra la red local de la UDCH?

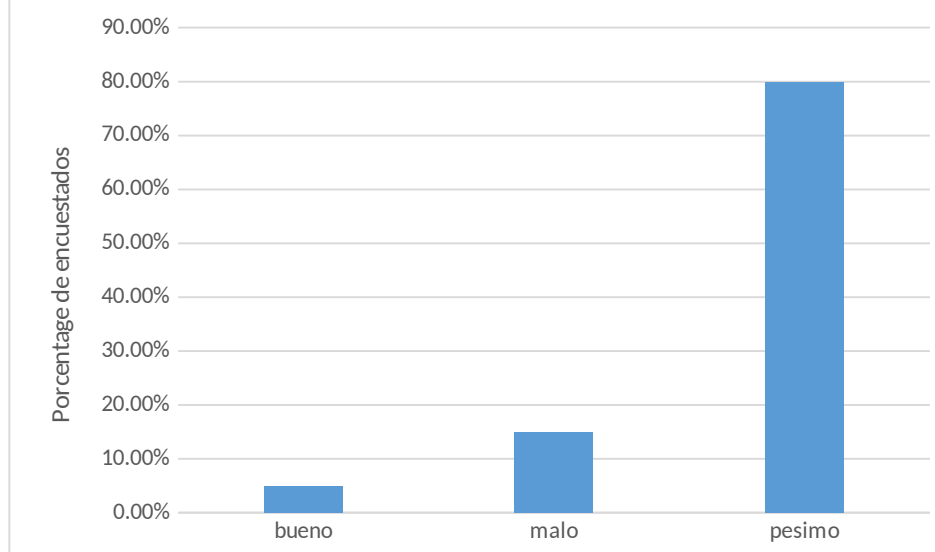


Tabla 5 Resultado Grafico a Pregunta 04 de la primera encuesta

¿Cuántas veces lo han regresado por no haber sistema disponible?

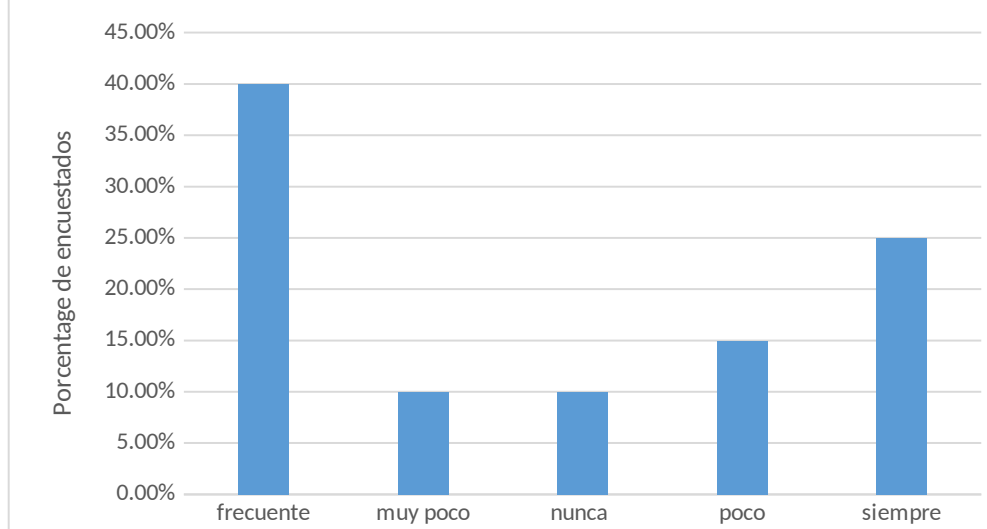


Tabla 6 Resultado Grafico a Pregunta 05 de la primera encuesta

¿Cree usted que si se implementa un comparador y sincronizador de bases de datos se podría mejorar la calidad del servicio que se le brinda?

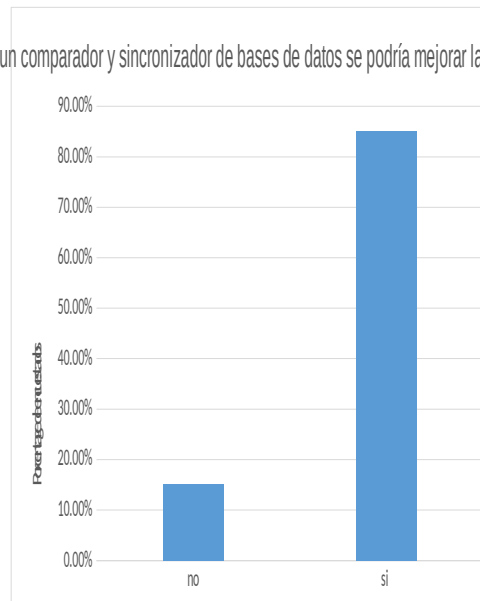


Tabla 7 Resultado Grafico a Pregunta 06 de la primera encuesta

En los siguientes cuadros que se muestran anteriormente se puede observar de manera gráfica y en forma porcentual la molestia que en muchos casos tienen los clientes que usan este servicio, también podemos darnos cuenta las causas de estas molestias.

Basándonos en los resultados obtenidos por las encuestas hechas a los clientes de la Universidad Particular de Chiclayo, podemos apreciar que la falta de disponibilidad de información es un problema muy común el cual está causando un constante malestar a los que solicitan esa información que se encuentra guardada en la base de datos central.

Como resultado estas cifras nos dan un indicador muy alto y a su vez alarmante, ya que se observa niveles elevados de insatisfacción y molestias por parte de los clientes que usan este servicio y que es necesario con carácter de urgencia tomar medidas drásticas para poder dar una solución a este caos y poder terminar con esto de una vez.

IV.2. Resultados encuesta 02

Al momento de simular la replicación ¿Con que rapidez se le es entrega la información que requirió?

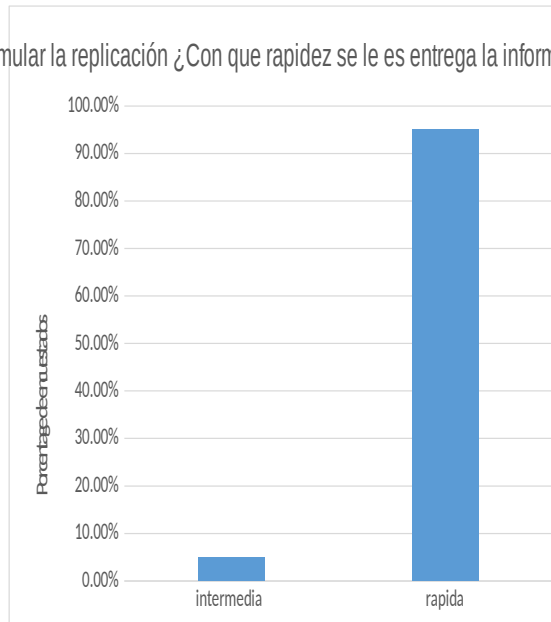


Tabla 8 Resultado Grafico a Pregunta 01 de la segunda encuesta

Teniendo en cuenta el ejemplo de simulación ¿Se le haría más fácil si hubiera replicación y sincronización de las bases de datos entre todas las sedes de la Universidad Particular de Chiclayo?

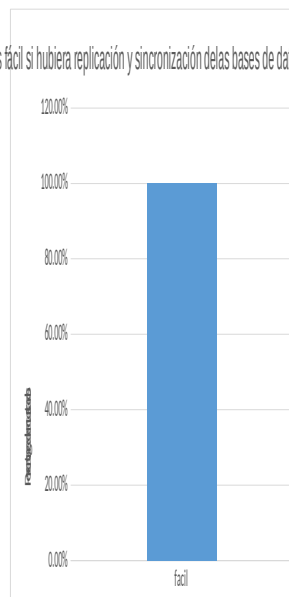


Tabla 9 Resultado Grafico a Pregunta 02 de la segunda encuesta

Con este ejemplo de simulación de replicación y sincronización de base de datos ¿Qué tan buena le parece la idea de que haya en todas las sedes de la Universidad Particular de Chiclayo toda la información que usted requiere?

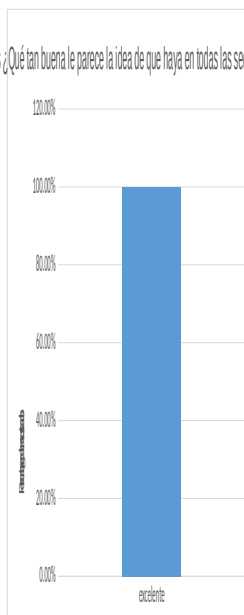


Tabla 10 Resultado Gráfico a Pregunta 03 de la segunda encuesta

¿Después de haber visto cómo funciona el comparador y sincronizador y teniendo en cuenta la eficacia que se obtuvo califique la forma en que se desempeñó?

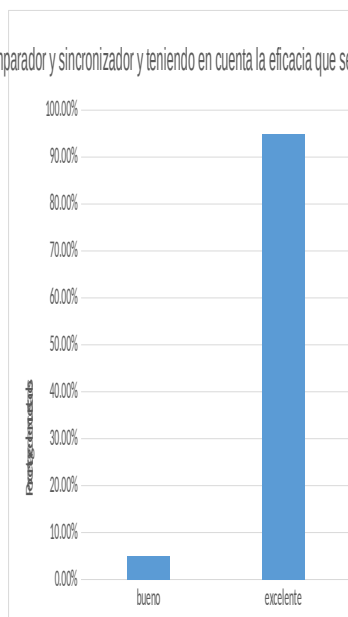


Tabla 11 Resultado Gráfico a Pregunta 04 de la segunda encuesta

Después de haber obtenido los resultados de la segunda encuesta, se tabularon en los cuadros que se muestran anteriormente, se puede observar que las mejoras son notorias de una manera resaltante en todos los sentidos

al realizar la simulación de replicación y sincronización de base de datos relacionales en tiempo real.

Basándonos en los resultados obtenidos en la segunda encuesta realizada a los clientes de la Universidad Particular de Chiclayo, podemos darnos cuenta de manera notoria que al realizar la simulación los problemas que sucedían en la mayoría de casos desaparecen, además que ayuda en la rapidez de los tramites y los clientes se van agradecidos por la rapidez del servicio brindado.

Como resultado estas cifras nos dan un indicador muy alto, ya que se observa niveles elevados de satisfacción por parte de los clientes que usan este servicio.

V. DISCUSIÓN

Esta investigación tuvo como propósito identificar y describir aquellas causas que impiden que se pueda visualizar sin tanta demora la información de los estudiantes mediante los sistemas de información de la universidad particular de Chiclayo, así como también sugerir una solución que sea efectiva, económica, y que no sea necesario realizar cambios drásticos y costosos en la arquitectura de red de los locales de la universidad particular de Chiclayo.

De los resultados obtenidos en esta investigación, se puede deducir que los principales causantes de que los sistemas de información muestren la información con un tiempo de retraso, es el mal estado en que se encuentra el cableado de red y el considerable deterioro que presentan los equipos encargados de la transmisión y ruteamiento de datos por la red.

Por otro lado, el proyecto se trabajó sobre el ambiente Linux, el proyecto demanda de mucha RAM (Memoria de acceso aleatorio) ya que el servidor antepondrá las necesidades de sincronización y replicación en lugar del rendimiento de las aplicaciones. Lo que reduce el tiempo de respuesta de los usuarios.

Se buscó una hora del día en el que haya poca saturación de la red para empezar el proceso de sincronización entre las bases de datos, este tipo de sincronización es llamada comúnmente como replicación asíncrona lo que nos permite sincronizar la data de las bases de datos a una hora en que se determinó que haya un poco tráfico en la red para evitar pérdida de información en la replicación.

El proceso de replicación se llevó a cabo mediante la virtualización de sistemas operativos sobre los cuales se desempeña todo el proyecto, para esta finalidad se tuvo que usar una herramienta que nos permita virtualizar los

servidores con sus respectivos sistemas operativos (CentOS), la herramienta utilizada fue VMWare Workstation 10.0 para realizar dicha funcionalidad, en otras investigaciones como la de Raynaud, 2010 en la que también fue necesario utilizar la virtualización se utilizó VirtualBox como principal medio de virtualización de sistemas operativos.

VI. CONCLUSIONES

1. Se logró de una manera fácil la instalación y configuración del sistema operativo CentOS 6.3 en todos los servidores involucrados en la replicación, se establecieron los lineamientos necesarios para que el sistema operativo funcione adecuadamente según los requerimientos necesarios en la implementación del comparador y sincronizador de bases de datos relacionales.
2. Al analizar la red estructurada de red de la universidad particular de Chiclayo se logró identificar que el principal motivo de la saturación de red es causada por el poco ancho de banda contratado y el mal estado en que se encuentra el cableado estructurado de red, la saturación mencionada es provocada en las principales horas punta de la jornada laboral administrativa.
3. Se logró implementar la instalación y configuración exitosa tanto del servidor web apache como de las bases de datos sobre el gestor de bases de datos relacionales PostgreSQL 8.0, así como también se detalló todo el procedimiento de instalación y configuración en un manual para que en futuros casos la dificultad de este tipo de implementación sea mínima, por ende quedó todo listo para poder realizar la sincronización con la herramienta de replicación Bucardo.
4. Se logró poner a prueba el Servidor Comparador y Sincronizador de bases de datos utilizando software libre como son el CentOS, PostgreSQL, Bucardo, Apache los cuales en conjunto lograron una adecuada sincronización entre bases de datos de los servidores locales de la Universidad Particular de Chiclayo

VII. RECOMENDACIONES

- Se recomienda establecer políticas de seguridad para que el sistema operativo no este expuesto a vulnerabilidades tales como GHOST que permiten tomar el control absoluto del sistema, es por ello que se debe contratar a personas en cargadas de mantener la seguridad tanto del hardware como del software y así evitar futuras perdidas.
- Se recomienda hacer una reinstalación de todos los equipos y cableado de red que se encuentren en mal estado con la finalidad de que haya una óptima comunicación de red entre los servidores y clientes que interactúen con las bases de datos.
- Es necesario mantener un personal capacitado con respecto a las herramientas utilizadas en este proyecto tales como apache, postgresql, sistemas operativos Linux, bases de datos, etc. para que pueda brindar una solución ante cualquier inconveniente a lo largo del funcionamiento de los servidores de sincronización.
- El proyecto fue diseñado para ser implementado en cualquier ambiente Linux y en los casos que sea necesario mantener una alta disponibilidad de la información en esta investigación se tomó como marco de referencia el problema con el que se lidia actualmente en la Universidad particular de Chiclayo. Es por ello que es recomendable que la persona administradora de los servidores encargados de la sincronización y replicación sean personas conocedoras del tema de Bases de Datos Relacionales con el gestor de bases de datos PostgreSQL y del sistema operativo Linux para que pueda dar el respectivo soporte y soluciones a los diferentes inconvenientes que puedan presentarse y que impidan el normal funcionamiento.

VIII. PROPUESTA TECNOLÓGICA

VIII.1. Generalidades

Para comenzar a generalizar este proyecto de investigación definamos primero el término de replicación de base de datos, es una técnica mediante la cual copiamos de forma exacta en otra ubicación una instancia de la base de datos. En la mayoría de casos se utiliza en entornos distribuidos de Sistemas de Gestión de Bases de Datos donde una sola base de datos tiene que ser utilizada y actualizada en varios lugares de forma simultánea.

En la actualidad existen en la red multitud de aplicaciones y de sistemas que tienen por debajo una base de datos que sigue el modelo cliente-servidor. Muchas veces esos sistemas deben de tener garantizada la accesibilidad por lo que para evitar problemas es necesario utilizar este tipo de técnicas de replicación de base de datos de forma que un fallo en uno de los servidores de base de datos no impida a los usuarios seguir utilizando la aplicación.

Mediante la replicación de base de datos, usuarios de todo el mundo pueden estar accediendo a lo que para ellos son los mismos datos, aunque en realidad, físicamente esos datos pueden estar de forma transparente para el usuario, en diferentes nodos o localidades.

En el siguiente proyecto de investigación que se está presentando constará de un análisis, diseño e implementación de un comparador y sincronizador de bases de datos relacionales en el cual se tomará como caso de estudio y aplicación un problema que actualmente está afectando a la disponibilidad eficaz de la información almacenada en su base de datos central de la Universidad Particular de Chiclayo.

El objetivo de este proyecto en sí es realizar una replicación de Bases de Datos PostgreSQL, que optimice en una mayor escala los recursos de la empresa o institución en este caso la Universidad Particular de Chiclayo mediante la replicación de data de dos o más bases para hacer una sincronización de la información que tiene una base datos con la información de otra base de datos con la que se replicara y así poder solucionar los diversos problemas que suceden y causan malestar en los clientes.

La implementación de esta función se ha realizado utilizando una herramienta de replicación de bases de datos PostgreSQL llamada bucardo, el cual permite realizar replications asíncronas maestro – maestro entre dos o más bases de datos relacionales que utilicen el manejador de base de datos PostgreSQL.

Definamos primero que es PostgreSQL, es un potente sistema de base de datos objeto-relacional de código abierto. Cuenta con más de 15 años de desarrollo activo y una arquitectura probada que se ha ganado una sólida reputación de fiabilidad e integridad de datos. Se ejecuta en los principales sistemas operativos que existen en la actualidad como: Linux, UNIX (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64), Windows.

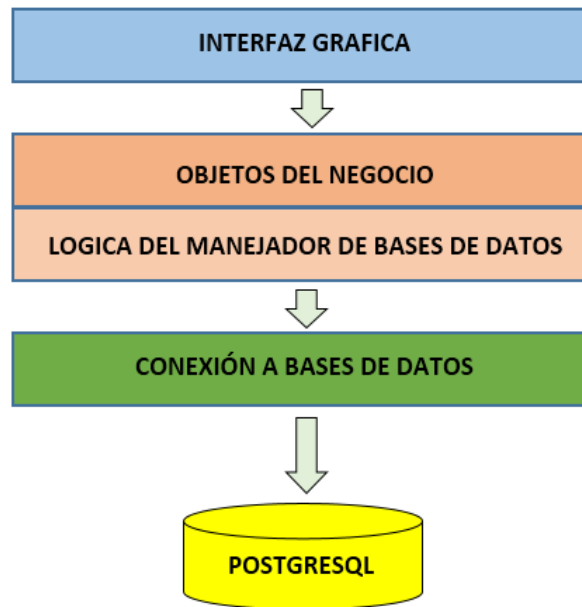


Imagen 5 Modelo Conexión desde una aplicación a Postgresql

Es totalmente compatible con ACID, tiene soporte completo para claves foráneas, uniones, vistas, disparadores y procedimientos almacenados (en varios lenguajes). Incluye la mayoría de los tipos de datos del SQL 2008, incluyendo INTEGER, numérico, BOOLEAN, CHAR, VARCHAR, DATE, INTERVAL, y TIMESTAMP. También soporta almacenamiento de objetos binarios grandes, como imágenes, sonidos o vídeo. Cuenta con interfaces nativas de programación para C / C + +, Java,. Net, Perl, Python, Ruby, Tcl, ODBC, entre otros, y la documentación que actualmente existe es realmente excepcional.

Una base de datos de clase empresarial, PostgreSQL cuenta con características avanzadas tales como Multi-Version Control de concurrencia (MVCC), puntos en tiempo de recuperación, tablespaces, replicación asincrónica, transacciones anidadas (savepoints), respaldos online/hot, un sofisticado query planner/optimizer. Soporta el conjunto de caracteres internacional, codificaciones de caracteres multibyte, Unicode, mayúsculas y minúsculas.

Es altamente escalable, tanto en la enorme cantidad de datos que puede manejar y en el número de usuarios concurrentes que puede administrar. Hay sistemas activos en PostgreSQL en entornos de producción que manejan más de 4 terabytes de datos.

Debido a que PostgreSQL no dispone de un sistema nativo para replicar, por lo que, para llevar a cabo esta gestión, podemos hacer uso de la más común de las herramientas destinadas a ello: Bucardo.

Ahora definamos que es Bucardo, es una herramienta que nos permite replicar dos o más bases de datos idénticas, tanto en modo master/Slave como master/master.

Los pasos a llevar a cabo son relativamente sencillos, aunque eso siempre depende del entorno en el que nos encontremos. Para simplificar la instalación del Bucardo, recomendamos hacer uso de un servidor basado en Debian/Ubuntu, porque podemos hacerlo desde los mismos repositorios con un simple “apt-get install bucardo “, lo cual instalará todas las dependencias necesarias para su uso. Una vez instalado, todas las gestiones se realizan con el comando “bucardo_ctl“. Es importante remarcar que el servicio de bucardo no necesariamente debe estar donde reside el motor de base de datos a replicar. Dicho de otro modo, podemos tener un servidor independiente que se dedique exclusivamente a realizar la replicación de dos bases de datos remotas.

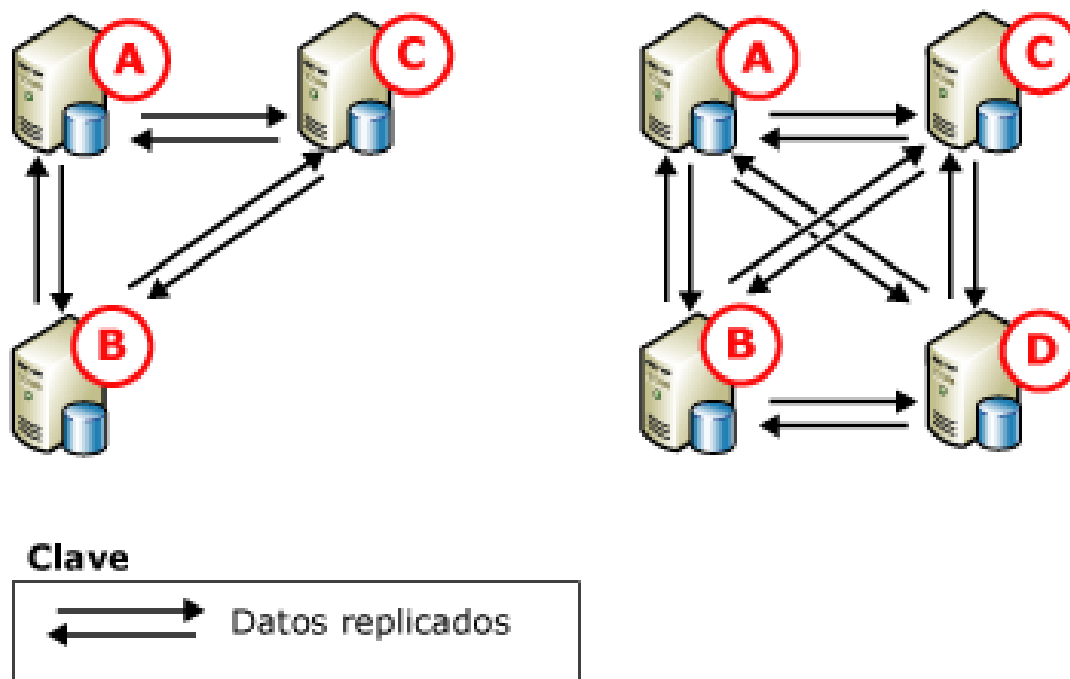


Imagen 6 Diseño gráfico del modelo de replicación entre varios servidores

En el esquema actual que se implementara será el de maestro – maestro para que ambas bases trabajen de igual forma y se pueda aprovechar todos los recursos de las mismas tales como la posibilidad de insertar registros, actualizar registro, o eliminarlos. Uno de los objetivos al realizar esta implementación será el de mantener la información segura y un ágil acceso a ella.

VIII.2. Análisis

Una vez obtenida toda la información requerida en la parte anterior, se realizó una evaluación de la situación actual a nivel de equipos, configuración y distribución. Esto permitió la primera etapa de la investigación conocer la situación actual.

Una vez echo la primera etapa de nuestra investigación que consiste en conocer la situación actual en que se encuentra la Universidad Particular de Chiclayo, para los fines de crear una propuesta base y alternativas que fueran factibles para la institución, se realizó una encuesta a los sectores que inciden de manera directa en el propósito de esta investigación llegando a identificar los siguientes puntos:

- Se tomó una muestra durante el transcurso de todo el día durante horas determinadas con la finalidad de identificar la fluctuación del ancho de banda para conocer las ventajas y desventajas que este tiene y su funcionalidad e importancia. La recolección de esta información será importante para la toma de decisiones.
- Para conocer también cómo funciona la conectividad de red en las sedes en la Universidad Particular de Chiclayo se llegó a la conclusión que personal administrativo y población estudiantil hacen un consumo alto del ancho de banda sobrepasando en algunos casos el límite de este.
- Mediante el uso de la técnica de observación llegamos a ver el estado en que se encuentra el cableado estructurado en las sedes de la Universidad Particular de Chiclayo observando lo siguiente: mala estructura del cableado de red en las diversas oficinas, aulas y laboratorios.
- Se observó que la gran mayoría de equipos que controlan el tráfico de red se encuentran en mal estado producto de los años y se encuentran mal ubicados en condiciones que podrían dañar el funcionamiento de estos.

Partiendo de los resultados de las mediciones del ancho de banda, se realizó un análisis donde, según la experiencia de los proponentes e investigaciones al respecto, se identificó el origen de los principales problemas y las necesidades reales en todas las sedes de la Universidad Particular de Chiclayo. Este análisis conocido como Análisis de las necesidades nos brinda una visión general de los problemas y sirve para establecer prioridades. El mismo, en resumen, implica:

- Evaluar la situación del ancho de banda en la Universidad Particular de Chiclayo.
- Analizar las horas en que el ancho de banda es mayor.
- Detectar las horas en que el ancho de banda es menor.

Una vez obtenido los resultados de las encuestas, continuamos con el proceso de identificación, se realizó un análisis completo y se logró obtener la opinión de los que fueron encuestados, llegando a las siguientes conclusiones:

- Los encuestados califican de una manera baja la rapidez en la que son atendidos.
- Las respuestas de los encuestados nos dieron a entender la necesidad de mejorar esta situación y tomar medidas urgentes para mejorar la disconformidad de estos y así poder brindar un servicio de mejor calidad.

1era Pregunta

a)	13	ptos
b)	7	ptos
c)	0	ptos

6ta Pregunta

a)	17	ptos
b)	3	ptos

2da Pregunta

a)	3	ptos
b)	2	ptos
c)	15	ptos

3ra Pregunta

a)	18	ptos
b)	2	ptos
c)	0	ptos
d)	0	ptos

4ta Pregunta

a)	16	ptos
b)	3	ptos
c)	1	ptos
d)	0	ptos

5ta Pregunta

a)	5	ptos
b)	8	ptos
c)	3	ptos
d)	2	ptos
e)	2	ptos

Imagen 7 Comparativa resultados de encuestas hechas

Hora	Mbps de descarga	Mbps de carga
08:30 a.m.	3.75	1.8
09:30 a.m.	1.75	1
10:30 a.m.	0.95	0.51
11:30 a.m.	0.86	0.46
12:30 p.m.	0.95	0.51
01:30 p.m.	3	1.8
02:30 p.m.	3.74	1.9
03:30 p.m.	0.92	0.55
04:30 p.m.	0.83	0.49
05:30 p.m.	0.99	0.55
06:30 p.m.	3.2	1.93
07:30 p.m.	3.9	2
08:30 p.m.	3.6	1.94
09:30 p.m.	4	1.92
10:30 p.m.	3.6	2
11:30 p.m.	5	1.95

Ancho de banda contratado	6 Mbps descarga	2 Mbps subida
------------------------------	--------------------	------------------

Tabla 12 Comparativa Fluctuaciones de ancho de banda en universidad particular de Chiclayo

VIII.3. Diseño

El diseño que se propone a continuación, se basa en estudios realizados partiendo de la observación, también se aplicó una serie de encuestas para poder llegar a la causa del problema, asimismo una vez que se llegó a concluir el problema se propuso una solución y realizar un correcto diseño del comparador y sincronizador para poder solucionar los diversos problemas que se suscitan en cada una de las sedes de la Universidad Particular de Chiclayo y a sí acabar con estos.

En este apartado se describe el diseño de la solución propuesta. La primera parte comprende el diseño en alto nivel de la arquitectura de red justificando la elección de un patrón arquitectónico. Respecto proceso de replicación, se mencionan los patrones y estándares adoptados para llevar a cabo la sincronización entre las bases de datos.

VIII.3.1. Arquitectura de la red

En esta sección se explica el diseño a alto nivel y los paradigmas arquitectónicos de cómo está conformado y configurado actualmente el servidor con los respectivos nodos (sucursales) que realizan solicitudes al servidor de base de datos central.

Se puede apreciar de que todos se están conectando al mismo servidor, sin embargo, algunos nodos se encuentran a una distancia considerablemente apartada del servidor de base de datos central dando como resultado una respuesta tardía por parte del servidor al nodo alejado que hizo la solicitud. Y sí a eso se le agrega los problemas encontrados y mencionados anteriormente nos dará como resultado una congestión de la información solicitada.

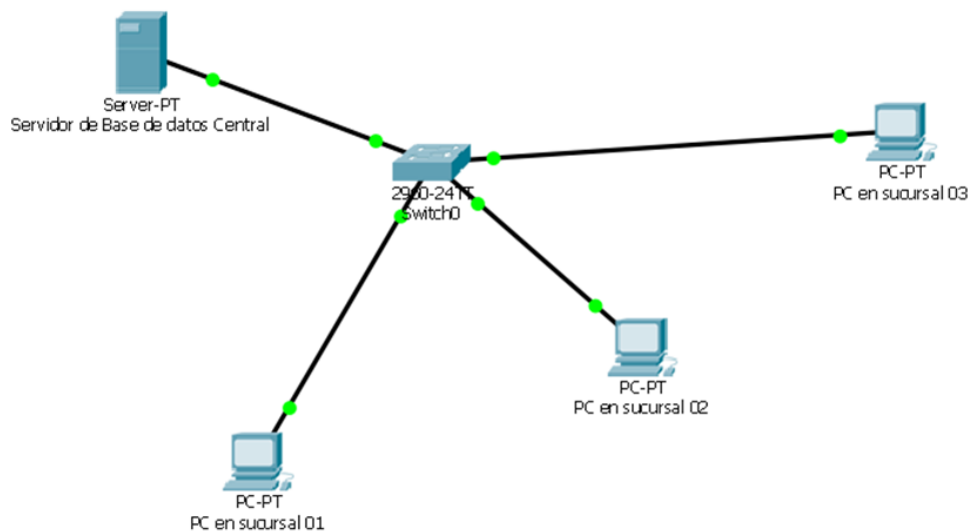


Imagen 8 Estructura del diseño actual de la red y su servidor de base de datos

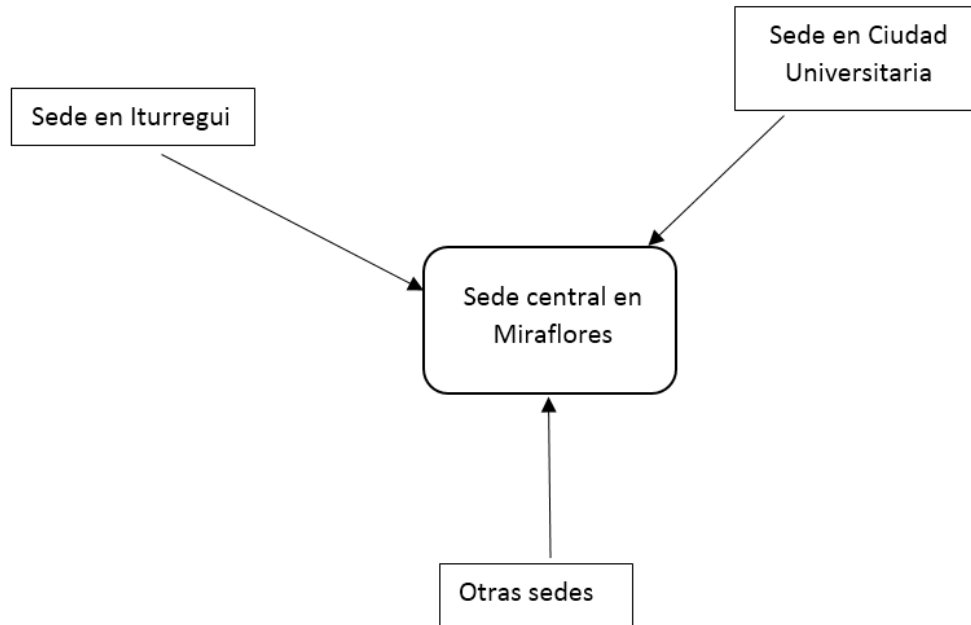
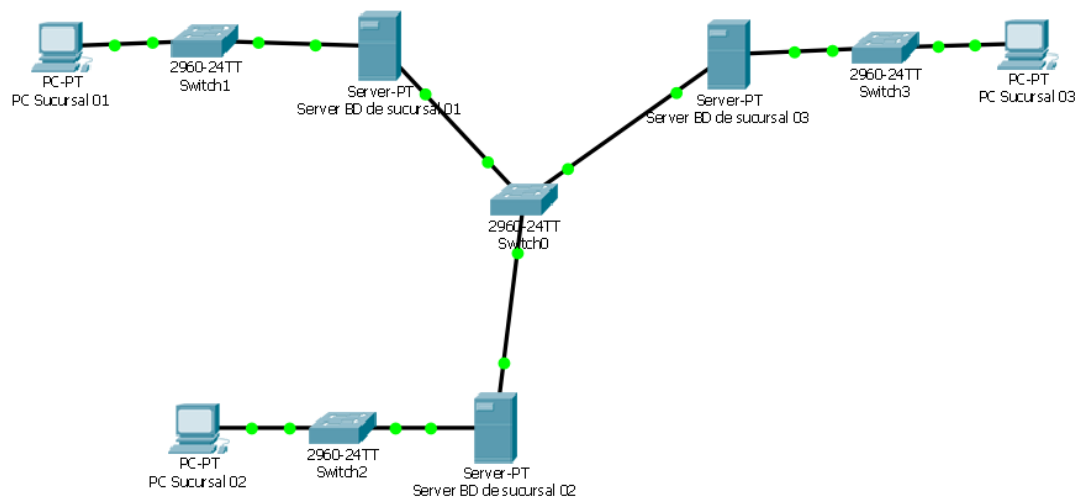


Imagen 9 Estructura del diseño actual de la red y su servidor de base de datos

A continuación, se describirá el diagrama de la arquitectura de red que se está proponiendo, el cual presenta mejoras para solucionar los problemas que actualmente se presentan.

La siguiente arquitectura estará conformada por un servidor de base de datos en cada sucursal quienes se estarán sincronizando a una hora determinada del día con la finalidad de que todos cuenten con la misma información.



En
el

Imagen 10 Estructura de distribución de servidores de bases de datos propuesta en el presente proyecto
 diagrama planteado cada sucursal de la Universidad Particular de Chiclayo cuenta con un servidor de base de datos al que solo los nodos que se encuentren en su misma sucursal van a ser la solicitud de información.

VIII.3.2. Diseño de la tabla a replicar

La siguiente tabla ALUMNO se utilizará para almacenar los datos que seguidamente

serán replicados hacia otras bases de datos con la misma estructura de tabla.

ALUMNO	
idalumno	int
nombre	varchar(40)
apellidos	varchar(100)

Imagen 11 Estructura de la tabla Alumno que se utilizara para hacer las pruebas de replicación

VIII.3.3. Diseño del formulario para las pruebas

A continuación, se mostrarán los diseños de formularios que se utilizarán para realizar las pruebas, servirán para poder insertar, actualizar y visualizar registros a la tabla alumno de la base de datos hecha en el manejador de bases de datos PostgreSQL.

Los siguientes formularios están moqueteados a base de HTML y diseñados con css3, la información ingresada en los formularios será procesada por el lenguaje de programación PHP.

Se trató de hacer los formularios de una manera sencilla y básica para centrarnos solo en el proceso de replicación,

Administración de alumnos

Insertar

Código	Nombre	Apellido			
6	Cristian Arturo	Cruz Benel	Ver	Editar	Eliminar
1	Andres	Paredes	Ver	Editar	Eliminar

Imagen 12 Vista principal del formulario que interactuara con las bases de datos

Insertar alumno

Nombre del alumno

Apellido del alumno

Guardar

Imagen 13 Formulario para Insertar un nuevo registro en la base de datos PostgreSQL

Detalles del alumno

Código: 6
Nombre: Cristian Arturo
Apellido: Cruz Benel

Imagen 14 Vista para visualizar los detalles de un registro guardado en la base de datos PostgreSQL

Editar alumno

Nombre del alumno

Cristian Arturo

Apellido del alumno

Cruz Benel

Guardar

Imagen 15 Formulario para editar un registro guardado en la base de datos PostgreSQL

VIII.4. Implementación

En este capítulo se explicarán los detalles relacionados a la fase de implementación de la solución tales como tecnologías a utilizar y los tipos de pruebas a utilizar para verificar las funcionalidades del aplicativo implementado en este proyecto de investigación.

Empezaremos por discutir la elección de las tecnologías a utilizar para el desarrollo e implementación de la solución propuesta en esta investigación. Esto incluye que plataformas se utilizaran para ejecutar las aplicaciones, tales como máquinas virtuales, lenguajes de programación en el backend, frontend y de maquetado, tipo de replicación que se utilizara, configuración del sistema operativo sobre el que se llevara a cabo el replicador, configuración creación de la base de datos y formularios de pruebas.

VIII.4.1. Virtualización con VMware Workstation

En lo que compete a la plataforma sobre la que se llevará a cabo todas las configuraciones será en un ambiente virtualizado por lo que el programa para dicha tarea será VMware Workstation 10.0

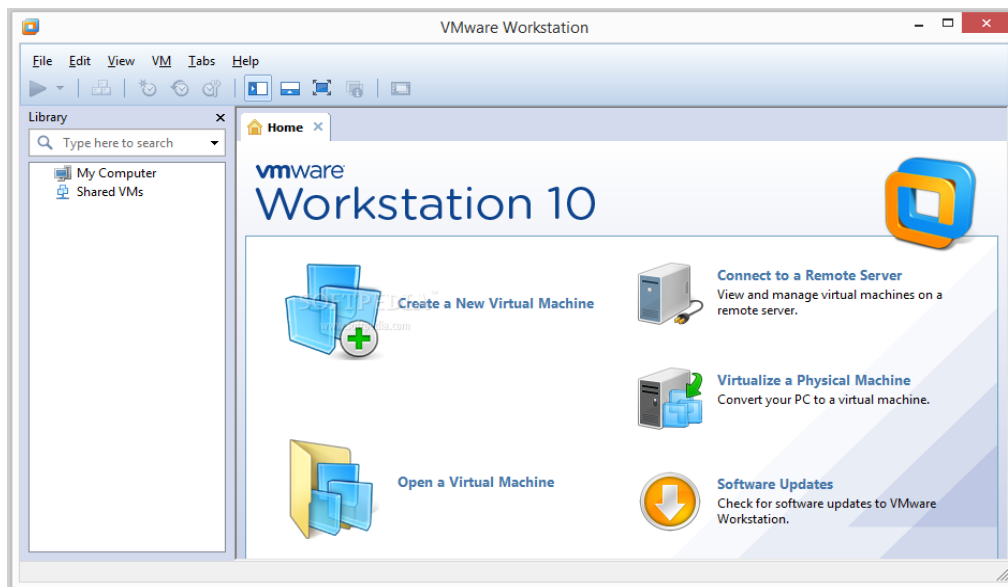


Imagen 16 Programa VMWare en el que se virtualizaran los sistemas operativos CentOS 6.3

Características de WMware 10.0
Ejecución de máquinas virtuales con hasta 16 CPUs virtuales.
Soporte discos duros virtuales de hasta 2TB
Configuración de discos duros como SATA, IDE y SCSI
Detección de Discos SSD y optimización de los procesos
Navegación con Múltiple-Monitor

Tabla 13 Características principales de VMWare Workstation

VIII.4.2. Sistema Operativo CentOS

El sistema operativo en mención será instalado en dos máquinas virtuales para que la replicación sea hecha entre ambas máquinas, sin embargo, es importante mencionar que la replicación funcionaria efectivamente con más de dos.

En lo que respecta al sistema operativo es necesario configurar unos parámetros para el óptimo funcionamiento y poder continuar con la instalación del comparador y sincronizador.

Las dos máquinas involucradas en la replicación a partir de ahora serán referidas como “servidor01” quien hará referencia a una sucursal de la universidad particular de Chiclayo y “servidor02” que será el que hará referencia a otra sucursal de la misma, como primer paso en la configuración será asignarles una IP estática a cada máquina. Para el servidor01 se le asignara la IP 192.168.1.50 y para el servidor02 la IP 192.168.1.60

```

GNU nano 2.0.9
127.0.0.1                localhost.localdomain localhost
::1                    localhost6.localdomain6 localhost6
192.168.1.50            sucursal1.tesis.com      sucursal1
192.168.1.60            sucursal2.tesis.com      sucursal2

```

Imagen 17 Contenido del archivo hosts del servidor 01

Archivo hosts del servidor01

```

GNU nano 2.0.9                                Fichero: /etc/hosts
127.0.0.1                localhost.localdomain localhost
::1                    localhost6.localdomain6 localhost6
192.168.1.60            sucursal2.tesis.com      sucursal2
192.168.1.50            sucursal1.tesis.com      sucursal1

```

Imagen 18 Contenido del archivo "hosts" del servidor 02

Archivo hosts del servidor02

```

GNU nano 2.0.9                                Fichero: /etc/sysconfig/network
NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=sucursal1.tesis.com

```

Imagen 19 Archivo "network" del servidor01

Archivo network del servidor01

```
GNU nano 2.0.9 Fichero: /etc/sysconfig/network

NETWORKING=yes
NETWORKING_IPV6=no
HOSTNAME=sucursal2.tesis.com
```

Imagen 20 Archivo "network" del servidor02

Archivo network del servidor02

Asignación de IP Estática

```
GNU nano 2.0.9 Fichero: ifcfg-eth0

DEVICE=eth0
IPADDR=192.168.1.50
NETMASK=255.255.255.0
ONBOOT=yes
NAME=eth0
TYPE=Ethernet
HWADDR=00:0c:29:e9:ce:5e
BOOTPROTO=none
GATEWAY=192.168.1.1
IPV6INIT=no
USERCTL=no
```

Imagen 21 Archivo "ifcfg-eth0" del servidor01

En este archivo de establece la dirección estática que tendrá el servidor

```
GNU nano 2.0.9 Fichero: ifcfg-eth0

DEVICE=eth0
IPADDR=192.168.1.60
NETMASK=255.255.255.0
ONBOOT=yes
NAME=eth0
TYPE=Ethernet
HWADDR=00:0c:29:b3:97:5d
BOOTPROTO=none
GATEWAY=192.168.1.1
IPV6INIT=no
USERCTL=no
```

Imagen 22 Archivo ifcfg-eth0 del servidor02

En este archivo de establece la dirección estática que tendrá el servidor

Para que los cambios hechos en los archivos de configuración sean reconocidos por el Sistema Operativo es necesario ejecutar una serie de comandos que serán descritos a continuación

```
service NetworkManager stop  
  
chkconfig NetworkManager off  
  
service network restart
```

Imagen 23 Comandos para configurar IP estática

Se deben ejecutar estos comandos para que los cambios hechos en los archivos anteriores sean reconocidos por el servidor

VIII.4.3. Servidor web Apache

Apache es uno de los programas open-source más utilizados para poder montar un servidor web en una computadora, es multiplataforma por lo que funciona en cualquier sistema operativo ya sea Windows, IOS, o en este caso en la distribución CentOS de Linux.

Para tener nuestro servidor web apache, primero es necesario hacer su respectiva instalación haciendo uso de una serie de comandos, el resultado final de la ejecución de dichos comandos sería el mostrado en la imagen inferior. El proceso realizado debe hacerse por igual en todos los servidores involucrados en la sincronización en este caso será en el servidor01 y servidor02.

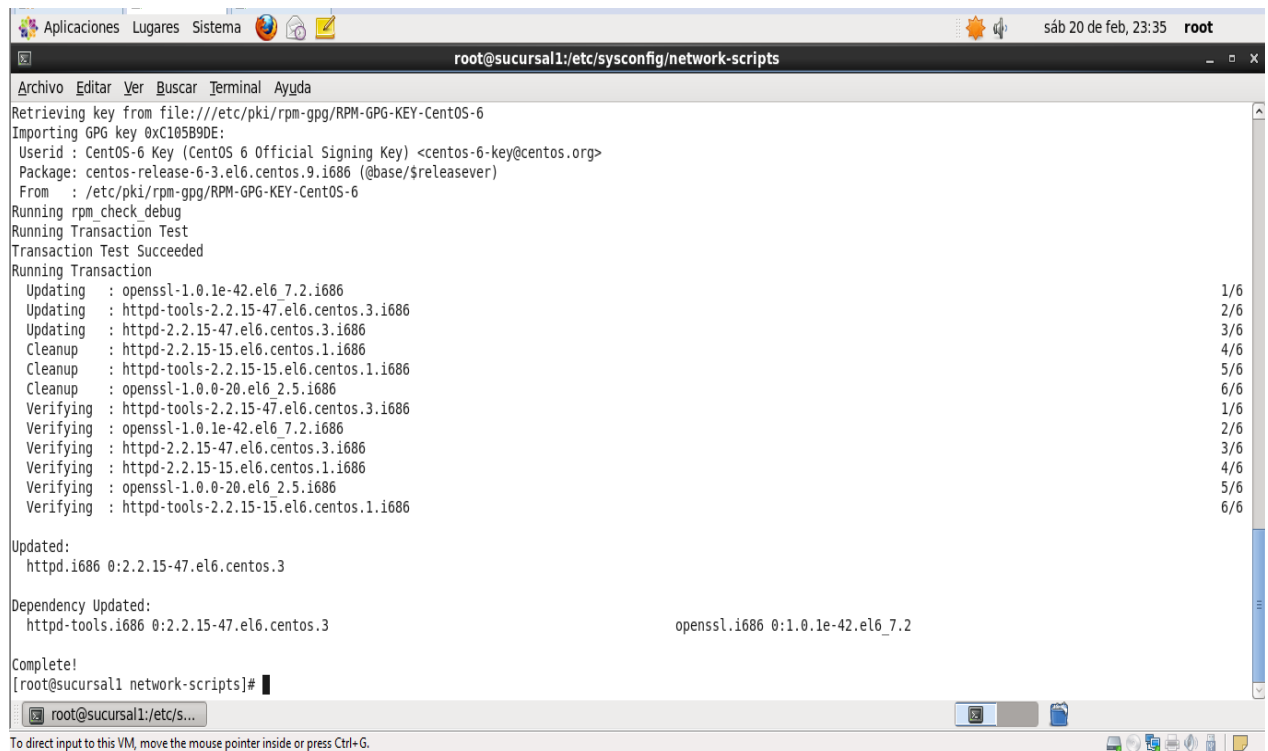


Imagen 24 Instalación del servidor web Apache

Los comandos utilizados son mencionados a continuación:

```
yum -y install httpd
service httpd start
chkconfig httpd on
```

Imagen 25 comandos para instalar y ejecutar el servidor web

VIII.4.4. Lenguaje de programación PHP 5.6.

Entre los candidatos de tecnologías para el procesamiento de datos que se llenaran por el formulario, se ha escogido PHP 5.6 como lenguaje de programación ya que el proceso de llenado de datos va enfocado a un entorno web.

Con respecto al lenguaje de programación PHP tiene soporte para conectarse con bases de datos hechas en PostgreSQL y hace la gestión de datos de una forma fácil y eficientemente.

Se utilizara el framework de PHP “Laravel 5.4” para agilizar la implementación de los formularios que nos permitirán realizar la creación, lectura, actualización y eliminación de los registros en cada una de las base de datos.

Para instalar la versión de php 5.6 es necesario realizar el siguiente procedimiento:

Registrar los repositorios donde se encuentra disponible la versión 5.6 de php

```
# wget http://dl.fedoraproject.org/pub/epel/6/i386/epel-release-6-8.noarch.rpm
# wget http://rpms.famillecollet.com/enterprise/remi-release-6.rpm
# rpm -Uvh remi-release-6*.rpm epel-release-6*.rpm
```

Imagen 26 Comandos para agregar los repositorios de donde se descargara PHP 5.6

Modificar el archivo **/etc/yum.repos.d/remi.repo** y habilitar el repositorio remi

```
[remi]
name=Remi's RPM repository for Enterprise Linux 6 - $basearch
baseurl=http://rpms.remirepo.net/enterprise/6/remi/$basearch/
#mirrorlist=https://rpms.remirepo.net/enterprise/6/remi/httpsmirror
#mirrorlist=http://cdn.remirepo.net/enterprise/6/remi/mirror
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-remi
```

Imagen 27 Configuración para descargar desde el repositorio agregado anteriormente

Habilitar la versión de php que se desea descargar por defecto.

```
[remi-php55]
name=Remi's PHP 5.5 RPM repository for Enterprise Linux 6 - $basearch
baseurl=http://rpms.remirepo.net/enterprise/6/php55/$basearch/
#mirrorlist=https://rpms.remirepo.net/enterprise/6/php55/httpsmirror
#mirrorlist=http://cdn.remirepo.net/enterprise/6/php55/mirror
# NOTICE: common dependencies are in "remi-safe"
enabled=0
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-remi

[remi-php56]
name=Remi's PHP 5.6 RPM repository for Enterprise Linux 6 - $basearch
baseurl=http://rpms.remirepo.net/enterprise/6/php56/$basearch/
#mirrorlist=https://rpms.remirepo.net/enterprise/6/php56/httpsmirror
#mirrorlist=http://cdn.remirepo.net/enterprise/6/php56/mirror
# NOTICE: common dependencies are in "remi-safe"
enabled=1
gpgcheck=1
gpgkey=file:///etc/pki/rpm-gpg/RPM-GPG-KEY-remi
```

Imagen 28 Configuración 2 para descargar desde el repositorio agregado anteriormente

Registrar los certificados del repositorio remi en centos.

```
# yum --disablerepo=epel -y update ca-certificates
```

Imagen 29 Comando para registrar los repositorios REMI

Proceder con la instalación de php 5.6.

```
# yum install php php-mcrypt php-cli php-gd php-curl php-mysql php-ldap php-zip php-fileinfo
```

Imagen 30 Comando para iniciar la descarga de PHP 5.6

VIII.4.5. PostgreSQL 8.4

Una de los principales objetivos de este proyecto es hacer uso de herramientas poderosas y que al mismo tiempo sean de código libre para que la implementación

sea considerablemente económica ya que al hacer uso de este tipo de aplicaciones nos olvidamos del pago de licencias por su uso, como lo es en el caso de PostgreSQL, el que es un potente gestor de bases de datos y con su código fuente libre.

Primero procederemos con su instalación procedida por su configuración para que empiece su ejecución. Al igual que con el servidor web estos procedimientos se tienen que hacer en todos los servidores involucrados.

```
yum -y install postgresql-server  
service postgresql initdb  
service postgresql start  
chkconfig postgresql on
```

Imagen 31 Comandos para descargar e inicializar PostgreSQL

Una vez inicializado el proceso de postgresql ya se puede acceder a su administrador de bases de datos, para ello es necesario conocer el usuario y contraseña por defecto con la cual se configura cada vez que se instala.

```
su - postgres
```

Imagen 32 Comando para entrar al gestor de bases de datos postgresql

El usuario por defecto es “postgres” y la contraseña en blanco, una vez dentro del gestor de base de datos podremos proceder con la creación de la base de datos que se utilizara para hacer las pruebas de replicación.

Ejecutamos el comando “psql” para ejecutar comandos en el gestor de base de datos postgresql.

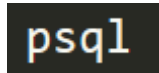
A terminal window with a dark background. The text 'psql' is written in a light blue, monospaced font.

Imagen 33 Comando para abrir una consola en postgresql

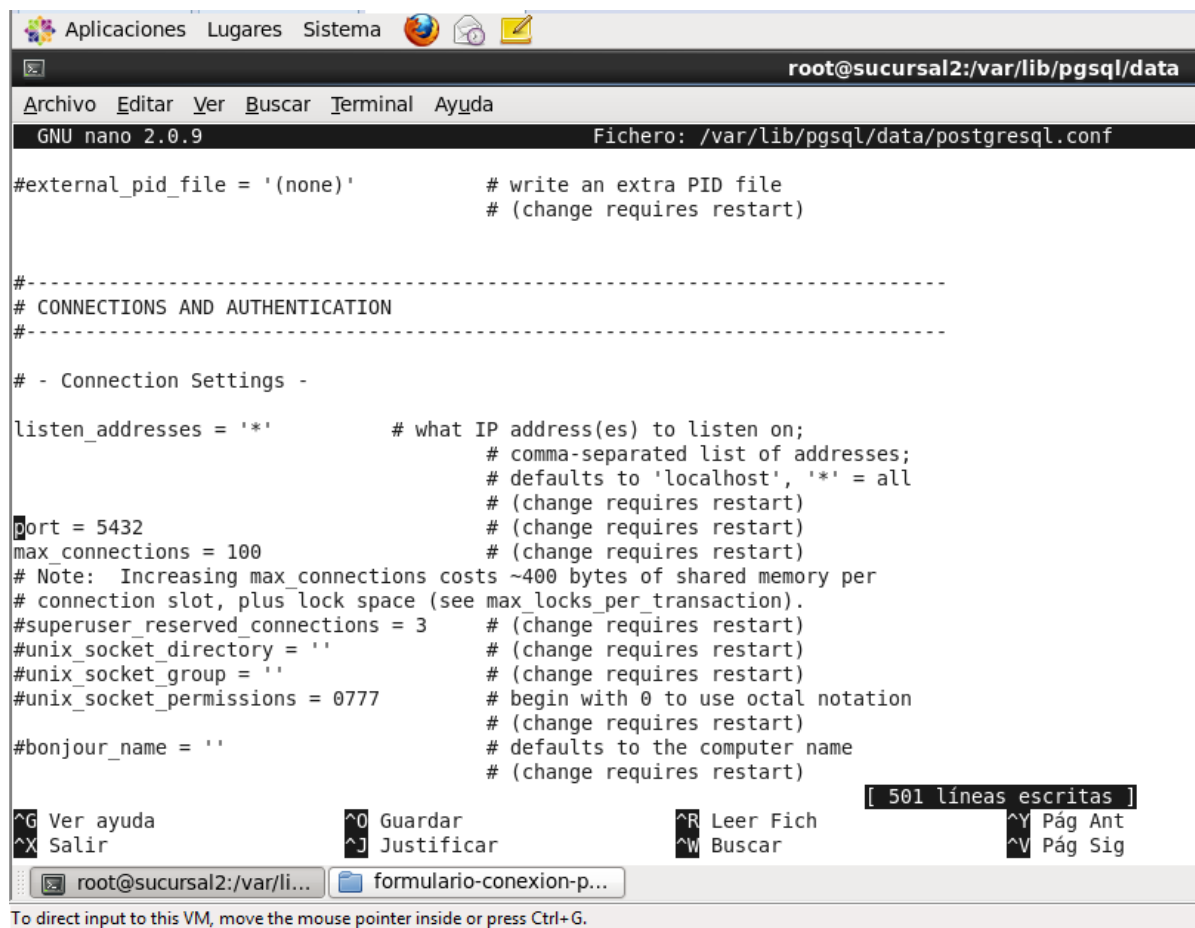
Al usuario postgres le asignaremos una contraseña para aumentar la seguridad y conectarnos desde el backend ingresando su respectivo usuario y contraseña.

A terminal window with a dark background. The text '\password postgres' is written in a light blue, monospaced font.

Imagen 34 Comando para establecer una contraseña al usuario "postgres "

Una vez instalado el postgresql se le tiene que configurar para habilitar las conexiones TCP/IP y que escuche por el puerto 5432, los cambios para dicha configuración deben ser hechos en el archivo postgresql.conf.

El primer parámetro a modificar es “listen_addresses” el cual estará comentado y con un valor de ‘localhost’ el cambio a hacer es el de descomentarlo y poner un ‘*’ en lugar de ‘localhost’ y por último descomentar el parámetro “port” y asignarle como valor “5432”:



```
Aplicaciones Lugares Sistema
root@sucursal2:/var/lib/pgsql/data
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.0.9 Fichero: /var/lib/pgsql/data/postgresql.conf

#external_pid_file = '(none)'          # write an extra PID file
                                       # (change requires restart)

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

listen_addresses = '*'                # what IP address(es) to listen on;
                                       # comma-separated list of addresses;
                                       # defaults to 'localhost', '*' = all
                                       # (change requires restart)
port = 5432                           # (change requires restart)
max_connections = 100                 # (change requires restart)
# Note: Increasing max_connections costs ~400 bytes of shared memory per
# connection slot, plus lock space (see max_locks_per_transaction).
#superuser_reserved_connections = 3   # (change requires restart)
#unix_socket_directory = ''           # (change requires restart)
#unix_socket_group = ''               # (change requires restart)
#unix_socket_permissions = 0777       # begin with 0 to use octal notation
                                       # (change requires restart)
#bonjour_name = ''                   # defaults to the computer name
                                       # (change requires restart)

[ 501 líneas escritas ]

^G Ver ayuda      ^O Guardar        ^R Leer Fich      ^Y Pág Ant
^X Salir          ^J Justificar     ^W Buscar         ^V Pág Sig

root@sucursal2:/var/li... formulario-conexion-p...
```

Imagen 35 Configuración del archivo "pg_hba.conf "

Sera necesario también hacer unas modificaciones en el archivo de configuración de postgresql llamado "pg_hba.conf" con la finalidad de configurar los permisos de postgresql para que cuando reciba peticiones mediante un rango de IP local no pida autenticación. Al igual que las configuraciones anteriores esta también tiene que ser aplicada en ambos servidores.

```
GNU nano 2.0.9                                Fichero: /var/lib/pgsql/data/pg_hba.conf
#
# If you want to allow non-local connections, you need to add more
# "host" records. In that case you will also need to make PostgreSQL listen
# on a non-local interface via the listen_addresses configuration parameter,
# or via the -i or -h command line switches.
#
#
# TYPE  DATABASE    USER        CIDR-ADDRESS          METHOD
# "local" is for Unix domain socket connections only
local   all             all                                trust
# IPv4 local connections:
host    all             all          127.0.0.1/32          trust
host    all             all          192.168.1.0/21        trust
# IPv6 local connections:
host    all             all          ::1/128               trust
```

Como último paso de configuración del postgresQL será el de reiniciarlo para que los cambios hechos sean reconocidos.

```
service postgresql restart
```

Imagen 36 Comando para reiniciar el servicio postgresql

VIII.4.6. Base de datos

La base de datos que se utilizara para realizar las pruebas de replicación entre las diferentes sucursales será hecha con una estructura básica con la finalidad de enfocarnos enteramente en la configuración del comparador y sincronizador de bases de datos, debido a que una vez teniendo el servidor replicador funcionando será posible adaptarlo a cualquier base de datos.

Cabe resaltar que para que la replicación se realice correctamente y evitar errores de escritura o lectura es importante que las bases de datos junto con sus respectivas tablas tengan la misma estructura, es decir, las tablas deben tener la misma cantidad de atributos, con los mismos tipos de datos y longitud.

A continuación se listan los comandos de postgresql que se utilizaran a lo largo de toda esta implementación.

1. Login usuario postgres:
2. `$ sudo su – postgres`
- 3.
4. Creación de un usuario:
5. **CREATE USER** nombre_usuario **WITH PASSWORD** '123456'
- 6.
7. Eliminar usuario:
8. **DROP USER** nombre_usuario
- 9.
10. Crear base de datos:
11. **CREATE DATABASE** nombre_db **WITH OWNER** nombre_usuario;
- 12.
13. Eliminar base de datos:
14. **DROP DATABASE** nombre_db
- 15.
16. Acceder a una base de datos con usuario x:
17. `psql -U nombre_usuario nombre_db`
- 18.
19. Obtener ayuda:
20. `\h`
21. Salir
22. `\q`
- 23.

24. Leer comandos desde un archivo:

25. \i script.sql

26.

27. Listar bases de datos:

28. \l

29.

30. Listar tablas de una base de datos:

31. \d

32.

33. Detallar una tabla:

34. \d table_name

35.

36. Detallar una tabla:

37. \d+ table_name

38.

39. Usar una base de datos:

40. \c nombre_db

41.

42. Mostrar usuarios:

43. **SELECT * FROM "pg_user";**

44. # o tambien

45. \du

Inciamos creando la base de datos para cada servidor, el nombre de la base de datos del servidor01 tendrá el nombre “bdservidor01” y la del servidor02

“bdservidor02”. Los comandos a utilizar para este procedimiento son los siguientes:

1. #EN SERVIDOR01
2. **CREATE DATABASE** bdservidor01;
3. \c bdservidor01
- 4.
- 5.
6. #EN SERVIDOR02
7. **CREATE DATABASE** bdservidor02;
8. \c bdservidor02

Con respecto a la creación de las tablas se hará uso de un script.sql para importar las sentencias SQL a la base de datos creada anteriormente en su respectivo servidor, para lograr dicho objetivo se hará uso del comando:

```
\i script.sql
```

```
\i '//home//script.sql';
```

Imagen 37 Comando para importar un script con código sql

El contenido del script.sql para la base de datos “bdservidor01” será descrito a continuación.

```
1 CREATE TABLE alumno
2 (
3
4     codigo SERIAL PRIMARY KEY,
5     nombre VARCHAR(200) NOT NULL,
6     apellido VARCHAR(200) NOT NULL
7
8 );
9
10
11 INSERT INTO alumno (nombre,apellido)
12 VALUES
13 ('cristian arturo', 'cruz benel'),
14 ('jose miguel','oblitas jimenez'),
15 ('carlos','Sanchez Paredes');
```

Imagen 38 Contenido del script que se ejecutara en la base de datos del servidor 01

Y el contenido del script.sql para la base de datos “bdservidor02” será el siguiente:

```
1 CREATE TABLE alumno(  
2  
3  codigo SERIAL PRIMARY KEY,  
4  nombre VARCHAR(200) NOT NULL,  
5  apellido VARCHAR(200) NOT NULL  
6  
7 );  
8  
9  
10 INSERT INTO alumno (nombre,apellido)  
11 VALUES  
12 ('Anderson', 'Del Valle Santa Cruz'),  
13 ('Peter','Ferrari'),  
14 ('Eduardo','Santana Lopez');  
15
```

Imagen 39 Contenido del script que se ejecutara en la base de datos del servidor 02

VIII.4.7. Formularios CRUD

Los formularios que interactuaran con las bases de datos serán desarrollados con HTML, JavaScript y PHP 5.6 utilizando el framework de PHP Laravel 5.4, es por ello que al utilizar un framework tendremos una estructura de archivos definida.

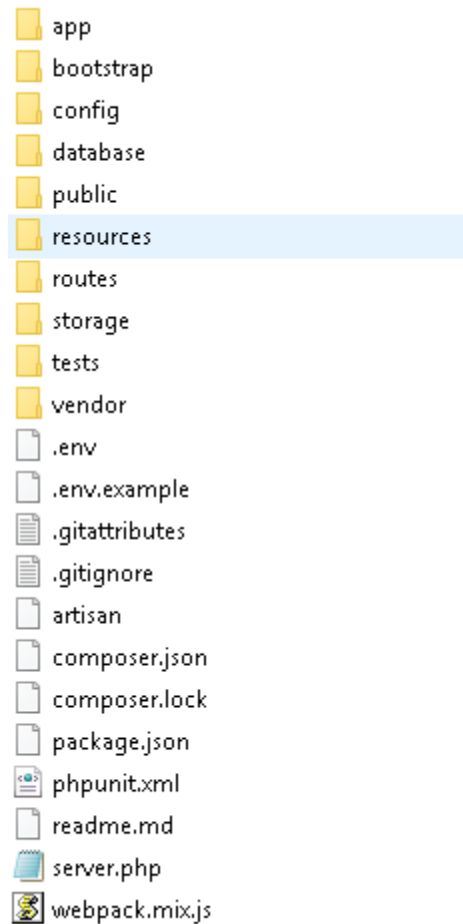


Imagen 40 Estructura de carpetas en un proyecto de Laravel 5.4

Las carpetas y archivos del framework con el que se trabajara en esta aplicación serán los siguientes:

Carpeta “app”: En ella se encuentran las clases que darán la funcionalidad a los formularios, tales como, los controladores, los modelos, y clases extras que podrían ser creadas para complementar la funcionalidad de la aplicación.

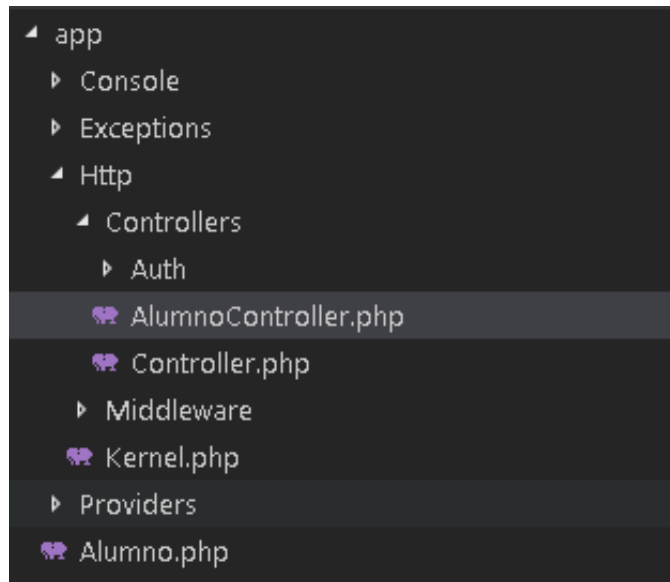


Imagen 41 Contenido de la carpeta "app" en un proyecto de laravel 5.4

Carpeta "public": En esta carpeta serán colocados todos los recursos estáticos de la aplicación es decir, archivos css, js, imágenes, fuentes, etc.

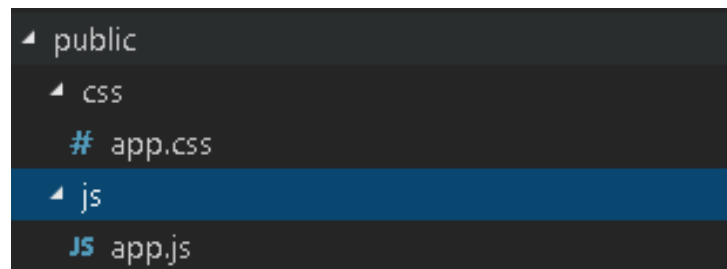


Imagen 42 Contenido de la carpeta "public" en un proyecto de laravel 5.4

Carpeta “resources”: Aquí se guardaran todas las vistas relacionadas a la interfaz de usuario y que serán mostradas en el navegador.

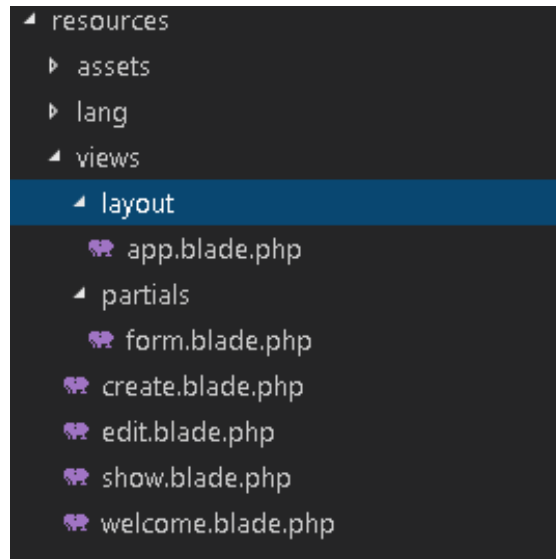


Imagen 43 Contenido de la carpeta “resources” en un proyecto de laravel 5.4

Carpeta “routes”: En esta carpeta estarán almacenados todas las URL’s enlazadas a su respectivo controlador el cual será ejecutado cuando se haga una petición a una URL.

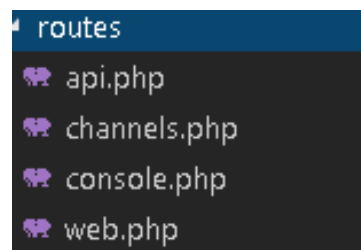


Imagen 44 Contenido de la carpeta “routes” en un proyecto de laravel 5.4

Archivo “.env”: En él se configurara la conexión a la base de datos y la conexión con el servidor de correo electrónico.

```
DB_CONNECTION=pgsql
DB_HOST=127.0.0.1
DB_PORT=5432
DB_DATABASE=bd_tesis
DB_USERNAME=postgres
DB_PASSWORD=postgres

BROADCAST_DRIVER=log
CACHE_DRIVER=file
SESSION_DRIVER=file
QUEUE_DRIVER=sync

REDIS_HOST=127.0.0.1
REDIS_PASSWORD=null
REDIS_PORT=6379

MAIL_DRIVER=smtp
MAIL_HOST=smtp.mailtrap.io
MAIL_PORT=2525
MAIL_USERNAME=null
MAIL_PASSWORD=null
MAIL_ENCRYPTION=null
```

Imagen 45 Contenido del archivo “.env” en un proyecto de laravel 5.4

La vista principal de la aplicación que interactuara con la base de datos se muestra a continuación, en ella se puede observar que en la parte superior se muestra el nombre del presente proyecto de investigación y en la parte inferior observamos un botón para insertar un nuevo alumno seguido por un listado de todos los alumnos almacenados en la base de datos PostgreSQL, cada uno con sus respectivas

opciones de ver, editar y eliminar

Comparador y Sincronizador de Bases de Datos Relacionales

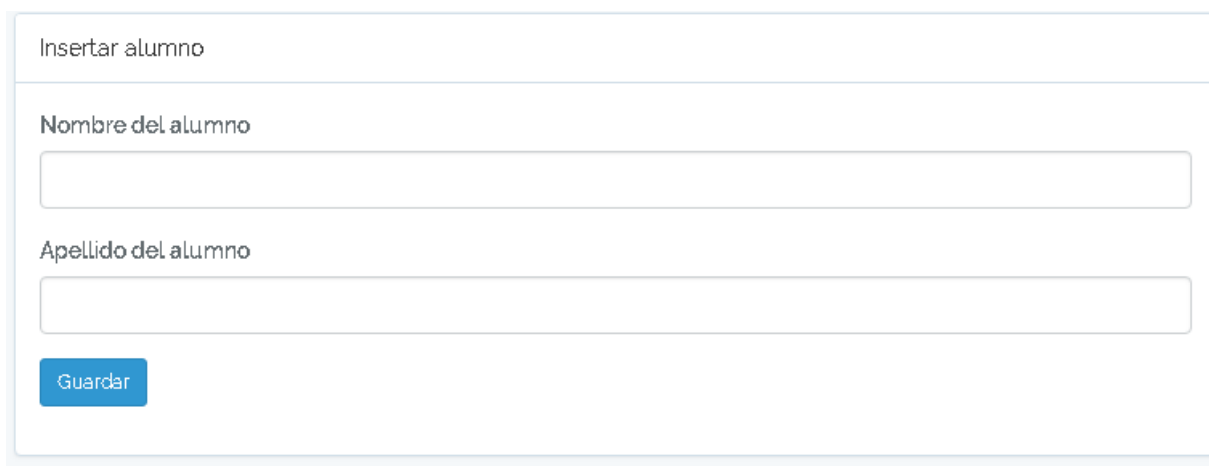


The screenshot shows a web interface titled "Administracion de alumnos". At the top right is a blue "Insertar" button. Below the title is a table with three columns: "Codigo", "Nombre", and "Apellido". The table contains three rows of student data. To the right of each row are three buttons: "Ver" (light blue), "Editar" (light blue), and "Eliminar" (red).

Codigo	Nombre	Apellido	Ver	Editar	Eliminar
7	Jose Miguel	Oblitas Jimenez	Ver	Editar	Eliminar
6	Cristian Arturo	Cruz Benel	Ver	Editar	Eliminar
1	Andres	Paredes	Ver	Editar	Eliminar

Imagen 46 Vista principal del formulario que interactuara con las bases de datos

En el formulario de inserción se solicita un nombre de alumno y su apellido, una vez llenados todos los campos se procede a dar clic en el botón “Guardar” para almacenar el alumno y redirigir a la vista principal de la aplicación.



The screenshot shows a form titled "Insertar alumno". It has two input fields: "Nombre del alumno" and "Apellido del alumno". Below the fields is a blue "Guardar" button.

Imagen 47 Formulario para insertar un nuevo registro en la base de datos

Para Editar un alumno se selecciona el botón editar del respectivo alumno que se desea editar.

Administración de alumnos			Insertar		
Código	Nombre	Apellido			
7	Jose Miguel	Oblitas Jimenez	Ver	Editar	Eliminar
6	Cristian Arturo	Cruz Benel	Ver	Editar	Eliminar
1	Andres	Paredes	Ver	Editar	Eliminar

Imagen 48 Opción de editar un registro de la base de datos

El cual nos llevara a un formulario de edición, donde se nos mostraran los datos actuales del alumno con la opción de poder modificarlos y guardar los cambios hechos.

Editar alumno
Nombre del alumno
Jose Miguel Orlando
Apellido del alumno
Oblitas Jimenez
Guardar

Imagen 49 Formulario para editar un registro de la base de datos

En este caso se le agrego un tercer nombre al campo “Nombre del alumno”.

Una vez guardadas las modificaciones al del alumno se nos redijera a la vista principal de la aplicación y se nos mostrara un mensaje de notificación diciendo si se ha modificado correctamente o no.

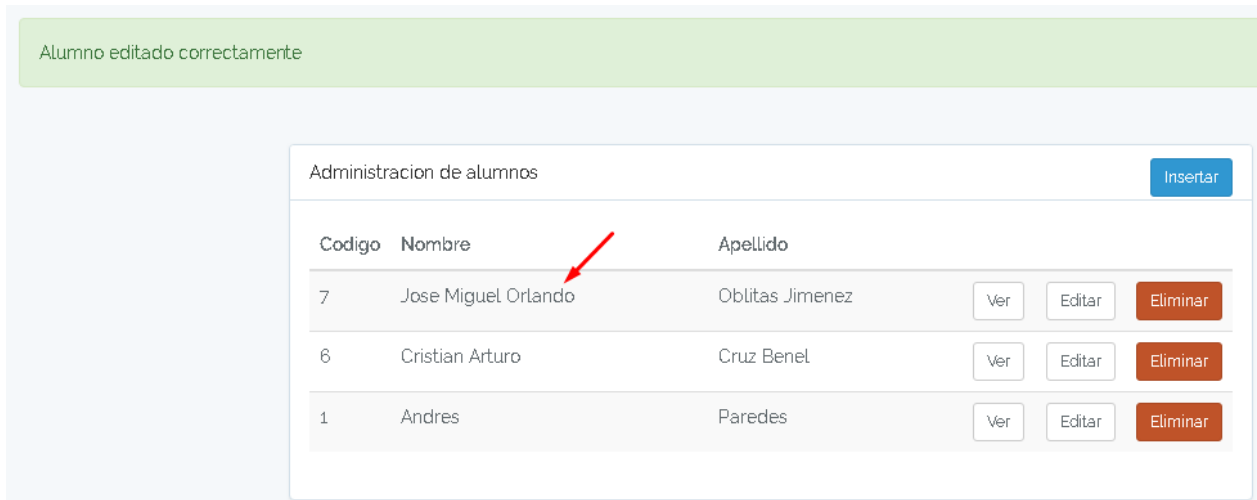


Imagen 50 Resultado después de la edición de un registro

Para eliminar un alumno se procede a hacer clic en la opción eliminar del respectivo alumno.



Imagen 51 Opción de eliminar un registro

La vista principal se recargara y nos mostrara un mensaje de notificación diciendo si se eliminó o no el alumno, de ser sí entonces el alumno ya no será mostrado en el listado.



Imagen 52 Resultado después de realizar la eliminación de un registro

Para que la aplicación anterior se ejecute correctamente en su servidor correspondiente se deben seguir los siguientes pasos:

Copiar todo el proyecto en la siguiente ruta del servidor (/var/www/html/), en esta locación es donde se ponen los archivos que serán mostrados por el servidor web.

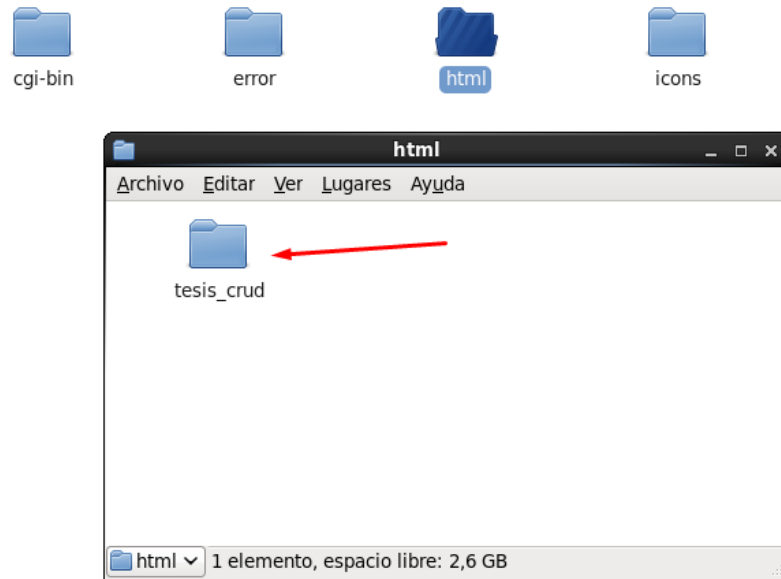


Imagen 53 Ubicación de los formularios en el servidor web

Ahora se debe modificar el archivo `httpd.conf` para direccionar la ruta raíz hacia la ubicación de nuestra aplicación, para ello se hará uso de un servidor web virtual.

```
[root@sucursal2 html]# nano /etc/httpd/conf/httpd.conf
```

Imagen 54 Comando para editar el archivo "httpd.conf"

```
#<VirtualHost *:80>
#   ServerAdmin webmaster@dummy-host.example.com
#   DocumentRoot /www/docs/dummy-host.example.com
#   ServerName dummy-host.example.com
#   ErrorLog logs/dummy-host.example.com-error_log
#   CustomLog logs/dummy-host.example.com-access_log common
#</VirtualHost>

<VirtualHost *:80>
  DocumentRoot "/var/www/html/tesis_crud/public"
  RewriteEngine On
  ServerName catchall

  <Directory "/var/www/html/tesis_crud/public">
    AllowOverride all
  </Directory>
</VirtualHost>
```

Imagen 55 Líneas de código para agregar un nuevo servidor virtual web

Procedemos a guardar el archivo y reiniciar el servicio `httpd` para que los cambios

hechos se ejecuten.

```
[root@sucursal2 html]# service httpd restart
```

Imagen 56 comando para reiniciar los servicios web apache

Configuramos los permisos necesarios para las carpetas

```
# chown -R apache:apache /var/www/html/tesis_crud
```

Imagen 57 Asignacion al usuario apache como propietario de la carpeta tesis_crud

```
# chmod -R 755 /var/www/html/tesis_crud/storage
```

Imagen 58 permisos de lectura y escritura para el usuario propietario de la carpeta "storage"

Por ultimo ejecutar el siguiente comando para que la aplicación pueda acceder a la base de datos postgresql.

```
# setsebool -P httpd_can_network_connect_db on
```

Imagen 59 Comando para permitir la conexion entre los formularios y la base de datos Postgresql

VIII.4.8. Replicación con Bucardo

El programa que se encargara de realizar la replicación entre las bases de datos será “**Bucardo**” en su versión 5.4.1, este programa de software libre puede ser utilizado en cualquier distribución de Linux ya sea Ubuntu, fedora, RedHat, CentOS,etc. en este caso lo utilizaremos sobre la distribución CentOS 6.3 para realizar sincronizaciones master-to-master es decir para poder leer y escribir sobre cualquier base de datos y que estas se sincronicen entre ellas.

Es necesario llevar a cabo una serie de instalaciones de las dependencias que bucardo requiere, tales como: perl, perl-DBD-Pg package, postgresql-plperl, , el modulo DBIx::Safe, perl-Time-HiRes, perl-boolean.

VIII.4.8.1. Instalacion de perl

```
yum -y install perl*
```

Imagen 60 Comandos para instalar los paquetes perl

VIII.4.8.2. Instalacion de Perl-DBD-Pgpackage

```
yum provides "*/DBD/Pg.pm"  
yum install perl-DBD-Pg
```

Imagen 61 Comando para instalar el paquete Perl-DBD-Pgpackage

VIII.4.8.3. Instalación de Postgresql-plperl

```
yum install postgresql-plperl
```

Imagen 62 Comando para instalar Postgresql-plperl

VIII.4.8.4. Instalación del módulo DBIx::Safe

Para realizarla instalación de este módulo es necesario descargar el archivo dbix_safe.tar.gz desde su página oficial http://bucardo.org/downloads/dbix_safe.tar.gz, copiar el archivo descargado hacia

alguna ruta del servidor donde se instalara Bucardo, en este caso se optó por copiarlo en el escritorio de CentOS.

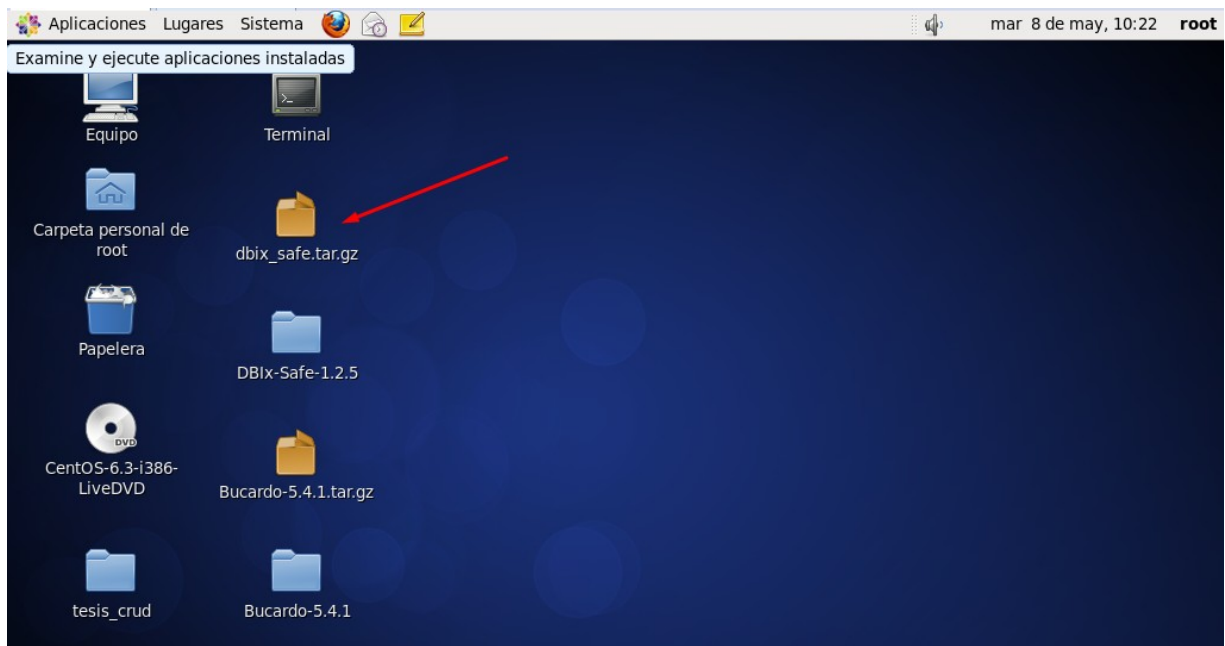


Imagen 63 Ubicación de donde se copió el archivo dbix_safe.tar.gz

Es necesario descomprimirlo y acceder a la carpeta descomprimida.

```
tar xvfz dbix_safe.tar.gz  
|cd DBIx-Safe-1.2.5
```

Imagen 64 descompresion del archivo dbix_safe.tar.gz

Una vez ubicados en la carpeta contenedora del módulo DBIX_SAFE ejecutamos los siguientes comandos para llevar a cabo su instalación.

```
perl Makefile.PL  
|make  
make test  
sudo make install
```

Imagen 65 Comandos para hacer la instalación del archivo dbix_safe.tar.gz

VIII.4.8.5. Instalación de perl-Time-HiRes

```
yum install perl-Time-HiRes
```

Imagen 66 comando para instalar perl-Time-Hires

VIII.4.8.6. Instalación de perl-boolean

```
yum install perl-boolean
```

Imagen 67 Comando para instalar perl-boolean

Por ultimo para tener listo todo para la instalación de bucardo es necesario crear una carpeta donde se guardaran todos los archivos .log que cree bucardo

```
mkdir /var/log/bucardo
```

Imagen 68 Creación de una carpeta para que bucardo guarde los .log que genera

```
chmod 777 /var/log/bucardo
```

Imagen 69 Asignación de permisos de lectura y escritura para bucardo en la carpeta /var/log/bucardo

VIII.4.8.7. Instalación de Bucardo

Cuando ya están instaladas todas las dependencias de Bucardo realizaremos los siguientes pasos para llevar a cabo con su instalación.

Descargar la versión 5.4.1 de bucardo desde el siguiente enlace <http://bucardo.org/downloads/Bucardo-5.4.1.tar.gz>.

Copiarlo a una ruta del servidor en este caso al escritorio de CentOS.

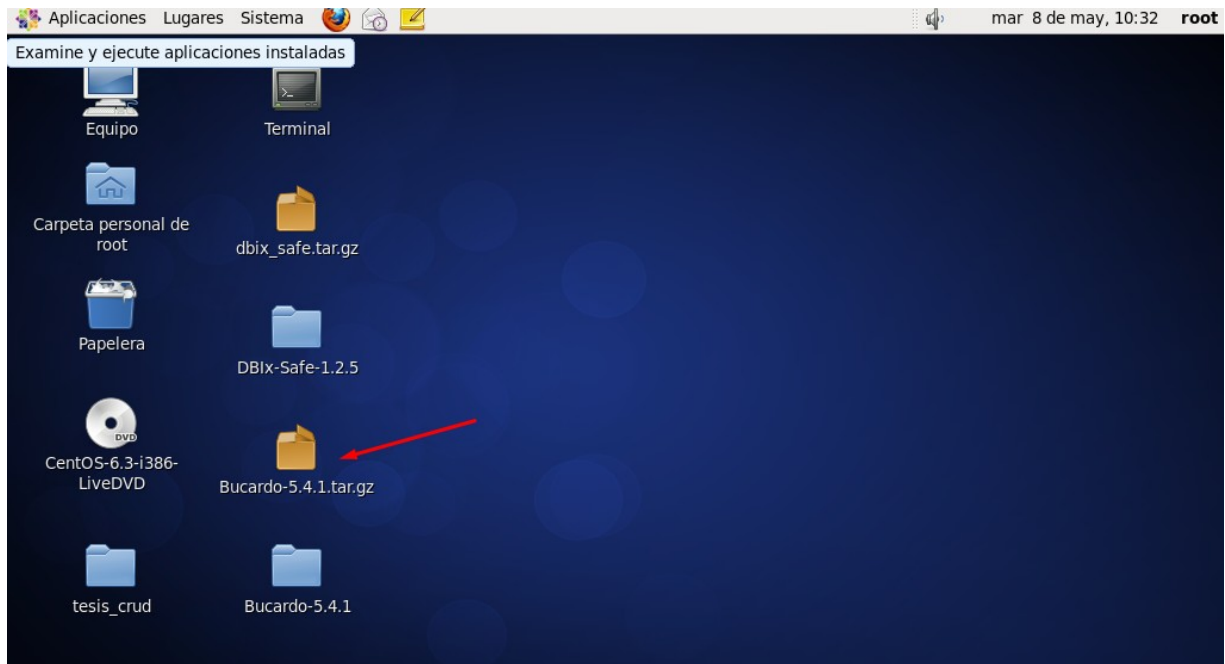


Imagen 70 Ubicación de donde se copió el instalador de bucardo

Ejecutar los siguientes comandos para descomprimirlo e instalarlo.

```
# tar xvfz Bucardo-5.4.1.tar.gz
# cd Bucardo-5.4.1
# perl Makefile.PL
# make
# sudo make install
```

Imagen 71 Comandos para descomprimir e instalar Bucardo 5.4.1

Ahora que todo lo necesario está instalado y configurado correctamente llevaremos a cabo la replicación para ello se deben seguir los pasos mencionados a continuación.

Configurar a bucardo para que reconozca las bases de datos que se sincronizaran.

```
[root@sucursal1 ~]# bucardo add database bdservidor01 dbname=bdservidor01
Added database "bdservidor01"

[root@sucursal1 ~]# bucardo add database bdservidor02 dbname=bdservidor02 host=sucursal2.tesis.com
Added database "bdservidor02"
```

Imagen 72 Agregación de las bases de datos que se replicarán

Ahora es necesario agregar las tablas de las bases de datos que se replicarán.

```
[root@sucursal1 ~]# bucardo add table public.alumno db=bdservidor01
Added the following tables or sequences:
public.alumno

[root@sucursal1 ~]# bucardo add table public.alumno db=bdservidor02
Added the following tables or sequences:
public.alumno
```

Imagen 73 Asignación de las tablas que se sincronizarán

Es necesario también configurar los **herd** que vienen a ser las relaciones que hay entre las bases de datos a replicar.

```
[root@sucursal1 ~]# bucardo add herd bdservidor01_bdservidor02 alumno db=bdservidor01
Created relgroup "bdservidor01_bdservidor02"
The following tables or sequences are now part of the relgroup "bdservidor01_bdservidor02":
public.alumno

[root@sucursal1 ~]# bucardo add herd bdservidor02_bdservidor01 alumno db=bdservidor02
Created relgroup "bdservidor02_bdservidor01"
The following tables or sequences are now part of the relgroup "bdservidor02_bdservidor01":
public.alumno
```

Imagen 74 Asignación de los herd

Para crear los sincronismos se siguen haciendo los siguientes pasos.

```
[root@sucursal1 ~]# bucardo add sync sync_bdservidor01 relgroup=bdservidor01_bdservidor02 dbs=bdservidor01,bdservidor02
Added sync "sync_bdservidor01"
Created a new dbgroup named "sync_bdservidor01 3"
```

```
[root@sucursal1 ~]# bucardo add sync sync_bdservidor02 relgroup=bdservidor02_bdservidor01 dbs=bdservidor02,bdservidor01
Added sync "sync_bdservidor02"
Created a new dbgroup named "sync_bdservidor02 3"
```

Imagen 75 Creación de los sincronismos en Bucardo

Por ultimo iniciamos la replicación.

```
[root@sucursal1 ~]# bucardo start
Checking for existing processes
Removing file "/tmp/bucardo.mcp.pid" with stale PID 15272
Removing file "/tmp/fullstopbucardo"
Starting Bucardo
- - -
```

Imagen 76 Comando para iniciar la replicación

Podemos ver el estado de la replicación entre los servidores ejecutando el comando "bucardo status":

```
[root@sucursal1 ~]# bucardo status
PID of Bucardo MCP: 3390
```

Name	State	Last good	Time	Last I/D	Last bad	Time
sync_bdservidor01	Good	13:35:42	1m 34s	0/0	none	
sync_bdservidor02	Good	13:35:42	1m 34s	0/0	none	

Imagen 77 Estatus actual de la replicación realizada

VIII.4.9. Automatización de la sincronización

Una vez analizadas las fluctuaciones de ancho de banda de la red se determinó que la sincronización se llevará a cabo todos a la 1:00 de la madrugada y terminará a las 5:00 am de todos los días del año, así mismo debido a que la sincronización entre todos los servidores se hará al día siguiente, es necesario tener en cuenta que la información estará desactualizada con un día de diferencia.

La herramienta para automatizar la replicación será CRONTAB que viene pre instalada en el sistema operativo utilizado (CentOS 6), para iniciar la configuración se debe modificar el archivo "/etc/crontab"



Imagen 78 Comando para editar el archivo crontab

Se procede a configurar los parámetros necesarios de inicio y detención de la sincronización automática.

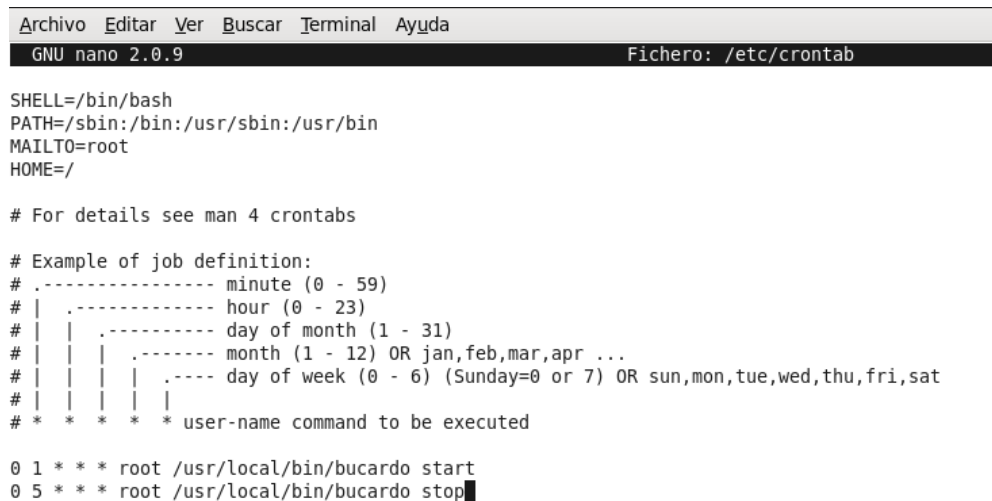


Imagen 79 Contenido del archivo crontab

Por último, se reinicia el demonio para que los cambios hechos los lea el programa.

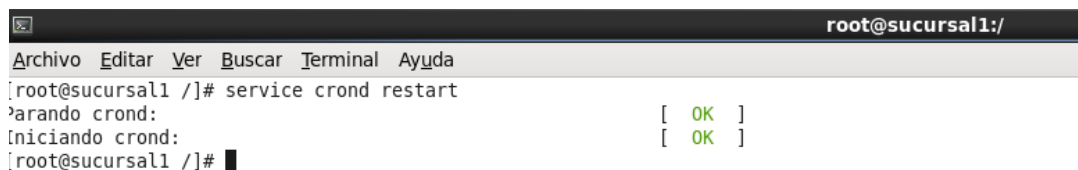


Imagen 80 Reinicio del demonio cron

I.1. Presupuesto

En este apartado hablaremos sobre los costos que conllevara la puesta en marcha del proyecto, los costos en los que se va a incurrir para el desarrollo de la implementación son evitados mediante el uso de software gratuito y de libre distribución.

Para establecer este análisis es necesario identificar los beneficios y ventajas que ofrecerá la herramienta a los usuarios. Como se mencionó en la descripción de la solución, el buen uso de esta herramienta permitirá que desarrolladores y administradores realicen la tarea de sincronizar la base de datos de manera rápida y segura, evitando cometer errores, los cuales pueden ser difíciles de encontrar en caso de que sea una sincronización que abarque muchas entidades de la base de datos.

Tomando en cuenta lo expuesto anteriormente, se puede decir que el proyecto es viable en los aspectos antes mencionados y los resultados que esperan obtener van a permitir que el usuario final tenga una herramienta confiable y automatizada para realizar la sincronización de bases de datos.

El análisis costo/beneficio, tiene por objetivo fundamental proporcionar una medida de los costos en que se incurre en la realización de un proyecto y comparar dichos costos previstos con los beneficios esperados de la realización del proyecto. Se contemplan únicamente los costos que influyen directamente en la propuesta.

I.1.1. Inversión

Los gastos a los que debemos incurrir en la primera instancia están determinados por la capacidad instalada con al que planteamos implementar el proyecto de esta investigación. Hemos decidido también incidir en el uso de software libre para así poder disminuir los gastos por las licencias de estos y comprar el equipamiento necesario para la producción que necesitamos.

INVERSIÓN	CANTIDAD	UNIDAD	PRECIO UNITARIO	COSTO TOTAL
Hardware				
Servidor de base de datos	3	Unidad	3000	9000
Monitor	3	Unidad	310	930
Teclado	3	Unidad	30	90
Mouse	3	Unidad	25	75
UPS	3	Unidad	210	630
Software				
CentOS	3	Unidad	0	0
PostgreSQL	3	Unidad	0	0
Bucardo	1	Unidad	0	0
Capacitación				
Manejo / Mantenimiento	1	Capacitación	120	120
Diseño y Implementación				
Profesional Informático	2	Honorarios	1250	2500
Documentación	2	Documento	150	300
TOTAL				13645

Tabla 14 Cuadro de la Inversión que se realizara

I.1.2. Beneficios

A) Beneficios Tangibles

Con el comparador y sincronizador de base de datos se podrá tener la información de los clientes tales como sus estados de cuenta, su récord de notas y toda información almacenada en la base de datos.

Asimismo, se evitarán las pérdidas y una mayor disponibilidad de la información referente a la base de datos de la Universidad Particular de Chiclayo.

Por lo explicado, se asume que lo propuesto será de gran ayuda para evitar toda serie de problemas que se suscitan en el día a día.

B) Beneficios Intangibles

- Disminución de errores del 100% al momento de acceder a las consultas referentes a los estudiantes de la Universidad Particular de Chiclayo.
- Satisfacción del 100% de los trabajadores y clientes que hacen uso de este servicio ya que podrán seguir con su trabajo sin ser interrumpidos de sus labores cotidianas.

Mes	Costo Inicial	Primer Mes	Recuperado
1	13645	2000	11645
2	11645	2000	9645
3	9645	2000	7645
4	7645	2000	5645
5	5645	2000	3645
6	3645	2000	1645
7	1645	2000	-355

Tabla 15 Cuadro para calcular en que mes se recuperar la inversión del proyecto

Se logró recuperar el costo total de la inversión al séptimo mes de haber implementado el comparador y sincronizador de bases de datos relacionales.

II. BIBLIOGRAFÍA

- A. C. (13 de Julio de 2015). *Desarrollolibre*. Obtenido de <http://www.desarrollolibre.net/blog/tema/188/linux/bucardo-el-sistema-de-replicacion-para-postgresql>
- Abraham Silberschatz, H. F. (2002). *FUNDAMENTOS DE BASES DE DATOS*. Madrid: MC GRAW HILL.
- Antonio Carrillo Ledesma, K. I. (2016). *Máquinas Virtuales*, editorial Alfaguara, Venezuela.
- Aulbach, A. (2002). *Manual de PHP*. Free Software Foundation, editorial Almadía, Argentina.
- Ballester, E. G. (s.f.), editorial Gallo negro, España.
- Bermúdez. (2008). *Configuración servidores en centos*, editorial Impedimenta, Madrid.
- Camps, P. R. (2005). *Bases de datos*, editorial Nevsky, Barcelona.
- Centro de Investigación de la Web. (2008). *Como funciona la web*, editorial Sexto Piso,. Chile: Claudio Gutiérrez Gallardo.
- Corona, A. E. (s.f.). PROTOCOLOS TCP/IP DE INTERNET. *Revista Digital Universitaria*., editorial Urano, Chile.
- Date, C. J. (2001). *INTRODUCCIÓN A LOS Sistemas De Bases De Datos*,: Pearson Educación, Editorial Suruela, Caracas.
- Denzer, P. (2002). *PostgreSQL*, editorial Safori, Costa rica.
- Fernández, V. (2002). *Manual de PHP*. Grupo de Documentación de PHP, editorial Lumen, Mexico.
- G. J., M. L., U. E., H. R., & C. A. (2012). *Replicación y Cluster de Bases de Datos*, editorial Herder, El Salvador: Lulu.

- Gilfillan, I. (s.f.). *la biblia mysql*. anaya multimedia, editorial Claridad, argentina.
- Jacqueline, G., M. L., U. E., H. R., & C. A. (2012). *Replicacion y Cluster de Bases de Datos*. El Salvador: Lulu.
- Jon Postel, S. C. (2009). *Redes de comunicaciones*. España: editorial Güimi.
- Jorge Ferrer, V. G. (2010). *Curso completo de HTML*, editorial minotauro, Caracas.
- Jorge Ferrer, V. G. (2014). *Curso completo de HTML*, editorial minotauro, Caracas..
- kabir, m. j. (2013). *La biblia del apache*, editorial grupo planeta, Barcelona.
- Lockhart, T. (2012). *Guia del Programador PostgreSQL*. Postgres Global Development Group.
- Márquez, J. E. (2005). *Transmisión de datos*.: Taller de Publicaciones de la Facultad de Ingeniería, editorial RBA, Venezuela.
- Martín, L. M. (octubre de 2010). *Diseño y Construcción de Bases de. Diseño y Construcción de Bases de*, editorial Nostra ediciones, Mexico.
- Ministerio de Educación, C. y. (2012). *Aulas en red. Aplicaciones y Servicios. Linux. Instituto Nacional de Tecnologías Educativas y de Formación del Profesorado*, 21.
- Richzendy. (11 de mayo de 2011). *richzendy*. Obtenido de <http://www.richzendy.org/2011/05/22/instalando-rubyrep-en-centos-5-x.html>
- Rosas, A. C. (2016). *Máquinas Virtuales*.
- Santis, A. H. (2012). *Transmisión de datos*. Tlalnepantla: Red Tercer Milenio.
- Sierra, A. A. (2002). *Curso básico de Linux*. SSA.
- Stallman, R. M. (2004). *Software libre para una sociedad libre*. Madrid: GNU Press.
- Tanenbaum, A. S. (2009). *SISTEMAS OPERATIVOS*. Atlacamilco: Cámara Nacional de la Industria.

TANENBAUM, A. S. (2009). *SISTEMAS OPERATIVOS*. Amsterdam, Holanda:
PEARSON EDUCACIÓN.

Vazquez, A. (2016). *CentOS Linux y Servicios de Red* Antonio Vazquez. Leanpub.

VMware, I. (2009). *Workstation User's Manual*.ediotrial Nation Up, Palo Alto.

III. ANEXOS