



UNIVERSIDAD NACIONAL DE CHIMBORAZO

FACULTAD DE INGENIERÍA

ESCUELA DE INGENIERÍA EN SISTEMAS Y COMPUTACIÓN

**“Trabajo de Grado previo a la obtención del Título de Ingeniero en Sistemas
y Computación.”**

TRABAJO DE GRADUACIÓN

Título del proyecto

ESTUDIO DE LAS TÉCNICAS DE REPLICACIÓN DE BASES DE DATOS Y
SU INCIDENCIA EN EL ALTO DESEMPEÑO, CASO APLICATIVO
REPLICACIÓN DE LA BASE DE DATOS ORACLE BAJO PLATAFORMA
GNU-LINUX DE LA EMPRESA TEVCOL.

Autor: Walter Ramiro Padilla Narvárez

Directora: Ing. Lida Barba

Riobamba – Ecuador

2013

UNIVERSIDAD NACIONAL DE CHIMBORAZO
FACULTAD DE INGENIERÍA
ESCUELA DE INGENIERÍA EN SISTEMAS Y
COMPUTACIÓN

Título del Proyecto

ESTUDIO DE LAS TÉCNICAS DE REPLICACIÓN DE BASES DE DATOS Y
SU INCIDENCIA EN EL ALTO DESEMPEÑO, CASO APLICATIVO
REPLICACIÓN DE LA BASE DE DATOS ORACLE BAJO PLATAFORMA
GNU-LINUX DE LA EMPRESA TEVCOL.

Los miembros del Tribunal de Graduación del proyecto de investigación de título: ESTUDIO DE LAS TÉCNICAS DE REPLICACIÓN DE BASES DE DATOS Y SU INCIDENCIA EN EL ALTO DESEMPEÑO, CASO APLICATIVO REPLICACIÓN DE LA BASE DE DATOS ORACLE BAJO PLATAFORMA GNU-LINUX DE LA EMPRESA TEVCOL. Presentado por: Walter Ramiro Padilla Narváez y dirigida por: Ing. Lida Barba.

Una vez escuchada la defensa oral y revisado el informe final del proyecto de investigación con fines de graduación escrito en la cual se ha constatado el cumplimiento de las observaciones realizadas, remite la presente para uso y custodia en la biblioteca de la Facultad de Ingeniería de la UNACH.

Para constancia de lo expuesto firman:

Presidente del Tribunal	Firma
--------------------------------	--------------

Miembro del Tribunal	Firma
-----------------------------	--------------

Miembro del Tribunal	Firma
-----------------------------	--------------

AUTORÍA DE LA INVESTIGACIÓN

“La responsabilidad del contenido de este Proyecto de Graduación, nos corresponde exclusivamente a: Walter Padilla (autor) y de la Ing. Lida Barba (directora); y el patrimonio intelectual de la misma a la Universidad Nacional de Chimborazo.

AGRADECIMIENTO

Agradezco a Dios por la vida brindada para poder alcanzar mis metas y culminar este proyecto, a mi madre que ha sido el eje fundamental para salir adelante, a mis amigos por su apoyo incondicional y a todos los docentes por su guía durante toda mi formación académica.

Walter Ramiro Padilla Narváez.

DEDICATORIA

Dedico a mi madre por su apoyo por ser el eje para luchar y lograr esta meta propuesta a una persona especial que siempre ha estado presente en los momentos más difíciles y en los más llevaderos de mi carrera le guardo una gratitud inmensa por su ayuda en todo momento y a todas las personas que con su ayuda, guía o consejo hicieron posible la realización de un logro más en mi vida

INDICE GENERAL

ÍNDICE DE FIGURAS.....	xii
ÍNDICE DE TABLAS	xiii
RESUMEN	xiv
SUMMARY	xv
INTRODUCCION	1
CAPITULO I	2
MARCO REFERENCIAL	2
1.1 Identificación y descripción del Problema	2
1.2 Formulación del Problema	3
1.3 Objetivos	3
1.3.1 Objetivo General	3
1.3.2 Objetivos Específicos	3
1.4 Justificación	3
1.5 Hipótesis	4
1.6 Delimitación.....	4
CAPÍTULO II.....	5
MARCO TEORICO.....	5
2.1 Introducción a las Bases de Datos Distribuidas	5
2.1.1 Definiciones	6
2.1.1.1 Base de Datos (BD).....	6
2.1.1.2 Sistema de Gestión de BD (SGBD).....	7
2.1.1.3 Red de Computadoras	8
2.1.1.4 Bases de Datos Distribuidas	8
2.1.1.5 Sistema de Gestión de Bases de Datos Distribuido (SGBDD)	9

2.1.2 Características de las BDD	9
2.1.3 Ventajas de las BDD	10
2.1.4 Desventajas de las BDD	13
2.2 Sistemas Gestores de Bases de Datos Distribuidas (SGBDD).....	14
2.2.1 Características y arquitectura de los SGBDD.....	14
2.2.2 Clasificación de los SGBDD	16
2.2.2.1 Distribución.....	16
2.2.2.2 Autonomía.....	16
2.2.2.3 Heterogeneidad	17
2.3 Diseño de BDD	18
2.3.1 Propuesta de replicación de base de datos Oracle	18
2.3.2 Alternativas de replicación	18
2.4 Replicación Oracle	20
2.4.1 Vista general de la replicación	20
2.5 Objetos, Grupos y Sitios Replicables	23
2.6 Tipos de ambientes de Replicación	25
2.6.1 Replicación Multimaster	25
2.6.2 Replicación de vistas materializadas	27
2.6.2.1 Vistas materializadas de solo lectura	28
2.6.2.2 Vistas materializadas actualizables	29
2.6.2.3 Vistas materializadas “writeable”	30
2.6.2.4 Subconjuntos de Filas y Columnas con vistas materializadas	30
2.6.2.5 Actualización de vistas materializadas	30
2.6.2.6 Grupos de refresco	31
2.6.2.7 Registro de vistas materializadas.....	31
2.6.3 Configuraciones Híbridas: Multimaster y vistas materializadas	31
2.6.4 Conflictos de Replicación	32

2.6.5 Detección de Conflictos y Replicación Procedural.....	33
2.7 Replicación multimaster.....	33
2.7.1 Conceptos de replicación multimaster.....	33
2.7.1.1 Por qué utilizar replicación multimaster.....	35
2.7.1.2 Proceso de replicación multimaster.....	36
2.7.2 Arquitectura de la replicación Multimaster.....	39
2.7.2.1 Mecanismos del sitio maestro.....	39
2.7.2.2 Mecanismos administrativos.....	39
2.8 Vistas Materializadas.....	40
2.8.1 Conceptos de vistas materializadas.....	40
2.8.1.1 Por que usar vistas materializadas.....	40
2.8.1.2 Vistas materializadas en múltiples capas.....	41
2.8.1.3 Mecanismos del master site y del master site de vista materializada.....	41
2.8.1.4 Trigger interno para el log de vistas materializadas.....	41
2.8.1.5 Log de vista materializada.....	41
2.8.1.6 Mecanismos de los sites de vistas materializadas.....	42
2.8.1.7 Tabla base y vista.....	42
2.8.1.8 Log de vista materializada actualizable.....	42
2.8.1.9 Trigger para el Log de vista materializada actualizable.....	42
2.9 Conflictos.....	42
2.9.1 Tipos de conflictos.....	42
2.9.2 Conflictos de datos y orden de transacción.....	43
2.9.3 Detección de conflictos.....	43
2.9.3.1 Como lo hace Oracle.....	43
2.9.3.2 Resolución de conflictos.....	44
2.9.4 Técnicas para evitar conflictos.....	44
2.9.4.1 Utilizar grupos de columnas.....	44

2.9.4.2 Sitio primario y sitio dinámico propiedad de modelos de datos	44
2.9.4.3 Evitando conflictos de unicidad	45
2.9.5 Arquitectura de resolución de conflictos	45
2.9.5.1 Mecanismos de soporte	45
2.9.5.2 Métodos de resolución de conflictos	46
2.10 Desarrollo de replicación de la base de datos Oracle en la empresa Tevcol	46
2.10.1 Base De Datos Standby	46
2.10.2 Opciones de configuración	47
2.10.3 La aplicación de registros de rehacer archivados a la base de datos standby	47
2.10.4 Ventajas y Desventajas.....	48
2.10.5 Modos Standby Database	49
2.10.6 Modo de recuperación Manual	50
2.10.7 Modo de recuperación Gestionado	52
2.10.8 Modo de sólo lectura	53
2.10.9 Replicación en una Base de Datos Standby (Manual)	53
2.10.11 Visión general de la replicación	54
2.10.12 Disponibilidad.....	54
2.10.13 Aplicaciones que utilizan la replicación	55
2.10.14 Objetos de replicación	55
2.11 Análisis de diferentes técnicas de replicación	56
CAPITULO III.....	58
METODOLOGIA	58
3.1 Método.....	58
3.2Tipo de Estudio	58
3.3 Población y Muestra.....	58
3.4 Procedimientos.....	58
3.4.1 Fuentes de Información	58

3.4.2 Procesamiento y Análisis	59
3.4.2.1 Teoría fundamentada en datos	59
3.4.2.2 Análisis de tareas.....	59
CAPITULO IV	60
RESULTADOS	60
4.1 Análisis de resultados.....	60
4.2 Elección de herramienta de Estudio	60
4.3 Metodologías de Estimación	60
4.3.1 Análisis de Puntos Función	60
4.4 Metodología escogido para la estimación de costos	61
4.5 Comprobación de Hipótesis	61
4.5.1 Hipótesis	61
4.5.2 Operacionalización de las variables	61
4.5.3 Comprobación.....	63
CAPITULO V.....	67
PROPUESTA REPLICACION BASE DE DATOS TEVCOL.....	67
5.1 La Empresa TEVCOL.....	67
5.1.1 Entorno de la Base de Datos TEVCOL Sin Replicación	68
5.1.2 Entorno de la Base de Datos de TEVCOL con Replicación	69
5.2 Instalación Oracle Database 10g Release 2 (10.2.0.1) Installation On Red Hat Enterprise Linux 5 (RHEL5) e implementación de la réplica.....	69
5.2.1 Proceso Preliminar	69
Paso siguiente solo se pulsa siguiente	74
5.2.2 Comprobación de los requisitos	74
5.3 Requerimientos para la replicación	78
5.4 Proceso de RECUPERACION	82
5.4.1 Preparación del servidor.....	82
5.4.2 Verificación de logs	86

5.4.3 Subir scripts de usuarios, dblinks , Jobs y Sequences de la base de datos	86
5.4.4 Checklist de Procesos de Recuperación.....	90
5.5 Desarrollo de la replicación utilizando base de datos standby, como manejador de base	91
CAPITULO VI	98
CONCLUSIONES Y RECOMENDACIONES	98
6.1 CONCLUSIONES	98
6.2 RECOMENDACIONES	99
CAPITULO VII	100
Bibliografía	100
CAPITULO VIII.....	101
ANEXOS	101

ÍNDICE DE FIGURAS

Figura 1: Base de Datos.....	6
Figura 2: Red de Computadoras.....	8
Figura 3: BDD (Base de Datos Distribuidas).....	9
Figura 4: Diseño De Bases de Datos Distribuida.....	18
Figura 5: Vista General de la Replicación.....	20
Figura 6: Replicación Multimaster.....	26
Figura 7: Replicación Master Group Quiesce.....	27
Figura 8: Replicación de Vistas Materializadas.....	28
Figura 9: Ambiente de Replicación Hibrido.....	32
Figura 10: Esquema de Modo de Recuperación Manual.....	50
Figura 11: Esquemas de Modo de Recuperación Gestionado.....	52
Figura 12: Modo de Solo Lectura.....	53
Figura 13: Pantalla inicio instalación base datos.....	73
Figura 14: Establecer directorio donde se instala la base de datos.....	74
Figura 15: Pantalla de comprobación del sistema.....	74
Figura 16: Pantalla en la cual se muestra los recursos a ser instalados.....	75
Figura 17: Proceso de la instalación.....	75
Figura 18: Confirmación del path.....	76
Figura 19: Final instalación software base de datos.....	76
Figura 20: Asistente para crear una base de datos.....	77

ÍNDICE DE TABLAS

Tabla 1: Configuraciones Posibles.....	48
Tabla 2: Resumen Tecnicas de Replicacion.....	56
Tabla 3: Operacionabilidad de Variables.....	62
Tabla 4: Configuracion de tablespace.....	64
Tabla 5: Descripcion Procesos.....	65

RESUMEN

La empresa TEVCOL se creó en el año de 1967 está dedicada al servicio de guardianía transporte de valores y custodia de joyas está al servicio del Ecuador en el transcurso de 45 años esta empresa se encuentra en 22 ciudades del país distribuidas en las diferentes sucursales tanto en la costa, sierra, oriente y las islas Galápagos se caracteriza por ser una empresa sólida brindando sus servicios al público.

La empresa cuenta con un sistema de base de datos, en el que se almacena la información correspondiente a custodia, logística, facturación, contabilidad, nómina, sib entre otros.

La problemática parte de saber que si bien la replicación es una de las mejores opciones para resguardar la integridad de una base de datos, es necesario conocer cuál de las técnicas existentes será la que mejor y la que más se ajuste a la necesidad de la empresa.

El primer paso se procedió es analizar las diferentes técnicas de replicación de Oracle, y en especial las técnicas de Oracle 10g Estándar ya que la empresa trabaja con este servidor de base de datos.

Debido a su amplia de cobertura, la empresa necesita debe tener una base de datos sólida para brindar un servicio eficiente a todos los usuarios y cumplir con los pedidos a tiempo, evitar demora en el proceso y caídas del sistema o cuellos de botella, de ahí la necesidad de tener una base de datos replicada y lista para ser utilizada cuando se presente cualquier inconveniente, por lo que se procedió a implementar la replicación de la base de datos tomando como método el base de datos en standby.

SUMMARY

The TEVCOL company was established in 1967 is dedicated to serving the guardianship, transportation of securities and custodian of jewelry is in the service of Ecuador in the course of 45 years, this company is located in twenty two cities across the country distributed in different branches both the coast, mountains, east and the Galapagos Islands is characterized by a solid company offering its services to the public.

The company has a database system, in which is stored the information on custody, logistics, billing, accounting, payroll, sib among others.

The problematic part of know that although replication is one of the best options to protect the integrity of a database, you must know which of the existing techniques will best and the one that best suits the needs of the company. The first step is to analyze the different proceeded Oracle replication techniques and especially Standard Oracle 10g techniques as the company works with the database server.

Because of its broad coverage, the company needs have a solid database to provide an efficient service to all users and fulfill orders on time, to avoid delay in the process and system crashes or bottlenecks, hence the need to have a replicated database and ready to be used when submitting any inconvenience, so we proceeded to implement the replication of the database as a method taking the standby database.

INTRODUCCION

La carencia de seguridad de la base de datos involucra que el trabajo desempeñado día a día no se realice de la mejor manera o la más eficiente ya que al no haber seguridades el desempeño de los usuarios está expuesto a factores como, el alto consumo del ancho de banda, cola de procesos, tráfico de datos etc.

Los mencionados factores crean problemas palpables principalmente en el recurso tiempo ya que provocan caídas del sistema, pérdida de datos, cuellos de botella, inconvenientes que deben ser minimizados con la aplicación de una replicación de la base de datos.

La consolidación de datos de diferentes sucursales hacia la matriz y un intercambio bidireccional de información hacia diferentes usuarios, con la réplica se logra minimizar el tráfico de datos entre los diferentes puntos donde se accede a la información.

Distribución de datos en otras ubicaciones y acceso a una misma tabla de diferentes usuarios con la réplica reducimos la carga en la red y tenemos un flujo de información más fluido.

Si bien la empresa TEVCOL trabaja con un distribuidor de base de datos tan robusto y eficiente como Oracle, no quiere decir que deje de estar susceptible a ataques por la red o a otros factores que puedan perjudicar a la empresa.

En este sentido es la empresa quien debe implementar políticas de seguridad sobre la información que maneja ya que la pérdida de información podría verse reflejada directamente en pérdidas económicas.

De este modo es imprescindible que la empresa aplique las seguridades necesarias para su base de datos y así evitar al máximo que la empresa se vea afectada por los factores antes mencionados.

CAPITULO I

MARCO REFERENCIAL

1.1 Identificación y descripción del Problema

La empresa TEVCOL se creó en el año de 1967 está dedicada al servicio de guardianía transporte de valores y custodio de joyas, se encuentra al servicio del Ecuador durante 45 años, esta empresa se encuentra en 22 ciudades del país distribuidas en las diferentes sucursales tanto en la costa, sierra, oriente y las islas Galápagos se caracteriza por ser una empresa sólida brindando sus servicios al público.

Empresas que tratan con información de tal importancia como TEVCOL deben estar al tanto de todos los peligros y riesgos de seguridad que en la actualidad puede con llevar el trabajar con una base de datos sin tomar las precauciones necesarias.

La empresa TEVCOL cuenta con un sistema de base de datos, en el que se almacena la información correspondiente a custodia, logística, facturación, contabilidad, nómina, entre otros.

La problemática parte de saber que si bien la replicación es una de las mejores opciones para resguardar la integridad de una base de datos, es necesario conocer cuál de las técnicas existentes será la que mejor y la que más se ajuste a la necesidad de la empresa.

El primer paso a realizar será el análisis de técnicas de replicación de Oracle, y en especial las técnicas de Oracle 10g Estándar ya que la empresa trabaja con este servidor de base de datos.

Debido a su amplia de cobertura, la empresa necesita debe tener una base de datos sólida para brindar un servicio eficiente a todos los usuarios y cumplir con los pedidos a tiempo, evitar demora en el proceso y caídas del sistema o cuellos de botella, de ahí la necesidad de tener una base de datos replicada y lista para ser utilizada cuando se presente cualquier inconveniente.

1.2 Formulación del Problema

La falta de aplicación de técnicas de replicación de la base de datos de la empresa TEVCOL impide su alto desempeño y disponibilidad.

1.3 Objetivos

1.3.1 Objetivo General

Estudiar las técnicas de replicación de bases de datos y su incidencia en el alto desempeño, caso aplicativo replicación de la base de datos Oracle bajo plataforma GNU-LINUX de la Empresa TEVCOL.

1.3.2 Objetivos Específicos

- Realizar el estudio de las diferentes técnicas de replicación de bases de datos.
- Seleccionar la técnica apropiada de replicación para plantear una propuesta de solución a la problemática de TEVCOL.
- Implementar la replicación con la técnica seleccionada.
- Determinar la incidencia de las técnicas de replicación en el alto desempeño de la base de datos de TEVCOL.

1.4 Justificación

La empresa TEVCOL es una empresa sólida con sucursales a nivel nacional por lo que se necesita que la base de datos no de problemas para que el trabajo de los usuarios sea eficiente el mismo que se lo logra teniendo una base de datos replicada.

De ese modo si la base tiene algún inconveniente, la base que se encuentra en standby tomará su lugar y no existirá ni cuellos de botella ni caídas de sistema y así los usuarios podrán realizar su trabajo de una forma más eficiente sin perder información.

El presente Proyecto de investigación permitirá a los usuarios de los distintos departamentos de la Empresa TEVCOL realizar un trabajo eficiente facilitando la toma de decisiones procesando datos para obtener información de calidad, de esta forma llevar una comunicación más fluida, segura y confiable.

Emprender una investigación sobre las técnicas de replicación de la base de datos bajo plataforma Linux a partir de la técnica más adecuada y compatible con el

sistema operativo utilizado en la empresa TEVCOL y su correspondiente implementación en este ayudará a mantener la base de datos más protegida contra las frecuentes amenazas que recibe Windows por virus o código malicioso.

1.5 Hipótesis

El uso de técnicas de replicación incide en el desempeño de la base de datos de la empresa TEVCOL.

1.6 Delimitación

- * Área: Departamento de Sistemas
- * Espacial: TEVCOL - Quito.
- * Temporal: 2011 – 2012.

Como limitante para el desarrollo de este proyecto se encontró el costo para la aplicación de cualquiera de las técnicas debido a que la empresa cuenta con licencias de Oracle Estándar y para ese tipo de software solo se puede aplicar ciertas técnicas sobre todo de aplicación gráfica.

CAPÍTULO II

MARCO TEORICO

2.1 Introducción a las Bases de Datos Distribuidas

Las Bases de Datos Distribuidas (BDD) son la unión de dos conceptos que, en principio, pueden parecer opuestos: los Sistemas Gestores de Bases de Datos (SGBD), y por otro lado, los sistemas en red. La primera parte, los SGBD, son sistemas, con sus correspondientes aplicaciones y archivos, de gestión y almacenamiento de datos que están disponibles en un solo lugar.

Bajo estos sistemas se implantan programas independientes entre sí, los cuales son totalmente indiferentes a cambios en la organización lógica y física de los datos y viceversa. La tecnología de redes de computadores, por otra parte, promueve un modo de trabajo que va en contra de todo esfuerzo de centralización

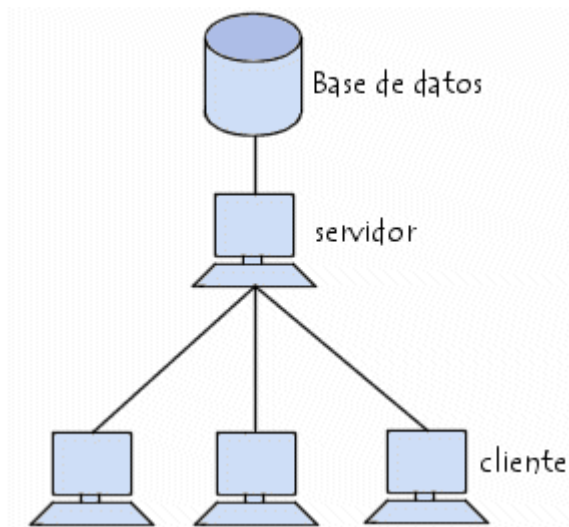
A simple vista resulta difícil entender cómo estos dos enfoques, que se oponen, pueden ser sintetizados para producir una tecnología que es mucho más poderosa que cualquiera de los dos enfoques por separado.

La clave para entender esto es darse cuenta de que en la tecnología de las BDD el objetivo principal no es la centralización sino más bien la integración. La idea consiste en lograr la integración y distribución y que ambos enfoques no pierdan su naturaleza esencial¹ (MARTÍNEZ, 2010).

¹ *Diseño y Construcción de Bases de Datos Distribuidas Heterogéneas sobre Oracle y SQL Server. MARTÍNEZ, L. (2010). Madrid: McGrawHill. Pag.262*

2.1.1 Definiciones

2.1.1.1 Base de Datos (BD)



Fuente:

Figura 1: Base de Datos

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

La definición anterior es muy general pero la aceptación común del término base de datos suele ser más restringida. Una BD tiene las siguientes propiedades implícitas:

Una BD representa algún aspecto del mundo real, en ocasiones llamado universo del discurso. Las modificaciones del universo del discurso se reflejan en la BD.

- Una BD es un conjunto de datos lógicamente coherente, con cierto significado incoherente.

- Una colección aleatoria de datos no puede considerarse propiamente una BD.
- Toda BD se diseña, construye y puebla con datos para un propósito específico.

Está dirigida a un grupo de usuarios y tiene ciertas aplicaciones preconcebidas que interesan a dichos usuarios.

En otras palabras, una BD tiene una fuente de la cual se derivan los datos, cierto grado de interacción con los acontecimientos del mundo real y un público que está activamente interesado en el contenido de la BD.

Las BBDD pueden ser de cualquier tamaño y tener diversos grados de complejidad pero deben organizarse y controlarse para que los usuarios puedan buscar, obtener y actualizar los datos cuando sea necesario. (LAURA, 2010)

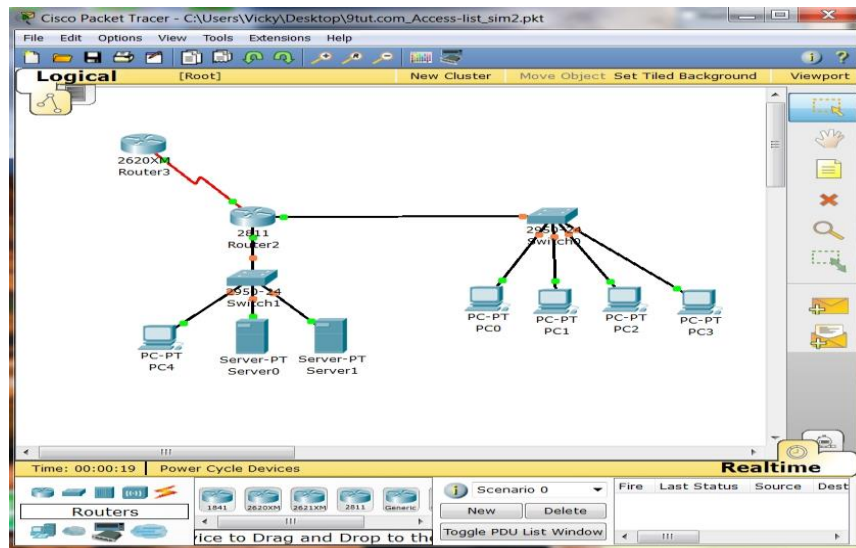
2.1.1.2 Sistema de Gestión de BD (SGBD)

Un SGBD es un conjunto de programas que permite a los usuarios crear y mantener una BD. Por tanto, el SGBD es un sistema de software de propósito general que facilita el proceso de definir; construir y manipular BBDD para diversas aplicaciones. Par definir una BD hay que especificar los tipos de datos, las estructuras y las restricciones de los datos que se almacenarán en ella. Construir una BD es el proceso de guardar los datos mismos en algún medio de almacenamiento controlado por el SGBD. En la manipulación de una BD intervienen funciones como consultar la BD para obtener datos específicos, actualizar la BD para reflejar cambios en el mini mundo y generar informes a partir de los datos.

No es necesario un software de SGBD de propósito general para implementar una BD computarizada. Se podría escribir un conjunto de programas propio para crear y mantener la BD. (LAURA, 2010)

2.1.1.3 Red de Computadoras

Una red de computadoras, también llamada red de ordenadores o red informática, es un conjunto de equipos (computadoras y/o dispositivos) conectados por medio de cables, señales, ondas o cualquier otro método de transporte de datos, que comparten información (archivos), recursos (CD-ROM, impresoras, etc.) y servicios (acceso a Internet, e-mail, Chat, juegos), etc.



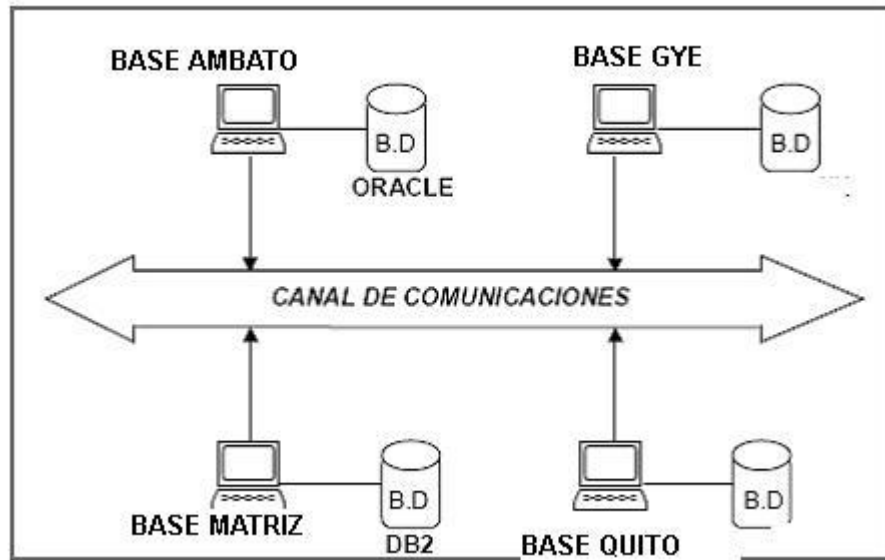
Fuente: Walter Padilla

Figura 2: Red de Computadoras

Una red de comunicaciones es un conjunto de medios técnicos que permiten la comunicación a distancia entre equipos autónomos (no jerárquica *-master/slave-*). Normalmente se trata de transmitir datos, audio y vídeo a través de diversos medios (aire, vacío, cable de cobre, fibra óptica, etc.). (Engeleiter, 2009)

2.1.1.4 Bases de Datos Distribuidas

Una Base de Datos Distribuida o BDD es una colección de datos integrados lógicamente en una BD, pero que físicamente pueden ser procesados y almacenados en varios nodos distribuidos sobre una red de ordenadores. (Luis, 2006)



Fuente: Walter Padilla

Figura 3: BDD (Base de Datos Distribuidas)

2.1.1.5 Sistema de Gestión de Bases de Datos Distribuido (SGBDD)

Un SGBDD es un producto software que permite el manejo de Bases de Datos Distribuidas (BDD) y hace la distribución transparente a los usuarios.

2.1.1.6 Sistema de Bases de Datos Distribuidas (SBDD)

Un SBDD es la unión entre un SGBDD y una BDD. Es decir, es un sistema en el cual múltiples sitios de BD están conectados en una red de computadoras de tal forma que, un usuario en cualquier sitio puede acceder a los datos en cualquier parte de la red exactamente como si los datos estuvieran siendo accedidos de forma local. (LAURA, 2010)

2.1.2 Características de las BDD

Podemos caracterizar las BDD en varios aspectos:

- Los datos deben estar físicamente en más de un ordenador, es decir, los datos se encuentran almacenados en distintas sedes.
- Las sedes deben estar interconectadas mediante una red de computadoras (Cada sede será un nodo de la red). Para realizar el diseño no se tendrá en cuenta la topología, tipo y rendimiento de la red, aunque estas propiedades tengan relevancia en el buen funcionamiento del sistema.

- Los datos han de estar lógicamente integrados en una única estructura o esquema lógico global común.
- Los usuarios han de tener acceso (recuperación y actualización) a los datos pertenecientes a la BDD, ya residan éstos en la misma sede (acceso local) o en otra sede (Acceso remoto).
- Cada nodo o emplazamiento facilita un entorno para la ejecución de transacciones tanto locales como globales.
- En una única operación, tanto de consulta como de actualización, se puede acceder a datos que se encuentran en más de una sede sin que el usuario sepa la distribución de los mismos en las distintas sedes. Es decir, que la distribución de la información sea transparente para el usuario. (Luis, 2006)

2.1.3 Ventajas de las BDD

Entre las ventajas potenciales de los sistemas de BDD se encuentran las siguientes:

- **La naturaleza distribuida de algunas aplicaciones de bases de datos**
Muchas de estas aplicaciones están distribuidas naturalmente en diferentes lugares. Por ejemplo, una compañía puede tener oficinas en varias ciudades, o un banco puede tener múltiples sucursales. Es natural que las bases de datos empleadas en tales aplicaciones estén distribuidas en esos lugares. Muchos usuarios locales tienen acceso exclusivamente a los datos que están en el lugar, pero otros usuarios globales pueden requerir acceso ocasional a los datos almacenados en varios de estos emplazamientos.
- **Mayor fiabilidad y disponibilidad.** La fiabilidad se define a grandes rasgos como la probabilidad de que un sistema esté en funciones en un momento determinado. La disponibilidad es la probabilidad de que el sistema esté disponible continuamente durante un intervalo de tiempo. Cuando los datos y el software del sistema gestor de base de datos están distribuidos en varios sitios, un sitio puede fallar mientras que los demás siguen operando. Sólo los datos y el software que residen en el sitio que falló están inaccesibles.

Esto mejora tanto la fiabilidad como la disponibilidad. En un sistema centralizado, el fallo de un solo sitio hace que el sistema completo deje de estar disponible para todos los usuarios.

- **Posibilidad de compartir los datos al tiempo que se mantiene un cierto grado de control local.** En algunos tipos de Sistemas de Bases de Datos Distribuidas (SBDD), es posible controlar los datos y el software localmente en cada sitio. Sin embargo, los usuarios de otros sitios remotos pueden tener acceso a ciertos datos a través del software del SGBDD. Esto hace viable poder compartir de forma controlada los datos en todo el sistema distribuido.
- **Mejor rendimiento.** Cuando una base de datos grande está distribuida en múltiples sitios, hay bases de datos más pequeñas en cada uno de éstos. En consecuencia, las consultas locales y las transacciones que tienen acceso a datos de un solo sitio tienen un mejor rendimiento porque las bases de datos locales son más pequeñas. Además, cada sitio tiene un menor número de transacciones en ejecución que si todas las transacciones se enviaran a una sola base de datos centralizada. En el caso de transacciones que impliquen acceso a más de un sitio, el procesamiento en los diferentes sitios puede efectuarse en paralelo, reduciéndose así el tiempo de respuesta.
- **Plataforma autónoma.** Dado que las aplicaciones y bases de datos no residen en las mismas máquinas, no hay ninguna razón particular para residir en el mismo tipo de máquina. Esta libertad permite a los administradores de bases de datos, desarrolladores y usuarios de escritorio elegir sus plataformas sin arriesgar cambios de funcionalidad en el motor de base de datos.
- **Localización transparente.** Esto significa que ni las aplicaciones ni los usuarios necesitan preocuparse de dónde residen los datos o cómo se distribuyen. Al ser protegidos por estas características específicas la usabilidad de una base de datos aumenta ya que los desarrolladores y los usuarios no necesitan tener en cuenta detalles tales como cadenas de conexión. Por otra parte, los datos pueden ser trasladados de una instancia

de base de datos a otra con un impacto mínimo sobre los usuarios y aplicaciones.

- **Autonomía del sitio.** Las BDD permiten varias localizaciones para compartir sus datos sin ceder el control administrativo. Si la sede de una instancia de base de datos contiene información especialmente sensible o tiene requerimientos de alta disponibilidad, todavía puede compartir datos sin comprometer su seguridad o la disponibilidad. Además, un sitio determinado en un entorno de BDD puede seguir sus propios procedimientos administrativos y actualización de rutas, dentro de lo razonable.
- **Seguridad mejorada.** Los componentes de la arquitectura distribuida son completamente independientes unos de otros, lo que significa que cada sitio puede mantenerse de forma independiente. Se pueden compartir los datos sin compartir cuentas y contraseñas. Cada sitio puede tener sus propios administradores y sus propios conjuntos de cuentas, y los datos privados se pueden mantener en privado. También se puede configurar un entorno distribuido para proporcionar seguridad en el sentido de la supervivencia, es decir, puede mantener dos o más versiones del esquema completo de la BD replicados en diferentes máquinas en diferentes lugares. ²En BDD no hay ninguna razón para que los desarrolladores o usuarios finales tengan su cuenta en un servidor de base de datos, porque todos los accesos de base de datos se producen a través de APIs (*Application Programming Interfaces*) de red. De este modo, se minimiza la exposición de dicho servidor de base de datos a posibles intrusiones maliciosas o a usuarios poco cuidadosos. (Engeleiter, 2009)

² Oracle Database. Redwood Shores, Engeleiter, K. (2009). CA 94065 EEUU: Oracle, Corporation. Pag 117

2.1.4 Desventajas de las BDD

En contra, la distribución produce un aumento en la complejidad del diseño y en la implementación de nuevos sistemas.

- **Seguridad.** Esta característica también aparece en el apartado de ventajas, dado que tiene dos caras, debido a que puede ser difícil saber y controlar quién está accediendo a una base de datos mediante un enlace de BD. Las cuentas a las cuales se conecta un enlace de BD no debe dar más derechos de acceso que los absolutamente necesarios.

En un entorno de replicación avanzada, los problemas de seguridad pueden llegar a ser complicados porque la comunidad de usuarios puede ser el conjunto de todos los usuarios de todas las bases de datos que participan en la distribución. El mantenimiento de las cuentas en sí mismas puede convertirse en un trabajo de tiempo completo. Habrá que decidir si las operaciones de replicación deben ser realizadas en sitios remotos por el usuario original o por una cuenta de replicación genérica.

- **Coherencia de los datos.** Si existen varias réplicas una de las tareas de mayor esfuerzo será garantizar la coherencia de datos. Esta responsabilidad es compartida entre Diseño y Construcción de Bases de Datos Distribuidas Heterogéneas sobre Oracle Y SQL Server los diseñadores, desarrolladores y administradores de bases de datos.
- **Administración de transacciones.** Si se necesita hacer un número de operaciones, pero tenemos la información en varios sitios replicada, debemos entender que el conjunto de operaciones debe ser realizado para cada sitio replicado. Si una solución posible es realizar la actualización por la noche, hora española, ¿qué ocurre con las sedes que están fuera de España y tienen un horario diferente y que quizá corresponda con su horario normal de trabajo? La cuestión es que las actualizaciones en un entorno replicado deben ser cuidadosamente coordinadas con todos los sitios a fin de evitar conflictos masivos y congestiones. En la carga inicial y la distribución de datos entre los sitios también se requiere coordinación. Por ejemplo, puede ser interesante bloquear a los usuarios de todos los

nodos hasta que se pueda garantizar que los datos son idénticos en todas las sedes.

- **Seguimiento.** La carga de trabajo de un DBA en un entorno distribuido puede ser considerable. El DBA debe controlar objetos tales como los registros de *snapshots*, las colas de trabajos, colas de transacciones, y las colas de errores. Si los problemas en un entorno distribuido se dejan sin resolver, puede ser tan difícil corregirlos que incluso sea más fácil recargar los datos desde el principio que tratar de solucionar errores específicos.
- **Recuperación.** Si una base de datos que forma parte de un entorno distribuido falla, el proceso de recuperación no sólo debe garantizar la restauración completa de los datos locales, sino también la restauración de los datos distribuidos, tales como las transacciones e instancias suspendidas.
- **Rendimiento.** Varios factores pueden afectar el rendimiento en una base de datos distribuida. Si la aplicación referencia datos desde un enlace de base de datos, el rendimiento de la red tendrá una influencia directa en el funcionamiento de la BDD. Los componentes de replicación que utilizan técnicas de almacenamiento y reenvío, tales como instantáneas y maestros de replicación, también tendrán su coste en el rendimiento global del sistema. El almacenamiento y la transmisión de operaciones impacta en el rendimiento general del sistema, y se debe tener en cuenta lo que esto significa cuando se especifican los requisitos de hardware. (Engeleiter, 2009)

2.2 Sistemas Gestores de Bases de Datos Distribuidas (SGBDD)

2.2.1 Características y arquitectura de los SGBDD

Los SGBDD son un producto software que permiten la gestión de la base de datos y posibilitan una distribución transparente a los usuarios.

El SGBDD debe contar con funciones de un SGBD centralizado y además con funciones tales como:

- Capacidad de acceder a sitios remotos y transmitir consultas y datos entre diversos sitios a través de una red de ordenadores.

- Capacidad de rastrear la traza de distribución y de replicación de los datos en el Diseño y Construcción de Bases de Datos Distribuidas Heterogéneas sobre Oracle Y SQL Server catálogo de SGBDD.
- Capacidad de elaborar estrategias de ejecución para consultas y transacciones con acceso a datos situados en nodos diferentes.
- Capacidad de mantener la consistencia en las réplicas de un elemento de información.
- Capacidad de decidir cuál de las copias de un elemento de información será accedida.
- Capacidad de recuperación ante caídas de sitios individuales y fallos de un enlace de comunicación.

El SGBDD debe contar con los siguientes componentes para poder ofrecer estas funcionalidades descritas:

- **Procesador de datos locales** que se encarga de la gestión local de los datos de forma parecida al software de un SGBD centralizado, por lo que además de ejecutar transacciones locales se encarga de la concurrencia y la recuperación ante fallos a nivel local.
- **Diccionario o directorio global** donde se guardará información acerca de dónde y cómo se almacenan los datos, el modo de acceso y otras características físicas. En resumen, contiene las especificaciones necesarias para pasar de la representación externa o esquema externo de los datos a la representación interna de los mismos.
- **Procesador de aplicaciones distribuidas** que es el responsable de las funciones distribuidas. Accede a la información sobre la ubicación de los datos, que se encuentra en el diccionario, y se ocupa de procesar todas las peticiones que involucran más de una sede para generar un plan de ejecución distribuido. Es el elemento diferenciador en los sistemas distribuidos, dada una operación se encargará de repartir el trabajo a los distintos procesadores locales que intervienen en dicha operación.
- **Software y red de comunicaciones.** No forma parte estrictamente del SGBDD sino que provee al procesador de aplicaciones distribuidas

primitivas y servicios de comunicaciones para que éste lleve a cabo su labor. (LAURA, 2010)

2.2.2 Clasificación de los SGBDD

Esta clasificación se realiza en base a tres características: la distribución, la autonomía y la heterogeneidad del sistema local.

2.2.2.1 Distribución

Los datos pueden ser distribuidos entre múltiples bases de datos. Estas bases de datos pueden estar almacenadas en un único sistema informático o en varios, distribuidos geográficamente, pero interconectados por un sistema de comunicación. Los datos pueden ser distribuidos entre múltiples bases de datos de diferentes maneras. Estos incluyen, en términos relacionales, particiones de base de datos verticales y horizontales.

Además pueden mantenerse varias copias de todo o parte de los datos. Estas copias no tienen por qué tener una estructura idéntica.

Algunos de los beneficios de la distribución de los datos son bien conocidos, tales como una mayor disponibilidad y fiabilidad de la información, así como una mejora en los tiempos de acceso

2.2.2.2 Autonomía

Se refiere a la distribución del control, no de los datos. Indica el grado en el que los SGBD pueden actuar independientemente y es función de factores tales como el nivel de intercambio de información entre los componentes del sistema, la ejecución independiente de transacciones, la tolerancia a modificaciones respectivas en los datos, etc.

Para lograr la independencia entre bases de datos que componen un sistema distribuido, es necesario considerar los tipos de autonomía descritos por:

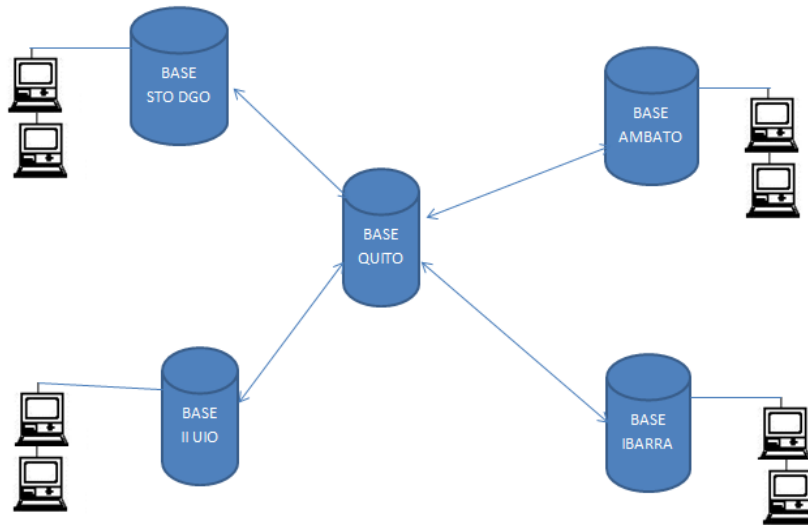
- **Autonomía de diseño:** La capacidad de que cada nodo de la base de datos decida los aspectos relacionados con su diseño. Es decir, las personas involucradas son libres de decidir cualquier particularidad e incluso decidir qué SGBD utilizar. Los aspectos a considerar en la autonomía de diseño son:
 - El universo de datos relevante para el sistema

- La representación (modelo de datos, lenguaje de consultas) y el nombrado de los datos.
- La conceptualización o interpretación semántica de los datos (heterogeneidad semántica)
- Restricciones usadas para administrar los datos
- La funcionalidad del sistema
- La asociación y compartición con otros sistemas
- La implementación (por ejemplo: registros y estructuras de archivos).
- **Autonomía de comunicación:** es la habilidad de que un nodo decida comunicarse o no con otro componente de una misma federación
- **Autonomía de ejecución:** es la habilidad de un nodo para ejecutar operaciones locales sin la interferencia de operaciones externas, en el orden en que el nodo lo decida.
- **Autonomía de asociación:** cada nodo cuánto y cuándo puede compartir su funcionalidad y recursos con otros componentes, inclusive la capacidad de asociarse o retirarse de una o más federaciones. (Luis, 2006)

2.2.2.3 Heterogeneidad

La heterogeneidad puede referirse a diferentes aspectos, desde diferencias hardware o en los protocolos de red, hasta variaciones en los SGBD propiamente (modelos de datos, lenguaje de manipulación de datos, protocolos de gestión de transacciones).

2.3 Diseño de BDD



Fuente: Walter Padilla

Figura 4: Diseño De Bases de Datos Distribuida.

En el diseño de las BDD, una cuestión clave es la distribución de los datos. El encargado de esta tarea es el DBA, el cual establece en tiempo de diseño el modelo de distribución de los datos. Esta definición es progresiva, de acuerdo a la evolución de la BD y al modo en que el SGBD resuelve:

- La distribución de las funciones
- La sincronización de las operaciones
- La descomposición de las consultas

2.3.1 Propuesta de replicación de base de datos Oracle

2.3.2 Alternativas de replicación

La replicación de datos se refiere al almacenamiento de copias de datos en múltiples sitios conectados mediante una red de ordenadores. Pueden guardarse copias fragmentadas en varios sitios para satisfacer requerimientos de información específicos.

Como la existencia de copias de fragmentos puede mejorar la disponibilidad de los datos y el tiempo de respuesta, estas copias reducen los costes de comunicación y de consultas totales.

Supóngase que la tabla A está dividida en dos fragmentos A1 y A2. Dentro de una base de datos distribuida replicada, es posible el escenario de la Ilustración 6 en el que el fragmento A1 se guarda en los sitios S1 y S2, mientras que el A2 se guarda en los sitios S2 y S3.

Los datos replicados se someten a la regla de consistencia mutua. La regla de consistencia mutua requiere que todas las copias de fragmentos de datos sean idénticas.

Por consiguiente, para mantener la consistencia de los datos entre las réplicas, el SGBDD debe garantizar que se realice una actualización de la base de datos en todos los sitios donde existan réplicas.

Suponiendo que la base de datos está fragmentada correctamente, hay que decidir la asignación de los fragmentos a diversos sitios en la red. Cuando se asignan los datos, o bien se pueden replicar o mantenerse una sola copia de los mismos. Las razones para la replicación son la disponibilidad, fiabilidad y la eficiencia de consultas de sólo lectura.

Si hay múltiples copias de un elemento de datos, hay una buena probabilidad de que alguna copia esté accesible en algún lugar, incluso cuando se producen fallos en el sistema. Por otra parte, las consultas de sólo lectura que acceden a los mismos elementos de datos pueden ser ejecutados en paralelo desde la salida de copias en múltiples sitios.

Por otra parte, la ejecución de las transacciones de actualización causa problemas ya que el sistema tiene que garantizar que todas las copias de los datos se actualizan correctamente.

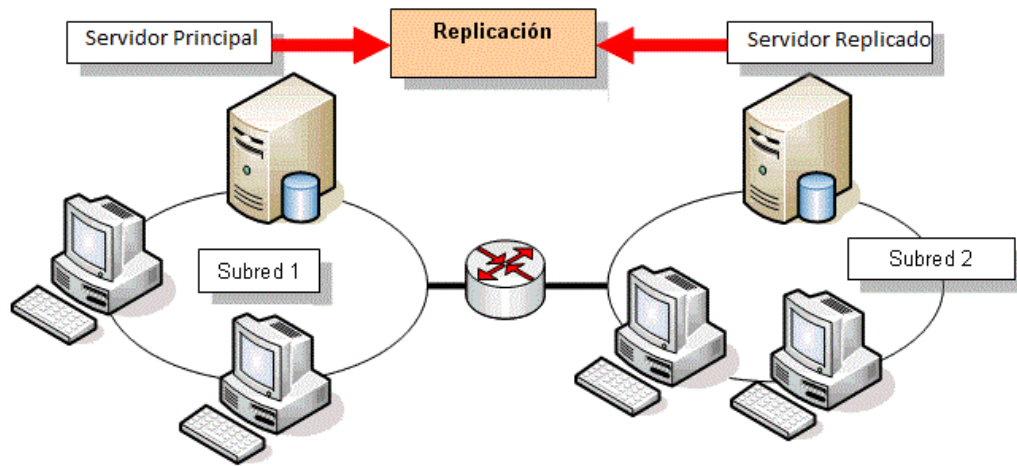
De ahí que la decisión con respecto a la replicación dependa de la relación existente entre consultas de sólo lectura y consultas de actualización. Esta decisión afecta a casi todos los algoritmos de SGBDD y las funciones de control.

Una base de datos no replicada (comúnmente se llama base de datos particionada) contiene fragmentos que se asignan a los diferentes sitios, y sólo hay una copia de cada fragmento en la red. En el caso de realizarse replicación, la base de datos

puede ubicarse en su totalidad en cada lugar (base de datos totalmente replicada), o los fragmentos pueden distribuirse en los nodos de tal manera que las copias de un fragmento puedan residir en varios emplazamientos (base de datos parcialmente replicada). En este último caso el número de copias de un fragmento puede suponer un aumento de coste para el algoritmo de asignación.³(Luis, 2006)

2.4 Replicación Oracle

2.4.1 Vista general de la replicación



Fuente: Walter Padilla

Figura 5: Vista General de la Replicación

Replicación es el proceso de copiado y mantenimiento de objetos de la base de datos, tales como tablas, en bases de datos múltiples “simulan” un sistema de bases de datos distribuido. Los cambios hechos en un sitio son capturados y almacenados localmente antes de ser enviados y aplicados en cada una de las locaciones remotas. La replicación avanzada de Oracle es una característica del servidor Oracle, no es un servidor separado.

La replicación utiliza tecnología de bases de datos distribuidas para compartir datos entre múltiples sitios, pero una base de datos replicada y una base de datos distribuida no son lo mismo. En una base de datos distribuida, los datos están disponibles en muchas locaciones, pero una tabla en particular reside solo en una

³ *Bases de Datos Distribuidas. Luis, P. M. (2006). Barcelona Esp.Pag.115.*

locación. Replicación significa que el mismo dato está disponible en múltiples locaciones. (Heurtel, 2009)

Algunas de las razones principales para utilizar la replicación son:

- **Disponibilidad:** La replicación mejora la disponibilidad de las aplicaciones, ya que provee opciones alternativas de acceso a los datos. Si un sitio no se encuentra disponible, los usuarios pueden continuar con consultando o actualizando las locaciones restantes. En otras palabras la replicación provee una excelente protección contra fallos.
- **Rendimiento:** La replicación provee un rápido acceso local para compartir datos ya que balancea la actividad sobre múltiples sitios. Algunos usuarios pueden acceder a un servidor mientras que otros acceden a diferentes servidores, reduciendo la carga de todos los servidores. También, los usuarios pueden acceder a datos de un sitio replicado que tiene un menor costo de acceso, el cual es típico que el sitio se encuentre geográficamente más cerca.
- **Computación Desconectada:** Una vista materializada es una copia (replica) completa o parcial de una tabla en un determinado momento. Las vistas materializadas habilitan a los usuarios trabajar sobre un subconjunto de la base de datos mientras están desconectados del servidor de base de datos central. Más tarde, cuando se establezca una conexión, los usuarios pueden sincronizar (refresh) las vistas materializadas.

Cuando los usuarios refrescan las vistas materializadas, actualizan la base de datos central con todos los cambios que hayan hecho y recibirán los cambios que puedan haber ocurrido mientras estaban desconectados.

- **Reducción de la carga:**
La replicación puede ser utilizada para distribuir datos sobre múltiples localizaciones regionales. Así, las aplicaciones pueden acceder a varios servidores regionales en vez de acceder a un servidor central. Esta configuración puede reducir la carga de la red drásticamente.
- **Despliegue en masa:** Cada vez más, las organizaciones necesitan hacer uso de algunas aplicaciones que requieren la capacidad de usar y

manipular datos. Con la replicación de Oracle, las plantillas de uso habilitan crear múltiples ambiente de vistas materializadas rápidamente. Se pueden utilizar variables para personalizar cada ambiente de vistas materializadas según las necesidades individuales.

Replicación lo soportan una variedad de aplicaciones que a menudo tiene diferentes requerimientos. Algunas aplicaciones permiten sitios de vistas materializadas relativamente autónomos. Por ejemplo, automatización de ventas, servicio de campo, venta al por menor, despliegue en masa, típicamente requieren datos para ser periódicamente sincronizados entre el sistema central de base de datos y un gran número de sitios pequeños remotos, los cuales a menudo son desconectados desde la base de datos central. En ese caso, los sitios remotos deben ser autónomos.

Por otra parte, aplicaciones tales como centros de llamadas y sistemas de Internet requieren datos en múltiples servidores para ser sincronizados de manera continua, casi instantánea para asegurarse que el servicio que se provee está disponible y equivalente en todo momento. Por ejemplo, un sitio Web de ventas al por menor en Internet debe asegurarse que los clientes vean la misma información en el catálogo online en cada sitio. Aquí, la consistencia de datos es más importante que la autonomía del sitio.

Replicación en Oracle puede ser usada para cada tipo de aplicaciones descritas en el párrafo anterior, y para sistemas que combinen aspectos de ambos tipos de aplicaciones. De hecho, la replicación en Oracle puede soportar ambos, despliegue en masa y replicación servidor a servidor, habilitando la integración en un ambiente único y coherente. En tal ambiente pueden compartir datos.⁴

Replicación Oracle puede replicar datos en ambientes que usen diferentes versiones de Oracle y en ambientes que corran Oracle en diferentes sistemas operativos. Por lo tanto, aplicaciones que usen datos en tales ambientes pueden usar replicación Oracle. (Engeleiter, 2009).

⁴ *Oracle Database. Redwood Shores, Engeleiter, K. (2009). CA 94065 EEUU: Oracle Corporation. Pag.181*

2.5 Objetos, Grupos y Sitios Replicables

Las siguientes secciones explican los componentes básicos de un sistema de replicación, incluyendo objetos, grupos y sitios de replicación.

- **Objetos Replicables:** Un objeto replicable es un objeto de una base de datos existente en múltiples servidores en un sistema de base de datos distribuido. En un ambiente de replicación, cualquier actualización hecha a un objeto replicable en un sitio es aplicada a las copias en todos los otros sitios.

Replicación Oracle permite replicar los siguientes tipos de objetos:

- Tablas
- Índices
- Vistas y Object Views
- Packages y Packages Bodies
- Procedimientos y Funciones
- Tipos definidos por el usuario y Type Bodies
- Triggers
- Sinónimos
- Indextypes
- Operadores definidos por el usuario.

En cuanto a tablas, la replicación soporta avanzadas características tales como tablas particionadas, tablas organizadas por índices, tablas conteniendo columnas que se basan en tipos definidos por el usuario y tablas objetos.

- **Grupos de replicación:** En un ambiente de replicación, Oracle maneja objetos replicables usando grupos replicación. Un grupo replicación es una colección de objetos replicables que están relacionados lógicamente.

Organizando objetos de bases de datos relacionadas dentro de un grupo replicación, es más fácil administrar muchos objetos juntos. Típicamente, se crea y usa un grupo de replicación, para organizar el esquema de objetos necesarios para soportar una aplicación de base de datos particular. Sin embargo, grupos de replicación y esquemas no necesitan corresponder uno con otro. Un grupo de replicación puede contener objetos de múltiples esquemas y un esquema singular puede tener objetos en múltiples grupos de replicación.

Sin embargo, cada objeto replicable puede ser miembro sólo de un grupo de replicación.

- **Sitios Replicables:** Un grupo de replicación puede existir en múltiples sitios replicables. Los ambientes de replicación soportan dos tipos básicos de sitios: sitios maestros y sitios de vistas materializadas. Un sitio puede ser tanto un sitio maestro para un grupo replicable y un sitio de vistas materializadas para otro grupo replicable diferente. Sin embargo, un sitio no puede ser a la vez sitio maestro y sitio de vista materializada de un mismo grupo replicable.

Las diferencias entre sitio maestro y sitio de vista materializada son las siguientes:

- Un grupo de replicación en un master site es referido más específicamente como un master group. Un grupo de replicación en un sitio de vista materializada está basado en un master group y es referido más específicamente como un grupo de vista materializada. Adicionalmente, cada master group tiene exactamente un master definition site. Un master definition site de un grupo de replicación es un master site, sirviendo como el centro de control, para manejo de grupos de replicación y objetos en el grupo.
- Un master site mantiene una copia completa de todos los objetos en un grupo de replicación, mientras que las vistas materializadas en un materialized view site pueden contener todos o un subconjunto de la tabla de datos dentro de un master group. Por ejemplo, si el master group x contiene las tablas y, z, entonces todos los master sites participando en el master group deben mantener una copia completa de y,z. Sin embargo, una vista de un materialized view site debería contener solo una vista materializada de la tabla de y, mientras que otra materialized view site debería contener vistas materializadas de ambas tablas, de y,z.
- Todos los master sites en un ambiente de replicación multimaster se comunican directamente con uno y otro para propagar continuamente los cambios de datos en el grupo de replicación. Materialized view sites contienen una imagen, o vista materializada, de la tabla de datos desde un cierto punto en el tiempo.
- Típicamente, una vista materializada es refrescada periódicamente para sincronizarla con su master site. Se puede organizar vistas materializadas en

grupos de refresco. Vistas materializadas en un grupo de refresco pueden pertenecer a uno o más grupos de vistas materializadas, y ellas son actualizadas al mismo tiempo para asegurar que los datos en todas las vistas materializadas en el grupo de refresco correspondan al mismo, tradicionalmente consistente punto de tiempo. (Luis, 2006)

2.6 Tipos de ambientes de Replicación

Replicación Oracle soporta los siguientes tipos de ambientes de replicación:

- Replicación Multimaster
- Replicación de Vistas Materializadas.
- Configuraciones híbridas Multimaster y Vistas Materializadas.

2.6.1 Replicación Multimaster

Replicación Multimaster permite sitios múltiples, actuando como pares iguales, para dirigir grupos de objetos de base de datos replicables. Cada sitio en un ambiente de replicación multimaster es un master site, y cada sitio se comunica con los otros master sites.⁵

Las aplicaciones pueden actualizar cualquier tabla replicada en cualquier sitio en una configuración multimaster. Los servidores de base de datos Oracle operando como master sites en un ambiente multimaster, automáticamente trabajan para reunir los datos de todas las tablas de réplicas y para asegurar la consistencia global de la transacción y la integridad de los datos.

La replicación asincrónica es el medio más común para implementar la replicación multimaster. Otros medios incluyen replicación sincrónica y replicación procedural.

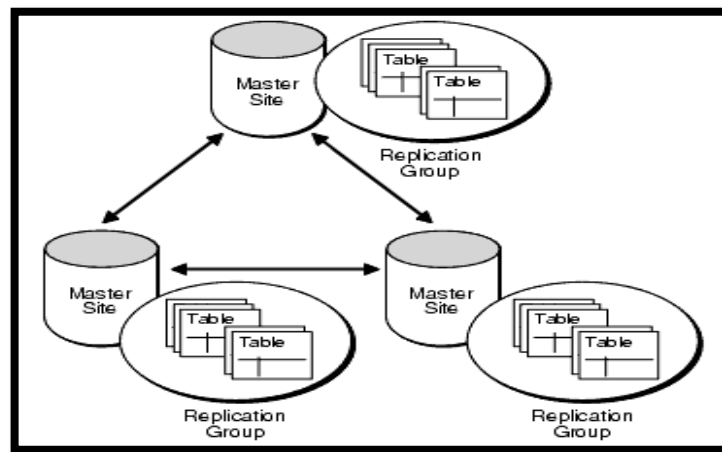
Cuando se usa replicación asincrónica, la información acerca del lenguaje de manipulación de datos (DML) que cambia en una tabla es guardado en la cola de transacciones diferidas en el master site donde el cambio ocurrió.

⁵ *Bases de Datos Distribuidas. Luis, P. M. (2006). Bcelona Esp.Pag.194*

Estos cambios son llamados transacciones diferidas. Las transacciones diferidas son empujadas o propagadas hacia el otro participante del master site en intervalos regulares. Se puede controlar la cantidad de tiempo de un intervalo.

Usar replicación asincrónica significa que los conflictos de datos son posibles, porque el mismo valor de fila puede ser actualizada en dos master sites diferentes, casi al mismo tiempo. Sin embargo, se puede usar técnicas para evitar conflictos y, si el conflicto ocurre, Oracle provee mecanismos preconstruidos que pueden ser implementados para resolverlos.

La información sobre los conflictos sin resolver es almacenada en un Log de errores. (Luis, 2006)

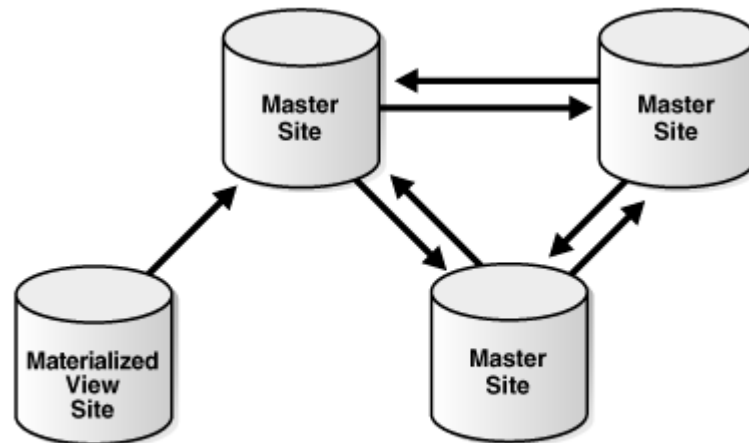


Fuente: Luis, P. M. (2006).

Figura 6: Replicación Multimaster

- **Master Group Quiesce:** A veces, es necesario parar toda actividad de replicación de un master group para poder ejecutar ciertas tareas administrativas en el master group. Por ejemplo, se debe parar toda actividad de replicación de un master group para agregar un nuevo objeto de master group. Parar toda la actividad de replicación de un master group es llamado “quiescing” del grupo. Cuando un master group es parado, los usuarios no pueden llamar declaraciones DML de ninguno de los objetos del master group. También, todas las transacciones diferidas deben ser propagadas antes de que se pueda parar un master group. Los usuarios

pueden continuar consultando las tablas en un master group parado.
(Engeleiter, 2009)⁶



Fuente: Engeleiter, K. (2009).

Figura 7: Replicación Master Group Quiesce

2.6.2 Replicación de vistas materializadas

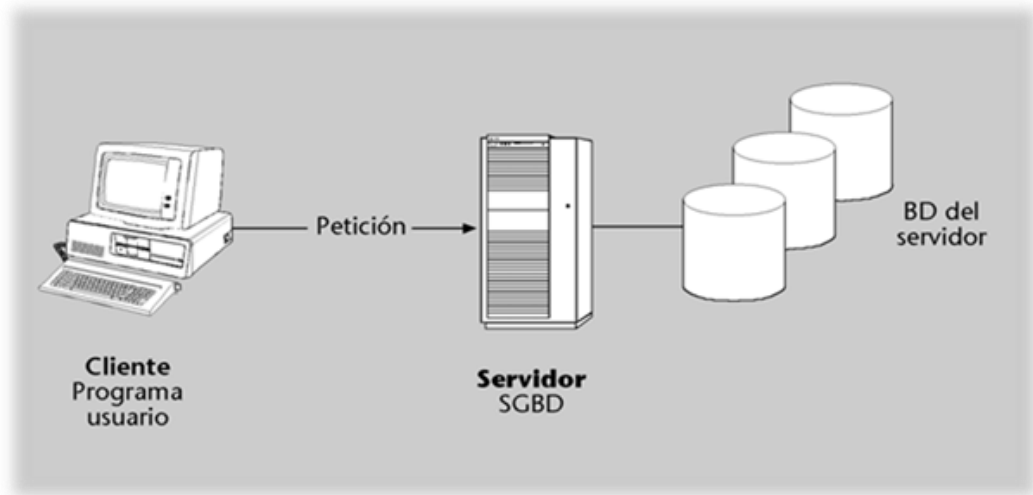
Una vista materializada contiene una copia completa o parcial de un “target master” desde un punto específico en el tiempo. El “target master” puede ser a la vez una tabla master en un master site o una vista materializada master en un site de vistas materializadas. Una vista materializada master es una vista materializada que funciona como un master para otra vista materializada. Una vista materializada multimaster es una que está basada en otra vista materializada, en lugar de una tabla master.

Las vistas materializadas proveen los siguientes beneficios:

- Permite accesos locales, lo cual mejora los tiempos de respuesta y la disponibilidad.
- Carga de queries sin conexión del master site o del sitio de vista materializada master, porque en su lugar los usuarios pueden consultar la vista materializada local.

⁶ Engeleiter, K. (2009). *Oracle Database*. Engeleiter, K. (2009). Redwood Shores, CA 94065 EEUU: Oracle Corporation. Pag.179

- Incrementa la seguridad de los datos permitiendo replicar solo un subconjunto.
- Un materialized view puede leerse en solo lectura, actualizarse, o escribirse, y estos tipos de vistas materializadas proveen beneficios en suma de aquellos enumerados previamente. (Luis, 2006)



Fuente: Luis, P. M. (2006).

Figura 8: Replicación de Vistas Materializadas.

2.6.2.1 Vistas materializadas de solo lectura

En una configuración básica, las vistas materializadas pueden proveer acceso de solo lectura a la tabla de datos que se originan desde un sitio master o un sitio master de vistas materializadas. Las aplicaciones pueden consultar datos de vistas materializadas de solo lectura para evitar accesos al sitio master a través de la red, prescindiendo de la disponibilidad de la red. Sin embargo, las aplicaciones a lo largo de todo el sistema deben acceder a datos en el sitio master para ejecutar sentencias DML. Las tablas master y las vistas materializadas master de solo lectura no necesitan pertenecer a un grupo de replicación.

(Heurtel, 2009)

Las vistas materializadas de solo lectura proveen los siguientes beneficios:

- Eliminan la posibilidad de conflictos porque no pueden ser actualizados.

- Soportan vistas materializadas complejas. Vistas materializadas complejas son por ejemplo las que contienen operaciones “set” o una cláusula CONNECT BY. (LAURA, 2010)⁷

2.6.2.2 Vistas materializadas actualizables

En una configuración más avanzada, se puede crear una vista materializada actualizable que permita a los usuarios insertar, actualizar, y borrar filas de las tablas del master objetivo o del master de vistas materializadas ejecutando estas operaciones en el materialized view. Una vista materializada actualizable puede también contener un subconjunto de los datos en el master.

Las vistas materializadas actualizables están basadas en tablas o en otras vistas materializadas que han sido establecidos para soportar la replicación. De hecho, las vistas materializadas actualizables deben ser parte de un grupo de vistas materializadas que esté basado en otro grupo de replicación.

La vistas materializas actualizables tiene las siguientes propiedades.

- Están siempre basados en una tabla única, aunque tablas múltiples pueden ser referenciadas en una subconsulta. Pueden ser actualizadas de forma incremental o rápida.
- Oracle propaga los cambios hechos a una vista materializa actualizable a una tabla de vistas remota master o al master de vistas materializas. Las actualizaciones del master se replican en cascada a los otros sitios de replicación.

Las vistas materializas actualizables proveen los siguientes beneficios:

- Permiten a los usuarios consultar y actualizar datos locales replicados aun cuando está desconectado del master site o del sitio master de las vistas materializas.
- Requiere menos recursos que una replicación multimaster, mientras todavía está soportando la actualización de datos. Las vistas materializadas pueden reducir la cantidad de carga en los recursos de red porque las vistas materializas pueden ser actualizadas según la demanda, mientras que la

⁷ *Diseño y Construcción de Bases de datos Distribuidas. LAURA, M. (2010). Madrid: madrid.Pag.232*

replicación multimaster propaga los cambios a intervalos regulares. Además, debido a que las vistas materializadas pueden residir en una base de datos que contienen menos información, el espacio del disco y los requerimientos de la memoria para los clientes de vistas materializadas pueden ser menores que los requerimientos para un servidor de Oracle que contenga master sites. (Heurtel, 2009)

2.6.2.3 Vistas materializadas “writeable”

Se puede crear una vista materializada usando la cláusula FOR UPDATE durante la creación pero luego nunca agregar la vista materializada a un grupo de vista materializada.

En ese caso, los usuarios pueden mejorar el lenguaje de manipulación de datos (DML) en la vista materializada, pero esos cambios no pueden ser propagados al master y se pierden si la vista materializada es actualizada. Tales vistas materializadas son llamadas vistas materializadas “writeable”. (Heurtel, 2009)

2.6.2.4 Subconjuntos de Filas y Columnas con vistas materializadas

Ambos, subconjuntos de filas y columnas dan la posibilidad a crear vistas materializadas que contienen una copia parcial de los datos en una tabla master o master de vistas materializadas. Tales vistas materializadas pueden ser útiles para oficinas que no requieran el conjunto de datos completo. (Heurtel, 2009)

2.6.2.5 Actualización de vistas materializadas

Para asegurarse que una vista materializada es consistente con su tabla master o su master de vista materializada, se necesita actualizar periódicamente la vista materializada. Oracle provee los siguientes tres métodos para actualizar las vistas materializadas:

- a) Refresco rápido que utiliza logs de vistas materializadas para actualizar solo las filas que han sido cambiadas desde la última actualización.
- b) Refresco completo que actualiza toda la vista materializada.
- c) Refresco forzado ejecuta una actualización rápida cuando es posible. Cuando una rápida actualización no es posible, el refresco forzado ejecuta un refresco completo. (Heurtel, 2009)

2.6.2.6 Grupos de refresco

Cuando es importante para las vistas materializadas ser transaccionalmente consistentes con las otras, se pueden organizar en grupos de refresco.

Actualizando el grupo de refresco, se puede asegurar que los datos en todas las vistas materializadas del grupo de refresco corresponden a la misma transacción consistente en el tiempo. Ambos, vistas materializadas de sólo lectura y las actualizables, pueden ser incluidas en un grupo de refresco. Una vista materializada en un grupo de refresco puede ser actualizada individualmente, pero haciendo eso, se anula los beneficios del grupo de refresco, porque actualizando las vistas materializadas individualmente no se actualizan las otras vistas materializadas del grupo de refresco. (Luis, 2006)

2.6.2.7 Registro de vistas materializadas

Un registro de vista materializada es una tabla en el master site o en el master site de vistas materializadas que guarda todos los cambios de DML en la tabla master o en el master de vistas materializadas. Un registro de vista materializada es asociado con una tabla master simple o master de vistas materializadas, y cada uno de esos tiene solo un registro de vista materializada. Un refresco rápido de una vista materializada es posible solo si el master de vistas materializadas tiene un registro de vistas. Cuando una vista materializada es refrescada con el método rápido, las entradas de las vistas materializadas asociadas al registro de vistas materializadas que han aparecido desde la última vez que fue actualizado son aplicadas a la vista materializada. (Luis, 2006)⁸

2.6.3 Configuraciones Híbridas: Multimaster y vistas materializadas

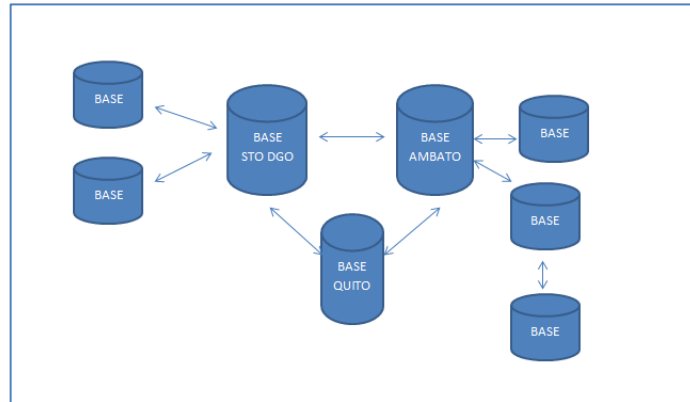
La replicación multimaster y materialized views pueden ser combinadas en configuraciones híbridas o mixtas para reunir los distintos requerimientos de aplicación. Configuraciones híbridas pueden tener cualquier número de master sites y múltiples sitios de vistas materializadas para cada master.

Replicación multimaster entre dos masters puede soportar replicación de tablas completas, entre las bases de datos que dan soporte a dos regiones geográficas.

⁸ *Bases de Datos Distribuidas. Luis, P. M. (2006). Barcelona Esp.Pag.235*

Las vistas materializadas pueden ser definidas en los masters para replicar tablas completas o subconjuntos de tablas a sitios dentro de cada región.

La siguiente figura muestra un ambiente de replicación híbrido. (Luis, 2006)



Fuente: Walter Padilla.

Figura 9: Ambiente de Replicación Híbrido

Diferencias clave entre las vistas materializadas y tablas master replicadas:

- Las tablas maestras replicadas deben contener datos para la tabla completa que está siendo replicada, mientras que las vistas materializadas pueden replicar subconjuntos de la tabla master de datos.
- La replicación Multimaster permite replicar cambios de cada transacción mientras los cambios ocurren. Las actualizaciones de vistas materializadas son orientadas a conjuntos, propagando cambios desde múltiples transacciones en una operación más eficiente, orientadas a lotes, pero con intervalos menos frecuentes. (Luis, 2006)

Si ocurren conflictos debido a los cambios hechos a copias múltiples de los mismos datos, entonces la detección y resolución de conflictos siempre ocurre en un master site o en un master site de vista materializada.

2.6.4 Conflictos de Replicación

Los ambientes de replicación asincrónicos multimaster y de vistas materializadas actualizables deben tener la posibilidad de tener conflictos de replicación que puedan ocurrir cuando, por ejemplo, dos transacciones originadas desde diferentes

sitios actualizan la misma fila casi al mismo tiempo. Cuando ocurren los conflictos de datos, se necesita un mecanismo para asegurar que el conflicto sea resuelto de acuerdo con las reglas de negocio y para asegurar que la información converja correctamente en todos los sitios.

Además para terminar con cualquier conflicto que pudiera ocurrir en un ambiente de replicación, la replicación Oracle ofrece una variedad de métodos preconstruidos de resolución de problemas que habilita la posibilidad de definir un sistema de resolución de conflictos para la base de datos que resuelve los conflictos de acuerdo con las reglas del negocio. Si se tiene una situación única que los métodos preconstruidos de resolución de problemas no pueden resolver, entonces se tiene la opción de construir y utilizar métodos propios de resolución de conflictos. (Engeleiter, 2009)

2.6.5 Detección de Conflictos y Replicación Procedural

Cuando un dato replicado utiliza replicación procedural, los procedimientos que replican los datos son responsables de asegurar la integridad del dato replicado.

Estos se deben diseñar procedimientos tanto para evitar o detectar conflictos de replicación y para resolverlos apropiadamente. En consecuencia, la replicación procedural es principalmente utilizada cuando las bases de datos son modificadas solo en grandes operaciones por lote. En tales situaciones, los conflictos de replicación son poco probables ya que numerosas transacciones no disputan el mismo dato. (Heurtel, 2009)

2.7 Replicación multimaster

2.7.1 Conceptos de replicación multimaster

Para entender los detalles arquitectónicos de la replicación maestra, es necesario entender los conceptos de la replicación maestra. Ya vimos anteriormente conceptos de replicación maestra, pero en este apartado se pretende profundizar un poco más. Conocer cómo y por qué la replicación es usada proveerá de una comprensión mucho mayor de cómo los elementos arquitectónicos individuales trabajan juntos para crear un ambiente de replicación multimaster.

- **Replicación multimaster:** La replicación multimaster también conocida “peer-to-peer” o replicación en caminos está compuesta de múltiples sitios

maestros que participan igualmente en un modelo que soporta actualizaciones desde cualquier sitio. Las actualizaciones que se realicen a un sitio individual son propagadas a todos los demás sitios maestros que participan.

Los servidores de bases de datos Oracle que operan como sitios maestros en un ambiente de replicación multimaster trabajan automáticamente para uniformizar los datos de todas las réplicas de tablas y aseguran la consistencia global de las transacciones y la integridad de los datos. La resolución de conflictos es manejada independientemente en cada uno de los sitios maestros. La replicación multimaster provee la replicación completa de cada tabla replicada en cada uno de los sitios maestro.

Si el ambiente de replicación es un ambiente híbrido (tienen muchos sitios maestros soportando una o más sitios de vistas materializadas) entonces el sitio maestro principal propaga cualquiera de las actualizaciones realizadas sobre la vista materializada a todos los otros sitios maestros en el ambiente de replicación multimaster. Después, cada uno de los sitios maestro propaga los cambios a sus vistas materializadas durante el refresco de ellas. (Engeleiter, 2009)

La siguiente figura muestra un posible ambiente de replicación multimaster:

- **Replicación de un solo maestro:** Un sitio maestro único puede también funcionar como el sitio maestro principal para uno o más sitios de vistas materializadas. Contrariamente a la replicación multimaster donde las actualizaciones son propagadas a todos los demás sitios, las actualizaciones de las vistas materializadas son realizadas solo en el sitio maestro principal.
- **Sitios maestros:** Un sitio maestro puede ser un nodo en un ambiente de replicación multimaster y ser el maestro para uno o más sitios de vistas materializadas en un ambiente de replicación de maestro único o multimaster. Los objetos replicados son guardados en el sitio maestro y están disponibles para ser accedidos por los usuarios.
- **Sitio maestro de definición:** En un ambiente de replicación multimaster, un sitio maestro opera como el sitio maestro de definición para un grupo

maestro. Este sitio particular realiza muchas de las tareas de mantenimiento y administrativas para el ambiente de replicación multimaster. Cada grupo maestro puede tener un único sitio de definición maestro, aunque el sitio de definición maestro puede ser cualquiera de los sitios maestros en el ambiente multimaster. Tradicionalmente el sitio de definición maestro puede ser cambiado a otro sitio maestro si es necesario. Un sitio maestro único soportando replicación de vistas materializadas es por defecto el sitio de definición maestro. (Engeleiter, 2009)

2.7.1.1 Por qué utilizar replicación multimaster

Desde un punto de vista muy básico, la replicación es usada para asegurar que los datos estén disponibles cuando y donde se los necesite. El siguiente apartado describe varios ambientes diferentes que tienen necesidades de disponibilidad distintos. Un ambiente de replicación puede tener uno o más de los siguientes requerimientos:

- **Tolerancia a fallos:** La replicación multimaster puede ser utilizada para proteger la disponibilidad de una base de datos crítica. Por ejemplo, un ambiente de replicación multimaster puede replicar los datos en su base de datos para establecer un sitio tolerante a fallos si el sitio primario deja de estar disponible debido a problemas en el sistema o en la red. Un sitio tolerante a fallos de este tipo puede servir también como una base de datos totalmente funcional para soportar el acceso de aplicaciones cuando el sitio primario opera concurrentemente. Es posible usar también Oracle net para configurar el tiempo de conexión automática cuando ocurre un fallo, lo cual le permite a Oracle net cambiar a un sitio maestro diferente si el primer sitio maestro falla.
- **Balanceo de carga:** La replicación multimaster es útil para las aplicaciones de procesamiento de transacciones que requieren múltiples puntos de acceso a información de bases de datos para los siguientes propósitos:
 - Distribuir la carga de una aplicación pesada
 - Asegurar disponibilidad continua
 - Proveer acceso localizado a los datos

2.7.1.2 Proceso de replicación multimaster

Como vimos anteriormente hay dos tipos de replicación multimaster, asincrónica y sincrónica.

- **Replicación asincrónica:** La replicación asincrónica propaga independientemente cualquier DML o ejecución de un procedimiento replicado a todos los otros sitios maestro que participan en el ambiente de replicación multimaster.

La propagación ocurre en una transacción separada, después que el DML o el procedimiento de replicación hayan sido ejecutados localmente.

La replicación asincrónica es el modo por defecto para la replicación debido a que requiere menos recursos de red y hardware que la replicación sincrónica, resultando una mejor disponibilidad y fiabilidad.

La replicación asincrónica sin embargo significa que los conjuntos de datos de los distintos sitios maestros del ambiente de replicación pueden ser diferentes durante un periodo de tiempo hasta que los cambio hayan sido propagados.

También pueden ocurrir conflictos de datos en un ambiente de replicación sincrónica.

A continuación se describe el proceso de replicación asincrónica:

- Un usuario publica una sentencia de DML o ejecuta un wrapper para un procedimiento replicado. Después que una tabla haya sido acondicionada para replicación, cualquier DML que el usuario realice un commit sobre la tabla es capturada para replicación en todos los sitios maestros.

Para cada fila que es insertada, actualizada o borrada, un gatillo crea una llamada diferida a procedimiento remoto (RPC) y la coloca en la cola de transacciones diferidas. La cola de transacciones diferidas contiene todas las RPCs diferidas.

Si un procedimiento ha sido replicado y su wrapper es ejecutado en un sitio maestro entonces la llamada al procedimiento es colocada en la cola de transacciones diferidas.

- La cola de transacciones diferidas guarda RPCs diferidas.

Cada transacción en la cola de transacciones diferidas tiene una lista de destinatarios que define donde debe ser propagada la transacción diferida; esta lista contiene todos los sitios maestros excepto el sitio original. Existe una cola de transacciones diferidas para cada sitio, y esta cola puede ser usada en grupos de replicación múltiple.

- La propagación envía transacciones diferidas a su destino a intervalos preestablecidos o bajo demanda. Las transacciones diferidas en la cola de transacciones diferidas son propagadas a sus destinos objetivos. Cada destino puede tener diferentes intervalos.
- La transacción diferida es aplicada en el destino remoto mientras que una transacción diferida está siendo propagada a su destino objetivo, cada RPC diferida es aplicada en el sitio destino llamando a un paquete interno. Si la transacción diferida no puede ser aplicada exitosamente en el sitio destino, entonces es reenviada y colocada en la cola de errores en el sitio destino, donde el DBA puede revisar y reparar los errores y reaplicar la transacción diferida.

Cuando una transacción diferida es aplicada en el sitio destino, Oracle chequea posibles conflictos de datos. Si un conflicto es detectado, entonces es registrado en el sitio remoto y, opcionalmente, un método de resolución de conflictos es invocado.

- Cuando una transacción diferida ha sido exitosamente propagada a todos los sitios maestros, no es purgada de la cola de transacciones diferidas del sitio original inmediatamente. Esta podría ser purgada más tarde por un trabajo de purgación el cual corre a intervalos definidos por el usuario. (Engeleiter, 2009)
- **Replicación sincrónica:** La replicación sincrónica propaga cualquier cambio hecho en el sitio local a todos los otros sitios maestros conectados sincrónicamente en un mismo ambiente de replicación durante la misma transacción que el cambio inicial. Si la propagación falla en cualquiera de los sitios maestro, entonces la transacción entera, incluyendo el cambio

inicial en el sitio maestro local es deshecha. Esta estricta medida asegura la consistencia de datos a lo largo del ambiente de replicación.

Contrariamente a la replicación asincrónica, no existe un periodo de tiempo en el cual los datos de los sitios maestros no coincidan. La replicación sincrónica también asegura que no se introduzcan conflictos de datos en el ambiente de replicación. Estos beneficios tienen el costo de requerir mucho hardware y recursos de red sin flexibilidad de tiempos. Por ejemplo, si un único sistema maestro en un ambiente mutimaster no está disponible, entonces una transacción no puede ser completada en ningún sitio maestro. Sin embargo, en la replicación asincrónica, la transacción diferida es mantenida en el sitio original hasta que el sitio caído se vuelva disponible. Mientras tanto, la transacción puede ser exitosamente propagada y aplicada a los otros sitios de replicación. (Heurtel, 2009)

Adicionalmente mientras el desempeño de las consultas se mantiene alto, ya que son realizadas localmente con replicación sincrónica, las actualizaciones son más lentas debido al protocolo de dos fases que se utiliza para realizar los commits y que asegura que cualquier actualización sea exitosamente propagada y aplicada en los sitios remotos.

A continuación se describe proceso de replicación sincrónica:

- Un usuario publica una sentencia de DML o ejecuta un wrapper de un procedimiento replicado.

Después que una tabla haya sido acondicionada para replicación, cualquier DML sobre el cual el usuario realice commit y que tenga como objetivo dicha tabla es capturado para replicación en todos los otros sitios maestros de replicación. Si un procedimiento ha sido replicado y su wrapper es ejecutado en un sitio maestro, entonces el procedimiento llamado es capturado para replicación.

- El DML o la ejecución del wrapper es inmediatamente propagado a los sitios destino. El gatillo interno captura cualquier DML e inmediatamente propaga estas acciones a todos los otros sitios maestros del ambiente de replicación. El gatillo interno aplica estas acciones en el contexto seguro del enlace de la base de datos del

propagador y usa un RPC interno para aplicar estas acciones en el sitio destino.

Como un gatillo interno, un wrapper propaga inmediatamente al procedimiento replicado, la llamada al procedimiento a todos los otros sitios maestros del ambiente de replicación.

Si la transacción falla en cualquiera de los sitios maestros de replicación entonces la transacción es desecha en todos los sitios maestros. Esta metodología asegura la consistencia de datos en todos los sitios maestros de replicación. Debido a la necesidad de deshacer una transacción si cualquier sitio falla, la replicación sincrónica es extremadamente dependiente de redes de alta disponibilidad, bases de datos de alta disponibilidad y hardware asociado. (Engeleiter, 2009)

2.7.2 Arquitectura de la replicación Multimaster

Este apartado describe la arquitectura de la replicación en términos de mecanismos y procesos.

2.7.2.1 Mecanismos del sitio maestro

Para soportar un ambiente de replicación, Oracle utiliza los siguientes mecanismos en cada sitio maestro que participa, ya sea en ambientes de replicación multimaster o replicación master. Algunos de estos mecanismos son requeridos solo en circunstancias especiales.

2.7.2.2 Mecanismos administrativos

Muchos mecanismos son requeridos para manejar las tareas administrativas que son llevadas a cabo para soportar un ambiente de replicación. Estos mecanismos permiten encender y apagar un ambiente de replicación, así como monitorizar tareas administrativas que son generadas cuando se construye o modifica el ambiente de replicación.

Modos de operación de replicación

Hay tres modos de operación:

- **Normal:** Este modo permite que la replicación ocurra. El ambiente de replicación está corriendo en este modo. Cualquier transacción sobre un objeto replicado es permitida y es propagada apropiadamente.
- **Quiescing:** Es el modo para pasar del modo normal al modo quiesced. En este modo el usuario no está habilitado a ejecutar transacciones sobre objetos replicados, pero existen transacciones diferidas que son propagadas. Las consultas sobre tablas quiescing están permitidas. Cuando todas las transacciones diferidas fueron exitosamente propagadas a sus respectivos destinos, el ambiente de replicación precede al modo Quiesced.
- **Quiesced (Inactivo):** Puede ser considerado deshabilitado para el uso normal de la replicación. Es principalmente para propósitos administrativos (tales como agregar o eliminar objetos replicados). La replicación está detenida en este modo. (Luis, 2006)

2.8 Vistas Materializadas

2.8.1 Conceptos de vistas materializadas

Como se vio anteriormente, una vista materializada es una réplica de un master en un instante de tiempo. Estas vistas son actualizadas mediante procesos por lotes conocidos como refreshes.

Como varios de los conceptos de las vistas materializadas ya fueron explicados, vamos a ver algunas características extra que son de interés.

2.8.1.1 Por que usar vistas materializadas

Ya vimos anteriormente cuales eran las ventajas de la base de datos replicadas.

Las ventajas de las vistas materializadas son las siguientes:

- Reducir la carga de la red.
- Replicación de subconjuntos: Esto permite la replicación de datos pertinentes para el sitio en particular. Es decir que permite replicar el mínimo imprescindible.
- Permite computación off-line.
- Desarrollo en masa.

2.8.1.2 Vistas materializadas en múltiples capas

Oracle permite la creación de vistas materializadas, basadas en otras vistas materializadas.

Esto genera un ambiente de varias capas de vistas, una jerarquía. Las vistas en este tipo de ambientes pueden ser de escritura o sólo lectura.

A las vistas que actúan de master de otras vistas se les llama “master de vistas materializadas”. Cualquier sitio puede ser un master y cada master puede tener más de una vista dependiente.

2.8.1.3 Mecanismos del master site y del master site de vista materializada

- **Tabla maestra o vista materializada maestra:** La tabla maestra o la vista materializada maestra es la base de las vistas materializadas. La tabla maestra está ubicada en un sitio master mientras que la vista materializada está ubicada en un sitio master de vistas materializadas, es decir en un nivel 1 como mínimo. Los cambios efectuados en la tabla maestra son propagados a las vistas materializadas durante el refresco.

2.8.1.4 Trigger interno para el log de vistas materializadas

Cuando se efectúan cambios en la tabla maestra o en el master de la vista materializada mediante DML, el trigger interno se encarga de registrar información sobre las filas afectadas en el log. Esta información incluye la clave primaria, el ROWID o object ID y otros valores. Este trigger es el último en activarse. (LAURA, 2010)

2.8.1.5 Log de vista materializada

Este log es necesario en un sitio master, si queremos poder efectuar el refresco rápido sobre vistas dependientes del master. Cuando se crea un log para una tabla maestra o Master de vista materializada, Oracle crea una tabla “oculta” que servirá de log.

2.8.1.6 Mecanismos de los sites de vistas materializadas

Cuando se crea una vista materializada, se crean ciertos mecanismos para soportar las vistas. Se crea una tabla base, por lo menos un índice, y posiblemente una vista. Si creamos una vista materializada modificable, entonces se crean también un trigger interno y un log local, en el sitio.

2.8.1.7 Tabla base y vista

La tabla base de una vista materializada tiene diferentes funciones dependiendo del nivel de compatibilidad definido (esto es para integrar bases de datos de distintas versiones). Para las versiones superiores a la 8.1.0, la tabla base es la propia vista materializada y no se requiere de la vista extra. En cambio para versiones inferiores, la tabla base es el “soporte” y la vista es la vista materializada.

2.8.1.8 Log de vista materializada actualizable

Este log es utilizado para saber que filas deben ser sobrescritas o eliminadas al efectuarse un refresco rápido. Si la vista es de sólo lectura entonces este log no es creado.

2.8.1.9 Trigger para el Log de vista materializada actualizable

Como en el caso del master, este trigger se encarga de registrar en la vista materializada actualizable los cambios efectuados mediante MDL. Si la vista es sólo lectura, entonces este trigger no es creado.

2.9 Conflictos

Ya hemos mencionado a lo largo del proyecto el problema de los conflictos. En este apartado veremos los tipos de conflictos junto con métodos para la prevención y solución de ellos.

Los conflictos aparecen en sistemas que permiten actualizaciones a los mismos datos en diferentes sitios. Lo primero es tratar de diseñar el sistema para que no ocurran conflictos. Hay casos en los que no es posible, resolver conflictos. (LAURA, 2010)

2.9.1 Tipos de conflictos

Se pueden encontrar los siguientes conflictos en un ambiente replicado.

- **Conflicto de actualización:** Estos conflictos ocurren por querer actualizar una entrada en distintos sitios al mismo tiempo.
- **Conflicto de unicidad:** Este ocurre cuando una actualización viola la restricción de unicidad de una columna. Por Ej. Si se introducen dos registros con la misma clave primaria en distintos sitios al mismo tiempo, se tiene un conflicto.
- **Conflicto de borrado:** Ocurre cuando un registro es borrado en un sitio y en otro sitio es borrado o actualizado al mismo tiempo. Ese registro ya no existe para ser borrado o actualizado. (LAURA, 2010)

2.9.2 Conflictos de datos y orden de transacción

En ambientes de replicación con 3 o más sitios maestros, puede ocurrir que un sitio esté inaccesible durante varias actualizaciones, pero la replicación de los datos continúa propagándose a lo largo del resto de master sites. Cuando la propagación continúe la actualización de los datos en el site que estaba inaccesible es posible que se realicen en distinto orden.

También pueden ocurrir errores de integridad referencial. Por ejemplo, si se creó un nuevo cliente en una base de datos y a su vez se le asignó su primera compra, puede que en la propagación llegue primero la transacción y luego la creación del cliente, lo cual es incorrecto por no existir un cliente para la compra en ese momento. (LAURA, 2010)

2.9.3 Detección de conflictos

2.9.3.1 Como lo hace Oracle

El sitio master receptor o el sitio de la vista materializada en cada sistema de propagación detecta los conflictos de actualización, de unicidad y eliminación de la siguiente manera:

- El sitio master receptor detecta un conflicto de actualización si hay alguna diferencia entre los valores anteriores y los valores actuales para una misma fila. Para esto el sitio que propaga el cambio debe suministrar información extra sobre los valores anteriores.

- Los errores de unicidad los detecta el sitio master receptor si viola una restricción de unicidad al hacer una actualización o una inserción de una fila replicada.
- Los conflictos de borrado son detectados si no se encuentra una fila para la sentencia de actualización o eliminación porque la clave primaria de la fila no existe. (Engeleiter, 2009)

2.9.3.2 Resolución de conflictos

Oracle provee con métodos de resolución de conflictos de actualización que aseguran la convergencia de los datos en todos los sitios. También provee métodos de resolución de conflictos de unicidad, pero éstos no aseguran la convergencia de datos.⁹

Por otro lado Oracle no provee métodos de resolución de conflictos de borrado ni conflictos de orden, pero permite al usuario crear sus propios métodos de acuerdo a las necesidades del negocio.

Cualquier tipo de método de resolución de conflictos es aplicado en el momento de la detección del mismo. En el caso en que el conflicto no sea posible resolverlo, lo logea en la cola de errores. Incluso es posible la definición de varios métodos de resolución para un mismo conflicto, en el caso en que uno falle, se aplica el segundo método. (Heurtel, 2009)

2.9.4 Técnicas para evitar conflictos

2.9.4.1 Utilizar grupos de columnas

Al definir grupos de columnas, cada grupo es analizado de manera independiente al otro. Por lo tanto si por ejemplo ocurren actualizaciones simultáneas en columnas pertenecientes a diferentes grupos, no se produce un conflicto. La idea es que el grupo debe ser definido de manera lógica, agrupando las columnas lógicamente relacionadas entre sí.

2.9.4.2 Sitio primario y sitio dinámico propiedad de modelos de datos

⁹ Engeleiter, K. (2009). *Oracle Database*. Engeleiter, K. (2009). Redwood Shores, CA 94065 EEUU: Oracle Corporation. Pag.286

Una forma de evitar conflictos de replicación es limitando el número de sitios con la capacidad de actualizar tablas simultáneamente. Para esto se tienen dos modelos diferentes:

- **Primary:** Este modelo evita todo tipo de conflictos debido a que sólo un servidor permite la actualización de datos.
- **Dinamic:** Este modelo es menos restrictivo que el anterior. En este caso la capacidad de actualizar datos replicados se traslada de sitio en sitio, aun conservando la no-simultaneidad de las actualizaciones.

2.9.4.3 Evitando conflictos de unicidad

En estos casos se pueden generar conjuntos disjuntos de números de secuencia en cada sitio para usarlos de clave primaria. Esto puede tornarse complicado si aumentan los sitios. Otra opción es agregar un prefijo que identifique al sitio como parte de una clave compuesta. Finalmente se puede utilizar la función SYS_GUID para generar números únicos globalmente. (Heurtel, 2009)

2.9.5 Arquitectura de resolución de conflictos

2.9.5.1 Mecanismos de soporte

El mecanismo más importante es el de “grupos de columnas”. Esto es así porque el grupo de columnas es la base de la detección y resolución de conflictos de actualización. Por otro lado la cola de errores es importante para llevar registros de los conflictos ocurridos.

- **Grupos de columnas:** Un grupo de columnas es una agrupación lógica de columnas en una tabla replicada. Tienen que tener las siguientes propiedades:
 - Una columna puede pertenecer a sólo un grupo.
 - Un grupo puede tener varias columnas de una tabla.
 - La resolución de conflictos sólo es aplicable a columnas en un grupo de columnas.

La definición de grupos de columnas es importante para mantener la integridad de los datos y permite además aplicar distintas reglas a cada grupo. Hay que elegir de forma inteligente como agrupar las columnas de una tabla. Si dos columnas o más

deben permanecer consistentes consigo mismas, entonces es conveniente ponerlas en el mismo grupo. (LAURA, 2010)

- **Cola de error:** Si un método de resolución de conflictos falla o si no se ha definido ningún método para el conflicto encontrado, la cola de errores registra información sobre el conflicto.

2.9.5.2 Métodos de resolución de conflictos

- **Última marca de tiempo:** Este método resuelve el conflicto basado en la última actualización ocurrida, identificando la última actualización ocurrida mediante una marca de tiempo.

Este método asegura la convergencia de datos en una ambiente de múltiples masters y múltiples vistas materializadas.

Hay que tener en cuenta los husos horarios, por lo que es conveniente utilizar el GMT.

- **Sobrescritura:** Este método reemplaza los datos en el sitio receptor con los datos recibidos desde el sitio emisor, por lo cual no puede asegurar la convergencia de los datos en ambientes multimaster. Este método es para ser usado en ambientes con un único master y varias vistas materializadas. En definitiva lo que se hace al detectar un conflicto es utilizar el valor de la vista materializada que hizo la actualización más recientemente. (LAURA, 2010)

2.10 Desarrollo de replicación de la base de datos Oracle en la empresa Tevcol

2.10.1 Base De Datos Standby

Una base de datos Standby es una configuración de alta disponibilidad y contingencia relativa que respalda el nodo de producción primario de una instalación Oracle. En el caso de Bases de Datos Standard Edition este proceso se realiza manualmente mediante el modo de recuperación manual (manual recovery mode), ya que la versión de Oracle Estándar no soporta modo de recuperación administrado (managed recovery mode), la cual realiza todo el proceso automáticamente a través de Oracle Net. En estas bases de datos Standby también podremos realizar el reporting y los backups.

Incluso podemos usar la base de datos Standby como banco de pruebas, ya que con la función de *Flashback Database* podremos abrir la BBDD Standby en modo lectura-escritura, realizar los cambios y pruebas que queramos, y una vez finalizadas volver al estado anterior “rebobinando” la BBDD sin que pierda su función DR.

Una base de datos de reserva tiene los propósitos principales siguientes:

- Desastres protección
- Protección contra la corrupción de los datos
- Información suplementaria

Si la base de datos principal se destruye o se corrompe sus datos, puede realizar una conmutación por error a la base de datos standby, en cuyo caso la base de datos standby se convierte en la base de datos primaria nueva. También puede abrir una base de datos en espera con la opción de sólo lectura, lo que permite que funcione como una base de datos de información independiente.

Si bien esta alternativa es económica, se corre el riesgo de pérdida de datos en el caso de que una falla no permita acceder a las transacciones que en el nodo primario se encuentran en los online redolog y que no alcanzaron a ser archivadas en un archived redolog y luego transmitida al nodo standby. Cuando esta última situación no es tolerable por los SLA de la organización, entonces se debe recurrir a Oracle Dataguard. (Heurtel, 2009)

2.10.2 Opciones de configuración

Se puede configurar una base de datos en espera de varias maneras diferentes, dependiendo del método para:

2.10.3 La aplicación de registros de rehacer archivados a la base de datos standby

Por ejemplo, entorno gestionado de modo de espera de Oracle permite que la base de datos primaria para automáticamente los registros de archivado rehacer al sitio de base de datos en espera siempre que la instancia en espera se inicia. Si implementa un entorno de reserva no administrado, debe transferir los registros de

forma manual. Si la base de datos standby está en modo de recuperación administrado, la base de datos standby aplica automáticamente los registros recibidos de la base de datos primaria. También puede aplicar manualmente los registros a la base de datos en espera, colocándolo en el modo de recuperación manual. En cualquier momento se puede abrir la base de datos standby en modo de sólo lectura para los informes. (LAURA, 2010)

En la tabla siguiente se detallan las configuraciones posibles en función del ambiente que le conviene más:

Tabla 1: Configuraciones Posibles

AMBIENTE	METODO DE TRANSFERENCIA	MODOS DE BASE DE DATOS STANDBY	REQUISITOS DE RED
Gestionado	Automático (o manual si es necesario)	Recuperación gestionada, recuperación manual, o de solo lectura	Net8
No Gestionado	Solo manual	Recuperación manual, o solo de lectura	Ninguno

Fuente: LAURA, M. (2010).

2.10.4 Ventajas y Desventajas

Una base de datos standby puede ser una herramienta poderosa para la prevención de desastres y la información complementaria a la vez. Por ejemplo, usted puede:

1. Mantener una base de datos en espera en una ubicación que está geográficamente alejada de la base de datos principal, o mantener varias bases de datos en espera en diversas ubicaciones geográficamente.
2. Mantener las bases de datos principal y de reserva en unidades de disco diferentes de la misma máquina, de modo que si la unidad de la base de datos primaria falla, puede activar la base de datos en espera y reanudar las operaciones normales.

3. Implementar una configuración administrada de espera, por lo que una base de datos standby aplica automáticamente los registros de rehacer archivados que son automáticamente enviados al sitio en espera de una base de datos primaria. De esta manera, los cambios en una base de datos primaria se propagan con regularidad a una base de datos en espera.
4. Hacer una base de datos en espera de la nueva base de datos principal con una mínima pérdida de tiempo y de datos si la base de datos principal está completamente destruida.
5. Proporcionar protección posible contra los trabajos por lotes erróneos, errores de usuario (por ejemplo, truncando la mesa equivocada), o corrupciones de la aplicación en la base de datos principal al no aplicar registros archivados que contienen datos corruptos de la base de datos standby. A continuación, puede activar la base de datos standby incorrupto, por lo que es la base de datos primaria.

Mientras que una base de datos standby puede ser una enorme ventaja en su estrategia de backup y recuperación, se trata de los costos también. Por ejemplo, una base de datos en espera requiere:

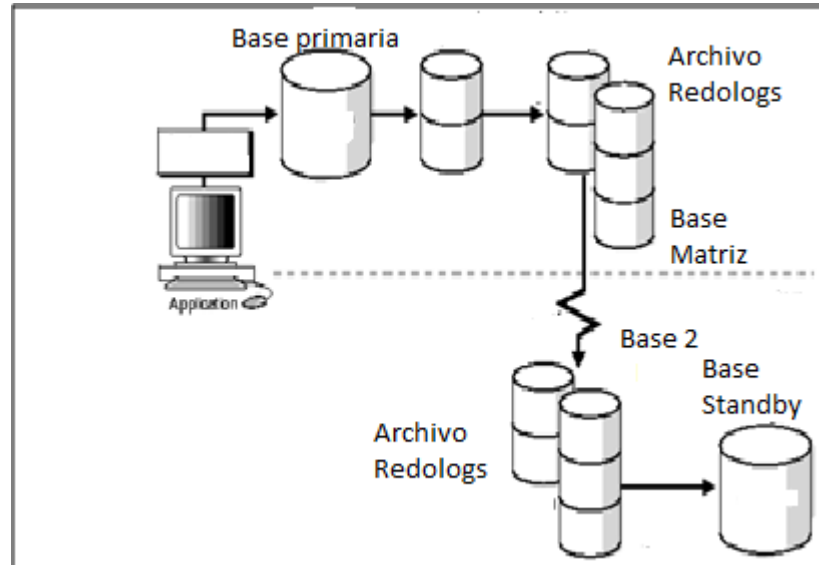
1. Un equipo adicional si se desea maximizar la prevención de desastres mediante el mantenimiento de una base de datos standby en máquinas separadas
2. Implementación y mantenimiento de una conexión Net8 si utiliza el entorno de reserva administrado
3. Recursos adicionales del sistema y el espacio de almacenamiento extra sin importar qué aplicación que usted elija
4. La administración de la base de datos en espera para reflejar algunas operaciones estructurales (por ejemplo, la adición de un espacio de tabla o archivo de datos) realizados en la base de datos primaria. (LAURA, 2010)¹⁰

2.10.5 Modos Standby Database

¹⁰ *Diseño y Construcción de Bases de datos Distribuidas. LAURA, M. (2010). Madrid: madrid.Pag.295*

Puede realizar cualquiera de las siguientes operaciones que se excluyen mutuamente en una base de datos en espera.

2.10.6 Modo de recuperación Manual



Fuente: LAURA, M. (2010).

Figura 10: Esquema de Modo de Recuperación Manual

Usted tiene la opción de colocar la base de datos en modo de recuperación manual, en cuyo caso habrá que continuamente y de forma manual transferir y aplicar los registros de rehacer archivados a la base de datos standby para mantenerlo sincronizado con la base de datos primaria. El Modo de recuperación manual es útil en entornos en los que no quieren conectar las bases de datos principal y de reserva a través de Net8. Además, si por alguna razón la base de datos principal no puede transferir automáticamente los registros de rehacer archivados a una base de datos en espera en un entorno de recuperación administrado, puede ser necesario realizar una recuperación manual para actualizar la base de datos Standby.

Podemos cambiar los roles de las BBDD en menos de 5 minutos y, con solo unos clicks de ratón o una orden de línea de comandos, pasar la BBDD principal a Standby y la Standby a principal. Esto nos facilitara mucho la vida si, por ejemplo, queremos realizar tareas de mantenimiento en el servidor principal: solo

tenemos que cambiar los roles, realizar las modificaciones y volver a cambiar los roles una vez finalizadas (todo ello con un mínimo corte de servicio y sin perder datos). Esta opción también permite intercambiar los roles cada cierto tiempo para “asegurar” que nuestra solución DR funciona correctamente.

Cabe destacar que podemos mantener diferentes niveles de seguridad/prestaciones en una Standby y cambian de uno a otro nivel bajo demanda. Desde un modo en que se favorezca la BBDD principal, enviando los cambios de manera asíncrona a la Standby evitando que se penalice su rendimiento en lo más mínimo, hasta un modo en que prime la seguridad, no devolviendo el “*commit completed*” al usuario hasta que el cambio se haya almacenado tanto en la principal como en la Standby.

Finalmente, comentar que todo esto lo podemos configurar desde *Oracle Grid Control* o por línea de comandos mediante la utilidad *DGMGRL*, el propio software de BBDD incorpora el llamado *Data Guard Broker* que vigilará y mantendrá la BBDD Standby por nosotros en todo momento.

Podemos crear una BBDD Standby, preparar un sistema de scripts que envíe los redo logs archivados de la primaria a la Standby, y finalmente otro que los aplique. En el fondo lo que haremos será una especie de “recover” continuado de la BBDD.

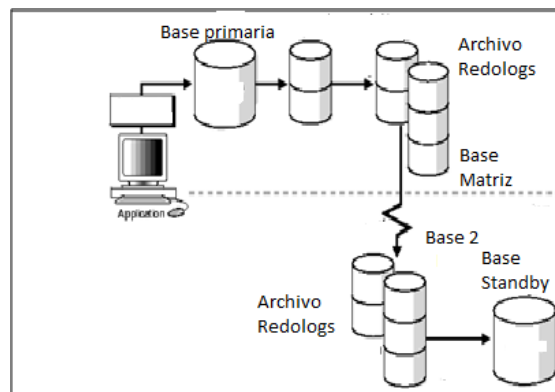
De nuestra pericia con los scripts dependerá el grado de automatización y funcionalidades que lleguemos a tener (hemos realizado algunas instalaciones de este tipo en las que se ha llegado a simular el “aplicado retardado” que nos ofrece Data Guard por ejemplo). No podremos abrir en lectura-escritura al no disponer de *Flashback Database* (solo disponible con *Enterprise Edition*), pero si podremos abrir la Standby en solo lectura, realizar consultas y volverla a activar posteriormente (todo ello mediante comandos SQL).

Otro inconveniente es que no disponemos del “Real Time Apply” traspaso síncrono de cambios de manera que en caso de problemas es probable que perdamos algunos datos (pero la cantidad de datos perdidos es controlable por la frecuencia de enviado de redos que decidamos configurar).

A nivel de licenciamiento rigen las mismas consideraciones que ya hemos comentado en el punto anterior, se deberá disponer de licencias para ambos gestores.

No tendremos todas las ventajas de una Standby “real” pero, dependiendo del caso, puede cubrir nuestras necesidades de DR, y todo ello con BBDD *Standard Edition*. (LAURA, 2010)¹¹

2.10.7 Modo de recuperación Gestionado



Fuente: Engeleiter, K. (2009).

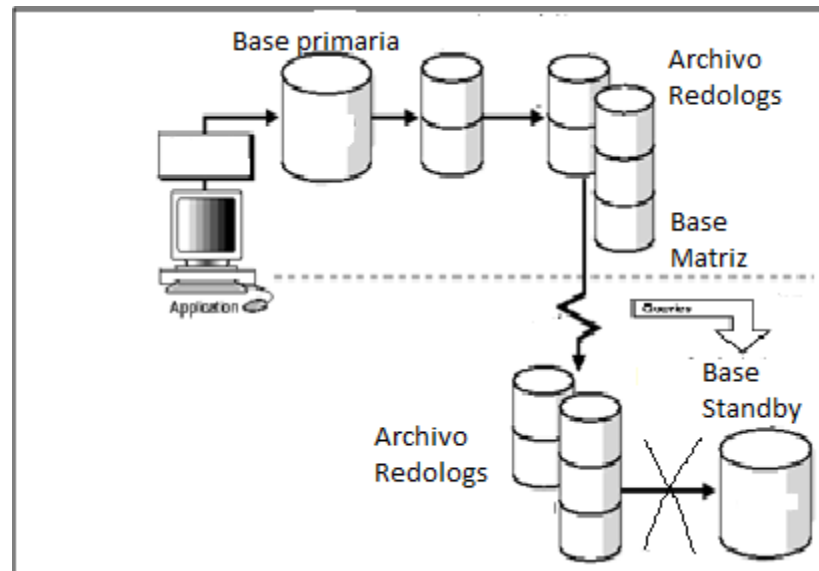
Figura 11: Esquemas de Modo de Recuperación Gestionado

Puede colocar la base de datos standby en modo de recuperación administrado, en cuyo caso la base de datos standby aplica automáticamente los registros de rehacer archivados, ya que los recibe de la base de datos primaria. Para iniciar la recuperación administrada, debe conectarse a la base de datos standby utilizando SQL * Plus y emitir la sentencia ordenar la base de datos standby en el modo de recuperación gestionado.

La principal ventaja de correr una base de datos en modo de recuperación administrada es que usted no tiene que transferir o aplicar los registros de rehacer archivados manualmente: Oracle automatiza el procedimiento. (Engeleiter, 2009)

¹¹ Engeleiter, K. (2009). *Oracle Database. Redwood Shores, CA 94065 EEUU: Oracle Corporation. Pag.313*

2.10.8 Modo de sólo lectura



Fuente: Engeleiter, K. (2009).

Figura 12: Modo de Solo Lectura

También puede abrir la base de datos standby en modo de sólo lectura después de terminar la recuperación manual o administrada. A continuación, puede consultar la base de datos e incluso almacenar datos en los espacios de tablas temporales (siempre que ya existen en la base de datos standby) sin afectar a los archivos de datos o registros de rehacer. Puede devolver la base de datos standby al modo de recuperación manual o administrada en cualquier momento, sin tener que apagarlo.

2.10.9 Replicación en una Base de Datos Standby (Manual)

La replicación es el proceso de utilización de un conjunto de bases de datos suministrados de paquetes PL / SQL y procedimientos, junto con un conjunto de herramientas de utilidades de red suministrada para permitir el intercambio de objetos de bases de datos y datos entre bases de datos. Para mantener replicados objetos de base y los datos entre múltiples bases de datos, un cambio en uno de estos objetos de base de datos en una base de datos se comparte con otras bases de datos relacionadas. De este modo, los objetos de la base de datos y los datos están sincronizados dentro de todas las bases de datos en el entorno de replicación.

Con la replicación, los datos se pueden compartir y replicar entre bases de datos que le permite poner los datos en los que se necesita.

Usted desea aumentar su estrategia de copia de seguridad: En conjunto con la copia de seguridad de rutina, la replicación proporciona niveles alternativos de protección de datos y la integridad, y reducir al mínimo las interrupciones de usuario. (Luis, 2006)

2.10.11 Visión general de la replicación

La replicación es el proceso de copiar y mantener objetos de la base, como tablas, en varias bases de datos que componen un sistema de base de datos distribuida. Cambios aplicados a un sitio son capturados y almacenados localmente antes de ser enviados y aplicados a cada uno de los lugares remotos. La Replicación avanzada es un sistema totalmente integrado característica del servidor Oracle, no es un servidor independiente.

La réplica utiliza la tecnología de base de datos distribuida para compartir datos entre múltiples sitios, pero una base de datos replicada y una base de datos distribuida no son los mismos. En una base de datos distribuida, los datos están disponibles en muchos lugares, pero una tabla en particular reside en una sola ubicación. Por ejemplo, la tabla de empleados sólo reside en la ny.world base de datos en un sistema de base de datos distribuido que también incluye la hk.world la.world y bases de datos. Replicación significa que los mismos datos es disponible en varias ubicaciones. Por ejemplo, la tabla de empleados está disponible en ny.world, hk.world, y la.world. Algunas de las razones más comunes para el uso de la replicación se describen como sigue: (Luis, 2006)

2.10.12 Disponibilidad

La replicación proporciona un acceso rápido y local a los datos compartidos, ya que equilibra la actividad en múltiples sitios. Algunos usuarios pueden tener acceso a un servidor mientras otros usuarios acceder a diferentes servidores, reduciendo así la carga en todos los servidores. Además, los usuarios pueden acceder a los datos desde el sitio de replicación que tiene el costo más bajo de acceso, que es típicamente el sitio que está geográficamente más cerca de ellos.

2.10.13 Aplicaciones que utilizan la replicación

La replicación es compatible con una variedad de aplicaciones que a menudo tienen diferentes requisitos. Por ejemplo, la automatización de fuerza de ventas, servicio de campo, comercio minorista, y otras aplicaciones de despliegue masivo típicamente requieren que los datos de forma periódica sincronizada entre los sistemas de bases de datos centrales y un gran número de pequeños, sitios remotos, que están a menudo desconectados de la base de datos central. Los miembros de una fuerza de ventas debe ser capaz de completar las transacciones, independientemente de si están conectados a la base de datos central. En este caso, los sitios remotos deben ser autónomos.

Por otra parte, las aplicaciones tales como centros de llamadas y los sistemas de Internet requieren datos en varios servidores para sincronizar de manera continua, casi instantánea manera de garantizar que el servicio proporcionado está disponible y equivalente en todo momento.

Por ejemplo, un sitio web de venta al por menor a través de Internet deben asegurarse de que los clientes ven la misma información en el catálogo en línea en cada sitio. Aquí, la coherencia de datos es más importante que la autonomía sitio. (Engeleiter, 2009)

2.10.14 Objetos de replicación

Un objeto de replicación es un objeto de replicación de bases de datos existentes en varios servidores de un sistema distribuido de bases de datos. En un entorno de replicación, todas las actualizaciones hechas a un objeto de replicación en un único sitio se aplican a las copias en todos los otros sitios.

La replicación avanzada le permite reproducir los siguientes tipos de objetos:

- Tablas
- Índices
- Vistas y vistas de objetos
- Los bultos y cuerpos de paquete
- Procedimientos y funciones
- Tipos definidos por el usuario y Cuerpos de tipo
- Disparadores
- Sinónimos
- Indextypes
- Los operadores definidos por el usuario

En cuanto a las tablas, la replicación soporta características avanzadas tales como tablas particionadas, organizador índice-tablas, tablas que contienen columnas que están basadas en el usuario define tipos y tablas de objetos.

Si se tiene una base de datos de única instancia de Oracle para configurar la replicación entre dos sitios, puede utilizar Storage Foundation for Databases (SFDB) herramientas para clonar la base de datos standby para análisis y ensayos. En esta configuración, Storage Foundation está instalado en ambos sitios y Oracle Data Guard se utiliza para la replicación entre los dos sitios. Su equipo de análisis debe generar informes sobre una base diaria para acciones de seguimiento e inventario. Mientras su equipo de análisis es la generación de los informes, se debe mantener la replicación en asegurar delta mínima entre los dos sitios.

Data Guard cuenta con el conocimiento necesario de la base de datos Oracle para proteger de forma fiable la base de datos standby de corrupciones que tratan de propagar a partir de la base de datos primaria. Es fácil de implementar y administrar.

También permite a las bases de datos en espera, tanto físicos como lógicos, que se utilizarán para la producción, mientras que en papel blanco (pruebas, copias de seguridad, actualización). (Luis, 2006)

2.11 Análisis de diferentes técnicas de replicación

A continuación analizaremos la diferentes técnicas estudiadas para definir la más se adecua a la empresa TEVCOL.

Tabla 2. Resumen técnicas replicación

TECNICA	CARACTERISTICA	VENTAJA	DESVENTAJA
Replicación Multimaster	También conocida “peer-to-peer” o replicación en caminos está compuesta de múltiples sitios maestros	Múltiples maestros en varios sitios	Alto costo de la licencia Oracle
Replicación Vistas	Una vista materializada es una réplica de un	Vista materializada es	No se tiene los datos actualizados

Materializadas	master en un instante de tiempo	una réplica de un master en un instante de tiempo.	al mismo tiempo en todos los nodos.
Replicación Sincrónica	Propaga cualquier cambio hecho en el sitio local a todos los otros sitios maestros conectados sincrónicamente en un mismo ambiente de replicación durante la misma transacción que el cambio inicial	Existe sincronización de los datos en un mismo tiempo y todos los datos están actualizados en un mismo tiempo.	Para la aplicación se necesita una versión del Oracle 11G.
Replicación Base de Datos Standby	Una base de datos Standby es una configuración de alta disponibilidad y contingencia relativa que respalda el nodo de producción primario de una instalación Oracle.	Existen datos actualizados en un mismo instante tanto en el servidor principal como en el replicado.	NO admite interfaz gráfica y el costo de las licencias para la actualización es demasiado alto.

Fuente: Walter Padilla

Luego de analizado las técnicas antes descritas se optó para aplicar en la replicación la técnica de base de datos standby debido a que la empresa cuenta con el Oracle 10G Standar debido a costos de licencias no se cuenta con un manejador de base de datos más avanzado.

CAPITULO III

METODOLOGIA

3.1 Método

Se utilizó el Método de Investigación Científica, ya que fue el más óptimo por cuanto utiliza conocimiento probado en anteriores investigaciones y pudo ser utilizado en la investigación para aplicar conocimientos y teorías ya elaborados para usarlos en la implementación de la Replicación del Sistema de Base de Datos bajo plataforma Linux en Oracle 10g Estándar en la empresa privada TEVCO y tratar de dar solución al problema planteado.

3.2 Tipo de Estudio

Investigación de Campo.- Se efectuó en el lugar y tiempo real en que ocurren los fenómenos objetos de estudios. Se refiere a un estudio en una situación real, que al efectuarse en condiciones no artificiales permite detectar mejor las posibles consecuencias de la investigación.

Investigación Aplicada.- El investigador busca resolver un problema conocido y encontrar respuestas a preguntas específicas. En otras palabras, el énfasis de la investigación aplicada es la resolución práctica de problemas, El objetivo es predecir un comportamiento específico en una configuración muy específica

3.3 Población y Muestra

Para la presente investigación no fue necesario la utilización de una muestra por cuanto el sistema no es de afectación directa al usuario final sino únicamente es de incumbencia del departamento técnico de sistemas el mismo que se encuentra en la Matriz ubicada en la ciudad de Quito.

3.4 Procedimientos

3.4.1 Fuentes de Información

Entre las fuentes de información consta la Primaria y Secundaria

- a) **Primarias.-** Esta información se obtuvo a través de la Observación y conversación con el gerente nacional de sistemas de la empresa, usuarios que utilizan la base de datos.

- b) **Secundarias.**- Las fuentes secundarias se obtuvo de folletos, libros, documentos y bibliotecas virtuales.

3.4.2 Procesamiento y Análisis

3.4.2.1 Teoría fundamentada en datos

La teoría fundamentada en datos es un método de investigación cualitativa que ayuda en la colecta, análisis sistemático de datos y en la generación de la teoría.

En el desarrollo de esta investigación este método se utilizó para precisar la colecta y el análisis general de la información pertinente a la implementación de la réplica en sistemas de bases de datos.

3.4.2.2 Análisis de tareas

En este proceso se describen las tareas realizadas actualmente por los usuarios, sus patrones definidos de flujo de trabajo, los cuales se originan de sus esquemas mentales y las necesidades de información para realizar su trabajo. Es decir, se procura identificar “qué el usuario hace”, “de qué manera lo hace”, y “qué necesita para hacerlo”. De esa manera, se logra el entendimiento conceptual de las tareas que deberán formar parte del sistema en desarrollo. Para la obtención de dicho entendimiento se pueden utilizar varias técnicas tales como, observación sistemática, etc.

CAPITULO IV

RESULTADOS

4.1 Análisis de resultados

Se ha desarrollado una replicación en el sistema de la base de datos la misma que permitirá a los usuarios manejar la información de la empresa de forma más fluida sin caídas ni cuellos de botella en el sistema, para que dicha información sea confiable para los usuarios.

La replicación es una aplicación realizada en una plataforma de Linux Red Hat 5 y testeado para sistemas operativos Windows para versiones de Xp en adelante, está desarrollado con herramientas bajo licenciamiento de Oracle Estándar Edition 10g software que no soporta orientación a objetos en esta versión.

Con la implementación de la réplica antes mencionada el usuario podrá manipular los datos de forma más fluida debido a que los datos se encuentran replicados y si tenemos algún problema con la base principal la base de datos replicada tomará su lugar, este proceso se realiza de forma transparente para el usuario.

4.2 Elección de herramienta de Estudio

Después del estudio de las herramientas de Linux Red Hot 5.0 se han seleccionado la herramienta ORACLE después de la toma de decisiones por la gerencia, por cuestiones de mayor eficiencia en el desempeño para le replica se procedió a elegir al manejador de base de datos Oracle.

4.3 Metodologías de Estimación

4.3.1 Análisis de Puntos Función

El Análisis de Punto Función es una técnica que, mediante la descomposición de un sistema en componentes más pequeños, permite que éstos puedan ser mejor comprendidos y analizados en forma individual.

El Análisis de Punto Función se basa en la teoría de que las funciones de una aplicación son la mejor medida del tamaño de un sistema. El Punto Función mide el software mediante la cuantificación de la funcionalidad que el sistema le brinda al usuario basado fundamentalmente en el diseño lógico.

4.4 Metodología escogido para la estimación de costos

En el caso de la estimación de costos se escogió como metodología COCOMO II, aunque esta es un tanto complicada, debido a la utilización de varias fórmulas que estiman el costo de un proyecto, cuenta con la ventaja de usar para su estimación un amplio dominio de factores que inciden sobre el costo del proyecto, tales como: retrasos en el cronograma, pérdida de personal, etc.

4.5 Comprobación de Hipótesis

4.5.1 Hipótesis

El uso de técnicas de replicación incide en el desempeño de la base de datos de la empresa TEVCOL.

4.5.2 Operacionalización de las variables

Con la replicación de la base de datos dependiendo de la técnica escogida se mejorarán los siguientes aspectos.

- Disponibilidad de la aplicación y de los datos
- Rendimiento (tiempo acceso)
- Computación desconectada (sincronización del servidor con otros equipos que poseen una réplica completa o parcial de las tablas)
- Reducción de Carga en la Red

Tabla 3: Operacionalidad de variables

VARIABLE	TIPO	DEFINICION CONCEPTUAL	INDICADORES
Técnicas de replicación.	Independiente	Técnicas científicas y analíticas que permiten identificar, preservar, analizar y presentar datos.	<p>El modo en que la replicación incrementa la disponibilidad de los datos para los usuarios y aplicaciones.</p> <p>Al haber múltiples copias de los datos disponibles en el sistema, se dispone de un mecanismo excelente de recuperación cuando existan fallos en nodos</p> <p>Estándares ISO.</p> <p>Protección y control de acceso al sistema.</p> <p>Seguridad en los Servicios.</p>
Desempeño de la base de datos.	Dependiente	El desempeño de una base de datos está influenciado por el tiempo de ingreso de datos, su procesamiento y emisión de resultados, en base a lo cual existen criterios de disponibilidad, rendimiento, carga en la red, entre otros que permiten determinar su alto o bajo rendimiento.	<p>Se mejora las transacciones de consulta cuando se introduce la replicación en un sistema que estuviera aquejado de sobrecarga de recursos centralizados.</p> <p>Modo en que se utiliza la replicación para distribuir datos en ubicaciones remotas.</p>

4.5.3 Comprobación

Para la comprobación de la hipótesis descrita anteriormente se realizó una estimación en tiempos de eficiencia de procesos en transacción antes y después de la implementación de la Réplica de Sistemas de bases de datos, las mismas que arrojaron que se incrementaba la eficiencia en el rendimiento con la implementación de la réplica quedando demostrada nuestra hipótesis como afirmativa.

Para recalcar lo mencionado anterior se procedió a tomar en cuenta los tiempos del procesamiento de la información de una transacción sin la base replica así como con la base de datos ya replicada se tomó en cuenta también tiempos de procesamiento de información para extracción de backups sin replicación y con replicación

Para la comprobación se tomó en cuenta valores en horas de tiempos que se demora en realizar un backup de la base que no está replicada y con respecto cuando ya está replicada la base.

Definición de desempeño El desempeño de una base de datos está influenciado por el tiempo de ingreso de datos, su procesamiento y emisión de resultados, en base a lo cual existen criterios de disponibilidad, rendimiento, carga en la red, entre otros que permiten determinar su alto o bajo rendimiento.

Los valores medidos son

Hipótesis general:

El uso de técnicas de replicación incide en el desempeño de la base de datos de la empresa TEVCOL.

Hipótesis específica:

El uso de técnicas de replicación incide en el tiempo de ejecución de procesos de la base de datos de la empresa TEVCOL.

H0: $M_1 = M_2$ (media el grupo 1 es igual a la media del grupo 2, no existe diferencia entre las medias)

H1: $M_1 > M_2$ (media el grupo 1 es mayor a la media del grupo 2, hay diferencia entre las medias)

La Prueba z para medias de dos muestras se aplica con dos muestras conocidas, con varianzas conocidas.

Esta herramienta se utiliza para comprobar las hipótesis nulas relativas a que no existen diferencias entre dos medias de población frente a las hipótesis alternativas en uno u otro sentido.

La herramienta de Prueba z también puede utilizarse para el caso en que la hipótesis nula sea que existe un valor distinto de cero específico para la diferencia entre las dos medias de población. En este caso se ha aplicado esta prueba para determinar las diferencias entre el desempeño de la base de datos de TEVCOL sin replicación y con replicación.

En tal virtud los tiempos promedio totales luego de la medición y suma de tiempos individuales de cada proceso, son los siguientes:

Tabla 4. Descripción de los procesos

PROCESO	DEFINICION
CAPITAL	En este proceso se ingresa la información de contabilidad como son cuentas por cobrar, egresos, cheques ,cuentas por pagar etc
TRANSVAL	Se ingresa todo lo referente a facturación.
SIB	Se ingresa todo proceso referente a lo que es custodia dinero y custodia joyas.
SCPV	Guarda información de despachos de materiales de bodega.
REQ	Se procesa información concerniente a compras para la empresa.
ARCHIVO	Se procesa toda la información de la empresa

Tabla 5: Procesamiento de datos

No.	TIEMPO PROMEDIO (SIN REPLICACIÓN)	(X-X) ²	TIEMPO PROMEDIO (CON REPLICACIÓN)	(X-X) ²
	SR		CR	
1	6,4	0,04	3,7	1E-04
2	6,2	0,16	3,6	0,0121
3	5,9	0,49	3,2	0,2601
4	7,5	0,81	3,6	0,0121
5	7,1	0,25	3,9	0,0361
6	6,8	0,04	3,6	0,0121
7	6,5	0,01	3,7	1E-04
8	6,5	0,01	3,8	0,0081
9	6,6	0	3,7	1E-04
10	6,1	0,25	4,3	0,3481
MEDIA	6,6		3,71	
DESVIACIÓN ESTÁNDAR	0,48		0,28	
VARIANZA	0,23		0,08	

A simple vista se determina que el tiempo empleado por los procesos de la base de datos de TEVCOL es mayor sin el uso de técnicas de replicación, frente al uso de las técnicas de replicación.

Para determinar la prueba z se ha aplicado la siguiente fórmula:

$$Z_{prueba} = \frac{\bar{x}_1 - \bar{x}_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}}$$

Donde X1 y X2 son las medias del grupo 1 y 2 respectivamente.

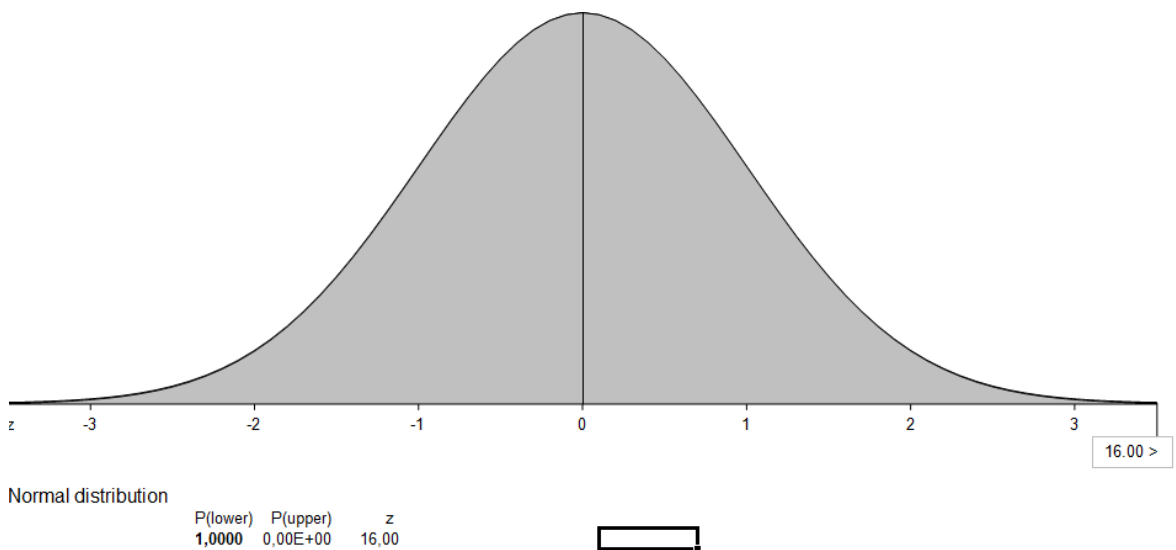
σ_1^2 y σ_2^2 es la desviación estándar.

n1 y n2 son el tamaño de las muestras.

Resultados:

	SR	CR
Media	6,56	3,71
Varianza (conocida)	0,23	0,08
Observaciones	10	10
Diferencia hipotética de las medias	0	
Z	16,2	
P(Z<=z) una cola	0,0	
Valor crítico de z (una cola)	1,6	
Valor crítico de z (dos colas)	0,0	
Valor crítico de z (dos colas)	2,0	

Se obtuvo el valor de $z=16,2$ y este es mayor que 2,3 (región crítica), por lo tanto se acepta la hipótesis de investigación y se rechaza la hipótesis nula.



CAPITULO V

PROPUESTA REPLICACION BASE DE DATOS TEVCOL

5.1 La Empresa TEVCOL

La empresa TEVCOL está ubicada en la ciudad de Quito con su base matriz y con diferentes oficinas a nivel nacional en 22 ciudades del Ecuador es una empresa solida brindando servicios de transporte de valores, custodia de joyas, custodia móvil, seguridad privada y vigilancia armada.

La problemática de TEVCOL parte tener una base de datos robusta para brindar un servicio eficiente a los usuarios ya que el factor tiempo es imprescindible para brindar un servicio de calidad a los usuarios, entonces la necesidad de implementar una replicación a la base de datos, para evitar caídas en el sistema cuellos de botellas, carga en la red.

La base de datos se encuentra ubicada en la Matriz de la empresa TEVCOL y a su vez la réplica en la Base 2 de Tevcol que se encuentra en la ciudad de quito pero en otro punto de la ciudad, se detalla en el mapa siguiente los puntos desde donde nomas se accede a la base principal.

De ahí la necesidad de implementar una replicación a la base de datos debido a que la empresa ha incrementado sus puestos de trabajo cada vez más en varias ciudades del país.

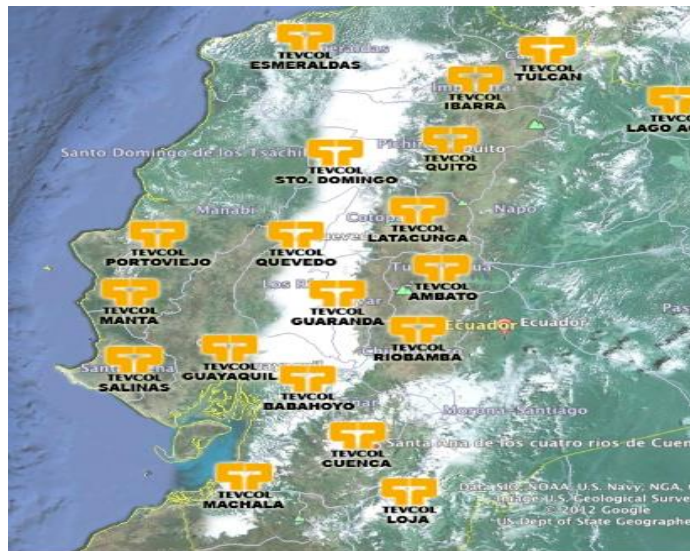
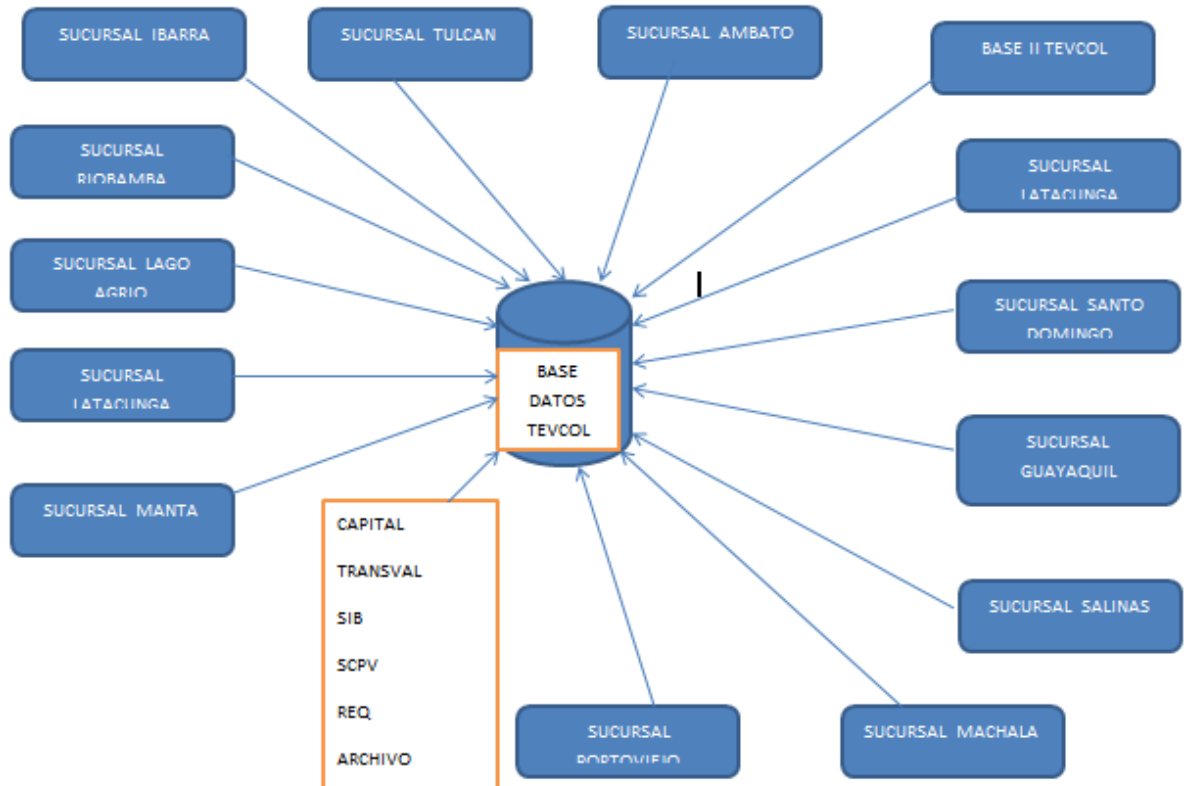


Figura 14. Ubicación puntos de TEVCOL

5.1.1 Entorno de la Base de Datos TEVCOL Sin Replicación

La base de datos TEVCOL sin replicación todos las sucursales acceden a un mismo punto esto se demuestra en el siguiente diagrama.

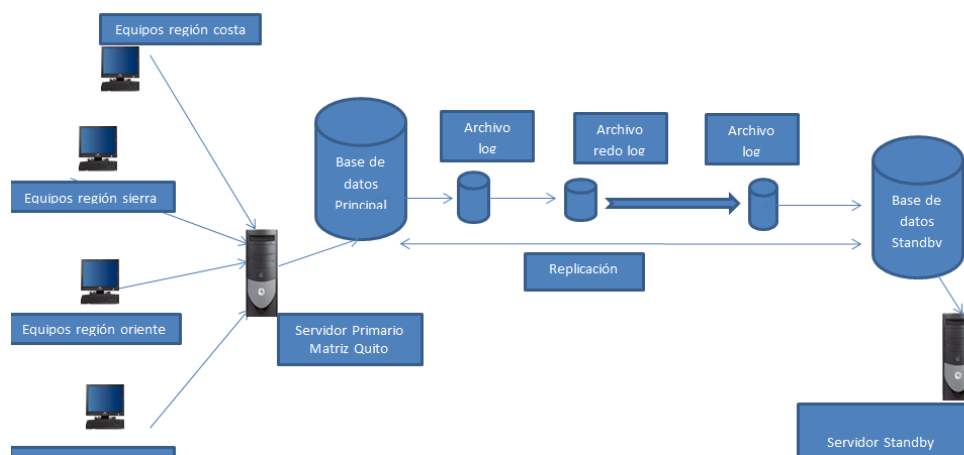


Fuente: Walter Padilla

Figura 15: Modo de Solo Lectura

5.1.2 Entorno de la Base de Datos de TEVCOL con Replicación

En este entorno se encuentra la base de datos sincronizado de tal manera que los usuarios ingresan a la base de datos principal y en el caso de que exista alguna caída de la base principal la base Standby toma el lugar de la principal sin causar efecto en el instante en el procesamiento de la información.



Fuente: Walter Padilla

Figura 16. Entorno con replicación

5.2 Instalación Oracle Database 10g Release 2 (10.2.0.1) Installation On Red Hat Enterprise Linux 5 (RHEL5) e implementación de la réplica.

5.2.1 Proceso Preliminar

A continuación se describirá la instalación de Oracle Database 10g Release 2 (10.2.0.1) en Red Hat Enterprise Linux 5 (RHEL5). La descripción se basa en una instalación de servidor similar a este, con un mínimo de intercambio 2G de swap Linux instalara los siguientes paquetes descritos a continuación:

- GNOME Desktop Environment
- Editors
- Graphical Internet
- Text-based Internet
- Development Libraries
- Development Tools
- Legacy Software Development
- Server Configuration Tools

- Administration Tools
- Base
- Legacy Software Support
- System Tools
- X Window System

Si se realiza instalaciones alternativas pueden requerir más paquetes a ser cargado, además de los enumerados a continuación:

- Download Software
- Unpack Files
- Hosts File
- Set Kernel Parameters
- Setup
- Installation
- Post Installation
- Se procede a descargar el software siguiente:
- Oracle Database 10g Release 2 (10.2.0.1) Software

El software descargado está en un archive zip se procede a descomprimir los archivos:

```
unzip 10201_database_linux32.zip
```

Ahora debe tener un solo directorio que contiene los archivos de instalación. Dependiendo el tamaño de la descarga de este bien puede ser llamado "db/Disk1" o "base de datos"

El archivo / etc / **hosts** debe contener un nombre completo para el servidor para nuestro caso se lo llamara REPLICA:

```
<IP-address> <fully-qualified-machine-name> <machine-name>
```

Se agrega las líneas siguientes al archivo / etc / sysctl.conf:

```
#kernel.shmall = 2097152
#kernel.shmmax = 2147483648
kernel.shmmni = 4096
# semaphores: semmsl, semmns, semopm, semmni
kernel.sem = 250 32000 100 128
#fs.file-max = 65536
net.ipv4.ip_local_port_range = 1024 65000
net.core.rmem_default=262144
net.core.rmem_max=262144
```

```
net.core.wmem_default=262144
net.core.wmem_max=262144
```

Para cambiar cualquiera de los parámetros del kernel se ejecuta el siguiente comando:

```
/sbin/sysctl -p
```

Agregue las líneas siguientes al archivo / etc / security / limits.conf:

```
*      soft  nproc  2047
*      hard  nproc  16384
*      soft  nofile 1024
*      hard  nofile 65536
```

Agregue la línea siguiente al archivo / etc / pam.d / login, si no existe:

```
session required /lib/security/pam_limits.so
```

Desactivar la seguridad de linux editando el archivo / etc / selinux / config, asegurándose de que la bandera SELINUX se establece de la siguiente manera:

```
SELINUX=disabled
```

Alternativamente, esta alteración se puede hacer usando la herramienta de interfaz gráfica de usuario (System > Administration > Security Level and Firewall).

En el cual se instala los siguientes paquetes:

```
# From RedHat AS5 Disk 1
cd /media/cdrom/Server
rpm -Uvh setarch-2*
rpm -Uvh make-3*
rpm -Uvh glibc-2*
rpm -Uvh libaio-0*
cd /
eject

# From RedHat AS5 Disk 2
cd /media/cdrom/Server
rpm -Uvh compat-libstdc++-33-3*
rpm -Uvh compat-gcc-34-3*
rpm -Uvh compat-gcc-34-c++-3*
rpm -Uvh gcc-4*
```

```
rpm -Uvh libXp-1*
cd /
eject

# From RedHat AS5 Disk 3
cd /media/cdrom/Server
rpm -Uvh openmotif-2*
rpm -Uvh compat-db-4*
cd /
eject
```

Se procede a crear los nuevos grupos y usuarios:

```
groupadd oinstall
groupadd dba
groupadd oper

useradd -g oinstall -G dba oracle
passwd oracle
```

En este estado de la instalación se crea los directorios donde se instala el software de Oracle:

```
mkdir -p /u01/app/oracle/product/10.2.0/db_1
chown -R oracle.oinstall /u01
```

Luego de la instalación se inicia una sesión como root y ejecute el siguiente comando:

```
xhost +<machine-name>
```

Edite el archivo / etc / redhat-release sustitución de la información de la versión actual (Red Hat Enterprise Linux Server versión 5) con el texto siguiente:

```
redhat-4
```

Se inicia la sesión como usuario oracle y añadir las siguientes líneas al final del archivo bash_profile.:

```
# Oracle Settings
TMP=/tmp; export TMP
TMPDIR=$TMP; export TMPDIR

ORACLE_BASE=/u01/app/oracle; export ORACLE_BASE
ORACLE_HOME=$ORACLE_BASE/product/10.2.0/db_1; export
ORACLE_HOME
```



```

ORACLE_SID=TSH1; export ORACLE_SID
ORACLE_TERM=xterm; export ORACLE_TERM
PATH=/usr/sbin:$PATH; export PATH
PATH=$ORACLE_HOME/bin:$PATH; export PATH

LD_LIBRARY_PATH=$ORACLE_HOME/lib:/lib:/usr/lib; export
LD_LIBRARY_PATH
CLASSPATH=$ORACLE_HOME/JRE:$ORACLE_HOME/jlib:$ORACLE_H
OME/rdbms/jlib; export CLASSPATH

if [ $USER = "oracle" ]; then
  if [ $SHELL = "/bin/ksh" ]; then
    ulimit -p 16384
    ulimit -n 65536
  else
    ulimit -u 16384 -n 65536
  fi
fi

```

Inicie sesión en el usuario oracle. Si está utilizando la emulación de X a continuación, establezca la variable DISPLAY del medio ambiente:

```
DISPLAY=<machine-name>:0.0; export DISPLAY
```

Inicie el Oracle Universal Installer (OUI) emitiendo el siguiente comando en el directorio de base de datos:

```
./runInstaller
```

Durante la instalación, escriba el nombre ORACLE_HOME apropiado y luego continúe con la instalación. Para una visión más detallada del proceso de instalación, haga clic en los enlaces de abajo para ver capturas de pantalla de cada etapa.

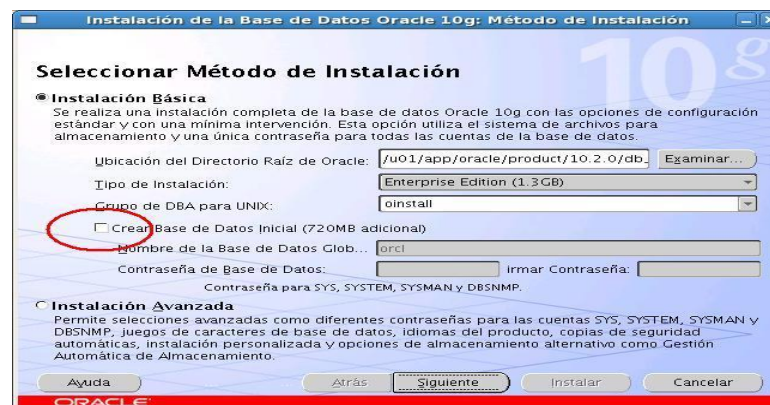


Figura 13: Pantalla inicio instalación base datos

Paso siguiente solo se pulsa siguiente



Figura 14: Establecer directorio donde se instala la base de datos

5.2.2 Comprobación de los requisitos

Debería emitir un sonido por la versión del sistema operativo y algunas veces por la red. Como el servidor esta con IP fija no debe haber complicación. Igual dele un vistazo a las demás comprobaciones, puede que te esté fallando otra cosa.



Figura 15: Pantalla de comprobación del sistema

Luego se muestra una pantalla con el resumen de lo que se va a instalar. Click en **INSTALAR**



Figura 16: Pantalla en la cual se muestra los recursos a ser instalados.

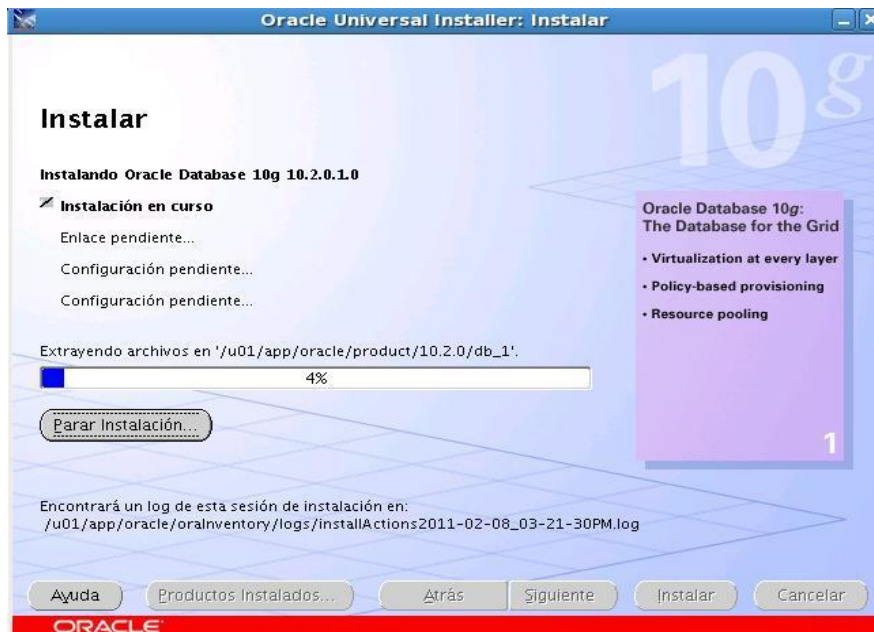


Figura 17: Proceso de la instalación.

Al final el instalador pedirá ejecutar un par de sentencias con la cuenta root. Abre un cliente ssh con la cuenta root y ejecútalas. La segunda sentencia pedirá confirmación de un path, no le pongas nada, sólo dale ENTER.



Figura 18: Confirmación del path

Con eso se terminó de instalar el software. Todavía no hay ninguna base de datos para trabajar.



Figura 19: Final instalación software base de datos

Se va a crear una base de datos. Si las variables de ambiente quedaron bien se ejecuta la shell para crear bases de datos, el Database Configuration Assistant, a.k.a. DBCA.

Ejecutar #dbca
y se abrirá una ventana de instalación inicial



Figura 20: Asistente para crear una base de datos

De ahora en adelante se puede poner screenshots sólo si lo considera muy necesario.

La opción siguiente ofrecerá creación de base de datos, plantillas y ASM. Se aplica la opción Crear Base de Datos.

La opción siguiente pone alternativas que son básicamente configuraciones de memoria inicial, los cuales puedes cambiarlos después. Para efectos prácticos usaremos Uso General.

Post Installation

Edit the /etc/redhat-release file restoring the original release information:

```
Red Hat Enterprise Linux Server release 5
```

Edit the /etc/oratab file setting the restart flag for each instance to 'Y':

```
TSH1:/u01/app/oracle/product/10.2.0/db_1:Y
```

For more information see:

5.3 Requerimientos para la replicación

Como requerimiento obligatorio para superar la contingencia debemos verificar que los siguientes puntos se cumplan:

- El enlace con Base2 deberá estar listo para soportar las conexiones a nivel nacional, es decir verificar que las rutas de todas las subredes de TEVCOLO estén habilitadas para dicho uso:

<i>RED</i>	<i>CIUDAD</i>
192.168.1.0	QUITO
192.168.3.0	AMBATO
192.168.4.0	BRIGHTSTAR
192.168.5.0	QUEVEDO
192.168.6.0	PORTOVIEJO
192.168.7.0	MANTA
192.168.8.0	IBARRA
192.168.9.0	STO. DOMINGO
192.168.11.0	BABAHOYO
192.168.12.0	ANGAR GYE
192.168.14.0	CELLISTIC
192.168.16.0	VIA DAULE
192.168.17.0	MACHALA
192.168.18.0	SALINAS
192.168.19.0	CALIFORNIA
192.168.20.0	ESMERALDAS

192.168.21.0	TULCAN
192.168.22.0	LATACUNGA
192.168.23.0	RIOBAMBA

Figura 30: Verificación subredes

- El Servidor de respaldo ubicado físicamente en Base 2 debe levantarse correctamente con la versión de la base de datos Oracle actual (V. 10.2.0.3.0)
- El tnsnames de todos los equipos que ingresan a los diferentes sistemas de la empresa deberán contener un en su configuración una entrada adicional que contendrá la conexión al servidor de respaldo y deberá estar nombrada como REPLICA, el personal de soporte deberá tomar en consideración esta configuración.
- Copiar los archivos de respaldo (.zip) que contienen a su vez los respaldos .dmp de las diferentes bases de datos de la Empresa ubicados en el equipo de respaldos con IP 192.168.1.245 en la ubicación C:\TCQ.

- SIB
- TRANSVAL
- RASTRAC
- LOGISTICA
- CAPITAL
- SCPV
- REQ
- ARCHIVO QUITO
- ARCHIVO AMBATO
- ATCHIVO GUAYAS

- ✓ El servidor deberá previamente tener definidos y configurados los tablespace de las bases de datos indicadas de la misma manera que se han configurado en el servidor de producción como se indica a continuación:

Tabla 3. Configuración de Tablespace

BASE DE DATOS	TABLESPACE	UBICACIÓN DEL ARCHIVO .DBF	TAM AÑO	AUTOEXTEND
SIB	SIB_DATA	/u01/app/oracle/product/10.2.0/db_1/dbs/SIB_DATA.DBF	6 GB	24 MB
TRANSVAL	TRANSVAL	/u01/app/oracle/product/10.2.0/db_1/dbs/TRANSVAL.DBF	8 GB	50 MB
RASTRAC	RASTRAC	/u01/app/oracle/product/10.2.0/db_1/dbs/RASTRAC.DBF	12 GB	64 MB
LOGISTICA	LOGISTICA	/u01/app/oracle/product/10.2.0/db_1/dbs/LOGISTICA.DBF	10 GB	50 MB
CAPITAL	INTER_DATA	/u01/app/oracle/product/10.2.0/db_1/dbs/INTER_D.DBF	10 GB	50 MB
SCPV	SCPV	/u01/app/oracle/product/10.2.0/db_1/dbs/SCPV.DBF	1 GB	5 MB
REQ	REQ	/u01/app/oracle/product/10.2.0/db_1/dbs/REQ.DBF	1 GB	5 MB
ARCHIVO UIO	ARCHIVO_DATA	/u01/app/oracle/product/10.2.0/db_1/dbs/ARCHIVO_DATA.DBF	200MB	64 MB
ARCHIVO AMB	ARCHIVO_AMB	/u01/app/oracle/product/10.2.0/db_1/dbs/ARCHIVO_AMB.DBF	200MB	64 MB
ARCHIVO GYE	ARCHIVO_GYE	/u01/app/oracle/product/10.2.0/db_1/dbs/ARCHIVO_GYE.DBF	150 MB	2 GB

- De igual manera los usuarios propietarios de los tablespaces deberán estar ya creados

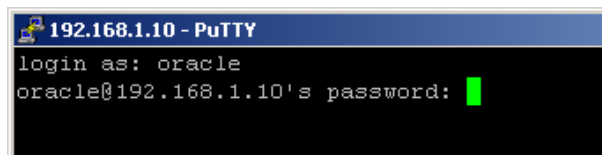
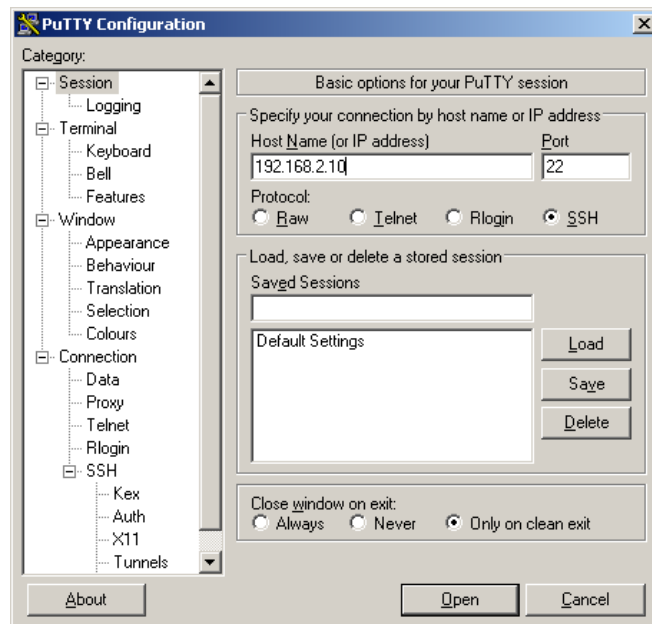
<i>BASE DE DATOS</i>	<i>USER</i>	<i>TABLESPACE</i>	<i>Temporary tablespace</i>	<i>ROLES</i>
SIB	SIB	SIB_DATA	TEMP	CONNECT,DBA Y RESOURCE
TRANSVAL	ATRANSVAL	TRANSVAL	TEMP	CONNECT,DBA Y RESOURCE
RASTRAC	RASTRAC	RASTRAC	TEMP	CONNECT,DBA Y RESOURCE
LOGISTICA	LOGISTICA1	LOGISTICA	TEMP	CONNECT,DBA Y RESOURCE
CAPITAL	INTER	INTER_DATA	TEMP	CONNECT,DBA Y RESOURCE
SCPV	SCPV	SCPV	TEMP	CONNECT,DBA Y RESOURCE
REQ	REQ	REQ	TEMP	CONNECT,DBA Y RESOURCE
ARCHIVO UIO	ARCHIVO	ARCHIVO_DATA	TEMP	CONNECT,DBA Y RESOURCE
ARCHIVO AMB	ARCHIVO_AMB	ARCHIVO_AMB	TEMP	CONNECT,DBA Y RESOURCE
ARCHIVO GYE	ARCHIVO_GYE	ARCHIVO_GYE	TEMP	CONNECT,DBA Y RESOURCE

5.4 Proceso de RECUPERACION

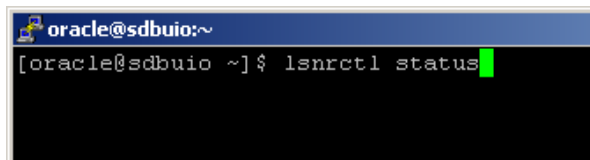
5.4.1 Preparación del servidor

Una vez que el servidor de contingencia ha sido encendido, es necesario realizar lo siguiente:

- Logearse con el usuario oracle ya que solo este podrá levantar los servicios de la base de datos desde un equipo remoto, podemos utilizar nuestro utilitario Putty



- Verificar el estatus del servicio de listener (por default el servicio deberá estar levantado), dicha verificación se muestra con el comando:



- Cuando el listener se encuentra levantado nos mostrará en pantalla el estado del servidor como se muestra a continuación

```

oracle@sdbuio:~
[oracle@sdbuio ~]$ lsnrctl status

LSNRCTL for Linux: Version 10.2.0.3.0 - Production on 20-NOV-2012 16:33:31

Copyright (c) 1991, 2006, Oracle. All rights reserved.

Connecting to (DESCRIPTION=(ADDRESS=(PROTOCOL=IPC)(KEY=EXTPROC)))
STATUS of the LISTENER
-----
Alias                     LISTENER
Version                   TNSLSNR for Linux: Version 10.2.0.3.0 - Production
Start Date                12-NOV-2012 10:38:02
Uptime                    8 days 5 hr. 55 min. 28 sec
Trace Level               off
Security                  ON: Local OS Authentication
SNMP                      OFF
Listener Parameter File  /u01/app/oracle/product/10.2.0/db_1/network/admin/listener.ora
Listener Log File        /u01/app/oracle/product/10.2.0/db_1/network/log/listener.log

Listening Endpoints Summary...
  (DESCRIPTION=(ADDRESS=(PROTOCOL=ipc)(KEY=EXTPROC)))
  (DESCRIPTION=(ADDRESS=(PROTOCOL=tcp)(HOST=sdbuio.tevcol.com.ec)(PORT=1521)))

Services Summary...
Service "GLTEVCOL" has 1 instance(s).
  Instance "GLTEVCOL", status READY, has 1 handler(s) for this service...
Service "GLTEVCOLXDB" has 1 instance(s).
  Instance "GLTEVCOL", status READY, has 1 handler(s) for this service...
Service "GLTEVCOL_XPT" has 1 instance(s).
  Instance "GLTEVCOL", status READY, has 1 handler(s) for this service...
Service "PLSExtProc" has 1 instance(s).
  Instance "PLSExtProc", status UNKNOWN, has 1 handler(s) for this service...
The command completed successfully

```

Levantar el servicio de listener

En el Shell de Linux una vez logeados digitaremos el siguiente comando:

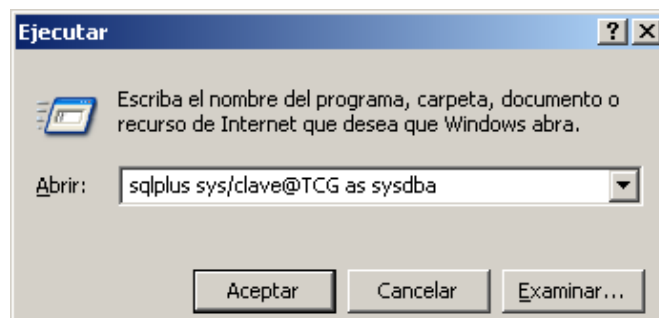
- En caso de que el servicio no esté levantado procederemos con el siguiente comando:

```

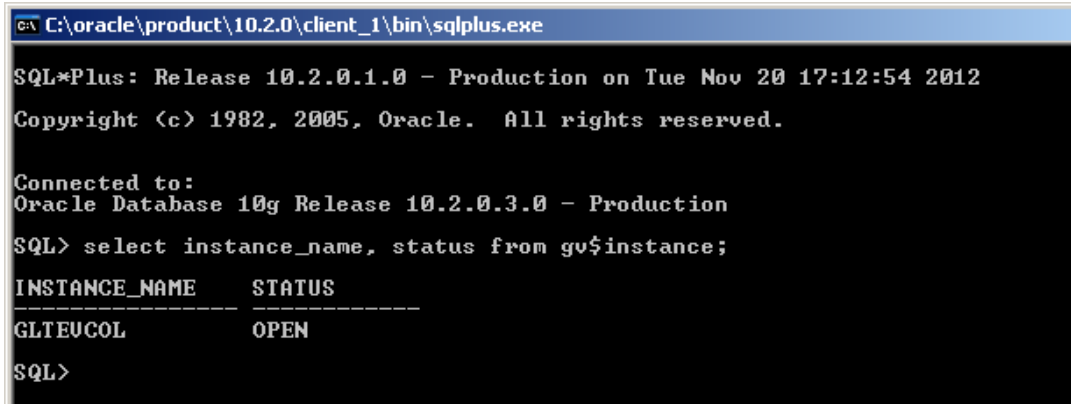
oracle@sdbuio:~
[oracle@sdbuio ~]$ lsnrctl start

```

- A continuación verificamos de igual manera si la base de datos está levantada



- Una vez que hemos ingresado al sqlplus podemos verificar el estatus de la instancia GLTEVCOL



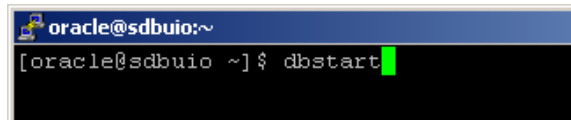
```
C:\oracle\product\10.2.0\client_1\bin\sqlplus.exe
SQL*Plus: Release 10.2.0.1.0 - Production on Tue Nov 20 17:12:54 2012
Copyright (c) 1982, 2005, Oracle. All rights reserved.

Connected to:
Oracle Database 10g Release 10.2.0.3.0 - Production
SQL> select instance_name, status from gv$instance;

INSTANCE_NAME      STATUS
-----
GLTEUCOL           OPEN

SQL>
```

- Si la instancia no estuviera levantada podemos hacerlo desde el Shell de Linux

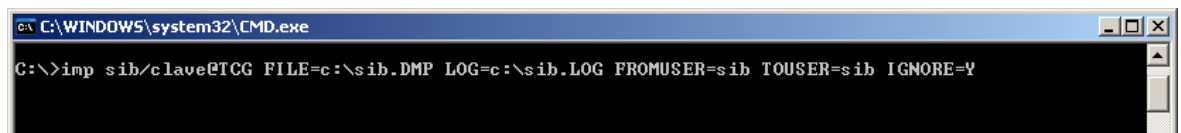


```
oracle@sdbuio:~
[oracle@sdbuio ~]$ dbstart
```

Una vez que los respaldos han sido copiados a un PC se procedera a desempaquetar lo mismo para poder proceder con la carga de información a la base de datos, para ello desde el editor de comandos de Windows haremos lo siguiente:

Data de SIB

```
imp sib/clave@TCG FILE=c:\sib.DMP LOG=c:\sib.LOG FROMUSER=sib TOUSER=sib IGNORE=Y
```



```
C:\WINDOWS\system32\CMD.exe
C:\>imp sib/clave@TCG FILE=c:\sib.DMP LOG=c:\sib.LOG FROMUSER=sib TOUSER=sib IGNORE=Y
```

Data de TRANSVAL

```
imp atransval/clave@TCG FILE=c:\transval.DMP LOG=c:\transval.LOG  
FROMUSER=transval TOUSER=transval IGNORE=Y
```

Data de RASTRAC

```
imp rastrac/clave@TCG FILE=c:\rastrac.DMP LOG=c:\rastrac.LOG  
FROMUSER=rastrac TOUSER=rastrac IGNORE=Y
```

Data de LOGISTICA

```
imp logistica1/clave@TCG FILE=c:\logistica.DMP LOG=c:\logistica.LOG  
FROMUSER=logistica1 TOUSER=logistica1 IGNORE=Y
```

Data de CAPITAL

```
imp inter/clave@TCG FILE=c:\capital.DMP LOG=c:\capital.LOG  
FROMUSER=inter TOUSER=inter IGNORE=Y
```

Data de SCPV

```
imp scpv/clave@TCG FILE=c:\scpv.DMP LOG=c:\scpv.LOG FROMUSER=scpv  
TOUSER=scpv IGNORE=Y
```

Data de REQ

```
imp req/clave@TCG FILE=c:\req.DMP LOG=c:\req.LOG FROMUSER=req  
TOUSER=req IGNORE=Y
```

Data de ARCHIVO QUITO

```
imp archivo/clave@TCG FILE=c:\archivo.DMP LOG=c:\archivo.LOG  
FROMUSER=archivo TOUSER=archivo IGNORE=Y
```

Data de ARCHIVO AMBATO

```
imp archive_amb/clave@TCG FILE=c:\archivo_amb.DMP  
LOG=c:\archivo_amb.LOG FROMUSER=archive_amb TOUSER=archive_amb  
IGNORE=Y
```

Data de ARCHIVO GUAYAS

```
imp archive_gye/clave@TCG FILE=c:\archivo_gye.DMP LOG=c:\archivo_gye.LOG  
FROMUSER=archive_gye TOUSER=archive_gye IGNORE=Y
```

5.4.2 Verificación de logs

En cuanto la data termine de subir es necesario chequear los archivos log que se crean en el equipo desde el cual se subieron los dmp, verificando las novedades o inconsistencia que pudieran presentarse en la migración de datos.

Cada usuario genera un archivo log, por lo tanto deberemos verificar:

- ✓ Sib.log
- ✓ Transval.log
- ✓ Rastrac.log
- ✓ Logistica.log
- ✓ Capital.log
- ✓ Scpv.log
- ✓ Req.log
- ✓ Archivo.log
- ✓ Archivo_amb.log
- ✓ Archivo_gye.log

Es necesario realizar este proceso puesto que ahí podemos evidenciar triggers,SP, Package y Vistas que suban sin compilarse, de ser así procederemos a chequear cada uno de estos objetos de la base de datos.

5.4.3 Subir scripts de usuarios, dblinks , Jobs y Sequences de la base de datos

Finalmente es necesario ejecutar los scripts que se encuentran al igual que los respaldos como archivos. sql, pueden ser ejecutados desde el TOAD o desde el sqlplus, los archivo son los siguientes:

- ✓ Usuarios.sql
- ✓ Dblinks,sql
- ✓ Jobs.sql

La parte de Sequences es vital realizar una verificación exhaustiva de dichos secuenciales a fin de mantener consistencia en la data respaldada.

```
CREAR USUARIO GLTEVCOL
IDENTIFICADO POR <contraseña>
Streams_tbs tabla predeterminado
Temp TABLESPACE TEMPORAL
CUOTAS SIN LÍMITES EN streams_tbs;
```

```

GRANT CONNECT, RESOURCE, DBA
A GLTEVCOL;
conectar / as sysdba
EMPEZAR
DBMS_STREAMS_AUTH.GRANT_ADMIN_PRIVILEGE (
  concesionario => 'GLTEVCOL',
  grant_privileges => true);
END;

```

Esto se hace sólo de origen (si se trata de una forma de replicación)

```

ALTER DATABASE AGREGAR DATOS DE REGISTRO COMPLEMENTARIOS
(ALL) COLUMNS;

```

subvención DBA, IMP_FULL_DATABASE, EXP_FULL_DATABASE a
GLTEVCOL;

Conectado con el usuario corrientes en ambas bases de datos, crear las colas:

```

comenzar
DBMS_STREAMS_ADM.SET_UP_QUEUE (queue_name => 'streams_queue');

```

- Si el supuesto previsto en el nodo que va a ser el primario

```

/* DBMS_AQADM.ALTER_QUEUE_TABLE (

  queue_table => 'streams_queue_table',
  primary_instance => 2,
  secondary_instance => 1);
*/

```

Importar una base de datos :

- Crear tablas si no existen en la base destino

- Aquí estoy usando algunos espacios de tabla de demostración (utilizar su propia tabla que necesita ser movido

```

crear tablas WS_DATA datafile '/ oracle/oradata/destdb-
aix_ORCLPLY1/datafile/destdb-aix/ws_data.dbf' tamaño 4120 M AUTOEXTEND
EN SIGUIENTE MAXSIZE 10240K 32767M;

```

```

crear tablas WS_DATA_LARGE datafile '/ oracle/oradata/destdb-
aix_ORCLPLY1/datafile/destdb-aix/ws_data_large.dbf' talla M AUTOEXTEND EN

```

```

SIGUIENTE MAXSIZE 10240K 32767M 2000;
crear tablas WS_IDX datafile '/ oracle/oradata/destdb-
aix_ORCLPLY1/datafile/destdb-aix/ws_idx.dbf' tamaño 3500 M AUTOEXTEND
EN SIGUIENTE MAXSIZE 10240K 32767M;
crear tablas WS_IDX_LARGE datafile '/ oracle/oradata/destdb-
aix_ORCLPLY1/datafile/destdb-aix/ws_idx_large.dbf' tamaño 1300 M
AUTOEXTEND EN SIGUIENTE MAXSIZE 10240K 32767M;
crear tablas temporales temp2 tempfile '/ oracle/oradata/destdb-
aix_ORCLPLY1/datafile/destdb-aix/temp2.dbf' 1000m AUTOEXTEND EN
SIGUIENTE MAXSIZE 10240K 3276M;

```

```

impdp gltevcoll / strmadminpw parfile = <location> / impdestdb-aix.dat

```

contenido de imp.dat:

```

DIRECTORIO = orclply1exp
TOTAL = Y
DUMPFIL = SourceDB-solaris_orclply1_20110113.dmp
logfile = imp_orclply1_20110113.log
paralelo = 4

```

Crear regla de propagación:

```

EMPEZAR
(DBMS_STREAMS_ADM.ADD_GLOBAL_PROPAGATION_RULES
streams_name => 'streams_propagation',
source_queue_name => 'strmadmin.streams_queue',
destination_queue_name => 'strmadmin.streams_queue @ fuente-db_streams',
include_dml => true,
include_ddl => true,
source_database => null,
inclusion_rule => true);
END;

```

Crear Aplicar reglas a puerta:

```

comenzar
(DBMS_STREAMS_ADM.ADD_GLOBAL_RULES
streams_type => 'Aplicar'
streams_name => 'streams_apply',

```



```

queue_name => 'streams_queue',
include_dml => TRUE,
include_ddl => true,
source_database => 'SourceDB-solaris',
inclusion_rule => TRUE);
fin;

```

Iniciar procesos de la fuente:

EMPEZAR

- Empezar captura

```

DBMS_CAPTURE_ADM.START_CAPTURE (
    capture_name => 'streams_capture');

```

- Propagación de inicio (después de la importación se realiza)

para rc en (seleccione propagation_name de dba_propagation)

bucle

```

DBMS_PROPAGATION_ADM.START_PROPAGATION (rc.propagation_name);

```

Terminar el ciclo;

```

END;

```

Crear el directorio

orclply1exp COMO

```

'/ Oracle/backup/destdb-aix_ORCLPLY1/export';

```

comenzar

```

(DBMS_STREAMS_ADM.ADD_GLOBAL_RULES

```

```

    streams_type => 'Aplicar'
    streams_name => 'streams_apply',
    queue_name => 'streams_queue',
    include_dml => TRUE,
    include_ddl => true,
    source_database => 'SourceDB-solaris',
    inclusion_rule => TRUE);

```

```

fin;

```

5.4.4 Checklist de Procesos de Recuperación

Finalmente en resumen los pasos a seguir:

<i>PROCESO</i>	<i>VERIFICACION</i>
Verificación de Enlace con base 2	✓
Habilitar subredes de TEVCOL	✓
Verificación de la versión de la BD Oracle	✓
Verificación de tnsnames en los equipos de los usuarios	✓
Copia de archivos de respaldo .dmp	✓
Verificación de tablespace	✓
Verificación de user DBA	✓
Levantar listener	✓
Levantar base de datos	✓
Subir data	✓
Verificación de logs	✓
Subir scripts de users, dblinks, jobs y sequences.	✓

Figura 31: Desarrollo de los procesos de replicación

5.5 Desarrollo de la replicación utilizando base de datos standby, como manejador de base código.

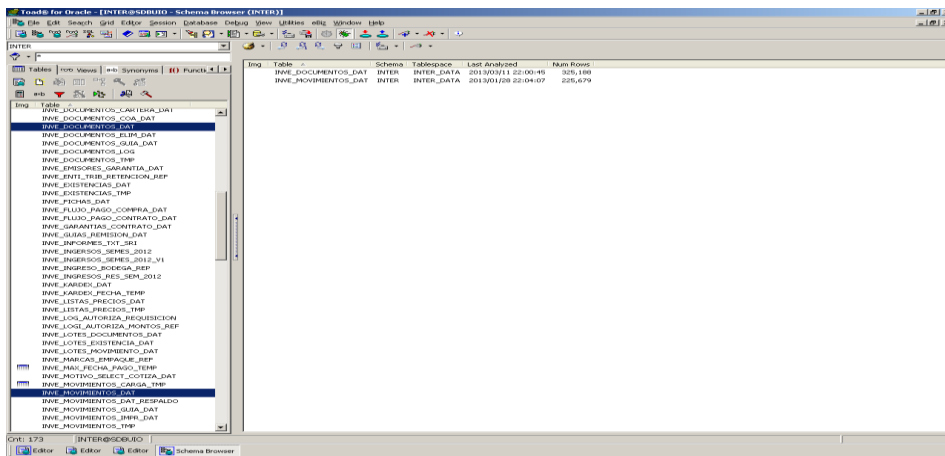


Figura 31: Manejador de base de datos

Para generar el código SQL realizamos el proceso como muestra en la pantalla.

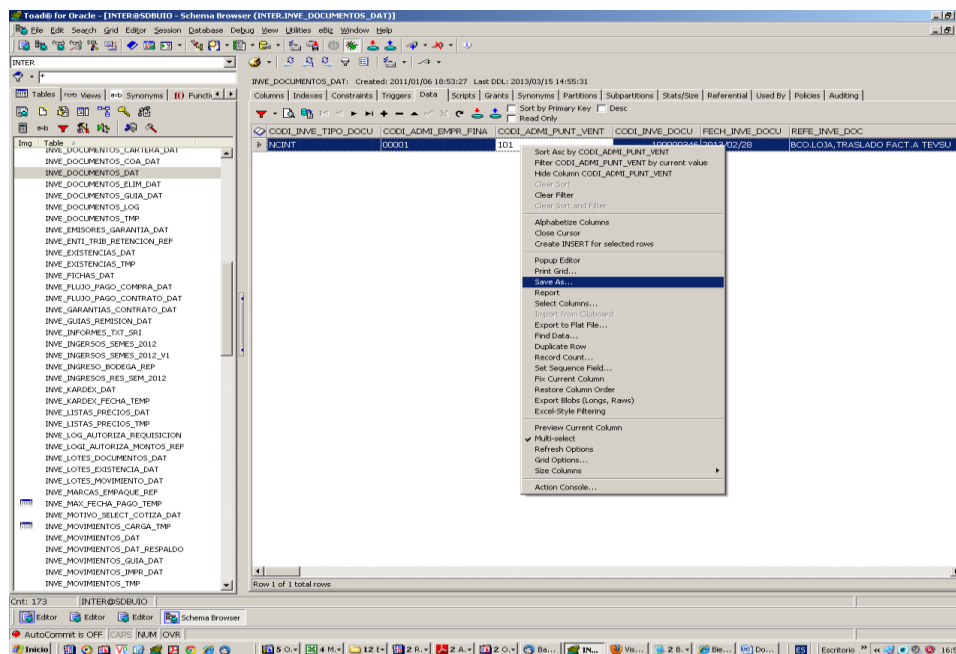


Figura 32: Pantalla para ingresar código SQL

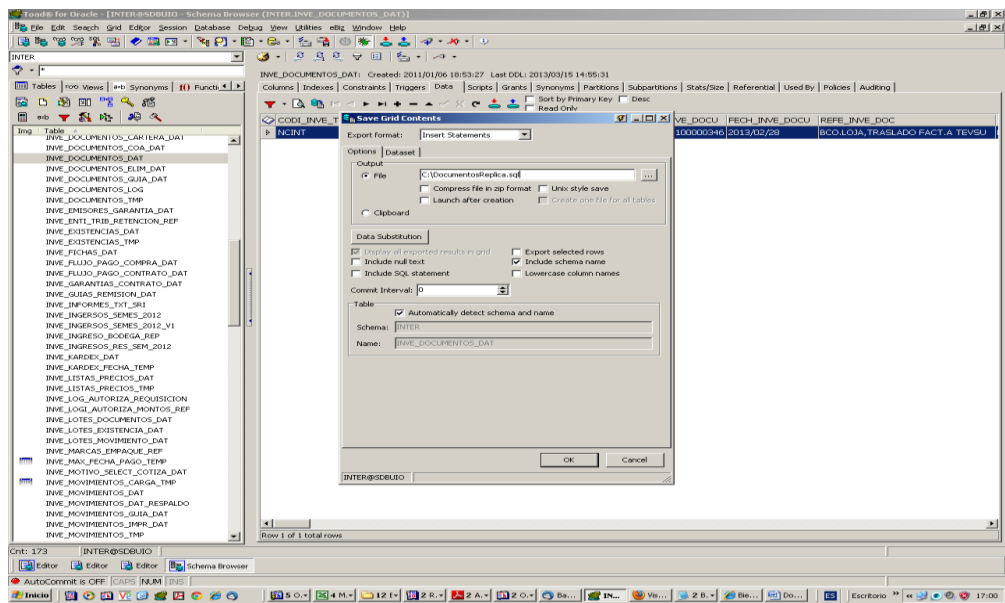


Figura 33: Pantalla filtrada de información de la base

Para realizar un filtrado de los datos de la base procedemos de la siguiente manera como se muestra en la siguiente pantalla.

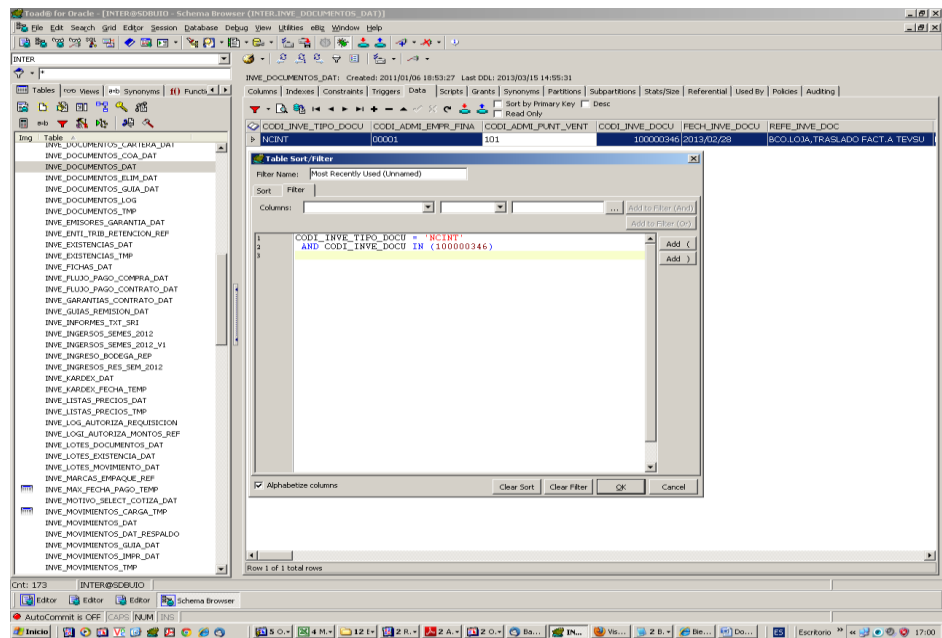


Figura 34: Búsqueda de datos en la base

A continuación pondremos el código de la factura como es original y el código en Oracle.

Cabecera documentos (facturas)

INVE_DOCUMENTOS_DAT

```
(  
  CODI_INVE_TIPO_DOCU      VARCHAR2(5 BYTE) NOT NULL,  
  CODI_ADMI_EMPR_FINA      VARCHAR2(5 BYTE) NOT NULL,  
  CODI_ADMI_PUNT_VENT      VARCHAR2(3 BYTE) NOT NULL,  
  CODI_INVE_DOCU           NUMBER(9)   NOT NULL,  
  FECH_INVE_DOCU           DATE        NOT NULL,  
  REFE_INVE_DOC            VARCHAR2(30 BYTE) NOT NULL,  
  COME_INVE_DOCU           VARCHAR2(500 BYTE),  
  CODI_INVE_BODE           NUMBER(2)   NOT NULL,  
  FECH_VCTO_INVE_DOCU     DATE,  
  CODI_ADMI_MONE           VARCHAR2(5 BYTE) NOT NULL,  
  ORIG_INVE_DOCU          NUMBER(1)   NOT NULL,  
  IMPO_IVA_INVE_DOCU      NUMBER(17,2),  
  IMPO_NETO_INVE_DOCU     NUMBER(17,2),  
  IMPO_PEND_INVE_DOCU     NUMBER(17,2),  
  IMPO_TOTA_INVE_DOCU     NUMBER(17,2),  
  IMPO_COMI_VNTA_INVE_DOCU NUMBER(17,2) DEFAULT 0,  
  CODI_INVE_CAUS_DEVO      VARCHAR2(5 BYTE),  
  CODI_INVE_TIPO_CONC      VARCHAR2(1 BYTE),  
  CODI_INVE_CONC           VARCHAR2(4 BYTE),
```

IMPR_ESTA_CUEN_INVE_DOCU NUMBER(1) DEFAULT 0,
 TIPO_DOCU_INVE_DOCU VARCHAR2(5 BYTE),
 CODI_INVE_AGEN NUMBER(9),
 AGEN_CODI_ADMI_EMPR_FINA VARCHAR2(5 BYTE),
 AGEN_CODI_ADMI_PUNT_VENT VARCHAR2(3 BYTE),
 BODE_TRAS_CODI_INVE_BODE NUMBER(2),
 BODE_TRAS_CODI_ADMI_EMPR_FINA VARCHAR2(5 BYTE),
 BODE_TRAS_CODI_ADMI_PUNT_VENT VARCHAR2(3 BYTE),
 CODI_INVE_PROV NUMBER(9),
 PROV_CODI_ADMI_EMPR_FINA VARCHAR2(5 BYTE),
 CODI_INVE_CLIE NUMBER(9),
 CLIE_CODI_ADMI_EMPR_FINA VARCHAR2(5 BYTE),
 COTI_CODI_ADMI_MONE VARCHAR2(5 BYTE),
 COTI_CODI_ADMI_PAIS VARCHAR2(24 BYTE),
 COTI_FECH_ADMI_COTI DATE,
 FECH_PEDI_INVE_DOCU DATE,
 ORDE_COMP_INVE_DOCU VARCHAR2(30 BYTE),
 FECH_ENVI_INVE_DOCU DATE,
 CODI_INVE_TRSP VARCHAR2(5 BYTE),
 CODI_INVE_TRAS VARCHAR2(5 BYTE),
 GUIA_ENVI_INVE_DOCU VARCHAR2(30 BYTE),
 FECH_VENC_PPAG_INVE_DOCU DATE,
 CONC_ENTR_INVE_DOCU VARCHAR2(4 BYTE),
 TIPO_CONC_ENTR_INVE_DOCU VARCHAR2(1 BYTE),

CONC_SALI_INVE_DOCU VARCHAR2(4 BYTE),
 CONT_INVE_DOCU NUMBER(1) DEFAULT 0,
 TIPO_CONC_SALI_INVE_DOCU VARCHAR2(1 BYTE),
 CONT_INVE_DOCU NUMBER(1) DEFAULT 0,
 FECH_CONT_INVE_DOCU DATE,
 CODI_ADMI_ESTA VARCHAR2(5 BYTE) DEFAULT 'C',
 CODI_ADMI_TABL VARCHAR2(5 BYTE) DEFAULT 'IN006' NOT
 NULL,
 IVA_INVE_DOCU NUMBER(1),
 EMPR_FINA_INVE_TIPO_DOCU VARCHAR2(5 BYTE),
 PORC_IVA_INVE_DOCU NUMBER(6,2),
 SEGU_INVE_DOCU NUMBER,
 FETE_INVE_DOCU NUMBER,
 CODI_ADMI_EMPR_FINA_PADR VARCHAR2(5 BYTE),
 CODI_ADMI_PUNT_VENT_PADR VARCHAR2(3 BYTE),
 CODI_INVE_TIPO_DOCU_PADR VARCHAR2(5 BYTE),
 CODI_INVE_DOCU_PADR NUMBER(9),
 CODI_INVE_PRVI NUMBER(2),
 CODI_INVE_TIPO_COMP NUMBER(2),
 CODI_INVE_DEST NUMBER(2),
 CODI_INVE_COMP NUMBER(2),
 FECH_VALI_INVE_DOCU DATE,
 AUTO_IMPR_INVE_DOCU VARCHAR2(10 BYTE),
 NUME_SERI_INVE_DOCU VARCHAR2(3 BYTE),
 AUTO_CONT_INVE_DOCU VARCHAR2(37 BYTE),
 NUME_COMP_INVE_DOCU VARCHAR2(13 BYTE),
 ESTA_TRAN_INVE_DOCU VARCHAR2(1 BYTE),
 PAGO_CONT_INVE_DOCU NUMBER(1),
 ESTA_IMPR_INVE_DOCU VARCHAR2(1 BYTE),
 AFECTA_IVA_INVE_DOCU NUMBER(1) DEFAULT 1
 NOT NULL,

AFECTA_VP_IVA_INVE_DOCU NUMBER(1) DEFAULT 1
 NOT NULL,
 AFECTA_VP_FUE_INVE_DOCU NUMBER(1) DEFAULT 1
 NOT NULL,
 USUA_CREA_INVE_DOCU VARCHAR2(30 BYTE),
 FECH_CREA_INVE_DOCU DATE,
 USUA_ACTU_INVE_DOCU VARCHAR2(30 BYTE),
 FECH_ACTU_INVE_DOCU DATE,
 USUA_AUTO_PREC_INVE_DOCU VARCHAR2(30 BYTE),
 FECH_AUTO_PREC_INVE_DOCU DATE,
 USUA_AUTO_LIMI_INVE_DOCU VARCHAR2(30 BYTE),
 FECH_AUTO_LIMI_INVE_DOCU DATE,
 FECH_BASE_INVE_DOCU DATE,
 TIPO_PAGO_INVE_DOCU VARCHAR2(1 BYTE) DEFAULT 'P',
 CHEQ_ENTR_INVE_DOCU NUMBER(1) DEFAULT 0
 NOT NULL,
 FECH_ENTR_CHEQ_DOCU DATE,
 USER_ENTR_CHEQ_DOCU VARCHAR2(50 BYTE),
 IMPO_DEBE_INVE_DOCU NUMBER,
 CENT_COST_AUX_DOCU VARCHAR2(15 BYTE),
 CENT_COST_DOCU VARCHAR2(15 BYTE),
 DOCU_AUTO_INVE_DOCU VARCHAR2(1 BYTE) DEFAULT 'M'
 CONSTRAINT INVE_DOC\$N\$DOC_AUT_INV_DOC NOT NULL,
 CODI_INVE_TIPO_OPER VARCHAR2(5 BYTE),
 CODI_INVE_DEST_OPER VARCHAR2(5 BYTE),
 NUME_FACT_INVE_RETE VARCHAR2(25 BYTE),
 FECH_CADU_INVE_DOCU DATE,
 FECH_EMIS_INVE_DOCU DATE,
 ESTA_TRAN_INVE_DOC VARCHAR2(1 BYTE),
 DIFE_CAMB_ADMI_COTI NUMBER,
 RETE_IVA_INVE_DOCU NUMBER(1),
 RETE_FUEN_INVE_DOCU NUMBER(1),
 ESTA_MATR_INVE_DOCU VARCHAR2(1 BYTE),
 NUME_AUTO_INVE_DOCU VARCHAR2(25 BYTE),
 CODI_TIPO_OPER_DOCU VARCHAR2(3 BYTE),
 FECH_EMIS_SERV_DAT DATE,
 TIPO_NOVE_SERV_DAT VARCHAR2(2 BYTE),
 CODI_ESTA_PAGO_CASH VARCHAR2(1 BYTE),
 FECH_INVE_TRANS_DOCU DATE,

FECH_INVE_PAGO_DOCU	DATE,
USUA_CARG_TRANS_DOCU	VARCHAR2(30 BYTE),
USUA_CARG_PAGO_DOCU	VARCHAR2(30 BYTE),
USER_AUTO_ORDEN_MOVI	VARCHAR2(20 BYTE),
CODI_ORDEN_COMPR_MOVI	VARCHAR2(40 BYTE)

)

Los demás códigos se encuentran en los anexos adjuntos en el final del documento.

CAPITULO VI

CONCLUSIONES Y RECOMENDACIONES

6.1 CONCLUSIONES

- Se podría definir para mejor comprensión del tema que una base de datos es un conjunto de datos relacionados entre sí.
- TEVCOL para brindar un servicio eficiente necesita que su base de datos sea robusta frente a diversas cargas de información desde distintos puntos al mismo tiempo, por cuanto la necesidad de implementar una réplica de la base.
- Según estándares establecidos por la Empresa TEVCOL se estudiaron los sistemas de bases de datos más óptimos para un adecuado manejo de la información y estos son la replicación con vistas materializadas y la replicación con base de datos en standby.
- Finalmente el sistema de bases de datos que se ajustó a la mayoría de necesidades requeridas por los usuarios tanto internos como externos fue la replicación con base de datos en standby.
- Con el uso de técnicas de replicación ha incidido positivamente en el desempeño de la base de datos de la empresa TEVCOL.
- Un manejador de base de datos tan robusto como es Oracle interfiere en el desempeño de la base de datos sobre todo en el manejo de la información vía remota.
- Con la implantación de la replicación se determina que el tiempo influye en las bases sin replicación y con las bases con replicación.

6.2 RECOMENDACIONES

- El presente trabajo puede servir como base de consulta referencial tanto para la aplicación de bases de datos en todo tipo de entidades así también como material de apoyo para estudiantes y docentes en carreras afines al tema.
- Para aplicación de sistemas de bases de datos en empresas con similares requerimientos se recomienda en primera instancia revisar el presupuesto con el que se cuenta para la aplicación ya que los costos varían en todos los tipos de Bases de Datos.
- Actualmente en el mercado se pueden encontrar diversos manejadores de bases de datos cuyo costo en cuanto a licenciamiento a nivel de servidor y de usuario son costo accesible.
- Se recomienda programar los tiempos de implementación del sistema en un periodo mínimo de tres meses para una correcta ejecución del sistema de bases de datos.
- Se recomienda llevar una documentación de los procesos de la base de datos de la empresa TEVCOL.

CAPITULO VII

Bibliografía

Engeleiter, K. (2009). Oracle Database. Redwood Shores, CA 94065 EEUU: Oracle Corporation.

Heurtel, O. (2009). Oracle 11g Administración. Barcelona: Ediciones ENI.

LAURA, M. (2010). Diseño y Construcción de Bases de datos Distribuidas. Madrid: madrid.

Luis, P. M. (2006). Bases de Datos Distribuidas. Barcelona Esp.

NASSIR, S. (2009). PREPARACION Y ELABORACION PROYECTOS.
BOGOTA: GRAW.

Kevin Loney, Bob Bryla, Oracle Database 10g, McGraw-Hill Interamericana de España S.L., 26/11/2005

Chris Ostrowski, Bradley Brown, Oracle Application Server 10g Web Development, McGraw-Hill Education, 2005

CAPITULO VIII

ANEXOS

ANEXO 1. Detalle factura original

ANEXO 2. Presentación de la factura en el servidor original el cual se llama TEVCOL NIIF SERVIDOR 192.168.1.10 en la imagen muestra el nombre del servidor.

ANEXO 3. Código insert cabecera

ANEXO 4. Código inserts detalle.

ANEXO 5. Además de eso lo que también se puede replicar son las notas de crédito que quedarían de la siguiente manera esta pantalla está en el servidor NIIF.

ANEXO 6. La visibilidad de la misma nota de crédito en el servidor de replicado en el mismo tiempo nos muestra una nota de crédito igual al del servidor principal pero nos muestra el nombre del otro servidor.

ANEXO 7. Código insert cabecera.

ANEXO 8. Código insert detalle.

ANEXO 9. Código insert cruce nota de crédito con factura

ANEXO 25. Inserción de un cliente en replica

ANEXO 24. Inserción de un cliente en producción

ANEXO 23. Inserción de un proveedor en replica

ANEXO 22.Insercion de un proveedor en producción

ANEXO 21. Inserción de una baja de activo fijo en replica

ANEXO 20. Inserción de una baja de activo fijo en producción

ANEXO 19. Inserción de una adición de activo fijo en producción

ANEXO 18. Inserción de una adición de activo fijo en producción

ANEXO 17. Registro de activo fijo en servidor de replica

ANEXO 16. Registro de activo fijo en servidor de producción

ANEXO 15. Inserción cta contable en servidor de replica

ANEXO 14. Inserción cta contable en servidor de producción

ANEXO 13. Inserción asiento contable en servidor de replica

ANEXO 12.Insercion asiento contable en servidor de producción

ANEXO 11. Inserción empleados en server replica

ANEXO 10. Inserción empleados en server producción

ANEXO 1. Detalle factura original

INVE_MOVIMIENTOS_DAT

```
(
  NUME_INVE_MOVI      NUMBER(5) CONSTRAINT
INVE_MOV$$NUM_INV_MOV NOT NULL,
  CODI_INVE_TIPO_DOCU  VARCHAR2(5 BYTE) CONSTRAINT
INVE_MOV$$COD_INV_TIP_DOC NOT NULL,
  CODI_INVE_DOCU       NUMBER(9) CONSTRAINT
INVE_MOV$$COD_INV_DOC NOT NULL,
  CODI_ADMI_EMPR_FINA  VARCHAR2(5 BYTE) CONSTRAINT
INVE_MOV$$COD_AMD_EMP_FIN NOT NULL,
  CODI_ADMI_PUNT_VENT  VARCHAR2(3 BYTE) CONSTRAINT
INVE_MOV$$COD_ADM_PUN_VEN NOT NULL,
  CODI_INVE_PROD       VARCHAR2(24 BYTE) CONSTRAINT
INVE_MOV$$COD_INV_PRO NOT NULL,
  DESC_INVE_MOVI      VARCHAR2(1000 BYTE),
  UNID_INVE_MOVI      NUMBER(12,2) CONSTRAINT
INVE_MOV$$UNI_INV_MOV NOT NULL,
  PREC_UNIT_INVE_MOVI NUMBER(17,2)  DEFAULT 0,
  IMPO_DESC_AUTO_INVE_MOVI NUMBER(17,2),
  PORC_DESC_ESPE_INVE_MOVI NUMBER(6,2),
  IMPO_NETO_INVE_MOVI  NUMBER(17,2)  DEFAULT 0
CONSTRAINT INVE_MOV$$IMP_NET_INV NOT NULL,
  IMPO_TOTA_INVE_MOVI  NUMBER(17,2)  DEFAULT 0
CONSTRAINT INVE_MOV$$IMP_TOT_INV NOT NULL,
  IMPO_IVA_INVE_MOVI   NUMBER(17,2),
  IMPO_COMI_INVE_MOVI  NUMBER(17,2)  DEFAULT 0,
  IMPO_DESC_ESPE_INVE_MOVI NUMBER(17,2),
  COST_INVE_MOVI       NUMBER(25,7)  DEFAULT 0,
  CANC_INVE_MOVI       NUMBER(1),
  IVA_INVE_MOVI        NUMBER(1),
  ESTA_INVE_MOVI       VARCHAR2(1 BYTE),
  UNID_PEND_INVE_MOVI  NUMBER(12,2),
  UNID_PROC_INVE_MOVI  NUMBER(12,2),
  CODI_ADMI_EMPR_FINA_PADR VARCHAR2(5 BYTE),
  CODI_ADMI_PUNT_VENT_PADR VARCHAR2(3 BYTE),
  CODI_INVE_TIPO_DOCU_PADR VARCHAR2(5 BYTE),
  CODI_INVE_DOCU_PADR  NUMBER(9),
```

```

NUME_INVE_MOVI_PADR    NUMBER(5),
RETE_IVA_INVE_MOVI    NUMBER(17,2),
RETE_FUEN_INVE_MOVI    NUMBER(17,2),
UNID_ANTE_INVE_MOVI    NUMBER(12,2),
ESTA_TRAN_INVE_MOVI    VARCHAR2(1 BYTE),
CODI_INVE_TIPO_PAGO    VARCHAR2(25 BYTE),
PORC_IVA_INVE_MOVI    NUMBER          DEFAULT 0
CONSTRAINT INVE_MOVSN$POR_IVA_INV_MOV NOT NULL,
TIPO_UNID_INVE_MOVI    VARCHAR2(5 BYTE),
INDI_AUTO_INVE_MOV     VARCHAR2(1 BYTE)  DEFAULT 'N',
UNID_REQU_INVE_MOVI    NUMBER(20,8),
UNID_AUTO_INVE_MOVI    NUMBER(20,8),
ESTA_ANTE_INVE_MOVI    VARCHAR2(5 BYTE),
CENT_COST_INVE_MOVI    VARCHAR2(15 BYTE)  DEFAULT NULL,
CENT_COST_AUX_INVE_MOVI VARCHAR2(15 BYTE),
NUME_DISC_INVE_MOVI    VARCHAR2(5 BYTE),
CODI_INVE_TIPO_GASTO    NUMBER(5),
UNID_PEND_DESP_INVE_MOVI NUMBER(30),
UNID_DESP_INVE_MOVI    NUMBER(30),
NUME_PERCHA_MOVI       VARCHAR2(20 BYTE),
FECHA_CREA             DATE          DEFAULT SYSDATE,
USER_CREA              VARCHAR2(30 BYTE)  DEFAULT USER,
CODI_ADMI_TABL         VARCHAR2(5 BYTE)  DEFAULT 'IN015',
UNID_ANUL_INVE_MOVI    NUMBER          DEFAULT 0
CONSTRAINT INV_MOVSN$UNI_ANU_INV_MOV NOT NULL,
NUME_COMPR_INVE_MOVI   NUMBER,
CODI_TIPO_OPER_MOVI    VARCHAR2(3 BYTE)
)

```


ANEXO 2. Presentación de la factura en el servidor original el cual se llama TEVCOL NIIF SERVIDOR 192.168.1.10 en la imagen muestra el nombre del servidor

The screenshot shows the 'Capital' software interface for 'MODULO DE CUENTAS POR COBRAR - [Facturación - IN2111]'. The window title is 'Capital >> MODULO DE CUENTAS POR COBRAR - [Facturación - IN2111]'. The interface includes a menu bar (Acción, Edición, Consulta, Bloque, Registro, Ayuda, Ventana), a toolbar, and a main form area. The form contains the following data:

- Sucursal: 10 QUITO
- Servicio: TV TRANSPORTE DE VALORES
- Tipo Documento: FACTU FACTURA
- Código: 100046007
- Estado: PROCESADO
- Referencia: 001-001-000046007
- Fecha: 2013/02/26
- Comentario: LIQUIDACION POR SERVICIO DE FUNDAS MAL O
- Fecha Vencimiento: 2013/02/26
- Tipo Novedad: 01 NINGUNA
- Fecha del Servicio: 2013/01/25
- Cliente: 135 BANCO DE LA PRODUCCION S.A.- PRODDUBANCO
- Tipo Operac: OPERACION
- Autorización: 1110959596
- IVA a nivel de Documento:
- Moneda: DOLAR

The 'Movimientos' table is as follows:

Num.	Producto	Descripción	Sucursal	Precio	Descuento	Neto	Total
1	Y-TV-05	ESPECIES NO UTILIZADA	10	30.00	0.00	30.00	33.60
2	Y-TV-05	ESPECIES NO UTILIZADA	20	60.00	0.00	60.00	67.20
3	Y-TV-05	ESPECIES NO UTILIZADA	30	1.50	0.00	1.50	1.68
4	Y-TV-05	ESPECIES NO UTILIZADA	40	10.50	0.00	10.50	11.76
5	Y-TV-05	ESPECIES NO UTILIZADA	50	30.00	0.00	30.00	33.60

Summary values:

- Unidades: 8.00
- Pendiente: 183.12
- IVA: 19.62
- Neto: 163.50
- Total: 183.12

The 'Detalle Impresión Factura' section is empty.

Figura 35: Pantalla a replicar Servidor NIIF

Presentación de la misma factura ya realizada la réplica en el cual se presenta el nombre del otro servidor en el cual está la otra base de datos replicada SERVIDOR 192.168.1.45 TEVCOL REPLICA

The screenshot shows the 'Capital' software interface for 'MODULO DE CUENTAS POR COBRAR - [Facturación - IN2111]'. The window title is 'Capital >> MODULO DE CUENTAS POR COBRAR - [Facturación - IN2111]'. The interface is identical to Figure 35, but the server name in the title bar is 'TEVCOL REPLICA'.

The 'Movimientos' table is as follows:

Num.	Producto	Descripción	Sucursal	Precio	Descuento	Neto	Total
1	Y-TV-05	ESPECIES NO UTILIZADA	10	30.00	0.00	30.00	33.60
2	Y-TV-05	ESPECIES NO UTILIZADA	20	60.00	0.00	60.00	67.20
3	Y-TV-05	ESPECIES NO UTILIZADA	30	1.50	0.00	1.50	1.68
4	Y-TV-05	ESPECIES NO UTILIZADA	40	10.50	0.00	10.50	11.76
5	Y-TV-05	ESPECIES NO UTILIZADA	50	30.00	0.00	30.00	33.60

Summary values:

- Unidades: 8.00
- Pendiente: 183.12
- IVA: 19.62
- Neto: 163.50
- Total: 183.12

The 'Detalle Impresión Factura' section is empty.

Figura 36: Pantalla replicada en servidor REPLICA

ANEXO 3. El código de la factura anterior se nos presenta de la siguiente manera, **CODIGO INSERT CABECERA**

Insert into INTER.INVE_DOCUMENTOS_DAT

(CODI_INVE_TIPO_DOCU, CODI_ADMI_EMPR_FINA,
CODI_ADMI_PUNT_VENT, CODI_INVE_DOCU, FECH_INVE_DOCU,
REFE_INVE_DOC, COME_INVE_DOCU, CODI_INVE_BODE,
FECH_VCTO_INVE_DOCU, CODI_ADMI_MONE, ORIG_INVE_DOCU,
IMPO_IVA_INVE_DOCU, IMPO_NETO_INVE_DOCU,
IMPO_PEND_INVE_DOCU, IMPO_TOTA_INVE_DOCU,
IMPO_COMI_VNTA_INVE_DOCU, IMPR_ESTA_CUEN_INVE_DOCU,
TIPO_DOCU_INVE_DOCU, AGEN_CODI_ADMI_EMPR_FINA,
AGEN_CODI_ADMI_PUNT_VENT, CODI_INVE_CLIE,
CLIE_CODI_ADMI_EMPR_FINA, COTI_CODI_ADMI_MONE,
COTI_CODI_ADMI_PAIS, COTI_FECH_ADMI_COTI, CONT_INVE_DOCU,
CODI_ADMI_ESTA, CODI_ADMI_TABL, IVA_INVE_DOCU,
EMPR_FINA_INVE_TIPO_DOCU, PORC_IVA_INVE_DOCU,
FETE_INVE_DOCU, PAGO_CONT_INVE_DOCU,
AFECTA_IVA_INVE_DOCU, AFECTA_VP_IVA_INVE_DOCU,
AFECTA_VP_FUE_INVE_DOCU, USUA_CREA_INVE_DOCU,
FECH_CREA_INVE_DOCU, TIPO_PAGO_INVE_DOCU,
CHEQ_ENTR_INVE_DOCU, IMPO_DEBE_INVE_DOCU,
CENT_COST_AUX_DOCU, CENT_COST_DOCU,
DOCU_AUTO_INVE_DOCU, NUME_AUTO_INVE_DOCU,
CODI_TIPO_OPER_DOCU, FECH_EMIS_SERV_DAT,
TIPO_NOVE_SERV_DAT)

Values

('FACTU', '00001', '101', 100046007, TO_DATE('02/26/2013 00:00:00',
'MM/DD/YYYY HH24:MI:SS'), '001-001-000046007', 'LIQUIDACION POR
SERVICIO DE FUNDAS MAL O
NO UTILIZADAS, DEL 08 AL 25-01-2013', 10, TO_DATE('02/26/2013
00:00:00', 'MM/DD/YYYY HH24:MI:SS'), '1', 3, 19.62, 163.5, 183.12, 183.12, 0,
0, 'FACTU', '00001', '101', 135, '00001', '1', 'EC', TO_DATE('02/26/2013
00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 0, 'P', 'IN006', 1, '00001', 12, 0, 1, 1,
1, 1, 'ATRANSVAL', TO_DATE('02/26/2013 00:00:00', 'MM/DD/YYYY
HH24:MI:SS'), 'P', 0, 183.12, 'TV', '10', 'M', '1110959596', 'OPE',
TO_DATE('01/25/2013 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), '01');
COMMIT;

ANEXO 4. Código inserts detalle

Insert into INTER.INVE_MOVIMIENTOS_DAT

(NUME_INVE_MOVI, CODI_INVE_TIPO_DOCU, CODI_INVE_DOCU,
CODI_ADMI_EMPR_FINA, CODI_ADMI_PUNT_VENT,
CODI_INVE_PROD, DESC_INVE_MOVI, UNID_INVE_MOVI,
PREC_UNIT_INVE_MOVI, IMPO_DESC_AUTO_INVE_MOVI,
PORC_DESC_ESPE_INVE_MOVI, IMPO_NETO_INVE_MOVI,
IMPO_TOTA_INVE_MOVI, IMPO_IVA_INVE_MOVI,
IMPO_COMI_INVE_MOVI, IMPO_DESC_ESPE_INVE_MOVI,
COST_INVE_MOVI, IVA_INVE_MOVI, UNID_PEND_INVE_MOVI,
UNID_PROC_INVE_MOVI, UNID_ANTE_INVE_MOVI,
PORC_IVA_INVE_MOVI, INDI_AUTO_INVE_MOV,
CENT_COST_INVE_MOVI, FECHA_CREA, USER_CREA,
CODI_ADMI_TABL, UNID_ANUL_INVE_MOVI,
NUME_COMPR_INVE_MOVI, CODI_TIPO_OPER_MOVI)

Values

(1, 'FACTU', 100046007, '00001', '101', 'V-TV-05', 'FUNDAS TV NO
UTILIZADAS : QUITO, BANCO DE LA PRODUCCION S.A. ,
CORRESPONDIENTE A : 2013/01/08 - 2013/01/25 , FACTURA : 0. SEGUN
19 ANEXOS ADJUNTOS', 1, 30, 0, 0, 30, 33.6, 3.6, 0, 0, 0, 1, 0, 0, 0, 0, 'N', '10',
TO_DATE('02/26/2013 00:00:00', 'MM/DD/YYYY HH24:MI:SS'),
'ATRANSVAL', 'IN015', 0, 0, 'OPE');

Insert into INTER.INVE_MOVIMIENTOS_DAT

(NUME_INVE_MOVI, CODI_INVE_TIPO_DOCU, CODI_INVE_DOCU,
CODI_ADMI_EMPR_FINA, CODI_ADMI_PUNT_VENT,
CODI_INVE_PROD, DESC_INVE_MOVI, UNID_INVE_MOVI,
PREC_UNIT_INVE_MOVI, IMPO_DESC_AUTO_INVE_MOVI,
PORC_DESC_ESPE_INVE_MOVI, IMPO_NETO_INVE_MOVI,
IMPO_TOTA_INVE_MOVI, IMPO_IVA_INVE_MOVI,
IMPO_COMI_INVE_MOVI, IMPO_DESC_ESPE_INVE_MOVI,
COST_INVE_MOVI, IVA_INVE_MOVI, UNID_PEND_INVE_MOVI,
UNID_PROC_INVE_MOVI, UNID_ANTE_INVE_MOVI,
PORC_IVA_INVE_MOVI, INDI_AUTO_INVE_MOV,
CENT_COST_INVE_MOVI, FECHA_CREA, USER_CREA,
CODI_ADMI_TABL, UNID_ANUL_INVE_MOVI,
NUME_COMPR_INVE_MOVI, CODI_TIPO_OPER_MOVI)

Values

(2, 'FACTU', 100046007, '00001', '101', 'V-TV-05', 'FUNDAS TV NO UTILIZADAS : GUAYAQUIL, BANCO DE LA PRODUCCION S.A. , CORRESPONDIENTE A : 2013/01/08 - 2013/01/25 , FACTURA : 0. SEGUN 10 ANEXOS ADJUNTOS', 1, 60, 0, 0, 60, 67.2, 7.2, 0, 0, 0, 1, 0, 0, 0, 0, 'N', '20', TO_DATE('02/26/2013 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'ATRANSVAL', 'IN015', 0, 0, 'OPE');

Insert into INTER.INVE_MOVIMIENTOS_DAT

(NUME_INVE_MOVI, CODI_INVE_TIPO_DOCU, CODI_INVE_DOCU, CODI_ADMI_EMPR_FINA, CODI_ADMI_PUNT_VENT, CODI_INVE_PROD, DESC_INVE_MOVI, UNID_INVE_MOVI, PREC_UNIT_INVE_MOVI, IMPO_DESC_AUTO_INVE_MOVI, PORC_DESC_ESPE_INVE_MOVI, IMPO_NETO_INVE_MOVI, IMPO_TOTA_INVE_MOVI, IMPO_IVA_INVE_MOVI, IMPO_COMI_INVE_MOVI, IMPO_DESC_ESPE_INVE_MOVI, COST_INVE_MOVI, IVA_INVE_MOVI, UNID_PEND_INVE_MOVI, UNID_PROC_INVE_MOVI, UNID_ANTE_INVE_MOVI, PORC_IVA_INVE_MOVI, INDI_AUTO_INVE_MOV, CENT_COST_INVE_MOVI, FECHA_CREA, USER_CREA, CODI_ADMI_TABL, UNID_ANUL_INVE_MOVI, NUME_COMPR_INVE_MOVI, CODI_TIPO_OPER_MOVI)

Values

(3, 'FACTU', 100046007, '00001', '101', 'V-TV-05', 'FUNDAS TV NO UTILIZADAS : AMBATO, BANCO DE LA PRODUCCION S.A. , CORRESPONDIENTE A : 2013/01/08 - 2013/01/25 , FACTURA : 0. SEGUN 1 ANEXOS ADJUNTOS', 1, 1.5, 0, 0, 1.5, 1.68, 0.18, 0, 0, 0, 1, 0, 0, 0, 0, 'N', '30', TO_DATE('02/26/2013 00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'ATRANSVAL', 'IN015', 0, 0, 'OPE');

Insert into INTER.INVE_MOVIMIENTOS_DAT

(NUME_INVE_MOVI, CODI_INVE_TIPO_DOCU, CODI_INVE_DOCU, CODI_ADMI_EMPR_FINA, CODI_ADMI_PUNT_VENT, CODI_INVE_PROD, DESC_INVE_MOVI, UNID_INVE_MOVI, PREC_UNIT_INVE_MOVI, IMPO_DESC_AUTO_INVE_MOVI, PORC_DESC_ESPE_INVE_MOVI, IMPO_NETO_INVE_MOVI, IMPO_TOTA_INVE_MOVI, IMPO_IVA_INVE_MOVI, IMPO_COMI_INVE_MOVI, IMPO_DESC_ESPE_INVE_MOVI, COST_INVE_MOVI, IVA_INVE_MOVI, UNID_PEND_INVE_MOVI, UNID_PROC_INVE_MOVI, UNID_ANTE_INVE_MOVI, PORC_IVA_INVE_MOVI, INDI_AUTO_INVE_MOV, CENT_COST_INVE_MOVI, FECHA_CREA, USER_CREA,

CODI_ADMI_TABL, UNID_ANUL_INVE_MOVI,
NUME_COMPR_INVE_MOVI, CODI_TIPO_OPER_MOVI)

Values

(4, 'FACTU', 100046007, '00001', '101', 'V-TV-05', 'FUNDAS TV NO
UTILIZADAS : RIOBAMBA, BANCO DE LA PRODUCCION S.A. ,
CORRESPONDIENTE A : 2013/01/08 - 2013/01/25 , FACTURA : 0. SEGUN 6
ANEXOS ADJUNTOS', 1, 10.5, 0, 0, 10.5, 11.76, 1.26, 0, 0, 0, 1, 0, 0, 0, 0, 'N',
'40', TO_DATE('02/26/2013 00:00:00', 'MM/DD/YYYY HH24:MI:SS'),
'ATRANSVAL', 'IN015', 0, 0, 'OPE');

Insert into INTER.INVE_MOVIMIENTOS_DAT

(NUME_INVE_MOVI, CODI_INVE_TIPO_DOCU, CODI_INVE_DOCU,
CODI_ADMI_EMPR_FINA, CODI_ADMI_PUNT_VENT,
CODI_INVE_PROD, DESC_INVE_MOVI, UNID_INVE_MOVI,
PREC_UNIT_INVE_MOVI, IMPO_DESC_AUTO_INVE_MOVI,
PORC_DESC_ESPE_INVE_MOVI, IMPO_NETO_INVE_MOVI,
IMPO_TOTA_INVE_MOVI, IMPO_IVA_INVE_MOVI,
IMPO_COMI_INVE_MOVI, IMPO_DESC_ESPE_INVE_MOVI,
COST_INVE_MOVI, IVA_INVE_MOVI, UNID_PEND_INVE_MOVI,
UNID_PROC_INVE_MOVI, UNID_ANTE_INVE_MOVI,
PORC_IVA_INVE_MOVI, INDI_AUTO_INVE_MOV,
CENT_COST_INVE_MOVI, FECHA_CREA, USER_CREA,
CODI_ADMI_TABL, UNID_ANUL_INVE_MOVI,
NUME_COMPR_INVE_MOVI, CODI_TIPO_OPER_MOVI)

Values

(5, 'FACTU', 100046007, '00001', '101', 'V-TV-05', 'FUNDAS TV NO
UTILIZADAS : SANTO DOMINGO, BANCO DE LA PRODUCCION S.A. ,
CORRESPONDIENTE A : 2013/01/08 - 2013/01/25 , FACTURA : 0. SEGUN
10 ANEXOS ADJUNTOS', 1, 30, 0, 0, 30, 33.6, 3.6, 0, 0, 0, 1, 0, 0, 0, 0, 'N', '50',
TO_DATE('02/26/2013 00:00:00', 'MM/DD/YYYY HH24:MI:SS'),
'ATRANSVAL', 'IN015', 0, 0, 'OPE');

Insert into INTER.INVE_MOVIMIENTOS_DAT

(NUME_INVE_MOVI, CODI_INVE_TIPO_DOCU, CODI_INVE_DOCU,
CODI_ADMI_EMPR_FINA, CODI_ADMI_PUNT_VENT,
CODI_INVE_PROD, DESC_INVE_MOVI, UNID_INVE_MOVI,
PREC_UNIT_INVE_MOVI, IMPO_DESC_AUTO_INVE_MOVI,
PORC_DESC_ESPE_INVE_MOVI, IMPO_NETO_INVE_MOVI,
IMPO_TOTA_INVE_MOVI, IMPO_IVA_INVE_MOVI,
IMPO_COMI_INVE_MOVI, IMPO_DESC_ESPE_INVE_MOVI,
COST_INVE_MOVI, IVA_INVE_MOVI, UNID_PEND_INVE_MOVI,

UNID_PROC_INVE_MOVI, UNID_ANTE_INVE_MOVI,
PORC_IVA_INVE_MOVI, INDI_AUTO_INVE_MOV,
CENT_COST_INVE_MOVI, FECHA_CREA, USER_CREA,
CODI_ADMI_TABL, UNID_ANUL_INVE_MOVI,
NUME_COMPR_INVE_MOVI, CODI_TIPO_OPER_MOVI)

Values

(6, 'FACTU', 100046007, '00001', '101', 'V-TV-05', 'FUNDAS TV NO
UTILIZADAS : QUEVEDO, BANCO DE LA PRODUCCION S.A. ,
CORRESPONDIENTE A : 2013/01/08 - 2013/01/25 , FACTURA : 0. SEGUN 4
ANEXOS ADJUNTOS', 1, 9, 0, 0, 9, 10.08, 1.

9, 0, 0, 9, 10.08, 1.08, 0, 0, 0, 1, 0, 0, 0, 0, 'N', '52', TO_DATE('02/26/2013
00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'ATRANSVAL', 'IN015', 0, 0, 'OPE');

Insert into INTER.INVE_MOVIMIENTOS_DAT

(NUME_INVE_MOVI, CODI_INVE_TIPO_DOCU, CODI_INVE_DOCU,
CODI_ADMI_EMPR_FINA, CODI_ADMI_PUNT_VENT,
CODI_INVE_PROD, DESC_INVE_MOVI, UNID_INVE_MOVI,
PREC_UNIT_INVE_MOVI, IMPO_DESC_AUTO_INVE_MOVI,
PORC_DESC_ESPE_INVE_MOVI, IMPO_NETO_INVE_MOVI,
IMPO_TOTA_INVE_MOVI, IMPO_IVA_INVE_MOVI,
IMPO_COMI_INVE_MOVI, IMPO_DESC_ESPE_INVE_MOVI,
COST_INVE_MOVI, IVA_INVE_MOVI, UNID_PEND_INVE_MOVI,
UNID_PROC_INVE_MOVI, UNID_ANTE_INVE_MOVI,
PORC_IVA_INVE_MOVI, INDI_AUTO_INVE_MOV,
CENT_COST_INVE_MOVI, FECHA_CREA, USER_CREA,
CODI_ADMI_TABL, UNID_ANUL_INVE_MOVI,
NUME_COMPR_INVE_MOVI, CODI_TIPO_OPER_MOVI)

Values

(7, 'FACTU', 100046007, '00001', '101', 'V-TV-05', 'FUNDAS TV NO
UTILIZADAS : PORTOVIEJO, BANCO DE LA PRODUCCION S.A. ,
CORRESPONDIENTE A : 2013/01/08 - 2013/01/25 , FACTURA : 0. SEGUN 3
ANEXOS ADJUNTOS', 1, 9, 0, 0, 9, 10.08, 1.08, 0, 0, 0, 1, 0, 0, 0, 0, 'N', '60',
TO_DATE('02/26/2013 00:00:00', 'MM/DD/YYYY HH24:MI:SS'),
'ATRANSVAL', 'IN015', 0, 0, 'OPE');

Insert into INTER.INVE_MOVIMIENTOS_DAT

(NUME_INVE_MOVI, CODI_INVE_TIPO_DOCU, CODI_INVE_DOCU,
CODI_ADMI_EMPR_FINA, CODI_ADMI_PUNT_VENT,
CODI_INVE_PROD, DESC_INVE_MOVI, UNID_INVE_MOVI,
PREC_UNIT_INVE_MOVI, IMPO_DESC_AUTO_INVE_MOVI,
PORC_DESC_ESPE_INVE_MOVI, IMPO_NETO_INVE_MOVI,

IMPO_TOTA_INVE_MOVI, IMPO_IVA_INVE_MOVI,
 IMPO_COMI_INVE_MOVI, IMPO_DESC_ESPE_INVE_MOVI,
 COST_INVE_MOVI, IVA_INVE_MOVI, UNID_PEND_INVE_MOVI,
 UNID_PROC_INVE_MOVI, UNID_ANTE_INVE_MOVI,
 PORC_IVA_INVE_MOVI, INDI_AUTO_INVE_MOV,
 CENT_COST_INVE_MOVI, FECHA_CREA, USER_CREA,
 CODI_ADMI_TABL, UNID_ANUL_INVE_MOVI,
 NUME_COMPR_INVE_MOVI, CODI_TIPO_OPER_MOVI)

Values

(8, 'FACTU', 100046007, '00001', '101', 'V-TV-05', 'FUNDAS TV NO
 UTILIZADAS : MANTA, BANCO DE LA PRODUCCION S.A. ,
 CORRESPONDIENTE A : 2013/01/08 - 2013/01/25 , FACTURA : 0. SEGUN 6
 ANEXOS ADJUNTOS', 1, 13.5, 0, 0, 13.5, 15.12, 1.62, 0, 0, 0, 1, 0, 0, 0, 0, 'N',
 '61', TO_DATE('02/26/2013 00:00:00', 'MM/DD/YYYY HH24:MI:SS'),
 'ATRANSVAL', 'IN015', 0, 0, 'OPE');
 COMMIT;

ANEXO 5. Además de eso lo que también se puede replicar son las notas de crédito que quedarían de la siguiente manera esta pantalla está en el servidor NIIF

The screenshot shows the 'MODULO DE CUENTAS POR COBRAR' interface. The main data fields are as follows:

Sucursal	10	Servicio	ADM	ADMINISTRACION			
Cliente	771	BANCO DE LOJA S.A.	Fecha	2013/02/28			
Fecha Vcto	2013/02/28	Documento	100000346	Estado	CONTABILIZADO		
Tipo Documento	NCINT	Moneda	1	DOLAR	Saldo Pendiente	.00	
No. Serie (Documento)	BCO.LOJA.TRASLADO FACT.A TEVSU		Comentario	BCO.LOJA.TRASLADO FACT.A TEVSU		Tipo Oper NIIF	OPERATIVO

Tipo Gasto	Neto	IVA	Total	Descripción
1010501005	5,718.17	0.00	5,718.17	BCO.LOJA.TRASLADO FACT.A TEVS
	5,718.17	0.00	5,718.17	

Documento	Tipo Documento	Moneda	Monto Documento	Saldo Pendiente	Saldo Monto Local	Valor del Pago	Pago Monto Local
100046248	FACTURA	DOLAR	3,327.63	.00	.00	3,268.21	3,268.21
100046249	FACTURA	DOLAR	259.35	.00	.00	254.71	254.71
100046250	FACTURA	DOLAR	481.60	.00	.00	473.00	473.00
						5,718.17	5,718.17

Figura 37: Nota de crédito en servidor NIIF

ANEXO 6. La visibilidad de la misma nota de crédito en el servidor de replicado en el mismo tiempo nos muestra una nota de crédito igual al del servidor principal pero nos muestra el nombre del otro servidor.

Capital >> MODULO DE CUENTAS POR COBRAR - [Notas de Crédito -BANC0003]

Acción Edición Consulta Bloque Registro Ayuda Ventana

TEVCOL REPLICA PRINCIPAL INTER 2013/03/15

Sucursal: QUITO Servicio: ADM ADMINISTRACION

Cliente: 771 BANCO DE LOJA S.A. Fecha: 2013/02/28 Fecha Vcto: 2013/02/28 Documento: 100000346 Estado: CONTABILIZADO

Tipo Documento: NCINT Moneda: DOLAR Saldo Pendiente: .00

No. Serie (Documento): BCO.LOJA.TRASLADÓ FACT.A TEVSU Comentario: BCO.LOJA.TRASLADÓ FACT.A TEVSU Tipo Oper. NIIF: OPERATIVO

Notas de Crédito				
Tipo Gasto	Neto	IVA	Total	Descripción
1010501005 CTAS X COB. TEVSUR	5,718.17	0.00	5,718.17	BCO.LOJA.TRASLADÓ FACT.A TEVS
	5,718.17	0.00	5,718.17	

Documentos							
Documento	Tipo Documento	Moneda	Monto Documento	Saldo Pendiente	Saldo Monto Local	Valor del Pago	Pago Monto Local
100046248	FACTURA	DOLAR	3,327.63	.00	.00	3,268.21	3,268.21
100046249	FACTURA	DOLAR	259.35	.00	.00	254.71	254.71
100046250	FACTURA	DOLAR	481.60	.00	.00	473.00	473.00
						5,718.17	5,718.17

Figura 38: Nota de crédito en servidor replicado

ANEXO 7. El código de la nota de crédito queda de la siguiente manera.

CODIGO INSERT CABECERA

Insert into INTER.INVE_DOCUMENTOS_DAT

(CODI_INVE_TIPO_DOCU, CODI_ADMI_EMPR_FINA,
 CODI_ADMI_PUNT_VENT, CODI_INVE_DOCU, FECH_INVE_DOCU,
 REFE_INVE_DOC, COME_INVE_DOCU, CODI_INVE_BODE,
 FECH_VCTO_INVE_DOCU, CODI_ADMI_MONE, ORIG_INVE_DOCU,
 IMPO_IVA_INVE_DOCU, IMPO_NETO_INVE_DOCU,
 IMPO_PEND_INVE_DOCU, IMPO_TOTA_INVE_DOCU,
 IMPR_ESTA_CUEN_INVE_DOCU, CODI_INVE_CLIE,
 CLIE_CODI_ADMI_EMPR_FINA, CONT_INVE_DOCU,
 CODI_ADMI_ESTA, CODI_ADMI_TABL,
 EMPR_FINA_INVE_TIPO_DOCU, AFECTA_IVA_INVE_DOCU,
 AFECTA_VP_IVA_INVE_DOCU, AFECTA_VP_FUE_INVE_DOCU,
 USUA_CREA_INVE_DOCU, FECH_CREA_INVE_DOCU,
 USUA_ACTU_INVE_DOCU, FECH_ACTU_INVE_DOCU,
 TIPO_PAGO_INVE_DOCU, CHEQ_ENTR_INVE_DOCU,

IMPO_DEBE_INVE_DOCU, CENT_COST_AUX_DOCU,
CENT_COST_DOCU, DOCU_AUTO_INVE_DOCU,
CODI_TIPO_OPER_DOCU, CODI_ESTA_PAGO_CASH)

Values

('NCINT', '00001', '101', 100000346, TO_DATE('02/28/2013 00:00:00',
'MM/DD/YYYY HH24:MI:SS'), 'BCO.LOJA,TRASLADO FACT.A TEVSU',
'BCO.LOJA,TRASLADO FACT.A TEVSU', 10, TO_DATE('02/28/2013
00:00:00', 'MM/DD/YYYY HH24:MI:SS'), '1', 5, 0, 5718.17, 0, 5718.17, 0, 771,
'00001', 1, 'O', 'IN006', '00001', 1, 1, 1, 'JPAREJA', TO_DATE('03/06/2013
00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'JPAREJA', TO_DATE('03/06/2013
00:00:00', 'MM/DD/YYYY HH24:MI:SS'), 'P', 0, 5718.17, 'ADM', '10', 'M',
'OPE', 'O');
COMMIT;

ANEXO 8. Código insert detalle

Insert into INTER.BANC_MOVIMIENTOS_DAT

(CODI_INVE_DOCU, CODI_INVE_TIPO_DOCU,
CODI_ADMI_PUNT_VENT, CODI_ADMI_EMPR_FINA,
SECU_BANC_MOVI, CODI_BANC_TIPO_PAGO, DESC_BANC_MOVI,
MONT_BANC_MOVI, CONC_BANC_MOVI, NETO_BANC_MOVI,
IVA_BANC_MOVI, USER_CREA, FECHA_CREA,
ESTA_CRUZ_BANC_MOVI, CODI_TIPO_OPER_MOVI)

Values

(100000346, 'NCINT', '101', '00001', 1, '1010501005',
'BCO.LOJA,TRASLADO FACT.A TEVSU', 5718.17, 'N', 5718.17, 0,
'JPAREJA', TO_DATE('03/06/2013 16:17:22', 'MM/DD/YYYY HH24:MI:SS'),
'S', 'OPE');
COMMIT;

ANEXO 9. Código insert cruce nota de crédito con factura

Insert into INTER.BANC_PAGOS_DAT

(SECU_BANC_PAGO, CODI_ADMI_EMPR_FINA,
CODI_ADMI_PUNT_VENT, CODI_INVE_TIPO_DOCU,
CODI_INVE_DOCU, CODI_ADMI_MONE, MONT_BANC_PAGO,
CODI_ADMI_EMPR_FINA_2, CODI_ADMI_PUNT_VENT_2,
CODI_INVE_TIPO_DOCU_2, CODI_INVE_DOCU_2,
CODI_ADMI_MONE_2, MONT_BANC_PAGO_2, USER_CREA,
FECHA_CREA, CODI_SUCU_BANC_PAGO)

Values

(1000059974, '00001', '101', 'NCINT', 100000346, '1', 3268.21, '00001', '101',
'FACTU', 100046248, '1', 3268.21, 'JPAREJA', TO_DATE('03/06/2013
16:17:22', 'MM/DD/YYYY HH24:MI:SS'), '10');

Insert into INTER.BANC_PAGOS_DAT

(SECU_BANC_PAGO, CODI_ADMI_EMPR_FINA,
CODI_ADMI_PUNT_VENT, CODI_INVE_TIPO_DOCU,
CODI_INVE_DOCU, CODI_ADMI_MONE, MONT_BANC_PAGO,
CODI_ADMI_EMPR_FINA_2, CODI_ADMI_PUNT_VENT_2,
CODI_INVE_TIPO_DOCU_2, CODI_INVE_DOCU_2,
CODI_ADMI_MONE_2, MONT_BANC_PAGO_2, USER_CREA,
FECHA_CREA, CODI_SUCU_BANC_PAGO)

Values

(1000059975, '00001', '101', 'NCINT', 100000346, '1', 254.71, '00001', '101',
'FACTU', 100046249, '1', 254.71, 'JPAREJA', TO_DATE('03/06/2013 16:17:28',
'MM/DD/YYYY HH24:MI:SS'), '10');

Insert into INTER.BANC_PAGOS_DAT

(SECU_BANC_PAGO, CODI_ADMI_EMPR_FINA,
CODI_ADMI_PUNT_VENT, CODI_INVE_TIPO_DOCU,
CODI_INVE_DOCU, CODI_ADMI_MONE, MONT_BANC_PAGO,
CODI_ADMI_EMPR_FINA_2, CODI_ADMI_PUNT_VENT_2,
CODI_INVE_TIPO_DOCU_2, CODI_INVE_DOCU_2,
CODI_ADMI_MONE_2, MONT_BANC_PAGO_2, USER_CREA,
FECHA_CREA, CODI_SUCU_BANC_PAGO)

Values

(1000059976, '00001', '101', 'NCINT', 100000346, '1', 473, '00001', '101',
'FACTU', 100046250, '1', 473, 'JPAREJA', TO_DATE('03/06/2013 16:17:37',
'MM/DD/YYYY HH24:MI:SS'), '10');

Insert into INTER.BANC_PAGOS_DAT

(SECU_BANC_PAGO, CODI_ADMI_EMPR_FINA,
CODI_ADMI_PUNT_VENT, CODI_INVE_TIPO_DOCU,
CODI_INVE_DOCU, CODI_ADMI_MONE, MONT_BANC_PAGO,
CODI_ADMI_EMPR_FINA_2, CODI_ADMI_PUNT_VENT_2,
CODI_INVE_TIPO_DOCU_2, CODI_INVE_DOCU_2,
CODI_ADMI_MONE_2, MONT_BANC_PAGO_2, USER_CREA,
FECHA_CREA, CODI_SUCU_BANC_PAGO)

Values

(1000059977, '00001', '101', 'NCINT', 100000346, '1', 1677.5, '00001', '101',
'FACTU', 100046251, '1', 1677.5, 'JPAREJA', TO_DATE('03/06/2013 16:17:54',
'MM/DD/YYYY HH24:MI:SS'), '10');

Insert into INTER.BANC_PAGOS_DAT

(SECU_BANC_PAGO, CODI_ADMI_EMPR_FINA,
CODI_ADMI_PUNT_VENT, CODI_INVE_TIPO_DOCU,
CODI_INVE_DOCU, CODI_ADMI_MONE, MONT_BANC_PAGO,
CODI_ADMI_EMPR_FINA_2, CODI_ADMI_PUNT_VENT_2,
CODI_INVE_TIPO_DOCU_2, CODI_INVE_DOCU_2,
CODI_ADMI_MONE_2, MONT_BANC_PAGO_2, USER_CREA,
FECHA_CREA, CODI_SUCU_BANC_PAGO)

Values

(1000059978, '00001', '101', 'NCINT', 100000346, '1', 44.75, '00001', '101',
'FACTU', 100046252, '1', 44.75, 'JPAREJA', TO_DATE('03/06/2013 16:18:03',
'MM/DD/YYYY HH24:MI:SS'), '10');

COMMIT;

ANEXO 10. Inserción empleados en server producción

The screenshot displays the 'Capital' HR system interface. The title bar reads 'Capital >> MODULO DE NOMINA - [Maestro de Personal - NOMI0120]'. The menu bar includes 'Acción', 'Edición', 'Consulta', 'Bloque', 'Registro', 'Ayuda', and 'Ventana'. The toolbar contains various icons for file operations and navigation. The main window has a header with the 'CAPITAL' logo, 'TEVCOL ROLES', 'PRINCIPAL', 'INTER', and a date '2013/05/29'. Below the header are tabs for 'Datos Básicos', 'Datos Comunes', 'Nómina', 'Seguro Social', 'Cargas Familiares', and 'Otros'. The 'Datos Básicos' tab is active, showing a form for employee data. The form includes fields for 'Código del Empleado' (100005), 'Tipo de Identificación' (CEDULA DE ID), 'Identificación' (1701458224), 'Ruc', 'Código Tarjeta', 'Libreta Militar', 'Fecha Caduc. Tarjeta', 'Apellido Paterno' (MEDINA), 'Apellido Materno' (MEDINA), 'Primer Nombre' (AURA), 'Segundo Nombre' (MARIA IDELGARDE), 'Nacionalidad' (Ecuatoriana), 'Estado Civil' (Divorciado), 'Sexo' (Femenino), 'Fecha de Nacimiento' (1946/06/12), 'Pais Nacimiento', 'Ciudad de Nacimiento' (QUITO), 'Provincia Nacimiento' (PICHINCHA), and 'Parroquia Nacimiento'. There are also empty fields for 'Servicio Militar' and a 'Fotografía' section with a large empty box. The 'Firma' section also has an empty box. The bottom right corner of the form area says 'TEVCOL S.A.'.

ANEXO 11. Inserción empleados en server replica

Capital >> MODULO DE NOMINA - [Maestro de Personal - NOMI0120]

Acción Edición Consulta Bloque Registro Ayuda Ventana

TEVCOL REPLICA PRINCIPAL INTER 2013/05/29

Datos Básicos | Datos Comunes | Nómina | Seguro Social | Cargas Familiares | Otros

Código del Empleado: 100005
 *Tipo de Identificación: CEDULA DE ID
 *Identificación: 1701458224
 Ruc:
 Código Tarjeta:
 Libreta Militar:
 Fecha Caduc. Tarjeta:
 *Apellido Paterno: MEDINA
 *Apellido Materno: MEDINA
 *Primer Nombre: AURA
 *Segundo Nombre: MARIA IDELGARDE
 *Nacionalidad: EC Ecuatoriana
 *Estado Civil: DI Divorciado
 *Sexo: F Femenino
 *Fecha de Nacimiento: 1946/06/12
 *Pais Nacimiento:
 *Ciudad de Nacimiento: QUITO
 *Provincia Nacimiento: PICHINCHA
 *Parroquia Nacimiento:
 Servicio Militar:

Fotografía

Firma

TEVCOL S.A.

Anexo 12. Insercion asiento contable en servidor de producción

Capital >> MODULO DE CONTABILIDAD - [Mantenimiento de Lotes - CONTAB0500]

Acción Edición Consulta Bloque Registro Ayuda Ventana

TEVCOL NIIF PRINCIPAL INTER 2013/05/29

Lotes | Asientos

Lote: 100096216 Asiento: 1004001757 F. Ingreso: 2013/04/30 Cierre Anual

Tipo: N-DIARIO Documento: No.Doc.: Ingresado Autorizado Rechazado

Concepto: ROL DE PAGOS ABRIL 2013 RUIO Cuadrado

Proceso: No Mayorizado Mayorizado Error al Mayorizar

Detalle de Asiento

Cuenta Contable	Referencia	Detalle	Moneda Nacional		Moneda Ext
			Debe	Haber	
1010101001 CAJA GENERAL			0.01	0.00	
1010101001 CAJA GENERAL			0.00	0.01	
6010101001 SUELDOS	SUELDO		9,043.43	0.00	
6010101001 SUELDOS	RELIQ.		3,028.29	0.00	
6010101002 HORAS EXTRAS			510.44	0.00	
6010101003 SUBSIDIO FAMILIAR	SUB.FAM.		85.50	0.00	
			615,507.10	615,507.10	

NIIF 2011

Anexo 13. Inserción asiento contable en servidor de replica

Capital >>> MODULO DE CONTABILIDAD - [Mantenimiento Plan de Cuentas Base - CONTAB0050]

Acción Edición Consulta Bloque Registro Ayuda Ventana

TEVCOL REPLICA PRINCIPAL INTER 2013/05/29

Lotes Asientos

Lote: 100096216 Asiento: 1004001757 F. Ingreso: 2013/04/30 Cierre Anual

Tipo: N-DIARIO Documento: No.Doc.: Ingresado
 Autorizado
 Rechazado

Concepto: ROL DE PAGOS ABRIL 2013 RUIO

Proceso: No Mayorizado Mayorizado Error al Mayorizar Cuadrado

Detalle de Asiento

Cuenta Contable	Referencia	Detalle	Moneda Nacional		Moneda Ext
			Debe	Haber	Det
1010101001 CAJA GENERAL			0.01		
1010101001 CAJA GENERAL				0.01	
6010101001 SUELDOS	SUELDO		9,043.43		
6010101001 SUELDOS	RELIQ.		3,028.29		
6010101002 HORAS EXTRAS			510.44		
6010101003 SUBSIDIO FAMILIAR	SUB.FAM.		85.50		
			615,507.10	615,507.10	

NIIF 2011

Anexo 14. Inserción cta contable en servidor de producción

Capital >>> MODULO DE CONTABILIDAD - [Mantenimiento Plan de Cuentas Base - CONTAB0050]

Acción Edición Consulta Bloque Registro Ayuda Ventana

TEVCOL NIIF PRINCIPAL INTER 2013/05/29

Plan Base: 100 PB TEVCOL Formato: 1 TEVCOL FORMATO 1

Tipo Empresa: Real Ficticia Empresa: 00001 TRANSPORTADORA ECUATORIANA DE VAL

Activo Congelado Temporal Moneda: 1 DOLAR

6.01.01.01 GASTOS PERSONAL A
 6.01.01.01.001 SUELDOS
 6.01.01.01.002 HORAS EX1
 6.01.01.01.003 SUBSIDIO F
 6.01.01.01.004 SUBSIDIO A
 6.01.01.01.005 BONO RESI
 6.01.01.01.006 BONO ESPE
 6.01.01.01.007 BONO EFICI
 6.01.01.01.008 BONO EXTF
 6.01.01.01.009 BONO TRAT
 6.01.01.01.010 BONO VOLL
 6.01.01.01.011 SERVICIOS
 6.01.01.01.012 COMISIONE
 6.01.01.01.013 BONOS Y G
 6.01.01.01.014 BONOS NO
 6.01.01.01.015 APORTE PA

Nivel Cuenta Padre Cuenta
 6 6010101 001

Descripcion Corta SUELDOS Descripcion Larga SUELDOS

Descripcion en Inglés SUELDOS

Clase Detalle X Tipo de Proceso MIXTO

Tipo de Cuenta 6 GASTO ADMINISTRATIVO Naturaleza 0 DEUDOR

Moneda 1 DOLAR Centro Costo SI Centro Costo Auxiliar SI

Estado Activo? Formato: 6.01.01.01.001

Plan de Ctas Super (Ingresar con formato)

ANEXO 15. Inserción cta contable en servidor de replica

Capital >> MODULO DE CONTABILIDAD - [Mantenimiento Plan de Cuentas Base - CONTAB0050]

Acción Edición Consulta Bloque Registro Ayuda Ventana

TEVCOL REPLICA PRINCIPAL INTER 2013/05/29

Plan Base: 100 PB TEVCOL Formato: 1 TEVCOL FORMATO 1

Tipo Empresa: Real Ficticia Empresa: 00001 TRANSPORTADORA ECUATORIANA DE VAL

Activo Congelado Temporal Moneda: 1 DOLAR

6.01.01.01 GASTOS PERSONAL A
 6.01.01.01.001 SUELDOS
 6.01.01.01.002 HORAS EX1
 6.01.01.01.003 SUBSIDIO F
 6.01.01.01.004 SUBSIDIO A
 6.01.01.01.005 BONO RESF
 6.01.01.01.006 BONO ESPE
 6.01.01.01.007 BONO EFICI
 6.01.01.01.008 BONO EXTF
 6.01.01.01.009 BONO TRAI
 6.01.01.01.010 BONO VOLL
 6.01.01.01.011 SERVICIOS
 6.01.01.01.012 COMISIONE
 6.01.01.01.013 BONOS Y G
 6.01.01.01.014 BONOS NO
 6.01.01.01.015 APORTE PA

Nivel: 5 Cuenta Padre: 6010101 Cuenta: 001

Descripcion Corta: SUELDOS Descripcion Larga: SUELDOS

Descripcion en Inglés: SUELDOS

Clase: Detalle Tipo de Proceso: MIXTO

Tipo de Cuenta: GASTO ADMINISTRATIVO Naturaleza: DEUDDR

Moneda: DOLAR Centro Costo: SI Centro Costo Auxiliar: SI

Estado ¿Activo?: Formato: 6.01.01.01.001

Plan de Ctas Super: (Ingresar con formato)

ANEXO 16. Registro de activo fijo en servidor de producción

Capital >> MODULO DE ACTIVOS FIJOS - [Mantenimiento de Activos - ACTI0042]

Acción Edición Consulta Bloque Registro Ayuda Ventana

TEVCOL NIF PRINCIPAL INTER 2013/05/29

Sucursal: QUITO Servicio: SIB SERVICIO INTEGRAL DE BOVEDA

Código: 1000007208 Cod. Barras: Nombre: ZONA TECNOLOGICA F106031 LECTORBARRA 5 12355000502260

Nivel: TDECP EQUIPOS DE COMPUTACION TD-ECP Cod Ant.: Estado: A ACTIVO

Proveed.: 100005082 ZONA TECNOLOGICA Factura: 106031 Fecha Compra: 2013/04/24

Custodio: Fecha Recep.: 2013/04/24 Moneda: 1 DOLAR

Clasifica.: SubClasif.: Descr.: ZONA TECNOLOGICA F106031 LECTORBARRA 5

Caracac.: ZONA TECNOLOGICA F106031 LECTORBARRA 5 Desc. U. Física: ZONA TECNOLOGICA F106031 LECTORBARRA 5

Ub. Física: Condición: Las características con * son obligatorias

Características Generales Valor/Depreción de la Compra Equipos de Computación

Marca: * 4 MOTOROLA

Modelo: SYMBOL

Serie: S1235500502260

Color:

Path Archivo:

ANEXO 23. Inserción de un proveedor en replica

Capital >> MODULO DE BANCOS - [Proveedores - INI_3]

Acción Edición Consulta Bloque Registro Ayuda Ventana

TEVCOL REPLICA PRINCIPAL INTER 201

Información de Proveedores Proveedores por Empresa Productos por Proveedor

Código: 100005078 Persona Natural? Tipo Ident: RUC RUC #Ident: 0916649171001

Tipo Proveedor: NAC NACIONAL Razón Social: ALFONSO VILLACIS ELINA JACKELINE

Denominación: DELICIAS COSTEÑAS Estado: A ACTIVO
 Comercial: Fecha Alta: 2013/04/01 Moneda: 1 DOLAR Fecha Baja:

Representante Legal: ALFONSO VILLACIS ELINA JACKELINE Identificación:
 Representante Ventas: Identificación:

Ubicación Geográfica: EPIQU QUITO Actividad: REST RESTAURANTES
 Calle: GUSTAVO DARQUEA OE-137 Y VERSALLES Número:
 E-Mail: Teléfono (1): 0987459087 Teléfono (2):
 Teléfono (3): Fax: Contacto:

Contribuyente Especial? Auto. Imprenta: 7751 Auto Contribu.: 1112289275 Fecha Cadu.: 2014/02/05 Lleva Contabilidad Retenedor Fuente
 RISE Fecha Inicial de Cont.Esp.: No Resolución: Se le retiene IVA Se le retiene Fuente

Información Cash Management

Tipo de Cuenta: Número de Cuenta:
 Institución Financiera:

No.	Factura Inicial	Factura Final	Autoriz. Imprenta	Nume Auto Contri	Fecha Caducidad

CASH Y NIIF 2013

ANEXO 24. Inserción de un cliente en producción

Capital >> MODULO DE CUENTAS POR COBRAR - [Clientes]

Acción Edición Consulta Bloque Registro Ayuda Ventana

TEVCOL NIIF PRINCIPAL

Cientes Cientes por Empresa

Código: 159 COOPERATIVA DE AHORRO Y CREDITO VICENTINA MANUEL GODOY

Tipo Ident: RUC RUC Identificación: 1190068389001 Institución Pública
 Tipo: NORM CLIENTE NORMAL Tipo de Cliente: Juridico Cliente Confidencial
 Nombre: Segmento: Contribuyente Especial
 Den. Comercial: COOPERATIVA DE AHORRO Y CREDITO VICENTINA MANUEL GODOY Lleva Contabilidad
 Retenedor Fuente

Moneda: 1 DOLAR Estado: A ACTIVO Genera Estado de Cuenta?
 Fecha Alta: 2006/03/03 Fecha Baja:

Agentes
 Agente de Cobro: Agente de Venta:

Localización
 Ubicación geográfica: ECAZCU CUENCA
 Dirección: BOLIVAR 10-56 Y AZUAY Número:
 Teléfono (1): Teléfono (2):
 Fax: E-Mail:

Agente de Cuenta TEVCOL
 Agente Comercial 0 NO ASIGNADO

ANEXO 25. Inserción de un cliente en replica

Capital >> MODULO DE CUENTAS POR COBRAR - [Clientes]

Acción Edición Consulta Bloque Registro Ayuda Ventana

TEVCOL REPLICA PRINCIPAL

Cientes Cientes por Empresa

Código: 159 COOPERATIVA DE AHORRO Y CREDITO VICENTINA MANUEL GODOY

Tipo Idem: RUC RUC Identificación: 1190068389001

Tipo: NORM CLIENTE NORMAL Tipo de Cliente: Jurídico

Nombre Segmento

Den. Comercial: COOPERATIVA DE AHORRO Y CREDITO VICENTINA MANUEL GODOY

Institución Pública
 Cliente Confidencial
 Contribuyente Especial
 Lleva Contabilidad
 Retenedor Fuente

Moneda: 1 DOLAR Estado: A ACTIVO Genera Estado de Cuenta?

Fecha Alta: 2006/03/03 Fecha Baja:

Agentes
Agente de Cobro: Agente de Venta:

Localización
Ubicación geográfica: ECAZCU CUENCA
Dirección: BOLIVAR 10-56 Y AZUAY Número:
Teléfono (1): Teléfono (2):
Fax: E-Mail:

Agente de Cuenta TEVCOL
Agente Comercial: NO ASIGNADO