

41132  
64



# UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES  
ARAGÓN

**“ADMINISTRACIÓN DEL SERVIDOR DE BASE DE DATOS EN INFORMIX APLICADO AL SISTEMA INTEGRAL DE PERSONAL DE LA SECRETARÍA DE EDUCACIÓN PÚBLICA (SIAPSEP)”**

**T E S I S**  
QUE PARA OBTENER EL TÍTULO DE:  
**INGENIERO EN COMPUTACIÓN**  
P R E S E N T A N:  
**ALEJANDRO SOLIS MELÉNDEZ**  
CUAUHTÉMOC EDUARDO MENDOZA MORALES

ASESOR:  
ING. BLANCA ESTELA CRUZ LUEVANO

MÉXICO

TESIS CON  
FALLA DE ORIGEN

2003

A



Universidad Nacional  
Autónoma de México



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

**TESIS  
CON  
FALLA DE  
ORIGEN**



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO  
ESCUELA NACIONAL DE ESTUDIOS PROFESIONALES ARAGÓN  
SECRETARÍA ACADÉMICA

**M. en C. JESÚS DÍAZ BARRIGA ARCEO**  
**Jefe de la Carrera de Ingeniería en Computación,**  
**Presente.**

En atención a la solicitud de fecha 20 de febrero del año en curso, por la que se comunica que los alumnos ALEJANDRO SOLIS MELENDEZ y CUAUHTEMOC EDUARDO MENDOZA MORALES, de la carrera de Ingeniero en Computación, han concluido su trabajo de investigación intitulado "ADMINISTRACIÓN DEL SERVIDOR DE BASE DE DATOS EN INFORMIX APLICADO AL SISTEMA INTEGRAL DE PERSONAL DE LA SECRETARÍA DE EDUCACIÓN PÚBLICA (SIAPSEP)", y como el mismo ha sido revisado y aprobado por usted, se autoriza su impresión; así como la iniciación de los trámites correspondientes para la celebración del Examen Profesional.

Sin otro particular, reitero a usted las seguridades de mi atenta consideración.

**Atentamente**  
**"POR MI RAZA HABLARÁ EL ESPÍRITU"**  
**San Juan de Aragón, México, 20 de febrero del 2003**  
**EL SECRETARIO**

**Lic. ALBERTO IBARRA ROSAS**

C p Asesor de Tesis.  
C p Interesado.

AIR/vr

TESIS CON  
FALLA DE ORIGEN

**"ADMINISTRACIÓN DEL SERVIDOR DE BASE DE DATOS EN INFORMIX  
APLICADO AL SISTEMA INTEGRAL DE PERSONAL DE LA SECRETARÍA  
DE EDUCACIÓN PÚBLICA (SIAPSEP)"**

TESIS CON  
FALLA DE ORIGEN

## **AGRADECIMIENTOS**

### **A mis padres**

Rosa Morales de Mendoza  
Cauhtémoc Eduardo Mendoza Rojo

Por su apoyo y amor brindado durante todos estos años de mi vida, por sus atinados consejos, quiero con ustedes compartir este logro ya que sin ustedes no lo hubiera alcanzado. "Gracias por ser mis Padres".

### **A mis hermanos**

Fernando  
Laura Lilia  
Nancy Guadalupe  
David Eduardo

Por todos los momentos que hemos estado juntos, por su ayuda y confianza, ya que no hay nada más grande y fuerte que ser una familia unidad y feliz,

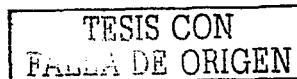
### **A mis familiares**

Por ser gente que ha llegado a formar parte de mi familia y de manera muy especial a mi cuñado César y a mis sobrinos César y Judith +, de los cuales he aprendido muchas cosas para seguir adelante.

### **A Alejandro**

Por su apoyo para culminar este logro y poderlo compartir conjuntamente, por su amistad, esfuerzo, conocimientos y dedicación puesta.

### **A mis amigos y compañeros**



CUAUHTÉMOC

### **A Dios:**

Por que gracias a él he podido llegar a cubrir cada una de las metas que me he forjado, por que día a día ilumina mi vida y me va guiando, por ser mi mas grande ejemplo a seguir.

### **A mi padre**

Paulo Solis Calcanas

Quien con su apoyo y consejos inculcó en mi el valor que tiene la vida, y lo difícil que resulta llegar a ser alguien, por que gracias a él cubro una etapa mas en mi vida, gracias a sus desvelos, al sudor de su frente, gracias a cada sacrificio que hizo por mi, gracias por que ha sido mi mas grande maestro.

### **A mi madre**

Sofía Meléndez Escobar

Por enseñarme tantas cosas, por todos sus buenos consejos que siempre me ha inculcado, en una palabra por ser quien me dio la vida.

### **A mis hermanos**

Juanita  
Cruz María  
Chuy

Por que gracias a ellos hemos fortalecido los lazos de unión que existen en nuestra familia. Por que siempre me han apoyado cuando los he buscado, por que han sido mis mejores amigos en toda mi vida..

### **A mi novia**

Gaby

Gracias por ser mi fuente de inspiración, por estar siempre ahí cuando te necesito, por darme aliento, por ser parte de mi vida, por ser mi mas grande amiga, excelente novia y compañera.

### **A mis abuelos**

Papá Chuy y Mama Lupe

Por que gracias a sus sabios consejos he sabido afrontar todos aquellos obstáculos que se me han ido presentado, gracias por que el ha sido uno de mis mas grandes maestros en la vida. Por el amor y afecto que siempre nos ha transmitido, por los consejos que le ha inculcado a mi madre para poder educarnos, gracias.

### **A mis maestros**

Por inculcarme el valor de ser universitario, por cada conocimiento que me han transmitido, por ser parte muy importante de esta gran meta que hoy veo realizada.

### **A Cuauhtémoc**

Por todo el trabajo que hemos venido realizando durante todo este tiempo juntos, por esa amistad tan sólida que hemos mantenido, por el apoyo brindado para el logro de esta meta que hoy hemos cumplido, gracias.

### **A Gilberto**

Por ser uno de mis mejores amigos, por estar conmigo en toda situación, por tanta convivencia, por los ánimos que me daba cuando era estudiante gracias.

### **A mis amigos y familiares**

ALEJANDRO

11

### **A nuestra asesora**

Queremos brindar nuestro más grande agradecimiento a la profesora Blanca Estela Cruz Luevano por todo el apoyo que nos brindó durante todo este tiempo para la realización de este proyecto, por todos los consejos que nos aportó, por todos los conocimientos que nos transmitió cuando fuimos estudiantes.

### **A nuestra Universidad Nacional Autónoma de México**

Por ser nuestro templo del saber, por permitirnos desarrollarnos espiritual y tecnológicamente, por formar parte de tan prestigiada Institución.

### **A nuestra Escuela Nacional de Estudios Profesionales Plantel Aragón**

Por ser parte fundamental en nuestro desarrollo profesional, por todos los conocimientos adquiridos durante nuestra estancia en ella, por la satisfacción que nos deja el decir que formamos parte de ella de igual forma el agradecimiento a los profesores de esta institución por los conocimientos transmitidos durante esos años de estudio.



## **PREFACIO**

**Hoy cumplimos una de nuestras más grandes metas en la vida.**

**Hoy no concluimos nuestra carrera, puesto que esta apenas inicia**

**y nos reclama el compromiso de ejercer con respeto y valor**

**Hoy podemos decir con orgullo que somos INGENIEROS EN COMPUTACIÓN de la UNAM**

**Pero el conocimiento avanza,**

**y nosotros debemos marchar a su paso**

**Al estudio consciente**

**frente del saber. . .**

# INDICE

<b>INTRODUCCIÓN</b>	<b>1</b>
<b>I.- CONCEPTOS GENERALES</b>	
I.1 Sistema Operativo Unix	5
I.2 Bases de Datos	12
I.3 Informix Dinamic Server	20
I.4 Sistema Integral de Administración de la Secretaría de Educación Pública	26
<b>II.- SERVIDORES DE BASE DE DATOS</b>	
II.1 Tipos de Administradores	34
II.2 Características del Servidor Dinámico	35
II.3 Arquitectura Cliente/Servidor y tipos de aplicaciones	36
II.4 Arquitectura Escalable Dinámica	38
II.5 Arquitectura de Procesamiento en Paralelo	39
II.6 Escalabilidad	41
II.7 Alto rendimiento	41
II.7.1 Manejo de discos o particiones crudas	41
II.7.2 Manejo Dinámico de la Memoria Compartida	42
II.7.3 Asignación dinámica de hilos	42
II.7.4 Fragmentación y Paralelismo	42
II.8 Procesadores Virtuales	46
II.9 Extensibilidad	48
II.10 Múltiple Residencia	49
II.11 Tareas de Administración de bases de datos	54
<b>III- CONFIGURACION REQUERIDA PARA LA IMPLANTACIÓN DEL SISTEMA SIAPSEP EN INFORMIX</b>	
<b>III.1 SISTEMA OPERATIVO UNIX</b>	
III.1.1 Modificación de los Parámetros del Kernel de Unix	60
III.1.2 Asignación de espacio en disco	64
III.1.3 Creación de Dispositivos Crudos o Archivos sin buffer	65
III.1.4 Creación de nombres de dispositivos estándar	66
III.1.5 Asignación de permisos, propietarios y grupos	67

### III.2 INFORMIX DINAMIC SERVER

III.2.1	Instalación del Servidor de Base de Datos	73
III.2.2	Instalación del ESQ/C	75
III.2.3	Definición del tipo de comunicación y servicio entre el servidor y el manejador de Base de Datos.	76
III.2.4	Alta del Servicio en el archivo Services	83
III.2.5	Ambientación del usuario de informix.	86
III.2.6	Modificación de los parámetros de Informix	
III.2.6.1	Disco	92
III.2.6.2	Memoria Compartida (Shared Memory)	93
III.2.6.3	Ejecución y Afinación (tuning and performance)	97
III.2.6.4	Replicación de datos	99
III.2.6.5	Diagnóstico	100
III.2.6.5	Consultas paralelas a Bases de Datos	101
III.2.7	Áreas Físicas y Lógicas	
III.2.7.1	Creación de los Dbspaces y Chunks	106
III.2.7.2	Creación de Archivos Logs	108

### IV.- OPERACIÓN Y MANTENIMIENTO DEL SERVIDOR DE BASE DE DATOS

IV.1	Modos de operación	
IV.1.1	Cambios de estatus	117
IV.1.2	Inicialización del Servidor de BD	122
IV.2	Herramientas de consulta	126
IV.4	Herramientas de modificación	143
IV.3	Herramientas de respaldo y restauración de Base de Datos	155
IV.5	ISA (Informix Server Administrator)	190

### V.- SEGURIDAD

V.1	Seguridad en la Red de Datos	193
V.2	Sistema Operativo Unix	203
V.3	Servidor de Base de Datos	212
V.4	SIAPSEP	243

CONCLUSIONES	250
--------------	-----

GLOSARIO	253
----------	-----

BIBLIOGRAFIA	257
--------------	-----

## INTRODUCCIÓN

En la actualidad, la necesidad de modernizar todos los procesos que aún son realizados de forma manual, la modernización de aquellos demasiado obsoletos, o bien, permanecer a la vanguardia con la constante actualización, cualquiera que sea el caso, el mercado del Software ofrece a los usuarios una gran variedad de alternativas para la resolución de sus problemas, para ello es necesario tener bases de conocimiento firmes para poder explotar al máximo dicho software, en este caso se maneja el tópicó de bases de datos ya que es uno de los pilares para el buen manejo de la información.

Dentro de los objetivos principales para el desarrollo de este trabajo de investigación es analizar la importancia del manejo de la información en la base de datos y las formas de cómo llevar a cabo la administración, así como proporcionar la información necesaria para llevar a cabo la administración del servidor de base de datos Informix. Identificar todos los factores que interactúan con una base de datos para su funcionamiento, así como identificar las características y conocimientos que debe tener un administrador de servidores de base de datos. Realizar el proceso de instalación y configuración de Informix. Detallar las definiciones de cada uno de los parámetros de configuración de Informix. Aplicar los conceptos de administración al Sistema SIAPSEP (Sistema ya creado) como parte aplicativa, identificar las herramientas en cuanto a mantenimiento y seguridad, que el servidor de base de datos Informix proporciona para mantener la integridad de la información, identificar la forma en como cada uno de los factores que interactúan con una base de datos proporcionan y refuerzan la seguridad (Sistema Operativo, Red y la propia aplicación).

Las bases de datos son una herramienta útil en el crecimiento de cualquier organización, el cúmulo y control de la información permite conocer índices y puntos para la organización, la información está disponible en momentos precisos y claves para el desarrollo de la misma, para la toma de decisiones debe ser oportuna y confiable. El obtener este punto implica implantar políticas y estrategias respecto a la información, y el Administrador de la base de datos es quien debe poner en práctica estas decisiones.

Las ventajas principales de las bases de datos como son de todos conocidas involucran la recuperación y manejo rápido y eficiente de la información, el control de la redundancia, evitar la inconsistencia de la información y el tener una mayor integridad de ella. Aunado a lo anterior podemos recalcar el poder de las aplicaciones *distribuidas* y los sistemas Cliente - Servidor.

En una base de datos la información se encuentra en diversos archivos (tablas) y a su vez estos pueden alojarse en diversos dispositivos de almacenamiento (discos), incluso en diferentes servidores, sin embargo la información se maneja como un todo, de hecho se dice que la información es integrada, la mayor ventaja de esto es el compartir información, dando como resultado que varios usuarios pueden acceder al mismo tiempo a la base de datos, incluso desde diferentes terminales y la transparencia del sistema evita que el usuario perciba la trascendencia y alcance de la aplicación.

Otro de los factores que sin duda alguna han ayudado al desarrollo de las bases de datos son las nuevas tecnologías de almacenamiento y acceso a la información a través de diferentes medios, así como la madurez de los sistemas operativos que han creado bases sólidas para este tipo de aplicaciones.

Este trabajo de investigación contiene información importante sobre la administración de Informix Dynamic Server 2000 proporcionando temas sobre como instalar, configurar, administrar y el uso de servidores de bases de datos, así como características de conceptos y procedimientos para realizar las tareas administrativas de un servidor de base de datos aplicado a un sistema de nómina.

Dentro de este documento informativo se manejan tres tipos de tópicos iniciando con el sistema operativo UNIX, como segundo punto y siendo la parte medular de este trabajo de investigación es el administrar un manejador de base de datos Informix y finalmente la parte aplicativa que es el sistema de nómina de la SEP.

En lo que se refiere al sistema de manejo de la nómina, lleva el nombre de Sistema Integral de Administración de la Secretaría de Educación Pública, es un sistema que actualmente está utilizando Informix Dynamic Server como manejador de base de datos y con sistema operativo unix, permitiendo ejemplificar y aplicar los conceptos de administración de base de datos relacionales presentados en este tema de investigación.

La mayoría de los productos comerciales de bases de datos están enfocados al modelo relacional, de hecho este modelo representa la tendencia dominante en el mercado actual de estos sistemas y se dice que constituye el avance más importante en el campo de los estudios sobre las mismas. Esta es una de las razones por las cuales gran número de sistemas están basados en él, además de que la tecnología relacional cuenta con bases sólidas en ciertos aspectos de las matemáticas, esto se refiere, a que la parte manipulativa del modelo relacional está sustentado en el álgebra relacional. Esta ofrece un conjunto de operaciones explícitas como unión, reunión, restricción, intersección, proyección, producto, división y diferencia, que sirven para indicar al sistema como constituir alguna relación deseada partiendo de las relaciones dadas en la base de datos. De hecho, el álgebra relacional en su descripción se acerca a un lenguaje de programación, la cual se sustenta matemáticamente en la lógica de conjuntos. Existe otra relación similar entre el álgebra relacional y el SQL, ciertamente dentro de la historia del modelo relacional el SQL ha tenido un papel fundamental en la evolución de éste. En 1985 aparece el Manejo de bases de datos por Ejemplos (QBE) y en 1986 el Instituto Nacional Americano de Normas (ANSI) y la Organización Internacional para la Estandarización (ISO) coinciden en adoptar el dialecto SQL como interfaz "oficial" para sistemas relacionales.

La integridad de la información es un punto básico en los sistemas de bases de datos, que nos garantiza que la información que acumulamos es confiable. Existen reglas de integridad que son específicas a la información de la base de datos. Estas implican que lo que trata de hacer el usuario es correcto, es decir que los datos introducidos tengan una relación con el mundo real (que sean válidos). Para esto se especifican rangos de valores que puedan aceptar los datos o si estos se obtendrán de una lista de valores y cuai debe ser esa lista. Esta integridad la marca la naturaleza de la información que se maneja en el sistema. Por otro lado el modelo relacional incluye dos reglas generales de integridad en el sentido en que se aplican a toda la base de datos. Estas dos reglas se refieren a las claves primarias y a las claves ajenas.

TESIS CON  
FALLA DE ORIGEN

Otro punto que el administrador realiza cotidianamente, es el mantenimiento del servidor de base de datos, parte muy importante para evitar errores y corregirlos de manera anticipada, considerados dentro de este punto los respaldos y restauraciones ya sea de una base de datos o de todo el servidor. Los manejadores de bases de datos proporcionan herramientas para cubrir estas necesidades, por lo que el administrador debe de conocerlas y explotárlas al 100% para asegurar el buen funcionamiento del servidor.

Cabe recalcar que otro punto fundamental dentro de los sistemas es la seguridad. Entendiendo por seguridad de la base de datos la protección de la información contra una revelación, alteración o destrucción no autorizada. De esta forma el sistema debe estar al tanto de ciertas restricciones que no deben ser violadas por los usuarios. Para esto debe existir un catálogo de usuarios y su correspondiente alcance. El administrador de la base de datos debe asignar los permisos correspondientes a cada grupo de usuarios, con esto se restringen derechos por cada grupo dependiendo del nivel y responsabilidad de cada uno de ellos dentro de la empresa, existen restricciones de aspecto legal, social o ético que deben ser aplicados en las políticas de seguridad del sistema e implantados en el alcance a los usuarios, de esta forma cada grupo ve el sistema desde un punto diferente, así se protege la información dentro del mismo.

Pero también se debe contemplar la seguridad de la información fuera del sistema de base de datos por eso esta administración va de la mano con la administración de los recursos de la red. Se puede tener un sistema de base de datos en un ambiente muy seguro, pero si fuera del sistema se puede copiar y editar la información, de nada serviría, para evitar un incidente se deben planear los directorios en donde se alojará la información y asignar de esta forma derechos sobre ellos a cada usuario en coordinación con el administrador de la red, para evitar un dolor de cabeza se deben plantear políticas a niveles empresa y de sistema ya que la seguridad debe ser global.

Informix se especializa en tecnologías avanzadas para la administración de información, el proceso transaccional altamente escalable, así como un sistema de seguridad confiable.

A lo largo de su historia, la empresa informix ha sido pionera en el área de bases de datos relacionales la arquitectura OLPT y lenguajes de cuarta generación para plataformas Unix, así como en el lanzamiento al mercado de tecnología de bases de datos relacionales con orientación a objetos, tanto para Unix como para Windows NT.

Actualmente, las soluciones de bases de datos de Informix, que se encuentran soportadas por los más importantes proveedores de hardware, desarrolladores de software e integradores de soluciones, incluyen sistemas de alto desempeño para ambientes corporativos muy diversos: desde pequeños grupos de trabajos hasta las más complejas aplicaciones de procesamiento paralelo, entre ellas destacan: servidores de bases de datos y herramientas para el desarrollo de aplicaciones en áreas de procesamiento de transacciones en línea, aplicaciones datawarehouse, manejo dinámico de contenidos Web y sistemas empresariales de computación distribuida.

La empresa Informix fue comprada por la empresa IBM, con la finalidad de reforzar una parte que le faltaba cubrir en el mercado el servidor de Base de Datos Informix, debido a que IBM cuenta con un manejador de Base de Datos denominado DB2 el cual no es muy

demandado dentro del mercado, por lo que la adquisición de Informix viene a complementar todos los productos que esta empresa proporciona, brindando en la actualidad los servidores de Bases de Datos Informix y DB2.

Este trabajo de investigación se encuentra estructurado en los siguientes capítulos. El primer capítulo, describe los conceptos generales inmersos en este trabajo de investigación, en donde básicamente se tocan cuatro puntos que son: el sistema operativo Unix, bases de datos, informix y el sistema SIAPSEP, donde de manera general se darán cada una de las características de dichos tópicos, con la finalidad de ver como interactúan y a su vez su funcionamiento, para brindar un panorama general de los conceptos manejados en este trabajo de investigación.

El segundo tema denominado Servidores de Base de Datos describe las características del servidor de base de datos Informix, los tipos de administrador de acuerdo a la actividad que realizan y las tareas de los mismos.

En el tercer tema se realiza la explicación del proceso de instalación y configuración del servidor de base de datos informix, determinando los pasos a seguir del administrador, aplicándolo al sistema SIAPSEP, considerando la parte de configuración del sistema operativo como son: los servicios, usuarios, permisos etc. para que pueda funcionar Informix sobre plataformas Unix.

En el cuarto tema se mencionan cada una de las herramientas que informix proporciona para el mantenimiento del servidor de base de datos, como son respaldos y restauraciones, revisión de inconsistencias en las bases de datos del servidor, así como los estatus de operación.

En el tema cinco se habla de la seguridad, considerando dos partes que son: sistema operativo y servidor de base de datos. Actualmente en lo que a seguridad se refiere están muy ligados el sistema operativo e Informix ya que uno es el complemento del otro, para poder resguardar la seguridad de ambos. Así mismo se proporcionan códigos del lenguaje ESQL/C para procesos de permisos en las bases de datos.

La última parte son las conclusiones obtenidas del desarrollo del presente trabajo y las referencias de investigación.

TESIS CON  
FALLA DE ORIGEN

**TEMA I****CONCEPTOS GENERALES**

En este capítulo se describen de manera general los puntos relevantes del propósito de este trabajo de investigación, en donde el primer punto hace referencia al Sistema Operativo Unix, que como ya es de todos conocido son programas que controlan los recursos de la computadora y es en esa parte donde se hace la administración y distribución de todos y cada uno de los programas que van a ser ejecutados en el sistema operativo, así como el acceso a los usuarios, esto mediante el monitoreo del Administrador del Sistema Operativo que a su vez es el que va a dar pauta para que el usuario administrador del Servidor de Base de Datos genere las Bases de Datos requeridas, las cuales a su vez necesitan de espacio y configuración para su funcionamiento.

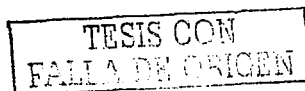
Así mismo este capítulo hace referencia a lo que son Bases de Datos, definidas como el conjunto de datos ordenados en forma de filas y columnas, siendo uno de los tópicos fundamentales de este trabajo de investigación, además de que se presentan las características y forma en como se organiza la información, entre las cuales podemos mencionar la integridad, redundancia y seguridad, todo esto en un entorno de Bases de Datos relacionales, lo anterior se realiza mediante programas denominados Sistemas Manejadores de Base de Datos (DBMS).

Los manejadores de Bases de Datos son herramientas que permiten la administración de los datos, en la actualidad existen varios manejadores, entre los principales se encuentran Informix, Sybase y Oracle, en este trabajo utilizamos el manejador de Informix y se hace una breve descripción y comparación con otros manejadores, dicho lo anterior todo esto se ve plasmado en una aplicación real, la administración del Sistema Integral de Administración de Personal de la SEP.

Este sistema como se hará mención esta sobre una plataforma UNIX y un Manejador de BD Informix.

**I.1 SISTEMA OPERATIVO UNIX**

Durante los últimos veinte años, el sistema operativo UNIX se ha convertido en un sistema operativo potente, flexible y versátil. Sirve como sistema operativo para todo tipo de computadoras, incluyendo las computadoras personales monousuario, estaciones de trabajo de ingeniería, microcomputadoras, minicomputadoras y supercomputadoras.





En sentido estricto, UNIX es el núcleo de un sistema operativo de tiempo compartido: un programa que controla los recursos de una computadora y los asigna entre los usuarios. Permite a los usuarios ejecutar sus programas; controla los dispositivos periféricos (discos, terminales, impresoras y otros) conectados a la máquina; y proporciona un sistema de archivos que administra el almacenamiento a largo plazo de la información tal como programas, datos y documentos.

En un sentido más amplio, "UNIX" abarca no sólo el núcleo, sino que incluye también programas esenciales, entre ellos: compiladores, editores, programas para copiado e impresión de archivos, etc. De igual forma este sistema operativo "UNIX" puede incluir programas desarrollados por usuarios para ser ejecutados en el sistema; por ejemplo, herramientas para preparar documentos, rutinas para análisis estadísticos y paquetes gráficos.

A continuación se describen algunas de las características que hacen a éste sistema operativo muy importante:

- . El sistema está escrito en un lenguaje de alto nivel, haciéndolo fácil de leer, comprender, cambiar y mover a otras máquinas.
- . Tiene una interfaz con el usuario simple que tiene el poder de suministrar los servicios que quiere el usuario.
- . Permite la realización de programas complejos a partir de programas más simples.
- . Usa un sistema de archivos jerárquico que permite un fácil mantenimiento.
- . Usa un formato para los archivos consistente en el flujo de bytes, haciendo que los programas de aplicación sean fáciles de escribir.
- . Provee una simple y consistente interfaz con los dispositivos periféricos.
- . Es un sistema multiusuario y multitarea, cada usuario puede ejecutar varios procesos simultáneamente.
- . Oculta la arquitectura de la máquina del usuario, haciendo fácil el escribir programas que corran en diferentes plataformas de hardware.

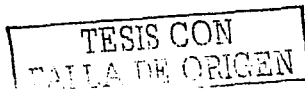
Además de que el sistema operativo y muchos de los programas y comandos están escritos en lenguaje C, UNIX soporta otros lenguajes, incluyendo Fortran, Basic, Pascal, Ada, Cobol, Lisp, Prolog y Java entre otros. UNIX soporta cualquier lenguaje de programación que tenga un compilador o intérprete y una interfaz con el sistema que permita transformar las peticiones del usuario de servicios del sistema, al conjunto estándar de peticiones usadas en el sistema UNIX.

## **Estructura del Sistema Operativo UNIX**

### *Sistema de Archivos*

El sistema de archivos o file system es la forma en el que UNIX gestiona los archivos existentes en el sistema. El sistema de archivos está caracterizado por:

- . Una estructura jerárquica.
- . Un tratamiento consistente de los archivos de datos.
- . Habilidad para crear y borrar archivos.
- . Crecimiento dinámico de los archivos.
- . Protección de los archivos de datos.



### . Tratamiento de los dispositivos periféricos como archivos.

El sistema de archivos está organizado como un árbol, con un único nodo raíz llamado root (se representa con "/"); cada nodo que no es hoja de la estructura del sistema de archivos es un directorio de archivos. Cada archivo está identificado por un nombre de archivo. El nombre de archivo viene dado por un nombre de camino o ruta (path name) que describe como localizarlo en la jerarquía del sistema de archivos. Un nombre de camino es una secuencia de nombres de componentes separados por caracteres "/"; un componente es una secuencia de caracteres que identifica un archivo que está contenido en el componente precedente (directorio). Un camino completo comienza con un carácter "/" y especifica un archivo que puede ser encontrado empezando en el nodo raíz del sistema de archivos.

#### Acceso a los archivos

Desde el punto de vista del acceso a un archivo, existen tres tipos de usuarios a los que se les pueden dar o denegar permisos sobre un archivo:

(u)	user	Propietario del archivo.
(g)	group	Usuarios pertenecientes al grupo del propietario.
(o)	others	Resto de usuarios que no pertenecen al grupo.

La capacidad de un usuario para trabajar con archivos depende del tipo de acceso que tenga a dicho archivo. Los accesos disponibles en UNIX son:

Para un archivo:

Permiso de lectura (r):	Permite ver el contenido del archivo.
Permiso de escritura (w):	Permite cambiar el contenido del archivo.
Permiso de ejecución (x):	Permite ejecutar un archivo (como cualquier orden de UNIX).

Para un directorio:

Permiso de lectura (r):	Permite ver los nombres de los archivos de un directorio. Si se quiere información detallada sobre dichos archivos el directorio tiene que tener el permiso de ejecución para dicho usuario.
Permiso de escritura (w):	Permite cambiar el contenido de dicho directorio; crear nuevos archivos, suprimir los existentes (este último caso depende de los permisos de escritura de los propios archivos).
Permiso de ejecución (x):	Se debe de hablar más bien de permiso de búsqueda ya que permite situarse en dicho directorio y según el resto de los permisos, permitirá crear, borrar, modificar o copiar archivos.

Además de los permisos de acceso rwx para el propietario del archivo, grupo al que pertenece el propietario y resto de usuarios, existen tres permisos especiales que afectan cuando se emplea el archivo como programa ejecutable. Estos modos sólo se aplican a archivos ejecutables y sólo el superusuario puede fijarlos, son los siguientes:



<b>set-uid</b>	Permite fijar el identificador de usuario (valor octal 4000), e indica que cuando el programa se ejecuta, el identificador de usuario pasa a ser el del propietario del archivo original.
<b>set-gid</b>	Permite fijar el identificador de grupo (valor octal 2000), e indica que cuando el programa se ejecuta, el identificador de grupo pasa a ser el del grupo propietario del archivo.
<b>sticky bit</b>	Se denomina "bit de adherencia" (valor octal 1000) y se aplica a programas que son compartidos por muchos usuarios. Con el bit de adherencia se consigue que el programa no abandone el espacio de intercambio (swap) aunque nadie lo esté utilizando. Se suele aplicar a programas de uso intensivo para mejorar sus tiempos de respuesta.

Los dos primeros permisos son de utilidad para programas, como mail, que deben crear archivos en directorios no necesariamente poseídos por la persona que ejecuta el programa. Mas adelante hablaremos en forma mas detallada acerca de estos permisos especiales y que son de gran importancia para la seguridad del sistema de archivos en Unix.

### *Shell*

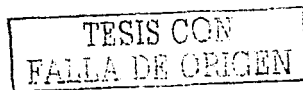
El Shell lee las órdenes y las interpreta como peticiones de ejecución de un programa o programas, lo que realiza posteriormente. Debido a este papel, el shell se denomina interprete de ordenes. Además de ser un interprete de ordenes, el shell también es un lenguaje de programación. Como lenguaje de programación, permite controlar cómo y cuándo se llevan a cabo las ordenes.

### *Kernel*

El Kernel ó núcleo es la parte del sistema operativo que interactúa directamente con el Hardware de una computadora, a través de los controladores de dispositivos que están incorporados en el núcleo. Proporciona conjuntos de servicios que pueden ser utilizados por programas, aislando estos programas del hardware subyacente. Las funciones del núcleo es la gestión de memoria, el control de acceso a la computadora, el mantenimiento del sistema de archivos, el manejo de las interrupciones (señales que finalizan la ejecución), el manejo de errores, la realización de los servicios de entrada y salida (que permiten a las computadoras interactuar con terminales, dispositivos de almacenamiento e impresoras) y la asignación de recursos de la computadora.

Un administrador del sistema puede ajustar las características operativas del sistema mediante la configuración del kernel. Por ejemplo para tener un mejor performance en el manejo de una base de datos, en ocasiones es necesario hacer ciertas modificaciones al kernel.

La configuración del kernel es la acción de indicar, de que tamaño deben ser las tablas que utiliza UNIX para el manejo de sus recursos; dicho proceso se hace mediante la modificación de un archivo de parámetros. Después se toma el kernel y se enlaza con ciertas subrutinas y se genera un nuevo kernel con las nuevas entradas. Este proceso no queda operativo hasta que no se inicie de nuevo el equipo. Esta configuración solamente la puede realizar el administrador del sistema.



Por ejemplo, cuando se configura el kernel es el momento donde se determina el número de archivos que se pueden encontrar abiertos, el número de procesos que se pueden estar ejecutando al mismo tiempo, y el tamaño de los buffers, todos estos recursos los maneja UNIX por medio de tablas y listas. El tamaño de estas tablas es definido por el administrador del sistema, por consiguiente este tamaño no se puede modificar en medio de la ejecución.

Cuando no se configura bien el kernel con una debida estructuración y se deja exceso de espacio para tablas se está desperdiciando la memoria, quitándole espacio a los procesos, debido a que tiene un kernel muy grande.

### **Filosofía de UNIX**

La Filosofía de UNIX se basa en la idea de que un sistema informático potente y complejo, debe ser *simple, general y extensible* y que esto proporciona importantes beneficios tanto para los usuarios como para los que desarrollan programas.

El sistema operativo UNIX contempla los archivos de manera extremadamente simple y general dentro de un modelo único. Ve de la misma manera los directorios, los archivos ordinarios, los dispositivos, tales como las impresoras, discos, teclados y terminales de pantalla. El sistema de archivos oculta al usuario detalles del hardware subyacente.

Una característica única del sistema UNIX es la gran colección de órdenes o herramientas software que proporciona. Esta es otra gran expresión de la filosofía básica. Estas herramientas son pequeños programas, cada uno diseñado para realizar una función específica y todos diseñados para operar juntos. En lugar de pocos programas grandes, cada uno tratando de hacer muchas cosas, el sistema UNIX proporciona muchas herramientas simples que deben combinarse para realizar un amplio rango de cosas. Algunas herramientas llevan a cabo una tarea básica y tienen nombres especiales, otras son lenguajes de programación y por tanto con sintaxis complicadas.

### *Capacidades multiusuario y multitarea*

El sistema operativo UNIX puede ser utilizado por computadoras con muchos usuarios o con un único usuario, ya que es un sistema multiusuario. También es un sistema operativo multitarea ya que un único usuario puede llevar a cabo más de una tarea al mismo tiempo.

### *Entorno excelente para redes*

El sistema UNIX proporciona un entorno excelente para redes. Ofrece programas y facilidades que proporcionan los servicios necesarios para construir aplicaciones basadas en red, base de la Compilación Distribuida. En las computadoras en red, la información y su procesamiento es compartida por diferentes computadoras de red. El sistema UNIX ha demostrado ser útil en la arquitectura cliente-servidor donde máquinas de una red pueden ser al mismo tiempo clientes y servidores.



## Aplicaciones sobre el Sistema Operativo UNIX

En el pasado, el crecimiento del sistema operativo UNIX estaba limitado por la falta de software para aplicaciones y otros tipos de software. No obstante, en los últimos años ha habido una explosión en el desarrollo del Software UNIX. Como consecuencia, ahora podemos obtener software para cualquier aplicación o utilidad del sistema operativo UNIX, solo que ahora nos enfocaremos en los Sistemas referentes al manejo de datos y su administración ya que es parte medular en el desarrollo de este trabajo.

### Software de Gestión de Bases de Datos

Un sistema de gestión de bases de datos suministra un sistema computarizado de gestión de registros que satisface sus necesidades. El software para gestión de bases de datos es a menudo el software más comúnmente utilizado en computadoras personales. Las aplicaciones de negocio construidas sobre sistemas de gestión de bases de datos se utilizan de manera extensiva en minicomputadoras y sobre mainframes. Los sistemas de gestión de bases de datos suministran un lenguaje de consulta, utilizado para borrar, recuperar, modificar y añadir datos. (Muchos productos de bases de datos utilizan el lenguaje de consulta SQL que es un estándar ANSI) La mayoría de los sistemas de gestión de bases de datos utilizan un modelo relacional, que almacena los registros en forma de tablas y permite operaciones que enlazan bases de datos, seleccionan registros de las bases de datos y crean proyectos utilizando campos específicos de los registros en la base de datos.

Estos sistemas de gestión de bases de datos, a menudo suministran herramientas que pueden ser utilizadas para crear aplicaciones personalizadas, incluyendo *lenguajes de 4ª generación (4GL)*

Los desarrolladores de aplicaciones podrán utilizar 4GL para crear de manera rápida aplicaciones de bases de datos, debido a que las declaraciones en 4GL corresponden a funciones convencionales llevadas a cabo en las bases de datos. Cada declaración en un lenguaje de 4ª Generación corresponde a múltiples declaraciones en un lenguaje de 3ª Generación como C, COBOL o Fortran.

Generalmente la elección de un Sistema Operativo para la implementación de Sistemas de Gestión de Bases de Datos sea de tipo Relacional o no, infiere mucho en las capacidades y seguridad que este proporciona, y UNIX es un sistema Operativo que reúne muchas características que se requieren.

En el caso de Informix, suministra un amplio rango de productos de manejo de bases de datos que corren sobre una gran variedad de sistemas UNIX, entre los que se soportan HP-UX, Solaris 2.x, SCO UNIX y UNIX SVR4.

Informix proporciona una diversidad de herramientas que facilitan el desarrollo de aplicaciones que involucran el manejo de datos, así como mecanismos para realizar tareas de consultas paralelas que incrementan el rendimiento hasta treinta veces más que los mecanismos de acceso a bases de datos en serie. Esto es extremadamente útil, ya sea en bases de datos muy grandes como la que se hace referencia en este trabajo de investigación, o en aplicaciones con requisitos de tiempo de recuperación de datos.

TESIS CON  
FALLA DE ORIGEN

### *Administración de Bases de Datos*

**Para la gestión, el almacenamiento y la recuperación de la información que necesitan las aplicaciones que corren en el ordenador central y dan servicio a las diferentes unidades que lo solicitan.**

**Entre las tareas que tiene encomendadas esta área podemos destacar:**

- **Administrar y mantener los sistemas de gestión de bases de datos, con el fin de optimizar el rendimiento de las aplicaciones puestas en servicio.**
- **Estudiar, diseñar y mantener la estructura de la información de las diferentes bases de datos que necesitan las aplicaciones.**
- **Establecer los procedimientos necesarios para la seguridad del sistema:**
  - Autorización de accesos**
  - Copia y recuperación en caso de desastre.**
  - Operaciones de carga y descarga masiva de información de/para usos externos.**
- **Desarrollar programas para verificar e incorporar información externa a las bases de datos.**

**En el siguiente subtema se describe más a fondo los conceptos enfocados a bases de datos, así como el objetivo del uso de un manejador de bases de datos y sus características.**



**1.2 BASES DE DATOS**

Una base de datos es, esencialmente, un conjunto de datos ordenados en filas y columnas que se cargan y se ven en un programa o plantilla de cálculos. Esta compuesta por campos que varían de acuerdo al rubro y el universo seleccionados.

Así mismo "base de datos es la representación de la realidad (entiéndase como organización) en forma de datos; que están entrelazados de la manera más coherente posible, almacenados con una redundancia calculada y estructurados de tal manera que facilite su explotación, y que se puedan satisfacer las necesidades de información de los diferentes usuarios".

En la figura 1.1 se esquematiza de manera simple los componentes principales de un Sistema de base de datos:

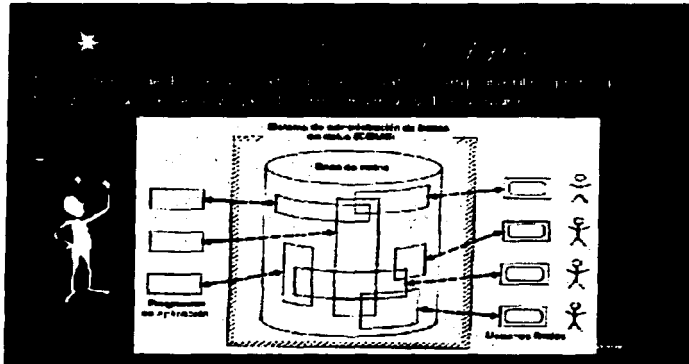


Figura 1.1 Componentes principales de un Sistema de Base de Datos

TESIS CON  
FALLA DE ORIGEN

Un Sistema de base de datos ofrece las siguientes características:

- Ofrece a la organización un control centralizado de su información.
- Esto contrasta con un enfoque de archivos donde cada aplicación tiene sus propios datos (archivos), de modo que los datos están dispersos y son difíciles de controlar.
- Ofrece un acceso dinámico a la información mediante el uso de índices.
- Se mantiene un alto nivel de integridad de los datos, así como la consistencia de los mismos.
- Se tiene un acceso directo a la información gracias al manejo de índices.

*Ventajas de las bases de datos*

- Disminución de redundancias de datos.
- Evitar inconsistencias de datos.
- Compartir datos.
- Mantener la integridad.
- Es posible definir y aplicar restricciones de seguridad.
- Independencia de datos.

*Conceptos Asociados a bases de datos*

Cuando hablamos de bases de datos, estamos haciendo referencia a un conjunto de conceptos que manejamos cotidianamente y que están fuertemente ligados al concepto mismo de base de datos. Así tenemos que:

- Base de datos involucra un conocimiento profundo y detallado de la realidad referida a la organización.
- La base de datos debe ser la representación más fidedigna de la organización.
- Los datos que representan la organización deben ser utilizados de manera coordinada e integral que facilite la "difusión" de los mismos.
- El proceso de difusión de los datos obliga a definir y establecer objetivos que protejan la organización.



En consecuencia, existen cuatro conceptos involucrados:

- 1) Coherencia, asociado a la validez de los datos.
- 2) Integridad, sobre el total de la información a representar.
- 3) Seguridad, como garantía de los datos en su representación.
- 4) Confidencialidad, otorgando acceso de acuerdo a los niveles de la organización.

Las personas encuentran una serie de razones para decidir sobre la necesidad de transformar (convertir) un sistema orientado a archivos en un sistema apoyado en una base de datos por las siguientes cuestiones:

1. Relaciones complejas en el modelo de información.
2. Alta volatilidad en los requerimientos de las aplicaciones y, como consecuencia, una excesiva carga de re programación.
3. Necesidad de integración de nuevas aplicaciones junto con las ya existentes.
4. Exceso de redundancias no controladas con resultados incoherentes.
5. Requerimientos de diferentes tipos de registros en los archivos.
6. Conveniencia de modos múltiples de acceso (secuencial, directo, indexado, etc).
7. Exigencia de modos múltiples de tratamiento (batch u on-line) para los mismos conjuntos de datos.
8. Requerimientos en cuanto a Seguridad, Confidencialidad y Rendimiento.

Pero el manejo de base de datos involucra aspectos tecnológicos, organizacionales y económicos.

#### - Motivaciones de orden Tecnológico y Técnico

La informática provee en la actualidad un sin número de herramientas, entre las que podemos incluir a los Sistemas Manejadores de bases de datos ó Motor de base de datos. Estas herramientas vienen desarrollándose en una diversidad de formas (técnicas) que están asociadas directamente al aumento de capacidad del hardware (velocidad de procesamiento y almacenamiento), la baja de precio, que tienen como objetivo la mejora de la productividad y calidad en el desarrollo y uso del software de aplicación.

#### - Motivaciones de orden Organizacional

El estudio de la organización de la empresa es de vital importancia en BD, en donde se debe determinar el funcionamiento de la misma y clasificar los diversos procesos en los tres niveles de la empresa (estratégico, técnico y operacional). Es decir, que la ejecución de un Planeamiento Estratégico de Información pasa necesariamente por el uso de técnicas y herramientas asociadas a BD.



### - Motivaciones de orden económico

Si bien es cierto, el esfuerzo inicial en BD puede ser costoso, este es, con el tiempo, de gran rentabilidad por las ventajas intrínsecas que redundarán en beneficio de la empresa; claro está que una mala aplicación de la misma, puede traer consecuencias imprevisibles y altos costos.

### Concepto de un Sistema Manejador de base de datos (DBMS)

Conjunto de programas, procedimientos, lenguajes, etc. que suministra a los usuarios los medios necesarios para describir, recuperar y manipular los datos integrados en la base, asegurando su confidencialidad y seguridad. Un manejador de bases de datos es capaz de realizar lo siguiente:

- definición de datos
- privacidad y seguridad
- modificación y/o recuperación de datos
- backup (respaldo) y/o restore (recuperación)
- mantenimiento
- consulta
- reportes

### Objetivos en el uso de un manejador de base de datos

- 1) Oportunidad, asociado a la eficiencia y eficacia.
- 2) Disponibilidad, permitiendo la accesibilidad de datos.
- 3) Consistencia (Oportunidad + Disponibilidad), como calidad de datos.
- 4) Evolución, para adaptarse al entorno.
- 5) Integridad, en el nivel de los datos así como del sistema.

### Objetivos de los DBMS

Los objetivos del DBMS que podemos identificar son:

- Independencia de datos
- Accesibilidad limitada
- Datos al día y sin redundancia
- Consistencia
- Interfaz técnica
- Entrada directa a los datos
- Recuperación por diferentes accesos
- Función completa de interrogación
- Estandarización
- Seguridad



Existen cuatro productos básicos en los DBMS para sistemas host (propietarios). Estos son:

1. Repositorio / Diccionario / Directorio / Enciclopedia
2. Desarrollador de Aplicaciones (Prototipo, Lenguaje(s) de Programación, basado en intérpretes).
3. Lenguaje de Consulta
4. Lenguaje Reporteador

#### 1. Repositorio, Diccionario o Enciclopedia

El concepto ha ido evolucionando y han surgido una serie de términos que están referidos a lo mismo, esto es a un "almacén" donde se guardan o almacenan las características (entiéndase atributos) de los datos. Históricamente los términos son:

- Directorio, asociado al "Directorio del Sistema" que identifica los archivos del sistema.
- Diccionario, referido al concepto de Diccionario de Datos de los grandes DBMS.
- Enciclopedia, utilizado para el desarrollo de herramientas para generar reportes.
- Repositorio, el más reciente, referido a herramientas basadas en el Modelado.

#### 2. Desarrollador de Aplicaciones (Prototipo, Lenguaje(s) de Programación Basado en intérpretes)

Son herramientas que nos van a permitir aplicar los criterios y reglas del sistema, por ejemplo el lenguaje de programación "C".

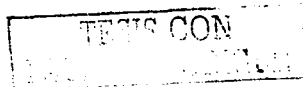
#### 3. Lenguaje de Consulta

Es el lenguaje que nos permite hacer las manipulaciones de borrado, actualización, selección o inserción de nuevos datos a la base de datos así como modificaciones del esquema de la misma.

#### 4. Lenguaje Reporteador

Este lenguaje es mediante el cual se generan los productos finales al usuario, en esta parte el Sistema SIAPSEP utiliza el lenguaje denominado SQR (Struct Query Report), y el mismo lenguaje "C", que por ejemplo van a generar nominas, resúmenes, pólizas contables, etc.

Un manejador dentro de sus características en el manejo de datos, es permitir la integridad de los datos, seguridad de los datos y que no exista la redundancia de los mismos como se explica a continuación:



## **Integridad**

La integridad de la información es un punto básico en los sistemas de bases de datos, que nos garantiza que la información que acumulamos es confiable. Existen reglas de integridad que son específicas a la información de la base de datos. Estas implican que lo que trata de hacer el usuario es correcto, es decir que los datos introducidos tengan una relación con el mundo real (que sean válidos). Para esto se especifican rangos de valores que puedan aceptar los datos o si estos se obtendrán de una lista de valores y cual debe ser esa lista. Esta integridad la marca la naturaleza de la información que se maneja en el sistema. Por otro lado el modelo relacional incluye dos reglas de integridad generales en el sentido en que se aplican a todas las bases de datos. Estas dos reglas se refieren generalmente a las claves primarias y a las claves ajenas.

La regla de integridad de las entidades hace referencia especialmente a las claves primarias y dice así:

\* Ningún componente de la clave primaria de una relación base puede aceptar nulos.

Entiéndase por nulos como información faltante por alguna razón, de esta forma en la práctica no se puede registrar información si se carece de una clave primaria, esto se justifica con la simple razón de que en una base de datos no registraremos información de algo que no podamos identificar.

La regla de integridad referencial está orientada a las claves ajenas y dice así:

\* La base de datos no debe contener valores de clave ajena sin concordancia.

De este modo si los valores de las claves primarias representan identificadores a entidades, así los valores de clave ajena representan referencias a entidades, una expresión lógica sería de la siguiente manera: Si B hace referencia a A, entonces A debe existir. Con esto advertimos que la integridad referencial exige concordancia de las claves ajenas, muy especialmente con las claves primarias.

## **Redundancia**

Entiéndase por redundancia a la información que está repetida dentro de la base de datos. Ciertamente en una base de datos existe información que se encuentra más de una vez, pero esto debe ser en una expresión mínima, el motivo fundamental de la relación entre entidades dentro del modelo relacional es el evitar la redundancia de la información, pues con tan solo conocer la clave de una tupla se puede conocer el resto de la información correspondiente. El que en una base de datos exista redundancia de información implica el riesgo de tener datos inconsistentes si no se toman las medidas pertinentes. Sin embargo en modelos distribuidos la redundancia es una técnica de mejora al sistema, como lo veremos en capítulos posteriores.

## Seguridad

Cabe recalcar que otro punto fundamental dentro de los sistemas es la seguridad. Entendiendo por seguridad de la base de datos a la protección de la información contra una revelación, alteración o destrucción no autorizada. De esta forma el sistema debe estar al tanto de ciertas restricciones que no deben ser violadas por los usuarios. Para esto debe existir un catálogo de usuarios y su correspondiente alcance. El administrador de la base de datos debe asignar los permisos correspondientes a cada grupo de usuarios, con esto se restringen derechos por cada grupo dependiendo del nivel y responsabilidad de cada uno de ellos dentro de la empresa, existen restricciones de aspecto legal, social o ético que deben ser aplicados en las políticas de seguridad del sistema e implantados en el alcance a los usuarios, de esta forma cada grupo ve el sistema desde un punto diferente, así se protege la información dentro del mismo.

Pero también se debe contemplar la seguridad de la información fuera del sistema de base de datos por eso esta administración va de la mano con la administración de los recursos de la red. Se puede tener un sistema de base de datos en un ambiente muy seguro, pero si fuera del sistema se puede copiar y editar la información, de nada serviría, para evitar un incidente se deben planear los directorios en donde se alojará la información y asignar de esta forma derechos sobre ellos a cada usuario en coordinación con el administrador de la red, para evitar un dolor de cabeza se deben plantear políticas a niveles empresa y de sistema ya que la seguridad debe ser global

## Base de Datos del Modelo Relacional

La mayoría de los productos comerciales de bases de datos están enfocados al modelo relacional, de hecho este modelo representa la tendencia dominante en el mercado actual de estos sistemas y se dice que constituye el avance más importante en el campo de los estudios sobre las mismas. Esta es una de las razones por las cuales gran número de sistemas están basados en él, además de que la tecnología relacional cuenta con bases sólidas en ciertos aspectos de las matemáticas, esto se refiere a que la funcionalidad del modelo relacional está sustentado en el álgebra relacional. Esta ofrece un conjunto de operaciones explícitas como unión, reunión, restricción, intersección, proyección, producto, división y diferencia, que sirven para indicar al sistema como constituir alguna relación deseada partiendo de las relaciones dadas en la base de datos. De hecho, el álgebra relacional en su descripción se acerca a un lenguaje de programación, la cual se sustenta matemáticamente en la lógica de conjuntos. Existe otra relación similar entre el álgebra relacional y el SQL, ciertamente dentro de la historia del modelo relacional el SQL ha tenido un papel fundamental en la evolución de éste. En 1985 aparece el Manejo de bases de datos por Ejemplos (QBE) y en 1986 el Instituto Nacional Americano de Normas (ANSI) y la Organización Internacional para la Estandarización (ISO) coinciden en adoptar el dialecto SQL como interfaz "oficial" para sistemas relacionales. A continuación describiremos los principios básicos del modelo relacional.

TESIS CON  
FALLA DE ORIGEN



**Independencia de Datos**

- Se define como "la inmunidad de las aplicaciones ante cambios en la estructura de almacenamiento y en la técnica de acceso a los datos"(C.J.Date).
- Las aplicaciones no dependen de la estructura o técnica de acceso de los datos.
- Para lograr este objetivo se concibe un sistema de base de datos como una arquitectura de tres niveles.

**Arquitectura de tres niveles**

- Nivel Interno: Más cercano al almacenamiento físico, es decir se ocupa de cómo se almacenan físicamente los datos.
- Nivel Externo: Más cercano a los usuarios, se ocupa de como los usuarios perciben los datos (vistas individuales).
- Nivel Conceptual : Es el nivel intermedio entre el interno y el externo, además de que presenta una vista global o comunitaria de los datos.

El esquema interno describe la estructura física de almacenamiento de la base de datos.

El esquema conceptual describe la estructura de toda la base de datos para una comunidad de usuarios. Describe entidades, atributos, vínculos, restricciones, operaciones , etc.

Esquema externo o vista de usuario describe la parte de la base de datos que interesa a un grupo de usuarios y oculta a ese grupo el resto.

En la actualidad la mayoría de las empresas trabajan bajo el concepto de bases de datos para tener un buen manejo de la información, como es el pago de la nómina en el sistema SIAPSEP que lleva el control del pago de los maestros a nivel central de 250,000 maestros y de manera descentralizada a nivel estatal, utilizando un manejador de bases de datos Informix y del que se hace referencia en el siguiente tema.

### **I.3 INFORMIX DINAMIC SERVER**

#### **Historia de Informix**

Software Informix Inc. se creó en 1980 por Roger Sippl y Laura King que tuvo como nombre inicial "Sistema de base de datos Relacionales Inc.". Informix es un descendiente de Postgres.

Informix es una de las primeras compañías comercializadoras de bases de datos relacionales, con protocolos para el procesamiento de transacciones en línea (OLTP), y de lenguajes de cuarta generación (4GL) para plataformas UNIX.

En 1981, esta compañía saca al mercado C-ISAM, que son rutinas del lenguaje C para crear y usar índices de archivos secuenciales.

En el año de 1982 la compañía se dedicó a investigar y realizar pruebas lanzando posteriormente en el año de 1984 una segunda generación de lenguajes SQL (conteniendo herramientas basadas en SQL y C).

En 1986, Informix sale al público. Además ese mismo año sacan al mercado Informix-4GL un lenguaje de cuarta generación.

En 1987, crean Informix SE. (Informix Standar Engine) que es un sistema pequeño sin necesidades grandes, requerimientos de administración mínimos sobre lo que pueden ser pequeñas bases de datos.

En 1988, Informix toma el nombre (oficialmente) de Informix Software Inc.

En 1990, sacan al mercado Informix-OnLine (llamado anteriormente Informix Turbo). El cual es un sistema para base de datos grandes y con una mayor necesidad de administración.

En 1994, Informix presenta Informix Dinamic Server: que maneja bases de datos en "multihilos" (multithread).

En 1996, las compras de Informix resaltaron (base de datos Relacionadas a objetos). Notemos eso en 1996, el peso de Informix en el mercado se valora, tiene \$558 Millones, o 15.6% del mercado de las bases de datos. Informix es considerado en la tecnología de la computadora y de la innovación arquitectónica en bases de datos.

En octubre de 1998, Informix compra el almacenamiento de la parte de datos de Red Brick junto con su tecnología, esta herramienta dedicada exclusivamente a Dataware House, proporcionando carga (Alto desempeño en un solo paso), consulta (El desempeño lo es todo), escalabilidad (De Gigas a Terabytes), disponibilidad (Datos disponibles en todo momento), administración (Fácil y Flexible) y acceso (Acceso desde cualquier plataforma/herramienta).

En el siguiente esquema (Figura 1.2) se muestra como ha ido evolucionando la tecnología de Informix Dynamic Server, en sus diferentes versiones, identificando las características particulares de cada una de ellas.

Data Management Solutions

## La Evolución de IDS

**Mejor base de datos para propósitos generales, OLTP, nivel empresarial y de grupos de trabajo**

9.30

Con características de extensibilidad que son la llave para descubrir los valores escondidos de la base de datos

7X

Sistemas con bases de datos de usuario final...

- Particionamiento, Parallel Data Query, HDR y ER, soporte para gran cantidad de usuarios, etc.

6X

Incremento en el desempeño y la funcionalidad con...

- Arquitectura DSA Multi-threaded, I/O asíncrono, respaldos y administración de dispositivos, etc.

5X

Bases de datos con...

- Optimizador basado en costos, two-phase commit, integridad referencial, etc




Figura 1.2 Evolución de la Tecnología de Informix Dinamic Server

### Informix Dinamic Server

Es un Servidor de base de datos Multitareas que provee un excelente rendimiento, administración y escalabilidad. Se encuentra en producción en miles de sitios hoy en día, Informix Dinamic Server es tecnología de vanguardia en la industria de las bases de datos.

Disponible para plataformas UNIX, LINUX y Windows NT, Informix Dynamic Server dentro de sus características tiene un procesamiento de transacciones superior y un óptimo soporte a la toma de decisiones a través de tecnología de consultas paralelas de datos (query paralela data), alta disponibilidad, integridad de datos, administración de sistema, facilidades de replicación, y herramientas de monitoreo gráfico.

Con su tecnología y las flexibles opciones de configuración, Informix Dynamic Server provee una plataforma de base de datos para toda empresa, soportando aplicaciones de grupo de trabajo descentralizadas y departamentales, así como aplicaciones OLTP, Web y Dataware house ejecutándose en ambientes paralelos. Informix Dynamic Server incluye

TESIS CON  
 FALLA DE ORIGEN



una colección de productos de conectividad de alto rendimiento que aseguran acceso seguro y transparente a diferentes sistemas.


La tecnología de Informix Dynamic Server es compatible con un número considerable de proveedores de hardware y software en la industria, en la figura 1.3 se muestran algunos de ellos.

**Data Management Solutions**


## Neutral en las plataformas

**Las plataformas OEM incluyen...**

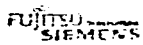
- Sun Microsystems
- HP
- Compaq
- Fujitsu-Siemens
- EMC
- Network Appliance
- StorageTek
- Caldera/SCO
- .....




take it to the n



invent





**COMPAQ**




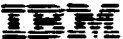





Figura 1.3 Software y Hardware compatible con Informix

El rendimiento de este servidor permite a los clientes procesar más datos en menos tiempo para obtener rápidamente respuestas a las preguntas de negocios y críticas. La escalabilidad del servidor permite a los clientes crecer con facilidad y personalizar su base de datos mientras cumple con las exigencias de rendimiento y preserva sus inversiones iniciales de tecnología de información.

Las capacidades de Informix Dynamic Server pueden ser extendidas añadiendo una o más opciones de configuración. Estas opciones añaden una funcionalidad clave para permitirle construir aplicaciones que resuelven las complejas necesidades del mercado.

En la próxima generación de informix para servidores de bases de datos ofrece una herramienta de transacciones probado por la industria para aplicaciones de misión crítica, así como la versión para internet, capaz de soportar miles de usuarios coexistentes.

TESIS CON  
 FALLA DE ORIGEN

Informix Dinamic Server entrega máxima fiabilidad, disponibilidad y es escalable para impulsar sistemas que procesan transacciones grandes. En la siguiente figura se esquematizan estas características.

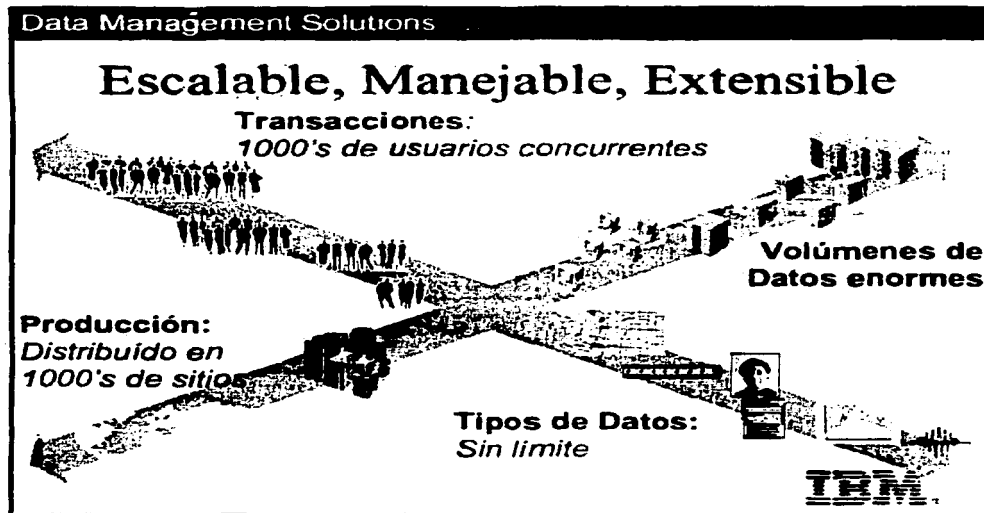


Figura. 1.4 Características de Informix

Informix Dinamic Server ofrece la siguiente característica y perfeccionamiento:

Incremento de procesamiento beneficiando de un 15 a un 30%

**Comparación con otros Manejadores de bases de datos**

*Comparación con Oracle*

La guerra entre Oracle e Informix es intensa debido a que sus opciones de aplicación son muy similares y que apuntan hacia los mismos mercados.

Los dos servidores de bases de datos salieron entre el año de 1996 del mes de marzo (Oracle) y el año de 1996 del mes de diciembre (Informix). Sistemas que integran la orientación a objetos.

La tecnología usada son los DataBlades. La diferencia principal entre Oracle (Servidor 7.3.2 Universal) e Informix (Informix Dinamic Server) es la arquitectura de aplicación del DataBlades.

Nota: Un datablade está compuesto por una colección de objetos de Base de Datos y código que extienden la funcionalidad del manejador de base de datos. En cierta forma puede considerarse como una librería de clases C++ o Java que encapsula tipos de datos especializados

Informix da el acceso directo al centro, que es un valor por defecto serio que tiene la seguridad: los DataBlades no son ¡necesariamente implementadas por Informix!.

Oracle, no da el acceso al centro, por lo que es más seguro. El inconveniente es que es menos rápido. (La seguridad siempre tiene un precio por lo que se refiere a velocidad).

Otra diferencia es el proceso que tiene Oracle e Informix, tienen objetos relacionales que en la actualidad la versión de IDS2000 ya es orientada a objetos.

La estrategia de Oracle es más bien de tener una "imagen instantánea" de la base de datos que tiene en determinado momento, para encontrar información necesaria para contestar a las demandas.

Un último punto de comparación interesante es el mercado apuntado por Oracle: La aplicación cliente - servidor, y las redes LAN, más las aplicaciones UNIX. Informix, apunta las mismas marchas, excepto los microordenadores.

#### *Comparación con PostgreSQL*

La comparación entre PostgreSQL e Informix empieza con un punto esencial: PostgreSQL es gratuito e Informix no.

Otra diferencia es la dirección del tiempo. Informix tiene una herramienta poderosa y completa (FECHA & el INTERVALO), mientras PostgreSQL tiene un tipo de tiempo de los datos (VIAJE de TIEMPO) eso es fácil para algunas aplicaciones.

El tiempo en Informix es simplemente por un módulo de DataBlade cuyo principio es poner un TIME\_STAMP (marcador temporal) para cada objeto en los momentos necesarios.

Otra diferencia es que Informix es monolítico, mientras la arquitectura de PostgreSQL no, parece finalmente que la comparación es inútil, marchas apuntadas por Informix y PostgreSQL son muy diferentes a pesar de que en sus inicios informix descendió de Postgres.

#### *Comparación con Sybase*

Entre las comparaciones se puede hablar del manejo de procesadores virtuales en Informix que Sybase no maneja, lo que permite que el tiempo de procesamiento en informix sea mucho más rápido, el esquema cliente/servidor es el mismo, aunque el hablar de la seguridad Sybase profundiza más en este aspecto manejando distintos los

TESIS CON  
FALLA DE ORIGEN

usuarios de Unix y de base de datos, pero de manera general comparativamente su esquema en cuanto a sistemas de administración de BD es diferente, observando que el esquema de Sybase apunta al mismo esquema que Oracle.

### *Comparación con DB2*

Uno puede comparar Informix brevemente con DB2: En primer lugar, uno de los rasgos principales es que DB2 gira en los sistemas Mainframes IBM (normal para un producto de IBM), en LAN's, microordenadores o las estaciones UNIX.

DB2 es extensible. Esta característica también encuentra de nuevo a Informix como servidor dinámico.

Informix fue el primero en lanzar la idea de objeto - relacional en el año de 1996 de enero. Actualmente Informix pertenece a IBM por lo que a un futuro se puede entender que se identificarán las características de DB2 e Informix para fusionarlo en un solo servidor con las bondades de cada uno.

Informix Software figura dentro de los sistemas de bases de datos escalable y confiable.

Las opciones de configuración proporcionan funcionalidad de ambientes donde la base de datos se usa. Herramientas que hacen accesibles las aplicaciones de desarrollo. El software puede comprarse para ejecutar multiprocesadores simétricos (SMP) y el procesamiento en paralelo masivo (MPP). El uso de extensiones lo que permite apoyo de programación adicional para ejecución de procesos alternos, como el desarrollo de tipos de fechas especiales como son los DataBlades.

Informix Dynamic Server puede configurarse para apoyar aplicaciones de Web/Intranet, los multi-medios de comunicación, alto-volumen, las transacciones de OLTP, el almacenamiento de dicha información.

Informix Dynamic Server; esta disponible en Windows NT y la mayoría de las plataformas de UNIX, incluyendo, LINUX, apoyando necesidades de dirección de datos de las aplicaciones departamentales.

Todo esto nos lleva al buen manejo, funcionamiento y a su vez conocer todo lo que un manejador de base de datos ofrece y si cumple nuestros requerimientos, así como la flexibilidad en el manejo de los datos, seguridad, tiempo de respuesta, etc., cabe señalar que esto no culmina en el manejador de base de datos, sino que todo se ve reflejado en el buen funcionamiento de cualquier sistema. En el siguiente punto hablaremos del sistema SIAPSEP mencionando sus características.

TESIS CON  
FALLA DE ORIGEN

#### **1.4 SISTEMA INTEGRAL DE ADMINISTRACIÓN DE LA SECRETARÍA DE EDUCACIÓN PÚBLICA SIAPSEP**

En la actualidad existen sistemas desarrollados en diferentes lenguajes de programación los cuales utilizan para el manejo de datos alguno de los sistemas manejadores de bases de datos más comunes, el sistema SIAPSEP utiliza Informix y es un sistema que nos puede hablar de la potencialidad y rapidez de este manejador.

La Secretaría de Educación Pública ha desarrollado, el Sistema Integral de Administración de Personal y Pagos SIAPSEP el cual tiene como objetivo primordial el pago a maestros de educación docentes y administrativos de todo el Distrito Federal, este sistema cuenta con una plataforma Cliente-Servidor, corriendo en equipos clientes de Microsoft Windows, utilizando como vista (Front-End) un 4GL llamado UNIFACE Ver. 7.1, interactuando con una base de datos tipo relacional INFORMIX DINAMIC SERVER, en procesos pesados usa lenguaje de tercera generación 3GL (Lenguaje "C") y en aspectos de comunicación utiliza el protocolo TCP/IP.

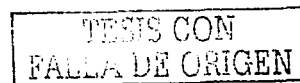
Como aspecto histórico este sistema fue creado en sus inicios en programación COBOL, con un sistema operativo primos y un equipo PRIME manejando archivos planos por lo que no tenía integridad referencial, la redundancia de información predominaba y a su vez permitía la modificación de archivos provocando un descontrol e inseguridad de la información y la incorrección de los datos emitidos para el pago de la nómina.

Posteriormente se decidió migrar de un sistema de archivos planos a un sistema manejador de bases de datos denominado Sybase que en esos momentos presentaba mejores condiciones de procesamiento con gran cantidad de información el lado inverso del sistema que se venía manejando un esquema de seguridad, integridad referencial, evitar redundancia de información así como tiempo de respuesta más rápido, esto de manera comparativa, el proceso del cálculo de la nómina utilizando el sistema anterior era de cuatro días y se redujo a 2 días pero todavía seguía siendo mucho el tiempo de respuesta con el sistema manejador de base de datos Sybase. El sistema se conformaba de la siguiente manera.

Contaba con la siguiente estructura:

- Una plataforma cliente – servidor corriendo sobre Windows 3.11
- Utilizaba una vista (Front-End) desarrollada con una herramienta llamada UNIFACE de origen holandesa que permitía realizar las formas para interactuar con el manejador de BD.
- Lenguaje de programación "C" para la realización de procesos pesados
- Utilización del protocolo TCP/IP.
- Los reportes que se emitían eran con un lenguaje interprete SQR (Reportes de consultas estructuradas).

En la Figura 1.5 se muestra la pantalla inicial en la parte del Front-End o cliente al acceder al sistema SIAPSEP



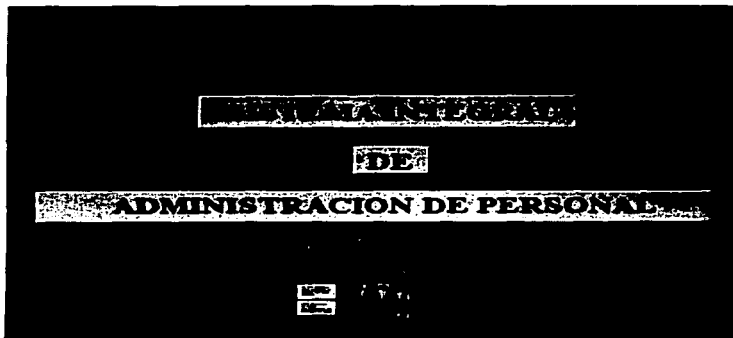


Figura. 1.5 Presentación inicial del Sistema SIAPSEP

Posteriormente dadas las características que presentaba el producto de informix como son los procesadores virtuales y tiempo de respuesta superior al que presentaba sybase, se decidió cambiar de servidor de base de datos iniciando con la versión de informix 7.14, el cual continuaba con una plataforma cliente – servidor, corriendo sobre Microsoft Windows 95, utilizando como lenguaje 4GL UNIFACE versión 6, lenguaje "C" (Compilador Apogee "C"), ESQL/C que es una combinación del lenguaje estructurado de consultas y el lenguaje C, como protocolo de comunicación TCP/IP y un lenguaje reporteador SQR.

En el siguiente párrafo se describen las partes que conforman la aplicación del SIAPSEP.

### **Plataforma de Desarrollo**

*Interfaz de usuario final.*- Es la interfaz gráfica que le va a permitir al usuario, consultar y actualizar la base de datos, esta interfaz está basada en una herramienta denominada UNIFACE.

*Manejador de base de datos relacional.*- El manejador de base de datos es donde se van a encontrar todas las tablas o entidades así como los datos, en este caso la herramienta utilizada es Informix Dinamic Server.

*Sistema operativo estándar.*- El sistema operativo es el que nos permite llevar a cabo toda la administración de los recursos, y de acuerdo al sistema operativo y la versión tendrán que ser los productos correspondientes, para nuestro caso es el sistema operativo Unix

*Programación paramétrica .-* En el sistema SIAPSEP existe una parte en la cual se llevan a cabo las modificaciones así como la adición de conceptos de pago de una nómina, una de las razones del porque tenga que ser parametrizable, es para que el usuario simplemente con definirlo en algunas pantallas, pueda dar de alta o cambiar datos, esto apoyado de programación y catálogos contenidos en la base de datos.

En la figura 1.6 se muestran las capas que intervienen para que la aplicación funcione.

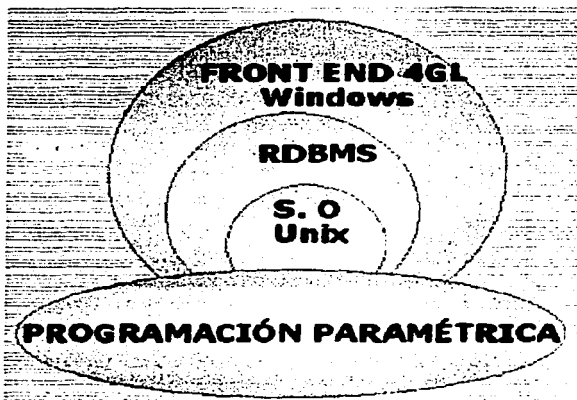


Figura 1.6 Capas que intervienen en el SIAPSEP

Características del equipo y sistema actualmente instalado en oficinas centrales de la Secretaría de Educación Pública

Red	Software	Sistema
Superservidor SUN 10000 1 GB EN RAM HD 48 GB 4 Procesadores 95 Estaciones Pentium II 128 MB en RAM HD 500 MB Cableado estructurado ETHERNET 10BaseT Servidor de Comunicaciones y Archivos (Linux)	S.O. UNIX Manejador de base de datos INFORMIX UNIFACE 7 vista al usuario WINDOWS 95,98,2000 y XP Compilador "C" ANSI SQR TCP/IP OnNet(Comunicador del cliente con la BD)	420 Formas en Uniface 380 Objetos en Lenguaje 750 Reportes SQR 350,000 Plazas

TESIS CON  
FALLA DE ORIGEN

La figura 1.7 muestra como se hace la conexión de un cliente hacia un servidor con los programas mencionados. Cada uno de los productos como son el Sistema Operativo, el manejador de base de datos y la interfaz gráfica, para que el cliente o el usuario las puedan ver requieren de programas de comunicación conocidos como protocolos y también ODBC para la conexión a la base de datos que cada uno de los manejadores contiene.

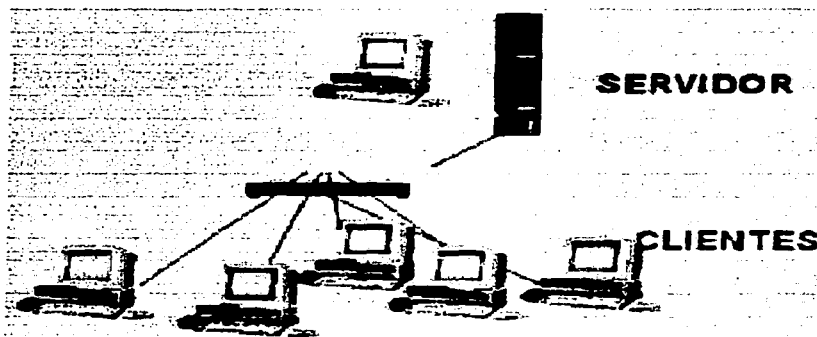


Figura. 1.7 Conexión Cliente/Servidor (Utilizada por el sistema SIAPSEP)

### Características Técnicas del Sistema SIAPSEP

**Integral.-** Relación con diversos subsistemas, por ejemplo: Presupuesto, Contabilidad, Tesorería, etc.

**Paramétrico y flexible a cambios.-** Operación sencilla en las modificaciones; reglas del negocio inmersas en la misma base de datos.

**Eficiente** como producto de misión crítica con alto volumen de información.

**Portable** a diversas plataformas de cómputo, base de datos y sistema operativo.

**Ambiente de Interfaz Gráfica (Windows),** con ayuda en línea.

**Plataforma de Última Tecnología (Cliente/Servidor).**

TESIS CON  
FALLA DE ORIGEN



Es un producto durable y redituable.

Orientado a los Procesos, pre-empleo, empleo y post-empleo.

Interoperable.- La operación se realiza desde cualquier punto de la Red.

Seguridad de acceso.

**Esquema Conceptual**

El esquema conceptual describe los procesos que el sistema realiza para los requerimientos de los usuarios finales como reportes, archivos, etc. así como los cambios que la BD detecta por los movimientos de personal como son: altas, bajas, cambios ó licencias de personal que labora en la Secretaría de Educación Pública, y se muestra en la figura 1.8.

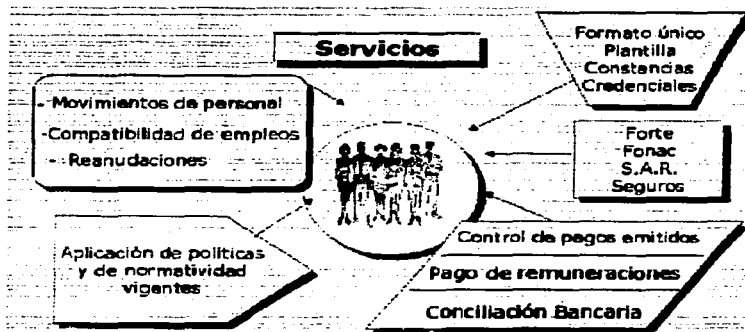


Figura. 1.8 Esquema del funcionamiento del Sistema SIAPSEP

TESIS CON  
FALLA DE ORIGEN

Es la presentación de cómo van afectándose las diferentes entidades o tablas de la BD por los efectos del desarrollo de la nómina de este sistema. Lo siguiente con la finalidad primordial del pago de la nómina, pero para llegar a este paso se ven involucradas varias actividades previas que alimentan a la base de datos en varias de sus tablas e información que servirá para procesos posteriores o generación de procesos previos, la figura 1.9 nos muestra como punto central al empleado, pero que se ven

afectados por cambios de tablas que hacen referencia, en este caso a la tabla del empleado y por los movimientos que anteriormente se mencionaron como son altas bajas o cambios de registros contenidos en la base de datos.

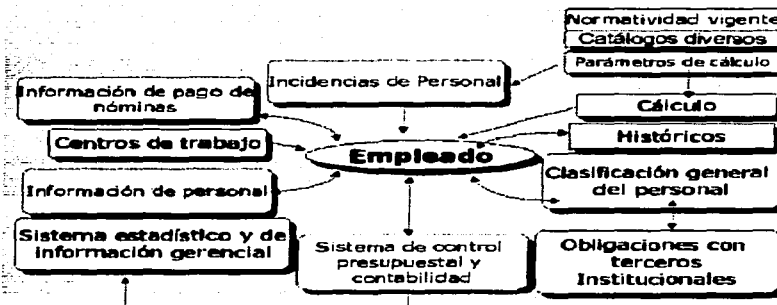


Figura 1.9 Esquema de entidades principales en el SIAPSEP y su relaciones

Los esquemas antes mencionados deben de cumplir con la seguridad, integridad, flexibilidad de los datos, todas estas características conllevan a la entrega puntual de la información.

La figura siguiente (Fig 1.10) muestra que características debe de tener un sistema para la entrega oportuna de la información como finalidad primordial.

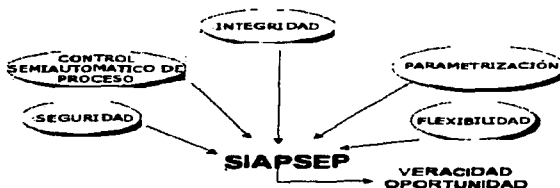


Figura 1.10 Características que proporciona el sistema SIAPSEP

**Proceso de Información**

La concentración de todo el software, programación y utilización del manejador de base de datos va a permitir llevar a cabo todo el proceso, integrando los esquemas antes mencionados como se muestra a continuación en la figura 1.11:

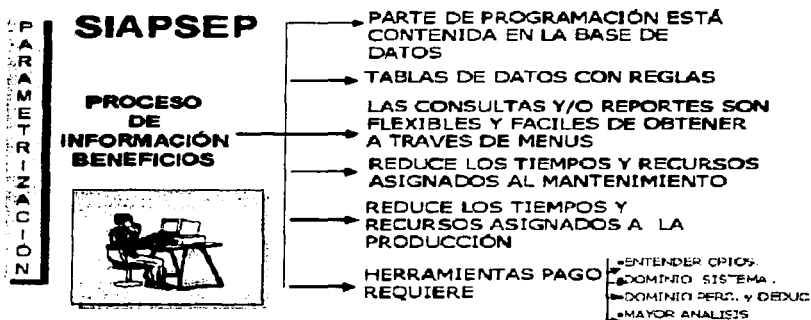


Figura. 1.11 Accesibilidad para el procesamiento de la Información en el sistema SIAPSEP

**Proceso del SIAPSEP**

Este siguiente esquema nos permite observar como se realiza el trabajo por parte de todas las personas involucradas en el sistema, como es el administrador del Sistema Operativo, de la base de datos los cuales van a ser los que estén dando el mantenimiento correcto de cada uno de los programas y en base a ello el operador del sistema y el capturista podrán hacer el trabajo correspondiente, con la finalidad de terminar el ciclo que sería la salida, con la entrega de reportes, pagos y término de la quincena.

Todo lo mencionado nos permite observar que una de las partes más importantes para poder llegar a la salida según la figura 1.12 es el buen control que tengan los administradores sobre el sistema, que es el punto primordial en este trabajo de investigación referente a base de datos.

TESIS CON  
FALLA DE ORIGEN

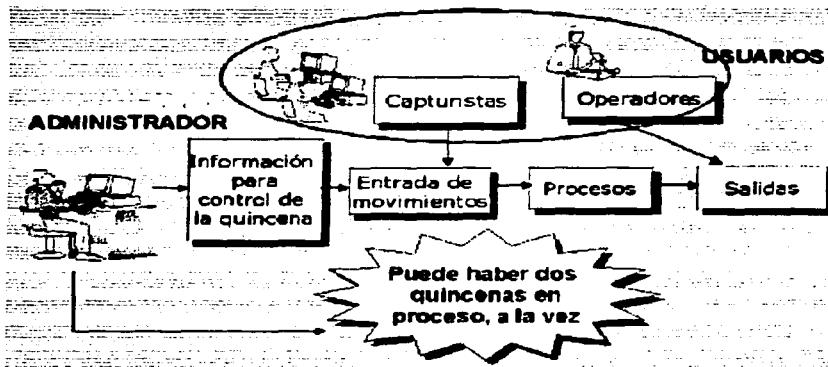


Figura 1.12 Flujo de procesamiento de información en el SIAPSEP

Todo esto corresponde a la parte aplicativa de este trabajo de investigación, a continuación iniciaremos con la parte central que es la administración del servidor de base de datos, encargados de administrar una o más bases de datos que se tengan creadas. Un punto muy importante que hay que recalcar es que nosotros no desarrollamos el sistema de base de datos, por el contrario el objetivo de este trabajo corresponde a la administración enfocada al sistema SIAPSEP para su mejor funcionamiento en la toma de decisiones y tiempos de respuesta oportunos.

Cabe señalar que las bases de datos que utiliza este sistema son dos: la nómina\_ofm que es la que contiene los datos y unifaceseven que es el repositorio de datos.

TESIS CON  
FALLA DE ORIGEN

## TEMA II

## SERVIDORES DE BASES DE DATOS

Dentro de la administración de una base de datos, existen diferentes roles que son desempeñados por diferentes personas en una organización, dependiendo de las necesidades que se tengan, el hablar de administración es hablar de organización ya sea de archivos, monetarios, personales, etc. Y bajo el cual se va a regir una forma de manejo estructural como en nuestro caso especial hablamos del manejo de archivos que a su vez contienen datos, los cuales por sus características están almacenados en catálogos, contenidos en una base de datos.

En este capítulo se describen las distintas funciones que son llevadas a cabo por el administrador de un servidor de base de datos, así como la distinción de los administradores de acuerdo a la función que realizan ya que puede haber un administrador del servidor de base de datos y otro administrador de alguna base de datos en especial, también se hará referencia a las características del servidor de base de datos Informix en su versión 2000.

Se mencionan los tipos de aplicaciones cliente que operan sobre una base de datos de tipo relacional, además de que se mencionan aspectos importantes con relación a la arquitectura que implementa Informix Dynamic Server para el mejor rendimiento y tiempo de respuesta, así como las características que posee un servidor de manera dinámica, como son el manejo de memoria compartida, el manejo de disco, así como la seguridad en el servidor de base de datos.

## II.1 TIPOS DE ADMINISTRADORES

TESIS CON  
FALLA DE ORIGEN

Debido a las diversas funciones que tiene un administrador y para poder llevar a cabo un buen funcionamiento tanto del servidor en general como las bases de datos se tiene que estructurar o designar actividad propias a cada administrador para tener mayor seguridad y disponibilidad de datos de manera específica (Determinada base de datos), como de manera general (Servidor de Base de Datos). A continuación se listan los administradores así como los apertadores que son los que ejecutan las instrucciones encomendadas por un administrador (Figura 2.1)

- *Administradores de bases de datos* El administrador de base de datos (DBA) es el primer responsable del manejo del control de accesos a una base de datos, mediante el uso de sentencias SQL permite y quita privilegios para asegurar el uso correcto de manera individual y realizar acciones necesarias para el buen rendimiento ya que en ocasiones usuarios demandan de manera potencial recursos o realizan de manera inapropiada actividades.
- *Administrador del Servidor de base de datos* Distinto a la función de DBA, un administrador de servidores de base de datos, es el responsable del mantenimiento,



- administración y operación del servidor entero de base de datos, el cual contiene distintas bases de datos de manera individual.
- *Operador del Servidor de base de datos* Es el responsable de las tareas rutinarias asociadas a la administración del servidor de base de datos incluyendo respaldos y restauración de una base de datos. La misma persona puede ocupar las reglas de administrador y operador.

El siguiente esquema muestra los niveles de administración involucrando las actividades del operador como parte importante en la administración.

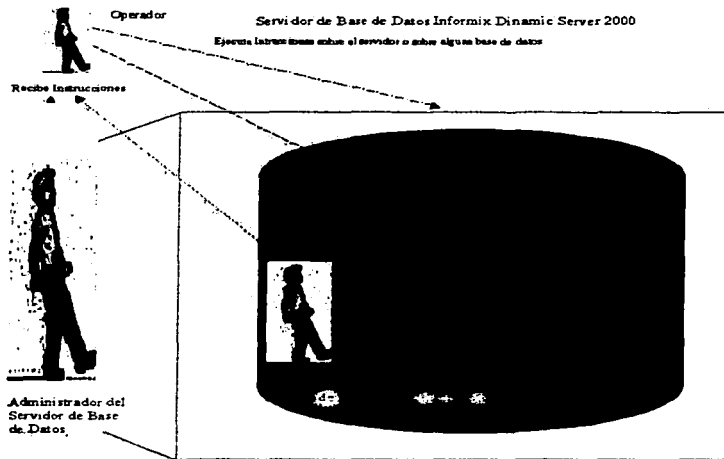


Figura 2.1 Niveles de administración en un Servidor de Base de Datos

## II.2 CARACTERÍSTICAS DEL SERVIDOR DINÁMICO (DINAMIC SERVER)

Es un sistema manejador de bases de datos de objetos relacionales (ORDBMS), combinando orientado - objetos y correlación, la capacidad de representar todos los datos en tablas compuestas de regiones y columnas. Además proporciona ayuda para crear en datos alfanuméricos, tales como cadenas de caracteres, enteros, decimales y datos. El servidor de base de datos proporciona capacidades orientado a objetos, habilitado para añadir nuevos tipos de datos y rutinas de apoyo para extender sus capacidades de manejo de datos.

El servidor de base de datos proporciona las siguientes características:

- Arquitectura cliente/servidor
- Arquitectura Escalable Dinámica
- Alto rendimiento
- Alta disponibilidad
- Administración del sistema dinámico
- Seguridad del servidor de base de datos
- Auditable
- Distribución de consultas en la base de datos
- Extensibilidad

A continuación se explican cada una de las características anteriores.

### II.3 ARQUITECTURA CLIENTE/SERVIDOR

El servidor de base de datos proporciona acceso fiable a la base de datos para la aplicación de programas del cliente. Un cliente es un programa de aplicación que el usuario corre para regresar información de una base de datos que maneja el servidor de base de datos.

Cuando la aplicación de un cliente se conecta a un servidor de base de datos y regresa información, el servidor de BD localiza la respuesta de datos dentro de la BD y regresa el resultado. El servidor de base de datos puede regresar renglones seleccionados de tablas dentro de la BD, añade o borra renglones ó actualiza columnas particulares.

La aplicación del cliente usa Lenguaje Estructurado de consultas (SQL) para mandar las instrucciones al servidor de base de datos y recibir respuesta. Lo programas del cliente incluyen utilerías como el dbaccess y programas fuentes usando Informix API con herramientas tales como Informix-ESQL /C, Informix-CLI, ODBC, C++ ó Java.

#### Tipo de aplicación de clientes

Hay dos clases de aplicaciones de acuerdo a las necesidades del cliente, que operan sobre datos en una base de datos relacional:

- Aplicación de procesos de transacciones en línea (OLTP)
- Aplicación de sistemas soporte – decisión (DDS)

*Aplicaciones OLTP (OnLine Transaction Processing)*

Son usadas frecuentemente para la captura de nuevos datos o actualización de datos ya existentes. Estas operaciones involucran rapidez, acceso indexado, número pequeño de

TESIS CON  
FALLA DE ORIGEN

registros. Las aplicaciones OLPT son frecuentemente aplicaciones multiusuarios con respuesta aceptable con tiempo de medida en fracciones de segundos. Las aplicaciones OLPT tienen las siguientes características:

- Transacciones sencillas que involucran cantidades pequeñas de datos.
- Acceso indexado a datos.
- Muchos usuarios
- Respuestas frecuentes
- Tiempo de respuesta rápida.

### Aplicaciones DSS ó DDS

Son utilizadas para reportes o consolidar datos que fueron capturados mediante operaciones OLPT. Estas aplicaciones proporcionan información usada para planear estrategia de contabilidad y toma de decisiones. Los datos contenidos dentro de la BD son consultas que no se pueden actualizar durante operaciones DDS. Las aplicaciones DDS incluyen nómina, inventarios y reportes financieros.

La figura 2.2 ilustra la forma de conexión de un cliente a través del Informix-CLI el servidor de base de datos Informix.

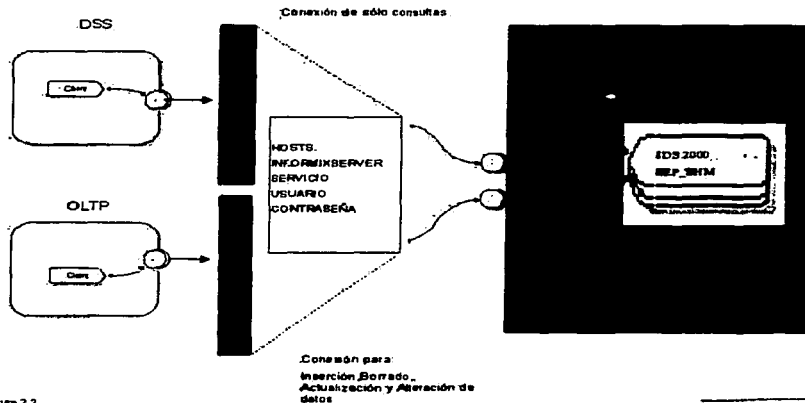


Figura 2.2

Figura 2.2 Esquema de conexión de clientes de aplicaciones DSS y OLTP mediante el cliente de informix ( Informix- CLI )

TESIS CON  
FALLA DE ORIGEN



## II.4 ARQUITECTURA DINÁMICA ESCALABLE

Los servidores de BD de informix implementan una arquitectura avanzada que informix llama Arquitectura Escalable Dinámica (DSA Dinamic Scalable Architecture). DSA proporciona distintas ventajas de rendimiento para iniciar un procesador y una computadora multiprocesador simétrico. Las ventajas son las siguientes:

- Un número pequeño de procesos de servidores de BD, da servicio a una cantidad considerable de procesos de aplicaciones de clientes, con los siguientes beneficios:
  - Reduce procesamiento del sistema operativo (menos procesos de corrimiento)
  - Reduce requerimientos de memoria global.
  - Reduce la contención para los recursos dentro del sistema manejador de BD (DBMS).
- DSA proporciona más control de prioridades y tareas programáticas para la BD que el sistema operativo hace.

La figura 2.3 esquematiza estas características del servidor de base de datos.



Figura 2.3 Esquema de la Arquitectura Dinámica Escalable

## II.5 ARQUITECTURA DE PROCESAMIENTO EN PARALELO

Los servidores de BD multitareas explotan procesadores simétricos (SMP) y arquitecturas uniprocadoras.

DSA explota particularmente sistemas de computadoras (SMP symmetric multiprocessor) proporcionando los siguientes beneficios sobre esos sistemas:

- Múltiples procesos que pueden trabajar en paralelo para un cliente.
- Sobre algunas computadoras multiprocesadores pueden ligar procesos de servidores de BD para CPU's específicos.

La figura 2.4 muestra los tres principales componentes de esta arquitectura del servidor de base de datos Informix: memoria compartida (shared memory), discos y procesadores virtuales:

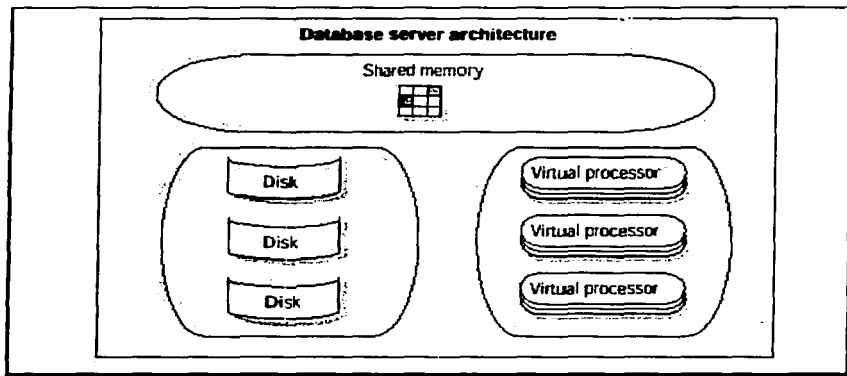


Figura 2.4 Componentes de la Arquitectura de Procesamiento en Paralelo

TESIS CON  
FALLA DE ORIGEN

### Componentes de la arquitectura de un servidor de BD

El cual incluye los siguientes componentes:

- **Procesadores Virtuales.-** Un procesador virtual que el sistema operativo programa para la ejecución sobre el CPU. Una tarea del servidor de BD es una tarea que el procesador virtual programa internamente para su procesamiento. Los procesadores virtuales del servidor de BD son multiejecuciones porque se pueden correr múltiples ejecuciones concurrentes.
- **Memoria compartida (Shared Memory) .-** La memoria compartida consiste de una porción residente y una virtual. El servidor de BD usa la porción residente para los datos cache del disco para el acceso rápido a múltiples aplicaciones del cliente. El servidor de BD de datos usa la porción virtual para mantener y controlar los recursos que el procesador virtual requiere y las lecturas grandes de blocks de acceso de datos infrecuente.
- **Componentes de disco.-** Los componentes de disco es una colección de una o más unidades de espacio en disco asignadas para el servidor de BD. Todos los datos en la BD y toda la información necesaria para mantener al sistema residente del servidor de base de datos dentro de los componentes del disco.

El servidor de base de datos multitareas maneja accesos a uno o más objetos de las BD relacionales para aplicaciones de clientes. En un objeto de una base relacional los datos son organizados en tablas que consisten de renglones y columna.

La figura 2.5 esquematiza el procesamiento en paralelo, en donde el cliente realiza una petición, de indexación, ordenamiento y recuperación y la petición es atendida de manera paralela por el cpu1, cpu2, cpu3 y cpu4.

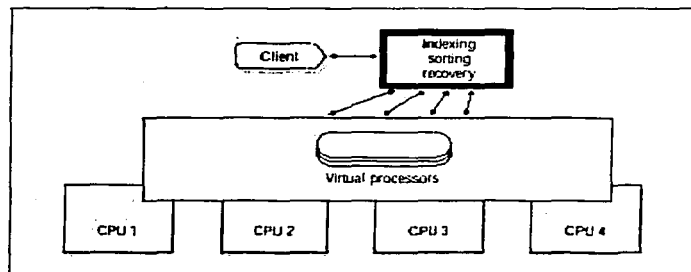


Figura 2.5 Esquema sobre como Informix lleva a cabo el procesamiento paralelo

## II.6 ESCALABILIDAD

La arquitectura dinámica Escalable (DSA) de Informix describe la capacidad de un servidor de BD para escalar sus recursos, debido a la demanda de actualización. Una llave fundamental de DSA son los procesadores virtuales que son la parte central de manejo, disco E/S, red, y funciones ópticas en paralelo.

Explota las capacidades al máximo de multiprocesadores simétricos y plataformas uniprosesor, para liberar la escalabilidad, manejabilidad y rendimiento.

Esta es una propiedad de la arquitectura dinámica escalable a través de las propiedades de disco, memoria virtual y procesadores virtuales.

## II.7 ALTO RENDIMIENTO (HIGH PERFORMANCE)

El servidor dinámico lleva su alto rendimiento a través de los siguientes mecanismos:

- Manejo de disco ó particiones crudas
- Manejo dinámico de la memoria compartida.
- Asignación dinámica de hilo (thread)
- Fragmentación y Paralelismo

### II.7.1 Manejo de discos o particiones crudas

El servidor de base de datos Informix utiliza los archivos unbuffered para mejorar el acceso de E/S. Unix soporta este tipo de particiones conocidas como raw disk devices. Cuando se almacenan tablas en este tipo de dispositivos, el servidor de base de datos controla la organización física de datos y minimiza el acceso a disco de E/S. Cuando son almacenados los datos de esta manera, se tienen las siguientes ventajas:

- El servidor de base de datos optimiza el acceso a tablas garantizando que los registros son almacenados de manera contigua.
- El servidor de base de datos es la vía de paso de entrada/salida transfiriendo directamente datos entre el disco y la memoria compartida.

Si para el sistema operativo no es de prioridad el rendimiento, puede ser configurado el servidor de base de datos para que almacene datos de manera regular ya sea en un file system especial o en un archivo buffered, conocidos comúnmente como archivos cocidos. Cuando se utilizan este tipo de archivos el servidor de base de datos no maneja el contenido de los archivos, debido a que el control de E/S a disco lo toma el Sistema Operativo.

TESIS CON  
FALLA DE ORIGEN

### **II.7.2 Manejo Dinámico de la Memoria Compartida**

Todas las aplicaciones que usan una instancia de un servidor de BD comparten datos en el espacio de memoria del servidor de BD. Después una aplicación lee datos de una tabla, otras aplicaciones pueden acceder cualquier dato en memoria. Esta compartición de datos en memoria previene redundancia de E/S al disco y la correspondiente degradación en el rendimiento.

La memoria compartida del servidor de BD contiene datos inicializados de la BD y control de información. Por lo que los datos necesitan estar en varias aplicaciones, estos son localizados en una porción de memoria compartida. Todo el control de la información necesita manejar accesos para que los datos puedan ser localizados en el mismo lugar. Los servidores de bases de datos añaden dinámicamente memoria como se requiera y el administrador puede añadir segmentos de memoria compartida si es necesario.

Es un parámetro configurable y va depender directamente de la memoria que el equipo maneja y de acuerdo a la demanda de los usuarios.

### **II.7.3 Asignación Dinámica de Hilos (Thread)**

El servidor de base de datos soporta múltiples conexiones de clientes, usando un número pequeño de procesadores, llamados procesadores virtuales. Un procesador virtual es un proceso multihilos que puede servir a múltiples clientes, donde múltiples hilos corren de manera paralela para una simple sentencia ó petición a base de datos, de esta forma el servidor de base de datos proporciona una arquitectura flexible para balancear de manera dinámica la carga en aplicaciones OLTP ( Procesamiento de Transacciones en Línea ) y DSS (Aplicaciones de soporte/decisión)

### **II.7.4 Fragmentación y Paralelismo**

El servidor de base de datos usa partición de tablas de manera local, también denominado fragmentación, para una distribución inteligente de tablas en los discos, proporcionando una mejora en el rendimiento, esto es aplicable cuando se tiene una base de datos muy grande y esta es una buena técnica para hacer más eficiente el manejo de datos.

La fragmentación de una tabla permite almacenar partes de una tabla en diferentes discos, permitiendo el almacenamiento de cantidades grandes de datos, provocando un balance de carga de datos debido a que para la recuperación de datos no es sobre un solo disco sino en múltiples.

La figura 2.5 muestra el acceso de varios clientes a una tabla fragmentada. En la primera parte, la tabla es vista por la aplicación cliente. En la parte de abajo del esquema, los fragmentos de tabla son almacenados por el servidor de base de datos



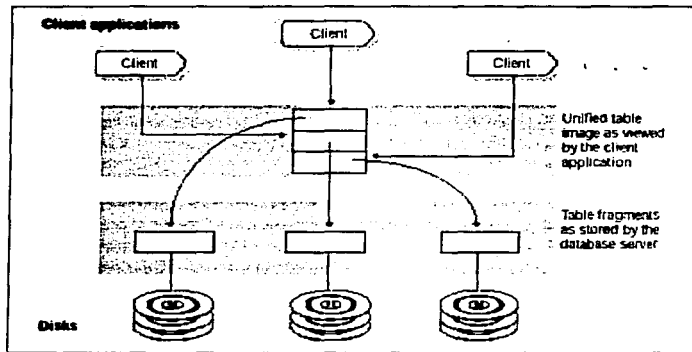


Figura 2.6 Esquema de Fragmentación y Paralelismo

### Estrategias de Fragmentación

- Un esquema de distribución.- Este tipo de fragmentación agrupa registros o llaves de índices dentro de fragmentos. La forma de realizar una fragmentación de este tipo es mediante la cláusula `FRAGMENT BY` de las sentencias `CREATE TABLE`, `CREATE INDEX` ó `ALTER FRAGMENT`.
- Asignación del dbspace donde se va a alojar algún fragmento.- La forma en como se fragmenta hacia un dbspace determinado es con la cláusula `IN` de las sentencias antes mencionadas indicando el dbspaces donde se va a alojar la información.
- Round-robin.- En este tipo de fragmentación se van colocando los registros uno tras otro en los fragmentos definidos haciendo una rotación cada que es agregado un registro. Para una sentencia `INSERT`, el servidor de base de datos usa una función hash, con un número aleatorio para determinar en que fragmento va a ser asociado el registro. Para el caso de un cursor, el primer registro asignado es de manera aleatoria pero los siguientes registros son asignados a fragmentos en secuencia. Si algún fragmento ya está lleno se salta al siguiente que encuentre vacío.

TESIS CON  
REGISTRO



- **Basado en Expresión.-** Este tipo de fragmentación mete los registros basado sobre la expresión de fragmentación que se especifica. Esta expresión está definida mediante un criterio o una regla. La expresión puede ser un rango o alguna regla arbitraria.

La figura 2.7 muestra los esquemas de particionamiento soportados por Infomix.



Figura 2.7 Tipos de particionamiento de datos en Informix

El servidor de base de datos puede alojar y asignar múltiples hilos para trabajar en paralelo, sobre una sola petición, lo cual agiliza el tiempo de respuesta, a esto se le conoce como PDQ (Parallel Database Query), que significa consulta de base de datos en paralelo, esta característica adicionada a la fragmentación hace más eficiente los procesos.

La figura 2.8 muestra un ejemplo de la ventaja de este tipo de funcionalidad.



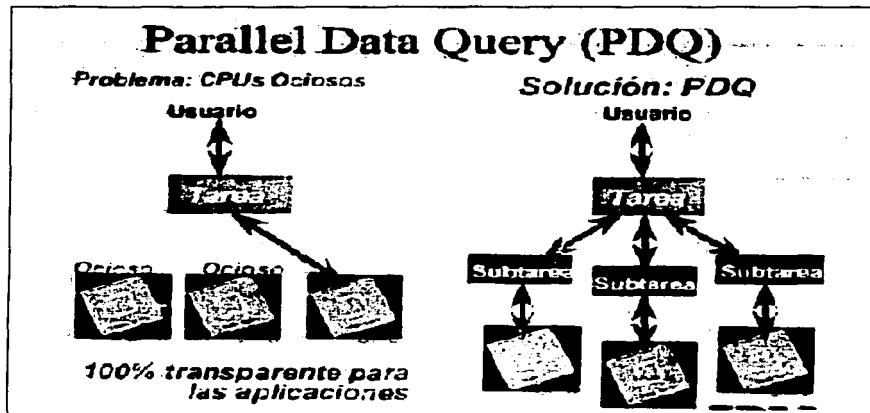


Figura 2.8 Esquema del proceso llevado a cabo en la consulta de datos en paralelo

### Paralelismo

El grado de paralelismo para una consulta se refiere al número de subplanes de ejecución que el servidor de base de datos ejecuta en paralelo para correr la consulta. Por ejemplo una unión de dos tablas con seis hilos de ejecución tiene un alto grado de paralelismo que uno de dos hilos de ejecución. A continuación se presentan las 5 formas de paralelismo que puede utilizar el servidor de base de datos dependiendo la complejidad de la consulta.

1. Scan Paralelo (Escaneo completo)
2. Join Paralelo (Uniones)
3. Sort Paralelo (Ordenamiento)
4. Aggregate Paralelo (Funciones agregadas)
5. Group Paralelo (Agrupaciones)

TESIS CON  
FALLA DE ORIGEN





## II.8 PROCESADORES VIRTUALES

Los procesos del servidor de base de datos son llamados procesadores virtuales por que funcionan en forma muy similar a un CPU de una computadora, solo que un CPU corre múltiples procesos del sistema operativo para servicios de múltiples usuarios. Un procesador de base de datos virtual corre múltiples procesos hijos (conocidos como hilos), para atender peticiones cliente a través de instrucciones SQL de varias aplicaciones.

La figura 2.9 ilustra la relación existente de las aplicaciones cliente con los procesadores virtuales; el número de procesadores virtuales casi siempre es mucho mayor que el número de aplicaciones cliente o consultas.

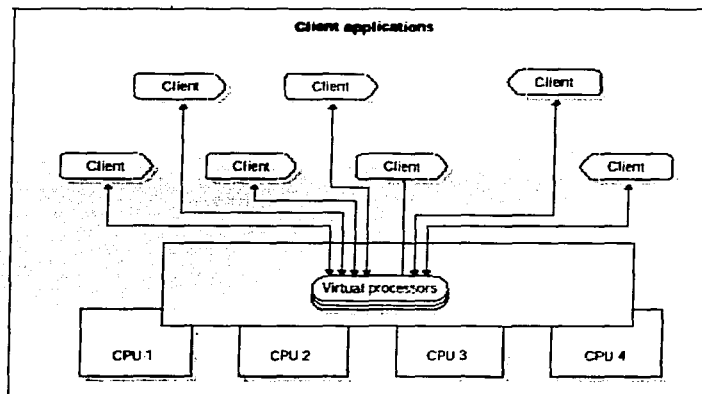


Figura 2.9 Utilización de procesadores virtuales en la conexión de clientes

### Tipos de Procesadores Virtuales

La tabla 2.1 muestra las clases de procesadores virtuales y los tipos de procesamiento que realizan. Cada clase de procesador virtual está dedicado a procesar ciertos tipos de procesos hijos.



Clase de Procesador Virtual	Categoría	Propósito
CPU	Procesamiento Central	Corre todas las sesiones de procesos hijos. Corre además algunos procesos hijos asincrónicos de E/S disponibles del kernel. Puede correr un conjunto de hilos a la vez, dependiendo de la configuración.
PIO	E/S Disco	Escribe al archivo de log físico, si es que es espacio de disco cocido.
LIO	E/S Disco	Escribe al archivo de log lógico, si es que es espacio de disco cocido.
AIO	E/S Disco	Si la entrada/salida asíncrona del kernel es usada, los procesadores virtuales realizan esta entrada/salida hacia los espacios de disco.
SHM	Red	Lleva a cabo la comunicación de memoria compartida.
TLI	Red	Utiliza la interface de la capa de transporte para llevar a cabo la comunicación.
SOC	Red	Utiliza sockets para llevar a cabo la comunicación.
OPT	Optico	Lleva a cabo entradas/salidas hacia discos ópticos.
ADM	Administración	Realiza funciones administrativas.
ADT	Auditoría	Realiza funciones de Auditoría.
MSC	Misceláneos	Requiere servicios para llamadas al sistema que necesitan gran almacenamiento.

Tabla 2.1 Clases de Procesadores Virtuales

TESIS CON  
FALLA DE ORIGEN

## II.9 EXTENSIBILIDAD

El servidor de base de datos tiene la capacidad de aceptar tipos de datos definidos por el usuario. La combinación de un nuevo tipo de dato y rutinas es a lo que se le denomina datablade.

Tipos de extensibilidad permitidos por informix:

- Tipo de datos definidos por el usuario.
- Acceso y almacenamiento de objetos grandes simples e inteligentes.
- Rutinas definidas por el usuario.
- Módulos datablade.
- Procesadores virtuales definidos por el usuario.
- Métodos de accesos definidos por el usuario.

La figura 2.10 ilustra la extensibilidad de módulos datablades añadidos a un servidor de base de datos.

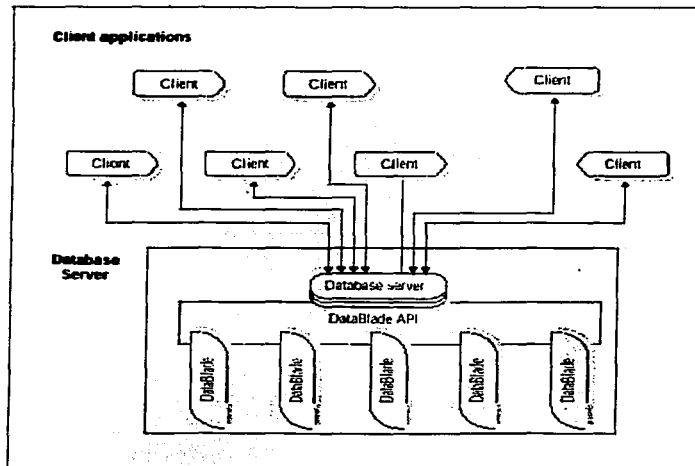


Figura 2.10 Manejo de la Extensibilidad (Datablade) en Informix

## II.10 MÚLTIPLE RESIDENCIA

Se puede utilizar un servidor de base de datos en alguna de las 2 siguientes formas:

- Correr múltiples instancias del servidor de base de datos en un solo equipo (host).
- Accesar varios servidores de bases de datos en toda la red.

Cuando múltiples servidores de bases de datos, su memoria compartida y estructuras de disco coexisten en un solo equipo, se le conoce como residencia múltiple.

### *Beneficios:*

Creando ambientes independientes de bases de datos en un mismo host, permite ejecutar las siguientes acciones:

- Separar ambientes de producción y ambientes de desarrollo
- Aislar bases de datos sensitivas
- Probar transacciones de datos distribuidas en un solo host

Cuando se utiliza la múltiple residencia, cada servidor de base de datos, tiene su propio archivo de configuración, por lo que se puede crear un archivo de configuración para cada servidor de base de datos que se requiera para realizar su mantenimiento por separado como son respaldos, uso de memoria compartida y prioridades de afinación o tuning.

Es posible separar ambientes de producción y desarrollo, para proteger el sistema. Podría ser además muy usual el aislar aplicaciones de bases de datos que sean críticamente importantes.

Si se está desarrollando una aplicación para utilizarla en una red, se puede utilizar la dirección de loopback para la simulación de datos distribuida y probarlo en una sola computadora. Después, cuando la red esté lista, se puede utilizar la aplicación, sin hacer cambios en el código fuente de la aplicación.

Sin embargo, corriendo múltiples servidores de bases de datos en un mismo host no es tan eficiente que si se corriera uno solo, por lo que es necesario balancear las ventajas al separar servidores de bases de datos contra el costo extra de performance.

### **Funcionamiento de la Múltiple Residencia**

La residencia múltiple es posible, ya que el sistema operativo puede mantener áreas separadas de memoria compartida y en disco para cada instancia del servidor de base de datos; cada instancia pasa un valor al sistema operativo. Este valor, el cual es una función del parámetro de configuración **SERVERNUM**, especifica las direcciones de memoria compartida las cuales el proceso del servidor de base de datos debe asignar para cada una de las instancias.

Se debe especificar un solo nombre de servidor de base de datos, así como una sola locación de almacenamiento para cada servidor de base de datos.



**Variable de Ambiente ONCONFIG**

Los parámetros en el archivo de configuración **ONCONFIG**, describen cada instancia del servidor de base de datos especificando el nombre del actual archivo de configuración **ONCONFIG**. Los siguientes parámetros deben tener valores únicos para cada servidor de base de datos.

DBSERVERALIASES  
 DBSERVERNAME  
 MIRRORPATH  
 MSGPATH  
 ROOTPATH  
 SERVERNUM

**Parámetros de Configuración***SERVERNUM*

Para mantener la separación entre las instancias de los servidores de base de datos, se deben mantener múltiples archivos de configuración, con un valor único de **SERVERNUM**.

Cuando se inicializa el servidor de base de datos, este lee la variable de ambiente **ONCONFIG**, para el nombre del archivo de configuración. Como siguiente paso, el servidor de base de datos lee este archivo para obtener el valor del parámetro de configuración **SERVERNUM**. El servidor de base de datos puede entonces utilizar este valor para calcular las direcciones de memoria compartida requeridas.

Por ejemplo, los archivos de configuración para dos servidores de bases de datos en una plataforma Unix pueden contener los siguientes parámetros:

<i>ONCONFIG file: onconfig.1</i>	<i>ONCONFIG file: onconfig.2</i>
<i>DBSERVERNAME dbsvr_1</i>	<i>DBSERVERNAME dbsvr_2</i>
<i>SERVERNUM 1</i>	<i>SERVERNUM 2</i>
<i>ROOTPATH /dev/area1</i>	<i>ROOTPATH /dev/area2</i>

En caso de que el servidor utilice nombres alias se deberá de asignar de igual forma para cada una de las instancias.

**Ejemplo:** En la figura 2.11 se utilizan los archivos de configuración mostrados anteriormente y que proveen un ejemplo de residencia múltiple. Cada servidor de base de datos tiene su propio nombre y su propia sección de memoria compartida, así como su propia área de almacenamiento en disco.

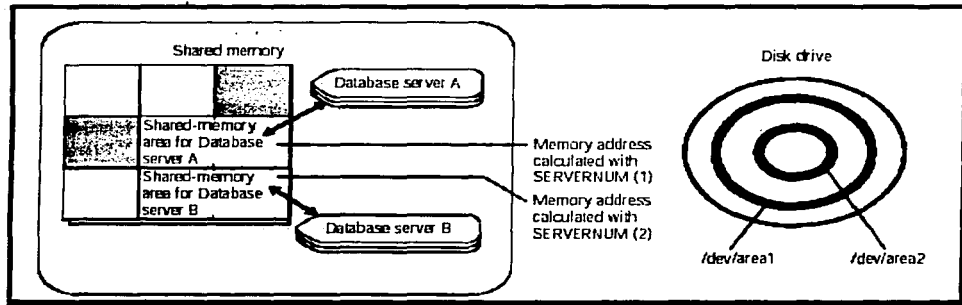


Figura 2.11 Esquema de Múltiple Residencia

### Planeación para la Múltiple Residencia

Cuando se planea utilizar esta característica de Informix Dynamic Server denominada multiple residencia en un servidor se deben considerar los siguientes factores:

#### - Espacio de almacenamiento.

Cada servidor de base de datos debe tener su propio espacio de almacenamiento, no es posible utilizar un mismo espacio en disco para más de una instancia. Cuando se prepara un servidor de base de datos adicional, es necesario repetir algo de la planeación que se hizo cuando se instaló el primer servidor. Es necesario considerar los siguientes cuestionamientos:

- ¿Se utilizarán archivos con buffer o sin buffer?
  - ¿Los archivos sin buffer comparten partición de disco con otras aplicaciones?
  - ¿Se van a utilizar espejos (mirror)?
  - ¿Dónde residirán los espejos (mirror's)?
  - ¿Dónde residirá la bitácora de mensajes?
- Es posible dedicar una unidad de cinta al servidor de base de datos, para estos logs lógicos.
- ¿Que tipo de respaldos se realizarán?

## - Memoria

Cada servidor de base de datos tiene su propia memoria. A continuación se presentan algunos pasos a seguir cuando se desea crear un servidor adicional de base de datos, o múltiple residencia en un Host.

### Procedimiento para la creación de una nuevo instancia en Unix

#### *Creación de múltiple residencia de un servidor de base de datos*

##### 1.-Se crea un nuevo archivo de configuración onconfig

Cada instancia del servidor de base de datos debe tener su propio archivo de configuración **ONCONFIG**. Se realiza una copia del archivo de configuración **ONCONFIG** que contiene las características básicas que se desean para el nuevo servidor de base de datos. Se asigna un nombre de archivo el cual sea fácil de asociar con esta función. Por ejemplo, se puede elegir el nombre de archivo onconfig.prod\_conta para indicar el archivo de configuración de un sistema en producción que contiene información de contabilidad.

##### 2.-Se coloca la variable de ambiente **ONCONFIG** en el nuevo archivo de configuración

La variable de ambiente **ONCONFIG** se coloca en el archivo profile del usuario con el nombre del nuevo archivo de configuración (ONCONFIG), especificando solo el nombre del archivo, no la ruta completa ya que por default la trayectoria de los archivos de configuración son reconocidas por informix en el directorio donde se instaló, en el subdirectorio \$Dir\_Informix/etc .

##### 3.-Modificación del nuevo archivo de configuración **ONCONFIG**

Para editar el archivo de configuración nuevo, se puede utilizar un editor de texto como vi de Unix u ON-Monitor (Utilería de administración de Informix).

##### 4.-Modificación del archivo sqlhosts para incluir el dbservername de los nuevos servidores de bases de datos

El archivo sqlhost debe tener una entrada para cada servidor de base de datos. Si otros productos de informix instalados en otras máquinas accesan una instancia de este servidor de base de datos, los administradores de estas máquinas deben actualizar sus archivos de configuración de sqlhosts.

Si se va hacer uso de conexiones de tipo TCP/IP con alguna instancia de un servidor de base de datos, el administrador de la red debe actualizar el archivo hosts de sus sistemas y el archivo services respectivamente.

##### 5.-Inicialización del espacio en disco para el nuevo servidor de base de datos.

Antes de inicializar el espacio en disco, se debe verificar que esté colocada la variable de ambiente en el archivo ONCONFIG .



Si no están colocadas correctamente, se corre el riesgo de sobrescribir datos de otro servidor de base de datos.

Cuando se inicializa espacio en disco, el servidor de base de datos inicializa el espacio en disco especificado en el actual archivo de configuración.

### **Asegurar propiamente respaldos, inicialización y conexión de clientes**

#### 1.-Se preparan los dbspaces y los calendarios de respaldos

Cuando se utiliza múltiple residencia se debe mantener sbspaces separados y respaldos de log lógicos para cada instancia del servidor de base de datos. Cuando se realizan respaldos de logs lógicos o de dbspaces con múltiple residencia, se debe compartir el dispositivo de respaldo.

Se puede colocar un dispositivo de cinta dedicado para cada servidor de base de datos, de otra forma se puede calendarizar el respaldo de los diferentes dbspaces que se tengan

#### 2.-Modificación del archivo de boot (inicialización) del sistema operativo

Para inicializar un script para una simple instancia de un servidor de base de datos, se deben de colocar las variables de ambiente **INFORMIXDIR**, **PATH**, **ONCONFIG**, e **INFORMIXSERVER** y posteriormente ejecutar la instrucción oninit (Da de alta el Servidor de Base de datos). Para inicializar una segunda instancia de un servidor de base de datos, se cambian las variables de ambiente **ONCONFIG** y **INFORMIXSERVER** en el archivo de configuración del segundo servidor de base de datos y después se ejecuta la utilería oninit nuevamente.

No se deben cambiar las variables de ambiente **INFORMIXDIR** ó **PATH**, similarmente, se puede solicitar al administrador del sistema el modificar el programa shutdown para incluir un script que dé de baja las instancias del servidor de base de datos en forma exitosa.

#### 3.-Finalmente se verifican las variables de ambiente de **INFORMIXSERVER** para usuarios

Si una nueva instancia de la base de datos debe ser por default el servidor de base de datos, los usuarios necesitan asignar su variable de ambiente **INFORMIXSERVER** para la nueva instancia.





## II.11 TAREAS DE ADMINISTRACIÓN DE BASE DE DATOS

De la misma manera que se tienen diferentes roles para la administración de una base de datos, existen de igual manera diferentes tareas administrativas que se llevan a cabo y que van desde la planeación, configuración y mantenimiento de la base de datos.

En este punto detallamos cuáles con las actividades realizadas en torno a una base de datos de tipo relacional.

Las tareas administrativas del servidor de base de datos requieren que se tengan los privilegios acordes para llevar a cabo la administración.

### *Tareas Iniciales*

Cuando se tiene primeramente el servidor de BD se necesita realizar algunas instalaciones iniciales y tareas de configuración.

### *Tareas de Configuración*

Las tareas de configuración son generalmente tareas de organización que envuelven funcionalidad o mantenimiento y ajustes de rendimiento tareas que son requeridas para el uso del servidor de BD.

### *Manejo de Espacio en Disco*

Es responsabilidad del administrador planear e implementar el esquema de información manejado por el servidor de BD sobre discos. El camino de distribución de los datos puede afectar de manera considerable el funcionamiento del servidor de BD.

### *Manejo del estatus de la base de datos*

Implica el cambio de modo de operación de la base de datos en cuanto al proceso de transacción ya que permite tener la base en modo login lo que facilita la recuperación de errores mediante el rollback o el manejo de la base en modo no login que para el uso de procedimientos que no requieren transacciones incrementa el rendimiento y rapidez en cuanto a tiempo de respuesta por lo que el administrador decidirá en que momento o en que proceso requiere el cambio de estatus de la BD.

### *Manejo de Logs Lógicos*

A pesar de que el respaldo de los logs lógicos es una tarea rutinaria, la administración de los mismos (colocación y tamaño de los logs así como señales de desbordamiento) es requerida incluso cuando no está la BD en modo login.

### *Manejo de los logs físicos*

El administrador puede cambiar el tamaño y localización del log físico como parte del manejo efectivo de discos.

### *Uso de Espejos (Mirror)*

**Nota:** El mirror consiste en tener redundancia de datos u objetos en un sistema.

El uso de espejos va a permitir al administrador tener una réplica de una o varias áreas que él crea convenientes y que tendría un efecto grave en caso de que la información ahí contenida se perdiera, esto quiere decir que el espejo contiene toda la información que se tiene en el área especificada con la finalidad de que si el área se daña se pueda activar el espejo para continuar el proceso.

### *Uso de Replica de datos*

El uso de replicación de datos por parte del administrador proporciona una alta disponibilidad de la información, este proceso permite duplicar una base de datos entera hacia otro servidor de base de datos proporcionando un estado de resguardo en caso de problemas irremediables en la falla de un servidor.

Informix soporta replicación de datos asincrónica y permite seleccionar los datos exactos que se requieren replicar.

### *Manejo de la Memoria Compartida*

El manejo de la memoria compartida incluye las siguientes tareas:

- Cambio del tamaño o número de buffers (cambiar el tamaño del buffer de los logs físicos o lógicos), para la agrupación de la memoria compartida.
- Cambiar el valor de los parámetros de la memoria compartida, si es necesario.
- Cambio de forzar residencia (encendido o apagado temporal para las sesiones)
- Intervalos de revisión de checkpoint.
- Añadir segmentos para la porción virtual de la memoria compartida.

### *Manejo de procesadores virtuales*

La configuración y manejo de los procesadores virtuales (VPs) tiene un impacto directo sobre el rendimiento de la BD, el número de procesadores virtuales va a depender del hardware y del tipo de aplicaciones que soporte el servidor de base de datos.

### *Manejo de consultas paralelas a bases de datos*

El administrador puede controlar los recursos que usan las bases de datos para el funcionamiento de las consultas en paralelo. El cual requiere el balance de requerimientos de consultas contra las consultas de los procesos de transacciones en línea (OLTP). Los recursos que se necesitan considerar son memoria compartida y espacios de tablas temporales.



## Tareas Rutinarias

### Cambios de estatus de la BD

El administrador se encarga de colocar el servidor de base de datos en modo fuera de línea (Off-line) lo cual significa que se va a dar de baja de manera completa, o colocar la BD en modo Monousuario (Quiescent) el cual permite darle el mantenimiento al manejador, por lo tanto el único usuario que puede acceder es el administrador y por último el modo en línea (On-line) en este modo todos tienen acceso a la BD contemplando las restricciones que el administrador haya asignado.

### Respaldo de datos y Archivos Logs Lógicos

Para asegurar que se podrá recuperar las BD en caso de fallas informix recomienda hacer respaldos frecuentes de espacios de almacenamiento y logs lógicos.

Cuando se llevan a cabo respaldos o recuperaciones de información con herramientas propias de Informix suelen ocurrir errores de E/S, lo que ocasiona que falle una transacción de este tipo. A diferencia de otros manejadores de bases de datos, Informix puede reiniciar las tareas de recuperación o restauración a partir de donde ocurrió el error y no desde el inicio, lo cual incrementa en gran medida el tiempo de respuesta a una solicitud de este tipo; la figura 2.12 ejemplifica el proceso de restauración cuando ocurre un error.



Figura 2.12 Restauración de información en un servidor de BD Informix

De manera programática se deben realizar respaldos completos (nivel-0) cada semana, respaldo a nivel 1 de manera diaria y respaldo de nivel 2 cada hora, esto a consideración de cada administrador y dependiendo las características del sistema.

A continuación se presenta un resumen de las tareas administrativas

#### *Tareas Iniciales*

- Diseño para el servidor de base de datos
- Pre - instalación del sistema operativo y cambios
- Instalación del servidor de base de datos
- Configurar el servidor de BD
- Definir conectividad
- Inicialización del servidor de BD

#### *Tareas de configuración*

- Manejo de espacio en disco y espacio de almacenamiento
- Análisis de configuración del disco
- Cambio de estatus a la base de datos
- Administrar los logs lógicos
- Tamaño y colocación de los logs lógicos
- Modificar la configuración de los logs lógicos
- Administración de los logs físicos
- Manejo de Espejos
- Replicación de Datos
- Manejo de la memoria compartida
- Manejo de los procesadores virtuales
- Manejo de la consulta de datos en paralelo (PDQ)
- Migración a diferentes servidores de datos

#### *Tareas Rutinarias*

- Cambios del modo de operación del servidor de BD
- Actualización de datos y tablas de almacenamiento
- Respaldo de datos y archivos de logs lógicos
- On - Archive (UNIX)
- ontape
- On - Bar
- Manejo de almacenamiento de Informix
- Verificación de respaldos
- Funcionamiento externo de respaldos y restauraciones
- Monitoreo de la actividad del servidor de BD
- Revisión de consistencia de datos
- Afinación del funcionamiento del servidor de BD



En el siguiente tema se contemplan aspectos relacionados a la configuración del Sistema Operativo en cuanto a los parámetros que el manejador de bases de datos requiere para su funcionamiento, como lo es el manejo de semáforos, memoria compartida, etc., así como las variables de entorno, y para el caso del servidor de base de datos, la instalación y configuración para que funcione el sistema SIAPSEP.

## TEMA III

### CONFIGURACIÓN REQUERIDA PARA IMPLANTACIÓN DEL SISTEMA SIAPSEP EN INFORMIX

Este tema está conformado por dos subtemas generales, los cuales se consideran importantes para poder cumplir los requisitos del sistema SIAPSEP y son el Sistema Operativo e Informix Dynamic Server.

Antes de iniciar la configuración del servidor de base de datos, como requisito se debe haber configurado apropiadamente el sistema operativo, ya que es el que permite administrar los programas de instalación y configuración del manejador de base de datos informix

#### III.1 SISTEMA OPERATIVO UNIX

Dentro de la preparación del sistema operativo para la instalación de productos (Informix), unas actividades son realizadas previas y otras durante la instalación, a continuación se listan estas actividades:

*Configuración previa a la instalación de Informix.*

- Crear un usuario para realizar la instalación.
- Verificar que se cuenta con los requerimientos mínimos del sistema operativo y disponibilidad de espacio en disco.

*Configuración durante la instalación de informix*

- Modificación de los parámetros del Kernel.
- Alojamiento de espacio en disco.
- Creación de dispositivos crudos o archivos sin buffer.
- Creación de nombres de dispositivos estándar.
- Asignación de permisos, propietarios y grupos.

#### **Configuración Previa a la Instalación**

- *Creación de Cuentas de Instalación*

Antes de realizar la instalación de informix IDS 2000 ver. 9.2 se debe crear un usuario para llevar a cabo este procedimiento, así como asignarle un espacio en disco en donde van a residir todos los archivos de instalación.

### **Procedimiento para crear una cuenta en UNIX.**

Para tener control de las cuentas de usuarios creadas y los roles que tienen especificados, se crean grupos con características distintas, como pueden ser para administradores del Sistema Operativo, Administradores de Bases de Datos, Desarrolladores, Operadores, etc.

Para poder crear un grupo se utiliza la siguiente instrucción desde la vía de comandos:

```
#groupadd -g gid -o nombre del grupo
```

gid - Es el identificador del nuevo grupo, este identificador debe ser un número decimal no negativo a partir del número 100 debido a que del 0-99 son identificadores de grupo reservados para el sistema operativo.

Creado el grupo el siguiente procedimiento es crear el usuario:

```
#useradd -u uid -g grupo -d directorio -s shell -c comentario -mk skel_dir login
```

uid - Es el identificador del nuevo usuario, el cual no debe ser un número negativo y debe ser mayor al identificador del último usuario creado.  
Este valor debe ser > a 100 ya que de 0 a 99 se reservan para el sistema mismo

grupo - Debe ser un identificador de grupo existente.

directorio - Es el directorio en donde va a residir el usuario al momento de acceder.

shell - Pueden ser cualquiera de los siguientes sh,csh,ksh.

Comentario- Este comentario le sirve al administrador para ubicar más fácilmente a los usuarios.

skel\_dir - Sirve para inicializar al usuario colocando en el archivo profile valores estándar y se debe introducir el directorio donde reside esa información, por lo regular se encuentra ubicado en /etc/skel.

Al dar de alta un usuario en el sistema, es necesario asignarle un password, esto se lleva a cabo con la siguiente instrucción:

```
#passwd usuario  
Usuario - Es un login válido en el sistema.
```

### **III.1.1 Modificación de los Parámetros del Kernel de Unix**

Así como el manejador de informix se configura a partir de información especificada y definida por el sistema operativo, este también requiere saber datos técnicos que el manejador recomienda y que básicamente tiene que ver con la memoria que el sistema operativo va a compartir así como otros tipos de datos denominados semáforos.

Estos datos técnicos se proporcionan de acuerdo a la plataforma sobre la cual se lleva a cabo la instalación (HP, Sun, Linux ó Windows NT), localizados en uno de los directorios



en donde se descargan los programas para la instalación de informix con la siguiente trayectoria \$DIR-INFORMIX/release/en\_us/0333/IDS9.20. En general, el espacio disponible para la memoria compartida mejora el funcionamiento del mismo.

A continuación se presentan los parámetros que son añadidos al kernel del sistema operativo.

Como ya se mencionó, el kernel es el núcleo principal del sistema operativo, y tiene las siguientes características:

- Encargado de activar las interrupciones necesarias.
- Es una parte de UNIX que permite la existencia de un S.O. multiusuario, multitarea y multiproceso.
- Funciona como un asignador de recursos para cualquier proceso de usuario.
- Calendarizador y despachador de procesos.
- Administración de espacio de archivos y direccionamiento.
- Comunicación entre usuarios y procesos.
- Administración de E/S y de periféricos.
- Respuesta a las llamadas del sistema.

#### *Carga de Parámetros al Kernel*

La forma más rápida de que un proceso tenga la misma información que otro es no mover los datos entre procesos sino compartílos. El que envía S (Sender) y el que recibe R (Receiver) comparten alguna zona de memoria principal, y cuando el dato es colocado aquí por el S, el dato es instantáneamente disponible para ser utilizado por el R. Un semáforo o mensaje deberá ser utilizado para evitar que el R no lea los datos antes que los mismos estén disponibles y prevenir que el S escriba nuevos datos antes que los anteriores hayan sido leídos por el R. Unix soporta shared memory entre un conjunto (cualquiera) de procesos. La memoria compartida es llamada segmento. Cada proceso puede tener diferentes zonas de memoria compartida entre diferentes subconjuntos de procesos y cada proceso podrá acceder a las zonas de memoria que comparte. El segmento de memoria es creado al inicio fuera del espacio de direcciones de cada proceso, por ello cada proceso que desee acceder a esta zona compartida ejecutará una llamada al sistema para "mapear" esta zona dentro de su espacio de direcciones. Una vez realizada esta operación, acceder a la zona compartida es equivalente a acceder a una variable local del proceso.

Los parámetros a cargar referentes al Kernel, para la utilización de la memoria compartida por parte de informix, son mediante la utilidad sam de equipos HP ó para equipos Sun de manera estándar, en el archivo system que se encuentra en la trayectoria /etc/system.

Para que los cambios se vean reflejados se tiene que reinicializar el servidor.

El siguiente archivo muestra los parámetros que Informix recomienda cargar en el kernel del sistema operativo.



## Archivo Online

=====

MACHINE SPECIFIC NOTES  
FOR  
INFORMIX-OnLine PRODUCT RELEASE


=====

PORT NUMBER: 120285(DAT),120292(TAPE CARTRIDGE),120299(TAPE QIC  
525),120305(CD ROM)

INFORMIX-OnLine Dynamic Server Shared Memory and Kernel Parameters:

=====

Estos son los Parámetros que se cargan en el Kernel



SHMMAX: 67108864  
SHMMIN: 1  
SHMALL: Not applicable  
SHMMNI: 256  
SHMSEG: 120  
SHMBRK: Not applicable  
MAXMEM: Not applicable  
SEMMNI: 4096  
SEMMNS: 4096  
SEMMSL: Not applicable  
NFILES: 790

LOCATION OF SHARED MEMORY:  
=====

The ONCONFIG variable SHMBASE should be set to the following:

ANSI C compiler cumulative patch

*Nota:* El parámetro *Shared Memory Max* debe ser = ó > que el total de la memoria real o memoria RAM. Esto tomando en cuenta la cantidad que se requiere para inicializar cada instancia, así como para lo necesario para el crecimiento de la porción virtual.

### *Parámetros de Configuración del Kernel*

La utilización de mensajes para sincronizar procesos, si bien es mucho más eficiente que utilizar archivos, introducen una carga considerable en el sistema ya que el contenido del mensaje se 'mueve' entre los procesos. Una alternativa más eficiente sería incrementar/decrementar en forma controlada una variable. Unix posee un conjunto de llamadas específicas para trabajar con estas variables denominadas semáforos.

Los Semáforos son recursos del kernel con un tamaño de 1 byte cada uno.

Cada instancia de Dynamic Server (Informix) requiere el siguiente conjunto de semáforos:

- Un conjunto para cada grupo de más de 100 procesadores virtuales (VPs) los cuales son inicializados con Dynamic Server.
- Un conjunto para cada VP adicional que se pueda añadir dinámicamente mientras Dynamic Server está corriendo.
- Un conjunto para cada grupo de 100 o menor número de sesiones conectadas a través de interfaces de comunicación de memoria compartida.

Debido a que las utilerías como **onmode** utilizan conexiones de memoria-compartida, se debe haber configurado un mínimo de 2 conjuntos de semáforos para cada instancia de Dynamic Server; una para el conjunto inicial de Procesadores Virtuales, (VPs) y una para las conexiones de memoria compartida que las utilerías de Dynamic Server utiliza.

El parámetro de configuración de Sistema Operativo **SEMMNI**, típicamente especifica el número de conjuntos de semáforos a alojar.

El parámetro **SEMMSL** especifica básicamente el número máximo de semáforos por grupo. Como mínimo este parámetro debe ser de 100.

Algunos sistemas operativos requieren que se configure el número máximo de semáforos a través de todos los grupos. Típicamente son dados a través del parámetro de configuración **SEMMNS**. La siguiente formula se utiliza para calcular el número total de semáforos para cada instancia que Dynamic Server requiere.

**SEMMNS = init\_vps + added\_vps + (2 \* shmem\_users)+concurrent\_utils**

*init\_vps* - Es el número de VP's que son inicializados con Dynamic Server, incluyendo CPU, PIO, LIO, AIO, SHM, TLI, SOC, y ADM VP's

**CPU** es la identificación de los procesadores  
**PIO** es un procesador virtual que se encarga de la I/O en el archivo de log físicos  
**LIO** es un procesador virtual que se encarga de la I/O en el archivo de log lógico  
**AIO** es un procesador virtual que se encarga kernel-asynchronous I/O  
**SHM** es el parámetro para utilizar la memoria compartida en las conexiones de red  
**TLI** es un programa de interfaces de red (Transport Layer Interface).  
**SOC** es otro programa de interfaz de red basado en sockets.  
**ADM** es el administrador de procesadores virtuales.

### *Configuración de los Parámetros de Memoria Compartida*

1.- Si el sistema operativo no tiene límite de tamaño para segmentos de memoria compartida, se consideran los siguientes criterios:

- a.- El parámetro de configuración de SHMMAX o SHMSIZE debe ser igual o mayor al requerido por la configuración del Servidor Dinámico.  
Este tamaño incluye la cantidad de memoria que es requerida para inicializar la instancia de Servidor, y la cantidad que se asigna para el crecimiento de la porción virtual.
- b.- Se coloca el parámetro de configuración para el número máximo de segmentos, conocido como SHMMIN, como un mínimo de 1 por instancia del Servidor Dinámico.

2.- Si el sistema operativo tiene un tamaño límite de segmento, se toman las siguientes acciones:

- a.- Colocar el parámetro de configuración para el tamaño máximo de segmento, conocido como SHMMAX o SHMSIZE, al mismo valor que el sistema presenta.
- b.- Utilizar la siguiente fórmula, para calcular el número de segmentos para una instancia del Servidor Dinámico:

$$\text{SHMMNI} = \text{total\_shmем\_size} / \text{SHMMAX}$$

**total\_shmem\_size** Es la cantidad total de memoria compartida que se puede colocar para que Dynamic Server pueda utilizar.

- 3.- Colocar el parámetro de configuración SHMMNI o de máximo número de segmentos, al valor que rinde la cantidad total de memoria compartida para Dynamic Server.  
Si la computadora o Servidor es dedicada a una sola instancia de Dynamic Server, entonces el total puede ser arriba del 90 % del tamaño de la memoria virtual (Memoria Física + Área de swap)
- 4.- Si el sistema operativo utiliza el parámetro de configuración SHMSEG, el cual indica el número máximo de segmentos de memoria compartida que el procesador puede "cachar". Se debe colocar este parámetro a un valor igual o mayor que el número de segmentos que se pueden colocar para una instancia de Dynamic Server.

### **III.1.2 Asignación de Espacio en Disco**

La configuración del disco es una de las tareas más importantes para obtener óptimos rendimientos con data marts y data warehouses que son las nuevas tendencias en el manejo de la información, que básicamente es tener en la red la información y desde cualquier parte poder manipularla en línea y de manera transaccional. La E/S al disco es otra de las partes que influye en el buen funcionamiento y accesos rápidos y constantes que es donde se da la porción más prolongada del tiempo de respuesta de una operación



SQL. El servidor de base de datos ofrece accesos paralelos a múltiples discos sobre una computadora por lo que el manejo y configuración de las áreas de disco tendrán que configurarse ya sea con áreas crudas ó sistemas de archivos (file system) como se mencionará más adelante.

Una óptima distribución en cuanto a estas áreas de disco influye de manera directa en el performance de la base de datos.

Antes de asignar el espacio en disco se requiere saber la configuración actual para planear los espacios disponibles en el equipo en donde se va a crear el ambiente para informix:

Para el almacenamiento en disco del servidor de BD se deben considerar los siguientes pasos:

- Configurar un dispositivo crudo o crear un archivo unbuffered (sin buffer) para cada disco.
- Crear nombres de dispositivos estándar; esto quiere decir manejar una misma nomenclatura.
- Asignar propietarios, permisos y grupos para cada dispositivo crudo o archivo sin buffer.

### III.1.3 Creación de Dispositivos Crudos ó Archivos sin Buffer

Para efectuar el mejor rendimiento UNIX utiliza dispositivos de disco crudos y Windows NT usa archivos NTF unbuffered, que son buffers utilizados como vía de paso de entrada y salida al disco (E/S) para el normal funcionamiento del sistema operativo.

Así mismo el sistema operativo utiliza el concepto de volumen lógico y otros usan la unidad lógica, cada uno de estos términos representa la unidad básica del espacio de disco físico que el sistema operativo reserva para el servidor de base de datos.

En términos del manejador de BD Informix, las áreas crudas van a recibir nombres como dbspace o chunk en donde el primero es un área que su tamaño no debe de exceder de 2GB y el segundo es una partición de un disco, que tiene que ser asignado a un dbspace para ampliar su tamaño.

#### Archivo buffer (File System)

Es la colección de archivos y directorios usados para almacenar y organizar la información.

Es una partición que ha sido formateada en bloques de datos y contiene una estructura de tablas definiendo la localización de archivos y directorios

#### Archivo sin Buffer (Partición Cruda RAW)

Una partición cruda tiene definido el inicio y el tamaño, pero no tiene un Sistema de Archivos instalado.

TESIS CON  
FALLA DE ORIGEN

Una partición puede ser usada como un dispositivo crudo y como swap

Los nombres lógicos para distinguirlos son:

Archivo Sin Buffer	Archivo Buffer
/dev/rdisk	/dev/dsk

### III.1.4 Creación de Nombres de Dispositivos Estandar

Informix recomienda utilizar ligas simbólicas abreviadas para cada dispositivo de disco crudo. El utilizar ligas simbólicas, permite reemplazar un disco físico que haya fallado, por un nuevo disco asignando a la liga simbólica en otra área del disco.

Para crear una liga entre el nombre de dispositivo de carácter especial y otro nombre de archivo se usa el comando que permite hacer ligas en UNIX, que es usualmente el comando **ln**.

Una liga es un apuntador hacia un archivo o directorio, básicamente existen 2 tipos de ligas: Ligas duras y ligas suaves, las diferencias entre estas se mencionan a continuación:

- **Liga Dura**
  - Se crea con: `#ln <archivo_fuente> <archivo_destino>`
  - No apunta a directorios
  - No apunta a otros Sistemas de archivos (Partición)
  - Es idéntica al archivo fuente y no crea espacio en disco duro
- **Liga Suave (Simbólica)**
  - Se crea con: `#ln -s <archivo_fuente> <archivo_destino>`
  - Si apunta a directorios
  - Si apunta a otros Sistemas de archivos (Partición)
  - Ocupa espacio en disco duro

Nota: La liga únicamente es un apuntador o referencia

La liga suave se realiza ejecutando el comando **ln -s** en UNIX sobre el directorio en donde va a apuntar, verificando el inicio del dispositivo y la liga existente. Si el sistema operativo no soporta las ligas simbólicas se pueden utilizar ligas a discos duros, para poder ver las ligas simbólicas se ejecuta el siguiente comando **ls -lg** sobre el directorio en donde fueron creadas. El siguiente ejemplo muestra cómo se crean las ligas para dispositivos crudos, así como el despliegado de las mismas.

```
# ln -s /dev/raw/datos dbspace_datos
# ls -lg
crw-rw--- /dev/rxy0h
crw-rw--- /dev/rxy0a
lrwxrwxrwx /dev/my_root@->/dev/rxy0h
lrwxrwxrwx/dev/raw_dev2@->/dev/rxy0a
```

TESIS CON  
FALLA DE ORIGEN

### III.1.5 Asignación de Permisos Propietarios y Grupos

Los archivos o dispositivos crudos que el servidor de base de datos utiliza tienen un propietario y permisos apropiados.

Sobre el sistema operativo UNIX el propietario y el grupo deben ser de informix y los permisos deben de estar asignados para lectura y escritura para el usuario y el grupo pero no para otros.

Con la cuenta de root y la utilización del comando `chown` se indica quien o quienes van a ser los propietarios de un archivo o área como sigue:

```
$ chown -R informix:informix /dev/vg00/rdbtemp
$ chown -R informix:informix /dev/vg00/rdbdats
$ chown -R informix:informix /dev/vg00/rdbroot
$ chown -R informix:informix /dev/vg00/rdblogs
```

Nota: La opción `-R` es equivalente a recursividad.

Después de que `informix` es el propietario de las áreas corresponde dar los permisos para que se pueda leer y escribir realizándolo de esta forma:

*Cambiar a las ligas los permisos de lectura-escritura*

Posicionados en el directorio `$INFORMIXDIR` se utiliza el comando `chmod` que permite que un usuario pueda escribir, leer o ejecutar indicándolo con las siguientes letras `w`, `r` y `x` respectivamente como a continuación se muestra:

```
$ chmod a +rw lrdbtemp
$ chmod a +rw lrdbdats
$ chmod a +rw lrdbroots
$ chmod a +rw lrdblogs
```

ó de igual manera, en notación octal:

```
$ chmod 660 lrdbtemp
$ chmod 660 lrdbdats
$ chmod 660 lrdbroots
$ chmod 660 lrdblogs
```

*Agregar al archivo `.profile` las variables de ambiente*

Para que se pueda llevar a cabo la instalación de `informix`, se requiere dar de alta unas variables de ambiente en el archivo de inicialización del usuario `informix`, las cuales indican la trayectoria en donde se encuentra el directorio de `informix` y la ubicación de los archivos ejecutables, esto se hace de la siguiente manera:

Colocado en el directorio residente del usuario `informix` se edita el archivo `.profile`, con el editor de texto `vi` para UNIX.

TESIS CON  
FALLA DE ORIGEN

**\$ vi .profile**

```
# @(#) $Revision: 66.1.1.1 $
umask 027

# Default user .profile file (/bin/sh initialization).

# Set up the terminal:
if [ "$TERM" = "" ]
then
    eval ` tset -s -Q -m `:?hp` `
else
    eval ` tset -s -Q `
fi
stty erase "^H" kill "^U" intr "^C" eof "^D"
stty hupcl ixon ixoff
tabs

# Set up the search paths:
PATH=$PATH:.

# Set up the shell environment:
set -u
trap "echo 'logout'" 0

# Set up the shell variables:
stty erase "^?" kill "^U" intr "^C" eof "^D"
TERM=hp
export TERM
EDITOR=vi
export EDITOR
```

```
#####
echo " .....
echo " * .....
echo " *          BIENVENIDO AL USUARIO:  INFORMIX .....
echo " *          .....
echo " .....
pwd
```

Insertar en este punto las siguientes líneas:

```
INFORMIXDIR=/informix
PATH=$INFORMIXDIR/bin:$PATH:
INFORMIXSERVER=sep_shm
TERMCAP=$INFORMIXDIR/etc/termcap
export INFORMIXDIR PATH INFORMIXSERVER TERMCAP
```

TIENE CON  
FALLA DE ORIGEN

Lo anterior, describe las actividades que se requieren para que se pueda llevar a cabo la instalación de los productos de Informix Dynamic Server 2000 versión 9.20.

En la siguiente parte de este mismo tema se llevará a cabo la instalación y configuración del manejador, así como de los productos necesarios para llevar a cabo el funcionamiento de la aplicación SIAPSEP como son ESQL/C, SQR entre otros.

TESIS CON  
FALLA DE ORIGEN



### III.2 INFORMIX DINAMIC SERVER

El hablar de la configuración de un sistema manejador de BD requiere mucha decisión, para actividades como almacenamiento de datos, acceso a datos y la protección de los mismos. El cómo instalar y configurar el servidor de BD puede afectar en gran medida el rendimiento de operación de la BD, así como del sistema operativo

Este capítulo tiene como objetivos instalar el Servidor de Base de Datos utilizando como sistema aplicativo el SIAPSEP, apoyados en los conocimientos de administración del Sistema Operativo que requiere el Servidor de BD para su instalación, así como la configuración del mismo.

Previo a la instalación se tiene que hacer una recopilación de información respecto a los recursos con los que se cuenta para llevar a cabo la instalación, considerando las siguientes características:

- El nombre de la máquina (hostname)
- Dirección IP sobre la red.
- Información del servicio de red que soporta la plataforma sobre la que se va a instalar.
- Número de dispositivos y sus respectivos tamaños
- Cantidad de memoria para determinar el valor que va a ser asociado al Servidor de BD.

La figura 3.1 muestra de manera general un ejemplo de las características que presenta un equipo y las cuales se deben de considerar antes de instalar y configurar el servidor de Base de Datos.



Informix Dinamic Server y para el caso de la versión de Informix 2000 el producto se instala después del IDS 2000.

La tabla 3.1 muestra de acuerdo a la versión con que se cuente de Informix y del ESQ/C (Producto para programación), el orden en que se deben de instalar.

Database Server Versions	2.0a	2.1 and 2.2	2.3	2.4	2.5
5.x	Database server first	Database server first	Database server first	Database server first	Database server first
7.1a and 7.2a	Database server first	Database server first	Database server first	Database server first	Database server first
7.3a	Client products first	Database server first	Database server first	Database server first	Database server first
8.1a	Client products first	Database server first	Database server first	Database server first	Database server first
8.2a	Client products first	Client products first	Database server first	Database server first	Database server first
8.3	Client products first	Client products first	Database server first	Database server first	Database server first
9.1a	Client products first	Client products first	Database server first	Database server first	Database server first
9.2	Client products first	Client products first	Database server first	Database server first	Database server first

Tabla 3.1 Orden de Instalación de los productos de Informix

### Requerimientos de Instalación

Este tipo de software lo proporciona el proveedor y requiere de licencia para su uso por lo que antes de instalar el/los productos se debe contar con:

- NO. DE LICENCIA
- LLAVE DE ACTIVACIÓN

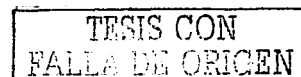
Otro requerimiento previo al sistema operativo, es el compilador de "C" el cual debe estar instalado, ya que es utilizado por el producto ESQ/C".

De acuerdo a la información anterior el proceso de instalación es el siguiente:

1. Instalación de INFORMIX DINAMIC SERVER (IDS 2000)
- 2.- Instalación ESQ/C
- 3.- Configuración de IDS 2000
- 4.- Inicialización del manejador de Base de Datos.

### Variables de Ambiente Requeridas

Las variables de ambiente son localidades o direcciones de memoria donde se almacena información sobre Unix que es el sistema operativo sobre el que estamos instalando



informix, existen variables con distintas funciones como por ejemplo la variable **PATH** guarda las direcciones o caminos (de directorios) donde buscar un programa ejecutable.

Para el producto IDS 2000 se tienen que dar de alta las siguientes variables de ambiente:

**INFORMIXDIR.**- Especifica el directorio en donde se encuentran instalados los productos de informix, como por Ej. **INFORMIXDIR=/informix**.

**PATH.**- Es la variable de ambiente que especifica la localización de los archivos ejecutables. Ej. **PATH=\$INFORMIXDIR/bin**

**INFORMIXSERVER.**- Especifica el nombre del servidor de BD default. Al cual se hace referencia en el archivo de configuración **ONCONFIG** en la etiqueta **DBSERVERNAME**.

**TERMCAP.**- Variables de ambiente que especifican el tipo de interfaz de terminal.

Dentro de las variables de ambiente existe una que especifica las librerías compartidas (shared – library) la cual, a comparación de las variables anteriores cambia dependiendo de la plataforma sobre la que se instale, el esquema siguiente nos muestra el nombre de las variables según la plataforma en que se encuentra.

Plataforma	Variable de Ambiente
AIX	LIBPATH
HP-UX	SHLIB_PATH
Solaris y otras Plataformas	LD_LIBRARY_PATH

### III.2.1 Instalación del Servidor de Base de Datos.

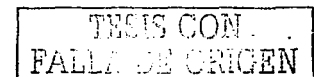
1 .- Forma de declarar las variables de ambiente para el sistema SIAPSEP.

```
$ INFORMIXDIR = /informix
$ TERMCAP = $INFORMIXDIR/etc/termcap
$ export TERMCAP
$ INFORMIXSERVER = sep_shm (Nombre del servidor)
$ export INFORMIXSERVER
```

2 .-Como siguiente paso se descargan los archivos de disco en donde apunta la variable de **INFORMIXDIR** en este caso la trayectoria **informix**

Existen dos formas de extraer los archivos dependiendo el formato en que se encuentre el software a instalar, la primera es con la utilería de unix **cpio**, la cual extrae los archivos en la ubicación donde se ejecutó el programa en este caso en el directorio asignado para **informix** como se muestra a continuación:

```
$ cd $INFORMIXDIR
$ cpio -icdumB < /dev/rmt/0m (Unidad de CD o DAT)
```



La otra forma de extraer los archivos es con el comando **tar** el cual extrae de y para un archivo tarfile, la forma de ejecutarlo es la siguiente:

```
$ cd $INFORMIXDIR
$ tar -xvf /dev/rmt/0m
```

Al término de la extracción en el directorio de informix deja una serie de archivos así como el archivo ejecutable de instalación de Informix Dinamic Server denominado **installserver**:

drwxrwx---	2	informix	informix	aaodir
drwxrwxr-x	2	informix	informix	bin
drwxr-xr-x	2	informix	informix	bitmaps
drwxrwx---	2	informix	informix	dbssodir
drwxr-xr-x	5	informix	informix	demo
drwxrwxr-x	2	informix	informix	etc
drwxr-xr-x	2	informix	informix	forms
drwxr-xr-x	5	informix	informix	gis
drwxr-xr-x	3	informix	informix	hhhelp
drwxr-xr-x	4	informix	informix	incl
drwxr-xr-x	2	informix	informix	ism
drwxr-xr-x	3	informix	informix	lib
drwxr-xr-x	3	informix	informix	msg
drwxr-xr-x	3	informix	informix	release
drwxr-xr-x	4	informix	informix	snmp
rwxf-xr-x	1	informix	informix	<b>installserver</b>

El siguiente paso es proceder a la instalación de IDS 2000 ejecutando el programa de instalación (**install**) que tiene el siguiente nombre:

**Installserver**

En unix la forma de instalar algún producto es anteponiendo al archivo ejecutable **./** como se muestra a continuación:

```
$ ./installserver
```

Este tipo de productos viene acompañado por licenciamiento por lo que es solicitado al momento de la instalación con los dos requerimientos siguientes:

No. de Licencia  
Llave de Activación

Estos dos datos finales los proporciona la empresa que vendió el software.

Finalmente la instalación genera un archivo el cual va a permitir la liga entre informix y el sistema operativo que de igual forma se tiene que ejecutar y su nombre es el siguiente:

*RUN\_AS\_ROOT.server*

La ejecución es de la siguiente forma:

*./RUN\_AS\_ROOT.server*

En versiones inferiores a la 9.x (versión 2000) la instalación se tenía que hacer con el usuario root (usuario administrador de informix) por lo que al momento de instalar ligaba librerías, a partir de la versión 2000 el usuario que instala el producto es el usuario de informix por lo que para realizar la liga de librerías con el sistema operativo tiene que ser mediante la ejecución del archivo RUN\_AS\_ROOT.server como se mencionó en el párrafo anterior.

Termina la Instalación de Informix Dinamic Server

El sistema SIAPSEP requiere de una herramienta de programación para interactuar con la Base de Datos y realizar la toma de decisiones. Informix proporciona un producto denominado ESQL/C que es la combinación del lenguaje de programación "C" y el lenguaje estructurado de consultas (SQL). Este producto en versiones anteriores a la 9.x venía como un producto aparte ahora viene acompañado de otros productos con el nombre de ClientSDK, el procedimiento de instalación es el siguiente:

### III.2.2 Instalación del ESQL/C

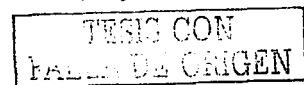
Para la descarga a disco de los archivos de instalación se debe estar con el usuario informix (administrador del servidor de base de datos).

- 1.- Cargar el CD ó Dat que contiene los productos del cliente en la unidad respectiva.
- 2.- De igual forma se prepara el ambiente (declaración de variables de ambiente)  
Con la cuenta **root** se definen las siguientes variables de ambiente:  
\$ INFORMIXDIR = /informix (Ubicación donde está instalado informix)  
\$ export INFORMIXDIR = /informix (Se da de alta )  
\$ export PATH = \$INFORMIXDIR/bin:\$PATH

La siguiente instrucción debe aplicarse dentro del directorio de informix y con el usuario de informix como se mencionó anteriormente.

- 3.- Descarga la información de la cinta ó dat, que de igual forma que el proceso de instalación hay dos maneras de realizarlo, con el comando cpio y el tar.

\$ cpio -icudunB < /dev/rm/0m



ó

```
$ tar -xvf /dev/rmt/0m
```

4.- Después de la extracción de los archivos se procede a instalar el ESQI.

Al terminar de bajar el CD ó dat deja un archivo con el sig. nombre **installesq**

El cual se ejecutará de la siguiente manera:

```
$ ./installesq
```

El programa pide:

```
No. de Serie          < copiarlo de la Tarjeta de Licencia >  
Llave de Activación  < copiarlo de la Tarjeta de Licencia >
```

Para poder realizar las ligas con las librerías de informix y del Sistema Operativo se genera un archivo con el siguiente nombre **RUN\_AS\_ROOT.clientsdk**

```
./RUN_AS_ROOT.clientsdk
```

5.- Instalación del ESQI Finalizada.

Posterior a la instalación de los recursos requeridos sigue la parte de configuración de conectividad que nos va a permitir acceder a los datos, como a continuación se menciona.

### III.2.3 Definición del Tipo de Comunicación y Servicio entre el Servidor y el Manejador de Base de Datos.

Para lograr establecer la conexión entre el cliente y el servidor que más adelante se mencionará se requiere saber el nombre del servidor de Base de Datos, el tipo de conexión que utiliza el cliente para comunicarse, el nombre de la máquina (hostname) o nodo sobre el cual está funcionando el servidor de BD y el nombre del servicio.

La tabla 3.2 muestra las conexiones que informix soporta para las comunicaciones en estructuras cliente/servidor en donde el identificador nettype es una simplificación del producto de informix con las dos primeras posiciones on (online), las tres posiciones siguientes corresponden al protocolo (ipc) y finalmente las tres posiciones siguientes corresponden al tipo de conexión que se va a utilizar (shm memoria compartida) como se muestra a continuación:

TESIS CON  
FALLA DE ORIGEN

nettype	Description	Connection Type
onipcshm	The database server using shared-memory communication	IPC
onipcstr	The database server using stream-pipe communication	IPC ♦
onipcnmp	The database server using named-pipe communication	IPC ♦
ontlitcp	The database server using TLI with TCP/IP protocol	Network ♦
onsoctcp	The database server using sockets with TCP/IP protocol	Network
ontlisp	The database server using TLI with IPX/SPX protocol	Network ♦

Tabla 3.2 Tipos de conexiones cliente/servidor soportadas por informix

Informix está conformado para trabajar en un modelo llamado cliente/servidor el cual permite, que una aplicación de un cliente se encuentre en una computadora y el servidor de base de datos en otra (esto no impide que se puedan encontrar sobre la misma máquina), esto mediante un protocolo de red y una interfaz de programación de red para conectarse y hacer la transferencia de datos entre el cliente y el servidor de BD.

Los protocolos de red son un conjunto de reglas que gobiernan la transferencia de los datos entre una aplicación, en nuestro caso entre un cliente y un servidor de base de datos. Esta información, específica, entre otras cosas que formato toman los datos cuando son enviados por la red. Un ejemplo de un protocolo de red y muy comúnmente utilizado es el TCP/IP, que contiene el código del formato en que se envía la información del cliente al servidor y viceversa, denominado driver protocolo red.

El cliente y el servidor de BD accesan al driver de red por medio de un camino denominado programa de interface de red, el cual contiene llamadas al sistema o librerías de rutinas que proporcionan comunicación de red.

La información de conectividad permite a las aplicaciones de clientes conectarse a cualquier servidor de BD Informix sobre la red. Los datos de conectividad para un servidor de Base de Datos en particular incluyen lo siguiente:

- Nombre del servidor de BD.
- Tipo de conexión que los clientes soportan.
- Nombre hosts de la computadora o nodo sobre el cual corre el servidor de BD.
- Nombre del servicio

TESIS CON  
FALLA DE ORIGEN



La tabla 3.3 nos muestra las conexiones soportadas en las diferentes plataformas:

Tipo de Conexión	Windows NT	UNIX	Local
Sockets	Sí	Sí	Sí
TLI (TCP/IP)	Sí	Sí	Sí
TLI (IPX/SPX)	Sí	Sí	Sí
Shared memory	Sí	Sí	
Stream pipe	Sí	Sí	
Named pipe	Sí		

Tabla 3.3 Conexiones soportadas en diferentes plataformas por Informix

### Los Sockets

Los *sockets* no son más que puntos o mecanismos de comunicación entre procesos, que permiten que un proceso hable (emita o reciba información) con otro proceso incluso estando estos procesos en distintas máquinas. Esta característica de interconectividad entre máquinas hace que el concepto de *socket* nos sea de gran utilidad. Esta interfaz de comunicación es una de las distribuciones de Berkeley al sistema UNIX, implementándose las utilidades de interconectividad de este Sistema Operativo ( *rlogin, telnet, ftp, ...* ) usando *sockets*.

La comunicación entre procesos a través de *sockets* se basa en la filosofía **CLIENTE-SERVIDOR**: un proceso en esta comunicación actuará de **proceso servidor** creando un *socket* cuyo nombre conocerá el **proceso cliente**, el cual podrá "hablar" con el proceso servidor a través de la conexión con dicho **socket nombrado**. El proceso crea un *socket* sin nombre cuyo valor de vuelta es un descriptor sobre el que se leerá o escribirá, permitiéndose una comunicación **bidireccional**, característica propia de los *sockets* y que los diferencia de los **pipes**, o canales de comunicación unidireccional entre procesos de una misma máquina.

### Tipo de Comunicación TLI

Es una interfaz de comunicación a través de capas, esta interfaz general permite la comunicación entre aplicaciones, accedendo a la vía de transporte. Las conexiones pueden ser realizadas desde otra aplicación ubicada en otro equipo, colocando el protocolo de transporte, el cual se pone en ejecución a través de rutinas y librerías arriba del mecanismo Stream que es el que lleva a cabo las secuencias de transporte.

### El Shared Memory

Cuando el cliente y el Servidor se encuentran en el mismo equipo, se comunican mediante una estructura llamada Shared Memory, utilizando el protocolo **onipcshm**.

TESIS CON  
FALLA DE ORIGEN

Una conexión shared memory usa un área de la memoria compartida como canal a través del cuál el cliente y el servidor se comunican entre ellos como lo muestra la figura 3.2:

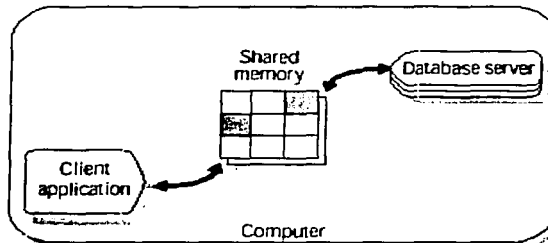


Figura 3.2 Conexión mediante la memoria compartida

En este ejemplo vemos como el proceso cliente intercambia información con el Servidor de BD, utilizando esta comunicación.

Este tipo de conexión proporciona acceso rápido al servidor, pero no es muy seguro y es vulnerable en cuanto a errores de código de programación así como a modificaciones que no se ven realizadas en el servidor de BD. El cliente puede tener solamente una conexión de este tipo.

#### *Conexión Stream-Pipe*

Es una comunicación interprocesos (IPC) en UNIX que facilita y permite que varios procesos sobre la misma computadora se comuniquen entre sí. El uso del Stream Pipe se puede realizar en el momento en el que el cliente y el servidor de Base de Datos se encuentren en el mismo equipo. Las ventajas que proporciona este tipo de conexión son que no ponen en riesgo la seguridad de los datos como lo hace el tipo de conexión shared-memory, permiten el uso de transacciones distribuidas entre servidores de Base de datos que están en la misma máquina. Por otra parte presenta las siguientes desventajas que son: más lentitud en la conexión comparándolos con los otros tipos de conexiones, así como que no están disponibles en todas las plataformas.

### *Conexión Named-Pipe*

Son aplicaciones de interfaces de programación (API's) para comunicación de interprocesos de manera bidireccional (IPC) sobre Windows NT. Permite un alto rendimiento para conectar redes de computadoras usando el transporte en capas, la información la almacenan en memoria para cuando se haga una solicitud sea rápida la respuesta. Las conexiones son soportadas para conexiones locales en el servidor de Base de Datos.

Actualmente el tipo de conexión que está utilizando la aplicación SIAPSEP es TLI (TCP/IP) para poder hacer uso de este tipo de comunicación en informix se tiene que configurar y dar de alta en algunos archivos la indicación de este tipo de comunicación que a continuación se describe.

El punto siguiente es definir que tipo de conexión se va a utilizar de acuerdo a la ubicación del cliente con respecto al servidor de BD, para lo cual existen los archivos de conectividad que contienen información que habilita la comunicación cliente/servidor, los cuales son divididos en 3 grupos:

- Archivos de configuración de red.
- Archivos de seguridad de red.
- El archivo **SQLHOSTS**

### *Archivo SQLHOSTS*

Este archivo contiene información para habilitar que una aplicación cliente encuentre y se conecte a cualquier servidor de base de datos Informix sobre la red. Este archivo es generado cuando se hace la instalación del servidor de BD (IDS2000), colocando el archivo en el directorio */etc*.

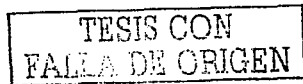
El protocolo que vamos a utilizar es TCP/IP para la aplicación del SIAPSEP, y en base a esto se configura el archivo SQLHOSTS.

En caso de cambiar la trayectoria para hacer referencia a este archivo se tiene que dar de alta la variable de ambiente **INFORMIXSQLHOSTS**.

La información requerida para este archivo de configuración es la siguiente:

- Dbservername
- Nettype
- Hostname
- Servicename
- Options

Nombre del Servidor de Base de Datos (dbservername)



Es el nombre del servidor de Base de Datos que en el archivo de configuración va a estar definido por DBSERVERNAME y DBSERVERALIASES.

*Campo del tipo de conexión (Nettype)*

Describe el tipo de conexión que puede haber entre un servidor de BD y un cliente, el cual se subdivide en tres secciones: Productos del servidor de BD, tipo de interfaz y protocolo de red como se muestra en la figura 3.3:



Figura 3.3 División del campo Nettype

Referente a la figura anterior tenemos lo siguiente:

Las dos primeras letras del tipo de conexión representan el producto de informix que se está instalando **on** ó **ol** que representa Online.

Las tres siguientes letras indican el tipo de interfaz que se utilizará.

Las tres ultimas letras representan el mecanismo de comunicación o protocolo de red.

El tipo de interfaz representa el programa de interfaz de red que son los siguientes:

ipc	IPC (Comunicación Interprocesos)
soc	Sockets
tli	TLI ( Interface de transportación en capas)

En la subdivisión del nettype referente a protocolos de red se manejan 5 que son los siguientes:

shm	Comunicación mediante la memoria compartida
spx	Protocolo de red IPX/SPX
str	Comunicación Stream-pipe
nmp	Comunicación Name -pipe
tcp	Protocolo de red TCP/IP

TESIS CON  
FALLA DE ORIGEN

El siguiente esquema muestra las opciones completas del identificador nettype y sobre las plataformas que soporta ya sea de sistemas operativos unix ó sistemas operativos con Windows NT:

nettype value (UNIX)	PROTOCOL value (Windows NT)	Description	Connection Type
unipcshm		Shared-memory communication	IPC
unipcstr		Stream-pipe communication	IPC
	unipcstrp	Named-pipe communication	IPC
uniltcp		TLI with TCP/IP protocol	Network
unioctcp	unioctcp	Socket with TCP/IP protocol	Network
uniltsp		TLI with SPX/SPP protocol	Network

El tercer campo que tiene que ser definido en el archivo SQLHOST es el hostname, que contiene el nombre de la computadora donde reside el servidor de base de datos, el dato que será asignado se encuentra en el archivo **hosts** el cual proporciona la dirección de red de la computadora.

Finalmente sigue el campo del nombre del servicio, que se va a definir de acuerdo al tipo de conexión que se va a utilizar. Este servicio debe de darse de alta previamente en el archivo services de UNIX, en donde está definido el nombre del servicio, puerto/protocolo y finalmente su descripción (opcional).

A continuación observamos una edición del archivo sqlhosts en donde podemos visualizar los datos utilizados por el sistema SIAPSEP.

### \$ vi sqlhosts

```

#.....
#          INFORMIX SOFTWARE, INC.
#
#          PROPRIETARY DATA
#
#THIS DOCUMENT CONTAINS TRADE SECRET DATA WHICH IS THE PROPERTY OF
#INFORMIX SOFTWARE, INC. THIS DOCUMENT IS SUBMITTED TO RECIPIENT IN
#CONFIDENCE. INFORMATION CONTAINED HEREIN MAY NOT BE USED, COPIED
OR
#DISCLOSED IN WHOLE OR IN PART EXCEPT AS PERMITTED BY WRITTEN
#AGREEMENT
#SIGNED BY AN OFFICER OF INFORMIX SOFTWARE, INC.

```

```
#
#THIS MATERIAL IS ALSO COPYRIGHTED AS AN UNPUBLISHED WORK UNDER
#SECTIONS 104 AND 408 OF TITLE 17 OF THE UNITED STATES CODE.
#UNAUTHORIZED USE, COPYING OR OTHER REPRODUCTION IS PROHIBITED BY
#LAW.
#
#
demo_on      onipcshm      on_hostname on_servername
demo_se      seippcip      se_hostname sqlxec
#-----
```

Estas líneas, son un ejemplo que trae por default el servidor, por lo cual deben de comentarse o borrarse para poder añadir las líneas que requiere la configuración de determinado sistema

**Las siguientes líneas son insertadas para el sistema SIAPSEP en el archivo sqlhosts antes mencionado**

#SERVIDOR	TIPO CONEXION	HOST	SERVICIO
sep_shm	onipcshm	cpar02	ninguno
sep_tcp	onsoctcp	cpar02	sqlxecdsa

La primera línea es para la conexión mediante la memoria compartida y la segunda mediante el protocolo tcp/ip.

EL hosts es el nombre del equipo en donde se encuentra instalado el servidor de base de datos Informix.

### III.2.4 Alta del Servicio en el Archivo Services.

Cuando se utiliza el protocolo de conexión TCP/IP el nombre del servicio debe de corresponder con el que se encuentra en el archivo services, en él se encuentra el número de puerto asociado al servidor de base de datos en el hosts correspondiente. Este número de puerto especificado en el archivo services le permite a los programas de red encontrar el servidor de Base de Datos, que posteriormente se va a definir.

En el siguiente esquema se encuentra la relación entre el archivo sqlhosts de Informix, el archivo de servicios y el archivo hosts del Sistema Operativo.

Como el archivo services es un archivo de configuración del Sistema Operativo el usuario que puede hacer modificaciones es el administrador. La ubicación de este archivo es bajo el directorio /etc. El archivo que a continuación se edita es el que actualmente está utilizando el Sistema Integral de Administración de la SEP.

La forma de editar un archivo utilizando el sistema operativo unix es el siguiente:

\$ vi services

Buscar que NO exista el puerto **1525** ni **sqlxecdsa** que son los nombres que se van a designar.

Insertar la siguiente línea, la cual puede ir al inicio o al final del archivo, de igual forma el número de puerto no necesariamente debe ser el 1525, pero es el estándar con la finalidad de tener bien ubicado el puerto por donde se está conectando el cliente y el servido de BD.

```
sqlxecdsa    1525/tcp    # servidor sep_tcp de informix
```

A continuación se ejemplifica como quedaría el archivo services

**\$ vi services.**

```
# @(#)services $Revision: 1.26.111.4 $ $Date: 95/02/17 16:32:46 $
#
# This file associates official service names and aliases with
# the port number and protocol the services use.
#
# Some of the services represented below are not supported on HP-UX.
# They are provided solely as a reference.
#
# The form for each entry is:
# <official service name> <port number/protocol name> <aliases>
#
# See the services(4) manual page for more information.
# Note: The entries cannot be preceded by a blank space.
#
tcpmux      1/tcp          # TCP port multiplexer (RFC 1078)
echo        7/tcp          # Echo
#
# UNIX services
#
biff        512/udp    comsat      # mail notification
exec        512/tcp
login       513/tcp
#
lansrm      570/udp
DAServer    987/tcp
nfsd-keepalive 1110/udp
nfsd-status 1110/tcp
rib         1260/tcp
clvm-cfg    1476/tcp
diagmond    1508/tcp
#
# SRM/UX Server
# SQL distributed access
# Client status info
# Cluster status info
# remote loopback diagnostic
# HA LVM configuration
# Diagnostic System Manager
```

TESIS CON  
FALLA DE ORIGEN

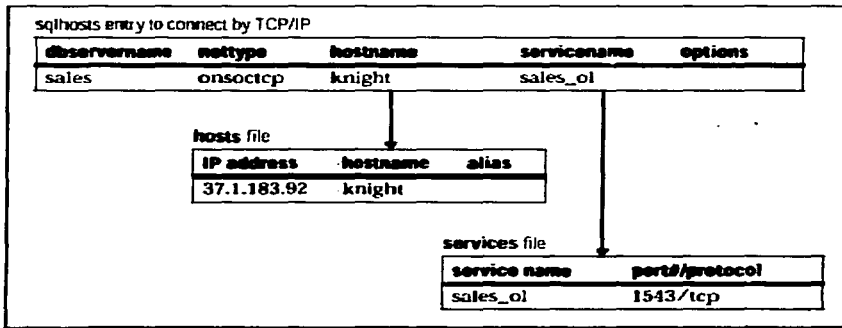
*Registro Insertado*

```

.....
sqlxecdsa 1525/tcp # Servidor sep_tcp de informix
.....

nft          1536/tcp          # NS network file transfer
pmlockd     1889/tcp          # SynerVision locking daemon
pmlockd     1889/udp          #
nfsd        2049/udp          # NFS remote file system
netdist     2106/tcp          # update(1m) network distribution service
rfa         4672/tcp          # NS remote file access
veesm       4789/tcp          # HP VEE service manager
hacl-hb     5300/tcp          # High Availability (HA) Cluster heartbeat
hacl-probe  5303/tcp          # HA Cluster TCP probe
    
```

El siguiente esquema ilustra los archivos de configuración antes mencionados y como son referenciados para configurar el archivo de informix:



Archivos de Comunicación utilizados por Informix (Sqlhosts, hosts y services)

TESIS CON  
FALLA DE ORIGEN





### III.2.5 Ambientación del Usuario Informix

El usuario Informix o cualquier otro usuario que requiera realizar una conexión a informix necesita ambientarse dando de alta todas las variables de ambiente mencionadas al inicio de la instalación.

Esto se realiza en el archivo profile de cada usuario, a continuación se muestra un ejemplo de la forma en que se ambienta el usuario informix.

#### \$ vi .profile

```
# @(#) $Revision: 66.1.1.1 $
umask 027

# Default user .profile file (/bin/sh initialization).

# Set up the terminal:
if [ "$TERM" = "" ]
then
    eval `tset -s -Q -m `:?hp` `
else
    eval `tset -s -Q `
fi
stty erase "^H" kill "^U" intr "^C" eof "^D"
stty hupcl ixon ixoff
tabs

# Set up the search paths:
PATH=$PATH:..
# Set up the shell environment:
set -u
trap "echo 'logout'" 0

# Set up the shell variables:
stty erase "^?" kill "^U" intr "^C" eof "^D"
TERM=hp
export TERM
EDITOR=vi
export EDITOR
```

TESIS CON  
FALLA DE ORIGEN

```
#####
echo "
echo "
echo "          BIENVENIDO AL USUARIO:  INFORMIX
echo "
echo "
pwd
```

Las siguientes líneas son las que se agregan al profile del usuario para indicarle a que servidor de BD conectarse. Este es un ejemplo:

```

.....
INFORMIXDIR=/informix
PATH=$INFORMIXDIR/bin:$PATH:
INFORMIXSERVER=sep_shm
TERMCAP=$INFORMIXDIR/etc/termcap
export INFORMIXDIR PATH INFORMIXSERVER TERMCAP
.....

```

```

# Edited by Uniface (poly) Install Script on Thu Jan 23 16:39:14 CST 1997
# polyserver with drivers  CHR INF SEQ TXT  TCP
. /psys/inspoly
psv="$PSV TCP:";export psv
PSV="$PSV TCP:";export PSV
# INFORMIXDIR=/informix;export INFORMIXDIR
. /psys/insunis
# INFORMIXDIR=/informix;export INFORMIXDIR

```

#### *Cambiar de Propietario las Areas que van a alojar los dbspaces*

Un dbspace es una unidad de almacenamiento lógica, la cual proporciona el enlace crítico entre la unidad de almacenamiento lógico y físico, esto permite que se asocien unidades físicas (tales como chunks) con unidades lógicas (como son las tablas).

Con la cuenta de **root**

```

$ chown informix:informix /dev/vg00/rdbtemp
$ chown informix:informix /dev/vg00/rdbdats
$ chown informix:informix /dev/vg00/rdbroot
$ chown informix:informix /dev/vg00/rdblogs

```

#### **CREAR LAS LIGAS CON In -s**

*Se cambia a las ligas los permisos de lectura y escritura como sigue:*

Posicionarse en el directorio \$INFORMIXDIR.

```

$ chmod a+rw lrdbtemp
$ chmod a +rw lrdbdats
$ chmod a +rw lrdbroots
$ chmod a +rw lrdblogs

```



### III.2.6 Modificación de los Parámetros de Informix:

- DISK PARAMETERS
- SHARED MEMORY PARAMETERS
- PERFORMANCE TUNING PARAMETERS
- DATA REPLICATION PARAMETTERS
- DIAGNOSTIC PARAMETERS
- PARALELL DATABASE QUERIE PARAMETERS

Estos parámetros están en función de:

El nombre que se le hayan dado a los dbspaces.  
El tamaño en kbytes de los dbspaces.  
Las trayectorias de archivos  
El No. de archivos lógicos  
Si los dbspaces son o no espejeados.

A continuación se muestran las pantallas de la configuración de la instalación realizada en el CPAR (Centro de Procesamiento de Atención Remota) en una maquina SUN Enterprise 10000.

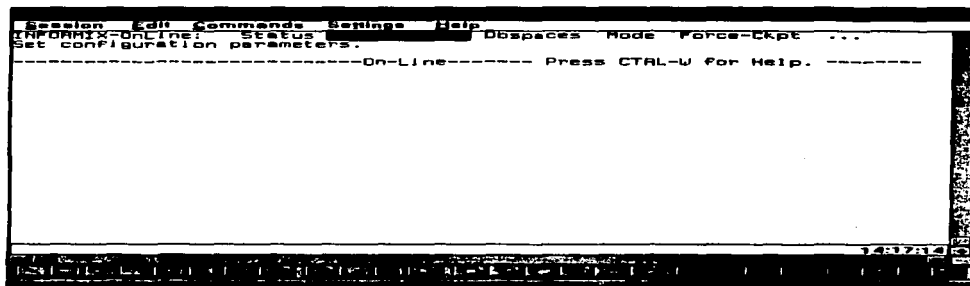
Los campos que se modificarán son los que están marcados. Posteriormente al hacer el Tunning del manejador, se modifican los restantes.

Existen dos formas de realizar una afectación en la configuración del servidor de Base de Datos, una es editando el archivo de configuración con el editor de textos y modificando los parámetros que se requieren o con la utilería onmonitor, la utilizada a continuación:

Para lo anterior debe ejecutarse el programa **onmonitor**, con la cuenta **informix** desde cualquier directorio ó en caso de que no esté definida la trayectoria de los ejecutables trasladarse al subdirectorío /informix/bin.

A continuación se presentan los pasos a seguir para poder configurar el servidor de base de datos informix:

1.- **\$ onmonitor** y aparece la siguiente pantalla.



Esta pantalla es con la que inicia la utilería onmonitor, en la parte superior tiene un menú el cual tiene todas las opciones de configuración que se presentarán a continuación estos cambios que se vana realizar mediante esta vía también pueden ser modificados directamente en el archivo de configuración de informix ONCONFIG.

Aparecen las siguientes opciones:

- Status
- Parameters
- Dbspaces
- Mode
- Force-Ckpt
- Logical-Logs
- Exit

TESIS CON  
FALLA DE ORIGEN

A continuación se describe cada uno de los menús pero de manera detallada en la parte final lo referente a parámetros:

**Menú de Estatus**

Submenú	Descripción
Profile	Despliega las estadísticas del rendimiento del servidor de Base de datos.
Userthreads	Despliega el estatus de actividad de los hilos de usuario.
Spaces	Despliega información acerca de espacios de almacenamiento y chunks del servidor de BD.
Databases	Despliega el nombre, propietarios y modo de logeo de las 100 primeras BD.
Logs	Despliega la información de los logs físicos

	y lógicos.
Archive	Despliega todas las cintas de respaldo de los archivos de logs lógicos que para su restauración es mediante la utilería ontape.
Data-Replication	Despliega el estatus de replicación de datos en Alta Disponibilidad y como está configurado.
Output	Almacena la salida de información de otros estatus en un archivo especificado.
Configuration	Copia la configuración actual del Servidor de BD a un archivo.

### Menú de Parametros

Submenú	Descripción
Initialize	Inicializa el espacio en disco del Servidor de BD ó modifica parámetros de espacio en disco.
Shared-Memory	Inicializa la memoria compartida del Servidor de BD ó modifica los parámetros de memoria compartida.
Performance	Especifica el número de procesadores virtuales por cada clase de procesador virtual.
Data - Replication	Especifica los parámetros de Replicación de alta disponibilidad.
Diagnostics	Especifica el valor de los parámetros de diagnósticos.
PdQ	Cambia los parámetros para la consulta de datos en paralelo.
Add-log	Añade un archivo de log lógicos a un dbspace.
Drop Log	Borra los archivos de logs lógicos de un dbspaces determinado.
Physical Log	Cambia el tamaño o la localización de los logs físicos del Servidor de BD.

### Menú de Dbspaces

Submenú	Descripción
Create	Creación de un dbspaces
BLOBSpace	Creación de un blobspace
Mirror	Añade espejos a un espacio de almacenamiento existente o finaliza un espejo del mismo.
Drop	Borra un espacio de almacenamiento.
Info	Despliega el número de identificación, localización de cada chunk asignado a un espacio de almacenamiento

Add-chunk	Añade un chunk a un espacio de almacenamiento.
Dataskip	Cambia el parámetro database.
Status	Cambia el estatus de un chunk a espejo.

Menú Mode

Submenú	Descripción
Startup	Inicializa la memoria compartida y pone al servidor en modo monousuario.
On-Line	Cambia el servidor de Bd de datos de modo monousuario a en-línea.
Graceful-Shutdown	Cambia el servidor de Online a Quiscent hasta que termine la última actividad de usuario.
Immediate Shutdown	Cambia el servidor de online a Quiscent en un lapso de 10 segundos.
Take - Offline	Desasigna la memoria compartida y da de baja el servidor de informix.
Add - Proc	Añade procesadores virtuales
Drop-Proc	Borra procesadores virtuales.
Decision-Support	Asigna dinámicamente los parámetros de soporte decisión.

Menú Force-Ckpt

Submenú	Descripción
Force-Ckpt	Despliega el tiempo del más reciente checkpoint realizado por el servidor de BD.

Menú Archive

Submenú	Descripción
Tape-Parameters	Modifica los parámetros del ontape para el respaldo en cinta.

Menú Logs - Lógicos

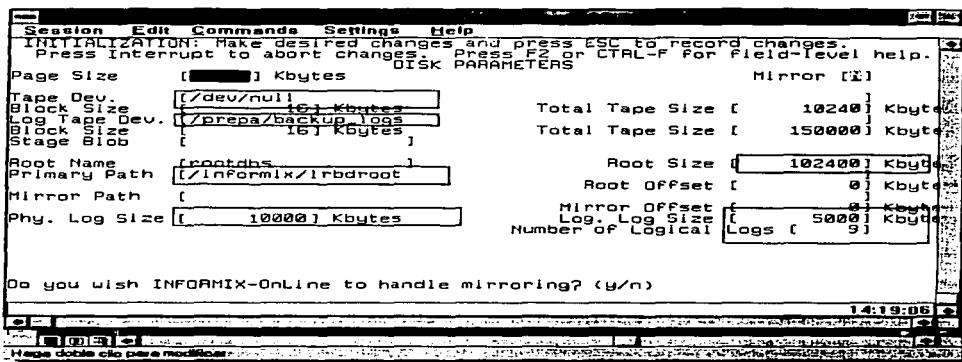
Submenú	Descripción
Databases	Modifica el estatus de logeo de una base de datos.
Tape-Parameters	Modifica los parámetros de ontape del dispositivo de respaldo.

TESIS CON  
FALLA DE ORIGEN

A continuación se realiza el proceso de configuración del Servidor de Base de Datos Informix para que pueda cargarse la Base de datos del SIAPSEP.

2.- Después de presentar la primer pantalla del onmonitor se selecciona la opción de Parameters e Initialize presentando la siguiente pantalla:

### III.2.6.1 Disco



En esta pantalla básicamente se definen los parámetros de Disco y son los siguientes:

Page Size .- Es el tamaño de la página del sistema y depende de la plataforma sobre la que se vaya a trabajar (IBM,HP, SUN ó NT). En este servidor HP el tamaño de las paginas de datos es de 2Kbytes.

Mirror (MIRROR).- Es la bandera que indica si es habilitado un espejo para el servidor de base de datos. Habilitar el espejo si el esquema es crear un espejo para el dbspace de root como parte de la inicialización, en nuestro ejemplo no configura un espejo por lo que se coloca la letra N en caso de si habilitarlo se le coloca la letra Y.

Tape Dev (TAPEDEV).- Especifica el dispositivo sobre el cual se va a realizar el respaldo de los datos (dbspace), de igual forma especifica el dispositivo default para la carga o descarga de datos con las opciones de onload y onunload. Puede ser representado mediante ligas simbólicas o utilizando el dispositivo desde otro equipo.

TESIS CON  
FALLA DE ORIGEN

- Block Size (TAPEBLK).**- Especifica el tamaño de un bloque cuando se ejecuta la utilería de ontape al momento de escribir cuando se realiza el respaldo de dato, de la misma manera cuando se realiza la carga o descarga de datos con las utilerías de onload y onunload.
- Total Tape Size (TAPESIZE).**- Este parámetro especifica el tamaño del dispositivo cuando se utiliza ontape para el respaldo de datos.
- Log Tape Dev (LTAPEDEV).**- Especifica el dispositivo en el cual se van a realizar los respaldos de los logs lógicos con la utilería ontape.
- Block Size ( LTAPEBLK).**- Especifica el tamaño de un bloque de información cuando se realiza el respaldo de los logs lógicos.
- Total Tape Size(LTAPESIZE).**- Especifica el tamaño máximo de la cinta donde se va a llevar a cabo el respaldo de los logs lógicos con la utilería ontape.
- Stage Blob (STAGEBLOB).**- Este parámetro se utiliza únicamente cuando se manejan tipos de datos TEXT y BYTE enviándolos a un tipo de almacenamiento óptico.
- Root Name (ROOTNAME).**- Especifica el nombre del dbspace raíz del servidor de base de datos, el máximo número de caracteres es de 18 y no puede contener números.
- Root Size.(ROOTSIZE).**- Especifica el tamaño del chunk inicial del dbspace de root, expresado en kilobytes.
- Primary Path (ROOTPATH).**- Especifica la ruta completa incluyendo el dispositivo o el nombre del archivo del chunk inicial del dbspace de root.
- Root Offset (ROOTOFFSET).**- Especifica la compensación de espacio en disco (archivo, partición de disco ó dispositivo) en el inicio del dbspace de root.
- Mirror Path (MIRRORPATH).**- Especifica la ruta completa del chunk que el servidor de BD va a utilizar como espejo para el dbspace inicial (root).
- Mirror Offset (MIRROROFFSET).**- Especifica la compensación dentro de una partición de disco ó dentro de un dispositivo.
- Phy. Log Size (PHYSFILE).**- Especifica el tamaño del log físico.
- Log. Log Size (LOGSIZE).**- Especifica el tamaño usado cuando los archivos de logs lógicos son creados.
- Number of logical logs (LOGFILE).**- Especifica el número de archivos de logs lógicos cuando es inicializado el servidor de BD.

TESIS CON  
FALLA DE ORIGEN



### III.2.6.2 Memoria Compartida (Shared Memory)

```

Session Edit Commands Settings Help
SHARED MEMORY: Make desired changes and press ESC to record changes.
Press Interrupt to abort changes. Press F2 or CTRL-F for field-level help.
SHARED MEMORY PARAMETERS
Server Number [ 1 ] Server Name [ lsep_shm ]
Server Aliases [ lsep_tcp ]
Dbspace Temp [ tempodbs ]
Deadlock Timeout [ 60 ] Secs Number of Page Cleaners [ 1 ]
Forced Residency [ M ] Stack Size (K) [ 32 ]
Non Res. SegSize (K) [ 16000 ] Optical Cache Size (K) [ 0 ]
Physical Log Buffer Size [ 32 ] K Dbspace Down Option [ 0 ]
Logical Log Buffer Size [ 32 ] K Preserve Log For Log Backup [ N ]
Max # of Logical Logs [ 20 ] Transaction Timeout [ 300 ]
Max # of Locks [ 100000 ] Long TX HWM [ 50 ]
Max # of Buffers [ 50000 ] Long TX HWM Exclusive [ 60 ]
Index Page Fill Factor [ 90 ]
Add SegSize (K) [ 8192 ]
Total Memory (K) [ 0 ]

Resident Shared Memory size [ 16000 ] Kbytes Page Size [ 8192 ] Kbytes

Enter a unique value to be associated with this version of INFORMIX-OnLine.

14:19:51
    
```

En esta pantalla se configura lo referente a la memoria compartida, los parámetros son los siguientes:

**Server Number (SERVERNUM).**- Especifica una localización relativa en la memoria compartida, no es importante el número pero debe estar en el rango de 0-255 y para cada instancia de servidor de BD debe ser único.

**Server Name (DBSERVERNAME).**- Especifica un nombre único asociado con un servidor de BD. La aplicación del cliente utiliza el dbservername en la variable de ambiente INFORMIXSERVER para que el cliente pueda establecer conexión con el servidor de BD correspondiente.

**Server Aliases (DBSERVERALIAS).**- Especifica una lista de alternativas del DBSERVERNAME. Si el servidor de BD soporta más de un protocolo de comunicación, se puede especificar un alias al servidor para poder asociar en el archivo de comunicaciones SQLHOST un alias para poder utilizar cualquier otro tipo de protocolo.

**Dbspace Temp (DBSPACETEMP).**- Especifica lista de dbspace que el servidor de BD maneja para almacenamiento global de tablas temporales.

**Deadlock Timeout (DEADLOCK\_TIMEOUT).**- Especifica EL MÁXIMO número de segundos que un proceso del servidor de BD puede esperar para ser

TESIS CON  
FALLA DE ORIGEN

bloqueado. Este parámetro se usa únicamente para consultas distribuidas en servidores remotos, cuando no es distribuido no lo utiliza.

**Secs Number of Page Cleaners (CLEANERS).**- Especifica el número de páginas limpias por proceso.

**Forced Residency (RESIDENT).**- Este parámetro especifica los segmentos residentes y los virtuales de memoria compartida residente en la memoria física del sistema operativo, por lo que dependerá del sistema si soporta o no la residencia forzada.

**Stack Size (STACKSIZE).**- Especifica el tamaño de pila que puede utilizar un proceso de usuario en el servidor de BD, este parámetro no tiene un límite pero repercute directamente en la memoria compartida.

**Non Res. SegSize (SHMVIRTSIZE).**- Especifica el tamaño inicial de un segmento virtual en la memoria compartida.

**Optical Cache Size (OPCACHEMAX)** .- Especifica el tamaño de la memoria cache para el subsistema óptico. El servidor de BD almacena piezas de datos TEXT y BYTE en la memoria cache.

**Dbospace Down Option (ONDBSPACESDOWN).**- Define la acción que el servidor de BD puede tomar, cuando cualquier evento ocurre deshabilitando un dbospace no crítico.

**Preserve Log For Log Backup(LBU\_PRESERVE).**- Reserva el último log lógico para tareas administrativas, para definir la marca más alta para determinar que el log ya está lleno.

**Heterogeneous Commit (HETERO\_COMMIT).**- Parámetro de configuración que especifica o no si el servidor de BD está preparado para participar con productos de informix para puerta de enlace en la terminación de distintas transacciones.

**Physical Log Buffer Size.(PHYSBUFF).**- Especifica el tamaño en kilobytes de cada uno de los logs físicos en la memoria compartida. El mínimo valor que puede permitir es de 1 página pero el valor recomendado es el de 16 páginas.

**Long TX HWM (LTXHWM).**- Especifica la marca de la transacción más larga que puede haber, el tamaño es el porcentaje de espacio disponible en los logs lógicos cuando se llena. Si una transacción de usuario llega a esta marca se realiza un rollback de lo realizado y envía un mensaje de error.

**Logical Log Buffer Size (LOGBUFF).**- Especifica el tamaño en kilobytes de cada uno de los logs lógicos en memoria compartida, se recomienda que el valor esté entre 16 o 32 kilobytes, ó en el caso de cargas de trabajo altas hasta 64 kilobytes.

**Long TX HWM Exclusive (LTXEHWM).**- Especifica la marca de la transacción más larga de acceso exclusivo, se especifica en porcentaje en base al tamaño de los

logs lógicos en caso de que la marca sea excedida realiza un rollback de dicha transacción y manda un mensaje a la terminal del usuario.

**Max # of Logical Logs (LOGSMAXS).**- Especifica el máximo número de archivos de logs lógicos en cada instancia de servidor de BD. El servidor requiere como mínimo de 3 archivos de logs lógicos para funcionar.

**Index Page Fill Factor (FILLFACTOR).**- Especifica el porcentaje de llenado de las páginas de índice. Si es un porcentaje bajo proporciona disponibilidad de espacio para más datos en una página, en caso de que el fill factor este al 100% implica que cuando haya más información se tienen que crear paginas adicionales para la agrupación de la información de acuerdo al índice.

**Max # of Lock (LOCKS).**- Especifica el número máximo de bloqueos disponibles para el servidor de BD durante los procesos, cada bloqueo toma 44 byte de memoria compartida. Por ejemplo si se asigna 1, 000,000 al parámetro LOCKS, el servidor aloja 40 Megabyte de memoria compartida para los bloqueos.

**Add Seg Size(K) (SHMADD)** .- Especifica el tamaño en segmentos que dinámicamente se añaden para la porción virtual de la memoria compartida. Se recomienda de acuerdo a la memoria disponible la siguiente cantidad de segmentos:

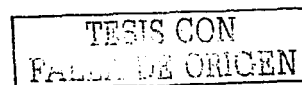
Cantidad de memoria fisica	SHMADD Recomendado
Menor que 256 Megabytes	8192
Mayor a 256 pero menor a 512 Megabytes	16,384
Mayor a 512 megabytes	32,768

**Max # of Buffers (BUFFERS).**- Especifica el número máximo de buffers, que los procesos de los usuarios tienen disponibles para la lectura y escritura en disco. Por ejemplo si al servidor de BD accesan el 15 % de las aplicaciones de datos el 90 % del tiempo, entonces se necesitan asignar un 15% de buffers considerando el valor total de la memoria física.

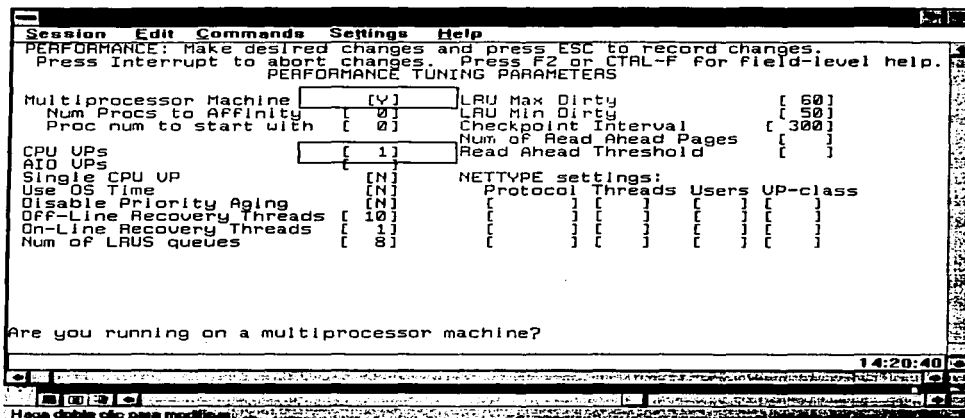
**Total Memory (SHMTOTAL).**- Especifica la cantidad de memoria compartida que podrá ser usada por el servidor de base de datos, el valor de 0 implica que el servidor no tiene límite en cuanto a memoria, el límite estaría marcado por la cantidad de memoria que tiene el equipo.

**Resident Shared Memory size** .- Este parámetro está en función de la cantidad que se este asignando y los buffers que se estén definiendo.

**Page Size.**- Es el tamaño de una página de datos.



### III.2.6.3 Ejecución y Afinación (Tuning and Performance)



**Multiprocesador Machine (MULTIPROCESSOR).**- Este parámetro se activa cuando el equipo donde está instalado el servidor de Base de Datos contiene más de un procesador, en caso de no ser así se coloca en el parámetro N .

**Num Proc to Affinity (AFF\_NPROCS).**- Este parámetro se utiliza cuando un equipo es multiprocesador y especifica el número de CPU's que el servidor de base de datos puede enlazar como procesadores virtuales.

**Proc num to start whit (AFF\_SPROCS).**- Unicamente este parámetro lo soportan los equipos multiprocesador. Especifica el CPU, iniciando con 0, sobre el cual inicia el servidor de base de datos el enlace de los procesadores virtuales con los CPU's.

**CPU VPs (NUMCPUVPS).**- Especifica el número de procesadores virtuales de los CPU's que van a correr.

**AIO VPs (NUMAIOVPS).**- Especifica el número de procesadores virtuales de la clase AIO que van a correr, afectan directamente la lectura y escritura en disco.

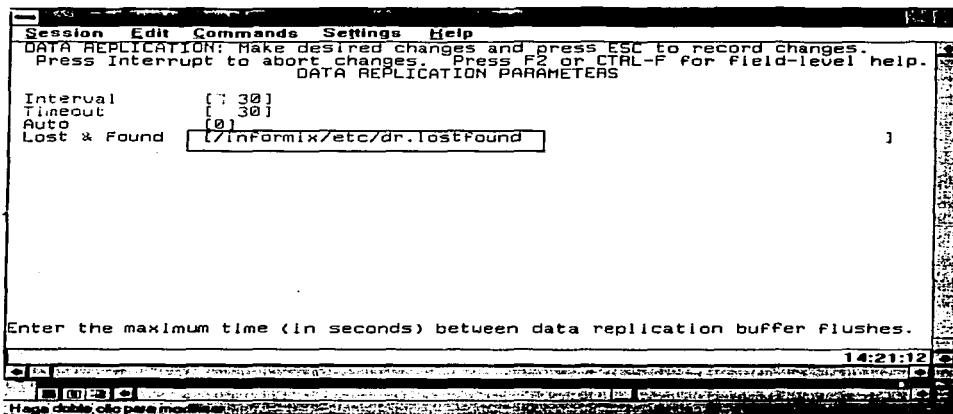
TESIS CON  
FALLA DE ORIGEN

- Single CPU VP (SINGLE\_CPU\_VP).**- Especifica o no si el servidor de BD está corriendo con únicamente un procesador virtual.
- Use OS Time (USEOSTIME).**- Especifica si el servidor de Base de Datos va a usar de manera granular en segundos cuando se obtiene el tiempo del sistema operativo para las sentencias SQL.
- Disable Priority Aging(NOAGE).**- Algunos sistemas operativos bajan la prioridad de los procesos cuando duran un periodo largo de tiempo, cuando se activa este parámetro no baja la prioridad de procesos con un largo tiempo de procesamiento. Si el sistema maneja este tipo de prioridad informix recomienda que se active.
- Off-Line Recovery Threads(OFF\_RECVRV\_THREADS).**- Es el número de hilos(procesos) que se van a recuperar en recuperaciones lógicas cuando el servidor esta fuera de línea (durante una restauración fría).
- On-Line Recovery Threads(ON\_RECVRV\_THREADS).**- Es el máximo número de hilos que se deben recuperar cuando el servidor de BD hace una recuperación lógica (esto es en una restauración en caliente).
- Num of LRUS queues (LRU).**- Especifica el número de LRU (paginas con uso reciente) en las colas de almacenamiento en la memoria compartida.
- LRU Max Dirty (LRU\_MAX\_DIRTY).**- Especifica el porcentaje de paginas modificadas en el LRU, cuando es alcanzado este porcentaje la cola en la memoria compartida es limpiada.
- LRU Min Dirty (LRU\_MIN\_DIRTY).**- Especifica el porcentaje de páginas modificadas en la cola del LRU, cuando alcanza este porcentaje es limpiado.
- Checkpoint Interval (CKPTINTVL).**- Especifica la frecuencia expresada en segundos en el cual el servidor de base de datos revisa y determina si un checkpoint es necesario. Cuando un checkpoint ocurre las paginas en la memoria compartida son escritas a disco.
- Num of Read Ahead Pages (RA\_PAGES).**- Especifica el número de paginas en disco que intenta leer durante una secuencia de revisión de datos ó índices registrados.
- Read ahead Threshold (RA\_THRESHOLD).**- Es utilizado con el parámetro anterior cuando el servidor lee durante una secuencia de registros de datos e índices. RA\_THRESHOLD especifica el umbral de lecturas.
- NETTYPE.**- Ese parámetro permite implementar múltiples conexiones.

TESIS CON  
FALLA DE ORIGEN



### III.2.6.4 Replicación de datos



**Interval (DRINTERVAL).**- Especifica el máximo intervalo de tiempo en segundos durante el vaciado de la replicación de datos. Para la actualización sincronía se debe activar este parámetro con 1.

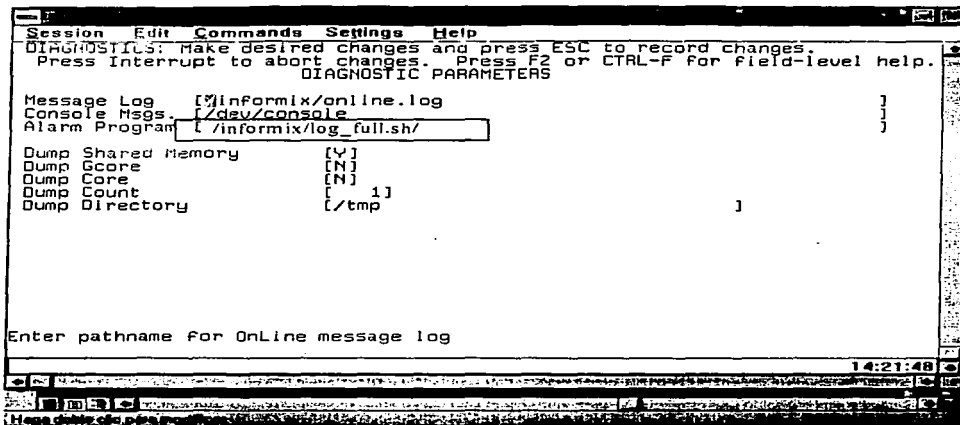
**Timeout (DRTIMEOUT).**- Aplica únicamente en la replicación de datos y especifica la longitud de tiempo en segundos que puede esperar al momento de realizar la replicación de datos.

**Auto (DRAUTO).**- Determina como un servidor de base de datos secundario reacciona ante una falla de replicación.

**Lost & Found (DRLOSTFOUND).**- Especifica la ruta del archivo que contiene las transacciones terminadas en el servidor de base de datos primario pero no las del servidor secundario cuando el servidor primario tiene una falla.

TESIS CON  
FALLA DE ORIGEN

### III.2.6.5 Diagnóstico



**Message Log (MSGPATH).**- Especifica la ruta completa del archivo de bitácora, el servidor escribe el estatus de mensajes y diagnósticos durante la operación del servidor.

**Console Msgs (CONSOLE).**- Especifica la ruta destino para los mensajes de la consola.

**Alarm Program (ALARMPROGRAM).**- Especifica la ruta completa de los archivos ejecutables escritos y ejecuciones del servidor de BD cuando un evento ocurre.

**Dump Shared Memory (DUMPSHMEM).**- Indica si la memoria compartida puede ser respaldada si una falla ocurre. Toda la memoria compartida usada por el servidor de BD es respaldada, este puede resultar bastante largo. La memoria compartida es respaldada en el directorio especificado por el parámetro DUMDIR. El archivo queda con el nombre de **shmем.pid.cnt**. El pid es el identificador del proceso virtual, el cnt es un contador por cada procesador virtual que tiene alguna falla.

**Dump Gcore (DUMPGCORE).**- Es usado cuando el sistema operativo soporta gcore. Si se activa esta opción pero el sistema operativo no la soporta los mensajes son enviados a la bitácora (Message Log). Este archivo es salvado en directorio especificado por la variable DUMDIR y queda con el nombre de **core.cnt.pid**.

TESIS CON  
FALLA DE ORIGEN



**Dump Core (DUMPCORE).**- Controla las fallas ocasionadas en el servidor de BD en un procesador virtual enviándolos a un respaldo central . El archivo core es almacenado en la trayectoria donde apunta el parámetro DUMPDIR.

**Dump Count (DUMPCNT).**- Especifica el número de fallas que un servidor de BD puede respaldar tanto de la memoria compartida como los archivos core generados por el gcore.

**Dump Directory (DUMPDIR) .** - Especifica el directorio donde el servidor de base de datos respalda la memoria compartida, los archivos gcore o mensajes cuando una inserción falla. Los archivos pueden generarse muy grandes por lo que es recomendable que la trayectoria a donde apunta este parámetro tenga suficiente espacio en disco.

### III.2.6.6 Consultas paralelas a Bases de Datos

The screenshot shows a window titled "PARALLEL DATABASE QUERIES PARAMETERS". The window has a menu bar with "Session", "Edit", "Commands", "Settings", and "Help". Below the menu bar, there are instructions: "PDQ: Make desired changes and press ESC to record changes. Press Interrupt to abort changes. Press F2 or CTRL-F for field-level help." The main area contains the following configuration:

```

PARALLEL DATABASE QUERIES PARAMETERS

PDQ Priority                [ 2 0 ]
Max PDQ Priority           [ 100 ]
Decision Support Queries   [          ]
Decision Support Memory (Kbytes) [          ]
Maximum Decision Support Scans [ 1048576 ]
Dataskip [off]
Optimizer Hint             [ 2 ]
    
```

At the bottom of the window, it says "Enter the pdq priority." and the time "14:22:27" is displayed in the bottom right corner.

**Max PDQ Priority (MAX\_PDQPRIORITY).**- Este parámetro limita los recursos de la consulta a BD en paralelo que el servidor de BD puede alojar para una consulta en un sistema de soporte decisión. Este factor es usado para escalar el valor de la prioridad de las consultas a BD en paralelo asignado para los usuarios.

TESIS CON  
FALLA DE ORIGEN



**Decisión Support Queries (DS\_MAX\_QUERIES).**- Es el número máximo de consultas que pueden ejecutarse concurrentemente. El manejador de acceso a memoria para las consultas utiliza la siguiente fórmula:

$$\text{Memoria reservada} = (\text{DS\_TOTAL\_MEMORY} / \text{DS\_MAX\_QUERIES}) * \text{DS\_MAX\_QUERIES} * (\text{PDQPRIORITY} / 100) * (\text{MAX\_PDQPRIORITY} / 100)$$

**Decisión support Memory (kbytes)(DS\_TOTAL\_MEMORY).**- Especifica la cantidad de memoria disponible para las consultas PDQ. Depende directamente de la cantidad de la memoria física. No se debe confundir este parámetro con el parámetro SHMTOTAL. Para aplicaciones de transacciones en línea (OLTP), este parámetro se asigna entre un 20 y un 50 % del valor de SHMTOTAL en kilobytes, para aplicaciones con consultas de soporte-decisión (DSS) el valor de DS\_TOTAL\_MEMORY debe estar entre el 50 y el 80 % de SHMTOTAL. Si la instancia de servidor de BD de datos es exclusiva para DSS este parámetro se debe asignar para el 90 % del total asignado al parámetro SHMTOTAL.

**Maximum Decisión Support Scans (DS\_MAX\_SCANS).**- Limita el número de revisiones en el PDQ por procesos que el servidor de BD puede ejecutar de manera concurrente.

**Dataskip (DATASKIP).**- Permite evitar puntos de fallas medianas. Esta capacidad puede resultar para la alta disponibilidad de los datos. Esto hace que el servidor de BD brinque algún o todos los fragmentos no disponibles cuando es activado este parámetro.

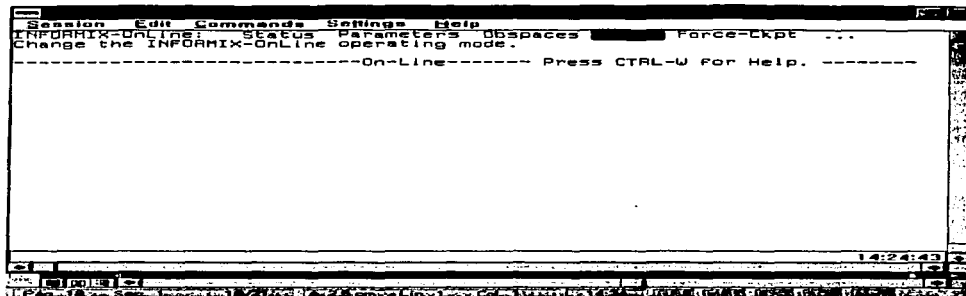
**Optimizer Hint (OPTCOMPIND).**- Ayuda a que el optimizador seleccione un apropiado método de acceso para las aplicaciones.

La forma de almacenar los cambios en cada una de las pantallas es utilizando la tecla ESC.

Finalmente se realiza el formateo de acuerdo a la configuración realizada en las pantallas anteriores, confirmando en la pantalla siguiente la inicialización para guardar todos los datos configurados.

TESIS CON  
FALLA DE ORIGEN

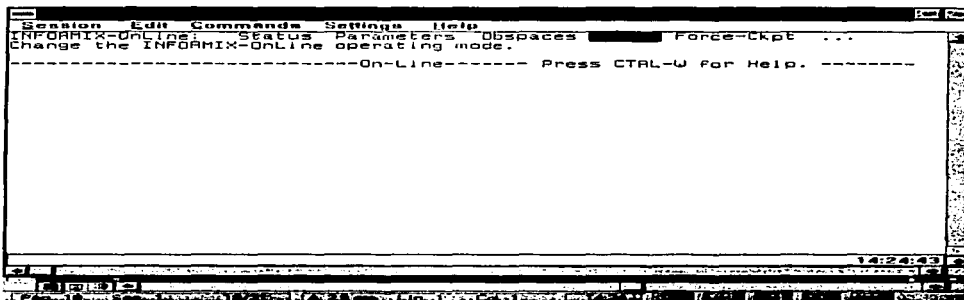




*Dar de alta el Manejador de Informix.*

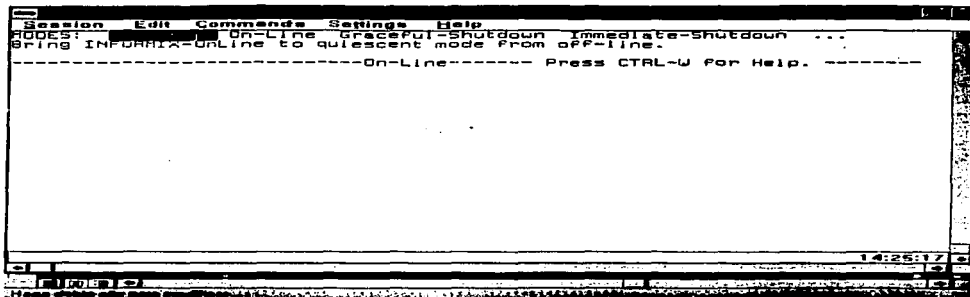
Al terminar de asignar todos los parámetros para verificar que no hubo ningún problema en la configuración se da de alta el Servidor, con la misma utilería onmonitor de la siguiente forma:

En la opción MODE del menú del programa onmonitor



Al seleccionar la opción de mode se presentará la siguiente pantalla que contendrá las opciones para dar de alta el servidor de base de datos.

TESIS CON  
FALLA DE ORIGEN



Se selecciona **Start\_Up**, para que el servidor pase al estatus de **On\_Line**.

Para la revisión del manejador para que o haya presentado ningún error al momento de la inicialización se debe realizar lo siguiente:

Se edita el archivo **online.log** del directorio informix y se verifica que aparezca la línea **sysmaster database built successfully**

vi online .log

Thu Jan 23 16:01:57 1997

16:01:57 DR: DRAUTO is 0 (Off)

16:01:58 listener-thread: err = -930: Cannot connect to database server (cepar02).

16:01:58 Attempting to bring listener thread down

16:01:58 PANIC: Attempting to bring system down

Thu Jan 23 16:04:01 1997

16:04:01 DR: DRAUTO is 0 (Off)

16:04:24 INFORMIX-OnLine Initialized -- Complete Disk Initialized.

16:04:24 Warning: Invalid dbspace 'tempodbs' listed in DBSPACETEMP.

16:04:24 Checkpoint Completed: duration was 0 seconds.

16:04:24 Dataskip is now OFF for all dbspaces

16:04:24 Quiescent Mode

16:04:43 On-Line Mode

16:04:43 Building 'sysmaster' database ...

16:05:06 **'sysmaster' database built successfully.**

TESIS CON  
FALLA DE ORIGEN

Si no aparece la última línea en negritas el manejador de Base de Datos Informix, NO SE GENERO ADECUADAMENTE. Lo cual requiere nuevamente realizar la configuración apoyados en el error que está enviando

### III.2.7 Áreas Físicas y Lógicas

#### III.2.7.1 Creación de los Dbspaces y Chunks (Áreas Físicas)

Para crear un dbspace puede ser con la utilería del onmonitor o desde la vía de comandos.

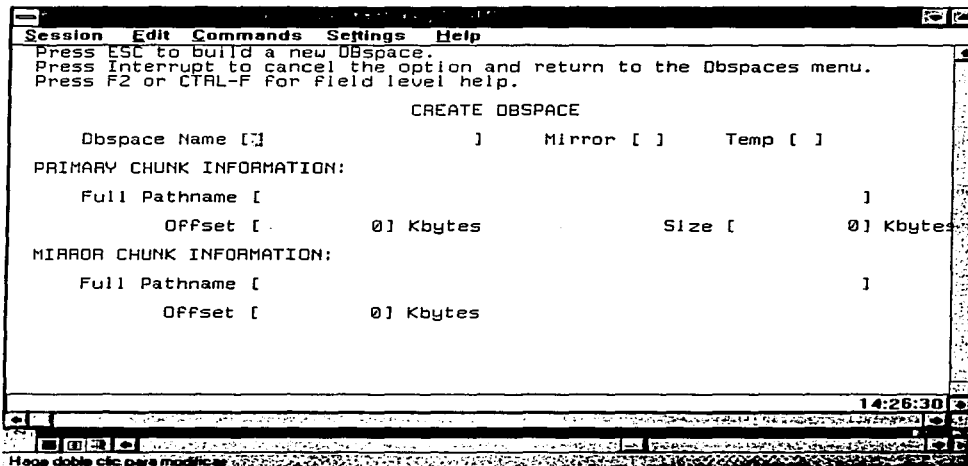
El siguiente procedimiento es para dar de alta un dbspace con la utilería onmonitor.

\$ onmonitor

Seleccionar dbspaces

Seleccionar create

Aparece la siguiente pantalla



La pantalla de creación muestra la siguiente información:

### III.2.7.1 Creación de los Dbspaces y Chunks

Dbospace Name .- es el nombre lógico de un área de BD.

Mirror.- Este parámetro indica si el dbospace va a tener un espejo.

Temp.- Indica si el dbospace va a ser temporal.

#### PRIMARY CHUNK INFORMATION:

Full Pathname.- Es la ruta donde se encuentra el área donde van a residir los datos ya sea un área cruda o un file system.

Offset.- Espacio de compensación del dbospace determinado en kilobytes.

Size.- Tamaño del dbospace en kilobytes.

#### MIRROR CHUNK INFORMATION:

Full Pathname.- En caso de que la bandera de espejo haya sido activada en este apartado se colocará la ruta donde va a residir el dispositivo espejo.

Offset.- Espacio de compensación para el área espejo en kilobytes.

La segunda forma de dar de alta un dbospace, es vía de comandos con la siguiente instrucción:

#### **onspaces -c**

Ejemplo: Con la siguiente instrucción se crea un dbospaces de 2.00 GB

```
onspaces -c -d dbspce1 -p /dev/raw_dev1 -o 5000 -s 2048000 -m /dev/raw_dev2 5000
```

-d .- Especifica el nombre del dbospace.

-p .- Es la ruta donde se encuentra el área donde va a residir el dbospace

-o .- Es el offset (espacio de compensación.)

-s .- Tamaño del dbospaces en kilobytes

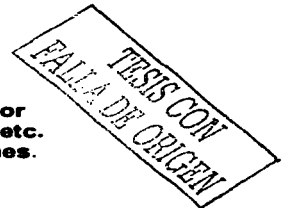
-m.- La trayectoria donde se encuentra el área del espejo del dbospace que se está creando.

El número que aparece al final es el offset del área espejo.

A continuación se presentan los nombres lógicos con los que están dados de alta los dbospace para la aplicación SIAPSEP.

Nombre de las ligas correspondientes /informix/nombre liga

<b>rdbdats</b>	--> Para alojar la Tabla de Datos
<b>rdbroots</b>	--> Para alojar la Estructura del manejador
<b>rdbtemp</b>	--> Para alojar Tablas temporales, Sorts etc.
<b>rdblogs</b>	--> Para alojar el archivo de Transacciones.



En caso de que el área del dbspace se llene se tiene que agregar más espacio para el futuro crecimiento por lo que se tienen que añadir uno ó más chunks dependiendo el crecimiento de la base de datos y el espacio en disco disponible. Los chunks son áreas físicas (particiones de disco).

En el caso de querer añadir un chunk a un dbspace se tienen dos opciones una utilizando el onmonitor y otra la vía de comandos, para el caso del onmonitor se realizan los siguientes pasos:

Con el usuario de informix y desde cualquier directorio aplicar la utilería onmonitor

Seleccionar del menú **Dbspaces**

Después seleccionar **Add\_chunk**

Aparecerán los dbspaces que se encuentran dados de alta, después se procede a seleccionar el dbspace al que se quiere añadir el chunk, posicionado en el dbspace donde se va a añadir el chunk se presionan las teclas **CTRL-B ó F3**.

La siguiente pantalla indica si va a ser espejo o nó, presionar la letra Y(si) ó N(no). Si el chunk que va a ser añadido es temporal colocar Y ó N.

En caso de que el Dbspace o el Blobspace sean espejos, se tiene que definir un chunk primario y un chunk espejo y dar la trayectoria completa del nuevo chunk primario en la indicación **Full Pathname** de la Sección de Chunk primario.

Añadir el tamaño del chunk en kilobytes, en donde está el artículo **Size**.

Si el chunk es espejo se completa toda la información, del pathname, size y offset opcional en la sección de **mirror-chunk** en la misma pantalla.

La otra forma de dar de alta un chunk es con la utilería onspaces y la opción -a como se muestra a continuación:

```
onspaces -a bddata3 -p /dev/raw_dev1 -o 200 -s 10000 -m /dev/raw_dev2 200
```

En está opción al dbspace bddata3 se le está agregando un chunk ubicado en /dev/raw\_dev1 con un offset de 200 kilobytes y un tamaño de 10000 kilobytes, la opción m se agrega en caso de que el dbspace haya sido dado de alta con un espejo en caso de no ser así se omite ese parámetro.

Después de haber agregado las áreas en donde se va a realizar la carga de datos se tienen que dar de alta los logs ya que es recomendable quitarlos del dbspace de root y colocarlos en un dbspace aparte. La forma de dar de alta los logs lógicos se explica a continuación.

TESIS CON  
FALLA DE ORIGEN

### III.2.7.2 Creación de Archivos Logs (Áreas Lógicas)

De acuerdo al valor establecido para el dbspace de logs, deben darse de alta estos colocando un tamaño de acuerdo al tamaño de BD de datos y las transacciones que se puedan realizar.

Antes de dar de alta los logs el manejador de Base de Datos debe estar en modo Monousuario (Quiescent), los estatus ó modos de la base de datos se explicarán a detalle en el siguiente capítulo.

Para poner en el modo monousuario el manejador con el usuario informix se ejecuta el siguiente comando:

**\$ onmode -s**

Para generar los logs se tecléa el siguiente comando :

**\$ onparams -a -d <nombre del dbspace de logs> -s <tamaño>**

**Este comando se repite de acuerdo al número de logs que se necesiten crear sin exceder el tamaño del dbspaces de logs.**

Al termino de la creación de todos los logs, se deben de activar para lo cual se efectúa la siguiente instrucción desde la línea de comandos

**\$ onmode -l**

Para realizarlo utilizando el onmonitor se realiza lo siguiente:

Se coloca en modo monousuario el manejador de BD y se selecciona del menú Parameters → Add-Log.

Ubicados en ese submenú se indica el dbspace donde van a residir los nuevos archivos de log lógicos. Automáticamente aparece el tamaño de los logs en el campo *Logical Log Size* .

Hasta este punto termina la configuración del manejador de Base de Datos de Informix 2000, por consiguiente se requiere hacer la importación de la Base de Datos que contiene como mencionamos anteriormente 381 tablas.

Como se ha mencionado el Sistema Integral de Administración de la SEP es nuestro sistema de aplicación, el cual consta de las siguientes tablas, para ello el servidor se configura para soportar está cantidad de información:

TESIS CON  
FALLA DE ORIGEN

## Relación de Tablas del SIAPSEP

'informix'.a\_fup  
'informix'.a\_pg9999  
'informix'.a\_s102  
'informix'.a\_sub4+0  
'informix'.act\_inact  
'informix'.acum\_cheque\_cpto  
'informix'.acum\_cheque\_sap  
'informix'.acumulados  
'informix'.agrupacion  
'informix'.agrupacion\_cpto  
'informix'.agrupacion\_ct  
'informix'.aguinaldo  
'informix'.ahisa\_desc  
'informix'.ahisa\_desc75  
'informix'.ahisa\_desc\_paso  
'informix'.area\_sep  
'informix'.baja\_persona  
'informix'.ban\_cam\_plaza  
'informix'.ban\_cat\_puesto  
'informix'.ban\_ins\_cpto  
'informix'.ban\_ins\_mens  
'informix'.ban\_pago  
'informix'.ban\_pago\_aut  
'informix'.banco\_plaza  
'informix'.bancos  
'informix'.bancos\_cuenta  
'informix'.basifica\_emp  
'informix'.benef\_pen\_alim  
'informix'.bolsa\_trabajo  
'informix'.calc\_errorlog  
'informix'.calc\_errorlog\_ms  
'informix'.camdaper  
'informix'.cancela\_pago  
'informix'.cancelacion  
'informix'.cancelaux  
'informix'.canje  
'informix'.capitulos\_pres  
'informix'.cat\_act\_ins  
'informix'.cat\_ct\_ocu  
'informix'.cat\_directivas  
'informix'.cat\_fuente\_fin  
'informix'.cat\_funcion  
'informix'.cat\_gasto\_esp  
'informix'.cat\_mms  
'informix'.cat\_ogt  
'informix'.cat\_paso  
'informix'.cat\_prog\_esp  
'informix'.cat\_prog\_sec  
'informix'.cat\_proyecto  
'informix'.cat\_puesto\_cptos  
'informix'.cat\_servicios\_ct  
'informix'.cat\_subfuncion  
'informix'.cat\_subpart  
'informix'.cat\_tipo\_gasto  
'informix'.cat\_ur\_e  
'informix'.cat\_ur\_s  
'informix'.categoria direc  
'informix'.categoria\_puesto  
'informix'.cates\_48hrs  
'informix'.cates\_bas  
'informix'.cates\_dir  
'informix'.cecoban  
'informix'.ced\_forte  
'informix'.cedula\_forte  
'informix'.censo\_eco  
'informix'.centro\_computo

'informix'.centro\_serpub  
'informix'.centro\_trabajo  
'informix'.cheq\_cpo\_dif\_sit  
'informix'.cheque2  
'informix'.cheque3  
'informix'.cheque4  
'informix'.cheque\_banamex  
'informix'.cheque\_banamex\_pa  
'informix'.cheque\_cc  
'informix'.cheque\_concal  
'informix'.cheque\_cpto  
'informix'.cheque\_cpto\_24  
'informix'.cheque\_cpto\_241  
'informix'.cheque\_cpto\_becas  
'informix'.cheque\_cpto\_cancel  
'informix'.cheque\_cpto\_dif  
'informix'.cheque\_cpto\_dif\_pa  
'informix'.cheque\_cpto\_est  
'informix'.cheque\_cpto\_msys  
'informix'.cheque\_cpto\_ori  
'informix'.cheque\_cpto\_ori\_pa  
'informix'.cheque\_cpto\_pa  
'informix'.cheque\_cpto\_pa24  
'informix'.cheque\_cpto\_pa\_241  
'informix'.cheque\_cpto\_pamsys  
'informix'.cheque\_cpto\_rei  
'informix'.cheque\_cpto\_rei\_pa  
'informix'.cheque\_dif  
'informix'.cheque\_dif\_paes  
'informix'.cheque\_difes  
'informix'.cheque\_pa2  
'informix'.cheque\_pa3  
'informix'.cheque\_pa4  
'informix'.cheque\_paso  
'informix'.cheque\_paso\_pa  
'informix'.cheque\_sap  
'informix'.cheque\_sustra  
'informix'.cheques\_err  
'informix'.cif\_ct\_ap\_mv  
'informix'.cifras\_control  
'informix'.clasif\_serv  
'informix'.clasificadores  
'informix'.clave\_nep  
'informix'.clave\_presupuesto  
'informix'.cod\_paren  
'informix'.codigos\_operacion  
'informix'.concepto  
'informix'.concepto49  
'informix'.concepto\_calculo  
'informix'.conper\_aux2  
'informix'.conperdec2  
'informix'.cons\_hist\_pago  
'informix'.control\_acum\_cptos  
'informix'.control\_qna\_proc  
'informix'.cpto\_34  
'informix'.cpto\_dese  
'informix'.cptos\_pres  
'informix'.cptos prorrateados  
'informix'.crea\_can\_coderr  
'informix'.crea\_can\_error  
'informix'.crea\_can\_pza  
'informix'.credito\_salario  
'informix'.ct\_audi2  
'informix'.ct\_audita  
'informix'.ct\_basico\_nep  
'informix'.ct\_concepto





CONFIGURACIÓN REQUERIDA PARA LA IMPLANTACIÓN  
DEL SISTEMA SIAPSEP EN INFORMIX

TEMA III

'informix'.ct\_servicio  
'informix'.ctrl\_prog  
'informix'.cuentas\_cheque  
'informix'.curp\_sp  
'informix'.cve\_cong  
'informix'.cve\_mov  
'informix'.debito\_janamax  
'informix'.ded\_hist\_pago  
'informix'.deduccion  
'informix'.dep\_admva  
'informix'.dep\_norm  
'informix'.dependientes  
'informix'.des\_baja  
'informix'.del\_reintegr  
'informix'.detalle\_mov1  
'informix'.dir\_reg  
'informix'.doc\_control  
'informix'.doc\_mov\_tec\_fin  
'informix'.e\_issste  
'informix'.edo\_civil  
'informix'.edo\_curp  
'informix'.edo\_mun\_loc  
'informix'.emp\_cpto  
'informix'.emp\_gobfed  
'informix'.emp\_ni  
'informix'.emp\_plaza  
'informix'.emp\_plaza\_cmsys  
'informix'.emp\_plaza\_cpto  
'informix'.emp\_plaza\_fal  
'informix'.emp\_plaza\_faltas  
'informix'.emp\_plaza\_l819  
'informix'.emp\_plaza\_l9dt  
'informix'.emp\_plaza\_msg  
'informix'.emp\_plaza\_tempo  
'informix'.emp\_plaza\_tempo2  
'informix'.emp\_pza\_cpto\_aux  
'informix'.emp\_sp  
'informix'.empleado  
'informix'.empleado\_banco  
'informix'.empleado\_curp  
'informix'.empleado\_nss  
'informix'.entero\_shcp  
'informix'.entidades\_fed  
'informix'.error\_de\_sistema  
'informix'.errores\_de\_movimie  
'informix'.errores\_prueba  
'informix'.errores\_rec  
'informix'.escuela\_origen  
'informix'.estatus  
'informix'.estatus\_cheque  
'informix'.estatus\_ct  
'informix'.estatus\_mov\_tec  
'informix'.estatus\_plaza  
'informix'.estr\_org\_ct  
'informix'.estr\_org\_ls  
'informix'.etapas\_calc  
'informix'.fecha\_qna  
'informix'.fecha\_qna2  
'informix'.formas  
'informix'.formato  
'informix'.fovim\_desc  
'informix'.fovim\_p  
'informix'.fovim\_r  
'informix'.fovissste\_desc  
'informix'.grupo\_gen  
'informix'.grupo\_usu  
'informix'.grupos  
'informix'.guiones  
'informix'.hist\_cheque\_paso

'informix'.hist\_paso  
'informix'.hist\_plaza\_ct  
'informix'.hist\_plaza\_estatus  
'informix'.historia\_cancelaci  
'informix'.historia\_cheque  
'informix'.historia\_cheques  
'informix'.historia\_pago  
'informix'.historia\_pago\_aux  
'informix'.historia\_pagohis  
'informix'.historia\_sar  
'informix'.historia\_sp  
'informix'.hillist\_ep  
'informix'.hillist\_hpe  
'informix'.ident\_doc1  
'informix'.ident\_doc2  
'informix'.identificadores  
'informix'.importes\_ue  
'informix'.inasistencias  
'informix'.ispt  
'informix'.issste\_desc  
'informix'.issste\_nss  
'informix'.issste\_nssl  
'informix'.issste\_nsnl  
'informix'.l\_totfallas  
'informix'.licencias  
'informix'.menu  
'informix'.mod\_pago\_cptos  
'informix'.mod\_sub  
'informix'.mod\_sub\_ssub  
'informix'.modalidad  
'informix'.modelos  
'informix'.modsub\_msg  
'informix'.mot\_serpub  
'informix'.motivo\_cancelacion  
'informix'.motivo\_canje  
'informix'.motivo\_lnu  
'informix'.motivo\_reexpedicio  
'informix'.motivo\_reintegr  
'informix'.motivo\_reposicion  
'informix'.motivo\_sus  
'informix'.motivos\_mov  
'informix'.mov\_tecfin\_bt\_nep  
'informix'.movd\_bajapp  
'informix'.movm\_bajapp  
'informix'.movr\_detalle  
'informix'.movs\_arch\_pza  
'informix'.movs\_plaza  
'informix'.movs\_prep\_catalogo  
'informix'.movs\_preparados  
'informix'.movs\_reclamo  
'informix'.movs\_reintegr  
'informix'.movs\_tec\_fin\_nep  
'informix'.movtos\_archivo  
'informix'.movtos\_archivo\_dia  
'informix'.movtos\_archivo\_vio  
'informix'.movtos\_sp  
'informix'.msg\_cheque  
'informix'.municipios  
'informix'.nips\_ct  
'informix'.nivel\_academico  
'informix'.niveles\_reporte  
'informix'.nom\_coyo  
'informix'.nvoim  
'informix'.obl\_gasto  
'informix'.oficina\_pago  
'informix'.operador\_cpto\_exc  
'informix'.pago\_debito  
'informix'.parametros  
'informix'.partida

TESIS CON  
TABLA DE ORIGEN

CONFIGURACIÓN REQUERIDA PARA LA IMPLANTACIÓN  
DEL SISTEMA SIAPSEP EN INFORMIX

TEMA III

'informix'.paso111  
'informix'.paso\_19  
'informix'.paso\_600  
'informix'.paso\_601  
'informix'.paso\_concepto  
'informix'.paso\_cpto\_beca  
'informix'.paso\_cpto\_imp  
'informix'.paso\_cptoh  
'informix'.paso\_ct  
'informix'.paso\_ct1  
'informix'.paso\_ctb  
'informix'.paso\_ctb\_hist  
'informix'.paso\_igdt  
'informix'.paso\_resumen  
'informix'.paso\_rfc  
'informix'.paso\_sp  
'informix'.pasoche\_est  
'informix'.pen\_alim  
'informix'.perc\_ded  
'informix'.perc\_hist\_pago  
'informix'.percepcion  
'informix'.perdedinf  
'informix'.permsys  
'informix'.plaza  
'informix'.prima\_ant\_aux  
'informix'.proc\_log  
'informix'.procesos  
'informix'.ptda\_concepto  
'informix'.ptda\_mot\_mov  
'informix'.puesto\_solicitud  
'informix'.r\_cheque\_cpto  
'informix'.ramo\_ur  
'informix'.ramos  
'informix'.ran\_cheq\_inu  
'informix'.ran\_cheq\_sus  
'informix'.ran\_cheq\_sustra  
'informix'.ran\_cheq\_util  
'informix'.ran\_cheq\_x\_usar  
'informix'.rango\_cheque  
'informix'.reexpedicion  
'informix'.ref\_x\_cptos  
'informix'.reposicion  
'informix'.responsabilidades  
'informix'.resumen\_pzaact  
'informix'.rfc2  
'informix'.rfc3  
'informix'.rfc4  
'informix'.rfc\_calculo  
'informix'.rfc\_calculo2  
'informix'.rfc\_carga  
'informix'.rfc\_falta\_h  
'informix'.rfc\_forte  
'informix'.rfc\_fup  
'informix'.rfc\_id  
'informix'.rfc\_msys  
'informix'.rfc\_paso2

'informix'.rfc\_uni  
'informix'.rfcepc  
'informix'.rfchom  
'informix'.salario\_minimo  
'informix'.serv\_pub  
'informix'.serv\_pub2  
'informix'.servicio\_nep  
'informix'.severidades  
'informix'.sexo  
'informix'.situacion\_cheque  
'informix'.sostenimiento  
'informix'.subunidad  
'informix'.system\_user1  
'informix'.t\_cheque\_cpto  
'informix'.t\_sar\_qna  
'informix'.t\_sar\_qna\_rec  
'informix'.tab\_141  
'informix'.tab\_141a  
'informix'.tab\_141b  
'informix'.tab\_ptos  
'informix'.tab\_rev  
'informix'.tab\_ur  
'informix'.tabla\_aux  
'informix'.tabla\_aux2  
'informix'.tables  
'informix'.taux\_q1415  
'informix'.tec\_fin\_ejer\_nep  
'informix'.tec\_fin\_imp\_nep  
'informix'.techo  
'informix'.techo\_aux  
'informix'.techo\_fin\_nep  
'informix'.ter\_paso  
'informix'.ter\_rfc  
'informix'.tipo\_calculo  
'informix'.tipo\_inasistencia  
'informix'.tipo\_licenciatura  
'informix'.tipo\_localidad  
'informix'.tipo\_mov  
'informix'.tipo\_pago  
'informix'.tipo\_plantel  
'informix'.tipo\_titulacion  
'informix'.tipos\_sosten  
'informix'.tot\_hist\_pago  
'informix'.total\_faltas  
'informix'.unidad  
'informix'.unidad\_dist\_cheq  
'informix'.unidad\_resp  
'informix'.unidad\_serpub  
'informix'.unidad\_subunidad  
'informix'.usuario\_modalidad  
'informix'.usuario\_nep  
'informix'.usuario\_ur  
'informix'.vales\_despensa  
'informix'.xserver\_info  
'informix'.xsql\_languages  
'informix'.zona\_e

TEJES CON  
FALLA DE ORIGEN

En este capítulo, se realizó la configuración tanto del sistema operativo Unix, como de Informix (IDS200), para poder poner en marcha las bases de datos del SIAPSEP, en el siguiente capítulo se explicará la forma en que se le da mantenimiento al manejador de BD, de manera general la forma en como se lleva a cabo la administración del manejador, así como examinaremos algunas utilerías que nos servirán para revisión de inconsistencias generales.

TESIS CON  
FOTOCOPIA DE ORIGEN

## TEMA IV

### OPERACIÓN Y MANTENIMIENTO DEL SERVIDOR DE BASE DE DATOS

El mantener un margen alto de disponibilidad en los sistemas y equipos de toda organización es un reto constante, por lo que es responsabilidad de los administradores de sistemas lograr mantener esa disponibilidad. Para que esto se pueda llevar a cabo es necesario dar mantenimiento a todo lo que se encuentra alrededor de dichos sistemas, como lo son los servidores en donde están las bases de datos, sistemas operativos, sistemas de red, sistemas de aire acondicionado, etc.

Cada uno de estos elementos son un punto importante, y es por ello que no deben ser descuidados en lo mínimo. Nosotros nos enfocaremos en el mantenimiento al servidor de base de datos, el cual es nuestro punto central.

En este capítulo hablaremos acerca de los modos en que opera el servidor de base de datos Informix y las instrucciones que nos permiten llevar a cabo los cambios, así como utilerías de mantenimiento dentro de las cuales se consideran los respaldos y restauraciones del Servidor de Base de Datos de Informix, como parte fundamental dentro de las tareas de mantenimiento. Para ello existen varias herramientas que complementan esta parte, y es en base a las necesidades de cada organización, lo que determina la utilería o las herramientas que serán utilizadas.

Podemos considerar tres categorías de herramientas de mantenimiento, que son de consulta, modificación, respaldo y restauración.

Las herramientas de consulta proporcionan al administrador un panorama amplio, de cómo está funcionando el servidor de base de datos que son: onstat, oncheck, onlog y el archivo de bitácora (informix.log), en este último van quedando registrados los cambios o fallas que pudiera presentar el servidor de BD.

Las herramientas de modificación son las que permiten efectuar cambios ó reparar fallas de acuerdo al diagnóstico llevado a cabo por las herramientas de consulta dentro de las cuales consideraremos las siguientes: oncheck, onmode, onparams, onspaces y oninit.

Las utilerías de respaldo y restauración son las mismas lo que cambia son las opciones ó parámetros que se utilicen, las cuales permiten tener asegurados los datos para errores que pudieran presentarse ó recuperación de información histórica de las cuales Informix ofrece las siguientes utilerías de respaldo: dbexport, dbimport, ontape, ondblog, onbar y onarchive.


Las utilerías antes mencionadas son ejecutadas desde la vía de comandos, pero existe una utilería gráfica que permite realizar algunas opciones de mantenimiento denominada onmonitor.

El hablar de operación y mantenimiento viene ligado, debido a que por ejemplo el querer realizar el cambio de logeo de una base de datos, requiere que el servidor este operando únicamente para el administrador ó el querer realizar un respaldo con dbexport requiere que el servidor este operando en línea, de ahí que para realizar alguna actividad de mantenimiento en específico se requiera saber en que modo debe estar operando el servidor de Base de Datos.

#### IV.1 MODOS DE OPERACIÓN

La forma en que opera el servidor de base de datos, indica si está o no disponible y para que tipo de usuarios, por ejemplo, cuando se va a dar mantenimiento al servidor y no se desea dar acceso a los usuarios del sistema, únicamente al administrador del servidor de BD, otra forma de operar del servidor puede ser cuando se necesita recuperar el servidor de base de datos de respaldos, entre otros. Esta flexibilidad, es la que hace ver la importancia que tiene el modo de operar del servidor de base de datos.

La forma en que el administrador visualiza el estado actual del servidor de Base de Datos es con la utilería onstat desde la vía de comandos. El modo es desplegado en el encabezado. Como se muestra:



Información Database Server: 1111 On Line Up 1 10 18 01 11

↑  
Indica que el Servidor de BD está en línea.

El servidor de BD tiene tres modos de operación principales los cuales el administrador puede cambiar y son los siguientes:

Modo de Operación	Descripción
Modo off line	Es cuando el servidor no está corriendo. La memoria compartida no ha sido asignada.
Quiescent ó Modo de Administración	Cuando el proceso de inicialización está en ejecución (oninit), los recursos de la memoria compartida son asignados. Pero el sistema no permite el acceso a los usuarios a la base de datos.
Modo On line	En este modo los usuarios pueden conectarse a las Bases de Datos y realizar todas las actividades del servidor de base de datos. Este es el modo de operación normal.

TESIS CON  
FALLA DE ORIGEN



La figura 4.1 muestra cuando el servidor esta fuera de línea (Off-line). En donde se puede visualizar que cualquier usuario que se quiere conectar es rechazado además la memoria compartida aún no ha sido inicializada.

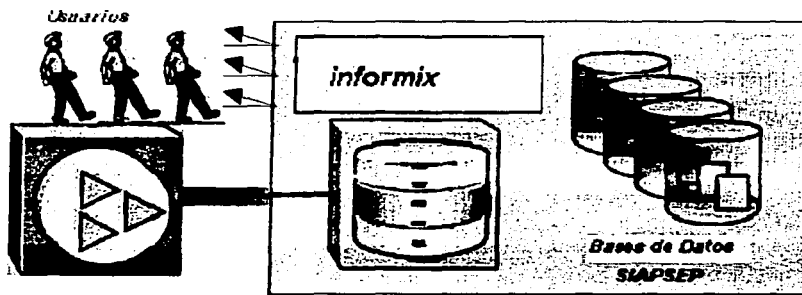


Figura 4.1 Estatus del Servidor de BD Informix fuera de línea

En la figura 4.2 se visualiza el modo de operación Quiescent, en donde los usuarios siguen sin poder acceder a excepción del administrador que puede realizar actividades de mantenimiento, en este modo la memoria compartida es inicializada.

TESIS CON  
FALLA DE ORIGEN



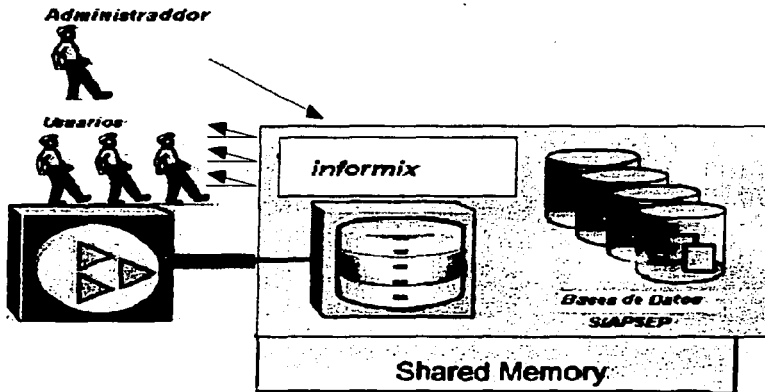


Figura 4.2 Estatus del Servidor de BD Informix monousuario (Quiscent)

La figura 4.3 muestra el modo de operación en línea (On-line) en el cual todos los usuarios permitidos tienen acceso al servidor de base de datos.



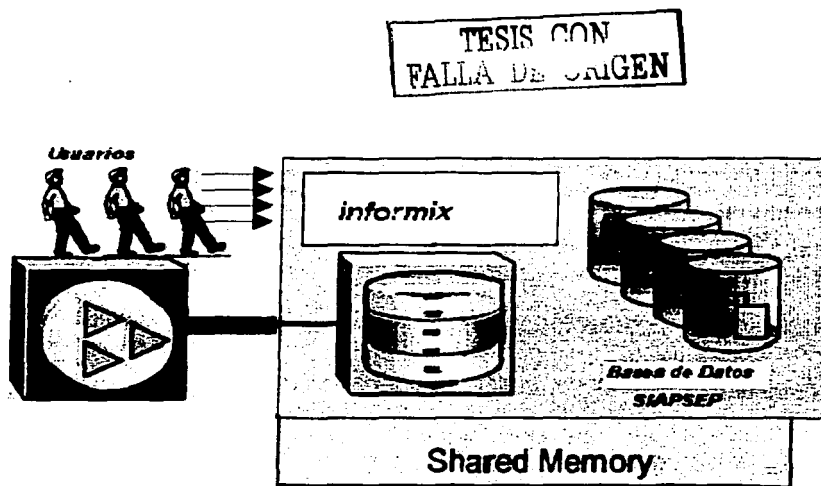


Figura 4.3 Estatus del Servidor de BD Informix en línea (Online)

También de manera adicional el servidor de Base de Datos puede presentar cualquiera de los siguientes estatus de manera automática:

- **Modo Read-only** .- Es usado por el servidor de base de datos secundario (Es el servidor donde se está replicando la información) en alta disponibilidad en la parte de replicación de datos la cual vemos a detalle en el siguiente capítulo. Una aplicación puede consultar el servidor de BD en el modo read-only (sólo lectura), pero las aplicaciones no pueden escribir.
- **Modo Recovery** .- Este modo es transitorio. Esto ocurre cuando el servidor de Base de Datos está en fase de recuperación rápida (fast recovery) ó cuando se restaura de un respaldo, el modo de recuperación ocurre cuando se realiza el cambio del modo fuera de línea (off-line) a modo (monousuario) quiescent.
- **Modo Shutdown** .- Es transitorio y ocurre cuando el servidor de BD cambia de modo on-line a inactivo. Los usuarios actuales siguen realizando sus transacciones activas hasta concluiras, pero nuevos usuarios ya no pueden conectarse. Una vez que el modo shutdown es iniciado, no puede ser cancelado.





### IV.1.2 Cambio de Estatus

Existen dos formas de efectuar el cambio de operación del Servidor de Base de Datos ya sea desde la vía de comandos con las utilerías *oninit* y *onmode* ó con *onmonitor* (utilería visual de *informix*). Estos cambios pueden ser realizados únicamente por el administrador, ya sea del sistema operativo ó del servidor de base de datos.

#### *De Modo Off-line (Inactivo) a Modo Quiescent (Monousuario)*

Cuando el servidor de base de datos cambia de modo *off-line* a *quiescent*, es inicializada la memoria compartida para el servidor de BD, en estos momentos ninguna sesión puede acceder al servidor de BD, los usuarios pueden ver únicamente la información del estatus del servidor y para el caso del usuario administrador del sistema operativo (*root*) y del servidor de base de datos (*informix*) pueden acceder a las opciones administrativas.

La forma de realizar los cambios de estatus en el servidor de base de datos mediante la vía de comandos es la siguiente:

Con la utilería *oninit* y la subopción *-s*.

Y utilizando el *onmonitor*:

Se selecciona la opción de *Mode* → *Startup*

#### *De Modo Off-line a Modo On-line*

Cuando se realiza el cambio del servidor de Base de datos de *off-line* (fuera de línea) a *on-line* (en línea), el servidor de base de datos inicializa la memoria compartida. En este modo de operación el acceso es completo para todos los usuarios.

Para el cambio del servidor de base de datos puede ser solamente desde la vía de comandos con la opción *oninit*, debido a que la utilería *onmonitor* no soporta este cambio.

El siguiente código es un shell de ejecución el cual permite dar de alta el manejador de base de datos:

```
# Shell que da de alta el Manejador de Base de Datos
# Esta parte del shell es en caso de que se quiera dar de alta odebaja el servidor

# Se ejecuta
menu

menu ()
while :
do
clear
encabezado
echo "\n\ TECLEE LA OPCION DE LA ACTIVIDAD A REALIZAR:\n\n"
echo "\t\t 1. ALTA DE SERVIDOR DE BASE DE DATOS\n"
echo "\t\t 2. BAJA DE SERVIDOR DE BASE DE DATOS\n"
echo "\t\t 3. Salir\n\n"
echo "\t\t Opcion : \c"

```

TESIS CON  
FALLA DE ORIGEN



```

read opcion
case $opcion in
  1)
    altaserver
    continue
    break
    ;;
  2)
    bajaserver
    continue
    break
    ;;
  6) echo exit >> $Archsai
    trap 1 2 3
    exit 255;;
altaserver()
(
# Rutina de alta del server seleccionado.
"Archsai=ALTASRV_`date +%y%m%d.%H:%M`"
date > "$Archsai"
# Se solicita la confirmacion de alta del servidor.
namesrv
clear
echo " ALTA DEL SERVIDOR DE BASE DE DATOS
=====

CONFIRMA DAR DE ALTA EL SERVIDOR: $INFORMIXSERVER (s/n)? lc"
Read Respuesta
Echo "respuesta: $Respuesta" >> $Archsai
Case $Respuesta in
s|S|i|I)
  clear; echo "UN MOMENTO ..."
  tmp=""onstat -| grep On-Line | wc -l"
  if [ $tmp = "1" ]
  then (
    echo "ERROR: El servidor ${INFORMIXSERVER} ya estaba dado de alta.";
    date; onstat -
    echo "EL PROCESO DE ALTA FRACASO.":sleep 10;
    ) | tee -a "$Archsai"
  else tmp=""oninit 2>&1| grep "\error in shared memory" | wc -l"
    if [ $tmp = "1" ]
    if ( oninit 2>&1 )
    then (
      date; echo "EL PROCESO DE ALTA FRACASO.";
      echo " *** REPORTE PROBLEMA DE error in shared memory AL AREA DE BASE DE DATOS *****";
      sleep 30;
      ) | tee -a "$Archsai"
    else (
      date;
      tmp3=""onstat -| grep On-Line | wc -l";
      while [ $tmp3 != "1" ]
      do
        tmp3=""onstat -| grep On-Line | wc -l"
        sleep 5
        done
        onstat -;
      date;
      echo "EL SERVIDOR ${INFORMIXSERVER} HA SIDO DADO DE ALTA.";
      echo "presione CTRL-C para terminar...";
      sleep 6;

```

TESIS CON  
FALLA DE ORIGEN

```

) | tee -a "$Archsai"
fi
# ) | tee -a "$Archsai"
;;
) | clear
PSS=$PS1
PS1=""hostname">"
Archsai=MANTO_date +%y%m%d.%H:%M"
date > "$Archsai"
(
echo "Modo mantenimiento!! \n"
echo "exit para regresar...\n"
script
date
echo "SALIO DE SISTEMA OPERATIVO."
) | tee -a "$Archsai"
PS1=$PSS
rm typescript
trap 1 2 3
continue
;;
*) echo "\t\tPor que se arrepintio? ..."
echo "\t\tBYE!!"
esac
;;
# Para restablecer la operacion
#trap 1 2 3
#exit 255
}
# Fin del Archivo

```

Programa Código shell que permite dar de alta el Servidor de BD

### De Modo Quiescent a On-line

El cambio del servidor de base de datos de Quiescent (modo para administración) a On-line implica que los usuarios permitidos puedan tener acceso.

Para realizar este cambio desde la vía de comandos se aplica la siguiente instrucción:

*onmode* y la subopción *s* (*onmode -s*)

y para realizarlo con la utilería *onmonitor* se selecciona *Mode* → *On-line*

Para realizar el cambio inverso de modo en línea a modo Quiescent pueden adoptarse dos estatus, *Gracefully* ó *Immediately*, dependiendo de la sub-opción seleccionada, las cuales son los siguientes:

### De Modo On-line a Quiescent "Gracefully"

De esta forma se restringe el acceso al servidor de base de datos sin interrumpir los procesos actuales. Después de hacer el cambio de modo de operación, el servidor asigna una bandera que previene la conexión de nuevas sesiones y permite que las conexiones actuales terminen su sesión.

TESIS CON  
FALLA DE ORIGEN



Una vez iniciado el cambio de modo de operación ya no puede ser interrumpido. Para realizar el cambio desde la línea de comandos se ejecuta la instrucción siguiente:

*onmode -m ó onmode -sy* la y indica que ya no es necesaria la confirmación

Utilizando la utilidad *onmonitor*:

Se selecciona *Mode* → *Gracefull Shutdown*.

*De Modo On-line a Quiescent "Immediately"*

Este cambio de modo de operación restringe el acceso de manera inmediata al servidor de base de datos y los trabajos en progreso pueden perderse.

Al momento de realizar este cambio se requiere la confirmación del usuario, si la contestación es afirmativa el servidor de base de datos envía una señal de desconexión a todas las sesiones existentes en la memoria compartida.

Los usuarios cuando intentan conectarse estando en el modo *Quiescent Immediately* manda el número de error -459 indicando que el servidor de base de datos fue dado de baja y en el caso de las sesiones existentes al realizar el cambio manda el mensaje de error -457 indicando que la sesión ya fue terminada.

Al realizar este cambio el servidor de base de datos limpia todas las sesiones que fueron terminadas al momento del cambio de operación. En cuanto a la actividad de las transacciones es realizado un *rollback*.

Para realizar el cambio desde la vía de comandos se utiliza el siguiente comando:

*onmode -uy ó onmode -u*

y para el caso de la utilidad *onmonitor* las siguientes opciones:

Se selecciona *Mode* → *Immediate Shutdown*

*De cualquier Modo de Operación a directamente Off-line*

Desde cualquier modo que se encuentre el servidor de base de datos puede pasar a fuera de línea, al momento de realizar el cambio el servidor solicita la confirmación de que se va a dar de baja, si es confirmado el servidor inicia una petición de *checkpoint* con la finalidad de que toda la información que esté en memoria sea descargada a disco, y manda una señal de desconexión a todas las sesiones adjuntas a la memoria compartida, si alguna sesión no recibe señal alguna ó no es capaz de cumplir automáticamente un lapso de 10 segundos, el servidor termina las sesiones.

La forma de realizar el cambio desde la vía de comandos es la siguiente:

*onmode -k ó onmode -ky*

Y utilizando el *onmonitor* se realiza la siguiente secuencia de opciones:

Se selecciona *Mode* → *Take Offline*.





```

    ) | tee -a "$Archsal"
  else if onmode -uky
  then (
      date; onstat -;
      echo EL SERVIDOR ${INFORMIXSERVER} HA SIDO DADO DE BAJA.;
      sleep 10;
      ) | tee -a "$Archsal"
  else (
      date; onstat -;
      echo "EL PROCESO DE BAJA FRACASO."
      ) | tee -a "$Archsal"
  fi
fi
# trap 1 2 3
# exit 255
::
*) echo "¡¡¡Por que se arrepiñto? ..."
echo "¡¡¡BYE!!"
::
esac
::
# Para restablecer la operacion de las senales
#trap 1 2 3
#exit 255
}
# Fin del Archivo

```

Programa Código shell que permite dar de baja el Servidor de BD

Es importante mencionar que el manejo de estatus es muy importante ya que permite que el administrador tenga un control sobre el servidor de base de datos pero para el caso de la nómina del sistema SIAPSEP requiere una alta disponibilidad pero hay ocasiones en que es necesario realizar algún mantenimiento al servidor de BD ó al S.O esto implica darlo de baja. Esto quiere decir que los administradores tanto del S.O. como del servidor de BD deben saber en que tiempo no hay mucha actividad por parte de los usuarios para poder realizar el mantenimiento de ahí derivada la importancia de saber cuando realizar un cambio de estatus. A continuación se detalla el proceso de inicialización del servidor de BD, cada una de las partes que da de alta dependiendo el tipo de inicialización ya sea de memoria compartida o disco.

#### IV.1.2 Inicialización del Servidor de BD

Al momento de inicializar el Servidor de Base de datos se distinguen dos porciones de inicialización, la memoria compartida y espacio en disco.

Cuando se realiza la inicialización de la memoria, se dan de alta las tablas internas, buffers y el área de comunicación de la memoria compartida, esto ocurre cada vez que se inicia el servidor de BD.

La inicialización de espacio en disco depende del archivo de configuración y el chunk primario denominado root, el cual es creado cuando el servidor es puesto en línea por primera vez durante una recuperación *en frío*, este cambio implica que se destruyan todos los datos anteriores en caso de que hubiera información en el servidor de Base de Datos.

TESIS CON  
FALLA DE ORIGEN

**Nota:** En frío se refiere a que el servidor de base de datos se encuentra en modo fuera de línea al momento de recuperar datos.

La siguiente tabla muestra los procesos mediante los cuales es inicializado el servidor de base de datos en memoria compartida y disco:

<b>Inicialización de la Memoria Compartida</b>	<b>Inicialización de disco</b>
Procesos del Archivo de Configuración	Procesos del Archivo de Configuración
Creación de segmentos de la memoria compartida	Creación de segmentos de la memoria compartida
Inicialización de las estructuras de la memoria compartida	Inicialización de las estructuras de la memoria compartida
Inicialización de todos los procesadores virtuales requeridos	Inicialización de todos los procesadores virtuales requeridos
Realización de conversiones necesarias	
Inicialización de una recuperación rápida	
Iniciar un checkpoint	Iniciar un checkpoint
Cambios de configuración	
Actualizar el archivo <code>oncfg_servername.servernum</code>	Actualizar el archivo <code>oncfg_servername.servernum</code>
Cambio a modo Quiescent	Cambio a modo Quiescent
Borrar <code>tblspaces</code> temporales	
Asignar <code>forcer-residency</code> si responde	Asignar <code>forcer-residency</code> si se especifica
Cambio a modo on-line para regresar el control a los usuarios	Cambio a modo on-line para regresar el control a los usuarios
Si las tablas SMI no son actuales, actualiza las tablas	Crea tablas SMI

Tabla. Inicialización de la memoria compartida y disco

El siguiente término es muy usado por un administrador SMI (System-Monitoring Interface) y es una interface de monitoreo de sistema compuesta de tablas y pseudo - tablas que el servidor de base de datos Informix mantiene de manera automática y son diferentes a las tablas de usuario, teniendo una ubicación distinta en disco, así mismo la información de las tablas es cargada a memoria dependiendo de la demanda de información a la base de datos. Cuando se realiza un query referenciando a las tablas SMI se lee la información de las estructuras de la memoria compartida. La información contenida en estas tablas es referente al estado actual del servidor por lo que continuamente cambian.

#### *Información proporcionada por las tablas SMI*

- Auditorías
- Espacio de disco usado
- Profile de usuarios
- Estatus de las Bases de datos
- Tablas
- Chunks
- Acceso de Entrada - Salida a los Chunks

TESIS CON  
FALLA DE ORIGEN

- Dbspaces
- Locks
- Extents
- Estadísticas de sentencias SQL.
- Uso de los procesadores virtuales CPU
- Profile del Sistema

Es importante identificar cuando se realiza un cambio en la configuración, para saber la forma en que puede afectar al momento de que se lleva a cabo la inicialización del Servidor de Base de Datos.

Cuando se realiza la inicialización del servidor, se considera la siguiente información para en caso de haber cambios validarlos y posteriormente aplicarlos en caso de que no se hayan encontrado errores:

#### *Archivo de Configuración*

- 1.- Si la variable de ambiente ONCONFIG fue asignada el servidor lee el archivo en *\$INFORMIXDIR/etc/\$ONCONFIG*, si este valor fue asignado pero no se encuentra el archivo de configuración el servidor retorna un mensaje de error.
- 2.- Si la variable de ambiente no es asignada, el servidor lee los valores del archivo onconfig en la siguiente trayectoria: *\$INFORMIXDIR/etc/onconfig*

#### *Creación de la Porciones de Memoria Compartida*

El servidor de BD utiliza los valores de configuración para calcular el tamaño requerido para determinar la memoria compartida.

Para crear memoria compartida, el servidor toma espacio del sistema operativo para separarlo en tres tipos de memoria:

- Porción residente usada para almacenamiento de tablas y de datos.
- Porción virtual usada de manera más interna para sesiones de usuarios.
- Y la porción de la comunicación IPC para la conexión a la memoria compartida.

#### *Inicialización de las Estructuras de Memoria Compartida*

Después de que el servidor de BD define el espacio de la memoria compartida, inicializa las estructuras de la memoria compartida, registrando el tamaño en el encabezado de la memoria compartida.

#### *Inicialización de Espacio en Disco*

Después de que las estructuras de la memoria compartida son inicializadas el servidor de Base de Datos da inicio al espacio en disco, con las páginas reservadas encontradas en el dbspace de root.





### *Inicialización de todos los procesadores virtuales Requeridos*

Inicializa todos los procesadores virtuales necesarios partiendo de lo definido en el archivo de configuración ONCONFIG en el parámetro NETTYPE parámetro que influye en el número y tipo de procesadores inicializados para realizar las conexiones.

### *Realizar Conversiones Necesarias*

El servidor realiza una revisión de los archivos internos para ver las versiones de los archivos, esto aplica cuando se realizan migraciones de versiones.

### *Inicialización de Recuperación Rápida (Fast Recovery)*

El servidor verifica si es necesario efectuar la recuperación rápida durante la inicialización del espacio en disco. Esto es debido a que pudiera ser que la última vez que se dio de baja el servidor no se concluyeron algunas actividades y por lo tanto es necesario realizar rollback.

### *Inicio de un Checkpoint*

Después de realizar una recuperación rápida el servidor realiza un checkpoint y esto ocurre cuando el estatus del servidor cambia de modo Quiescent a modo On-line. Toda la información contenida en los logs lógicos es descarga a disco.

*Nota:* *checkpoint* se refiere a un punto en que se hace la revisión de algo, para verificar que no haya inconsistencias.

### *Cambios en el Documento de Configuración*

El servidor compara los valores actuales del archivo de configuración con los valores contenidos en el dbspace de root en las páginas reservadas definidas por PAGE\_CONFIG. Cuando existen diferencias el servidor envía en el archivo de bitácora los mensajes de los cambios encontrados.

### *Crear el archivo oncfg\_servername.servernum*

El servidor crea este archivo y actualiza cada adición o borrado de un dbspace, blobspace, archivo logical-log ó chunk, este archivo no requiere de manipulación y es utilizado cuando se realiza una restauración completa.

### *Borrado de Tblspaces temporales*

El servidor de BD selecciona todos los dbspaces y los pasa a un tblspaces durante una inicialización, y el siguiente proceso borra todos los tblspaces temporales.

### *Regresar el control al Usuario*

Después de que los pasos previos son completados el servidor de Base de datos escribe en los mensajes "inicialización completa". En este punto el usuario puede tener acceso al servidor, cualquier mensaje de error puede ser enviado de tres formas:



- En la línea de comandos.
- En el archivo de mensajes especificado por la variable de ambiente MSGPATH.
- Y en caso de que se utilice onmonitor en la pantalla aparece el error.

#### *Prepara las tablas SMI (System-Monitoring Interface)*

Al regresar el control al usuario ahí no finaliza el proceso de inicialización del servidor de base de datos, ya que tiene que revisar la interface de monitoreo del sistema, si estas tablas no son las actuales el servidor las actualiza, en el caso de la inicialización si no existen estas tablas el servidor las crea, conformando la base de datos sysmaster, después de haber sido satisfactoria la creación, el servidor está listo para poder ser utilizado.

Como mencionamos es de suma importancia la inicialización ya que en caso de algún error, proporciona la información necesaria al administrador para poder realizar las correcciones con las herramientas que proporciona el mismo, dar mantenimiento al servidor de base de datos ayuda a evitar algunos errores que pudieran presentarse, apoyados por herramientas que el sistema operativo tiene y es en base a las necesidades del administrador, lo que va a determinar la selección de alguna de ellas. Haciendo mención al sistema SIAPSEP su base de datos debe estar bien identificada y resguardada y el tener en cuenta el proceso de inicialización se tendrá un tiempo de respuesta en caso de algún error además de tener el conocimiento de este proceso es importante monitorear constantemente la actividad del manejador de base de datos y cada servidor proporciona sus propias herramientas. A continuación mencionaremos herramientas del servidor de base de datos Informix.

## IV.2 HERRAMIENTAS DE CONSULTA

Estas herramientas permiten al administrador del Servidor de BD monitorear las actividades realizadas e identificar de manera anticipada o en su caso corregir problemas registrados en el servidor, las herramientas de consulta son las siguientes:

- Onstat
- Oncheck
- Onlog
- Archivo de bitácora (informix.log)

### **Onstat**

Esta herramienta lee estructuras de memoria compartida y provee estadísticas acerca del servidor de base de datos. Esta utilería no coloca ningún candado sobre la memoria compartida por lo que no afecta el rendimiento del sistema, además de que proporciona información completa sobre cualquier parte del servidor de base de datos.

TESIS CON  
FALLA DE ORIGEN



Una función muy usual de la salida de onstat es una cabecera que muestra el estatus del servidor de base de datos como sigue:

Blocked: *reason*

La variable *reason* puede tomar uno de los siguientes valores mostrados en la tabla 4.1

Reason	Descripción
CKPT	Checkpoint
LONGTX	Transacción larga
ARCHIVE	Continuando con el respaldo
MEDIA FAILURE	Falla en la unidad de media
HANG_SYSTEM	Falla en el servidor de base de datos
DBS_DROP	Borrando un dbspace
DDR	Replicación discreta de alta disponibilidad de datos
LBU	Logs llenos hasta la marca máxima

Tabla 4.1 Valores de la variable *reason*

Todas las salidas de onstat incluyen una cabecera. El signo de – después de onstat despliega solo la cabecera de la instrucción la cual es muy usual para checar el modo del servidor de base de datos. La cabecera considera los siguientes datos:

*Version--Mode (Type)--(Checkpoint)--Up Uptime--Sh\_mem Kbytes*

donde:

Version	Es el número de versión y el nombre del producto
Mode	Es el actual modo de operación
(Type)	Si el servidor de base de datos utiliza replicación de datos en alta disponibilidad, indica si el tipo de replicación es primario o secundario.
Chkptn	Es la bandera de checkpoint
Uptime	Indica cuanto tiempo el servidor de base de datos ha estado corriendo
Sh_mem	Es el tamaño de la memoria compartida dado en Kilobytes

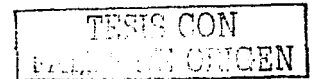
Por Ejemplo:

**Dynamic Server Version 9.20.UC1--On-Line--Up 15:11:41--9216 Kbytes**

*Opciones de la Utilería Onstat*

onstat - -

La opción - - despliega un listado de todas las opciones de onstat y sus funciones. Esta opción es la única que no se puede combinar con ninguna otra bandera.



**onstat -a**

La opción **-a** es interpretada como **onstat -cuskbtdlp**, y la salida es desplegada en ese orden. Esta opción presenta una información general de todo el servidor de base de datos.

**onstat -b**

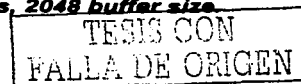
La opción **-b** despliega información acerca de los buffers actualmente en uso, incluyendo el número total de páginas residentes en el pool del buffer, esta salida muestra la siguiente información:

<b>address</b>	Es la dirección del buffer en la tabla de buffers
<b>userthread</b>	Es la dirección de los hilos de usuario mas recientes para acceder la tabla de buffer. Muchos hilos de usuarios pueden estar leyendo el buffer concurrentemente.
<b>flags</b>	Utiliza los siguientes bits de bandera para describir el buffer:
	0x01            Datos Modificados
	0x02            Datos
	0x04            LRU
	0x08            Error
<b>pagenum</b>	Es el número de página física en el disco
<b>memaddr</b>	Es la dirección de memoria del buffer
<b>nslots</b>	Este campo indica el número de registros que están almacenados en una página.
<b>pgflgs</b>	Utiliza los siguientes valores, solos o en combinación para describir el tipo de página.
	1. Páginas de datos
	2. Páginas de Tblspace
	4. Lista de páginas libres
	8. Lista de páginas libres en un chunk.
	9. Páginas de datos restantes
<b>waitlist</b>	Es la dirección del primer hilo de usuario que esta esperando para acceder al buffer.

El máximo número de buffers disponibles está especificado como **BUFFERS** en el archivo **ONCONFIG**.

La opción **-b** y **B** también proveen un resumen de información acerca del número de buffers modificados, el número total de páginas residentes en el pool del buffer y el número total de buffers disponibles, así como el tamaño de cada buffer dado en bytes (tamaño en páginas), como lo muestra la siguiente línea:

**123 modified, 23 resident, 2000 total, 2048 hash buckets, 2048 buffer size.**



La siguiente programación Shell ejecuta esta opción del onstat:

```
#!/bin/sh
#####
#
# // IDStools // For Informix Dynamic Server //
#
# Copyright (C) 1999, 2000 Tim Schaefer All Rights Reserved
#
# Program: IDStools_idstools
#
# version 1
#
onstat -b | grep "buffer size" | awk '
BEGIN {
    tok_cnt=0 ;
    buffer_size="" ;
}
{
    tok_cnt = split ( $0 , tok_arr , " " ) ;
    for ( i=0 ; i <= tok_cnt ; i++ )
    {
        # { printf ("%d %s\n" , i , tok_arr[i] ) }
    }
}
buffer_size=tok_arr[i-3]
{ printf ("%s\n" , buffer_size ) }
}'

#
# version 2
#
BUFF_SIZE=onstat -b | grep "buffer size" | awk '
BEGIN {
    tok_cnt=0 ;
    buffer_size="" ;
}
{
    tok_cnt = split ( $0 , tok_arr , " " ) ;
    for ( i=0 ; i <= tok_cnt ; i++ )
    {
        if ( tok_arr[i] == "buffer" )
        {
            buffer_size=tok_arr[i-1]
            { printf ("%s" , buffer_size ) }
        }
    }
}
}'
page_size=$BUFF_SIZE
echo $BUFF_SIZE
```

TESIS CON  
FALLA DE ORIGEN



**onstat -c**

Esta opción despliega el contenido del archivo **ONCONFIG**, el servidor primero verifica si se tiene asignado un valor para la variable de ambiente **ONCONFIG**, en caso de no existir toma el archivo por default de la trayectoria **\$INFORMIXDIR/etc/onconfig**.

**onstat -d**

Esta opción despliega información de chunks en cada espacio de almacenamiento así como el tamaño de **sbspaces**, áreas de datos de usuarios, y áreas de megadatos.

Para obtener estadísticas actuales para un chunk de un **bospace**, se utiliza el comando **onstat**.

**-d update**. Esta utilidad actualiza la memoria compartida, con el número actual de páginas libres para cada **dbspace**. El servidor desplegaría el siguiente mensaje:

Waiting for server to update BLOB chunk statistics ...

**onstat -D**

Esta opción despliega información de las páginas leídas y las páginas escritas. Se pueden interpretar los dos campos que son únicos para la salida de la opción **-D**, como sigue:

<b>pageRd</b>	Es el número de páginas leídas
<b>pageWr</b>	Es el número de páginas escritas

**onstat -f**

Lista los **dbspaces** con las recientes fechas de actualización. Cuando se ejecuta la opción **onstat -f**, el servidor de Base de Datos despliega una de las siguientes tres salidas:

- **Dataskip** is off para todos los **dbspaces**
- **Dataskip** is on para todos los **dbspaces**
- **Dataskip** is on para los siguientes **dbspaces**:  
**dbspace1 dbspace2 ...**

**onstat -F**

Esta opción despliega una cuenta por cada tipo de escritura a disco de la siguiente forma:

<b>Fg Write</b>	Es el número de tiempos de las páginas de primer plano ocurridas.
<b>LRU Write</b>	Es la cantidad de tiempo en que transcurre una escritura LRU.
<b>Chunk writes</b>	Es el tiempo que transcurre en la escritura a chunks.
<b>Address</b>	Es la dirección de la estructura de usuario asignadas a los hilos de las páginas limpias.

<b>Flusher State</b>	Es el número de páginas limpias. Para identificar el estado de las páginas se utilizan las siguientes opciones:
----------------------	--



C Escritura a chunk  
 E Salida  
 I Limpiador inactivo  
 L Cola de LRU  
 Exit El estatus de exit indica que el servidor de BD esta dado de baja.

**data** Proporciona información adicional sobre el estado de algún campo.

**onstat -g** (Opción de monitoreo)

Esta opción es para realizar el soporte y rastreo general del servidor de BD. La siguiente tabla 4.2 muestra las subopciones:

Opciones de Onstat -d	Descripción de la función
-g act	Imprime hilos activos
-g afr pool name session id	Imprime la asignación de la memoria fragmentada o el pool de la memoria compartida para una sesión específica.
-g all	Imprime la información de todos los multihilos en ejecución.
-g ath	Imprime todos los hilos. El hilo sqlmain representa la sesión del cliente.
-g cat [modificador]	Imprime la información del catálogo global de replicación. El catálogo global contiene un sumario de información acerca de los servidores definidos replicados.
-g cac agg	Imprime la definiciones agregadas por usuarios que están actualmente en el cache.
-g cac stmt	Imprime el contenido de las funciones SQL contenidas en cache.
-g con	Imprime las condiciones en espera.
-g ddr	Imprime el estatus del log de replicación de BD
-g dfm	Imprime la información del flujo de datos entre coservidores .
-g dict table	Sin cualquier parámetro, imprime una línea de información para cada tabla en la memoria cache en el diccionario de la memoria compartida.
-g dis	Imprime una lista de todos los servidores de base de datos y sus estatus e información de cada uno de ellos como son INFORMIXDIR, el archivo sqlhosts, ONCONFIG y el hostname.
-g dll	Imprime una lista de todas las librerías que han sido cargadas.
-g dri	Despliega información sobre la replicación de datos.
-g dsc	Imprime información de la distribución de datos en cache.
-g dss [modificador]	Imprime información estadística detallada de la actividad individual de datos sincrónica y acerca de tipos de datos definidos por el usuario.
-g dtc	Imprime estadísticas acerca del borrado de registros de tablas.
-g ffr pool name session id	Despliega los fragmentos libres para un pool de la memoria compartida.
-g glo	Imprime información de multihilación global. Esta información incluye información acerca de los procesadores virtuales, el número total de sesiones y otros contadores adicionales.
-g grp [modificador]	Imprime información estadística de la replicación.
-g imc	Imprime información del número máximo de conexiones para un servidor de BD.
-g ioa	Imprime información combinada a partir de las siguientes opciones: -g ioq y -g iov.
-g iof	Despliega información de E/S asincrónica para un chunk o un archivo, esta opción es similar a -D excepto que no muestra información sobre chunks, temporales ni archivos de ordenamiento de trabajo.
-g iog	Imprime información global de los AIO's.
-g ioq queue name	Imprime información pendiente de operaciones de E/S.
-g iov	Imprime estadísticas asincrónicas de E/S para los procesadores virtuales.
-g lmx	Despliega todos los bloqueos anulados.
-g mem pool name session id	Despliega estadísticas del pool de la memoria, nombre del pool tipo de segmento de memoria compartida, dirección, tamaño total, el número de bytes de memoria libre y el número de segmentos libres asignados en el pool

TESIS CON  
 FALLA DE ORIGEN

-g mgm	Despliega información del recurso del manejo de los permisos a la memoria.
-g nif [modificador]	Despliega estadísticas de las interfaces de red.
-g nsc client id	Despliega el estatus de la memoria compartida por identificador de cliente.
-g seg	Despliega estadísticas de los segmentos de memoria compartida.
-g sle	Imprime todos los hilos que no están inactivos.
-g wal	Despliega todo los hilos que están en espera.

Tabla 4.2 Opciones de monitoreo del servidor de BD Informix

La siguiente programación shell utiliza la utilidad `onstat` con la opción `g` y las subopción `sess`, para eliminar sesiones en unix mediante la herramienta `kill`.

```
#!/bin/sh
#####
#
# // IDStools // For Informix Dynamic Server //
#
# Copyright (C) 1999, 2000 Tim Schaefer All Rights Reserved
#
# Program: IDSkill_pid
#####
# DESCRIPTION: This program kills an informix session based on a process id
# passed to it. The user of this program can optionally call this
# program with a process-id or will be prompted for a process-id
# if one is not supplied at runtime. The user is prompted a second
# time to make sure this is the process they want killed.
#
# If the user presses return at the "Enter PID:" prompt, a
# listing of available PIDs will be scrolled to the screen, and
# another opportunity will be given to enter the desired PID.
#
# If the user presses return at the "Kill this process? Y/N "
# prompt, no kill will be performed. The default is to abort the
# kill process.
#
# If the user enters a "Y" or a "y" at the "Kill this process? Y/N"
# prompt, an onmode -z action will be performed.
#
#####
# substitute '.x' with a .profile or other environment file for the informix
# environment you wish to work with:
# .x >/dev/null 2>/dev/null
#
# or
#
# add explicit environment variables here:
#
# export INFORMIXDIR=
# export INFORMIXSERVER=
# export ONCONFIG=
# export PATH=$PATH:$INFORMIXDIR:$INFORMIXDIR/bin:
#
#####
# BUFF_SIZE='onstat -b | grep "buffer size" | awk ' { print $8 } ''
AWK=awk
FILE_PATH=/tmp
TMP=/tmp
ISQL=dbaccess
#####
prompt_for_pid()
{
    echo "Enter PID or q to quit or ENTER/RETURN key for list : \c"
    read pid
}
}
```





```

#####
show_all_sessions()
{
    onstat -g ses | pg -p "Press Return:"
}

#####
get_session_id()
{
    session_id=`onstat -g ses | grep ${pid} | awk ' { print $1 } '`
}

#####
show_session_detail()
{
    onstat -g ses $session_id | awk '
/session/|^S/ { print $0 }
/Sess /|^S/ { print $0 }
/Last parsed /|^S/ { print $0 }
'
}

#####
kill_informix_session()
{
    echo "Killing informix session: $session_id"
    #
    # use this if you must use su
    # /bin/su informix -c "/products/informix/bin/onmode -z $session_id"
    #
    onmode -z $session_id
    sleep 2
}

#####
kill_unix_process()
{
    echo "Killing unix pid: $pid"
    #
    # use this if you must use su
    #
    # /bin/su root -c "kill -9 $pid"
    #
    kill -9 $pid
}

#####
prompt_for_kill()
{
    echo "\nKill this process? Y/N \c"
    read yesno
    case $yesno in
    Y|y) kill_informix_session
        kill_unix_process
        ;;
    *) echo "No kill performed"
        ;;
    esac
}

#####
run_program()
{
    get_session_id
    show_session_detail
    prompt_for_kill
}
#####

```

TESIS CON  
FALLA DE ORIGEN



```

if [ $# -le 0 ] ; then
    prompt_for_pid
    while [ -z "$pid" ]
    do
        show_all_sessions
        prompt_for_pid
    done
    if [ $pid = 'q' ] ; then
        exit
    else
        run_program
    fi
else
    pid=$1
    while [ -z "$pid" ]
    do
        show_all_sessions
        prompt_for_pid
    done
    if [ $pid = 'q' ] ; then
        exit
    else
        run_program
    fi
fi

```

TESIS CON  
FALLA DE ORIGEN

#### onstat -i

Funciona de manera interactiva, para que la misma instrucción la ejecute varias veces en determinado intervalo de tiempo por default es cada 5 segundos.

#### onstat -k

Esta opción despliega información acerca de los bloqueos (locks) activos, mostrando lo siguiente:

address	Es la dirección del bloqueo en la tabla de bloqueos. Si un hilo de un usuario está en espera para un bloqueo en este campo al momento de desplegar aparece <i>wait</i>
wtlist	Si hay algún hilo de usuario en espera en esa línea aparecerá.
owner	Es la dirección de la memoria compartida del hilo que contiene el bloqueo.
lklist	Es el siguiente bloqueo de la lista.
type	Se utiliza el siguiente código:
	HDR Encabezado
	B Bytes
	S Compartido
	X Exclusivo
	I Intento
	U Actualización
	IX Intento-exclusivo
	IS Intento-compartido
	SIX Compartido Intento compartido
rowid	Es el identificador del número del registro.
key#/bstz	Es el número de la llave del índice.



**onstat -l**

Despliega información acerca de los logs lógicos y físicos con las siguientes características:

Buffer	Es el número de buffers del log lógico.
Bufused	Es el número de páginas del buffer del log lógico que se están usando.
Bufsize	Es el tamaño de cada buffer de log físico en páginas.
Numpages	Es el número de páginas escritas en el log físico.
Numwrits	Es el número de escrituras a disco.
Pages/io	Es el cálculo de numpages/numwrits
Phybegin	Es el número de páginas físicas donde está iniciando el log.
Physize	Es el tamaño del log físico en páginas.
Phypos	Es la posición actual en el log del último registro en el log.
Phyused	Es el número de páginas usadas en el log.
%used	Es el porcentaje de páginas usadas.
Flags	Proporciona el estatus de cada log como sigue: A Recientemente añadido ( y listo para usar) B Respaldado C Archivo de log lógico actual. D Marcado para borrar. F Libre, disponible para usar L El checkpoint más reciente registrado. U Usado.
Size	Es el tamaño del log en páginas
Used	Es el número de páginas usadas.
%used	Es el porcentaje de páginas usadas.
Total	Es el número total de logs lógicos.

**onstat -t y -T**

Despliega información de los tblspace.

**onstat -u**

Despliega información de los usuarios conectados al servidor de Base de Datos:

Address Es la dirección de la memoria compartida del hilo del usuario.

Códigos de bandera para la posición 1:

Flags	Proporciona el estatus de la sesión.
B	Esperando por un buffer
C	Esperando por un checkpoint
G	Esperando para la escritura del buffer del log lógico
L	Esperando para un bloqueo
T	Esperando para un transacción
Y	Esperando una condición

TESIS CON  
FALLA DE ORIGEN



**X** Esperando para realizar un rollback  
**DEFUNCT** El hilo del usuario a provocado una falla por lo que es suspendido para permitir que otros hilos continúen con su trabajo.

Códigos de bandera para la posición 2:

\* Transacción activa durante una falla de Entrada/Salida

Códigos de bandera para la posición 3:

A Un hilo de respaldo para un dbspace

Códigos de bandera para la posición 4:

P Hilo primario para la sesión.

Códigos de bandera para la posición 5:

R Leyendo

X Hilo en sección crítica.

Códigos de bandera para la posición 7:

B Limpieza de hilos

C Termina el hilo del usuario esperando a ser limpiado.

D Un proceso demonio de un hilo.

F Página limpia de un hilo.

M Hilo especial de onmonitor.

**sessid** Es el número de identificación de la sesión. Durante operaciones tales como ordenamiento paralelo y construcción de índices en paralelo, un usuario puede tener muchos hilos asociados pero la forma de saber a donde están asociados es mediante este campo.

**User** Es el nombre del login del usuario (derivado del sistema operativo).

**tty** Indica el tty que el usuario está utilizando (derivado del sistema operativo)

**wait** Indica si el hilo del usuario está en espera desplegando la dirección del recurso.

**tout** Es el número de segundos que está esperando actualmente.

**locks** Es el número de bloqueos que el hilo del usuario está ocupando.

**nreads** Es el número de lecturas de disco que el hilo del usuario está ejecutando.

**nrites** Es el número de llamadas a escrituras que el hilo del usuario está ejecutando. Todas las llamadas a escrituras son escrituras al buffer cache de la memoria compartida.

**onstat -x**

Despliega información sobre transacciones en el servidor de Base de Datos

**onstat -z**



Limpia las estadísticas del servidor de base de datos, incluyendo las estadísticas de replicación.

Esta es la herramienta de consulta que proporciona de manera completa información de cualquier índole referente al servidor de base de datos.

El siguiente código shell genera un bitácora de la utilidad onstat:

```
#!/bin/sh
#####
#
# // IDStools // For Informix Dynamic Server //
#
# Copyright (C) 1999, 2000 Tim Schaefer All Rights Reserved
#
# Program: IDSonstat_log
#####
echo "-----" > $ONCONFIG.onstat
echo "hostname '*** 'uname -a '*** 'date' " >> $ONCONFIG.onstat
echo "-----" >> $ONCONFIG.onstat
for option in -b -c -d -f -h -k -l -m -p -s -t -u -x -C -D -F -R -G -X
do
echo "----- onstat $option"
onstat $option
echo "-----"
done 1 >> $ONCONFIG.onstat 2 > &1
#####
for option in ath wai act rea sle sch lmx wmx con stk glo mem seg iov lof log iob ppf tpf ntu nlt ntm ntd nss nsc nsd sts dlc
opn qst wst ses sqf stq dri pos mgm lap ddr dmp src spi
do
echo "----- onstat -g $option"
onstat -g $option
echo "-----"
done 1 >> $ONCONFIG.onstat 2 > &1
#####
```

La siguiente utilidad sirve como herramienta de consulta pero también como herramienta de mantenimiento.

### Oncheck

Dependiendo de las opciones que sean seleccionadas, oncheck puede realizar las siguientes funciones:

- Verificar las estructuras del disco para inconsistencias
- Desplegar información acerca de la estructura del disco

Con las opciones siguientes:

```
oncheck -cd
oncheck -cD
oncheck -ci
oncheck -cl
```

TESIS CON  
FALLA DE ORIGEN



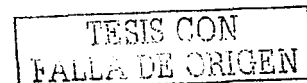
Si no se desea utilizar esta opción, oncheck envía el prompt cuando encuentra inconsistencias, permite que se repare de manera independiente. Si la opción `-n` se especifica, oncheck no envía el prompt, ya que esta opción indica a oncheck a no realizar reparaciones.

OPCIONES	PROPOSITO
<code>-cc</code>	Verifica las tablas de catálogo del sistema, para la base de datos especificada
<code>-cd</code>	Lee todas las páginas excepto los objetos largos simples del tblspace para la base de datos especificada, tabla o fragmento y chequea cada página para inconsistencias
<code>-cD</code>	Igual que la opción anterior, pero además lee la cabecera de cada blobpage y lo chequea para consistencias
<code>-ce</code>	Checa cada chunk libre y el correspondiente espacio libre de cada tblspace exento.
<code>-ci</code>	Checa el ordenamiento de los valores de las llaves y la consistencia de las ligas de los nodos verticales y horizontales para todos los índices asociados con la tabla especificada. También chequea los índices que utiliza el método de acceso definido por un usuario
<code>-cr</code>	Checa cada una de las páginas reservadas del dbspace raíz para algunas condiciones.
<code>-cR</code>	Igual que el anterior, además de las páginas de log físicas y páginas de log lógicas.
<code>-n</code>	Indica que los índices no reparados no deben permanecer, siempre que algún error sea detectado
<code>-pB</code>	Despliega estadísticas que describen el promedio de llenado de un blobpage o blobpage de una tabla especificada.
<code>-pc</code>	Igual que la opción <code>-cc</code> , pero además despliega información del catálogo del sistema, como chequea las tablas de catálogo del sistema
<code>-pe</code>	Lo mismo que <code>-ce</code> pero despliega información extra del chunk y del tblspace, lista el espacio libre de cada chunk, y el correspondiente espacio libre en cada tblspace
<code>-pk</code>	Muestra lo mismo de <code>-ci</code> , pero además despliega los valores de las llaves para todos los índices en la tabla especificada, así como los chequea.

Tabla. Descripción de las opciones de la utilidad oncheck

Estas son las opciones más importantes de esta utilidad.

### Onlog



Esta utilidad despliega el contenido de los archivos de log lógicos. La salida de onlog es muy usual en situaciones de depuración, cuando se desea "identificar" una transacción específica, o cambios que se han venido realizando en el servidor de BD.

Cualquier usuario puede utilizar todas las opciones de onlog, excepto la opción `-l`, en este caso solo puede ser el usuario root o informix.

Si el servidor de base de datos está en línea (on-line), cuando se ejecuta esta utilidad sólo los archivos en disco pueden ser leídos, pero si el servidor de base de datos se encuentra en modo inactivo o en modo monousuario, onlog solo puede leer los registros lógicos de log almacenados en los buffers de log lógicos en memoria compartida.

Cuando el servidor de base de datos lee los archivos de log lógicos con estatus U, del disco, mientras se encuentra en modo online, niega todos los accesos hacia los archivos de log lógicos, de manera efectiva detiene las actividades en la base de datos, para todas



las sesiones. Por esta razón, informix recomienda esperar a que los archivos hayan sido respaldados, y luego leer el contenido de los archivos de log lógicos desde el respaldo.

### Sintaxis

Comando	Opciones
onlog -q	Suprime la cabecera inicial y la única línea de cabecera que aparece cada 18 registros por default

Por default onlog despliega las cabeceras de los archivos de log lógicos, los cuales describen el número de transacción y el tipo de registro. El tipo de registro identifica el tipo de operación realizada.

Así como se puede desplegar la cabecera para cada archivo de log lógico, es posible especificar la salida basada en el siguiente criterio.

- Registros asociados con una tabla específica.
- Registros iniciados con un usuario específico.
- Registros asociados con una transacción específica.

La utilidad onlog utiliza las trayectorias almacenadas en las páginas reservadas para el dbospace de root, para alojar los archivos de log lógicos.

Los filtros para la recuperación de información específica en los logs son los siguientes:

Opciones	Descripción
-b	Esta opción despliega los registros asociados con un blobspace ó blobpages.
-d device	El dispositivo es la ruta en donde se respaldan los logs lógicos.
-n uniqid	Direcciona a onlog para que realice la lectura de los registros contenidos en los archivos log especificados en unqid.

La siguiente pantalla muestra la salida de esta utilidad:

```

48254      188  BEEPREP  4          0  48032          0  1
30000000  30000041  30000000  00344038  .....E
30027e79  00300002  61887669  73770000  .....GRISW..
30000000  00300000  00000000  00000000  .....T.....
30000015  00300000  00000000  00300000  ..R.....
80000000  00300000  00000000  00300000  .....
30000000  00300000  00000000  00300000  .....
30000000  00300000  00000000  00300000  .....
30000000  00300000  00000000  00300000  .....
00000000  00300000  00000000  00300000  .....
30000001  00300000  00000000  00300000  .....Judica_s
30000000  00300001  68755469  74895773  .....
87500000  73476374  63760000  ..cc..root  cp..

```

Desplegado de la utilidad onlog

TESIS CON  
FALLA DE ORIGEN

En la parte inicial de la pantalla viene el identificador del inicio de una transacción global definido como GTRID, el cual indica que tipo de proceso se va a realizar en este caso **BEGPREP** no indica que se está iniciando la preparación de una instrucción con el identificador 188 en el cual el ejecutor de esta instrucción es **chrisw**, las columnas indican las direcciones en el log lógico donde son registradas las transacciones.

El siguiente ejemplo muestra la utilidad que tiene la utilidad onlog:

Este ejemplo muestra un proceso pequeño el cual involucra una recuperación manual.

La siguiente instrucción fue ejecutada por el usuario a713ap00. La cual regresa un error -698.

```
$dbaccess
create database nomina_des with log;
create table empleado ( rfc char(13) );
close database;
create database nomina_des2@sep_tcp with log;
create table empleado_curp (curp char(18));
close database;
database nomina_des;
begin work;
insert into empleado values ("SOMA750201AA1");
insert into empleado_curp@sep_tcp values ("SOMA750201HDF09IO2W")
commit work;
## Código de error regresado -698
```

El siguiente desplegado es tomado de los logs lógicos de las instrucciones realizadas en el servidor actual:

addr	len	type	xtid	id	addr	user
19018	15	LOGCKPOINT	3	19018	0	
19018	7	LOGBEGIN	1	19018	08/27/91 12:55:57	3482 a713ap00
1902c	12	LOGINSERT	3	19018	1620018 102	4
1904c	40	LOGCKPOINT	3	19018	:	:
		begin	xtid	id	addr	user
19018	7	LOGBEGIN	2	1902c	16063	1
19069	15	LOGCOMMIT	3	19018	08/27/91 11:01:38	
19018	15	LOGENDTRANS	2	19043	890543	

Y el siguiente fragmento del archivo de log lógico es tomado del servidor de base de datos sep\_tcp:

addr	len	type	xtid	id	addr	user
19018	7	LOGBEGIN	2	19018	08/27/91 12:57:07	3483 a713ap00
1902c	12	LOGINSERT	2	19018	1620018 102	4
1904c	40	LOGCKPOINT	2	19018	en	1
17018	15	LOGCURTX	2	1902c		
17028	12	LOGCLR	2	1902c		
17034	15	LOGROLLBACK	2	17018	08/27/91 11:01:22	
17044	40	LOGCKPOINT	2	17018	:	:
		begin	xtid	id	addr	user
19018	7	LOGBEGIN	2	17034	-----	
18018	15	LOGENDTRANS	2	17034	890543	

TESIS CON  
FALLA DE ORIGEN



Primero se debe identificar la transacción en el actual servidor de BD, con la transacción en el servidor `sep_tcp`. Los registros del log `BEGPREP` y `PREPARE` cada uno contiene su `GTRID`, el cual es un número de identificación único denominado *identificador de transacciones globales*, que se obtiene cuando se ejecuta la instrucción `onlog -l` delimitándose al área del usuario que se quiere verificar.

Para realizar la corrección del error enviado por el servidor de base de datos se realiza lo siguiente:

1. Identificar los registros en donde hubo actualización.
2. Identificar que tipo (`insert`, `update`, `delete`) usando `onlog` y la tabla de tipos de registros.
3. Se utiliza la salida de la instrucción `onlog -l` por cada registro para obtener el *identificador xid* el cual es el número de *identificador de transacciones locales*, que se obtiene del registro `HEURTX`.
4. Posteriormente se debe de mapear el número de `tblspace` para el nombre de la tabla, comparando el número del `tblspace` en la columna `partnum` de la tabla `systables`.
5. Posteriormente se identifica el número de error y las posibles causas para realizar la corrección.

#### *Finderr*

Esta utilidad proporciona información de los códigos de error, y las soluciones para su corrección.

#### **Bitácora de Informix (Archivo)**

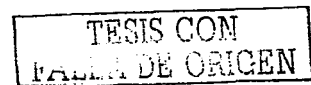
Este archivo se encuentra en la trayectoria que define el administrador en la variable de configuración `MSGPATH`.

En este archivo el servidor de BD Informix escribe información referente a estatus y errores registrados en algún proceso determinado. Así mismo ayuda al administrador a tomar medidas sobre lo que se debe hacer para corregir dichas inconsistencias.

Para visualizar los mensajes de este archivo es mediante la utilidad `tail` de unix ó con la utilidad `onstat` y el parámetro `-m`.

La categoría de mensajes que son registrados en este archivo son los siguientes:

- Información de rutina como por ejemplo `checkpoint`
- Mensajes de fallas
- Acciones administrativas necesarias
- Detección de errores fatales



A continuación se presentan unos ejemplos de errores que pueden ocurrir:

Error	Causa	Acción
Aborting Long Transaction: <i>tx Oxn</i> .	Indica que el espacio especificado para una transacción fue excedido. parámetro (LTXHWM), por lo que se está realizando un roll back.	Acción adicional no necesaria.
Affinitied VP <i>mm</i> to phys proc <i>nn</i> .	EL servidor de BD concluyó satisfactoriamente los procesadores virtuales.	Nada requerido
Building 'sysmaster' database requires <i>-mm</i> pages of logical log. Currently there are <i>nn</i> pages available. Prepare to back up your logs soon.	No se tiene actualmente la cantidad libre de espacio en los logs para completar la construcción de la base de datos sysmaster.	Respaldar los logs
Cannot Allocate Physical-log File, <i>mm</i> wanted, <i>nn</i> available.	El servidor de base de datos intento inicializar la memoria compartida con un tamaño de log físico que excede la cantidad de espacio contiguo disponible en el dbspace. Ambas cantidades de espacio, la deseada y la disponible estan expresadas en kilobytes.	Se debe reducir el tamaño del log físico ó cambiar la localización del log físico a un dbspace que contenga una cantidad adecuada de espacio contiguo para alojar estos logs físicos.
The chunk <i>pathname</i> must have <i>owner-ID</i> and <i>group-ID</i> set to informix.	El chunk no tiene asignado el dueño y el identificador de grupo correcto.	Asegurarse de que el dispositivo en el cual el chunk esta localizado tiene el dueño correcto. En Unix, ambos, el dueño y el grupo deben ser informix.

### Explain

Esta opción se utiliza para desplegar los planes de ejecución del optimizador, estimando el número de registros recuperados y el costo de la sentencia sql en tiempo.

Para activar esta opción se ejecuta de la siguiente manera:

```
$dbaccess
set explain on
```

Y para desactivarlo:

```
Set explain off
```

TESIS CON  
FALLA DE ORIGEN



Esta herramienta proporciona información sobre medidas de trabajo que afectan directamente el rendimiento. Los resultados los escribe en un archivo de texto en el directorio en donde se invocó esta opción. En caso de que el optimizador determine que debe realizar una acción correctiva o modificación al lenguaje SQL proporciona alternativas.

El optimizador es el que genera un plan de ejecución para recuperar la información solicitada por algún usuario, por lo que debe evaluar los diferentes caminos para poder llegar a una respuesta de la manera más óptima, referenciado por una serie de estadísticas que tienen información sobre los objetos como cantidad de registros, llaves primarias y secundarias, índices entre otros generados por la utilidad `update statistics` que se mencionará posteriormente.

Esta utilidad ayuda tanto a administradores como a programadores en sentencias SQL, para obtener el resultado más óptimo y menos consumo de recursos.

### IV.3 HERRAMIENTAS DE MODIFICACION

Estas herramientas sirven para realizar modificaciones al servidor de Base de Datos derivados del análisis y estadísticas reportadas por las herramientas de consulta. La utilerías que consideraremos son las siguientes:

- Update Statistics
- Oncheck
- Onmode
- Onparams
- Onspaces
- Oninit
- Onmonitor

#### Update Statistics

Esta utilidad permite tener información actualizada sobre la tabla ó tablas que están en constante movimiento. Actualiza tablas de sistemas de catálogos para optimizar consultas de ejecución, básicamente para que el acceso y que el tiempo de respuesta sea rápido, de igual forma aplica para los procedimientos almacenados.

Internamente el manejador cuando se hace uso de esta utilidad recalcula la información en los catálogos `systables`, `syscolumns`, `sysindexes` y `sysdistrib`, con la finalidad de determinar la mejor ruta de ejecución de consultas a bases de datos. Esta utilidad no se ejecuta de manera automática por lo que el administrador debe determinar el tiempo en que debe de realizar un `update statistics`. De ahí parte importante del administrador realizar las consultas a la base de datos para ver que tablas u objetos de las bases de datos tienen mayor demanda.



Para el caso de los procedimientos almacenados (store procedure) el catálogo de estadísticas es sysprocplan, al momento de realizar update statistics implica que cada que se ejecuta un procedimiento, se reoptimiza el plan de ejecución, dado que cualquier objeto puede cambiar y así mismo el plan de ejecución.

La actualización de estadísticas puede ser para una tabla o procedimiento específico o para toda la base completa dependiendo la instrucción a ejecutar.

El administrador debe de considerar los siguientes puntos para que sea aplicado un update statistics:

- Cuando una tabla tiene modificaciones constantes.
- Cuando cambia la estructura de una tabla.
- Al momento de incluir en los procedimientos tablas adicionales.
- Cuando se cambia de versión de Servidor de BD

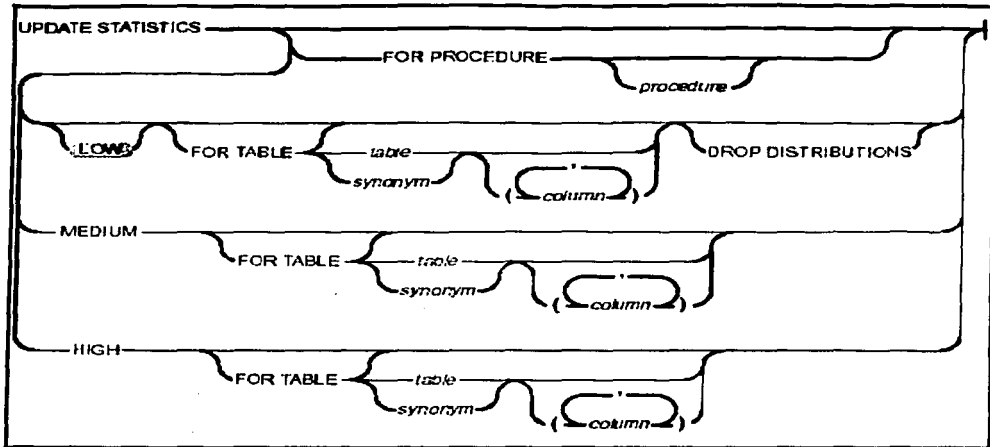
La siguiente tabla muestra las tres diferentes opciones en que puede ejecutarse la utilería update statistics pero aplicable unicamente a tablas y no a procedimientos:

Modo de Ejecución	Descripción
LOW	El servidor de BD actualiza las estadísticas de los datos en los siguientes sistemas de catálogos: Systables, syscolumns y sysindexes lo cual causa que se genere menos información y por consiguiente es más rápido que las opciones HIGH y MEDIUM.
HIGH	Genera información más completa que la opción anterior por lo que actualiza estadísticas de los catálogos de la opción LOW y además del catálogo sysdistrib, que sirve para la generación de una distribución exacta.
MEDIUM	Este proceso actualiza las estadísticas de distribución de datos en el catálogo sysdistrib. Este modo se ejecuta más rápido que el modo LOW pero más lento que el modo high.

Tabla. Niveles de actualización de estadísticas en las tablas de la Base de Datos

El siguiente esquema muestra la sintaxis de ejecución de esta utilería:





Sintaxis de la utilidad Update Statistics

Donde:

<i>column</i>	Nombre de una columna en la tabla especificada.
<i>procedure</i>	Procedimiento para el cual se requiere realizar la actualización de estadísticas.
<i>synonym</i>	Nombre del sinónimo de la tabla que se desea realizar la actualización de estadísticas.
<i>table</i>	Nombre de la tabla a la que se le van a actualizar las estadísticas.

### Oncheck

Esta utilidad puede reparar los siguientes tipos de estructuras de disco

- Páginas de Bitmap
- Particiones de blobspaces
- Páginas de blobspaces
- Índices
- Páginas de Sbspaces
- Particiones de Metadatos para sbspaces

TESIS CON  
FALLA DE ORIGEN

En caso de que `oncheck` detecte errores en otras estructuras, envía mensajes de alerta de las inconsistencias

Cuando se utiliza la opción `-y oncheck` realiza la reparación de manera automática, como se muestra en el siguiente ejemplo:



```
oncheck -cd -y
oncheck -cD -y
oncheck -ci -y
oncheck -cl -y
```

La utilidad `oncheck` no puede reparar una tabla en un `dbspace`, `sbspace` ó espacio externo. Lo que si puede reparar son índices en un `sbspace` ó espacio externo.

Al momento de ejecutar la utilidad `oncheck` se realiza un bloqueo compartido, por lo que otros usuarios no pueden actualizar, insertar ó borrar hasta que terminen las siguientes revisiones:

- Cuando se checan datos.
- Cuando se revisan índices (`-ci`, `-cl`, `-pk`, `-pK`, `-pl` ó `-pL`) y las tablas están utilizando modo de bloqueo a nivel página.
- Cuando se le adiciona el parámetro `-x` a las opciones del párrafo anterior y el bloqueo de la tabla está a nivel registro.

También esta opción lleva a cabo la revisión del sistema de catálogos con `-cc`, llevando una revisión completa de todos los catálogos de sistema de todas las bases de datos. Es recomendable realizar la actualización de estadísticas para realizar una revisión segura y así poder revisar el sistema de catálogos de la base de datos de la `nomina_ofm`; esto se realizaría de la siguiente manera:

```
oncheck -cc nomina_ofm
```

Las opciones `-cd` y `cD` revisan páginas de datos, para el caso de la opción `-cc` revisa todas las páginas excluyendo `blobpages` y `ysbpages`. La opción `-cD` es similar a la anterior pero de manera adicional incluye la revisión de `blobpages`.

Estas son las opciones más utilizadas de esta herramienta.

### **onmode**

Las opciones de `onmode` determinan cual de las siguientes operaciones pueden realizarse.

- Cambiar el modo de operación del servidor de base de datos
- Forzar a que el servidor realice un checkpoint
- Cambiar la residencia de las porciones virtuales y residentes de memoria compartida.
- Cambiar (Switchear) el archivo de log lógico.
- Eliminar las sesiones del servidor de base de datos.



- Añadir un segmento de memoria compartida hacia la porción de memoria compartida virtual.
- Añadir o remover procesadores virtuales
- Regenerar un archivo .infos
- Colocar parámetros para soporte de decisiones
- Liberar segmentos no utilizados de memoria.
- La opción onmode -d para colocar tipos de replicación de datos.
- Colocar opciones de caché de sentencias SQL

Esta utilidad sólo puede ser ejecutada por el Administrador del Servidor de Base de Datos.

Sintaxis de la utilidad onmode:

Para el caso del cambio de estatus del servidor de Base de Datos se utilizan las siguientes opciones:

Opción	Descripción
-k	Toma el servidor de base de datos en modo offline para remover memoria compartida. En el prompt se pregunta para confirmar, si se desea eliminar esta parte, se utiliza la opción -y seguido de la opción -s (-ky).
-m	Toma el servidor de base de datos de modo inactivo a modo online
-s	Da de baja el servidor de base de datos
-u	Da de baja el servidor de base de datos inmediatamente

Para forzar a un checkpoint se utilizan las siguientes opciones:

Opción	Descripción
-c	Forza a un checkpoint que descargue los buffers a disco.
block	Bloquea el servidor de base de datos de cualquier transacción, de esta forma los usuarios pueden acceder en un modo de sólo lectura. Esta opción se usa comúnmente para respaldar.
unblock	Desbloquea el servidor de Base de Datos, de esta forma el servidor opera de manera normal.

Opciones para cambiar la residencia de la memoria compartida:

Opción	Descripción
-n	Finaliza la residencia forzada de la porción residente de la memoria compartida
-r	Inicia la residencia forzada de la porción residente de la memoria compartida

Para poder utilizar estas opciones en el archivo de configuración ONCONFIG el parámetro *RESIDENT* debe estar asignado a 1.

Opciones para switchear el archivo de log lógico

Opción	Descripción
-l	Cambia del actual archivo de log lógico al siguiente archivo de log lógico

Opciones para eliminar las sesiones del servidor de Base de Datos

TESIS CON  
DE ORIGEN



Opción	Descripción
-z sid	Elimina las sesiones especificadas por el sid. El sid es el identificador de la sesión que se desea eliminar.

#### Opciones para añadir segmentos de memoria compartida

Opción	Descripción
-a seg_size	Permite añadir un nuevo segmento de memoria compartida virtual. El tamaño se especifica en kilobytes. Una de las restricciones es que no debe exceder el límite del tamaño de la memoria compartida del sistema operativo.

#### Opciones para añadir o remover Procesadores Virtuales

Opción	Descripción
-p number	Añade o borra los procesadores virtuales. El argumento number indica el número de los procesadores virtuales a agregar o remover. Si este valor es negativo los procesadores son borrados y si es positivo los agrega. Los tipos de procesadores virtuales son: AIO,LIO,PIO,SHM,SOC y TLI.

### Onparams

Cualquier comando de onparams falla, si un respaldo de algún espacio de almacenamiento está en progreso. La función de esta utilería está orientada al mantenimiento tanto de logs físicos como lógicos la siguiente tabla muestra las opciones:

Parámetros de onparams	Función	Estatus del servidor
onparams -a -d dbSPACE [-l] -s size	Añade un archivo de log lógico	Modo online, quiescent ó fast recovery
onparams -d -l lognum	Borra un archivo de log lógico	Modo online, quiescent ó fast recovery
onparams -p	Cambia el tamaño o localización del log físico	Únicamente modo quiescent

El usuario puede ser root o informix para ejecutar esta utilería

#### Descripción de los parámetros

-a -d dbSPACE	Añade un archivo de log lógico, al final del archivo de log listado en el dbSPACE especificado.
-i	Insertar un archivo de log lógico, después del actual archivo de log.
-s size	Especifica el tamaño en kilobytes para un nuevo archivo de log lógico.
-d -l lognum	Permite al usuario liberar el archivo de log lógico especificado por el número de archivo de log.
-y	Provoca que el servidor de base de datos automáticamente responda con "yes" a las peticiones.
-p	Cambia la localización del archivo de log físico
-d dbSPACE	Cambia la localización del log físico en el dbSPACE especificado
-s size	Cambia el tamaño (en kilobytes) del log físico.





Otra forma de cambiar el tamaño del log físico, es editar el archivo **ONCONFIG** y reinicializar el servidor de base de datos

A continuación veremos unos ejemplos de comandos con esta utilidad:

```
onparams -a -d rootdbs -s 1000      # añade un archivo de log lógico de 1000KB para el
                                   dbspaces rootdbs
onparams -a -d rootdbs -i          # inserta el archivo de log lógico después del log actual
onparams -d -l 7                   # borra el log 7.
onparams -p -d dbspace1 -s 3000    # cambia de tamaño y mueve el archivo de log físico
                                   para el dbspace1
```

## Onspaces

La función primordial de esta utilidad es el manejo de los espacios de almacenamiento.

Se pueden realizar las siguientes tareas con la utilidad **onspaces**:

- Crear un dbspace, un dbspace temporal, un blobspace o un extspace.
- Crear un sbpace o un sbpace temporal
- Cambiar las especificaciones por default de un sbpace
- Borrar un dbspace, blobspace, sbpace o extspace
- Añadir un chunk a un dbspace o blobspace
- Añadir un chunk a un sbpace
- Borrar un chunk en un sbpace, blobspace o sbpace
- Inicializar un espejo
- Terminar un espejo
- Cambiar el status de un chunk en mirror
- Especificar el parámetro **DATASKIP**

TESIS CON  
FALLA DE ORIGEN

Cuando se utiliza **onspaces** para administrar espacios de almacenamiento, el servidor de base de datos actualiza información acerca del espacio en el archivo **oncfg\_servername.servnum**.

Se puede especificar un máximo de 2047 chunks para un espacio de almacenamiento, y un máximo de 2047 espacios de almacenamiento en el servidor de base de datos del sistema.

Para la ejecución de esta utilidad debe ser el usuario **root** o **informix**.

La siguiente tabla muestra los parámetros y su funcionalidad para la creación de un **dbspace**, **dbspace temporal**, **blobspace** o **extspace**:



Parámetro	Descripción
-c	Crea un dbspace, blobospace, sbspace ó extspace
-d dbspace	Nombre del dbspace que va a ser creado
-g pageunit	Especifica el tamaño del blobospace en términos de páginas
-l location	Especifica la localización del extspace
-m pathname offset	Especifica una ruta opcional para el chunk espejo del chunk inicial.
-o offset	Identifica en kilobytes, la compensación de la partición o el dispositivo del chunk inicial.
-p pathname	Indica el dispositivo o la partición en disco, del chunk inicial, blobospace ó dbspace que se está creando.
-s size	Indica en kilobytes el tamaño del chunk inicial del nuevo blobospace ó dbspace
-t	Crea un dbspaces temporal para almacenar tablas temporales.
-x extspace	Nombre del extspace que va a ser creado.

Cuando se crea un dbspace temporal con onspaces, el servidor de base de datos utiliza los dbspaces que son creados, después se realizan los siguientes pasos:

-Se añade el nombre del nuevo dbspace temporal en la lista de dbspaces temporales, en el parámetro de configuración DBSPACETEMP, en la variable de ambiente DBSPACETEMP o ambos.

- Se reinicializa el servidor de base de datos.

Para la creación de un sbspace ó un sbspace temporal se utilizan las siguientes opciones:

Opcion	Descripción
-S Sbspace	Nombre del sbspace que se va a crear
-c	Crea un sbspace. Se pueden crear hasta 2047 espacios de almacenamiento de un solo tipo.
-m pathname offset	Especifica una trayectoria opcionaly offset para el chunk que espejea el chunk inicial del nuevo sbspace.
-Mo offset	Indica en kilobytes el offset dentro de una partición del disco o dentro del dispositivo donde los megadatos serán almacenados.
-Ms mdszie	Especifica el tamaño, en kilobytes, del área de megadatos alojados en el chunk inicial.
-t	Opción para la creación de un sbspace temporal.
- Df default list	Lista de especificaciones default para un smart large objects almacenados en el sbspace.

## Oninit

TESIS CON  
FALLA DE ORIGEN

Esta herramienta es utilizada desde la línea de comandos para inicializar: el servidor de bases de datos, memoria compartida y colocar el estatus del servidor de bases de datos a modo online (en línea). Si es utilizado con la opción -i (oninit -i) se puede inicializar también el espacio en disco.

Los usuarios root o informix son los que pueden ejecutar oninit. Antes de que se inicialice el Servidor de Base de Datos, se deben definir las variables de ambiente INFORMIXSERVER con el dbservername que se eligió cuando se colocaron los parámetros de configuración, DBSERVERNAME. La variable de ambiente INFORMIXSERVER no es requerido para la inicialización. Sin embargo, si esta variable



no es colocada, el servidor de base de datos no construye la tabla sysmaster, de igual manera la herramienta DB-Access, requiere que esta utilería sea colocada.

La forma de ejecutar esta utilería es la siguiente:

#### Sintaxis

Comando	Opciones
oninit	-y -p -s

- y Hace que el servidor de base de datos responda "yes" de manera automática.
- p Con esta opción, se direcciona a oninit a no buscar tablas temporales. El servidor de base de datos retorna a modo online mas rápidamente, pero el espacio utilizado por las tablas temporales no es reivindicado.
- s Inicializa memoria compartida y el servidor de base de datos se va a modo inactivo.

Forma de inicializar la memoria compartida sin opciones:

Cuando se ejecuta oninit sin opciones, el servidor de base de datos es dejado en modo online, después de que la memoria compartida es inicializada, por ejemplo los siguientes comandos toman el servidor de base de datos en modo off-line y lo retornan a modo online.

```
onmode -ky
oninit
```

Inicializando memoria compartida con la opción -s

Los siguientes comandos dan de baja el servidor de base de datos y lo resetean nuevamente en modo inactivo.

```
onmode -ky
oninit -s
```

- i Causa que el servidor de base de datos inicialice espacio en disco y memoria compartida. Se recomienda mover el servidor de BD a modo online después de que es inicializado el espacio en disco.
- s Cuando es utilizada la opción -i, causa que el servidor de base de datos se vaya a modo inactivo, después de que se inicializa el espacio en disco

Nota: Cuando se inicializa espacio en disco, la inicialización destruye todos los datos que el servidor de datos actualmente maneja. Además de que el servidor de base de datos debe estar en modo offline cuando se inicializa espacio en disco.

TESIS CON  
FALLA DE ORIGEN



**Cambiando el estado del servidor de base de datos en modo recuperación**

Opción	Propósito
-r	Inicializa el servidor de base de datos en modo recuperación. También cambia el modo del servidor de base de datos para replicación de datos en alta disponibilidad.

**Onmonitor**

Con esta herramienta se pueden realizar varias tareas administrativas, para invocar esta utilidad se ejecuta desde la línea de comandos, como a continuación se muestra:

```
% onmonitor
```

Para el administrador del servidor de base de datos como son root o informix, aparece el menú principal que contiene todas las opciones administrativas y para los demás usuarios tienen acceso pero a nivel consultas únicamente.

El menú principal de ON-Monitor despliega los siguientes submenús

- Status
- Parameters
- Dbspaces
- Mode
- Force Ckpt
- Archive
- Logical Logs
- Exit option

Opciones de pantalla de ON-Monitor

Menú de Estatus

Menu	Descripción
Profile	Despliega estadísticas del rendimiento de la base de datos.
Userthreads	Despliega el estatus de los procesos de usuarios actuales.
Spaces	Despliega información acerca de los espacios de almacenamiento y de los chunks del servidor de base de datos.
Databases	Despliega el nombre, dueño y modo de loggeo de las 100 primeras bases de datos.
Logs	Despliega información del status acerca de los buffers físicos y lógicos del log.
Archive	Despliega una lista de todas las cintas de respaldo y los archivos de log lógicos que se requiere para restaurar datos utilizando ontape.
Data-Replication	Despliega el status y la configuración del Alta Disponibilidad y Replicación de datos (HDR).
Output	Almacena la salida de información acerca de otro status en un archivo especificado.
Configuration	Copia la actual configuración del servidor de base de datos hacia un archivo.

TESIS CON  
FALLA DE ORIGEN



## Menú de Parámetros

Menú	Descripción
Initialize	Inicializa espacio en disco del servidor de base de datos o modifica parámetros del espacio en disco.
Shared Memory	Inicia memoria compartida o inicializa parámetros de la misma del servidor de base de datos.
Performance	Especifica el número de procesadores virtuales por cada tipo de procesador virtual (VP).
data-replication	Especifica parámetros de HDR (Replicación Alta Disponibilidad).
Diagnostics	Especifica valores para los parámetros de diagnósticos.
PdQ	Cambia este parámetro para consultas paralelas a base de datos.
Add-log	Añade archivos de logs lógicos a un dbspaces.
Drop-log	Borra un archivo de log lógico de un dbspaces.
Physical-log	Cambia el tamaño o la localización del log físico en el servidor.

## Menú de Dbspaces

Menú	Descripción
Create	Crea dbspaces
BLOBspace	Crea un blobspace
Mirror	Agrega ó borra un espejo a un espacio de almacenamiento existente.
Drop	Borra un espacio de almacenamiento del servidor
Info	Despliega el número de identificación, localización y espacio ocupado de cada chunk asignado a un espacio de almacenamiento.
Add chunk	Añade un chunk a un espacio de almacenamiento
Dataskip	Cambia el parámetro de la Base de Datos.
Status	Cambia el estatus de un chunk a un espejo.

## Menú de modos (estatus)

Menú	Descripción
Startup	Inicializa la memoria compartida y cambia el estatus del servidor a estado monousuario (Quiescent).
On-line	Cambia el estado del servidor de modo monousuario a estatus en línea.
Gracefull-Shutdown	Cambia el servidor de base de datos del estatus online a modo Quiscent permitiendo que todos los trabajos de los usuarios concluyan
Immediate - Shutdown	Cambia el estado del servidor de BD de online a modo Quiscent en un lapso de 10 segundos.
Take - Offline	Separa la memoria compartida y cambia de manera inmediata el servidor de BD a fuera de línea (off-line)
Add - Proc	Añade procesadores virtuales
Drop - Proc	Borra procesadores virtuales
Decisión - Support	Asigna el parámetro de soporte decisión dinámicamente

## Menú de Force - Ckpt

Menú	Descripción
Force - Ckpt	Despliega el tiempo del checkpoint más reciente ó fuerza al servidor de base de datos a ejecutar un checkpoint.

TESIS CON  
FALLA DE ORIGEN



entre otros, con la finalidad de que el usuario del Sistema pueda tener la respuesta rápida y oportuna de datos. Estas utilerías que porporciona el servidor de Base de Datos Informix son del uso diario de los administradores.

En el caso del SIAPSEP en cuanto a las tablas que tienen mayor contención pueden utilizarse varias formas de identificarlos, como puede ser la utilería onstat ó directamente con las tablas SMI. Otras de las funciones de estas utilerías es proporcionar un estadístico, en cuento al espacio utilizado de base de datos, usuarios que se conectan, periodos en los que hay más actividad, objetos que son más consultados entre otros. A continuación se tienen las herramientas de respaldo que son de igual forma de uso muy frecuente para resguardar la información de los sistemas.

#### IV.4 HERRAMIENTAS DE RESPALDO Y RESTAURACIÓN DE BASE DE DATOS

Este tipo de utilerías le sirven al administrador para realizar respaldos y restauraciones de Bases de Datos, los niveles de respaldo que a continuación se señalan van del más sencillo hasta el más complejo.

##### Unload

Esta también es considerada una herramienta de respaldo mediante la cual se puede descargar datos de objetos en específico por ejemplo tablas con información crítica la instrucción se ejecuta desde el dbaccess y tiene la siguiente sintáxis:

Unload to nombre\_archivo delimiter 'delimitador' criterios de descarga.

Ej.

```
unload to empleado.txt delimiter '|' select * from empleado;
```

En el ejemplo anterior la información queda en el archivo de texto empleado.txt. Se puede tener un detalle mayor utilizando las condiciones where.

##### Load

Esta herramienta restaura la información descargada por la utilería unload y la sintaxis es la siguiente:

```
load from nombre_archivo delimiter 'delimitador' insert into nombre_tabla
```

Retomando el ejemplo de descarga anterior la carga se realizaría de la siguiente forma:



### Load from empleado.txt delimiter '|' insert into empleado

Esta herramienta se utiliza mucho cuando se desea cargar información de una base de datos a otra ó cuando se están realizando pruebas y no se desea respaldar la base de datos completa, pero si los objetos de la base de datos van a cambiar es muy útil esta herramienta.

### Dbschema

Esta utilidad tiene los siguientes propósitos:

- Desplegar sentencias SQL's requeridas para replicar una base de datos, tablas, vistas ó procedimientos.
- Despliega información sobre permisos
- Despliega información sobre tipo de datos definidos por el usuario.

Cuando se realiza un esquema completo de la base de datos se obtiene la siguiente información.

- Script completo de todo los sinónimos con la instrucción completa de creación (create synonym)
- Script completo de todas las tablas (create table), vistas (create vista), funciones (create function) y procedimientos almacenados (create procedure).
- Contiene toda la información referente a permisos, de todos los objetos en la base de datos (Grant).

La instrucción para poder realizar el esquema completo de la base de datos del SIAPSEP (nomina\_ofm) sería de la siguiente forma:

```
dbschema -d nomina_ofm > nomina_ofm.sql
```

El signo mayor que indica que la salida va a ser registrada en el archivo nomina\_ofm.sql.

El uso común de esta herramienta es cuando se requiere pasar una base de datos pero sin tener información, o cuando se pasa de un ambiente de pruebas para liberación y únicamente se desea crear el cascaron de la base de datos pero conservando todos los permisos y objetos.

A continuación se presentan fragmentos del esquema de la base de datos de la nomina\_ofm:

Parte de permisos a nivel base de datos:





```

DBSCHEMA Schema Utility      IDS Version 9.20.UCS
Copyright (C) Informix Software, Inc., 2000
Software Serial Number INF#J010494
grant dba to "informix";
grant resource to "public";
grant resource to "a713ap00";
grant dba to "al6jap00";

```

Parte del esquema referente a tablas:

```

{ TABLE "informix".empleado row size = 355 number of columns = 28 index size = 85
}
create table "informix".empleado
(
  u_version char(1),
  rfc char(13) not null constraint "informix".n171_604,
  nom_emp char(60) not null constraint "informix".n171_605,
  sexo char(1) not null constraint "informix".n171_606,
  edo_civil smallint not null constraint "informix".n171_607,
  edo_nac smallint not null constraint "informix".n171_608,
  qna_ini integer not null constraint "informix".n171_609,
  act_inact char(1) not null constraint "informix".n171_610,
  ban_pago smallint not null constraint "informix".n171_611,
  mot_baja smallint not null constraint "informix".n171_612,
  qna_baja integer not null constraint "informix".n171_613,
  qna_ing_gob integer not null constraint "informix".n171_614,
  qna_ing_sep integer not null constraint "informix".n171_615,
  niv_academico char(2) not null constraint "informix".n171_616,
  tipo_lit smallint not null constraint "informix".n171_617,
  tipo_lic smallint not null constraint "informix".n171_618,
  num_lit integer not null constraint "informix".n171_619,
  num_lic integer not null constraint "informix".n171_620,
  emp_cta_sar char(18) not null constraint "informix".n171_621,
  acumulado_horas_42 float not null constraint "informix".n171_622,
  acumulado_horas_48 float not null constraint "informix".n171_623,
  emp_dom_char(60) not null constraint "informix".n171_624,
  emp_dom_col char(60) not null constraint "informix".n171_625,
  emp_dom_pob char(60) not null constraint "informix".n171_626,
  emp_dom_cp char(5) not null constraint "informix".n171_627,
  emp_cta_bancaria char(18) not null constraint "informix".n171_628,
  bco_admdor smallint not null constraint "informix".n171_629,
  bco_plaza smallint not null constraint "informix".n171_630,
  primary key (rfc) constraint "informix".empleado1
);
revoke all on "informix".empleado from "public";

create index "informix".empleado1 on "informix".empleado (emp_cta_bancaria);

create index "informix".empleado_1 on "informix".empleado (rfc,
act_inact);

{ TABLE "informix".empleado_curp row size = 32 number of columns = 3 index size =
25 }
create table "informix".empleado_curp
(
  u_version char(1),
  rfc char(13) not null ,
  cve_unica char(18) not null .

```

TESIS CON  
FALLA DE ORIGEN



```

primary key (rfc) constraint "informix".empleado_curp1
);
revoke all on "informix".empleado_curp from "public";

( TABLE "informix".historia_pago row size = 497 number of columns = 19 index size
= 339 )
create table "informix".historia_pago
(
  u_version char(1),
  rfc char(13) not null constraint "informix".n209_906,
  cod_pago smallint not null constraint "informix".n209_907,
  unidad smallint not null constraint "informix".n209_908,
  subunidad smallint not null constraint "informix".n209_909,
  cat_puesto char(7) not null constraint "informix".n209_910,
  horas smallfloat not null constraint "informix".n209_911,
  cons_plaza integer not null constraint "informix".n209_912,
  qna_pago integer not null constraint "informix".n209_913,
  tipo_calculo char(1) not null constraint "informix".n209_914,
  qna_proc integer not null constraint "informix".n209_915,
  cons_qna_proc smallint not null constraint "informix".n209_916,
  nivel_sueldo smallint not null constraint "informix".n209_917,
  tipo_pago smallint not null constraint "informix".n209_918,
  num_perc smallint not null constraint "informix".n209_919,
  num_desc smallint,
  cadena_cpto varchar(80,40) not null constraint "informix".n209_920,
  cadena_importe1 varchar(180,90) not null constraint "informix".n209_921,
  cadena_importe2 varchar(180,90),
  primary key
(rfc,cod_pago,unidad,subunidad,cat_puesto,horas,cons_plaza,qna_pago,tipo_calculo,qna_proc,cons_qna_
proc)
  constraint "informix".u331_2658
);
revoke all on "informix".historia_pago from "public";

create index "informix".cred_sal on "informix".historia_pago (rfc,
qna_pago,cod_pago,tipo_pago);
create index "informix".historia_pago2 on "informix".historia_pago
(rfc,cod_pago,unidad,subunidad,cat_puesto,horas,cons_plaza,
tipo_calculo,qna_pago,cons_qna_proc);
create index "informix".historia_pagoidx81 on "informix".historia_pago
(rfc,cod_pago,unidad,subunidad,cat_puesto,horas,cons_plaza,
qna_proc);
create index "informix".historia_pagoidx82 on "informix".historia_pago
(rfc,cod_pago,unidad,subunidad,cat_puesto,horas,cons_plaza,
qna_pago);
create index "informix".qnapago on "informix".historia_pago (qna_pago);

create index "informix".cat_qna on "informix".historia_pago (cat_puesto,
qna_proc);

```

#### Parte referente a los permisos sobre tablas:

```

grant all on "informix".empleado to "public" as "informix";
grant all on "informix".empleado_nss to "public" as "informix";
grant all on "informix".emp_plaza_msg to "public" as "informix";
grant all on "informix".entidades_fed to "public" as "informix";
grant all on "informix".errores_de_movimie to "public" as "informix";
grant all on "informix".error_de_sistema to "public" as "informix";
grant all on "informix".escuela_origen to "public" as "informix";
grant all on "informix".estatus to "public" as "informix";
grant all on "informix".estatus_cheque to "public" as "informix";

```

TESIS CON  
DE ORIGEN

```
grant all on "informix".estatus_ct to "public" as "informix";
grant all on "informix".estatus_mov_tec to "public" as "informix";
grant all on "informix".estatus_plaza to "public" as "informix";
grant all on "informix".estr_org_ct to "public" as "informix";
grant all on "informix".estr_org_ls to "public" as "informix";
grant all on "informix".etapas_calc to "public" as "informix";
grant all on "informix".formas to "public" as "informix";
grant all on "informix".grupos to "public" as "informix";
grant all on "informix".grupo_gen to "public" as "informix";
grant all on "informix".grupo_usu to "public" as "informix";
```

#### Parte del esquema referente a procedimientos:

```
create procedure "informix".act_inact_s1( where1 char(1) )
    returning char(1),
    char(1),
    char(8);

    define p_field1 char(1);
    define p_field2 char(1);
    define p_field3 char(8);

    foreach select u_version, act_inact, situacion
    into p_field1,
    p_field2,
    p_field3
    from act_inact
    where act_inact = where1
    return p_field1,
    p_field2,
    p_field3
    with resume;
end foreach;
end procedure;

create procedure "informix".acumulados_u( uversion_used smallint,
    update1 char(1),
    update2 char(13),
    update3 integer,
    update4 money(19,2),
    update5 money(19,2),
    update6 money(19,2),
    update7 money(19,2),
    where1 char(13),
    where2 integer,
    where3 char(1) )

if (uversion_used = 1) then
update acumulados
set u_version = update1,
rfc = update2,
qna_proc = update3,
tot_sar = update4,
tot_gravable = update5,
tot_no_gravable = update6,
tot_ispt = update7
where rfc = where1
and qna_proc = where2
and u_version = where3;
else
update acumulados
```

TESIS CON  
FALLA DE ORIGEN



```

set u_version = update1,
rfc = update2,
qna_proc = update3,
tot_sar = update4,
tot_gravable = update5,
tot_no_gravable = update6,
tot_lspt = update7
where rfc = where1
and qna_proc = where2;
end if;

if (dbinfo('sqlca.sqlerrd2') != 1) then
raise exception 25;
end if;
end procedure;

create procedure "informix".acumulados_i( insert1 char(1),
insert2 char(13),
insert3 integer,
insert4 money(19,2),
insert5 money(19,2),
insert6 money(19,2),
insert7 money(19,2)

insert into acumulados(u_version, rfc, qna_proc, tot_sar, tot_gravable, tot_no_gravable, tot_lspt )
values (insert1,
insert2,
insert3,
insert4,
insert5,
insert6,
insert7);
end procedure;

```

#### Parte de permisos a procedimientos almacenados:

```

grant execute on "informix".basifica_emp_e to "public" as "informix";
grant execute on "informix".basifica_emp_f to "public" as "informix";
grant execute on "informix".movr_detalle_j to "public" as "informix";
grant execute on "informix".movr_detalle_d to "public" as "informix";
grant execute on "informix".movr_detalle_s to "public" as "informix";
grant execute on "informix".cifras_control_e to "public" as "informix";
grant execute on "informix".cifras_control_f to "public" as "informix";
grant execute on "informix".codigos_operac_e to "public" as "informix";
grant execute on "informix".codigos_operac_f to "public" as "informix";
grant execute on "informix".control_acum_c_i to "public" as "informix";
grant execute on "informix".cptos_prorrate_e to "public" as "informix";
grant execute on "informix".movs_reclamo_s to "public" as "informix";
grant execute on "informix".movs_reclamo_s1 to "public" as "informix";
grant execute on "informix".crea_can_coder_s to "public" as "informix";
grant execute on "informix".crea_can_error_f to "public" as "informix";
grant execute on "informix".aguinaldo_s to "public" as "informix";

```

#### Parte de creación de vistas:

```

create view "informix".tables (table_schema,table_name,table_type,remarks) as
select TRIM ( BOTH '' FROM x0.owner ) ,TRIM ( BOTH '' FROM x0.tabname
) ,TRIM ( BOTH '' FROM "informix".ansitabtype(x0.tabtype )
) ,"informix".null()from "informix".systables x0 where ((x0.tabtype
IN ('T','V' )) AND ((x0.owner = USER ) OR EXISTS (select x1.grantor
.x1.grantee .x1.tabid .x1.tabauth from "informix".systabauth

```



```
x1 where ((x0.tabid = x1.tabid ) AND (x1.grantee IN (USER
,'public' ) ) ) ) ;
```

### Parte de actualización de estadísticas

```
Update statistics high for table emp_plaza
(rfc, cod_pago, unidad, subunidad, cat_puesto, horas, cons_plaza);
update statistics medium for table historia_pago
(rfc, cod_pago, unidad, subunidad, cat_puesto, horas, cons_plaza, qna_pago);
```

La forma en como se puede restatarar un esquema de una base de datos generado por el dbschema es como cualquier instrucción SQL pero se debe eliminar el encabezado o comentarlo debido a que no son instrucciones permitidas.

### Dbexport

La utilería dbexport descarga una base de datos dentro de archivos de texto y crea un archivo con el esquema de la misma. Una de las ventajas de dejar este formato de archivos es que pueden ser editados y modificados.

El respaldo puede quedar en los siguientes destinos:

- Descarga la base de datos y el archivo que contiene el esquema hacia disco
- Descarga la base de datos y el archivo que contiene el esquema hacia cinta
- Descargar el archivo que contiene el esquema de la Base de datos a disco y descargar los datos a cinta.

El nombre del directorio de respaldo queda con el nombre de la base de datos con extensión exp por ejemplo nomina\_ofm.exp

La sintaxis es la siguiente:

```
dbexport -d nombre_bd
```

Ej:

```
dbexport -d nomina_ofm
```

La forma de ejecución es muy sencilla y es muy utilizado este tipo de respaldo debido a que es fácil de entender y manipular.

### Dbimport

La utilería dbimport crea una base de datos y carga los datos de los archivos de texto generados por el dbexport, el archivo de entrada para esta utilería es el archivo que contiene el esquema de la Base de Datos, los archivos de datos no necesariamente deben de ser los generados por el dbexport ya que se pueden utilizar otros siempre y cuando cumplan el formato.

TESTA CON  
FALLA DE ORIGEN

Las opciones de localización de los archivos pueden ser las siguientes:

- Pueden ser recuperados en disco
- Pueden ser recuperados en cinta
- El esquema de la Base de Datos puede ser localizado en disco y los archivos de datos en cinta.

La sintaxis de carga es la siguiente:

```
dbimport archivo -d dbspace
```

Ejemplo:

```
dbimport nomina_ofm -d bddats1
```

La opción `archivo` se refiere al archivo descargado por la opción `dbexport` pero sin extensión (.exp), el parámetro `-d` indica el `dbspace` en donde debe ser cargada la base de datos, en caso de no ser colocado este parámetro la base de datos es creada en el `dbspace` de root.

## Ontape

Cuando se está respaldando y la información cambia constantemente, pero parte de esta información se mantiene estable, no se recomienda respaldar la información volátil.

La utilidad `ontape` se utiliza para crear respaldos de datos y archivos de logs lógicos.

Por lo anterior Informix soporta los siguientes niveles de respaldo:

- Nivel 0.-** Respalda todas las páginas utilizadas. Un respaldo de este tipo contiene una copia de cada página de disco usada (`dbspaces` y `blobspaces`) que son necesarias para restaurar el Servidor de base de datos. Una desventaja es que se pueden consumir grandes cantidades de tiempo debido a que Informix debe escribir todas las páginas hacia cinta.
- Nivel 1.-** Respalda todos los cambios hechos después del último respaldo de nivel 0. Todos los datos copiados hacia el respaldo reflejan el estado de los datos a la hora que el respaldo de nivel 1 inició.
- Nivel 2.-** Respalda todos los cambios hechos después del último respaldo de nivel 1. Al igual que el anterior, los datos copiados a cinta reflejan el estado de los datos a la hora que comenzó el respaldo de nivel 2. Obviamente un respaldo de este tipo toma menos tiempo que un respaldo de nivel 1, debido a que solo se respaldan los cambios previamente hechos al último respaldo.



Es necesario hacer un plan para calendarizar los respaldos en forma adecuada para evitar largos retrasos al momento de estar respaldando ó restaurando datos.

A continuación se muestra un esquema que nos serviría para tomarlo como patrón a la hora de definir un esquema de respaldos:

Nivel	Calendario Diario	Calendario Semanal	Calendario Mensual
0	Cada noche	Domingos	Una vez al mes
1	Medio día	Miercoles	Una vez a la semana
2	Cada dos horas	Lunes, Martes, Jueves Viernes	Una vez al día

La figura 4.4 ilustra las unidades en donde se guarda la información cuando se realiza un respaldo de datos utilizando la herramienta ontape.

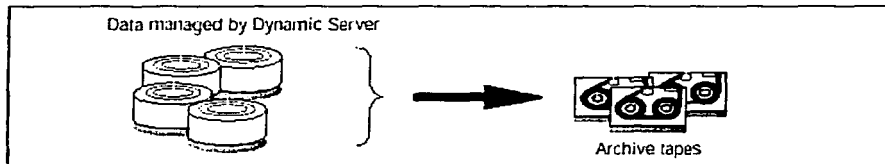


Figura 4.4 Unidades de almacenamiento de la utileria ontape

Consideraciones para que los respaldos sean más rápidos:

- Crear un dbspace temporal y almacenar las tablas temporales almacenadas durante un respaldo.
- Crear un dbspace normal donde se puedan almacenar todas las tablas temporales, y entonces liberar el dbspace antes de crear un respaldo.
- Liberar todas las tablas temporales antes de respaldar los dbspaces

Es necesario realizar un respaldo de nivel 0 después de haber creado un dbspace.

Para minimizar el tiempo para una restauración, se debe considerar lo siguiente:

- Tamaño y número de archivos
- El mínimo número de respaldos necesarios para restaurar algún respaldo de nivel 0.
- El número máximo es tres para cada nivel de archivo.

TESIS CON  
FUELE DEL ORIGEN

- Cantidad de datos que se intenta restaurar
- Tipo de restauración

Cuando se lleva a cabo una restauración completa del sistema, es posible restaurar unos dbspaces primero mientras Informix se encuentra en modo fuera de línea. Estos dbspaces permanecen disponibles mientras otros dbspaces están siendo restaurados, este tipo de restauración incrementa la disponibilidad de algunos dbspaces pero también incrementa el tiempo de la restauración total. Cuando los logs lógicos se llenan, Informix suspende el procesamiento antes de respaldar.

El tiempo requerido para alguna posible restauración está limitado por el tiempo necesario para leer y procesar los siguientes datos:

- Un respaldo de nivel 0 de los dbspaces que están siendo restaurados.
- Un respaldo de nivel 1, representando de 1 a 3 días de actividad en los dbspaces que están siendo restaurados.
- Archivos de log lógicos, representando menos de 1 día de actividad en el dbspace o dbspaces que están siendo restaurados.

Antes de crear un respaldo se deben tomar las siguientes consideraciones:

- Evitar utilizar tablas temporales durante una ardua actividad
- Cuando se crea una tabla temporal durante un respaldo mientras se está utilizando la utilidad ontape, esta tabla es colocada en DBSPACETEMP
- Asegurarse de que se tiene suficiente espacio para los logs lógicos a la hora de crear un respaldo.
- Realizar una copia del archivo ONCONFIG
- Verificar la consistencia de datos
- Correr Informix Dynamic Server en el modo apropiado
- Tomar en cuenta la disponibilidad del operador
- No utilizar el modo background
- Etiquetar las cintas apropiadamente

Cuando se lleva a cabo un respaldo Informix realiza lo siguiente:

- ◆ Compara el nivel de respaldo especificado con la información en la página reservada del respaldo.
- ◆ Cuando Informix no encuentra un registro de un respaldo previo en la página reservada, un respaldo de nivel 0 llega a ser el único nivel de respaldo válido.
- ◆ Temporalmente libera el estatus de los archivos de log lógicamente utilizados, y checa la cantidad total de espacio libre en los logs.
- ◆ Cuando el espacio libre es menor que el del archivo de log, Informix rechaza la requisición de respaldo y se recomienda que respalden los archivos de log lógicos
- ◆ Inicializa un checkpoint (llamado checkpoint de respaldo), el cual marca el inicio del respaldo.
- ◆ Sincroniza con otros procesos de respaldo para garantizar que no se respalden 2 procesos simultáneamente del mismo dbspace o blobospace





- Las direcciones del mas reciente registro escrito en el actual archivo de log lógico es también tomado en cuenta durante el checkpoint. Este registro llega a ser el último registro del archivo de log lógico que se copia como parte del respaldo de Dynamic Server.

Cuando se lleva a cabo un respaldo de tipo completo, Informix marca todos los archivos de log lógicos que contienen los registros de log de transacciones abiertas, por lo que no pueden ser liberadas antes de que se respalden.

Orden en que se ejecuta un respaldo:

- 1.-Informix añade una sección incluyendo las páginas reservadas desde el dbspace.raiz al respaldo
- 2.-Añade una sección incluyendo un snapshot (una imagen) de los logs lógicos, que contienen transacciones abiertas al mismo tiempo que se respalda el archivo de checkpoint.
- 3.-Enseguida se respaldan los dbspaces en un orden no particular
- 4.-Cuando Informix respalda un dbspace o blobspace, añade las páginas de las tablas temporalmente usadas para almacenar antes imágenes de logs físicos.
- 5.- La porción de disco de lectura del respaldo procede a completar cuando el servidor dinámico alcanza la ultima página

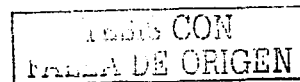
Antes de recibir datos de respaldo la utilería ontape escribe una página de cabecera en el dispositivo de respaldo. Esta página contiene la siguiente información:

- El tamaño de bloque del dispositivo de cinta
- El tamaño de la cinta
- Una bandera que indica que la cinta es para un respaldo
- Un timestamp que indica la fecha y la hora del respaldo
- El nivel de respaldo
- La localización física de un registro de checkpoint en el archivo de log lógico.

Cuando se decide que es necesario respaldar los archivos de bitácora lógica, se debe determinar si se quiere ejecutar un respaldo automático o continuo.

Se deben considerar los siguientes dos puntos cuando se usan tipos de datos de texto y BYTE en una base de datos que usa modo de transacción logging:

- Se deben de reutilizar los blobspaces, respaldar los archivos de bitácora lógica. Cuando los usuarios borran valores de texto o Byte en blobspace no llegan a ser liberadas para rehusos mientras se libera el archivo de bitácora que contiene los registros borrados. Para liberar el archivo de bitácora, se debe de respaldar.
- Cuando se tiene que recuperar un blobspace no disponible, en cinta se salta y hace casi imposible la recuperación de los valores de texto o Byte cuando llegan a ser necesarios. Sin embargo, los blobspace de los valores de texto o blobspace borrados llegan a ser libres cuando el blobspace llega a estar disponible aún cuando los valores de texto o Byte no fueron respaldados.



La utilidad `ontape` y `ON-Archive` proveen tipos de archivos incompatibles, por lo que no es posible crear un respaldo con `ontape` y restaurarlo con `ON-Archive`. De igual manera, no es posible restaurar logs lógicos creados con otra herramienta o viceversa.

`Ontape` utiliza 6 parámetros en el archivo de configuración `ONCONFIG` para crear archivos y respaldar archivos de log lógicos. El archivo `ONCONFIG` está localizado en el directorio `$INFORMIXDIR/etc`.

Los seis parámetros que `ontape` utiliza se dividen en 2 conjuntos: El primer conjunto especifica las características del dispositivo de cinta para el respaldo de datos. El segundo conjunto especifica las características del dispositivo de cinta para respaldos de archivos de log lógicos.

La siguiente lista muestra los parámetros de respaldo para datos:

<code>TAPEDEV</code>	Es el dispositivo de cinta utilizado para respaldar
<code>TAPEBLK</code>	Indica el tamaño de bloque de las cintas utilizadas para estar respaldando, en kilobytes
<code>TAPESIZE</code>	Indica el tamaño de las cintas utilizadas para estar respaldando, en kilobytes.

La siguiente lista muestra los parámetros para respaldos lógicos.

<code>LTAPEDEV</code>	Es el dispositivo de cinta de archivos lógicos
<code>LTAPEBLK</code>	Indica el tamaño de bloque de las cintas utilizadas para respaldos de archivos lógicos, en kilobytes
<code>LTAPESIZE</code>	Indica la capacidad de las cintas utilizadas para respaldos de archivos lógicos en kilobytes

Se deben de considerar dispositivos diferentes para el respaldo de datos y de logs lógicos, para el manejo independiente de respaldos.

Cuando estos 2 parámetros especifican el mismo dispositivo, los logs pueden llenarse y puede ocasionar que el servidor de informix, detenga el procesamiento durante un respaldo de datos y liberarlos hasta que este concluya.

Cuando solo existe un solo dispositivo de cinta y se desean crear respaldos mientras Informix está en modo on-line, se deben tomar las siguientes precauciones:

- Configurar Informix con una gran cantidad de espacio para las bitácoras lógicas mediante una combinación de muchos archivos de log o grandes archivos de log.
- Almacenar todas las tablas temporalmente creadas explícitamente en un `dbspace` dedicado y luego dar de baja el `dbspace` antes que sea respaldado.
- Crear un respaldo cuando existen bajas actividades de la base de datos.
- Liberar tantos archivos de log lógicos como sea posible, antes que se inicialice el respaldo



Se pueden especificar los valores de LTAPEDEV y TAPEDEV como trayectorias simbólicas. El utilizar esta opción habilita que se pueda switchear hacia otro dispositivo de cinta, u otro compatible, sin cambiar la trayectoria en el archivo ONCONFIG. Por ejemplo, se puede especificar la siguiente liga simbólica en el dispositivo de cinta /dev/rst0:

```
In -s /dev/rst0 /dbfiles/logtape
```

Luego cuando se coloca el parámetro LTAPEDEV, se hace como se muestra a continuación:

```
LTAPEDEV /dbfiles/logtape
```

Se puede cambiar hacia un dispositivo diferente, sin cambiar el parámetro LTAPEDEV, solo se necesita cambiar la liga simbólica como se muestra en el siguiente ejemplo:

```
In -s /usr/backups /dbfiles/logtape
```

Para solucionar el que se tenga sólo un dispositivo de respaldo se pueden especificar dispositivos de respaldo conectado a otro host o sistema de computo, y realizar un archivo o respaldo a través de la red hacia un dispositivo remoto, esto quiere decir que el dispositivo está conectado en otro sistema de computo. Para especificar un dispositivo de cinta en otro sistema de cómputo se tiene la siguiente sintaxis:

```
host_machine_name : tapedevice_pathname
```

El siguiente ejemplo especifica un dispositivo de cinta en el sistema de computo "mitla":

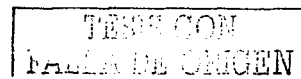
```
mitla : /dev/rst01
```

Exite una opción en donde los archivos de logs lógicos son vaciados a null esto quiere decir que cada que se llenan se van reutilizando. Informix recomienda que no se utilice /dev/null como un dispositivo cuando se está respaldando. Ahora bien, cuando se especifica /dev/null como un dispositivo de cinta se puede evitar el sobrepaso de un respaldo de nivel 0 que se requiere después de algunas operaciones. Obviamente no se puede restaurar Informix desde un archivo que se direccionó hacia /dev/null.

Se especifica esta trayectoria como un dispositivo de cinta cuando se decide que no se necesita recobrar transacciones de bitácoras lógicas. Cuando se especifica el dispositivo de cinta como /dev/null, el tamaño de bloque y el tamaño de la cinta son ignorados.

El parámetro de block size debe ser tan grande como el dispositivo lo permita y está dado en kilobytes.

Se puede cambiar la configuración de los parámetros de la utilería de respaldo ontape, utilizando el usuario de root (superusuario) o informix, desde onmonitor ó editando el archivo ONCONFIG. Estos cambios se pueden llevar a cabo cuando Informix está en modo on-line, toman efecto inmediatamente.



Cuando se va a cambiar el valor de LTAPEDEV hacia /dev/null, Informix debe operar en modo off-line, ya que el cambiarlo en modo on-line puede interrumpir el procesamiento, ya que Informix se detiene cuando encuentra que el siguiente archivo de log (en secuencia) no está libre.

Cuando se coloca LTAPEDEV hacia /dev/null, el Servidor de Base de Datos libera los logs lógicos sin que se requiera, que se respalden estos logs. Los logs lógicos no quedan marcados como libres, pero el Servidor de Base de Datos puede reutilizarlos

Seleccionando del menú Archive, la opción Tape-Parameters para cambiar los valores de TAPEDEV, TAPEBLK , and TAPESIZE. ON-Monitor despliega los valores actuales

Se debe Introducir la trayectoria completa del valor del dispositivo de cinta de log lógico, en el campo de Log Tape Device, igualmente se deben introducir los nuevos valores en los campos del dispositivo Block Size y Tape Size cuando sea apropiado. Estos cambios toman efecto inmediatamente.

Para cambiar los Parámetros de cinta de Respaldo de Logs Lógicos se realiza lo siguiente:

Con la utilería on-monitor se selecciona Logical-Logs del menú, después la opción Tape Parámetros para cambiar los valores de LTAPEDEV, LTAPEBLK y LTAPESIZE. ON-Monitor despliega los valores actuales.

Para cambiar los valores de TAPEDEV, TAPEBLK y TAPESIZE desde la línea de comandos se utiliza un editor de texto para modificar el archivo ONCONFIG.

El servidor de BD puede operar en modo en línea (on - line) cuando se respaldan archivos de bitácora lógicos.

Para respaldar archivos de bitácora lógicos, se usa la opción -a del comando ontape.

% ontape -a

-a envía el comando ontape para respaldar todos los archivos de bitácora llenos.

Cuando una cinta montada en LTAPEDEV llega a estar llena antes del fin del archivo de bitácora lógico, el comando ontape avisa que se debe montar una nueva cinta.

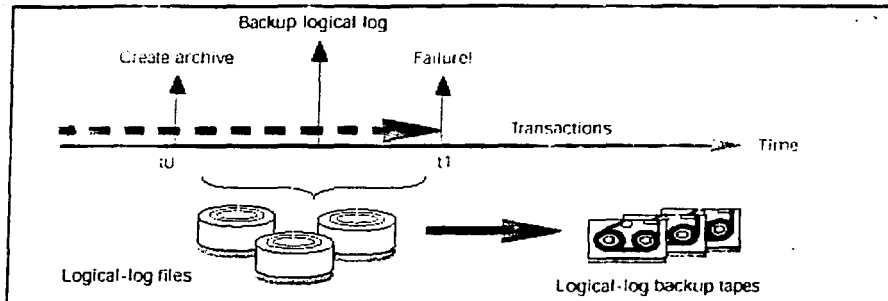
Para empezar un respaldo continuo de archivos de bitácora lógicos, se usa la opción -c del comando ontape.

% ontape -c

El servidor de BD respalda cada archivo de bitácora lógico hasta que este llega a llenarse. Los respaldos continuos no respaldan el archivo que se encuentre actualmente en uso y puede operar en modo On - line.



La figura 4.5 muestra cuando se realiza un respaldo de los logs lógicos.



### Restauraciones con Ontape

#### Restauración mixta

Una restauración de este tipo es una restauración en frío seguida de una restauración en caliente. Se restauran algunos dbspaces y blobspaces durante una restauración en frío (estando las bases de datos cerradas ó en estado fuera de línea) Informix opera en modo off-line, en cambio en la restauración en caliente (estando las bases de datos abiertas), Informix opera en modo on-line, y de igual manera se restauran algunos dbspaces y blobspaces. Se puede realizar una restauración de tipo mixta cuando se lleva a cabo una restauración completa, pero es necesario proveer acceso a una tabla en particular o a un conjunto de tablas tan pronto como sea posible, en este caso se

Figura 4.5 Respaldo de los archivos de logs lógicos con la utilería ontape

realiza una restauración en frío para restaurar los dbspaces críticos y los dbspaces que contienen las tablas importantes.

Una restauración en frío toma menos tiempo para restaurar todo el Servidor de datos de Informix, que una restauración de tipo mixta, siempre pensando que Informix está en modo on-line durante parte de una restauración de tipo mixta, ya que una restauración mixta requiere 2 restauraciones de tipo lógico (una restauración en caliente, y una en frío)

Para una restauración completa lógica y física del servidor dinámico, se utiliza la opción -r de la utilería ontape.

Para restaurar solo dbspaces y blobspaces seleccionados se utiliza la opción -D, como se muestra a continuación:

TESIS CON  
FALLA DE ORIGEN

**#ontape -r** (La utilería ontape realiza la restauración de datos tanto física como -lógica)

**#ontape -D dbspace**

(Dicha utilería restaura solo los dbspaces y blobspaces que se especifican como parte de una restauración en caliente)  
(*dbspace* es el nombre del dbspace o blobspace a restaurar)

**Nota:** Cuando no se especifica la opción -D, ontape realiza una restauración completa del sistema, por lo que Informix debe estar en modo off-line para poder llevar a cabo esta restauración.

Durante una restauración en frío no se puede reinicializar la memoria compartida, añadir chunks o cambiar los dispositivos de cinta, esto por que cuando se inicia la restauración, la actual configuración de Informix debe permanecer compatible.

Se debe de asegurar que los parámetros de memoria estén colocados hacia el valor máximo después de un respaldo de nivel 0. Por ejemplo, cuando se decrece el valor de USERTHREADS de 45 a 30 después de un respaldo de nivel 0, se debe iniciar una restauración con el valor de USERTHREADS colocado en 45 y no en 30, siempre pensando que la ultima copia del archivo de configuración para el último respaldo debió registrar el valor de USERTHREADS colocado en 30.

Se debe verificar que la actual configuración de espejo concuerde con la configuración que tuvo efecto a la hora de efectuar el ultimo respaldo de nivel 0. Debido a que Informix recomienda crear un respaldo de nivel 0, después de cada cambio hecho en la configuración de espejo, aunque no crea problemas. Los parámetros mas críticos son los parámetros de espejo que aparecen en el archivo de configuración de Informix, los cuales son MIRRORPATH y MIRRORFFSET.

Se recomienda también que los dispositivos crudos o raw devices y los archivos que son usados por el servidor (de los dbspaces y blobspaces que están siendo restaurados) después de un respaldo de nivel 0 estén disponibles.

La siguiente lista describe los pasos principales para una restauración en caliente.

Antes de iniciar una restauración, se deben de recolectar todas las cintas del último respaldo de nivel 0 que contiene los dbspaces y blobspaces que se están restaurando y algún subsecuente respaldo de nivel 1 o de nivel 2.

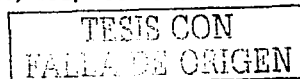
*Verificar la configuración de Informix*

Checar que se están utilizando las particiones crudas o archivos que son usados por Informix (de los dbspaces y blobspaces que están siendo restaurados) después del ultimo archivo de nivel 0.

*Respaldar los archivos de bitácora lógicos*

Antes de iniciar una restauración en caliente, se deben respaldar los archivos de bitácora.

Para llevar a cabo una restauración caliente se debe operar en modo online.



Debe ser el usuario informix o root quien utilice la utilidad `ontape`, dicha utilidad se debe utilizar como se muestra en el siguiente ejemplo, para restaurar `dbspaces` seleccionados y `dbspaces` que Informix maneja:

```
# ontape -r -D dbspace1 dbspace2
```

No es posible restaurar `dbspaces` críticos durante una restauración en caliente, se deben restaurar como parte de una restauración en frío.

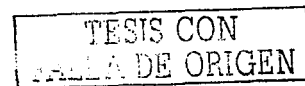
Durante la restauración, `ontape` indica el prompt para montar las cintas con los apropiados `dbspaces` o archivos de log.

Al final de la restauración en caliente, los `dbspaces` o `blobspaces` que estaban abajo se ponen en modo online

La utilidad `ontape` emplea parámetros definidos en el archivo `ONCONFIG` para definir el dispositivo de cinta para respaldos de bitácora lógicas. Sin embargo se deben considerar los siguientes puntos cuando se elige un dispositivo para respaldar una bitácora lógica:

- Cuando un dispositivo de bitácora lógico entra en conflicto con un archivo de dispositivo, se pueden planear los respaldos sin considerar las necesidades competentes de la agenda del archivo.
- Cuando se especifica `/dev/null` como un dispositivo de respaldo de bitácora lógico en el parámetro `ONCONFIG LTAPEDEV`, se evita tener que montar y mantener cintas de respaldo. Sin embargo, solo se puede recuperar datos arriba del punto de la cinta con el archivo mas reciente. No se puede recuperar el trabajo hecho después de archivarlo. Observe este aviso acerca de como manejar el `LTAPEDEV` en el `/dev/null` en "Cuando no se necesita recuperar".
- Cuando el dispositivo de cinta corre lento, la bitácora lógica podría llenarse mas rápido que si se hiciera una copia a la cinta. En este caso, se podría considerar el ejecutar el respaldo a disco y entonces copiar el disco respaldado a una cinta.

Las figuras 4.6 y 4.7 muestran el proceso de restauración física y restauración lógica respectivamente.



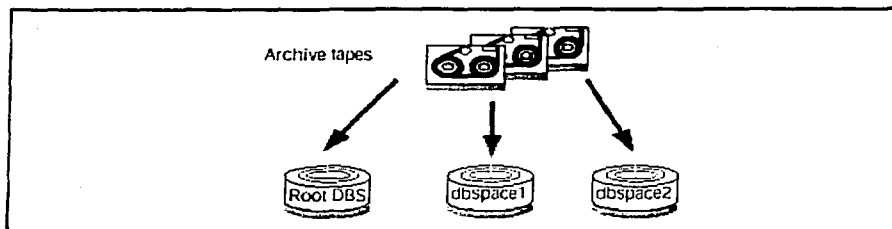


Figura 4.6 Restauración física con la utilidad ontape

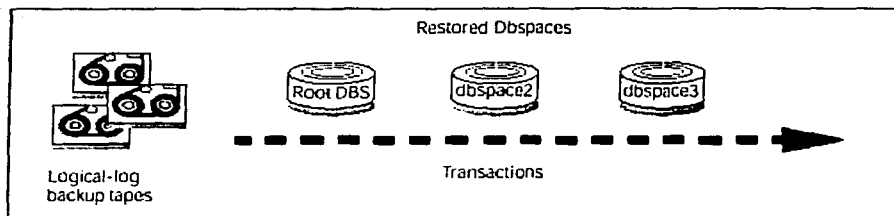


Figura 4.7 Restauración lógica con la utilidad ontape

### On-bar

ON-Bar es una utilidad para respaldar y restaurar el servidor de Informix, tanto en sistemas UNIX como en NT, se utiliza para hacer un respaldo de la base de datos y de las bitácoras lógicas y asegurarse nuevamente de la integridad, en caso de pérdida o corrupción de datos.

La corrupción de datos o pérdida de éstos puede ser por razones de falla en los programas ó falla en los discos y hasta un desastre en el lugar donde los equipos residen.

Para recobrar los datos, la restauración de la base de datos se hace en 2 pasos: Primero se restaura el respaldo de la copia de los datos, y después se restauran los logs para traer los datos y tenerlos en un estado lo mas cerca posible al mas reciente.





**Opciones:**

**Onbar.-** Esta utilidad es un shell script editable en UNIX y archivo batch en NT, que inicializa el driver de ON-Bar. Esta utilidad se emplea para checar la versión del storage manager y personalizar operaciones de respaldo y recuperaciones.

**Onbar\_d.-** Es el driver de ON-Bar el cual inicializa y controla actividades, respaldos y restauraciones.

Estas utilerías se pueden invocar desde la línea de comandos, un script o en forma calendarizada desde un cron.

On-bar se comunica con Informix y el Storage Manager (Figura 4.8). El driver de ON-Bar (onbar\_d) crea procesos hijos que efectúan operaciones de respaldo y recuperación.

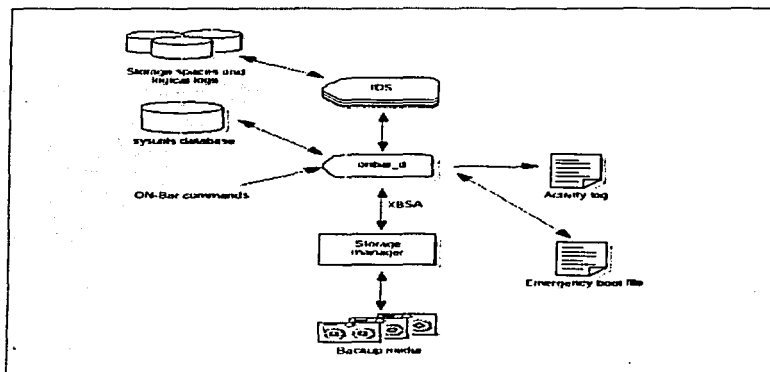


Figura 4.8 Componentes de la herramienta de respaldo Onbar

La utilería ON-bar respalda la siguiente información:

- Espacios de almacenamiento que contienen tablas e índices
- Archivos de logs lógicos los cuales contienen un registro de cada transacción que se realiza en la base de datos. Es posible respaldar los archivos de logs lógicos en forma separada o junto con los espacios de almacenamiento (áreas de datos). Se deben respaldar los logs lógicos tan pronto como se llenen para que puedan ser reutilizados.
- El catálogo de ISM (Informix Storage Manager), el cual contiene información acerca de los datos respaldados.

TESIS CON  
FALLA DE ORIGEN



Este catálogo se encuentra bajo la trayectoria **\$INFORMIXDIR/ism**

On-bar utiliza las siguientes tablas de catálogos en la base de datos *sysutils* para checar la compatibilidad de versiones de componentes.

A continuación se muestra una lista de tablas de la base de datos *sysutils*:

- La tabla *base\_server* toma fragmentos de instancias de Dynamic Server
- La tabla *bar\_object* toma objetos de respaldos. Un objeto de este tipo puede ser un *dbspace*, *blobospace* o archivo de bitácora lógica.
- La tabla *bar\_instance* describe cada objeto que es respaldado durante un respaldo satisfactorio
- La tabla *bar\_version* lista las versiones compatibles de ON-Bar y storage manager

Así como esta utilería respalda y restaura datos, periódicamente escribe hacia la bitácora de actividades. Cuando encuentra un error o una condición de emergencia, escribe un mensaje hacia esta bitácora. También anexa cuales espacios de almacenamiento y cuales archivos lógicos fueron incluidos en un respaldo (backup) o una operación de restauración y cuanto tiempo tomó esta.

Se puede especificar la localización de esta bitácora en el parámetro de configuración de *BAR\_ACT\_LOG* ó utilizar la localización por default */tmp/bar\_act.log*

Los siguientes parámetros se pueden colocar en el archivo *ONCONFIG* para informar al Administrador de eventos al momento de realizar un respaldo de este tipo.

### **ALARMPROGRAM**

Automáticamente respalda las bitácoras lógicas cuando se llenan. La utilería *onbar* provee un shell script llamado *log\_full.sh* que puede ser usado cuando inicia un respaldo de archivos de log lógicos cuando *Informix* funciona como una alarma de eventos de logs. El especificar una alarma de eventos es opcional a la hora de realizar un respaldo de logs lógicos y se utiliza el parámetro de configuración *ALARMPROGRAM*, y es colocado como sigue:

***\$INFORMIXDIR/etc/log\_full.sh*** para respaldar los logs lógicos automáticamente

Si no se desean respaldar estos logs, dicho parámetro debe ser colocado como sigue:

***\$INFORMIXDIR/etc/no\_log.s***

### **BAR\_ACT\_LOG**

Especifica la localización del archivo de bitácora de actividad de *ON-Bar*. Cuando sea que se lleve a cabo una actividad de respaldo, restauración o un error ocurra, *onbar* escribe una breve descripción en el log de actividad

Valor por default: */tmp/bar\_act.log*



**BAR\_BSALIB\_PATH**

Especifica la trayectoria de la librería del storage manager. Este parámetro es soportado solo en algunas plataformas. Para determinar si este parámetro está soportado en alguna plataforma en específico, se revisa en las especificaciones del sistema.

Valor por default: /usr/lib/ibsad001.platfform\_extension

**BAR\_MAX\_BACKUP**

Especifica el número máximo de procesos en paralelo por comando de ON-Bar. Por ejemplo, si este parámetro es colocado en 5 y se ejecutan 2 comandos de ON-Bar, el número máximo de procesos que onbar correrá concurrentemente es diez.

Valor por default	:	sin limite =0
unidades	:	procesos de onbar
rango de valores	:	>=0 a infinito
toma efecto	:	Cuando onbar inicializa

La utilidad onbar ignora este parámetro para un respaldo o restauración del sistema completo, ya que este tipo de respaldos son siempre seriales.

La cantidad de memoria disponible está basada en el parámetro de SHMTOTAL. ON-Bar realiza el siguiente cálculo, donde N es el número máximo de procesos de onbar que son presentados:

$$N = \text{SHMTOTAL} / (\# \text{ transport buffers} * \text{size of transport buffers} / 1024)$$

**BAR\_NB\_XPORT\_COUNT**

Especifica el número de buffers de datos de memoria compartida por cada proceso de onbar\_d.

El valor de este parámetro afecta el performance de ON-Bar. Por ejemplo, si BAR\_MAX\_BACKUP es colocado a 5 y BAR\_NB\_XPORT\_COUNT colocado a 5 consecuentemente funcionan 5 comandos de ON-Bar, resultando 25 procesos hijos de ON-Bar utilizarán un total de 125 buffers.

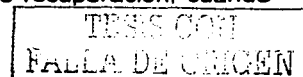
valor por default: 10

**BAR\_RETRY**

Especifica cuantas veces ON-Bar debe reintentar un respaldo de logs lógicos, u operaciones de restauración si el primer intento falla.

La forma en que sea colocado este parámetro determinará la forma en que ON-Bar actuará:

-Si es colocado hacia BAR\_ABORT, ON-Bar aborta el respaldo o recuperación, cuando un error ocurre, retorna el error y sale.



-Si es colocado como **BAR\_CONT**, ON-Bar aborta el respaldo o restauración por un espacio de almacenamiento en particular, retorna el error, e intenta respaldarlo o restaurarlo si el espacio de almacenamiento permanece.

-Si es colocado un número en específico, ON-Bar intenta respaldar o restaurar este espacio de almacenamiento el número de veces antes de que falle y se mueva hacia el siguiente número.

valor por default	:	1
rango de valores	:	BAR_ABORT (0), BAR_CONT (1), o n
toma efecto	:	Cuando onbar inicia

### **BAR\_XFER\_BUF\_SIZE**

Especifica el tamaño de página de los buffers que IDS (Informix Dinamic Server) utiliza para extraer datos de cada proceso de onbar o el tamaño de cada buffer de transferencia.

El servidor de base de datos pasa este buffer hacia ON-Bar y al storage manager. Para calcular el tamaño de transferencia del buffer en el espacio de almacenamiento o en el respaldo de logs se realiza la siguiente fórmula:

**BAR\_XFER\_BUF\_SIZE = PAGESIZE**

Para calcular cuanta memoria el servidor de base de datos necesita, utilizar la fórmula: **(BAR\_XFER\_BUF\_SIZE\* PAGESIZE) +500**

Los 500 extras son para overhead, por ejemplo, si **BAR\_XFER\_BUF\_SIZE** es de 15 el tamaño de transferencia del buffer debe ser de 61,940 bytes.

valor por default	:	15 cuando el PAGESIZE es de 4k 31 cuando el PAGESIZE es de 2 k
unidades	:	PAGESIZE
rango de valores	:	1 a 1,000.000 páginas
toma efecto	:	Cuando onbar inicia

### **LTAPEDEV**

Especifica el dispositivo de cinta donde los logs lógicos están respaldados, no colocar **LTAPEDEV** hacia /dev/null, ya que los respaldos de logs lógicos no trabajarán debido a que el servidor de base de datos, debe marcarlos como respaldados.

### **RESERTABLE\_RESTORE**

Coloca como apagado (off) este tipo de restauración.

valor por default:	OFF
rango de valores:	OFF (Si no se desea utilizar este tipo de restauración) ON (En caso contrario)
toma efecto:	Cuando el servidor de base de datos inicia



Un respaldo de este tipo, es un respaldo de todos los espacios de almacenamiento y todos los archivos de log. ON-Bar respalda los espacios de almacenamiento y dichos logs en forma serial durante un respaldo de este tipo. La ventaja de respaldar de esta forma es que puede ser utilizado para restaurar espacios de almacenamiento solamente.

No siempre se respaldan todas las tablas o todos los índices a la vez, por ejemplo si algunas tablas cambian pero otras no lo hacen, es ineficiente respaldar las tablas que no cambian cada vez que se respalda el servidor de base de datos.

Para proveer un ambiente de respaldo más flexible ON-Bar soporta 3 niveles de respaldo:

-*Nivel 0* respalda todos los datos en los espacios de almacenamiento especificados.

-*Nivel 1* respalda solo los datos que han cambiado desde el último respaldo de nivel 0 de los espacios de almacenamiento especificados.

-*Nivel 2* respalda solo los datos que han cambiado desde el último respaldo de nivel 1.

Un respaldo de tipo lógico copia todos los archivos de log lógicos completos que no hayan sido respaldados por el storage manager. Estos logs lógicos contienen registros de todos los cambios (checkpoints) que se efectuaron en una base de datos, durante el periodo en el que el log estuvo activo.

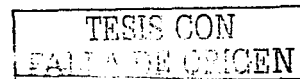
Informix continuamente escribe y salva nuevos registros de log lógicos en caso de que se deban restaurar estas transacciones.

Informix recomienda que se respalde cada archivo de log lógico tan pronto como este se llene. Para respaldar estos logs que se llenan periódicamente hay que configurar a ON-Bar para que se respalden en forma automática, de igual manera Informix reutiliza los archivos de log lógicos para minimizar la cantidad de espacio en disco requerida para las transacciones de logging .

Antes de crear un respaldo con ON-Bar, se debe configurar el ISM (o algún otro administrador de almacenamiento) en el servidor de bases de datos. Dicho administrador debe estar listo para recibir datos antes de que se comience hacer el respaldo o la recuperación, se debe reservar espacio en los dispositivos de almacenamiento y en los logical-log si es posible. Etiquetar y montar todas las cintas en el dispositivo de almacenamiento. El respaldo o la recuperación se encontrará en pausa hasta que no se monte la cinta requerida o el disco óptico.

Para asegurarse que se pueden recuperar datos, se necesita hacer un respaldo de nivel 0 del root dbspace cuando:

- Se haga un espejeo (mirroring)
- Se adicione un archivo de log lógico
- Se cambie el tamaño o la localidad de algún log físico
- Se de de baja un Chunk o dbspace



Para recuperar espacio o crear nuevos espacios de base de datos o archivos de bitácora lógicos, se debe hacer un respaldo de nivel 0 de todos los espacios de datos afectados cuando se hacen los siguientes cambios:

- Después de hacer cambios en la configuración del administrador del almacenamiento
- Después de adicionar un dbspace o blobspace y antes de recuperarlos
- Después de comenzar un espejeo para un dbspace que contiene los archivos de bitácora lógicos.
- Después de adicionar archivos de bitácora lógicos (para hacer accesibles los logs).
- Después de dejar caer un archivo de log lógico
- Después de mover uno o mas archivos de logs lógicos, y rechazar el antiguo archivo de log lógico, y adherir un nuevo archivo de logs lógico
- Después de cambiar el tamaño o localidad de un log físico y reinicializar la memoria compartida.
- Después de rechazar un chunk (Antes de eso podemos reusar el dbspace que contiene a dicho chunk).

Antes de crear un respaldo, se deben ejecutar las siguientes tareas para ayudar a asegurarnos que el respaldo haya sido exitoso:

- Asegurarse de que se tiene suficiente espacio en los logs lógicos para crear un respaldo.
- Imprimir o mantener una copia en esencia de la configuración del servidor de bases de datos.
- Verificar la consistencia de los datos.

Antes de respaldar espacios de almacenamiento, se debe asegurar que se tiene una copia del respaldo en curso de los siguientes archivos de configuración de la base de datos:

- Sqlhost
- Oncfg
- Archivo de boot de emergencia
- ONCONFIG
- Archivos del administrador de almacenamiento

Verificación de la consistencia de la información antes de hacer un respaldo de nivel 0

Para estar seguros de la integridad de los respaldos, periódicamente se tiene que verificar todas las bases de datos del servidor, obviamente antes de crear el respaldo de nivel 0. Por otro lado, no se necesita checar la consistencia antes de cada respaldo de nivel 0. Por su parte, Informix recomienda, sin embargo, que no se descarte un respaldo que fuera verificado en cuanto a su consistencia hasta la próxima vez que se verifique la consistencia de las bases de datos.



El servidor de bases de datos deberá estar en línea o en modo de esperar una petición, para ejecutar un respaldo. Usar la opción `onbar -b` para que automáticamente se respalden los storage spaces y los logical logs. Solo los espacios de almacenamiento que están en línea serán respaldados. Para determinar cuales son los espacios de almacenamiento que se encuentran en línea se utiliza la opción `-d` en la utilidad `cnstat`.

Importante : No se pueden respaldar o restaurar dbspaces temporales.

Cuando el servidor ISM recibe una petición de respaldo desde ON-Bar, muestra una etiqueta y monta las peticiones en la ventana de los dispositivos del programa administrador ISM.

Si se esta usando ISM, entonces `onbar -b` también respalda el catálogo de ISM el cual contiene información acerca de los datos respaldados. Durante la operación de respaldo, ISM crea puntos de salvamento (save sets) de los datos respaldados e inserta los registros del catálogo de ISM.

Para ejecutar un respaldo incremental (nivel 1), usa la opción `-L1`. Si no se especifica ningún nombre de los storage-space, entonces todos los espacios de almacenamiento serán respaldados:

```
Onbar -b -L 1G
```

Para ejecutar un respaldo completo de espacios de almacenamiento específicos (por ejemplo, dos dbspaces llamados `fin_dbspace1` y `fin_dbspace2`), se debe usar la opción `-b`, el siguiente ejemplo nos muestra la forma con se realizaría el respaldo:

```
Onbar -b fin_dbspace1 fin_dbspace2
```

Para respaldar una lista de espacios de almacenamiento, se usa el siguiente comando:

```
Onbar -b -f /usr/informix/backup_list/listfile3
```

La figura 4.9 esquematiza el proceso de respaldo de datos y logs con la herramienta `onbar`.

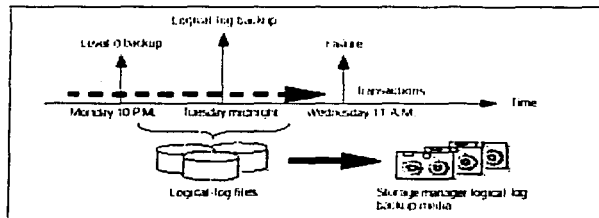


Figura 4.9 Respaldo de datos y logs con la herramienta Onbar

TESIS CON  
FALLA DE ORIGEN



Para realizar una restauración con la utilería On-bar se utilizan las siguientes opciones:

- Opción:**        **Propósito:**  
**-r**                Especifica una restauración.  
**Consideraciones:**  
 Ninguna.
- e**                **Propósito:**  
 Especifica una restauración externa.  
**Consideraciones:**  
 Debe ser usado con la opción `-r`
- t time**        **Propósito:**  
 Especifica el tiempo de la última transacción de una restauración de los archivos lógicos en una restauración en frío.  
**Consideraciones:**  
 Utilizar esta opción para restaurar las bases de datos hacia un estado muy cercano al último. Se deben restaurar todos los espacios de almacenamiento.
- p**                **Propósito:**  
 Especifica solamente una restauración física  
**Especificaciones:**  
 Esta opción debe ser seguida por una restauración lógica antes de que los datos sean accesibles. Esta opción apaga el log durante una restauración en frío.
- f file\_w\_names**  
**Propósito:**  
 Restaura los dbspaces o blobspaces que están listados (uno por línea) en el archivo de texto cuya trayectoria la provee `file_w_names`  
**Especificaciones:**  
 Esta opción se utiliza para evitar introducir una lista larga de dbspaces o blobspaces cada vez que se utilice esta opción .
- dbspace\_list** **Propósito:**  
 Nombres de uno o más dbspaces o blobspaces a ser restaurados  
**Consideraciones:**  
 Si no se especifica el `dbspace_list` o `-f file_w_names`, ON-Bar restaura todos los espacios de almacenamiento incluyendo los dbspaces críticos, si se desean introducir mas de un nombre de espacio de almacenamiento, utilizar un espacio para separar los nombres.

Ejemplos:

Para restaurar los chunks para todos los espacios de almacenamiento y las bitácoras lógicas utilizar el siguiente comando:

```
onbar -r -e
```





Si son creados varios respaldos de espacios de almacenamiento seleccionados, se deben restaurar ambos, los dbspaces y las bitácoras de log, para hacer esto se deben ejecutar los siguientes comandos:

```
onbar -or -e -p
onbar -r -l
```

La figura 4.10 ilustra el proceso de restauración con On-bar

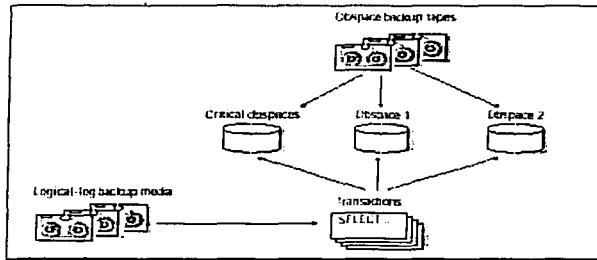


Figura 4.10 Proceso de restauración con la herramienta Onbar

### On-archive

La recuperación del Servidor de Informix, implica que se pueda recuperar tanto el Servidor, como los datos que llegan a ser corruptos o inaccesibles.

Un "archive" (respaldo) es una copia de cualquiera o todos los datos que IDS maneja, para ser mas precisos, un respaldo es una copia de uno o mas dbspaces. Se puede crear un respaldo de los datos del Servidor de Base de Datos en cinta o disco.

On-Archive opera de las 3 siguientes formas:

- Operaciones atendidas.- En este tipo de operación ON-Archive asume que un operador está trabajando tanto con Informix en modo en línea como con ON-Archive; dicho operador da la instrucción a ON-Archive para realizar el respaldo de los datos, o de las bitácoras lógicas. Además indica en la línea de comandos al operador la unidad de respaldo que se necesitará, cuando llegan a ocurrir errores se envían mensajes hacia la terminal para que el operador tome acciones.

TRABAJO CON  
FALLA DE ORIGEN



- Operaciones sin atender.- En este tipo de operaciones, un programa de ON-Archive llamado *onautovop* actúa como un operador virtual para efectuar los respaldos. El operador virtual asume que no existe un operador humano de por medio, por lo que omite el envío de una petición para montar una cinta cuando es requerida. En lugar de enviar mensajes de error hacia la terminal, el operador virtual informará a la persona designada acerca de los errores en forma electrónica, por ejemplo el correo electrónico (e-mail).
- Situaciones de emergencia.- Una situación de este tipo existe cuando Informix no puede acceder a un catálogo de ON-Archive, por lo que en esta situación el operador utiliza un programa de propósito especial llamado *ondatastr* para traer los datos de alguna cinta, disco ó archivos de bitácora lógica respaldados.

Un comando de ON-Archive que efectúa ya sea un respaldo o una operación de restauración, se distingue de otros comandos, ya que no se ejecutan inmediatamente. Estos comandos especiales llamados *requests* se almacenan en un catálogo de ON-Archive para ser ejecutados posteriormente.

Lo siguiente son programas que componen On-Archive

- onarchive*.- Este programa habilita, el poder ejecutar comandos de ON-Archive; y provee ambas interfaces, tanto la línea de comandos, como un menú de interface. Ambas interfaces proveen la misma funcionalidad a ON-Archive.
- ontalgr*.- Este programa es un demonio como proceso que ON-Archive utiliza para acceder a los catálogos. Este proceso mantiene una conexión abierta hacia la base de datos *sysmaster* donde las tablas de los catálogos de ON-Archive residen. Este proceso debe estar corriendo para utilizar ON-Archive en operaciones atendidas o sin atender.
- onautovop*.- Este programa actúa como un operador virtual. Ejecuta requerimientos de ON-Archive en forma calendarizada sin requerir intervención humana. Cuando *onautovop* ejecuta requerimientos, estos son llamados "operaciones sin atender".
- ondatastr*.- Esta utilidad permite al usuario *root* o *informix* ejecutar operaciones en situaciones de emergencia cuando no se utilice ON-Archive debido a que los catálogos no están disponibles. Esta utilidad utiliza un lenguaje de comandos similar al de *onarchive* pero tiene un menú como interface.

Este catálogo es parte central de las operaciones de ON-Archive, el cual es un conjunto de tablas de bases de datos que son parte de la base de datos *sysmaster*. Las tablas de catálogos de ON-Archive se almacenan en disco como tablas regulares de IDS, los cuales mantienen información acerca de datos y operaciones de ON-Archive, por ejemplo:

- \*Definiciones de conjunto de volúmenes (forma en como agrupa los respaldos)
- \*Un registro de archivos y requerimientos de respaldos
- \*Localizaciones de volúmenes de cintas



El archivo de configuración para ON-Archive es el archivo *config.arc* localizado en el directorio \$INFORMIXDIR/etc

Cuando se desea crear y utilizar diferentes archivos de configuración para ON-Archive, se puede colocar la variable de ambiente *ARC\_CONFIG* con el nombre del archivo e IDS busca dicho archivo, que es especificado en *ARC\_CONFIG* en el directorio \$INFORMIXDIR/etc. Esta variable de ambiente da la posibilidad de que se cambien parámetros de configuración mientras se está preservando el archivo por default *config.arc*.

ON-Archive también provee un archivo llamado *operdeflt.arc* que define valores por default para calificadores de comandos utilizados comunmente.

La siguiente tabla define los parámetros en el archivo de configuración ON-Archive:

Parámetro	Definición
CATALOG MESSAGE	Especifica el archivo de mensajes oncatlgr
DEVICE	Especifica los dispositivos que ON-Archive puede utilizar para escribir archivos y respaldos de bitácoras lógicas
DEFAULT	Especifica el nombre del archivo que contiene defaults para calificadores de comandos
ENGLISH	Indica que el lenguaje, por default es el inglés.
ERROR	Especifica los nombres de los archivos de mensajes de error que ON-Archive utiliza
HELP	Especifica el nombre del archivo de ayuda de onarchive
KEYM_HELP	Especifica los archivos de ayuda para onkeymgr
MESSAGE	Especifica los nombres de archivos de formato
NB_DISK_SPACE_EXTENT	Define la unidad de localización de espacio en volúmenes de disco
PRIVILEGE	Indica las restricciones que ON-Archive utiliza
TIME_OUT	Especifica un valor de tiempo fuera.

Tabla. Parámetros de Configuración para la utilería On-Archive

Ejemplos:

**CATALOG MESSAGE:**

CATALOG MESSAGE = -----pathname-----  
 pathname es una trayectoria relativa de un archivo  
 CATALOG MESSAGE = /msg/cm\_exnglish.dat

**DEFAULT.-**

DEFAULT = /etc/dflts.arc

**DEVICE.-**

DEVICE = logical device = physical device  
 donde:

TESIS CON  
 FALLA DE ORIGEN



- logical\_device.-** Es un nombre que el usuario asocia a un grupo de dispositivos físicos.
- physical\_device.-** Es una especificación de un dispositivo físico asociado con el nombre de un dispositivo lógico. La trayectoria hacia el dispositivo físico puede ser una liga.

Este parámetro soporta los siguientes dispositivos de cinta:

Dispositivos de cinta	
DAT	(4mm digital audio tape)
Exabyte	(8-mm Helical scan)
9-track	(half-inch reel to reel)
QIC	(quarter-inch cartridge)

**ENGLISH.-** Provee opciones de lenguaje para inglés y francés.

**ERROR.-**

El siguiente ejemplo ilustra este parámetro en inglés y español.

```
ERROR ENGLISH = /arc_fsl/ee.dat
ERROR FRENCH  = /arc_fsl/ef.cat
```

**HELP.-** Este parámetro especifica la localización de archivos de punto de ayuda

**KEYM\_HELP.-**

EL siguiente ejemplo ilustra el parámetro de KEYM\_HELP en inglés y francés:

```
KEYM_HELPENGLISH = /arc_fsl/ke.hpf
KEYM_HELPFRENCH  = /arc_fsl/kf.hpf
```

**MESSAGE.-** Este parámetro especifica la localización de los archivos de mensajes

**NB\_DISK\_SPACE\_EXTENT.-**

```
NB_DISK_SPACE_EXTENT = -----integer-----
```

**Integer.-** Especifica un valor en el rango de 1 a 100 que cuando se multiplica por el valor del calificador BLOCKSIZE, determina cuanto espacio se puede alojar en un volumen de disco cada vez que se requiere mas espacio.

Ejemplo de uso de este parámetro:

```
NB_DISK_SPACE_EXTENT = 30
```

**PRIVILEGE.-** Cuando se define este modo, se puede limitar accesos a ON-Archive. La sintaxis de uso es la siguiente:



PRIVILEGE = OPERATOR

**OPERATOR.-**

En el siguiente ejemplo ON\_Archive otorga acceso a root, al usuario informix y a todos los miembros del grupo.

PRIVILEGE = OPERATOR, GROUP

*Nota:* Como todos los parámetros, cada vez que el parámetro PRIVILEGE sea modificado, se debe detener y reiniciar **oncatlgr** para que los cambios tomen efecto.

**TIME\_OUT.-**

TIMEOUT = integer

integer.- Especifica el número de minutos antes de una condición de time-out.

En el siguiente ejemplo ON-Archive espera 15 minutos a que el operador monte un volumen en un dispositivo físico, si dicho montaje permanece incompleto después de 15 minutos, ON-Archive termina:

TIMEOUT = 15

*Funcionamiento de ON-Archive*

Se debe inicializar el programa *oncatlgr*, también conocido como *catalogador*, antes de que se inicialicen operaciones atendidas u operaciones sin atender. Informix debe estar en modo on-line para inicializar *oncatlgr*. Con el usuario root o informix, se ejecuta el siguiente comando en el prompt para inicializar este programa:

```
% start_oncatlgr
```

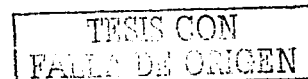
Se debe estar seguro de que hay solo un catalogador corriendo para cada instancia de Informix. Cuando se inicializa el catalogador, primero se checa la existencia de algún proceso de *oncatlgr*. Cuando otro proceso de *oncatlgr* está corriendo, la utilería *start\_oncatlgr* pregunta si se desean detener los procesos de catalogadores existentes.

Cuando múltiples instancias de Servidor de Base de Datos residen en una computadora, se puede determinar cuáles procesos de *oncatlgr* están asociados con una instancia en particular de Servidor de BD examinando el archivo */tmp/oncatlgr.out.pid* donde pid es el identificador de proceso para cada proceso de *oncatlgr*.

Este archivo contiene el número de instancia de Informix Dynamic Server con su particular proceso de *oncatlgr*.

```
%grep Server /tmp/oncatlgr.out
```

Debido a que es necesario correr *oncatlgr* antes de ejecutar cualquier comando de ON-Archive. Se puede asegurar que esté corriendo cuando sea que IDS esté corriendo. Para



inicializar oncatlgr automáticamente cuando se inicializa Informix, se agrega la siguiente línea en los archivos de inicialización:

```
% INFORMIXDIR/bin/start_oncatlgr
```

#### *Detener el Catalogador*

El catalogador debe ser apagado antes de que Informix este fuera de línea. Para detener el proceso catalogador, se ejecuta el siguiente comando con el usuario ó root informix:

```
%stop_oncatlgr
```

Cuando son múltiples instancias de Informix las que están corriendo, en un equipo, se debe tener cuidado de dar de baja solo aquellos procesos asociados con cada instancia de Informix que está siendo usada.

#### *Inicializando ON-Archive*

Una vez ya inicializado el catalogador, se puede inicializar la utilería *onarchive* que habilita el poder ejecutar comandos desde la línea tradicional o a través de un menú de interface. No se requieren privilegios especiales para correr onarchive, aunque los comandos llegan a ser diferentes dependiendo de los privilegios de ON-Archive.

Con el siguiente comando se inicializa esta utilería:

```
%onarchive
```

El siguiente mensaje aparece cuando se inicializa onarchive con la línea de comandos (el texto puede diferir dependiendo del sistema)

```
% onarchive
ON-Archive: Version 9.2
Copyrigh © 1994 – Informix Software, Inc.
Onarchive>
```

El prompt de *Onarchive>* indica que pueden ser introducidos comandos de ON-Archive, por ejemplo:

```
Onarchive>ARCHIVE/DBSPACESET=*
```

On-Archive procesa el comando y luego despliega un nuevo prompt para indicar que hay cabida para un nuevo comando.

EXIT.- Este comando termina la interacción con ON-Archive, este comando no tiene restricciones.



*Creando y ejecutando Requests*

Los requests son comandos que no se ejecutan inmediatamente, en lugar de esto, el comando generado es almacenado en los catálogos de ON-Archive. Se Pueden ejecutar comandos en forma explícita utilizando este comando o en forma implícita utilizando el operador virtual, *onaautop*.

Los requests se ejecutan de la misma manera que otro tipo de comandos. Los siguientes comandos de ON-Archive pueden llegar a ser requests:

- ARCHIVE
- BACKUP
- COPY
- RETRIEVE/DBSPACESET
- RETRIEVE/LOGFILE
- REMOVE with WAIT

Cuando un request es almacenado, se le asocia un id request (rid) Por ejemplo:

```
Onarchive>ARCHIVE/DBSPACESET=cust_set/LEVEL=0/SET=cust
```

ON-Archive responde con un mensaje como se muestra a continuación

```
Request 0000015 registered in the catalog
```

EL dueño del request es el usuario mismo que lo creó, el dueño no es necesariamente la misma persona que lo ejecuta; se tienen además dos formas de ejecutar un request:

- En forma manual
- En forma automática

Las siguientes restricciones aplican para el comando EXECUTE:

- Cuando ON-Archive está corriendo con privilegios de Operador, el usuario debe logearse como el usuario informix o el usuario root, Estos usuarios pueden ejecutar requests como cualquier usuario creado.
- Cuando ON-Archive está corriendo con privilegios de GROUP, el usuario debe firmarse en el sistema como parte del grupo super\_archive, los usuarios en este grupo pueden ejecutar requests como cualquier usuario perteneciente al grupo super\_archive

**Status de Requests**

Un request siempre tiene uno de los siguientes status:

CANCELED  
COMPLETED

TESIS CON  
FALLA DE ORIGEN



EXECUTING  
 FAILED  
 NEW  
 PARENT  
 UNCOMPLETED

Cuando un error ocurre o un requerimiento es ejecutado, ON-Archive reporta el error, que depende de cómo fue ejecutado el requerimiento. La siguiente lista describe las diferentes formas en que ON-Archive reporta errores:

- Cuando onarchive ejecuta un requerimiento, ON-Archive envía un correo de error hacia la terminal.
- Cuando onautopop ejecuta el requerimiento, ON-Archive envía un mensaje de error hacia el dueño del requerimiento.
- Cuando un requerimiento incluye el calificador de LOG, ON-Archive escribe el error hacia el archivo arcxxx.log en el actual directorio de trabajo, donde xxx es un ID de requerimiento.

#### *Conjunto de Volúmenes y Volúmenes*

Antes de que se lleve a cabo un respaldo de bitácoras lógicas, se debe definir los volume sets y volúmenes que ON-Archive utiliza para almacenar información. Se utilizan los comandos DEFINE/VSET y DEFINE/VOLUME para crear vsets y volúmenes.

Las siguientes restricciones aplican para el comando DEFINE/VSET

- Cuando ON/Archive está corriendo en modo OPERADOR, solo el usuario informix o root pueden definir un vset.
- De igual manera, si está corriendo con privilegios de GROUP, solo usuarios que están en el grupo super\_archive podrán definir un vset.

Cuando se define un vset, se debe proveer valores para los siguientes calificadores:

- VSET
- CLASS
- DEVICE\_TYPE
- DRIVER
- ACCESABILITY

La siguiente lista muestra calificadores opcionales, los cuales ofrecen funciones especiales como protección de almacenamiento, área temporal de almacenamiento de overflow, descripción de localización e intercambio de datos entre 2 servidores de Informix Dynamic Server.

1. PROTECTION
2. PERMANENTLY\_MOUNTED
3. LOCATION
4. TRANSIT





## 5. IMPORTED

### Ejemplos de DEFINE/VSET

El siguiente ejemplo define un vset llamado cust que es un vset de sistema, consiste de volúmenes de cinta, tiene alta accesabilidad, y asigna el vset a un dispositivo lógico llamado tape:

```
Onarchive>DEFINE/VSET=cust/CLASS=SYS/DRIVER=TAPE/ACCESS=O/DEVICE_TYPE=tape
```

El siguiente ejemplo define un vset llamado test, que es un vset de usuario accesible solo para los usuarios "pat" y "terry". Es un vset de disco asignado al dispositivo lógico llamado HO:

```
Onarchive>DEFINE/VSET=test/ACCESABILITY=30/DEVICE_TYPE=HO/CLASS=USER-/USER      =
(pat,terry) / DRIVER=DISK
```

ON-Archive inicializa volúmenes no importados que son parte de un vset definido con DRIVER=TAPE o parte de un vset definido con DRIVER=DISK y el calificador NOPERMANENTLY\_MOUNTED.

ON-Archive no inicializa un volumen que pertenezca a otro vset definido con el calificador IMPORTED

Cuando ON-Archive inicializa un volumen, se indica el prompt para seleccionar uno de los dispositivos físicos definidos para los dispositivos lógicos que se especificaron para el vset. En el siguiente ejemplo, ON-Archive pregunta al operador para seleccionar uno de los dos dispositivos de cinta asociados con el dispositivo lógico.

Devices available for ONARCHIVE

DEVICE # DEVICE NAME

```
1          /dev/rst0
2          /dev/rst1
```

Enter one of the device numbers (1-2) or (C) cancel : 1

Aquí se selecciona uno de los dispositivos especificados

Nota : On-archive permite que se inicialicen volúmenes que actualmente contienen datos, pero siempre hay que pensar en este punto, y no inicializar (definir) un volumen que pudiera contener datos que se pudieran necesitar para una restauración por que podrían ya estar inaccesibles.

TESIS CON  
FALLA DE ORIGEN



Estas son las herramientas de respaldo que proporciona el servidor de Informix Dynamix Server para el administrador.

Como última herramienta de respaldo tenemos Informix Server Administrator que es una herramienta visual que permite ejecutar varias de las herramientas antes mencionadas.

#### IV.5 ISA (INFORMIX SERVER ADMINISTRATOR)

Informix Server Administrator permite administrar bases de datos, ejecutando comandos desde un browser. No se necesita estar familiarizado con la sintaxis y el formato de los comandos, ISA presenta la salida de los comandos en un formato fácil de lectura.

Con ISA se pueden realizar las siguientes tareas administrativas:

- Cambiar parámetros de configuración en forma temporal o permanente
- Utilizar el setup para configurar o reconfigurar el servidor de base de datos
- Cambiar el modo de operación del Servidor
- Modificar la información de conectividad en el servidor en el archivo sqlhost
- Verificar los dbspaces, blobspaces y sbspaces
- Administrar los logs lógicos y físicos
- Examinar y modificar la memoria utilizada
- Leer los mensajes de log
- Restaurar y respaldar los dbspaces y sbspaces
- Correr varios comandos de onstat para monitorear el performance
- Añadir y remover chunks, dbspaces, blobspaces y sbspaces
- Examinar y administrar las sesiones de los usuarios
- Examinar y administrar los procesadores virtuales
- Utilizar el cargador de alto performance (HPL), dbimport y dbexport
- Configurar un servidor base de datos primario y secundario para la replicación de datos en alta disponibilidad
- Utilizar las siguientes utilerías: dbaccess, dbschema, onbar, oncheck, ondblog, oninit, onlog, onmode, onparams, onspaces, onstat, onpladm.

En el proceso de administración de cualquier servidor de base de datos es indispensable tener un procedimiento de respaldo con el cual se asegure la información contenida en las bases de datos, el sistema SIAPSEP contiene la información de miles de maestros por lo cual es necesario tener un esquema de respaldo seguro y confiable Informix proporciona varias opciones, los respaldos que se utilizan actualmente para el servidor de base de datos de la nómina es el ontape y onbar, pero cualquiera de la opciones antes mencionadas son muy aplicables como por ejemplo para cuando se requiere respaldar una base de datos en específico (dbexport) ó alguna tabla que se verá afecta en un proceso (unload), todo dependerá de las necesidades del usuario, una parte muy importante que debe tener el administrador es una comunicación tanto como con el administrador del S.O. y el del Sistema SIAPSEP para poder de acuerdo a los requerimientos de cada uno de estos los datos disponibles.

TESIS CON  
FALLA DE ORIGEN

En el siguiente tema se hablará acerca de la seguridad del sistema operativo unix y el manejador de bases de datos informix, ya que la seguridad es un punto que no debe de ser descuidado en ningún sistema.



## TEMA V

### SEGURIDAD

La seguridad en todos los sistemas y además parte fundamental en la administración del mismo, como lo es el sistema SIAPSEP engloba varios aspectos, como son: seguridad en la red, seguridad en el sistema operativo donde convive la aplicación, seguridad en la base de datos del sistema, así como la seguridad de la aplicación misma, todo con el fin común de proteger la información contenida en los servidores de Base de Datos.

Es muy importante no descuidar alguno de estos puntos ya que, el tomar en cuenta cada uno de ellos, robustece de manera significativa la aplicación haciendola menos susceptible a una intrusión a cualquier nivel de seguridad de un sistema.

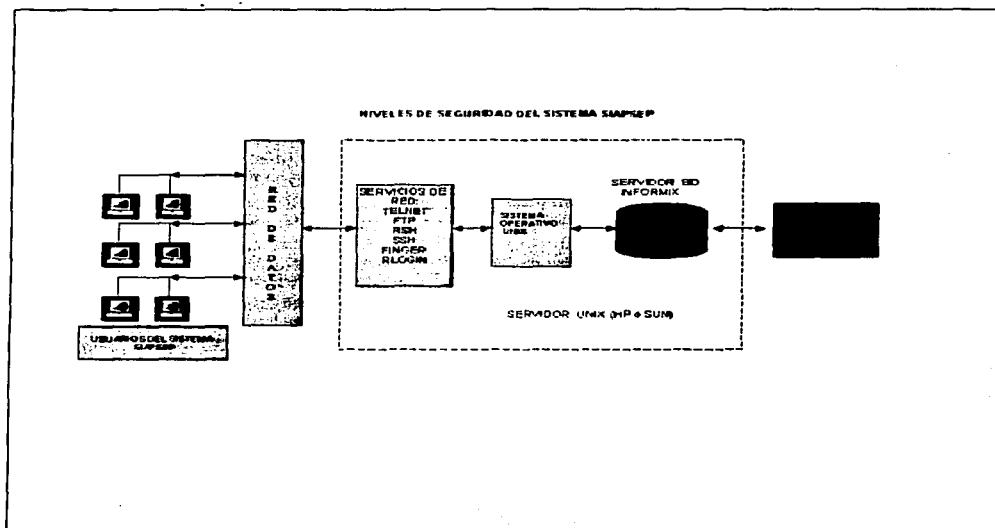


Figura 5.1 Niveles de Seguridad en el Sistema SIAPSEP

TESIS CON  
FALLA DE ORIGEN



La figura 5.1 esquematiza los niveles de seguridad por los que debe pasar un usuario para poder llegar a la aplicación SIAPSEP, que son: Seguridad en la Red de Datos, Seguridad en el Sistema Operativo, Seguridad en el Servidor de Base de Datos y Seguridad del SIAPSEP.

Comenzaremos hablando de la seguridad en la red de datos, la cual va de la mano con la seguridad en el sistema operativo de red, en este caso el sistema operativo Unix.

## V.1 SEGURIDAD EN LA RED DE DATOS

### Supervisión de la red

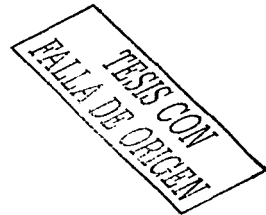
En lo que a seguridad en la red se refiere, existen diferentes tareas que lleva a cabo el administrador de la red, como lo es la supervisión de la misma, para lo cual existen diferentes formas para llevarla a cabo, así como diferentes tipos de software los cuales facilitan en gran medida esta labor tan importante.

A continuación se mencionan algunas herramientas que el sistema operativo brinda para asegurar el buen funcionamiento de la red del sistema.

El programa syslog

El syslog, es un mecanismo mediante el cual se manda cualquier mensaje de error a la consola del sistema. Normalmente los mensajes se almacenan en la trayectoria /usr/adm/messages, con la fecha, el tiempo que aparecieron, nombre del programa que envió el mensaje, y el identificador del proceso. Un ejemplo de salida de este archivo se presenta a continuación:

```
Oct 26 10:48:21 ribera vmunix: esp0 at SBus slot f 0x800000 pri 4 (onboard)
Oct 28 13:13:59 ribera vmunix: fd0: drive not ready Oct 28 13:51:13 ribera
vmunix: panic on cpu 0: allocgbik: can&acute; find blk in cyl Oct 31
08:17:28 ribera vmunix: sd0 at esp0 target 3 lun 0 Oct 31 08:17:28 ribera
vmunix: st0 at esp0 target 4 lun 0 Nov 6 11:28:49 ribera vmunix: root on sd0a
fstype 4.2 Nov 16 19:36:48 ribera su: &acute;u root& acute;succeeded for
juaase on /dev/ttyq3 Nov 17 11:52:03 ribera vmunix: NFS write error: on host
palomino remote file system full
```



Esta opción puede ser de gran utilidad para el administrador ya que los reportes generados por el sistema SIAPSEP son almacenados en el área del usuario que está enviando el reporte, por lo que en algún momento puede llenarse esa área designada por lo tanto el administrador podría identificar el error de que ya no puede realizarse la transferencia del reporte a la cuenta del usuario y comentar al administrador del sistema



operativo para que lleve a cabo la depuración o agregar más espacio según lo que se determine.

La figura 5.2 esquematiza la funcionalidad anterior del programa syslog aplicado a la parte de reportes del sistema SIAPSEP:

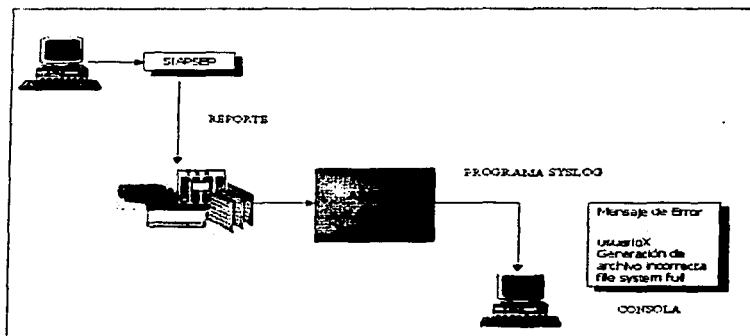


Figura 5.2 Procedimiento de la utilidad syslog

### Programas login y su

Son de particular interés los mensajes de los programas login y su. Se debe suprimir la posibilidad de que alguien entre directamente como superusuario (root), porque no se conoce quién es el que está usando la cuenta. También se almacena cualquier caso en el que una persona intente repetidas veces entrar en una determinada cuenta sin éxito. Si se encuentran mensajes de estos, es posible que haya un hacker intentando descubrir los passwords de los usuarios.

También almacena los momentos en los que se usa el comando su, para superusuario (root) o para otro nombre de usuario. Esta bitácora se encuentra en el sistema en la trayectoria /var/adm/sulog, en el cual se puede observar si hay usuarios que comparten los passwords, o para ver si existe algún usuario que ha entrado en el sistema y pretende acceder a otras cuentas.

Para tareas de administración de la base de datos del sistema SIAPSEP, los programas su y login son muy importantes, ya que los administradores, tanto del sistema operativo,

TESIS CON  
FALLA DE ORIGEN



como de la base de datos, en ocasiones requieren hacer uso de estos programas para realizar tareas en las que se involucran el manejo de puertos, ya sea que la base de datos utiliza o de subir o dar de baja la aplicación. Es recomendable que los usuarios y operadores del sistema SIAPSEP no tengan acceso a estas herramientas para evitar que un usuario pueda acceder a otra cuenta desde su usuario los únicos que deben tener acceso a estos programas son root e informix.

### *NFS (Sistemas de archivos en Red)*

El trabajar con sistema de archivos en red es muy importante, debido a que en ocasiones, es necesario compartir un recurso de nuestro sistema por la red hacia otro host, en el caso del SIAPSEP cuando se necesita respaldar algún dbspace o instalar algún tipo de software, si nuestro host no cuenta con unidad de CDROM, o espacio disponible, es posible compartir el recurso de otro host que se encuentre en nuestra red, solo que esto representa un factor de riesgo, ya que el recurso está disponible hacia los demás usuarios de la red.

Este esquema es utilizado actualmente por el sistema SIAPSEP para compartir las pantallas (front-end) de la aplicación, de donde todos los clientes se conectan para poder acceder a la aplicación.

Para tener este acceso a un NFS, se instala un programa denominado interdrive el cual permite que un usuario se conecte al área NFS designada en el sistema operativo desde su PC (Figura 5.3).

La figura 5.3 representa el uso de esta propiedad de sistemas de archivo en red.

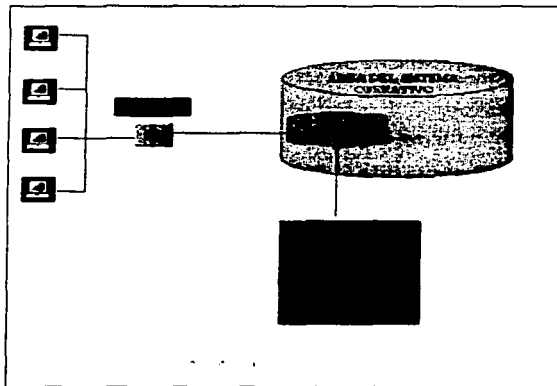


Figura 5.3 Sistemas de Archivos en Red

TESIS CON  
FALLA DE ORIGEN



Cabe señalar que esta parte en lo que a seguridad se refiere, recae en las propiedades o permisos que designen a los archivos ó directorios en este caso las pantallas del SIAPSEP.

### *showmount*

Este comando se puede usar en un servidor de NFS (Network File System), para ver todos los hosts que tienen algún sistema de archivos montado sobre el servidor. Con la opción -a se muestra tanto el host, como los directorios.

```

cigales://lostrego/home2                nava://lostrego/usrl
palomino://lostrego/home1
pesquera://lostrego/usrl protos:/usr/include rueda:/home/lostrego
tierra://lostrego/usrl                  verdejo:/home/lostrego
viura://lostrego/home2

```

La salida de showmount puede usarse para dos asuntos, la primera es que sólo máquinas locales a nuestro sistema deben estar presentes. Sólo se deben encontrar los directorios habituales. Si existen directorios que no lo son hay que encontrar quién los está montando y por qué, aunque es una manera inocente de entrar en el sistema, puede significar un intento de rotura de la seguridad.

### **Seguridad del DNS**

El servicio DNS (Domain Name System), es un amplio conjunto de bases de datos distribuidas, usado a lo largo de la internet, proporcionando correspondencia entre los nombres de host y las direcciones de IP de los mismos. Existe la posibilidad de abusar de este servicio para poder entrar en un sistema. La autenticación se lleva a cabo en muchos casos a partir del nombre del host o la dirección Ip. Las aplicaciones de alto nivel, usan en la mayoría de los casos los nombres para la autenticación, puesto que las tablas de direcciones son mucho más difíciles de crear, entender y mantener. Si por ejemplo alguien quiere suplantar una máquina por otra no tiene más que cambiar una de las entradas de la tabla que relaciona su nombre con su dirección. Este es el problema fundamental de DNS.

Actualmente se está llevando a cabo la implementación de autenticación para el sistema SIAPSEP en el cual se requiere un DNS, en donde se encuentra el servidor de autenticación.

### *El servidor inetd*

Este programa controla todos los servicios de red del sistema y tiene toda una lista de servicios a la espera de recibir respuesta de sus peticiones. Algunos servicios pueden deshabilitarse para mejorar la seguridad. Los servicios que se quieren dar se incluyen en el archivo /etc/inetd.conf. La mayor parte de ellos corren como root y por tanto si algún usuario consigue tomar el control de este servidor el resultado puede ser serio.

TESIS CON  
FALLA DE ORIGEN



Algunos de los servicios más importantes del servidor inetd son:

*in.fingerd:*

Permite a los usuarios remotos listar todos aquellos conectados al sistema. Los hackers lo utilizan para ver si el administrador del sistema está en él.

*rcp.rexd:*

Permite a los usuarios ejecutar comandos en máquinas remotas con muy poca autenticación. Es mejor usar en su lugar /usr/bin/rsh que es mucho más seguro. El SIAPSEP para la ejecución de programas o procesos desde el cliente utiliza esta herramienta ya que desde el cliente se ejecutan procesos de manera remota, el producto con el que se cuenta es el REXEC.

*in.tftpd:*

Una versión reducida de ftp, con poca autenticación.

*adminda:*

Como este programa es el que se encarga de administrar los servicios de red en un ambiente Cliente/Servidor, para el caso de nuestro sistema SIAPSEP, es recomendable solo habilitar los servicios de TCP/IP que sean necesarios, y deshabilitar los que se consideran vulnerables como los que se mencionaron anteriormente, por ejemplo:

*El demonio in.routed*

Este demonio gestiona automáticamente las tablas de enrutamiento. Se intercambia con otras máquinas de la red intercambiando información de enrutamiento, usando el protocolo RIP (Routing Information Protocol). Lo que hace es indicar a las máquinas que lo solicitan, la dirección de aquellas que conoce. Si la máquina tiene una única red conectada no es necesario que esté corriendo. Es mucho mejor decir, en cuanto a mejora del rendimiento, a la máquina explícitamente dónde enviar los paquetes no locales. Esto se consigue estableciendo un router por defecto:

```
/sbin/route -f add default <comms_server_name> 1
```

Desde el punto de vista de, seguridad esta no es la forma idónea. Si se usa más de un gateway se puede mantener manualmente la tabla de enrutamiento con:

```
/sbin/route add net <red> <gateway> 1
```

Donde <red> es uno de los nombres de /etc/networks , y <gateway> es el nombre de la máquina local que conecta la red.

<p>TESIS CON FALLA DE ORIGEN</p>
--------------------------------------



### *FTP y Telnet*

El protocolo de transferencia de archivos (FTP), se implementa mediante los programas ftp y ftpd. Las viejas versiones de los mismos presentaban ciertos fallos de seguridad.

Una de las posibilidades que ofrece este servicio es el FTP anónimo, para la distribución de software público.

Debido a que estos servicios presentan muchas vulnerabilidades, se tienen hoy en día herramientas, las cuales sustituyen a estos servicios de red, pero que además de proporcionar el mismo servicio, proporciona un nivel de seguridad muy alto en el sistema al pasar la información de manera cifrada, sobre un canal inseguro, como lo es la red de datos, y que en el caso del SIAPSEP por ejemplo, al hacer una consulta hacia una tabla o hacer una inserción en algún campo de una tabla, si hubiese algún tipo de dispositivo o software que estuviese rastreando la información que está pasando sobre la red, no habría manera de conocer la información que se está solicitando al hacer una consulta al sistema, o algún tipo de conexión con la base de datos del SIAPSEP.

El uso de los programas anteriores es común en cualquier sistema por lo que son susceptibles de la manipulación de cualquier usuario por ello requieren del monitoreo o procesos especiales para mantener la seguridad, a continuación se presentan algunas herramientas que permiten monitorear dichos programas.

#### *Detección de intrusos*

Dentro de las labores de supervisión se encuentra la detección de actos que supongan una violación de la seguridad del sistema y atenten contra la disponibilidad, integridad, o privacidad de la información y recursos del sistema. A esto se le conoce como detección de intrusos y se constituye como la última barrera de protección en un sistema de seguridad.

Existen infinidad de políticas y procedimientos, que pueden facilitar la administración de seguridad, así como herramientas específicas que pueden llevar una bitácora e incluso negar el acceso a ciertos usuarios hostiles para el servidor, existen herramientas capaces de "esconder" con gran eficacia los puntos críticos de nuestra seguridad como el password de root, dos herramientas que actualmente se utilizan en lugares tan importantes como universidades, centros de investigación espacial, o gubernamental son TCP-wrapper y SSH, herramientas "básicas" para comenzar con la seguridad en los servidores donde conviven nuestros sistemas, en especial nuestro sistema SIAPSEP.

No podemos decir que con la implantación de estas herramientas estamos 100 % seguros de cualquier ataque, un hacker siempre va a encontrar alguna manera de entrar a un sistema, pero sin embargo podemos aumentar las medidas de seguridad para que la intrusión sea más difícil, detectada, registrada y de ser posible detenida.

#### *TCP-Wrapper*

TCP-wrapper es una herramienta que permite controlar y proteger los servicios de red y algunos mas, limitando el acceso como sea posible, registrando todas las



conexiones para hacer el trabajo de detectar y resolver problemas de seguridad más eficientemente.

Un wrapper es un programa que literalmente cubre la identidad de un segundo programa, obteniendo con esto un más alto nivel de seguridad. Estos programas nacieron por la necesidad de modificar el comportamiento del sistema operativo sin tener que modificar su funcionamiento.

Los Wrappers son ampliamente utilizados, y han llegado a formar parte de herramientas de seguridad por las siguientes razones:

- Debido a que la seguridad lógica esta concentrada en un solo programa, los Wrappers son fáciles y simples de validar.
- Debido a que el programa protegido se mantiene como una entidad separada, éste puede ser actualizado sin necesidad de cambiar el Wrapper.
- Debido a que los Wrappers llaman al programa protegido mediante la llamada al sistema estándar `exec()`, se puede usar un solo Wrapper para controlar el acceso a diversos programas que se necesiten proteger.

#### *Funcionamiento de TCP-wrapper*

Para explicar cómo trabaja el **TCP Wrappers**, primero necesitamos entender cómo funcionan los servicios de red TCP/IP en sistemas UNIX. Los servicios de red se basan en el modelo **Cliente-Servidor**. Por ejemplo, cuando alguien usa el comando `telnet` para conectarse a un servidor, el demonio del proceso `telnet` dentro del server es ejecutado actuando así como servidor, dando al usuario la capacidad para poder acceder al sistema. Lo más común en sistemas Server es correr un **Daemon** que espera cualquier solicitud de conexión a través de la red. Cuando una solicitud de conexión tiene lugar, este **Daemon** (Usualmente llamado `inetd` en Unix) corre el servicio apropiado y regresa a su estado latente, en espera de otras solicitudes de conexiones (Figura 5.4).

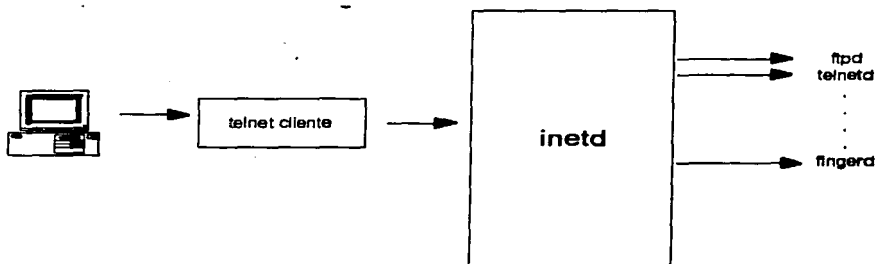


Figura 5.4 Servicios de red en un sistema Unix

TESIS CON  
FALLA DE ORIGEN



Aquí se mueve el programa servidor de red a otro lugar, y se instala un programa trivial en lugar del servidor original de red. Siempre que una conexión tiene lugar, este programa registra el nombre del servidor remoto y entonces corre el servicio de red correspondiente (Figura 5.5).

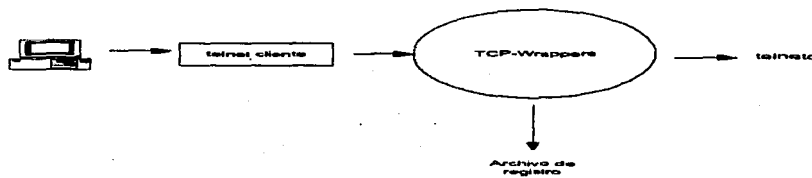


Figura 5.5 Arquitectura TCP - Wrappers

### Firewalls (Pared)

En la actualidad los firewalls son ampliamente utilizados en el mundo de internet, debido a la cantidad de redes que existen, así como del tipo de información que es filtrada entre subredes.

El objetivo principal de un firewall es el aislar u ocultar la topología de una red ante internet, así como permitir tanto la salida, como la entrada de determinado tipo de servicios de red a través de puertos específicos. Es decir verifica tanto el origen, como el punto de entrada o de llegada de información hacia la red que esta protegiendo.

Todo esto lo lleva a cabo mediante una serie de reglas, definidas en un conjunto de tablas, las cuales se consultan en el firewall y es en base a su definición si se permite o no, una petición hacia el exterior de la red interna, o viceversa, desde el internet hacia algún servicio de un host de la red interna que se esta protegiendo.

La implantación de un Firewall como medio para proteger los servicios de TCP/IP en una red corporativa, no es la única solución, sino es una solución alternativa para proteger la información, cuando esta tiene que viajar a lo largo de internet y se desea restringir los puntos hacia donde llega esta, así como de donde proviene.

TESIS CON  
FALLA DE ORIGEN



Son las reglas del negocio las que determinan si es costeable o no el pensar en una solución de seguridad como la implementación de un Firewall para proteger los datos de la organización.

Para el caso del SIAPSEP, pudieran existir hosts que se encuentren en el exterior de la red corporativa (internet) que tuviesen instalado un cliente para acceder a la aplicación ya sea para hacer una consulta a la base de datos, o actualizar algún campo en una tabla específica de la base de datos.

A continuación se muestra la figura 5.6 con la implementación si se tuviesen clientes instalados a lo largo de internet y en el caso de que existiera la necesidad de pensar en un firewall como una solución de seguridad para proteger, tanto la red, como los datos que se filtran a través de ella.

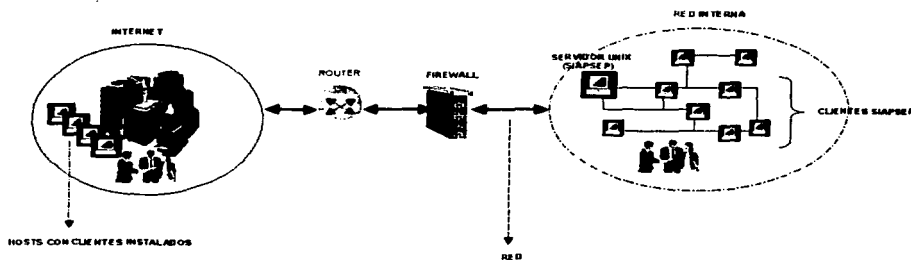


Figura 5.6 Arquitectura de seguridad en red para el SIAPSEP utilizando firewall

### OpenSSH

OpenSSH es una versión "free" de Secure Shell (**ssh**) que a su vez es un programa que permite realizar conexiones seguras con otras máquinas a través de una red, ejecutar programas en una máquina remota, y copiar archivos de una máquina a otra.

**Ssh** provee fuerte autenticación y comunicación segura sobre un canal inseguro. **Ssh** nace como un remplazo a los comandos **rlogin**, **rsh**, y **rcp**, los cuales proporcionan gran flexibilidad en la administración de una red, sin embargo, presentan grandes riesgos en la seguridad de un sistema.

TESIS CON  
FALLA DE ORIGEN

- 1.- Los servicios **r (rlogin, rsh, rcp)** son vulnerables a cierto tipo de ataques. Cualquiera que tenga acceso a una máquina en la red puede llegar por ciertos trucos a obtener acceso físico a la red de cómputo, puede obtener acceso no autorizado por una gran variedad de formas.
- 2.- Con ssh, se puede crear una sesión **X remota** segura que permanece transparente al usuario.
- 3.- Protección de ataques sobre protocolo de comunicación y el medio.

La siguiente lista muestra los tipos de ataques que podemos evitar al utilizar ssh:

*IP spoofing.*- Donde un host remoto envía paquetes a otro host aparentando ser un host conocido por este.

*Ruteo de IP spoofing.*- Donde un host puede pretender que un paquete de IP proviene de otro hosts valido.

*DNS spoofing.*- Cuando un ataque provoca que los registros de servidor de nombre se pierdan.

*Intersección de password en claro*, u otra información intercambiada entre dos hosts.

*Manipulación de datos* que se encuentran en intercambio entre hosts.

### **COPS**

COPS es una herramienta de seguridad, que comprueba muchos de los problemas de seguridad de nuestro sistema. COPS está formado por varios programas en shell y otros escritos en C.

Entre las comprobaciones que lleva a cabo destacan los siguientes:

- Detecta si existen passwords fácilmente adivinables.
- Comprueba que no haya identificadores de usuario o de grupo duplicados.
- Examina los archivos .cshrc, .login, .profile y .rhosts, para comprobar que no tengan problemas de seguridad.
- Comprueba que ni los comandos de /etc/rc ni los archivos *cron* tengan permisos de escritura para todo el mundo.
- Verifica que no existan malos caminos de búsqueda para el root y si el sistema NFS se exporta a todo el mundo.

Existen otras herramientas de seguridad las cuales son de dominio publico y son de mucha utilidad en el robustecimiento de la seguridad tanto del sistema operativo como de la red, pero nosotros solo hacemos referencia a estas.

TESIS CON  
FALLA DE ORIGEN



## V.2 SISTEMA OPERATIVO UNIX

Un requisito importante de seguridad es la posibilidad de visualizar qué es lo que está pasando y qué ha pasado en nuestro sistema. Para ello hay que controlar los archivos a los que se acceda de forma no autorizada, así como la supervisión de nuestro propio sistema para detectar cualquier agujero de seguridad.

### Supervisión del sistema de cuentas

Hay que llevar un seguimiento del sistema de cuentas para comprobar dos sucesos:

- Usuarios que han accedido al sistema cuando no deberían estar conectados (por la noche, durante las vacaciones...)
- Programas que los usuarios normalmente no deberían estar usando. Veamos las órdenes que ofrece el sistema para llevar a cabo este control.

Para esto hay que llevar un control muy preciso de las cuentas que se tienen en el sistema, y de las que no deberían existir en el mismo.

Esto es de mucha importancia, por que en el caso del sistema SIAPSEP, cuando se hacen consultas a la base de datos, en un horario fuera de lo normal (en la madrugada, por ejemplo), nos serviría para rastrear intentos de acceso maliciosos, o no autorizados, de igual forma aplicaría para que no accesen a la programación fuente o shells de ejecución.

#### *El archivo lastlog*

El archivo `/usr/adm/lastlog` almacena la hora del más reciente acceso del usuario al sistema y desde dónde se ha llevado a cabo el acceso. Este archivo es el que usa el comando `finger`. Un usuario podría conocer si alguien ha entrado en su cuenta, recordando cuándo fue la última vez que accedió al sistema.

#### *Los archivos utmp y wtmp*

El archivo `/etc/utmp` se utiliza para la visualización de aquellos conectados al sistema en cada instante. El comando `who` muestra por pantalla este archivo. Para cada usuario se da el nombre, de la terminal usada, el momento de acceso al sistema, y el host remoto si se conecta vía red. El archivo `/usr/adm/wtmp`, almacena cada acceso al sistema y cada salida del mismo. Se puede mostrar con `who`:

```
% who /usr/adm/wtmp
```

En el caso de que varios usuarios entren al sistema al mismo tiempo, es posible, que los tiempos de acceso y abandono se mezclen dando una salida confusa. Esto también se puede visualizar utilizando el comando `last`. En las nuevas versiones del sistema operativo se incluyen también los "pseudoaccesos" como el `ftp`.



### *El archivo acct*

El archivo `/etc/adm/acct` almacena cada ejecución de un proceso en el sistema, quién lo ejecutó, cuándo y cuanto tiempo tardó en ejecutarse. La información se registra cuando se completa el comando pero siempre que se haya compilado el kernel con la opción `SYSACCT`, que por defecto no está activada. Esta información es de gran utilidad para el administrador debido a que hay procesos que después de varias ejecuciones se vuelven más lentos lo que prendería un foco rojo para la revisión del porque se está tardando más.

### *Supervisión del sistema de archivos*

Tareas que se deben llevar a cabo en el control del sistema de archivos son:

- Buscar archivos que pueden modificarse por usuarios no autorizados.
- Detectar cambios no autorizados.
- Poder recuperar el sistema, dentro de lo posible, después de que se hayan dado cambios no autorizados.
- Parches y nuevas versiones

Uno de los problemas que encontramos en seguridad, son las brechas que aparecen en ciertos programas sensibles, como son el ftp, sendmail. Están saliendo continuamente nuevas versiones y "parches", para poder tapar los agujeros de seguridad que existían en las antiguas.

## **Permisos en Archivos**

### *Categorías de Permisos*

Un archivo ordinario, generalmente es representado por el siguiente símbolo en todos los tipos de Unix ( - ), por lo que un directorio es representado por la letra ( d ).

### *Usuario*

El dueño es representado por los siguientes tres caracteres, los cuáles muestran el tipo de acceso para el dueño del archivo o el directorio en su caso.

### *Grupo*

Los siguientes tres caracteres identifican los permisos para el grupo para el cual es dueño el archivo.

Los administradores del sistema definen grupos del sistema y determinan que usuarios pertenecen a que grupos.





Los usuarios en el mismo grupo pueden acceder cada archivo perteneciente a la categoría de "otros" basados en los permisos de grupo.

### Otros (Public)

El último conjunto de caracteres, llamado "permisos para otros", son los permisos que cualquiera puede tener. Otros es cualquiera que no es dueño del archivo, ni pertenece al grupo del cual es dueño el archivo, pero que tiene acceso al sistema.

### Permisos por default

Cuando un usuario accesa al sistema, archivos y directorios están protegidos por los permisos por default. Estos se colocan en un lugar automáticamente, cuando el archivo o directorio es creado.

#### Ejemplo:

```
$mkdir secure; touch prueba
$ls -l
drwxr-xr-x2user2
-rw-r--r--1user2
```

(La salida de este comando sería la siguiente: )  
 staff\$12May 2417:25secure  
 staff 0May 2417: prueba

Los permisos de las utilerías de informix por lo regular aplican el mismo criterio ya que cuando son instalados hay permisos que no deberían de darse de alta para el usuario por lo que se debe determinar de acuerdo al perfil de cada usuario que permisos debe de tener.

### Tipos de permisos

Los permisos son representados por caracteres (Tabla 5.1), que controlan quien puede leer, escribir o ejecutar el contenido de un archivo o directorio.

Permiso	Símbolo	Archivo Plano	Directorio
Lectura	r	El contenido puede ser desplegado ó copiado	El contenido puede ser listado con el comando ls
Escritura	w	El contenido del archivo puede ser modificado	El archivo puede ser sobrescrito o borrado
Ejecución	X	El archivo puede ser ejecutado (shell scripts o ejecutables solamente)	El acceso al directorio es controlado

Tabla 5.1 Permisos en Directorios y Archivos



## Ejemplos:

Los siguientes son ejemplos de diferentes tipos de permisos para archivos y directorios.

- `rwX---` En este ejemplo, el archivo tiene permiso de lectura/escritura/ejecución para el dueño solamente.
- `d r-x r-x ---` En este ejemplo, el directorio tiene permisos de lectura/ejecución para el dueño y el grupo
- `d rwx r-x -x` El directorio tiene permisos de lectura/escritura/ejecución para el dueño, lectura/ejecución para el grupo y ejecución para otros.

*Permisos en modo Octal*

El modo octal está se basa en los números del 0 al 7. Cada permiso tiene un valor octal como sigue en la tabla 5.2:

Valor Octal	Permisos
4	Lectura
2	Escritura
1	Ejecución

Tabla 5.2 Permisos utilizados valores Octales

Los valores octales a manera de permisos y las diferentes variantes básicas que podemos encontrar en los archivo se muestran en la tabla 5.3 como sigue:

Valor Octal	Permisos
7	r w x
6	r w -
5	r - x
4	r - -
3	- w x
2	- w -
1	- - x
0	- - -

Tabla 5.3 Correspondencia entre el valor octal y los permisos



## Permisos de archivos avanzados

### Controlando Modificación de Archivos

#### Permisos de setuid y setgid

El dueño y el superusuario pueden también colocar los permisos de setuid y setgid, en un archivo y setgid para un directorio. Estos permisos especiales habilitan que se controle la modificación de archivos y la creación de directorios compartidos.

#### Programas Ejecutables

Si un programa tiene el permiso de setuid y setgid, cualquiera que tenga permisos para correr el programa, es tratado como si el fuera el dueño del programa.

Si un programa tiene permisos setgid, cualquiera que tenga permisos para correr el programa es tratado como si el perteneciera al grupo al que pertenece el programa.

Los programas ejecutables con permisos setuid o setgid obtienen los UIDs (Identificador del usuario) y GIDs (Identificación del grupo), del dueño y el grupo del archivo del programa, en lugar de obtener sus UIDs y GIDs de sus procesos (usualmente un shell), con el que inician. Esto es utilizado cuando un programa debe acceder archivos que son normalmente solo accesibles al dueño o al grupo al que pertenece el programa.

Los programas ejecutables definen la interacción con el archivo (o archivos) que están siendo modificados.

#### Directorios

Los Directorios que tienen permisos setgid, propagan sus GID a archivos creados a partir de ellos. Esto es, nuevos archivos y directorios pertenecerán al mismo grupo del directorio del cual dependen.

El permiso setgid es una característica muy utilizada para directorios protegidos compartidos.

En nuestro Sistema SIAPSEP, utilizamos ampliamente este tipo de permisos sobre archivos.

Los bits setuid y setgid son desplegados con la letra "s" sobre el campo de ejecución, para el dueño y el grupo.

Por ejemplo:

```
$ ls -l /bin/passwd /etc/shadow
-r-s r-s r-x 1 root sys 22208 Mar 27 06:21 /bin/passwd
-r-- --- --- 1 root sys 529 Apr 12 16:03 /etc/shadow
```



Estos permisos permiten a los usuarios cambiar ciertos campos sobre el archivo `/etc/shadow` cuando el comando `passwd` es utilizado. Si apareciera una "S", este es un error de condición indicando que el bit de `setuid` o `setgid`, en el modo de ejecución está en off.

Estos permisos se manipulan con el comando `chmod` en Unix utilizando ya sea la notación simbólica o numérica. La notación numérica requiere cuatro números octales, cuando se está especificando `setuid` o `setgid`, y se usan los números de la izquierda para referirse a estos permisos especiales.

```
4 = setuid
2 = setgid
1 = sticky bit
```

Control de la modificación de archivos utilizando:

```
#chmod 4755 setuid_program
#chmod 2755 setgid_program
```

Crear archivos compartidos utilizando:

```
#chmod g+s directorio-x
```

El bit de `setgid` sobre un directorio debe ser cambiado utilizando notación simbólica

#### Utilizando el Sticky bit

Si un directorio es públicamente escribible, y tiene el sticky bit colocado, archivos con el directorio, pueden ser removidos o renombrados solo si algo de lo siguiente se cumple:

- El usuario es dueño del archivo
- El usuario es dueño del directorio
- El archivo es escribible por el usuario
- El usuario es root

Esto previene a usuarios el borrado de directorios públicos como el `/var/tmp` por ejemplo.

El sticky bit es desplegado con la letra "t" en el campo de ejecución para otros. Una "T" es un bit de estado indefinido, indicando que el sticky bit está en on y la ejecución está en off. Archivos de swap tienen este particular estado del sticky bit activado.

Ejemplo:

```
$ls -ld /var/tmp
d rwx rwx rwt2 sys      sys      512    Mat 26 11:02 /var/tmp
```

TESIS CON  
FALLA DE ORIGEN



El siguiente ejemplo muestra como se asignan los permisos de Sticky

Ejemplos:

```
#chmod 1777 project
#ls -ld project
  d rwx rwx rwt2rootother512Nov 1514:30 project
# chmod a=rwxt project
#ls -ld project
  d rwx rwx rwt2rootother512Nov 1514:30 project
```

#### Listas de Control de Accesos (ACL's)

Un ACL provee un gran control sobre todos los permisos de archivos. La tradicional protección de archivos en Unix provee permisos de lectura, escritura y ejecución sobre las tres clases de usuario: dueño, grupo y otros.

Un ACL da la posibilidad de definir estos permisos y aún más. Estos permisos adicionales pueden ser colocados en forma individual a un usuario y a grupos adicionales.

El programa en Unix que nos permite el manejo de ACL's es el comando **setfacl**

El formato es el siguiente:

```
setfacl options acl_entry file_name1 [filename2.....]
```

Donde: acl\_entry Es una entrada de ACL

Filename Es un archivo o directorio el cual se le va a aplicar este tipo de permiso.

El siguiente esquema muestra las Entradas Básicas de ACL's

Entrada de ACL	Significado
u[ser] : perms	Permisos para el dueño
g[roup] : perms	Permisos para el dueño del grupo
o[ther] : perms	Permisos para otros usuarios que sean dueños o miembros del grupo
m[ask] : perms	Máscara de ACL. La entrada de máscara indica el número máximo de permisos permitidos a usuarios y para grupos. Esta máscara es un forma rápida de cambiar permisos para todos los usuarios y grupos
u[ser] : uid:perms	Permisos para un usuario en específico
G[roup] : gid : perms	Permisos para un grupo en específico

Tabla 5.4 Entradas Básicas ACL's

Opciones:

-m Crea o modifica un ACL

-s Reemplaza un ACL entero con un nuevo ACL



```
-d   Borra las entradas de ACL's
-r   Recalcula los permisos de ACL's
```

Ejemplos del uso de este comando

Añadiendo permisos de lectura/escritura para el grupo ssa20

```
# setfacl -m user:ssa20:6 prueba
```

Para verificar que un archivo tiene una entrada de ACL utilizando el comando **ls**, un signo de (+) debe aparecer del lado derecho, en el campo de modos de permisos, indicando que el archivo tiene una entrada de ACL.

Ejemplo:

```
#ls -l prueba
```

```
-rwx r-- ---+ 1 william syaadmin 163 Nov11 11:12 prueba
```

En el siguiente ejemplo, se muestra como se borra una entrada de ACL:

```
#setfacl -d user:ssa20:6 prueba
```

En el siguiente ejemplo, se le colocan permisos de lectura/escritura para el usuario, para el grupo permisos de solo lectura, y para otros no se tienen permisos sobre el archivo prueba.txt: en adición el usuario ssa20 es asignado a permisos de lectura/escritura sobre el archivo, y el permiso de máscara ACL es colocado como lectura/escritura, lo cual quiere decir que ni el usuario, ni el grupo tienen permisos de ejecución.

```
# setfacl -s user: :rw-, group: :r--, other: ---, mask:rw-, user:ssa20: rw- prueba.txt
```

El siguiente comando se utiliza para verificar que la entrada de ACL ha sido colocada:

```
#ls -l
```

```
total124
```

```
-rw- r--- ---+1 william sysadmin 34816 Nov 12 14:15 prueba.txt
```

```
#getfacl ch3.txt
```

```
# file:prueba.txt
```

```
# owner: william
```

```
# group: sysadmin
```

```
user: :rw-
```

```
user:ssa20:rw- #effective:rw-
```

```
group: :r-- #effective:r--
```

```
mask:rw-
```

```
other:---
```

Para el caso de nuestro sistema SIAPSEP, los archivos pertenecientes al usuario informix, tienen la siguiente configuración:

```
-rwxr-s---+ 1 informix informix 1403176 Dec 21 1998 onstat
```



Por lo que podemos ver, este archivo tiene habilitado el sticky bit sobre el grupo, además de que tiene un ACL activado y como se mencionó anteriormente, esta es la configuración actual de los archivos en nuestro sistema.

Esta parte de ACL's se pretende utilizar en el archivo dbaccess de informix que es el que nos permite acceder a las Bases de Datos y realizar varias actividades, actualmente cuenta con los siguientes permisos:

```
-rwxr-xr-x  informix  informix  25667      Oct 12 12:50 dbaccess
```

Con estos permisos, actualmente todos los usuarios puedan acceder a dicho archivo por lo que se pretende utilizar esta característica de ACL's para que todos los usuarios (público) no puedan hacer uso de esta herramienta a excepción de uno de ellos como que podría denominarse administrador en el grupo público como se muestra en la figura 5.7.

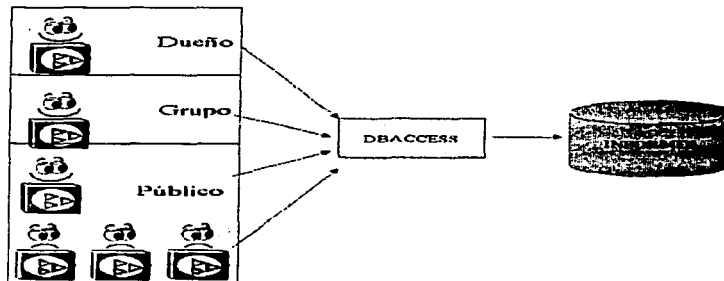


Figura 5.7 Permisos en la utilerías de informix

El archivo de informix dbaccess debe quedar con la siguiente característica:

```
-rwxr-s---+ 1 informix informix  25667      Oct 12 12:50 dbaccess
```

Todo lo que se mencionó anteriormente, es lo que podemos resaltar en cuanto a seguridad de la red de datos del sistema, así como en el sistema operativo donde convive, ya que al tomarse en cuenta estos puntos, se robustecerá de manera significativa la seguridad del sistema SIAPSEP, ya que tanto la red como el sistema operativo, son el caparazón de la seguridad de dicho sistema.

TESIS CON  
FALLA DE ORIGEN

A continuación hablaremos de lo que a seguridad en la Base de Datos se refiere, para lo cual cabe aclarar que la seguridad de la base de datos depende en un porcentaje representativo del Sistema Operativo.

### **V.3. SERVIDOR DE BASE DE DATOS INFORMIX**

La seguridad de la base de datos esta implementada en varios niveles:

La protección de los datos deberá llevarse a cabo contra fallos físicos, fallos lógicos y fallos humanos (intencionados o no). Estos fallos alteran indebidamente los datos, los corrompen con lo que la base de datos ya no puede servir a los fines para los que fue creada.

El SGBD (Sistema de Gestión de Bases de Datos) facilita normalmente mecanismos para prevenir los fallos (subsistema de control), para detectarlos una vez que se han producido (subsistema de detección) y para corregirlos después de haber sido detectados (subsistema de recuperación).

#### **Aspectos fundamentales de la seguridad**

- **Confidencialidad.** No revelar datos a usuarios no autorizados. Comprende también la privacidad (protección de datos personales).
- **Accesibilidad.** La información debe estar disponible.
- **Integridad.** Permite asegurar que los datos no han sido falseados.

La seguridad en la base de datos abarca varios temas:

- Cuestiones éticas y legales relativas al derecho a tener acceso a cierta información.
- Cuestiones de política en el nivel gubernamental, institucional o corporativo relacionadas con la información que no debe estar disponible para el público.
- Cuestiones relacionadas con el sistema.
- Necesidad en algunas organizaciones de identificar múltiples niveles de seguridad y de clasificar los datos y los usuarios según estos niveles.

Informix Dynamic Server debe proveer técnicas que permitan a ciertos usuarios tener acceso a porciones selectas de una base de datos sin tener acceso al resto. Por lo regular un manejador de base de datos cuenta con un subsistema de seguridad de autorización de la base de datos que se encarga de garantizar la seguridad de porciones de la base de datos contra el acceso no autorizado.

Una técnica de seguridad es el cifrado de datos que sirve para proteger datos confidenciales que se transmiten por satélite o algún tipo de red de comunicaciones.

En el caso del SIAPSEP, se mencionó en el subtema anterior, la existencia de algunas herramientas, las cuales cumplen con este objetivo y que proporcionan fuerte autenticación y cifrado en la comunicación, al momento de establecer una consulta hacia la base de datos por ejemplo: El cliente de informix requiere tener un usuario dado de alta en el sistema operativo, dicho usuario tiene una contraseña la cual está encriptada,





posteriormente este usuario tiene dos formas de afectar la información, la primera conectándose mediante una sesión en unix y la otra mediante la aplicación, para el caso del usuario en unix para acceder al servidor de base de datos necesita tener los permisos de ejecución a la utilidad *dbaccess* para poder interactuar con la base de datos siguiendo este esquema de acceso cuando quiere acceder a la base de datos de la *nomina\_ofm* (Base de datos del SIAPSEP) necesita tener permisos de administrador, de recurso o de conexión, después para acceder a alguna tabla necesita tener los permisos de selección, actualización, borrado o inserción.

Como podemos observar el recorrido de cualquier usuario para acceder a un sistema requiere de una serie de validaciones para permitir su uso.

Asimismo el cifrado puede proveer protección adicional a secciones confidenciales de una base de datos. Los datos se codifican mediante algún algoritmo. Un usuario no autorizado tendrá problemas para descifrar los datos codificados, pero un usuario autorizado contará con algoritmos para descifrarlos.

Entre las obligaciones del DBA está otorgar privilegios a los usuarios y clasificar a los usuarios y los datos de acuerdo con la política de la organización. Las órdenes privilegiadas del DBA incluyen los siguientes tipos de acciones:

1. Creación de cuentas
2. Concesión de privilegios.
3. Revocación de privilegios.
4. Asignación de niveles de seguridad.

La acción 1 de la lista sirve para controlar el acceso al SGBD en general, la 2 y la 3 para controlar las autorizaciones discrecionales y la 4 controla la autorización obligatoria

### *Controles informáticos*

Estos controles informáticos para la seguridad incluyen:

- Autorizaciones
- Vistas
- Backup y recuperación
- Integridad
- Encriptado
- Procedimientos asociados

### *Confidencialidad*

En un SGBD existen diversos elementos que ayudan a controlar el acceso a los datos. En primer lugar el sistema debe identificar y autenticar a los usuarios utilizando alguno de las siguientes formas:

- código y contraseña
- conocimientos, aptitudes y hábitos del usuario
- información predefinida



Además, el administrador deberá especificar los privilegios que un usuario tiene sobre los objetos:

- Utilizar una BD
- Consultar ciertos datos
- Actualizar datos.

Para facilitar la administración, los SGBD suelen incorporar el concepto de perfil, rol o grupo de usuarios que agrupa una serie de privilegios por lo que el usuario que se asigna a un grupo hereda todos los privilegios del grupo.

El mecanismo de control de acceso se encarga de denegar o conceder el acceso a los usuarios. En un SGBD pueden existir diferentes tipos de autorización.

Una primera distinción puede hacerse entre:

- Autorización explícita. Normalmente usada en los sistemas tradicionales. Consiste en almacenar que sujetos pueden acceder a ciertos objetos con determinados privilegios para lo que suele utilizarse una matriz de control de accesos
- Autorización implícita. Consiste en que una autorización definida sobre un objeto puede deducirse a partir de otras

También se puede distinguir entre:

- Autorización fuerte. Las autorizaciones deducidas a partir de la misma no puedan ser invalidadas
- Autorización débil. Se permiten excepciones sobre las autorizaciones implícitas

Por último:

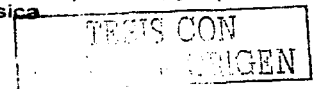
- Autorización positiva. Su presencia indica la existencia de autorización
- Autorización negativa. Es la denegación explícita de una autorización

### *Disponibilidad*

Los sistemas de bases de datos deben asegurar la disponibilidad de los datos a aquellos usuarios que tienen derecho a ello, por lo que proporcionan mecanismos que permiten recuperar la base de datos contra fallos lógicos o físicos que destruyan los datos en todo o en parte.

Aunque sólo nos centraremos en utilidades propias del manejador de bases de datos, sería conveniente además, contar con facilidades ajenas al SGBD como, por ejemplo, máquinas tolerantes a fallos, sistemas de alimentación ininterrumpida que es la parte de seguridad física de los equipos la cual debe de tener un cuidado muy importante para la disponibilidad diaria de la información y su acceso constante.

En el caso del SIAPSEP, no se está proponiendo un sistema de alta disponibilidad o redundante, en su lugar, se propone robustecer un poco mas el sistema operativo, ya que el contar con alta disponibilidad implica el doble de infraestructura física



### *Concepto de transacción*

Lo importante ante cualquier tipo de fallo es asegurar que, después de una actualización, la base de datos del Sistema SIAPSEP se queda en un estado consistente.

Para conseguir esto se crean unidades de ejecución denominadas transacciones que pueden definirse como secuencias de operaciones que han de ejecutarse de forma atómica, es decir, o bien se realizan todas las operaciones que comprenden la transacción globalmente o bien no se realiza ninguna.

Por definición, la base de datos se encuentra en un estado consistente antes de que se empiece a ejecutar una transacción y también lo deberá estar cuando la transacción termine de ejecutarse.

Una transacción puede terminar de dos formas diferentes:

- Con éxito, en cuyo caso las actualizaciones de que consta la transacción se graban (commit)
- Con fracaso, en cuyo caso debe ser restaurado el estado inicial en el que se encontraba la base de datos antes de que empezara a ejecutarse la transacción. Las actualizaciones de que consta la transacción deberán, por tanto, deshacerse (rollback).

Informix Maneja cuatro opciones mediante las cuales puede ser o no transaccional una base de datos, las cuales son las siguientes:

- Buffered transaction login
- Unbuffered transaction login
- Ansi - compliant transaction login
- No login

Esta característica es inherente a la base de datos y al momento de ser creada se puede asignar el estatus de login de la base de datos o ya estando creada. Cuando una falla ocurre el servidor de base de datos utiliza el log lógico visto en temas anteriores para recrear todas las transacciones y terminarlas.

Las bases de datos que están en modo transacciones permiten realizar las siguientes instrucciones: BEGIN, WORK, COMMIT WORK, ROLLBACK WORK, SET LOG y SET ISOLATION

### *Buffered Transaction Login*

En este tipo de estado de la base de datos, los registros son almacenados en el buffer del log lógico con cantidades grandes de almacenamiento, esto quiere decir que depende del tamaño del log será la información que podrá guardarse. Estos logs se van a limpiar de la memoria compartida siempre y cuando ocurra lo siguiente:

- El buffer este lleno.
- Un commit en la base de datos.

- Un checkpoint sea ejecutado por el manejador.
- Al momento de cerrar la conexión.

Si una falla ocurre, el administrador no puede esperar a que el servidor de base de datos recupere las transacciones que estaban en el buffer de los logs lógicos. Esto implica que pueden perderse algunas transacciones que no fueron terminadas al momento de ocurrir la falla.

Este estatus es mejor para las bases de datos que frecuentemente se actualizan (cuando la velocidad de actualización es importante). Con la finalidad que se puedan recrear las actualizaciones en caso de falla. Se puede poner a punto el tamaño de los logs lógicos para tener un equilibrio aceptable del sistema entre el rendimiento y el riesgo de pérdida de transacciones cuando ocurre una falla en el sistema.

Un ejemplo de crear una base de datos con este modo de transacción es la siguiente:

```
create database nomina_ofm with buffered log
```

Esta opción permite restablecer las operaciones después de algún error físico ó lógico. Actualmente las base de datos del SIAPSEP (nomina\_ofm) y el REPOSITORIO (unifaceseven) se encuentran en modo de buffered loggin. El identificador de este estatus es la letra B.

#### *Unbuffered Transaction Loggin*

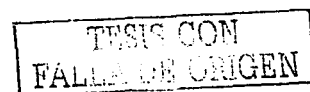
Si se realiza una transacción de este tipo, los registros en el buffer del log lógico se garantiza que son escritos a disco cuando se realiza un commit (término de transacción). Cuando se regresa el control a la aplicación después de una sentencia commit los registros del log lógico están sobre el disco. El servidor de base de datos limpia los registros del log tan pronto se ejecuta un commit.

Este estatus de loggin es una de las mejores selecciones para las bases de datos, porque garantiza que todas las transacciones terminadas (committed), pueden ser recuperadas. Cuando ocurre una falla únicamente las transacciones que no terminaron al momento de una falla se pierden. Sin embargo el servidor de base de datos limpia los logs y escribe la información a disco de manera frecuente permitiendo tener muchas paginas limpias en lo logs, evitando que se llenen, a diferencia del estatus buffered loggin no lo hace.

#### *ANSI Compliant*

Este estatus utiliza siempre las características de unbuffered transaction loggin, forzando las reglas ANSI para el procesamiento de transacciones. AL momento de ser asignado este estatus ya no puede ser cambiado por cualquiera de los antes mencionados.

#### *No Database Login*



Este estado basicamente se utiliza cuando se requiere realizar una carga masiva de datos ejecutando múltiples queries (sentencias SQL).

Si el administrador esta satisfecho con la fuente de recuperación, puede decidirse utilizar este estatus, con la finalidad de reducir el procesamiento, debido a que ya no son almacenadas las transacciones en los logs. Un ejemplo de uso de este estatus es cuando se están cargando muchos registros dentro de una base de datos partiendo de un dispositivo como puede ser una cinta ó de algún archivo, esto no requiere tener las bases de datos de manera transaccional, además de que la carga de datos es más rápida. Sin embargo si otros usuarios están activos en el servidor de base de datos no van a estar los registros en los logs lógicos hasta que se cambie el estatus de loggin.

#### *Recuperación de la base de datos*

Al ocurrir un fallo que dé lugar a pérdida de memoria volátil, es preciso realizar la operación que se suele denominar *recuperación en caliente*, en la que el sistema consulta el archivo diario para determinar las transacciones que hay que deshacer porque no han sido completadas y las que hay que rehacer porque, si bien se han completado, no habían sido grabadas en la base de datos cuando se produjo el fallo.

Con este fin, al recuperar la base de datos después de una caída del sistema, se obtiene la dirección del registro de recuperación más reciente y se recorre el archivo diario desde este punto hasta el final.

Este proceso ocurre cuando hay una caída del servidor de base de datos informix, en el momento que el administrador da de alta nuevamente el manejador, pasa por un proceso de recuperación denominado fast recovery, el cual revisa las transacciones que no concluyeron para deshacer todos los cambios realizados.

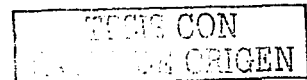
#### *Recuperación en frío*

En caso de un fallo de memoria secundaria que afecte a la base de datos, se lleva a cabo una *recuperación en frío*, que consiste en utilizar una copia de seguridad de la BD, también llamada de *respaldo (backup)*.

La copia de seguridad permitirá, junto con los archivos diarios que se han ido produciendo desde que se realizó la copia de seguridad, reconstruir la BD llevándola de forma consistente a la situación anterior a que se produjera el fallo.

Otro caso que se puede dar es el denominado *error fatal* que se produce cuando se pierde el archivo diario grabado en un soporte, en este caso resulta imposible recuperar la base de datos a su estado actual.

Esta parte se apoya mucho en lo que se explicó en el tema anterior de la variedad de respaldos que se pueden realizar en Informix que permite restaurar desde el log de transacciones, un dbspace en específico hasta el servidor de base de datos completo.



También se puede duplicar la base de datos. En general todas las técnicas de duplicación se conocen como *espejo (mirroring)* o *duplexación (duplexing)* y la replicación de datos se explicara a continuación.

### Espejo

Espejo es una estrategia de copiado de un dbspace, blobspace ó sbspace, con un tamaño igual al momento de realizarse el espejo. Cada escritura realizada al chunk primario, es acompañada por una escritura idéntica al chunk espejo. La figura 5.8 ilustra la forma como Informix realiza el espejeo de información.

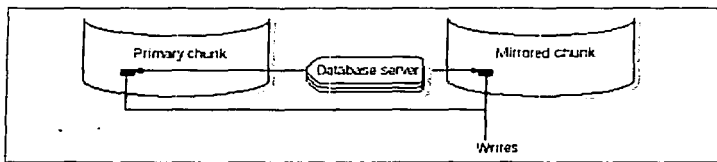


Figura 5.8 Espejeo en Informix

Si alguna falla ocurre en el chunk primario (Chunk del que se está copiando), se puede habilitar el chunk espejo para lectura y escritura, para la recuperación del chunk primario. No está soportado el espejeo de dispositivos en ubicaciones fuera de la red ya que deben estar registrados directamente en el servidor de base de datos.

Un espejo permite una recuperación en la cual no hay mucha pérdida de tiempo en caso de que se requiera restaurar un servidor de base de datos, ya que la otra opción sería desde un dispositivo de respaldo pero eso requiere mayor tiempo.

Cualquier base de datos crítica puede ser localizada en un dbspace espejo.

Toda seguridad requiere un costo y no es la excepción los espejos debido a que requieren espacio en disco para los dispositivos espejo así como hay un impacto en el rendimiento debido a que realiza escrituras en el chunk primario como en el chunk espejo.

Para saber que información debe ser espejeada, idealmente debería ser toda pero como ya vimos hay un impacto tanto en espacio en disco como en performance, cuando se está limitado en estos dos aspectos se deben de considerar los datos críticos, iniciando con el dbspace de root, el chunk que almacena los logs lógico y los logs físicos, ya que si cualquiera de éstas áreas falla el servidor se cae pasando a estatus off-line (fuera de línea). También son de gran importancia la información que contiene datos que son críticos para la empresa para el caso del SIAPSEP otro dispositivo que debe ser espejeado es el dbspace de datos debido a que ahí está contenida la información de toda la nómina de maestros del D.F.

Existen varias alternativas para seleccionar el dispositivo que se va a utilizar para la información espejo. Una es el utilizar un manejador de volúmenes lógicos, en la cual en un mismo file system hay múltiples discos. Otra opción es el hardware de espejo dentro del cual podemos considerar el RAID (redundant array of inexpensive disk), el cual requiere menos espacio en disco, entre otros.

Los pasos requeridos para el espejo de datos son los siguientes:

1. Poner el servidor de base de datos fuera de línea y habilitar el espejo.
2. Regresar el servidor de BD a modo Online
3. Asignar el espacio en disco del chunk espejo.
4. Seleccionar el dbspace, blobspace, or sbpace que se requiere espejear y crear el chunk espejo especificando la localización del mismo. Este paso se repite para todas las áreas de datos que se requiere tener espejo.

Para habilitar el espejo se debe realizar, modificando del archivo de configuración Informix el parámetro MIRROR colocandolo a 1 debido que por default maneja 0 y quiere decir que está deshabilitado.

Cuando un espejo es asignado puede cambiar su estatus, estos son los siguientes:

- Cambiar el chunk espejo de on - line a down
- Cambiar el chunk espejo de down a recovery

Cuando el espejo esta en estatus down el servidor de base de datos no puede leer ni escribir en él. Este estatus no es igual que hablar de un espejo end (finalizado) esto provoca que todos los espejos de los dbspaces sean liberados lo cual permite que sean reasignados posteriormente, cabe señalar que unicamente los usuarios de root e informix pueden realizar estos cambios.

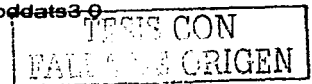
A continuación se muestra la forma como se espeja el dbspaces de datos de la base de datos del SIAPSEP que tiene por nombre lógico bddats.

Se utiliza la instrucción *onspaces* para iniciar el espejo, el dbspaces de la base de datos de la nomina\_ofm está asociado al dbspace de datos bddats, pero este a su vez tiene asociados dos chunks que son bddats1 y bddats2. La instrucción se aplicaría de la siguiente manera:

```
$ onspaces -m bddats -p /dev/data1 -o 0 -m /dev/mirror_data1 0
-p /dev/data2 -o 5000 -m /dev/mnirror_data2 5000
```

Este ejemplo es para el caso de que ya existan las áreas de datos, pero si se requiriera que al momento que se cree un área de datos ya contengan un espejo se utiliza la siguiente instrucción:

```
$ onspaces -c -d bddats3 -p /dev/bddats_3 -o 0 -s 2500 -m /dev/mirror_bddats3-0
```



En caso de que se requiera pasar un espejo a estatus down se utiliza onspaces como se muestra a continuación:

```
$ onspaces -s bddats1 -p /dev/data1 -o 0 -D
```

Para la recuperación de un espejo con estatus down se utiliza la siguiente instrucción:

```
$ onspaces -s bddats1 -p /dev/data1 -o 0 -O
```

Y para finalizar un espejo de algún área de datos específica se realiza lo siguiente:

```
$ onspaces -r bddats1
```

Actualmente el servidor de base de datos en donde están las bases de datos del SIAPSEP no utiliza el espejeo de datos, pero sería una buena alternativa para la disponibilidad inmediata de la información.

### *Replicación de datos*

La replicación es muy amplia y se refiere al proceso de representación de objetos de bases de datos en más de un sitio distinto al que se encuentran. Como por ejemplo una copia de una base de datos a un servidor de base de datos en otro computador diferente.

Las ventajas que proporciona la replicación de datos son las siguientes:

- Proporciona mayor rendimiento ya que los clientes pueden conectarse a los datos replicados evitando que haya mucha contención en un solo servidor de base de datos.
- Y hay mayor disponibilidad de datos con respecto al espejeo.

Obviamente la replicación de datos requiere de mayor espacio de almacenamiento y mucha actualización.

Todos los datos de un servidor de base de datos son replicados y actualizados dinámicamente a otro servidor de base de datos, frecuentemente en otra localización geográfica, proporcionando un camino para mantener una copia entera del servidor de base de datos la cual puede ser accesada ante cualquier falla ocurrida en el servidor replicado. Esta es una buena solución para la seguridad de datos cuando se daña el equipo, ó algún evento provoca la destrucción del mismo.

Cuando se realiza la replicación un servidor se denomina servidor de base de datos primario, el otro es llamado servidor de base de datos secundario. En operaciones normales los clientes suelen conectarse al servidor primario, de igual forma los clientes pueden usar el servidor secundario pero solo como lectura de datos debido a que el servidor secundario no permite actualizaciones del cliente (Figura 5.9).





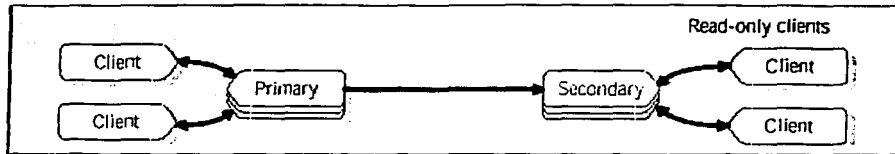


Figura 5.9 Acceso al servidor de Replicación secundario pero para consultas únicamente

Si uno de los servidores falla, pueden redireccionarse para que los clientes que utilizaban el servidor primario ahora utilicen el servidor secundario como se muestra en la figura 5.10

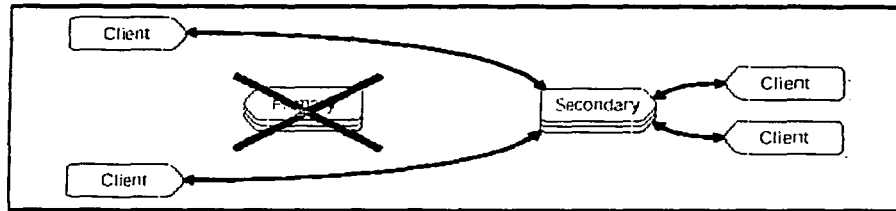


Figura 5.10 Redireccionamiento del servidor primario al secundario cuando el primero falla

Las características de la replicación son las siguientes:

- Proporciona recuperación inmediata si alguno de los servidores tiene fallas.
- Permite la carga balanceada entre los dos servidores de base de datos.

Las diferencias entre el espejeo de datos y la replicación de datos, es que el espejeo se hace sobre algunas o todas las porciones de datos y la replicación es sobre todo el servidor de base de datos (Figura 5.11).

TESIS CON  
FALTA DE ORIGEN

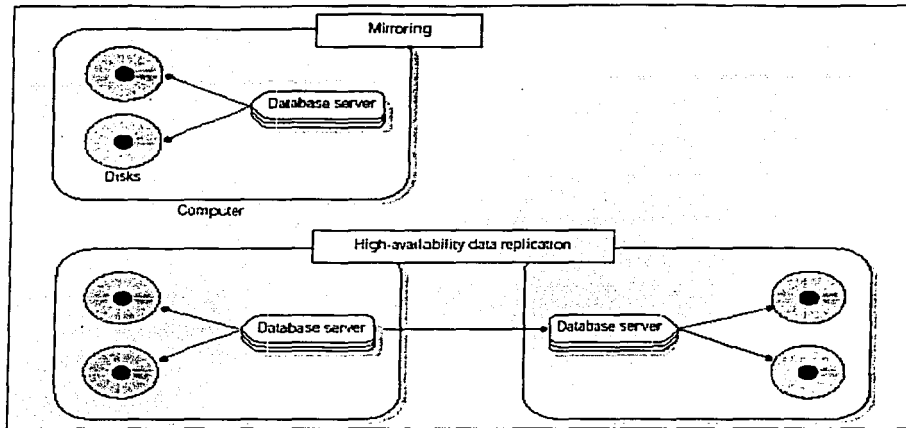


Figura 5.11 Espejeo y Replicación de Datos

Para iniciar la replicación se utilizan `dbspaces backup` y archivos de logs lógicos de un servidor a otro, el procedimiento es el siguiente:

- Determina el volumen de datos manejado por los dos servidores de base de datos creando un nivel 0 de respaldo, esto quiere decir un respaldo completo del servidor primario y posteriormente restaurar ese respaldo en el servidor secundario.
- El servidor de base de datos restaurado (servidor secundario), tienen que ser leídos los registros del log lógico para aplicar todas las transacciones realizadas.

Hay dos tipos de actualización que se realiza dependiendo del valor en el parámetro de configuración `DRINTERVAL` que son las siguientes:

**Actualización sincrónica.-** El parámetro `DRINTERVAL` debe -1. Tan pronto como el servidor de BD primario escribe el contenido del buffer de los logs lógicos al buffer de replicación, estos son enviados al buffer de replicación del servidor secundario. El buffer de los logs lógicos del servidor primario se limpian hasta que la transferencia es completada.

**Actualización Asíncrona.-** El valor de `DRINTERVAL` debe ser diferente a -1. El servidor de BD primario limpia los logs lógicos después de realizar la copia del contenido, independientemente de la acción realizada.

Cuando ocurre alguna falla en el servidor de base de datos primario y se desea redireccionar a los clientes se tiene que realizar lo siguiente:



Se utiliza la variable de ambiente *DBPATH* mediante la cual el cliente sabe a que servidor debe conectarse, indistintamente la variable de ambiente *INFORMIXSERVER* puede seguir direccionada al servidor primario. Como se muestra a continuación:

Suponiendo que el nombre del servidor primario es *sep\_shm* y el servidor secundario sea *sep1\_shm* las variables de ambiente quedarían así:

```
INFORMIXSERVER    sep_shm
DBPATH            sep1_shm
```

La siguiente esquema muestra como los clientes son redireccionados a otro servidor en caso de ocurrir alguna falla:

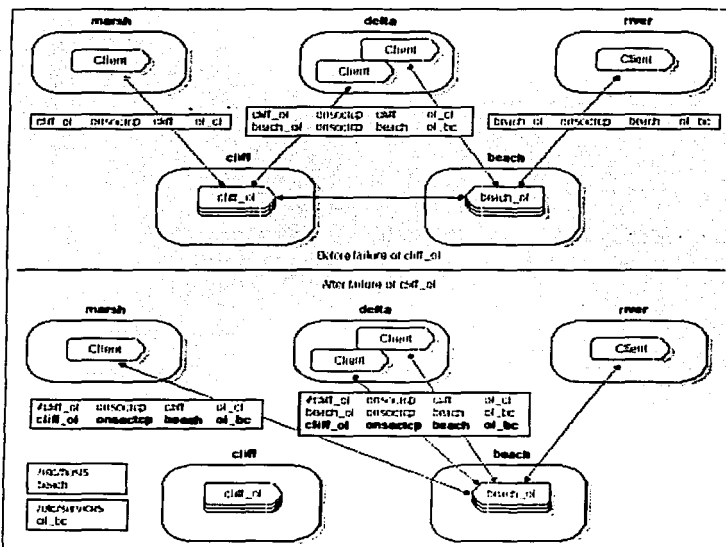


Figura 5.12 Redireccionamiento de clientes cuando hay replicación de datos

Para poder configurar la replicación se requiere lo siguiente:

- Hardware y software para el servidor de replicación.
- Requerimientos de base de datos.
- Configuración del servidor de base de datos.
- Configuración de la conectividad.

TESIS CON  
FALLA DE ORIGEN

Las bases de datos del servidor deben de estar en transaction loggin, así como la versión de informix debe ser la misma tanto para el servidor primario como el servidor secundario, el tamaño de dbspaces y chunks debe ser igual. Los siguientes parámetros del archivo ONCONFIG deben ser iguales:

- ROOTNAME
- ROOTOFFSET
- ROOTPATH
- ROOTSIZE

No deben estar asignadas las variables de espejo MIRROROFFSET y MIRRORPATH, los logs físicos deben ser idénticos en ambos servidor y los parámetros del ONCONFIG, PHYSDBS, PHYSFILES, LTAPEBLK, LTAPESIZE, TAPEBLK, TAPESIZE ( Variables de configuración de cinta), LOGFILE, LOGSIZE (Parámetros de logs lógicos), deben ser también idénticos en ambos servidores.

\*Todos los parámetros de configuración de la memoria compartida deben ser iguales en ambos servidores (SHMADD, SHMTOTAL, SHMVRTSIZE).

Los parámetros de replicación que se debe configurar son:

- DRINTERVAL
- DRLOFOUND
- DRTIMEOUT

Estos deben ser iguales en ambos servidores.

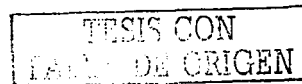
Este proceso de replicación es complicado en configuración pero uno de los mejores metodos para la alta disponibilidad y la seguridad de los datos.

### *Integridad*

El objetivo en cuanto a la integridad es proteger la base de datos contra operaciones que introduzcan inconsistencias en los datos, por eso hablamos de integridad en el sentido de *corrección, validez o precisión* de los datos de la base. El subsistema de integridad de Informix debe, por tanto, detectar y corregir, en la medida de lo posible, las operaciones incorrectas. Existen dos tipos de operaciones que pueden atentar contra la integridad de los datos que son las operaciones semánticamente inconsistentes y las interferencias debidas a accesos concurrentes. En lo que se refiere a la integridad de la Base de datos para el caso del sistema SIAPSEP la integridad referencial se encuentra en el front end.

### *Integridad semántica*

Existen operaciones que pueden violar restricciones definidas al diseñar la base de datos, como pueden ser restricciones sobre los dominios o sobre los atributos. Estas restricciones pueden ser estáticas (también llamadas de estado o situación) o dinámicas (denominadas de transición).



Un aspecto muy importante de estas reglas de integridad, es que se almacenan en el diccionario, como parte integrante de la descripción de los datos (*control centralizado de la semántica*) con lo que se consiguen las siguientes ventajas:  
 Son más sencillas de entender y de cambiar, facilitando su mantenimiento

- Se detectan mejor las inconsistencias
- Se protege mejor la integridad, ya que ningún usuario podrá escribir un programa que las viole llevando la base de datos a estados inconsistentes

El subsistema de integridad de Informix Dynamic Server debe realizar las siguientes funciones:

- Comprobar la coherencia de las reglas que se definen
- Controlar las distintas transacciones y violaciones de integridad
- Cuando se produce una violación, ejecutar las acciones pertinentes

### *Integridad operacional*

En sistemas multiusuario es imprescindible, además, un mecanismo de control de concurrencia para conservar la integridad de la base de datos, ya que se pueden producir importantes inconsistencias derivadas del acceso concurrente.

A continuación se presentan las técnicas de control de concurrencia más tradicionales:

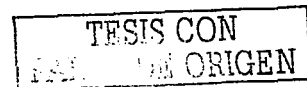
- Bloqueo
- Marcas de tiempo
- Marcas de tiempo multiversión
- Técnicas optimistas para, posteriormente, resumir las principales técnicas avanzadas:
- Transacciones anidadas
- Transacciones largas
- Transacciones de coordinación

### *Operacional*

Una vez que el usuario obtiene acceso al servidor de base de datos, los usuarios pueden acceder solo las bases de datos y los objetos para los cuáles tienen otorgados privilegios.

Por lo anterior Informix permite realizar a los DBA's

- Privilegios de nivel de la Base de datos
- Privilegios sobre nivel de objetos, por ejemplo:
  - Tablas
  - Columnas
  - Vistas
  - Rutinas
  - Fragmentos



**Nota: Todas las tareas de administración están restringidas al usuario informix**

Los tres niveles de acceso a una base de datos son:

- Connect
- Resource
- DBA

Mediante la utilidad dbschema se pueden identificar los permisos que actualmente tienen una base de datos a nivel base de datos. El siguiente ejemplo es del esquema de la base de datos *nomina\_ofm*:

```
DBSCHEMA Schema Utility          INFORMIX-SQL Version 9.20.UCS
Copyright (C) Informix Software, Inc., 2000
Software Serial Number INF#J010494
grant dba to "informix";
grant resource to "a713ap00";
grant connect to "public";
```

#### *El privilegio CONNECT*

Este privilegio permite conectarse a la base de datos, crear tablas e índices temporales, crear vistas y sinónimos, otorgar permisos a objetos que son creados y se liberan. No se pueden crear tablas e índices permanentes

#### *El privilegio RESOURCE*

Este privilegio nos da la posibilidad de crear tablas e índices permanentes, procedimientos almacenados y funciones.

#### *El privilegio DBA*

Un usuario con este privilegio tiene acceso completo al servidor de base de datos. Además de que puede dar de alta usuarios así como revocar permisos a todos los niveles en la base de datos.

A continuación se listan los diferentes privilegios que se tienen:

ALTER	Añadir, borrar o modificar columnas
DELETE	Remover renglones de una tabla
INDEX	Crear índices para una tabla
SELECT	Obtener columnas de información de una tabla
UPDATE	Obtener información de las columnas de una tabla
INSERT	Insertar renglones a una tabla
REFERENCES	Referenciar columnas en un constraint referencial
ALL	Permitir alguna o todas las operaciones precedidas

TESIS CON  
 FALLA DE ORIGEN

El siguiente esquema muestra los niveles de seguridad que el servidor de base de datos ofrece para la seguridad de la información.



Figura 5.13 Niveles de Seguridad en el Servidor de Base de Datos Informix

### *Privilegios de tablas*

Para las tablas, se pueden otorgar permisos de acceso de solo lectura (SELECT), modificación de acceso (DELETE, INSERT, UPDATE), o acceso de administración (ALTER, INDEX, REFERENCES).

El privilegio especial UNDER, aplica solo para unos tipos de tablas. Este privilegio da la posibilidad de crear una nueva subtabla, bajo un existente tipo de tabla. Para otorgar estos privilegios a un usuario, el administrador puede utilizar la palabra ALL.

### *Privilegios sobre columna*

Los privilegios SELECT, UPDATE, y REFERENCES pueden ser otorgados solo sobre una columna en específico sobre una tabla.

Informix otorga ciertos privilegios a nivel tabla y de base de datos por default

- Privilegios por default a una Base de Datos.

Si se desea permitir a los demás usuarios acceder a la base de datos, se deben otorgar los privilegios de CONNECT, RESOURCE, ó de DBA.



**Privilegios por default a una Tabla.**

Para una Base de datos, los niveles por default son automáticamente otorgados en *modo public* cuando una tabla es creada. Para una base de datos ANSI, estos privilegios no son automáticamente otorgados. Se deben otorgar explícitamente en *modo public* solo a usuarios en específico.

**NODEFDAC**

Para prevenir que se otorgue por default estos privilegios a una base de datos que no es del tipo ANSI, se coloca la variable de ambiente **NODEFDAC**, en modo "yes" cuando se ejecute la sentencia CREATE TABLE.

A continuación se muestran unos ejemplos de como se otorgan privilegios a usuarios:

```
GRANT CONNECT TO PUBLIC;    Connect es otorgado a todos los usuarios
GRANT RESOURCE TO maria, joe;
GRANT DBA TO janet;
```

Se puede utilizar esta sentencia GRANT para otorgar privilegios sobre la base de datos a otros usuarios. Los componentes de esta sentencia son los siguientes:

<i>Privilege</i>	Uno de los tipos de acceso a una base de datos: CONNECT, RESOURCE o DBA.
<i>PUBLIC</i>	Esta palabra se utiliza para especificar privilegios de acceso a todos los usuarios
<i>user-list</i>	Una lista de los nombres de usuario a los cuales se les está otorgando privilegios de acceso, se pueden introducir uno o mas nombres, separados por comas.

En el segundo ejemplo, el privilegio RESOURCE se otorga solo a los usuarios maria y joe.

En el tercer caso, janet tiene privilegios de DBA

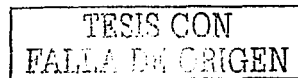
Se puede utilizar la sentencia REVOKE para revocar privilegios de acceso de los usuarios. Los componentes de esta sentencia son:

<i>privilege</i>	Uno de los tipos de niveles de acceso: CONNECT, RESOURCE o DBA.
<i>PUBLIC</i>	Esta palabra que se utiliza para especificar, privilegios de acceso para todos los Usuarios.
<i>User-list</i>	Lista de usuarios a quienes se les están otorgando los permisos de acceso.

Ejemplos:

```
REVOKE CONNECT FROM mike;
```

En este ejemplo, el privilegio CONNECT es retirado al usuario mike





**REVOKE RESOURCE FROM maria;**

En este ejemplo, el privilegio RESOURCE ya no es otorgado al usuario maría.

*Revocando privilegios a nivel Tabla*

Se puede utilizar la sentencia REVOKE para prevenir operaciones específicas que un usuario puede efectuar sobre una tabla, que se haya creado. Los componentes de la sentencia REVOKE son:

<i>Privilege</i>	Uno o mas tipos de acceso a la tabla: ALTER, DELETE, INDEX, INSERT, SELECT, UPDATE. REFERENCES y ALL.
<i>*tabla o vista</i>	El nombre de la tabla o vista, para la cual se están otorgando permisos.
<i>PUBLIC</i>	Es la palabra clave que se utiliza para especificar privilegios de acceso para todos los usuarios
<i>user-list</i>	Una lista de nombres de login para los usuarios a los cuales se les está otorgando privilegios de acceso, se pueden introducir uno o más nombres, separados por comas.

Aunque se puede otorgar privilegios de UPDATE y SELECT para columnas específicas, no se pueden revocar estos privilegios, columna por columna. Si se revocan los privilegios de UPDATE y SELECT de los usuarios, los privilegios de UPDATE y SELECT son revocados.

Ejemplos:

**REVOKE ALL ON orders FROM PUBLIC;**

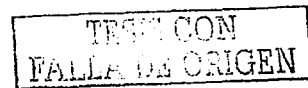
En este primer ejemplo todos los privilegios son revocados de todos los usuarios (PUBLIC) en la tabla orders

**REVOKE DELETE, UPDATE ON customer FROM mike, maria;**

En este ejemplo, los privilegios de DELETE y UPDATE son revocados de los usuarios mike y maría.

*Privilegios a nivel columna*

Solo privilegios de SELECT, UPDATE y PREFERENCES pueden ser otorgados hacia columnas individuales.



Los privilegios de columna son otorgados en la misma forma que se otorgan los privilegios a nivel tabla, excepto que una lista de columna deba tener el privilegio de la sentencia GRANT.

Cuando se asignan privilegios sobre una tabla, se pueden especificar los privilegios SELECT, UPDATE y PREFERENCES para aplicarlos solo sobre ciertas columnas en la tabla.

Ejemplo:  
GRANT SELECT (company, fname, lname)  
ON customer  
TO PUBLIC;

En este ejemplo, el privilegio SELECT se otorga a todos los usuarios, sobre las columnas company, fname y lname de la tabla customer

GRANT INSERT, UPDATE (quantity), SELECT  
ON items  
TO maria;

En este ejemplo, el privilegio UPDATE es otorgado solo en la columna quantity, pero los privilegios INSERT y SELECT se otorgan a todas las columnas sobre la tabla.

**Nota:** Los privilegios a nivel de columna no pueden ser revocados. Para remover un privilegio a nivel de columna, primero se revocan los privilegios a nivel de tabla, y se reasignan los privilegios sobre varias columnas como sea apropiado.

#### *Privilegios de Rutina*

Cuando se crea una rutina de usuario definida, de igual manera, un procedimiento almacenado o una función de la que el usuario es dueño, se otorgan privilegios de ejecución automáticamente para esa rutina.

Con el privilegio de ejecución es posible ejecutar las sentencias EXECUTE PROCEDURE y EXECUTE FUNCTION para la rutina

Si el administrador desea que otros usuarios tengan la posibilidad de ejecutar la rutina, debe de otorgárseles el privilegio de EXECUTE, utilizando la sentencia GRANT EXECUTE ON.

Ejemplos:

GRANT EXECUTE ON total\_orders TO PUBLIC;  
GRANT EXECUTE ON square ( x INT ) TO maria;  
REVOKE EXECUTE ON cancel\_orders FROM joe, tom;

TESIS CON  
FALLA DE ORIGEN



### *Roles*

Un rol es un grupo de usuarios, los cuales pueden tener un conjunto de privilegios otorgados. Los roles pueden hacer el trabajo del administrador más fácil. Una vez que un usuario es asignado a un rol, el administrador del sistema necesita solo el otorgarle o revocarle privilegios sobre una tabla a un rol.

#### Ejemplos de creación de roles

```
CREATE ROLE mkting;  
CREATE ROLE slsadmin;  
CREATE ROLE sales;  
GRANT mkting TO jim, mary, ram;  
GRANT slsadmin TO andy, liz, sam;  
GRANT sales TO mkting, slsadmin;
```

La sentencia **CREATE ROLE** se utiliza para crear un rol. Esta sentencia coloca una entrada en la tabla del catálogo `sysusers`, donde el tipo de usuarios es `G`. El nombre del rol debe ser menor o igual a 8 caracteres

Todos los usuarios y nombres de roles en el sistema deben de ser únicos. Por ejemplo si se tiene un nombre de usuario `gus`, que tiene la capacidad de conectarse a la base de datos del servidor, no es posible crear un rol llamado `gus`.

Un usuario no es capaz de conectarse si el nombre de usuario es creado como un nombre de rol.

La sentencia **CREATE ROLE** puede solo ser ejecutada por un usuario el cual tiene permisos de `DBA` sobre la base de datos.

Un rol es un objeto de la base de datos, esto quiere decir que solo es aplicable sobre la base de datos sobre la cual es creado el objeto.

Una vez que el rol es creado, el siguiente paso es asignar usuarios a los roles.

La sentencia **GRANT** asigna uno o mas usuarios hacia el rol especificado. Se puede también asignar un rol a otro rol. Una sentencia que es exitosa del comando **GRANT** coloca una entrada en la tabla de sistema `sysroleauth`.

#### Ejemplos de Asignación de roles:

```
REVOKE ALL ON orders FROM public;  
GRANT SELECT ON orders TO sales;  
GRANT INSERT, UPDATE, DELETE ON orders TO slsadmin;
```

Tablas y privilegios sobre tablas, pueden ser asignados a roles utilizando la sentencia **GRANT**, mas sin embargo, privilegios sobre la base de datos no pueden ser asignados a roles.



Antes de que los permisos a un rol puedan ser utilizados, la sesión debe ejecutar la sentencia SET ROLE, como a continuación se muestra:

```
SET ROLE slsadmin;
```

Esta se ejecutará por el usuario slsadmin:

```
INSERT INTO ORDERS(order_num, customer_num)
VALUES (0,104);
REVOKE ALL ON orders FROM slsadmin;
```

Se debe ejecutar la sentencia SET ROLE antes de recibir los permisos otorgados hacia un rol particular.

En el ejemplo mostrado anteriormente, el permiso slsadmin son activados para la actual sesión. Se obtienen los privilegios del rol, además de que se mantienen los privilegios del dueño, así como de tipo PUBLIC. Los permisos del rol están activos antes de que la sesión se interrumpa, o la actual base de datos sea cambiada. Para deshabilitar un rol se ejecuta:

```
SET ROLE {NONE|NULL}
```

Si se intenta ejecutar SET ROLE y el rol no fue asignado hacia usted, el siguiente usuario se genera:

```
19805: No privilege to set to the role.
```

Informix no asocia passwords con roles.

Dos ejemplos de la sentencia GRANT FRAGMENT y REVOKE FRAGMENT se muestran a continuación:

```
REVOKE FRAGMENT ALL Usuario1 puede ahora ejecutar UPDATE, INSERT
ON orders o DELETE Solo del fragmento en DBSPA-FROM user1;
```

```
GRANT FRAGMENT INSERT,UPDATE,DELETE
ON orders(dbSPACE1)
TO user1;
```

Estos ejemplos muestran como se puede otorgar privilegios de solo lectura sobre un fragmento pero solo sobre el dbSPACE1.

Si el usuario1 trata de hacer INSERT a un registro que está dentro de algún fragmento pero no en el dbSPACE1, el siguiente error ocurriría:

```
977: No permission on fragment (dbSPACE1).
271: Could not insert new row into the table.
```

Los privilegios de nivel de fragmento que se pueden otorgar, son INSERT, UPDATE, DELETE y ALL. Estos pueden ser otorgados se tenga o no privilegios a nivel de tabla

<p style="text-align: center;">TESIS CON FALLA DE ORIGEN</p>
--



Los privilegios de tabla toman precedencia de donde terminan los privilegios de nivel de fragmento.

Por ejemplo, si se tiene la capacidad de insertar a nivel tabla, los privilegios de insert a nivel de fragmento no son revisados.

Ejemplos:

La sentencia INSERT mostrada anteriormente sería exitosa si user1 tiene permisos de SELECT en la tabla. (Los permisos a nivel de fragmento son solo para las sentencias INSERT, UPDATE y DELETE).

```
INSERT INTO orders(cust_nbr) VALUES 100;
SELECT * FROM orders;
UPDATE orders SET cust_nbr = 12200
WHERE cust_nbr = 220;
```

La sentencia UPDATE mostrada en la parte superior fallaría ya que user1 no tiene permisos de UPDATE para el fragmento en el dbspace2. El usuario requiere permisos de UPDATE para el fragmento desde donde el registro es movido y el fragmento hacia donde el registro es movido.

La información que se presenta a continuación es fragmento del esquema actual de la base de datos del SIAPSEP, con los permisos default:

```
grant all on "informix".acumulados to "public" as "informix";
grant all on "informix".ahisa_desc75 to "public" as "informix";
grant all on "informix".area_sep to "public" as "informix";
grant all on "informix".banco_plaza to "public" as "informix";
grant all on "informix".ban_cam_plaza to "public" as "informix";
grant all on "informix".ban_cat_puesto to "public" as "informix";
grant all on "informix".ban_ins_cpto to "public" as "informix";
grant all on "informix".ban_ins_mens to "public" as "informix";
grant all on "informix".ban_pago to "public" as "informix";
grant all on "informix".ban_pago_auto to "public" as "informix";
grant all on "informix".bolsa_trabajo to "public" as "informix";
grant all on "informix".cates_bas to "public" as "informix";
grant all on "informix".centro_computo to "public" as "informix";
grant all on "informix".cifras_control to "public" as "informix";
grant all on "informix".clasif_serv to "public" as "informix";
grant all on "informix".codigos_operacion to "public" as "informix";
grant all on "informix".cod_paren to "public" as "informix";
grant all on "informix".clasificadores to "public" as "informix";
grant all on "informix".concepto to "public" as "informix";
grant all on "informix".concepto_calculo to "public" as "informix";
grant all on "informix".cons_hist_pago to "public" as "informix";
grant all on "informix".control_acum_cptos to "public" as "informix";
grant all on "informix".cptos_prorratedos to "public" as "informix";
grant all on "informix".crea_can_coder to "public" as "informix";
grant all on "informix".credito_salario to "public" as "informix";
grant all on "informix".ct_concepto to "public" as "informix";
grant all on "informix".cve_cong to "public" as "informix";
```

TESIS CON  
FALLA DE ORIGEN

```

grant all on "informix".cve_mov to "public" as "informix";
grant all on "informix".deduccion to "public" as "informix";
grant all on "informix".ded_hist_pago to "public" as "informix";
grant all on "informix".dependientes to "public" as "informix";
grant all on "informix".dep_admva to "public" as "informix";
grant all on "informix".dep_norm to "public" as "informix";
grant all on "informix".detalle_mov1 to "public" as "informix";
grant all on "informix".edo_civil to "public" as "informix";
grant all on "informix".edo_mun_loc to "public" as "informix";
grant all on "informix".empleado to "public" as "informix";
grant all on "informix".empleado_nss to "public" as "informix";
grant all on "informix".emp_plaza_msg to "public" as "informix";
grant all on "informix".entidades_fed to "public" as "informix";
grant all on "informix".errores_de_movimie to "public" as "informix";
grant all on "informix".error_de_sistema to "public" as "informix";
grant all on "informix".escuela_origen to "public" as "informix";
grant all on "informix".estatus to "public" as "informix";
grant all on "informix".estatus_cheque to "public" as "informix";
grant all on "informix".estatus_ct to "public" as "informix";
grant all on "informix".estatus_mov_tec to "public" as "informix";
grant all on "informix".estatus_plaza to "public" as "informix";
grant all on "informix".estr_org_ct to "public" as "informix";
grant all on "informix".estr_org_ls to "public" as "informix";
grant all on "informix".etapas_calc to "public" as "informix";
grant all on "informix".formas to "public" as "informix";
grant all on "informix".grupos to "public" as "informix";
grant all on "informix".grupo_gen to "public" as "informix";
grant all on "informix".grupo_usu to "public" as "informix";
grant all on "informix".identificadores to "public" as "informix";

```

Por lo que se realizaron dos programas en ESQ/C los cuales permiten revocar los permisos y asociarlos a los usuarios que se desean tengan acceso.

El programa 1 de acuerdo a la variable de ambiente DSQUERY determina la base de datos que deberá conectarse, de la cual selecciona todas las tablas de usuario creadas, de las que revoca todos los permisos y posteriormente extrae los usuarios a los cuales se les va a permitir el acceso de selección, inserción, borrado, actualización e indexado.

```

.....
/* Programa 1 : Permisos_globales.ec
* Autor   : Cuauhtemoc E. Mendoza Morales y Alejandro Solis M.
* Fecha   : 21 - Mayo - 2002
.....
*/
#include <stdio.h>
#include <stdlib.h>

/* VARIABLES ANFITRIONAS */
EXEC SQL BEGIN DECLARE SECTION:

char buf(300);
char buf1(300);
char nombre_tabla(19);
char nombre_user(11);
char szSQLRevoke(70);
char szSQLGrantS(70);
char szSQLGrantU(70);

```

TESIS CON  
FALLA DE ORIGEN

```

char szSQLGrantI[70];
char szSQLGrantD[70];
char szSQLGrantIndex[70];

char *qry_p =
"select tablename from systables \
where tabid > 99 and \
tablename not in ('xsql_languages','xserver_info','usuario_bd') \
order by tablename";

char *qry_p1 =
"select usuario \
from usuario_bd \
order by usuario";

char *qry_p2 =
"REVOKE all on %s from %s";

struct a{
char usuario[19];
}x[100];

EXEC SQL END DECLARE SECTION;

main()
{
EXEC SQL BEGIN DECLARE SECTION;
char base[16];
char banRevoke[2];
char banSelect[2];
char banInsert[2];
char banUpdate[2];
char banDelete[2];
char banIndex[2];
EXEC SQL END DECLARE SECTION;

int i = 0;
int y = 0;

printf ("Introduce el nombre de la BD : \n");
scanf ("%s",base);
EXEC SQL DATABASE :base;
if (sqlca.sqlcode < 0)
{
printf ( "ERROR: No se pudo Definir la BD: %s\n ", base );
printf ( "   ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
exit(0);
}

printf ("Revocar permisos           si/no \n");
scanf ("%s",banRevoke);

printf ("Permisos de Seleccion           si/no : \n");
scanf ("%s",banSelect);

printf ("Permisos de Actualizacion       si/no : \n");
scanf ("%s",banUpdate);

printf ("Permisos de Insercion           si/no : \n");
scanf ("%s",banInsert);

printf ("Permisos de Borrado             si/no : \n");
scanf ("%s",banDelete);

printf ("Permisos de Creacion de indices si/no : \n");
scanf ("%s",banIndex);

/* SELECCION DE USUARIOS. */

```



```

sprintf(buf1, qry_p1);
EXEC SQL PREPARE SQL_user FROM :buf1;

EXEC SQL DECLARE cur_user CURSOR FOR SQL_user;

EXEC SQL OPEN cur_user;

if ( sqlca.sqlcode < 0 )
{
  printf ( "ERROR: No se pudo abrir cursor 'cur_user'\n " );
  printf ( "   ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
}
/* FINALIZA DECLARACIÓN DE SELECT DE USUARIOS */

EXEC SQL FETCH cur_user
INTO :x[i].usuario;

while ( !sqlca.sqlcode )
{
  ++i;
  EXEC SQL FETCH cur_user
INTO :x[i].usuario;
}

/* SELECT PRINCIPAL. */
sprintf(buf, qry_p);
EXEC SQL PREPARE SQL_chcp FROM :buf;

EXEC SQL DECLARE cur_chcpto CURSOR FOR SQL_chcp;

EXEC SQL OPEN cur_chcpto;

if ( sqlca.sqlcode < 0 )
{
  printf ( "ERROR: No se pudo abrir cursor 'cur_chcpto'\n " );
  printf ( "   ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
}
/* FINALIZA SELECT PRINCIPAL */

EXEC SQL FETCH cur_chcpto
INTO :nombre_tabla;

while ( !sqlca.sqlcode )
{
  /* FOR POR USUARIO */
  for (y=0; y<41; y++)
  {
    /* REVOCAR PERMISOS A LOS USUARIOS */
    if (strcmp (banRevoke, "si") == 0)
    {
      sprintf (szSQLRevoke, "REVOKE all on %s from %s", nombre_tabla, x[y].usuario);
      EXEC SQL PREPARE RevokeSP FROM :szSQLRevoke;
      if ( sqlca.sqlcode < 0 )
      {
        printf ( "ERROR: No se pudo preparar la inf.\n " );
        printf ( "   ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
        printf ( "SQL: %s\n\n", szSQLRevoke );
      }
    }
  }

  EXEC SQL EXECUTE RevokeSP;
  if ( sqlca.sqlcode < 0 )
  {
    printf ( "ERROR: No se pudo insertar en 'mant_tabla'\n " );
    printf ( "   ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
  }
}

```

TESIS CON  
FALLA DE ORIGEN



```

printf ( "SQL: %s\n\n", szSQLRevoke );
}
}

/*          PERMISOS DE SELECCION */
if(strcmp (banSelect,"si")==0)
{
    sprintf(szSQLGrantS,"GRANT select on %s to %s",nombre_tabla,x[y].usuario);

    EXEC SQL PREPARE GrantSelect FROM :szSQLGrants;
    if ( sqlca.sqlcode < 0 )
    {
        printf ("ERROR: No se pudo preparar ln ");
        printf ( "    ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
        printf ( "SQL: %s\n\n", szSQLGrantS );
    }

    EXEC SQL EXECUTE GrantSelect;
    if ( sqlca.sqlcode < 0 )
    {
        printf ("ERROR: No se pudo grant ln ");
        printf ( "    ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
        printf ( "SQL: %s\n\n", szSQLGrantS );
    }
}

/*          PERMISOS DE INSERCION */
if(strcmp (banInsert,"si")==0)
{
    sprintf(szSQLGrantI,"GRANT insert on %s to %s",nombre_tabla,x[y].usuario);

    EXEC SQL PREPARE GrantInsert FROM :szSQLGrantI;
    if ( sqlca.sqlcode < 0 )
    {
        printf ("ERROR: No se pudo preparar la inf.\n ");
        printf ( "    ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
        printf ( "SQL: %s\n\n", szSQLGrantI );
    }

    EXEC SQL EXECUTE GrantInsert;
    if ( sqlca.sqlcode < 0 )
    {
        printf ("ERROR: No se pudo grant ln ");
        printf ( "    ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
        printf ( "SQL: %s\n\n", szSQLGrantI );
    }
}

/*          PERMISOS DE ACTUALIZACION */
if(strcmp (banUpdate,"si")==0)
{
    sprintf(szSQLGrantU,"GRANT update on %s to %s",nombre_tabla,x[y].usuario);

    EXEC SQL PREPARE GrantUpdate FROM :szSQLGrantU;
    if ( sqlca.sqlcode < 0 )
    {
        printf ("ERROR: No se pudo preparar la inf.\n ");
        printf ( "    ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
        printf ( "SQL: %s\n\n", szSQLGrantU );
    }

    EXEC SQL EXECUTE GrantUpdate;
    if ( sqlca.sqlcode < 0 )
    {
        printf ("ERROR: No se pudo grant ln ");
        printf ( "    ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
        printf ( "SQL: %s\n\n", szSQLGrantU );
    }
}
}

```

TESIS CON  
FALLA DE ORIGEN

```

/*      PERMISOS DE BORRADO */
if(strcmp(banDelete,"si")==0)
{
    sprintf(szSQLGrantD,"GRANT delete on %s to %s",nombre_tabla,x[y].usuario);
    EXEC SQL PREPARE GrantDelete FROM :szSQLGrantD;
    if ( sqlca.sqlcode < 0 )
    {
        printf ("ERROR: No se pudo preparar la inf.\n ");
        printf ( "      ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
        printf ( "SQL: %s\n\n", szSQLGrantD );
    }

    EXEC SQL EXECUTE GrantDelete;
    if ( sqlca.sqlcode < 0 )
    {
        printf ("ERROR: No se pudo \n ");
        printf ( "      ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
        printf ( "SQL: %s\n\n", szSQLGrantD );
    }
}

/*      PERMISOS DE INDEXADO */
if(strcmp(banIndex,"si")==0)
{
    sprintf(szSQLGrantIndex,"GRANT index on %s to %s",nombre_tabla,x[y].usuario);
    EXEC SQL PREPARE GrantIndex FROM :szSQLGrantIndex;
    if ( sqlca.sqlcode < 0 )
    {
        printf ("ERROR: No se pudo preparar la inf.\n ");
        printf ( "      ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
        printf ( "SQL: %s\n\n", szSQLGrantIndex );
    }

    EXEC SQL EXECUTE GrantIndex;
    if ( sqlca.sqlcode < 0 )
    {
        printf ("ERROR: No se pudo grant\n ");
        printf ( "      ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
        printf ( "SQL: %s\n\n", szSQLGrantIndex );
    }
}
}
/*FINALIZA EL FOR */

EXEC SQL FETCH cur_chcpto
INTO :nombre_tabla;
}

/* Cerrar Conexcion y Salir del Programa. */
EXEC SQL CLOSE cur_user;
EXEC SQL FREE cur_user;
}

```

Programa1 Permite seleccionar, insertar, borrar e indexar por catálogo de usuarios

Para el caso de usuarios que se dan de alta posteriormenete se creo el programa2 el cual permite darle el acceso de selección, actualización, inserción ó borrado para todas las tablas que existen a nivel usuario en la base de datos.



El programa recibe como parámetros de entrada, el nombre de la base de datos, el usuario al que se le van a asignar los permisos, de manera opcional si se le van a dar permisos de selección, inserción, actualización ó borrado. El código de este programa es el siguiente:

```

.....
* Programa2 : Permisos_x_usuario.ec
* Autor   : Cuauhtemoc E. Mendoza Morales y Alejandro Solis M.
* Fecha   : 21 - Mayo - 2002
.....

#include <stdio.h>
#include <stdlib.h>

/* VARIABLES ANFITRIONAS */
EXEC SQL BEGIN DECLARE SECTION;

char buf[300];
char buf1[300];
char nombre_tabla[18];
char tabla[18];
char nombre_user[11];
char szSQLRevoke[70];
char szSQLGrantS[70];
char szSQLGrantU[70];
char szSQLGrantI[70];
char szSQLGrantD[70];
char szSQLGrantIndex[70];

char *qry_p =
"select tablename \
from systables \
where tablename = '%s' \
order by tablename";

EXEC SQL END DECLARE SECTION;

main()
{
EXEC SQL BEGIN DECLARE SECTION;
char base[18];
char banRevoke[2];
char banSelect[2];
char banInsert[2];
char banUpdate[2];
char banDelete[2];
char banIndex[2];
EXEC SQL END DECLARE SECTION;

printf ("Introduce el nombre de la BD : \n");
scanf ("%s",base);
EXEC SQL DATABASE :base;
if (sqlca.sqlcode < 0)
{
printf ( "ERROR: No se pudo Definir la BD: %s\n", base );
printf ( "ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
exit(0);
}

printf ("Introduce el nombre de la tabla : \n");
scanf ("%s",tabla);

printf ("Introduce el nombre del usuario : \n");
scanf ("%s",nombre_user);

printf ("Revocar permisos           si/no \n");
scanf ("%s",banRevoke);

printf ("Permisos de Seleccion         si/no : \n");

```

TESIS CON  
FALLA DE ORIGEN



```

scanf ("%s",banSelect);
printf ("Permisos de Actualizacion      si/no : \n");
scanf ("%s",banUpdate);

printf ("Permisos de Insercion          si/no : \n");
scanf ("%s",banInsert);

printf ("Permisos de Borrado              si/no : \n");
scanf ("%s",banDelete);

printf ("Permisos de Creacion de indices  si/no : \n");
scanf ("%s",banIndex);

/* SELECT PRINCIPAL. */
sprintf(buf,qry_p.tabla);
EXEC SQL PREPARE SQL_chcp FROM :buf;

EXEC SQL DECLARE cur_chcpto CURSOR FOR SQL_chcp;

EXEC SQL OPEN cur_chcpto;

if ( sqlca.sqlcode < 0 )
{
  printf ( "ERROR: No se pudo abrir cursor 'cur_chcpto\n" );
  printf ( "   ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
}

EXEC SQL FETCH cur_chcpto
INTO :nombre_tabla;

if (sqlca.sqlcode == 100)
{
  printf("LA TABLA %s NO EXISTE EN LA BASE DE DATOS\n",tabla);
}

while ( !sqlca.sqlcode )
{

/* Revocar todos los permisos a la tabla */
if(strcmp (banRevoke,"si")==0)
{
  sprintf(szSQLRevoke,"REVOKE all on %s from %s",nombre_tabla,nombre_user);
  EXEC SQL PREPARE RevokeSP FROM :szSQLRevoke;
  if ( sqlca.sqlcode < 0 )
  {
    printf ("ERROR: No se pudo preparar la inf.\n ");
    printf ( "   ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
    printf ( "SQL: %s\n\n", szSQLRevoke );
  }

  EXEC SQL EXECUTE RevokeSP;
  if ( sqlca.sqlcode < 0 )
  {
    printf ("ERROR: No se pudo Insertar en 'mant_tabla\n ");
    printf ( "   ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
    printf ( "SQL: %s\n\n", szSQLRevoke );
  }
}

/* PERMISOS DE SELECCION */

if(strcmp (banSelect,"si")==0)
{
  sprintf(szSQLGrantS,"GRANT select on %s to %s",nombre_tabla,nombre_user);

```

TESIS CON  
FALLA DE ORIGEN



```

EXEC SQL PREPARE GrantSelect FROM :szSQLGrants;
if ( sqlca.sqlcode < 0 )
{
  printf ("ERROR: No se pudo preparar ln ");
  printf ( "   ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
  printf ( "SQL: %s\n\n", szSQLGrantS );
}

EXEC SQL EXECUTE GrantSelect;
if ( sqlca.sqlcode < 0 )
{
  printf ("ERROR: No se pudo grant ln ");
  printf ( "   ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
  printf ( "SQL: %s\n\n", szSQLGrantS );
}

/*PERMISOS DE INSERCIÓN */
if(strcmp (banInsert,"si")==0)
{
  sprintf(szSQLGrantI,"GRANT insert on %s to %s",nombre_tabla,nombre_user);

  EXEC SQL PREPARE GrantInsert FROM :szSQLGrantI;
  if ( sqlca.sqlcode < 0 )
  {
    printf ("ERROR: No se pudo preparar la inf.\n ");
    printf ( "   ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
    printf ( "SQL: %s\n\n", szSQLGrantI );
  }

  EXEC SQL EXECUTE GrantInsert;
  if ( sqlca.sqlcode < 0 )
  {
    printf ("ERROR: No se pudo grant ln ");
    printf ( "   ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
    printf ( "SQL: %s\n\n", szSQLGrantI );
  }
}

/*PERMISOS DE ACTUALIZACIÓN*/
if(strcmp (banUpdate,"si")==0)
{
  sprintf(szSQLGrantU,"GRANT update on %s to %s",nombre_tabla,nombre_user);

  EXEC SQL PREPARE GrantUpdate FROM :szSQLGrantU;
  if ( sqlca.sqlcode < 0 )
  {
    printf ("ERROR: No se pudo preparar la inf.\n ");
    printf ( "   ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
    printf ( "SQL: %s\n\n", szSQLGrantU );
  }

  EXEC SQL EXECUTE GrantUpdate;
  if ( sqlca.sqlcode < 0 )
  {
    printf ("ERROR: No se pudo grant ln ");
    printf ( "   ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
    printf ( "SQL: %s\n\n", szSQLGrantU );
  }
}

/*PERMISOS DE BORRADO*/
if(strcmp (banDelete,"si")==0)
{
  sprintf(szSQLGrantD,"GRANT delete on %s to %s",nombre_tabla,nombre_user);

  EXEC SQL PREPARE GrantDelete FROM :szSQLGrantD;
  if ( sqlca.sqlcode < 0 )
  {
    printf ("ERROR: No se pudo preparar la inf.\n ");

```

TESIS CON  
FALLA DE ORIGEN



```

        printf ( "      ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
        printf ( "SQL: %s\n\n", szSQLGrantD );
    }
}

EXEC SQL EXECUTE GrantDelete;
if ( sqlca.sqlcode < 0 )
{
    printf ("ERROR: No se pudo ln ");
    printf ( "      ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
    printf ( "SQL: %s\n\n", szSQLGrantD );
}
}

/*PERMISOS DE INDEXADO*/
if(strcmp(banIndex,"si")==0)
{
    sprintf(szSQLGrantIndex,"GRANT index on %s to %s",nombre_tabla,nombre_user);

    EXEC SQL PREPARE GrantIndex FROM :szSQLGrantIndex;
    if ( sqlca.sqlcode < 0 )
    {
        printf ("ERROR: No se pudo preparar la inf.\n ");
        printf ( "      ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
        printf ( "SQL: %s\n\n", szSQLGrantIndex );
    }

    EXEC SQL EXECUTE GrantIndex;
    if ( sqlca.sqlcode < 0 )
    {
        printf ("ERROR: No se pudo grant\n ");
        printf ( "      ERR_INFORMIX: %d\n\n", sqlca.sqlcode );
        printf ( "SQL: %s\n\n", szSQLGrantIndex );
    }
}

EXEC SQL FETCH cur_chcpto
INTO :nombre_tabla;
}

/* Cerrar Coneccion y Salir del Programa. */
EXEC SQL CLOSE cur_chcpto;
EXEC SQL FREE cur_chcpto;
}

```

Programa2 Permite Seleccionar, Insertar, actualizar e indexar a determinado usuario

Lo mencionado anteriormente es lo que podemos resaltar en cuanto a la seguridad de la base de datos del SIAPSEP, la cual se apoya como ya se mencionó en la seguridad del sistema operativo Unix.

A continuación y como último punto, nos enfocaremos en la Seguridad de la aplicación del Sistema SIAPSEP como el último nivel a considerar como parte de la seguridad de la información del servidor de base de datos.

TESIS CON  
FALLA DE ORIGEN



#### V.4 SIAPSEP

En lo que a seguridad respecta de la aplicación, es para reforzar los ámbitos anteriores el Sistema SIAPSEP actualmente cuenta con un módulo de Seguridad que veremos más adelante, en el que a través de grupos generales, grupos de usuarios y usuarios se van creando perfiles con la finalidad de controlar y gestionar el acceso al Sistema SIAPSEP como por ejemplo el personal que únicamente desea consultar información de empleados, al personal que realiza las incidencias de personal, al personal que ejecuta el cálculo de la nómina entre otros, esto proporciona seguridad en la integridad de la información así como restricción de acceso a usuarios a la aplicación.

Los procesos en el módulo de S.I.A.P.S.E.P. Seguridad solo podrán ser realizados por el Administrador del Sistema S.I.A.P.S.E.P. quien será la única persona con posibilidades para acceder a cualquier módulo y otorgar el permiso o permisos de acceso a los diferentes usuarios a las aplicaciones del S.I.A.P.S.E.P., siguiendo la filosofía de la seguridad teniendo un administrador en cada rubro.

La esquema representa la forma lógica en como estan organizandos los accesos al Sistema S.I.A.P.S.E.P.

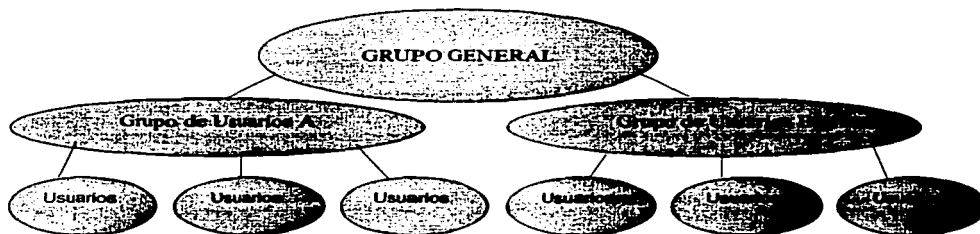


Figura 5.14 Niveles de Seguridad en el Sistema SIAPSEP

Puede haber varios grupos generales y cada uno de estos a su vez tiene grupos de usuarios, por consiguiente estos grupos de usuarios contienen varios usuarios.

TESIS CON  
FALLA DE ORIGEN



La esquema siguiente muestra el módulo del SIAPSEP Seguridad que permite dar mantenimientos de acceso a los usuarios del sistema.

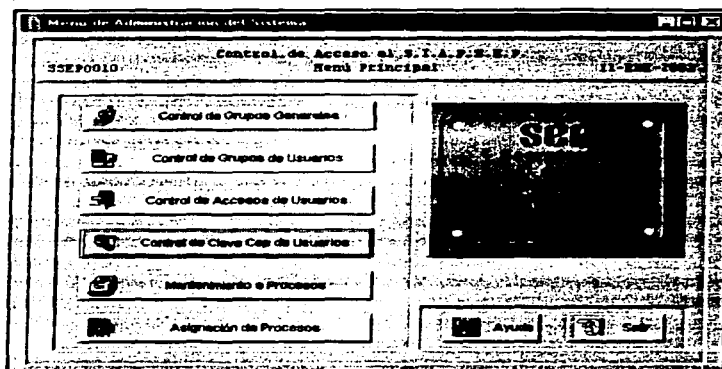


Figura 5.15 Pantalla de Control de Acceso a usuarios

#### *Control de Grupos Generales.*

Los usuarios deben ser clasificados por cada unidad en base a su responsabilidad en el proceso de Pagos y Administración de Personal. Por las razones anteriores podrán existir diferentes Grupos Generales (Figura 5.15) direccionados a diferentes servidores (unidades de red).

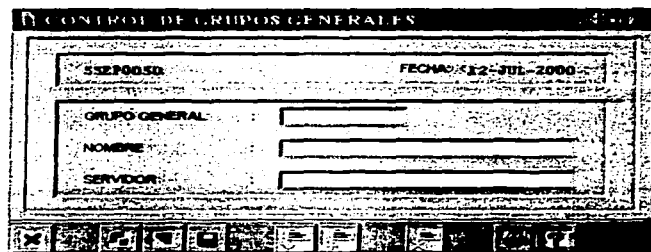


Figura 5.16 Módulo de designación de Grupos Generales



### Control de Grupos de Usuarios

Una vez creado el Grupo General, éste podrá contar con diferentes Grupos de Usuarios según la actividad que éstos realicen dentro del sistema S.I.A.P.S.E.P., (por ejemplo un grupo de Usuarios Capturistas, Auxiliares, Operadores, etc.). Cada administrador deberá encargarse de clasificar la actividad de sus usuarios con el fin de optimizar la administración de su sistema. La figura 5.17 Muestra el módulo de Control de Grupos de Usuarios.

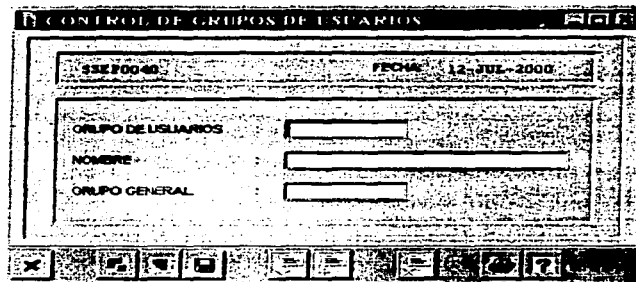


Figura 5.17 Creación de Grupo de Usuarios

### Control de Acceso de Usuarios.

Una vez que se han definido los Grupos de Usuarios Generales y sus Grupos de Usuarios respectivos, el siguiente paso es generar la(s) cuenta(s) del o los usuarios con su(s) respectiva(s) Clave(s) y Password de acceso al sistema.

Cada usuario deberá contar con una clave de acceso única (13 caracteres como máximo) que deberá ser asignada por el administrador del sistema, así como su password (8 caracteres como máximo), por último el nombre del Grupo de Usuarios al que pertenece dicho usuario. La figura 5.18 es el módulo donde se asocia el usuario con el grupo al que va a pertenecer de acuerdo al perfil asignado.

TESIS CON  
FALLA DE ORIGEN



Figura 5.18 Pantalla que define el usuario a que grupo pertenece

### *Mantenimiento a Procesos.*

El módulo de Mantenimiento a Procesos (Figura 5.19) debe contener toda la lista de pantallas que conforman el sistema, como son pantallas de Menú, pantallas de Submenú, Formas de Captura, Reportes, etc.

Figura 5.19 Procedimiento para el Mantenimiento de Procesos

TESIS CON  
FALLA DE ORIGEN



### Alta de una Nueva Forma o Módulo.

#### Asignación de Procesos.

Para la asignación de procesos es requisito haber creado Grupos Generales, Grupos de Usuarios y Control de Acceso a Usuarios (cuentas independientes), respectivamente, así como la declaración de todas las pantallas del sistema en el módulo Mantenimiento a Procesos.

La manera de asignar a un usuario los derechos para acceder a una pantalla o pantallas determinadas es básicamente, establecer dicha relación entre las mismas, este procedimiento deberá de realizarse por cada uno de los usuarios que tendrán acceso al Sistema, esta tarea puede resultar muy tediosa, para abreviar la rutina puede asignarse las relaciones a un Grupo de Usuarios o a un Grupo General, de esta forma todos los Usuarios pertenecientes a este Grupo, quedarán incluidos y por lo tanto gozarán de los mismos derechos del grupo.

#### Procedimiento para la Asignación de Procesos.

Las figuras 5.20, 5.21, 5.22 respectivamente son los recursos que se van dando de alta al sistema para que los usuarios que corresponde puedan acceder.

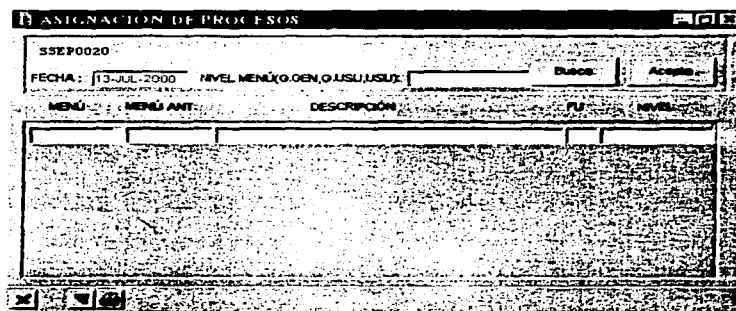


Figura 5.20 Asignación de Procesos

TESIS CON  
FALLA DE ORIGEN



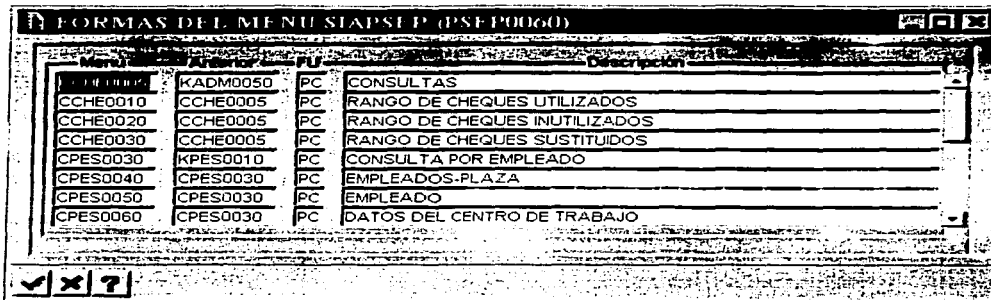


Figura 5.21 Despliegado de Asociación de Procesos

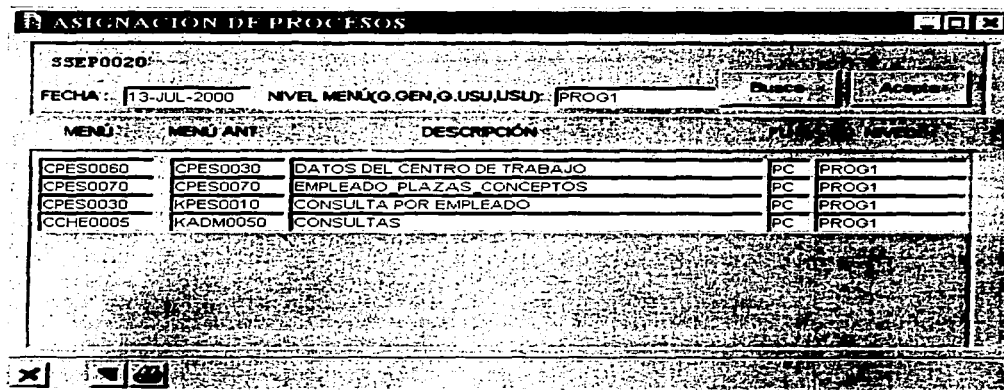


Figura 5.22 Módulo de Consultas a nivel grupo general, grupo de usuarios ó a nivel usuario

Todo lo anterior es lo que corresponde a la seguridad del sistema SIAPSEP en general.

TESIS CON  
FALLA DE ORIGEN

Con este último tema concluimos este trabajo de investigación en donde podemos señalar que la administración es una de las partes primordiales de un sistema y no solo de la base de datos sino de todo tipo de administración que involucra un sistema como es este de nómina (SIAPSEP). Esta tarea a nuestro criterio es una de las más completas y difíciles debido a que todo lo que interactúa en un sistema está ligado, implicando que si una de las partes falla, hay repercusión en los demás procesos.

El ser administrador de una base de datos implica tener un control y seguridad de todo lo contenido en el servidor, el practicar o crear metodologías de administración mediante la automatización de procesos como pueden ser programas de ayuda para agilizar en gran medida la seguridad y el mantenimiento de cada una de las bases de datos contenidas en el servidor.

Otro punto importante es que el administrador debe de conocer los requerimientos de informix, como lo es el sistema operativo sobre el cual opera, archivos de configuración, parámetros de memoria entre otros para poder explotar al máximo lo que el manejador puede ofrecer en cuanto a performance.

Consideramos que el ser administrador de Servidores de Bases de Datos es una actividad muy completa ya que se debe conocer varios aspectos no solo del servidor de base de datos sino que también del sistema operativo, la red, y el propio sistema.

TESIS CON  
FALLA DE ORIGEN

SEGURIDAD EN EL SIAPSEP

## CONCLUSIONES GENERALES

Las necesidades de información de toda organización giran entorno a mantener ésta misma en una forma consistente, actualizada, lo menos vulnerable posible y disponible para todo aquel que necesite accederla. Para lograr todo esto, se ven involucrados muchos elementos y factores que se entrelazan en conjunto para mantener la información disponible el 100 % del tiempo en una organización.

Esta es una de las razones que nos encaminaron a realizar este trabajo de investigación, ya que el administrar un sistema de información como lo es el SIAPSEP, envuelve varios factores, ya que es una labor realmente compleja, desde la instalación, administración y seguridad del sistema operativo donde van a residir las bases de datos, hasta la administración y seguridad del sistema, en nuestro caso el SIAPSEP, así como de las bases de datos que contempla el mismo.

Para que una base de datos pueda interactuar de manera óptima con la aplicación, requiere del sistema operativo ciertos requisitos o condiciones, como lo son: el manejo de archivos sin "buffer", manejo de memoria compartida, manejo dinámico de usuarios, seguridad sobre el sistema de archivos que tiene, etcétera.

El objetivo de hablar de un sistema operativo fuerte y robusto, que se adecuara a las necesidades de un sistema de información como lo es el SIAPSEP, es de gran importancia, ya que es la base medular del sistema, así mismo, la seguridad que proporciona el mismo repercute de manera directa en lo vulnerable que resulte ser el sistema SIAPSEP, en el caso del sistema operativo UNIX, es un sistema operativo muy robusto para un entorno cliente-servidor el cual proporciona un manejo consistente de la información, además de ser capaz de soportar un gran número de aplicaciones críticas, las cuales requieren en la mayoría de los casos de un 90 % disponibilidad de la información cuando se ven involucradas bases de datos, como es el caso de nuestro sistema SIAPSEP.

De igual manera, como parte del primer tema se vio la importancia que tienen las bases de datos como medio de representación de la realidad de una organización, así como lo flexible que representa el manejo de esta y el potencial que tiene el poder explotarla. Así mismo, hoy en día es de gran importancia que al momento de determinar que tecnología utilizar para el manejo de información, se considere la más apropiada que vaya de la mano con la tendencia actual, así como con las necesidades del negocio.

Para la elección de un sistema operativo como plataforma a la hora de implementar un sistema, se contemplan además de lo que se mencionó anteriormente, la disponibilidad que este brinda, así como las capacidades multiusuario y multitarea, para un entorno de trabajo cliente-servidor. En nuestro caso, es el sistema operativo UNIX el que se utiliza como plataforma o núcleo, ya que brinda todas estas capacidades como se resaltaron en el tema 1, y como es sabido, el 80 % de los servidores que existen en Internet son de tipo UNIX, lo cual es un buen indicador.

La seguridad que proporciona el sistema operativo es de vital importancia para la base de datos del sistema, ya que este es el núcleo de toda la aplicación.

## CONCLUSIONES GENERALES

Los servicios de red en el sistema operativo UNIX, están viajando a través de la red de manera insegura, es decir, con un dispositivo de hardware o un tipo de software específico es posible descifrar lo que está pasando a través de la red. De igual manera, por default no es posible obtener un reporte detallado de las conexiones que se llevan a cabo en un servidor UNIX. Por esta razón se propone la implementación del software TCP/Wrappers como medio para registrar todas las conexiones que se lleven a cabo en el servidor donde se encuentra el sistema SIAPSEP. Con esto se garantiza que en el caso de un intento de conexión fallido hacia el servidor, o un acceso no autorizado, se tendría evidencia sobre dicho hallazgo.

Para la elección del manejador de base de datos a utilizar, además de las necesidades del sistema a desarrollar, se consideran las facilidades del negocio para invertir en un software como este, además de conocer cual es la tendencia de la tecnología actual en torno a bases de datos de tipo relacional.

Para el caso del sistema SIAPSEP, se elige trabajar con Informix Dynamic Server como manejador de bases de datos, ya que es un manejador que brinda un gran rendimiento y excelente desempeño en ambientes Unix.

La tecnología que proporciona Informix para administración de bases de datos de tipo relacional es de tipo vanguardista ya que se encuentra implementado en miles de sistemas alrededor del mundo, y su uso esta mas que probado.

En comparación con otros manejadores de bases de datos, como se mencionó en el primer tema, Informix Dynamic Server es el manejador de bases de datos que más se adapta a las necesidades de nuestro sistema SIAPSEP.

Para nuestro sistema una de las necesidades que se tenían era el poder procesar datos en un tiempo mínimo, así como obtener tiempos de respuesta eficientes a consultas críticas, ya que dicho sistema estaba pensado en el pago de nomina a los profesores de la Secretaria de Educación Publica.

Uno de los puntos mas importantes que se tenían contemplados en un inicio, era el demostrar que la administración de un sistema envuelve muchos factores a los que no se les da tanta importancia, o se descuidan, como lo es el sistema operativo, la seguridad del mismo, el mantenimiento que hay que llevar a cabo en él, además de la administración del servidor de base de datos, lo cual repercute de manera directa en lo vulnerable que podría resultar la información en una organización.

Este trabajo aporta elementos que resultan clave para llevar a cabo la administración de un sistema en forma completa y eficiente.

Por otro lado queremos recalcar que el trabajo no es de una sola persona, sino que se deben involucrar tanto los administradores de la red, del sistema operativo, los administradores de las bases de datos y los del aplicativo, para que exista una sincronización de las actividades que se llevan a cabo. Por lo que es recomendable que los administradores de la aplicación tengan un poco de conocimiento sobre bases de datos, a su vez los DBA's deben conocer acerca de las actividades que realizan los administradores del sistema operativo, para que en el caso de una contingencia en la cual se vean involucradas ambas áreas, sepan resolver un problema mas rápidamente. Podemos concluir entonces, lo importante que resultó el resaltar cada uno de los aspectos que envuelven la administración de un sistema, para nuestro caso el

TESIS CON  
FALLA DE ORIGEN



## CONCLUSIONES GENERALES

SIAPSEP, aspectos que hoy en día en una organización no se ven como un todo, sino que se aíslan parcial o completamente.

Por lo que consideramos este trabajo puede servir como fuente de consulta para todo administrador de bases de datos o de sistemas, o a su vez como una guía, ya que las actividades que se resaltan, pretenden dar una visión mas amplia de los aspectos que se descuidan en la administración de sistemas en una organización, por lo que recalcamos nuevamente, se demostró lo importante que resulta el administrar un sistema como el SIAPSEP.

TESIS CON  
FALLA DE ORIGEN





## GLOSARIO

### API

Interfaz de Programación de Aplicación. Lenguaje y formato de mensaje utilizados por un programa para activar e interactuar con las funciones de otro programa o de un equipo físico.

### Batch

Término en inglés que en informática se traduce por los términos 'por lotes', "conjunto" o "grupo". Característica típica de ciertos procesos, que indica una serie de tareas que se ejecutan de forma sucesiva en el ordenador y se consideran como una unidad.

### Blobspace

Es una unidad de almacenamiento lógica compuesta de una o más chunks que almacenan datos únicamente de TEXTO y BYTE, este almacenamiento es de la manera más eficiente posible. Este tipo de dispositivo de almacenamiento guarda toda la información en disco sin pasar a la memoria compartida. De igual manera ayuda a la administración de este tipo de información ya que debido a que es en ocasiones muy grande, el manejarla en un dispositivo común, sería muy lenta su recuperación y almacenamiento.

### Buffer

Espacio reservado de memoria. Las posiciones de memoria que ocupa son consecutivas, por lo que podemos indexarlo.

### Datablade

Todo DataBlade está compuesto por una colección de objetos de Base de Datos y código que extienden la funcionalidad del Gestor Relacional Informix Dynamic Server. En cierta forma puede considerarse como una librería de clases C++ o Java que encapsula tipos de datos especializados, tales como imágenes, textos, etc. Generalmente, la construcción de un DataBlade implica la definición y el desarrollo de nuevos tipos de datos, rutinas, interfaces, tablas e índices y código de cliente.

### Datamart

Es un subconjunto de una bodega de datos para un propósito específico (e.g., un datamart financiero, uno de marketing, etc.).

Se puede ver como una vista de la bodega de datos orientada a un aspecto de un negocio, con un tiempo de vida reducido (e.g., 3 años).

Su función es apoyar a otros sistemas para la toma de decisiones.

Un datamart debe de permitir queries de muchas formas usando herramientas OLAP.

Para el proceso de construcción de bodegas de datos existen dos enfoques:

Construir primero un núcleo de la bodega de datos y luego hacer varios datamarts

Construir primero un datamart e ir expandiendo poco a poco el conjunto de datos y añadir nuevos Datamarts

### Dataware house

Repositorio completo de datos de la empresa, donde se almacenan datos estratégicos, tácticos y operativos, con objeto de obtener operación estratégica y táctica.

### DbSPACE

Es el espacio en donde el servidor de BD almacena los datos. Así mismo es la unidad de almacenamiento que proporciona la unión entre almacenamiento físico y lógico. Permite añadir partes físicas como lo son los chunks como unidades lógicas en este caso tablas. Los dbspaces pueden ser de distintas funciones como puede ser el dbSPACE inicial (root), dbspaces temporales y de datos.

### DDS

Puede considerarse como un sistema que se basa en un almacén de datos y crea una base de datos multidimensional, permitiéndole al usuario procesar analíticamente la información en línea (OLAP), con la habilidad de poder girar y taladrar dentro de los datos. Como utilidad al usuario final, un DSS se valora cuando se profundiza en la información para conocer los "porqués a" los indicadores presentados, pero la infraestructura y metodologías que soporten la explotación de información son las que completan el esquema de un DSS y le permiten hacer uso de OLAP, Dataware house y otros conceptos asociados.

### ExtSPACE

Es un nombre lógico asociado con una cadena arbitraria que significa la localización de datos externos. El recurso que el extSPACE referencia depende del método de acceso que el usuario tiene definido para el contenido. Por ejemplo un usuario requiere acceder a un archivo binario con un formato especial. Primeramente el desarrollador crea un método de acceso para ese recurso mediante rutinas de acceso a datos que van a permitir la interacción entre la base de datos y el archivo externo. Un DBA tiene que añadir un extSPACE que tendrá la etiqueta del archivo en la base de datos. Cada que se crea una tabla en el extSPACE, el usuario puede acceder al archivo vía sentencias SQL.

### Memoria compartida

Las utilidades de memoria compartida permiten crear segmentos de memoria a los que pueden acceder múltiples procesos, pudiendo definirse restricciones de acceso (sólo lectura). Para trabajar con un segmento de memoria compartida, es necesario crear un vínculo (attachment) entre la memoria local del proceso interesado y el segmento compartido.

### Microordenador

Sistema de cómputo, el cual está presente en todas las facetas de la vida moderna, de su uso en la industria y comercio a los juegos recreativos, pasando por las aplicaciones domésticas, por las transacciones financieras, por la automatización y finalmente por las telecomunicaciones.

### Multiproceso

Las computadoras que tienen más de un CPU son llamadas multiproceso. Un sistema operativo multiproceso coordina las operaciones de las computadoras que poseen más de un procesador. Ya que cada CPU en una computadora de multiproceso puede estar ejecutando una instrucción, el otro procesador queda liberado para procesar otras instrucciones simultáneamente. Al usar una computadora con capacidades de multiproceso incrementamos su velocidad de respuesta y procesamiento.

### Multitarea

El término multitarea se refiere a la capacidad del SO para correr más de un programa al mismo tiempo. Existen dos esquemas que los programas de sistemas operativos

TESIS CON  
FALLA DE ORIGEN

## GLOSARIO

utilizan para desarrollar SO multitarea, el primero requiere de la cooperación entre el SO y los programas de aplicación.

### **Multiusuario**

Un SO multiusuario permite a más de un usuario acceder una computadora. Claro que, para llevarse esto a cabo, el SO también debe ser capaz de efectuar multitareas. Unix es el Sistema Operativo Multiusuario más utilizado.

### **Multithread**

Se refiere a que el sistema operativo puede realizar varios procesos, por ejemplo, administrar en forma separada la memoria y en forma separada el manejo de las unidades de disco, todo esto al mismo tiempo.

### **ODBC**

ODBC es un término que significa conectividad abierta a bases de datos, es una librería de funciones de acceso a datos diseñada para ofrecer a las aplicaciones, un proceso estandarizado a las fuentes de datos.

### **OLTP (OnLine Transaction Processing)**

Sistema que procesa transacciones en el momento que se reciben, y actualiza sus archivos principales.

### **Overhead**

### **Protocolo**

Se refiere a un conjunto de reglas las cuales definen la forma en que se va a establecer un proceso.

### **Sbspace**

Es una unidad de almacenamiento lógica compuesta de uno o más chunks que almacena smart large objects. Smart large objects consiste de tipo de datos CLOB (character large objects) objetos de tipo carácter muy grandes y BLOB (binary large object) objetos de tipo binario muy grandes. En ocasiones también pueden ser almacenados los tipos de datos definidos por el usuario en sbspaces.

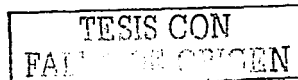
### **Sistema operativo**

Un Sistema Operativo es una parte importante de cualquier sistema de computación. Un sistema de computación puede dividirse en cuatro componentes: el hardware, el Sistema Operativo, los programas de aplicación y los usuarios.

Los Sistemas Operativos son ante todo administradores de recursos; el principal recurso que administran es el hardware del computador; además de los procesadores, los medios de almacenamiento, los dispositivos de entrada/salida, los dispositivos de comunicación y los datos.

### **Sockets**

Los sockets son puntos finales de enlaces de comunicaciones entre procesos. Los procesos los tratan como descriptores de ficheros, de forma que se pueden intercambiar datos con otros procesos transmitiendo y recibiendo a través de sockets.



## GLOSARIO

El tipo de sockets describe la forma en la que se transfiere información a través de ese socket.

### Software

El software es el conjunto de programas que las computadoras emplean para manipular datos. Sin el software, la computadora sería un conjunto de medios sin utilizar. Al cargar los programas en una computadora, la máquina actuará como si recibiera una educación instantánea; de pronto "sabe" como pensar y como operar.

### Tblspaces

Contiene todos los espacios de disco asignados a una tabla o a un fragmento de una tabla en caso de que esta esté fragmentada. Contiene los siguientes tipos de paginas:

- Asignación de paginas a datos
- Asignación de páginas a índices
- Páginas para el uso de BYTE ó TEXTO

### TCP/IP

TCP/IP es un conjunto de protocolos diseñados para las redes de Area Amplia (WAN). El protocolo TCP/IP está conformado por un modelo de cuatro capas: Interface de Red, Red, Transporte y Aplicación.

TESIS CON  
FALLA DE CARGEN



## BIBLIOGRAFIA

- **IBM Informix Administrator Reference.**  
Informix Dinamic Server 9.2  
August 2001  
Part No. 000-8323  
Informix Corporation
- **Administrator Guide fro Informix Dinamic Server.2000**  
Versión 9.2  
September 1999  
Part No. 000-6202  
Informix Corporation
- **Data Security**  
2001 International Business Machines Corporation  
Modulo 16  
Informix Corporation
- **Archive and Backup Guide**  
Informix Dinamic Server Developer Edition  
Informix Dinamic Server , Workgroup Edition  
Informix Corporation
- **Informix Guide to Database Design and Implementación**  
Informix Dinamic Server Versión 9.20  
February 2001  
Part No. 000-4364  
Informix Corporation
- **Informix Guide to SQL**  
Informix Dinamic Server 2000.Versión 9.2  
December 1999  
Part No. 000-6526  
Informix Corporation
- **INFORMIX-ESQL/C**  
Programmer's Manual  
Versión 9.21  
April 1999  
Part No. 000-5424  
Informix Corporation
- **Performance Guide for Informix Dinamic Server 2000**  
Versión 9.2  
September 1999  
Part No. 000-6213  
Informix Corporation

TESIS CON  
FALLA DE ORIGEN

- **Manual de Usuario del Sistema Integral de Administración de Personal de la Secretaría de Educación Pública.**  
Dirección General de Tecnología de la Información  
Ed. 2002
- **Solaris 2.x System Administration I**  
Enterprise services  
January 1999
- **Solaris 2.x System Administration II**  
Enterprise services  
January 1999
- **Seguridad en Unix. Sistemas Abiertos e Internet**  
A. Ribagorda, A. Calvo y M. A. Gallardo  
Ed. Paraninfo, 1996

<http://www.seguridad.unam.mx/>

<http://www.openssl.org/>

<http://www.openssh.com/>

<http://seguridad.diatel.upm.es/SeguridadUNIX.htm>

<http://www.asc.unam.mx>