

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.  
FACULTAD 2.**



**Réplica de la base de datos del sistema SIGAC.**

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

**Autor(es):**

**Yissell Fernández Aguiar  
José Luis Sierra Cerra**

**Tutor:**

**Ing. Yasser Azán Basallo**

**Co-Tutor:**

**Ing. Jorge Luis Olmedo Flores**

**Consultante:**

**Norge Fajardo Vega**

**Ciudad Habana, Julio 2008**

**“Año 50 de la Revolución”**

*Nos hemos convertido en expertos buscadores de información: entre la ensalada de datos que nos ofrecen los medios, elegimos selectivamente como niños melindrosos.*

*- David Fryxell*

## **DECLARACIÓN DE AUTORÍA**

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

-----

**Yissell Fernández Aguiar**

-----

**José Luis Sierra Cerra**

-----

**Yasser Azán Basallo**

### **DATOS DE CONTACTO**

Yasser Azán Basallo

Ing. Ciencias Informáticas (año 2007)

Instructor recién graduado

Departamento de Ingeniería, Gestión de Software y Práctica Profesional

Jorge Luis Olmedo Flores

Ing. Máquinas Computadoras (año 1994)

MSc. Informática Aplicada (año 2003)

Especialista "A" en Ciencias Informáticas -Especialista Principal

Trabaja en el tema Bases de Datos desde el año 1998.

Norge Fajardo Borges

Ing. Ciencias Informáticas (año 2007)

Instructor recién graduado

Trabaja en el tema Bases de Datos desde el año 2007.

### AGRADECIMIENTOS

*Quisiéramos agradecerles a todas aquellas personas que de una forma u otra han colaborado en la realización de este trabajo. A nuestros familiares en general los que con su ayuda han hecho que llegáramos a ser unos profesionales. Al compañero Jorge Luis Olmedo por habernos ayudado tanto y facilitado tanta información que nos fue de gran ayuda para la realización de este trabajo. A los compañeros del nodo central de la dirección de INFOMED.*

*A Norge Fajardo que trabaja el tema de réplica en UCIFAR por brindarnos toda su experiencia en este tema.*

**DEDICATORIA**

*Yissell*

*Este trabajo se lo quiero dedicar a mi mamá que me ha ayudado tanto en mi vida y en mi carrera profesional, fundamentalmente a mi papá que aunque ya no está conmigo, lo recuerdo siempre por haber dedicado toda su vida a enseñarme todo lo que sé y por haber sido el impulsor principal en mi vida como estudiante desde pequeña. A mi novio por su dedicación y paciencia. A mis amistades por haberme soportado durante estos cinco años de carrera, y por haberme alentado en aquellos momentos en que el ánimo y la perseverancia no me acompañaban.*

*José Luis*

*Este trabajo se lo quiero dedicar a mi mamá que tanto me ha ayudado en mi vida y en mi carrera profesional, a mi abuela por ser tan comprensible y tan buena conmigo, a mi papá que me ha enseñado muchas cosas en la vida al cual siempre tengo de ejemplo. A mi novia que tanto me ha ayudado y me ha comprendido, a mis tíos, a mis hermanos, a todos mis amigos a mi cuñado, en fin a todos aquellos que hicieron posible que este sueño se hiciera realidad.*

### RESUMEN

En el presente trabajo se expone una propuesta para solucionar los problemas de resguardo de información existentes en el Ministerio de Auditoría y Control. En este Ministerio actualmente toda la información se maneja de forma manual, por lo que el objetivo de este trabajo se basa esencialmente en el estudio de las diferentes réplicas de bases de datos existentes en las distintas organizaciones de nuestro país y en el mundo, para así llegar a dar una propuesta de réplica de bases de datos en el Sistema Gestor de Base Datos (SGBD) PostgreSQL que se llevará a cabo en dicho ministerio para garantizar la seguridad e integridad de los datos para evitar pérdida de información que pudieran ser lamentable para el Ministerio de Auditoría y Control y para nuestro país, logrando de esta manera un resguardo de la información, a dicha propuesta se le realizaron pruebas de validación de datos y de tolerancia a fallos, para verificar su factibilidad, las cuales arrojaron resultados satisfactorios cumpliendo así el objetivo trazado por este trabajo.

#### **Palabras Claves:**

Réplica de base de datos, Sistema Gestor de Base de Datos, Slony-I.

ÍNDICE

INTRODUCCIÓN..... - 1 -

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA ..... - 4 -

1.1- Introducción..... - 4 -

1.2 ¿Qué son las bases de datos? ..... - 4 -

    1.2.1 Definición de base de datos ..... - 4 -

    1.2.2 Estructura de una Base de Datos ..... - 4 -

    1.2.3 Tipos de base de datos. .... - 5 -

        1.2.3.1 ¿Qué es una base de datos de fichero plano? ..... - 5 -

        1.2.3.2 ¿Qué es una base de datos relacional? ..... - 5 -

        1.2.3.3 ¿Qué es una base de datos orientada a objeto? ..... - 5 -

        1.2.3.4 ¿Qué es una base de datos híbrida?..... - 6 -

1.3 Sistema de Gestión de Base de Datos (SGBD)..... - 6 -

    1.3.1 Tipos de SGBD ..... - 6 -

    1.3.2 Diferentes sistemas de gestión de bases de datos. .... - 7 -

    1.3.3 PostgreSQL ..... - 7 -

        1.3.3.1 Resumen Histórico..... - 7 -

        1.3.3.2 Arquitectura del PostgreSQL..... - 8 -

        1.3.3.3 Arquitectura del Libpq..... - 9 -

        1.3.3.4 Características técnicas del PostgreSQL ..... - 10 -

        1.3.3.5 Ventajas y mejoras del PostgreSQL. .... - 10 -

        1.3.3.6 Desventajas del PostgreSQL. .... - 11 -

    1.3.4 Software Existente Relacionado ..... - 11 -

    1.3.5 Sistema Operativo utilizado. Software libre (Linux) ..... - 14 -

        1.3.5.1 Características de Linux ..... - 15 -

    1.3.6 PL/PGSQL..... - 15 -

    1.3.7 pgAdmin III..... - 16 -

1.4 Réplicas de Bases de Datos ..... - 16 -

    1.4.1-Tipos de réplicas ..... - 16 -

    1.4.2 Entornos de Réplica..... - 17 -

    1.4.3 Modelo de distribución de datos ..... - 17 -

    1.4.4 Técnicas de replicación ..... - 18 -

    1.4.5 Conflictos de Replicación ..... - 19 -

    1.4.6 Replicar datos en un entorno de servidor a servidor ..... - 20 -

        1.4.6.1 Mejorar la escalabilidad y la disponibilidad ..... - 20 -

    1.4.7 Replicar datos entre un servidor y los clientes..... - 24 -

    1.4.8 Intercambiar datos con usuarios móviles ..... - 25 -

    1.4.9 Aplicaciones de punto de venta (POS) para el consumidor..... - 27 -

    1.4.10 Integrar datos de varios sitios (cliente)..... - 27 -

1.5 Actualidad de las Réplicas de Bases de datos en Cuba y el mundo. .... - 28 -



1.6 Conclusiones.....	- 30 -
<b>CAPÍTULO 2: ESTUDIO DE RÉPLICAS DE DATOS UTILIZADAS EN DIFERENTES EMPRESAS. PROPUESTA DE RÉPLICA PARA EL MAC.....</b>	<b>- 31 -</b>
2.1 Introducción.....	- 31 -
2.2 Réplicas utilizadas en proyecto Servicios Autónomos de Registros y Notarias (SAREN).- 31 -	- 31 -
-	
2.2.1 Misión del proyecto.....	- 31 -
2.2.2 Réplica utilizada por SAREN.....	- 31 -
2.2.2.1 Modelo de replicación.....	- 33 -
2.2.2.2 Replicar hacia todas las BD locales la información del CD.....	- 33 -
2.2.2.3 Replicar hacia las BD locales solo la información correspondiente. ....	- 33 -
2.2.2.4 Replicar desde una oficina información hacia el CD. ....	- 34 -
2.2.2.5 Replicar desde una oficina información hacia el CD, la cual debe actualizarse en todos los servidores locales restantes. ....	- 34 -
2.2.2.6 Replicar desde una oficina información hacia el CD, la cual debe actualizarse en todos los servidores locales restantes. Replicar hacia todas las oficinas información agregada en el CD.....	- 35 -
2.2.2.7 Replicar en ambos sentidos. ....	- 36 -
2.2.3 Mecanismo implementado.....	- 36 -
2.2.3.1 Captura o detección de cambios.....	- 36 -
2.2.3.2 Aplicación de cambios o sincronización.....	- 38 -
2.2.3.3 Alcance del mecanismo.....	- 38 -
2.3.1 Misión de la Empresa.....	- 38 -
2.3.2 Réplica Usada por ETECSA.....	- 39 -
2.3.3 Snapshot es útil cuando:.....	- 39 -
2.3.3.1 ¿Cómo trabaja la Replicación Snapshot?.....	- 40 -
2.3.4 Servicios que brinda ETECSA en los que se utiliza la replicación:.....	- 40 -
2.3.5 Características de los servidores que utiliza ETECSA para cada uno de los servicios que brinda:.....	- 41 -
2.4 Réplica usada en el Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR).....	- 42 -
2.4.1 Información de la Organización. ....	- 42 -
2.4.2 Réplica del MINFAR.....	- 42 -
2.4.1.1 Esquema de un nodo a la hora de Replicar Dato.....	- 43 -
2.5 Réplica de INFOMED.....	- 46 -
2.5.1 Misión de Infomed (Red de salud de Cuba).....	- 46 -
2.5.2 Estructura de red Infomed.....	- 46 -
2.5.3 Réplica de datos de Infomed.....	- 46 -
2.5.4 Propiedades de los servidores y de la red.....	- 47 -
2.6 Propuesta de Réplica para el Ministerio de Auditoría y Control (MAC).....	- 48 -

---

2.6.1 Características de cada uno de los dispositivos con los que cuenta el MAC .....	- 51 -
2.6.1.1 Características de los servidores (Nodo Central) .....	- 52 -
2.6.1.2 Características de las Estaciones de Trabajo (Organismo Central) .....	- 53 -
2.6.1.3 Características de los servidores en las Delegaciones .....	- 54 -
2.6.1.4 Características de las Estaciones de Trabajo (Delegaciones Provinciales) .....	- 54 -
2.7 Conclusiones.....	- 55 -
<b>CAPÍTULO 3: VALIDACIÓN Y RESULTADOS DE LA PROPUESTA DE RÉPLICA REALIZADA PARA EL MAC.....</b>	<b>- 56 -</b>
3.1 Introducción.....	- 56 -
3.2 Descripción del proceso de réplica.....	- 56 -
3.2.1 Pasos a seguir para replicar de un nodo maestro a un nodo esclavo.....	- 56 -
3.3-Pruebas realizadas a la propuesta del MAC.....	- 59 -
3.3.1-Prueba #1 .....	- 60 -
3.3.2-Prueba #2.....	- 60 -
3.3.3-Prueba #3.....	- 61 -
3.3.4-Prueba #4.....	- 62 -
3.4 Conclusiones.....	- 62 -
<b>CONCLUSIONES.....</b>	<b>- 63 -</b>
<b>RECOMENDACIONES.....</b>	<b>- 64 -</b>
<b>BIBLIOGRAFÍA.....</b>	<b>- 65 -</b>
<b>ANEXOS .....</b>	<b>- 68 -</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>- 71 -</b>

## INTRODUCCIÓN

Nuestro país en su larga lucha por alcanzar el desarrollo en todas las esferas de la sociedad se ha visto enfrascado en la ardua misión de llegar a ser el país más culto del mundo, se dio a la tarea de informatizar cada uno de los sectores tanto sociales, políticos como económicos, con el uso de las Tecnologías de la Informática y las Comunicaciones (TIC). Nuestra Universidad de las Ciencias Informáticas (UCI), fundada en el calor de la batalla de ideas forma parte de este proyecto de la revolución y una de las tareas que se le ha otorgado es la informatización de varios ministerios del país de los cuales podemos citar el Ministerio de Auditoría y Control (MAC).

Para este ministerio es trascendental mantener toda la información que aquí se maneja de forma segura y actualizada. Todo este proceso de actualización de datos se realiza mediante la red con la utilización de correo electrónico o correos de Cuba, por lo que toda esta información que se manda está propensa a pérdidas o a que muchas veces se guarde información repetida porque no hay constancia de que ésta haya sido guardada en algún lugar, ya que se realiza de forma manual y se coloca en gavetas, cajas, oficinas, etc. Lo que no garantiza ninguna seguridad para la misma, pues todo este proceso depende del factor humano con el consiguiente incumplimiento de los plazos requeridos por algunos, provocando la demora de la gestión de la información, así mismo dificultando la toma eficiente y rápida de decisiones.

Debido a todos estos problemas que presenta esta institución y teniendo en cuenta cada uno de los intereses y necesidades del ministerio se llegó a la conclusión de la necesidad que tiene éste para un control de todas las actividades que se llevan a cabo en el mismo, de la utilización de un gestor de base de datos para evitar pérdida de la información, así como de réplicas de base de datos para el aseguramiento de toda la información sin tener que afectar el trabajo de los usuarios, ya que siempre se debe contar con una copia actualizada de los datos para que en caso de posibles desastres informáticos como: virus, pérdidas de servidores, fallas en las redes y otros, la información no se pierda.

Si un servidor de base de datos deja de funcionar correctamente por problemas en la red o factores ajenos a la misma, se tendrá una réplica en otra base de datos que eliminará el problema existente, constituyendo así la réplica de base de datos un factor importante en todo este intercambio de información entre el ministerio y sus sucursales, pues de esta manera se podrá evitar que toda esta

información en caso de fallas en algunos de sus servidores se mantenga a salvo en las copias de seguridad, garantizando la integridad de los datos, ya que el ministerio no se puede dar el lujo de perder los datos al ser éstos confidenciales y de mucha importancia, por la información sensible que maneja sobre las empresas y sobre el gobierno, ya que las empresas que se recuperan con rapidez son las que ganan. La clave para recuperarse rápidamente pasa por contar con una copia actualizada de los datos de negocio críticos en un servidor que se encuentre listo y en espera.

Por lo que esto conlleva a nuestro **Problema Científico**: ¿Cómo garantizar la integridad y seguridad de los datos en el Ministerio de Auditoría y Control?

Para dar solución a nuestro problema científico nos planteamos el siguiente **Objeto de Estudio**: Réplicas de Bases de Datos en el SGBD PostgreSQL, y como **Campo de acción**: Réplicas de base de datos con el gestor de base de datos PostgreSQL para el Ministerio de Auditoría y Control.

**Objetivos Generales**: Dar una propuesta de solución de réplica de base de datos para el Ministerio de Auditoría y Control.

### Preguntas científicas

- ❖ ¿Qué mejoras pudieran introducirse en el Ministerio de Auditoría y Control que evitarán las pérdidas y errores en la información?
  
- ❖ ¿Qué se tiene que tener en cuenta para un resguardo de la información en el Ministerio de Auditoría y Control?

Todos los objetivos anteriormente expuestos se cumplen mediante las **Tareas de investigaciones siguientes**:

1. Realizar entrevistas y encuestas a personas especializadas en el tema que aporten elementos de interés al desarrollo de la investigación.
2. Realizar un estudio sobre las distintas réplicas de base de datos en PostgreSQL.
3. Estudiar los componentes en el gestor de base de datos PostgreSQL que se utilizan para la replicación.
4. Estudiar y analizar trabajos ya existentes sobre réplica de base datos en PostgreSQL.

## Métodos Científicos

### Métodos Teóricos:

El método **analítico-sintético** permitió el desglose de toda la información encontrada y llegar a conclusiones de la investigación para poder determinar que fue lo más apropiado para el problema planteado.

El método **modelación** permitió representar a través de distintos esquemas, la solución encontrada al problema y entender la arquitectura en la cual está conformado el SGBD PostgreSQL.

### Métodos empíricos:

El método de **observación** permitió tener una visión de las condiciones en las que se encontraba el ministerio para de esta manera darle solución.

El método de **entrevista** permitió obtener información que nos ayudó en la realización de nuestra investigación para darle solución a los objetivos propuestos.

El método de **experimentación** permitió mediante pruebas realizadas a nuestra propuesta ver la factibilidad de la misma.

Este trabajo esta compuesto de tres capítulos:

**En el capítulo 1:** En el primer capítulo se tratan aquellos temas que constituyen la fundamentación teórica de la investigación a realizar. Incluye un estado del arte del tema tratado en el ámbito tanto nacional como internacional, de las tendencias, técnicas, tecnologías, metodologías y software existentes que de una forma u otra están relacionados con el tema que se aborda, profundizando en ellos, así como la citación de los distintos tipos de réplicas existentes.

**En el capítulo 2:** En el capítulo 2 se realizará un estudio de las diversas réplicas de bases de datos utilizadas en algunas empresas, así como la cantidad de servidores utilizados en algunas de ellas, para así de una forma u otra llegar a dar la mejor propuesta de réplica que se utilizará en el ministerio.

**En el capítulo 3:** En el capítulo 3 haremos diferentes pruebas a la propuesta que se propondrá para el MAC para ver cada uno de los resultados que se obtendrán y verificar de esta manera si la propuesta es factible.

## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### 1.1- Introducción

En el presente capítulo se brinda una visión general de algunos aspectos relacionados con la réplica de base de datos en PostgreSQL y más específicamente la aplicación de estos conceptos en las bases de datos del Ministerio de Auditoría y Control. Por lo que a continuación se da una descripción de los principales conceptos relacionados con el problema a resolver, las tecnologías utilizadas, así como el uso de éstas en Cuba y el mundo en la actualidad.

### 1.2 ¿Qué son las bases de datos?

No podemos hablar de réplicas de bases de datos (BD) sin antes referirnos a ellas, el término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, United States of American (USA). Por lo que se puede definir que :

#### 1.2.1 Definición de base de datos

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular (1).

#### 1.2.2 Estructura de una Base de Datos

Una base de datos, a fin de ordenar la información de manera lógica, posee un orden que debe ser cumplido para acceder a la información de manera coherente. Cada base de datos contiene una o más tablas, que cumplen la función de contener los campos. Es necesario señalar que no todas las bases de datos presentan este orden jerárquico ya que de acuerdo a su clasificación pueden presentar otras estructuras jerárquicas adicionales a la que a continuación se exponen:

Por consiguiente una base de datos posee el siguiente orden jerárquico:

- ❖ Tablas
- ❖ Campos
- ❖ Registros
- ❖ Lenguaje SQL

## 1.2.3 Tipos de base de datos.

Las bases de datos se pueden dividir en cuatro tipos básicos:

- Bases de datos de fichero plano (o ficheros por bloques).
- Bases de datos relacionales.
- Bases de datos orientadas a objetos.
- Bases de datos híbridas.

A continuación se exponen de modo sencillo algunas características de estos cuatro tipos de bases de datos anteriormente mencionados.

### 1.2.3.1 ¿Qué es una base de datos de fichero plano?

Consisten en ficheros de texto divididos en filas y columnas. Estas bases de datos son las más primitivas y quizás ni tan siquiera merezcan considerarse como tales. Pueden ser útiles para aplicaciones muy simples, pero no para aplicaciones medianas o complejas, debido a sus grandes limitaciones (2).

### 1.2.3.2 ¿Qué es una base de datos relacional?

Su nombre proviene de su gran ventaja sobre las bases de datos de fichero plano: la posibilidad de relacionar varias tablas de datos entre sí, compartiendo información y evitando la duplicidad y los problemas que ello conlleva (espacio de almacenamiento y redundancia). Existen numerosas bases de datos relacionales para distintas plataformas (Access, Paradox, Oracle, Sybase) y son ampliamente utilizadas. Sin embargo, tienen un punto débil: la mayoría de ellas no admiten la incorporación de objetos multimedia tales como sonidos, imágenes o animaciones (2).

### 1.2.3.3 ¿Qué es una base de datos orientada a objeto?

Incorporan el paradigma de la Orientación a Objetos (OO) a las bases de datos. La base de datos está constituida por objetos, que pueden ser de muy diversos tipos, y sobre los cuales se encuentran definidas operaciones. Las bases de datos orientadas a objetos pueden manejar información binaria (como objetos multimedia) de una forma eficiente. Su limitación suele residir en su especialización, ya que suelen estar diseñadas para un tipo particular de objetos (2).

#### 1.2.3.4 ¿Qué es una base de datos híbrida?

Las bases de datos híbridas combinan características de las bases de datos relacionales y las bases de datos orientadas a objetos. Manejan datos textuales y datos binarios, a los cuales se extienden las posibilidades de consulta (2).

### 1.3 Sistema de Gestión de Base de Datos (SGBD)

Un Sistema de Gestión de Bases de Datos (SGBD) es el conjunto de programas que permiten definir, manipular y utilizar la información que contienen las bases de datos, realizan todas las tareas de administración necesarias para mantenerlas operativas, mantener su integridad, confidencialidad y seguridad. Una base de datos nunca se accede o manipula directamente sino a través del SGBD. Se puede considerar al SGBD como la interfaz entre el usuario y la base de datos (3).

#### 1.3.1 Tipos de SGBD

En un SGBD existen dos componentes importantes como son: cliente y servidor. El cliente es una aplicación y el servidor aquel en el que se encuentra el SGBD y la propia base de datos. Ahora de aquí se desprenden los diferentes tipos de SGBD entre ellos tenemos:

- **SGBD centralizados:** Los sistemas de bases de datos centralizados son aquellos que se ejecutan en un único sistema informático sin interactuar con ninguna otra computadora. Tales sistemas comprenden el rango desde los sistemas de bases de datos monousuarios ejecutándose en computadoras personales hasta los sistemas de bases de datos de alto rendimiento ejecutándose en grandes sistemas (3).
- **SGBD cliente – servidor:** Un sistema cliente-servidor es aquel en el que uno o más clientes y uno o más servidores, conjuntamente con un sistema operativo subyacente y un sistema de comunicación entre procesos, forma un sistema compuesto que permite cómputo distribuido, análisis, y presentación de los datos. Si existen múltiples servidores de procesamiento de base de datos, cada uno de ellos deberá procesar una base de datos distinta, para que el sistema sea considerado un sistema cliente-servidor. Cuando dos servidores procesan la misma base de datos, el sistema ya no se llama un sistema cliente-servidor, sino que se trata de un sistema de base de datos distribuido (3).



- **SGBD paralelos:** Los sistemas paralelos de base de datos se ejecuta sobre múltiples procesadores y utilizando múltiples discos y que está diseñado para ejecutar las operaciones en paralelo. Están basados en un sistema de compartición de recursos, dando lugar a arquitecturas paralelas de memoria compartida, de disco compartido, jerárquicas (híbridas) ó sin compartición, según sea el recurso que se esté compartiendo (4).
  
- **SGBD distribuidos:** Un sistema de base de datos distribuido (SBDD) consiste de una colección de lugares, conectados a través de alguna red de comunicaciones y cumple que:
  - Cada lugar es un SGBD que puede operar solo.
  - Los usuarios pueden acceder a los datos de cualquier lugar como si fueran datos locales.
  - Para el usuario el sistema distribuido debe “verse” igual que un sistema No-Distribuido.
  - Un SBDD se clasifica en homogéneo si cada lugar tiene el mismo SGBD, de otro modo se denomina heterogéneo (5).

### 1.3.2 Diferentes sistemas de gestión de bases de datos.

Rapidez, efectividad en los procesos y los grandes flujos de información están como primera necesidad a la hora de optimizar servicios y productos. Ante ésta notable demanda de soluciones informáticas han surgido multitud de gestores de bases de datos, siendo estos programas que permiten manejar la información de modo sencillo y que prestan servicios para el desarrollo y el manejo de bases de datos. Entre los que se encuentran Oracle, Microsoft SQL Server y Borland Interbase y en el mundo del software libre MySQL (MySQL Server) y PostgreSQL que será en el cual haremos énfasis por pertenecer a nuestro objeto de estudio.

### 1.3.3 PostgreSQL

#### 1.3.3.1 Resumen Histórico

PostgreSQL es un servidor de base de datos relacional, liberado bajo la licencia BSD<sup>1</sup>. Es una alternativa a otros sistemas de bases de datos de código abierto (como MySQL), así como sistemas propietarios como Oracle. PostgreSQL es el último resultado de una larga evolución comenzada con el

---

<sup>1</sup> La **licencia BSD** es la licencia de software otorgada principalmente para los sistemas BSD (*Berkeley Software Distribution*). Pertenece al grupo de licencias de software Libre. La licencia BSD permite el uso del código fuente en software no libre.

proyecto Ingres<sup>2</sup> en la Universidad de Berkeley. El líder del proyecto, Michael Stonebraker abandonó Berkeley para comercializar Ingres en 1982, el proyecto resultante, llamado Postgres, era orientado a introducir la menor cantidad posible de funcionalidades para completar el soporte de tipos.

A pesar de que el proyecto Postgres hubiese finalizado oficialmente, la licencia BSD bajo la cual Postgres había sido liberado, permitió a desarrolladores de código abierto el obtener una copia del código para continuar su desarrollo. La empresa Pervasive que comercializa su motor propietario PervasiveSQL, actualmente también distribuye una versión libre basada en PostgreSQL. A continuación se describen algunas características y funcionalidades del mismo.

### **1.3.3.2 Arquitectura del PostgreSQL.**

El siguiente gráfico muestra de forma esquemática las entidades involucradas en el funcionamiento normal del gestor de bases de datos:

---

<sup>2</sup> Proyecto, liderado por Michael Stonebraker, fue uno de los primeros intentos en implementar un motor de base de datos relacional.

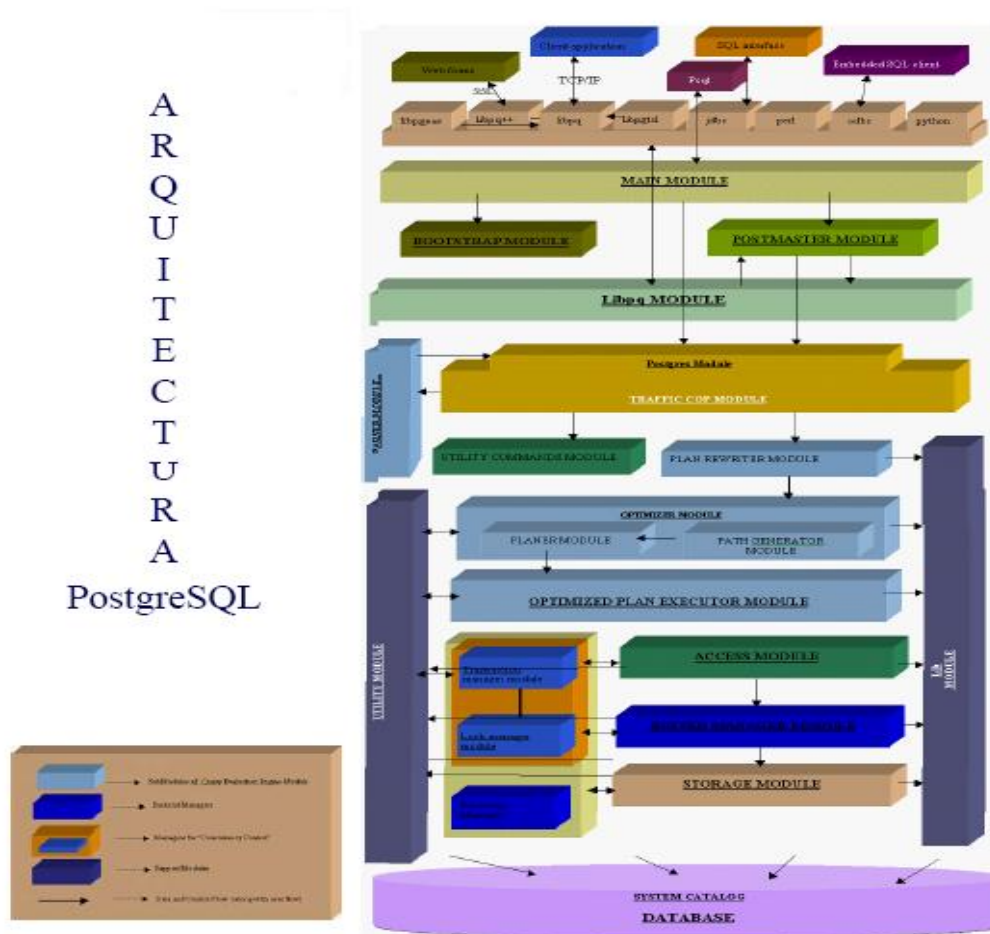


Figura 1.1 Arquitectura PostgreSQL

### 1.3.3.3 Arquitectura del Libpq

1) **Libpq** son responsables de manejar la comunicación con los procesos del cliente:

- establecer la conexión al postmaster.
- obtención del hilo de rosca del servidor del postgres para la sesión operacional.

2) El **Servidor** se compone de dos subsistemas: el **postmaster** y el **servidor del postgres**.

El **postmaster** es responsable de aceptar la petición de conexión entrante del cliente, de realizar control de la autenticación y de acceso en la petición del cliente, y de establecer al cliente la comunicación del servidor del postgres.

El **servidor del postgres** maneja todos los queries y comandos del cliente.

3) El **Store Manager** es responsable de la gestión de la memoria externa general y control de recurso en el back-end incluyendo la gerencia de almacenador intermediario compartido, de la gerencia de archivo, del control de la consistencia y del encargado de la cerradura (6)

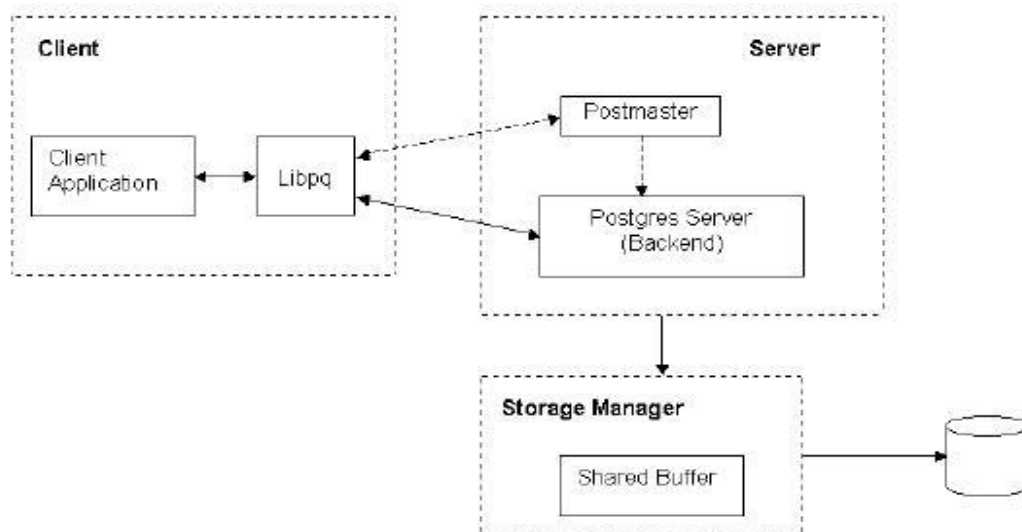


Figura 1.2 Arquitectura Libpq

#### 1.3.3.4 Características técnicas del PostgreSQL

Dentro de las características técnicas que posee el PostgreSQL se encuentran:

- Réplica (soluciones no comerciales y comerciales) que permite la duplicación de la base de datos principal a las máquinas auxiliares múltiples.
- Interfaces naturales para ODBC, JDBC, C, C++, PHP, Perl, TCL, ECPG, Python, y Ruby.
- Lenguaje procedural.
- Índices funcionales y parciales.
- Autenticación natural para kerberos (6).

#### 1.3.3.5 Ventajas y mejoras del PostgreSQL.

- Una ayuda mejor que los vendedores propietarios.
- Ahorro significativo en proveer de personal costes.
- Extensible

- Plataforma cruzada.
- Tiene soporte para triggers y procedimientos en el servidor.
- Diseñado para los ambientes de alto volumen (6).

### 1.3.3.6 Desventajas del PostgreSQL.

- Consume muchos recursos.
- Límite del tamaño de cada fila de las tablas a 8k (se puede ampliar a 32k recompilando, pero con un coste añadido en el rendimiento).
- Es un SGBD lento (6).

### 1.3.4 Software Existente Relacionado

Existen en el mundo múltiples soluciones que permiten llevar a cabo la réplica de datos enfocadas al gestor de base de datos utilizado en cuestión. Pueden citarse entre ellos Postgres-R, PgReplicator, Usogres, ERserver. Estas aplicaciones se dividen de acuerdo a los entornos de réplica donde pueden ser aplicadas, para el entorno “maestro-esclavo”, la más popular es:



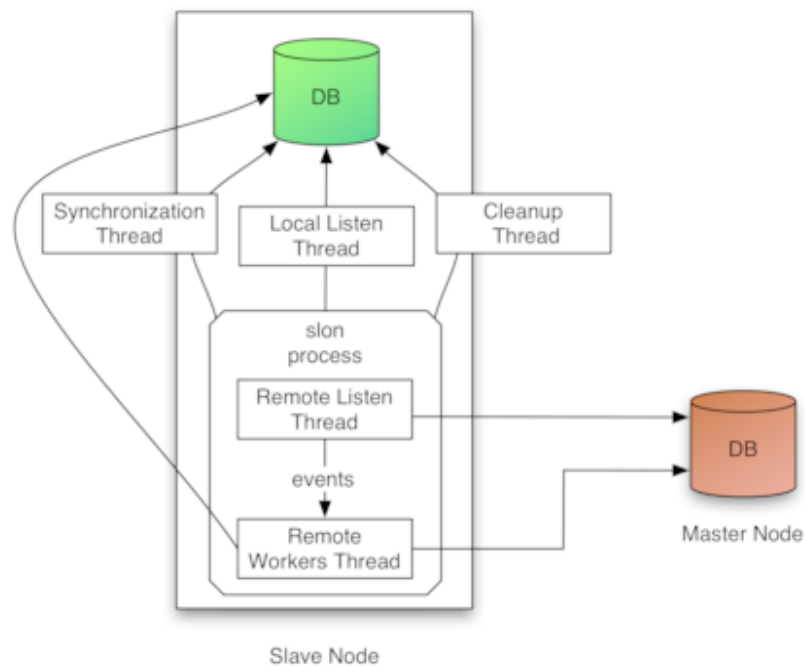
Figura 1.3 Logo del Slony I

Slony es el plural del elefante en el idioma ruso. La mascota para Slony, Slon es una muy buena variación del usual elefante mascota de PostgreSQL.

El cuadro grande para el desarrollo de Slony-I es que es un sistema maestro-esclavo de la réplica que incluye todas las características y capacidades necesitadas para replegar bases de datos grandes a un número razonablemente limitado de sistemas auxiliares. Si el número de servidores crece más allá de ese, el coste de comunicaciones aumentará prohibitivamente, y las ventajas incrementales de tener servidores múltiples fallarán en ese punto. Slony-I es un sistema diseñado para el uso en los centros de datos y los sitios de reserva, donde está el modo de operación normal que todos los nodos están disponibles.

Slony-I implementa la réplica asincrónica, usando disparadores para determinar las actualizaciones de las tablas, donde un solo “origen” (maestro) se puede replegar a los “suscriptores múltiples” (esclavos) incluyendo suscriptores conectados en cascada.

Slony I realiza una réplica de espejos, exactamente igual al origen de datos, no es posible actualizar los datos a medida que se produce algún cambio en ellos.



**Figura 1.4 Funcionamiento del Slony I**

Como se puede ver en la Figura 1.4, Slon es el principal motor de replicación, Slon, hace uso de muchos hilos. El hilo de sincronización verifica en un intervalo configurable si ha habido actividad de base de datos replicable, generando eventos de Sincronización (SYNC) si esta actividad ocurre. El hilo de escucha local escucha nuevos eventos de configuración y modificaciones de la configuración del clúster y la configuración en memoria del proceso Slon en consecuencia.

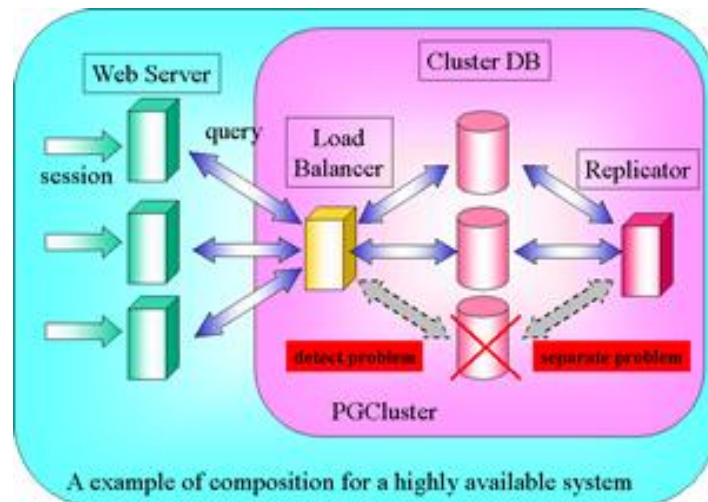
Como su nombre indica, el hilo de limpieza realiza el mantenimiento en el esquema Slony-I, al igual que la eliminación de antiguos eventos, o vacío de las tablas. El hilo de escucha remota se conecta al nodo remoto de la base de datos para recibir eventos de su proveedor de evento. Cuando se reciben eventos o confirmaciones, se selecciona la información correspondiente y se alimenta la cola de mensajes internos del hilo de trabajadores remoto. El hilo de trabajadores remoto, uno por cada nodo remoto, que tiene la replicación de datos actual, almacenamiento de eventos y la generación de confirmaciones. En cualquier momento, el esclavo sabe exactamente lo que los grupos de operaciones han consumido.

Para el caso del entorno “multi-maestro” es:

### **PgCluster**

Es un sistema de replicación sincrónico, “multi-maestro” para PostgreSQL.

PgCluster consiste en tres tipos de servidores, un servidor para balance de carga, clúster DataBase (DB) y un servidor de réplica.



**Figura 1.5 Funcionamiento del PgCluster**

Tiene dos funciones principales:

#### ➤ **Compartir carga**

La carga de la sesión de una petición de la referencia se distribuye. Es eficaz en el uso del Web con el cual una petición de la referencia vierte adentro.

Un objeto de la réplica se puede especificar por la tabla. Cuando diferencian las tablas que reciben una petición de puesta al día y una petición de la referencia, el PgCluster puede distribuir la tabla que recibe una petición de puesta al día y puede reproducir solamente la tabla que recibe una petición de la referencia.

#### ➤ **Alta disponibilidad**

Cuando la falta ocurre en el DB del racimo, un balanceador de la carga y un DB separado de la falta del servidor de la réplica de un sistema, y continúa el servicio usando el DB restante. Puesto que la separación del DB de la falta y la continuación del servicio se realizan simultáneamente, la mayoría de la hora de parada del servicio se hace a 0.

El DB del racimo que la reparación acabó se puede restaurar dinámicamente a un sistema, sin parar servicio.

Los datos se copian automáticamente al DB restaurado o agregado del otro DB. La pregunta que recibió durante la restauración isexecuted del servidor de la réplica después de la restauración.

Este último con objetivos definidos para soportar una alta disponibilidad, para controlar los fallos de los servidores.

Ambos sistemas son internacionales, en Cuba no se ha dado ninguna solución al respecto sobre PgCluster, pero si de Slony-I en la Universidad de las Ciencias Informáticas donde se ha trabajado en el tema.

El problema va por el lado de elegir la mejor opción para replicar bases de datos en PostgreSQL y que cumpla con las funcionalidades que se necesitan desarrollar en el Ministerio de Auditoría y Control.

### **1.3.5 Sistema Operativo utilizado. Software libre (Linux)**

Linux es una muy buena alternativa frente a los demás sistemas operativos. Más allá de las ventajas evidentes de costo, ofrece algunas características muy notables.

En comparación con las otras versiones de Unix para computadores, la velocidad y confiabilidad de Linux son muy superiores. También está en ventaja sobre la disponibilidad de aplicaciones, ya que no hay mucha difusión de estos otros Unixes (como Solaris, XENIX o SCO) entre los usuarios de computadores por sus altos costos.

Comparado con sistemas operativos como las diferentes versiones de Microsoft Windows, Linux también sale ganando. Los bajos requisitos de hardware permiten hacer un sistema potente y útil de aquel 486 que algunos guardan en un armario. Esta misma característica permite aprovechar al máximo las capacidades de las computadoras más modernas. Es poco práctico tener un computador con 16 Mega Bytes (Mb) de RAM y ponerle un sistema operativo que ocupa 13 (que es lo que reporta sobre Windows 95 el System Information de Symantec). No solo es superior respecto a el sistema de multitarea y de administración de memoria, sino también en la capacidades de networking (conectividad a redes) y de multiusuario. La única desventaja de Linux frente a estos sistemas, es la menor disponibilidad de software, pero este problema disminuye con cada nuevo programa que se escribe para el proyecto General Public License (GNU), y con algunas empresas que están desarrollando software comercial para Linux. A continuacion se presentan varias características que son representativas para linux .



### 1.3.5.1 Características de Linux

- **Multitarea:** La palabra multitarea describe la habilidad de ejecutar varios programas al mismo tiempo. Linux utiliza la llamada multitarea preventiva, la cual asegura que todos los programas que se están utilizando en un momento dado serán ejecutados, siendo el sistema operativo el encargado de ceder tiempo de microprocesador a cada programa.
- **Multiusuario:** Muchos usuarios usando la misma máquina al mismo tiempo.
- **Multiplataforma:** Las plataformas en las que en un principio se puede utilizar Linux son 386-, 486-, Pentium, Pentium Pro, Pentium II, Amiga y Atari, también existen versiones para su utilización en otras plataformas, como Alpha, ARM, MIPS, PowerPC y SPARC.
- **Multiprocesador:** Soporte para sistemas con más de un procesador está disponible para Intel y SPARC.
- Funciona en modo protegido 386.
- Protección de la memoria entre procesos, de manera que uno de ellos no pueda colgar el sistema.
- Carga de ejecutables por demanda: Linux sólo lee del disco aquellas partes de un programa que están siendo usadas actualmente.

### 1.3.6 PL/PGSQL

PL/PGSQL (Procedural Language/PostgreSQL Structured Query Language) es un lenguaje imperativo provisto por el gestor de base de datos PostgreSQL. Permite ejecutar comandos SQL mediante un lenguaje de sentencias imperativas y uso de funciones, dando mucho más control automático que las sentencias SQL básicas. Desde PL/PGSQL se pueden realizar cálculos complejos y crear nuevos tipos de datos de usuario. Como un verdadero lenguaje de programación, dispone de estructuras de control repetitivas y condicionales, además de la posibilidad de creación de funciones que pueden ser llamadas en sentencias SQL normales o ejecutadas en eventos de tipo disparador (trigger). Las funciones escritas en PL/PGSQL aceptan argumentos y pueden devolver valores de tipo básico o de tipo complejo (por ejemplo, registros, vectores, conjuntos o incluso tablas), permitiéndose tipificación polimórfica para funciones abstractas o genéricas (referencia a variables de tipo objeto).

## 1.3.7 pgAdmin III

Es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia Open Source. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres. pgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un editor de código de la parte del servidor, un agente para lanzar scripts programados y soporte para el motor de replicación Slony-I que es un software de replicación de bases de datos.

## 1.4 Réplicas de Bases de Datos

La réplica es el proceso de compartir datos entre las bases de datos en diversas localizaciones. Usando la réplica, se crean copias de la base de datos y se comparte la copia con diversos usuarios. Esto permite que realicen cambios a su copia local de la base de datos y que sincronicen más adelante los cambios a la base de datos de la fuente.

La réplica de la base de datos puede también suplir sus planes de desastre-recuperación duplicando los datos de un servidor local de la base de datos a un servidor alejado de la base de datos. Si el servidor primario falla, sus usos pueden cambiar a la copia replegada de los datos y continuar las operaciones (7).

### 1.4.1-Tipos de réplicas

Las réplicas se dividen en dos grupos dentro de los que se encuentran la:

#### ➤ **Replicación Básica**

Con la replicación básica, la replicación de los datos proporciona el acceso a sólo-lectura a los datos de las tablas que provienen de un sitio primario (maestro). Las aplicaciones pueden preguntar datos de las réplicas de datos locales para evitar el acceso a la red, sin importar su disponibilidad. Sin embargo, las aplicaciones a través del sistema deben tener acceso a los datos en el sitio primario cuando las actualizaciones sean necesarias (7).

## ➤ **Replicación Avanzada (Simétrica)**

Las características avanzadas de replicación amplían las capacidades básicas de sólo-lectura de la replicación, permitiendo que las aplicaciones hagan actualizaciones a las réplicas de las tablas, a través de un sistema replicado de la base de datos. Con la replicación avanzada, los datos pueden proveer lectura y acceso a actualizaciones a los datos de las tablas (7).

### **1.4.2 Entornos de Réplica**

Los tipos de entornos de réplica son los siguientes:

#### ➤ **Maestro-Esclavo:**

También conocida como de solo lectura, permite a un solo maestro recibir consultas de lectura/escritura, mientras los esclavos solo pueden aceptar consultas de lectura (8).

#### ➤ **Multi-Maestro:**

También llamada par-a-par o la réplica de camino de n, permite múltiples sitios, actuando como pares iguales. Cada sitio en un ambiente de réplica de multi-maestro es un sitio de maestro, y cada sitio se comunica con otros sitios maestros. Esta capacidad tiene también un severo impacto en el desempeño, debido a la necesidad de sincronizar los cambios entre los servidores. Este tipo de entorno puede ser usado para mantener sitios recuperables ante posibles desastres o caídas, así como para proveer sistemas con alta disponibilidad y para balancear la carga de consultas a través de las distintas ubicaciones (8).

### **1.4.3 Modelo de distribución de datos**

Existen dos modelos de distribución de datos esencialmente aplicados a cada uno de los entornos antes vistos:

#### ➤ **Asincrónica:**

A menudo llamada almacena-y-reenvía, captura cualquier cambio local, los almacena en una cola, y, a intervalos regulares, propaga y aplica estos cambios en sitios remotos. Con esta forma de réplica, hay un período de tiempo antes de que todos los sitios alcancen la convergencia de datos (8).

#### ➤ **Sincrónica:**

También conocida como la réplica en tiempo real, aplica cualquier cambio o ejecuta cualquier procedimiento reproducido en todos los sitios que participan en el ambiente de réplica como parte de una sola transacción. Si el procedimiento falla en cualquier sitio, entonces la transacción entera se anula. La réplica sincrónica asegura la consistencia de datos en todos los sitios en tiempo real (8).

#### 1.4.4 Técnicas de replicación

Existen varias técnicas de replicación de datos esencialmente aplicadas a cada uno de los entornos antes vistos:

➤ **Sincrónica**

Confirmación en dos fases (Two Phase Commit)

Esta tecnología surge a mediados de los años 80. Permite la sincronización de datos distribuidos. Cada transacción solamente es aceptada si todos los sistemas implicados en la réplica están conectados y listos para recibirla, si al menos uno falla, todo el proceso es anulado (8).

➤ **Asincrónica**

Descarga y Recarga (Dump and Reload)

Consiste en hacer un volcado de los datos, copiar la salva para un dispositivo de almacenamiento para luego distribuir la salva por los demás servidores. Esta técnica presenta el inconveniente de que en la mayoría de las ocasiones se consultaban datos que tenían semanas de desactualización, además de que el proceso se realizaba de forma manual (8).

➤ **Instantánea (Snapshot)**

Este tipo de replicación es el mecanismo más simple de todos, donde el publicador<sup>3</sup> replica datos tal como están en la base de datos en un momento dado. La publicación se puede realizar en forma cronogramada o por demanda. El funcionamiento es sencillo: el publicador simplemente envía una réplica de todos los datos hacia los suscriptores<sup>4</sup>, en vez de solamente enviar los datos que fueron alterados desde el último Snapshot realizado (9).

➤ **Transaccional**

Este tipo de replicación realiza un monitoreo de los cambios a los datos que son realizados en el publicador (inserciones, borrados y modificaciones de transacciones). Dichos cambios son propagados a los suscriptores en forma continua, de forma tal que se aproxima a una transacción en tiempo real (9).

---

<sup>3</sup> Es un servidor que es la fuente de datos que deben repetirse. El publicador define un artículo para cada tabla u otro objeto de base de datos para ser utilizada como una fuente de replicación. Uno o más artículos de la misma base de datos se organizan en una publicación.

<sup>4</sup> Es un servidor que recibe los datos replicados por el editor. El suscriptor define una suscripción para una publicación particular. La suscripción se especifica cuando el suscriptor recibe la publicación del Publicador y mapas de artículos a las tablas y otros objetos de base de datos en el suscriptor.

## ➤ **Mezcla**

Este tipo de replicación permite que los diferentes servidores actúen con alto nivel de independencia y desconectados entre sí. En un momento dado se provoca la consolidación de los datos que fueron alterados en cada sitio. Los posibles conflictos, tanto a nivel de tupla como de atributo, que surjan se resuelven automáticamente mediante un algoritmo basado en el uso de prioridades (9).

## **1.4.5 Conflictos de Replicación**

Existen varios tipos de conflictos a tener en cuenta, al tratar la replicación de datos. Los conflictos pueden ocurrir cuando estamos trabajando en un ambiente de replicación que permite actualizaciones concurrentes sobre los mismos datos en múltiples sitios.

### ➤ **Conflicto de actualización**

Un conflicto de actualización ocurre cuando se produce la replicación de una actualización (update) sobre un registro con otra actualización (update) sobre el mismo registro. Este conflicto ocurre cuando dos transacciones originadas desde distintos sitios actualizan el mismo registro, en forma cercana en el tiempo.

Resolución

Prioridad: Cada servidor obtiene una prioridad única, y el servidor de mayor prioridad “gana”, respecto de aquellos con prioridad menor.

Timestamp: La más nueva o la más antigua de las modificaciones es la considerada correcta, y por defecto, sino se eligió ninguno de los criterios “gana” la más nueva.

Particionamiento de datos: Se garantiza que cada registro sea manipulado por un único servidor, lo que simplifica la arquitectura.

### ➤ **Conflicto de unicidad**

El conflicto de unicidad sucede cuando la replicación de un registro intenta violar una restricción de integridad, ya sea por llave primaria o única (Primary Key o Unique). Por ejemplo considere lo que sucede, cuando dos transacciones originadas de dos sitios diferentes, cada una inserta un registro, en su respectiva tabla replicada, con el mismo valor de clave primaria. En ese caso sucede un conflicto de unicidad.

Resolución

Para cada servidor brindar un rango distinto de números para los generadores de clave (secuencias).

Agregar el identificador del servidor a la clave primaria.

Replicar en tablas separadas, y acceder a los datos a través de una vista formada por la unión de ellas. Para resolver el conflicto de potenciales claves duplicadas en la unión se usará una pseudo columna que representa la base de datos fuente.

### ➤ **Conflicto de supresión**

Un conflicto de supresión ocurre cuando dos transacciones originadas de sitios diferentes, una de ellas intenta borrar un registro, y la otra actualizar o borrar el mismo registro, ya que en este caso el registro no existe, tanto para ser actualizado como borrado.

Resolución

Para evitar este tipo de conflictos, una posible solución es que los sitios marquen lógicamente los registros a ser borrados y que periódicamente el sitio maestro corra un proceso que realice el borrado (“delete”) físico de los datos, es decir desde los sitios replicados no se puede ejecutar una sentencia para hacer el borrado de los datos (delete).

### ➤ **Conflicto de orden**

Los conflictos de orden pueden ocurrir en ambientes de replicación con tres o más sitios maestros. Si la propagación al sitio maestro X, está bloqueada por alguna razón, entonces la replicación de modificaciones en datos puede seguir siendo propagada a través de los otros sitios maestros; al finalizar la propagación estas modificaciones debieron ser propagadas al sitio X en un orden diferente a como ocurrieron en los otros sitios maestros, pudiendo producirse un conflicto.

Resolución

Éste tipo de conflicto suele resolverse asignándole distintas prioridades a los sitios maestros, de forma de ordenar las transacciones de acuerdo a ésta.

## **1.4.6 Replicar datos en un entorno de servidor a servidor**

Resulta útil dividir la réplica en dos categorías generales: replicar datos en un entorno de servidor a servidor y replicar datos entre un servidor y los clientes.

### **1.4.6.1 Mejorar la escalabilidad y la disponibilidad**

Muchas aplicaciones se pueden beneficiar de una mayor escalabilidad y disponibilidad. Mantener continuamente actualizadas las copias de datos permite realizar actividades de lectura en muchos servidores. La redundancia que resulta de mantener varias copias de los mismos datos es asimismo útil durante los errores previstos e imprevistos. Si un servidor no se encuentra disponible, las solicitudes se pueden enrutar a un servidor diferente que disponga de los mismos datos hasta que el servidor original (o el que lo sustituya) vuelva a estar conectado.

Existen varias formas de mejorar la escalabilidad y la disponibilidad mediante la réplica, en función de los requisitos de la aplicación:

Si la aplicación tiene una mayor actividad de lectura que de escritura, los datos almacenados en la caché mediante réplica en varios servidores se pueden utilizar para mejorar la escalabilidad de la aplicación.

Si la aplicación requiere la disponibilidad de los datos durante errores previstos e imprevistos de la base de datos.

Si la aplicación requiere tanto una escalabilidad de lectura alta como la posibilidad de controlar errores previstos e imprevistos.

### 1.4.6.1.1 Mejorar la escalabilidad

Las aplicaciones de nivel medio con frecuencia utilizan una sola base de datos para almacenar datos, lo que puede provocar limitaciones de escala cuando aumenta la carga respecto a la base de datos. Cuando las aplicaciones realizan más lecturas que escrituras, como en un catálogo basado en Web, es posible distribuir la parte de lectura de la carga de trabajo almacenando en caché los datos de sólo lectura en varias bases de datos y conectando los clientes de forma equilibrada entre las bases de datos para distribuir la carga.

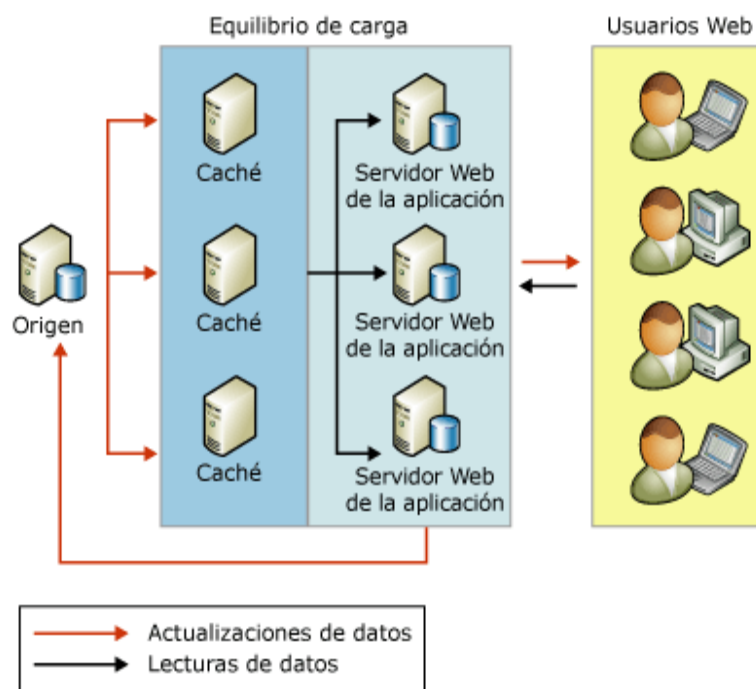


Figura 1.6 Mejora Escalabilidad

### 1.4.6.1.2 Mejorar la disponibilidad

La réplica se puede utilizar para replicar datos en un servidor en espera, que proporciona mayor disponibilidad en caso de inactividad prevista o imprevista del sistema. La réplica debe utilizarse para proporcionar datos en espera semiactiva si los datos requeridos en el servidor en espera son un subconjunto de los datos necesarios en el servidor primario.

Si la aplicación requiere que esté disponible una base de datos completa en un servidor en espera, se utiliza la creación de reflejo de la base de datos en vez de la réplica. La creación de reflejo de la base de datos es más eficaz si se necesita sincronizar toda la base de datos y no es necesario utilizar el servidor secundario para consultas.

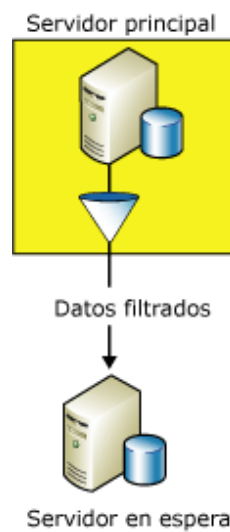


Figura 1.7 Mejora Disponibilidad

### 1.4.6.1.3 Mejorar la disponibilidad y la escalabilidad

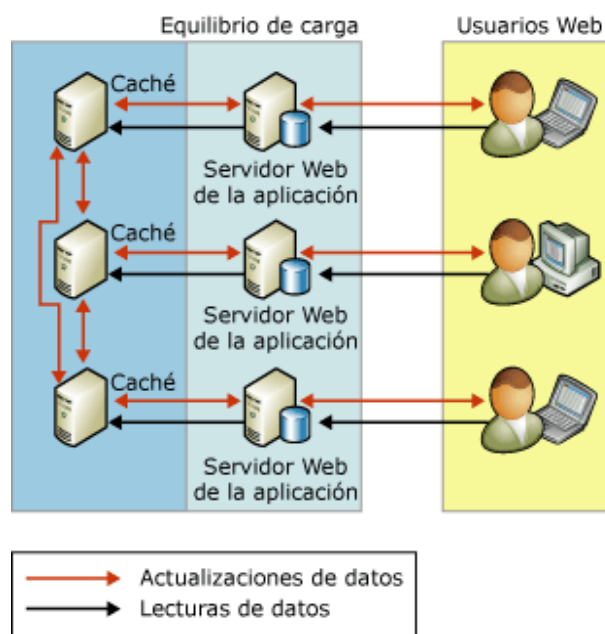
En muchas aplicaciones, es fundamental proporcionar mayor disponibilidad y escalabilidad de lectura. La réplica se puede utilizar como una parte fundamental de una solución para proporcionar ambas. En algunas aplicaciones, el objetivo podría ser mejorar la disponibilidad o la escalabilidad mediante la réplica.

En los siguientes diagramas se ilustran dos aplicaciones que aprovechan el uso de la réplica para aumentar la disponibilidad y la escalabilidad. En los dos casos, las tres bases de datos de los diagramas son pares entre sí: contienen datos y esquemas idénticos. La actividad de escritura para las bases de datos debe particionarse: si la base de datos contiene un catálogo de productos, por ejemplo, puede dirigir las actualizaciones a la primera base de datos para los nombres de productos que



empiezan por la letra A hasta la I, la segunda base de datos para los que empiezan por la letra J hasta la R, y la tercera base de datos para los que empiezan por la letra S hasta la Z. Las actualizaciones se replican posteriormente en las otras bases de datos.

En el primer diagrama se ilustra una configuración donde cada servidor de aplicaciones y Web utiliza datos de un servidor de almacenamiento en caché determinado. Las lecturas y actualizaciones de un usuario determinado pasan a un servidor de aplicaciones específico y, a continuación, a un servidor de almacenamiento en caché específico. Puesto que el servidor de aplicaciones actualiza la caché directamente, no se necesita un servidor de origen central. Las actualizaciones en cada caché se propagan a las demás cachés.



**Figura 1.8 Mejora de la disponibilidad y la escalabilidad**

En el segundo diagrama se muestran tres servidores separados geográficamente con datos que se transfieren entre los tres, lo que permite que cada servidor admita solicitudes de lectura y mejore la disponibilidad.

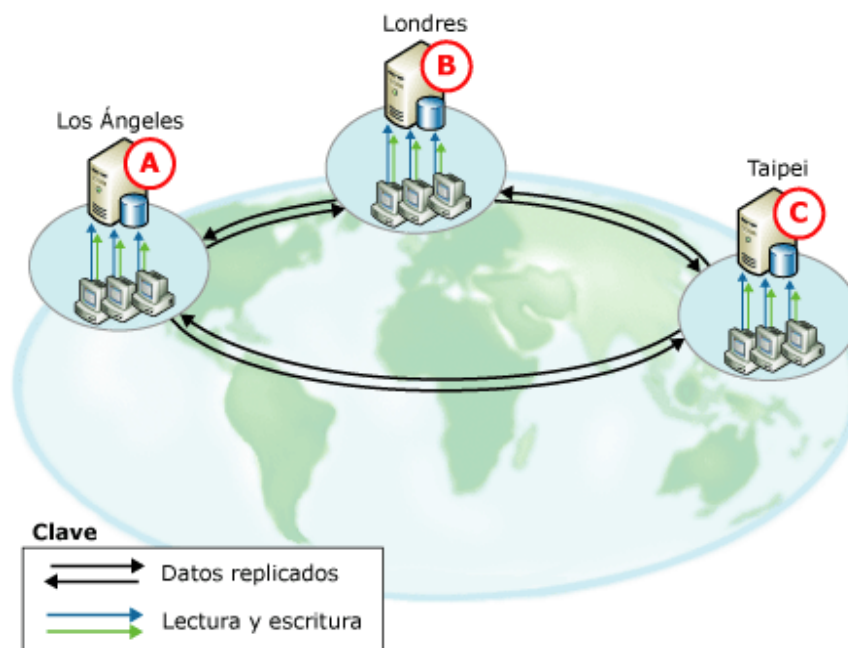


Figura 1.9 Diagrama se muestran tres servidores con datos que se transfieren entre los tres

#### 1.4.7 Replicar datos entre un servidor y los clientes

Es útil dividir la réplica en dos amplias categorías: replicar datos en un servidor para un entorno de servidor y replicar datos entre un servidor y los clientes.

##### Intercambiar datos con usuarios móviles

Muchas aplicaciones requiere que los datos estén disponibles para usuarios remotos, como personal de ventas, repartidores, etc. Estas aplicaciones incluyen las de administración de relaciones con los clientes (CRM), automatización del personal de ventas (SFA) y automatización del personal de campo (FFA).

##### Aplicaciones de punto de venta (POS) para el consumidor

Las aplicaciones POS, como los terminales de caja de salida y los cajeros automáticos, requieren que los datos se repliquen desde sitios remotos a un sitio central.

##### Integrar datos de varios sitios

Las aplicaciones a menudo integran datos de varios sitios. Por ejemplo, una aplicación de apoyo a oficinas regionales puede requerir que los datos fluyan en una o en ambas direcciones entre las oficinas regionales y la oficina central.

### 1.4.8 Intercambiar datos con usuarios móviles

Proporcionar y recopilar datos de usuarios móviles es una parte fundamental de muchas aplicaciones. La mayoría de aplicaciones que admiten usuarios móviles forman parte de una de dos grandes categorías:

#### **Administración de relaciones con clientes (CRM) y Automatización del personal de ventas (SFA)**

Por ejemplo, un vendedor puede utilizar una aplicación SFA para incluir datos de un pedido cuando visita a un cliente. Estos datos se transmiten posteriormente a una ubicación central, como la oficina central de una empresa o un centro de datos.

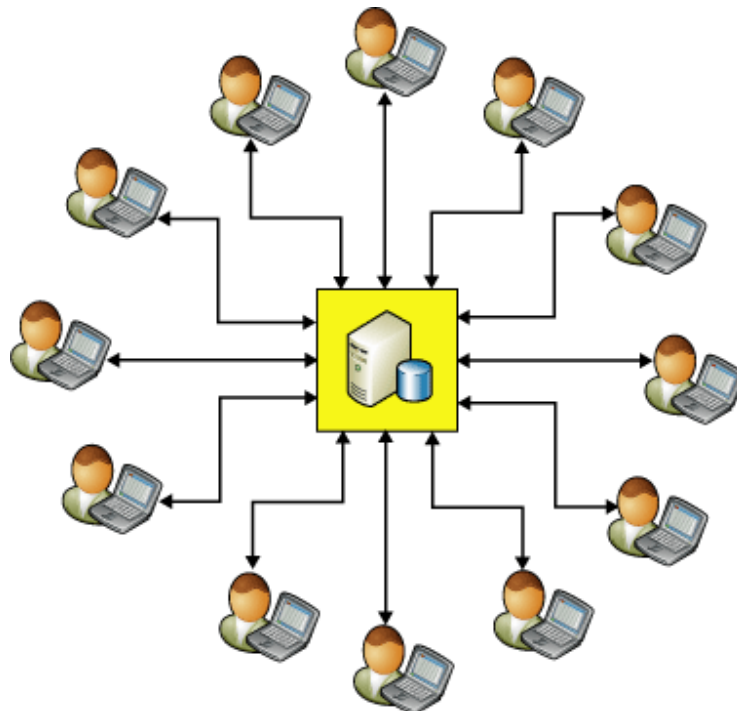
#### **Automatización del personal de campo (FFA)**

Por ejemplo, el personal de campo (repartidores, personal de mantenimiento, inspectores y otros) puede utilizar una aplicación FFA en un dispositivo de mano para recopilar y transmitir datos desde ubicaciones remotas. Un repartidor podría proporcionar datos sobre entregas de paquetes en los sitios de entrega y transmitir de nuevo estos datos a una ubicación central.

Las dos categorías de aplicaciones requieren características de réplica muy similares. La diferencia principal entre las aplicaciones es si los datos son actualizados por varios usuarios.

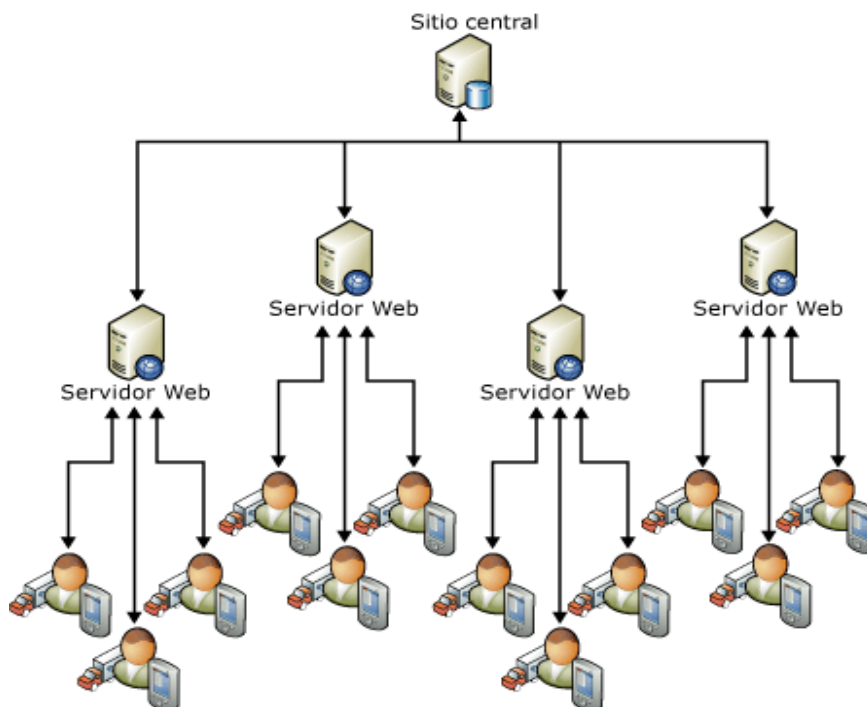
En los siguientes diagramas se ilustran dos métodos diferentes de entregar datos a usuarios móviles: en uno se utilizan equipos portátiles y en el otro se utilizan otros dispositivos. El primer método es el que se utiliza normalmente con aplicaciones SFA y CRM, y el segundo método se utiliza habitualmente con aplicaciones FFA. No obstante, podría utilizarse cualquier método en cada categoría de aplicaciones.

En el primer diagrama se ilustra un escenario en el que un grupo de usuarios con equipos portátiles se conecta directamente a un sitio central:



**Figura 1.10 Escenario de usuarios con equipos portátiles conectados a un sitio central.**

En el segundo diagrama se ilustra un escenario en el que los usuarios con dispositivos se conectan a un sitio central.



**Figura 1.11 Escenario donde usuarios con dispositivos se conectan a un sitio central.**

### 1.4.9 Aplicaciones de punto de venta (POS) para el consumidor

Las aplicaciones de punto de venta (POS) para el consumidor incluyen aquellas que los consumidores encuentran directa o indirectamente en el punto de venta. Algunos ejemplos de estas aplicaciones son los terminales utilizados por cajeros, cajeros automáticos y quioscos ubicados en tiendas. Estas aplicaciones recopilan datos en sitios remotos y los transmiten a una ubicación central, como una oficina central o un centro de datos. Lo habitual en estas aplicaciones es recopilar los datos principalmente en el punto de venta y, después, cargarlos en la oficina central sin que se produzca ningún conflicto, ya que en este caso solamente un usuario remoto (normalmente un cliente o un empleado) actualiza cada dato concreto.

En el siguiente diagrama se ilustra un escenario típico con flujo de datos en dos direcciones entre un sitio central y ubicaciones remotas:

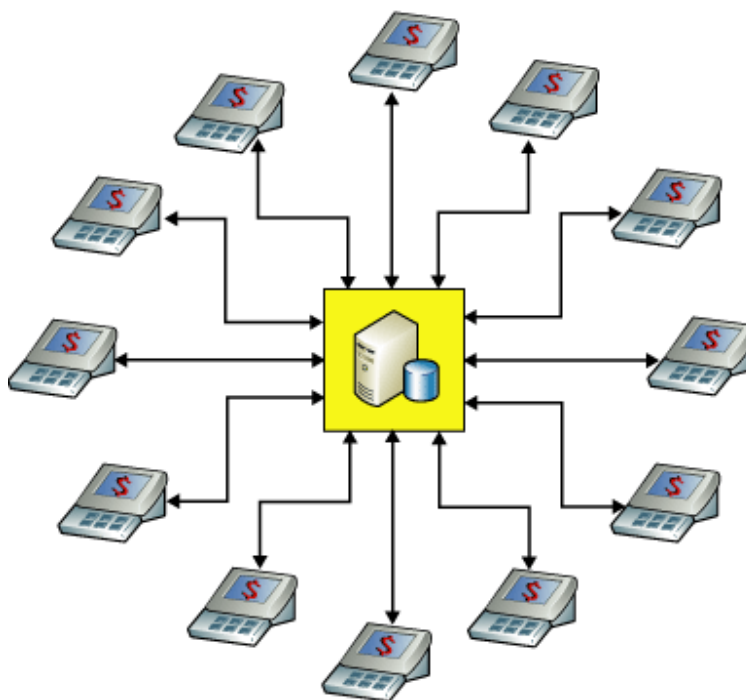


Figura 1.12 Escenario con flujo de datos en dos direcciones entre un sitio central y ubicaciones remotas.

### 1.4.10 Integrar datos de varios sitios (cliente)

Muchas empresas tienen varias oficinas o entidades que recopilan y procesan datos que se deben enviar a una ubicación central. Por ejemplo:

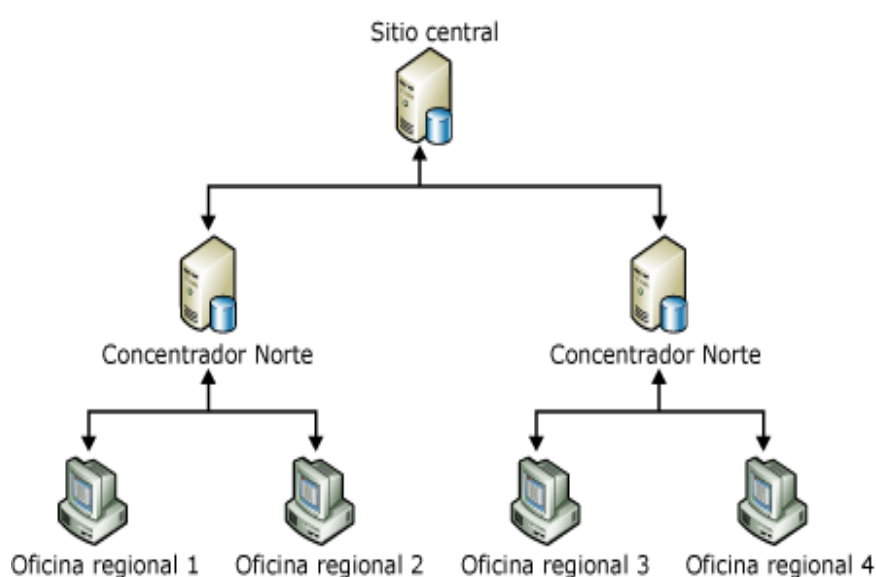
Los datos de inventario se pueden consolidar desde una serie de servidores situados en almacenes locales en un servidor de las oficinas centrales de la empresa.

La información de divisiones empresariales autónomas de una empresa se puede enviar a un servidor central.

Se puede consolidar la información de procesamiento de pedidos procedentes de ubicaciones dispersas.

En algunos casos, los datos también se envían desde el sitio central a sitios remotos. Normalmente, la intención es que estos datos sean de sólo lectura en el sitio remoto, por ejemplo un conjunto de tablas de inventario de productos que sólo se actualizan en un sitio central.

En el siguiente diagrama se ilustra un caso típico con datos que fluyen en dos direcciones entre un sitio central y ubicaciones remotas:



**Figura 1.13. Diagrama con datos que fluyen en dos direcciones entre un sitio central y ubicaciones remotas.**

En este diagrama, los datos fluyen primero a un concentrador antes de pasar a las oficinas a las que dicho concentrador da servicio. También es posible que los datos fluyan directamente entre el sitio central y las oficinas si la organización no tiene un nivel intermedio.

### **1.5 Actualidad de las Réplicas de Bases de datos en Cuba y el mundo.**

En Cuba como en todo el mundo hay un gran adelanto en cuanto a la gran cantidad de información que se debe de procesar en todas las empresas y entidades por lo que el resguardo de toda esta información es importantísimo ya que la pérdida de información sensible es fatal tanto para las medianas como para las grandes empresas, tanto es así que han surgido diferentes tecnologías de desarrollo o software de aplicación que mantiene la integridad de éstos datos .Un ejemplo de ello se ve

evidenciado en la industria porcina en nuestro país, como en todas las ramas de la industria y la economía en el mundo, se produce un aumento constante de la necesidad de la automatización de los procesos, ya que la velocidad de crecimiento de la industria electrónica a nivel mundial ha alcanzado niveles exponenciales, dotando al resto de las industrias de herramientas potentes para elevar la eficiencia y eficacia de sus producciones. En ésta institución se aplica un sistema llamado WinPORC. Se comprobó que el sistema posee dos mecanismos de replicación de datos: total o parcial. En la salva total el sistema realiza un respaldo en soporte magnético de toda la información contenida en las bases de datos y de la configuración general del sistema de los tres módulos. Éste mecanismo posibilita en caso de pérdida total o parcial del sistema, reiniciar el trabajo en el mismo punto donde se encontraba hasta el momento de la salva. El mecanismo de salva parcial solo hace un respaldo de la información contenida en las bases de datos utilizadas por el módulo seleccionado. Esto garantiza la actualización de la información en caso de problema con los ficheros de bases de datos, virus informáticos, pérdida de información, alteración u otras.

También en la Empresa de Telecomunicaciones de Cuba (ETECSA), se utiliza la réplica Snapshot para así tener un mejor resguardo de los datos, también tienen en cuenta otros tipos de réplicas como la réplica transaccional y la réplica Merge (Mezcla).

En la Universidad de las Ciencias Informáticas se llevó a cabo un proyecto para la informatización de los registros civiles y notarías en la ciudad de Caracas y la réplica que llevaron a cabo fue una réplica basada en la captura o detección de cambios, para lograr la sincronización de los datos.

También en la UCI se desarrolló un sistema de replicación basado en el trabajo con la herramienta Slony-I para SGBD PostgreSQL, este sistema se desarrolló para las Fuerzas Armadas Revolucionarias de Cuba (FAR).

En el mundo también existen muchas empresas que realizan réplicas de bases de datos para evitar la pérdida de información como **IPCorp** empresa que proporciona soluciones tecnológicas y servicios informáticos mediante una adecuada utilización de las tecnologías de la información, realiza una réplica sobre el SGBD PostgreSQL y el software Slony-I en un entorno maestro-esclavo.

También **MICROMOUSE (Red Eléctrica de España)** empresa líder de transporte de electricidad, utiliza una solución Double-Take® la cual es una herramienta de replicación de datos de alta calidad y fiabilidad que permite a empresas de cualquier tamaño la disponibilidad y seguridad de sus datos. Combina la replicación de datos en tiempo real con la recuperación automática de un servidor o de un clúster fallido.

### **1.6 Conclusiones**

En este capítulo se han introducido conceptos indispensables para la comprensión del proceso de distribución y resguardo de la información, con el uso de las tecnologías actuales. La mejor solución al problema de actualización de la información es la aplicación de la técnica de réplica de datos. Se hace necesaria la utilización de un software que ayude a configurar esta técnica en los servidores de datos del Ministerio de Auditoría y Control.

Se detalla un estudio de algunas de las últimas tendencias que a nivel mundial se vienen utilizando en el mundo del software. No se podía por tanto dejar de hablar de software libre cuando nuestro país, como muchos otros, ya se ha trazado la meta de implantarlo en numerosas esferas a nivel nacional, por lo que es uno de los aspectos que se ha considerado en el momento de escoger las herramientas a utilizar para referirnos a la solución que se propone.



## **CAPÍTULO 2: ESTUDIO DE RÉPLICAS DE DATOS UTILIZADAS EN DIFERENTES EMPRESAS. PROPUESTA DE RÉPLICA PARA EL MAC**

### **2.1 Introducción**

En el capítulo anterior se dio a conocer información sobre la replicación de datos, sus características, los diferentes tipos de réplicas existentes y la importancia de la misma, así como las características de otras herramientas necesarias para la replicación de datos. En el presente capítulo se realizará un estudio de las diversas réplicas de datos utilizadas en algunas empresas para así dar una propuesta que sea aplicable en el Ministerio de Auditoría y Control y sirva de utilidad en las tareas realizadas por dicho ministerio, por ser éste nuestro principal objetivo de estudio, haciendo énfasis en sus principales características y propiedades.

### **2.2 Réplicas utilizadas en proyecto Servicios Autónomos de Registros y Notarias (SAREN).**

#### **2.2.1 Misión del proyecto.**

La solución informática SAREN cuenta con un entorno distribuido, su misión se basa fundamentalmente en la informatización de los registros y notarías ubicados en la ciudad de Caracas en Venezuela, por lo que se hace necesaria la configuración de un ambiente de replicación entre los servidores de datos. A continuación se presentan las características, tanto del escenario de la solución como del mecanismo implementado.

#### **2.2.2 Réplica utilizada por SAREN**

##### **Escenario**

La solución SAREN se sostiene sobre una arquitectura de Bases de Datos Distribuidas (BDD). Esto se debe en gran medida a las características del sistema y la necesidad de brindar una cobertura de nivel nacional. El escenario se clasifica como homogéneo con diferentes plataformas. Las características se relacionan a continuación:

En un Centro de Datos existe un servidor central sobre el gestor Oracle DataBase Enterprise Edition 10g y la plataforma Red Hat Enterprise Edition Linux Advanced Server (RHEELAS) v 4.0.

Existe un número elevado de servidores de datos locales, sobre el gestor Oracle DataBase 10g Standard Edition One, establecidos en oficinas clientes. Los mismos poseen conexión completa con el servidor de datos central. Estos servidores están sobre la plataforma Windows Small Business Server 2003.

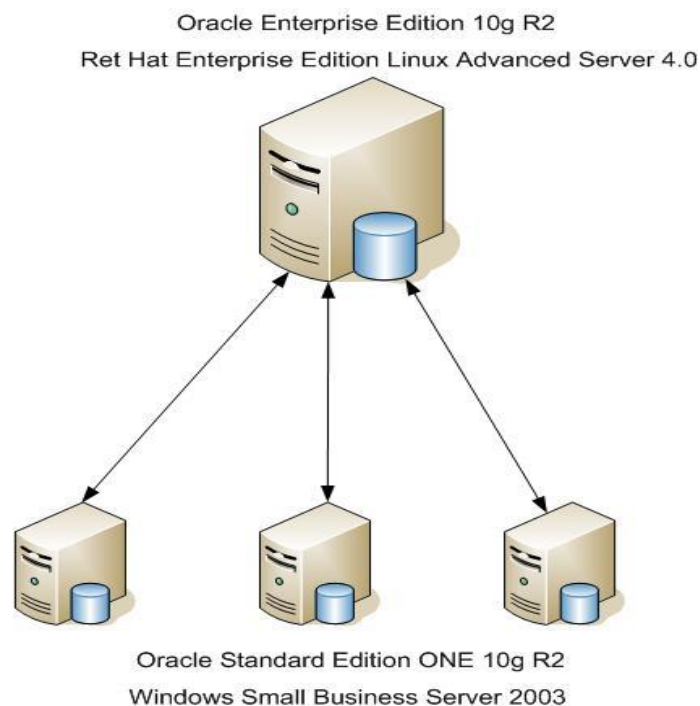
Se intercambia información entre las oficinas clientes y el centro de datos en ambas direcciones (replicación bidireccional).

No se intercambia información entre las oficinas.

Existen dos tipos de oficinas locales: oficinas mercantiles y oficinas públicas. Por este motivo la estructura de las BD locales no será idéntica en todas.

La estructura del servidor central contiene a todas las anteriores, existiendo un particionamiento tanto horizontal como vertical.

A continuación se presenta un esquema que muestra gráficamente las características expuestas.



**Figura 2.1 Esquema de arquitectura distribuida de servidores de datos para la solución SAREN**

### 2.2.2.1 Modelo de replicación

En dependencia de donde se esté realizando la actualización de datos, los sitios se comportan como Fuente (sitio donde se realiza una actualización) o Destino (sitio hacia donde deben replicar los datos actualizados). El proceso de sincronización garantiza la integridad de los datos en ambos sitios. Debido a que el sentido de la réplica es bidireccional entre el Centro de Datos (CD) y cada una de las oficinas, en algunos casos el CD se comportará como Fuente y en otros casos como Destino. Lo mismo pasa para los servidores de las oficinas.

Independientemente del sentido de la réplica se han identificado un conjunto de casos específicos. Debido a la distribución de la información por todas las BD locales, aquí fue necesario lograr un particionamiento horizontal de un conjunto de tablas. Las mismas en el servidor central contienen toda la información a nivel global, y en cada oficina solo la parte de la información que le corresponde. Para el proceso de replicación éste se implementó en forma de reglas, las cuales se validan a través de funciones en la BD. Una tupla de una tabla particionada, antes de replicar hacia un destino determinado, debe cumplir con la regla correspondiente a ese destino.

A continuación se presentan los principales casos de replicación:

### 2.2.2.2 Replicar hacia todas las BD locales la información del CD.

Conjunto de tablas que deben contener la misma información en el CD y en todas las oficinas.

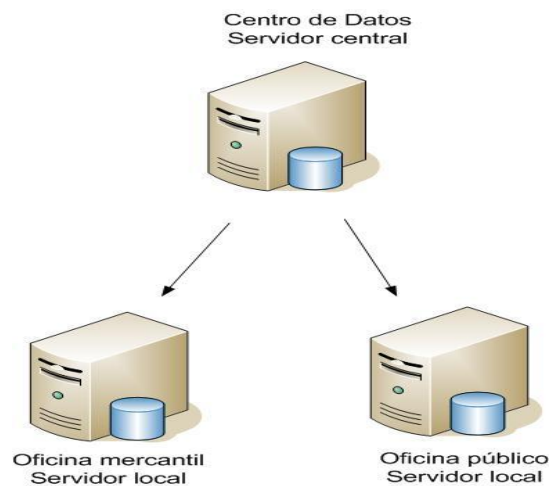


Figura 2.2 Modelo de replicación. Caso 1.

### 2.2.2.3 Replicar hacia las BD locales solo la información correspondiente.

Conjunto de tablas particionadas horizontalmente, cuya información se distribuye por los servidores según un criterio, y existe de forma íntegra en el CD.

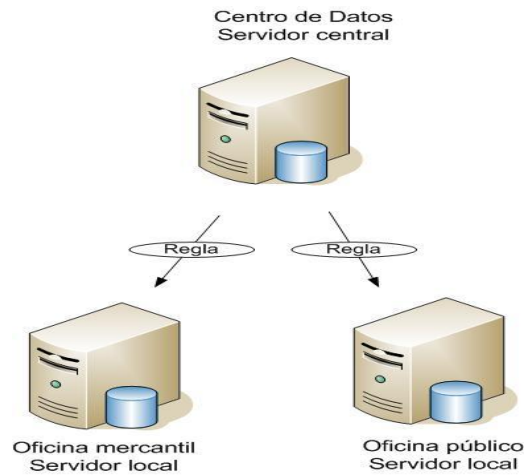


Figura 2.3. Modelo de replicación. Caso 2.

#### 2.2.2.4 Replicar desde una oficina información hacia el CD.

Conjunto de tablas que se actualizan en cada una de las oficinas y cuya información debe almacenarse de forma íntegra en el CD.

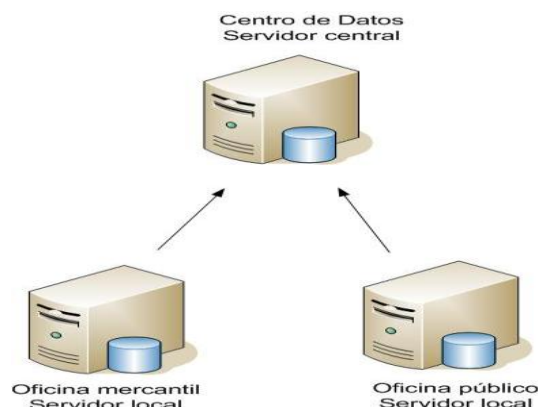


Figura 2. 4. Modelo de replicación. Caso 3.

#### 2.2.2.5 Replicar desde una oficina información hacia el CD, la cual debe actualizarse en todos los servidores locales restantes.

Conjunto de tablas que se actualizan en las oficinas, y cuya información debe estar sincronizada en todas las demás. Debido a la característica del sistema de que las oficinas no se comunican entre sí, es necesario utilizar al CD como puente para sincronizar la información en todos los sitios. La información sube de la oficina de donde se generó y luego baja desde el CD para todas las restantes. En este caso se valida que cuando la actualización se genera en una oficina, este no retorne a dicha oficina.

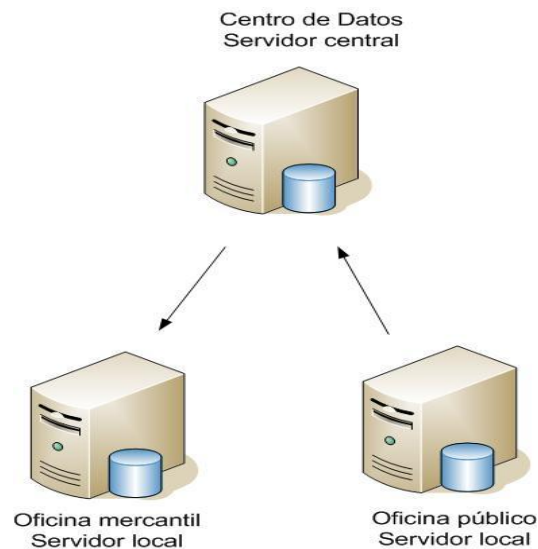


Figura 2.5. Modelo de replicación. Caso 4.

### 2.2.2.6 Replicar desde una oficina información hacia el CD, la cual debe actualizarse en todos los servidores locales restantes. Replicar hacia todas las oficinas información agregada en el CD.

Conjunto de tablas que se actualizan en las oficinas y en el CD. Si la actualización se realiza en la oficina la información debe llegar al CD y luego desde ahí replicar a todos los servidores restantes. Si la actualización se realiza en el CD la información debe replicar a todos los servidores locales. En este caso se valida que cuando la actualización se genera en una oficina, éste no retorne a dicha oficina. También se valida que no se produzcan colisiones producto de actualizaciones simultáneas sobre una misma tupla.

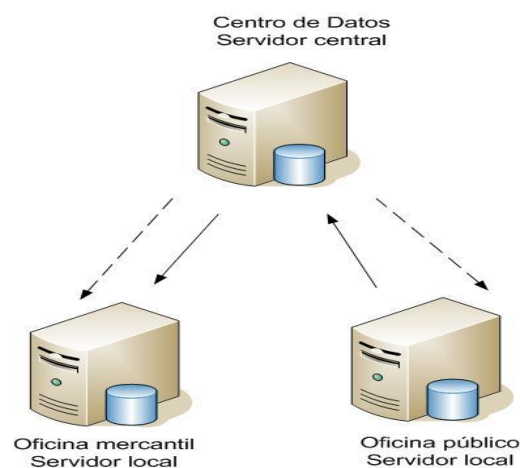


Figura 2.6. Modelo de replicación. Caso 5.

### 2.2.2.7 Replicar en ambos sentidos.

Conjunto de tablas que se actualizan en la oficina y en el CD. Si la actualización se realiza en el CD, la información replica hacia la oficina correspondiente. Si la actualización se realiza en la oficina, la información replica hacia el CD. En éste caso se debe tener en cuenta que pueden producirse colisiones en los datos producto de actualizaciones simultáneas, por lo que es necesario validar el tipo de actualización que se produce en cada sitio para evitar dichas colisiones.

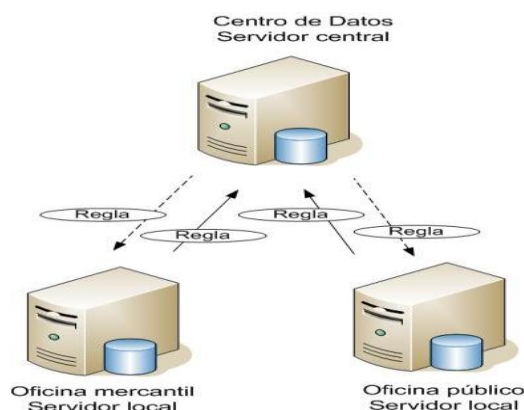


Figura 2.7. Modelo de replicación. Caso 6.

### 2.2.3 Mecanismo implementado

Para soportar los procesos de replicación necesarios se implementó un mecanismo que se encarga de realizar la sincronización de los datos entre los servidores. La cual presenta dos procesos principales: captura o detección de cambios y aplicación de los cambios en el destino para lograr la sincronización, los cuales se describen a continuación.

#### 2.2.3.1 Captura o detección de cambios

Para lograr la sincronización entre dos sitios inicialmente es necesario detectar qué información ha cambiado desde la última vez que se sincronizó. Para esto se definió la siguiente estructura:

Se crea una tabla de control para cada tabla de la BD que interviene en el proceso de replicación. En estas tablas de control se almacena información de cada tupla actualizada:

- id de la tupla
- fecha de la última actualización
- código de la operación(0 para INSERT o UPDATE, 1 para DELETE)
- sitio en que se produce la actualización

Esta información es la mínima indispensable para identificar los cambios realizados sobre las tablas de la BD, de modo que siempre se cuenta con la última actualización. Por ejemplo, si sobre una tabla se realizan 100 actualizaciones sobre la misma tupla, a los efectos de la sincronización solo sería necesario contar con la última operación para aplicarla en el sitio destino.

Para cada tabla de la BD que interviene en el proceso de replicación se asocia un disparador (triggers<sup>5</sup>) que se encarga de registrar en la tabla de control correspondiente la actualización realizada. La ejecución de este disparador se valida en dependencia del usuario que realiza la actualización, para de ésta forma aportar un grado de seguridad al mecanismo.

Se crea en cada sitio una tabla para registrar información sobre la fecha de sincronización del sitio con respecto a los demás. De ésta forma, entre los sitios A y B, todas las actualizaciones registradas en las tablas de control con fecha posterior a la fecha de sincronización de los dos servidores serán tenidas en cuenta, no así las que se hayan producido en una fecha anterior, debido a que éstas últimas ya fueron sincronizadas en algún momento.

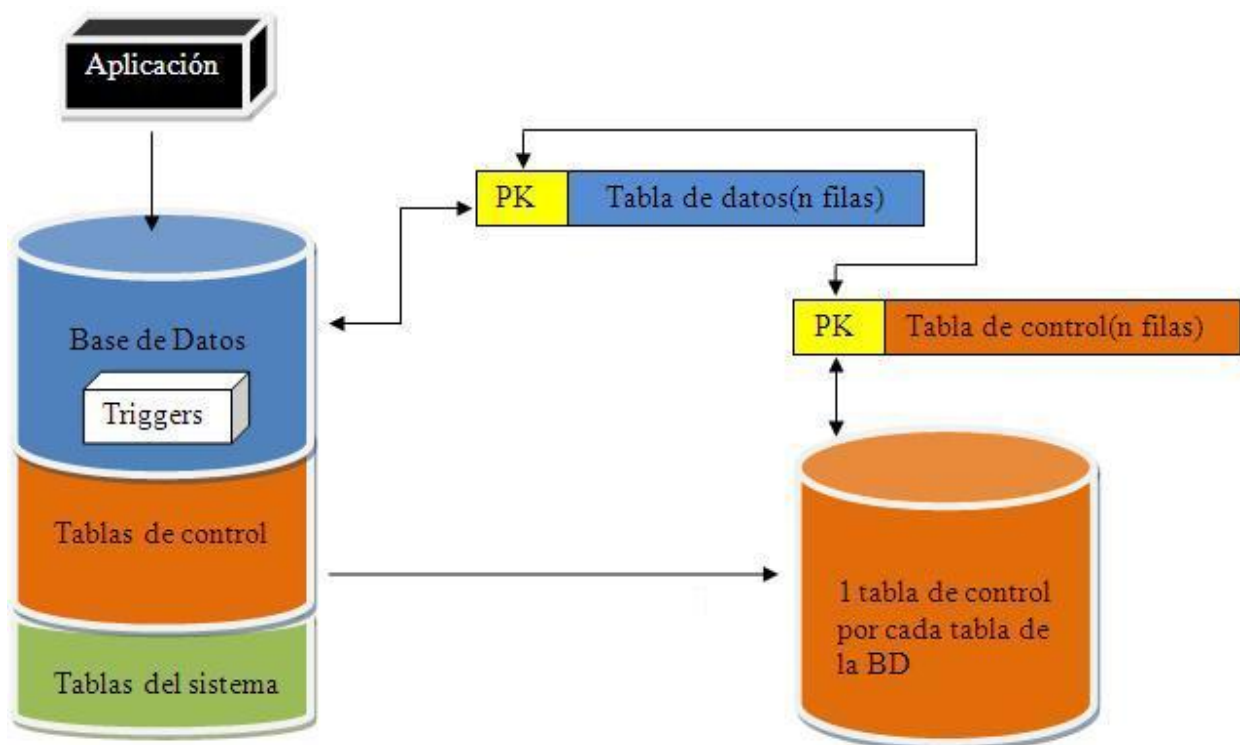


Figura 2.8. Estructura en la BD necesaria para capturar los cambios

<sup>5</sup> Son segmentos de código almacenados en la base de datos, y que se disparan como respuesta a sucesos que tienen lugar en las aplicaciones.

### 2.2.3.2 Aplicación de cambios o sincronización

Una vez que se tienen identificados cuales son los datos que se deben sincronizar, se utiliza una aplicación externa para llevar a cabo este proceso. La aplicación enlaza temporalmente los dos sitios por medio de una conexión peer-to-peer<sup>6</sup>, capturando en el sitio Fuente los datos actualizados y aplicando los cambios necesarios en el sitio Destino a través de llamadas a procedimientos almacenados. Una vez que culmina éste proceso de sincronización se cierra la sesión de replicación liberando de ésta forma los recursos y minimizando el impacto sobre el servidor. Debido al sentido bidireccional del proceso, la aplicación intercambia los sitios (Fuente-Destino, Destino-Fuente), posibilitando de ésta forma que en una sola corrida se sincronicen los servidores en ambos sentidos.

### 2.2.3.3 Alcance del mecanismo

Es importante resaltar que éste mecanismo de replicación es independiente del gestor sobre el que estén funcionando las BD. Debido a la simplicidad de sus procesos, para adaptarlo a algún gestor determinado solo sería necesario modificar la forma en que se crean los objetos de la BD necesarios (tablas de control, disparadores y procedimientos almacenados) y modificar la cadena de conexión al servidor. De esta forma se evidencia que está soportada la replicación entre servidores con cualquier gestor, e incluso entre servidores con gestores distintos.

## 2.3 Réplica utilizada por la Empresa de las Telecomunicaciones de Cuba (ETECSA).

### 2.3.1 Misión de la Empresa

La Empresa de Telecomunicaciones de Cuba (ETECSA), es una organización cubana de capital mixto y tiene como objeto social la prestación de los servicios públicos de telecomunicaciones, mediante la proyección, operación, instalación, explotación, comercialización y mantenimiento de redes públicas de telecomunicaciones en todo el territorio de la República de Cuba.

Esta empresa tiene una alta responsabilidad en el desarrollo socio-económico del país y en especial, en la informatización de la sociedad, garantizando una efectiva conectividad.

Como organización que presta servicios, centra su actividad en la calidad de los Recursos Humanos sobre la base de la gestión integral de éstos y se organiza en una estructura conformada por Unidades de Negocios que facilita la orientación de sus colaboradores a la satisfacción de las necesidades del mercado, dando soluciones creativas a los problemas mediante la utilización de una tecnología de

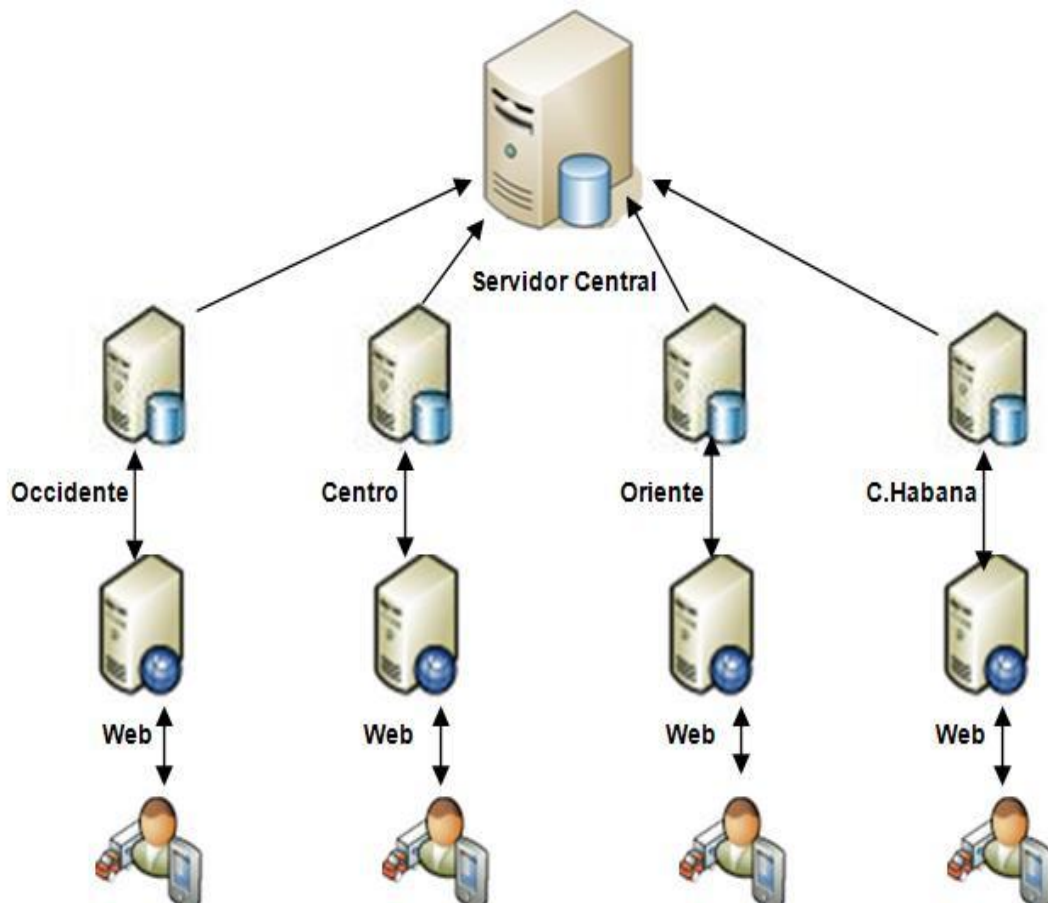
---

<sup>6</sup> Tipo de conexión en la que la comunicación se establece entre dos estaciones sin intermediarios.



avanzada y el compromiso con la calidad total, acorde a los valores y principios de nuestra sociedad, que se ponen de manifiesto en su visión y misión empresarial.

### 2.3.2 Réplica Usada por ETECSA



**Figura 2.9 Esquema de Réplica utilizada por ETECSA**

El tipo de réplica utilizada en ETECSA es Snapshot utilizando el gestor de Bases de Datos Oracle

### 2.3.3 Snapshot es útil cuando:

- Los datos son principalmente estáticos y no cambian a menudo. Cuando cambian, tiene más sentido publicar una nueva copia a los suscriptores.
- Es aceptable tener copias de datos que están fuera de fecha para un período de tiempo.
- Replicando pequeños volúmenes de datos en la que todo un refrescamiento de datos es razonable.

- Para decidir si la replicación Snapshot es apropiada, se debe considerar el tamaño de todo el conjunto de datos y la frecuencia de cambio en los datos.

### 2.3.3.1 ¿Cómo trabaja la Replicación Snapshot?

La replicación Snapshot es ejecutada por el agente de Snapshot y el agente de distribución. El agente de Snapshot prepara los archivos de Snapshot contenedores de esquemas y la información de las tablas publicadas y los objetos de la base de datos, y almacena los archivos en las carpetas de Snapshot, y los archivos de trabajos de sincronización en la base de datos de distribución en el distribuidor<sup>7</sup>. Por defecto la carpeta de Snapshot se localiza en el distribuidor, pero se puede especificar una localización alterna en lugar o en adición a la que está por defecto. EL agente de distribución mueve los Snapshot contenidos en las tablas de la base de datos de distribución hasta el punto de destino en las tablas de los suscriptores. La base de datos de distribución es usada solo por replicación y no contiene ninguna tabla de usuarios.

### 2.3.4 Servicios que brinda ETECSA en los que se utiliza la replicación:

ETECSA lleva a cabo varios servicios en los cuales utiliza la replicación, como son:

**Enterprise Resource Planning (ERP):** O sea, Planificación de Recursos de la Empresa. En ETECSA, éste sistema es SAP (Sistemas, Aplicaciones y Productos.), líder mundial en la materia. Estos sistemas se dedican a gestionar todos los recursos de una empresa, entre los que se enumeran la Contabilidad (General, Analítica), Gestión de Materiales (Logística), Recursos Humanos, Inversiones, Ventas, etc.

**Provisión:** Se refiere al sistema de provisión de servicios de ETECSA. En otras palabras, éste es el sistema que realmente contiene la información acerca de cada línea telefónica o de datos que la empresa activa para dar servicio, con todos los detalles relacionados (cable, par, gabinete, central, ubicación geográfica, etc.)

**Facturación:** es el más sencillo de entender, pues solo implica que dado un tráfico, según sea el segmento de mercado al que pertenece el cliente y así, las tarifas que se le aplican, se calcula cuanto debe pagar por usar el servicio y se le hace una factura, que se imprime y llega a los lugares establecidos, que en el caso de los clientes residenciales, es la casa.

---

<sup>7</sup> Es un servidor que realiza varias tareas cuando los artículos se desplazan desde Publicador a Suscriptores. Las tareas reales realizadas dependen del tipo de replicación realizada.

### 2.3.5 Características de los servidores que utiliza ETECSA para cada uno de los servicios que brinda:

Para llevar a cabo la réplica, ETECSA utiliza varios servidores de los cuales describimos sus características a continuación:

**Tabla 1. Características de los servidores que utiliza ETECSA**

Servicio	Servidor	Sistema operativo	Procesadores	Capacidad	RAM	SGBD
Provisión	BL-45	Windows 2000	8 procesadores de 1.8GHz ( en 4 pares dual-core)	10 Tbytes, propio de 40Gbytes	16 Gbytes	SQL Server 2000
	Servidor web(BL-20)		4 procesadores de 1.8GHz ( 2 pares dual-core)	120 Gbytes	4Gbytes	
ERP	HP-Compaq	Unix True64	8 procesadores de 2.7GHz	10Tbytes, propio de 80Ghz	16 Gbytes	Oracle 9i RDBMS
Facturación	HP-Compaq	Unix True64	4 procesadores de 2.7 GHz	propio de 1 Tbyte	16 Gbytes	Oracle 9i RDBMS

Las redes de acceso son redes IP. El servicio de provisión se ofrece por una red, con ancho de banda de 256 Kbps.

Los servicios de facturación y ERP se ofrecen por una red, con ancho de banda de 2Mbps (ver anexo 1).

## 2.4 Réplica usada en el Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR)

### 2.4.1 Información de la Organización.

Las Fuerzas Armadas Revolucionarias (FAR) es una institución que desempeña un papel determinante en la defensa del país, siendo el Ministerio (MINFAR), su entidad rectora.

Entre sus principales objetivos se encuentra la total informatización del organismo, el cual con el desarrollo de las (TIC) se ha dado a la tarea de informatizar todas sus entidades para garantizar la integridad y seguridad de toda la información que presenta que es de vital importancia para la organización y para el país.

### 2.4.2 Réplica del MINFAR

El Ministerio de las Fuerzas Armadas Revolucionarias (MINFAR) es una de las instituciones preocupada y ocupada en el desarrollo de soluciones informáticas aplicadas a su entorno. Dada la necesidad de compartimentación y vitalidad de la información, la totalidad de las aplicaciones informáticas son sistemas distribuidos a lo largo del país (sucursales), interconectados por una red con estructura de árbol, siendo el ministerio el nodo primario de la jerarquía, como se muestra en la Figura 2.10.

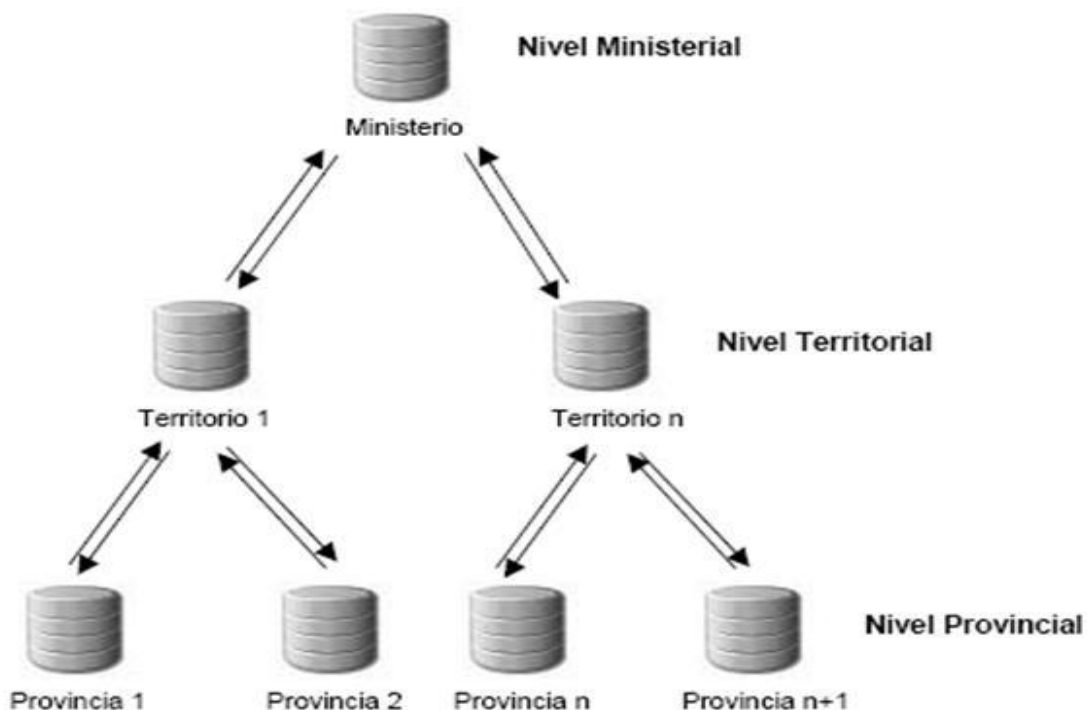


Figura 2.10. Esquema de réplica utilizada por las Fuerzas Armadas Revolucionarias (FAR).

Cada sucursal maneja sus propias bases de datos diseñadas en PostgreSQL (sistema gestor de base de datos), recibiendo y transmitiendo información, hacia uno y otro nivel de la jerarquía respectivamente. Se tienen tres niveles principales en los cuales se concentra la información de las sucursales subyacentes, las que variarán en cantidad y tipo, según las necesidades del ministerio. Un primer nivel, denominado Nivel Ministerial, un segundo nivel, denominado Nivel Territorial y por último, un tercer nivel, denominado Nivel Provincial. Atendiendo a la situación anterior se debe garantizar que la información de cada sucursal se transmita al nivel superior cumpliendo con una serie de normas y reglas que garanticen la compartimentación de la misma, así como el cumplimiento de los plazos requeridos.

### 2.4.1.1 Esquema de un nodo a la hora de Replicar Dato

Configuración de un esquema de réplica para replicar bases de datos mediante Slony I (Figura 2.11).

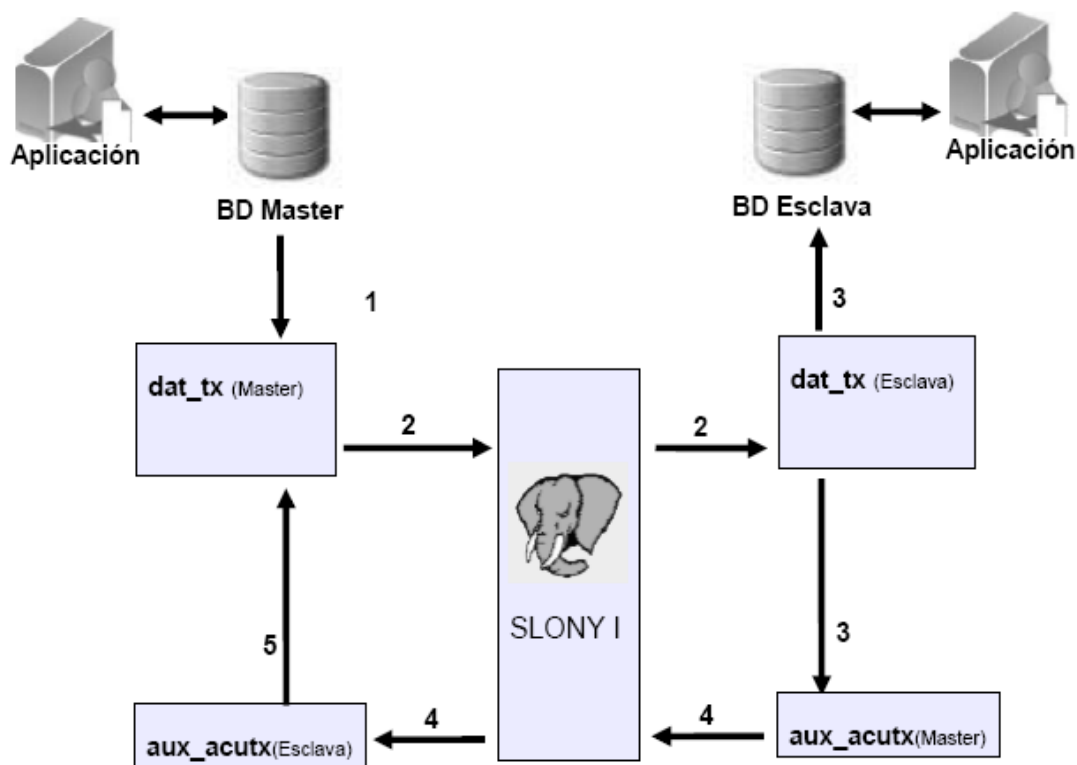


Figura 2.11. Réplica de la FAR entre dos servidores usando Slony-I.

A continuación se detallan la responsabilidad de cada flujo en la propuesta.

**Flujo 1**

Será un trigger escrito en pl/pgSQL que se dispare siempre luego de uno de los eventos de inserción, borrado o actualización (INSERT, DELETE o UPDATE), reconstruirá la sentencia ejecutada y la insertará en la entidad dat\_tx (master).

Según las reglas definidas decidirá si la sentencia SQL es o no propagada al nodo esclavo.

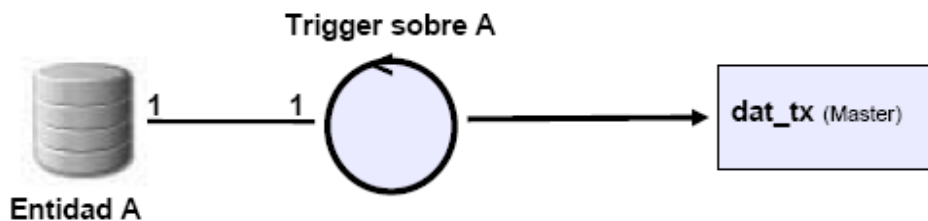


Figura 2.12 Descripción del Flujo 1

**Flujo 2**

Slony con su capacidad para crear espejos replicará cada tupla que pongamos en dat\_tx (master) en dat\_tx (esclava o espejo). Aún cuando se haya perdido la conexión, éste en cuanto se restablezca realizará la actualización de las mismas.

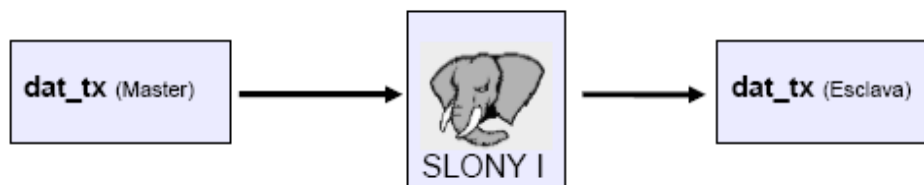


Figura 2.13 Descripción del Flujo 2

**Flujo 3**

Será un trigger que se dispare siempre luego de una inserción (after INSERT) sobre la entidad dat\_tx (esclava). Su cuerpo estará implementado en pl/pgSQL, posibilitando así la conexión a la “BD esclava” y la ejecución de las instrucciones SQL recibidas.

De esta última haberse ejecutado satisfactoriamente será insertado su identificador (llave) en la entidad dat\_acutx (master).

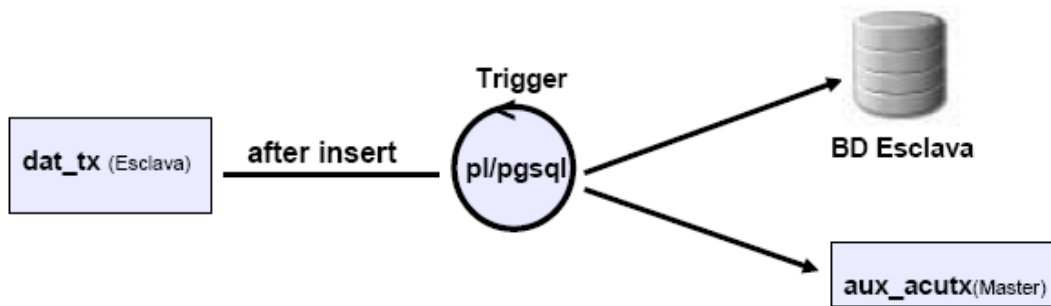


Figura 2.14 Descripción del Flujo 3

**Flujo 4**

Slony replicará cada tupla que pongamos en dat\_acutx (master) en dat\_acutx (esclava o espejo). Esta es la propuesta para mantener la entidad dat\_tx (master) vacía, evitando así la lentitud por sobrecarga del sistema.

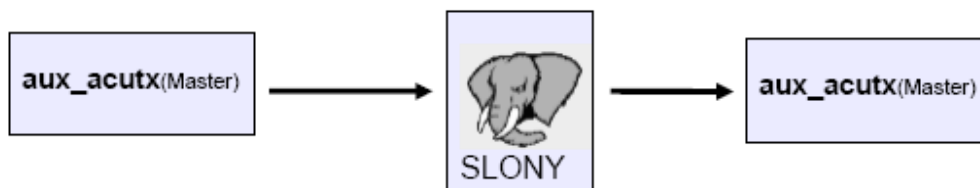


Figura 2.15 Descripción del Flujo 4

**Flujo 5**

Será un trigger (pl/pgSQL) que se dispare siempre luego de una inserción (after INSERT) sobre la entidad aux\_acutx (esclava), éste tendrá la responsabilidad de ejecutar el borrado de todas las tuplas que estén en dat\_tx (master) en la medida en que lleguen a él sus identificadores (así mismo se vacía dat\_tx (esclava)).

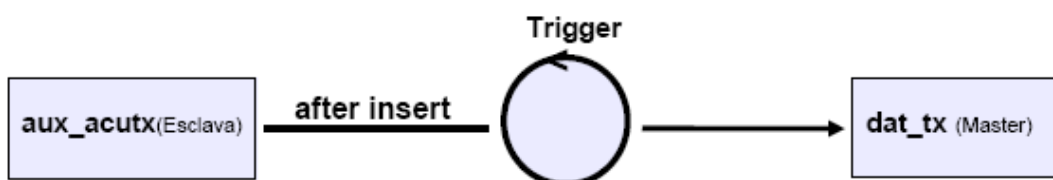


Figura 2.16 Descripción del Flujo 5

## 2.5 Réplica de INFOMED

### 2.5.1 Misión de Infomed (Red de salud de Cuba)

Infomed es la red de personas e instituciones que trabajan y colaboran para facilitar el acceso a la información y el conocimiento para mejorar la salud de los cubanos y de los pueblos del mundo, mediante el uso intensivo y creativo de las tecnologías de la información y la comunicación.

El Portal de Salud, Infomed, comenzó a desarrollarse a partir de 1994 con el objetivo de facilitar el acceso a la información relacionada con las ciencias de la salud y especialmente dar acceso a la información de salud producida en Cuba y cuenta para ello con la Biblioteca Virtual de Salud de Cuba y la Universidad Virtual de la Salud.

### 2.5.2 Estructura de red Infomed

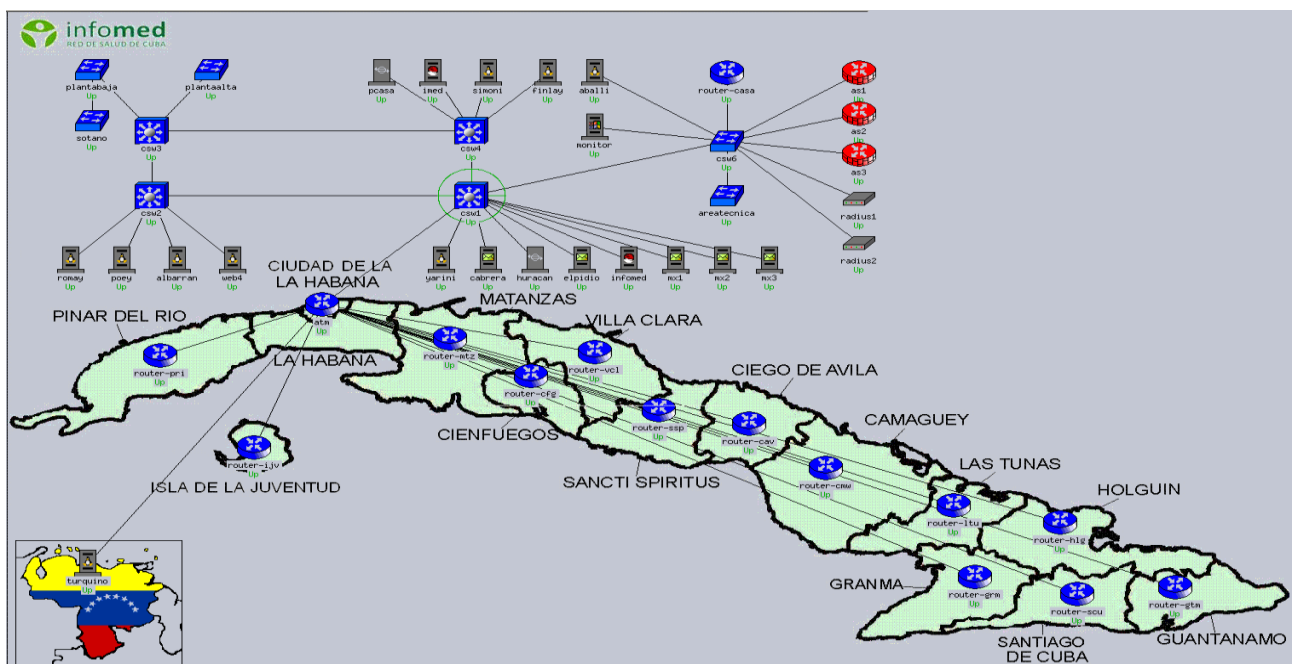


Figura 2.17 Red Infomed

### 2.5.3 Réplica de datos de Infomed

El instituto de Infomed utiliza el SGBD MySQL, con un servidor en cada provincia del país que se comunican con el servidor central ubicado en Ciudad de la Habana. Los servidores de provincia no utilizan ningún sistema de réplica pero sí hacen salvadas de sus bases de datos cada un tiempo determinado y la guardan para evitar la pérdida total de los datos en dichas provincias, mientras el



servidor central ubicado en Ciudad de la Habana y un servidor ubicado en Venezuela llamado Turquino que realiza una réplica asincrónica hacia un servidor esclavo que también se encuentra en Ciudad de la Habana, mediante un entorno de réplica maestro-esclavo, donde usan un servidor esclavo y dos servidores maestros (como se muestra en la figura 2.18), la réplica entre el servidor de Venezuela y el servidor esclavo de Ciudad de la Habana se realiza cada 20 minutos y entre el servidor de Ciudad de la Habana maestro y el esclavo se realiza en un horario determinado del día (no siempre es el mismo horario)(ver anexo 1).

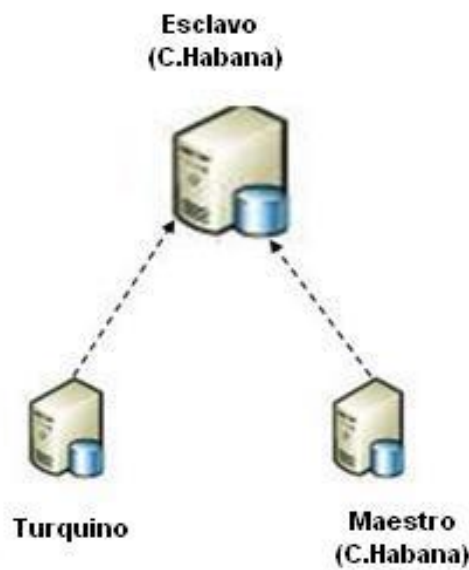


Figura 2.18 Esquema de Réplica de Infomed

#### 2.5.4 Propiedades de los servidores y de la red

Tabla 2. Características de los servidores que utiliza Infomed

Servidores	Capacidad	RAM	Procesador
Maestro(DL-560,G-3)	2disco,36GB	4GB-RAM	3Proc-2,8GHz
Esclavo(DL-360,G-3)	6disco,36GB	3GB-RAM	2Proc-3GHz

Tabla 3. Velocidad de la Red utilizada por Infomed.

Velocidad de la Red	Provincias
256Khz	Resto Provincias
128KHZ	Isla Juventud
4HZ	Ciudad Habana

## 2.6 Propuesta de Réplica para el Ministerio de Auditoría y Control (MAC)

Tabla 4. Comparación de algunas empresas en cuanto a las características que poseen para replicar datos.

Organización	SGBD	Entorno Réplica	Técnica Replicación	Tipo Réplica
INFOMED	MySQL	Maestro-Esclavo	Asincrónica	Básica
SAREN(RN)	Oracle	Maestro-Esclavo	Asincrónica	Básica
ETECSA	Oracle	Maestro-Esclavo	Snapshot	Básica
MINFAR	PostgreSQL	Maestro-Esclavo	Asincrónica	Básica

Dada la tabla comparativa (ver Tabla 4) entre todas las propiedades de las organizaciones que se estudiaron anteriormente, se puede ver que en casi todas coinciden el entorno de réplica, la técnica de replicación y el tipo de réplica por lo que se puede concluir que es una de las propiedades más usadas para la replicación de datos. Lo que varía es el tipo de SGBD ya que cada una utiliza el que más se adapte a sus intereses y posibilidades. Por lo que para la propuesta del MAC se tuvo en cuenta cada una de estas propiedades ya que se ajustan a las necesidades del ministerio, variando en el tipo de SGBD que es el PostgreSQL, pues es el gestor de bases de datos utilizado por el MAC. Después de todo este estudio realizado sobre estas empresas y el estudio de gran cantidad de fuentes bibliográficas acerca de las últimas tendencias y tecnologías que a nivel nacional e internacional se están usando en el mundo de la informática y en especial en la rama de la replicación de base de datos se escogió como técnica de replicación la asincrónica ya que la velocidad de red entre el ministerio y sus sucursales provinciales es de 56KBps por lo que no es factible usar una técnica de replicación sincrónica(tiempo real) para evitar congestionar la red con la transferencia constante de

datos entre el ministerio y sus sucursales provinciales, como entorno de réplica se escogió un entorno maestro-esclavo ya que los datos siempre serán replicados de las sucursales provinciales hacia el ministerio, nunca del ministerio hacia las sucursales provinciales por lo que no es necesario un entorno de réplica Multi-maestro.

Se escogió como software de replicación el Slony-I, ya que es un software diseñado para replicar específicamente sobre el SGBD PostgreSQL, Slony-I realiza una réplica asincrónica en un entorno maestro-esclavo cumpliendo con las características de replicación que se necesita en el MAC, además de ser un software popular en dicho entorno de réplica, fácil de usar e instalar y de tener una referencia del uso de este software en el MINFAR, debido a que éste se está utilizando en dicha organización cerca ya de un año sin haber presentado problemas en el tema de la replicación de datos, además de ser un software que presenta una muy buena tolerancia a fallos, ya que en casos de fallas en la red o de fluido eléctrico no ocasiona pérdidas de datos en el momento en que se está realizando la replicación, pues se puede seguir trabajando sobre el servidor maestro, así cuando se restablecen estos problemas, el se encarga de replicar los datos introducidos después de haber ocurrido dichos problemas hacia el servidor esclavo. El Slony-I no necesita conexiones especiales o protocolos para comunicar ambos servidores, sino que simplemente necesita tener acceso a las bases de datos PostgreSQL, como la conexión de un "superusuario" que tiene la capacidad de actualizar las tablas del sistema, ya que el PostgreSQL es el encargado de la seguridad de los datos en la replicación usando el protocolo Secure Socket Layer (SSL), básicamente SSL crea un "túnel" por donde los datos viajan de forma segura, SSL es capaz de enviar de forma cifrada la información requerida por un servidor, el cual una vez que los recibe los descifra para poderlos procesar. Esta operación también se lleva a cabo de forma inversa. Es un protocolo que trabaja entre la capa de aplicación y la capa de transporte (el protocolo TCP). La meta de SSL es proporcionar privacidad, integridad y autenticidad en la información que se maneja en las transacciones.

La estructura de replicación definida en nuestra propuesta consta de 3 niveles, (como se muestra en la figura 2.19), un primer nivel provincial donde se encontrarán todas las sucursales provinciales del MAC, debido a la necesidad que tiene el ministerio de utilizar la información de las sucursales encontradas en las distintas provincias del país y la velocidad de red entre las provincias y el MAC que es de 56KBps se decidió no conectar las provincias a una aplicación ubicada en el nodo central del MAC y que cada una se conecte en sus provincias a sus propias aplicaciones para evitar de esta forma que en caso de estar muchos usuarios conectados al mismo tiempo al servidor central y que la red no

sea capaz de soportar este tráfico de datos no ocurran problemas de conexión de red y no haya problemas de disponibilidad en los datos. Un segundo nivel regional que estaría constituido por tres servidores, Occidental, Central y Oriental y que nos garantizarían un esquema de réplica mas fuerte y prácticamente escaso o nulo a la pérdida de información, ya que dichos servidores garantizan que siempre halla un servidor disponible hacia donde replicar los datos en casos de pérdida de algún servidor y nos proporcionarían un tiempo más amplio de reponer el servidor que está en falla sin temor a que pueda ocurrir pérdida de información, ya que siempre se tendrá un resguardo de datos, además que serían los encargado del resguardo de información que se procese en cada sucursal provincial que le corresponda según su zona geográfica y también serían los encargados de replicar sus datos con un servidor central que se encontraría en un tercer nivel el cual es el encargado del resguardo de toda la información del Ministerio de Auditoria y Control(MAC),

De esta forma se lograría una gran seguridad y protección a la información procesada en el MAC en caso de fallas o problemas en el sistema, ya que cada sucursal replicaría los datos con el servidor regional que le corresponda y éste con el servidor central, en caso de alguna falla en los servidores regionales las sucursales seguirían trabajando con los datos del servidor central y en caso de falla en el servidor central pues la sucursales seguirían trabajando como lo venían haciendo con el servidor regional, logrando así que la pérdida de información sea escasa o nula.

A raíz del estudio realizado en las distintas empresas de nuestro país en cuanto a los requisitos de hardware que presentan cada una, y analizando que Infomed, siendo la red mas grande que tiene Cuba, la cual utiliza para la réplica servidores de 3 o 4 gigabytes de RAM y 72 gigabytes de capacidad en el servidor maestro y en el servidor esclavo 220 gigabytes, se considera que el MAC soporte para su replicación en sus sucursales provinciales como requisitos mínimos de hardware 1 gigabytes RAM, 2.0 GHZ de velocidad de procesadores y una capacidad de almacenamiento de 80 gigabytes, para los servidores regionales como requisitos mínimos de hardware 2 gigabytes RAM, 3.0 GHZ de velocidad de procesadores y una capacidad de almacenamiento de 160 gigabytes y para el nodo central como requisitos mínimos de hardware 2 gigabytes RAM, 3.0 GHZ de velocidad de procesadores y una capacidad de almacenamiento de 250 gigabytes ya que los datos que se replican en este ministerio son menores que los que se llevan hasta el momento replicando en Infomed y además que en el MINFAR con sus bases de datos que son bastante grandes se realizaron pruebas en servidores de 2 gigabytes de RAM y soportaron mas de 6 millones de registros sin problema, además de que este proceso de replicación en el MAC será realizado en un momento determinado del día donde no exista mucho tráfico en la red lo que facilitaría la replicación.

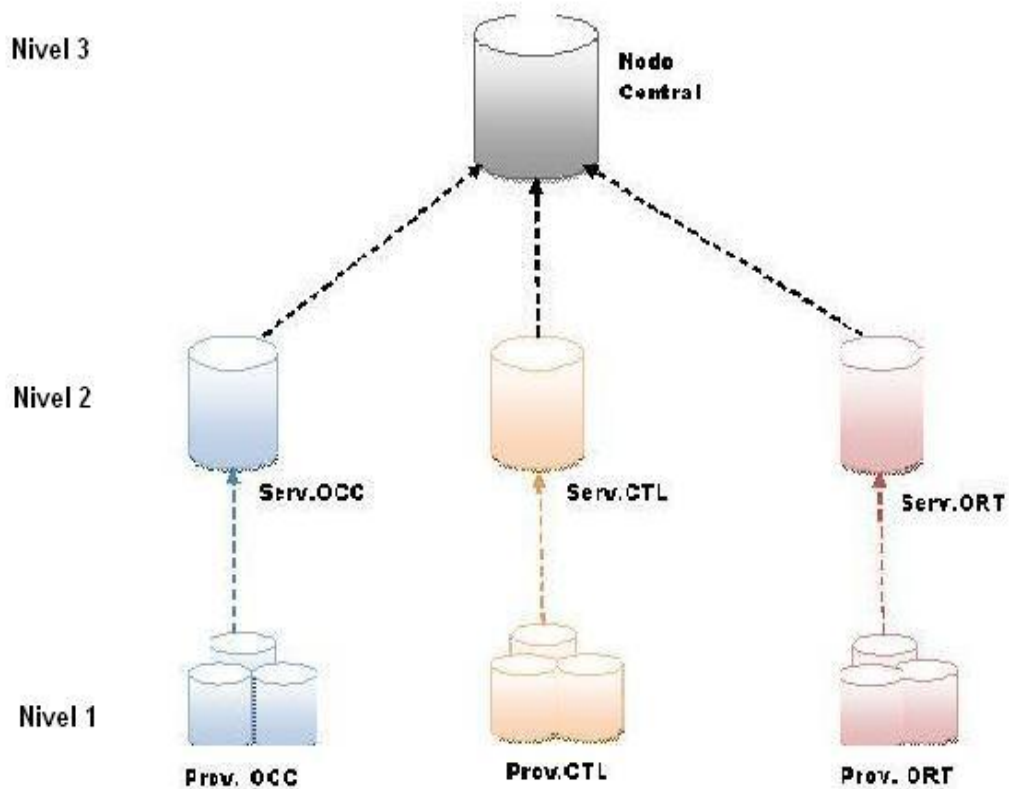


Figura 2.19 Esquema de la propuesta de réplica para el MAC

### 2.6.1 Características de cada uno de los dispositivos con los que cuenta el MAC

En el Ministerio de Auditoría y Control, en la cual se tiene previsto desarrollar una gran aplicación financiera (por los módulos de que se componen será una aplicación de buen tamaño), según la distribución de la empresa y sus sucursales se tendrá el siguiente entorno:

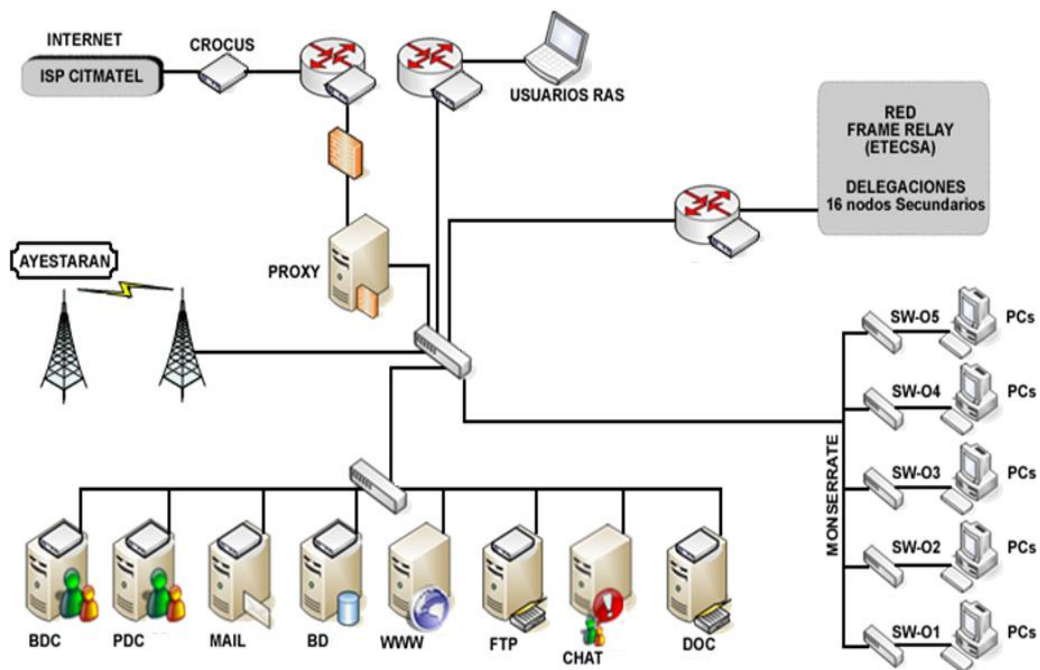


Figura 2.20 Estructura de Red del MAC.

### 2.6.1.1 Características de los servidores (Nodo Central)

Tabla 4. Características de los servidores Nodo Central que utiliza MAC  
Cantidad de Equipos: 10

Dispositivo:	Características	Observaciones
CPU	3.40 GHz Pentium D dual core	
Memoria RAM	1.00 Gbytes o 2.00 Gbytes	Según empleo y configuración
Capacidad de almacenamiento	Entre 160 Gbytes y 750 Gbytes	Según empleo y configuración
Network	Velocidad: 2 puertos a Giga bit	

Sistema Operativos	Windows 2003 y Linux	Según empleo
--------------------	----------------------	--------------

2.6.1.2 Características de las Estaciones de Trabajo (Organismo Central)

Tabla 5. Características de los equipos que utiliza el MAC en las Estaciones de Trabajo.

Equipos:	Características	Cantidad
PC de Mesa	Pentium IV	118
Laptop	Pentium IV	120
Impresoras	Laser, Deskjet y de Cinta	90
Sistema Operativos	Windows 2000 y XP	82.3% Windows XP

(Nodo de Comunicaciones Delegaciones)

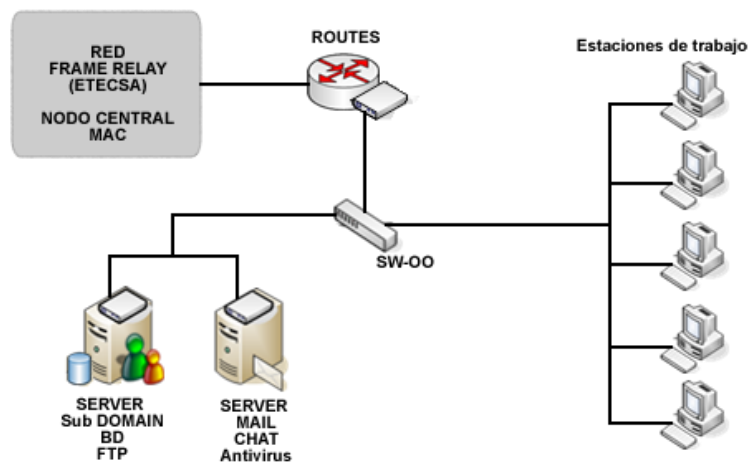


Figura 2.21. Estructura de Nodo de Comunicaciones en las delegaciones.

### 2.6.1.3 Características de los servidores en las Delegaciones

Tabla 6. Características de los servidores que utiliza el MAC en las delegaciones.

Cantidad de Equipos: 2

Dispositivo:	Características	Observaciones
CPU	3.40 GHz Pentium D dual core	
Memoria RAM	1.00 Gbytes	
Capacidad de almacenamiento	250 Gbytes	
Network	Velocidad: 1 puertos a Giga bit	
Sistema Operativos	Windows 2003	

### 2.6.1.4 Características de las Estaciones de Trabajo (Delegaciones Provinciales)

Tabla 7. Características de los equipos de las estaciones de trabajo que utiliza el MAC en las delegaciones provinciales.

Equipos:	Características	Cantidad
PC de Mesa	Pentium IV	9 Aproximadamente por Delegación
Laptop	Pentium IV	15 Aproximadamente por Delegación



---

<b>Impresoras</b>	<b>Laser, Deskjet y de Cinta</b>	<b>3 Aproximadamente por Delegación</b>
<b>Sistema Operativos</b>	<b>Windows 2000 y XP</b>	<b>71,6% Windows XP</b>

### **2.7 Conclusiones**

En este capítulo se llevó a cabo un estudio de las diferentes réplicas utilizadas en distintas organizaciones de nuestro país. Analizando el uso de las mismas y su importancia según el problema a resolver donde cada una tenía un objetivo fundamental, el de replicar los datos para evitar pérdidas lamentables, pero de manera diferente según el entorno y las necesidades de dichas organizaciones, todo este estudio realizado concluyó en una propuesta de réplica que será aplicable en el ministerio, según las necesidades y características del mismo.

## CAPÍTULO 3: VALIDACIÓN Y RESULTADOS DE LA PROPUESTA DE RÉPLICA REALIZADA PARA EL MAC

### 3.1 Introducción

En el capítulo anterior se llevó a cabo un estudio de las diferentes réplicas utilizadas en las distintas empresas de nuestro país, dando como resultado a dicho estudio la propuesta de réplica para el MAC. En el presente capítulo se realizarán diferentes pruebas para ver los diferentes resultados arrojados por dichas pruebas lo que nos dará una visión de lo factible que pueda ser la propuesta del MAC según las necesidades y características del mismo.

### 3.2 Descripción del proceso de réplica.

Previamente para aplicar la propuesta de réplica para el MAC, en los servidores que intervienen en el proceso de replicación hay que tener instalado las diferentes herramientas y software predefinidos por el proyecto MAC los cuales cumplen con los requisitos de la propuesta de réplica de base de datos. Como son:

- El sistema operativo Linux en su distribución Ubuntu 7.10.
- El gestor de base datos PostgreSQL8.2
- El Software de replicación Slony-I 1.1.2.9
- El pgAdminIII

Después de tener en los servidores las herramientas y Software planteados anteriormente, se define según la propuesta diseñada cuales serán los servidores maestros y los servidores esclavos según la estructura definida por la propuesta para el MAC, las base de datos de los servidores implicados en dicha réplica deben tener una misma estructura jerárquica para poder llevar a cabo la replicación entre ambos servidores.

#### 3.2.1 Pasos a seguir para replicar de un nodo maestro a un nodo esclavo.

- Permitir la conexión entre ambos servidores mediante la configuración del **pg\_hba.conf** y el **postgresql.conf**.
- Se instala el lenguaje PL/pgSQL en el SGBD PostgreSQL(ver anexo 2 )
- Se reinicia los SGBD en el servidor maestro y en el servidor esclavo (ver anexo 3).

- Se ejecuta el script **mainconfig** (ver anexo 4) solamente en el servidor maestro, el cual establece la conexión entre ambos servidores y especifica las tablas que van a ser replicadas entre dichos servidores.
- Se ejecuta el script **run\_master** (ver anexo 5) solamente en el servidor maestro, que es el encargado de crear previamente las condiciones para el escenario de replicación en el servidor maestro.
- Se ejecuta el script **run\_slave** (ver anexo 6) solamente en el servidor esclavo, que es el encargado de crear previamente las condiciones para el escenario de replicación en el servidor esclavo.
- Se ejecuta el script **start** (ver anexo 7) solamente en el servidor maestro el cual es el encargado de iniciar la réplica entre ambos servidores, después de este paso ya debe estar lista la réplica entre ambos servidores y debe quedar en el Pgadmin3 en el esclavo (como se muestra en la figura 3.1) y en el maestro (como se muestra en la figura 3.2).
- En caso de fallas en un servidor, por fluido eléctrico u otros factores y el mismo se apaga se deberá volver a ejecutar después de encendido en dicho servidor el **run\_master** o el **run\_slave** según el servidor que se apagó.
- En caso de pérdida de un servidor porque no funcione más se puede cambiar por otro y seguir los pasos explicados anteriormente o cambiar el funcionamiento de la réplica hacia otro servidor que se encuentre activo (funcionando) configurando los script y siguiendo los pasos anteriormente explicados.

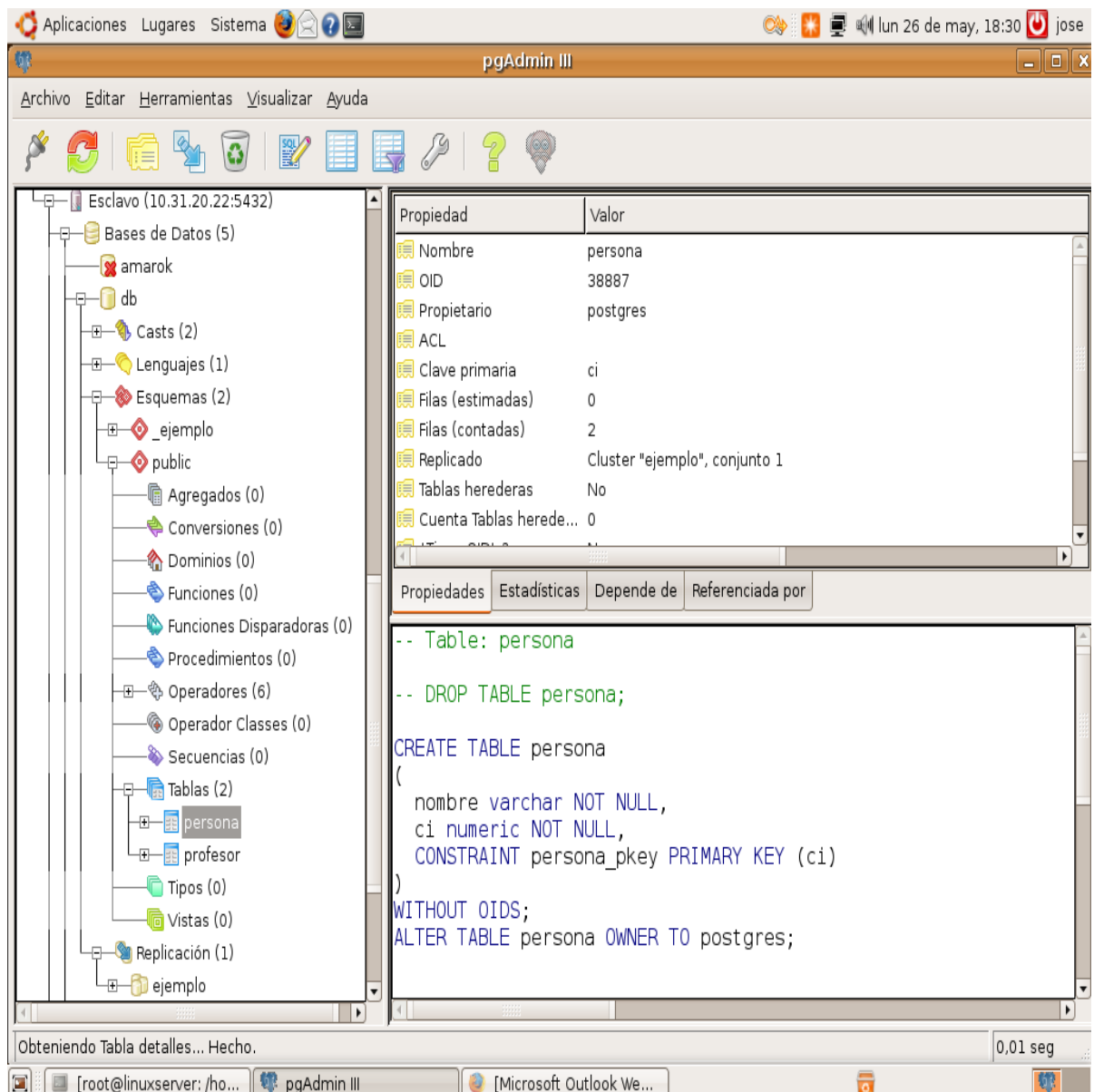


Figura 3.1 El servidor Esclavo luego de ejecutar el script start

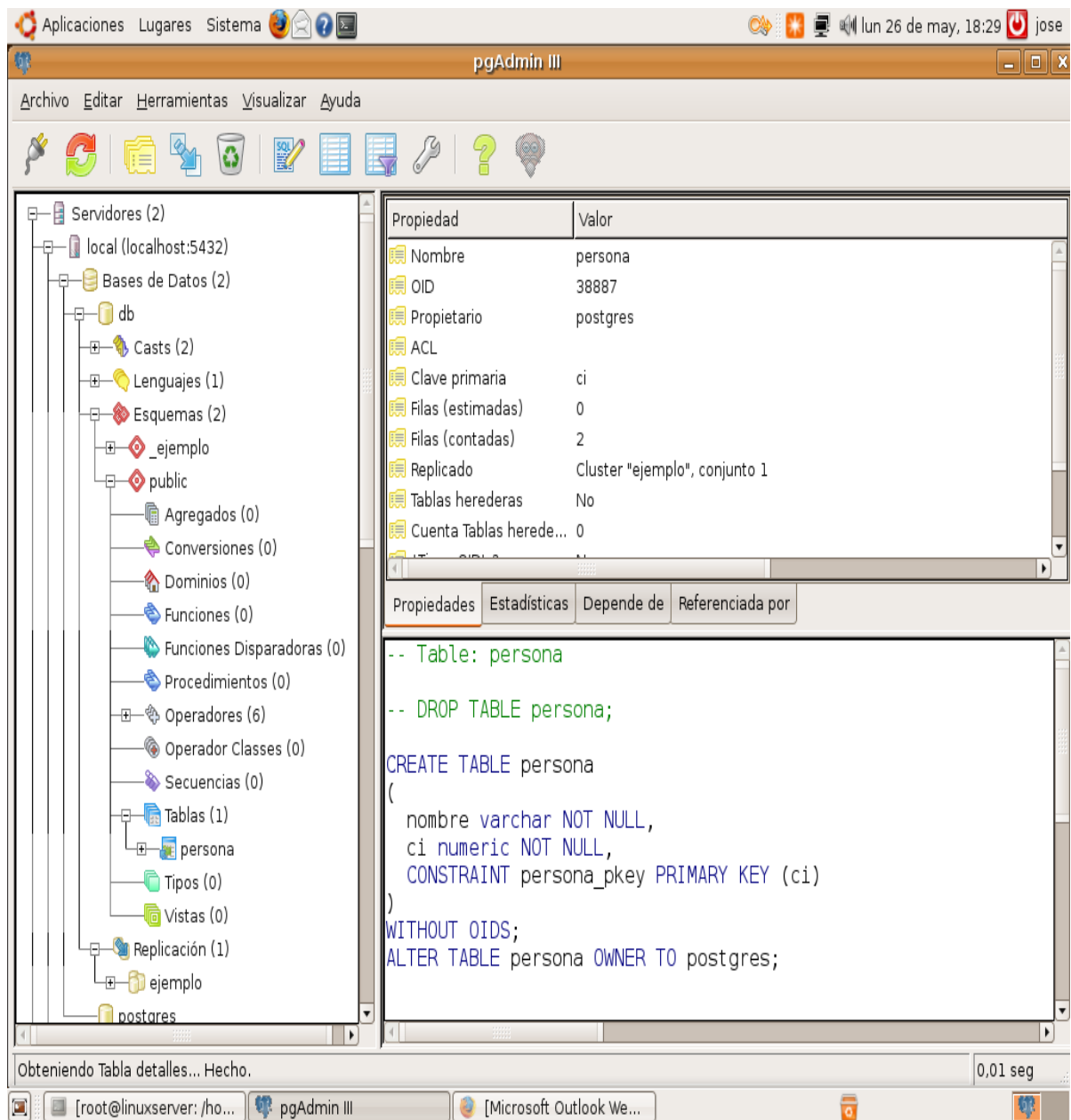


Figura 3.2 El servidor Maestro luego de ejecutar el script start

### 3.3-Pruebas realizadas a la propuesta del MAC.

Se realizaron diferentes pruebas a la propuesta del MAC expuesta en el capítulo anterior para verificar su funcionamiento y eficiencia, y comprobar si cumple con las propiedades necesarias que precisa dicho ministerio para garantizar un buen funcionamiento en sus bases de datos y una buena seguridad e integridad en sus datos. Estas pruebas se realizaron sobre computadores Pentium IV donde unas

van a funcionar como servidor esclavo y otras como servidor maestro, sobre una conexión de red de 100MBps, sobre el sistema operativo de Linux en su distribución Ubuntu7.10 y sobre el SGBD PostgreSQL en su versión 8.2, se escogió como muestra para realizar la replicación una base de datos de ejemplo en dicho SGBD con 5 tablas para comprobar el funcionamiento de la réplica en la propuesta realizada para el MAC.

### **3.3.1-Prueba #1**

#### **Objetivo:**

Verificar funcionamiento y rapidez de la réplica y comprobar si los datos son replicados de modo asincrónico y correctamente.

#### **Escenario:**

Se insertaron grandes cantidades de datos en las tablas del servidor maestro para verificar el funcionamiento y rapidez de la réplica entre el servidor maestro y el servidor esclavo, y se realizaron distintas operaciones sobre dichas tablas, como adicionar un dato, actualizar un dato, eliminar un dato, para comprobar la eficiencia de dicha réplica y la rapidez de realizar cada operación especificada anteriormente.

#### **Resultado:**

El resultado de dicha prueba fue satisfactorio ya que se replicaron los datos sin problema alguno en las distintas operaciones que se realizaron en las tablas maestras, con gran rapidez desde el servidor maestro hacia el servidor esclavo, logrando así el objetivo principal de dicha prueba que era comprobar si funcionaba la réplica entre ambos servidores de modo asincrónico. Dicha prueba arrojó las mismas tablas con iguales cantidades de datos en el servidor esclavo logrando así la integridad de los datos y la seguridad de los mismos.

### **3.3.2-Prueba #2**

#### **Objetivo:**

En medio del proceso de replicación dejar sin conexión de red ambos servidores para cuando se restableciera la conexión verificar si se replicaban los datos introducidos.

#### **Escenario:**

Se empezó a introducir datos en las tablas del servidor maestro y se replicaban en el servidor esclavo a medida que se introducían los mismos, luego se desconectó la red quedándose así sin conexión ambos servidores y se siguieron introduciendo datos en las tablas del servidor maestro, para

comprobar que pasaba con la replicación de datos cuando se restableciera la conexión entre ambos servidores.

**Resultado:**

El resultado de dicha prueba fue satisfactorio ya que al restablecer la conexión entre ambos servidores el Slony-I se encargó de verificar por donde se había quedado la replicación en las tablas esclavas al perder la conexión con el servidor maestro, y continuó a partir de ahí la replicación de los datos que le faltaban por replicar. Logrando así el objetivo principal de dicha prueba, que al restablecer la conexión se replicaran los datos que se introdujeron en la base datos después de haber estado sin conexión y no perder así la replicación de los mismos en el servidor esclavo. Por lo que se comprobó de esta manera la buena tolerancia a fallos del Slony-I como software de replicación.

### 3.3.3-Prueba #3

**Objetivo:**

En medio del proceso de replicación apagar el servidor esclavo y luego de un tiempo determinado encenderlo y verificar si se replicaron los datos que se introdujeron en las tablas del servidor maestro después de haber apagado el servidor esclavo.

**Escenario:**

Se empezó a introducir datos en las tablas del servidor maestro y se observó que los datos se replicaban hacia las tablas del servidor esclavo, después se apagó el servidor esclavo simulando su pérdida y se siguieron introduciendo datos en las tablas del servidor maestro para después de un tiempo determinado restablecer el servidor esclavo, ejecutar nuevamente el script run\_slave y observar que ocurría con la replicación entre dichos servidores.

**Resultado:**

El resultado de dicha prueba fue satisfactorio luego de restablecer el servidor esclavo y ejecutar el script run\_slave el Slony-I se encargó de verificar por donde se había quedado la réplica cuando se apagó dicho servidor y continuar replicando los datos que le faltaban hacia las tablas del servidor esclavo después de haber perdido la conexión con el mismo. Logrando así el objetivo principal de dicha prueba que consistía en verificar si se replicaban los datos después de apagado el servidor esclavo para así no perder la replicación de los mismos en dicho servidor.

### 3.3.4-Prueba #4

#### **Objetivo:**

Cambiar la replicación entre dos servidores (maestro-esclavo) hacia otro servidor activo en la réplica para comprobar si después de perdido un servidor en el esquema de replicación, por rotura u otro motivo se podría mantener la seguridad de los datos siendo replicados hacia otro servidor en la misma estructura de réplica hasta tanto se repare el servidor perdido.

#### **Escenario:**

Se apagó un servidor esclavo simulando su pérdida y se cambió la conexión de replicación hacia otro servidor superior donde se pudiera replicar dichos datos, siguiendo todos los pasos de reconfiguración de los script para dicho caso, para así comprobar que sucedía en caso de pérdida de un servidor, si se podía salvar los datos hasta tanto este se restableciera y así no tener pérdidas lamentables ya que estos datos no tendrían réplica (salva) ninguna hasta tanto no se repusiera dicho servidor.

#### **Resultado:**

El resultado de dicha prueba fue satisfactorio ya que al reconfigurar los script entre los nuevos servidores de replicación, cambiando así la replicación hacia otro servidor se replicaron los datos hacia el nuevo servidor sin problema alguno. Logrando así el objetivo principal de la prueba, que en caso de pérdida de un servidor, éste no dificultaría la replicación de los datos y la seguridad de los mismos, pues se cambió el destino de replicación de dicho servidor hacia otro servidor activo (funcionando) en el esquema de replicación.

### 3.4 Conclusiones

En este capítulo se llevaron a cabo diferentes pruebas realizadas a la propuesta que se expuso en el capítulo anterior, ya que se considera que se resolverán los problemas existentes en el ministerio con el resguardo y la seguridad de la información. Con el estudio del Slony-I se llegó a la conclusión de que éste software utilizado para la replicación resolvería completamente los intereses del ministerio de acuerdo a sus necesidades, ya que su utilización se ha puesto en práctica en algunos ministerios del país con magníficos resultados y además que las pruebas realizadas arrojaron magníficos resultados, ya que en cada caso de prueba se obtuvieron los resultados esperados y no hubo ninguna pérdida de información. Además de que el esquema de replicación se ajusta a las necesidades del MAC y es un esquema prácticamente escaso o nulo a la pérdida de información.



### CONCLUSIONES

Para el desarrollo del presente trabajo de diploma se realizó un estudio de las diferentes condiciones informáticas en las cuales se encontraba el Ministerio de Auditoría y Control mediante entrevistas a funcionarios del mismo y visitas a dicha institución para comprobar en que condiciones se encontraba el MAC, para luego desarrollar un estudio de réplica de base de datos en el SGBD PostgreSQL consultando una gran cantidad de fuentes bibliográficas acerca de las últimas tendencias y tecnologías que a nivel nacional e internacional se están usando en el mundo de la informática y en especial en la rama de la replicación de base de datos, para definir aquellas que mejores se ajustaban y mejor respuesta darían al problema planteado por el MAC para lograr darle solución al mismo satisfactoriamente, partiendo de la previa definición de herramientas y software con la cual se diseñó y se propuso un esquema de réplica para el Ministerio de Auditoría y Control para así garantizar la seguridad, eficiencia e integridad de los datos en todo éste proceso de intercambio de información entre el ministerio y sus sucursales provinciales.

Lográndose de esta manera los objetivos propuestos desde el comienzo de este trabajo y demostrándose además con el mismo que con la creación de un sistema que permita la seguridad, confiabilidad e integridad de los datos será posible mantener el intercambio de información entre el ministerio y sus sucursales provinciales.

### **RECOMENDACIONES**

Se recomienda que se lleve a cabo la puesta en práctica de esta propuesta, ya que con las pruebas que se realizaron se puede afirmar que dicha propuesta resolverá los problemas de resguardo de información en el MAC. Así como también que en próximos proyectos sea utilizada, pues es aplicable a otras empresas e instituciones que manejen información confidencial, siempre y cuando sigan la política de software libre, además de que se realice un testeo automático para verificar posibles fallas ocurridas a la hora de replicar grandes cantidades de datos en el MAC.

---

## BIBLIOGRAFÍA

### Consultada:

- Maestros del Web. *Maestros del Web*. [Online] Maestros del Web, 1997-2008.  
<http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>.
- Ciencomm.net. *Ciencomm.net*. [Online] Ciencomm. [Cited: enero 13, 2008.]  
<http://www.ciencomm.net/preguntas-frecuentes/bases-de-datos.html>.
- Alarcos. *Alarcos*. [Online] [Cited: enero 25, 2008.] [http://alarcos.inf-cr.uclm.es/doc/bda/doc/trab/T9900\\_OGonzalez.pdf](http://alarcos.inf-cr.uclm.es/doc/bda/doc/trab/T9900_OGonzalez.pdf).
- Universidad Alfonso X el sabio. *Universidad Alfonso X el sabio*. [En línea] [Citado el: 10 de febrero de 2008.] [http://www.uax.es/publicaciones/archivos/TECELS08\\_001.pdf](http://www.uax.es/publicaciones/archivos/TECELS08_001.pdf).
- **Vásquez, Gabriel**. Universidad del Cauca. *Universidad del Cauca*. [Online] [Cited: enero 20, 2008.] [atenea.unicauca.edu.co/~gavasquez/res/BD/material/IntroSBBD.pps](http://atenea.unicauca.edu.co/~gavasquez/res/BD/material/IntroSBBD.pps).
- **PostgreSQL, Grupo de desarrollo global de**. PostgreSQL. *PostgreSQL*. [En línea] 1996-2008. [Citado el: 13 de febrero de 2008.] <http://www.postgresql.org/>.
- ITESCAM (Instituto Tecnológico Superior de Calkini). [En línea] [Citado el: 3 de febrero de 2008.] <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r11294.PDF>.
- **Vega, Norge Fajardo**. *Sistema de réplica para bases de*. Ciudad de La Habana : s.n., 2007.  
[http://bibliodoc.uci.cu/TD/TD\\_0440\\_07.pdf](http://bibliodoc.uci.cu/TD/TD_0440_07.pdf)
- Carbonell, Luis Enrique Carrazana. *Replika Saren*. Ciudad de La Habana : s.n., 2008.
- Tripod. *Tripod*. [En línea] [Citado el: 10 de febrero de 2008.]  
<http://gustavolarriera.tripod.com/doc/tech/sql7repl.pdf>.
- **Slony-I, Grupo Desarrolladores de**. Slony-I. *Slony-I*. [Online] [Cited: febrero 22, 2008.]  
<http://www.slony.info>.
- **Desarrolladores, Grupo**. Trac. *Trac*. [Online] [Cited: marzo 1, 2008.] <http://www.pgcluster.org/>.
- **Linux, Grupo Desarrolladores de**. [www.linux-es.org](http://www.linux-es.org) El rincón de linux para hispanohablantes.  
*www.linux-es.org El rincón de linux para hispanohablantes*. [Online] [Cited: marzo 5, 2008.]  
[http://www.linux-es.org/sobre\\_linux..](http://www.linux-es.org/sobre_linux..)
- **Medina, José Manuel Alarcón**. Conselleria de Infraestructuras y Transporte. *Conselleria de Infraestructuras y Transporte*. [Online] Octubre/Noviembre 2006. [Cited: febrero 25, 2008.]  
<http://www.gvpontis.gva.es/fileadmin/conselleria/images/Documentacion/migracionSwAbierto/SI TARGES/manual.pdf>.
- **Hernandez, Luis**. Instituto universitario de Microelectrónica aplicada. *Instituto universitario de*

- Microelectrónica aplicada*. [Online] [Cited: Febrero 22, 2008.]  
<http://www.iuma.ulpgc.es/users/lhdez/inves/pfcs/memoria-ivan/node7.html>.
- **Luna, Luis**. [Online] noviembre 2005. [Cited: marzo 1, 2008.]  
Universidad de Carabobo facultad experimental de ciencia y tecnología. *Universidad de Carabobo facultad experimental de ciencia y tecnología*. [Online] noviembre 2005. [Cited: marzo 1, 2008.] <http://alfa.facyt.uc.edu.ve/computacion/pensum/cs0347/download/exposiciones2005-2006/Oracle%20Postgre%20MySQL.pdf>.
- **Desarrolladores, Grupo**. IpCorp Software. *IpCorp Software*. [Online] [Cited: Enero 25, 2008.]  
<http://www.ipcorp.com.ar/>.
- **Alonzo, Luis Ignacio**. Facultad de ingeniería Universidad de Uruguay. *Facultad de ingeniería Universidad de Uruguay*. [Online] [Cited: marzo 26, 2008.]  
<http://www.fing.edu.uy/inco/cursos/tagsi/Trabajos/2003/TAGSI03-TareaTecnSI-grupo4.pdf>.
- Ubuntu. *Ubuntu*. [Online] [Cited: mayo 10, 2008.]  
[http://www.guia-ubuntu.org/index.php?title=PgAdmin\\_III](http://www.guia-ubuntu.org/index.php?title=PgAdmin_III).
- Universidad Autónoma del Estado de México. Universidad Autónoma del Estado de México. [En línea] [Citado el: 12 de junio de 2008.] <http://www.uaemex.mx/plin/iuni/rev27/>.
- Micromouse. *Micromouse*. [En línea] [Citado el: 25 de febrero de 2008.]  
<http://www.micromouse.com/referencias/almacenamiento/altadisponibilidad-ree.asp>.

**Citada:**

1. Maestros del Web. Maestros del Web. [Online] Maestros del Web, 1997-2008.  
<http://www.maestrosdelweb.com/principiantes/%C2%BFque-son-las-bases-de-datos/>.
2. Ciencomm.net. Ciencomm.net. [Online] Ciencomm. [Cited: enero 13, 2008.]  
<http://www.ciencomm.net/preguntas-frecuentes/bases-de-datos.html>.
3. Alarcos. Alarcos. [Online] [Cited: enero 25, 2008.]  
[http://alarcos.inf-cr.uclm.es/doc/bda/doc/trab/T9900\\_OGonzalez.pdf](http://alarcos.inf-cr.uclm.es/doc/bda/doc/trab/T9900_OGonzalez.pdf).
4. Universidad Alfonso Xel sabio. Universidad Alfonso Xel sabio. [En línea] [Citado el: 10 de febrero de 2008.] [http://www.uax.es/publicaciones/archivos/TECELS08\\_001.pdf](http://www.uax.es/publicaciones/archivos/TECELS08_001.pdf).
5. Vásquez, Gabriel. Universidad del Cauca. Universidad del Cauca. [Online] [Cited: enero 20, 2008.] [atenea.unicauca.edu.co/~gavasquez/res/BD/material/IntroSBBD.pps](http://atenea.unicauca.edu.co/~gavasquez/res/BD/material/IntroSBBD.pps).
6. PostgreSQL, Grupo de desarrollo global de. PostgreSQL. PostgreSQL. [En línea] 1996-2008. [Citado el: 13 de febrero de 2008.] <http://www.postgresql.org/>.
7. ITESCAM(Instituto Tecnológico Superior de Calkini). [En línea] [Citado el: 3 de febrero de

2008.] <http://www.itescam.edu.mx/principal/sylabus/fpdb/recursos/r11294.PDF>.

8. Vega., Norge Fajardo. Sistema de réplica para bases de. Ciudadde La Habana : s.n., 2007.

[http://bibliodoc.uci.cu/TD/TD\\_0440\\_07.pdf](http://bibliodoc.uci.cu/TD/TD_0440_07.pdf)

9. Tripod. Tripod. [En línea] [Citado el: 10 de febrero de 2008.]

<http://gustavolarriera.tripod.com/doc/tech/sql7repl.pdf>

## ANEXOS

### Anexo1

#### Entrevistas realizadas a distintas empresas.

- ¿Que características tiene la empresa?
- ¿Objetivo que se busca con la aplicación de la réplica?
- ¿Tipo de réplica utilizada?
- ¿Utilidad de la réplica?
- ¿Tipo de gestor de base de datos utilizado?
- ¿Tipos de servidores utilizados?
- ¿Características de estos servidores?

### Anexo 2

#### Lenguaje PL/pgSQL

```
# createlang -U postgres -h 10.31.20.16 plpgsql base_dato
```

### Anexo 3

#### Restart SGBD

```
# /etc/init.d/postgresql-8.2 restart (tanto en master como en esclavo)
```

### Anexo 4

#### Script mainconfig

```
#!/bin/bash
CLUSTERNAME=ejemplo
DBNAME1=db
DBNAME2=db
MASTERHOST=10.31.20.16
SLAVEHOST=10.31.20.22
REPLICATIONUSER=postgres
slonik << _EOF_
cluster name=$CLUSTERNAME;
node 1 admin conninfo = 'dbname=$DBNAME1 host=$MASTERHOST user=$REPLICATIONUSER';
node 2 admin conninfo = 'dbname=$DBNAME2 host=$SLAVEHOST user=$REPLICATIONUSER';
```

```

init cluster (id=1, comment = 'Nodo Habana_Matanzas');
create set ( id=1, origin=1,comment= 'Cluster de prueba');
#tablas schema persona-----
set add table (set id=1, origin=1, id=1, fully qualified name = public.ejemplo', comment='tabla de
prueba');
#-----
store node (id=2, comment = 'Nodo Ministerial Esclavo CHabana');
store path (server = 1, client = 2, conninfo='dbname=$DBNAME1 host=$MASTERHOST
user=$REPLICATIONUSER ');
store path (server = 2, client = 1, conninfo='dbname=$DBNAME2 host=$SLAVEHOST
user=$REPLICATIONUSER ');
store listen (origin=1, provider = 1, receiver=2);
store listen (origin=2, provider = 2, receiver=1);
_EOF_

```

## **Anexo 5**

### **Script run\_master**

```

#!/bin/bash
CLUSTERNAME=ejemplo
DBNAME1=db
MASTERHOST=10.31.20.16
REPLICATIONUSER=postgres
slon $CLUSTERNAME "dbname=$DBNAME1 user=$REPLICATIONUSER host=$MASTERHOST"

```

## **Anexo 6**

### **Script run\_slave**

```

#!/bin/bash
CLUSTERNAME=ejemplo
DBNAME1=db
SLAVEHOST=10.31.20.22
REPLICATIONUSER=postgres
slon $CLUSTERNAME "dbname=$DBNAME1 user=$REPLICATIONUSER host=$SLAVEHOST"

```

## **Anexo 7**

### **Script start**

```

#!/bin/sh

```

```
CLUSTERNAME=ejemplo
DBNAME1=db
DBNAME2=db
MASTERHOST=10.31.20.16
SLAVEHOST=10.31.20.22
REPLICATIONUSER=postgres
slonik <<_EOF_
cluster name = $CLUSTERNAME;
node 1 admin conninfo = 'dbname=$DBNAME1 host=$MASTERHOST user=$REPLICATIONUSER';
node 2 admin conninfo = 'dbname=$DBNAME2 host=$SLAVEHOST user=$REPLICATIONUSER';
subscribe set (id = 1, provider = 1, receiver = 2, forward = yes);
_EOF_
```



### GLOSARIO DE TÉRMINOS

**Nodo:** En términos de la propuesta para replicar base de datos, son dos bases de datos PostgreSQL que intervienen en la réplica.

**Clúster:** En términos de Slony-I, un clúster son dos bases de datos PostgreSQL que intervienen en la réplica.

**Publicador:** Es un servidor que es la fuente de datos que deben repetirse. El publicador define un artículo para cada tabla u otro objeto de base de datos para ser utilizada como una fuente de replicación. Uno o más artículos de la misma base de datos se organizan en una publicación.

**Suscriptores:** Es un servidor que recibe los datos replicados por el editor. El suscriptor define una suscripción para una publicación particular. La suscripción se especifica cuando el suscriptor recibe la publicación del Publicador y mapas de artículos a las tablas y otros objetos de base de datos en el suscriptor.

**Peer-to-peer (punto a punto):** Comunicación segura en la cual intervienen solo dos máquinas.

**Distribuidor:** Es un servidor que realiza varias tareas cuando los artículos se desplazan desde Publicador a Suscriptores. Las tareas reales realizadas dependen del tipo de replicación realizada.

**Triggers:** Son segmentos de código almacenados en la base de datos, y que se disparan como respuesta a sucesos que tienen lugar en las aplicaciones.

**Monousuario:** En términos de base de datos un sistema monousuario sólo atiende a un usuario a la vez, y su principal uso se da en los ordenadores personales.

**Querys:** Un query es una herramienta del sistema que nos permite la posibilidad de extraer información. Consulta. Inserción, actualización, búsqueda o eliminación en una base de datos.

**Kerberos:** Es un sistema de autenticación utilizado para comprobar la identidad de un usuario o máquina.

**Licencia Open Source:** Open Source es código abierto en inglés es el término con el que se conoce al software distribuido y desarrollado libremente.