

ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL

Facultad de Ingeniería en Electricidad Y Computación

“Sistema Optimizador de Base de Datos”

TESIS DE GRADO

Previa la obtención del Título de:

INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN

SISTEMAS TECNOLÓGICOS

INGENIERO EN COMPUTACIÓN ESPECIALIZACIÓN

SISTEMAS DE INFORMACIÓN

Presentado por:

Alex Paúl Tapia Chichande

Luis Miguel Escobar Larenas

Guayaquil – Ecuador

2005

AGRADECIMIENTO

Agradecemos a DIOS quien con su infinita bondad, misericordia y amor nos ha permitido culminar esta tesis de graduación.

A la ESPOL en cuyas aulas se fue forjando nuestro conocimiento diario para formarnos como profesionales de bien.

A nuestro Director de Tesis, Ing Fabricio Echeverría quien con su ayuda incondicional, comprensión, ejemplo de superación y exigencia diaria supo como transmitirnos un gran legado intelectual, cualidades que lo han hecho un guía profesional en nuestras vidas.

A nuestros maestros por los conocimientos otorgados y a nuestros familiares y amigos por el apoyo brindado.

DEDICATORIA

A Dios por haberme permitido culminar los estudios, con su bendición y protección en todo este tiempo de preparación.

A mi esposa por su incondicional apoyo, y confianza demostrado durante todos estos años, a mi hija por la inspiración y motivación que me entrega cada día, a mis padres, tíos, suegros y familiares que han sido un pilar fundamental en mi vida y por el respaldo brindado durante el transcurso de estos años para la consecución de esta obra.

Alex Paúl Tapia Chichande.

DEDICATORIA

Ahora puedo comprender las palabras de mi padre “Principio y fin de mi jornada”, haciendo referencia a la razón su vida.

Tomando su incomparable ejemplo dedico esta obra a mis seres amados empezando por la razón de mi existencia mi incondicional esposa y mi inefable hijo, quienes me alientan día a día.

A mis padres por aquella sacrificada abnegación y guía entregada a lo largo de todos los días de mi vida.

A mis hermanos por ese espíritu de superación indomable y esa fe expresada en amor en cada instante de mi ser.

Luis Miguel Escobar Larenas

TRIBUNAL DE GRADUACIÓN

	MSC. Miguel Yapur Auad	
	SUBDECANO FIEC	
	PRESIDENTE DEL TRIBUNAL	

	Ing. Fabricio Echeverría Briones	
	DIRECTOR DE TESIS	

	Ing. Galo Valverde Landívar	
	VOCAL PRINCIPAL	

	Ing. Marcelo Loor	
	VOCAL PRINCIPAL	

DECLARACIÓN EXPRESA

“La responsabilidad del contenido de esta Tesis de Grado, nos corresponden exclusivamente y el patrimonio intelectual de la misma a la ESCUELA SUPERIOR POLITÉCNICA DEL LITORAL”

(Reglamento de Graduación de la ESPOL)

Luis Escobar Larenas

Alex Tapia Chichande

RESUMEN

Ante el inminente uso de las bases de datos, su constante evolución, integración y el potencial económico que esta representa por el manejo de datos; y considerando el arduo trabajo que realizan los DBA en sus diferentes labores; se implementó un CASE denominado SYOPBASE (Sistema Optimizador de Base de Datos), el cual permitirá realizar un enlace entre el divorcio producido por el modelo lógico y el modelo físico del almacenamiento de los datos, contemplando sólo la optimización

basada en la ubicación física de las tablas de acuerdo a las características físicas de los dispositivos de almacenamiento

Para la realización de la presente tesis en primera instancia se consideró un estudio la naturaleza de las tablas contenidas en las diferentes bases de datos, para posteriormente relacionarlas con los discos duros a utilizar dependiendo de las características propias de los mismos o de los tipos de RAID usados.

La siguiente fase fue implementar un Algoritmo Genético:

Algoritmo de búsqueda basado en la mecánica de la selección natural y de la genética natural. Combina la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorizado, para constituir así un algoritmo de búsqueda que tenga algo de las genialidades de las búsquedas humanas (Goldberg, 1989)

El cual se utilizó para determinar la mejor ubicación de cada una de las tablas en un disco específico, optimizando de esta manera aquel trabajo que hasta la actualidad se realizaba empíricamente, sin ninguna base científica sino basado únicamente en la experiencia propia de la persona encargada de la base de datos.

ÍNDICE GENERAL

INDICE GENERAL	I
INDICE DE ABREVIATURAS	II
INDICE DE FIGURAS	III
INDICE DE ANEXOS	IV
CAPITULO 1 INTRODUCCION.....	1
1.1. DESCRIPCIÓN DEL PROBLEMA.....	1
1.2. AMBIENTES DONDE SE PRODUCE EL PROBLEMA	4
1.3. SOLUCIÓN PROPUESTA Y SU FACTIBILIDAD.....	6
1.4. OBJETIVOS DE LA SOLUCIÓN.....	7
CAPITULO 2 FUNDAMENTOS TEORICOS	8
2.1. GENERALIDADES DE LOS SISTEMAS OPERATIVOS	8
2.2. GENERALIDADES DE LAS BASES DE DATOS	16
2.3. CARACTERÍSTICAS DEL MODELO LÓGICO	20
2.4. CARACTERÍSTICAS DEL MODELO FÍSICO	23
2.5. CARACTERÍSTICAS DE LOS DISPOSITIVOS FÍSICOS DE ALMACENAMIENTO.....	25
2.6. NATURALEZA DE LAS TABLAS DE LAS BASES DE DATOS	26
CAPITULO 3 DISEÑO DEL ALGORITMO GENETICO COMO SOLUCION DEL PROBLEMA.....	34
3.1. DISEÑO DEL ALGORITMO GENÉTICO	34
3.2. JUSTIFICATIVO PARA EL USO DEL ALGORITMO GENÉTICO	43
CAPITULO 4 BENCHMARK.....	45
4.1. JUSTIFICACIÓN	45
4.2. SERVIDOR IBM XSERIES 235	46
4.3. EQUIPO PARA ALMACENAMIENTO MASIVO (SAN) STORAGETECH.....	47
4.4. SERVIDOR INTEL SBT2 (1).....	48
4.5. SERVIDOR INTEL SBT2 (2).....	49
4.6. SERVIDOR IBM NETFINITY 3500.....	50
4.7. SERVIDOR HP PROLIANT DL 380	50

4.8.	TIPOS DE PRUEBAS	51
4.8.1.	CONFIGURACIÓN 1	53
4.8.2.	CONFIGURACIÓN 2	54
4.8.3.	CONFIGURACIÓN 3	55
4.9.	COMPARACIÓN DE RESULTADOS.....	56
CAPITULO 5 IMPACTO FINANCIERO.....		60
5.1.	BENEFICIOS CUALITATIVOS	60
5.2	BENEFICIOS CUANTITATIVOS.....	61
5.3.	RECURSOS Y COSTOS	62
5.3.1	BENEFICIOS CUALITATIVOS	62
5.3.2	RECURSOS Y COSTOS	63
5.3.3	RECURSOS ADICIONALES REQUERIDOS PARA ADMINISTRAR EL SISTEMA ..	64
5.3.4	COSTO DEL SOFTWARE.....	64
5.4.	ROI.....	65
CONCLUSIONES Y RECOMANDACIONES.....		66
BIBLIOGRAFIA		68

ÍNDICE DE ABREVIATURAS

Abreviatura	Ingles	Español
DBA	Database administrator	Administrador de base de datos
DB / BDD	Database	Base de datos
SYOPBASE		Sistema optimizador de base de datos
DBMS / SGBD	Database Management System	Sistema gestor de bases de datos
RAID	Redundant Array of Independent Disks	Arreglo redundante de discos independientes
FMS	Fortran Monitor System	Sistema monitor de Fortran (Sistema Operativo)
IBSYS / SOS	SHARE Operating System	Sistema operativo de IBM
IBM	International Business Machines Corporation	
OS / SO	Operating System	Sistema Operativo
MS DOS	Microsoft Disk Operating System	Sistema operativo de Microsoft perteneciente a la familia DOS.
THE	Technische Hogeschool Eindhoven	Universidad Técnica de Eindhoven
E/R	Entity-Relationship	Entidad relación
DED	Data Element Definition	Diagrama estructura de datos
DDL	Data Definition Language	Lenguaje de definición de datos
I/O	Input / output	Entrada / salida
CASE	Computer Aided Software Engineering	Ingeniería de Software Asistida por Ordenador
GIS	Geographic Information System	Sistemas de información Geográfica
ANSI	American National Standards Institute	Instituto Nacional de Estándares Americanos
SPARC	Scalable Processor ARChitecture	Arquitectura escalable para procesador
SAN	Storage Area Network	Red de área de almacenamiento
DINFOR		Dirección de Informática de la Armada
DIGPER		Dirección General del Personal
DIGPERGYE		Base de datos de personal de la DIGPER Guayaquil

ÍNDICE DE FIGURAS

Figura 1.1 Niveles de arquitectura de bases de datos	5
Figura 2.1 Sistema operativo de estructura monolítica.	10
Figura 2.2 Sistema operativo de estructura jerárquica.	11
Figura 2.3 Sistema operativo de estructura anillos concéntricos.....	12
Figura 2.4 estructura de máquina virtual.....	13
Figura 2.5 Estructura de datos jerárquica.....	18
Figura 3.1 Diseño de la base de datos	43

ÍNDICE DE TABLAS

Tabla 2.1 Características principales a considerar	24
Tabla 2.2 Tabla de resumen de análisis de las bases de datos.....	31
Tabla 2.3 Matriz relación Naturaleza de tabla vs. características de dispositivo	32
Tabla 3.1 Matriz disco	35
Tabla 3.2. Matriz tabla	35
Tabla 3.3. Matriz pesos tablas vs. disco.....	36
Tabla 3.4. Matriz tabla vs. disco convertida	37
Tabla 3.5. Matriz tabla vs. disco depurada.....	38
Tabla 3.6. Matriz sumatoria tabla vs. disco.....	39
Tabla 3.7. Matriz de tablas pares.....	39
Tabla 4.1 Servidor IBM x Series 235.....	46
Tabla 4.2 Equipo SAN STORAGE TECH	47
Tabla 4.3 INTEL SBT2 (1)	49
Tabla 4.4 INTEL SBT2 (2)	50
Tabla 4.5 Tipos de pruebas DIGPERGYE	52
Tabla 4.6 Características Base Sueldos	53
Tabla 4.7 Configuración No. 1	53
Tabla 4.8 Configuración No. 2	54
Tabla 4.9 Configuración No. 3	55
Tabla 4.10 Comparación de resultados	56
Tabla 4.11 Resumen de resultados.....	57
Tabla 5.1 Recursos y costos	62
Tabla 5.2 Recursos utilizados en implementación.....	63
Tabla 5.3 Recursos mínimos necesarios	63
Tabla 5.4 Recursos adicionales	64
Tabla 5.5 Costo del CASE	64

ÍNDICE DE ANEXOS

<u>ANEXO A</u>	(No. registros y columnas DIGPERGYE)
<u>ANEXO B</u>	(Promedios escritura DIGPERGYE)
<u>ANEXO C</u>	(Promedios lectura DIGPERGYE)
<u>ANEXO D</u>	(No. registros y columnas SUELDOS)
<u>ANEXO E</u>	(Promedios escritura SUELDOS)
<u>ANEXO F</u>	(Promedios lectura SUELDOS)
<u>ANEXO G</u>	(Codigo de case implementado)
<u>ANEXO H</u>	(Matriz posibles cruzamientos de tablas)
<u>ANEXO J</u>	(Cotizacion empresa MAGNANI)
<u>ANEXO K</u>	(Cotizacion Empresa TECNOPLUS)
<u>ANEXO L</u>	(Cotizacion Empresa MC System)
<u>ANEXO M</u>	(Cotizacion Empresa ORCOIN S.A.)
<u>ANEXO N</u>	(Cotizacion Empresa TECMOCOMP)
<u>ANEXO O</u>	(Cotizacion FACTURA MC SYSTEM)
<u>ANEXO P</u>	(Trigger de la tabla PERSONA)
<u>ANEXO Z</u>	(Manual de Usuario)

CAPITULO 1 INTRODUCCION

1.1.Descripción del problema

Como es bien conocido las bases de datos "... no son simplemente un conjunto de archivos, en vez de ellos, una base de datos es una fuente central de datos que está pensada para que sea compartida por muchos usuarios con una diversidad de aplicaciones."⁽¹⁾ Cuyos datos se encuentran organizados y almacenados en diferentes dispositivos, la cual puede ser manipulada por medio de programas que nos permiten un acceso directo a los mismos, para de esta forma obtener: la información requerida en mucho menos tiempo, manejo de mayor cantidad de datos, manejo de varios usuarios simultáneamente (concurrentes), la independencia de su uso, coherencia de resultados y el hecho de evitar la redundancia entre muchas otras virtudes.

⁽¹⁾ Kendal & Kendal, ANALISIS Y DISEÑO DE SISTEMAS, tercera edición, México, 1997, Pág. 588.

Es así como surgen las bases de datos a mediados de los años sesenta sin embargo no fue sino hasta 1970 en que Codd⁽²⁾ propone el modelo relacional en donde los datos se estructuran lógicamente formando relaciones entre las tablas que lo componen, su objetivo primordial era el de mantener la independencia de su estructura lógica con su modo de almacenamiento. Es el nacimiento de los sistemas de gestión de datos conocidos como SGBD o sus siglas en inglés DBMS (DataBase Management System).

Muchos autores consideran la existencia de dos niveles en las bases de datos para el tratamiento de las entidades o de los objetos los cuales son: nivel lógico y el nivel físico; el primero (conocido muchas veces como conceptual) se refiere a las características de identificación de las entidades junto con su descripción y el lugar que ocupa en la organización; en cambio el nivel físico se refiere a la forma, modo, tipo, lugar de almacenamiento o repositorio físico en donde se almacenan los datos.

De la misma manera existen otros autores que consideran a un tercer nivel conocido como el nombre de Nivel Visión, el cual es parte del conceptual, el cual es el

⁽²⁾Edgar F. "Ted" Codd (Agosto 23, 1923 – abril 18, 2003), científico británico que contribuyó con la teoría relacional de bases de datos.

encargado de dividir a la base de datos en pequeñas parcelas útiles para los usuarios, un ejemplo clásico ha mencionar es el hecho de que un empleado que no debe tener acceso al sueldo de sus compañeros o superiores; este hecho esta relacionado con un Esquema de Visión.

Como se hace mención en los párrafos anteriores generalmente el nivel físico es facilitado por los SGBD, lo que quiere decir que el usuario esta en capacidad de decidir el lugar en donde se realizará la operación de almacenamiento; sin embargo esta operación se realiza sobre la base de la experiencia propia del DBA, lo cual ocasiona una operación poco eficiente con los recursos propios, e ineficaz puesto que se basa únicamente en las observaciones anteriores realizada por el mismo. El problema se ve agravado por el hecho que existe la posibilidad que el DBA no tenga la experiencia suficiente para determinar el lugar mas adecuado para el almacenamiento.

En este punto se considera la existencia de un divorcio entre el conocimiento de los datos a ser almacenados (con las características propias de cada uno de ellos) y el lugar establecido como repositorio de datos en donde serán almacenados. Para hacer esta elección de lugar no se ha tomado en cuenta ninguna de las características propias ni de las bases, ni de los datos, ni del software por una parte y por otra no se

ha considerado características de los discos, tipo de RAID a ser utilizado, velocidades de operación, almacenamiento o algún tipo de degradación.

Es el momento en el cual la interrogante para resolver este problema mediante la creación de un CASE capaz de guiar al DBA, a realizar un trabajo eficaz y eficiente en la administración de las mismas considerando factores que hasta el momento se los había considerado empíricamente o tal vez no se los había considerado.

1.2. Ambientes donde se produce el problema

La reunión de ciertas condiciones tales como excesiva cantidad de tablas, demasiados datos, utilización de diversos tipos de arreglos de discos para el almacenamiento, tipo de operación a realizar con las mencionadas tablas y datos, crecimiento indiscriminado de datos a ser guardados, hacen un ambiente propicio y adecuado para dificultar la toma de decisiones. Por lo cual pensamos que los ambientes más

propensos a este problema son los sistemas bancarios, sistemas de simulación, sistemas creadores de modelos (tales como el GIS⁽³⁾) entre otros.

Específicamente el problema surge en los dos niveles inferiores, entre el nivel interno y el nivel conceptual, adicionalmente se considera como un problema netamente del DBA, pues este será la persona encargada de tomar decisiones sobre los posibles lugares de almacenamiento de las diferentes tablas de las bases de datos.

Las bases de datos en general tienen una arquitectura de tres niveles (definida por el grupo ANSI⁽⁴⁾/SPARC⁽⁵⁾), la cual puede apreciarse en la figura 1.1:

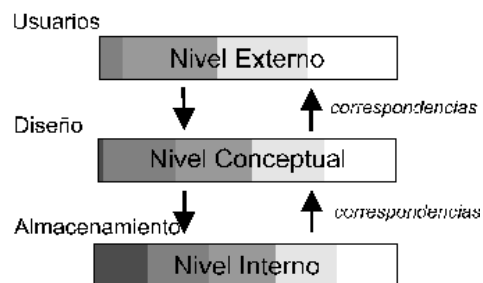


Figura 1.1 Niveles de arquitectura de bases de datos

⁽³⁾ GIS: (Geographical Information System) Sistema de Información Geográfica.- Es una metodología para adquirir, almacenar, gestionar, editar y mostrar datos relacionados con mapas y planos geográficos.

⁽⁴⁾ ANSI: (American National Standard's Institute) Organización americana para definir estándares.

⁽⁵⁾ SPARC: Estandar RISC desarrollado por Sun Microsystems.

1.3.Solución propuesta y su factibilidad

Una solución para el problema en mención y basados en la aplicación de un algoritmo genético, el cual toma la naturaleza de las tablas, las propiedades de los arreglos, las características técnicas de los discos, para de esta manera indicar los repositorios adecuados para los datos por medio de una selección natural basada en genética.

Este algoritmo esta en capacidad de crear un conjunto de soluciones óptimas las cuales se almacenan en una base de datos que se considera como el conjunto de soluciones universo, de este universo y aplicando nuevamente el algoritmo se obtiene una población de soluciones las cuales son por selección natural mejores respecto a las anteriores y así sucesivamente, de esta manera va evolucionando cada vez más, se logra conseguir individuos óptimos, los cuales han sido el producto de la selección del universo de las mejores poblaciones y los mejores individuos.

El hecho de poder abstraer todas las características mencionadas y luego plasmarlas en código ha permitido optimizar las acciones realizadas por los DBA en cuanto se refiere a la toma de decisión del lugar más adecuado para su almacenamiento. Adicionalmente se puede mencionar que en el mercado actual existen pocos

programas capaces de realizar las acciones antes mencionadas para automatizar estas funciones tales como el RSS⁽⁶⁾ (Research Storage System), el RSI⁽⁷⁾ (Research Storage Interface).

Cabe mencionar que el sistema implementado consideró las velocidades estándares de los de los discos (no las optimizadas por el RAID), de igual manera no se consideró el almacenamiento de los índices de las bases de datos.

1.4. Objetivos de la solución

1. Optimizar la capacidad de almacenamiento de los discos utilizados en los arreglos de discos.
2. Optimizar la velocidad de los procesos realizados en las diferentes transacciones ordinarias de las bases de datos.
3. Disponer de varias alternativas óptimas para elección de almacenamiento.
4. Crear modelos basados en una base de datos.

⁽⁶⁾ RSS es el subsistema de almacenamiento de system R, este es el responsable de mantener el almacenamiento físico de las relaciones, de los caminos de acceso a ellas, bloqueos (en un ambiente multiusuario), y de las capacidades de registro de transacciones y recuperación.

⁽⁷⁾ RSI (Research Storage Interface) es la interfaz orientada a la tupla que system R presenta al usuario final.

CAPITULO 2 FUNDAMENTOS

TEORICOS

2.1.Generalidades de los Sistemas Operativos

A partir de los años 50 surge la necesidad imperiosa del uso de la computadora, y sus primeras versiones eran simplemente tableros con capacidad de ser conectados (primera generación), posteriormente apareció la segunda generación implementando el trabajo en lotes (sistemas como FMS⁽⁸⁾ o el IBSYS⁽⁹⁾). Con el surgimiento de las computadoras de la tercera generación aparecen los sistemas operativos OS/360 de IBM⁽¹⁰⁾, es importante destacar que en este período también aparece el concepto de

⁽⁸⁾ FMS (Fortran Monitor System) fue el sistema concebido específicamente para trabajar en el lenguaje de programación FORTRAN y así algunos llevar al máximo su potencial.

⁽⁹⁾ IBSYS es un sistema operativo creado por IBM en los años 60 para el 7094.

⁽¹⁰⁾ IBM (International Business Machines) es una de las compañías más grandes de computadoras en el mundo. IBM empezó en 1911 como un productor de máquinas para tabular fichas perforadas. En 1953, introdujo su primera computadora, el 701. Durante el 60's y 70's, IBM vino a dominar el campo nuevo de central y las minicomputadoras. En 1981, IBM lanzó su primera computadora personal.

multiprogramación aunque en general se los considera de lote. El avance de la electrónica a gran escala provocó que en la cuarta generación se hable de computadoras personales y estaciones de trabajo con interfases amigables por lo que se hacen populares el MS-DOS⁽¹¹⁾ y UNIX⁽¹²⁾ en las mencionadas computadoras.

Se da un brinco gigantesco cuando en los años 80 comienza el auge de las redes de computadoras y por ende la necesidad de sistemas operativos capaces de administrar redes y sistemas distribuidos. El crecimiento indiscriminado en todo el mundo de la Internet hace posible el acceso a personas e instituciones a toda la información disponible en sus computadoras personales, claro está que para la época existía una diversidad de sistemas operativos por lo cual se busca una solución para hacer trabajar a estos sistemas operativos diferentes. Sin embargo no fue sino hasta la década de los 90, época en la cual se crean las aplicaciones para ser ejecutadas en una plataforma y los resultados pueden ser observados en otra diferente por lo que se verifica que el nivel de interacción cada vez se hace más profundo.

⁽¹¹⁾ MS DOS (MicroSoft-Disk Operating System) es el sistema operativo creado para las Pc de Microsoft, cuyo funcionamiento es muy parecido a la versión del IBM para sus computadoras personales, la peculiaridad de llamarlos por DOS se debe a que anteriormente los sistemas operativos obligatoriamente arrancaban desde el disco.

⁽¹²⁾ Unix: es uno de los más populares sistemas operativos multiusuario y multitarea desarrollado por Labs Bell en los años 70 con las consideraciones que sea liviano, flexible, barato y de uso exclusivo de los desarrolladores. Fue el primer SO desarrollado en C, lo que significa que puede ser instalado prácticamente en cualquier Pc.

Algunos autores consideran que los sistemas operativos se basan en tres clasificaciones las cuales se mencionan a continuación: sistemas operativos según su estructura, sistemas operativos de acuerdo a los servicios que ofrecen y finalmente sistemas operativos de servicios para los usuarios.

Los sistemas operativos de acuerdo a su estructura cumplen con dos requisitos que son: fácil de usar y de aprender; pero en cuanto a la parte técnica considera: mantenimiento, forma de operación, restricciones de uso, eficiencia, tolerancia a errores y flexibilidad. Tomando en cuenta su estructura podemos mencionar:

- a. Estructura monolítica.- cuentan con un solo programa entrelazados por un conjunto de rutinas, de tal manera que se puedan llamar entre sí. Su construcción final son módulos compilados por separado.

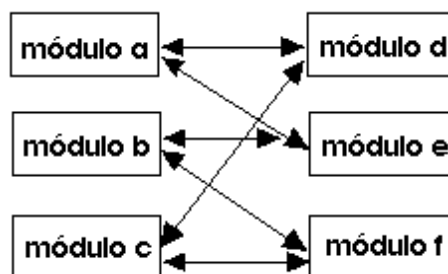


Figura 2.1 Sistema operativo de estructura monolítica.

- b. Estructura jerárquica.- se dividió al sistema operativo en pequeñas partes las cuales tenían una interfase con el resto de elementos, adicionalmente su

organización era por niveles. Esa fue la razón para denominarla estructura jerárquica o de niveles; el primero creado fue denominado THE⁽¹³⁾ (Technische Hogeschool Eindhoven), perteneciente a Dijkstra⁽¹⁴⁾. Otros autores los consideran de multicapas. Su estructura es la base para la actual estructura de muchos sistemas operativos.

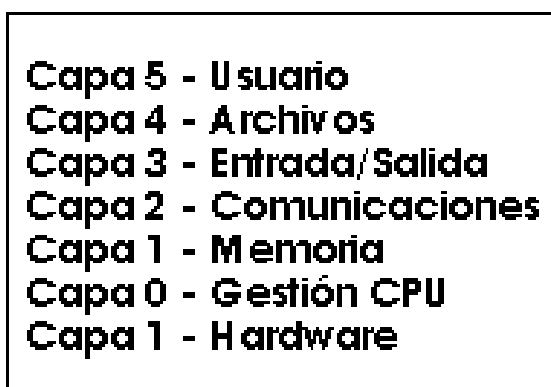


Figura 2.2 Sistema operativo de estructura jerárquica.

Otra forma de denominación es la de estructura de anillos concéntricos o rings.

⁽¹³⁾ THE (Technische Universiteit Eindhoven o TU/E) es una universidad técnica localizada en Eindhoven, (Holanda). Fue fundada el 15 de junio de 1956 por el gobierno holandés. Era la segunda institución de su clase en los países bajos, sólo ha sido precedida por la Universidad Tecnológica de Delft. Originalmente, se llamó el Eindhoven (EL) de Technische Hogeschool.

⁽¹⁴⁾ Edsger Wybe Dijkstra fue uno de los miembros más influyentes de la generación fundadora de la ciencia informática. Sus contribuciones científicas más importantes son los lenguajes de programación de diseño del algoritmo, sistemas operativos de diseño distribuidos procesando el diseño formal de la especificación y comprobación de argumentos matemáticos además, Dijkstra se interesó intensamente en la enseñanza, y en las relaciones lo académico, la ciencia y la industria de software. Durante sus cuarenta años como científico, logro posicionarse como uno de los mejores en el campo de la informática, así mismo en la industria y por sus contribuciones le otorgaron muchos premios inclusive el honor más alto de la ciencia, el Premio de ACM Turing.

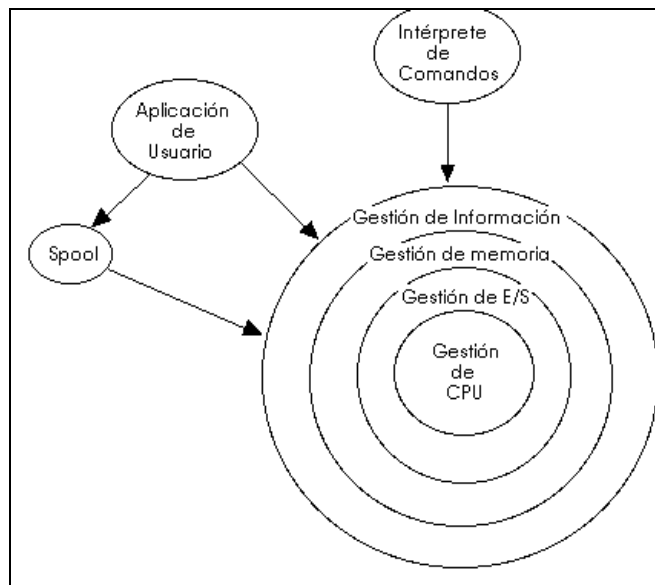


Figura 2.3 Sistema operativo de estructura anillos concéntricos.

En cada uno de los anillos se pueden apreciar puertas de acceso o trampas (traps), que sirven de ingreso para las partes inferiores esta es la manera para proteger las partes internas de accesos no deseados.

- c. Estructura de maquina virtual.- en este tipo de sistema operativo cada proceso presenta una interfase la cual es idéntica a la maquina real, en este tipo de estructura se separan la multiprogramación y la máquina extendida, la idea principal es la de dar la sensación de disponer de varias máquinas diferentes las cuales pueden ejecutar sistemas operativos diferentes como se puede apreciar en la figura 2.4.

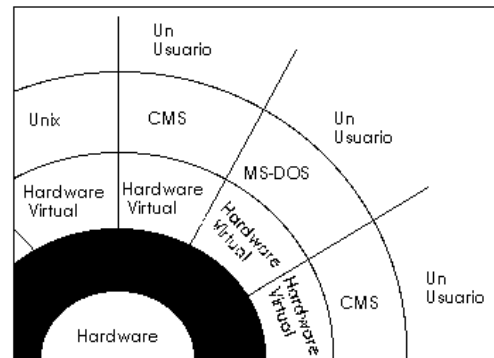


Figura 2.4 estructura de máquina virtual

- d. Estructura cliente servidor.- son de propósito general y sirve para toda clase de aplicaciones; puede ser ejecutado en la mayoría de las computadoras. Lo mas destacado es que los procesos pueden ser tanto en clientes como en servidores.

De la misma manera podemos diferenciar a los sistemas operativos por el tipo de servicios para lo cual los hemos dividido de la siguiente manera:

- División por el número de usuarios
 - a. Monousuarios.- son aquellos SO que soportan únicamente a un usuario sin tomar en cuenta al número de procesadores que disponga la computadora o número de tareas o procesos que tenga que realizar en un instante de tiempo.

- b. Multiusuarios.- estos sistemas operativos permiten dar servicio a más de un usuario a la vez, por medio de accesos remotos o la utilización de varias terminales conectadas a la computadora mediante el uso de una red.
- División por el número de tareas
 - a. Monotarea.- la característica principal de estos SO es que permiten la realización de una tarea a la vez por usuario, es importante mencionar que se permite la combinación de multiusuario monotarea, es decir muchos usuarios realizando una sola tarea.
 - b. Multitarea.- permite al usuario realizar muchas labores al mismo tiempo, como ejemplo se puede editar el código fuente mientras se compila otro programa y al mismo tiempo recibir un mensaje electrónico, lo cual aumenta notoriamente su productividad.
- División por el número de procesos
 - a. Uniproseso.- este SO es capaz de manejar únicamente un procesador en la computadora.
 - b. Multiproseso.- se refiere al trabajo específico de varios procesadores del sistema de tal manera que se puede distribuir la carga del proceso; se los considera de dos formas de trabajo: simétrica si todos los procesos o parte de ellos conocidos como (threads), son ejecutados indistintamente en

cualesquiera de los procesadores disponibles por medio de un algoritmo de mejor distribución y de manera asimétrica en el cual uno de los procesadores será considerado como principal y los demás como esclavos para la ejecución de los mencionados procesos.

La siguiente división de los sistemas operativos es dependiente de los servicios que ofrecen (es decir es una visión desde el punto de vista del usuario) la que se puede apreciar mediante la siguiente división:

- a. SO de red.- son aquellos que nos permiten interactuar con los SO de otras computadoras por algún medio de transmisión con el afán de realizar un sin número de actividades entre sí, tales como transferencia de archivos, accesos remotos, instalar utilitarios, tener acceso a bases de datos etc., es decir lo más importante es disponer del recurso y poderlo compartir.
- b. SO distribuidos.- en realidad no se ha desarrollado un SO distribuido a plenitud pues la idea principal es la de abarcar todos los servicios de red integrando los recursos en una sola máquina virtual a la cual se puede acceder de manera transparente, necesitando conocer únicamente el nombre y de esta manera los recursos como si fuesen locales. El grave problema resulta de la complejidad de distribuir los procesos en varias unidades de procesamiento, reintegrar los resultados al mismo tiempo de resolver

problemas de concurrencia y paralelismo. Las ventajas principales que se puede mencionar son: velocidad, capacidad de crecimiento, confiabilidad y costos. En cambio las desventajas se basan el aspecto de concurrencia y paralelismo. Puesto que debido a la concurrencia se debe implantar mecanismos para tratar de evitar la competencia entre procesos, evitar una postergación indefinida, evitar condiciones circulares y el hecho de ocupar recursos en espera de otro. En realidad todos estos problemas presentan la mayoría de los sistemas operativos multiusuarios y multitareas; claro está que para los sistemas operativos distribuidos el problema es aún más complejo, razón por la que se necesitará algoritmos mejor desarrollados. Debido al paralelismo el solo hecho de encontrar las secciones de algún código paralelizable es un trabajo intenso, sin embargo es necesario dividir a un proceso en varios sub-procesos y redirigirlos a diferentes unidades de procesamiento para su distribución.

2.2 Generalidades de las bases de datos

Ampliando aquellos conceptos que se pudieron apreciar en la Descripción del problema perteneciente al Capítulo I, en donde se hace mención brevemente a las

bases de datos, consideramos que es importante mencionar que las bases de datos deben cumplir con algunos objetivos tales como::

1. Todos los usuarios compartan la base.
2. Controlar la veracidad y consistencia de los datos.
3. Disponibilidad de los datos.
4. Considerar la evolución y crecimiento de la base de datos.
5. Todo el proceso sea transparente para el usuario.

Un punto extremadamente importante es como guardar los datos, es así como se considera que existen dos maneras para realizar esta labor, la primera es crear archivos independientes e individuales y la segunda y más desarrollada en la actualidad es la creación de una base de datos que pueda ser compartida por una infinidad de usuarios y una gran cantidad de aplicaciones, lo cual resulta muy útil pues se facilita la captura de datos, almacenamiento y actualización por una sola vez, para lo cual es necesario conocer tres conceptos claramente: el mundo real, los datos y los metadatos.

El mundo real representa todas aquellas entidades, objetos o eventos de los cuales deseamos recolectar y almacenar datos relevantes, sus atributos son las características de estas entidades. En cambio los datos son los valores que pueden ser organizados en

registros los cuales serán accedidos por medio de claves o llaves. Finalmente los metadatos describen a los datos de acuerdo a las restricciones impuestas a estos.

De esta manera consideramos que las bases de datos pueden tener tres tipos de estructuras que son:

1. Estructuras de datos jerárquicos- conocidas también como árboles, implican que una entidad no puede tener más de una entidad que la posea. Es decir se la considera como una estructura compuesta de muchas asociaciones 1:M o 1:1; otros tipos de asociaciones tales como M: 1 o M:N no son permitidas.

Gráficamente se la puede apreciar de la siguiente manera:

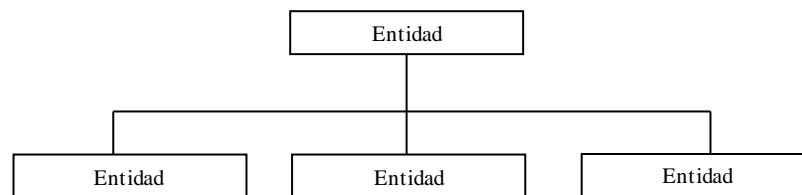


Figura 2.5 Estructura de datos jerárquica

Algunos autores la consideran como ramificaciones donde una entidad puede tener ramificaciones subordinadas semejantes a las ramas de un árbol

2. Estructura de datos en red.- este tipo de estructura permite tener entidades con cualquier cantidad de subordinados o superiores las cuales se encuentran vinculadas por medio de enlaces que representan a conceptos de datos comunes a ambas entidades conectadas. Es una estructura mucho más compleja que su predecesora, su principal ventaja es que el mismo dato

puede ser utilizado por varias entidades, la desventaja por el contrario es que fácilmente podría existir redundancia de datos, en la figura a continuación hemos representado al mundo real como registros lógicos que se relacionan entre sí por medio de flechas hacia los respectivos datos comunes.

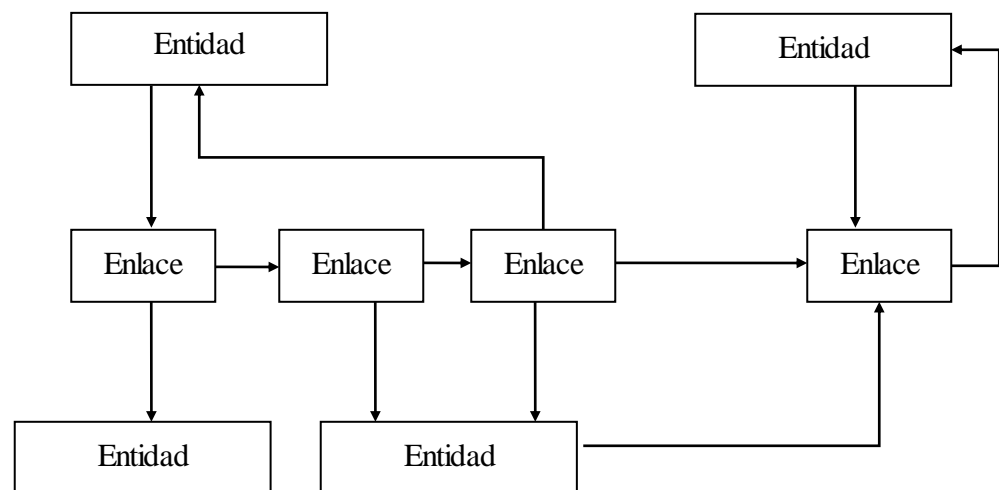


Figura 2.6 Estructura de datos en red

3. Estructura de datos relacional.- representan una o varias tablas bidimensionales (conocidas como relaciones), los renglones de las tablas contienen registros mientras que las columnas representan a sus atributos. Podemos llamar relacional a la base de datos construida por relaciones entre dos tablas o más, las cuales disponen de ventajas como eficiencia en las consultas y factibilidad de crecimiento en la base de datos.

2.3. Características del Modelo Lógico

Considerando que cualquier cambio en la estructura física o lógica de los datos afectaba a las aplicaciones directamente, ANSI publicó la definición de una arquitectura de tres niveles la cual se debe seguir para el diseño de la base de datos en donde hace énfasis a la independencia entre las referencias externas a los datos y la forma física de almacenamiento, considera también la organización estos niveles de la siguiente manera:

1. Nivel Externo.- este es únicamente de interacción con el usuario en donde se representa una visión parcial de los datos respecto al usuario, muestra únicamente la parte que le interesa al usuario.
2. Nivel Conceptual.- este en realidad representa el esquema lógico de los datos en donde se visualiza su estructura y sus relaciones, sin embargo no considera especificaciones físicas. En este punto se excluyen datos innecesarios y se define que alternativa de modelo se va a utilizar.
3. Nivel físico.- este debe ser transparente para el usuario y va a depender de la estructura de los datos, hardware, software, así como el modo de almacenamiento de los datos, sistemas operativos utilizados, SGBD etc....

El modelo conceptual es un modelo abstracto que pertenece al nivel externo que permite al lector tener una visión general del asunto a tratar, es decir son todos los procesos del caso en estudio pero con una visión global de la situación que permitirá obtener una imagen amplia de todo el proceso; sin pensar en las limitaciones físicas, ni lógicas, ni estructura alguna, adicionalmente no considera al tipo de base de datos, ni tampoco al SGBD. El modelo entidad relación es un modelo conceptual, (fue propuesto inicialmente por CHEN⁽¹⁵⁾ para posteriormente dar lugar al modelo E/R extendido); es decir se puede expresar que el modelo conceptual toma a las relaciones establecidas entre las entidades participantes en el tema a solucionar.

Los componentes del modelo E/R⁽¹⁶⁾ son los siguientes: entidades, relación, grados de relación, cardinalidad, atributos. Adicionalmente se pueden mencionar conceptos como: jerarquía, agregación y exclusividad.

Una vez establecido correctamente el modelo conceptual, se considera pertinente continuar con el modelo lógico perteneciente al nivel conceptual, denominado por

⁽¹⁵⁾Dr. Peter P. Chen es el científico creador del modelo entidad relación, su labor se la considera como la piedra angular de dotación lógica, su trabajo de investigación y metodologías estructuradas han hecho un impacto significativo en la mayoría de las herramientas tipo CASE.

⁽¹⁶⁾Llamado también un modelo de la entity relationship, es una representación gráfica de las entidades y sus relaciones uno al otro, típicamente utilizado en los sistemas de bases de datos. Una entidad es un grupo de datos, un objeto es el concepto genérico de los datos que se almacenan. Una relación es cómo los datos se comparten entre entidades.

algunos autores como Esquema de Base de Datos (Database Schema), y como se verificará posteriormente será el punto de partida para realizar el modelo físico. En este modelo lógico transformamos las entidades obtenidas y relaciones aceptadas en el modelo anterior a tablas, para lo cual utilizamos las reglas de normalización.

Cada entidad del modelo conceptual será transformada en una tabla y los atributos de dicha entidad serán los atributos de dicha tabla, las relaciones de muchos a muchos se transforman en tablas en donde la clase estará formada por la clave primaria de las entidades relacionadas; en cambio las relaciones de uno a muchos se propaga la clave principal de la entidad cuya cardinalidad es 1 a la de cardinalidad M. Podemos valernos de los Diagramas de Estructura de Datos (DED).

Es importante determinar que este tipo de esquema permite relaciones entre dos entidades; bajo la salvedad de modelar relaciones de dos o más entidades se deberán redefinir todas a relaciones binarias.

Existe una correspondencia entre los diagramas E/R y los DED, permitiendo el paso de uno a otro y viceversa.

Los problemas más importantes que se pueden evidenciar son: redundancia de datos y ambigüedades entre datos; de esta manera surgen las soluciones a los mismos como las reglas de normalización, para lo cual consideramos los conceptos de: superclave, clave primaria, clave candidata; dependencia funcional, transitiva y completa.

Las principales formas de normalización son las siguientes:

1. Primera Forma Normal (1FN).
2. Segunda Forma Normal (2FN).
3. Tercera Forma Normal (3FN).
4. Forma Normal de Boyce Codd⁽¹⁷⁾
5. Cuarta Forma Normal (4FN).
6. Quinta Forma Normal (5FN).

2.4. Características del Modelo Físico

Una vez que hemos finalizado con los dos pasos anteriores el lector del presente estará en capacidad de diferenciar el modelo de las tablas (más sus derivadas de las

⁽¹⁷⁾ Edgar F. Codd científico que contribuyó notablemente en el estudio del modelo entidad relación y a quien se le atribuye una de las formas normales de las bases de datos.

relaciones), lo que representa una metáfora de la conversión de esta tabla en un mapeo físico para lo cual se debe considerar el conocimiento de características tales como se presenta en tabla 2.1:

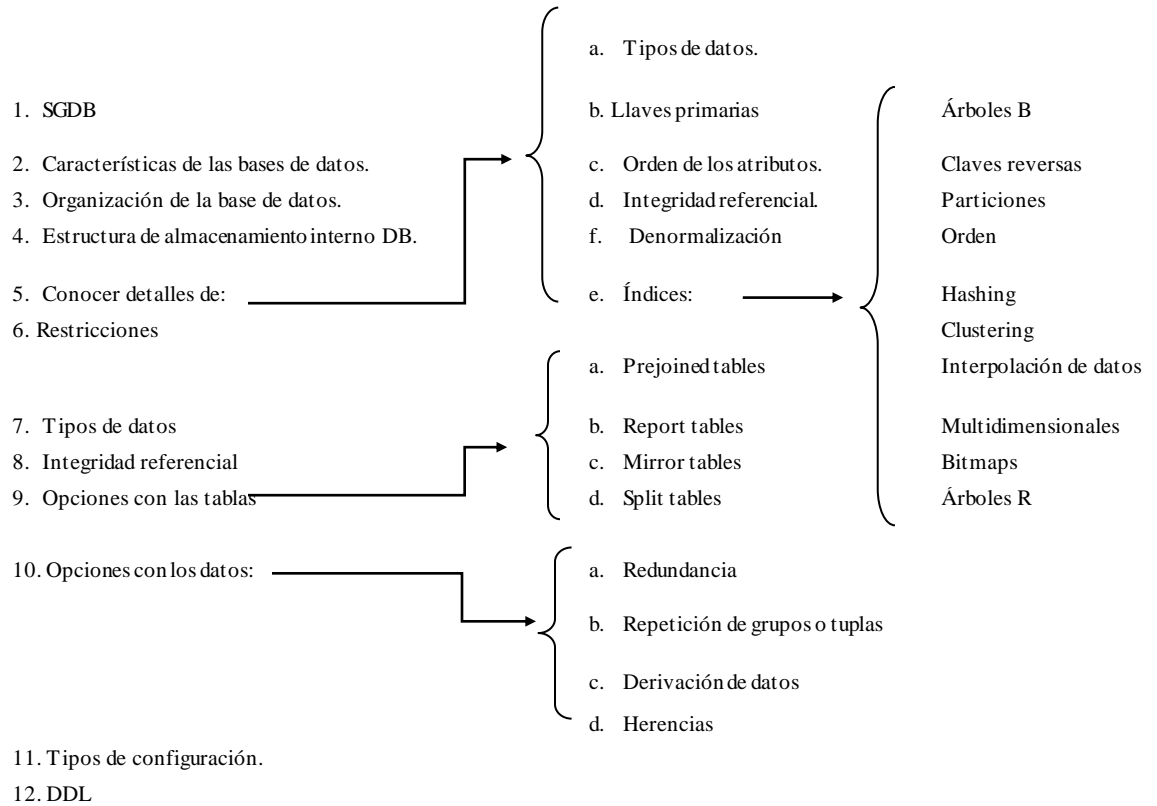


Tabla 2.1 Características principales a considerar

2.5. Características de los dispositivos físicos de almacenamiento

Como es bien conocido los datos contenidos en las tablas será almacenada en discos duros los cuales disponen de ciertas características propias de cada uno de ellos, las principales características se las mencionan a continuación:

1. Capacidad de almacenamiento.- Es una característica propia del disco que limita la cantidad de información que puede ser almacenada.
2. Rendimiento.- Es la suma de factores que implican la capacidad de los discos para efectuar operaciones de I/O, es importante mencionar que los discos dependerán de dos factores importantísimos, factores mecánicos y factores electrónicos.
3. Velocidad.- Es la característica que nos indica las revoluciones por minuto del dispositivo en mención
4. Facilidad de manejo.- Es la habilidad que brinda al operador a fin de realizar las tareas de lectura, escritura, almacenamiento y borrado.
5. Costos.- Es el valor económico que implica la utilización de los mencionados dispositivos.
6. Escalabilidad.- Es la capacidad que tienen los dispositivos que le permiten aumentar en cantidades y seguir trabajando manteniendo o mejorando su rendimiento

7. Tipos de RAID.- Es el conjunto de discos redundantes e independientes usados a fin de preservar la integridad de los datos, mejorar rendimiento y mejorar la tolerancia a errores.

2.6 Naturaleza de las tablas de las bases de datos

Consideramos que las tablas que componen las bases de datos, tienen propiedades entre las cuales mencionamos a:

1. Propiedad de acceso- son todas aquellas tablas a las cuales se las utiliza en las bases de datos.
2. Propiedad de lectura- son todas aquellas tablas las cuales han sido leídas para la obtención de diferentes datos.
3. Propiedad de modificación- son todas aquellas tablas las cuales han sido cambiadas o alteradas de alguna manera.
4. Propiedad de consulta- son aquellas tablas las cuales se han utilizado para realizar las diferentes consultas de los procesos realizados.

Para el trabajo en mención hemos dividido a las tablas de la base de datos de DIGPERGYE en dos cuyas características son:

1. Propiedad concurrente de escritura.- son aquellas que han sido alteradas y nuevamente grabadas y las hemos dividido en:
 - a. Altamente concurridas de escritura.- hemos considerado valores promedio de 144.59 operaciones diarias, aunque el rango va desde 100 hasta 726.13 operaciones diarias. ([ver Anexo B](#))
 - b. Medianamente concurridas de escritura.- son aquellas que tienen interacciones entre 15 y 100 (diariamente), con un promedio de 67.65 operaciones diarias. ([ver Anexo B](#))
 - c. Bajamente concurridas de escritura.- son aquellas que tienen interacciones entre 0 y 15 (diariamente), con un promedio de 2.28 operaciones diarias, ([ver Anexo B](#)).
2. Propiedad concurrente de lectura.- son aquellas que no han sido alteradas sino que únicamente han sido accedidas como consulta y las hemos dividido en:
 - a. Altamente concurridas de lectura.- son aquellas tablas que han sido consultadas mas de 50 veces diarias, el promedio referencial tomado es de 258.76 consultas diarias. ([ver Anexo C](#)).

- b. Medianamente concurridas de lectura.- son aquellas que han sido consultadas entre 15 y 50 veces al día, con un promedio de consulta de 38.91 consultas diarias. ([ver Anexo C](#)).
- c. Bajamente concurridas de lectura.- son aquellas que tienen menos de 15 consultas al día, con un promedio de 2.39 consultas al día. ([ver Anexo C](#))

Adicionalmente se trabajó en la base de datos de SUELDOS cuyas características se mencionan a continuación:

- 1. Propiedad concurrente de escritura.-
 - a. Altamente concurridas de escritura.- el valor promedio es 80.7 operaciones diarias, aunque el rango va desde 50 hasta 142.4 operaciones diarias. ([ver Anexo E](#))
 - b. Medianamente concurridas de escritura.- son aquellas que tienen interacciones entre 15 y 50 (diariamente), con un promedio de 25.3 operaciones diarias. ([ver Anexo E](#))
 - c. Bajamente concurridas de escritura.- son aquellas que tienen interacciones entre 0 y 15 (diariamente), con un promedio de 1.6 operaciones diarias, ([ver Anexo E](#)).
- 2. Propiedad concurrente de lectura.-

- a. Altamente concurridas de lectura.- son aquellas tablas que han sido consultadas mas de 15 veces diarias, el promedio referencial tomado es de 28.9 consultas diarias, ([ver Anexo F](#)).
- b. Medianamente concurridas de lectura.- son aquellas que han sido consultadas entre 5 y 15 veces al día, con un promedio de consulta de 5.9 consultas diarias. ([ver Anexo F](#)).
- c. Bajamente concurridas de lectura.- son aquellas que tienen menos 5 consultas al día, con un promedio de 1.5 consultas diarias. ([ver Anexo F](#)).

Los valores mencionados fueron tomados como parámetros gracias al siguiente procedimiento:

1. Se determinó el número de tablas de la base de datos en donde se aplicará el algoritmo genético, (para nuestro caso específico fueron: DIGPERGYE 614 tablas, sueldos 124 tablas)
2. Se determinó el número de registros y el número de columnas en cada una de las tablas mediante la aplicación de las siguientes líneas de código (los resultados se pueden apreciar en el [anexo A](#) y [anexo D](#)):

```
select TRA_PER_ID, fec_mod  
  
from transaccion  
  
where TRA_TABLA = '-----nombre de la tabla-----'  
  
group by TRA_PER_ID, fec_mod
```

```
order by fec_mod
```

3. Se determinó el número de transacciones realizadas en cada una de las tablas que disponen de un trigger (disparador)⁽¹⁸⁾ para lo cual se utilizó las siguientes líneas de código (el resultado se lo puede apreciar en el [anexo B](#) y [anexo E](#)), los diferentes tipos de trigger se los aprecia en el [anexo P](#):

```
select year(fec_mod), TRA_TABLA, count(*)
from transaccion
group by year(fec_mod), TRA_TABLA
order by year(fec_mod), TRA_TABLA
```

4. Posteriormente se determinó el número de consultas realizadas en cada una de las tablas para lo cual se utilizó el siguiente código (el resultado se lo puede apreciar en el [anexo C](#) y [anexo F](#)):

```
select year(FECHA), count(*)
from TRANSACCION_CONSULTA
group by year(FECHA)
order by year(FECHA)
```

⁽¹⁸⁾ Disparador es un procedimiento especializado, que se almacenan dentro de la base de datos y es manejado por el DBMS. Este no puede ser llamado o ejecutado. El DBMS enciende automáticamente al disparador como resultado de una modificación de los datos a la tabla asociada. Los disparadores son utilizados para mantener la integridad de referencia de datos cambiando los datos en una manera sistemática.

5. La tabla de valores obtenidos se la promedio para el número de días que la base de datos ha sido utilizada arrojando los valores expuestos en el [anexo B](#) y [anexo C](#), (para el caso de la base de datos DIGPERGYE), los valores referenciales de días de uso datan desde el 11-ENE-2000 al 04-ABR-2005 (aproximadamente 1365 días). Igual procedimiento se realizo para la base de datos de sueldos con resultados expuestos en el [anexo E](#) y [anexo F](#), el periodo referencial es de 60 días.

El resumen de anexos de análisis se los puede apreciar en la siguiente tabla:

Tabla de resumen de Análisis			
Base de datos DIGPERGYE			
Promedios de escritura diaria:	144.5	66.7	2.28
Total tablas:	135	237	42
Promedios de lectura diaria:	258.8	38.9	2.4
Total tablas:	127	29	45
Base de datos SUELDOS			
Promedios de escritura diaria:	80.7	25.3	1.6
Total tablas:	31	17	8
Promedios de lectura diaria:	28.9	5.9	1.5
Total tablas:	24	23	21

Tabla 2.2 Tabla de resumen de análisis de las bases de datos

Tomando las características de los ítems 2.5 y 2.6 hemos creado la matriz relación naturaleza de tabla vs. características de dispositivo, la cual presentamos en la tabla 2.3:

Naturaleza de tabla	Característica del dispositivo		
	Rápido	Medianamente rápido	Lento
Altamente Accesada	10	5	0
Medianamente accesada	5	10	0
Poco acceso	0	5	10

Tabla 2.3 Matriz relación Naturaleza de tabla vs. características de dispositivo

En la matriz anteriormente indicada hemos calificado como optimas a las relaciones:

- Rápido vs. Altamente accesada (ideal)
- Medianamente rápido vs. medianamente accesada (ideal)
- Lento vs. poco acceso (ideal)

De la misma manera se ha calificado como regulares a las siguientes relaciones:

- Medianamente rápido vs. altamente accesada (con un disco medianamente rápido podría ser aceptable un acceso a datos elevado, aunque este no sea el ideal)
- Rápido vs. medianamente accesada (con un disco rápido podría ser aceptable un acceso mediano, aunque exista desperdicio de recursos)

- Medianamente rápido vs. poco acceso (con un disco de velocidad media podría ser aceptable tener tablas de poco acceso, aunque existe desperdicio de recursos)

De igual forma los autores del presente han calificado como relaciones no deseadas a:

- Lento vs. altamente accesada (al disponer de un disco lento la peor condición que pueda suceder es tener una tabla altamente accesada)
- Lento vs. medianamente accesada (sí el disco es lento una de las peores condiciones es que disponga de tablas medianamente accesadas)
- Rápida vs. poco accesado (si el disco en mención es rápido una de las peores condiciones sería que tenga tablas de muy poco acceso, puesto que estaríamos desperdiciando recursos)

CAPITULO 3 DISEÑO DEL ALGORITMO GENETICO COMO SOLUCION DEL PROBLEMA

3.1.Diseño del algoritmo genético

El algoritmo genético es un método repetitivo de búsqueda basado en la mecánica de la selección natural y de genética, mediante una secuencia modificada de instrucciones generadas de manera recursiva, de tal manera que “combina la supervivencia del más apto entre estructuras de secuencias con un intercambio de información estructurado, aunque aleatorio”⁽¹⁹⁾

Consideramos la creación de un algoritmo genético basada en una simple parametrización de matrices optimizando el lugar de almacenamiento en una base de datos.

⁽¹⁹⁾Goldberg, Glosario de Carlos Von Der Becke.

Utilizando las matrices de la tabla 3.1 y 3.2, posteriormente la combinación de las mismas basados en la tabla 2.2 obtenemos la matriz inicial del algoritmo expuesta en la tabla 3.3:

Matriz Disco	D1	D2	D3	D4	D5
	Rápido	Lento	Rápido	Medianamente Rápido	Medianamente Rápido

Tabla 3.1 Matriz disco

Matriz Tabla		
	T1	Altamente Accesada
	T2	Altamente Accesada
	T3	Poco acceso
	T4	Altamente Accesada
	T5	Poco acceso
	T6	Medianamente accesada
	T7	Medianamente accesada
	T8	Altamente Accesada
	T9	Poco acceso
	T10	Altamente Accesada

Tabla 3.2. Matriz tabla

De la combinación de estas dos matrices se obtiene:

MATRIZ PESOS TABLA VS DISCO					
	D1	D2	D3	D4	D5
T1	10	0	10	5	5
T2	10	0	10	5	5
T3	0	10	0	5	5
T4	10	0	10	5	5
T5	0	10	0	5	5
T6	5	0	5	10	10
T7	5	0	5	10	10
T8	10	0	10	5	5
T9	0	10	0	5	5
T10	10	0	10	5	5

Tabla 3.3. Matriz pesos tablas vs. disco

El algoritmo diseñado llena una matriz randómicamente con 0 y 1, para posteriormente convertirlas en las posibilidades ideales (10) en 1, a las regulares (5) en 1 con la opción de poder cambiarla a 0, o mantenerse como tal y finalmente a las posibilidades no deseadas las convierte en 0, de esta manera la matriz convertida quedaría como se muestra a continuación:

MATRIZ TABLA VS DISCO CONVERTIDA					
	D1	D2	D3	D4	D5
T1	1	0	1	1/0	1/0
T2	1	0	1	1/0	1/0
T3	0	1	0	1/0	1/0
T4	1	0	1	1/0	1/0
T5	0	1	0	1/0	1/0
T6	1/0	0	1/0	1	1
T7	1/0	0	1/0	1	1
T8	1	0	1	1/0	1/0
T9	0	1	0	1/0	1/0
T10	1	0	1	1/0	1/0

Tabla 3.4. Matriz tabla vs. disco convertida

Se clasificó inicialmente la distribución de tablas en los discos, sin embargo observamos que para las tablas: T1, T2, T4, T8 y T10 las posiciones ideales serían que se guarden en los discos: D1 o D3, recomendable en los discos D4 o D5 y no deseado en el disco D2, de esta manera si empezamos a colocar todas las tablas en los primeros discos saturaremos en poco tiempo los mismos mientras que los siguientes estarían poco utilizados siendo esto ineficiente.

El algoritmo genético trabaja con una función conocida como $fitness^{(20)}$, (la cual se la puede evidenciar en el [anexo G](#)), la misma que compara el cromosoma actual con un promedio y de ser mejor lo aprueba, recalculando el nuevo promedio o desecha al

⁽²⁰⁾ Función del algoritmo genético que compara al cromosoma actual con un promedio establecido anteriormente.

inicial, para de esta manera evaluar todas las posibilidades de los cromosomas y determinar cuales serían las mejores posiciones para las diferentes tablas; es así como escoge en segunda instancia a la matriz indicada en la tabla 3.5:

MATRIZ TABLA VS DISCO DEPURADA POR UNA CORRIDA DE ALGORITMO GENETICO					
	D1	D2	D3	D4	D5
T1	1	0	0	0	0
T2	0	0	1	0	0
T3	0	1	0	0	0
T4	1	0	0	0	0
T5	0	1	0	0	0
T6	0	0	0	1	0
T7	0	0	0	1	0
T8	1	0	0	0	0
T9	0	1	0	0	0
T10	0	0	1	0	0

Tabla 3.5. Matriz tabla vs. disco depurada

En la tabla anterior podemos observar claramente como el problema de distribución se ha resuelto puesto que las tablas: T1, T4, T8 han sido colocadas en el disco D1, y que a su vez este disco era el ideal para las mismas. De la misma manera para las tablas T2 y T10 se encuentran ubicadas en el disco D3 y este disco a su vez es uno de

los ideales para las mismas. La misma situación ocurre para el resto de tablas las cuales han sido colocadas en los lugares ideales para su almacenamiento.

Sin embargo el sistema ahora es poco eficaz puesto que no existe ninguna tabla que haya sido guardada en el disco D5, (es decir los recursos han sido mal utilizados), por lo cual el sistema no ha sido explotado en su mayor capacidad, razón por la cual se ha creado la siguiente matriz, la cual nos ha servido para fijarnos en este error:

MATRIZ SUMATORIA TABLA VS DISCO					
	D1	D2	D3	D4	D5
Sumatoria	3	3	2	2	0

Tabla 3.6. Matriz sumatoria tabla vs. disco

Entonces al algoritmo genético inicia un proceso de cruzamiento (cross over), el cual consiste en tomar los 2 predecesores anteriores y de esta manera realizar un proceso de cruzamiento de genes en los mismos, los cuales se verán reflejados en un nuevo cromosoma, de características diferentes respecto a los dos anteriores, para visualizar con un ejemplo disponemos de:

Tomado de Matriz Tabla vs. disco convertida					
T1	1	0	1	1/0	1/0
Tomado de Matriz vs. disco depurada por una corrida de algoritmo genético					
T1	1	0	0	0	0

Tabla 3.7. Matriz de tablas pares

De los cuales se ha obtenido la siguiente tabla de combinaciones en las cuales se aprecia los posibles cromosomas válidos, a los cuales se les aplica nuevamente a cada uno de ellos la función fitness del algoritmo genético, [ver anexo H](#).

Como se puede apreciar muchos de los cromosomas se repiten, esa es la razón para lo cual colocamos literales para agruparlos por las similitudes de sus genes, para adicionalmente calcular la probabilidad de cruzamiento.

MATRIZ POSIBLES GENES REPETIDOS	
TIPO DE CROMOSOMA	No. DE REPETICION
A	8
B	5
C	7
D	4
E	8
F	4
G	8
H	4

Tabla 3.9 Matriz posibles genes repetidos

En caso de existir un cruzamiento no aceptable, es decir, que no cumpla con la función fitness, el algoritmo esta en capacidad de devolver aleatoriamente un descendiente.

Una vez realizado el cruzamiento el algoritmo esta diseñado para realizar un proceso conocido como mutación el cual arbitrariamente altera uno o más genes de un

cromosoma seleccionado; para de esta manera cubrir con posibilidades que no han sido consideradas por el cruzamiento, como lo apreciamos en el siguiente cuadro:

Tomado de Matriz Tabla vs. disco convertida					
T1	1	0	1	1/0	1/0

Tabla 3.10 Matriz antes de mutación

MATRIZ POSIBILIDADES DE MUTACION						
T1	1	0	1	1	1	D
T1	1	0	1	1	0	C
T1	1	0	1	0	1	B
T1	1	0	1	0	0	A
T1	1	0	0	1	1	H
T1	1	0	0	1	0	G
T1	1	0	0	0	1	F
T1	1	0	0	0	0	E
T1	1	1	1	1	1	No considerada
T1	1	1	1	1	0	No considerada
T1	1	1	1	0	1	No considerada
T1	1	1	1	0	0	No considerada
T1	1	1	0	1	1	No considerada
T1	1	1	0	0	1	No considerada
T1	1	1	0	0	0	No considerada
T1	0	0	1	1	1	No considerada
T1	0	0	1	1	0	No considerada
T1	0	0	1	0	1	No considerada
T1	0	0	1	0	0	No considerada
T1	0	0	0	1	1	No considerada
T1	0	0	0	1	0	No considerada
T1	0	0	0	0	1	No considerada
T1	0	0	0	0	0	No considerada
T1	0	1	1	1	1	No considerada
T1	0	1	1	1	0	No considerada
T1	0	1	1	0	1	No considerada
T1	0	1	1	0	0	No considerada
T1	0	1	0	1	1	No considerada
T1	0	1	0	1	0	No considerada
T1	0	1	0	0	1	No considerada
T1	0	1	0	0	0	No considerada

Tabla 3.11 Matriz posibilidades de mutación

De esta manera se puede apreciar como existen posibilidades que no han sido consideradas hasta el momento para su aplicación, de la misma manera al igual que en el caso anterior se calcula la probabilidad de mutación y la creación de nuevos cromosomas a los cuales nuevamente se los vuelve a aplicar la función fitness, para determinar si estos son ideales, regulares o no deseados.

Al algoritmo genético trabaja con una aproximación igual a 1, y una vez obtenida esta la convierte en posibilidades de almacenamiento binario los cuales se puede apreciar como una Matriz de Tabla vs. Disco, indicando con el número 1 como al ideal y al número 0 como el no deseado. Adicionalmente nos indica el número de transacciones que realizó para llegar a este resultado, el cual después de varias pruebas realizadas lo estimamos en 650000 instancias (se debe considerar que mientras mayores sean las ocasiones de aplicación más preciso será el resultado). Como es obvio suponer no existe una respuesta única ideal y de la misma manera al realizar varias ejecuciones sobre un mismo problema los resultados serán diferentes, los mismos que han sido guardados en una base de datos cuyo esquema se pone de manifiesto a continuación:

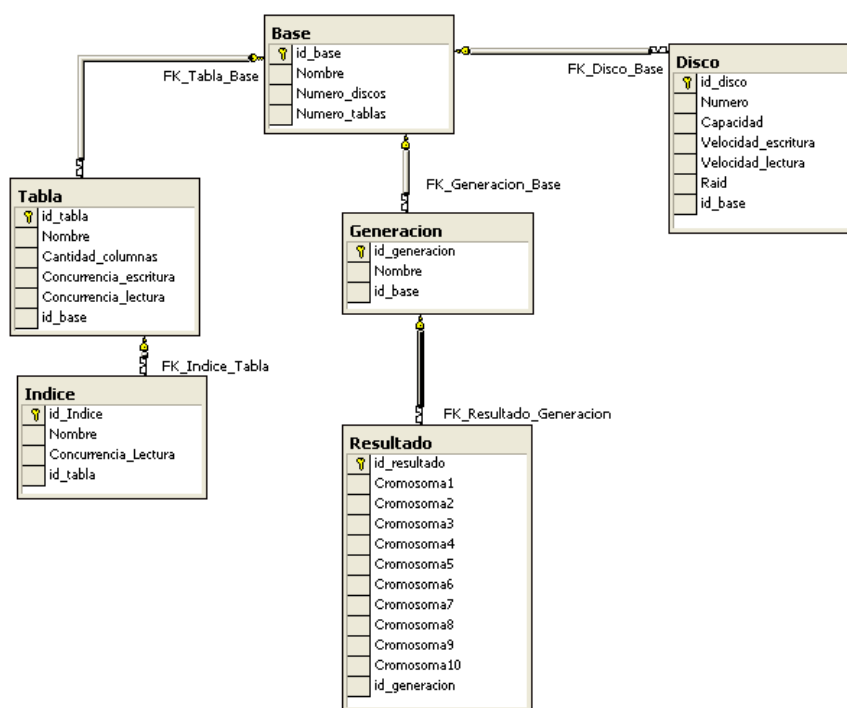


Figura 3.1 Diseño de la base de datos

3.2. Justificativo para el uso del algoritmo genético

Sus inicios datan a partir de los años sesenta en donde surgieron muy primitivamente y hasta la actualidad se han ido desarrollando gracias al crecimiento de la tecnología, evolución de la naturaleza de los programas, técnicas de programación entre otros.

Los algoritmos genéticos han sido desarrollados basados en el principio de la evolución es decir, el de mejor estado o condición es el que se impone sobre los demás y sobre la base de él se genera el resto de su especie. Es así como aprovechando técnicas de programación basados en estos mencionados algoritmos los

cuales son aplicados sobre todo en la resolución de problemas de optimización, optimización de funciones lineales y no lineales, problemas del viajero (la ruta mas corta), problemas de itinerarios, partición y control; han sido utilizados cada día mas en diferentes áreas cada día.

Basándonos en las necesidades evidenciadas por parte de los DBA, creemos que existe la necesidad de crear un programa capaz de decidir cual es el lugar más adecuado para los respaldos realizados en las bases de datos razón por la cual era necesario la construcción de un CASE capaz de brindar asesoría para la mencionada labor.

El hecho de ser CASE, ha permitido que sea de fácil instalación y uso, para lo cual se recomienda al lector familiarizarse con el mismo mediante la lectura del manual de usuario en el [anexo Z](#).

CAPITULO 4 BENCHMARK

4.1. Justificación

Los autores del presente hemos trabajado en la Dirección General del Personal de la Armada (DIGPER), en donde nos encontramos con los siguientes equipos y configuraciones:

Los servidores que posee DIGPER son los siguientes:

1 servidor IBM xSeries 235.

1 equipo para almacenamiento masivo (marca STORAGETECH).

1 servidor HP Proliant recientemente adquirido.

2 servidores INTEL SBT2.

1 servidor NETFINITY 3500.

La base de datos que utilizan todos es SQL 7.0, con las licencias de DINFOR (Dirección de Informática de la Armada). Se dispone de estas configuraciones:

4.2. Servidor IBM xSeries 235

Este servidor tiene 6 HD Hot Swap, 5 discos de 73.4 GB de capacidad y 10K rpm, el HD último tiene una capacidad de 36.4 GB.

Numero de discos	Capacidad	velocidad
1	73.4 Gb	10 KRPM
2	73.4 Gb	10 KRPM
3	73.4 Gb	10 KRPM
4	73.4 Gb	10 KRPM
5	73.4 Gb	10 KRPM
6	36.4 Gb	10 KRPM

Tabla 4.1 Servidor IBM x Series 235

Los 5 HD están configurados en RAID 5 con un disco como Hot Spare, por si uno de ellos falla este disco lo reemplaza sin tener problemas de funcionamiento. La capacidad total del arreglo es de 205 GB. El HD de 36.4 GB esta configurado con RAID 0, en el cual se encuentra instalado el sistema operativo. Estos discos se instalaron el mes de Octubre del 2004 y no se han tenido ningún tipo de problemas.

4.3. Equipo para almacenamiento masivo (SAN)⁽²¹⁾ STORAGETECH

Este equipo tiene instalados 6 HD de 73 GB Hot Swap para fibra óptica de 10K rpm. Los discos están configurados en RAID 5 con un disco como Hot Spare. La capacidad total de los 6 discos es de 273 GB.

Numero de discos	Capacidad	Velocidad	Conectividad
1	73 Gb	10 KRPM	Fibra óptica
2	73 Gb	10 KRPM	Fibra óptica
3	73 Gb	10 KRPM	Fibra óptica
4	73 Gb	10 KRPM	Fibra óptica
5	73 Gb	10 KRPM	Fibra óptica
6	73 Gb	10 KRPM	Fibra óptica

Tabla 4.2 Equipo SAN STORAGETECH

⁽²¹⁾ SAN (Storage Area Network) es una red o subred compartida de media o alta velocidad compuesta por dispositivos de almacenamiento.

Adicional, DIGPER adquirió 2 HD de iguales características de los 6 anteriores, únicamente con diferencia en el firmware. Uno de estos discos está instalado en el SAN debido a una falla que presentó uno de los originales.

Problemas. Este equipo ha presentado 4 fallas desde su instalación que fue realizada el mes de Mayo del 2003.

1. La primera se dio a fines del mes de Octubre del 2003 en la cual falló uno de los discos.
2. Un mes después otro disco falló (sin motivo aparente).
3. La tercera falla se produjo en el mes de Julio del 2004, por problemas de la controladora; esto fue solucionado el mes de Febrero del 2005, es decir el SAN no funcionó alrededor de seis meses.
4. La última falla, se produjo en uno de los discos, a los 2 días de entrar nuevamente en funcionamiento el equipo.

4.4 Servidor INTEL SBT2 (1)

El servidor que actualmente tiene la BDD y la aplicación de Personal, tiene 4 HD, 3 Discos tienen 16.94 GB de capacidad y un HD tiene 34.18 GB de capacidad. El servidor no tiene una tarjeta RAID para poder crear un arreglo de discos.

Numero de discos	Capacidad	Velocidad
1	16.94 Gb	10 KRPM
2	16.94 Gb	10 KRPM
3	16.94 Gb	10 KRPM
4	34.18 Gb	10 KRPM

Tabla 4.3 INTEL SBT2 (1)

4.5. Servidor INTEL SBT2 (2)

El servidor que actualmente tiene la BDD y la aplicación de Sueldos, tiene 5 HD, 2 Discos tienen 17.10 GB de capacidad y 3 HD tiene 16.94 GB de capacidad. El servidor no tiene una tarjeta RAID para poder crear un arreglo de discos.

Numero de discos	Capacidad	Velocidad
1	17.10 Gb	10 KRPM
2	17.10 Gb	10 KRPM
3	16.94 Gb	10 KRPM
4	16.94 Gb	10 KRPM
5	16.94 Gb	10 KRPM

Tabla 4.4 INTEL SBT2 (2)

4.6.Servidor IBM NETFINITY 3500

Este servidor tiene un disco de 12 GB SCSI, no es Hot Swap. Este servidor actualmente se lo utiliza para el servicio de Internet e ICQ. No se lo utilizó para las pruebas

4.7.Servidor HP Proliant DL 380

Este servidor dispone de 2 HD Hot Swap de 36.4 GB de capacidad y 15K rpm. Desgraciadamente no se obtuvo autorización para la manipulación de servidor para las pruebas.

4.8. Tipos de pruebas

Es importante mencionar que el CASE desarrollado considera que el DBA tiene un conocimiento previo de la base de datos, en vista de que este deberá conocer características específicas de la misma, es decir, cualquier persona o un DBA que se encuentre pobremente relacionado con la base de datos no podrá conseguir el provecho esperado del CASE. Para las pruebas, utilizamos las bases de datos que se detallan a continuación:

1. DIGPERGYE.- es la base que almacena datos de todo el personal activo y pasivo de la Armada del Ecuador, el equivalente a 10985 personas activas, y es el equivalente a 30 generaciones (promociones), aproximadamente desde el año 1969. El número de usuarios es de 68 internos y 43 desde puntos remotos. La mencionada base de datos dispone de 614 tablas, las cuales después de un análisis evidenciado en los anexos A, B y C muestra los siguientes resultados:

Tipo de tabla	Numero de tabla
Altamente concurridas de escritura	135
Medianamente concurridas de escritura	236
Bajamente concurridas de escritura	42
Altamente concurridas de lectura	127
Medianamente concurridas de lectura	29
Bajamente concurridas de lectura	45
Total de tablas	614
Numero de usuarios internos	68
Numero de usuarios remotos	43
Tamaño	2.636.330 Mb

Tabla 4.5 Tipos de pruebas DIGPERGYE

2. SUELDOS.- es la base de datos que maneja los pagos de salarios de todo el personal de la Armada del Ecuador, su uso es por 22 personas calificadas para ello, y el estudio de las tablas emitió el siguiente resultado:

Tipo de tabla	Numero de tabla
Altamente concurridas de escritura	31
Medianamente concurridas de escritura	17
Bajamente concurridas de escritura	8
Altamente concurridas de lectura	24
Medianamente concurridas de lectura	23
Bajamente concurridas de lectura	21
Total de tablas	124
numero de usuarios	22
Tamaño	4.265.139 Mb

Tabla 4.6 Características Base Sueldos

4.8.1. Configuración 1

Se utilizaron las siguientes configuraciones dando estos resultados:

Base de datos	DIPERGYE	SUELDOS
Servidor	5117Mb	4313 Mb
STORAGETECH	145 minutos	6 fallas
Problemas	Integridad de las tablas	No fue posible de realizar

Tabla 4.7 Configuración No. 1

Durante la realización de las pruebas se presentaron innumerables incidentes tales como: pérdida de datos, pérdida de integridad de las tablas, pérdida de relaciones de las tablas, en conclusión los parámetros ingresados en las tablas que debían ser ordenadas no sirvieron de ayuda, por el contrario el respaldo de los mismos tuvo una duración mayor a la esperada puesto que las mencionadas bases tomaban menos tiempo en sus servidores originales. Al buscar explicación mencionamos que se mantiene un criterio de falla posiblemente en esta configuración por el tipo de servidor y los problemas presentados últimamente.

4.8.2. Configuración 2

Base de datos	DIPERGYE	SUELDOS
Servidor	5117Mb	4313 Mb
Servidor IBM x SERIES 235	96 minutos	52 minutos
Problemas	Nada a mencionar	Nada a mencionar

Tabla 4.8 Configuración No. 2

Durante la realización de estas pruebas en este servidor se evidenció una mejora significativa respecto al anterior y cierta mejora respecto a los servidores originales de las bases de datos.

4.8.3. Configuración 3

Realizamos respaldos en los servidores originales pero con la nueva deposición de orden de las tablas recomendadas por el CASE implementado, obteniéndose los siguientes resultados en promedio. Se debe considerar que los servidores originales no disponen de tarjeta RAID.

Base de datos	DIPERGYE	SUELDOS
Servidor	5117Mb	4313 Mb
INTEL SBT2 (1)	112 minutos	No aplicable
INTEL SBT2 (2)	No aplicable	52 minutos

Tabla 4.9 Configuración No. 3

4.9. Comparación de resultados

Servidor	Base de datos	DIPERGYE	SUELDOS
		5117Mb	4313 Mb
Servidores de prueba			
STORAGETECH		145 minutos	6 intentos fallidos
Servidor IBM x SERIES 235		96 minutos	52 minutos
Servidores originales con nueva disposición de tablas			
Intel SBT2 (1)		112minutos	No aplicable
Intel SBT2 (2)		No aplicable	52minutos
Servidores originales disposición original de tablas			
Intel SBT2 (1)		122minutos	No aplicable
Intel SBT2 (2)		No aplicable	60 minutos

Tabla 4.10 Comparación de resultados

Todos los valores mencionados aquí fueron tomados después de varias pruebas y de esta manera obtuvimos un promedio, tanto en los servidores de prueba como en los originales.

Servidor	Base de datos	DIPERGYE	SUELDOS	
		5117Mb	4313 Mb	
Servidores de prueba				
STORAGETECH		145	6 intentos fallidos	Configuración No 1
Servidor IBM x SERIES 235		96	52	Configuración No 2
Servidores originales con nueva disposición de tablas				
Intel SBT2 (1)		112	No aplicable	Configuración No 3
Intel SBT2 (2)		No aplicable	55	
Servidores originales disposición original de tablas				
Intel SBT2 (1)		122	No aplicable	
Intel SBT2 (2)		No aplicable	60	
desventajas en tiempo de respuesta STORAGETECH		118.85%	No aplicable	
beneficios en tiempo de respuesta IBM x SERIES 235		78.69%	86.67%	
beneficios en tiempo de respuesta Intel SBT2 (1).		91.80%	No aplicable	
beneficios en tiempo de respuesta Intel SBT2 (2)		No aplicable	91.66%	

Tabla 4.11 Resumen de resultados

Analizando el cuadro anterior concluimos que:

1. Se obtienen muy malos resultados con el servidor STORAGETECH (Configuración No 1), dando una disminución de rendimiento equivalente al 18.85%, siendo la posible causa los problemas mencionados en la descripción del equipo, la degradación del mismo o la incompatibilidad de hardware y software.
2. En el mencionado servidor se produjo excesiva cantidad de errores por lo cual creemos que en versiones posteriores de este CASE se debería considerar la posibilidad de tomar en cuenta el uso y degradación de los discos, para de esta manera tratar de colocar un parámetro más de medición y lograr corregir situaciones como esta.
3. Se tiene buenos resultados con el servidor IBM x SERIES 235 (Configuración No.2), dando una mejora en rendimiento medido en tiempo a un equivalente de 21.31% (para la base de datos DIGPERGYE) y del 13.33% (para la base de datos de SUELDOS). Sin embargo se debe considerar que los servidores disponen de RAID, lo cual no permite verificar la eficiencia auténtica del CASE, en vista de que los servidores originales no disponen de RAID.
4. Con el mencionado servidor no se produjo ningún tipo de conflicto.
5. En los servidores originales, con la nueva disposición de las tablas (Configuración No.3), se obtuvieron mejoras del 8.196% (para la base de datos DIGPERGYE) y del 8.33% (para la base de datos de SUELDOS)

6. Para la utilización del CASE se considera la velocidad standard de los discos, la cual viene como característica propia dada por el fabricante, en ningún momento se considera la velocidad optimizada del RAID.
7. En el CASE elaborado se ha considerado únicamente las tablas en las cuales se guardan datos, motivo por la cual no se ha manipulado ninguna tabla perteneciente al sistema. Esta es la razón por la que los índices se han guardado de acuerdo a la disposición recomendada por el SGBD.

CAPITULO 5 IMPACTO FINANCIERO

5.1. Beneficios cualitativos

Los beneficios cualitativos que se pueden mencionar entre otros son:

1. La utilización de un CASE amigable y sencillo que permita mejorar las tareas del DBA (Ver anexo I, Manual de Usuario).
2. La utilización correcta de los dispositivos de almacenamiento.
3. El mejoramiento de los históricos de las bases de datos.
4. Aprovechamiento de mejor manera de las características tanto de las bases de datos como de los dispositivos utilizados en las operaciones de las mismas.
5. Creación de un control de rendimiento de equipos y su utilización de base de datos.
6. Detección de posibles fallas en dispositivos utilizados por medio de comparación de parámetros guardados en los históricos.
7. Detección de incompatibilidad entre software y hardware.

8. Detección de degradación de equipos por medio de comparaciones de los históricos de grabación.

5.2 Beneficios cuantitativos

Los beneficios cuantitativos más relevantes son:

1. Existe una mejora de los tiempos empleados para las operaciones de mantenimiento de las bases de datos.
2. El algoritmo permite mejorar los procedimientos de respaldo de las bases de datos, logrando de esta manera optimizar este proceso.
3. Al momento disponemos de varias opciones de almacenamiento óptimas, las cuales pueden ser almacenadas en una base de datos, lo que nos permite llevar un histórico de las actividades realizadas en la base de datos.
4. Las disposiciones de almacenamiento recomendadas han sido obtenidas por medio de la aplicación de métodos y técnicas mediante la aplicación de conocimientos establecidos que no son empíricos.
5. Al disminuir el tiempo en una de sus actividades, el DBA, podrá dedicar mayor cantidad de tiempo a otras labores igualmente importantes; como el mantenimiento de la base de datos, la seguridad de la red, la implementación de otros case de optimización, etc.

5.3. Recursos y costos

Recursos	Costos	Justificación
Servidores para pruebas		Se ha considerado los valores referenciales que costaron los servidores en los que se realizaron las pruebas. En realidad fueron usados únicamente para la elaboración de las pruebas.
IBM x SERIES 235	\$ 45000,00	
STORAGETECH	\$ 36000,00	
Intel SBT2 (1)	\$ 35000,00	
Intel SBT2 (2)	\$ 42000,00	
Bases de datos		
DIGPERGYE	No aplicable	Puesto que se encuentra el histórico de toda la Armada del Ecuador con datos desde 1969.
Documentos	No aplicable	Tiene valor solo para la Armada del Ecuador
Sueldos	No aplicable	Tiene valor solo para la Armada del Ecuador

Tabla 5.1 Recursos y costos

5.3.1 Beneficios cualitativos

Recursos	Costos	Justificación
Java	No aplicable	La herramienta que los autores han escogido debido a que es un lenguaje universal y libre, que se encuentra al alcance de todos de fácil aprendizaje y orientado a objetos.
PC (para la creación del case)	\$ 400,00	Hemos considerado que una Pc Pentium III de 800 MHz con 128 Mb en RAM, tarjeta de video de 8 Mb, espacio libre en el disco de 100 Mb, con teclado, ratón, monitor de 15'', CdRom y tarjeta fax MODEM o de red

Impresora	\$ 80,00	Lexmark
Banda ancha 128 Kbps	\$ 80,00 mensuales	De esta manera evitamos el gasto excesivo en teléfono.
Regulador de voltaje	\$ 20,00	
Cables cruzados		Construcción casera
Software SQL 7.0	\$ 650,00	La licencia para cada cliente.

Tabla 5.2 Recursos utilizados en implementación

5.3.2 Recursos y costos

PC (para la utilización del case)	<p>Pc Pentium IV de 2800 MHz</p> <p>512 Mb en RAM</p> <p>Tarjeta de video de 8 Mb, Espacio libre en el disco de 100 Mb</p> <p>Teclado</p> <p>Ratón</p> <p>Monitor de 15''</p> <p>Cd Rom</p>
-----------------------------------	---

Tabla 5.3 Recursos mínimos necesarios

5.3.3 Recursos adicionales requeridos para administrar el sistema

Administración del case	Base de datos SQL 7.0 o Access
	Java

Tabla 5.4 Recursos adicionales

5.3.4 Costo del software

Recursos	Costos
Desarrollo 1500 por autor	\$ 3000
Java	No aplicable
PC (para la creación del case), una por autor	\$ 800,00 ver anexo J , ver anexo K
Impresora anexo J , anexo K	\$ 80,00
Banda ancha 128 KPSS (\$80 mensuales por 12 meses)	\$ 960,00
Regulador de voltaje	\$ 20,00
Cables cruzados	Construcción casera
Software SQL 7.0, anexo L , anexo M , anexo N , anexo O	\$ 650,00
Total	\$ 6160

Tabla 5.5 Costo del CASE

5.4.ROI

Lastimosamente los autores del presente no tienen valores económicos para compararlos con los beneficios obtenidos sobre la inversión puesto que las pruebas realizadas fueron hechas en entidades estatales que no producen dinero sino por el contrario el ente DIGPER, es un organismo netamente consumidor de recursos del estado. Los beneficios obtenidos han sido parametrizados en cuestión de tiempo y ventajas respecto a los usuarios del mismo, DBA.

CONCLUSIONES Y RECOMENDACIONES

1. Se optimizó la capacidad de almacenamiento en los discos utilizados, tanto en los sistemas originales como en los que disponen de RAID.
2. Se mejoró la velocidad en los procesos de respaldo realizado en las diferentes transacciones ordinarias en las bases de datos.
3. Actualmente se dispone de varias alternativas óptimas para la elección del sitio de almacenamiento de las diferentes tablas de las bases de datos, y no simplemente estar a la disposición de lo que nos indica la base de datos.
4. Se puede crear un modelo de comportamiento, Histórico del CASE, basados en las acciones realizadas sobre la misma; una vez que se obtenga la suficiente cantidad de datos.

RECOMENDACIONES

1. Para versiones futuras se podrá considerar el uso y la degradación de los discos duros y dispositivos, para de esta manera obtener un parámetro más a ser medido y mejorar los valores obtenidos para equipos con problemas.
2. Creación de históricos de bases de datos ya utilizadas.
3. Creación de modelos aplicables a situaciones parecidas de diferentes bases de datos.
4. Creación de índices con la ubicación y manipulación física de estos

BIBLIOGRAFIA

1. ZBIGNIEW, Michalewicz, "GENETIC ALGORITHMS + DATA STRUCTURES = EVOLUTION PROGRAMS", third extended revised edition, Springer Verlag heidenberg New York, 1999.
2. KENDAL & KENDAL, "ANALISIS Y DISEÑO DE SISTEMAS", tercera edición, México, 1997.
3. GILL & RAO, DATA WARE HOUSING, edición en español, Prentice Hall, Hispanoamericana, Naucaplan de Juárez, Edo. México, México, 1996.
4. FLANAGAN, David, "JAVA EN POCAS PALABRAS", Mc Graw Hill, segunda Edición, 1998.
5. NAUGHTON y SCHILDT "JAVA MANUAL DE REFERENCIAS" Mc Graw Hill, 1999.
6. LATHROP, WEBSTER, TEMPLE, "ARIADNE: PATTERN DIRECTED INFERENCE AND HIERARCHICAL ABSTRACTION IN STRUCTURE RECOGNITION", paper, Journal ACM, Massachusetts, USA, 1997.
7. GRIMSON, Eric, "THE COMBINATORICS OF LOCAL CONSTRAINTS IN MODEL BASED", paper, Journal ACM, California, USA, 2003.

8. LIBERATORE, Paolo, “COMPABILITY AND COMPACT REPRESENTATION OF REVISION OF HORN KNOWLEDGE BASES”, paper, Journal New York, USA, 2000.
9. AGUILAR, Luis, “FUNDAMENTOS DE PROGRAMACION: ALGORITMOS Y ESTRUCTURAS DE DATOS”, segunda edición, Mc Graw Hill, Madrid, España, 1996.
10. <http://ccp.servidores.net/genetico.html> COSTA, Carlos, “Algoritmos Genéticos”, 2003
11. <http://geocities.com/CapeCanaveral/9802/3d5ca000.htm> Larraña, “Algoritmos Genéticos”, Departamento de Ciencias de Computación e inteligencia artificial, Universidad del país vasco, 2003.
12. <http://www.ianet.com/users/jcontre/genetic> CONTRERAS, J, “Tutorial sobre algoritmos genéticos ”, Maracaibo, 2003.
13. <http://redcientifica.com/doc/doc199904260011.html> COELLO, Carlos, “Introducción a los algoritmos genéticos” , 2003.
14. <http://geocities.com/ohcop/busqueda.html>
15. <http://geocities.com/ohcop/index.html>
16. <http://www.programacion.com/tutorial/moddatos/4/>

17. <http://tramullas.com/documatica/indice.html>
18. <http://www.programacion.com/tutorial/moddatos/4/>
19. <http://www.tau.org.ar/base/lara.pue.udlap.mx/sistoper/index.html>
20. <http://www.cindoc.csic.es/isis/01-3-1.htm>
21. <http://ict.udlap.mx/people.html>
22. <http://www.inei.gob.pe/>
23. http://exa.unne.edu.ar/depar/areas/informatica/anasistem2/public_html/apuntes/maf/anexos/estructura.htm
24. <http://perseo.dif.um.es/~fernan/clasifica.ppt>
25. http://docentes.usaca.edu.co/wildiaz/BDII_02.html
26. <http://www.algoritmia.net/articles.php?id=12>

27. http://www.lafacu.com/apuntes/informatica/algorit_ordena/default.htm

28. http://decsai.ugr.es/docencia/ii/ii_mod_av_bd.html

29. http://www.itlp.edu.mx/publica/tutoriales/admonbasedat/tema2_1.htm