



UNIVERSIDAD NACIONAL AUTONOMA  
DE MEXICO

110  
2EJ

FACULTAD DE INGENIERIA

**REINGENIERIA DEL SISTEMA DE  
INFORMACION DE LA FACULTAD DE  
INGENIERIA PARA SU IMPLEMENTACION EN  
UN AMBIENTE CLUSTER.**

FALLA DE ORIGEN

**T E S I S**

QUE PARA OBTENER EL TITULO DE:  
INGENIERO EN COMPUTACION  
P R E S E N T A  
GERARDO VILLALPANDO ROMO



ASESOR: ING. DOMINGO PALAO MUÑOZ

MEXICO, D. F.

1995



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



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

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

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

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

---

**Reingeniería del Sistema de Información de la  
Facultad de Ingeniería para su implementación en un  
ambiente Cluster.**

---

---

## **OBJETIVO**

**Plantear y aplicar las ventajas tecnológicas de los equipos de cómputo, configuraciones y software con que cuenta la Facultad de Ingeniería para la implementación de las aplicaciones y la Base de Datos en un ambiente VAX-Cluster.**

---

---

**Agradecimientos.**

Al personal de USECAFI por las facilidades proporcionadas.

Al apoyo de mi padre aunque ahora no esté ya con nosotros.

A mi madre y hermanos por su aliento y entusiasmo.

A mis amigos por su insistencia.

---

---

## INDICE

1 .	Introducción	1 - 1
2 .	Estructura actual en la Facultad de Ingeniería	2 - 1
	Políticas	2 - 1
	Recursos de Hardware	2 - 2
	Recursos de Software	2 - 7
	Recursos Humanos	2 - 9
3 .	Arquitecturas y Procesos	3 - 1
	Uniprosesador	3 - 7
	Simétricamente Paralelo	3 - 8
	Cluster	3 - 10
	Paralelismo Masivo	3 - 11
4 .	Característica Tecnológicas del RDBMS Oracle7	4 - 1
	Estructuras de memoria y procesos	4 - 8
	Estructura de la Base de Datos	4 - 19
	Integridad de los datos	4 - 25
	Seguridad de la Base de Datos	4 - 27
	Consideraciones adicionales	4 - 29

---

---

5 .	Procedimiento de migración	5 - 1
	Migración de la Base de Datos	5 - 2
	Copia de información a través de SQL*Net	5 - 6
	Utilería de Migración	5 - 8
	Exportar / Importar	5 - 27
	Migración de las aplicaciones	5 - 32
6 .	Arquitectura y configuración de Oracle en Cluster	6 - 1
	Parallel Cache Management	6 - 4
	Archivos de LOG en una Base de Datos en modo paralelo	6 - 14
	Segmentos de Rollback en una Base de Datos en modo paralelo	6 - 17
	Como dar de alta y baja la Base de Datos en modo paralelo	6 - 20
7 .	Arquitectura y configuración del VAX - Cluster	7 - 1
	Antecedentes	7 - 2
	Características	7 - 3
	Estructura del Hardware de un VAX - Cluster	7 - 5
	Software del VAX - Cluster	7 - 10
	Configuración e implementación del Cluster	7 - 12
8 .	Caso Práctico	8 - 1
	Primera parte : Proceso de Desarrollo de la Base de Datos	8 - 1
	Segunda parte : Proceso de Desarrollo del Sistema de Exámenes Extraordinarios	8 - 16
9 .	Conclusiones	9 - 1
10 .	Bibliografía	

---

## **INTRODUCCION**

Cuando se habla de reingeniería me parece necesario acotar los términos y alcances para un esquema como el que en este trabajo se está planteando, ya que aquí se manifiesta el aprovechamiento y revisión del esquema actual para el manejo de la información académico - administrativa de la Facultad de Ingeniería para optimar la operación con dicha información.

Lo ideal es llegar a una verdadera reingeniería del negocio, lo que significa "empezar de nuevo". Este concepto que no es nuevo pero está funcionando y con gran interés en nuestra cultura no se trata de remendar nada, de componer los sistemas existentes para que funcionen mejor simplemente. Lo que significa es abandonar procedimientos establecidos hace mucho tiempo y examinar otra vez el trabajo que se requiere para crear lo que se nos pide, ya sea un sistema de información o un producto en una compañía, que para este caso es la Facultad de Ingeniería, y entregarle algo de valor a los funcionarios de la misma.

No se está empezando de cero con este trabajo debido a que ya ha habido un análisis de procesos y flujo de información que han determinado un hardware y software a emplear, pero sí se plantea en este trabajo la integración de los nuevos sistemas, optimación y reglas básicas para una buena funcionalidad del manejo de información.

Para apoyar todo el proceso de reingeniería se están realizando y planteando nuevos esquemas de trabajo organizacional inclusive en la misma Unidad de Servicios de Cómputo para la Administración de la Facultad de Ingeniería (USECAFI), porque la reingeniería significa olvidarse de cómo se realizaba el trabajo en la época del procesamiento masivo a través de esquemas en batch o por lotes, es decir con la objetividad de cómo se puede realizar mejor ahora. En la reingeniería de negocios los viejos títulos y formas organizacionales ( departamentos, divisiones, grupos, etc. ) dejan de tener importancia. Lo que importa en la reingeniería es cómo queremos hoy organizar el trabajo, dadas las exigencias de las necesidades y el potencial de las tecnologías actuales, puntos que son tratados en los capítulos 2 y 3; planteando las arquitecturas de cómputo recientes y la manera de organizar los trabajos en la USECAFI.

Trabajos que mantienen las características de sus integrantes como es la innovación, individualismo, confianza en sí mismo, voluntad de correr riesgos y actitud de frente al cambio; siendo de esta manera coherentes con los planteamientos de la reingeniería de negocios que no pretende modificar el comportamiento de los trabajadores o de los líderes. Por el contrario, aprovecha sus disposiciones naturales y da todos los incentivos y motivación al ingenio creativo de cada ser que participa en todo proceso de superación y servicio.

Una vez sentadas las bases y el ambiente en el que se ha de continuar este trabajo, en el capítulo 4 se explican y ejemplifican las principales características tecnológicas del manejador de Bases de Datos Oracle versión 7, que se empleará para el cumplimiento del objetivo principal, que se resume en plantear e implementar las ventajas tecnológicas que se puedan aplicar de acuerdo a la estructura computacional con que cuenta la Facultad de Ingeniería.

Los capítulos 5 y 6 contienen los procedimientos y recomendaciones para realizar la configuración del manejador de la Base de Datos y de esta forma las aplicaciones pueden aprovechar lo que se ha explicado en los capítulos anteriores.

De la misma forma que se debe configurar el software del manejador de la Base de Datos, se hace para los equipos de cómputo que participan en este esquema del Cluster; este procedimiento es explicado en el capítulo 7.

Trabajando de manera activa con los responsables del desarrollo e implementación de los sistemas de información, se ha llegado a obtener un modelo de entidades y relaciones que refleja el esquema actual de datos e información que se emplea en la Facultad de Ingeniería. Este diagrama de entidades y relaciones es expuesto en el capítulo 8 de una manera clara. Durante este proceso se descubrieron algunas tareas y sistemas que se encontraban aislados y otros que se ejecutaban simplemente para satisfacer exigencias internas de la propia organización y que no son necesarios ya que generan inconsistencia y falta de integridad en la información; estos aspectos son comentados en cada capítulo en el caso que amerite alguna aclaración particular, pero de manera general en el capítulo 9 de conclusiones y sugerencias.

---

En general se plantean una serie de procedimientos y reglas para continuar, dirigir y llevar a un término el proceso de reingeniería global de la Facultad de Ingeniería, iniciando con este trabajo que se enfoca al Sistema de Información que sustenta los procedimientos y procesos planteados hasta este momento por los dirigentes de esta entidad educativa de la UNAM.

## **ESTRUCTURA ACTUAL EN LA FACULTAD DE INGENIERIA**

Es importante considerar el medio ambiente donde se ha desarrollado este proyecto ya que existen una serie de políticas y recursos con los que se cuenta tanto de hardware y software a los que nos hemos apegado para el mejor funcionamiento y aprovechamiento de la configuración actual.

En este capítulo se mencionan los recursos con los que cuenta actualmente la Facultad de Ingeniería para automatizar el proceso académico - administrativo; así como los organismos y personal que interviene para efectuar tales tareas.

### **Políticas:**

La Unidad de Servicios de Cómputo para la Administración (USECAFI) es el órgano de la Facultad de Ingeniería en la UNAM, dependiente de la Secretaría de Servicios Académicos, encargada de administrar el esquema de información académica de la Facultad.

USECAFI es creado al reestructurarse el Centro de Cálculo (CECAFI) con el fin de optimar los servicios que se ofrecen a la comunidad de la Facultad de Ingeniería, con el objetivo fundamental de dar servicio con calidad.

La organización interna de USECAFI está basada en la administración por proyectos, la cual tiene una jefatura que se encarga de coordinar el trabajo, varios administradores de proyectos quienes conceptualizan y planifican las soluciones, y un grupo de becarios que desarrollan los sistemas de solución que logran la vinculación de la Unidad con las necesidades de la Facultad. Además se cuenta con apoyo de secretarías, capturistas y operadores del equipo de cómputo.

---

La implementación de soluciones se lleva a cabo utilizando la tecnología más actual en el mercado tanto en equipos de cómputo como en software.

USECAFI trabaja de manera muy cercana con distintas instancias de la Facultad tales como el Departamento de Administración Escolar, Las Secretarías Académicas y algunos otros departamentos de las Divisiones.

La información es el recurso más importante de una organización, sin importar su tipo. La Unidad de Servicios de Cómputo para la Administración es el órgano encargado de mantener y administrar la información académica de la Facultad de Ingeniería y es el lugar en el que se ha propuesto e implementado el ambiente VAX-Cluster conjuntamente con el RDBMS Oracle versión 7; aspectos que son tratados particularmente en capítulos subsecuentes con la finalidad de documentar y apoyar esta decisión.

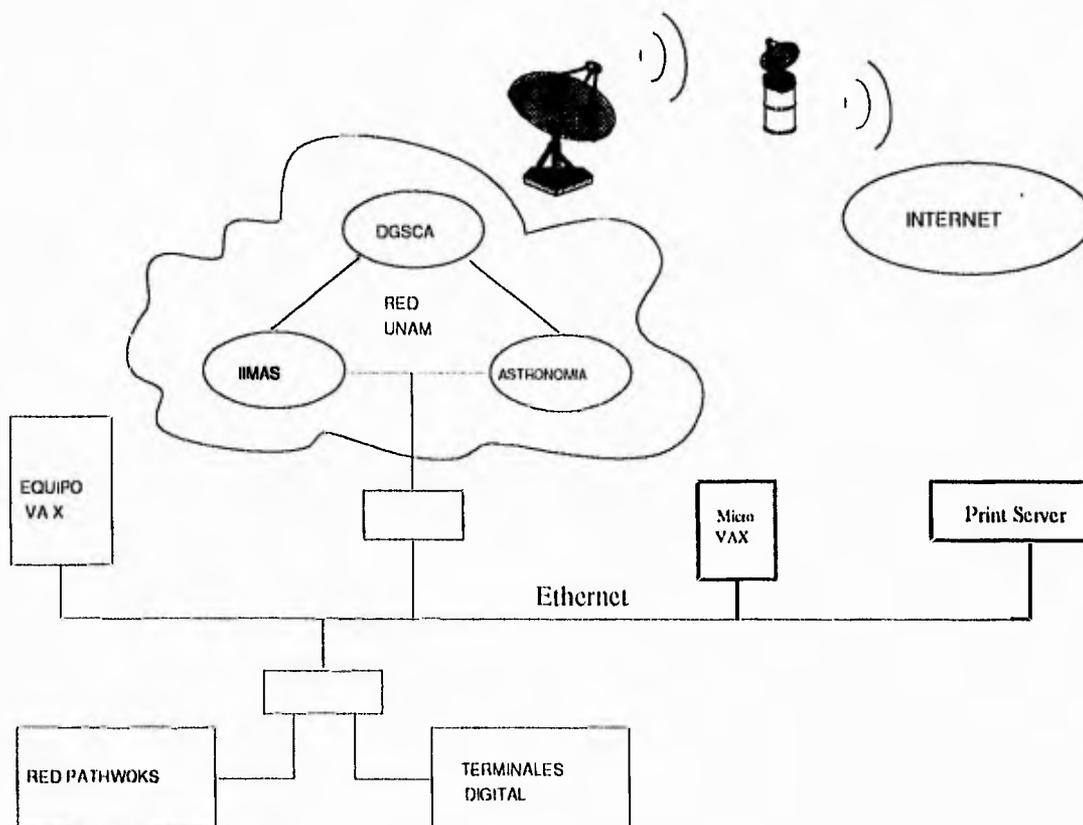
### **Recursos de Hardware**

A principios de 1990 cuando todavía existía el Centro de Cálculo en la Facultad de Ingeniería se instaló una pequeña red local para unir los equipos de cómputo principales. Hasta hace poco tiempo el uso de dicha red se había limitado prácticamente a la transferencia de archivos entre los sistemas internos del Centro de Cálculo. Sin embargo al definirse la nueva estructura de trabajo en la Facultad y el surgimiento de USECAFI, se han aprovechado tres sucesos que han incrementado notablemente las aplicaciones de esta red :

- La integración de USECAFI a la RedUNAM.
- La conexión de computadoras personales a la red de USECAFI.
- La implementación del Cluster con los equipos existentes.

La red de USECAFI se basa en la tecnología Ethernet, que es una de las más populares en la actualidad. La topología Ethernet es una derivación de la topología de bus lineal. Con la implementación de la red en USECAFI se tienen varios buses conectados mediante dispositivos llamados repetidores, que se encargan de duplicar la señal en cada uno de los canales de comunicación (buses).

La topología de la red en USECAFI se muestra en la siguiente figura :



La red de USECAFI se encuentra conectada a la RedUNAM mediante un repetidor de fibra óptica. La RedUNAM tiene sus nodos principales en la Dirección General de Servicios de Cómputo Académico, el Instituto de Matemáticas Aplicadas y Sistemas; y en el Instituto de Astronomía. Además de proporcionar la capacidad de comunicación entre cualquiera de sus nodos, la RedUNAM presta los servicios de acceso a los ambientes de supercómputo y de conexión con la Internet.

De manera general se muestra el esquema actual y a continuación se explicará a detalle la configuración de cada equipo, tanto VAX como computadoras personales. Este equipo se utiliza con diversos fines y cuenta con puntos de acceso en varios lugares de la Facultad, como son : USECAFI; la División de Ciencias Básicas; la División de Ingeniería Mecánica e Industrial; la División de Ingeniería Eléctrica, Electrónica y en Computación; la División de Ingeniería Civil; la División de Ingeniería en Ciencias de la Tierra; la División de Ciencias Sociales y Humanidades; el Departamento de Administración Escolar y la Oficina de Servicios Escolares.

#### **Configuración de los nodos en USECAFI:**

El nodo principal de la red en USECAFI se llama ATZIN y es una minicomputadora VAX6000/210 construida por Digital Equipment Corporation. El hardware de esta computadora incluye :

- 32 Mbytes de memoria principal.
  
- Dos discos duros de 1.2 Gbytes.
  
- Dos discos duros de 1.5 Gbytes.

- Dos discos duros de 300 Mbytes.
- Una unidad de cartucho TK70.
- Una unidad de cartucho TF86 de 6Gbytes por cartucho.
- Una unidad de cinta TU81.
- Un graficador LG02 de 600 líneas por minuto.
- Una impresora LP37 de 1200 líneas por minuto.
- Una impresora láser LN03.

Otro sistema conectado a la red es una minicomputadora MicroVAX 3400 que se llama **KELEM**, construida también por Digital Equipment Corporation. Esta computadora tiene las siguientes características :

- 12 Mbytes de memoria principal.
  - Dos discos duros de 400 Mbytes.
  - Un disco duro de 1 Gbytes.
  - Una unidad de cartucho TK70.
  - Una unidad de cinta TS05.
  - Una impresora LPA0 de 600 líneas por minuto.
-

El resto de los nodos de la red son computadoras personales con diferentes configuraciones, que pueden ser :

- Computadoras 286, con 1 Mbyte de RAM, 40 Mbytes en disco duro y monitor VGA.
  
- Computadoras 286, con 6 Mbytes de RAM, 80 Mbytes en disco duro y monitor Super VGA.
  
- Computadoras 386, con 8 Mbytes de RAM, 80 Mbytes en disco duro y monitor super VGA.

La red cuenta con cuatro servidores de terminales que permiten la conexión con los nodos ATZIN y KELEM desde terminales de tipo carácter; y adicionalmente se tiene un "Print Server 20 Turbo" de 20 paginas por minuto.

Es importante notar que, a pesar de que cada máquina tiene sus propios recursos, el software de la red permite usar los recursos de hardware y software de un nodo desde cualquier otro. Por mencionar un ejemplo, una computadora personal puede usar como almacenamiento secundario el disco de ATZIN o utilizar la impresora láser de esa máquina.

El hecho de contar ya con esta red nos ha permitido proponer e implementar el VAX Cluster entre las máquinas VAX 6000/210 y la MicroVAX 3400, lo que se traduce en distribución de proceso y aprovechamiento de recursos de cómputo al máximo en USECAFI, ya que no solo se comparten los discos que conforman el Cluster, sino que también las unidades de almacenamiento secundario y las colas de proceso batch y de impresión se están utilizando de forma común.

### Recursos de Software

Los nodos de la red en USECAFI cuentan con un conjunto significativo de utilerías que pueden ser empleas por cualquiera de sus usuarios. En esta sección se hace una descripción de este software.

El Sistema Operativo OpenVMS (Virtual Memory System), fue desarrollado por Digital Equipment Corporation para su serie de computadoras VAX. OpenVMS ha llegado a ser un Sistema Operativo muy común en diversos ambientes, tales como universidades, empresas gubernamentales y de la iniciativa privada. La Unidad de Servicios de Cómputo para la Administración (USECAFI), ha contado con este tipo de sistemas desde 1982, año en el que se adquirió una computadora VAX 11/780; y actualmente cuenta una computadora VAX 6000/210 y una microVAX 3400 con la versión 5.5 de Open VMS.

Este cluster tendrá ahora usos diversos, entre los que destacan el desarrollo de aplicaciones académico-administrativas para la Facultad de Ingeniería (sistema de inscripciones, generación de horarios, asistencia de profesores, etc.).

Las utilerías de software que se encuentran instaladas en estas máquinas actualmente son las siguientes :

- Compiladores
  - ANSI C
  - Basic
  - Cobol
  - Fortran
  - Pascal

- Procesador de textos
  - Runoff

- Productos de comunicación

- Decnet
  - Pathworks
  - LAT (Local Area Transport)
  - TCP/IP

- Uso general

- DEC-FORMS (Sistema Administrador de Formas)

- Manejador de Bases de Datos y herramientas

- Oracle7
  - SQL\*Forms
  - SQL\*Menu
  - PRO\*C
  - PRO\*Fortran

Este software se encuentra propiamente configurado para operar y aprovechar la configuración del Cluster, aspectos que son tratados en los capítulos subsecuentes.

Por otra parte USECAFI cuenta también con más de 20 computadoras personales con MS-DOS y Windows, así como un conjunto de paquetes para estas computadoras que incluyen procesadores de texto (Wordperfect), paquetes de dibujo (AUTOCAD) y Microsoft Office entre otros.

---

### **Recursos Humanos**

Como se comentó al principio de este capítulo el personal que labora en USECAFI en su mayoría son Becarios y líderes de proyectos. Cada persona es responsable de una serie de actividades y objetivos que permiten mantener y continuar con la forma de trabajo actual que a continuación se describe.

La filosofía de trabajo dentro de la Unidad constituye todo un marco sistemático dentro del cual, un total de 32 personas actualmente desarrollan su trabajo con el firme objetivo de brindar el mejor servicio a las diferentes dependencias de la propia Facultad e incluso a otras dependencias universitarias y compañías privadas.

El jefe de USECAFI es el responsable ante la Secretaría de Servicios Académicos del establecimiento y funcionamiento de los sistemas que se desarrollan dentro de esta Unidad, de manera que satisfagan las necesidades de información de los usuarios finales a corto y largo plazo.

Se cuenta con un Secretario Auxiliar que es el responsable de brindar apoyo a la jefatura en todas las actividades de administración y técnicas que realice USECAFI y asumir el cargo de Jefe de la Unidad en caso de que así se requiera en determinado momento.

Por otro lado, los líderes de proyectos son responsables ante el jefe de la Unidad de la puesta en marcha de los sistemas que procesan la información de la Base de Datos de acuerdo al plan de automatización definido. Dirigen a los becarios, quienes trabajan sobre el proyecto desarrollando diversas actividades, entre las más importantes el desarrollo y programación de los Sistemas.

Los becarios constituyen aproximadamente el ochenta por ciento del personal que labora en USECAFI; son alumnos regulares de la Facultad, que por lo general es su primera experiencia laboral en este ámbito profesional lo que constituye por un lado un gran entusiasmo por participar en el desarrollo de las actividades propias de la Facultad, así como una oportunidad muy alta de constante capacitación. La rotación de personal es considerablemente alta por lo que no se alcanza a aprovechar al máximo las capacidades y conocimientos que con la experiencia en este Centro de trabajo van adquiriendo.

Esta rotación de personal provoca que no se mantenga un seguimiento fiel a los proyectos y sobre todo a la documentación de los mismos. Por otra parte, el software que se tiene para la administración de la Base de Datos permite realizar varias actividades tecnológicas de manera muy eficiente, pero el hardware con el que se cuenta es insuficiente en cuanto a recursos de memoria, disco e incluso de procesador, por lo que USECAFI se deberá enfocar en obtener mayores recursos para garantizar la eficiencia en los tiempos de ejecución de los sistemas que se desarrollan. Un punto que impera en esta configuración es que se utiliza el mismo equipo para el desarrollo y para la producción, lo cual origina que algunos usuarios constantemente reclamen por la lentitud de sus procesos, cuando éstos están diseñados e implementados de manera eficiente pero los recursos de hardware son insuficientes.

De esta forma es que se ha determinado integrar la Base de Datos actual de la Facultad de Ingeniería en un ambiente que resulta óptimo como es el caso del VAX Cluster y aprovechar las ventajas tecnológicas que provee el RDBMS Oracle versión 7; pero es recomendable que se contemple la posibilidad de crecer estos equipos ya que las mismas necesidades de la Facultad crecen, se propone eficiencia y calidad que puede ser incrementada con recursos acordes a las mismas necesidades y avances tecnológicos que en la propia Facultad de Ingeniería se enseñan y difunden a través de sus profesores; pero no se están llevando completamente a la práctica en esta institución que insiste en la vanguardia tecnológica.

## **ARQUITECTURAS Y PROCESOS**

En este capítulo se introduce el concepto de proceso en un ambiente computacional para poder tener una mejor comprensión de los esquemas de hardware actuales. Se presentan algunas de las definiciones más populares para poder llegar a definir las configuraciones de equipos de cómputo actuales y sus principales características.

Las definiciones y conceptos introducidos aquí sirven de base para las exposiciones en capítulos posteriores que hacen referencia a la operación del manejador de la Base de Datos y los equipos bajo una configuración en cluster.

Se han utilizado muchas definiciones del término proceso y aunque no hay un acuerdo universal sobre su definición, parece hacerse referencia más frecuentemente al concepto de programa que se está ejecutando.

El término proceso puede resultar confuso cuando se le usa para describir un entorno de multiprogramación, porque usualmente, se entiende que multiproceso significa un sistema con múltiples procesos que se ejecutan de manera simultánea. Cuando es necesario distinguirlo de multiprogramación, usamos el término multiproceso para indicar un sistema operativo que soporta la ejecución simultánea de programas en un solo procesador, sin complicar demasiadas formas de administración de archivos. A esta forma de funcionamiento se le conoce también como multitarea. La multiprogramación es un concepto más general que indica un sistema operativo que proporciona administración de memoria y de archivos, además de soportar la ejecución concurrente de programas. Así el sistema operativo multiprogramación es también un sistema operativo multiproceso, mientras que a la inversa no es implícito.

La manifestación de un proceso en un sistema operativo es un Bloque de Control de Proceso (Process Control Block PCB). El PCB es una estructura de datos que contiene cierta información importante acerca del proceso, incluyendo :

- Estado actual de proceso
- Identificación única del proceso
- Prioridad del proceso
- Apuntador para localizar la memoria del proceso
- Apuntadores para asignar recursos

El PCB es un almacenamiento central de información que permite al sistema operativo localizar toda la información clave sobre el proceso. Cuando el sistema operativo cambia la asignación del CPU entre los procesos, utiliza las áreas de preservación del PCB para mantener la información que necesita para reiniciar el proceso cuando consiga de nuevo la atención del CPU.

Las operaciones básicas para la administración de procesos son :

- Creación y destrucción
- Suspensión y activación
- Cambio de prioridad
- Bloquear y despertar

Para la creación de un proceso se deben contemplar algunas operaciones, entre ellas dar el nombre al proceso, insertarlo en la lista de procesos conocidos del sistema, determinar la prioridad inicial, crear el Bloque de Control (PCB) y asignar los recursos iniciales.

Un proceso puede crear un nuevo proceso. Si lo hace, el proceso que lo crea se llama proceso padre, y el proceso creado se llama proceso hijo. Tal creación produce una estructura jerárquica de procesos en la cual cada hijo tiene un solo padre, pero cada padre puede tener varios hijos.

Un proceso suspendido no puede proseguir hasta que otro proceso lo reanude. La suspensión es una operación importante y ha sido implementada de diversas formas en muchos sistemas diferentes.

Las suspensiones suelen ser de un breve período de tiempo. A menudo son realizadas por el sistema para sacar temporalmente ciertos procesos durante una situación de carga. Para las suspensiones de larga duración, los recursos del proceso deben ser liberados. La decisión sobre la liberación de los recursos depende, en gran parte, de la naturaleza del recurso. Por ejemplo, la memoria principal debe ser liberada inmediatamente después de la suspensión del proceso y una unidad de cinta puede ser retenida por el proceso durante una suspensión corta.

Reanudar o activar un proceso implica reiniciarlo en el punto donde fue suspendido.

La destrucción de un proceso es mucho más complicada si ha creado otros procesos. En algunos sistemas, un proceso hijo se destruye automáticamente al destruir el padre; en otros sistemas, los procesos hijos continúan independientemente de sus padres, pero la destrucción de los padres no tiene ningún efecto sobre los hijos que ha creado.

Durante su existencia, un proceso pasa por una serie de estados discretos. Varias circunstancias pueden hacer que un proceso cambie de estado.

Se dice que un proceso se está ejecutando, es decir, está en estado de ejecución, si tiene en ese momento el CPU. Un proceso se dice que está listo, cuando podría usar un CPU, si hubiera uno disponible. Se dice que un proceso está bloqueado cuando, espera que ocurra algo, como la terminación de una entrada salida, para poder ponerse en acción nuevamente. Hay mas estados de proceso, pero éstos son los más importantes que permiten generalizar el funcionamiento.

Si se considera un equipo de cómputo con un solo procesador, sólo puede estar corriendo un proceso al mismo tiempo, pero puede haber varios procesos listos y varios más bloqueados. Por lo tanto, se establece una lista de procesos en estado de listos, y una lista de bloqueados. La lista de los procesos listos se mantiene en orden prioritario; sin embargo, la lista de procesos bloqueados está desordenada porque no se desbloquean en orden prioritario; en lugar de esto, los procesos se desbloquean en el orden en que tienen lugar los eventos por los que están esperando.

Cuando un trabajo es admitido por el Sistema Operativo es insertado en la última parte de la lista de procesos listos. El proceso se va moviendo gradualmente hacia la cabeza de esta relación a medida que se completan los procesos anteriores. Cuando el proceso llega a la cabeza de la lista, y cuando el CPU se encuentra disponible, el proceso recibe el CPU y se dice que hace una transición de estado, del estado listo al estado en ejecución. La asignación del CPU al primer proceso de la lista de procesos listos es llamado despacho, y es ejecutado por la entidad del sistema llamado despachador.



*Transiciones de un proceso*

Mientras el proceso tenga el CPU, se dice que está en ejecución. Para prevenir que cualquier proceso monopolice el sistema, ya sea de manera accidental o maliciosamente, el Sistema Operativo ajusta un reloj de interrupción del hardware para permitir al usuario ejecutar su proceso durante un intervalo de tiempo específico. Si el proceso no abandona voluntariamente el CPU antes de que expire el intervalo, el reloj genera una interrupción, haciendo que el Sistema Operativo recupere el control. El Sistema Operativo, entonces, hace que el proceso que anteriormente se hallaba en estado de ejecución pase al de listo, y hace que el siguiente proceso de la lista de listos pase al estado en ejecución.

---

Si un proceso que se encuentra en estado de ejecución inicia una operación de entrada/salida antes de que termine su intervalo de tiempo, el proceso voluntariamente abandona el CPU, es decir, que el proceso se bloquea a sí mismo hasta la terminación de la operación de entrada/salida.

La otra transición posible en este modelo de tres estados ocurre cuando acaba una operación de entrada/salida, o alguna otra causa por la que esté esperando el proceso. El proceso cambia del estado bloqueado al estado listo.

Vale la pena hacer notar que la única transición de estado iniciada por el propio proceso es el bloqueo; las otras tres transacciones son iniciadas por entidades ajenas.

---

### MULTIPROCESAMIENTO.

A medida que ha bajado el precio del hardware de las computadoras se ha producido una tendencia hacia el multiprocesamiento, es decir la configuración de un sistema de cómputo con varios procesadores y la masificación del paralelismo. Si ciertas operaciones se pueden ejecutar en paralelo de forma lógica entonces el reto está en que la nueva generación de computadoras ejecuten físicamente en paralelo.

La concepción del procesador como el recurso más valioso de un sistema de computación dado, fue una idea durante las dos primeras décadas de la computación moderna, pero ya no resulta aplicable. Con el multiprocesamiento, es ahora más importante considerar los temas de confiabilidad, paralelismo en computación, esquemas óptimos de conexión y contención entre procesadores que intentan acceder a los mismos recursos.

El procesamiento paralelo es interesante por varias razones. La gente parece más capaz de centrar su atención en una sola actividad a la vez que de pensar en paralelo. Es difícil determinar cuáles actividades pueden o no ejecutarse en paralelo. Los programas en paralelo son mucho más difíciles de depurar que los programas secuenciales.

Entre los esquemas de procesamiento actuales se encuentran clasificados por el número y arreglo de los procesadores las siguientes arquitecturas :

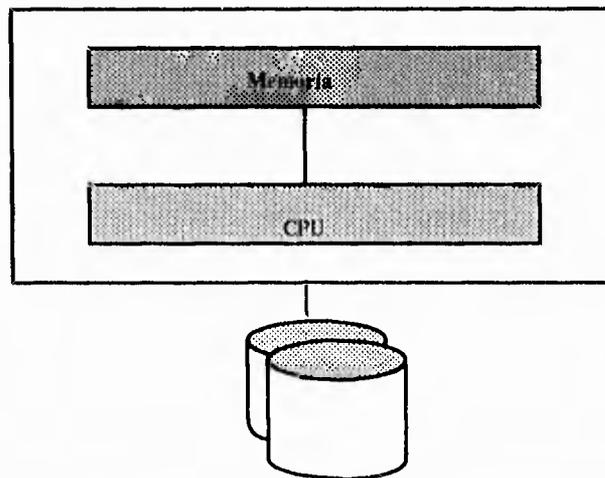
- Uniprocador
- Simétricamente paralelo ( Symmetric Multiprocessing (SMP) )
- Cluster o Arquitectura de bajo acoplamiento ( Loosely Coupled )
- Masivamente paralelo ( Masively Processor Parallelism (MPP) )

La asignación de los procesadores físicos a los procesos es lo que permite que estos últimos realicen su trabajo. Esta asignación es un problema complejo que maneja el sistema operativo; dado que debe determinar cuándo deben asignarse los procesadores y a qué procesos.

### Uniprocador

Es el esquema de cómputo en el que la unidad central cuenta exclusivamente con un procesador para la solución y ejecución de las tareas que se activan.

Los recursos, esencialmente el CPU se dedica a un solo proceso a la vez; por lo tanto cada programa tendrá un tiempo determinado el procesador para la solución y culminación de las actividades que deba realizar, y se tendrá que alternar este recurso con los otros procesos que estén activos en el Sistema.



**Simétricamente Paralelo (SMP)**

Una tendencia significativa en el campo de la computación actual es hacia el uso, cada vez mayor, del multiprocesamiento, es decir, de configurar un sistema de computación con varios procesadores. El multiprocesamiento se ha utilizado en los sistemas durante varias décadas, pero está siendo de nuevo objeto de un intenso interés debido a la disponibilidad de microprocesadores de bajo costo.

Los microprocesadores han hecho posible conseguir ciclos de procesamiento por un costo mínimo, y su tamaño reducido hace razonable pensar en empaquetar varios de ellos en un solo sistema.

Uno de los atractivos de los sistemas de multiprocesamiento es que si un procesador falla, los procesadores restantes continúan operando. Esto no es automático y requiere de un diseño cuidadoso. Un procesador que falla tendrá que informar de alguna manera a los demás procesadores, que deben hacerse cargo del trabajo. Los procesadores en funcionamiento deben pues, ser capaces de detectar el fallo de un procesador determinado.

El sistema operativo debe percibir que ha fallado un procesador determinado y que ya no se encuentra disponible para ser asignado. Además, como ahora el sistema tiene menos capacidad, el sistema operativo deberá ajustar sus estrategias de asignación de recursos para evitar la sobrecarga del sistema.

Es interesante observar que la mayoría de los sistemas de multiprocesamiento tienen como meta principal el incremento de la capacidad de ejecución. La programación sigue siendo una actividad esencialmente secuencial y pocos programas son escritos para explotar la concurrencia. Existen varias razones para esto :

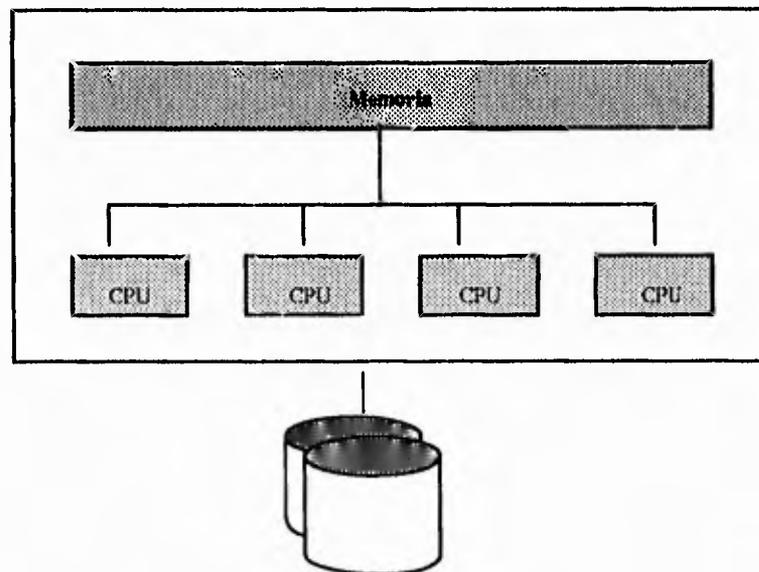
- Las personas pensamos en forma secuencial.
- Ningún lenguaje humano proporciona la expresión adecuada de paralelismo.
- Se están apenas usando estas arquitecturas y existe aun poca experiencia.
- El hardware de la mayoría de las computadoras está orientado hacia la operación secuencial.

---

El multiprocesamiento simétrico es una configuración en la que todos los procesadores son idénticos, el sistema operativo los administra dado que cualquiera de ellos puede ser utilizado para controlar algún dispositivo de Entrada/Salida o hacer referencia a cualquier unidad de almacenamiento.

Un proceso en ejecución en un sistema de multiprocesamiento simétrico puede ser ejecutado en diferentes momentos por cualquiera de los procesadores equivalentes. Todos los procesadores pueden cooperar en la ejecución de un proceso determinado.

Una característica importante en esta arquitectura es que todos los procesadores están conectados de tal forma que comparten la memoria del sistema a través de un bus común. Se genera por lo tanto una restricción en el número de procesadores que se pueden tener conectados para obtener un mayor rendimiento, dado que no por aumentar el número de procesadores se obtendrá necesariamente un mayor rendimiento, ya que se genera un cuello de botella con el bus de comunicación entre los procesadores y el arreglo de memoria.

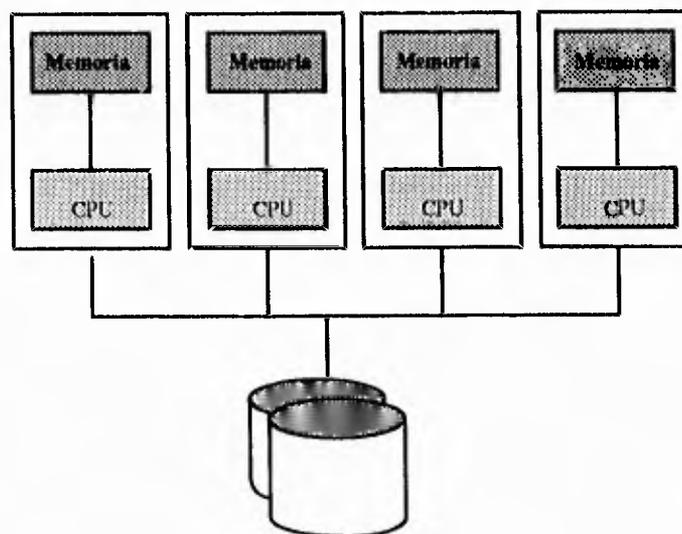


## Cluster

El Cluster o Loosely Coupled utiliza arreglos de almacenamiento compartido principalmente y está compuesto por varios procesadores. Esta clase de multiprocesamiento incluye la conexión de dos o mas sistemas de cómputo independientes por medio de un enlace de comunicación. Cada equipo tiene su propio Sistema Operativo y almacenamiento. Los equipos pueden funcionar de manera independiente y comunicarse cuando sea necesario. Los sistemas separados pueden acceder a los archivos del otro a través del enlace de comunicación.

En esta arquitectura no existen recursos de memoria compartida. Cada procesador tiene su propio recurso de memoria local y los procesadores se comunican a través de un bus de área local, lo cual limita su separación en unos cuantos metros.

Este arreglo de procesadores y memoria propia, resuelve de alguna forma el problema que se presenta con los equipos Simétricamente Paralelos (SMP's), ya que se tiene un acceso directo del procesador al arreglo de memoria local.



Se puede observar que existe el bus que conecta cada módulo o equipo independiente, lo que indica que se tendrá un canal de comunicación común entre los equipos y si se aumenta el número de equipos no implica que se aumentará el rendimiento en las transacciones que se ejecutarán.

Esta arquitectura está diseñada para aplicaciones que requieren dividir el proceso de ciertas transacciones particulares en algún equipo, utilizando los recursos de CPU y memoria de ese equipo sin afectar el rendimiento que se desea tener en otras transacciones que se estén ejecutando en otro equipo con otros recursos de procesadores y memoria propios; pero sin embargo comparten dispositivos de almacenamiento, como disco y unidades de cinta.

### **Paralelismo Masivo.**

El paralelismo masivo nos ofrece una forma de ejecutar hasta su terminación un programa dado en el tiempo más corto posible.

Al utilizar muchos CPU's se pretende aumentar la confiabilidad, disponibilidad y el poder de cómputo. La confiabilidad se mejora debido a que si un procesador falla, los otros pueden continuar funcionando. El sistema continuará operando, aunque con una reducción de su capacidad. El aumento del poder de computación es posible gracias a la combinación del poder de mezclar varios procesadores.

Algunas veces resulta más atractiva la conversión a multiprocesamiento para incrementar la capacidad que añadir sistemas de computación adicionales completos. Los sistemas múltiples pueden implicar considerables problemas de espacio, pueden requerir de labor adicional para mantenerlos, etc.

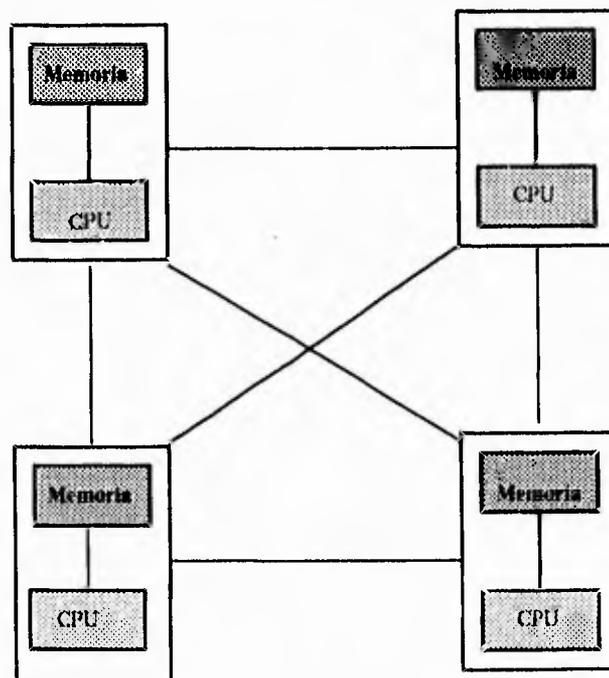
Los sistemas de multiprocesamiento proporcionan una flexibilidad importante en los sistemas computacionales. Su diseño modular puede facilitar la expansión de la capacidad de una instalación al añadir procesadores a medida que se hacen necesarios.

En la arquitectura masivamente paralela se elimina el tener que compartir un solo arreglo de memoria en un equipo de cómputo como lo es con los simétricamente paralelos (SMP's), ya que se cuenta con varios módulos de manera independiente, que están compuestos por un procesador o un arreglo pequeño de procesadores con un arreglo de memoria local. Por otra parte el bus de conexión que se tiene entre los módulos de CPU's y memoria como en el caso del cluster se elimina con un bus que va de cada módulo a cada uno de los otros, de tal forma que se tienen tantas conexiones como módulos existan.

La arquitectura masivamente paralela es apropiada para aplicaciones que requieren una alta escalabilidad, alto rendimiento y sobre todo aplicaciones orientadas a transacciones en línea (OLTP), altamente complejos y de grandes volúmenes de información.

Se tiene un alto rendimiento en estos equipos, no sólo por la capacidad de procesamiento, sino también porque este tipo de módulos de memoria y procesadores independientes se utilizan para soportar los dispositivos de entrada/salida, comunicaciones y en general las interfaces de red.

Este hardware está diseñado para manejar cientos de nodos independientes, para que si alguno falla durante alguna transacción, cualquiera de los demás sean capaces de concluir la tarea, sin afectar los procesos que se puedan estar ejecutando de manera concurrente o paralela.



## CARACTERISTICAS TECNOLOGICAS DEL RDBMS ORACLE7

La versión mas reciente del manejador de Bases de Datos Oracle es la versión 7, y es reconocida en el mercado y por la literatura como Oracle7. Las principales características de operación son discutidas en este capítulo; ya que si se desea operar con esta nueva versión es conveniente que se tengan en consideración las tendencias en el manejo de la información por este manejador de Base de Datos para que sea aprovechado al máximo.

El diseño del RDBMS y herramientas de Oracle están orientadas al soporte de sistemas que requieren trabajar grandes volúmenes de información, transacciones en línea, y garantía de soporte para los sistemas abiertos y tecnologías de punta como cliente/servidor o esquemas distribuidos de información.

Antes de comenzar con las características de Oracle es conveniente mencionar unos aspectos generales de esta compañía y la evolución de sus productos, ya que considero de valiosa ayuda como apoyo a la justificación del uso de este software en la solución que se está planteando en este trabajo.

En 1969, en un laboratorio de IBM en San José California, un ingeniero llamado E.F. Codd inició una revolución en el concepto de la manipulación de la información cuando propuso un nuevo enfoque para el manejo de los datos. Este enfoque "**manejador de datos relacionales**" permitía a los usuarios organizar los datos de una forma mas intuitiva que antes. En lugar de imponer un orden jerárquico para la información, el enfoque relacional acomoda los datos en tablas de información relacionada. Diez años después de que Codd introdujera el método relacional, esta tecnología fue introducida al campo comercial, no por medio de IBM, sino por Oracle Corporation. Con cuatro personas y un solo contrato, los fundadores de Oracle no solo trajeron al mercado el primer Sistema Manejador de Bases de datos Relacionales (RDBMS) basado en SQL, sino que fueron hacia adelante para desarrollar una de las compañías de tecnología de gran crecimiento en esta industria.

En el camino Oracle ha dado una serie de primicias tecnológicas que ha publicado en distintas fuentes de información :

1983 Primer RDBMS portable.

Primer RDBMS en soportar multiprocesadores simétricos.

1984 Primer RDBMS para PC's basado en SQL.

1985 Primer RDBMS cliente-servidor.

1986 Primer RDBMS con capacidad para consultas distribuídas.

1987 Primer RDBMS para redes de PC's.

1988 Primer marca de 100tps en el Benchmark TP1.

1990 Primer servidor de Base de Datos para Macintosh.

1991 Primer RDBMS abierto para ambientes paralelos.

Primer Benchmark TPC/B con más de 1,000 tps.

Primer RDBMS certificado como 100% estándar.

1992 Primer RDBMS Netware Loadable Module (NLM) certificado por Novell.

Primer Benchmark TPC/A de Sistemas Abiertos con mas de 500tps

Primer RDBMS de servidores cooperativos.

En 1982, los fundadores de Oracle reconocieron la tendencia hacia Sistemas Abiertos y reescribieron el software de la compañía para que pudiera ser portado a cualquier plataforma de cómputo. Esta estrategia les ha funcionado bien hasta ahora, ya que Oracle es la compañía independiente de Bases de Datos mas grande del mundo y la tercer compañía mas grande en la industria de software.

Oracle Corporation provee el Sistema Manejador de Bases de Datos Relacionales (RDBMS), herramientas para el desarrollo de sistemas, aplicaciones y servicios a grandes corporaciones alrededor del mundo. La estrategia de Oracle es proveer la tecnología y los recursos necesarios para desarrollar soluciones eficientes en cuanto a costo y rendimiento de los aspectos complejos en el manejo de información; y así mejorar los resultados del negocio. En un medio ambiente caracterizado por los cambios constantes, el software de Oracle permite a los clientes proteger las inversiones en hardware y software asegurando la continuidad hacia los avances futuros. Los productos de Oracle se van adecuando a las necesidades de las instituciones ya que se caracterizan por ser para ambientes distribuidos, portables, abiertos e integrados. Los clientes típicos de Oracle son los que usan este software para el desarrollo de aplicaciones de misión crítica, desde soporte a la toma de decisiones hasta sistemas de gran escala de procesamiento de transacciones en línea (OLTP On Line Transaction Processing) y explotando la tecnología cliente-servidor.

El producto que lleva el nombre de la compañía es el "Oracle Server" (cuya última versión es Oracle7), una Base de Datos de servidores cooperativos muy avanzada. Oracle también provee una variedad de herramientas para el desarrollo de aplicaciones y automatización de oficinas; paquetes contables, de manufactura y de recursos humanos; así como de servicios de consultoría, capacitación y soporte técnico. El software de Oracle está disponible para las principales plataformas de cómputo, desde PC's hasta mainframes y sistemas masivos paralelos.

Para aplicaciones en línea críticas que demandan alto rendimiento y confiabilidad, Oracle ofrece una alternativa para los esquemas de mainframes y altos costos. Decenas de empresas de todo el mundo han adoptado Oracle como manejador de Base de Datos para hacer un "downsizing" de sus sistemas de información al construir sistemas distribuidos y flexibles. El software de Oracle se adapta automáticamente a las redes de las empresas en la mayoría de los casos.

Con el crecimiento que ha tenido Oracle Corporation, se han hecho grandes inversiones en investigación y desarrollo para ofrecer tecnología sólida que provea soluciones computacionales completas para los negocios. La trayectoria que ha tenido Oracle en innovación y liderazgo ha asegurado a los usuarios que tendrán la tecnología para adaptarse a los cambios que surjan en las necesidades de sus instituciones.

Oracle ha enfocado sus esfuerzos al desarrollo de importantes tecnologías emergentes y ha liderado la industria en la implementación de soluciones basadas en multiprocesadores simétricos, sistemas abiertos y plataformas masivas en paralelo.

---

La tecnología de Oracle está orientada a ambientes de cómputo heterogéneos; por lo que el software de Oracle y los servicios hacen mas accesible la información reduciendo los costos de los sistemas. Los componentes claves de la estrategia de Oracle son :

Software portable que corre virtualmente en cualquier tipo de computadora, desde PC's hasta mainframe.

Software abierto que puede operar con Bases de Datos, aplicaciones y ambientes de programación de otros proveedores.

Servidores de Bases de Datos que permitan a los usuarios acceder automáticamente múltiples fuentes de datos.

Herramientas de desarrollo y aplicaciones horizontales que provean soluciones completas.

Servicios de consultoría orientados a lograr la convergencia entre los sistemas de información y las necesidades del negocio.

Para cumplir estas metas, Oracle consistentemente ha sido el líder en la industria en inversión para investigación y desarrollo. En el año fiscal 1989 se invirtieron 89 millones de dólares en este rubro; para el año fiscal 1992 esta cifra subió a 153 millones. Con esta inversión en tecnología, Oracle ha logrado varias primicias importantes en la industria como las que se enunciaron anteriormente.

Este compromiso permite a la compañía invertir en nuevas tecnologías, como sistemas masivos en paralelo, tecnología orientada a objetos, comunicaciones satelitales, etc., asegurando que los productos de Oracle se adaptarán a las tecnologías nacientes.

El mercado de Bases de Datos ha cambiado porque las empresas han movido sus Bases de Datos de mainframes costosos y de alto nivel de mantenimiento a computadoras mas pequeñas y mas económicas. Desde 1985 la proporción de RDBMS's corriendo en computadoras medianas ha crecido de apenas el 5% a mas de 60%.

En 1992 el mercado mundial para servidores de Bases de Datos relacionales era de aproximadamente 2.1 billones de dólares de acuerdo a Gartner Group (Grupo de investigadores independientes de aspectos tecnológicos de interés global que emiten sus estudios e investigaciones de manera imparcial); y proyecta un crecimiento a 5.3 billones de dólares para 1997. El crecimiento mas rápido para Oracle y para el mercado de RDBMS's en general ha sido en sistemas UNIX. En el semestre terminado en Noviembre de 1992, los ingresos de Oracle por ventas en el mercado de UNIX creció mas del 56% con respecto al año anterior, mientras los ingresos para servidores "desktop" creció mas del 44% en el mismo período. Los ingresos de estos dos segmentos representaron mas de 65% del ingreso total de licencias de Oracle para el año fiscal 1992.

Como se ha comentado, la versión mas reciente del RDBMS de Oracle es Oracle7 y se caracteriza por resolver algunos puntos que hasta el momento no se habían podido solucionar de una manera simple; como es el caso de consultar o actualizar información de mas de un servidor al mismo tiempo sin programación adicional, en un ambiente de Base de Datos distribuidas.

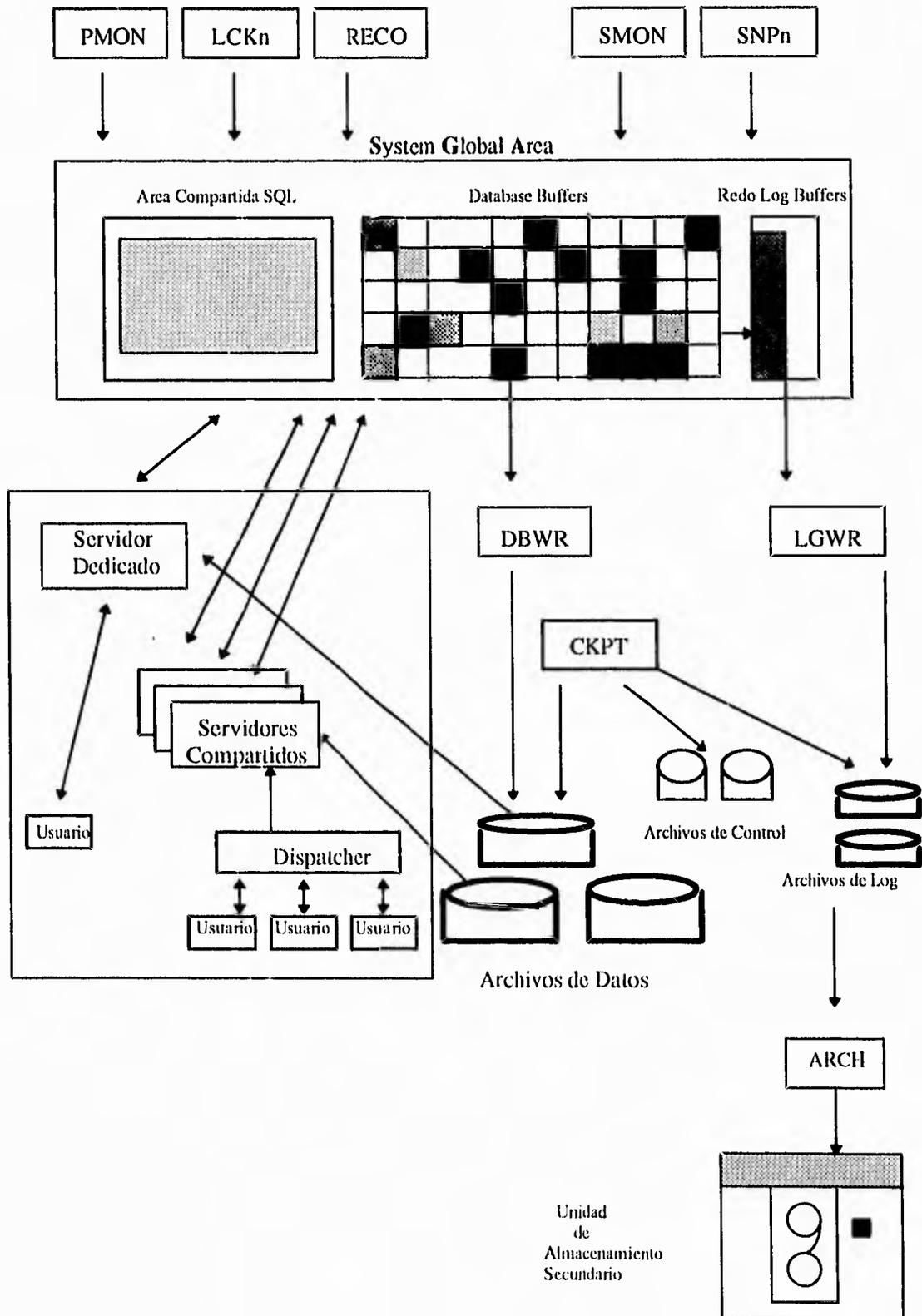
Con Oracle7 las aplicaciones pueden acceder información en múltiples servidores sin codificación adicional. En términos técnicos una consulta o actualización (Select o Update) estándar de SQL trabaja de forma idéntica si los datos están almacenados en una sola o en múltiples computadoras.

Oracle7 es completamente portable, lo que es indispensable para las instituciones que quieren proteger sus inversiones en equipo de cómputo. Debido a que Oracle7 es abierto, las aplicaciones pueden acceder a la información que se encuentre almacenada en Oracle, DB2 o casi cualquier otro producto de distintos proveedores. Al mismo tiempo, la funcionalidad de Oracle7 esconde la complejidad dando a los usuarios la impresión de que toda la información viene de la misma fuente.

Oracle7 también incluye soporte a procedimientos almacenados e integridad referencial de acuerdo a los estándares definidos por el comite ANSI.

Las características adicionales en Oracle7 es la optimización del manejo de Bases de Datos de gran escala y el soporte real a Bases de Datos distribuidas. Por ejemplo, el "two-phase-commit" transparente y automático de Oracle7 asegura que los cambios a los datos ocurran simultáneamente en todas la Bases de Datos involucradas. Oracle7 también permite a los usuarios centralizar reglas del negocio en el RDBMS en lugar de definir las en las aplicaciones, ofreciendo mayor consistencia y eficiencia.

A continuación se hace una explicación mas detallada de algunas de estas características que son importantes tener en cuenta para el proceso de optimación que se llevará a cabo con el presente trabajo. Comencemos pues con la explicación de la arquitectura del RDBMS con respecto al manejo de memoria y procesos.



Los componentes principales de una Instancia y Base de Datos Oracle7 son los archivos físicos, el Area Global de memoria (SGA) y los procesos.

Vale la pena enfatizar que una instancia en Oracle se define como un SGA y los procesos de background asociados a los archivos que conforman una Base de Datos física. Estos procesos de Background se activan automáticamente al momento de levantar la Base de Datos con el comando STARTUP desde la utilería SQL\*DBA de Oracle y realizarán actividades específicas que se explicarán más adelante. De igual forma el SGA es un área de memoria compartida que está definida y apartada al momento de levantar la Base de Datos; es por esto que el tamaño que se tiene de memoria principal para tal efecto, dependerá de los valores especificados en el archivo INIT.ORA.

### **1. Estructuras de memoria y procesos.**

En esta sección se explican los componentes y procesos de los que el RDBMS de Oracle se vale para la administración de los usuarios y la información de la Base de Datos.

El SGA (System Global Area) es una región de memoria principal compartida que Oracle aparta al momento de activar o levantar la Base de Datos, en esta zona se almacenan datos e información de control general de una instancia de Oracle. Esta memoria que se aparta y es conocida como SGA se activa cada vez que se ejecuta un STARTUP a la Base de Datos y se liberará cuando se da SHUTDOWN, es por esto que cada vez que se activa una instancia se generará un SGA y su tamaño dependerá de los valores indicados a los parámetros del archivo INIT.ORA.

Los datos almacenados en el SGA son compartidos por todos los usuarios que se conectan a través de alguna herramienta de Oracle a la Base de Datos para manipular la información. Por razones de rendimiento es recomendable tener un SGA lo suficientemente grande para mantener bastantes datos en memoria principal y realizar menos accesos a disco.

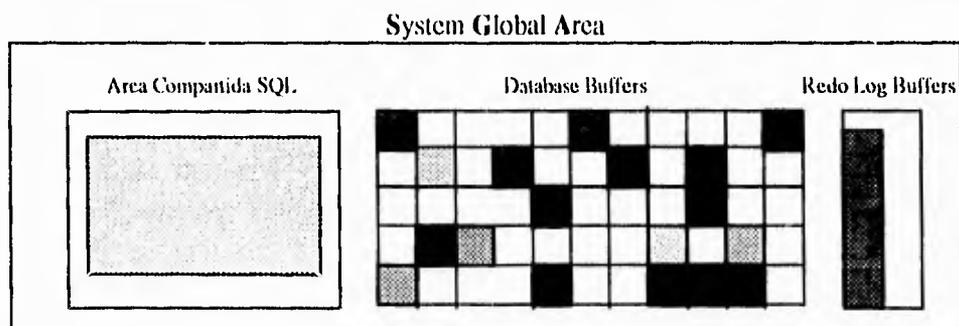
---

La información almacenada en el SGA está dividida en distintos tipos de estructuras de memoria, estas son *database buffers*, *redo log buffers* y *shared pool o área de SQL compartida*. Estas áreas son de tamaño fijo y son creadas al ejecutar el startup a la Base de Datos como se ha comentado.

**Database Buffers :** Son los bloques mas recientemente usados de la Base de Datos en memoria. El conjunto de estos bloques de datos en el SGA se llama *database buffer cache* y pueden haber sido modificados y accedidos por los usuarios y no necesariamente se han hecho permanentes estos cambios en los archivos que corresponden a la Base de Datos.

**Redo Log Buffers :** En esta zona se almacenan unicamente los cambios realizados a los Datos; es decir si se realiza un cambio en un bloque del Database Buffer por ejemplo, en esta zona se tendrá registrado solamente el dato que se modificó y no el bloque completo, ya que un bloque de información contiene varios bytes y solo se modifican algunos del total del bloque. Esta información se baja a los archivos de log (REDO LOG) y se emplea cuando es necesaria una recuperación de la Base de Datos.

**Shared Pool :** Es una zona compartida de sentencias SQL de igual forma en el SGA que contiene la ruta y plan de ejecución de dicha instrucción. Con esta característica se tienen las instrucciones de SQL en un lugar compartido optimizando las aplicaciones que utilizan las mismas sentencias de SQL, liberando de esta forma memoria que se emplearía si cada usuario que ejecuta una aplicación común mantiene la instrucción en su zona de memoria independiente, como se realizaba hasta la versión 6 de Oracle.



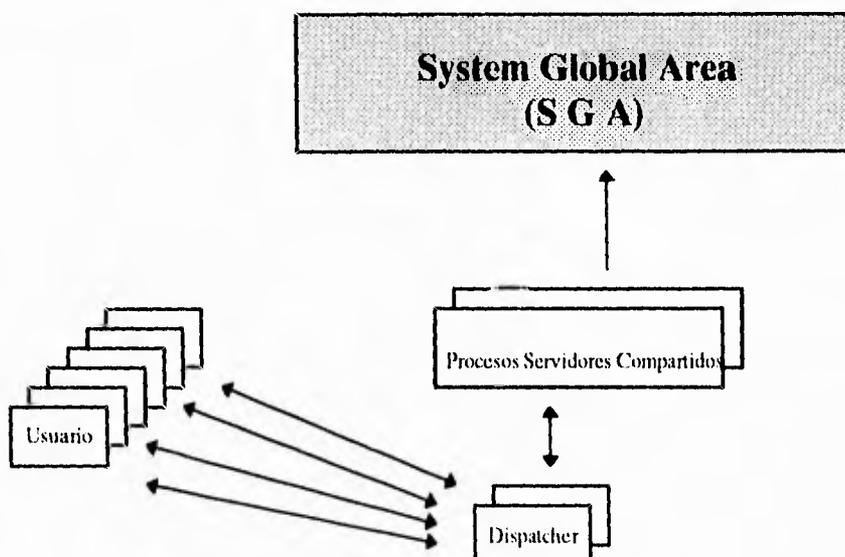
En una configuración con Oracle, se reconocen básicamente dos tipos de procesos : procesos de usuarios o procesos clientes y procesos de Oracle.

Un proceso cliente o de usuario es creado cuando se utiliza alguna herramienta de Oracle (como SQL\*DBA) o se ejecuta algún programa desarrollado con las utilerías de Oracle (como un programa de PRO\*C).

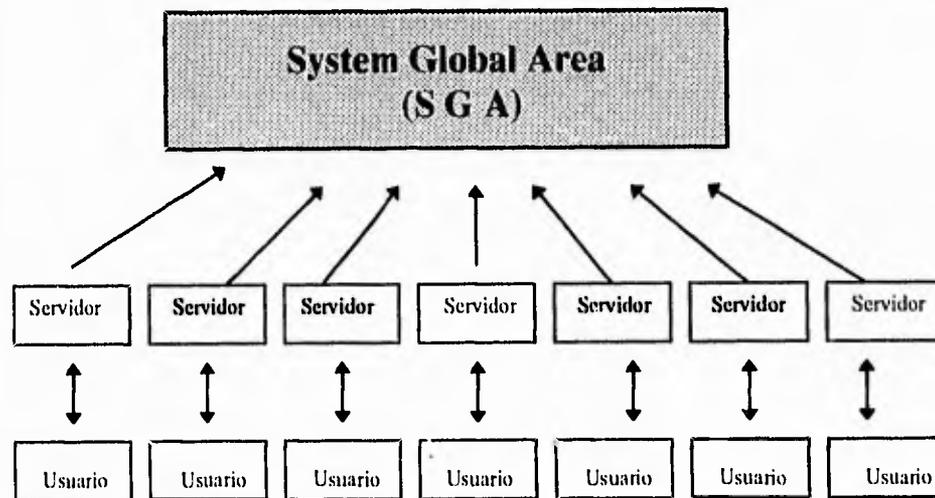
Los procesos de Oracle son activados por otros procesos para ejecutar alguna actividad específica. Estos procesos son :

**Procesos Servidores :** Oracle crea procesos servidores para manejar los requerimientos establecidos por los procesos clientes. Un proceso servidor es el encargado de establecer la comunicación entre el proceso cliente y las estructuras de memoria de Oracle para las solicitudes de dicho proceso cliente. Por ejemplo, si un usuario consulta algunos datos que aún no se encuentran en el SGA , el proceso servidor asociado lee dichos bloques de los archivos de datos y los coloca en el SGA.

Se pueden configurar un determinado número de procesos servidores compartidos para todos los usuarios que deseen acceder la información de la Base de Datos. Adicionalmente el Administrador de la Base de Datos puede configurar procesos servidores dedicados. Esta configuración de compartir procesos servidores por un número determinado de usuarios se llama *multi-threaded*.



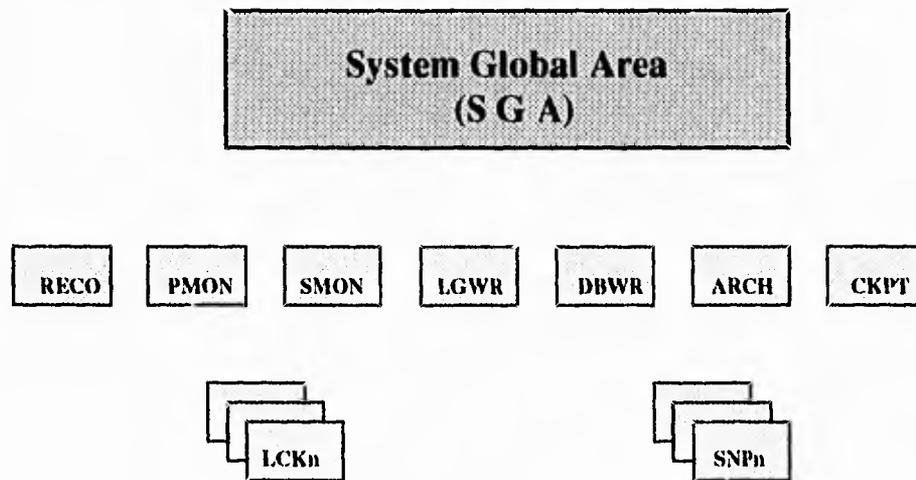
Oracle7 se puede configurar para trabajar bajo una arquitectura multi-threaded y para que trabaje en forma tradicional en la que cada proceso cliente cuenta con un proceso servidor dedicado automáticamente, como se hacía hasta la versión 6.



Las aplicaciones no se tienen que modificar dado que ésto es solo una característica del RDBMS para disminuir el consumo de recursos que se emplean al levantar tantos procesos servidores como procesos clientes se tengan activos en una instalación.

**Procesos de Background :** Oracle crea un conjunto de procesos para la administración y operación de una instancia, estos procesos son activados automáticamente al momento de levantar la Base de Datos y realizan una serie de actividades que consolidan las operaciones realizadas por los procesos de los usuarios.

Para cada instancia se pueden activar los distintos procesos de background, pero indistintamente se activarán al menos los que se identifican como DBWR, LGWR, SMON y PMON. Dependiendo de la configuración a través de parámetros se pueden activar: CKPT, ARCH, RECO, Dnnn, LCKn; que son descritos a continuación:



Esta figura muestra lo que se conoce como una instancia en Oracle, un SGA y una serie de procesos de background para la administración de procesos e información con la Base de Datos.

**Database Writer (DBWR) :** Escribe los bloques modificados del database buffer a los archivos de la Base de Datos. Los bloques menos recientemente usados son los que se escriben primero. Este proceso solo se dedicará a escribir o bajar dichos bloques sin importarle si se le ha dado commit o no a la transacción. Se optimiza el acceso a disco ya que en un solo acceso al SGA se pueden bajar varios bloques.

Otra causa por la que se activa este proceso es cuando se ha llenado la zona reservada para los buffers de la Base de Datos y algún proceso de usuario requiere manipular información que no se encuentra en el SGA y se tienen que buscar bloques libres para cargar los que el proceso esté solicitando.

**Log Writer (LGWR) :** Escribe la información almacenada en la zona del redo log buffer en los archivos de log (redo log's). Este proceso de background se activará cuando una transacción se concluya con commit, en este momento se baja esta información que contiene principalmente la dirección y el dato que se modificó de la Base de Datos. Hasta este punto pudo no haberse bajado el bloque real que está en el database buffer a la Base de Datos.

La consistencia de la Base de Datos se encuentra en los archivos de log, debido a que si algún usuario requiere ahora consultar el dato que se acaba de modificar y consolidar con commit, se está observando el dato real y último en el database buffer y eventualmente el DBWR bajará ese bloque a la Base de Datos. En caso de que haya habido alguna falla antes de que se active el DBWR Oracle podrá realizar la recuperación de la transacción a partir de lo que se almacenó en los archivos de log.

**Checkpoint (CKPT) :** En intervalos de tiempo, los bloques modificados del database buffer son escritos a los archivos de la Base de Datos por el DBWR; este evento es llamado checkpoint. Con este evento además de realizar la escritura de los bloques de memoria a disco, implica modificar todos los encabezados de los archivos de la Base de Datos y de control indicando el checkpoint más reciente. El proceso CKPT es opcional, y si no existe, LGWR asume la responsabilidad para efectuar las tareas del checkpoint.

**System Monitor (SMON) :** Este proceso realiza la recuperación de una instancia al momento del startup; esto quiere decir que si la Base de Datos no fue dada de baja con shutdown normal, al momento de levantarla se necesitará realizar una recuperación a partir de la información que se encuentra en los archivos de log, ya que en ellos se tiene la consistencia de la información.

Antes de dejar la Base de Datos activa para que sea utilizada por todos los usuarios, se verificará que los cambios registrados en los archivos de log se hayan aplicado físicamente en la Base de Datos.

---

El SMON también realiza la liberación de los bloques temporales utilizados durante el proceso de una operación con order by, group by, etc.

**Process Monitor (PMON)** : Este proceso realiza la recuperación necesaria cuando un proceso cliente o de usuario falla o ya no existe. El PMON es el responsable de liberar los recursos que tenía apartados el proceso de usuario que falló. Esta falla puede ser por ejemplo, la interrupción del proceso, o que se haya abortado o destruido a nivel sistema operativo por el administrador del equipo.

**Archiver (ARCH)** : La función del proceso de archiver es copiar los archivos de log a otro dispositivo (disco o cinta) cuando se han llenado. De esta manera se podrá realizar alguna recuperación de la Base de Datos hasta el último momento en que se causó la pérdida de la información.

Es conveniente aclarar que el proceso ARCH se activa únicamente cuando la Base de Datos está en modo archive log.

**Recover (RECO)** : Este proceso es usado para resolver transacciones distribuidas que han quedado pendientes por fallas en el canal de comunicación entre las Bases de Datos distribuidas involucradas para concluir la transacción.

El proceso se activará en intervalos de tiempo para intentar conectarse al otro nodo y automáticamente concluir la transacción con commit o rollback en ambos nodos.

**Dispatcher (Dnnn)** : Los dispatchers son procesos opcionales que son empleados en la configuración multi-threaded de Oracle7 únicamente. Se crearán varios procesos dispatcher para rutear los procesos de usuarios o procesos clientes con un proceso servidor compartido.

**Lock (LCKn)** : Se pueden activar hasta 10 procesos lock (LCK0, ..., LCK9) para el manejo de candados entre instancias en una configuración de Parallel Server, como es el caso de un cluster; en el que a través de dos o mas instancias se manipula la misma Base de Datos.

El siguiente ejemplo indica de manera simple la forma en que Oracle trabaja bajo una arquitectura cliente-servidor con servidores dedicados por default, es decir sin configurar el multi-threaded server :

1. La Base de datos está activa y disponible para los usuarios en la máquina que funciona como servidor; y por lo tanto los procesos de background están preparados para realizar las operaciones correspondientes.
2. Un usuario desde su estación de trabajo se conecta a la Base de Datos a través de un producto de Oracle como puede ser SQL\*Plus. Esta conexión se lleva a cabo con el driver de SQL\*Net correspondiente al protocolo con el que se desee trabajar.
3. En el servidor se está ejecutando también el correspondiente driver de SQL\*Net. Se detecta la conexión y se realiza la petición para la creación de un proceso servidor dedicado que corresponda a la conexión del proceso cliente.
4. El usuario ejecuta alguna sentencia SQL. Por ejemplo inserta un registro en una tabla.
5. El proceso servidor dedicado, recibe la instrucción. En este punto se pueden seguir dos caminos para procesar dicha instrucción :

- En caso de que en el área compartida de SQL se encuentre una instrucción idéntica, el proceso servidor podrá utilizar el plan de ejecución que se ha encontrado.

- Si la instrucción no existe en el área compartida de SQL, se aparta espacio para almacenar dicha instrucción.

El proceso servidor checa que el usuario tenga los privilegios necesarios sobre la Base de Datos para efectuar la operación.

6. El proceso servidor extrae los datos de los archivos de la Base de Datos si es necesario, o usa los datos ya existentes en el database buffer según sea el caso.

7. El proceso servidor ejecuta la instrucción de SQL almacenada en la zona de SQL compartido. Los datos o cambios son almacenados en el SGA. El DBWR eventualmente se activará para hacer permanentes los cambios en disco. El LGWR almacena los datos de la transacción en los archivos de log, únicamente si la transacción fue concluida con commit.

8. De acuerdo con el resultado de la transacción, el servidor de la Base de Datos envía un mensaje a través de la red al usuario.

9. Mientras se está ejecutando esta operación, los otros procesos de background se activarán en caso de que sea necesario de acuerdo a las condiciones propias de la configuración con que se esté trabajando. Adicionalmente el RDBMS estará manejando otras transacciones y validando la contención entre dichas transacciones y requerimientos de acceso a los datos por los procesos de los usuarios.

El siguiente ejemplo indica, de igual forma, de manera simple la operación de Oracle7 bajo una arquitectura con la configuración del multi-threaded server :

1. El servidor de la Base de Datos tiene activa una instancia con una configuración de multi-threaded server.
2. Un usuario desde su estación de trabajo ejecuta una aplicación que puede ser de SQL\*Forms por ejemplo. Esta comunicación se establece a través del driver de algún protocolo y SQL\*Net.
3. En el servidor se está ejecutando de igual forma SQL\*Net para el driver apropiado. Se detecta la petición del proceso que desea conectarse y si está empleando versión 2 de SQL\*Net se establece la conexión a través de algún proceso dispatcher disponible.

En caso de que el usuario esté utilizando SQL\*Net versión 1 ó 1.1, SQL\*Net crea un proceso servidor dedicado para la atención del proceso del usuario y la operación será semejante en adelante a la descrita en el ejemplo anterior.

NOTA : La arquitectura multi-threaded está diseñada para trabajar con SQL\*Net versión 2.

4. El usuario ejecuta alguna instrucción de SQL. Por ejemplo modifica un registro de una tabla.
5. El proceso dispatcher coloca el requerimiento del proceso cliente en una cola que está en el SGA y es compartida para todos los procesos dispatchers.
6. Algún proceso servidor compartido libre, checa la cola común que registra los requerimientos a través de los procesos dispatchers y toma la siguiente instrucción de SQL en la cola. Se procesa dicha instrucción como se describió en los pasos 5,6 y 7 del ejemplo anterior.

7. Una vez que se finalizó con el proceso de la instrucción de SQL, el proceso pone el resultado en una cola de respuestas, de la cual el proceso dispatcher tomará el resultado correspondiente.

8. El proceso dispatcher checa la respuesta que se encuentra en la cola de procesos atendidos y le manda al usuario que originó la transacción el mensaje correspondiente.

De esta forma se puede apreciar de una manera simple la operación básica de Oracle con respecto a las transacciones que tiene que atender.

## **2. Estructura de la Base de Datos.**

En esta sección se discute la estructura de almacenamiento de información en una Base de Datos Oracle, a la cual se le hace referencia de manera lógica y física.

### **2.1 Estructura física**

Está determinada por los archivos a nivel Sistema Operativo que constituyen la Base de Datos. Cada Base de Datos Oracle está compuesta por tres tipos de archivos :

- Uno o varios archivos de datos.
- Dos o mas archivos de log (REDO LOG).
- Uno o mas archivos de control.

Los archivos de datos contienen la información real que se consulta y manipula por los procesos de los usuarios, en ellos se tienen las tablas, índices, diccionario de datos etc.

En los archivos de log se lleva una especie de bitácora de los movimientos realizados sobre los datos; se tiene almacenada la información necesaria de direcciones y status de dichos datos para poder realizar una recuperación en caso de falla o pérdida de los archivos de datos. En estos archivos de log se lleva la consistencia de la Base de Datos y son actualizados por el proceso de background LGWR cada vez que se ejecuta alguna consolidación de una transacción con la instrucción *commit work*; y el proceso que se sigue con estos archivos es el siguiente :

LGWR estará escribiendo en uno de los archivos de Log hasta que se haya llenado, debido a que son de tamaño fijo definido al momento de crear la Base de Datos. Una vez lleno este primer archivo se realiza un checkpoint y se pasa el control al segundo archivo de log para que se opere con él de igual forma. Cuando este segundo archivo se llenó, se realiza otra vez un checkpoint y se pasa el control al primer archivo de log; y el proceso continua hasta que la Base de Datos se da de baja.

Este trabajo de transferencia de control entre un archivo de log y otro sirve para que el proceso ARCH ejecute el respaldo en otro dispositivo (disco o cinta) de aquel archivo de log que no se esté utilizando en ese momento, de esta forma se puede apreciar que el LGWR por un lado está trabajando con un archivo de log, mientras que opcionalmente se puede tener al proceso ARCH respaldando el otro que se acaba de liberar.

En caso que no se tenga la Base de Datos en modo *archivelog* sencillamente el LGWR estará escribiendo de manera alternada en uno y otro archivo de log.

Por otra parte en el archivo de control se tiene información básica de la estructura de la Base de Datos y el número de secuencia del archivo de log que se está utilizando en el momento por el LGWR. La información que se tiene de la Base de Datos es la ubicación física de los archivos de datos y log, así como el status de la última operación con la Base de Datos, es decir si se dio de baja con shutdown normal o no, debido a que de esto dependerá si se realiza una recuperación al momento de darla de alta nuevamente.

Por ejemplo se puede apreciar en la siguiente figura una Base de Datos en la que se tienen 3 archivos de datos distribuidos en dos discos y con los tamaños especificados en mega-bytes.

Los archivos de log se muestran de igual forma asociados a esta Base de Datos. Es recomendable que los archivos de Log estén en algún disco distinto al que se utiliza para los archivos de datos, ya que se puede generar contención al acceder los discos por los procesos de background que trabajan de manera independiente a las transacciones de los usuarios.

El archivo de control es generalmente muy pequeño, pero en él se tiene la información general del estado de operación de la Base de Datos y si llegará a perderse, prácticamente se ha perdido la información de la Base de Datos; por tal motivo es recomendable generar mas de una copia en distintos discos, ya que este archivo se actualiza únicamente al momento de levantar y tirar la Base de Datos; así como cuando se altera la Base de Datos al añadir archivos o tablespaces, como se explicará en este mismo capítulo.

Este trabajo de transferencia de control entre un archivo de log y otro sirve para que el proceso ARCH ejecute el respaldo en otro dispositivo (disco o cinta) de aquel archivo de log que no se esté utilizando en ese momento, de esta forma se puede apreciar que el LGWR por un lado está trabajando con un archivo de log, mientras que opcionalmente se puede tener al proceso ARCH respaldando el otro que se acaba de liberar.

En caso que no se tenga la Base de Datos en modo *archivelog* sencillamente el LGWR estará escribiendo de manera alternada en uno y otro archivo de log.

Por otra parte en el archivo de control se tiene información básica de la estructura de la Base de Datos y el número de secuencia del archivo de log que se está utilizando en el momento por el LGWR. La información que se tiene de la Base de Datos es la ubicación física de los archivos de datos y log, así como el status de la última operación con la Base de Datos, es decir si se dio de baja con shutdown normal o no, debido a que de esto dependerá si se realiza una recuperación al momento de darla de alta nuevamente.

Por ejemplo se puede apreciar en la siguiente figura una Base de Datos en la que se tienen 3 archivos de datos distribuidos en dos discos y con los tamaños especificados en mega-bytes.

Los archivos de log se muestran de igual forma asociados a esta Base de Datos. Es recomendable que los archivos de Log estén en algún disco distinto al que se utiliza para los archivos de datos, ya que se puede generar contención al acceder los discos por los procesos de background que trabajan de manera independiente a las transacciones de los usuarios.

El archivo de control es generalmente muy pequeño, pero en él se tiene la información general del estado de operación de la Base de Datos y si llegará a perderse, prácticamente se ha perdido la información de la Base de Datos; por tal motivo es recomendable generar mas de una copia en distintos discos, ya que este archivo se actualiza únicamente al momento de levantar y tirar la Base de Datos; así como cuando se altera la Base de Datos al añadir archivos o tablespaces, como se explicará en este mismo capítulo.

Archivos de datos:



DISKSSISTEMAS:[ORACLE7.DB_DEMO]SYSTEM01.DBS	150 MB
DISKSSISTEMAS:[ORACLE7.DB_DEMO]SYSTEM02.DBS	50 MB
SYSSYSDEVICE:[DATOS.DB_DEMO]SYSTEM03.DBS	100 MB

Archivos de log:



DISK\$ARCHIVOS:[LOGS.DB_DEMO]ARCH1_DEMO.RDO	1MB
DISK\$ARCHIVOS:[LOGS.DB_DEMO]ARCH2_DEMO.RDO	1MB

Archivos de control :



DISKSSISTEMAS:[ORACLE7.DB_DEMO]CONTROL01.CTL	50K
DISK\$ARCHIVOS:[CONTROL.DB_DEMO]CONTROL02.CTL	50K
SYSSYSDEVICE:[CONTROL.DB_DEMO]CONTROL03.CTL	50K

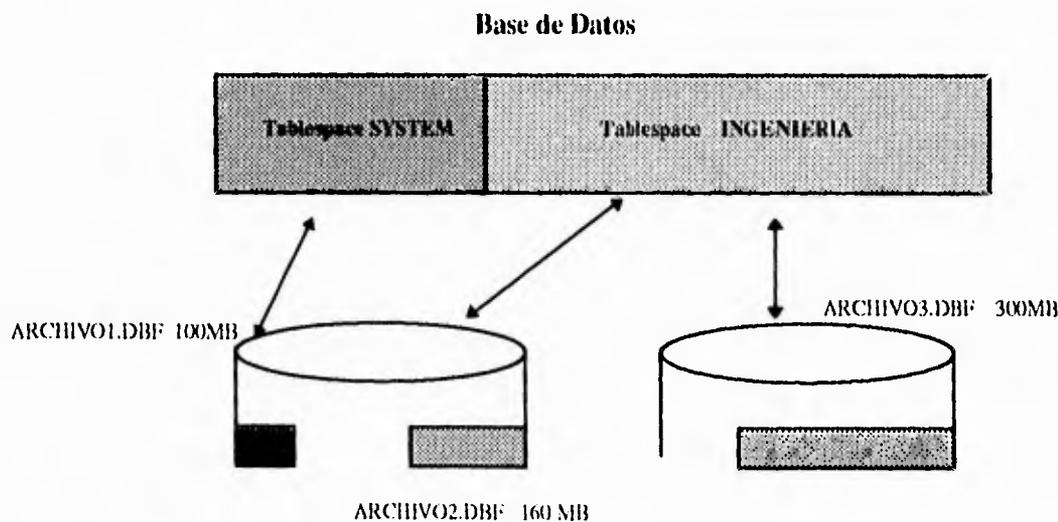
## 2.2 Estructura lógica

Está compuesta por :

- Uno o varios tablespaces
- Objetos que pertenecen a los usuarios y algunos pueden ser manipulados de acuerdo a los privilegios que se les otorgan. (ej. tablas, índices, vistas, clusters, secuencias, procedimientos almacenados, triggers).

Las estructuras lógicas que se manejan como tablespaces, segmentos, extents y bloques determinan la manera en que físicamente Oracle utiliza el espacio en disco. Los objetos y las relaciones existentes entre ellos están determinados por el análisis y diseño de la Base de Datos.

Como se había comentado, la Base de Datos está dividida en unidades lógicas llamadas tablespaces. Un tablespace es usado para agrupar estructuras que se relacionan entre sí; por ejemplo en un tablespace se pueden tener agrupados todos los objetos que pertenecen a la información administrativa de la Facultad de Ingeniería, como las tablas de alumnos, profesores, etc, y en otro tablespace se tiene la información del diccionario de datos exclusivamente, como se muestra en la figura :

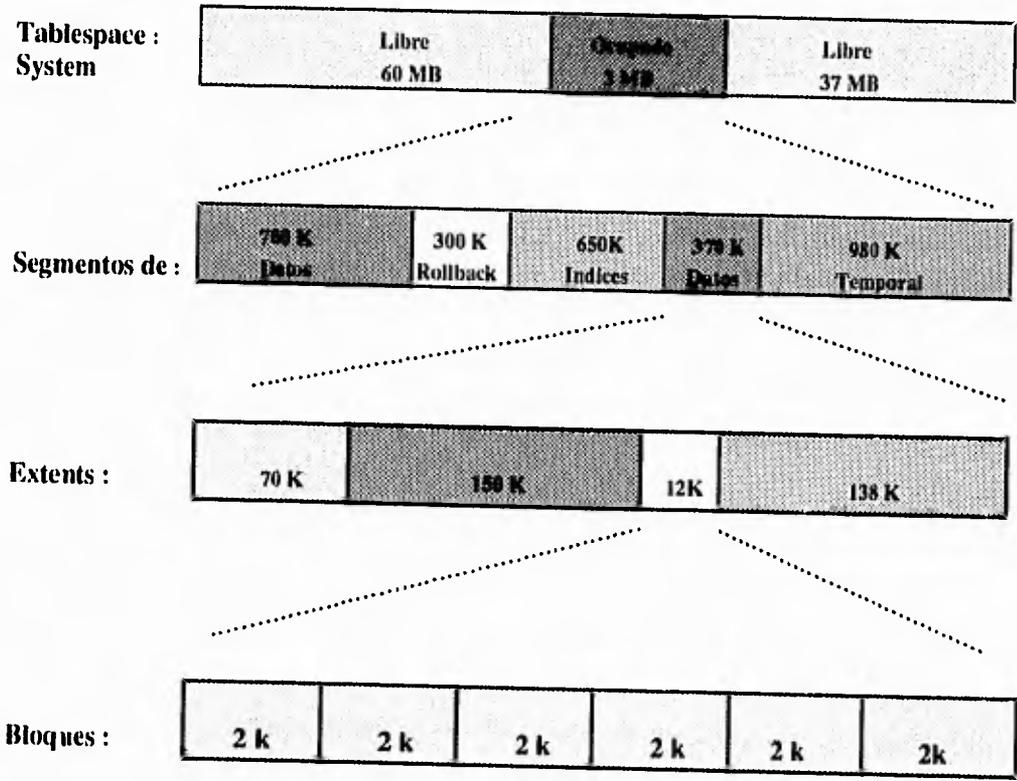


En resumen se pueden apreciar en esta y las siguientes ilustraciones la estructura que Oracle sigue internamente :

- Cada Base de Datos está dividida en uno o varios tablespaces.
  
- Se crean uno o varios archivos físicos para asociarlos a un tablespace, estos archivos pueden estar inclusive en distintos discos y hacer referencia al mismo tablespace. El hecho de distribuir los tablespaces en distintos discos permite que se tengan accesos simultáneos de lectura o escritura a datos o índices que se encuentren ubicados en otros dispositivos.
  
- El tamaño total de un tablespace está determinado por el espacio que ocupan archivos en disco. En este caso el tablespace system ocupa 100 MB y el tablespace ingeniería 460 MB.
  
- Internamente cada tablespace está dividido de manera lógica en segmentos de datos, índices, rollback y temporales, estos últimos son empleados cuando se ejecuta una instrucción con order by o group by. Este espacio es reutilizable por otros procesos una vez que se ha terminado la instrucción.
  
- Cada segmento está dividido en extents, que son un conjunto de bloques contiguos.
  
- Los bloques son definidos al momento de crear la Base de Datos; pero en la mayoría de los Sistemas Operativos un bloque es de 2048 bytes por default.

A continuación se presenta de una manera gráfica los puntos anteriores :

---



### **3. Integridad de los datos.**

Es muy importante garantizar que los datos permanezcan en la Base de Datos cumpliendo las reglas del negocio, integridad y consistencia. Oracle ha implementado una serie de técnicas para cumplir con estas necesidades en el manejo de la información.

Un *constraint* o restricción es una regla de negocio que se especifica al momento de crear la tabla y se asocia a las columnas. Esta restricción forma parte de la tabla y está registrada en el diccionario de datos, por lo que desde cualquier herramienta que se manipule la información de la Base de Datos, ésta permanecerá íntegra ya que se mandará el error correspondiente cuando se viole dicha regla. Si una regla de este tipo cambia, solamente se tendrá que realizar la modificación una vez y tendrá efecto para todas las aplicaciones, ya que éstas no se deberán modificar.

Las siguientes son las reglas que Oracle soporta para la integridad de las tablas :

- NOT NULL - no permite que la columna quede vacía, siempre se le deberá insertar un valor.
- UNIQUE - No se permitirán duplicados en la columna.
- PRIMARY KEY - Deshabilita los valores duplicados y nulos de una columna o conjunto de columnas. Será la llave primaria de la tabla.
- FOREIGN KEY - Garantiza que cada valor en esta columna o columnas existan previamente como valor único y llave primaria en otra tabla.
- CHECK - No permite la inserción de valores que no satisfagan con una expresión lógica definida.

Las reglas de negocio que no se pueden resolver con integridad de la entidad y referencial a través de las restricciones mencionadas se podrán resolver con procedimientos almacenados o triggers a nivel de la Base de Datos.

Los procedimientos y funciones almacenados son escritos con PL/SQL que es el lenguaje procedural que Oracle ha desarrollado como extensión a SQL. Las instrucciones que se manejan son las que en tercera generación se conocen como IF THEN ELSE, WHILE y LOOP. Los procedimientos se desarrollarán solamente una vez y quedan en formato compilado en la Base de Datos lo que hace más rápida la ejecución de las aplicaciones y el mantenimiento a las mismas, ya que se tienen centralizados los procedimientos y las aplicaciones solo los mandarían llamar pasando y recibiendo parámetros. Los procedimientos como todos los objetos en la Base de Datos serán referenciados por los usuarios siempre y cuando tengan los privilegios correspondientes para poder realizar las operaciones pertinentes.

Adicionalmente Oracle permite escribir procedimientos que se ejecutarán automáticamente antes y/o después de ejecutar instrucciones como insert, update o delete sobre alguna tabla. Estos procedimientos se llaman triggers.

Los triggers de la Base de Datos se ejecutarán sin que el usuario tenga que realizar ninguna instrucción adicional a las de DML (Data Manipulation Language "insert , update y delete" ). Estos son empleados de igual forma para garantizar la integridad de la información, sobre todo cuando se sabe de antemano que se deberán realizar siempre una serie de operaciones antes o después que se realice un insert, update o delete sobre una tabla.

#### **4. Seguridad de la Base de Datos**

Debido a que Oracle es un manejador de Bases de Datos multiusuario, incluye una serie de características y de control y acceso a la información. Los mecanismos de seguridad mecanismos se resumen en :

- Prevenir acceso a usuarios no autorizados.
- Prevenir el acceso a objetos no autorizados.
- Control de uso del espacio en disco.
- Control de los recursos del sistema, como uso de CPU por ejemplo.
- Auditoría de acciones y comandos de los usuarios.

La seguridad de la Base de Datos puede ser clasificada en dos diferentes categorías : seguridad del sistema y seguridad de la información.

La seguridad del sistema incluye mecanismos que controlan el acceso y uso de la Base de Datos en general y se realiza mediante :

- Username y password que determinan los privilegios por usuario.
- Asignación de cuota para utilización del espacio en disco.
- Limitación y control de los recursos consumidos.
- Auditoría de acceso a la información de la Base de Datos.
- Operaciones e instrucciones que cada usuario puede utilizar.

La seguridad en los datos se basa en mecanismos para el control de acceso y uso de los objetos de la Base de Datos :

- Verificación de las operaciones que se pueden realizar por cada usuario, sobre cada columna y tabla de la Base de Datos. Por ejemplo un usuario puede realizar inserciones y borrar registros, pero no podrá modificar información de alguna tabla en particular.

Oracle7 propone el concepto de administración basada en roles, como un mecanismo de control y administración de la Base de Datos.

Cada Base de Datos tiene una lista de usuarios dados de alta que son los que podrán conectarse y realizar operaciones sobre la información, para realizar esto el administrador de la Base de Datos deberá definir los privilegios con los que cada usuario actuará. Se puede delimitar el espacio en disco y por tablespace que cada usuario podrá utilizar, el tablespace que por default se asignará para crear sus objetos y el tablespace que se utilizará para realizar el proceso temporal del usuario al ejecutar una instrucción con order by o group by por ejemplo.

### **5. Consideraciones Adicionales**

Los siguientes son los tipos de datos que se pueden utilizar para la definición de las columnas de las tablas en una Base de Datos en Oracle :

**CHAR (n)** - almacena una cadena de longitud fija de hasta 255 caracteres. El valor 'n' determina el tamaño máximo de la cadena; por ejemplo si se define una columna como char(50) se podrán almacenar 50 caracteres en dicha columna, en caso que no se inserten los 50 caracteres se llenará con espacios en blanco hasta cubrir la longitud definida. Esta consideración es importante tenerla en cuenta ya que en Oracle 6 no se completaba con espacios en blanco lo que restaba de la cadena.

**VARCHAR2(n)** - almacena una cadena de caracteres variable de hasta 2000 caracteres. El valor 'n' determina el tamaño máximo de la cadena y no completará con espacios en blanco lo que reste en caso de que no se llene hasta el valor de 'n'; por ejemplo si se definió una columna varchar2(50) y se insertan 20 caracteres, solamente se tendrán almacenados en la Base de Datos 20 caracteres.

**VARCHAR(n)** - actualmente es sinónimo de varchar2, pero en futuras versiones Oracle utilizará este tipo de dato para columnas de longitud variable y distinta comparación semántica. Lo recomendable es utilizar varchar2 o char según sea la necesidad, ya que este tipo de dato está reservado para uso posterior.

**NUMBER** - almacena valores numéricos fijos y de punto flotante hasta con 38 dígitos de precisión.

**DATE** - almacena valores que hacen referencia a algún valor en el tiempo; se almacena el año, el mes, día, hora, minutos y segundos.

**LONG** - las columnas definidas como tipo long son utilizadas para almacenar texto en la Base de Datos. Solo se permite una columna tipo long por tabla y se pueden almacenar hasta 2GBytes de información en este tipo de columna.

**RAW y LONG RAW** - estos tipos de datos son utilizados para almacenar datos que no serán interpretados por Oracle. Son ideales para almacenar datos binarios; por ejemplo en una columna long raw se pueden almacenar imágenes, gráficas, sonido, documentos y su interpretación depende del uso que se le vaya a dar con alguna interfase que lo interprete; raw es equivalente a varchar2 y long raw a long, pero la diferencia radica en que SQL\*Net y las utilerías de export/import de Oracle no interpretarán ni realizarán ninguna conversión de estos datos.

Otra característica importante a considerar es el cálculo del espacio en disco que se estima tendrán los principales objetos en la Base de Datos, esto es tablas e índices. Una vez calculados los espacios se pueden utilizar las instrucciones de initial, next, pctincrease, etc, para la creación de estos objetos.

A continuación se describe este procedimiento propuesto por Oracle para el cálculo de espacios que ocuparán inicialmente las tablas y los índices que se vayan a crear dependiendo del número de registros y columnas de la tablas.

Al utilizar este procedimiento se obtendrán una serie de valores estimados para el espacio inicial que se requerirá, por lo que se debe tener en cuenta que los valores nulos no representan espacio en la Base de Datos y el valor "n" que se indica en una columna varchar2(n), no necesariamente representa los bytes que ocupará el campo en la Base de Datos, ya que si se insertan un número menor a "n" de caracteres, el resto no se complementa con espacios en blanco, como es el caso de las columnas que se definen como char(n).

Una vez calculado el espacio inicial estimado que tendrá la tabla, es recomendable añadirle un 10% a ese valor para poder trabajar de manera olgada con dicha tabla.

Los pasos para realizar esta estimación son :

- Calcular el tamaño del encabezado de los bloques.
- Calcular el espacio para datos por bloque.
- Calcular el tamaño combinado promedio de las columnas de la tabla.
- Calcular el tamaño promedio de un registro en la tabla.
- Calcular el promedio de registros que se pueden almacenar por bloque.
- Calcular el número de bloques y bytes requeridos para la tabla.

1. Calcular el tamaño del encabezado del bloque.

Encabezado parte **A** = Encabezado fijo + Encabezado variable

Donde:

Encabezado fijo = 57 bytes

Encabezado variable =  $23 * I$

**I** es el valor del INITRANS para la tabla. Por default es 1.

INITRANS es el número de transacciones que pueden hacer referencia al mismo tiempo a un bloque de la Base de Datos. El valor default de 1 es adecuado para operaciones comunes; este valor toma un valor mayor cuando se sabe de antemano que será un ambiente transaccional muy pesado y es muy probable que mas de una transacción de usuario necesite hacer referencia a bloques específicos de la Base de Datos.

Encabezado parte **B** =  $4 + 2 * R$

**R** es el número de registros en un bloque. (Será calculado en el paso 5).

El tamaño del encabezado está dividido en 2 partes, porque en otros puntos de este procedimiento se requerirá hacer referencia a la parte A ó B.

$$\text{Encabezado} = \text{Parte A} + \text{Parte B}$$

Por lo tanto si se toma el default de INITTRANS=1 se tendría que :

$$\text{Encabezado} = (57 + 23) + (4 + 2R) \text{ bytes}$$

$$\text{Encabezado} = 80 + (4 + 2R) \text{ bytes}$$

## 2. Cálculo del espacio para datos por bloque.

$$\text{Espacio datos} = (\text{tamaño\_bloque} - \text{encabezado}) - ((\text{tamaño bloque} - \text{encabezado parte A}) * (\text{PCTFREE} / 100))$$

El tamaño del bloque en una Base de Datos se indica al momento de creación con el parámetro db\_block\_size en el archivo INIT.ORA de esa instancia, y se puede consultar desde SQLDBA :

```
sqldba> connect internal;  
sqldba> show parameters db_block_size;
```

Para mayor información sobre los comandos de SQLDBA se puede consultar el manual "Oracle7 Server Utilities User's Guide".

PCTFREE es el porcentaje que quedará libre para posibles modificaciones de los datos que están en un bloque; en caso de que el PCTFREE sea 0, cuando el valor de una columna cambie y el espacio que ocupaba sea mayor, se encadenará un bloque para poder almacenar el dato completo.

Para este ejemplo, supongamos que se tiene un PCTFREE de 10 y el tamaño del bloque de la Base de Datos es de 2k, como en la mayoría de los Sistemas Operativos :

$$\text{Espacio datos} = (2048 - (80 + (4 + 2R)) - ((2048 - 80) * (10/100)) \text{ bytes}$$

$$\text{Espacio datos} = (1964 - 2R) - (1968 * 0.1) \text{ bytes}$$

$$\text{Espacio datos} = (1768 - 2R) \text{ bytes}$$

### 3. Calcular el espacio combinado por columnas

Para este punto se deben considerar los siguientes puntos :

- Número de columnas en la tabla.
- Tipo de dato por columna.
- Espacio promedio por columna.

Para calcular el espacio promedio nos podemos auxiliar de SQL\*Plus para los casos en los que la tabla ya ha sido creada y contiene datos.

```
SELECT AVG(NVL(VSIZE(col1), 0)) +
       AVG(NVL(VSIZE(col2), 0)) +
       ..... +
       AVG(NVL(VSIZE(coln), 0)) "Espacio Promedio del Registro"
FROM   tabla
```

Donde col1, col2, . . . , coln son los nombres de las columnas de la tabla, y TABLA es el nombre de la tabla que se desea consultar.

Por ejemplo si se tiene una tabla llamada STATUS y fue creada con la siguiente estructura:

```
CREATE TABLE status (  
  A      CHAR(10),  
  B      DATE,  
  C      NUMBER(8,2) );
```

Para determinar el espacio promedio requerido por renglón de esta tabla STATUS, se realizaría la siguiente consulta:

```
SELECT AVG(NVL(VSIZE(A),0) +  
          AVG(NVL(VSIZE(B),0) +  
          AVG(NVL(VSIZE(C),0)  " Promedio del Registro"  
FROM   STATUS;
```

Alternativamente, en caso de que no se tenga creada y con datos la tabla, se puede estimar este valor. Esto se puede hacer examinando cada columna y su tipo de dato. Para columnas con tipo de dato fijo, el valor de la columna será fijo. Para el caso de columnas de longitud variable se debe determinar el valor promedio de esta.

Por ejemplo, suponiendo que se quiere calcular el espacio promedio del registro de una tabla, se debe calcular el espacio promedio de cada columna. Para el caso de la tabla que se indica en este ejemplo, STATUS, el tamaño de las dos primera columnas es fijo, la primera es de 10 bytes y la segunda de 7 bytes. La tercer columna es de tipo numérica y se tiene una precisión de 8 dígitos. El cálculo promedio de un valor numérico es:

$$\text{columna\_num} = (n/2 + 1) \text{ bytes}$$

---

por lo que para este ejemplo se tendría :

$$\text{columna\_num} = (8/2 + 1) \text{ bytes}$$

$$\text{columna\_num} = 5 \text{ bytes}$$

Por lo tanto el espacio combinado por columna en un registro, identificado a partir de este punto como **D** es :

$$D = \text{col1} + \text{col2} + \text{col3}$$

$$D = ( 10 + 7 + 5 ) \text{ bytes}$$

$$D = 22 \text{ bytes}$$

#### 4. Cálculo del mínimo espacio requerido por registro en una tabla.

$$\text{Registro promedio} = \text{Encabezado registro} + F + V + D$$

Donde :

Encabezado registro = 3 bytes por registro en la tabla.

F = Total de columnas que almacenan 250 bytes o menos. Cada columna de este tipo representa 1 byte.

V = Cálculo que del total de columnas que almacenan mas de 250 bytes multiplicado por 3 bytes que representa cada columna de este tipo.

D = El espacio combinado por columnas en un registro. (Cálculo del paso 3 ).

Para el ejemplo que se está desarrollando:

$$\text{Registro Promedio} = (3 + (1*3) + (3*0) + 22) \text{ bytes}$$

$$\text{Registro Promedio} = 28 \text{ bytes}$$

5. Cálculo del número promedio de registros por bloque.

R identifica registros/bloque

$$R = \text{Espacio datos} / \text{Registro promedio}$$

Donde:

Espacio datos = Resultado que se obtuvo en el paso 2.

Registro promedio = Resultado que se obtuvo en el paso 4.

$$R[\text{registros/bloque}] = (1768 [\text{bytes}] - 2R [\text{bytes}] + [\text{registros/bloque}]) / 28 [\text{bytes}]$$

$$28R [\text{bytes}] * [\text{registros/bloque}] = 1768 - 2R [\text{bytes}] * [\text{registros/bloque}]$$

$$30R [\text{bytes}] * [\text{registros/bloque}] = 1768 [\text{bytes}]$$

$$R = 58 \text{ registros/bloque}$$

Esta R es la variable a la que se ha estado haciendo referencia desde el punto 1 y que se indicó que se calcularía en este paso del procedimiento.

6. Calcular el número de bloques y bytes.

Bloques por tabla = #registros / R

Suponiendo que se tienen 1000 registros en la tabla que se utiliza para este ejemplo, se tendría :

Bloques por tabla = 1000 / 58

Bloques por tabla = 173 Bloques

Como cada bloque de la Base de Datos es de 2048 bytes, ya se puede calcular el espacio aproximado en bytes que ocupará la tabla :

Tamaño inicial de la tabla = 173 bloques \* 2048 bytes/bloque

Tamaño inicial de la tabla = 354304 bytes = 346k

Este procedimiento provee un razonable estimado del tamaño de una tabla, no es un número exacto de bloques y bytes; pero es el valor que se puede utilizar en el parámetro INITIAL de la cláusula STORAGE al momento de crear la tabla. Es decir, en con el comando de creación de la tabla se indica el valor obtenido en el punto 6, de la siguiente manera:

```
CREATE TABLE STATUS (
  A CHAR(10),
  B DATE,
  C NUMBER(8,2) )
PCTFREE 10
PCTUSED 40
TABLESPACE DATOS
STORAGE ( INITIAL 346 K
          NEXT 346 K
          MAXEXTENTS 10
          PCTINCREASE 0 );
```

De igual forma existe un procedimiento para realizar el cálculo de espacio para los índices que se crearán sobre algunas columnas de la tabla. A continuación se describe este procedimiento, de igual forma que para el caso anterior se ejecutan una serie de pasos:

- Calcular el tamaño del encabezado de los bloques.
- Calcular el espacio para datos de cada bloque.
- Calcular la longitud de las columnas.
- Calcular el tamaño de la llave.
- Calcular el número de bloques y bytes.

#### 1. Cálculo del tamaño del encabezado de los bloques.

Encabezado = (Parte fija) + (Parte variable)

Donde:

Parte fija = 113 bytes

Parte variable =  $23 * I$        $I$  es el valor de INTRANS.

Para este ejemplo ahora asumamos que el valor del INTRANS no es default de 1, sino que es 2.

Encabezado =  $(113) + (2 * 23) = 159$  bytes

2. Cálculo del espacio para datos de cada bloque.

$$\text{Espacio datos} = (\text{Tamaño bloque} - \text{Encabezado}) * (1 - \text{PCTFREE}/100)$$

Supongamos que el tamaño del bloque es 2k y un PCTFREE de 10.

$$\text{Espacio datos} = (2048 - 159) * (1 - 10/100)$$

$$\text{Espacio datos} = 1889 * 0.9 = 1700 \text{ bytes}$$

3. Calcular el espacio combinado por columnas.

Para este caso se deberá saber el promedio en bytes que utiliza cada columna que formará el índice de la tabla. Para lo cual se puede utilizar el mismo esquema que se explicó en paso 3 del cálculo de espacios por columnas para las tablas.

Supongamos que se tienen dos columnas definidas como varchar2(150) y otra varchar2(6), que formarán el índice y en promedio la primer columna almacena 18 caracteres y la segunda 2.

Por lo tanto el espacio promedio, conocido como D es :

$$D = \text{col1} + \text{col2}$$

$$D = 18 + 2 = 20 \text{ bytes.}$$

4. Cálculo del tamaño de la llave.

Tamaño llave = Acceso al encabezado + F + V + D + ROWID

Donde :

Acceso al encabezado = 2 bytes fijos.

F = Total de columnas que almacenan 128 bytes o menos. Cada columna de este tipo representa 1 byte.

V = Cálculo el total de columnas que almacenan mas de 128 bytes multiplicado por 2 bytes que representa cada columna de este tipo.

D = El espacio combinado de las columnas indexadas. (Cálculo del punto 3).

ROWID = Es el identificador del registro y a través del cual se accedera a él cuando se utilice el índice. Se tiene el número del archivo, número del bloque y el número del registro. Esto representa 6 bytes.

Por lo tanto para este ejemplo tendríamos :

Tamaño llave =  $2 + | ( 2 * 1 ) + ( 1 * 2 ) | + 20 + 6$

Tamaño llave = 32 bytes

5. Cálculo del número de bloques y bytes.

Bloques para el índice =  $1.05 * (\#registros / Indices\ por\ bloque)$

Indices por bloque =  $(Espacio\ datos / Tamaño\ llave)$

Por lo tanto para este ejemplo en el que suponemos una tabla de 10000 registros y con respecto a los cálculos obtenidos en los pasos anteriores tenemos :

Indices por bloque =  $(1700 / 32) = 53$

Bloques para el índice =  $1.05 * (10000 / 53) = 199\ bloques$

Suponiendo que el tamaño del bloque de la Base de Datos es de 2048 bytes, se puede determinar en bytes el espacio que ocupará inicialmente el índice sobre las columnas que se han analizado :

Espacio del índice =  $199\ bloques * 2048\ bytes/bloque$

Espacio del índice = 400k

Por lo tanto al momento de crear el índice se indicará este valor al INITIAL del STORAGE en el comando CREATE INDEX como se indica :

```
CREATE INDEX idx_cols
ON status
STORAGE ( INITIAL 400K NEXT 400K )
TABLESPACE INDICES;
```

En caso que se tenga ya información en una Base de Datos, es conveniente utilizar las utilerías de Oracle EXPORT e IMPORT para corregir porosidad y fragmentación.

Realizando estas actividades adicionalmente se incrementará el rendimiento ya que la información quedará debidamente ordenada y el acceso a ésta será más eficiente por parte del manejador de la Base de Datos.

## **PROCEDIMIENTO DE MIGRACION**

En esta sección se describen los pasos y métodos que Oracle propone para realizar la migración de una Base de Datos y aplicaciones típicas que estén operando bajo Oracle versión 6. Se mencionan los procedimientos que implican la migración así como las ventajas y puntos a considerar en cada paso del proceso.

La información aquí descrita no es toda la que existe para realizar este trabajo; se recomienda que se involucre a los analistas de los sistemas y personas que conozcan del sistema operativo VMS y de la administración de Oracle, por lo que se pueden complementar los conceptos aquí tratados con los manuales de utilerías y administración que provee tanto Digital como Oracle.

La versión 6 de Oracle es compatible con la funcionalidad de Oracle7, sin embargo el proceso de migración consiste de una serie de procedimientos que transfieren una Base de Datos de Oracle versión 6 para que trabaje bajo la misma funcionalidad; y opcionalmente utilice las nuevas características que ofrece Oracle7.

Se tratan a continuación todos los aspectos referentes a la migración de la Base de Datos y posteriormente se comentarán los puntos mas relevantes para realizar la migración de las aplicaciones que existían con la versión 6 de Oracle.

Es importante hacer énfasis desde este punto que la versión 2.3 de SQL\*Forms de Oracle ya no es soportada ni distribuida con la versión 7 del RDBMS; por lo que habrá que migrar las formas que se pudieran tener con esta versión y accediendo Oracle6, pues de lo contrario es una aplicación que no se podrá utilizar nuevamente una vez concluido el trabajo que siguiendo los pasos descritos en este capítulo se proponen.

Los pasos para la migración de estas aplicaciones también serán tratados al final de este capítulo, una vez migrada la Base de Datos.

## MIGRACION DE LA BASE DE DATOS

### Puntos a considerar antes de iniciar la migración :

Antes de planear la migración de la Base de Datos, es conveniente que se tengan en cuenta de manera general los requerimientos y procedimientos que se utilizan. La siguiente lista es la presentación de este proceso de migración.

1. Familiarizarse con los conceptos y características que se manejan en Oracle7.

Varchar2

Comparación semántica de strings

Restricciones e Integridad

2. Elegir un método de migración : Import/Export, emplear la utilería de migración que provee Oracle o copia de la información a través de SQL\*Net.

3. Desarrollar un plan de pruebas antes de migrar a Oracle7. Ejecutar dichas pruebas con la Base de Datos en versión 6 y tener los resultados para compararlos con los que se obtendrán una vez instalado Oracle7.

4. Sí se emplea la utilería de migración, se debe crear una copia de la Base de Datos y migrar dicha Base de Datos de prueba.

5. Verificar las aplicaciones con Oracle7 y estar seguros que éstas funcionan de manera adecuada antes de migrar la Base de Datos de Producción.

6. Dar de baja la Base de Datos desde la utilería de administración de Oracle, **SQL\*DBA**, con el comando **SHUTDOWN NORMAL** para garantizar que no existen transacciones pendientes por aplicarse en la Base de Datos.
  
7. Realizar un respaldo de la Base de Datos con comandos del Sistema Operativo como puede ser **copy** o **backup**. Es importante señalar que se deberán respaldar todos los archivos que la componen; esto es, archivos de datos, de control, de log, de parámetros como **init.ora** y aquellos que sean para crear objetos locales con SQL.
  
8. Borrar o modificar aquellos parámetros obsoletos del **INIT.ORA**.
  
9. Modificar los scripts de SQL que se tengan para la creación de objetos en la Base de Datos.
  
10. Convertir la Base de Datos a Oracle7, empleando el procedimiento de migración o **Export/Import**. Ambos métodos presuponen la instalación ya realizada del software de Oracle7.
  
11. Realizar un nuevo respaldo de los archivos de datos, de control y log una vez que se haya habilitado exitosamente la Base de Datos con Oracle7.
  
12. Una vez realizada exitosamente la migración, verificar que las aplicaciones funcionen adecuadamente.
  
13. Probar las aplicaciones que se ejecutaron en el paso 3 y comparar los resultados; que deberán ser los mismos a los que se obtuvieron cuando se utilizó la versión 6.
  
14. Realizar los ajustes necesarios para obtener un mejor rendimiento durante la ejecución de las aplicaciones.
  
15. Determinar qué nuevas características de Oracle7 pueden ser apropiadas para implementarlas.

16. Desarrollar un nuevo procedimiento de administración de la Base de Datos que contemple algunas de las nuevas características de Oracle7.

Una vez concluidos estos pasos la Base de Datos debe estar completamente convertida y los usuarios podrán acceder a la información en Oracle7.

Típicamente el administrador de la Base de Datos es el responsable de que los procedimientos de migración se ejecuten de manera exitosa. Es decir, debe realizar los respaldos necesarios de la Base de Datos, determinar acciones de emergencia, planear las fechas de realización de las actividades, etc.

Mientras el administrador es el responsable de la migración de la información, los desarrolladores son los responsables de garantizar que los sistemas diseñados con la versión 6 funcionen de manera adecuada con Oracle7. Por ello, se deberá instalar Oracle7 para que tanto el Administrador como los desarrolladores puedan probar la funcionalidad de la aplicaciones y hacer las adecuaciones necesarias para que al momento de realizar la transferencia final con la Base de Datos de Producción no se presenten problemas.

6. Dar de baja la Base de Datos desde la utilería de administración de Oracle, **SQL\*DBA**, con el comando **SHUTDOWN NORMAL** para garantizar que no existen transacciones pendientes por aplicarse en la Base de Datos.
  
7. Realizar un respaldo de la Base de Datos con comandos del Sistema Operativo como puede ser **copy** o **backup**. Es importante señalar que se deberán respaldar todos los archivos que la componen; esto es, archivos de datos, de control, de log, de parámetros como **init.ora** y aquellos que sean para crear objetos locales con SQL.
  
8. Borrar o modificar aquellos parámetros obsoletos del **INIT.ORA**.
  
9. Modificar los scripts de SQL que se tengan para la creación de objetos en la Base de Datos.
  
10. Convertir la Base de Datos a Oracle7, empleando el procedimiento de migración o **Export/Import**. Ambos métodos presuponen la instalación ya realizada del software de Oracle7.
  
11. Realizar un nuevo respaldo de los archivos de datos, de control y log una vez que se haya habilitado exitosamente la Base de Datos con Oracle7.
  
12. Una vez realizada exitosamente la migración, verificar que las aplicaciones funcionen adecuadamente.
  
13. Probar las aplicaciones que se ejecutaron en el paso 3 y comparar los resultados; que deberán ser los mismos a los que se obtuvieron cuando se utilizó la versión 6.
  
14. Realizar los ajustes necesarios para obtener un mejor rendimiento durante la ejecución de las aplicaciones.
  
15. Determinar qué nuevas características de Oracle7 pueden ser apropiadas para implementarlas.

**MÉTODOS DE MIGRACIÓN PARA LA BASE DE DATOS :**

Como se ha mencionado existen tres propuestas para realizar la migración de la Base de Datos, en esta parte se mencionan los procedimientos y pasos a seguir para cada método; los cuales son :

- Copia de información a través de SQL\*Net
- Emplear la utilería de migración que Oracle provee con el software.
- Emplear las utilerías de export e import de Oracle.

Cada método tiene ciertas consideraciones como ventajas o desventajas, así como pre-requisitos de productos instalados, configuraciones o espacio en disco disponible que se deberán tomar en cuenta para cada caso. Es por estas razones que se asume que las personas que vayan a realizar esta tarea tienen al menos conocimientos básicos de VMS y Oracle; así como de las aplicaciones y ambiente local de la Base de Datos en cuestión, de otra forma si se presentan algunos problemas no se sabrá donde poder consultar cuestiones particulares.

### **I. Copia de Información a través de SQL\*Net**

Mediante SQL\*Net se pueden seleccionar los datos que se migrarán, la Base de Datos anterior se mantiene intacta y de esta manera se puede tener un medio ambiente ideal para realizar pruebas.

La desventaja existente es que se requiere trabajar con SQL\*Net y declarar Database Link's, no se realiza una conversión de datos automáticamente, es decir las columnas que son de tipo char seguirán siendo de tipo char y no varchar2 como sí se contempla en los otros casos; para este caso lo recomendable es que se realice la creación de las tablas en el ambiente de Oracle7 con la definición de las columnas que eran char en Oracle6 como varchar2 ahora. Además se requiere de espacio suficiente para mantener al menos las dos Bases de Datos.

También se debe tomar en cuenta que el hecho de trabajar con SQL\*Net, requiere de la configuración e instalación de este producto para el protocolo que se pretenda utilizar. Este método por lo tanto está orientado mas que nada para aquellos ambientes en los que se trabaja de manera distribuida y la información que se requiere migrar está en un equipo de cómputo distinto al que tiene instalado Oracle7.

El punto anterior no impide que se pueda realizar esta migración con una sola máquina y emplear SQL\*Net conectándose al mismo nodo, pero realmente esto no resulta práctico.

Las ventajas se resumen en que se pueden seleccionar solo los datos que se desean migrar, la Base de Datos anterior permanece intacta por lo que es ideal para realizar un ambiente de pruebas. Se debe tomar en cuenta por lo tanto el consumo del tiempo empleado ya que depende de la velocidad de transmisión y el canal de comunicación que se utilice en la red, se requiere tener instalado SQL\*Net para algún protocolo particular, no se realiza la conversión de datos automáticamente y se requiere de suficiente espacio en disco para mantener la Base de Datos anterior y la que se ha migrado; por otra parte se requiere que se conozca la forma de configurar SQL\*Net para el protocolo empleado y como definir los Database Link's que Oracle necesita para realizar las operaciones con SQL\*Net.

La manera de realizar esta migración es copiando los datos de un nodo a otro con los comandos de SQL create y copy empleado database link's, que en este ejemplo se llaman v6\_bd y o7\_bd y son definidos dependiendo del protocolo que se vaya a emplear con la sintaxis que Oracle especifica para cada protocolo en el manual "ORACLE SQL\*Net User's Guide".

```
SQL> Create table alumnos as select * from abd.alumnos@v6_dbd;
```

ó

```
SQL> Copy from abd/bdfiunam@v6_bd to abd/bdfiunam@o7_bd  
create table asignaturas using select * from asignaturas;
```

## **II. Emplear la utilería de migración**

Con el medio de distribución del software de Oracle7 y sus herramientas que apoyan a la manipulación y acceso a la Base de Datos, se incluye el procedimiento de instalación de este software. Dicho procedimiento cuenta con una serie de opciones que cada una es descrita en el manual "Oracle7 for VAX Open VMS Installation Guide"; y para esta sección puede utilizarse como complemento ya que se explicarán las opciones que se deben seleccionar para realizar la migración.

Esto quiere decir que esta utilería es proporcionada por Oracle y para que los resultados sean exitosos se deberán seguir al pie de la letra las recomendaciones y acciones a seguir en cada punto explicado durante esta sección.

Como se comentó anteriormente, se asume que el personal que va a realizar estas tareas cuenta con conocimientos de VMS, Oracle y de las aplicaciones y configuración local de la Base de Datos.

Se numerarán los puntos a seguir para poder hacer referencia a ellos en algún otro paso de este proceso de migración.

En este caso se trabajará con la Base de Datos que se encuentra activa con la versión 6 y la misma se dejará bajo el formato y configuración de Oracle 7, por lo que no se requerirá de espacio adicional para mantener ambas Bases de datos, pero por esta misma razón se deberán realizar los respaldos recomendados durante este trabajo.

### **1. Preparar el ambiente de Oracle7 que recibirá la Base de Datos que se migrará :**

Instalar y configurar el software de Oracle7, pero sin crear alguna Base de Datos. La utilería de migración convertirá los archivos de datos de la versión 6 a formato de Oracle7.

Es recomendable crear un usuario a nivel Sistema Operativo para mantener la instalación de Oracle7, como lo indica el manual "Oracle7 for VAX OpenVMS Installation Guide", dicho usuario y el que contiene la instalación y la Base de Datos de la versión 6 NO deberán pertenecer al grupo de SYSTEM, en el UAF; debido a que genera problemas al momento de migrar la Base de Datos.

Otro aspecto a considerar en este punto es que el Diccionario de Datos de Oracle7 ocupa aproximadamente un 50% más que el de Oracle6, por lo que se deberá alterar el tablespace SYSTEM en caso de que no haya al menos 3 MB libres.

Es importante que se conozcan y se tengan anotados en algún papel y accesibles a la mano durante el proceso de migración los siguientes nombres lógicos y variables. :

(\*) Nombre de la Base de Datos en Oracle6

(\*) ORA\_SID para Oracle6

(\*) ORA\_ROOT para Oracle6

(\*) ORA\_DB para Oracle6

(\*) ORA\_RDBMS en Oracle6

(\*) ORA\_RDBMS en Oracle7

(\*) ORA\_DB en Oracle7

(\*) ORA\_ROOT para Oracle7

2. Ejecutar el procedimiento de comandos ORA\_ROOT:[UTIL]ORAUSER.COM de la instalación de Oracle7.

3. Verificar que los valores de los símbolos ORA\_RDBMS y ORA\_ROOT estén definidos de acuerdo a la configuración de Oracle7.

```
$ show logical ORA_RDBMS
```

```
$ show logical ORA_ROOT
```

Crear el directorio para la administración de la BD en Oracle7.

4. Usar el procedimiento ORACLEINS para crear el directorio de administración de la Base de Datos.

Aparecerá el menú principal de instalación, como se muestra en la siguiente figura:

```
$ set default ORA_ROOT:[INSTALL]
```

```
$ oracleins
```

```
ORACLE Installation Startup Menu

Options :

1. Create a new ORACLE system
2. Upgrade your system from an ORACLE distribution tape
3. Reconfigure existing products, manage the database,
   or load the demo tables
4. Exit

Before attempting to upgrade, reconfigure, manage the
database, or load demo tables, please run
ORA_UTIL:ORAUSER.COM or, if you created an instance,
ORA_DB:ORAUSER-<databasename>.COM

Choose an option please:
```

Seleccionar la opción 3 "Reconfigure existing products, manage the database, or load demo tables".

Cuando se pida especificar el directorio de Oracle7, se mostrará el valor que tiene ORA\_ROOT; si este valor corresponde a Oracle6 o alguna otra ruta, tendrá que salir de este procedimiento y regresar al punto 2.

Cuando se pida el directorio o dispositivo en el cual se encuentran localizados los save sets, presione [ENTER] debido a que no se cargarán nuevos productos :

Reconfigure, Manage Database, and/or Load Demo Tables

ORACLE Installation Version 1.0.12.5 - Production on 26-OCT-1994 16:46:18.90

Copyright (c) 1992, Oracle Corporation, California, USA. All rights reserved.

Root directory? ( DISKSSISTEMAS:[ORACLE7] )  
The root directory will be DISKSSISTEMAS:[ORACLE7].

If you are loading products from save sets, enter the drive/directory where the save set are located (e.g. MUA0: or DISKSA:[ORACLE.SAVE\_SETS]). If you are loading from a remote device, do not include a username and password (you will be prompted instead).

If not loading save set, press [RETURN].

Save set location or [RETURN] :

Ahora aparecerá el menú principal llamado "Oracle Product Installation and Upgrade" como se muestra :

```

Main Menu

Oracle Product Installation and Upgrade

1. Software Installation and Upgrade Menu.

2. Instance Creation, Startup, and Shutdown Menu

3. Build or Upgrade Database Tables Menu

Enter a number or (E)XIT to exit installation procedure :
```

Seleccionar la opción 2 "Instance, Creation, Startup, and Shutdown Menu." Al entrar con esta opción, seleccione ahora la opción 1, "Create New Instance and Database".

Cuando se pida el SID (Identificador de la instancia), es conveniente que se indique el mismo nombre que se tiene en el símbolo ORA\_SID de la instalación de Oracle6. Ahora se solicitará el nombre de la Base de Datos, el cual deberá ser el nombre de la Base de Datos existente en versión 6 y que será migrada a Oracle7.

```
Currently known database SID's
<NONE>
Press [RETURN] to quit with no action
NOTE: The SID can be a maximum of 6 characters in length.
What is the SID for the instance you want to create ? DEMO
NOTE: The database name can be a maximum of 8 characters in length
What is the name of the database you want to create ? DEMO
```

Ahora aparecerá el menú de configuración de la Base de Datos, con una serie de valores predefinido, los cuales no se modificarán debido a que no se creará una Base de Datos, esto realmente servirá para que se cree un directorio debajo de ORA\_ROOT llamado ORA\_ROOT:[DB\_<db\_name>] ( en este ejemplo es ORA\_ROOT:[DB\_DEMO] ). Por lo tanto se deberá teclear E (Exit) para aceptar los valores propuestos.

Aparecerá un mensaje que indica que se ha creado un directorio para la Base de Datos que hemos indicado (DEMO en este caso). En este directorio se han creado también una serie de procedimientos de comandos y archivos de configuración de Oracle.

En cuanto se nos pregunte si deseamos continuar con el procedimiento de creación de la Base de Datos, se deberá responder con N indicando que no.

Una vez realizadas estas actividades se deberá salir del procedimiento ORACLEINS.

El nuevo directorio que se ha creado ( ORA\_ROOT:[db\_demo] ), contiene una serie de archivos que serán de utilidad para el procedimiento de migración posterior. Estos archivos son :

---

STARTUP_PARALLEL_<sid>.COM	startup_parallel_demo.com
STARTUP_PARALLEL_<sid>.SQL	startup_parallel_demo.com
INIT.ORA	
INITPS.ORA	
<nodename>_<sid>_INIT.ORA	kelem_demo_init.ora
ORAUSER_<sid>.COM	orauser_demo.com
CREATE_<sid>.COM	create_demo.com
CREATE_<sid>.SQL	create_demo.sql
SHUTDOWN_<sid>.COM	shutdown_demo.com
SHUTDOWN_<sid>.SQL	shutdown_demo.sql
STARTUP_EXCLUSIVE_<sid>.COM	startup_exclusive_demo.com
STARTUP_EXCLUSIVE_<sid>.SQL	startup_exclusive_demo.sql
ORA_DB_<sid>.COM	ora_db_demo.com
TRACE.DIR	

Para este caso se está haciendo referencia a una Base de Datos llamada **DEMO**, con un identificador de la instancia (SID) llamado **DEMO** también, en el nodo **KELEM**. Por lo general el SID tiene el mismo nombre que la Base de Datos.

En el archivo INIT.ORA se tienen los parámetros comunes que se requieren para activar una Base de Datos Oracle7 y en el archivo INITPS.ORA se tienen los parámetros para la configuración de la opción Oracle Parallel Server. Es conveniente que los parámetros que se tienen en el INIT.ORA de la configuración de la versión 6 se incluyan en este que se tiene para Oracle7 y modificar aquellos que lo requieran de acuerdo a la nueva instalación. Un parámetro que es recomendable que esté especificado es DB\_NAME ya que deberá corresponder al nombre de la Base de Datos que se ha definido para la versión 6.

#### Preparar los archivos de la Base de Datos.

6. Ejecutar el procedimiento de comandos ORAUSER\_<node>\_<sid>.COM que corresponda a la instalación del software de la versión 6.

Si la Base de Datos de Oracle6 no está activa se deberán instalar la imágenes de Oracle y levantar la Base de Datos, para lo cual se tendrán que ejecutar los procedimientos:

```
ORA_ROOT:[RDBMS]INSORACLE.COM  
ORA_ROOT:[RDBMS]INSCRTL.COM,  
ORA_DB:STARTUP_EXCLUSIVE_DEMO.COM.
```

7. Invocar el procedimiento de comandos ORA\_DB:ORA\_DB\_<dbname>.COM, el cual definirá los símbolos lógicos ORA\_PARAMS, ORA\_CONTROL1, y ORA\_CONTROL2.

8. Ejecutar el procedimiento MIGRATEUSER.COM, que se encuentra el directorio ORA\_RDBMS de la instalación de Oracle7. ORA\_RDBMS apunta en este momento a la instalación de versión 6, por lo cual, para ejecutar dicho procedimiento se deberá dar la ruta completa.

En este procedimiento se definirán los siguientes nombres lógicos :

```
ORA_MIGRATE  
ORA_MIG  
ORA_CNVFILE
```

9. Ejecutar el siguiente comando para ligar la utilería de migración

```
$@ORA_MIG:ORA_LINKMIG
```

Al invocar este procedimiento se generará el archivo ejecutable MIGRATEDB.EXE en el directorio ORA\_RDBMS de Oracle6.

10. Invocando SQL\*DBA crear el usuario MIGRATE con privilegios de connect. Una vez creado el usuario es conveniente que se le quite el privilegio de connect para evitar una mayor fragmentación del tablespace SYSTEM.

```
$ SQLDBA
```

```
sqldba> connect internal
```

```
sqldba> grant connect to MIGRATE identified by MIGRATE;
```

```
sqldba> revoke connect from MIGRATE;
```

11. Verificar que todos los Tablespaces existentes se encuentren en línea o activos. Se puede realizar una consulta a DBA\_TABLESPACES.

```
sqldba > select tablespace_name, status  
          from dba_tablespaces;
```

TABLESPACE_NAME	STATUS
SYSTEM	ONLINE
INGENIERIA	ONLINE
DATOS	ONLINE
INDICES	ONLINE
ROLLBACK	ONLINE
TEMPORAL	ONLINE

Si algún Tablespace no está en línea se deberá activar; de lo contrario **NO** se debe continuar con el proceso ya que se puede generar corrupción de la Base de Datos.

12. Dar de baja la Base de Datos con el comando SHUTDOWN NORMAL desde SQLDBA . Si se dio de baja con SHUTDOWN ABORT o SHUTDOWN IMMEDIATE, se deberá levantar la Base de Datos en modo DBA y posteriormente darla de baja con SHUTDOWN NORMAL.

En caso de que la Base de Datos no se haya dado de baja con el comando de SQLDBA, SHUTDOWN NORMAL, NO se debe continuar con el procedimiento de migración, ya que es muy probable que se genere corrupción de la Base de Datos.

13. Realizar un respaldo de todos los archivos de datos e índices de la Base de Datos, archivos de log y archivos de control. Este respaldo se deberá realizar empleado los comandos del Sistema Operativo, copy o backup. Para una mayor explicación de la sintaxis de estos comandos se puede consultar el help de VMS o consultar el manual de utilerías y comandos de Digital.

14. Posicionarse en el directorio ORA\_DB de la instalación de Oracle6.

```
$ set default ORA_DB
```

15. Correr la Fase I de la utilería de migración invocándola como se indica :  
( El símbolo migratedb se definió al momento de ejecutar el procedimiento MIGRATEUSER.COM en el paso 8. )

```
$ SHOW SYMBOL migratedb
```

```
$ MIGRATEDB
```

En caso de que no se tenga indicado el nombre de la Base de Datos con el parámetro DB\_NAME en el INIT.ORA se deberá ejecutar el comando de la siguiente manera :

```
$ MIGRATEDB DBNAME=<dbname>
```

```
$ MIGRATEDB DBNAME=demo      (para seguir con el ejemplo)
```

Esta primera fase de la migración generará un archivo llamado CONVERT.ORA en el directorio ORA\_DB de Oracle6. Este archivo contiene la descripción que se sabe de la Base de Datos a través del archivo de control. Hasta este punto la misma utilidad procesa las tablas de la Base de Datos para dejarlas preparadas para la fase 2 de migración.

Esta utilidad despliega los comandos de SQL que se están ejecutando. Para evitar que se desplieguen se puede incluir en la línea de comandos ECHO=FALSE y si se desea tener este desplegado en un archivo se debe incluir SPOOL=<archivo>.

#### Conversión de la Base de Datos a formato de Oracle7.

En esta fase, la utilidad de migración convierte los archivos de la Base de Datos a formato de Oracle7.

1. Asegurarse que se tengan definidos los símbolos para la Base de Datos de versión 6 ejecutando el archivo ORAUSER\_<dbname>.COM y adicionalmente, ejecutar el archivo de comandos ORA\_DB\_<dbname>.COM que define la ruta de ORA\_CONTROL1 y ORA\_CONTROL2. Ejecutar estos procedimientos en el orden indicado y bajo la configuración de versión 6.

```
$@ORA_DB:orauser_demo.com
```

```
$@ORA_DB:ora_db_demo.com
```

2. Es conveniente tener a la mano la ruta completa de los nombres lógicos ORA\_RDBMS, ORA\_ROOT y ORA\_DB para la configuración de versión 6.

3. Cambiar el valor del nombre lógico ORA\_PARAMS para que tenga el nombre completo del archivo de parámetros INIT.ORA de Oracle7, que deberá tener la siguiente sintáxis :

```
<nodename>_<sid>_INIT.ORA
```

```
$ DEFINE ORA_PARAMS kelem_demo_init.ora
```

Como se indica en este punto, solamente se deberá indicar el nombre del archivo, no la ruta completa donde se encuentra en el directorio de Oracle7. En este caso el nodo se llama KELEM y el identificador de la instancia (SID) se llama DEMO.

4. Posicionarse en el directorio ORA\_ROOT de la instalación de Oracle7 y ejecutar los siguientes procedimientos :

```
$ set def [.UTIL]
```

```
$@ORAUSER.COM
```

```
$ set def [-.RDBMS]
```

```
$@INSORACLE.COM
```

```
$@INSCRTL.COM
```

En caso que no se permita instalar las imágenes de Oracle7 a causa de secciones globales o paginas globales insuficientes, se deberán modificar los parámetros del Sistema Operativo para que se permita mantener activas e instaladas las imágenes de Oracle6 y Oracle7.

Para realizar estas modificaciones de parámetros del sistema operativo es conveniente que las realice el administrador del equipo y se puede auxiliar de los manuales de administración de Open VMS y en el manual de instalación de Oracle7 en la sección de pre-requisitos..

Si no están instaladas las imágenes de Oracle6 y Oracle7 **NO** se debe continuar con el procedimiento de migración, ya que se puede generar corrupción de la Base de Datos y la única forma de recuperarse será del respaldo que se recomendó realizar anteriormente.

5. Cambiarse al directorio que contiene todos los archivos para la administración de la Base de Datos en Oracle7, esto es [.DB\_<dbname>] que fue creado en el paso 4 de la sección anterior; donde <dbname> es el nombre de la Base de Datos (DEMO en este caso).

6. Ejecutar la utilidad de migración :

```
$@ORA_RDBMS:ORA_RDBMS_MIGRATE.COM
```

Al ejecutar este procedimiento se irán pidiendo una serie valores que permiten continuar exitosamente este proceso. Así mismo se generarán los siguientes archivos de log en el directorio ORA\_DB :

```
- CONVERT_DB.LOG  
- RESET_LOGS.LOG  
- LOAD_V7_CATALOG.LOG  
- LOAD_V7_PROCEXT.LOG  
- LOAD_V7_PARREXT.LOG
```

NOTA: ORA\_RDBMS apunta a la instalación de Oracle7 y  
ORA\_DB apunta a la instalación de Oracle6.

#### SANITY CHECK

To migrate successfully, you should do the following before continuing with this script. The VMS V6 to V7 migration documentation leads you through these steps, although in a different order.

- 1) Run MIGRATEDB against your V6 database (Phase I).
- 2) Make sure you **\*HAVE NOT\*** created V7 files to hold the migrated data. Your V6 data files are about to **\*become\*** your V7 data files.
- 3) If you haven't done a cold backup of your cleanly shut-down V6 database, **\*please\*** do it now.
- 4) Run the instance-specific ORAUUSER\_\*.COM and ORA\_DB:ORA\_DB\_<dbname> for the V6 database you took through Phase I of the migration.
- 5) Write down the translation of the V6 logical ORA\_RDBMS.
- 6) Run [.UTIL]ORAUUSER.COM under V7 ORA\_ROOT.
- 7) Create a V7 database administration directory.
- 8) Upgrade your INIT.ORA files from V6 to V7.
- 9) Run this script from your V7 database administration directory.

Is it OK to continue? (Y/N) [N] **Y**

Al contestar con **Y** para indicar que estos pasos ya se han previsto y ejecutado, se mostrarán las siguientes pantallas :

We show ORA\_DB pointing to the following location :

DISKSORACLE:[000000.ORACLE6.DB\_DEMO]

Is this correct for the database you wish to migrate from V6 to V7? Is CONVERT.ORA, created in Phase I, also in this location? (Y/N) [N] **Y**

## MIGRATION PHASE II

Here is what is about to happen:

- 1) We are going to point the ORA\_CNVFIL.E logical to the CONVERT.ORA file generated in Phase I.
- 2) We will temporarily point ORA\_SYSTEM to the V7 database administration area (rather than the V6 ORA\_INSTANC1) and ORA\_RDBMS.
- 3) The script will then redefine ORA\_CONTROL1 and ORA\_CONTROL2 to the names of your new control files.
- 4) It will invoke SQL\*DBA and perform the conversion of the data files from V6 format to V7 format.
- 5) If you indicate that the conversion was successful, we will reset your log files and load the V7 data dictionary.
- 6) Finally, we will register your V6 instances with V7 if everything else was successful.

**WARNING :** Keep a careful eye on the conversion process. You may corrupt your database if you reset the log files after a conversion error.

Are you ready to proceed ? (Y/N) [N] Y

If the conversion was successful, we are now ready to reset the log files. Please examine the above output and the contents of ORA\_DB:CONVERT\_DB.LOG before making your decision.

**WARNING :** Do **\*NOT\*** reset the log files if there was an error during conversion. You will need to read the documentation, and/or call Oracle Worldwide Support, before trying again. Proceeding further after a convert error can result in database corruption.

Should we reset the log files? (Y/N) [N]

Si se ha decidido continuar con el procedimiento de migración se tendrán los siguientes mensajes :

```
SQL*DBA: Release 7.0.12.0.0 - Production on Thu Nov 3 18:57:48 1994
Copyright (c) Oracle Corporation 1979, 1992. All rights reserved.
ORACLE7 Server Release 7.0.12.0.0 - Production
With the procedural and Parallel Server Option
PL/SQL Release 2.0.14.0.0 - Production
```

```
SQLDBA> Connected
File diskSoracle:[oracle6.db_demo]reset_logs.log opened Thu Nov 3
```

```
SQLDBA> Statement processed.
Database closed,
Database dismounted
ORACLE Instance shut down.
```

```
ORACLE Instance started.
Database mounted,
Database opened.
Total System Global Area          4137192 bytes
      Fixed Size                    32060 bytes
      Variable Size                 3974060 bytes
Database Buffers                   122880 bytes
Redo Buffers                        8192 bytes
SQL*DBA complete.
```

```
Your database's log files have been reset.
We will now load in the V7 data dictionary.
```

7. ORA\_RDBMS\_MIGRATE.COM concluye dando de baja la Base de Datos pero ahora ya está en formato de Oracle7.

Esto implica que ahora se deberá levantar la Base de Datos activando las imágenes de Oracle7, ejecutar el procedimiento de comandos ORAUSER\_<SID>.COM (ORAUSER\_DEMO.COM) que se encuentra en el directorio ORA\_DB de Oracle7 que definirá los símbolos para acceder a la Base de Datos a través de los productos instalados con la nueva versión del RDBMS.

En caso de que se tengan procedimientos particulares para la administración de la Base de Datos se deberán modificar para que contemplen las indicaciones anteriores.

8. Editar el archivo `ORA_DB_<dbname>.COM` (`ORA_DB_DEMO.COM`) que se encuentra en el directorio `ORA_DB` de la instalación de Oracle7 y modificar la definición lógica de la ubicación de los nuevos archivos de control que se han creado para la configuración de la Base de Datos con Oracle7, estos archivos de control serán referenciados por los nombres lógicos `ORA_CONTROL1` y `ORA_CONTROL2`, al igual que con Oracle6 pero se les agregan los caracteres `V7` para diferenciarlos de los que se emplearon con la versión 6.

Por ejemplo, si un archivo de control de la versión 6 de Oracle se llama `ORA_CONTROL1.CON` el nuevo archivo se llamará `ORA_CONTROL1V7.CON`.

Se debe hacer referencia explícitamente al valor que se tenía como `ORA_DB` para Oracle6 al especificar la ubicación de los archivos de control y dejar que el símbolo `ORA_DB` apunte al directorio que contiene los archivos de administración de la Base de Datos con Oracle7. Por lo tanto no se deberá modificar la ruta de `ORA_DB` y `ORA_ARCHIVE`.

Por ejemplo, si para la versión 6, `ORA_DB` está especificado como `DISK$ORACLE:|ORACLE6.DB_DEMO|` y para Oracle7 `ORA_DB` está como `DISK$SISTEMAS:|ORACLE7.DB_DEMO|`; el archivo que contiene las definiciones lógicas `ORA_DB_<dbname>.COM` será como el siguiente :

```
$ define/nolog ORA_DB          DISK$SISTEMAS:|ORACLE7.DB_DEMO|
$ define/nolog ORA_CONTROL1   ORA_DB:ORA_CONTROL1.CON
$ define/nolog ORA_CONTROL2   ORA_DB:ORA_CONTROL2.CON
$ define/nolog ORA_PARAMS     ora_db:'nodename$$'_'ora_local_sid'_INIT.ORA
$ define/nolog ORA_INITSQL    ORA_SYSTEM:SQL.BSQ
$ define/nolog ORA_ARCHIVE    disk$systemas:|oracle7.db_demo|
```

Una vez realizadas la modificaciones que se acaban de mencionar, este archivo deberá ser parecido al siguiente :

```

$ define/nolog      ORA_DB          DISKSSISTEMAS:{ORACLE7.DB_DEMO}
$ define/nolog      ORA_CONTROL1 -
                   DISK$ORACLE:{ORACLE6.DB_DEMO}ORA_CONTROL1V7.CON

$ define/nolog      ORA_CONTROL2 -
                   DISK$ORACLE:{ORACLE6.DB_DEMO}ORA_CONTROL1V7.CON

$ define/nolog      ORA_PARAMS      ora_db:'nodename$$'_ 'ora_local_sid'_INIT.ORA
$ define/nolog      ORA_INITSQL     ORA_SYSTEM:SQL.BSQ
$ define/nolog      ORA_ARCHIVE     disk$systemas:|oracle7.db_demo}

```

9. Si el procedimiento de migración resultó exitoso, es conveniente utilizar los procedimientos de Startup y Shutdown para comprobar que todo ha quedado bien configurado. Una vez que se han ejecutado estos procedimientos es conveniente realizar un respaldo de los archivos de la Base de Datos, archivos de log y de control .

En caso que al ejecutar el procedimiento ORAUSER\_<SID>.COM, se indica que el SID no está reconocido o definido, es conveniente checar que el archivo que se encuentra en ORA\_RDBMS:ORA\_RDBMS\_SIDS.DAT de la versión 6 corresponda al que se tiene en el directorio de Oracle7.

10. En este punto es conveniente planear y modificar los parámetros del INIT.ORA para empezar con las nuevas características que propone Oracle7.

11. Modificar los scripts de creación de objetos con SQL y verificar la nueva sintaxis para las instrucciones que lo ameriten o los tipos de datos.

12. Realizar las pruebas locales que sean convenientes para afinar aplicaciones, Base de Datos o procesos.

13. Una vez que se ha concluido este trabajo de migración y que se está seguro que todo opera como en la versión 6, se puede borrar el usuario MIGRATE con SQL\*DBA.

```
SQLDBA> DROP USER MIGRATE CASCADE;
```

La opción cascade es una característica de Oracle7 que implica que al borrar un usuario deshace automáticamente todos los objetos que había creado (tablas, índices, etc.).

Una vez que se ha verificado que la Base de Datos funciona adecuadamente con las adecuaciones mencionadas, es momento oportuno para borrar los directorios obsoletos.

Las ventajas se resumen en que se consume menos espacio en disco ya que se utilizan los mismos archivos de la Base de Datos anterior, es un método rápido y hace una conversión automática de los tipos de datos char de la versión 6 a varchar2 que es el que se maneja en Oracle7. Las desventajas son que se requiere preparar muy bien el ambiente y la Base de Datos de la versión 6, la fragmentación y porosidad que se tenía permanece, el tablespace SYSTEM crece y se requiere de aproximadamente 15 MB de espacio libre en disco para los archivos temporales que se generan durante este proceso de migración.

### **III. Exportar / Importar**

Esta es la tercera opción para realizar una migración de una Base de Datos que se encuentra en versión 6 de Oracle y se desea dejarla en operación con Oracle7. Como el nombre lo indica, en este método se emplean las utilerías de Oracle Export/Import para realizar el proceso de migración y es empleado cuando además de trasladar la información de una versión a otra, se pretende corregir fragmentación y/o porosidad de la información en la Base de Datos.

La información que aquí se presenta se puede complementar consultando el manual "ORACLE Utilities"; el cual tiene una explicación mas detallada de las opciones y uso del export e import.

Se presupone la instalación del software de Oracle7 y como en los casos anteriores de personal que tiene al menos conocimientos básicos de VMS, Oracle, las aplicaciones y configuración local de la Base de Datos, ya que de otra manera no se comprenderían los conceptos que aquí se manejan y no se podría determinar a que se refiere alguna eventualidad que pueda surgir en el transcurso de este procedimiento.

A continuación se numeran los pasos a realizar para la conclusión de esta actividad empleando el método de exportar e importar información entre Bases de Datos :

1. Instalar el software de Oracle7 y crear una Base de Datos inicial en la que se tendrá el resultado de la migración. Para realizar estas actividades se requiere del software en algún medio magnético y para la instalación se puede consultar el manual "Oracle7 for VAX OpenVMS Installation Guide"
2. Levantar la Base de Datos de Oracle6 en caso de que no esté activa. Vale la pena recordar que para realizar el export completo de la Base de Datos, no deberá haber ningún usuario conectado a la Base de Datos, ya que se puede respaldar información obsoleta hasta el momento que concluyó el export.

Es recomendable levantar la Base de Datos en modo DBA para asegurar que al menos no podrán entrar los usuarios que no tienen privilegios de DBA (administrador de la Base de Datos).

```
SQLDBA> startup dba open demo
```

3. Invocar la utilería export de Oracle6 para realizar el vaciado de la información que se migrará :

```
$EXP system/manager
```

```
Export: Version 6.0.37.3.0 - Production on Fri Nov 4 16:33:32 1994
```

```
Copyright (c) Oracle Corporation 1979, 1993. All rights reserved.
```

```
Connected to: ORACLE7 RDBMS V6.0.37.3.0 (6.2), - Transaction processing option -  
Production
```

```
PL/SQL V1.0.40.0.0 - Production
```

```
Enter array fetch buffer size: 4096 >
```

```
Export file: DISKSSISTEMAS:|ORACLE6|EXPDAT.DMP >
```

```
DISKSSISTEMAS:|ORACLE6.RESPALDOS|ORACLE6.DMP
```

```
E(ntire database), U(sers), or T(ables): U > E
```

```
Export grants (Y/N): N > Y
```

```
Export table data (Y/N): Y > Y
```

```
Compress extents (Y/N): Y > Y
```

```
Warning: constraints on tables not exported
```

```
About to export the entire database ...
```

```
. exporting tablespace definitions
```

```
. exporting user definitions
```

En este caso el respaldo que se realiza queda identificado con el archivo: DISKSSISTEMAS:|ORACLE6.RESPALDOS|ORACLE6.DMP. Este archivo es el que se deberá indicar al momento de importar la información a Oracle7.

3. Se deberá saber el nombre y tamaño de los tablespaces en Oracle6 para que se haga la creación del ambiente en Oracle7 de igual forma y evitar los problemas que se generarían al importar los datos.

Se puede consultar el diccionario de datos desde SQL\*DBA para obtener estos datos :

```
SQLDBA> select file_name, tablespace_name, bytes  
         from dba_data_files;
```

4. Levantar la Base de Datos de Oracle7 y crear los tablespaces del tamaño y nombre como los que se tienen en Oracle6.

Se pueden crear los usuarios y tablas antes de ejecutar el import. Esto con la finalidad de optimizar los espacios que se ocuparán al momento de crear las tablas e índices en la Base de Datos; además es un paso en el que se obliga a corregir los scripts de SQL que crean las tablas para cambiar los tipos de datos CHAR por VARCHAR2, incluir reglas de integridad, restricciones en las tablas como CHECK, DEFAULT, etc.

El manual de Oracle "SQL Language" puede servir como apoyo en caso de que no se conozca la sintaxis para la creación de usuarios, tablas y tablespaces.

5. Realizar el Import en la Base de Datos de Oracle7:

En este punto se supone que se está conectado al Sistema Operativo con el usuario que es el propietario del software de Oracle7, y se han definido los símbolos para los productos de Oracle7, esto es el ORAUSER.COM correspondiente.

---

```
$ IMP system/manager
```

```
Import: Release 7.0.12.0.0 - Production on Fri Nov 4 17:04:21 1994
```

```
Copyright (c) Oracle Corporation 1979, 1992. All rights reserved.
```

```
Connected to: ORACLE7 Server Release 7.0.12.0.0 - Production  
With the procedural and Parallel Server option  
PL/SQL Release 2.0.14.0.0 - Production
```

```
Import file: DISKSSISTEMAS:[ORACLE7]EXPDAT.DMP >  
              DISKSSISTEMAS:[ORACLE6.RESPALDOS]ORACLE6.DMP  
Enter insert buffer size (minimum is 4096) 30720>  
List contents of import file only (yes/no): no >  
Ignore create error due to object existence (yes/no): yes >  
Import grants (yes/no): yes >  
Import table date (yes/no): yes >  
Import entire export file (yes/no): yes >
```

El archivo que se indica para hacer el cargado de la información es el mismo que el que se indicó al realizar el export desde Oracle6.

En este momento la Base de Datos ya estará lista para realizar pruebas de funcionalidad con las aplicaciones y si no existen problemas se podrá borrar el software y Base de Datos de la versión 6.

Las ventajas de emplear este método es que se pueden seleccionar los usuarios y tablas que se desean migrar, la Base de Datos anterior permanece intacta, es un método óptimo para realizar pruebas, se corrige la porosidad y fragmentación en la Base de Datos así como se realiza la conversión de tipos de datos automáticamente de char a varchar2. Las principales desventajas son que se requiere de suficiente tiempo para realizar el export e import, lo cual depende del tamaño de la Base de Datos y además se necesita suficiente espacio en disco para mantener el software y las Bases de Datos de Oracle6 y Oracle7.

Este método es altamente recomendable ya que se corrigen varios aspectos de la Base de Datos y se puede incluso planear una real reorganización de usuarios y uso del espacio en disco que ocupa la Base de Datos. Con el simple hecho de realizar un export e import se está corrigiendo la fragmentación y porosidad de la Base de Datos, por lo que independientemente si se realiza o no la migración con este método es recomendable ejecutar estas operaciones eventualmente.

Por lo tanto, el tiempo que se tenga que invertir en estas actividades bien vale la pena planearlo para que no se vaya a afectar el acceso normal a la información de la Base de Datos.

El método de transferencia de información a través de SQL\*Net no se utilizó en este caso porque no aplica de acuerdo a la configuración actual de equipos y software en la USECAFI; sin embargo el método II que es el emplear la utilidad de migración fue útil para el planteamiento recomendado de pruebas. Sin embargo el método III que consiste en exportar e importar la información es el que se utilizó finalmente debido a que se aprovechó el tiempo para realizar tanto la migración como una compactación de la información, y con este trabajo corregir la porosidad y fragmentación existente en la Base de Datos.

---

## MIGRACION DE LAS APLICACIONES

La mayoría de las aplicaciones existentes en la USECAFI están desarrolladas con SQL\*Forms 3.0, SQL\*Menu 5.0 y Pro\*C.

Se deberán regenerar las aplicaciones de FORMS y MENU como se indica a continuación :

Para SQL\*Forms :

```
$ generate30 -t-b <nombre de la forma> <username/password>
```

Para SQL\*Menu :

```
$ genmenu50 <aplicación> <username/password>
```

Para mayor información sobre las opciones válidas y sintáxis se pueden consultar los manuales "Reference Guide" de SQL\*Menu y SQL\*Forms.

Para los módulos que están hechos en Pro\*C, se deberán precompilar, compilar y ligar para generar un nuevo ejecutable como se indica en el siguiente ejemplo :

```
$ proc iname=<archivo.pc> ireclen=132 oreclen=132  
      userid=<username/password>
```

```
$ cc <archivo.c>
```

```
$ lnproc <archivo ejecutable> <lista de módulos objeto>
```

Las aplicaciones que se encuentren con SQL\*Forms 2.3 deberán convertirse a SQL\*Forms 3.0; esta conversión solo servirá para que dichas aplicaciones operen funcionalmente con la versión 3.0, pero no se están aprovechando las características propias de SQL\*Forms 3.0, como los procedimientos hechos con PL/SQL por ejemplo.

Si se quieren aprovechar las características de SQL\*Forms 3.0 se tendrán que modificar los pasos de triggers que se mantienen de la versión 2.3 a procedimientos realizados con PL/SQL y que realicen las mismas operaciones; de lo contrario, como se ha mencionado, continuarán trabajando como lo hacían con SQL\*Forms 2.3.

La manera de convertir estas aplicaciones debe ser de acuerdo a la siguiente sintaxis :

```
$ convform30 -t-v <archivo.inp> <nombre de la forma> <username/password>
```

```
$ generarte30 -t-b <nombre de la forma> <username/password>
```

La generación de la forma se puede realizar directamente desde la herramienta SQL\*Forms 3.0 cargandola con la opción LOAD primeramente en caso de que no esté salvada dicha forma en la Base de Datos.

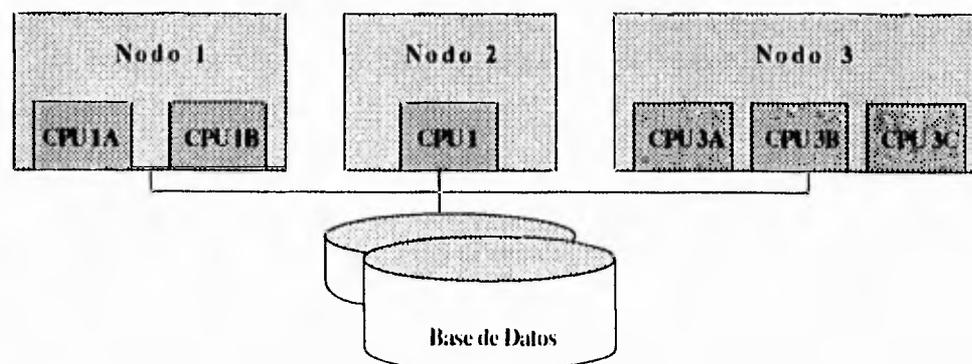
Las aplicaciones existentes funcionan adecuadamente y no se tendrán que modificar si se trabaja con la configuración de multi-trehaded server de Oracle7, ya que como se comentó en el capítulo anterior esto es una característica del RDBMS que no afecta a las aplicaciones.

Si se desean aprovechar las características de procedimientos almacenados y triggers a nivel de la Base de Datos en este ambiente en general, se deberán modificar las aplicaciones existentes y planear que procedimientos se deben centralizar, así como los permisos para cada usuario.

## ARQUITECTURA Y CONFIGURACION DE ORACLE EN CLUSTER

Como se ha mencionado en capítulos anteriores, existen distintos tipos de configuraciones de equipos de cómputo por lo que cada uno intenta mantener un mayor rendimiento y seguridad a la información de la Base de Datos.

Una de estas arquitecturas conocida como *tightly coupled* contiene varios procesadores accediendo el mismo arreglo de memoria a través de un bus común, este es el caso de las máquinas con procesadores simétricos (SMP); sin embargo la arquitectura *loosely coupled* opera con varios nodos independientes que comparten recursos como discos y dispositivos de almacenamiento secundario. Un nodo en esta arquitectura consiste de uno o varios procesadores con sus recursos de memoria independientes. La memoria no es compartida entre los nodos, solamente los nodos se comunican a través de un bus común de alta velocidad. Esto quiere decir que un nodo que pertenece al arreglo del Cluster puede ser una máquina con arquitectura *tightly coupled*, como se muestra en la siguiente figura :



En una red tradicional, las máquinas individuales se conectan con el fin básico de compartir información a través de algún canal. En este caso, el mantenimiento y acceso a la información se lleva a cabo por medio de un manejador para Bases de Datos distribuidas, o que al menos pueda contemplar un esquema de trabajo bajo una arquitectura cliente-servidor.

En un sistema SMP, todos los CPU's cooperan entre sí a través del acceso a memoria real compartida mediante un "bus" común. Debido a que este bus tiene un ancho de banda limitado, el número de CPU's que pueden conectarse es limitado también. Todos los CPU's se coordinan entre sí vía hardware y tienen acceso a la misma memoria, el uso de un manejador de Bases de Datos tradicional será suficiente para utilizar las capacidades de los CPU's del sistema, es decir no es necesario contemplar las capacidades distribuidas.

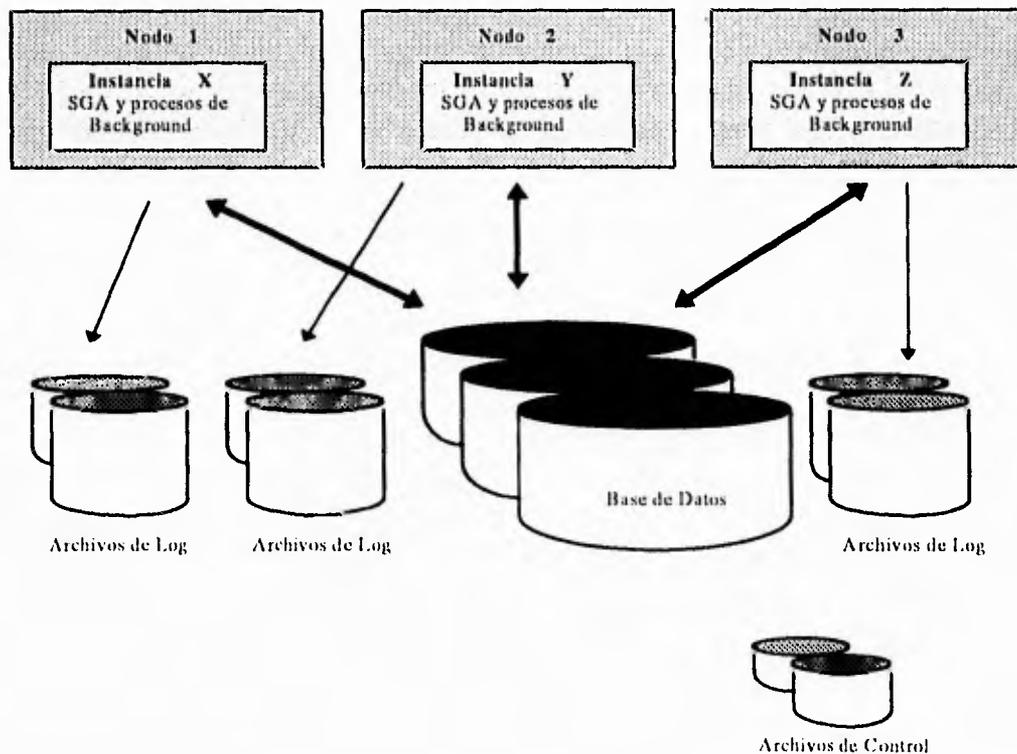
En la arquitectura de bajo acoplamiento o Cluster se mantiene un acceso al mismo arreglo de discos a través de varios nodos que contienen sus propios recursos de memoria y procesadores. La ventaja que se llega a tener con esta configuración, y en particular con un VAXCluster es que al fallar algún nodo, no se pierde el acceso a la información que se tiene en los discos.

Oracle maneja una opción en su software llamada "*Parallel Server*", que permite tener instancias separadas, corriendo simultáneamente en uno o más nodos de una configuración Cluster, accediendo una sola Base de Datos. Para tal efecto se vale de una tecnología que han llamado *parallel cache management (PCM)*.

Las características que se tienen con la opción de Parallel Server de Oracle son :

- Se puede activar una instancia en cada nodo que pertenezca al Cluster.
  
  - Cada instancia es un SGA y un conjunto de procesos de background separados en cada nodo del Cluster.
  
  - Todas la instancias comparten los mismos archivos de datos y de control.
-

- Cada instancia tiene sus propios archivos de log.
- Los archivos de datos, de log y control están en uno o varios discos que pertenecen a la configuración del cluster.
- Todas las instancias pueden ejecutar transacciones sobre la misma Base de Datos, y cada instancia puede tener varios usuarios ejecutando transacciones.
- El manejo de candados a nivel registro se mantiene y se garantiza la consistencia e integridad de la información, a pesar del acceso a la información a través de varios nodos, en los que cada uno tiene su propio SGA.



La manipulación de la Base de Datos puede ser a través de varias instancias con el Parallel Server o mediante una configuración distribuida o cliente-servidor. Un ambiente con el Parallel Server también puede ser parte de un ambiente distribuido conectado a través de SQL\*Net.

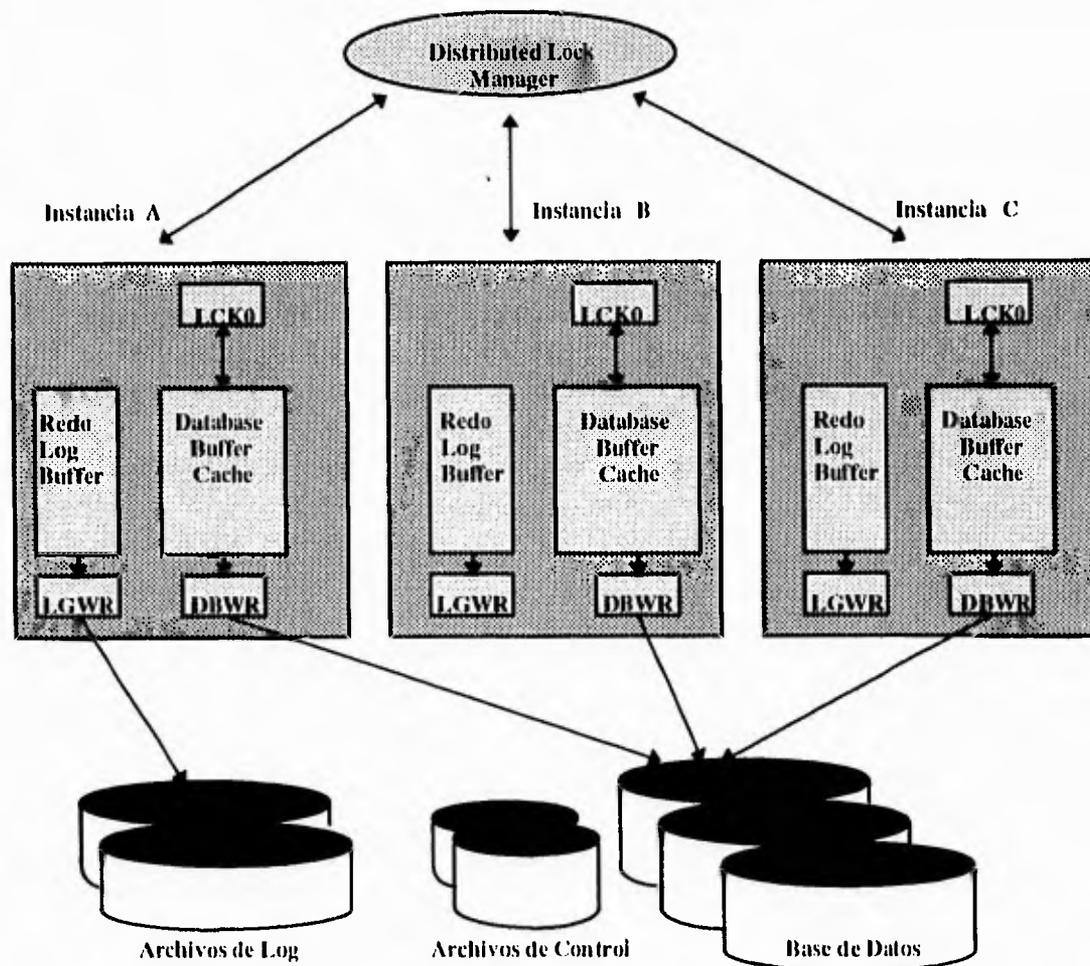
Con la opción del Parallel Server, el RDBMS de Oracle puede correr en modo exclusivo o modo paralelo. Con el modo exclusivo solamente una instancia podrá montar y manipular la Base de Datos; sin embargo con el modo paralelo múltiples instancias podrán montar y manipular la Base de Datos.

### **Parallel Cache Management (PCM)**

Para poder coordinar los recursos compartidos entre los nodos de un sistema con una configuración Cluster se hace uso del PCM. El Parallel Cache Management (PCM) usa candados distribuidos para coordinar los accesos a los recursos que se solicitan para cada instancia que está activa en el cluster. Los recursos que son manejados por los candados distribuidos son los segmentos de rollback, accesos al diccionario de datos a través del SGA y los datos en el SGA (database buffers).

Una instancia coloca un candado distribuido a los bloques que estará utilizando para su manipulación en su SGA propio y nunca lo libera hasta que otra instancia haga referencia a este dato que se está manipulando. Los esquemas de Cluster tienen un mecanismo de control y manejo de candados distribuidos llamado *distributed lock manager (DLM)*, que comunica los requerimientos de cada nodo y mantiene el status de los candados entre procesos e instancias dadas de alta en el Cluster.

La siguiente figura muestra la relación existente entre los procesos de background LCKn, el Database Buffer Cache y el Distributed Lock Manager (DML) en un ambiente de tres instancias con el Parallel Server de Oracle.



Con el DLM (Distributed Lock Manager) del Sistema Operativo, Oracle Parallel Server coordina múltiples buffers en memoria, rastreando el estado y localidad de los datos en cada nodo. Si alguna instancia necesita modificar datos que ya han sido previamente modificados dentro de los buffers en cache de alguna otra instancia, el DLM hace que la segunda instancia escriba los datos al disco de tal forma que la primera pueda accederlos.

Hay que hacer notar que Oracle Parallel Server hace uso de sus propios mecanismos para controlar la concurrencia sobre los objetos de la Base de Datos (índices, tablas, etc.); el DLM solo se utiliza para efectuar la comunicación y la coordinación de los bloques de datos entre instancias. También es importante mencionar que los candados a nivel renglón están presentes también con la configuración del Oracle Parallel Server.

Básicamente la tecnología del Oracle Parallel Server funciona de la siguiente forma : Solo se leen bloques de datos que aun no se encuentren en los buffers del SGA (System Global Area) de la instancia que requiere dicha información. Los bloques de datos que han sido modificados son escritos al disco únicamente cuando son requeridos por otra instancia, cuando se vacían para permitir la entrada de mas bloques al SGA de acuerdo al algoritmo LRU (Last Recently Used), o por checkpoints de la Base de Datos, los cuales aseguran que las transacciones registradas en los archivos de log se apliquen sobre los archivos de datos. Estos archivos de log se utilizan para garantizar la integridad de los datos cuando una transacción hace "commit" y permiten efectuar una recuperación de la Base de Datos en caso de una caída de la misma.

En caso que una instancia se desactive por caída de la máquina o alguna otra causa, alguna de las instancias restantes del cluster, efectuará una recuperación automática de la instancia que falló, permitiendo que los usuarios del nodo correspondiente puedan conectarse a la Base de Datos a través de otro nodo para poder continuar disponiendo de su información.

El parámetro de inicialización del INIT.ORA GC\_DB\_LOCKS indica el máximo número de PCM locks que se utilizarán para todos los archivos de datos y de índices. Un *PCM lock* es un candado distribuido que contempla uno o mas bloques de datos y de índices de la Base de Datos.

---

Un PCM Lock es utilizado por una instancia, cuando el bloque de datos se lleva al área del SGA "Database Buffers"; y será desactivado solamente cuando dicho bloque sea requerido para modificación por otra instancia. En caso que se requiera para lectura el bloque, no será liberado si es que también se utilizará para lectura por otra instancia. En este caso cada instancia tiene el mismo bloque en su SGA indicando que solo será para lectura.

Con el Oracle Parallel Server, solamente una instancia podrá modificar bloques de datos a un tiempo. Si una instancia modifica un bloque de datos que otra instancia necesita, la primera instancia deberá escribir el bloque en disco antes de que la otra instancia pueda leer este bloque. El proceso de bloqueo es realizado por el DLM (Distributed Lock Manager) del Sistema Operativo en un ambiente Cluster, como es el caso de OpenVMS de Digital.

El parámetro GC\_FILES\_TO\_LOCKS indica cuantos bloques de datos cubrirá cada PCM lock. Los PCM locks pueden corresponder uno a uno con el número de bloques de la Base de Datos. Comúnmente se asignan un número de bloques a cada PCM. El número de PCM locks asignados a cada archivo de datos está determinado por la siguiente fórmula:

$$\text{Promedio de bloques por PCM lock} = \frac{\text{Número de bloques en el archivo de datos}}{\text{Número de PCM locks asignados para el archivo}}$$

Cada PCM lock debe mapear ciertos bloques de los archivos de la Base de Datos. Por ejemplo, si una instancia A modifica datos en los archivos 1, 2 y 3; mientras que la instancia B modifica datos principalmente en los archivos 4 y 5; se pueden asignar PCM locks para los archivos 1, 2 y 3 y otro conjunto de candados para los archivos 4 y 5. Para este caso cada instancia tiene sus propios PCM locks para cubrir los bloques que modificará. De otra forma se puede definir un conjunto de PCM locks para los archivos 3 y 4 y ambas instancias usarán el mismo conjunto de candados pero el trabajo de Entrada/Salida a disco se incrementa.

El parámetro GC\_FILES\_TO\_LOCKS asigna el número de PCM locks para cada archivo de la Base de Datos y se realiza con la siguiente sintaxis :

```
GC_FILES_TO_LOCKS = "{#archivos=#PCM locks[!#bloques][EACH]}[:]..."
```

donde :

#archivos = uno o más archivos de datos, indicados por el número del archivo.  
#PCM locks = El número de PCM locks asignados a cada archivo.  
! = Indica el número de bloques contiguos de datos que se apartarán por PCM.  
EACH = Indica que el número de PCM locks será para cada archivo de datos.

Por ejemplo se puede tener este parámetro de la siguiente forma, el cual se explicará paso a paso para su mejor entendimiento :

```
GC_FILES_TO_LOCKS = "1-3=200EACH;4=50;5-7,9=100;8,10=20!50"
```

El número del archivo es el rango 1-3, que significa que son los archivos 1, 2 y 3 de la Base de Datos. Para saber el número que le corresponde a cada archivo se puede consultar la vista DBA\_DATA\_FILES desde SQL\*DBA :

```
sql> connect internal;  
sql> select file_id, file_name, tablespace_name  
       from dba_data_files;
```

---

FILE_ID	FILE_NAME	TABLESPACE_NAME
-----	-----	-----
1	disk\$rdbmsoracle:[oracle7]system.dbs	system
2	disk\$trabajo:[oracle7]indexes.dbs	indexes
3	disk\$sistemas:[oracle7]rollbacks.dbs	rollbacks
4	disk\$rdbmsoracle:[oracle7]data.dbs	data
5	disk\$archivos:[oracle7]temporal.dbs	temporal

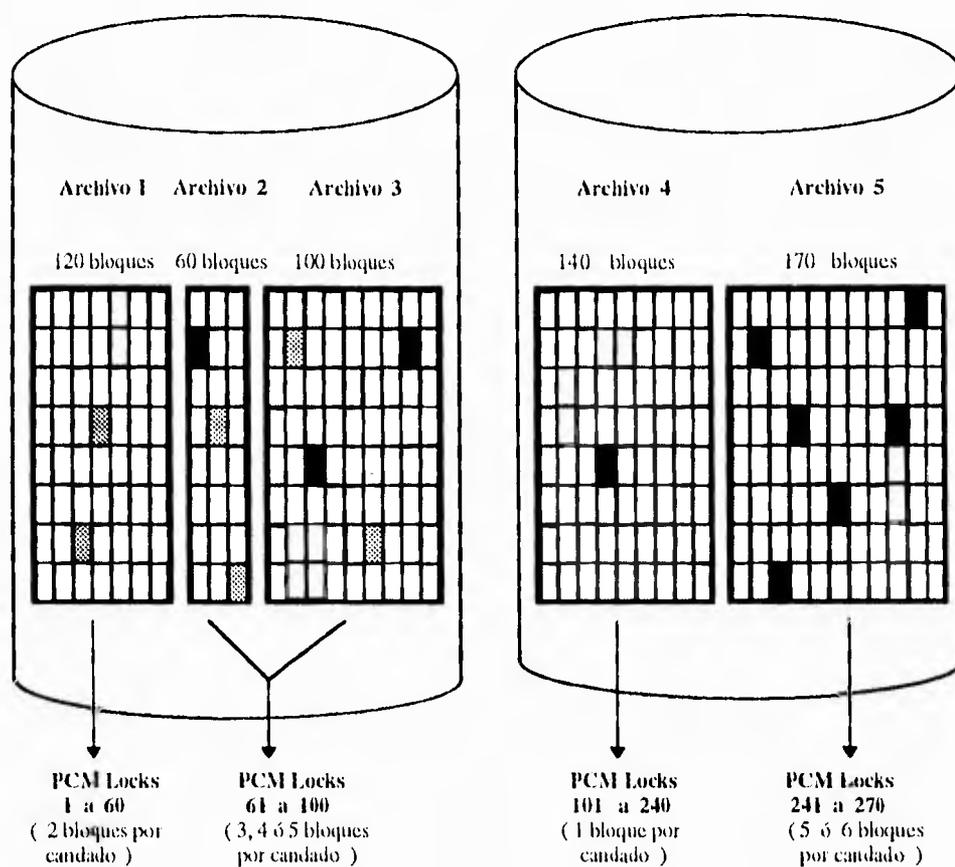
En este ejemplo se tienen 5 archivos para la Base de Datos, los cuales podremos especificar en el parámetro GC\_FILES\_TO\_LOCKS del INIT.ORA.

Cuando se añada o borre un archivo, se cree o se borre un tablespace se deberá también modificar este parámetro GC\_FILES\_TO\_LOCKS antes de dar de alta nuevamente Oracle en modo paralelo.

Otro aspecto que es importante considerar es que el valor de GC\_FILES\_TO\_LOCKS no deberá ser mayor al valor de GC\_DB\_LOCKS.

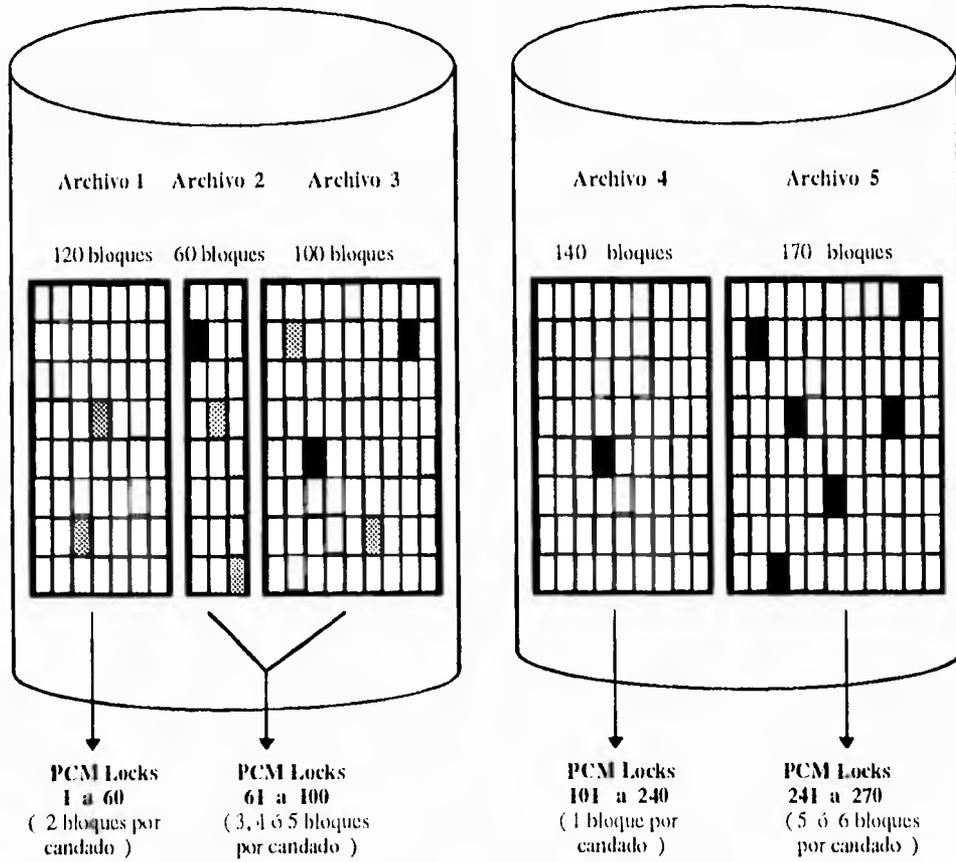
Es conveniente que el número de PCM locks sea un múltiplo del total de bloques de un archivo de datos. Por ejemplo, si se asignan 400 PCM locks para un archivo de datos que contiene 2500 bloques; 100 PCM locks podrían cubrir 7 bloques cada uno y 300 PCM locks cubrirán 6 bloques.

La siguiente figura ejemplifica el mapeo de bloques de la Base de Datos con los PCM locks con el valor del parámetro GC\_FILES\_TO\_LOCKS = "1=60:2=3=40:4=140:5=30".



Para este ejemplo, en el archivo 1 se tiene que 60 PCM locks son para mapear 120 bloques, que es un múltiplo de 60; por lo que cada PCM lock cubre dos bloques de datos.

En el archivo 2 y 3, con 40 PCM locks se mapean 160 bloques, 60 del archivo 2 y 100 del archivo 3, como se muestra en la figura. Como los bloques de los archivos 2 y 3 no son múltiplo de 40, cada PCM lock cubrirá en promedio 4 bloques de datos. Un PCM lock podrá cubrir 1 ó 2 bloques del archivo 2; y 2 ó 3 bloques del archivo 3. Por lo tanto un PCM lock podrá cubrir 3, 4 ó 5 bloques entre ambos archivos.



Para este ejemplo, en el archivo 1 se tiene que 60 PCM locks son para mapear 120 bloques, que es un múltiplo de 60; por lo que cada PCM lock cubre dos bloques de datos.

En el archivo 2 y 3, con 40 PCM locks se mapean 160 bloques, 60 del archivo 2 y 100 del archivo 3, como se muestra en la figura. Como los bloques de los archivos 2 y 3 no son múltiplo de 40, cada PCM lock cubrirá en promedio 4 bloques de datos. Un PCM lock podrá cubrir 1 ó 2 bloques del archivo 2; y 2 ó 3 bloques del archivo 3. Por lo tanto un PCM lock podrá cubrir 3, 4 ó 5 bloques entre ambos archivos.

En el archivo 4, cada PCM lock mapea exactamente un bloque de datos del archivo, debido a que existen el mismo número de PCM locks como bloques.

En el archivo 5, 30 PCM locks mapean 170 bloques de datos que no es múltiplo de 30, por lo tanto cada PCM lock cubrirá 5 ó 6 bloques de datos.

Por mencionar otro ejemplo, supongamos que se tiene el siguiente valor :

`GC_FILES_TO_LOCKS = "1=500:2-4,10-12=400EACH:5=150:6=250:7-9=300"`

Esto lo emplearíamos para indicar que se utilizarán 500 PCM locks para el archivo 1; 400 PCM locks para cada archivo del rango especificado, que son el archivo 2, 3, 4, 10, 11 y 12; 150 PCM locks para el archivo 5; 250 PCM locks para el archivo 6 y 300 PCM locks repartidos entre los bloques de datos que tengan los archivos 7, 8 y 9.

Para este ejemplo se están asignando un total de  $(500 + (6*400) + 150 + 250 + 300) = 3600$  PCM locks; por lo tanto el valor del parámetro `GC_DB_LOCKS` debe ser al menos 3600.

`GC_DB_LOCKS = 3600`

300 PCM locks son utilizados por los archivos 7, 8 y 9 de manera compartida porque se ha omitido la cláusula `EACH` y se ha indicado "`7-9=300`", lo que indica que los archivos del rango 7 al 9 compartirán 300 PCM locks. Si cada archivo estuviera formado por 900 bloques, para un total de 2700. De acuerdo con la fórmula para el cálculo promedio de bloques por PCM lock se tendría que cada PCM lock cubrirá 9 bloques de datos :

Promedio = Bloques / PCM locks

Promedio =  $2700/300 = 9$  bloques / PCM lock

En el siguiente ejemplo se tienen 200 PCM locks para cada archivo del 1 al 3; 50 PCM locks para el archivo 4; 100 PCM locks compartidos entre los archivos 5, 6, 7 y 9; y 20 PCM locks para grupos de 50 bloques contiguos en los archivos 8 y 10 de manera compartida:

```
GC_FILES_TO_LOCKS = "1-3=200EACH:4=50:5-7,9=100:8,10=20!50"
```

En caso que algún archivo de datos no se haya especificado en el parámetro GC\_FILES\_TO\_LOCKS, éstos utilizarán los PCM locks restantes. El número de PCM locks restantes es lo que sobra del total de PCM locks especificados en el parámetro GC\_FILES\_TO\_LOCKS y el valor indicado en el parámetro GC\_DB\_LOCKS; por esto se ha comentado que el valor de GC\_DB\_LOCKS debe ser al menos la suma de PCM locks indicados en GC\_FILES\_TO\_LOCKS.

Para el caso de la Base de Datos de la Facultad de Ingeniería que se ha configurado en un ambiente VAXCluster, tenemos los siguientes archivos y sus tamaños en bloques de 2048 bytes :

```
sql> connect internal;
sql> select FILE_ID, TABLESPACE_NAME, BLOCKS, BYTES
        from DBA_DATA_FILES;
```

FILE_ID	TABLESPACE_NAME	BLOCKS	BYTES
1	SYSTEM	10240	20971520
2	INDEXES	409600	838860800
3	DATA	409600	838860800
4	ROLLBACKS	256000	524288000
5	TEMPORAL	102400	209715200

Por ejemplo se pueden especificar los siguientes valores :

```
GC_FILES_TO_LOCKS = "1=500:2=2000:3=2000:4=500:5=500"
```

```
GC_DB_LOCKS = 5500
```

Esto indica que para el archivo 1 se contará con 500 PCM locks que corresponderán a 10240 bloques, por lo que cada PCM lock cubrirá aproximadamente 20 bloques.

Los archivos 2 y 3 cuentan con 2000 PCM locks cada uno, para cubrir los 409600 bloques con que cuenta cada archivo. Cada PCM lock cubrirá 204 ó 205 bloques.

Para el archivo 4 se especifican 500 PCM locks. Cuenta con 256000 bloques lo que indica que cada PCM lock cubrirá 512 bloques.

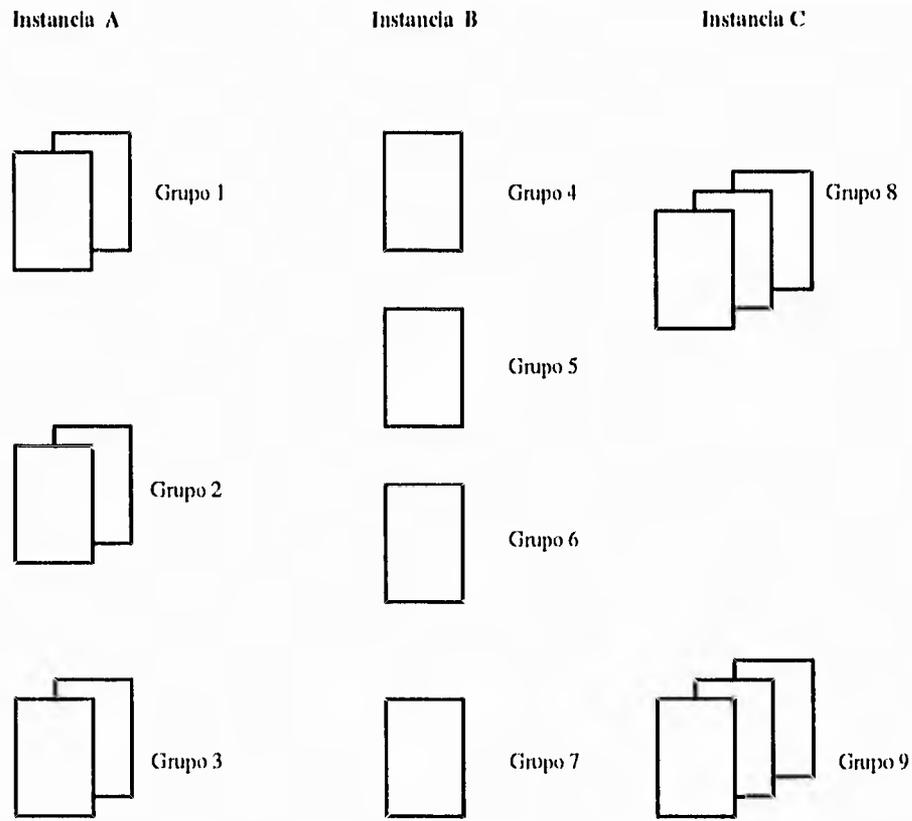
Al archivo 5 se le han asignado también 500 PCM locks y tiene 102400 bloques; por lo que cada PCM lock cubrirá 204 ó 205 bloques aproximadamente.

**Los archivos de log en una Base de Datos para operar en modo paralelo**

Cada instancia en una configuración con el Parallel Server de Oracle, tiene su propio conjunto de archivos de log. Estos archivos de log deberán estar en los discos compartidos del cluster.

Cada instancia debe tener al menos dos archivos de log por la manera de trabajar del RDBMS de acuerdo a lo que se ha explicado en el capítulo de características tecnológicas de Oracle7. Estos mismos archivos de log pueden estar en espejo para evitar la pérdida de alguno de ellos. Es decir, en caso de que alguien borre alguno de los archivos de log que están en línea, la operación de la Base de Datos se suspendería y la consolidación de la información que se encontraba en ese archivo de log se perdería y no habría forma de recuperarse.

Cada instancia en la configuración del Cluster o Parallel Server puede tener distintos grados de duplicación o de realizar el espejo de los archivos de log. Por ejemplo, una instancia puede tener tres archivos de log, donde cada uno se considera un grupo; y cada grupo tiene dos miembros, una segunda instancia tiene cuatro archivos de log que no están en espejo y la tercera instancia puede tener dos grupos con tres miembros cada uno, como se muestra en la siguiente figura :



En esta configuración del Parallel Server, cada instancia escribe en sus propios archivos de log. El conjunto de archivos de log que pertenecen a una instancia se llama **THREAD**. Cada archivo de log está asociado con un número particular de "thread"

De acuerdo con el ejemplo y figura anterior, la instancia A usa el "thread" 1, que contiene tres grupos de archivos de log (grupo 1, 2 y 3). Este conjunto de archivos de log está configurado en espejo para cada uno de sus miembros. La instancia B usa el "thread" 2, que contiene cuatro miembros de archivos de log (grupo 4, 5, 6 y 7) que no están configurados en espejo y la instancia C usa el "thread" 3 que contiene dos grupos de archivos de log (grupo 8 y 9) y existen dos archivos adicionales configurados en espejo para cada archivo de log del grupo.

---

Los números de los grupos deben ser únicos en la Base de Datos. Por ejemplo, en la figura anterior la instancia A o thread 1 contiene los grupos 1, 2 y 3, la instancia B contiene los grupos 4, 5, 6, y 7 y la instancia C contiene los grupos 8 y 9. La vista V\_SLOGFILE que le pertenece al usuario SYS de la Base de Datos contiene la información de los números de los grupos asociados a cada archivo de log.

Cada "thread" debe tener al menos dos archivos de log, o especificar grupos de archivos en espejo como copias en varios dispositivos.

Al momento de crear la Base de Datos con el comando CREATE DATABASE los archivos de log que se mencionan en ese comando se asignan automáticamente al "thread" 1 y quedan habilitados para esa instancia. Los conjuntos de archivos de log subsecuentes se deberán asignar con el comando ALTER DATABASE. Por ejemplo, el siguiente comando crea el conjunto o "thread" 2 con dos grupos de archivos y tres miembros cada uno, en este caso se crean en los discos DISKSSISTEMAS, DISK\$ARCHIVOS y SYSSSYSDEVICE.

```
SQLDBA> ALTER DATABASE ADD LOGFILE THREAD 2
        GROUP 4 ( DISKSSISTEMAS:|ORACLE7.LOGS|ORA_LOG4.RDO,
                 DISK$ARCHIVOS:|ORACLE7.LOGS|ORA_LOG4.RDO,
                 SYSSSYSDEVICE:|ORACLE7.LOGS|ORA_LOG4.RDO ) SIZE 1M
        GROUP 5 ( DISKSSISTEMAS:|ORACLE7.LOGS|ORA_LOG5.RDO,
                 DISK$ARCHIVOS:|ORACLE7.LOGS|ORA_LOG5.RDO,
                 SYSSSYSDEVICE:|ORACLE7.LOGS|ORA_LOG5.RDO ) SIZE 1M;
```

```
SQLDBA> ALTER DATABASE ENABLE PUBLIC THREAD 2;
```

En caso que no se quieran tener copias en espejo de los archivos de log, se puede omitir la cláusula GROUP y el comando sería de la siguiente manera:

```
SQLDBA> ALTER DATABASE ADD LOGFILE THREAD 2  
          'DISKSSISTEMAS:|ORACLE7.LOGS|ORA_LOG4.RDO' SIZE 1M,  
          'DISKSSISTEMAS:|ORACLE7.LOGS|ORA_LOG5.RDO' SIZE 1M;
```

```
SQLDBA> ALTER DATABASE ENABLE PUBLIC THREAD 2;
```

De esta forma se tienen activados dos conjuntos de archivos de log o thread's, lo que indica que se pueden habilitar hasta dos instancias en este momento, ya que el primer grupo es el que se activó al momento de crear la Base de Datos y fue el "thread" 1. En caso que se tengan más nodos o instancias que se activarán en la configuración del Cluster, se deberán crear los "thread's" necesarios para poder trabajar a través de dichas instancias.

### **Los segmentos de rollback en una Base de Datos en modo paralelo.**

Un segmento de rollback contiene información que se está modificando y no se le ha dado commit o rollback a la transacción; esta información sirve para mantener la consistencia en la lectura de la Base de Datos. Existe al menos un segmento de rollback que aparece automáticamente cuando se crea la Base de Datos, este segmento de rollback se llama SYSTEM. En una configuración del Parallel Server este segmento es común a todas las instancias que se activan en el cluster, pero cada instancia requiere al menos un segmento de rollback dedicado o privado.

Existen dos tipos de segmentos de rollback, públicos y privados. Los segmentos públicos serán utilizados por cualquier instancia que requiera de algún segmento disponible, pero los segmentos que han sido creados como privados solo podrán ser activados y utilizados por la instancia que los creó.

---

En esta configuración del Parallel Server de Oracle, se requieren tantos segmentos de rollback como el número de instancias que accederán a la Base de Datos sin contar el segmento de rollback SYSTEM. Es decir, si el cluster cuenta con tres nodos, serán tres las instancias a través de las cuales se manipulará la Base de Datos y por lo tanto se requieren al menos tres segmentos de rollback sin contar con el de SYSTEM. Estos segmentos de rollback que se requieren deben ser al menos uno por instancia y pueden ser públicos o privados.

Los segmentos de rollback pueden crearse en cualquier tablespace y si no se cuenta con suficientes segmentos de rollback en línea no se podrá dar de alta la Base de Datos en modo paralelo; por lo tanto se deben crear y poner en línea estos segmentos como se indica a continuación :

```
SQLDBA> connect internal;  
SQLDBA> select segment_name, status from dba_rollback_segs;
```

SEGMENT_NAME	STATUS
SYSTEM	ONLINE

```
SQLDBA> Create public rollback segment RBS1;  
SQLDBA> Create public rollback segment RBS2;  
SQLDBA> Create public rollback segment RBS3;  
SQLDBA> Alter rollback segment RBS1 on line;  
SQLDBA> Alter rollback segment RBS2 on line;  
SQLDBA> Alter rollback segment RBS3 on line;
```

```
SQLDBA> select segment_name, status from dba_rollback_segs;
```

SEGMENT_NAME	STATUS
SYSTEM	ONLINE
RBS1	ONLINE
RBS2	ONLINE
RBS3	ONLINE

---

Para este ejemplo se crean tres segmentos de rollback llamados RBS1, RBS2 y RBS3. En el esquema de la Facultad de Ingeniería se crearon 5 segmentos de este tipo; pero adicionalmente se crearon los segmentos RBS6 para la instancia en el nodo ATZIN y el segmento privado RBS7 para el nodo KELEM.

```
sql> connect internal;  
sql> create private rollback segment RBS6  
      storage ( initial 1M next 1M pctincrease 0 )  
      tablespace rollbacks;  
sql> alter rollback segment RBS6 online;
```

En caso de no tener los suficientes segmentos de rollback como instancias se pretendan activar, no se podrá levantar la Base de Datos en modo paralelo.

Para una mayor información sobre la manera de crear segmentos de rollback se puede consultar el manual de Oracle "SQL Language"; ya que se puede indicar algún tablespace particular para cada segmento.

**Como dar de alta y baja la Base de Datos en modo paralelo.**

En esta sección se describe como dar de alta, de baja y modificar los parámetros de inicialización de la Base de Datos.

Los pasos necesarios para dar de alta una Base de Datos son los siguientes :

1. Ejecutar los procedimientos de comandos ORAUSER.COM y ORA\_DB.COM que se encuentran en el directorio de administración de la Base de Datos que está debajo del directorio de Oracle7 y se reconoce por tener el nombre DB\_<dbname>.dir

```
$@DISKSSISTEMAS:|ORACLE7.DB_DEMO|ORAUSER_DEMO.COM
```

```
$@DISKSSISTEMAS:|ORACLE7.DB_DEMO|ORA_DB_DEMO.COM
```

Para este ejemplo suponemos que se tiene un usuario llamado ORACLE7 que tiene su directorio de entrada al Sistema Operativo en el dispositivo disk\$systemas y que la Base de Datos con la que estamos trabajando se llama DEMO.

2. Invocar SQL\*DBA y activar la Base de Datos que en este caso se llama DEMO. Si se indica el parámetro lmode=y, sqldba no presentará el menú de administración de la Base de Datos.

```
$ sqldba lmode=y  
SQLDBA > connect internal;  
SQLDBA> startup open DEMO;
```

Cuando se crea la Base de Datos con el procedimiento de instalación se levanta la Base de Datos en modo exclusivo, es decir solamente una instancia podrá hacer referencia a los archivos físicos de la Base de Datos; y por lo tanto no se puede trabajar en un ambiente Cluster.

Existen un par de procedimientos de comandos que permiten activar la Base de Datos en modo exclusivo, o en modo paralelo (modo para poder trabajar con el Cluster). Dependiendo de la forma en la que se desea operar con la Base de Datos es el procedimiento que se ejecutará, para el caso de una instalación en la que la Base de Datos se llama DEMO se ejecutarían de la siguiente manera :

```
S@DISKSSISTEMAS:[ORACLE7.DB_DEMO]STARTUP_EXCLUSIVE_DEMO.COM
```

o

```
S@DISKSSISTEMAS:[ORACLE7.DB_DEMO]STARTUP_PARALLEL_DEMO.COM
```

Cuando una instancia se da de alta con el comando STARTUP, usa un archivo de parámetros llamado INIT.ORA, y se hace referencia a él con el valor del nombre lógico ORA\_PARAMS, el cual indica el nombre completo de este archivo ya que dicho nombre se define al ejecutar el archivo de comandos que para este ejemplo se llamaría : DISKSSISTEMAS:[ORACLE7.DB\_DEMO]ORA\_DB\_DEMO.COM, debido a que la instalación del software de Oracle7 esté en el dispositivo y directorio que se ha indicado y la Base de Datos se llama DEMO. A través de este archivo de parámetros Oracle determina el tamaño de memoria principal que ocupará el SGA (System Global Area) para la instancia. Cada instancia en una configuración del Parallel Server de Oracle pueden tener distintos archivos de configuración o el mismo para todas ellas.

En caso de que se hayan definido segmentos de rollback privados, cada instancia deberá tener su propio archivo INIT.ORA de configuración e indicar con el parámetro ROLLBACK\_SEGMENTS los segmentos que se activarán para esa instancia

---

## **ARQUITECTURA Y CONFIGURACION** **DEL VAX - CLUSTER**

Un Cluster es una integración de software y dispositivos de almacenamiento que permiten habilitar un sistema unificado con el fin de compartir recursos de procesamiento, almacenamiento de datos y unidades de respaldo.

Digital Equipment Corporation diseñó el sistema OpenVMS Cluster por las siguientes razones principalmente :

- Permitir separar CPU's y dispositivos de almacenamiento para compartir archivos de datos.
  
- Permitir alta disponibilidad para las aplicaciones y datos.
  
- Permitir un fácil crecimiento en la implementación de un equipo de cómputo.
  
- Proveer un seguro en la inversión actual con los sistemas futuros.
  
- Incrementar la productividad, ya que varios sistemas pueden ser manejados como si fuera uno solo y esto provee una mayor capacidad del poder de cómputo que con sistemas independientes o aislados.

El Sistema Operativo que se emplea es OpenVMS y se cuenta además con un manejador de candados distribuidos que trabaja de manera prioritaria para controlar el acceso a los recursos compartidos.

Los recursos típicos a compartir son los archivos de datos; esto indica que se debe mantener un extremo cuidado en este proceso de administración ya que desde distintas máquinas que conforman el Cluster se tendrá acceso a cada archivo. Recursos que serán manipulados aun cuando el usuario no se entere que algún otro usuario podría utilizar el mismo recurso que él tiene en ese momento.

### **Antecedentes.**

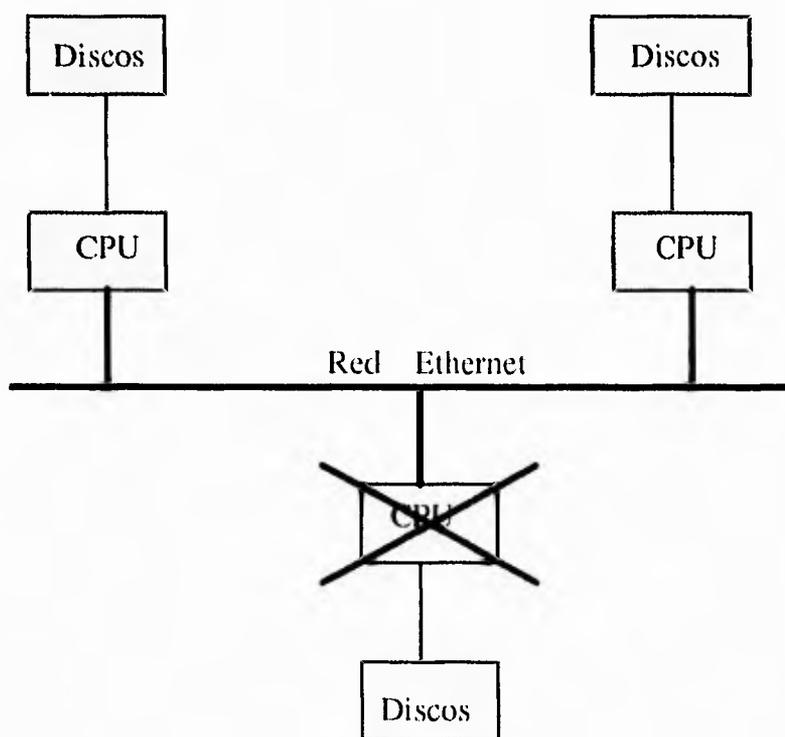
A principios de la década de los 80 comenzaron a existir centros de cómputo en los que se tenían activas distintas máquinas para procesamiento de información, pero éstas no compartían recursos a menos que realmente se conectaran en una red local. Como respuesta a las necesidades de alta disponibilidad, seguridad y para compartir información, Digital desarrolló e implementó una arquitectura en la que se permite interconectar distintos CPU's de manera independiente pero que pueden compartir dispositivos de almacenamiento en un mismo sistema llamado VAXCluster.

La implementación de esta configuración se realiza a través de una red de área local (LAN) con una comunicación por lo general de tipo Ethernet. Esta configuración es muy adecuada para aquellos ambientes en los que se tienen distintos grupos de usuarios que demanden recursos de procesamiento y memoria en equipos independientes pero requieren compartir información de un sólo conjunto de datos, de esta forma se llega a manifestar una alta disponibilidad y escalabilidad al distribuir los usuarios y procesos en equipos que se comportan como independientes pero que comparten la misma fuente de datos.

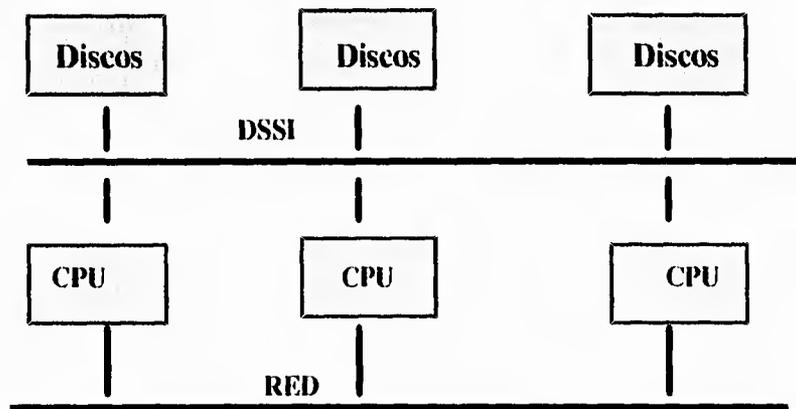
**Características.**

La característica más importante del software que administra el Cluster es la disponibilidad de permitir el acceso transparente a los dispositivos de almacenamiento y archivos.

En un esquema distribuido tradicional, cada equipo es una unidad independiente en las que si algún nodo se dá de baja, los discos de dicho nodo no pueden ser accedidos por los equipos que están activos. Es por esto que este tipo de configuración tiene una desventaja con respecto a la necesidad y disponibilidad para acceder la información cuando se presentan los casos de inoperabilidad o falla de alguna unidad central de proceso (CPU).



En contraste con el uso de algún modelo distribuido tradicional se propone la comunicación a través de un bus común de I/O que permita mostrar y acceder todos los dispositivos de almacenamiento desde cualquier equipo como si fueran propios, sin impedir el acceso a la información cuando algún nodo se dé de baja.



Existen cuatro formas de realizar la comunicación y son conocidas con las siglas : *CI*, *DSSI*, *Ethernet* y *FDDI*.

---

### Estructura del Hardware de un VAX Cluster

Es importante comentar acerca de los equipos y modelos de DIGITAL que se pueden utilizar para formar un VAXCluster.

La arquitectura **Alpha AXP** está basada completamente en procesadores RISC (Reduced Instruction Set Computing) de 64 bits, basado en el chip de Digital Alpha21064. Estos equipos se han combinado con una serie de subsistemas de gran velocidad de entrada/salida, gran capacidad y acceso a memoria, así como de dispositivos de almacenamiento masivo.

La arquitectura **VAX** está basada en procesadores CISC (Complex Instruction Set Computing) de 32 bits. Estos equipos han sido difundidos desde 1978 y actualmente existen varios de éstos instalados alrededor del mundo.

Debido a los avances tecnológicos ahora se requiere mantener ambas arquitecturas y proveer un mecanismo de comunicación entre ellas si es que se desea formar un Cluster entre arquitecturas distintas. Con el Sistema Operativo OpenVMS desarrollado por Digital, se permite mantener una compatibilidad entre aplicaciones y comunicación entre máquinas VAX y AXP.

A continuación se explican las opciones existentes de comunicación entre los distintos equipos que pertenecen a las arquitecturas VAX y AXP. Con el Sistema Operativo OpenVMS Cluster se soportan las siguientes conexiones : Computer Interconnect (CI), Digital Storage Systems Interconnect (DSSI), Ethernet y Fiber Distributed Data Interface (FDDI).

## 1. Configuración basada en CI-bus

Los elementos de esta arquitectura son : el acoplador estrella (star coupler), el puerto CI (CI port) y cable CI.

Los siguientes equipos son los que se pueden conectar bajo una arquitectura cluster con una configuración CI :

- Series DEC 7000 y DEC 10000
- Series VAX 6000, VAX 7000, VAX 8000, VAX 9000 y VAX 10000
- Series VAX 11/750, VAX 11/780 y VAX 11/785

Se maneja una filosofía de paso de mensajes a una alta velocidad entre los equipos interconectados, esto es conocido como canal CI o bus CI (Computer Interconnect). Se tiene una interface inteligente para el bus CI, llamada puerto CI (CI port) y un controlador inteligente que usa tanto la interfase del bus CI y del puerto CI para los dispositivos de almacenamiento (típicamente cintas y discos).

Para formar un cluster se requieren al menos dos equipos de cómputo o CPU's y se pueden incluir dos unidades de almacenamiento que serán uno de los recursos a compartir. Vale la pena mencionar que regularmente se utiliza un LAT server (Local Area Transport) para la conexión de las terminales.

El bus CI es una conexión dual y serial que soporta una transferencia de 70Mbits por segundo, se realiza a través de una implementación de dos cables coaxiales, uno para recibir y otro para la transmisión de señales. Lógicamente se considera al bus CI como un canal de comunicación, pero su implementación física es mediante una topología de estrella apoyándose con un hub central llamado Acoplador Estrella (Star Coupler) que conecta todos los nodos en un radio de hasta 45 metros.

Se seleccionó la topología de estrella debido a la eficiencia lograda con la comunicación serial entre nodos y adicionalmente el Central Coupler provee una simplicidad eléctrica y mecánica para añadir o retirar nodos de la estructura del Cluster.

El puerto CI es un adaptador inteligente que el microcódigo con el que cuenta permite comunicar equipos VAX y AXP.

## **2. Configuración basada en Digital Storage System Interconnect (DSSI).**

El DSSI es una alternativa de comunicación entre distintos CPU's a través de un canal de comunicación. Este bus es el único medio para la conexión de todos los dispositivos de almacenamiento y CPU's. Esta es la configuración que se tiene implementada en la USECAFI, como se explica en el capítulo 2.

Las siguientes máquinas pueden ser configuradas bajo esta arquitectura :

- DEC 4000 modelo 600 AXP server.
- DEC 7000 AXP y DEC 10000 AXP
- VAX 4000, VAX 6000, VAX 7000 y VAX 10000
- VAX 9000
- Sistemas Q-bus VAX y Micro VAX

La longitud del cable puede ser de 6 metros y se llega a tener una comunicación de hasta 32 Mbits por segundo. El adaptador con el que se cuenta es el que realiza la liga entre los equipos y dispositivos reduciendo significativamente el uso y overhead del CPU.

### **3. Configuración basada en Ethernet.**

En este caso el CI bus se ha sustituido por una configuración coaxial y los nodos VAX son MicroVAX o workstations, esto quiere decir que cualquier equipo DEC puede conectarse en cluster basado en una configuración Ethernet. Dichos nodos actúan como los propietarios de los dispositivos de almacenamiento que se compartirán entre los equipos que conforman el Cluster y todos los nodos podrán acceder y utilizar los discos que físicamente se tienen en los otros equipos.

Los límites de cables de conexión es de 2800 metros y la velocidad de transmisión es de 10 Mbits por segundo.

### **4. Fiber Distributed Data Interface (FDDI).**

Esta es una conexión a través de fibra óptica. El diseño de una red basada en FDDI representa actualmente el medio más avanzado y rápido de comunicación desarrollado por la ANSI (American National Standards Institute).

---

Los límites de conexión es de 40 kilómetros y se llega a tener una velocidad de hasta 100 Mbits por segundo. Básicamente se pueden comunicar los siguientes modelos de máquinas :

- VAX 6000, VAX 7000, VAX 8000, VAX 9000 y VAX 10000.
- DEC 3000, DEC 7000 y DEC 10000
- VAX Q-bus y Micro VAX
- VAXstation 4000 modelo 90

**Software del VAXCluster.**

Con el Sistema Operativo de Digital OpenVMS para una configuración en Cluster se tienen las siguientes características :

- Uso de manera compartida de dispositivos y archivos.
- Acceso controlado por usuarios.
- Uso de forma compartida de colas de impresión y batch.
- Administración unificada de los equipos.

Una de las características más importantes del Sistema Operativo OpenVMS para la administración de los recursos y procesos del Cluster es el manejador de candados distribuidos (Distributed Lock Manager), que se explica a continuación.

**OpenVMS Distributed Lock Manager.**

El manejador de candados distribuidos es el responsable de coordinar y sincronizar los recursos de los procesos a través del Cluster. Cuando una aplicación está trabajando en un ambiente Cluster, el lock manager envía el nombre del recurso y el modo de acceso como parámetro al kernel del Sistema Operativo. El nombre del recurso le permite al Sistema Operativo identificarlo como único para bloquearlo; por otra parte el modo de acceso puede ser de sólo lectura o compartido. El kernel validará los parámetros y modos de acceso para bloquear el recurso si es posible.

---

El hecho de estar trabajando en un ambiente de este tipo, en el que se está separando el proceso en varias máquinas que conforman el Cluster, podría hacernos pensar en una degradación del performance dado que el lock manager tendrá que estar enviando mensajes para mantener la comunicación y coordinación de recursos entre los nodos, pero no es así ya que se ha demostrado y es un punto de interés a considerar, que cuando se aumenta el número de nodos en una configuración con OpenVMS Cluster el número de mensajes que se manejan para la coordinación de los recursos en los nodos tiende a permanecer constante.

El lock manager opera directamente con el RMS (Record Management System) coordinando el acceso a los registros de los archivos. Como es sabido el RMS administra la estructura de archivos y directorios en un ambiente OpenVMS y maneja distintos tipos de archivos como secuenciales, indexados y de acceso directo; RMS conjuntamente con el lock manager permiten un acceso eficiente y seguro a los archivos de una manera compartida para que sean utilizados desde cualquier nodo que pertenezca al Cluster.

Estas características del Sistema Operativo han permitido a otros proveedores desarrollar aplicaciones y software complementario bajo esta arquitectura, como es el caso de Oracle que con la tecnología del "Parallel Server" (explicada en el capítulo 6) permite utilizar esta configuración de manera eficiente, lo cual hemos podido consolidar con la implementación que se ha propuesto para la administración de la información en la Facultad de Ingeniería.

### Configuración e implementación del Cluster

Los equipos con los que se cuenta en la Unidad de Servicios de Cómputo para la Administración en la Facultad de Ingeniería (USECAFI) para la implementación del Cluster son una VAX6000/210 y una MicroVAX3400, esta configuración está basada en la tecnología DSSI (Digital Storage System Interconnect) ya que este método es el que permite establecer la comunicación óptima entre los equipos y dispositivos periféricos que se tienen.

Para realizar la configuración se deben tomar en cuenta una serie de conceptos relacionados con los parámetros del Sistema Operativo para la correcta operación del Cluster.

El Sistema Operativo cuenta con un software llamado *connection manager* que corre en cada nodo activo del Cluster para coordinar el estado y buen funcionamiento de cada miembro de éste cuando se añade o se retira algún nodo. El *connection manager* realiza las siguientes actividades :

- Determina qué nodos forman el Cluster.
  
- Construye y mantiene las estructuras de datos necesarias para asegurar una vista uniforme del Cluster y sus recursos en cada nodo.
  
- Usa un esquema llamado "quorum" que le permite prevenir el particionamiento o fragmentación del Cluster.

El **Quorum** del Cluster es un valor dinámico calculado por el *connection manager* para prevenir la fragmentación o partición del Cluster.

A cada miembro del Cluster se le asignan un número fijo de votos que contribuyen al quorum. Para tal efecto, cada nodo que se añada al Cluster aportará un cierto número de votos. Se conocen como nodos satélites aquellas máquinas que no tienen Sistema Operativo propio para levantar o activar la máquina, es decir que no pueden realizar el proceso de "boot" independientemente. Los nodos no satélites son los que cuentan con su propio Sistema Operativo.

- Un nodo satélite tiene un valor de 0 votos por default.
- Un nodo no satélite tiene un valor de 1 voto por default.

Durante un estado de transición en el Cluster, es decir, cuando se activa o desactiva un nodo, el *connection manager* totaliza el número de votos de todos los miembros presentes y compara este valor con el del "quorum" para determinar las acciones a realizar.

- El Cluster funciona sí el número total de votos es al menos el valor del "quorum".
- Sí el total de votos es menor que el quorum del Cluster, se suspende el servicio hasta que haya suficientes votos para continuar.
- Un estado de transición en el Cluster ocurre cuando se añade o se retira un nodo.

El valor del "quorum" del Cluster es inicialmente el del parámetro EXPECTED\_VOTES del SYSGEN. Este valor le servirá para calcular el mínimo de votos necesarios para garantizar el funcionamiento y evitar que ocurra una fragmentación o partición del Cluster.

- El System Manager debe poner el valor de EXPECTED\_VOTES como la suma de votos necesarios para los miembros potenciales que formarán el Cluster.

- El valor inicial del "quorum" del Cluster es calculado usando la siguiente fórmula :

$$(EXPECTED\_VOTES + 2) / 2$$

Cuando ocurre un estado de transición en el Cluster, el *connection manager* recalcula el valor del "quorum" y le asigna el valor máximo que se obtenga de las siguientes condiciones :

- El valor actual del quorum del Cluster.

- El valor entero de  $(EXPECTED\_VOTES + 2) / 2$  del nodo principal, es decir del que realiza el arranque del sistema.

- El valor entero de  $(V + 2) / 2$ , donde V es el número de votos de los nodos presentes.

Para habilitar un VAXCluster es necesario tomar en cuenta y llevar a cabo los siguientes pasos :

- Asegurarse que todos los nodos y dispositivos están conectados y operando de manera adecuada.

- Instalar o actualizar la versión del Sistema Operativo más reciente en el disco del nodo que servirá para el arranque del sistema.
  
- Configurar el nodo principal durante la instalación o actualización del Sistema Operativo. En caso que ya esté instalado, solo se deberá configurar el nodo con el procedimiento de comandos CLUSTER\_CONFIG.COM.
  
- Configurar el software de DECnet y arrancar la red desde el nodo principal.
  
- Añadir los nodos satélites con la opción ADD del procedimiento CLUSTER\_CONFIG.
  
- Arrancar todos los nodos que pertenecen al Cluster. Se invocará automáticamente el AUTOGEN la primera vez para cada miembro del Cluster, con el fin de adecuar los parámetros seleccionados con el procedimiento de comandos CLUSTER\_CONFIG.
  
- Configurar los dispositivos, discos y cintas que se necesiten para el Cluster.
  
- Crear o modificar los procedimientos de comandos específicos de cada miembro del Cluster.

El Cluster que se ha formado en la USECAFI para los fines académico - administrativos es entre los nodos ATZIN y KELEM, es decir la VAX6000/210 y la MicroVAX3400. En estos equipos se mantiene la Base de Datos de la Facultad de Ingeniería con Oracle7 y desde cualquiera de los equipos se puede acceder a la información, lo que ha permitido una distribución de procesos.

---

El primer inconveniente que se ha manifestado es la falta de memoria principal (RAM) en el nodo KELEM ya que sólo tiene 12 Mbytes de RAM y esto ha originado una serie de inconvenientes; el principal es que no se pueden tener varios usuarios simultáneos ya que la máquina registra un bajo rendimiento. Si se aumenta la memoria principal en ambas máquinas con una configuración en ATZIN de 96Mbytes en RAM y en KELEM con 32Mbytes de RAM al menos, el rendimiento se vería incrementado notablemente. De hecho, estos equipo deberían ser utilizados solamente para un ambiente de producción o desarrollo y pensar en adquirir otro equipo con los recursos adecuados para que se realicen las otras tareas de producción o desarrollo, según sea el caso.

Es recomendable que para realizar el trabajo de configuración del Cluster se cuente con personal que conozca de la administración del equipo de cómputo, es decir el System Manager, ya que al ejecutar los procedimientos se modifican varios parámetros del SYSGEN y es conveniente que se contemplen las posibles fallas y cómo actuar en caso de ocurrir algún error.

Para una información complementaria sobre la operación y funcionalidad a nivel administración del los equipo Digital y en especial con el VAXCluster, se pueden consultar los manuales "VMS VAXCluster Manual" , "VAX System Generation Utility" y "VAX System Manager's Manual".

## **CASO PRACTICO**

Para llevar a cabo la implementación de un sistema con las características y esquemas planteados en los capítulos precedentes se ha propuesto la revisión del esquema del flujo de datos y el modelo de entidades y relaciones de la Base de Datos de la Facultad de Ingeniería y la implementación del Sistema de Exámenes Extraordinarios; por lo que en este capítulo se explicará la manera en que se llegó al esquema de la Base de Datos actual y como se desarrolló el Sistema de Exámenes Extraordinarios.

### **Primera Parte:**

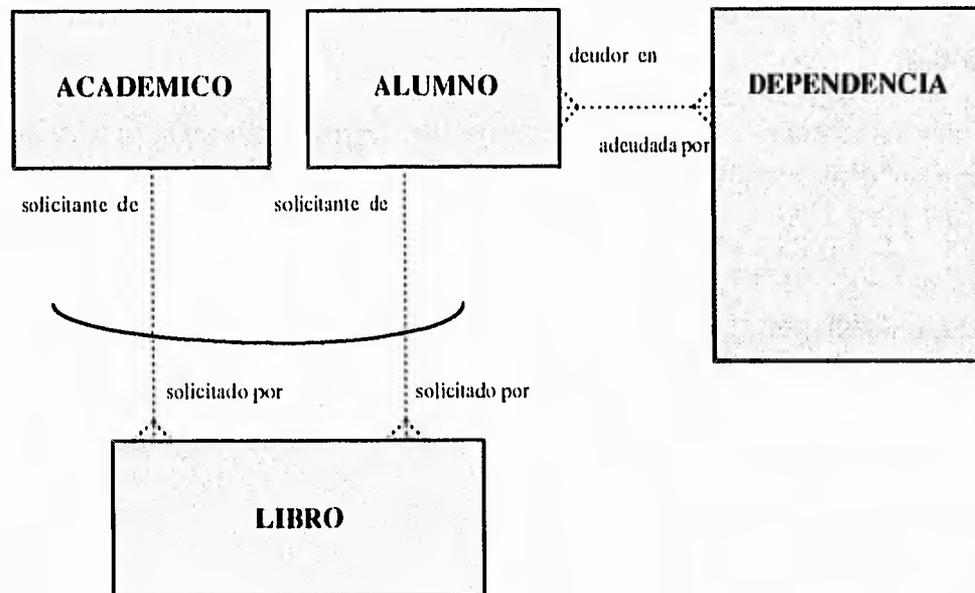
#### **Proceso de Desarrollo de la Base de Datos.**

El desarrollo de la Base de Datos es un enfoque top-down, que transforma los requerimientos de información en una Base de Datos operacional. Estos son los pasos que se siguieron para llegar al modelo conceptual de la información, los que se explicarán durante este capítulo.

El desarrollo top-down de la Base de Datos comienza con los requerimientos de información del negocio. Habrá que tener en cuenta que el alcance de un conjunto de requerimientos varía. Pueden ser diferentes las necesidades de cada departamento a las necesidades de toda la compañía que en este caso nos estamos refiriendo a la Facultad de Ingeniería de la UNAM. Este proceso es ejemplificado en la siguiente figura :



El modelo de datos conceptual define y modela los aspectos importantes acerca de la información que el negocio necesita saber o tener y las relaciones entre dicha información. Para este planteamiento surge el modelado de entidades y relaciones que debe contemplar adecuadamente las necesidades de información de la organización y soportar la funciones propias que la sustentan. Por ejemplo la siguiente figura muestra el modelo entidad - relación del Sistema de la Biblioteca:



Los componentes del modelo entidad - relación son :

- **ENTIDADES** - Son los aspectos importantes acerca de los cuales se necesita información (ACADEMICO, LIBRO, etc.)
- **RELACIONES** - Como se relacionan las entidades (solicitante de, deudor en, solicitado por, etc.)
- **ATRIBUTOS** - Información específica, la cual necesita ser almacenada (nombre, dirección, título, autor, etc. )

Un modelo entidad - relación es una forma efectiva para integrar y documentar los requerimientos de información de una Organización y mantiene los siguientes características :

**Sintaxis:**

Un modelo entidad - relación documenta los requerimientos de información de la organización en un formato preciso y claro.

**Comunicación con el usuario:**

Los usuarios pueden entender fácilmente la forma gráfica de un modelo entidad - relación.

**Fácil de desarrollar :**

Un modelo entidad - relación puede ser fácilmente desarrollado y refinado.

**Definición del alcance:**

Un modelo entidad - relación provee una clara imagen del alcance de los requerimientos de información de las organizaciones.

**Integración de múltiples aplicaciones :**

Un modelo entidad - relación nos provee una estructura adecuada para la integración de múltiples aplicaciones, desarrollar proyectos y/o paquetes de aplicación.

Este modelo conceptual es independiente del hardware o del software usados para la implementación de sistemas de información.

---

---

Es por estos motivos que el desarrollar un buen modelo nos ha llevado a la pronta implementación exitosa y a sustentar las bases para futuras adecuaciones o variaciones del hardware o del software que se pretenda emplear. Para el caso que aquí nos compete se ha definido ya el ambiente VAX-Cluster como plataforma de hardware y el manejador de Bases de Datos Oracle7; pero aún así este modelo podrá ser utilizado si estas condiciones cambian.

### **Entidades:**

Como se ha comentado anteriormente, una entidad es un aspecto importante acerca del cual se necesita tener o conocer información y los atributos describen a las entidades, ya que son las piezas específicas de información.

Una entidad debe de tener atributos que necesitan ser conocidos desde el punto de vista de la Organización, de otra manera no es una entidad que forme parte del alcance de los requerimientos del negocio.

Otra característica de las entidades son las ocurrencias o instancias. Cada ocurrencia debe ser identificada como única de otras ocurrencias de la misma entidad. Un atributo o conjunto de atributos que identifican de manera única a una ocurrencia dentro de una entidad son llamados Identificadores Unicos (UID's). Por ejemplo para la entidad ALUMNO se tienen varias ocurrencias, una de ellas sería :

<i>Num. Cuenta</i>	<i>Nombre</i>	<i>Carrera</i>
93136340	Teresa Montoya	32

Para este caso el nombre de la entidad es ALUMNO y se han identificado al menos tres atributos (num. cuenta, nombre y carrera) que son de interés para reconocer esta entidad. Se puede reconocer que el UID de esta entidad puede ser "num. cuenta", ya que identifica de manera única cada ocurrencia.

**Relaciones:**

Una relación es bidireccional y representa la asociación entre dos entidades, o entre una entidad consigo misma. La sintaxis de una relación es como sigue :

CADA	<i>entidad1</i>	DEBE SER		UNA O MAS	
		o	<i>Nombre de la</i>	o	<i>entidad 2</i>
		PUEDA SER	<i>relación</i>	UNA Y SOLAMENTE UNA	

La relación entre un ALUMNO y un LIBRO de la biblioteca de acuerdo al diagrama expuesto anteriormente sería :

CADA ALUMNO PUEDE SER SOLICITANTE DE UNO O MAS LIBROS

CADA LIBRO PUEDE SER SOLICITADO POR UNO Y SOLO UN ALUMNO

Cada dirección de una relación tiene :

Un nombre : Solicitante de; solicitado por.

Una opción : Debe ser o estar; puede ser o estar.

Un grado : Uno y solamente uno; uno o mas.

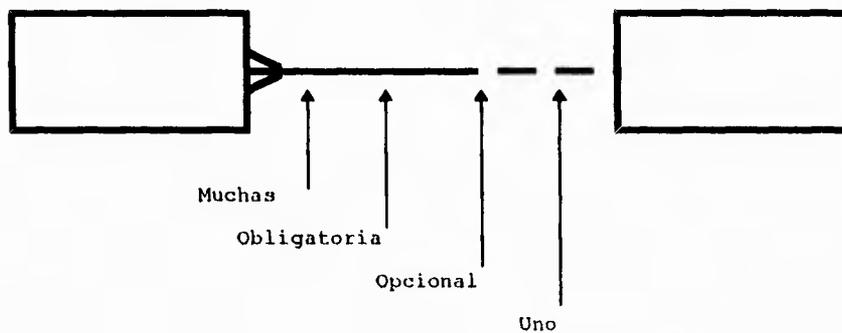
La siguiente figura ilustra los estándares de diagramación para el modelado de entidades y relaciones :

- Una línea entre dos entidades
- Nombres de relaciones en minúsculas
- Opcionalidad

----- Opcional (Puede ser)  
———— Obligatoria (Debe ser)

- Grado

>———— Uno o mas  
———— Uno y solo uno

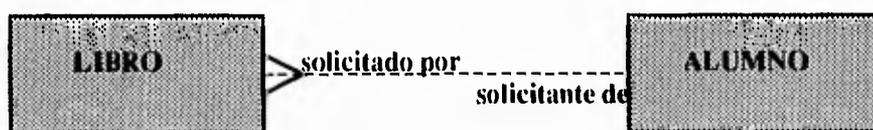


Existen tres tipos de relaciones :

- Relaciones muchos a uno
- Relaciones muchos a muchos
- Relaciones uno a uno

Todas las relaciones deben de representar los requerimientos de información y reglas que sustentan o definen a la Organización.

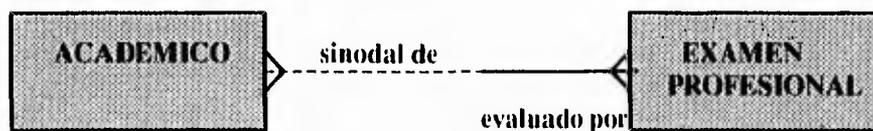
Una relación muchos a uno (M a 1 o M:1) tiene el grado de uno o mas en una dirección y el grado de uno y solamente uno en la otra dirección. Por ejemplo, para la siguiente figura se hace referencia a la relación existente entre las entidades ALUMNO y LIBRO del Sistema de la Biblioteca :



Cada libro puede ser solicitado por uno y solo un alumno.

Cada alumno puede ser solicitante de uno o varios libros.

Las relaciones muchos a muchos (M a M o M:M) tienen el grado de uno o mas en ambas direcciones. Por ejemplo, para la siguiente figura se hace referencia a la relación existente entre ACADEMICO y EXAMEN PROFESIONAL del Sistema de Titulados.



Cada académico puede ser sinodal de uno o varios exámenes profesionales.

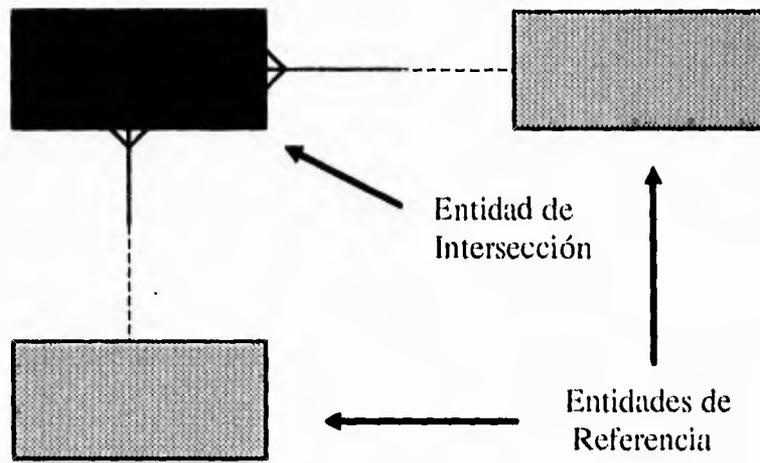
Cada examen profesional debe ser evaluado por uno o varios académicos.

Cuando se tienen relaciones M:M deben resolverse con una entidad de referencia que mantiene la intersección de aquellas que la originaron:

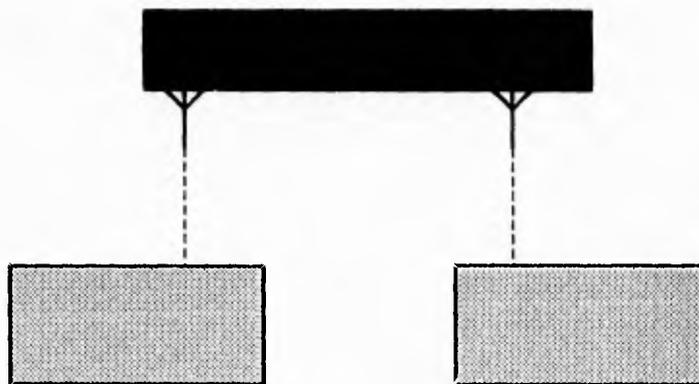
Forma de la relación M:M



Forma de la Entidad de intersección :

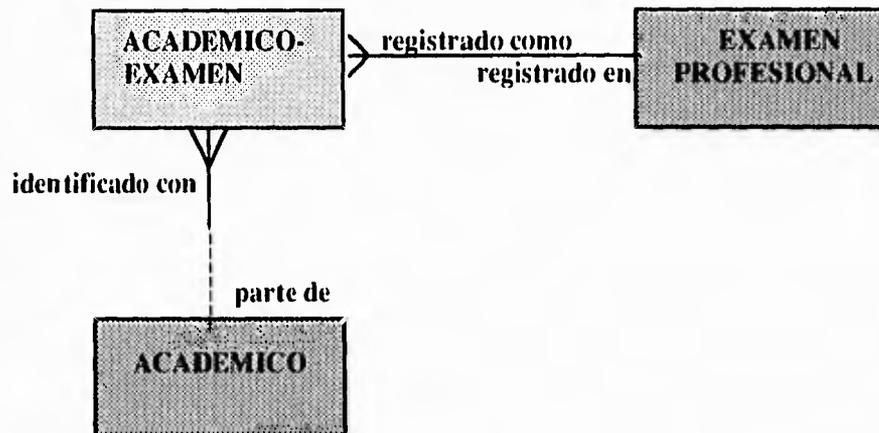


O



J

Por lo tanto para el ejemplo del Sistema de Titulados, las entidades de referencia son ACADEMICO y EXAMEN PROFESIONAL; y la entidad de intesección es ACADEMICO-EXAMEN.



Como se puede observar en este caso, cuando las relaciones M:M son resueltas, la forma del diagrama puede ser confusa por lo que se recomienda mantener el modelo original en el que se tiene el esquema del que se partió al modelo de entidades y relaciones final.

Una relación uno a uno (1 a 1 ó 1:1) tiene un grado de uno y solamente uno en ambas direcciones.

Estas relaciones son muy raras, ya que pueden ser en realidad la misma entidad y los atributos de una y otra pueden conformar una entidad global que contenga todos los atributos.

---

**Atributos:**

Los atributos son información que se necesita conocer o tener acerca de una entidad. Los atributos describen una entidad para calificar, identificar, clasificar, cuantificar o expresar el estado de la entidad.

Por ejemplo algunos atributos de la entidad ALUMNO son :

Número de cuenta,  
Nombre,  
Dirección, etc.

Para los atributos existen también una serie de consideraciones a tomar en cuenta para llegar a una estandarización en el desarrollo de sistemas:

- Los nombres de los atributos deben ser claros para un usuario, mas no codificado para el desarrollador.
- Los nombres de los atributos deben ser específicos, por ejemplo para una cantidad, es cantidad vendida o cantidad comprada.
- Clarificar siempre un atributo de fecha con una descripción o una frase, por ejemplo fecha de ingreso, fecha de titulación.
- Un atributo preferentemente debe ser asignado a una sola entidad.
- Verificar que un atributo no sea derivado o calculado de los valores existentes de otros atributos. Los datos derivados mas comunes son los contadores, totales, máximo, mínimo, promedio, etc.

Para llevar a cabo un buen modelado de entidades y relaciones, se tienen ciertos criterios adicionales que es conveniente explicar y ejemplificar para que se pueda interpretar el modelo final al que se ha llegado para la Facultad de Ingeniería.

---

---

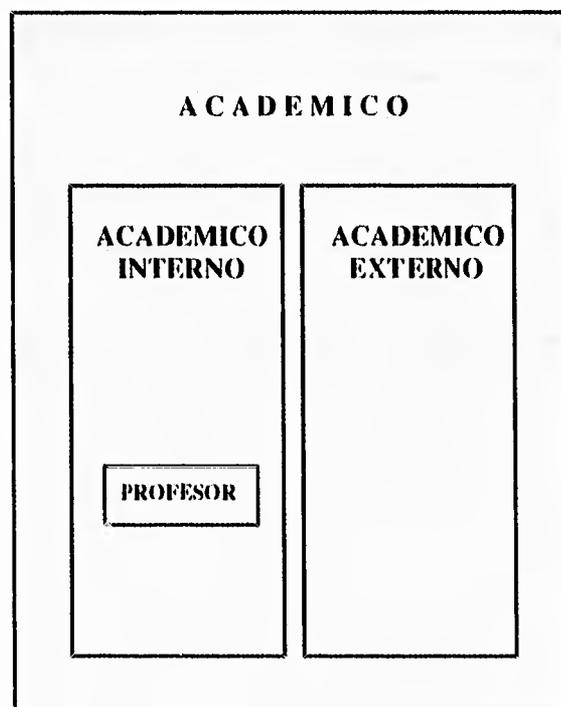
**Modelado de subtipos :**

Los subtipos se emplean para modelar exclusivamente tipos de entidades que tienen atributos o relaciones comunes.

Un supertipo es una entidad que tiene subtipos. Un supertipo puede ser separado en dos o más subtipos mutuamente excluyentes.

Por ejemplo para el caso de la entidad ACADEMICO, se tiene la existencia de personal académico que pertenece a la Facultad de Ingeniería (ACADEMICO INTERNO) y personal de este tipo que pertenece a otra dependencia distinta a la Facultad de Ingeniería (ACADEMICO EXTERNO).

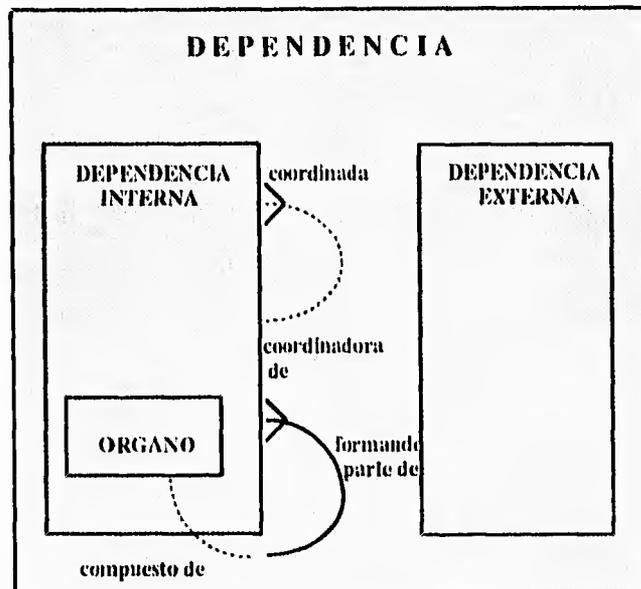
De igual forma un subtipo puede tener subtipos; tal es el caso de la entidad ACADEMICO INTERNO, que tiene personal que imparte clases (PROFESOR) y otros que no imparten clases. Este caso se modela como se indica en la figura.



### Modelado de Relaciones de Modo Recursivo :

Una relación recursiva es la relación de una entidad con ella misma; y para ejemplificarlo veamos la forma en que se modela la entidad DEPENDENCIA, que se refiere a los organismos que componen la estructura funcional tanto de la Facultad de Ingeniería como de otros organismos que cooperan con ésta.

De esta entidad, interesa identificar aquellas que son propias de la Facultad (DEPENDENCIA INTERNA) y las que solo intervienen de manera esporádica para algunas tareas muy específicas (DEPENDENCIA EXTERNA). Así mismo se tiene una sub-entidad o subtipo que es el ORGANO, como se muestra en la figura:



Cada organo puede estar compuesto de una o varias dependencias internas.  
Cada dependencia interna debe estar formando parte de uno y solo un organo.

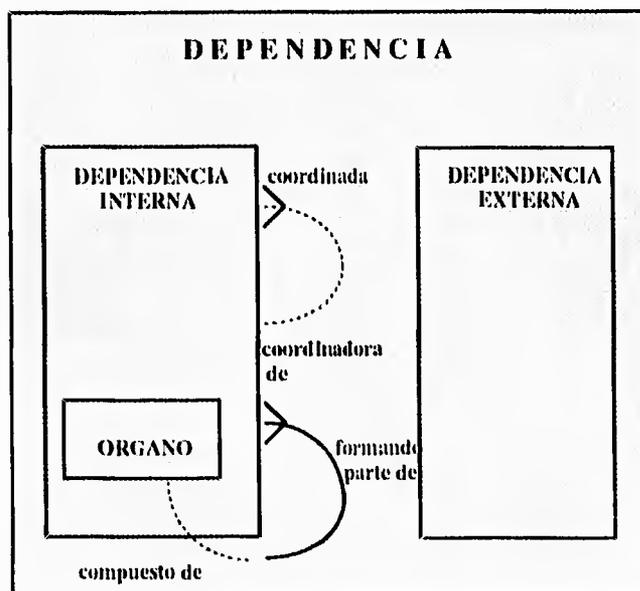
Cada dependencia interna puede ser coordinadora de una o varias dependencias internas.

Cada dependencia interna puede ser coordinada por una y solo una dependencia interna.

**Modelado de Relaciones de Modo Recursivo :**

Una relación recursiva es la relación de una entidad con ella misma; y para ejemplificarlo veamos la forma en que se modela la entidad DEPENDENCIA, que se refiere a los organismos que componen la estructura funcional tanto de la Facultad de Ingeniería como de otros organismos que cooperan con ésta.

De esta entidad, interesa identificar aquellas que son propias de la Facultad (DEPENDENCIA INTERNA) y las que solo intervienen de manera esporádica para algunas tareas muy específicas (DEPENDENCIA EXTERNA). Así mismo se tiene una sub-entidad o subtipo que es el ORGANO, como se muestra en la figura:



Cada organo puede estar compuesto de una o varias dependencias internas.  
Cada dependencia interna debe estar formando parte de uno y solo un organo.

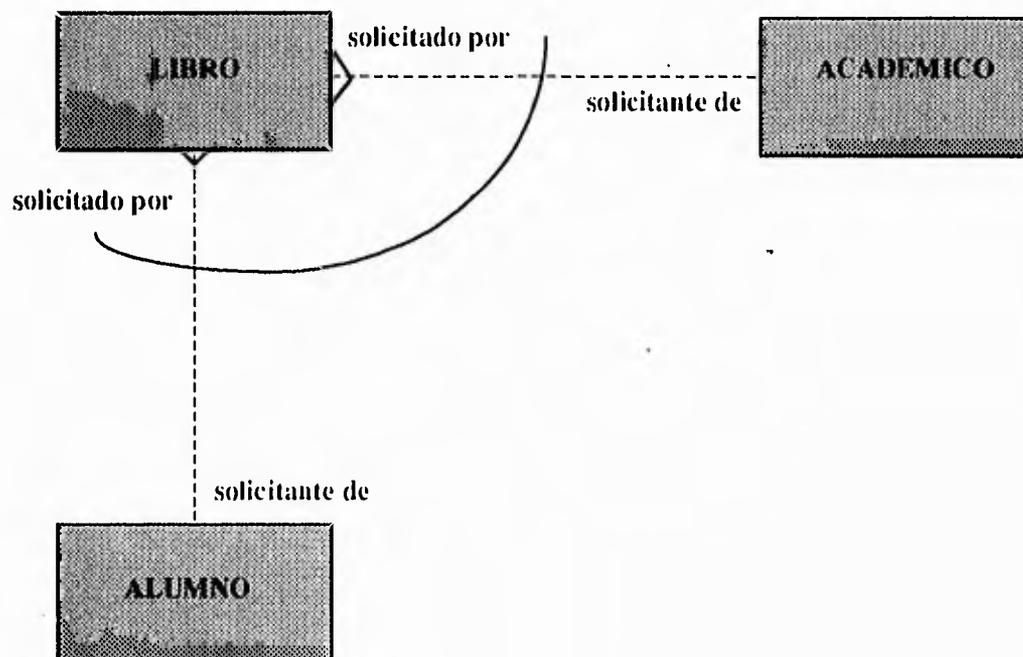
Cada dependencia interna puede ser coordinadora de una o varias dependencias internas.  
Cada dependencia interna puede ser coordinada por una y solo una dependencia interna.

**Modelado de Relaciones Mutuamente Excluyentes:**

Para modelar dos o mas relaciones mutuamente excluyentes de la misma entidad se emplea un arco. Las relaciones en arco frecuentemente tienen los mismos nombres de relación y deben ser o todas obligatorias o todas opcionales.

El arco indica que una relación específica solamente puede participar en un solo lado del arco a la vez, es decir cada entidad se relaciona o con una u otra de las entidades que están en el arco.

Por ejemplo en el Sistema de Biblioteca se tiene una relación mutuamente excluyente de la entidad LIBRO con las entidades ACADEMICO y ALUMNO.



La forma de lectura en este tipo de relación mutuamente excluyente es de la siguiente forma :

“Cada *entidad A* tiene una *relación1* con la *entidad1* o una *relación2* con la *entidad2* ”

Cada libro puede ser solicitado por uno y solo un académico o puede ser solicitado por uno y solo un alumno.

Cada alumno puede ser soliciante de uno o varios libros.

Cada académico puede ser solicitante de uno o varios libros.

Tomando en cuenta las consideraciones expuestas anteriormente se llegó a la definición del modelo de entidades y relaciones que actualmente contempla las necesidades de operación en la Facultad de Ingeniería a través de la Unidad de Servicios de Cómputo para la Administración (USECAFI).

En el apéndice A se tiene la definición de las entidades, en el apéndice B la explicación de las relaciones (entidades de intersección) que surgen de las relaciones M:M y en el apéndice C algunos diagramas entidad-relación de los sistemas que se manejan en la USECAFI a partir del diagrama general que se obtuvo y que se presenta también con este documento.

**Segunda Parte :****Proceso de Desarrollo del Sistema de Exámenes Extraordinarios.****I. DEFINICION DEL SISTEMA****Planteamiento del problema.**

El sistema de exámenes extraordinarios se diseñó por la necesidad de llevar un control confiable y rápido de los procesos administrativos que para cada período define la Facultad al realizar este tipo de exámenes.

**Dependencias que intervienen.**

Unidad de Servicios de Cómputo para la Administración (USECAFI).  
Responsable : Líder de proyecto.

Departamento de Administración Escolar  
Responsable : El Jefe del Departamento.

División de Ingeniería Civil, Topográfica y Geodésica (DICTG)  
Responsable : El Secretario Académico de la División.

División de Ingeniería Eléctrica, Electrónica y Computación (DIEEC).  
Responsable : El Secretario Académico de la División.

---

División de Ciencias de la Tierra (DCT).  
Responsable : El Secretario Académico de la División.

División de Ciencias Básicas (DCB).  
Responsable : El Secretario Académico de la División.

División de Ciencias Sociales y Humanidades (DCSH).  
Responsable : El Secretario Académico de la División.

División de Ingeniería Mecánica, Eléctrica e Industrial (DIMEI).  
Responsable : El Secretario Académico de la División.

Dirección General de Administración Escolar (DGAE).

### **Descripción del proceso.**

Existen tres períodos de exámenes extraordinarios al semestre, en los cuales se presentan como máximo 2 exámenes por período y hasta 4 durante el semestre por alumno con el fin de acreditar las asignaturas de acuerdo a la Legislación Universitaria y los acuerdos del Consejo Técnico de la Facultad de Ingeniería.

La operación administrativa está regulada por la Legislación Universitaria, que establece en los artículos 14, 15, 16 y 17 del Reglamento General de Exámenes Extraordinarios; y en los artículos 19 y 27 del Reglamento de Inscripciones, las normas que rigen la realización de estos exámenes :

---

División de Ciencias de la Tierra (DCT).

Responsable : El Secretario Académico de la División.

División de Ciencias Básicas (DCB).

Responsable : El Secretario Académico de la División.

División de Ciencias Sociales y Humanidades (DCSH).

Responsable : El Secretario Académico de la División.

División de Ingeniería Mecánica, Eléctrica e Industrial (DIMEI).

Responsable : El Secretario Académico de la División.

Dirección General de Administración Escolar (DGAE).

### **Descripción del proceso.**

Existen tres períodos de exámenes extraordinarios al semestre, en los cuales se presentan como máximo 2 exámenes por período y hasta 4 durante el semestre por alumno con el fin de acreditar las asignaturas de acuerdo a la Legislación Universitaria y los acuerdos del Consejo Técnico de la Facultad de Ingeniería.

La operación administrativa está regulada por la Legislación Universitaria, que establece en los artículos 14, 15, 16 y 17 del Reglamento General de Exámenes Extraordinarios; y en los artículos 19 y 27 del Reglamento de Inscripciones, las normas que rigen la realización de estos exámenes :

---

**Reglamento General de Exámenes Extraordinarios.**

Artículo 14 .- Los exámenes extraordinarios tienen por objeto calificar la capacitación de los sustentantes que no hayan acreditado las asignaturas correspondientes.

Artículo 15 .- Los exámenes extraordinarios se efectuarán en los períodos señalados en el calendario escolar. Estos serán realizados por dos sinodales, que deberán ser profesores definitivos de la asignatura correspondiente o de una afín.

Artículo 16 .- Los estudiantes tendrán derecho a presentar dos asignaturas por semestre mediante exámenes extraordinarios. Únicamente el Secretario General de la Universidad podrá conceder un número mayor de exámenes extraordinarios, previo informe favorable de la Dirección de la Facultad o Escuela y de la Coordinación de la Administración Escolar.

Artículo 17 .- En los exámenes extraordinarios se requerirá el acuerdo de ambos sinodales respecto a la calificación del sustentante. En caso de divergencia el Director de la Facultad o Escuela ordenará la revisión del examen a un tercer profesor definitivo de asignatura o una asignatura afín, quién fungirá como árbitro.

**Reglamento General de Inscripciones**

Artículo 19 .- Los alumnos que no terminen sus estudios en los plazos señalados no serán reinscritos y solo podrán acreditar sus materias por medio de exámenes extraordinarios

---

---

Artículo 27 .- Ningún alumno podrá ser inscrito más de dos veces en una misma asignatura, en caso de no acreditarla solo podrá hacerlo con examen extraordinario de acuerdo con lo dispuesto en el capítulo III del Reglamento General de Exámenes (Artículos 14, 15, 16 y 17).

Conviene aclarar que estos dos últimos artículos (19 y 27) generán la situación de los alumnos sin derecho a inscripción. Estos alumnos son inscritos de acuerdo a una petición realizada por los alumnos en Junio de 1976, donde se solicita cursar una asignatura como oyente con derecho a presentar exámenes parciales, resultando válida la calificación como extraordinario. Esta inscripción se limita exclusivamente a dos asignaturas.

Una vez explicadas la políticas a seguir para el registro de alumnos para realizar exámenes extraordinarios, se explica la manera de operar por las dependencias que intervienen en este proceso :

- Las Divisiones que conforman la Facultad de Ingeniería tienen la necesidad de capturar horarios para la realización de los exámenes extraordinarios para los tres períodos.

- Las Divisiones capturan horarios de exámenes extraordinarios; proporcionando la siguiente información : clave de la asignatura, grupo, período correspondiente, folio de los profesores que examinarán (dos por asignatura, conocidos como primero y segundo sinodal), salón, fecha y hora de realización del examen.

- Se generan dos listados que son enviados a las Divisiones correspondientes, para la revisión y corrección de horarios y fechas de realización de exámenes :

- El primero contiene la clave, nombre de la asignatura, grupo, nombre del primero y segundo sinodal asignado para cada período de exámenes extraordinarios. Este listado se genera una sola vez al semestre.

- El segundo contiene los horarios capturados para el período de exámenes extraordinarios próximo a realizarse. Esto indica que tiene que generarse tres veces durante el semestre.
  
  - Se corrigen los horarios y fechas de aplicación de exámenes extraordinarios.
  
  - Se publican los horarios en lugares estratégicos dentro de la Facultad de Ingeniería para que sean consultados por los alumnos.
  
  - El Departamento de Administración Escolar entrega las formas para el registro, las cuales serán llenadas, entregadas y capturadas en USECAFI por el alumno, de acuerdo a las fechas establecidas en el calendario escolar.
  
  - Durante la semana siguiente al último día de entrega de formas de inscripción, se realizarán las siguientes actividades en USECAFI :
    - Se generan los archivos que serán enviados a la Dirección General de Administración Escolar (DGAE). Estos archivos comúnmente son llamados MIDI y GP, que significa Maestro\_Detalle y Grupo\_Profesor.
  
    - Se generan e imprimen reportes de acuerdo a lo establecido por las Divisiones de la Facultad y la Secretaría de Servicios Escolares de acuerdo a sus necesidades.
  
    - Respaldo de la información y reporte de la misma.
-

---

## **II. ANALISIS DE REQUERIMIENTOS**

### **Requisitos Funcionales.**

El sistema de exámenes extraordinarios deberá realizar las siguientes funciones :

- Facilitar la captura de horarios de exámenes extraordinarios para los períodos normales, así como su verificación y corrección.
  
- Facilitar la captura de segundos sinodales para el período de alumnos sin derecho a inscripción.
  
- Generar los reportes que muestren la situación final de los alumnos inscritos a los exámenes extraordinarios del período en curso; así como la de los profesores que examinarán.
  
- Generación de archivos que muestren la situación final de los alumnos inscritos a extraordinarios; así como la de los profesores que examinarán. Estos archivos se pondrán a disposición de la Dirección General de Administración Escolar (DGAE) para los usos que considere pertinentes.
  
- Respaldo de la información utilizada en el período de exámenes extraordinarios en curso.

El diagrama de Entidades y Relaciones particular para este sistema se puede consultar en el apéndice C.

**Requisitos de Operación.**

- La interacción del sistema con el usuario debe ser fácil y bajo un enfoque amigable; esto se podrá lograr con un menú de control de operaciones.

- Debe enviar mensajes a la persona adecuada, ya sea al operador del equipo o al encargado del sistema. Estos mensajes deben ser preferentemente en español y debidamente explicados para que sean comprendidos por el usuario evitando confusión de conceptos o acciones.

**III. DISEÑO****Ambiente del Sistema.**

- Hardware : Equipo VAX configurado en Cluster con unidades de respaldo e impresoras.
  
- Software : Sistema Operativo VAX OpenVMS.  
Compilador y Ligador del Lenguaje C  
Manejador de Bases de Datos Oracle7  
Precompilador Pro\*C  
Manejador de formas SQL\*Forms  
Lenguaje de Consultas SQL\*Plus

**Documentación de referencia**

- SQL\*Plus User's Guide.
- SQL\*Plus Reference Manual
- SQL\*Forms Reference Manual
- Pro\*C User's Guide
- VAX C User's Guide
- Lenguaje de Programación C; Kerningham/Ritchie

**Estructura de Datos****GRUPO EXTRAORDINARIOS**

Nombre	Tipo	Longitud	Obligatoriedad	
Número_único	Numérico	4	Not Null	PK
Número	Numérico	2	Not Null	
Período	Carácter	1	Not Null	
Fecha	Fecha	7		
Salón	Carácter	4		
Asignatura	Númerico	4		FK

**SINODALES**

Nombre	Tipo	Longitud	Obligatoriedad	
Número_único	Numérico	4	Not Null	PK
RFC	Carácter	13		
Primero_segundo	Carácter	1	Not Null	

---

**INSCRIPCION\_EXTRAORDINARIOS**

Nombre	Tipo	Longitud	Obligatoriedad	
Número_único	Numérico	5	Not Null	FK
Cuenta	Numérico	8	Not Null	FK

**ACADEMICOS\_INTERNOS**

Nombre	Tipo	Longitud	Obligatoriedad	
RFC	Carácter	13	Not Null	PK
Título	Carácter	5		
Nombre	Carácter	30	Not Null	
Apellido1	Carácter	30		
Apellido2	Carácter	30		
Folio	Numérico	6		
Sexo	Carácter	1		
Profesor	Carácter	1		

**ASIGNATURAS**

Nombre	Tipo	Longitud	Obligatoriedad	
Clave	Numérico	4	Not Null	PK
Nombre	Carácter	28	Not Null	
Créditos	Number	2	Not Null	
Laboratorio	Carácter	1		
Inicio	Carácter	3		
Vigencia	Carácter	1		
Departamento	Numérico	3		
Nombrecompleto	Carácter	70		

---

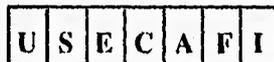
## ALUMNOS

Nombre	Tipo	Longitud	Obligatoriedad	
Cuenta	Numérico	8	Not Null	PK
Nombre	Carácter	50	Not Null	
Domicilio	Carácter	37		
Colonia	Carácter	30		
Delegación	Carácter	20		
CP	Numérico	5		
Teléfono	Numérico	7		
Nacimiento	Carácter	8		
Sexo	Carácter	1		
Ingreso_DGAE	Carácter	2	Not Null	
Ingreso_Facultad	Carácter	3	Not Null	
Tipo	Carácter	1		
Forma_ingreso	Carácter	1	Not Null	FK
Nacionalidad	Numérico	3	Not Null	FK
Estado	Numérico	1	Not Null	FK
Fecha_actualización	Fecha	7		
Tutor	Carácter	50		
Envio_archivo	Carácter	1		
Fecha_envío	Fecha	7		

**IV. IMPLEMENTACION.**

Con la estructura de datos anterior y de acuerdo al diagrama de entidades y relaciones general que se obtuvo, se implementó este Sistema; para el cual se explica la forma que permite realizar el registro de asignaturas que cada alumno desea presentar como examen extraordinario:

---

**Sistema de Exámenes Extraordinarios**

NO. CUENTA

NOMBRE

84565927

ROMERO ESPINO EDMUNDO

Al momento de entrar a sesión el alumno deberá introducir su número de cuenta. Se validará y se desplegará su nombre.

En la parte inferior de la forma aparece un mensaje que le indica al alumno que debe teclear su número de cuenta sin guión. En caso de que se inserten caracteres inválidos o un número de cuenta erróneo se desplegará un mensaje en la parte inferior de la pantalla que dice “**ERROR. Número de cuenta inexistente.**”; por lo tanto el alumno deberá checar cual fue su error de captura y corregirlo o solicitar apoyo con el personal de USECAFI que le informará lo que ocurre con su número de cuenta.

Una vez que el número de cuenta es válido se realiza una consulta a la Base de Datos para saber las inscripciones a exámenes extraordinarios que el alumno ha realizado hasta el momento durante el semestre, tanto en los tres períodos como a los grupos de oyentes. Se despliega esta parte complementaria en la forma debajo de los campos del número de cuenta y nombre del alumno, como se indica en la siguiente figura.

## Sistema de Exámenes Extraordinarios

U	S	E	C	A	F	I
---	---	---	---	---	---	---

NO. CUENTA

NOMBRE

84565927

ROMERO ESPINO EDMUNDO

TIENES REGISTRADAS 3 ASIGNATURAS HASTA EL MOMENTO

CVE. ASIGNATURA	NOMBRE	PERIODO
59	CALCULO DIFER. E INTEGRAL	SEGUNDO
59	CALCULO DIFER. E INTEGRAL	TERCERO
1402	HIDRAULICA BASICA	OYENTE

CVE. ASIGNATURA	NOMBRE
■	

TECLEA LA CLAVE DE LA ASIGNATURA.

Una vez despliega la información de las asignaturas que ha registrado hasta el momento, el cursor estará en el campo que solicita la clave de la asignatura, la cual será capturada por el alumno. Al teclear la clave se realizarán una serie de validaciones para asegurar que el alumno pueda registrar dicha asignatura.

Las validaciones se programaron como procedimientos de PL/SQL en la forma. Estos procedimientos se ejecutarán automáticamente al momento que el alumno oprima la tecla "ENTER" una vez que haya capturado la asignatura que desea registrar para presentar el examen extraordinario. A continuación se menciona el objetivo de cada procedimiento y el mensaje de error correspondiente al fallar la validación.

- VALIDA\_LIMITE\_INSCRIPCIONES.

Con este procedimiento se checa que el alumno no registre más de 4 asignaturas en el semestre o más de 2 en el período. En caso de que haya rebasado estos límites se desplegará el siguiente mensaje de error en la parte inferior de la forma : **“ERROR. No puedes continuar registrando asignaturas”**.

- VALIDA\_ASIGNATURA.

Se checa que la clave de la asignatura sea un número registrado en la tabla de asignaturas de la Facultad de Ingeniería. El mensaje de error en caso de que se inserten caracteres inválidos o una clave que no existe es : **“ERROR. Asignatura inexistente”**.

- VALIDA\_ASIGNATURA\_EN\_PERIODO.

Se checa que la asignatura esté registrada para poderse inscribir a examen extraordinario en el período vigente. El mensaje de error es **“ERROR. La asignatura no está registrada para este período.”**

- VALIDA\_INSCRIPCION\_REPETIDA.

Se checa que la asignatura que está capturando no la haya registrado ya para el período vigente. El error es : **“ERROR. Ya estás inscrito en esta asignatura para este período”**.

---

- VALIDA\_SERIACION\_Y\_APROBADA.

Se verifica que el alumno pueda registrar la asignatura de acuerdo al plan con el que se inscribió y por la tanto la seriación que debe respetar. Aquí mismo se checa que no tenga aprobada la asignatura, pues no tendría caso que se inscribiera para presentar el examen extraordinario de una asignatura aprobada.

El mensaje de error se obtiene de una tabla que tiene el tipo de error correspondiente a la falta que se esté registrando, pero será de la siguiente forma : **“ERROR. Sin derecho a inscripción : <mensaje complementario>”**.

- VALIDA\_IRREGULARIDAD

Se verifica que el alumno no esté dado de baja. El mensaje de error es : **“ERROR. Tienes registrada una suspensión, verifica en USECAFI”** .

- VALIDA\_PERTENECE\_A\_LA\_CARRERA

De acuerdo al plan de estudios del alumno se checa que la asignatura pertenezca a su carrera y a su plan. En caso de error, el mensaje es : **“ERROR. La asignatura solicitada no pertenece a tu carrera”**.

Para todos estos casos no se registrará la asignatura para ser presentado el examen extraordinario, se borrará la información del registro y se solicitará que capture la siguiente o que concluya su captura.

---

Cuando las validaciones hayan sido correctas y por lo tanto no se generen mensajes de error, se le preguntará al alumno si los datos capturados son correctos o no; para lo cual deberá responder con "S" para indicar que sí o con "N" para indicar que no. En caso de teclear algún carácter distinto se le mandará el mensaje de error : **"ERROR: Solo se acepta "S" o "N" "**.

Cuando la respuesta ha sido afirmativa se registrará la inscripción y se le mandará al alumno el siguiente mensaje en la forma : **"Asignatura registrada como inscripción a Examen Extraordinario"**. Y para los casos en que la respuesta sea negativa el mensaje es : **"La asignatura no se registro como inscripción"**.

**Sistema de Exámenes Extraordinarios**

U	S	E	C	A	F	I
---	---	---	---	---	---	---

NO. CUENTA

NOMBRE

84565927

ROMERO ESPINO EDMUNDO

**TIENES REGISTRADAS 3 ASIGNATURAS HASTA EL MOMENTO**

CVE. ASIGNATURA	NOMBRE	PERIODO
59	CALCULO DIFER. E INTEGRAL	SEGUNDO
59	CALCULO DIFER. E INTEGRAL	TERCERO
1402	HIDRAULICA BASICA	OYENTE

CVE. ASIGNATURA	NOMBRE
2120	INGENIERIA SISMICA

ESTOS DATOS SON CORRECTOS ? [S/N]
-----------------------------------

**RESPONDE CON 'S' PARA CONFIRMAR O 'N' PARA CANCELAR**

Cuando el alumno termine de registrar sus asignaturas, oprimirá la tecla que se encuentra en la parte derecha del teclado identificada como PF2 y la pantalla quedará limpia para que otro alumno pueda realizar el registro de asignaturas bajo el mismo procedimiento.

**Sistema de Exámenes Extraordinarios**

U	S	E	C	A	F	I
---	---	---	---	---	---	---

NO. CUENTA

NOMBRE

**TECLEA TU NUMERO DE CUENTA SIN GUION.**

---

## CONCLUSIONES

Con el desarrollo de este trabajo se han valorado y centrado varios de los conceptos que en esta época se han estado manifestando, como son los esquemas de cómputo con multiples procesadores, así como las recomendaciones de reingeniería en los procesos.

Considero importante hacer mención que los equipos de cómputo y el hardware en general están aumentando la capacidad de proceso y almacenamiento de información a precios que se pueden considerar accesibles para el potencial de uso que ofrecen. Ahora bien, esto no debe convertirse en un desperdicio desmedido de recursos, una falta de administración y mal aprovechamiento de los mismos.

Las arquitecturas de hardware, los Sistemas Operativos y los manejadores de Bases de Datos, podrán ser eficientes y de vanguardia tecnológica pero si nosotros los que somos los responsables de realizar las implementaciones de las aplicaciones reales para la solución de problemas específicos no somos capaces de mantener el sentido común y profesional de lo que significa el buen análisis y diseño de aplicaciones para plataformas y ambientes particulares, de nada servirá el enfoque que los proveedores ya han previsto para que dichas aplicaciones sean eficientes y productivas.

El hablar de reingeniería con este trabajo ha sido para manifestar que a nivel general se requiere de manera sustancial una revisión de los procesos y actividades desarrolladas en la Facultad de Ingeniería, para que se puedan involucrar de manera coordinada todos los responsables de los procesos de administración y operación; y garantizar que el trabajo que se está realizando actualmente es realmente lo que se espera en la Facultad. Esto lo digo porque a través de la USECAFI se están impulsando y desarrollando sistemas que apoyen al proceso académico-administrativo de la Facultad y posiblemente no se está analizando el esquema global de los procesos en la misma, ya que automatizar alguna actividad, a través de una aplicación, no debe traducirse en capturar tantas veces la información como se hacía manualmente.

Este tipo de circunstancias va a causar unicamente un desgaste en el desarrollo de futuras aplicaciones y falta de integración tanto de los sistemas como de los datos en la Facultad.

Lo que se ha podido realizar con este trabajo es la integración y análisis de los sistemas que actualmente se requieren en la Facultad en un modelo de entidades y relaciones que los contempla de una manera unificada para evitar esquemas aislados tanto de información como de responsabilidad sobre el uso común y consistente de la información a través de USECAFI.

Un factor que es importante destacar sobre la consistencia en el desarrollo de futuras aplicaciones, es que en USECAFI a pesar de manifestarse un gran entusiasmo y entrega en la realización de los trabajos designados, conserva un alto índice de rotación de personal lo que a su vez origina una falta de documentación en los sistemas que se han desarrollado; lo cual originó algunos problemas para saber la forma en que se podían integrar todos los sistemas que se habían contemplado.

Los equipos de cómputo con los que se cuenta en USECAFI son modelos que desde hace algunos años se les tuvo que incrementar sus recursos de CPU y memoria principalmente, ya que los procesos que se ejecutan cada vez son realizados con un mayor número de registros de información y hace que los procesos sean de igual forma cada vez más lentos sin que sea necesariamente por ineficiencia en el desarrollo de las aplicaciones. Por lo que de antemano es necesario recomendar la adquisición de hardware que permita un mejor tiempo de respuesta.

El sistema de exámenes extraordinarios que se implementó ha permitido comprobar las ventajas de utilizar y mantener actualizado un modelo de entidades y relaciones del esquema actual en la Facultad, ya que en línea se pueden realizar las inscripciones, lo que implica acceder y actualizar también en línea la Base de Datos mediante una aplicación eficiente y acorde con las tendencias en automatización de la administración escolar.

Así mismo reitero la recomendación de mantener un equipo para desarrollo y otro para producción ya que en períodos críticos se ve de manifiesto el disgusto tanto de desarrolladores como de usuarios finales por la lentitud tanto para desarrollar como para operar las aplicaciones.

No se puede hablar de reingeniería de procesos, eficiencia en sistemas académico-administrativos, equipos de cómputo de vanguardia y arquitecturas eficientes; aisladamente si no existe un compromiso de adecuarse a los recursos y esquemas de cómputo emergentes, al ingenio eficiente para la recomendación de sistemas y posibles configuraciones que apoyarán de manera óptima el desarrollo de las actividades cotidianas.

Se ha manifestado a través de este trabajo que los recursos que demanda el software son cada vez mayores por lo que el crecimiento en la demanda de manipulación de información y desarrollo de aplicaciones, debe ser consistente con el crecimiento en recursos de hardware que lo soporten; ya que de nada servirá el ingenio en el análisis, diseño e implementación de las aplicaciones, sino se cuenta con los recursos mínimos que los mismos usuarios finales están demandando.

Considero que el esfuerzo que sigue y trasmite este trabajo es sensibilizar sobre una cultura informática que debe ser llevada a todos los niveles para que se pueda indicar que la inversión en las herramientas de software y equipo de cómputo adecuado debe llevar consigo la eficiencia en las tareas y labores que se realicen en cada institución; bajo bases firmes de que lo que se realiza es por que se ha llevado un buen análisis de los procesos que generan el trabajo del personal y los sistemas o aplicaciones que les apoyen para el cumplimiento de su labor cotidiana; ya que si la eficiencia y la seguridad en los procesos de la Institución no es lo importante, de nada sirve que se contemple el uso de herramientas, hardware sofisticado y mucho menos una reingeniería del esquema establecido.

Como el planteamiento original de este trabajo lo contemplaba, se ha propuesto e implementado un esquema de configuración en Cluster con los equipos de cómputo de la Facultad y se ha obtenido el diagrama que define de manera real y actualizada la forma en que se opera la información a través del diagrama de entidades y relaciones.

Con el diagrama hemos podido validar los sistemas existentes y se preveen algunas de las aplicaciones que a consecuencia se habrán de realizar. Por otra parte, se contempla el poder generar un diagrama por cada aplicación, que se obtienen del diagrama general, así mismo los modos de acceso a la información, privilegios, etc.

Se han sentado las bases para implementar también un esquema mas real de administración y privilegios que tendrán los responsables y usuarios de cada aplicación, todo ello aprovechando las capacidades que otorgan las herramientas, el manejador de la Base de Datos y el Sistema Operativo.

**BIBLIOGRAFIA.**

Hammer, Michel / Champy, James  
Reingeniería  
Grupo Editorial Norma

Diaz de Salas Montañez, Héctor / González Arreola, Mireya /  
Guzmán Gutiérrez, Pablo / Palao Muñoz, Domingo  
Análisis, Diseño y construcción de un sistema de reinscripciones para la Facultad de  
Ingeniería bajo un entorno de Base de Datos.

Barker, Richard  
CASE\*Method Task and Deliverables  
Addison - Wesley

Barker, Richard  
CASE\*Method Entity - Relationship  
Addison - Wesley

Milenkovic, Milan  
Sistemas Operativos. Conceptos y Diseño  
Mc Graw Hill

Deitel, H.M.  
An Introduction to Operating Systems  
Addison - Wesley

Oracle7 Parallel Server  
Administrator's Guide  
Oracle Corporation 1992

---

ESTA TESIS NO DEBE  
SALIR DE LA BIBLIOTECA

Oracle7 Server  
SQL\*Language Reference Manual  
Oracle Corporation 1992

Oracle7 Server  
Utilities User's Guide  
Oracle Corporation 1992

Oracle, Relational Database Management System  
General Information Manual  
Oracle Corporation 1993

Oracle7 Server  
Concepts Manual  
Oracle Corporation 1992

OpenVMS Operating System Handbook  
Digital Equipment Corporation 1992.  
VAXcluster Systems

Digital Technical Journal  
Digital Equipment Corporation

OpenVMS Clusters Handbook  
Power Delivering Flexibility  
Digital Equipment Corporation 1993.

Distributed Systems  
Digital Technical Journal  
Digital Equipment

**APENDICE A**

## **ENTIDADES**

**ACADEMICO** .- Personal de la UNAM contratado para realizar actividades académicas en alguna dependencia universitaria. Para los fines del modelo de la Facultad de Ingeniería nos interesa separar aquellos que no pertenecen a esta dependencia ya que no se tendrá que llevar a cabo ningún registro ni proceso administrativo con ellos; por lo tanto esta entidad se manifiesta como el conjunto de dos subtipos (ACADEMICO\_EXTERNO y ACADEMICO\_INTERNO).

**ACADEMICO\_EXTERNO** . - Personal académico que no tiene alguna plaza en la Facultad de Ingeniería pero participa en ciertas actividades, por ejemplo ser sinodal en algún examen profesional.

**ACADEMICO\_INTERNO** . - Personal de la Facultad de Ingeniería contratado con una plaza para realizar actividades académicas en alguna dependencia interna. Este personal puede o no ser profesor de la Facultad.

**ALUMNO** .- Todos los alumnos registrados en la Facultad de Ingeniería (plantel 011). Para fines administrativos se requiere saber aquellos que son de primer ingreso y los que tienen derecho a inscripción, por lo que se han generado dos subtipos que mantienen atributos particulares de acuerdo a la condición del alumno durante su estancia en la Facultad.

**ALUMNO\_EN\_REINSCRIPCION** .- Estudiante de la UNAM que se puede reinscribir porque tiene derecho a reinscripción y número generado para la solicitud de grupos que están asociados a las asignaturas que desea cursar.

**ALUMNO\_PRIMER\_INGRESO** . - Estudiante que acaba de ingresar a la Facultad de Ingeniería para cubrir el plan vigente de asignaturas de acuerdo a la carrera elegida.

**ASIGNATURA** .- Contenido disciplinario contemplado en el mapa curricular de un plan de estudios y que posee una clave oficial otorgada por la Dirección General de Administración Escolar (DGAE).

**CARRERA** .- Area o rama profesional de licenciatura que se ofrece en esta Facultad y es coordinada por una División.

**CLASE** .- Cátedra que puede ser impartida por personal académico de la Facultad de Ingeniería en algún horario para algún grupo en particular.

**CONTRATO** .- Relación de trabajo que establece un académico para prestar sus servicios profesionales en la Facultad de Ingeniería.

**DEPENDENCIA** . - Unidades, centros, departamentos, coordinaciones, secretarías y demás organizaciones que pertenecen a la Facultad de Ingeniería; así mismo se les puede considerar como un lugar de adeudo.

**DEPENDENCIA\_EXTERNA** . - Organismos administrativos y académicos de la UNAM que intervienen en operaciones particulares con la Facultad de Ingeniería. Por ejemplo puede ser la coordinadora de algún servicio social propuesto y desarrollado por algún alumno de esta Facultad.

**DEPENDENCIA\_INTERNA** . - Organismos administrativos y académicos de la Facultad de Ingeniería.

**EDIFICIO** . - Lugar dentro de las instalaciones de la Facultad de Ingeniería en donde se localizan los salones.

**EQUIVALENCIA** . - El conjunto de asignaturas que se han reestructurado. Se mantiene la clave de asignatura anterior y actual para cada plan de estudio.

**ESTADO** . - Catálogo de los estados de la República Mexicana.

**EXAMEN PROFESIONAL** . - Trámite que cubren los alumnos de la Facultad de Ingeniería al concluir el plan de estudios de una carrera y desarrollar el trabajo de tesis para obtener el título de Ingeniero.

**FALTA** . - Registro de las faltas de los académicos a sus actividades cotidianas, con el fin de obtener estadísticas y otorgar incentivos.

**FECHA\_ESPECIAL** . - Fechas en las que se justifican las faltas a los académicos o se deben realizar procesos específicos de acuerdo al calendario vigente para la Facultad de Ingeniería.

**FORMA\_INGRESO** . - Catálogo de los métodos por los cuales son aceptados los alumnos en la Facultad de Ingeniería.

**FORMA\_REMUNERACION** . - Manera en que es remunerado el trabajo que realizan los alumnos como servicio social.

**GRUPO\_EXTRAORDINARIO** . - Registro de los grupos para presentar exámenes extraordinarios. Los exámenes de estos grupos se realizan en alguno de los tres períodos de fechas en un semestre.

---

**GRUPO\_ORDINARIO** . - Opción de una asignatura con uno o varios profesores a su cargo, dentro de un horario en el cual los alumnos con número de reinscripción lo solicitan para cursar la asignatura.

**GRUPO\_OYENTE** . - Registro de los grupos que aceptan alumnos oyentes.

**GRUPO\_PRIMER\_INGRESO** . - Registro de los grupos que atienden a los alumnos de primer ingreso.

**LIBRO** . - Catálogo de libros existentes en la Biblioteca de la Facultad de Ingeniería y que pueden ser consultados en las salas de la biblioteca o pueden ser solicitados como préstamo a domicilio por académicos y alumnos.

**NACIONALIDAD** . - Catálogo de claves y nacionalidades más comunes de los alumnos que se inscriben en la Facultad de Ingeniería.

**NOMBRAMIENTO** . - Título del puesto que ocupará algún académico de la Facultad de Ingeniería durante la prestación de sus servicios profesionales a la misma.

**ORGANO** . - Dependencia de alto nivel en la organización de la Facultad de Ingeniería.

**PERIODO\_CLASE** . - Bloques de tiempo en los que se imparten clases en la Facultad de Ingeniería.

**PLAN** . - El conjunto de asignaturas que contempla una carrera en algún período determinado. Un plan varía con respecto al paso del tiempo.

**PROFESOR .-** Personal académico de la Facultad de Ingeniería que imparte clases.

**PROGRAMA .-** Registro de los temas y programas en los que participan los alumnos para realizar el servicio social.

**REGISTRADO .-** Las ocurrencia que tiene un alumno de la Facultad de Ingeniería con respecto a cada carrera que esté cursando o que haya cursado en esta dependencia de la UNAM.

**SALON .-** Aula o laboratorio en que se imparte una asignatura de la Facultad de Ingeniería y se le atribuyen ciertas características como, acondicionamiento para proyección, ubicación, número de lugares, etc.

**SERIACION .-** Conjunto de asignaturas que exigen ciertos requisitos y antecedentes para ser cursadas de acuerdo a la vigencia de algún plan y con respecto a la carrera en la que esté inscrito cada alumno de la Facultad de Ingeniería.

**SERVICIO\_SOCIAL .-** Trabajo desarrollado por los alumnos como apoyo a la comunidad para cada carrera en la que se inscriba; y es un requisito para recibir el título de Ingeniero. Por este trabajo se puede incluso recibir algún tipo de remuneración.

**SITUACION .-** Medio ambiente o circunstancias por las que se genera la propuesta del servicio social que realizará cada alumno.

**TARJETA .-** Forma para registrar la asistencia a las actividades de los académicos. Se tiene un registro o tarjeta por cada asignatura que imparta un profesor.

---

**TESIS** . - Trabajo escrito que elabora un alumno de la Facultad de Ingeniería al finalizar el plan de estudios de cada carrera que estudie, con el objetivo de realizar el examen profesional que lo acreditará como Ingeniero.

**TIPO\_PROGRAMA** . - Catálogo de programas establecidos para que los alumnos desarrollen el servicio social.

**VIGENCIA\_PLAN** . - Control de la vigencia de algún plan de estudios así como la determinación del plan anterior a éste.

**APENDICE B**

## RELACIONES

**ADEUDA\_ACADEMICO** . - Identificación de los adeudos de material o bienes por parte de los académicos de la Facultad de Ingeniería en las dependencias de la misma.

**ADEUDA\_ALUMNO** . - Registro de los adeudos que tienen ciertos alumnos en alguna dependencia. El fin es determinar la situación actual de cada alumno y tomar medidas sobre su reinscripción por ejemplo.

**EXAMEN\_ACADEMICO** . - Registro de los académicos que participan en cada examen profesional.

**EXAMEN\_REGISTRADO** . - Registro de los alumnos por cada examen profesional presentado para cada carrera que curse.

**HA\_ESTUDIADO** . - Alumnos que cursaban alguna carrera en la Facultad de Ingeniería y se han dado de baja definitivamente así como aquellos que han cambiado de carrera. Un cambio de carrera consiste en renunciar a una carrera para inscribirse en otra.

**HISTORIA\_ACADEMICA** . - Registro de asignaturas que ha cursado cada alumno en la Facultad de Ingeniería .

**INSCRIPCION\_EXTRAORDINARIO** . - Registro de alumnos inscritos en grupos para presentar exámenes extraordinarios.

**INSCRIPCION\_NUEVO\_INGRESO** . - Registro de alumnos inscritos en los grupos de primer ingreso en la Facultad de Ingeniería.

**INSCRIPCION\_OYENTE** . - Registro de los alumnos que se inscriben a grupos que aceptan oyentes en el semestre vigente.

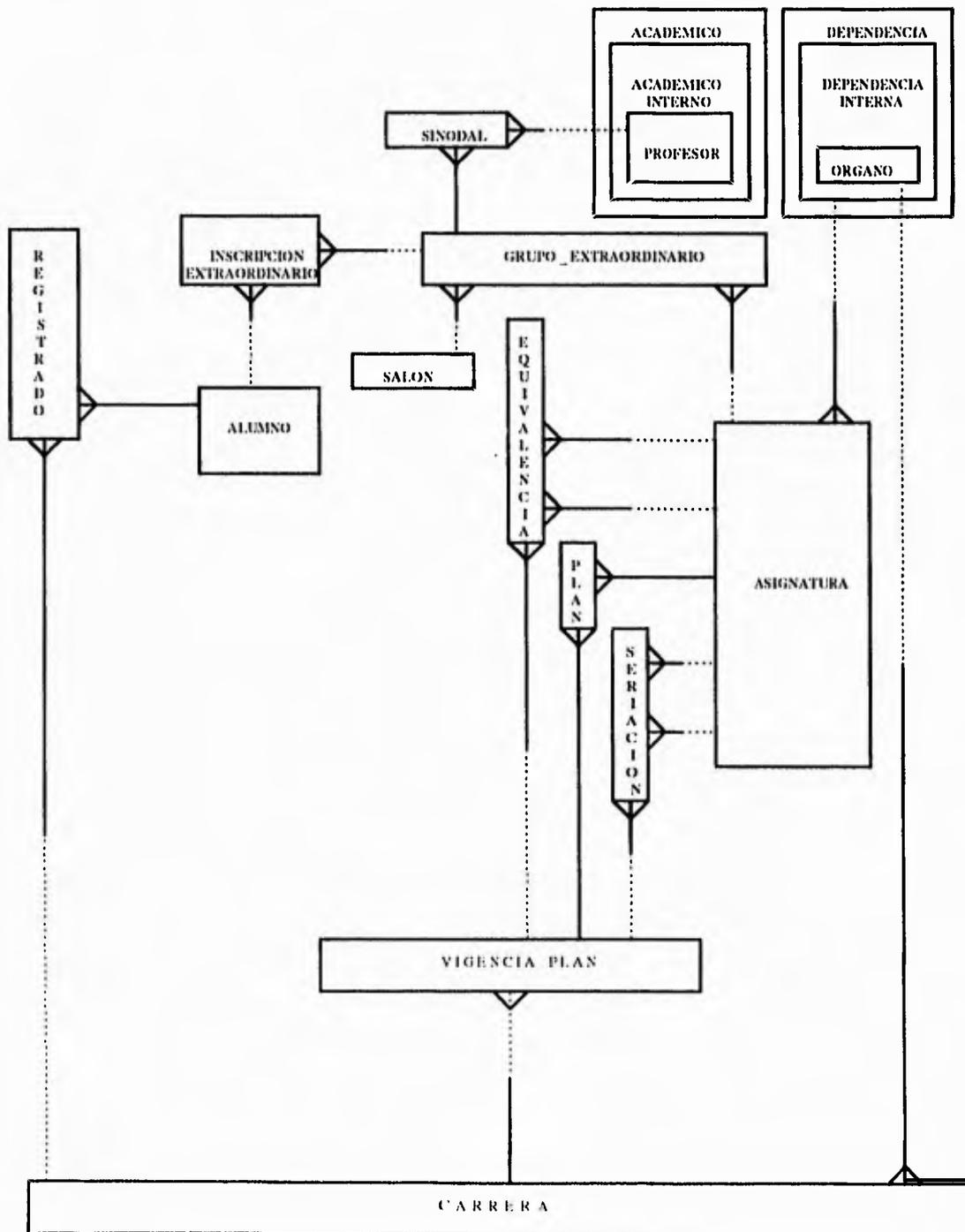
**INSCRIPCION\_ORDINARIA** . - Alumnos inscritos a un grupo ordinario para el semestre vigente.

**SINODAL** .- Académico que atiende a los grupos de exámenes extraordinarios. Los académicos son identificados como primero o segundo sinodal.

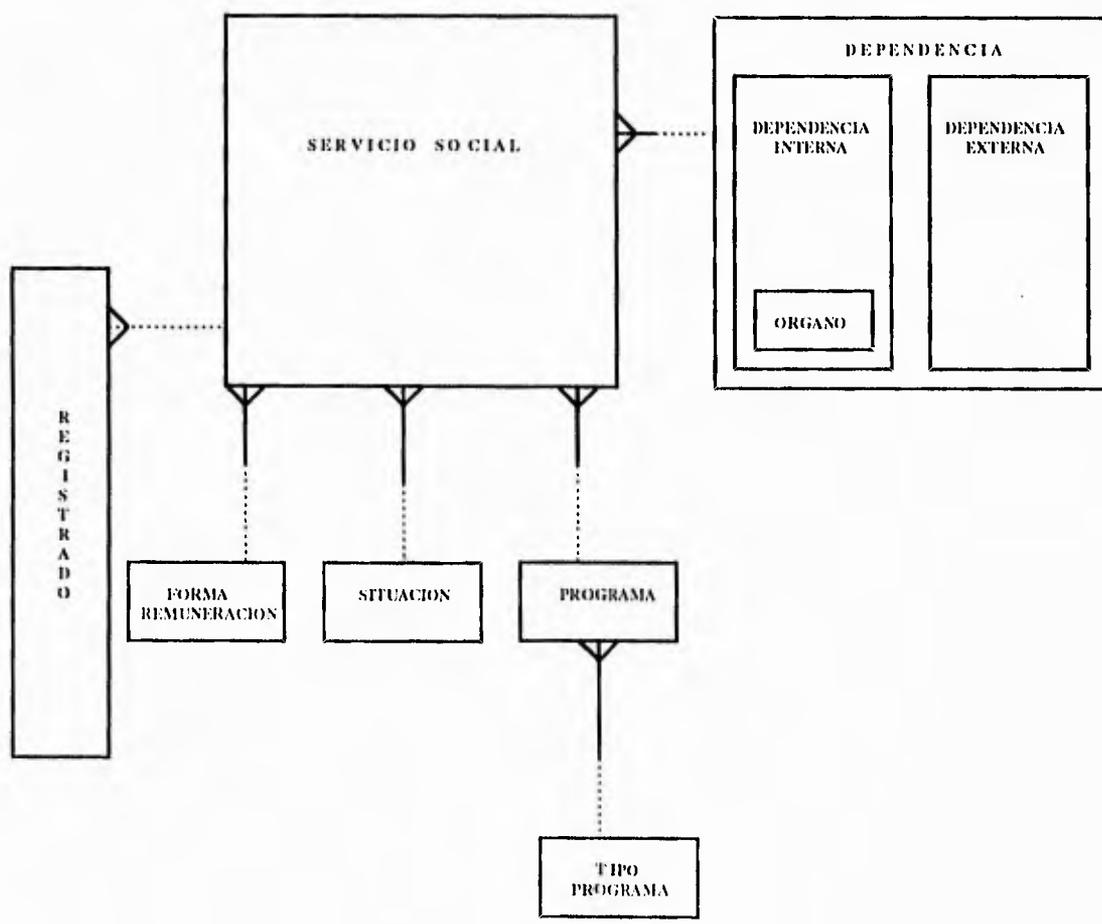
**SOLICITUD** . - Grupo de inscripción que solicita un alumno para cursar una asignatura. Existen dos opciones de grupo para cada asignatura solicitada que pueden ser iguales.

**APENDICE C**

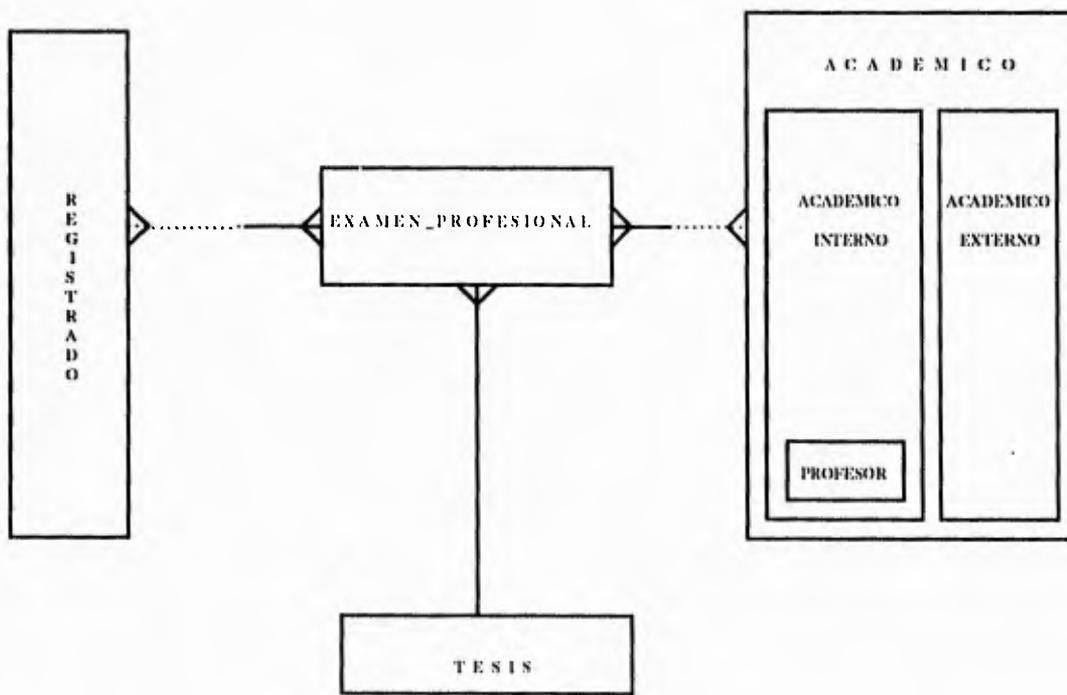
### EXTRAORDINARIOS



### SERVICIO SOCIAL



TITULADOS



### TITULADOS

