



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO
Facultad de Ingeniería

**Actualización tecnológica de equipos IBM
del modelo H50 a modelo System P5 52A**

**INFORME DE TRABAJO PROFESIONAL
QUE PARA OBTENER EL TÍTULO DE INGENIERA EN
COMPUTACIÓN PRESENTA:**

Alma Díaz Vázquez

Codirector: M. I. Aurelio Adolfo Millán Nájera



México, D.F. 2009

Índice

Agradecimientos.....	3
Introducción.....	4
Objetivo General.....	4
Objetivos específicos.....	4
Marco teórico.....	5
Antecedentes del proyecto.....	9
Definición del problema.....	11
Desarrollo.....	14
Análisis y metodología empleada.....	19
Instalación de productos Informix.....	27
Migración de información.....	33
Resultados y aportaciones.....	45
Conclusiones.....	45
Bibliografía.....	46
Anexo A.....	47

AGRADECIMIENTOS

Siempre pensé que sería fácil escribir una página de agradecimientos, creí vanidosamente que todo estaba “bajo control” para este momento y no es así. Sin embargo de cualquier manera deseo de todo corazón aunque sea torpemente, agradecer a muchas personas por el apoyo para la conclusión de este trabajo que finalmente me permitirá obtener mi título profesional.

Tengo que seguir un orden pero no quiero que se entienda que la gente que nombre antes parezca que es más importante que el resto, todos tienen el mismo valor para mí en estos momentos.

Agradezco mucho mis padres que me dieron las facilidades y la oportunidad de seguir una carrera universitaria, a mis hermanos que me han apoyado durante la realización de este trabajo y han creído en mí siempre.

Al Ing. Aurelio Adolfo Millán Nájera, por aceptar ser mi director universitario y ayudarme a corregir el trabajo hasta que quedara listo. Al Ing. Gustavo Torres Morales por ser mi aval externo y estar disponible todo el tiempo necesario.

A mi amigo Germán Costumbre, quién me ayudó con detalles de formato de la presentación, me respaldó y alentó todo el tiempo y se entusiasmó conmigo a la conclusión de este trabajo. A mis amigos Verónica Ramírez, Javier Casarrubias, Manuel Bautista, René Calderón que siempre estuvieron al tanto de mis progresos y con sus comentarios me impulsaban a seguir. A mis amigos y colaboradores cercanos Pedro Linares, Miguel Félix, Antonio Hernández que me apoyaron en el desarrollo del proyecto objeto de este trabajo.

A todos mis profesores de la Facultad de Ingeniería por aportar sus conocimientos en mi beneficio. Y a todas las personas que intervinieron de alguna manera en todos los trámites.

Además quiero dedicar el esfuerzo para la obtención del título profesional a todos ellos que son gente importante y muy querida; mil gracias por su amor y por su entrega en cada uno de los papeles que tienen en mi vida.

Alma.

Actualización tecnológica de equipos IBM del modelo H50 a modelo System P5 52A

Introducción.

Debido a que los equipos de cómputo que se desarrollan día con día cuentan con mayores capacidades de cómputo, los usuarios de ellos tienen posibilidades cada vez más variadas de elegir y mejorar su poder de cómputo para que el desempeño de sus actividades sea más acorde con los servicios que ofrecen y puedan proporcionar información de manera mucho más ágil, certera y precisa.

Este proyecto fue la propuesta de actualización a uno de nuestros clientes, para el cual el poder de cómputo es una de las características más importantes para su negocio. La propuesta en términos generales estuvo basada en las características del servicio que ofrece el cliente y en las características técnicas que facilitaban sus labores.

Se consideró también que la inversión incluyera los beneficios necesarios para que el cliente tuviera todo el soporte técnico necesario durante la etapa de estabilización de los equipos nuevos con la operación normal.

Para esto se realizó la propuesta de sustitución de equipos IBM modelo H50, por equipos modelo Series P5. El objetivo central de este proyecto fue la exitosa migración de información del cliente de un equipo al otro.

Objetivo General.

Propuesta de diseño de un procedimiento automatizado de migración de bases de datos relacionales en Informix para AIX y aplicaciones considerando los modelos y normas de seguridad actuales del cliente de un equipo antiguo a uno actualizado.

Objetivos específicos.

- Analizar el volumen de información de las bases de datos a fin de determinar el espacio necesario para la descarga de información.
- Clasificar la información histórica y la información dinámica de las bases de datos de acuerdo a los estándares actuales de la institución con los esquemas utilizados con Informix.
- Realizar la proyección del espacio necesario para la información histórica y dinámica para la migración a los equipos nuevos.
- Preparar el equipo nuevo con las nuevas versiones de software base para la administración de aplicaciones y bases de datos.
- Realizar una descarga y carga masiva de información que minimice la pérdida de información con las herramientas actualmente disponibles en la institución.
- Ajustar los parámetros de operación a las nuevas condiciones de equipo y software base.

- Minimizar tiempos fuera de producción derivados de mantenimientos preventivos y/o correctivos.
- Garantizar la integridad de la información residente en los equipos de producción.

Marco teórico.

Bases de datos

Una base de datos es una colección integrada de datos almacenados en distintos tipos de registros y relacionados de diversas formas de manera que sean accesibles para múltiples aplicaciones. En una base de datos las definiciones de los datos y las relaciones entre ellos están separadas de las declaraciones de procedimientos de un programa.

Para representar las entidades y relaciones de una base de datos se utiliza un Modelo de Datos, el cual permite representar de manera conceptual la organización de los datos. Este Modelo de Datos es independiente del DBMS y de la representación gráfica de los datos. Existen tres modelos de datos más utilizados: Relacional, Jerárquico y Reticular (o de RED), el tipo más utilizado actualmente es el modelo relacional, ya que las bases de datos cada vez se orientan más a objetos.

Modelo relacional:

En el modelo relacional las entidades y sus relaciones se representan con tablas bidimensionales. Las relaciones se consideran también como entidades, cada tabla representa una entidad y está compuesta de renglones y columnas. Las columnas de las tablas representan campos de datos y los renglones representan los registros de los datos. A una columna o conjunto de columnas que identifican de manera única a cada registro de la tabla se le llama “campo llave” o “clave”.

Propiedades de las bases de datos relacionales:

- No debe haber renglones duplicados.
- El orden de los renglones no es significativo.
- El orden de las columnas no es significativo.
- Provee mecanismos para asegurar la integridad de los datos.
- Permite acceso concurrente a datos.

Ventajas:

- Reduce al máximo la redundancia de datos.
- Asegura la integridad de los datos.
- Obliga la seguridad de los datos.
- Puede tener acceso concurrente a los datos.
- Soporta datos compartidos.

Sistema de administración de bases de datos (DBMS).

El software, hardware y los procedimientos para manejar la base de datos conforman un Sistema de Administración de Bases de Datos (DBMS por sus siglas en inglés). Un DBMS hace posible almacenar y acceder a la información contenida en una base de datos proporcionando seguridad a través de un conjunto de programas y una colección de datos interrelacionados. Si los datos van a ser compartidos por varios usuarios, el sistema debe evitar posibles resultados anómalos. Ejemplos de manejadores de bases de datos son: Informix, Oracle, Postgres y Sybase.

Los datos se almacenan en estructuras que se identifican como tablas, son arreglos de una o más dimensiones que representan una entidad en un esquema relacional. Una entidad es una persona, lugar, cosa evento u objeto acerca del cual se registra información. Los elementos que forman una tabla son:

Registros: son las diferentes instancias de la entidad que representa la tabla.

Campos: son los atributos que tiene la entidad.

Índices: son estructuras de la tabla que permiten acceder directamente a los datos de la misma, esto agiliza las búsquedas y se generan de manera más adecuada dependiendo de las relaciones entre tablas y de las aplicaciones que explotan la información. Para permitir el acceso aleatorio rápido a los registros de una tabla se utiliza una estructura de índice. Cada estructura de índice está asociada con una clave de búsqueda determinada. El índice es almacenado en una estructura adicional a la tabla y contiene la referencia a la posición en que se encuentran registros específicos dentro de la tabla.

El DBMS utilizado para este proyecto es Informix; es un administrador de bases de datos que particularmente explota los sistemas, multiprocesadores, usa memoria compartida y comunicación con cada uno cuando sea necesario.

Los 3 grandes componentes de la arquitectura del servidor de base de datos Informix son:

- Los procesos virtuales
- La memoria compartida
- Los discos

Procesador virtual.

Los procesadores virtuales son procesos servidores de base de datos, que funcionan igual que el CPU de una computadora personal. Así como un CPU corre múltiples procesos para servir a múltiples usuarios, un procesador virtual ejecuta múltiples threads para atender múltiples aplicaciones cliente SQL.

Thread.

Un thread es una pieza de trabajo para un procesador virtual de la misma forma que el procesador virtual es una pieza de trabajo para el CPU.

Tipos de procesadores virtuales (VP).

Los procesadores virtuales están divididos en clases basadas en el tipo de procesamiento que realizan. Cada clase de procesador virtual esta dedicado a cierto tipo de procesamiento de threads. Un VP puede pertenecer solo a una clase y una clase puede tener uno o más VP's. Existen clases de VP tales como CPU, AIO,PIO y LIO entre otros.

Memoria compartida.

La memoria compartida es una característica del sistema operativo que permite a los threads y procesos servidores de base de datos compartir datos de un pool de memoria, con los siguientes propósitos:

- Reducir el uso de memoria y E/S a disco.
- Ejecutar comunicación a alta velocidad entre procesos.

~

El servidor de base de datos crea las siguientes porciones de memoria compartida:

- Porción residente
- Porción virtual
- Porción de mensajes

Cada porción de la memoria compartida consume uno o más segmentos de sistema operativo. Cuando el servidor de base de datos inicializa la memoria compartida asigna al menos dos segmentos, uno para la porción residente y otra para la porción virtual.

Porción de memoria residente:

Contiene el buffer pool y otra información del sistema. Puede ser configurada para permanecer residente en memoria principal si el equipo lo permite.

Unix, cuando intercambia los procesos que corren en el sistema, normalmente "intercambia" el contenido de porciones de memoria a disco. Cuando una porción de memoria es designada como residente, esta no se "intercambia" a disco. Los datos mas frecuentemente accedidos residen en memoria, para mejorar el rendimiento pues reduce el I/O a los discos.

Aquí se encuentran los llamados LRU'S, los cuales ayudan a la administración de los buffers y al mejor aprovechamiento de la memoria y así evitar acceso innecesario a disco.

Physical-log Buffer:

El servidor de base de datos usa el buffer de log-físico para retener (before-image) imágenes-previas de páginas de datos e índices que están en el buffer pool antes de ser actualizadas, funcionando como un mecanismo tolerante a fallas.

El buffer de log-físico se compone de dos buffers, esto permite que los procesos del servidor de base de datos escriban al buffer de log-físico activo mientras el otro buffer esta siendo vaciado al archivo de log-físico en disco.

Porción de Memoria Virtual:

Contiene información acerca de threads, sesiones y los datos que son usados por ellos. Esta información aumenta y disminuye constantemente, así que el servidor es responsable de administrar la memoria de esta porción.

Cuando el servidor necesita memoria para un propósito específico, reserva memoria sin usar dentro de la porción virtual. Esta memoria es conocida como un pool de memoria compartida.

El servidor maneja la asignación de pools de memoria o bloques, siendo asignados en cualquier momento y reubicados cuando estos ya no son necesarios.

La porción virtual de memoria compartida es usada para muchos propósitos incluyendo:

- Datos de la sesión del usuario
- Información del diccionario (catálogo del sistema)
- Procedimientos almacenados (SQL y SPL)
- Información del thread
- Ordenamientos
- Información diversa

Porción de Mensajes

La porción de memoria de mensajes o de comunicación es alojada cuando la memoria compartida es inicializada. Contiene los buffers de mensajes para clientes locales que usen la memoria compartida para comunicarse con el servidor.

Las aplicaciones y el servidor apuntan al mismo segmento de memoria. El cliente deja los mensajes en esta porción y recoge los que el manejador deja para el.

El tamaño de la porción de mensajes es de aproximadamente 12k por cada usuario que usa memoria compartida para comunicarse con el servidor.

Componentes de disco para Informix.

El servidor de base de datos puede usar archivos de sistema operativo para almacenar datos, o si el sistema soporta disco raw (sin formato del sistema operativo), Informix recomienda el uso de ellas para almacenar datos para lograr mayor rendimiento y consistencia.

El servidor usa las siguientes unidades para el manejo del espacio en disco:

- Página
- Extent
- Tblspace
- Chunk
- Dbspace

Extent. Los extents son un grupo de páginas contiguas, el tamaño de los extents para una tabla, se define al momento de crearla de la siguiente manera:

```
CREATE TABLE tabla  
(  
  campo1 tipo,  
  .  
  .  
  CampoN tipo) EXTENT SIZE 100 NEXT SIZE 25
```

Página. Una página es la unidad física de almacenamiento en disco que el servidor de base de datos usa para leer y/o escribir a bases de datos Informix.

El tamaño de la página varía de computador a computador. Una página típicamente retiene 2 o 4 Kbytes, ya que el hardware determina el tamaño de la página, no se puede alterar el valor. Existen páginas de datos, índices y reservadas entre otras.

Informix cuenta con diferentes herramientas que permiten respaldar, cargar y descargar información de manera rápida y confiable, algunos de esas herramientas son: ontape, onbar, hpload, onload, onunload, dbimport y dbexport. Mismas que fueron evaluadas para elegir la más adecuada a usar en el proceso.

Antecedentes del proyecto.

Durante mi desarrollo profesional he tenido la oportunidad de permanecer en el área de Sistemas Computacionales todo el tiempo, sin embargo para los

propósitos de este trabajo solamente describiré el último empleo, mismo que ejerzo hasta el día de hoy.

El puesto en el que me he desarrollado los últimos diez años es el de Especialista en Apoyo Técnico, actualmente en el nivel de coordinación de área. Las actividades están enfocadas a la Administración de Bases de Datos y son principalmente las de análisis y solución de problemas reportados por los usuarios finales. Sin embargo el puesto incluye actividades tales como cálculos de espacio para objetos nuevos de bases de datos, monitoreo de desempeño, análisis de resultados de monitoreo, elaboración de propuestas de afinación de parámetros para el manejador de bases de datos, monitoreo de la utilización de espacio de almacenamiento de bases de datos, elaboración de proyecciones de utilización de espacio, propuestas para asignación de espacio para bases de datos, análisis de métodos de fragmentación para tablas grandes, revisión de auditorías de sistemas realizadas por el proveedor del manejador de bases de datos, evaluación de recomendaciones de afinación de parámetros, aplicación de las recomendaciones aceptadas, elaboración de reportes gerenciales de los problemas presentados y las soluciones aplicadas, así como del avance de proyectos del área. Participación en la planeación de proyectos y elaboración de planes de trabajo, realización de los mismos y capacitación de personal a mi cargo.

A mediados de la década de los 90's uno de nuestros clientes utilizaba equipos UNISYS U/6000 para albergar la información en las 8 ubicaciones que maneja. En esa época se realizó una actualización tecnológica para instalar equipos IBM-H50 que tenían un poder de cómputo mucho mayor, en especial de almacenamiento. Para este fin se contaba con arreglos de discos de 9 GB cada uno, que estaban asignados y configurados en los equipos IBM-H50 de acuerdo con el volumen de información que se concentraba en cada centro.

Los equipos H50 tenían instalado el sistema operativo AIX versión 4.3 (UNIX) y tenían como DBMS Informix versión 7.31. Tanto el equipo como las versiones de sistema operativo y manejador de bases de datos son bastante estables, se llegó a tal nivel de estabilidad que la administración se limitaba solamente a cuidar los recursos de memoria y almacenamiento de información que precisamente tenían muchas incidencias.

Como no se autorizaba una nueva actualización tecnológica, en el año 2003 se realizó una depuración de información y recuperación de espacio ocioso para poder continuar almacenando información. Este procedimiento consistió en el respaldo seguro de información y la depuración de la misma, dejando disponibles para efectos de consulta los 5 años más recientes de información en la base de datos.

Finalmente en noviembre de 2006 pensando en los problemas de mantenimiento de equipos obsoletos y la sustitución de discos duros que a últimas fechas se habían dañado con cierta frecuencia debido a su antigüedad, se propuso al cliente sustituir los equipos IBM-H50 por equipos nuevos IBM System P5 cuyo costo

estaría por debajo del costo de mantenimiento preventivo y correctivo del equipo antiguo, según los resultados del análisis financiero correspondiente.

Después de realizar el análisis de poder de cómputo, ventajas y requerimientos nuevos que se muestran en la siguiente sección, se determinó realizar la actualización tecnológica que se detalla en este trabajo.

Definición del problema.

Los equipos IBM-H50 que se utilizaban, tenían arreglos de discos SSA de 9 Gb cada uno, las dimensiones de los arreglos de discos variaban de acuerdo a la configuración de cada localidad. Este espacio en su asignación para bases de datos estaba dividido en chunks de 2 Gb cada uno en los diferentes dbspaces dependiendo la aplicación que lo requiriera. En el manejador de bases de datos Informix, el espacio para almacenamiento de datos está organizado de la siguiente manera: cada instancia bases de datos cuenta con dbspaces para almacenamiento de información, estos dbspaces a su vez están integrados por uno o más chunks, cada chunk es un espacio lógico que está relacionado con un espacio físico de un disco duro del equipo de que se trate. A continuación en la figura 1, se muestra gráficamente el concepto.

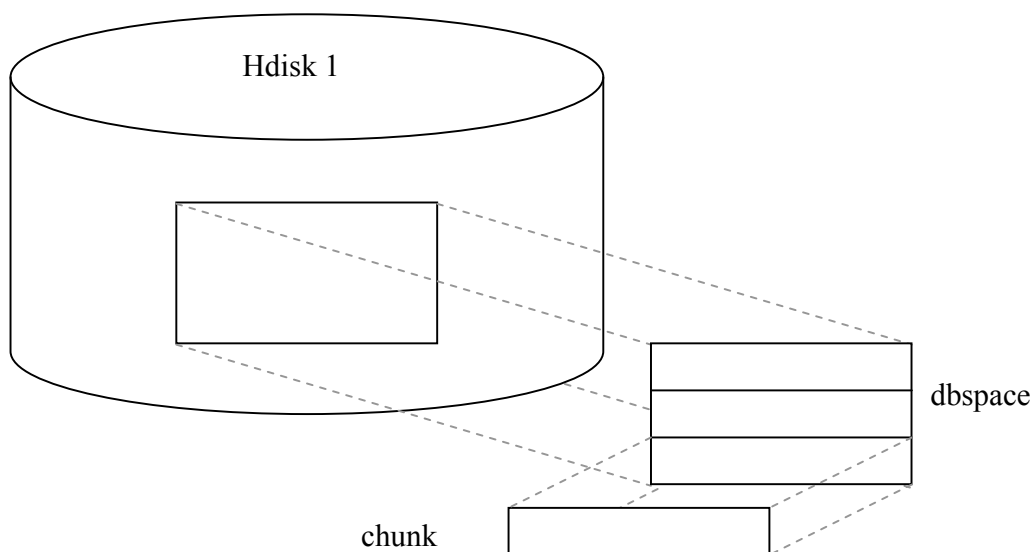


Figura 1. Descripción gráfica de un dbspace y un chunk

El procedimiento de administración de espacio es el siguiente: de acuerdo con las especificaciones de los contenidos de información en las tablas de las bases de datos de las aplicaciones, se realiza un cálculo de espacio requerido proyectado a

un año de acuerdo a los índices de crecimiento estimados por los desarrolladores. Este espacio es calculado de la siguiente manera: como principio se considera el tamaño de registro de la tabla, éste se obtiene de la suma del tamaño de cada campo, cuyo valor depende del tipo, esto es: integer 4 bytes, smallint 2 bytes, char(n) n bytes, varchar(m) m bytes, date 4 bytes, etc.

Debido a que la administración de información es mucho más ágil si las tablas están poco particionadas, se calcula el espacio que utilizará cada tabla considerando el número de registros que se espera almacenar en ella; el tamaño de la página lógica en este caso es de 4Kb; el tamaño del encabezado de la tabla o tblspace con esto se obtiene un valor que es considerado como el "extent inicial". Como medida preventiva en el caso de que el crecimiento de la tabla se vea superado respecto al cálculo inicial, se recomienda considerar un 15% adicional que será calculado como el "next extent" de la tabla.

Un extent es el espacio reservado por anticipado para el almacenamiento de una tabla de aplicación dentro del dbspace correspondiente. Cuando las tablas albergan una gran cantidad de registros, la mejor manera de almacenarlos es en forma continua, para que las búsquedas tengan costos bajos y las aplicaciones se desempeñen en forma adecuada.

Posteriormente se realiza la configuración y asignación del espacio suficiente determinado por el cálculo y por experiencia de acuerdo al monitoreo del consumo de espacio de estas bases de datos, se adiciona un quince por ciento de holgura para contingencia, como se comentó anteriormente.

Una vez en operación se colecta información semanalmente para determinar el crecimiento real de la información y obtener estimaciones de tiempo de vida del espacio aún disponible. Los reportes de utilización de espacio permiten prever los requerimientos reales de las aplicaciones y asignar oportunamente un nuevo chunk al dbspace correspondiente si es necesario.

En este caso los requerimientos de espacio para el año siguiente de operación normal, habían excedido las capacidades del equipo IBM-H50. Por otra parte, durante el último año los discos asignados al almacenamiento de información habían presentado muchos errores físicos, por lo que habían tenido que ser sustituidos; en muchas ocasiones estas sustituciones eran cada vez más difíciles ya que los discos disponibles son pocos en el mercado por estar discontinuados y por lo tanto los precios son muy altos, lo cual no era rentable ni recomendable.

Dadas las condiciones de riesgo en las que se encontraba el cliente, se le propuso realizar una actualización tecnológica de las 8 localidades donde manejaba 2 equipos por cada una con diferentes aplicaciones interrelacionadas entre sí. Como el nuevo equipo tenía grandes capacidades de cómputo, se podrían sustituir los 16 equipos involucrados en la operación por solamente 8 de ellos, realizando las adecuaciones correspondientes en cuanto a organización de la información y soporte de las aplicaciones. Adicionalmente se debería dotar al cliente con 2

equipos para desarrollo de sus aplicaciones y laboratorio de pruebas, con una configuración similar a los equipos de producción.

La propuesta fue realizar el cambio durante los meses de menor impacto en la operación: que de acuerdo a la planeación anual estos meses eran de julio a septiembre del 2007. Sin embargo mientras se logró la autorización, llegó el período pico en la operación de los servidores del cliente y se pospuso la migración hasta el siguiente mes de julio.

Desarrollo.

Para explicar al cliente los problemas a los que nos estábamos enfrentando como proveedores de servicios, se les presentó el cuadro comparativo de la tabla 1 donde se muestran las diferencias, ventajas y desventajas de cada equipo, las características de los equipos adicionales para desarrollo y pruebas y las necesidades de licenciamiento de productos.

RUBRO	IBM-H50	IBM SYSTEM P5 52A
Antigüedad	Obsoleto	Actualizado
Sistema Operativo	Fuera de soporte	Soportado por IBM
Tolerancia fallas	Fallas recurrentes	99% tolerante a fallas
Refacciones	Escasas	Sin problemas de surtido
Continuidad de la operación	En riesgo alto	En riesgo mínimo
Procesador	5 core 332 MHz 8 core 332 MHz	1 core 2.1GHz Power5+
Memoria	1.5 GB 2 GB	2 GB
DD para S.O.	8.5 GB	83 GB
DD para Datos	142 GB RAID1 364 GB RAID1	146 GB RAID1+0 584 GB RAID1+0
Red	2 Ethernet 10/100	2 Ethernet 10/100/1000
Periféricos	CD-ROM DAT Monitor	DVD-ROM DAT Monitor plegable Mouse
S.O.	AIX 4.3	AIX 5.3

Tabla 1. Cuadro comparativo de características de los equipos de cómputo

También se presentaron las especificaciones de las licencias necesarias de acuerdo con los productos que el cliente maneja; ya que son productos institucionales. En la tabla 2, se muestra la propuesta presentada al cliente.

PRODUCTO	VERSIÓN ACTUAL	VERSIÓN PROPUESTA	TIPO DE LICENCIA
S.O AIX	4.3	5.3	Licencia de uso incluida Licencia de soporte durante la duración del contrato
Informix	7.3	9.4	Licencia corporativa
Tuxedo	5.3	8.2.54	Licencia corporativa
Tivoli	4.1	4.1	Licencia corporativa

Tabla 2. Tablas de versiones de productos y licencias propuestas.

Como parte de esta propuesta se realizó el plan de trabajo global para determinar el tiempo que sería necesario utilizar por cada localidad a fin de negociar la ventana de tiempo necesaria para realizar este procedimiento.

En el plan de trabajo que se muestra en la tabla 3, se incluye una columna donde se indican las actividades en las que participé.

Actividad	Descripción	Duración (días)	Actividades en las que participé
	Migración Servidores IBM	129d	
1	• Análisis	42.3d	
2	▪ Análisis de factibilidad	25d	X
3	▪ Presentación de propuesta	0.3d	X
4	▪ Aceptación de propuesta	0.3d	
5	• Generación de planes de trabajo específicos	0.5d	
6	▪ Sistemas operativos	0.5d	
7	▪ Bases de datos	0.5d	X
8	▪ Aplicaciones	0.5d	
9	▪ Ubicación de equipos, soporte, respaldos	0.5d	X
10	• Construcción	53.5d	
11	▪ Generar lista de aplicaciones y programas fuente	2d	
12	▪ Generar documento de entrega de código a proveedor	1d	
13	▪ Entrega de aplicaciones y programas fuente	3d	
14	▪ Análisis de programas	3d	
15	▪ Revisión de palabras reservadas	5d	X
16	• Equipo de apoyo para la migración	44.25d	
17	▪ Instalación de equipo de desarrollo	1d	X
18	▪ Instalación de software y licencias	0.5d	X
19	▪ Generación de estructura de BD	0.5d	X
20	▪ Migración de información	3d	X
21	▪ Migración de usuarios y grupos	0.25d	
22	▪ Migración de políticas de seguridad	0.25d	
23	▪ Migración de estructura de directorios	0.25d	
24	▪ Entrega de análisis de impacto	1d	
25	▪ Definir estrategia a seguir	1.5d	
26	▪ Modificaciones a programación	24.5d	
27	▪ Generación de esquemas para réplicas, sinónimos, vistas	7d	X
28	▪ Recompilación del socket de seguridad	10d	
29	▪ Compilación	5d	
30	▪ Pruebas internas (aplicaciones y respaldos)	6d	X
31	▪ Definición de pruebas con el usuario	5d	
32	▪ Control de versiones	1d	
33	▪ Pruebas de validación con el usuario y laboratorio	6d	X
34	• Prueba Piloto	15.95d	
35	▪ Instalación de equipo para prueba piloto	1d	X
36	▪ Instalación de software y licencias	0.5d	X
37	▪ Generación de instancias y bases de datos	0.5d	X
38	▪ Migración de información, usuarios, grupos y estructura de directorios	2d	X
39	▪ Configuración final (ajustes)	1d	X
40	▪ Cambio de direcciones IP	0.05d	
41	▪ Pruebas de funcionalidad	1d	
42	▪ Operación normal	10d	
43	▪ Liberación de la localidad piloto	0.95d	X
44	• Liberación Nacional	36d	
	MIGRACIÓN LOCALIDADES A, B y C	16 d	
45	▪ Instalación de equipo (localidades A, B y C)	1d	X
46	▪ Instalación de software y licencias	0.5d	X
47	▪ Generación de instancias y bases de datos	0.5d	X
48	▪ Migración de información, usuarios, grupos y estructura de directorios	2d	X
49	▪ Configuración final (ajustes)	1d	X
50	▪ Cambio de direcciones IP	0.05d	
51	▪ Operación normal	10d	
52	▪ Liberación de las localidades A, B y C	0.95d	X
	MIGRACIÓN LOCALIDADES D, E y F	16 d	X
	MIGRACIÓN LOCALIDAD G	16 d	X

Tabla 3. Plan de trabajo global.

Nota: las actividades en el plan de trabajo global se repiten para las secciones “Migración de localidades D, E y F” y “Migración de localidad G”

Como en cada localidad se tenían 2 equipos IBM-H50 con diferentes aplicaciones, realicé un plan de trabajo específico para la transferencia de la bases de datos por cada uno de los equipos a migrar, se muestra en la tabla 4 sólo el plan de trabajo del equipo principal ya que con el secundario se haría lo mismo, excepto el punto de replicación de datos ya que en él no se maneja este esquema, por lo tanto sería redundante integrarlo en este trabajo.

Actividad	Descripción	Duración (horas)
	Migración de datos de los equipos Regionales de Recaudación	60.33h
1	• Actividades previas a la migración.	12.08h
2	○ Instalación de sistema operativo en el equipo nuevo AIX 5.3	2h
3	○ Instalación de Informix IDS Ver. 9.40.FC9 con herramientas en el nuevo equipo.	1h
4	○ Generación de la instancia de base de datos de PRODUCCIÓN (1a. Instancia)	2h
5	○ Ejecutar la actualización de estadísticas para optimización de la descarga de información.	3h
6	○ Detener la operación	5m
7	○ Realizar respaldo de nivel cero en modo quiescent del equipo original.	4h
8	• Actividades de migración de información equipos de producción.	40.75h
9	○ Eliminación de réplica de datos con IER.	0.58h
10	▪ Eliminación réplica de datos	5m
11	▪ Eliminación de servidores de réplica (Localidad afectada)	30m
12	• Base de datos de aplicación 1 (App1)	0.92h
13	○ Realizar respaldo de base de datos App1 a file system (dbexport)	20m
14	○ Transmisión del respaldo anterior al nuevo equipo.	15m
15	○ Recuperación de la base de datos App1 en el equipo nuevo.	20m
16	• Base de producción App2	30.5h
17	○ Obtener contadores iniciales de tablas de la base de datos App2	5m
18	○ Obtener esquema de la base de datos App2.	10m
19	○ Separación de esquemas en tablas, índices, constraints y otros objetos.	20m
20	○ Creación de jobs de descarga y carga de información	1h
21	○ Ejecución de jobs de descarga de información	14h
22	○ Verificación de registros descargados	20m
23	○ Creación de la estructura de base de datos en la instancia nueva	15m
24	○ Ejecución de jobs de carga de información	10h
25	○ Obtener contadores finales de tablas de la base de datos App2	5m
26	○ Verificación de contadores iniciales y finales iguales	15m
27	○ Generación de índices, constraints y otros objetos	4h
28	• Base de producción: App3	8.75h
29	○ Obtener contadores iniciales de tablas de la base de datos App3	5m
30	○ Obtener esquema de la base de datos App3.	10m
31	○ Separación de esquemas en tablas, índices, constraints y otros objetos	20m
32	○ Creación de jobs de descarga y carga de información	1h
33	○ Ejecución de jobs de descarga de información	1h
34	○ Verificación de registros descargados	15m
35	○ Creación de la estructura de base de datos en la instancia nueva	15m
36	○ Ejecución de jobs de carga de información	1h
37	○ Obtener contadores finales de tablas de la base de datos App3.	5m
38	○ Verificación de contadores iniciales y finales iguales	15m
39	○ "Generación de índices, constraints, etc."	3h
40	• Definición Réplica de datos con IER.	0.58h
41	○ Definición de servidores de réplica (Localidad afectada)	30m
42	○ Definición réplica de datos	5m
43	○ Pruebas de conectividad con Regionales	0.75h
44	○ Réplica de datos con IER	15m
45	○ Revisión de sinónimos en la instancia central	15m
46	○ Revisión de conectividad desde los equipos centrales	15m
47	• Actividades posteriores a la migración de información	7.5h
48	○ Mantenimiento a las bases de datos migradas.	6h
49	○ Respaldo de nivel cero del equipo migrado	1.5h
50	• Plan de contingencia	2.5 h
51	○ Encendido y redireccionamiento del equipo IBM-H50	2.5 h

Tabla 4. Plan de trabajo para migración de bases de datos.

En este proyecto de migración participé con dos posiciones: líder del proyecto y especialista en bases de datos, por lo que planeé y realicé la mayor parte de las

actividades de migración de información, además de la coordinación y soporte técnico al grupo de trabajo que me apoyó en las diferentes localidades.

Análisis y metodología empleada.

Para realizar el análisis se consideraron varios puntos, como inicio la factibilidad financiera, en la cual no participé. El segundo punto, fue el análisis de requerimientos de la operación y el empate con las características de los nuevos equipos; se verificaron las características de poder de cómputo de varios equipos considerando generar el menor impacto en cuanto a software base y forma de trabajo de las aplicaciones. Una vez determinado que el equipo IBM Series P5, con sistema operativo AIX 5.3 con kernel a 64 bits cubría e incluso excedía las necesidades de operación del cliente; de acuerdo con el análisis de capacidades actuales y futuras de ambos esquemas, se localizó la versión del DBMS soportado en esa plataforma y sus herramientas adicionales para así mismo homogenizarla con las versiones instaladas en los equipos concentradores del cliente. Una vez definida la versión de Informix 9.40.FC9 a 64 bits con sus herramientas correspondientes, estudié las consideraciones para instalación, las mejoras y ventajas del producto para determinar cuáles de ellas podrían ser aprovechadas para el mejoramiento de la operación.

Las ventajas más importantes sobre la versión anteriormente usada, son la posibilidad de usar chunks más grandes ya que la versión 7.31 de Informix estaba limitada a un tamaño máximo de chunk de 2GB, con la versión 9.40 este tamaño se puede incrementar indiscriminadamente habilitando la opción de chunks grandes antes de generarlos; mayor seguridad en el acceso al "system catalog" debido a que en la versión 9.40 de Informix, el acceso está limitado al usuario privilegiado DBA; optimizador mejorado y creación de índices con la metodología B-tree en las versiones anteriores a la 9.40, los índices no se creaban con esta metodología a menos que el DBA o el desarrollador lo especificara en las sentencias de creación de índices, en la nueva versión todos los índices se crean con esta metodología.

Posteriormente revisé las palabras reservadas de la nueva versión, para determinar si la programación de las aplicaciones las incluía y poder indicarlo al grupo de desarrollo para que se realizaran las modificaciones correspondientes de acuerdo con el resultado del análisis; las palabras reservadas son palabras en inglés que identifican procesos internos del DBMS, estas palabras no deben ser utilizadas por las aplicaciones como variables o nombres de funciones. No se encontraron palabras reservadas en la programación.

También el equipo de desarrollo realizó la compilación de todas las aplicaciones en la nueva plataforma de 64 bits ya que se encontraban trabajando a 32 bits; adicionalmente se modificaron algunas referencias a las librerías nuevas de AIX e Informix a 64 bits que serían utilizadas.

Hice una revisión del espacio asignado a las 3 bases de datos más grandes de cada una de las 8 localidades involucradas y determiné hacer un ajuste en el espacio asignado para datos ya que debido a la antigüedad del equipo, había dos situaciones, algunos dbspaces que estaban conformados con segmentos muy pequeños en disco y otros eran sobrados por lo que se desperdiciaba gran parte del segmento asignado.

Debido a que parte de la información que se alberga en esas bases de datos es histórica, la primera actividad fue clasificar las tablas históricas y las dinámicas. Posteriormente, hice una reorganización de los segmentos de tal manera que fueran uniformes en cuanto al tamaño de los chunks que los conforman y que la información estática no tuviera espacio ocioso asignado.

Adicionalmente, como las características de los nuevos equipos eran diferentes hubo que considerar para realizar este ajuste de espacio, que el tamaño físico de la página, que es el tamaño en Kb que el sistema operativo maneja por omisión para seccionar los discos duros en los equipos H50 es de 16 Kb y en los equipos Series P5 es variable en un rango de 8 a 128 Kb, aunque por omisión es de 64 Kb. Después de analizar el tamaño más conveniente para definir la página de acuerdo con los parámetros de lecturas eficientes de la base de datos Informix, se determinó establecer el tamaño de la página en 32 Kb, por lo que el tamaño de los segmentos se ajustó a múltiplos de 32.

Obtuve el reporte de espacio antiguo utilizado, asignado y libre y a partir de él, realicé el análisis del espacio asignado, ocupado y libre que en ese momento se tenía en la instancia; consideré el cálculo de espacio requerido para las tablas de aplicaciones y finalmente obtuve la nueva asignación de espacio para las instancias correspondientes que en la preparación de equipos nuevos se consideraría como la nueva configuración. En la tabla 5 y 6 se muestran los reportes de una localidad, este mismo análisis y cálculo se realizó para cada una de las 8 localidades en los equipos de producción: principal y secundario respectivamente.

Al inicio de este proceso se realizó la preparación de un equipo de pruebas con las mismas características de los equipos que se pondrían en producción, ahí se instaló el software base y se realizaron las actividades de compilación de aplicaciones en plataforma de 64 bits, así mismo se realizaron las pruebas tanto de los procedimientos de migración como de funcionamiento de las aplicaciones, que consistieron en recrear una migración desde un equipo de prueba H50 con bases de datos obtenidas de los equipos de producción realizando todas y cada una de las actividades de migración hasta completar el procedimiento sin fallas. Por otra parte, se hicieron las pruebas de funcionalidad de las aplicaciones con información de casos de prueba con el apoyo de los usuarios corrigiendo fallas como falta de permisos para archivos o tablas de bases de datos, configuración de usuarios y permisos en formas para la presentación de resultados de las aplicaciones.

NOMBRE	RUTA	TAMAÑO (PAGINAS)	LIBRE (PAGINAS)	NOMBRE NUEVO	RUTA NUEVA	TAMAÑO (PAGINAS)	TAMAÑO (KB)	MULTIPLS 32 MB
rootdbs	/dev/linfo/root	38000	29122	rootdbs	/dev/linfo/root	38000	152,000	160,000
dblogfisico	/dev/linfo/rlogfis	12800	7747	dblogfisico	/dev/linfo/rlogfis	12800	51,200	64,000
dblogslogicos	/dev/linfo/rlogslog	118750	34197	dblogslogicos	/dev/linfo/rloglog	118750	475,000	480,000
dbsono	/dev/linfo/runo	256000	75512	dbsono	/dev/linfo/runo1	512000	2,048,000	2,048,000
	/dev/linfo/runo2	512000	502941	dbsixuno	/dev/linfo/rixuno1	512000	2,048,000	2,048,000
dbsixuno	/dev/linfo/rixuno1	512000	346526	dbsdos	/dev/linfo/rdos	76000	304,000	320,000
dbsdos	/dev/linfo/rdos	76000	59613	dbstres	/dev/linfo/rtres	64000	256,000	256,000
dbstres	/dev/linfo/rtres	64000	45579	dbscuatro	/dev/linfo/rcuatro	28000	112,000	128,000
dbscuatro	/dev/linfo/rcuatro	28000	27947	dbsoapp1	/dev/linfo/rdbapp1	12000	48,000	64,000
dbsoapp1	/dev/linfo/rapp1	12000	5539	dbsoapp2mes	/dev/linfo/rapp2mes1	512000	2,048,000	2,048,000
dbsoapp2mes	/dev/linfo/rapp2mes1	512000	196943	dbsixapp2mes	/dev/linfo/rixapp2mes1	512000	2,048,000	2,048,000
dbsixapp2mes	/dev/linfo/rixapp2mes1	256000	91390	dbsixapp2mes	/dev/linfo/rixapp2mes2	75000	300,000	320,000
	/dev/linfo/rixapp2mes2	50000	48470	dbsoapp2	/dev/linfo/rapp21	512000	2,048,000	2,048,000
	/dev/linfo/rixapp2mes3	53248	53245	dbsoapp2	/dev/linfo/rapp22	384000	1,536,000	1,536,000
	/dev/linfo/rixapp2mes4	90112	90109	dbsixapp2	/dev/linfo/rixapp21	512000	2,048,000	2,048,000
	/dev/linfo/rixapp2mes5	131072	131069	dbapp3c	/dev/linfo/rapp3_1	128000	512,000	512,000
dbsoapp2	/dev/linfo/rapp21	512000	41440	dbixapp3c	/dev/linfo/rixapp3_1	128000	512,000	512,000
	/dev/linfo/rapp22	184000	0	dbapp402	/dev/linfo/rapp402_1	512,000	2,048,000	2,048,000
	/dev/linfo/rapp23	60000	27040	dbapp402	/dev/linfo/rapp402_2	512,000	2,048,000	2,048,000
	/dev/linfo/rotro	76000	14230	dbapp402	/dev/linfo/rapp402_3	512,000	2,048,000	2,048,000
dbsixapp2	/dev/linfo/rixapp21	512000	324693	dbapp402	/dev/linfo/rapp402_4	32,000	128,000	128,000
dbsoapp302	/dev/linfo/rapp302_1	128000	91110	dbixapp402	/dev/linfo/rixapp402_1	512,000	2,048,000	2,048,000
	/dev/linfo/rapp302_2	335872	335869	dbixapp402	/dev/linfo/rixapp402_2	512,000	2,048,000	2,048,000
dbsixapp302	/dev/linfo/rixapp302_1	128000	118368	dbixapp402	/dev/linfo/rixapp402_3	512,000	2,048,000	2,048,000
	/dev/linfo/rixapp302_2	176128	176125	dbixapp402	/dev/linfo/rixapp402_4	512,000	2,048,000	2,048,000
dbsoapp307	/dev/linfo/rapp307	4096	1537	dbixapp402	/dev/linfo/rixapp402_5	512,000	2,048,000	2,048,000
dbsixapp307	/dev/linfo/rixapp307	4096	4043	dbapp405	/dev/linfo/rapp405_1	393,216	1,572,864	1,600,000
dbsoapp402	/dev/linfo/rapp402_1	252000	46	dbixapp405	/dev/linfo/rixapp405_1	393,216	1,572,864	1,600,000
	/dev/linfo/rapp402_2	512000	107	dbapp406	/dev/linfo/rapp406_1	393,216	1,572,864	1,600,000
	/dev/linfo/rapp402_3	180224	142	dbixapp406	/dev/linfo/rixapp406_1	393,216	1,572,864	1,600,000

	/dev/linfo/rapp402_4	256000	202	dbapp407	/dev/linfo/rapp407_1	512000	2,048,000	2,048,000
	/dev/linfo/rapp402_5	335872	177443	dbapp407	/dev/linfo/rapp407_2	128,000	512,000	512,000
Dbisapp302	/dev/linfo/rixapp402_1	500000	3	dbixapp407	/dev/linfo/rixapp407_1	512000	2,048,000	2,048,000
	/dev/linfo/rixapp402_2	512000	0	dbixapp407	/dev/linfo/rixapp407_2	256000	1,024,000	1,024,000
	/dev/linfo/rixapp402_3	262144	20	dbapp408	/dev/linfo/rapp408_1	512000	2,048,000	2,048,000
	/dev/linfo/rixapp402_4	434176	96	dbixapp408	/dev/linfo/rixapp408_1	200000	800,000	800,000
	/dev/linfo/rixapp402_5	256000	108	dbtemp1	/dev/linfo/rtmp1	256000	1,024,000	1,024,000
	/dev/linfo/rixapp402_6	512000	473076	dbtemp2	/dev/linfo/rtmp2	256000	1,024,000	1,024,000
dbapp405	/dev/linfo/rapp405_1	258048	4265	dbreplica	/dev/linfo/rrep	128,000	512,000	512,000
	/dev/linfo/rapp405_2	126976	126973					50,592,000
dbixapp405	/dev/linfo/rixapp405_1	258048	94500					
dbapp406	/dev/linfo/rapp406_1	512000	255030					
dbixapp406	/dev/linfo/rixapp406_1	512000	346182					
dbapp407	/dev/linfo/rapp407_1	512000	106843					
dbixapp407	/dev/linfo/rixapp407_1	204800	0					
	/dev/linfo/rixapp407_2	262144	247147					
dbtemp1	/dev/linfo/rtmp1	256000	255897					
dbtemp3	/dev/linfo/rtmp3	128000	127889					
dbtemp4	/dev/linfo/rtmp4	124000	123881					
		48,034,424.00						

Tabla 5. Análisis de espacio y cálculo de espacio nuevo para el equipo principal.

NOMBRE	RUTA	TAMAÑO	LIBRE		NOMBRE	RUTA	TAMAÑO (PAG)	TAMAÑO (KB)	MULTIPLOS 32 MB
rootdbs	/dev/lifd/rrootd	38,000	31,324		rootdbs	/dev/lifp/rroot	38,000	152,000	160,000
dblogs	/dev/lifd/rlogs	40000	29947		dblogs	/dev/lifd/rlogs	40000	160000	160,000
dblogl	/dev/lifd/rlogl	40000	9947		dblogl	/dev/lifd/rlogl_d	40000	160000	160,000
dbixducto	/dev/lifd/rixduc1	256000	255947		dbalcon	/dev/lifd/rduc1	512000	2048000	2,048,000
dbtemp	/dev/lifd/rtmpd	256000	255947		dbixducto	/dev/lifd/rixduc1	256000	1024000	1,024,000
dbalcon	/dev/lifd/rduc1	512000	511997		db Sara	/dev/lifd/rdb Sara1	12000	48000	64,000
	/dev/lifd/rcon	256000	187559		dbadmocr	/dev/lifd/radmocr	28672	114688	128,000
db Sara	/dev/lifd/rdb Sara1	12000	11947		dbtemp	/dev/lifd/rtmpd	256000	1024000	1,024,000
dbadmocr	/dev/lifd/radmocr	28672	27851						4,768,000
		5,754,688							

Tabla 6. Análisis de espacio y cálculo de espacio nuevo para el equipo secundario.

A continuación realicé el análisis cuantitativo con relación al volumen de los datos en las diferentes localidades, hice una medición del número de registros por tabla y por base de datos que se requerían migrar para determinar la mejor manera de realizar la transferencia de datos. Considerando que Informix cuenta con una herramienta de descarga masiva de información en paralelo HP-Loader (High Performance Loader) que funciona óptimamente con tablas mayores de doscientos mil registros, obtuve el número de registros por tabla y una vez determinado, por cada localidad ordené las tablas de forma ascendente, como se muestra en la tabla 7, a fin de saber por aplicación, qué tablas se iban a descargar con HP-Loader y cuáles se descargarían con una sentencia de sql estándar. Solamente se muestran las tablas de una de las aplicaciones para evitar información repetitiva.

tab la1	-	tab la4 6	1	tab la9 1	67	tab la1 36	58 3	tab la1 81	2,41 2	tabla22 6	13,165	tabla271	148,626	ta bla316	2,079,890
tab la2	-	tab la4 7	1	tab la9 2	68	tab la1 37	60 1	tab la1 82	2,43 4	tabla22 7	13,234	tabla272	150,417	ta bla317	2,081,887
tab la3	-	tab la4 8	1	tab la9 3	69	tab la1 38	61 3	tab la1 83	2,46 1	tabla22 8	14,184	tabla273	158,423	ta bla318	2,121,576
tab la4	-	tab la4 9	1	tab la9 4	70	tab la1 39	65 1	tab la1 84	2,46 2	tabla22 9	14,589	tabla274	159,646	ta bla319	2,122,512
tab la5	-	tab la5 0	1	tab la9 5	76	tab la1 40	68 8	tab la1 85	2,49 4	tabla23 0	14,726	tabla275	162,016	ta bla320	2,215,477
tab la6	-	tab la5 1	1	tab la9 6	78	tab la1 41	72 6	tab la1 86	2,63 2	tabla23 1	14,809	tabla276	170,426	ta bla321	2,348,755
tab la7	-	tab la5 2	2	tab la9 7	82	tab la1 42	74 5	tab la1 87	2,69 6	tabla23 2	15,615	tabla277	170,627	ta bla322	2,711,940
tab la8	-	tab la5 3	2	tab la9 8	85	tab la1 43	74 6	tab la1 88	2,70 0	tabla23 3	16,018	tabla278	182,400	ta bla323	2,777,825
tab la9	-	tab la5 4	3	tab la9 9	86	tab la1 44	80 2	tab la1 89	2,91 4	tabla23 4	16,446	tabla279	195,614	ta bla324	3,152,521
tab la1 0	-	tab la5 5	5	tab la1 00	87	tab la1 45	88 0	tab la1 90	2,92 6	tabla23 5	17,141	tabla280	196,620	ta bla325	3,414,673
tab la1 1	-	tab la5 6	6	tab la1 01	96	tab la1 46	90 6	tab la1 91	2,93 7	tabla23 6	17,539	tabla281	203,536	ta bla326	3,462,408
tab la1 2	-	tab la5 7	7	tab la1 02	103	tab la1 47	91 1	tab la1 92	2,99 9	tabla23 7	18,473	tabla282	206,085	ta bla327	3,881,495
tab la1 3	-	tab la5 8	7	tab la1 03	110	tab la1 48	94 7	tab la1 93	3,03 6	tabla23 8	21,342	tabla283	209,796	ta bla328	4,009,210
tab la1 4	-	tab la5 9	8	tab la1 04	113	tab la1 49	1,00 1	tab la1 94	3,15 8	tabla23 9	22,635	tabla284	215,145	ta bla329	4,441,544
tab la1 5	-	tab la6 0	8	tab la1 05	119	tab la1 50	1,03 3	tab la1 95	3,16 0	tabla24 0	24,147	tabla285	225,330	ta bla330	4,453,555
tab la1 6	-	tab la6 1	8	tab la1 06	121	tab la1 51	1,03 3	tab la1 96	3,43 8	tabla24 1	24,267	tabla286	244,036	ta bla331	4,658,888
tab la1 7	-	tab la6 2	9	tab la1 07	123	tab la1 52	1,03 3	tab la1 97	3,60 9	tabla24 2	24,466	tabla287	246,003	ta bla332	4,867,249
tab la1 8	-	tab la6 3	10	tab la1 08	129	tab la1 53	1,04 0	tab la1 98	3,83 0	tabla24 3	25,284	tabla288	254,004	ta bla333	5,110,886
tab la1 9	-	tab la6 4	12	tab la1 09	129	tab la1 54	1,07 6	tab la1 99	3,89 1	tabla24 4	26,157	tabla289	265,220	ta bla334	5,161,507
tab la2 0	-	tab la6 5	12	tab la1 10	129	tab la1 55	1,08 8	tab la2 00	3,94 2	tabla24 5	27,164	tabla290	286,970	ta bla335	5,881,910
tab la2 1	-	tab la6 6	13	tab la1 11	135	tab la1 56	1,08 9	tab la2 01	4,12 9	tabla24 6	27,841	tabla291	288,718	ta bla336	5,835,339
tab la2 2	-	tab la6 7	16	tab la1 12	136	tab la1 57	1,09 9	tab la2 02	4,36 3	tabla24 7	31,040	tabla292	303,631	ta bla337	6,707,261
tab la2 3	-	tab la6 8	17	tab la1 13	136	tab la1 58	1,09 9	tab la2 03	4,43 2	tabla24 8	31,369	tabla293	419,309	ta bla338	7,125,062
tab la2 4	-	tab la6 9	18	tab la1 14	136	tab la1 59	1,09 9	tab la2 04	4,54 9	tabla24 9	36,814	tabla294	600,505	ta bla339	7,389,966
tab la2 5	-	tab la7 0	19	tab la1 15	147	tab la1 60	1,20 7	tab la2 05	4,85 8	tabla25 0	39,505	tabla295	603,625	ta bla340	7,730,661
tab la2 6	-	tab la7 1	19	tab la1 16	188	tab la1 61	1,23 6	tab la2 06	4,98 5	tabla25 1	41,545	tabla296	623,493	ta bla341	10,276,282
tab la2 7	-	tab la7 2	24	tab la1 17	198	tab la1 62	1,41 2	tab la2 07	5,07 8	tabla25 2	46,381	tabla297	663,186	ta bla342	11,353,603
tab la2 8	-	tab la7 3	26	tab la1 18	207	tab la1 63	1,43 4	tab la2 08	5,09 0	tabla25 3	46,929	tabla298	663,596	ta bla343	11,696,156
tab la2 9	-	tab la7 4	30	tab la1 19	217	tab la1 64	1,52 0	tab la2 09	5,11 3	tabla25 4	48,895	tabla299	776,094	ta bla344	11,750,707
tab la3 0	-	tab la7 5	32	tab la1 20	238	tab la1 65	1,57 1	tab la2 10	5,27 5	tabla25 5	55,599	tabla300	795,024	ta bla345	11,896,123
tab la3 1	-	tab la7 6	32	tab la1 21	252	tab la1 66	1,64 1	tab la2 11	5,34 5	tabla25 6	68,305	tabla301	873,197	ta bla346	12,653,009
tab la3 2	-	tab la7 7	32	tab la1 22	268	tab la1 67	1,67 7	tab la2 12	5,47 3	tabla25 7	69,739	tabla302	904,371	ta bla347	15,148,448
tab la3 3	-	tab la7 8	34	tab la1 23	291	tab la1 68	1,68 7	tab la2 13	6,27 8	tabla25 8	71,527	tabla303	926,087	ta bla348	16,453,878
tab la3 4	-	tab la7 9	34	tab la1 24	296	tab la1 69	1,73 9	tab la2 14	7,28 6	tabla25 9	80,211	tabla304	952,475	ta bla349	20,527,463
tab la3 5	-	tab la8 0	35	tab la1 25	311	tab la1 70	1,89 1	tab la2 15	8,13 4	tabla26 0	81,089	tabla305	965,008	ta bla350	25,133,293
tab la3 6	-	tab la8 1	39	tab la1 26	419	tab la1 71	1,93 5	tab la2 16	9,28 8	tabla26 1	83,880	tabla306	1,014,194	ta bla351	26,294,908
tab la3 7	-	tab la8 2	43	tab la1 27	438	tab la1 72	1,95 7	tab la2 17	10,05 0	tabla26 2	88,505	tabla307	1,027,289	ta bla352	37,360,356
tab la3 8	-	tab la8 3	43	tab la1 28	454	tab la1 73	2,00 3	tab la2 18	10,19 8	tabla26 3	100,468	tabla308	1,028,737	ta bla353	41,024,808
tab la3 9	-	tab la8 4	56	tab la1 29	476	tab la1 74	2,09 9	tab la2 19	10,75 5	tabla26 4	100,468	tabla309	1,103,483	ta bla354	42,654,917
tab la4 0	-	tab la8 5	58	tab la1 30	480	tab la1 75	2,10 9	tab la2 20	10,75 5	tabla26 5	116,701	tabla310	1,185,519	ta bla355	51,569,701
tab la4 1	-	tab la8 6	59	tab la1 31	481	tab la1 76	2,20 0	tab la2 21	10,79 2	tabla26 6	124,261	tabla311	1,192,336	ta bla356	65,572,988
tab la4 2	-	tab la8 7	61	tab la1 32	538	tab la1 77	2,28 0	tab la2 22	11,82 2	tabla26 7	125,395	tabla312	1,326,454	ta bla357	123,139,235
tab la4 3	-	tab la8 8	61	tab la1 33	563	tab la1 78	2,29 0	tab la2 23	11,95 4	tabla26 8	131,349	tabla313	1,331,842	ta bla358	141,049,528
tab la4 4	1	tab la8 9	65	tab la1 34	583	tab la1 79	2,29 6	tab la2 24	12,48 7	tabla26 9	135,003	tabla314	1,435,570		
tab la4 5	1	tab la9 0	66	tab la1 35	583	tab la1 80	2,32 5	tab la2 25	12,98 3	tabla27 0	136,899	tabla315	1,711,659		

Tabla 7. Análisis de número de registros por tabla para determinar forma de descarga: aplicación app1.

Nota: Para proteger la confidencialidad del cliente no se especifican los nombres de aplicación, ni tablas, únicamente es real el número de registros por tabla.

Las tablas mayores de doscientos mil registros fueron descargadas con la herramienta HP-Loader de Informix debido a que el desempeño de la herramienta es muy bueno pero durante las pruebas que se realizaron se utilizaron tablas de varios tamaños, se observó que en la descarga de las tablas menores a 200,000 registros el tiempo de acceso a la tabla a través de HP-Loader superaba el 500% (en tablas muy pequeñas) adicional al tiempo de descarga de registros; por lo tanto las tablas menores a doscientos mil registros se descargaron con una sentencia de descarga común en SQL.

Es importante hacer notar que la descarga se debe hacer de todas las tablas sin excepción aunque no tengan registros, pues pueden ser tablas de procesamiento que en cualquier momento pueden tener registros, aún cuando durante el análisis no los contengan.

Otra consideración importante es que este análisis solamente nos dará la tendencia, hay que recordar que una base de datos transaccional está constantemente cambiando de número de registros dependiendo de las transacciones ejecutadas y sólo puede garantizarse la migración exitosa haciendo la transferencia completa de las tablas.

El análisis anteriormente descrito lo plasmé igualmente para las bases de datos de las otras dos aplicaciones grandes. Las dos bases de datos de las aplicaciones pequeñas no se consideraron en este análisis, debido a que su migración fue realizada a través del comando dbexport que permite realizar un respaldo de la base de datos completa y llevarla a una nueva plataforma. Esta forma de migración es muy segura para bases de datos de pequeñas a medianas, debido a que las bases de datos de las aplicaciones pequeñas miden menos de 100Mb, determiné realizar la migración de esta forma.

El tiempo en el que debía ser migrada la información en las 8 localidades se limitaba a utilizar solamente los fines de semana, por lo que realicé un procedimiento escrito en el cual se especifica paso a paso desde la instalación de los productos Informix, hasta el final de la transferencia de información al equipo nuevo, este procedimiento se presenta en el **anexo A**. Así mismo, preparé dos shell's en uno de ellos se automatiza la instalación de los productos Informix y el otro contiene un grupo de comandos que servirían para realizar la migración de datos de una manera automatizada. Estos shell's se describen con detalle en la siguiente sección.

El procedimiento escrito junto con los shell's, fueron probados en el equipo de laboratorio realizando los ajustes referentes a pantallas presentadas por los productos o localización de rutas de los archivos que contenían los productos, hasta obtener los resultados esperados.

Se enviaron los 8 equipos a su localidad correspondiente con 2 discos duros internos de 82 Gb y 4 ó 6 discos duros externos (según el volumen de información) de 160 Gb y localmente se prepararon cada uno en un arreglo Raid 1+0. Se instaló el sistema operativo AIX 5.3 y se preparó la estructura de sistema de archivos (FS file system, por sus siglas en inglés) necesaria para el cliente, considerando la estructura original y aumentando ligeramente el tamaño de los espacios asignados. Se prepararon las áreas crudas, que son áreas en disco duro sin formato, para almacenamiento de datos de acuerdo con el cálculo realizado para cada una de las localidades. Se preparó un área de 50 GB para trabajar en ella durante la migración en el equipo nuevo, misma que se montó como FS remoto en el equipo antiguo.

Se preparó la configuración de periféricos (impresoras y concentradores) y se realizó la migración de usuarios y aplicaciones de cada uno de los equipos antiguos a los nuevos. Esta migración consistió en la copia del contenido de toda la estructura del sistema de archivos y configuraciones de usuarios del equipo antiguo al nuevo. En esta etapa del proceso, solamente actué como coordinadora para dar soporte y seguimiento al proceso en conjunto con los responsables de las aplicaciones.

Una vez preparados los equipos, se realizó la prueba piloto en una de las localidades de mediana magnitud, en esta ocasión aproveché para realizar un taller demostrativo del funcionamiento en ambiente real del shell de migración. En este taller solicité la presencia del personal técnico de las 8 localidades para realizar su capacitación en el proyecto, hacer la migración completa y posteriormente con mayor confianza cada uno de ellos replicara el procedimiento en su propia localidad.

Durante el taller se presentaron algunos inconvenientes: cuando se realizó la descarga de información estábamos usando la red descargando directamente al FS remoto en el equipo nuevo para no realizar transferencias de archivos por FTP que resultaba mucho más lento. Sin embargo no estaba enterada de una restricción de uso de la red que impide que se sature, de tal manera que cuando el monitoreo automático detecta que un proceso en particular está haciendo uso extremo de ella, ajusta el ancho de banda para que utilice una porción más pequeña y pueda haber disponibilidad para otros procesos. Esto perjudicó grandemente la tarea de descarga, ya que debido a que se realizó en paralelo y a alta velocidad, se hizo un mayor uso del ancho de banda de la red y la restricción de ajuste del ancho de banda provocó que el proceso de descarga demorara más del doble del tiempo estimado; por lo que se buscó la forma de evitarla y se llegó a la solución de usar un cable cruzado y conectar directamente los 2 equipos a través de las tarjetas de red. Esto benefició el proceso, pues disminuyó el tiempo de descarga estimado originalmente en un 30% aproximadamente.

Otro de los problemas que surgieron durante la prueba piloto, fue que algunas de las tablas más grandes no pudieron ser descargadas completamente ya que el

tamaño máximo de un archivo de acuerdo con las restricciones del sistema operativo es de 2GB y algunas de las tablas (de más de 20 millones de registros) excedían este tamaño, la solución fue generar el job de descarga de alta velocidad a 3 ó 5 dispositivos virtuales (en archivos diferentes, según la cantidad de registros de cada tabla) y realizar la carga de manera similar. Este problema no se había presentado durante las pruebas, ya que el ambiente fue establecido con las bases de datos de la localidad más pequeña.

Una vez concluido el proceso de migración en la localidad piloto, se probaron las aplicaciones y se obtuvieron los productos entregables correspondientes, se proporcionó al cliente un juego de productos obtenidos del equipo antiguo, y otro obtenido del equipo nuevo, mismos que fueron revisados y comparados por el cliente a fin de que fuera aprobado el procedimiento. Después de haber obtenido la aprobación del funcionamiento del nuevo equipo, se dio por válida la migración piloto y se acordaron con el cliente las fechas propuestas para realizar la migración del resto de las localidades con relación al plan de trabajo global presentado con anterioridad.

Durante las fechas acordadas con el cliente, acudí a una de las tres localidades programadas cada fin de semana de acuerdo con el plan, para apoyar a los técnicos de menor experiencia a fin de concluir exitosamente la migración de las 8 localidades y el último fin de semana realicé la migración del centro ubicado en la Ciudad de México, del cual soy responsable.

En la sección descrita a continuación se presentan los shell's que se utilizaron en la instalación de productos Informix y el de la migración de información en un orden que permite seguir la secuencia de cada uno de ellos.

Instalación de productos Informix.

Esta primera parte corresponde aún a la preparación de los equipos para la migración de información, por lo que fue la primera etapa. Se generó el file system de 2GB para la instalación de los productos de Informix, que son 4GL run time, SDK (paquete con herramientas de interpretación de programación en SQL y ESQL) e IDS (Informix Dynamic Server); el shell está estructurado en el orden específico de instalación de los productos correspondientes.

Teniendo depositados los productos en una ruta específica se desempacan e instalan los productos seleccionando las opciones correspondientes al paquete de instalación del producto mismo. Se incluyen algunas líneas para informar al usuario qué está haciendo.


```

        tar -xvf /PASO/C988UNA.tar
        sleep 2
        ./ids_install
        sleep 2
        ./RUN_AS_ROOT.server
        sleep 2
        echo "\t\t<<Enter>> para continuar..."
        read opc
        echo "Instalaste manejador Informix" > /PASO/instala
        clear;;

S|s) clear
    exit;;
*) echo "Opcion Incorrecta\c";sleep 2;clear;;
esac
done

```

Programa 1. Instalación de productos Informix: INSTALACION.SH

Como los productos tienen su propio programa de instalación, lo único que se hizo fue un shell que tomara los productos de la ruta donde estaban depositados y se ejecutaran los programas de instalación.

Se migraron también los archivos de configuración y trabajo del usuario administrador de las bases de datos, se resolvió con los dos shell's mostrados a continuación para mover los archivos correspondientes. El primero realiza un respaldo de los archivos en un formato compactado y el segundo desempaca y recupera la información en el equipo nuevo.

```

tar -cvf /PASO/PRINC/arch/monitores.tar /home/dba/MONITORES/*
tar -cvf /PASO/PRINC/arch/exporbd.tar /home/dba/EXPORBD/*

```

Script 1. Copiado de archivos de trabajo del DBA: COPIA_ARCH_PRINC.SH

```

tar -xvf /PASO/PRINC/arch/monitores.tar /home/dba/MONITORES/*
tar -xvf /PASO/PRINC/arch/exporbd.tar /home/dba/EXPORBD/*

```

Script 2. Recuperación de archivos de trabajo del DBA:RECUP_ARCH_PRINC.SH

Después de la instalación de los productos Informix, había que generar la instancia, por lo que fue necesario modificar los archivos de configuración de cada una de ellas: primaria y secundaria, especificando el nombre de la instancia, el rootdbs, que es el dbspace donde se guardan los catálogos de sistema para las bases de datos que se albergarán en esta instancia. Y algunos parámetros importantes como el dbspace donde se generaron los logical logs para la instancia por ejemplo, el tamaño de la memoria compartida para uso del DBMS, el nombre de la instancias, sus alias en caso de contar con ellos. También fue necesario especificar las nuevas direcciones IP de los equipos en el archivo de configuración sqlhosts y los nombres del parámetro DBSERVER para cada instancia.

En cuanto al sistema operativo, se ajustaron algunos parámetros del kernel como el número máximo de buffers que pueden ser manejados en la sesión y el número máximo de procesadores virtuales que pueden generarse en una sesión, así como la lectura y escritura asíncrona, que son los más importantes.

Se iniciaron las instancias y se verificó que el espacio para los catálogos del sistema estuviera en línea y se hubiera generado de manera correcta. El modelo de verificación fue el siguiente:

```
==>onstat -d

IBM Informix Dynamic Server Version 9.40.FC9      -- On-Line -- Up 0 days 00:00:06 -- 801472
Kbytes

Dbspaces
address          number  flags    fchunk  nchunks  flags    owner    name
70000003018d490  1      0x20001  1       1        N        informix rootdbs
1 active, 2047 maximum

Chunks
address          chunk/dbs  offset  size    free    bpages  flags  pathname
70000003018d8f8  1         1       0       49152  7189    PO--  /dev/linfp/rrootp

1 active, 2047 maximum

Expanded chunk capacity mode: disabled
```

Listado 1. Salida del comando “onstat -d”.

Note que en la segunda parte del reporte en la sección de banderas (flags) hay una letra “O”, eso indica que el dbspace se encuentra en línea. Y se verificó la bitácora de inicialización de Informix, con un mensaje semejante al siguiente:

```
==>onstat -m
13:28:59 On-Line Mode
13:29:06 'sysmaster' database built successfully.
13:29:07 'sysutils' database built successfully.
13:29:07 'sysuser' database built successfully.
```

Listado 2. Salida del comando “onstat -m”.

Que significa que la creación de los catálogos del sistema para el DBMS Informix fue exitosa. En caso de no haber sido así, habría necesidad de revisar los mensajes de error para determinar cómo corregir el problema. Las causas más comunes por las cuáles es posible que los catálogos no se generen correctamente son: que el producto esté mal instalado, o que el archivo que construye los catálogos no tenga permisos para ser ejecutado; esto último puede ocurrir cuando el producto haya sido instalado con un usuario privilegiado equivocado.

Para completar las instancias de Informix, fue necesario adicionar los dbspaces que albergan la información; para esto se utiliza el shell de creación de dbspaces para las bases de datos de las aplicaciones. Se utilizó un shell por cada una de las instancias.

```
onspaces -c -d dslogslog -p /dev/linfp/rlogslog -o 0 -s 65536
onspaces -c -d dbslogfis -p /dev/linfp/rlogfis -o 0 -s 327680
onspaces -c -d dbsuno -p /dev/linfp/runo -o 0 -s 1048576
onspaces -c -d dbsixuno -p /dev/linfp/rixuno -o 0 -s 2097148
onspaces -c -d dbsdos -p /dev/linfp/rdos -o 0 -s 1507328
onspaces -c -d dbstres -p /dev/linfp/rtres -o 0 -s 1507328
onspaces -c -d dbscuatro -p /dev/linfp/rcuatro_1 -o 0 -s 2097148
onspaces -a dbscuatro -p /dev/linfp/rcuatro_2 -o 0 -s 1507328
. . .
```

Script 3. Creación de dbspaces en la instancia principal: ESPACIOS_PRINC.SH

Nota: Se muestran solamente unas líneas ya que es un comando repetitivo.

Al finalizar se tendría el siguiente resultado:

```
==>onstat -d

IBM Informix Dynamic Server Version 9.40.FC9      -- On-Line -- Up 0 days 00:00:16 -- 801472
Kbytes

Dbspaces
address          number  flags      fchunk    nchunks  flags      owner      name
70000003018d490 1       0x20001   1         1        N          informix  rootdbs
700000031067028 2       0x1       2         1        N          informix  dbslogslog
700000031067490 3       0x1       3         1        N          informix  dbslogfis
7000000310678f8 4       0x1       4         1        N          informix  dbsuno
70000003106b028 5       0x1       5         1        N          informix  dbsixuno
70000003106b490 6       0x1       6         1        N          informix  dbsdos
70000003106b8f8 7       0x1       7         1        N          informix  dbstres
700000031070028 8       0x1       8         1        N          informix  dbscuatro
8 active, 2047 maximum

Chunks
address          chunk/dbs  offset    size      free      bpages    flags  pathname
70000003018d8f8 1          1         0         49152    7189     PO--   /dev/linfp/rrootp
7000000303c8790 2          2         0         212992   33739    PO--   /dev/linfp/rlogslog
7000000303c8928 3          3         0         81920    31867    PO--   /dev/linfp/rlogfis
7000000303c8ac0 4          4         0         262144   176516   PO--   /dev/linfp/runo
7000000303c8c58 5          5         0         524287   478060   PO--   /dev/linfp/rixuno
7000000303c8df0 6          6         0         376832   286022   PO--   /dev/linfp/rdos
700000031044028 7          7         0         376832   251977   PO--   /dev/linfp/rtres
7000000310441c0 8          8         0         32768    32715    PO--   /dev/linfp/rcuatr_1
700000031044358 9          8         0         524287   524234   PO--   /dev/linfp/rcuatr_2

9 active, 2047 maximum

Expanded chunk capacity mode: disabled
```

Listado 3. Salida del comando "onstat -d".

Note que la sección de banderas no tiene ninguna bandera "D" (down) por lo que se considera exitosa la asignación de espacio para almacenar las bases de datos.

Las instancias de Informix por omisión se generan con 3 logical logs de 200 Mb cada uno, que son alojados en el rootdbs, dbspace principal donde se almacenan las estructuras del diccionario de datos de la instancia; sin embargo con el objeto de no provocar saturación de ese dbspace se sugiere por buenas prácticas colocar los logical logs en un dbspace diferente exclusivo para este fin; con el shell que se muestra a continuación, se eliminan los logical logs de rootdbs y se adicionan los logs que requiere la instancia en un nuevo dbspace.

Es tarea del DBA determinar el número de logs que se deben usar, la recomendación por buenas prácticas es que sea el número de logs del tamaño suficiente para que logren guardar las transacciones generadas durante 24 horas de operación normal. El número de logs se establece respecto al tiempo en que tarda el manejador de bases de datos en hacer un cambio de logs y el número de transacciones que alberga cada log. Si el número de transacciones es alto entonces podrá establecerse un log de un tamaño más pequeño aunque haya más logs en la instancia para no guardar muchas transacciones en un solo log.

De acuerdo con los manuales de referencia de administración del DBMS Informix, no existe una regla exacta para determinar el número y el tamaño de los logs, esto depende del espacio con el que se cuente en disco para guardar logs y del tipo de operaciones que realicen las transacciones de la instancia ya que unas son más largas que otras y requieren más espacio para almacenarse. La forma más adecuada para determinar el número y tamaño de los logs es el monitoreo; el tamaño mínimo de un log es de 200 Kb y el máximo es de 2 Gb. El número mínimo de logs requerido para una instancia es 3 y el número máximo es de 32,767 por lo que es tarea del DBA establecer el número y el tamaño adecuado para que los logs cubran 24 horas de operación de una instancia en particular.

Es importante considerar que mientras más rápido se llenen los logs, más frecuentemente se realizarán los checkpoints (sincronización entre la memoria y el disco) en la base de datos y esto puede desembocar en un problema de desempeño; un log pequeño se llena más rápido que uno grande; lo que guardan los logs son registros, por lo que tienen relación directa con el tamaño del registro que tienen las tablas de bases de datos. También habría que considerar que un log muy grande puede tardar mucho tiempo en ser descargado a disco. Realicé el análisis del monitoreo de llenado de logs y determiné el tamaño y número de logs más conveniente para cada una de las 16 instancias, finalmente preparé los shells para la adición de logs a las instancias, así como la eliminación de los 3 primeros generados por omisión.

```
onparams -a -d dbslogslog -s 28672 -I /* adiciona log */
onparams -a -d dbslogslog -s 28672 -i
onparams -a -d dbslogslog -s 28672 -i
onparams -a -d dbslogslog -s 28672 -i
onparams -a -d dbslogslog -s 28672 -i
onparams -a -d dbslogslog -s 28672 -i
onparams -a -d dbslogslog -s 28672 -i
onparams -a -d dbslogslog -s 28672 -i
onparams -a -d dbslogslog -s 28672 -i
onparams -a -d dbslogslog -s 28672 -i
.
.
onmode -l; onmode -l; onmode -l; onmode -l; onmode -l; onmode -c;
onmode -l; onmode -l; onmode -l; onmode -l; onmode -l; onmode -c;
onmode -l; onmode -l; onmode -l; onmode -l; onmode -l; onmode -c;
/* recorre logs */
.
.
onparams -d -l 1 -y /* elimina logs creados durante la generación de la instancia */
onparams -d -l 2 -y
onparams -d -l 3 -y
```

Script 4. Generación de logs lógicos:CAMBIA_LOGS_P.SH

Con la ejecución del shell anterior se completa la generación de la instancias, hay que recordar que son 2 instancias; la primera albergará las bases de datos del equipo principal y la segunda las bases de datos del equipo secundario.

Migración de información.

En los equipos principales que fueron migrados, hay activo un proceso de replicación de datos, esta réplica se hace desde un equipo concentrador de

información de una aplicación que no se maneja a nivel local sino central. Por lo cual antes de iniciar con la migración de datos se requirió suspender la replicación de datos de manera controlada. Este procedimiento solamente es de diseminación por lo que no hubo más que suspenderlo para evitar que los datos que fueran llegando se insertaran en las tablas durante el procedimiento de migración.

Como los equipos H50 eran bastante antiguos se corría el riesgo de que al ser apagados no encendieran adecuadamente o que alguno de los discos de almacenamiento de datos fallara. Una de las medidas de seguridad fue iniciar con un respaldo de nivel cero del equipo en operación, este respaldo debía ser consistente ya que con él se garantiza el regreso de operación al equipo antiguo en caso de tener algún problema; además de cumplir con una de las buenas prácticas para concluir con un ciclo y guardar el respaldo como histórico.

Un respaldo de nivel cero guarda estructuras de espacios, logs lógicos y físico, estructuras de tablas, información y objetos relacionados tales como índices, vistas, sinónimos, etc. A partir de una recuperación de este respaldo podemos regenerar una instancia, ya sea por problemas de corrupción, o en un equipo nuevo con las mismas características del equipo original.

Otra consideración importante, es la prevención de que algún usuario que no esté enterado del procedimiento intente utilizar las bases de datos, esto no es deseable, así que además de realizar las notificaciones necesarias, se deberá cambiar en los archivos de configuración el nombre de las instancias, para que en el caso de alguna intromisión, el usuario vea la instancia fuera de línea y no pueda acceder a ella.

Estos archivos son: "onconfig" que contiene todos los parámetros de trabajo e identificación de instancias para el DBMS Informix tales como: nombre y alias de la instancia, tamaño y localización del dbspace principal rootdbs, tamaño de la memoria virtual, número de procesadores virtuales, número y tamaño de logs lógicos para guardar transacciones en línea, número de candados para garantizar transacciones completas, número y tamaño de colas de páginas leídas y usadas, tamaño del segmento de memoria que será usado para operaciones de optimización, etc.; y "sqlhosts" que contiene la descripción de los sockets de comunicación con el DBMS Informix, la IP o identificador de nombre del servidor, los puertos de comunicación y configuración de grupos de replicación de datos.

Por practicidad, la migración de información se realizó en 2 partes para cada equipo: descarga y carga de información; a continuación se muestra el shell que se ejecutó para realizar la descarga de información. Como preparación a la descarga se realizó el mantenimiento a las bases de datos, que consiste en una verificación de contenido, índices y encabezados de todas las tablas y una actualización de estadísticas para garantizar que el optimizador funcionara adecuadamente.

Considerando que el proceso de descarga y la transferencia de archivos planos conteniendo la información iba a representar un problema probable de desempeño del equipo antiguo, se consideró realizar la descarga en un FS remoto de 50 GB, localizado en el IBM-P5 de tal manera que la descarga fuera directa al equipo nuevo.

Dos de las cinco bases de datos migradas son pequeñas, tienen pocas tablas y poco volumen de información en ellas, por lo tanto para estas bases se utilizó una herramienta de exportación de información (dbexport), en la opción cero se realiza la exportación de la base de datos pequeña.

Como el resto de las bases contienen muchas tablas y cada tabla mucha información, se realizó extracción del esquema de las bases para llevarlo a las nuevas instancias y la descarga de información posteriormente. Para tener una referencia de verificación de la migración se tomaron los contadores de las tres bases de datos que serán descargadas tabla por tabla, que al final se compararían con los contadores de las tablas ya migradas.

```
#####
#*****
#*****
#*****          MENU PRINCIPAL          *****
#*****
#####

clear
cd /PASO
while true
do
echo
echo "\n\t=====
echo "\t|\t\tSOPORTE Y DESARROLLO          || "
echo "\t| |
echo "\t| |          ACTUALIZACION TECNOLOGICA          | | "
echo "\t| |          MIGRACION DE INFORMACION          | | "
echo "\t=====
echo "\t| |
echo "\t|\t\t0) Respaldo de base appl          | | "
echo "\t| |
echo "\t|\t\t1) Extraccion de estructura de tablas          | | "
echo "\t| |
echo "\t|\t\t2) Descarga de informacion          | | "
echo "\t| |
echo "\t|\t\ts) Salir          | | "
echo "\t| |
echo "\t=====
echo "\n\tTECLEE LA OPCION =====>\c"
read opcl

case $opcl in

0) clear
cd /PASO
echo "\tEste proceso respalda la base de datos appl\n"
echo "\tEspere un momento... \n"
onmode -D 80
dbexport appl -ss -t /PASO/PRINC/APP1 -b 2048 -s 1216000 -f /PASO/PRINC/appl.sql
sleep 2
echo "opcion0" > /PASO/PRINC/OPCION
echo "\n\n\t\t<<Enter>> para continuar..."
read opc
clear;;

1) clear
```

```

cd /PASO
cat /PASO/PRINC/OPCION
echo "Estas ejecutando la opcion 1, verifica secuencia\n"
echo "\n\t<<Enter>> para continuar, CTRL-C para salir..."
read opc
onstat -d > /PASO/PRINC/onstat_d_princ.ori
rm /PASO/PRINC/app2.sql > /dev/null 2>&1
rm /PASO/PRINC/app3.sql > /dev/null 2>&1
sleep 2
echo "\tExtrayendo estructura de tablas, espere un momento..."
dbschema -d app2 -ss > /PASO/PRINC/app2.sql
dbschema -d app2 -f all > /PASO/PRINC/APP2/proc.sql
dbschema -d app2 -s all > /PASO/PRINC/APP2/sinonim.sql
dbschema -d app3 -ss > /PASO/PRINC/app3.sql
dbschema -d app3 -f all > /PASO/PRINC/APP3/proc.sql
dbschema -d app3 -s all > /PASO/PRINC/APP3/sinonim.sql
sleep 2
cp /PASO/PRINC/app2.sql /PASO/PRINC/app2.sql.sql
cp /PASO/PRINC/app3.sql /PASO/PRINC/app3.sql.sql
echo "opcion1" > /PASO/PRINC/OPCION
echo "\n\n\t\t\t\t\t*** I M P O R T A N T E ***\n\n"
echo "\tSacar un contador de las B.D. ====> app2 y app3\n"
echo "\n\tRevisar los archivos de estructura ==> app2.sql y app3.sql\n"
echo "<<ENTER>> para continuar ..."
read opc
clear
clear;;

2) clear
cd /PASO
cat /PASO/PRINC/OPCION
echo "Estas ejecutando la opcion 2, verifica secuencia\n"
echo "\n\t<<Enter>> para continuar, CTRL-C para salir..."
read opc
clear
echo "\tDescargando la informacion de app2 y app3 al file system /PASO/PRINC\n"
echo "\tEspere un momento...\n\n"
sh e_jobs_desc_app2.sh
sleep 5
dbaccess app2 desc_app2.sql
sleep 3
sh e_jobs_desc_app3.sh
sleep 5
dbaccess app3 desc_app3.sql
sleep 3
echo "\t\t<<Enter>> para continuar..."
read opc
echo "opcion2" > /PASO/PRINC/OPCION
clear;;

S|s) clear
exit;;

*) echo "Opcion Incorrecta\c";sleep 2;clear;;
esac
done

```

Programa 2. Descarga de información a migrar: MIGRA_P_DESCARGA.SH

El siguiente shell se usó para generar los jobs de descarga en paralelo de alta velocidad con la herramienta HP-Loader de Informix, para las tablas de más de 200 mil registros. En el archivo se incluyen todas las tablas mayores de 200 mil registros, aquí solo se muestran unas cuantas líneas.

Si analizamos la primera línea, aquí se indica que se desea hacer un job de descarga llamado "d_tabla1" (el prefijo "d_" indica que será un job de descarga) de

la base de datos “app2” que descargará la tabla “tabla1” en el archivo tipo ASCII ubicado en “/PASO/PRINC/APP2/tabla1.asc” y queremos una bitácora de descarga en el archivo “/PASO/PRINC/tmp/d_tabla1.log” para verificar si fue exitosa o no.

```
sh onpp.sh -fu -j d_tabla296 -D app2 -t tabla96 -d /PASO/PRINC/APP2/tabla296.asc -l
/PASO/PRINC/tmp/d_tabla296.log
sh onpp.sh -fu -j d_tabla297 -D app2 -t tabla97 -d /PASO/PRINC/APP2/tabla297.asc -l
/PASO/PRINC/tmp/d_tabla297.log
sh onpp.sh -fu -j d_tabla298 -D app2 -t tabla98 -d /PASO/PRINC/APP2/tabla298.asc -l
/PASO/PRINC/tmp/d_tabla298.log
sh onpp.sh -fu -j d_tabla299 -D app2 -t tabla99 -d /PASO/PRINC/APP2/tabla299.asc -l
/PASO/PRINC/tmp/d_tabla299.log
sh onpp.sh -fu -j d_tabla300 -D app2 -t tabla100 -d /PASO/PRINC/APP2/tabla300.asc -l
/PASO/PRINC/tmp/d_tabla300.log
sh onpp.sh -fu -j d_tabla301 -D app2 -t tabla101 -d /PASO/PRINC/APP2/tabla301.asc -l
/PASO/PRINC/tmp/d_tabla301.log
. . .
sh onpp.sh -fu -j d_tabla366v -D app2 -t tabla366 -f /PASO/tabla366_lis -l
/PASO/PRINC/tmp/d_tabla366v.log
. . .
sh onpp.sh -fu -j d_tabla374v -D app2 -t tabla374 -f /PASO/tabla374_lis -l
/PASO/PRINC/tmp/d_tabla374v.log
```

Script 5. Creación de jobs de descarga de alta velocidad: C_JOBS_DESC_APP2.SH

La solución del problema que se presentó referente a la descarga de más de 20 millones de registros a un solo archivo está en las últimas líneas del shell anterior, se especifica que se requiere generar un job de descarga para la tabla “tabla366” de la base de datos “app2” descargando al grupo de archivos descrito en el archivo /PASO/tabla366_lis y queremos una bitácora de descarga en el archivo /PASO/PRINC/tmp/d_tabla366v.log

La descripción del archivo que contiene las rutas a diferentes archivos de depósitos para datos se muestra a continuación.

```
/PASO/PRINC/APP2/tabla366_1.asc
/PASO/PRINC/APP2/tabla366_2.asc
/PASO/PRINC/APP2/tabla366_3.asc
```

Archivo1. Archivo de descripción de depósitos para datos de la tabla 366: TABLA366_LIS

```
/PASO/PRINC/APP2/tabla374_1.asc
/PASO/PRINC/APP2/tabla374_2.asc
/PASO/PRINC/APP2/tabla374_3.asc
/PASO/PRINC/APP2/tabla374_4.asc
/PASO/PRINC/APP2/tabla374_5.asc
```

Archivo 2. Archivo de descripción de depósitos para datos de la tabla 374: TABLA374_LIS

Continuando con la secuencia, el shell que se lista en seguida, es el de ejecución de los jobs generados para la descarga de las tablas, se hace referencia a la herramienta de descarga con el comando “onpload” y los jobs generados anteriormente.

```
onpload -j d_tabla296 -fu
onpload -j d_tabla297 -fu
onpload -j d_tabla298 -fu
```

```

onpload -j d_tabla299 -fu
onpload -j d_tabla300 -fu
onpload -j d_tabla301 -fu
. . .
onpload -j d_tabla366v -fu
. . .
onpload -j d_tabla374v -fu

```

Script 6. Ejecución de jobs de descarga de datos: E_JOBS_DESC_APP2.SH

El resto de las tablas se descargaron con sentencias de descarga de información en SQL Language como se muestra inmediatamente.

```

unload to /PASO/PRINC/APP2/tabla1.asc select * from tabla1;
unload to /PASO/PRINC/APP2/tabla2.asc select * from tabla2;
unload to /PASO/PRINC/APP2/tabla3.asc select * from tabla3;
unload to /PASO/PRINC/APP2/tabla4.asc select * from tabla4;
unload to /PASO/PRINC/APP2/tabla5.asc select * from tabla5;
unload to /PASO/PRINC/APP2/tabla6.asc select * from tabla6;
. . .
unload to /PASO/PRINC/APP2/tabla95.asc select * from tabla295;

```

Script 7. Descarga de tablas con menos de doscientos mil registros: DESC_HDA.SQL

Para garantizar la descarga completa de los registros de las bases de datos, se revisan las bitácoras de descarga paralela de alta velocidad y se cuentan los registros de los archivos descargados con las sentencias SQL. Las bitácoras lucen como se muestra a continuación.

```

Fri Aug 22 13:28:03 2008

SHMBASE          0x30000000
CLIENTNUM        0x49020000
Session ID 368

Unload Database -> APP2
Query Name       -> d_tabla315
Device Array     -> d_tabla315
Query Mapping    -> d_tabla315
13:28:05 Records Processed -> 10000
13:28:15 Records Processed -> 20000
13:28:22 Records Processed -> 30000
13:28:35 Records Processed -> 40000
13:28:41 Records Processed -> 50000
. . .
13:45:40 Records Processed -> 820000
13:45:57 Records Processed -> 830000
13:46:11 Records Processed -> 840000
13:46:31 Records Processed -> 850000
13:46:50 Records Processed -> 860000
13:47:05 Records Processed -> 870000
Database Unload Completed -- Unloaded 873197 Records Detected 0 Errors
Fri Aug 22 13:47:18 2008

```

Archivo 3. Bitácora de descarga de información de alta velocidad.

El encabezado de la bitácora indica la fecha y hora en que se inició la descarga, la memoria que está usando, el identificador cliente y el número de sesión, posteriormente indica el job que se está ejecutando e inicia con la descarga. Al final muestra el resumen de la descarga, lo importante que debe revisarse es el


```

onstat -d > /PASO/PRINC/onstat_d_princ.fin
sleep 2
echo "\tExtrayendo estructura de tablas, espere un momento..."
separa_app2.sh
separa_app3.sh
dbaccess sysmaster /PASO/PRINC/APP2/tablas.sql
dbaccess sysmaster /PASO/PRINC/APP3/tablas.sql
echo "opcion1" > /PASO/PRINC/OPCION
echo "<<ENTER>> para continuar ..."
read opc
clear;;

2) clear
cd /PASO
cat /PASO/PRINC/OPCION
echo "Estas ejecutando la opcion 2, verifica secuencia\n"
echo "\n\t<<Enter>> para continuar, CTRL-C para salir..."
read opc
echo "\tCargando la informacion de las bases app2 y app3\n"
echo "\tEspere un momento...\n\n"
sh junta.sh
sh c_jobs_carga_app2.sh
sh c_jobs_carga_app3.sh
dbaccess app2 car_app2.sql
sleep 5
sh e_jobs_carga_APP2.sh
sleep 5
dbaccess app3 car_app3.sql
sleep 5
sh e_jobs_carga_app3.sh
sleep 5
echo "\t\t<<Enter>> para continuar..."
read opc
echo "opcion2" > /PASO/PRINC/OPCION
clear;;

3) clear
cd /PASO
cat /PASO/PRINC/OPCION
echo "Estas ejecutando la opcion 3, verifica secuencia\n"
echo "\n\t<<Enter>> para continuar, CTRL-C para salir..."
read opc
echo "\tCreando indices y asignando permisos ..."
echo "\tEspere un momento...\n\n"
sleep 5
dbaccess app2 /PASO/PRINC/APP2/indices.sql
sleep 5
dbaccess app2 /PASO/PRINC/APP2/proc.sql
sleep 5
dbaccess app2 /PASO/PRINC/APP2/sinonim.sql
sleep 5
dbaccess app3 /PASO/PRINC/APP3/indices.sql
sleep 5
dbaccess app3 /PASO/PRINC/APP3/proc.sql
sleep 5
dbaccess app3 /PASO/PRINC/APP3/sinonim.sql
sleep 5
echo "\t\t<<Enter>> para continuar..."
read opc
echo "opcion2" > /PASO/PRINC/OPCION
clear;;

S|s) clear
exit;;

*) echo "Opcion Incorrecta\c";sleep 2;clear;;
esac
done

```

Programa 3. Carga de información a migrar en el nuevo equipo: MIGRA_P_CARGA.SH

Debido a que la carga masiva de información es más eficiente deshabilitando los índices y constraints de las tablas ya que con esto se evita que se validn las restricciones implícitas, se realizó una extracción de estructuras en archivos separados para generar scripts con sentencias sql y efectuar la creación del esquema de la base de manera separada. De tal forma que al reconstruir el esquema en el nuevo equipo, se generen las tablas sin índices ni constraints, se cargue la información, posteriormente se generen los índices y finalmente se apliquen los constraints y permisos. El resultado funcional de estas actividades es el mismo que si se copia la estructura original completa, sin embargo representa mayor facilidad de manejo en caso de errores y ahorro de tiempo de carga de información.

El siguiente shell realiza la separación de las tablas, índices, constraints y demás objetos de la base de datos, para la base de datos app2. Como en este caso se está trabajando en una instancia que aún no está en producción se asigna a los procesos la máxima prioridad para ejecución de query's, esto no afectó el desempeño y nos permitió el uso de todos los recursos disponibles durante la carga de datos. Este shell toma como entrada el esquema de la base de datos obtenido en la primera parte del proceso, se ejecuta el mismo shell para separar los esquemas de las tres bases de datos de aplicaciones grandes: app2, app3 del equipo primario y app4 del equipo secundario.

Se dejan las sentencias separadas en el archivo "tablas.sql" que contiene sólo la declaración de la estructura de las tablas y en el archivo "indices.sql" conteniendo todas las sentencias de creación de objetos y restricciones adicionales a la base de datos en cuestión en directorios separados para cada base de datos. Se muestra solamente uno de los shell's utilizados.

```
## Shell para la extraccion de indices a partir de un dbschema

clear
cd /PASO/PRINC
rm /PASO/PRINC/APP2/tablas.sql > /dev/null 2>&1
rm /PASO/PRINC/APP2/indices.sql > /dev/null 2>&1
touch /PASO/PRINC/APP2/tablas.sql
touch /PASO/PRINC/APP2/indices.sql
echo "create database APP2;" > /PASO/PRINC/APP2/tablas.sql
echo "set PDQPRIORITY 100;" > /PASO/PRINC/APP2/indices.sql

clear

echo "Extraccion de tablas, espere un momento ..."
sleep 2
cat /PASO/PRINC/app2.sql.sql | awk ' BEGIN { bandera=0; termina=0 }
{
if ( $1 == "create" && ( $2 == "table" ) )

{ bandera = 1
}
if ( bandera == 1 && substr($0,length($0),1) == ";" )
{ termina = 1
printf("%s\n", $0 )
}
if ( bandera == 1 && termina == 0 ) { printf("%s\n ", $0) }
if ( bandera==1 && termina == 1) { bandera=0 ; termina=0 }
```

```

}
END {
} ' >> /PASO/PRINC/APP2/tablas.sql

*****
## Genera la extraccion de indices.

echo
echo "Extraccion de indices, espere un momento ..."
sleep 2
cat /PASO/PRINC/app2.sql.sql | awk ' BEGIN { bandera=0; termina=0 }
{
if ( $1 == "create" && ( $2 == "index" || $3 == "index" ) )
{ bandera = 1
}
if ( bandera == 1 && substr($0,length($0),1) == ";" )
{ termina = 1
printf("%s\n", $0 )
}
if ( bandera == 1 && termina == 0 ) { printf("%s ", $0) }
if ( bandera=1 && termina == 1) { bandera=0 ; termina=0 }
}
END {
} ' >> /PASO/PRINC/APP2/indices.sql

*****
## Genera los alter table.

echo
echo "Extraccion de alter table, espere un momento ..."
sleep 2
cat /PASO/PRINC/app2.sql.sql | awk ' BEGIN { bandera=0; termina=0 }
{
if ( $1 == "alter" && ( $2 == "table" ) )
{ bandera = 1
}
if ( bandera == 1 && substr($0,length($0),1) == ";" )
{ termina = 1
printf("%s\n", $0 )
}
if ( bandera == 1 && termina == 0 ) { printf("%s\n ", $0) }
if ( bandera=1 && termina == 1) { bandera=0 ; termina=0 }
}
END {
} ' >> /PASO/PRINC/APP2/indices.sql

*****
## Genera las vistas de la base de datos.

echo

echo "Extraccion de vistas, espere un momento ..."
sleep 2
cat /PASO/PRINC/app2.sql.sql | awk ' BEGIN { bandera=0; termina=0 }
{
if ( ( $1 == "create" || $1 == "CREATE" ) && ( $2 == "view" || $2 == "trigger" ) )
{ bandera = 1
}
if ( bandera == 1 && substr($0,length($0),1) == ";" )
{ termina = 1
printf("%s\n", $0 )
}
if ( bandera == 1 && termina == 0 ) { printf("%s\n ", $0) }
if ( bandera=1 && termina == 1) { bandera=0 ; termina=0 }
}
END {
} ' >> /PASO/PRINC/APP2/indices.sql

```



```

echo "\nExtraccion de permisos, espere un momento ..."
sleep 2
cat /PASO/PRINC/app2.sql.sql | awk ' BEGIN {   bandera=0; termina=0   }
{
if ( $1 == "grant" )
{ bandera = 1
}
if ( bandera == 1 && substr($0,length($0),1) == ";" )
{ termina = 1
printf("%s\n", $0 )
}
if ( bandera == 1 && termina == 0 ) { printf("%s\n ", $0) }
if ( bandera=1 && termina == 1) { bandera=0 ; termina=0 }
}
END {
} ' >> /PASO/PRINC/APP2/indices.sql

sleep 2
echo "\n\n\nPROCESO FINALIZADO !!!!!!! \n "
echo "<<ENTER>> para continuar ... "
read opc

```

Script 8. Separación de la estructura de tablas, índices y otros objetos: SEPARA_APP2.SH

Con la versión de Informix 9.40 que fue utilizada, se tiene una versión mejorada de HP-Loader, se deben ingresar las sentencias de creación de jobs para esta nueva versión, de tal manera que éstos sean generados y puedan usarse posteriormente para la carga. A diferencia de la versión anterior de Informix, 7.31, ahora en la versión 9.40 la herramienta está integrada en el producto, por lo que se hace referencia a ella a través del comando “onpladm” y sus diferentes opciones para indicar la clase de operación que se desea realizar.

A continuación se analiza la primera línea del shell, indica lo siguiente: crea un job con el prefijo “c_” que nos indicará que es un job de carga, que tome la información del archivo depositado en /PASO/PRINC/APP2/tabla96.asc para poblar la base de datos “app2”, la tabla a cargar será “tabla96” y el archivo está delimitado.

```

onpladm create job c_tabla296 -d /PASO/PRINC/APP2/tabla96.asc -D APP2 -t tabla296 -fl -z D
onpladm create job c_tabla297 -d /PASO/PRINC/APP2/tabla97.asc -D APP2 -t tabla297 -fl -z D
onpladm create job c_tabla298 -d /PASO/PRINC/APP2/tabla98.asc -D APP2 -t tabla298 -fl -z D
onpladm create job c_tabla299 -d /PASO/PRINC/APP2/tabla99.asc -D APP2 -t tabla299 -fl -z D
onpladm create job c_tabla300 -d /PASO/PRINC/APP2/tabla100.asc -D APP2 -t tabla300 -fl -z D
onpladm create job c_tabla301 -d /PASO/PRINC/APP2/tabla101.asc -D APP2 -t tabla301 -fl -z D
. . .
onpladm create job c_tabla366v -d /PASO/PRINC/APP2/tabla366_lis -D APP2 -t tabla366 -fa -z D
. . .
onpladm create job c_tabla374v -d /PASO/PRINC/APP2/tabla374_lis -D APP2 -t tabla374 -fa -z D

```

Script 9. Creación de jobs de carga de información de alta velocidad: C_JOBS_CARGA_APP2.SH

El siguiente shell se encarga de ejecutar los jobs generados anteriormente para la carga, analizando la primera línea se puede ver que se especifica lo siguiente: ejecuta el job c_tabla96, de carga y lleva la bitácora de ejecución en el archivo /PASO/PRINC/tmp/c_tabla96.log

```

onpladm run job c_tabla296 -fl -l /PASO/PRINC/tmp/c_tabla296.log
onpladm run job c_tabla297 -fl -l /PASO/PRINC/tmp/c_tabla297.log
onpladm run job c_tabla298 -fl -l /PASO/PRINC/tmp/c_tabla298.log

```

```

onpladm run job c_tabla299 -fl -l /PASO/PRINC/tmp/c_tabla299.log
onpladm run job c_tabla300 -fl -l /PASO/PRINC/tmp/c_tabla300.log
onpladm run job c_tabla301 -fl -l /PASO/PRINC/tmp/c_tabla301.log
. . .
onpladm run job c_tabla366v -fl -l /PASO/PRINC/tmp/c_tabla366v.log
. . .
onpladm run job c_tabla374v -fl -l /PASO/PRINC/tmp/c_tabla374v.log

```

Script 10. Ejecución de jobs de carga de información de alta velocidad: E_JOBS_CARGA_APP2.SH

Y finalmente se realizó la carga de información de las tablas más pequeñas con sentencias comunes de carga de SQL. Aquí se muestran algunas de ellas.

```

load from /PASO/PRINC/APP2/tabla1.asc insert into tabla1;
load from /PASO/PRINC/APP2/tabla2.asc insert into tabla2;
load from /PASO/PRINC/APP2/tabla3.asc insert into tabla3;
load from /PASO/PRINC/APP2/tabla4.asc insert into tabla4;
load from /PASO/PRINC/APP2/tabla5.asc insert into tabla5;
load from /PASO/PRINC/APP2/tabla6.asc insert into tabla6;
. . .
load from /PASO/PRINC/APP2/tabla95.asc insert into tabla295;

```

Script 11. Carga de información de tablas menores a 200,000 registros: CAR_APP2 .SQL

Para concluir el proceso de migración de equipos, se realizó una verificación de las bitácoras de carga en paralelo de alta velocidad de tablas y los contadores finales de todas las tablas respecto a los iniciales.

```

Sat Aug 23 01:28:03 2008

SHMBASE          0x30000000
CLIENTNUM        0x49020000
Session ID 368

Unload Database -> APP2
Query Name       -> c_tabla301
Device Array     -> c_tabla301
Query Mapping    -> c_tabla301
01:28:05 Records Processed -> 10000
01:28:15 Records Processed -> 20000
01:28:22 Records Processed -> 30000
01:28:35 Records Processed -> 40000
01:28:41 Records Processed -> 50000
. . .
01:45:40 Records Processed -> 190000
01:45:57 Records Processed -> 200000
01:46:11 Records Processed -> 210000
01:46:31 Records Processed -> 220000
01:46:50 Records Processed -> 230000
01:47:05 Records Processed -> 240000
Database Load Completed -- Loaded 244036 Records Detected 0 Errors
Sat Aug 23 01:47:18 2008

```

Archivo 4. Bitácora de carga de alta velocidad

Se verificaron las bitácoras de la carga de la misma manera que las de descarga. En el caso de la carga de información con HP-Loader, si existen errores, hay que borrar toda la información cargada en la tabla y realizar nuevamente la carga de esa tabla en particular. Se debe recordar que como no están generados aún los índices y constraints, de no eliminar toda la información, estaríamos cargando información duplicada.

Finalmente para concluir la verificación de carga, se toman contadores de las tablas completas, éstos se comparan con los contadores obtenidos al inicio del proceso (en el equipo antiguo). Deben ser iguales.

Después de esta verificación, se ejecuta la opción 3 para generar el resto de los objetos de las bases de datos y aplicar todas las restricciones del esquema original, esta operación llevó aproximadamente entre 3 y 5 horas, dependiendo del tamaño de la base de datos.

Una vez terminado el procedimiento, se cambió el nombre de las instancias a su nombre original, se dieron de baja las instancias y se cambió la dirección IP de los equipos a los originales, para que todas las referencias externas sean consistentes. Se volvieron a dar de alta las instancias para continuar el proceso.

Para concluir el procedimiento, se debe restablecer la replicación de datos desde el equipo central, por lo que se notificó al administrador del equipo para que se redefiniera la réplica hacia los nuevos equipos. Esto implica un cambio en la configuración debido al cambio de versión del software base, por lo que se requiere eliminar la definición original y realizar de cero la definición de servidores de réplica y réplicas de tablas. Una vez obtenida la notificación de conclusión de esta actividad se realizaron pruebas de replicación de datos y se verificaron los resultados; posteriormente se hicieron los respaldos de nivel cero de bases de datos y el total del sistema del nuevo equipo.

En ese momento, se entregó el equipo a los responsables de la operación de aplicaciones para que realizaran las pruebas de operación pertinentes y aprobar la migración. Durante las pruebas estuve presente para apoyar en los problemas que se presentaron: impedimento para leer algunas tablas, impedimento para escribir en algunas rutas de los equipos nuevos, otorgué los permisos correspondientes a cada usuario y se solucionó el problema.

Apoyé en la adecuación de las configuraciones para conexiones vía Web y se realizaron las pruebas de conectividad y consulta a bases de datos para garantizar el uso de la consulta Web que tiene el cliente implementada a algunas de las bases de datos.

Resultados y aportaciones.

El éxito en la migración de los equipos de nuestro cliente, tuvo los siguientes beneficios:

- Se tiene una base instalada de servidores IBM Series P5 de última generación.
- Tienen capacidad de crecimiento en cómputo.

- El sistema operativo está actualizado y hay ventajas operativas, las más importantes son la garantía de seguridad en la información.
- Hay posibilidades de crecimiento en el almacenamiento.
- Hay menor cantidad de licencias utilizadas y por administrar.
- Se redujo el espacio utilizado por los servidores y el consumo de energía eléctrica.
- Hay posibilidad de implementar monitoreo completo con Tívoli, de manera adicional a la distribución de software.
- Se realizó la homologación de versiones, en los 3 niveles de servidores de cómputo (local, regional y nacional) para Informix y Tuxedo.

Conclusiones.

Con este procedimiento se cubrió el objetivo de beneficiar al cliente, contando con equipos actualizados, tolerantes a fallas y que tienen refacciones disponibles en caso de fallas técnicas. Adicionalmente me permitió realizar investigaciones sobre las nuevas características de software DBMS de Informix y las ventajas de la versión 9.40 respecto a la 7.31 que estaba en uso. Considerando así mismo las ventajas que ofrece el soporte y asesoría del servicio de soporte técnico durante las 24 horas del día del proveedor de Informix.

Aprendí muchos detalles finos de la versión 9.40 del DBMS Informix, referentes al uso de la herramienta de carga paralela de alta velocidad, nuevos requerimientos para memoria compartida y la posibilidad de utilizar chunks de más de 2Gb para la generación de dbspaces en Informix para guardar BLOB's (Binary Large Objects, por sus siglas en inglés). Las características de creación de índices con la metodología B-tree y sus características para localización y obtención de datos de manera más eficiente.

Este proyecto también sirvió para capacitar al personal técnico en actividades de administración de bases de datos y en la realización de migraciones de información.

Así mismo me permitió diseñar un procedimiento automatizado para instalación de productos Informix y migración de información, que puede ser utilizado para posteriores actualizaciones tecnológicas.

Considero muy importante haber participado en este proyecto, ya que me permitió complementar y afianzar mis conocimientos sobre características físicas de equipos de cómputo y el sistema operativo AIX.

Y también considero relevante el desarrollo de este proyecto, porque me brindó la oportunidad de destacar mis habilidades en planeación de proyectos de ingeniería de cómputo, demostrar liderazgo durante el manejo y coordinación de personal

técnico, lo cual me dio una posición privilegiada en la organización después de la evaluación de los resultados de mi trabajo por los directivos de la empresa.

Bibliografía.

- ✓ Informix Dynamic Server Administrator's Guide Ver. 9.4
- ✓ Informix Dynamic Server Administrator's Reference Ver. 9.4
- ✓ Informix Dynamic Server Enterprise Replication Guide Ver. 9.4
- ✓ IBM Informix Migration Guide
- ✓ Manuales en línea de AIX 5L V5.3
- ✓ Hoja técnica: 625 - JLOGIN - A program to test JDBC connectivity to the server.
- ✓ Documentation Notes for the IBM Informix Dynamic Server Administrator's Guide
- ✓ IDS Machine notes 9.40
- ✓ IDS migrate docnotes 9.40 (en línea).

ANEXO A.

Procedimiento para la instalación de Informix y la migración de información a equipos nuevos.

Introducción.

Debido a que los equipos actuales en producción, son obsoletos y se tienen continuos problemas con discos y servicio de mantenimiento en general, se ha propuesto al cliente y aceptado por el usuario cambiarlos por equipos más recientes de IBM serie P. Esto nos permitirá mantener los niveles de servicio sin las limitaciones propias de la antigüedad de los equipos. Es por eso que deberá realizarse una migración de aplicaciones e información a los nuevos equipos.

Alcance.

Este procedimiento va dirigido al personal técnico regional, quienes participarán activamente en la realización de este proyecto. Contiene las consideraciones necesarias para la realización del proyecto y la forma y momento oportuno de usarlos.

Consideraciones.

1. Se tiene tanto en el equipo principal como en el secundario, un usuario: **migra07** con igual password que nos permitirá realizar estas actividades sin necesidad de depender de comandos autorizados del sistema de apoyo al operador. **Este usuario es un clon de informix por lo tanto es necesario tener suma precaución con él hay que recordar que aún son nuestros equipos de producción.**
2. Estaremos usando comandos de línea para realizar algunas actividades necesarias para la verificación del procedimiento y en casos de contingencia.
3. En el equipo nuevo el usuario para trabajar será: **migra07** con igual password.
4. Verificar que en el F.S. /PASO se encuentren los siguientes directorios:
 - a. /PASO/PRINC/APP2
 - b. /PASO/PRINC/APP3
 - c. /PASO/PRINC/tmp
 - d. /PASO/PRINC/arch
 - e. /PASO/SEC/APP4
 - f. /PASO/SEC/arch
5. Verificar que en el F.S. /PASO/PRINC se encuentre el archivo **vacío** APP1
6. Verificar que en el F.S. /PASO/SEC se encuentre el archivo **vacío** APP4
En el caso de que no se encuentre alguno de ellos hay que generarlos.
7. Copiar del equipo de producción todos los archivos en la ruta: /home/MIGRACION a la ruta /PASO.
8. Montar DVD con los productos en el equipo P5 y copiar los archivos a la ruta: /PASO

Actividades previas.

Verificar en el plan de trabajo que hayan sido realizadas las actividades previas que dependen de cada CRCD (tarea 3, 6 y 8 de los planes de trabajo de migración PRIMARIO y SECUNDARIO); estas actividades tienen que llevarse a cabo con la anticipación suficiente para que el día de la migración estén completadas.

Los equipos nuevos llegarán precargados con el sistema operativo, sin embargo será necesario instalar Informix con sus herramientas en el orden que se describe a continuación.

Se tiene un DVD con los productos de Informix para las herramientas y el manejador; se deben bajar los productos al F.S. /PASO en el equipo P5 (el nuevo).

En primer lugar habrá que firmarse con el usuario root y después se ejecutará el script: instalacion.sh que desplegará la siguiente pantalla:

```

=====
|               SOPORTE Y DESARROLLO               |
|               ACTUALIZACION TECNOLOGICA          |
|             INSTALACION DE INFORMIX             |
|-----|-----|
|            1) Instalacion de 4GL run time        |
|            2) Instalacion de herramientas SDK    |
|            3) Instalacion de manejador Informix  |
|            s) Salir                              |
|-----|-----|
=====

```

TECLEE LA OPCION =====>

Con la primera opción se cambiará el directorio al F.S. de Informix y se exportará la variable INFORMIXDIR y se desempacará la herramienta 4GL, el archivo es el C833MNA.tar; una vez que termine el desempacado de la herramienta, se ejecutará el archivo install4glrt para que la herramienta quede instalada.

A continuación con la opción número dos, se desempacará la herramienta SDK para instalarla, con el comando "tar"; al terminar el desempacado se instalará la herramienta ejecutando el archivo installclientsdk, se tiene un menú de instalación en el que hay que aceptar todas las opciones por omisión.

Y finalmente con la opción número tres, se desempacará e instalará el manejador de bases de datos, presentará otro menú para instalación, en la primera pregunta hay que contestar lo siguiente:

- 0) All products listed below
- 1) IBM Informix Dynamic Server 9.40
- 2) IBM Informix IConnect
- 3) IBM Informix JDBC
- 4) IBM Informix Server Administrator
- 5) Configure a Demo IDS Server (requires IDS)

Enter the number(s) of the products to install, separated by spaces
(i.e. "1 2 3"): 1 2 3

Y aceptar el resto de las preguntas por omisión, se desplegarán algunos mensajes y por último se ejecutará el archivo RUN_AS_ROOT.server para completar la instalación y saldremos del script.

Posteriormente se copiarán los archivos necesarios para generar las instancias en el equipo nuevo ejecutando desde el equipo H50 el script:

```
/PASO/copia_arch_princ.sh
/PASO/copia_arch_sec.sh
```

Y recuperarlos en el equipo nuevo, ejecutando desde el equipo P5 los siguientes scripts, es importante ejecutar primero el correspondiente a los archivos del equipo secundario y al final el correspondiente al equipo principal:

```
/PASO/recupera_arch_sec.sh
/PASO/recupera_arch_princ.sh
```

El siguiente paso será editar los archivos onconfig.xxx y onconfig.xxx2 donde hay que cambiar los siguientes parámetros:

```
ROOTSIZE          196608
PHYSDBS           rootdbs
LOGFILES          3
LOGSIZE           2000
DBSPACETEMP <espacios temporales> (ver arch. de espacios)
```

Revisar el archivo sqlhosts y verificar que existan las líneas:

```
xxx_lan_tcp       onsoctcp      IP_MIGRACION sqlexec_1525
xxx_lan           onipcshm     IP_MIGRACION sqlexec_shm1
```

y

```
xxx2_lan          onipcshm     IP_MIGRACION sqlexec_shm2
xxx2_lan_tcp     onsoctcp     IP_MIGRACION sqlexec_1529
```

donde xxx es el nombre de la localidad ; e IP_MIGRACION sea la IP actual del equipo nuevo. También escribir la IP del equipo nuevo en todos los lugares donde se haga referencia a xxx_lan o xxx2_lan. Si las líneas no existen habrá que incluirlas.

Así mismo es necesario verificar que las áreas crudas para los dbspaces estén listas, con la liga en /dev/linfp para la instancia principal y /dev/linfd para la instancia de secundaria; los permisos de las áreas deberán ser 660 y propietario y grupo "informix". Hay que verificar que existan todos los espacios de acuerdo con el archivo de dbspaces nuevos: espacios_princ.sh y espacios_sec.sh

Una vez verificado todo esto se procederá a inicializar las instancias, hay que estar seguros de encontrarse firmado en el equipo nuevo, para esto se sugiere cerrar todas las sesiones abiertas y abrir nuevamente una sola sesión en el equipo nuevo P5, ya que una vez inicializada la instancia la única forma de recuperar la información es realizando la recuperación del respaldo.

Después de firmarse con el usuario migra07, se deberá copiar el perfil de usuario al directorio casa del usuario de la siguiente manera:

```
cp /PASO/.profile2 .
mv .profile .profile.ori
mv .profile2 .profile
```


Y ejecutar el nuevo perfil para elegir primero la instancia de producción.

```
./profile
Indique Instancia 1:principal 2:secundaria
1
```

A continuación habrá que ejecutar los siguientes comandos:

```
onstat -
shared memory not initialized for INFORMIXSERVER 'xxx_lan'
oninit -iv
```

donde iv, significa initialize verbose.

Aparecerán muchos mensajes de inicialización, al final de cada línea deberá aparecer la palabra “suceded”, hasta llegar a lo siguiente:

```
Initialize Async Log Flusher...succeeded
Forking btree cleaner...succeeded
Initializing DBSPACETEMP list
Checking database partition index...succeeded
Checking location of physical log...succeeded
Initializing dataskip structure...succeeded
Checking for temporary tables to drop
Forking onmode_mon thread...succeeded
Verbose output complete: mode = 5
```

Al llegar a la línea Initializing DBSPACETEMP list se detendrá un poco el proceso, es normal porque el manejador está buscando los espacios temporales declarados en esta variable. Cuando se despliegue a la última línea del párrafo anterior, estaremos en condiciones de continuar construyendo nuestra instancia, hay que verificar que las siguientes líneas aparezcan en la bitácora de informix:

```
onstat -m
13:28:59 On-Line Mode
13:29:06 'sysmaster' database built successfully.
13:29:07 'sysutils' database built successfully.
13:29:07 'sysuser' database built successfully.
```

Continuando con la construcción de nuestra instancia, deberemos adicionar los dbspaces necesarios a la instancia para el almacenamiento de datos ejecutando el comando:

```
/PASO/espacios_princ.sh (espacios_sec.sh)
```

Se adicionarán los espacios y se desplegarán mensajes del tipo:

```
Verifying physical disk space, please wait ...
Space successfully added.
```

```
** WARNING ** A level 0 archive of Root DBSpace will need to be done.
```

Para finalizar en la construcción de la instancia, hay que cambiar de lugar los logical logs y el physical log, ejecutando el script:

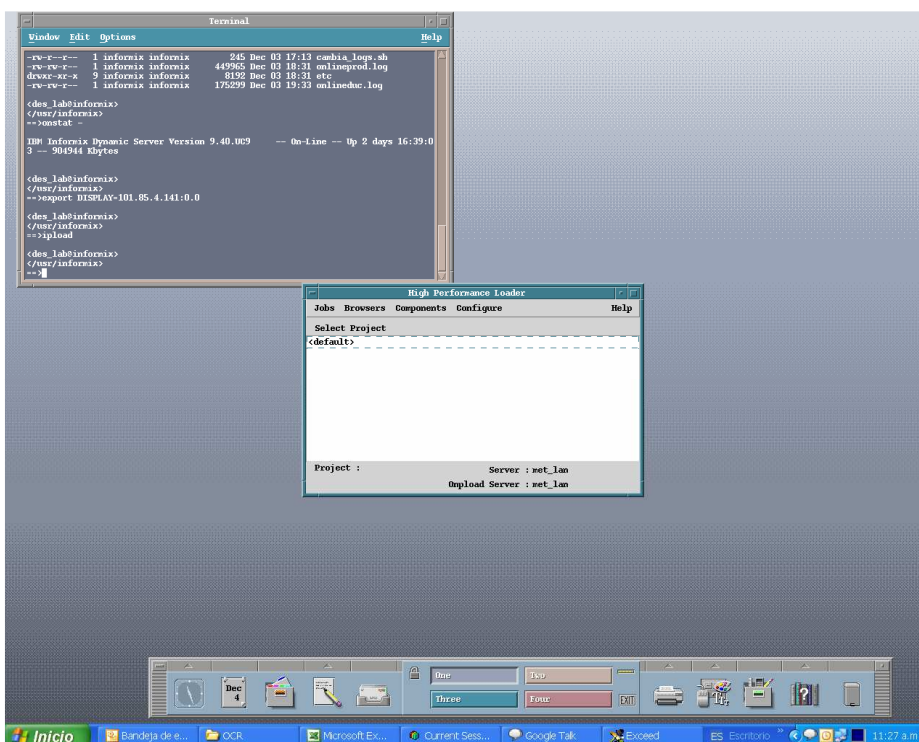
```
cambia_logs_p.sh (cambia_logs_s.sh)
```

Revisar que los logs lógicos sean el mismo número que en la instancia original H50 (ver archivo de espacios).

Para concluir con la instancia de producción, hay que conectarse al equipo nuevo a través de exceed y ejecutar los comandos:

```
export DISPLAY=<IP del equipo>:0.0  
ipload
```

Este último comando generará algunos mensajes (warning) y finalmente presentará una ventana pequeña blanca en medio de la pantalla, que es del high performance loader, la herramienta para carga y descarga de tablas en paralelo.



Cuando aparezca esta pantalla, se cerrará seleccionando del menú “Jobs” la opción “Exit”. Con esto hemos creado la base de datos onpload que nos servirá para realizar las cargas en paralelo en el nuevo equipo.

La instancia secundaria, se prepara de la misma manera, los scripts cambian, son los que se encuentran entre paréntesis.

Verificar notificación para detener operación y notificación a BD de equipos centrales para realizar la eliminación de la replicación de datos y respuesta de confirmación.

Descripción del procedimiento de migración.

Primera parte.

Para realizar esta actividad se requiere ingresar al equipo H50 principal/secundario con el usuario migra07 con igual password, con el objeto de aislar la instancia de los equipos y evitar que algún usuario pueda ingresar a ella, se dará de baja el manejador, se editarán las siguientes líneas del archivo de configuración de Informix:

```
Cd /usr/informix/etc
Vi onconfig.xxx
SERVERNAME      migra      #xxx_lan_tcp
DBSERVERNAME    #xxx_lan,yyy_lan...
```

Y en el archivo sqlhosts se copiará la línea:

```
xxx_lan      onipcshm      111.222.333.444      sqlexec_shm1
```

Y se modificará para que quede lo siguiente:

```
migra      onipcshm      111.222.333.444      sqlexec_shm1
#xxx_lan   onipcshm      111.222.333.444      sqlexec_shm1
```

En el equipo H50 estará montado un F.S. llamado /PASO que será remoto del equipo nuevo. Hay que cambiarse a ese directorio, ahí se encontrarán los archivos copiados en la etapa anterior que nos ayudarán con la migración de datos.

Los archivos que vamos a ejecutar para este propósito se llaman migra_p_descarga.sh (descarga de información de la instancia principal), migra_p_carga.sh (carga de información de la instancia principal), migra_s_descarga.sh (descarga de información de la instancia secundaria) y migra_d_carga.sh (carga de información de la instancia secundaria); Las instancias estarán inicializadas en el equipo nuevo.

Los nombres de las instancias no cambian una vez que los equipos se cambien, vamos a tener los mismos nombres y estructuras, excepto con respecto a los dbspaces que en el caso de las tablas de la aplicación app3 (editar app3.sql.sql) de la base de datos de app3, cambia a "dbsapp3" sin el número del ejercicio fiscal y los índices a "dbsixapp3". Así mismo para las tablas de 2007; por lo que también se tendrá que editar el archivo app2.sql.sql; donde se encontrarán las tablas correspondientes al año 2007; el dbspace donde deben residir en el equipo nuevo sera: dbsapp3 y los índices en dbsixapp3.

Ejecución de migra_p_descarga.sh

```
=====
|                                     |
|             SOPORTE Y DESARROLLO    |
|                                     |
|   ACTUALIZACION TECNOLOGICA        |
|   MIGRACION DE INFORMACION        |
|=====|
|                                     |
|   0) Respaldo de bases app1        |
|   1) Extraccion de estructura de tablas |
|   2) Descarga de informacion        |
|   s) Salir                          |
|                                     |
|=====|
```

TECLEE LA OPCION =====>

La opción cero, tiene una duración de entre 5 y 10 minutos, el respaldo se realiza con una instrucción dbexport y se deposita en el FS remoto para este propósito, en la ruta /PASO/PRINC/APP1, los archivos adicionales generados por el respaldo son: dbexport.out para referencias y app1.sql para la estructura de la base de datos.

Los mensajes desplegados en pantalla son:

```
Este proceso respalda la base de datos appl
Espere un momento...
```

Posteriormente se despliegan varios mensajes que indican que se está realizando el respaldo. Al finalizar se desplegará el mensaje :

```
*** end of dbexport ***
```

Para tratar de evitar errores por cansancio, las siguientes opciones tienen un mensaje de advertencia para seguir la secuencia.

La opción uno tardará de 5 a 10 minutos en realizarse, desplegará el siguiente mensaje antes de ejecutarse:

```
opcion0
Estas ejecutando la opcion 1, verifica secuencia
<<Enter>> para continuar, CTRL-C para salir ...
```

Esta opción obtiene una imagen de los dbspaces, y los esquemas de las bases de datos app2 y app3. Y después de oprimir <enter>, se despliega lo siguiente:

```
Extrayendo estructura de tablas, espere un momento...

***      I M P O R T A N T E      ***

Sacar un contador de las B.D. ====>  app2 y app3
<<Enter>> para continuar...
```

Es suficiente con depositar el contador a un archivo, pero si se prefiere soporte duro no hay ningún problema en obtenerlo en papel. Del archivo de contadores se puede insertar en el archivo "verifica contador.xls" como el contador original.

La opción número dos es la más larga, el tiempo estimado para ejecutarse es entre 7 y 10 horas dependiendo del volumen de información del CRCD del que se trate. Esta opción también tiene el mensaje para verificación de secuencia.

```
Opcion1
Estas ejecutando la opcion 2, verifica secuencia
<<Enter>> para continuar, CTRL-C para salir ...
```

Esta opción trabaja con una sentencia "unload" de SQL para las tablas pequeñas y jobs de descarga en paralelo para las tablas más grandes (más de 200,000 registros). Las tablas de la base de datos "app2", se descargarán en la ruta /PASO/PRINC/APP2 en archivos ASCII con los nombres de cada una de las tablas de la base de datos. Primero se descargarán las tablas más grandes de la base y después las tablas pequeñas. Posteriormente en la ruta /PASO/PRINC/APP3 se descargarán las tablas más grandes de la base de datos "APP3" y finalmente las más pequeñas.

Durante la ejecución de la opción 2 que tiene una duración más larga, se realizarán algunas actividades necesarias para la segunda parte del procedimiento.

Una actividad que puede llevarse a cabo durante la descarga de información, es la verificación de las bitácoras de descarga de las tablas más grandes. Estas se depositarán en la ruta /PASO/PRINC/tmp cada bitácora tendrá el siguiente formato:

```
d_<nombre_de_la_tabla>.log
```

el prefijo “d_” se refiere a que esa es la bitácora de la descarga. Se puede ver el archivo completo o solamente recortar la parte que nos interesa con el siguiente comando:

```
==>head d_tabla4.log ; tail d_tabla4.log
Fri Oct 26 13:28:03 2007

SHMBASE          0x30000000
CLIENTNUM        0x49020000
Session ID 368

Unload Database -> APP2
Query Name       -> d_tabla4
Device Array     -> d_tabla4
Query Mapping    -> d_tabla4
13:45:40 Records Processed -> 780000
13:45:57 Records Processed -> 790000
13:46:11 Records Processed -> 800000
13:46:31 Records Processed -> 810000
13:46:50 Records Processed -> 820000
13:47:05 Records Processed -> 830000

Database Unload Completed -- Unloaded 834658 Records Detected 0
Errors
Fri Oct 26 13:47:18 2007
```

La primera parte del comando nos muestra el nombre de la tabla y la segunda parte, muestra el número de registros descargados y la cantidad de errores detectados, aquí se va a validar el número del contador y los errores. Si hay alguna bitácora con errores, hay que tomar nota de la tabla de que se trate para realizar la descarga nuevamente en paralelo o bien con una sentencia “unload”. Es conveniente tener a la mano los contadores obtenidos en la opción uno para esta verificación.

Para las tablas menores de doscientos mil registros, la descarga deberá ser verificada contando el número de registros descargados con el comando:

```
$ wc -l > <nombre_archivo>
```

Y verificados con el contador que se obtuvo en la opción número uno, con el apoyo de la hoja de cálculo “verifica contador.xls”

Cuando termine la descarga de información y se termine la verificación de la misma, trabajaremos en el equipo nuevo.

Descripción del procedimiento.

Segunda parte.

Para realizar esta parte del procedimiento se requiere ejecutar otro script: `migra_p_carga.sh` que presentará la siguiente pantalla:

```
=====
|                 SOPORTE Y DESARROLLO                 |
| ACTUALIZACION TECNOLOGICA                             |
| MIGRACION DE INFORMACION                             |
|=====
|
|          0) Recuperacion de base app1                 |
|          1) Generacion de estructura de tablas        |
|          2) Carga de informacion                     |
|          3) Generacion de indices y constraints      |
|          s) Salir                                     |
|=====
TECLEE LA OPCION =====>
```

Esta parte del procedimiento es para recuperar la información respaldada en la sección anterior, es básicamente el mismo procedimiento, la opción cero recupera la base de datos `app1` con la instrucción `dbimport` los mensajes que se desplegarán en pantalla son semejantes a la primera parte del procedimiento:

```
Este proceso recupera la base de datos app1
Espere un momento...

. . .

*** end of dbimport ***
dbimport completed
```

Esta opción tardará en ejecutarse entre 5 y 10 minutos como máximo.

En la opción uno, se separa el esquema de la base de datos, en tablas y el resto de los objetos (índices, procedimientos almacenados, vistas, etc.) ya que las tablas sin índices ni otros objetos, son mejores para la carga de información. Esta opción puede durar algunas horas. Una vez separados los objetos, se ejecuta la creación de tablas tanto para la base de datos `app2` como para la base de datos `app3`. En este momento tendremos solamente los siguientes mensajes desplegados en pantalla:

```
Extrayendo estructura de tablas, espere un momento...
Database selected.
Table created.
```

```
Table created
```

```
. . .
```

```
Database closed.
```

```
<<ENTER>> para continuar ...
```

Igual que en la opción de descarga de información de la primera parte del procedimiento, esta opción número dos puede tardar en ejecutarse varias horas en promedio unas 10 horas. De la misma manera que durante la descarga, se generarán bitácoras de carga de tablas en paralelo en la ruta:

```
/PASO/PRINC/tmp
```

con el formato:

```
c_<nombre_tabla>.log
```

Que se verificarán de la misma forma que las de descarga, más la cuenta de registros de las tablas menores a doscientos mil registros. Esto se puede realizar de manera rápida con el comando:

```
./imprime <nombre_base>
```

Y usando nuevamente la hoja de cálculo "verifica contadores.xls; en el caso de que haya alguna diferencia habrá que repetir la carga borrando la información de la tabla y haciendo ya sea una carga en paralelo o no, dependiendo del volumen de información a cargar.

```
Cargando la informacion de las bases app2 y app3  
Espere un momento...
```

```
Database selected.
```

```
676 records loaded.
```

```
43897 records loaded.
```

```
. . .
```

```
Database closed.
```

```
<<Enter>> para continuar...
```

Después de haber verificado que los registros cargados son efectivamente los que se tenían en la base con referencia al contador obtenido en la opción uno de la primera parte del procedimiento, se deberá ejecutar la opción número tres, que creará el resto de los objetos alrededor de la base de datos (índices, sinónimos, vistas, procedimientos almacenados, etc.); los mensajes que se desplegarán en pantalla serán:

```
Creando indices y asignando permisos ...  
Espere un momento...
```

```
Database selected.
```

```
Index created.
```

```
Index created.  
  
. . .  
  
Database closed.  
<<Enter>> para continuar...
```

Esta operación también se ejecutará en varias horas, al menos 3. Mientras esto ocurre, en otra sesión se puede ingresar a la instancia secundaria e iniciar el procedimiento, que será el mismo, solo que ahora los scripts que se van a utilizar son: **migra_s_descarga.sh** y **migra_s_carga.sh**

El procedimiento para el equipo secundario, es un poco menos tardado debido a que solamente se estarán migrando dos bases de datos de dimensiones menores que las del equipo principal.

Actividades posteriores a la migración de información.

Una parte muy importante para adecuar los datos para su utilización, es realizar una actualización de estadísticas de los datos migrados, esto genera los histogramas de las distribuciones para columnas en las tablas y permite al optimizador de Informix tomar las mejores rutas para el acceso a datos. Esta actividad se realizará con el script:

```
/PASO/actualiza_princ.sh o /PASO/actualiza_sec.sh
```

Según el caso, se utilizará el script correspondiente. Hay que estar bien seguros de la instancia donde se está aplicando cada script.

Finalmente, se realizará el respaldo de nivel cero correspondiente de cada instancia editando el archivo onconfig.xxx y onconfig.xxx2 como sigue y ejecutando el comando:

```
$ vi onconfig.xxx (onconfig.xxx2)  
TAPEDEV /PASO/PRINC/nivel0  
TAPEDEV /PASO/SEC/nivel0  
  
$ ontape -s (en ambas instancias)
```

Posteriormente se realizará un respaldo a dispositivos de cinta, estos dispositivos hay que guardarlos y reciclarlos como normalmente se realiza en cada localidad.

Para concluir habrá que dar de baja la instancia en los equipos H50 y regresar los archivos onconfig y sqlhosts a su estado original, finalmente dar de alta las instancias.