

**MODELO DE TECNOLOGIA DE INFORMACION Y COMUNICACIONES
EN SISTEMAS DE BASES DE DATOS APLICADO A LAS
PEQUEÑAS Y MEDIANAS EMPRESAS
DEL DISTRITO BARRANQUILLA**

**JORGE GUILLERMO FEDERICO CASTRO CERVANTES
LENIS MARIA HERRERA RODRÍGUEZ
BERTHA ABIGAIL REYES SALAS
OLGA PATRICIA RODRÍGUEZ RUZ**

Asesores

**Ingeniero JOHAN MANCERA MORENO
Ingeniero WILSON NIETO**

**CORPORACIÓN EDUCATIVA MAYOR DEL DESARROLLO SIMÓN BOLIVAR
PROGRAMA DE INGENIERÍA DE SISTEMAS
ÁREA DE INVESTIGACIÓN FORMATIVA IV
BARRANQUILLA**

2004

TABLA DE CONTENIDO

INTRODUCCIÓN	16
1. PLANTEAMIENTO DEL PROBLEMA.....	19
1.1 DESCRIPCIÓN DEL PROBLEMA	19
1.2 FORMULACIÓN DEL PROBLEMA	21
2. OBJETIVOS	22
2.1 OBJETIVO GENERAL.....	22
2.2 OBJETIVOS ESPECIFICOS	22
3. JUSTIFICACIÓN	24
4. ALCANCES Y LIMITACIONES.....	26
5. SISTEMA DE BASE DE DATOS	27
5.1 MEDICIÓN DEL RENDIMIENTO DE LAS BASES DE DATOS	29
6. BASES DE DATOS DISTRIBUIDAS	34
6.1 DEFINICIÓN.....	34
6.2 TIPOS DE BASES DE DATOS DISTRIBUIDAS	37
6.3 DIVISIÓN PARA ALMACENAMIENTO.....	39
6.4 VENTAJAS DE LAS BASES DE DATOS DISTRIBUIDAS.....	40
6.5 DESVENTAJAS DE LAS BASES DE DATOS DISTRIBUIDAS	41
6.6 OBJETIVOS DE LAS BASES DE DATOS DISTRIBUIDAS	47
6.7 MODELO DE TRANSACCIONES DISTRIBUIDAS.....	53
6.8 SISTEMAS CLIENTE - SERVIDOR.....	56
7. BASES DE DATOS EN LA WEB	58
7.1 INTEGRACIÓN DE BASES DE DATOS EN LA WEB.....	58
7.2 ¿CÓMO FUNCIONA LA INTEGRACIÓN DE BASES DE DATOS EN LA WEB?.....	59
7.3 CATEGORIZACIÓN DE INTERFACES WEB/DBMS.....	60

7.4 TECNOLOGÍAS PARA LA INTEGRACIÓN DE BASES DE DATOS EN LA WEB.....	61
7.5 CUADRO DE RESUMEN INTERFACES WEB/DBMS.....	67
8. DATAWAREHOUSE	71
8.1 ¿CÓMO IMPLEMENTAR UN DATAWAREHOUSE CON ÉXITO?	72
8.2 ¿CÓMO TRABAJA UNA BODEGA DE DATOS?	73
8.3 ¿QUÉ HAY QUE HACER PARA CONFORMAR LA BODEGA DE DATOS?.....	74
8.4 ¿CUÁL ES EL RETORNO DE LA INVERSIÓN?	75
8.5 ARQUITECTURA DATAWAREHOUSE.....	75
9. CRM- CUSTOMER RELATIONSHIP MANAGEMENT	78
9.1 IMPLANTACIÓN INTERNA	79
9.2 ASP DE PRODUCTOS CRM	79
9.3 OUTSOURCING DEL CONTACT CENTER.....	80
9.4 HOSTING DE SOLUCIONES CRM.....	80
9.5 CARACTERÍSTICAS.....	80
9.6 TIPOS DE CRM	81
9.6.1 CRM Analítico	81
9.6.2 CRM Operacional.....	81
9.6.3 CRM Colaborativo.....	81
9.7 FUNCIONALIDAD CRM	82
9.8 ANÁLISIS DE LA INTEGRACIÓN	83
9.9 EVOLUCIÓN CRONOLÓGICA DE UN SISTEMA INTEGRADO CON CRM Y DATAWAREHOUSE.....	84
9.9.1 Fase I. Core Del Negocio	84
9.9.2 Fase II. Gestión De Clientes.....	84
9.9.3 Fase III. Data Warehouse.....	84
9.9.4 Fase IV.....	84
10. PRM (PARTNER RELATIONSHIP MANAGEMENT).....	85
11. BASES DE DATOS CENTRALIZADAS	88

11.1 GENERALIDADES	88
11.2 CARACTERÍSTICAS.....	89
11.3 VENTAJAS.....	89
11.4 DESVENTAJAS	90
12. S.D.K SOFTWARE DEVELOPER KIT	91
13. SOFTWARE DE FUENTES PÚBLICAS (OPEN SOURCE SOFTWARE).....	92
13.1 CARACTERÍSTICAS.....	92
13.2 TIPOS DE LICENCIAS DE FUENTE PÚBLICA.....	93
13.3 DESARROLLO DE SOFTWARE DE FUENTES PÚBLICAS	94
13.4 EMPRESAS ALREDEDOR DE LINUX	96
13.5 SEGURIDAD	97
14. MYSQL.....	99
14.1 BREVE HISTORIA DE MYSQL	99
14.2 ¿QUÉ ES MYSQL?	99
14.3 PRINCIPALES CARACTERÍSTICAS.....	102
14.3.1 Características (version 4.0 en adelante)	104
14.3.2 Mejoras desde la version 3.23.....	105
14.3.3 InnoDB: Transacciones ACID y más	106
14.4 ¿QUÉ ES LO QUE LE FALTA?	107
14.5 TAREAS DE UN ADMINISTRADOR DE BASE DE DATOS.....	108
14.6 MYSQL-FRONT	110
14.6.1 Instalación y configuración del Abria Merlin	110
14.6.2 Instalación y configuración del MySQL-Front.....	123
14.6.3 Conectarse a la Base de Datos	130
14.6.4 Creación de Base De Datos y/o Tablas	131
14.6.5 Creación de Tablas	132
14.6.6 Visualizar e Ingresar más Campos y Datos	134
14.7 PHPMYADMIN: ADMINISTRACION MYSQL A TRAVÉS DE LA WEB	135
14.7.1 Página de PhpMyAdmin	136
14.7.2 Instalación de PhpMyAdmin	137

14.7.3 Características	137
14.7.4 Requisitos	138
14.7.5 Configuración	138
14.7.6 Utilizando PhpMyAdmin.....	139
14.7.6.1 Creación de bases de datos.....	140
14.7.6.2 Creación de tablas	141
14.7.6.3 Configuración de los campos	141
14.7.6.5 Visualización de datos	143
14.8 MYSQLADMIN	147
14.9 MYSQL CONTROL CENTER.....	150
14.9.1 Administrador de Usuarios	151
14.9.2 Administración del Servidor	152
14.9.3 Ventana de Base de Datos.....	153
14.9.4 Copias de Seguridad en MySQL	156
15.9.4.1 Backup de los datos (backup table)	157
15.9.4.2 Recuperación de datos	158
15.9.4.3 Backup MySQL con mysqldump	159
15. ORACLE	166
15.1 HISTORIA DE ORACLE	166
15.2 ¿QUÉ ES ORACLE?	168
15.3 VERSIONES DE ORACLE	169
15.4 EL KERNEL DE ORACLE	171
15.5 ALGUNAS DE LAS HERRAMIENTAS DE ORACLE	172
15.6 VENTAJAS.....	172
15.7 COSTOS	173
15.8 INSTALACIÓN	173
15.8.1 Proceso de Pre-Instalación.....	173
15.8.2 Productos y Tamaño	173
15.8.3 Requerimientos	175
15.8.4 Instalación de Oracle	175

15.8.5 Desinstalación de Oracle.....	176
15.8.6 Arranque de una BD Oracle	177
15.8.7 Seguridad.....	179
15.8.7.1 Implementación de Seguridad.....	180
15.8.7.2 Creación de Usuarios.....	180
15.8.7.3 Eliminación de Usuarios.....	181
15.8.7.4 Privilegios del Sistema	182
15.8.7.5 Perfiles de Usuario.....	185
15.8.7.6 Gestionando Privilegios	186
15.8.7.7 Listar Privilegios Otorgados	187
15.9 ENFOQUE RELACIONAL	187
16. ADMINISTRACIÓN SGBD Y CONFIGURACIÓN	190
16.1 COMANDOS INTERACTIVOS ORACLE Y MYSQL	190
16.2 COMANDOS DDL (LENGUAJE DE DEFINICIÓN DE DATOS).....	190
16.2.1 Aplicación de Comandos DDL: (Tabla Estudiantes)	192
16.3 COMANDOS DML (LENGUAJE DE MANIPULACIÓN DE DATOS)	193
16.3.1 Aplicación de Comandos DML: (Tabla Estudiantes).....	196
16.4 DBA STUDIO – ADMINISTRADOR DE ORACLE.....	199
16.4.1 Conexión Directa	203
16.4.2 Conexión con OMS	205
16.4.3 Administrador de Instancias (Instance).....	206
16.4.4 Administrador de Esquemas (Schema)	214
16.4.5 Administrador de Seguridad (Security)	219
16.4.6 Administrador de Almacén (Storage)	226
17. MICROSOFT ACCESS	234
17.1 CARACTERÍSTICAS PRINCIPALES.....	234
17.2 INSTALACIÓN	235
17.2.1 Proceso de Instalación	235
17.3 CREACIÓN DE BASES DE DATOS.....	236
17.3.1 Creación de Tablas	238

17.3.2 Insertar Datos	239
17.4 FILTROS	242
17.4.1 Filtro por Selección	243
17.4.2 Filtro por Formulario	244
17.4.3 Filtro u Orden Avanzado	244
18. POSTGRESQL 7.1.....	246
18.1 HISTORIA DE POSTGRESQL	246
18.2 ¿QUÉ ES POSTGRESQL?	247
18.3 CARACTERÍSTICAS DE POSTGRESQL.....	250
18.4 ¿QUÉ ES LO QUE LE FALTA?	252
19. INTERBASE 6.0	253
19.1 ADMINISTRACIÓN	254
19.2 INSTALACIÓN	255
19.2.1 Instalación en Windows NT y Windows 95/98:	255
19.2.2 Instalación en UNIX:	260
19.2.2.1 Instalación en algunas plataformas UNIX.....	261
19.2.2.2 Desinstalación en UNIX	262
20. LAS COMPARATIVAS	263
20.1 COMPARATIVA DE CARACTERÍSTICAS TÉCNICAS	266
20.2 COMPARATIVA RENDIMIENTOS	268
20.2.1 Ataque Desde Línea De Comandos	269
20.2.2 Ataque Desde Php	271
21. SQL SERVER	277
21.1 OBJETIVOS DEL DISEÑO DE SQL SERVER	277
21.2 ARQUITECTURA RDBMS	281
21.3 DISPONIBILIDAD, FIABILIDAD Y TOLERANCIA A FALLOS.....	282
22. ESTADO DEL ARTE	283
23. ANTECEDENTES	286
24. MARCO CONCEPTUAL.....	290
25. METODOLOGIA.....	294

25.1 TIPO DE ESTUDIO	294
25.2 LINEA DE INVESTIGACION	294
26. RECURSOS	295
BIBLIOGRAFIA	297
ANEXOS	299

LISTA DE TABLAS

Tabla 1. Interfaces WEB/DBMS	67
Tabla 2. Resultado de ejecución del comando	149
Tabla 3. Resultado comando Backup table	158
Tabla 4. Resultado comando Restore table	159
Tabla 5. Arranque de una Base de datos Oracle	177
Tabla 6. Parámetros para la creación de usuarios	180
Tabla 7. Privilegios del sistema	182
Tabla 8. Privilegios de los roles CONNECT, RESOURCE y DBA	184
Tabla 9. Recursos limitados via perfil	185
Tabla 10. Privilegios otorgados sobre objetos	186
Tabla 11. Vistas del diccionario de datos	187
Tabla 12. Requisitos para instalación de Microsoft Access	235
Tabla 13. Características Principales	263
Tabla 14. Características Técnicas	266
Tabla 15. Otras características	267
Tabla 16. Script creación de tablas	269
Tabla 17. Resultados entrada fichero de texto	270
Tabla 18. Segundos en completar las 1000 peticiones	275
Tabla 19. Peticiones servidas por segundo	275

LISTA DE FIGURAS

Figura 1. Configuración de una base de datos distribuida	36
Figura 2. Base de datos distribuida particionada	37
Figura 3. Aumento de paralelismo.....	38
Figura 4. Base de datos combinando réplicas y particiones.	39
Figura 5. Una pasarela hipotética proporcionada por Ingres para Oracle.	53
Figura 6. Arquitectura del sistema global.....	55
Figura 7. Un sistema Cliente – Servidor	57
Figura 8. Arquitectura Datawarehouse.	77
Figura 9. Proceso de Instalación Abria Merlin (Paso 1)	110
Figura 10. Proceso de Instalación Abria Merlin (Paso 2)	111
Figura 11. Proceso de Instalación Abria Merlin (Paso 3)	112
Figura 12. Proceso de Instalación Abria Merlin (Paso 4)	113
Figura 13. Proceso de Instalación Abria Merlin (Paso 5)	114
Figura 14. Proceso de Instalación Abria Merlin (Paso 6)	115
Figura 15. Proceso de Instalación Abria Merlin (Paso 7)	116
Figura 16. Proceso de Instalación Abria Merlin (Paso 8)	117
Figura 17. Proceso de Instalación Abria Merlin (Paso 9)	118
Figura 18. Proceso de Instalación Abria Merlin (Paso 10)	119
Figura 19. Proceso de Instalación Abria Merlin (Paso 11)	120
Figura 20. Proceso de Instalación Abria Merlin (Paso 12)	121
Figura 21. Proceso de Instalación Abria Merlin (Paso 13)	122
Figura 22. Proceso de Instalación Abria Merlin (Paso 14)	123
Figura 23. Proceso de Instalación de MySQL Front (Paso 1)	124
Figura 24. Proceso de Instalación de MySQL Front (Paso 2)	125
Figura 25. Proceso de Instalación de MySQL Front (Paso 3)	126

Figura 26. Proceso de Instalación de MySQL Front (Paso 4)	127
Figura 27. Proceso de Instalación de MySQL Front (Paso 5)	128
Figura 28. Proceso de Instalación de MySQL Front (Paso 6)	129
Figura 29. Conectarse a la Base de Datos	131
Figura 30. Creación de Base de Datos	132
Figura 31. Creación de tablas y entidades	133
Figura 32. Creación de tablas por Query	134
Figura 33. Ingresar más campos	135
Figura 34. Página inicial de PhpMyAdmin	140
Figura 35. Crear base de datos	140
Figura 36. Crear tabla	141
Figura 37. Configurar campos	141
Figura 38. Grabar campos	142
Figura 39. Insertar datos	143
Figura 40. Ver datos	143
Figura 41. Welcome to PhpMyAdmin 2.5.4 (1)	144
Figura 42. Welcome to PhpMyAdmin 2.5.4 (2)	145
Figura 43. Exportar información	146
Figura 44. Ver los script de la base de datos	147
Figura 45. MySQL Control Center – Console Manager	150
Figura 46. Administración de usuarios	151
Figura 47. Administración del servidor	152
Figura 48. Ventana de Base de datos	154
Figura 49. Field Properties	154
Figura 50. Indexes	155
Figura 51. Table Properties	155
Figura 52. Plataforma Oracle 9i	171
Figura 53. Pantalla de instalación del cliente de Oracle	200
Figura 54. Primer pantallazo con Login to the Oracle Management Server seleccionado	201

Figura 55. Primer pantallazo con Launch DBA Studio standalone seleccionada	202
Figura 56. Oracle DBA Studio	202
Figura 57. Conexión Directa	203
Figura 58. Add Database To Tree	204
Figura 59. Edit Local Preferred Credentials	205
Figura 60. Conexión con OMS	206
Figura 61. Instance	207
Figura 62. Árbol Instance	208
Figura 63. Database	209
Figura 64. All Parameters	210
Figura 65. Stored Configurations	211
Figura 66. Sessions	212
Figura 67. Resource Consumer Groups	213
Figura 68. Resource Plans	213
Figura 69. Schema	214
Figura 70. Folders Schema	215
Figura 71. Menu Object	216
Figura 72. Creando una tabla (Paso 1).....	217
Figura 73. Creando una tabla (Paso 2).....	218
Figura 74. Editando un objeto.....	219
Figura 75. Security	220
Figura 76. Folders de Security.....	221
Figura 77. Iconos de los objetos	221
Figura 78. Crear usuario.....	222
Figura 79. Crear rol	223
Figura 80. Crear perfil.....	224
Figura 81. Show Grantees.....	225
Figura 82. Show Grantees.....	226
Figura 83. Storage.....	227
Figura 84. Listado del Storage.....	228

Figura 85. Tablespaces	229
Figura 86. Operaciones con Tablespaces	230
Figura 87. Operaciones con Datafiles.....	231
Figura 88. Operaciones con Segmentos de Rollback	232
Figura 89. Operaciones con Redo Log Group	233
Figura 90. Pantalla Inicial de Access	236
Figura 91. Creación de Base de datos	237
Figura 92. Creación de tablas en Vista Diseño	238
Figura 93. Diseño de la tabla.....	239
Figura 94. Click en el icono Diseño	240
Figura 95. Visualizar los campos de la tabla.....	240
Figura 96. Click en el icono Abrir.....	241
Figura 97. Ingresar datos en la tabla	241
Figura 98. Guardar la tabla	242
Figura 99. Menu Registro/Filtro	243

INTRODUCCIÓN

Las empresas están en un continuo desarrollo en pro de la seguridad de los datos generados por las diversas transacciones hechas en el área comercial y administrativa. Para proporcionar protección y utilización óptima de la información se emplean los Sistemas de Bases de Datos.

Los Sistemas de Bases de Datos son herramientas computarizadas para llevar registros. Es un depósito o contenedor de una colección de archivos de datos sistematizados. La manipulación de éstos permite la realización de funciones como adición, inserción, recuperación, modificación y eliminación de datos y archivos.

Desde hace diez años se ha incrementado el interés sobre los sistemas de bases de datos relacionales, especialmente los distribuidos, los centralizados, los data warehouses y en web. El modelo relacional es el fundamento de la tecnología moderna de bases de datos. Éstas se presentaron con el objetivo de almacenar los datos persistentes de una compañía y han ido perfeccionándose según las necesidades requeridas por los usuarios: como poder acceder desde un lugar distante a la misma base de datos, manejar objetos y operaciones sobre éstos y tomar decisiones sin afectar a los sistemas operacionales.

Los ingenieros se enfocan, frecuentemente, en áreas de aplicación como CAD, CASE, Geología, Ciencia y Medicina, Almacenamiento y recuperación de documentos, entre otros. En los últimos años varios fabricantes han elaborado productos DBMS o Servidores universales como Universal Database DB2,

Universal Data Option para Informix Dynamic Server, Oracle 8i Universal Server y Oracle 9i Universal Server, Database Server o Enterprise Server. Existen diversos lenguajes con los que se pueden manipular las bases de datos relacionales pero el más común entre todos es el SQL ya que la mayoría de los productos de bases de datos del mercado lo manejan.

Son muchos los benchmarks que se han publicado sobre estos gestores de bases de datos, aunque muchos de ellos tienen una clara tendencia hacia uno de los dos bandos. Es por esto que hay que saber extraer bien las conclusiones a partir de un benchmark. Por ejemplo, es importante que las comparaciones entre los gestores se realicen en igualdad de condiciones, cosa que por otra parte, muchas veces resulta complicado debido a las distintas implementaciones que estos poseen.

Además, resulta muy difícil diseñar un buen benchmark, que tenga en cuenta todas las posibles mejoras de rendimiento que pueda aplicar cada uno de estos gestores. Es lógico pues, que haya algunas pruebas que se le apliquen a un gestor, y que no tenga ningún sentido realizárselas al otro.

Todo apunta que las tecnologías *Open Source* cambiarán radicalmente la industria del *software* en 3 o 4 años. Las bases de datos son parte de esta transformación. Poco a poco, pero sin descanso, el *software Open Source* está adquiriendo una robustez y una potencia suficiente como para plantar cara al *software* comercial. La liberalización de *Interbase* es sólo el principio.

Según investigaciones consultadas, las bases de datos *Open Source* son utilizadas por los desarrolladores de *Web* y *software*, y pequeñas y medianas empresas. La mayoría también lo integran con otras aplicaciones *Open Source*. Es decir, normalmente, se tiende a radicalizar: o todo *Open Source*, o todo propietario.

En este trabajo compararemos algunas tipos Open Source (MySQL, PostgreSQL e *Interbase*) y otras propietarias (*Oracle* y *Access*):

Veremos que todas las bases de datos analizadas tienen sus fuertes y sus debilidades. Depende del proyecto que queramos realizar, escogeremos una u otra. Todo el mundo tiene una base de datos favorita, así como una distribución de Linux favorita, pero cada una tiene un propósito, y bajo unas condiciones cada una será mejor que las demás.

La finalidad de este proyecto investigativo es generar un documento que suministre información sobre la instalación, requerimientos y administración de bases de datos open source y propietarias que proporcione una ayuda en el conocimiento y la escogencia de las herramientas tecnológicas en cuanto a la base de datos mas apropiada para sus necesidades de información.

1. PLANTEAMIENTO DEL PROBLEMA

1.1 DESCRIPCIÓN DEL PROBLEMA

Las Bases de Datos son parte fundamental en el crecimiento y estructura organizacional de una empresa ya que ésta permite la optima integración y manipulación de la información mejorando así tiempo de respuesta sobre las transacciones que se realizan en un Sistema de Información.

Carecer de un Sistema de Bases de Datos genera problemas tales como la manipulación voluminosa de papelería, tiempos de respuesta muy extensos para recuperar información, la información pudiese estar desactualizada en cierto momento y se estaría propenso a la escasez de integridad, seguridad y consistencia de datos, entre otros.

Es un hecho notable que en Colombia, las Pymes participan en casi todos los sectores de la economía, desde la industria y el comercio hasta tecnología y comunicaciones. “Estudios de entidades y proveedores de soluciones tecnológicas demuestran que más de la mitad de las Pymes colombianas carecen de una infraestructura tecnológica adecuada. En la mayoría de los casos, se tiene un cierto número de computadores personales con software de oficina mal utilizado y que no genera crecimiento al negocio”¹. La adquisición de tecnología informática y de comunicaciones es una decisión necesaria que ayudará a las

¹ Revista INTER – Cambio. Bogotá: Agosto – Septiembre, 2002.

empresas a ser mas competitivas comercialmente frente a otras que se consolidan no solamente en el país sino en Latinoamérica.

A pesar de que existen muchos conceptos o teorías que permiten conocer el manejo de las Bases de Datos, existe dificultad al decidir como combinarlas con los diferentes lenguajes de programación y plataformas operacionales para resolver sus problemas de procesamiento, éstas son acciones importantes que permiten enriquecer los datos para obtener un análisis sobre ellos y las nuevas demandas futuras de la organización.

“Esta situación se agrava cuando se añaden al panorama anterior los aspectos **económico**: donde algunas veces el costo del motor de Base de Datos es demasiado elevado e inasequible al presupuesto de la empresa; y de **capacitación**: el usuario se limita a manejar aquellos sistemas de información que en algún momento han desarrollado, sin atreverse a optar por otras alternativas que pueden resultar más convenientes o más favorables para la aplicación que se desea realizar.”²

Las Pymes del distrito de Barranquilla (por definir en convenio con la Corporación Educativa Mayor del Desarrollo Simón Bolívar y ACOPI) son en este caso el sector objetivo para estudiar la forma en que manipulan sus datos, implicando el proceso de reconocimiento de tecnologías, recursos de hardware y software utilizados, información redundante, inconsistente, compartida y la seguridad sobre los datos con el propósito de generar un diagnóstico que permita conocer el estado actual de la manipulación de la información en las Pymes.

² JARAMILO MARÍN, Mauricio. Revista INTER-Cambio. Bogotá: Agosto –Septiembre, 2002

1.2 FORMULACIÓN DEL PROBLEMA

¿Este estudio permitirá crear una base de conocimientos para formular patrones que apoyen la toma de decisiones con respecto a la optimización de los recursos tecnológicos y la información de una Pyme del distrito de Barranquilla?

2. OBJETIVOS

2.1 OBJETIVO GENERAL

Generar un documento que suministre información sobre la instalación, requerimientos y administración de bases de datos open source y propietarias que proporcione una ayuda en el conocimiento y la escogencia de las herramientas tecnológicas en cuanto a la base de datos mas apropiada para sus necesidades de información.

2.2 OBJETIVOS ESPECIFICOS

- 🌐 Proporcionar una base sólida en los fundamentos y desarrollos de las tecnologías de Bases de Datos.
- 🌐 Conocer las ventajas y desventajas que ofrecen los diferentes Sistemas de Bases de Datos (Distribuidas, Data warehousing, en la WEB, Centralizadas, CRM, PRM, SDK, MySQL, Oracle, Access, PostgreSql e Interbase).
- 🌐 Enriquecer los conocimientos con diversos conceptos que proporcionan los Sistemas de Bases de Datos.
- 🌐 Comprender de qué se tratan los Sistemas de Bases de Datos y sus beneficios para con las organizaciones que las manejan para la administración de la información.

- ④ Estudiar las características de los sistemas de bases de datos, lo que proporciona al usuario y cómo se comunica con los sistemas en funcionamiento.
- ④ Suministrar información acerca del comportamiento de los motores de Bases de Datos frente a los diversos Lenguajes de Programación y Sistemas Operacionales para lograr una mejor utilización de éstas en pro de la conservación y la operatividad de la información en la organización.
- ④ Proporcionar información sobre la administración, instalación y configuración de Bases de Datos Open Source (MySQL, PostgreSql e Interbase) y Propietarias (Oracle, Access)
- ④ Realizar encuestas en una serie de empresas para conocer acerca de los Sistemas de Bases de Datos que utilizan o implementan para la manipulación de la información.

3. JUSTIFICACIÓN

Las tecnologías de la información y las comunicaciones (TIC), pertenecientes al producto resultante de la unión de las telecomunicaciones, la informática y la electrónica, se han convertido en un factor importante a la hora de mejorar los procesos dentro de una organización, determinando el buen desarrollo o el fracaso de los negocios.

Las empresas con mayor aptitud de adaptarse a los cambios comerciales obtendrán grandes beneficios de las tecnologías procedentes de las comunicaciones y el cómputo distribuido, pues estarán capacitadas para ordenar sus sistemas de información y arquitecturas de datos de acuerdo a su área comercial y por lo tanto podrán operar eficaz y eficientemente en la obtención acertada de información importante.

Particularmente, las Bases de Datos son un escenario sólido que permite fortalecer el procesamiento eficiente e integro de la información que es la parte vital de cualquier institución, razón por la cual se debe brindar el conocimiento, a los futuros ingenieros, sobre la buena administración de ésta herramienta que brinda la tecnología informática.

El poseer un esquema de Bases de Datos traería beneficios sustanciales para una empresa que considere sus datos como un activo valioso. "Entre estos beneficios encontramos:

- 🕒 Los datos pueden compartirse
- 🕒 Es posible reducir la redundancia
- 🕒 Es posible evitar la inconsistencia

- Es posible brindar manejo de transacciones
- Es posible mantener la integridad
- Es posible hacer cumplir la seguridad”³

El conocimiento acerca de sistemas de bases de datos se ha convertido en una parte esencial en la formación informática. El propósito de esta línea de investigación es presentar temas especializados aplicados en los diseños de Bases de Datos orientados a los sistemas de información en general, gestión en WEB, sistemas de datos distribuidos, centralizados y data warehousing.

³ DATE, C.J. Sistemas de Bases de Datos. Prentice Hall, 7ª edición. México: 2001.

4. ALCANCES Y LIMITACIONES

El alcance establecido en este trabajo de investigación, es un documento con información amplia y suficiente sobre las diferentes bases de datos existentes, lo cual servirá como base en la toma de decisiones en lo que se refiere a su tecnología y manejo de datos. El proyecto investigativo sobre Bases de Datos no desarrollará modelos.

5. SISTEMA DE BASE DE DATOS

Un sistema de base de datos es básicamente un sistema computarizado para manipular registros, cuya finalidad general es almacenar información y permitir a los usuarios recuperar y actualizar esa información con base en peticiones. La información puede ser cualquier cosa que sea de importancia para el individuo u organización; es decir, todo lo que sea necesario para auxiliarle en el proceso general de su administración. Los usuarios del sistema pueden realizar una serie de operaciones:

Agregar nuevos archivos vacíos a la base de datos sobre los archivos existentes.

Insertar datos dentro de los archivos existentes.

Recuperar datos de los archivos existentes.

Modificar datos en archivos existentes.

Eliminar datos de los archivos existentes.

Eliminar archivos existentes de la base de datos.

⁴Antes de las bases de datos se utilizaban *ficheros secuenciales* como almacenes de datos. Estos daban un acceso muy rápido pero sólo de forma secuencial, es decir, para acceder a una posición, tenías que recorrer todo el fichero. Más tarde aparecieron los *ficheros indexados*, donde el acceso ya podía ser aleatorio, es decir, acceder de una vez a la posición deseada del fichero.

⁴ VAQUER CRUSAT, Marc. Comparando MySQL, PostgreSQL, Interbase y SAP DB.

El sistema de ficheros era el sistema más común de almacenamiento de datos. Para compartir los datos entre varias máquinas surgió el *NFS*, y más tarde para evitar fallos en los sistemas de fichero aparecieron los sistemas *RAID 1 (mirror)*.

Pero los programas y datos cada vez eran más complejos y grandes. Se requería de un almacenamiento que garantizara un cierto número de condiciones y que permitiera operaciones complejas sin que se violaran estas restricciones. Además cada usuario que accediera a los datos debía tener su trabajo protegido de las operaciones que hicieran el resto de usuarios.

Respondiendo a estas necesidades, surgieron las *bases de datos jerárquicas* donde los datos se situaban siguiendo una jerarquía. Hoy en día, la mayoría de entidades bancarias aun funcionan con este tipo de base de datos, ya que no necesitan nada más: un cliente tiene X cuentas bancarias, una cuenta bancaria tiene X tarjetas, etc.

Las bases de datos jerárquicas tenían el problema que los accesos a los datos eran unidireccionales, y era más complicado pero posible, aunque el tiempo de cálculo era mayor para hacer el camino inverso. Por ejemplo, era fácil saber que cuentas tenía un cliente, pero no tanto saber de que cliente era una cierta cuenta.

Para dar absoluta libertad a las relaciones entre tablas surgieron las bases de datos relacionales (*RDBMS*). Las *RDBMS* trajeron dos cosas muy importantes: las propiedades *ACID* y un lenguaje común de acceso a los datos: *SQL*.

Mediante *SQL*, por primera vez, decías *QUE* datos querías y no *COMO* los tenías que sacar. Apareció el término de transacción: agrupación de instrucciones *SQL* (por ejemplo varios *Selects*, *Inserts* y *Updates*).

Las propiedades ACID son:

- ⌚ *Atomicidad*. Cada transacción del usuario debe tratarse de forma atómica. O se ejecuta todo o nada.
- ⌚ *Consistencia*. Las transacciones han de cumplir las restricciones definidas dentro la base de datos. Si no las pueden cumplir, se evita su ejecución. De esta forma se conserva la integridad y coherencia de los datos.
- ⌚ *Aislamiento (Isolation)*. Los resultados de una transacción son inaccesibles a las otras transacciones que no está terminada.
- ⌚ *Durabilidad*. Una vez se ha completado la transacción, los resultados de la misma han de ser permanentes y sobrevivir a posibles caídas del sistema o la base de datos.

La mayoría de desarrolladores prefieren hoy en día sacrificar la velocidad por las funcionalidades. Debido a que las *RDBMS* tienen que soportar todas estas propiedades, nunca serán tan rápidas como trabajar directamente sobre ficheros, aunque internamente trabajen sobre ellos.

5.1 MEDICIÓN DEL RENDIMIENTO DE LAS BASES DE DATOS

⁵En la actualidad, la información es un elemento esencial para que cualquier empresa, sin importar su monto de facturación anual o tamaño, pueda llegar a cumplir sus objetivos. Las bases de datos son el corazón del que muchas veces depende la organización interna de una empresa, la relación con sus proveedores y los consumidores. Las bases de datos de misión crítica son aquellas que por lo imprescindible de la información, o por lo que generaría su momentánea

⁵ www.clarin.com.ar/suplementos/informatica/2000-07-12/t-00201d.htm

inactividad, deben tener seguridad para que las operaciones sean lo menos vulnerables posible. Los sistemas de base de datos paralelas, la partición de los volúmenes de la información, los seguimientos de rendimiento y los diagnósticos prematuros de problemas son parte de las soluciones que ofrecen los diseñadores de software más utilizados por las grandes y medianas empresas.

El mercado de bases de datos en sistemas abiertos es uno de los más competitivos en la industria de la computación. Los principales proveedores son Informix, Oracle, y Sybase, tres compañías que poseen la base instalada, la calidad de producto y la capacidad técnica para mantenerse en actitud competitiva por largo tiempo.

Para diferenciar y evaluar estos productos, se realizan diversas pruebas de rendimiento también llamadas **benchmarks** definidas por el "Transaction Processing Performance Council" (Consejo de negociación de ejecución de procesos - TPC). El benchmarking es un proceso sistemático y continuo para evaluar productos, servicios y procesos de trabajo de las organizaciones que son reconocidas como representantes de las mejores prácticas con el propósito de realizar mejoras organizacionales. Es un **proceso** pues es una actividad que transcurre y se desarrolla; está **estructurado** ya que tiene un método para realizarlo; no es una actividad de un sólo momento ni a corto plazo, es una **labor de búsqueda** de información para la toma de decisiones aplicable a todas las áreas de la organización, **útil** para todo tipo de empresas, orientado a identificar a quienes se destacan por tener las mejores prácticas con el fin de conocerlas, hacer comparaciones, aprender de ellas y proponer cambios.

Por ejemplo, una de estas pruebas, conocida como TPC-C, modela un sistema de control de inventarios y pedidos. Cada prueba es implementada en cada una de las bases de datos. Luego cada aplicación se corre y su rendimiento se mide por la cantidad de transacciones por minuto (TpmC-C) logradas.

Los resultados que logra una base de datos aparecen prominentemente en los anuncios de cada una. Por ejemplo, a finales de 1994 Informix se anunciaba triunfante con 3,118.2 TpmC. Poco después, Oracle sobrepasó a Sybase e Informix, anunciando 5,369 TpmC. Luego, Sybase reportaba 4,545 TpmC, que aunque no superaba a Oracle, demostraba una capacidad muy similar.

Sin embargo, como en toda prueba de rendimiento, cada cual presenta sólo una pequeña parte de la verdad. Por ejemplo, aunque Oracle iba ganando a mediados de 1995, resultó que los productos utilizados en las pruebas no estuvieron disponibles sino hasta principios de 1996. Además, Oracle utilizó 3.5 gigabytes de memoria principal.

Por otra parte, nadie menciona a las bases de datos "propietarias": Tandem obtuvo 20,918.03 TpmC con su base de datos NonStop SQLMP. El DB2 de IBM sólo es superado por Sybase en costo de rendimiento. Y RDB de Digital fue en su momento, el ganador absoluto del TPC-A.

⁶Para evaluar el costo-rendimiento correctamente, la mejor opción es diseñar las pruebas de rendimiento de manera específica a partir del volumen de transacciones más significativo de cada empresa. La recompensa adquirida con las pruebas de las bases de datos es el cumplimiento de las metas propuestas por la empresa.

El objetivo de optimizar el rendimiento consiste en minimizar el tiempo de respuesta de cada consulta y maximizar la ganancia de todo el servidor de bases de datos al disminuir el tráfico de red, la entrada y salida de disco, y el tiempo de CPU. Este propósito se logra mediante la comprensión de los requisitos de las

⁶ www.encyclopedia-mex.galeon.com/Cad_Practicas.htm



aplicaciones, la estructura física y lógica de los datos, y las contrapartidas entre usos conflictivos de la base de datos, como el procesamiento de transacciones en línea (OLTP), frente a la ayuda a la toma de decisiones.

Se deben tener en cuenta los problemas de rendimiento durante todo el ciclo de desarrollo, no al final cuando el sistema está implementado. Muchos aspectos de rendimiento que causan mejoras significativas se consiguen mediante un cuidadoso diseño desde el principio. Para optimizar de forma eficaz el rendimiento de SQL Server, se deben identificar las áreas que producirán los mayores incrementos del rendimiento en la variedad más amplia de situaciones y centrar el análisis en ellas.

Algunos de los temas que se deben considerar para la medición de rendimiento o benchmarking de Bases de Datos son:

Diseño de bases de datos: Describe cómo el diseño de la base de datos es la forma más eficaz de mejorar el rendimiento global. El diseño de bases de datos incluye el esquema lógico de la base de datos (tablas y restricciones) y los atributos físicos (sistemas de discos, colocación de objetos e índices).

Optimización de consultas: Determina cómo el diseño correcto de las consultas utilizadas por una aplicación puede mejorar considerablemente el rendimiento.

Diseño de aplicaciones: Expone cómo el diseño correcto de la aplicación de usuario puede mejorar notablemente el rendimiento. El diseño de aplicaciones incluye los límites de las transacciones, los bloqueos y el uso de lotes.

Optimizar el rendimiento de programas y herramientas: Define cómo algunas de las opciones disponibles con los programas y herramientas proporcionados con Microsoft SQL Server versión 7.0 pueden resaltar la forma de mejorar el rendimiento de estas herramientas y el efecto de ejecutar las herramientas y la aplicación al mismo tiempo.

Optimizar el rendimiento del servidor: Establece cómo se puede cambiar la configuración de Microsoft Windows NT, Windows 95 y 98, y SQL Server para mejorar el rendimiento global.

⁷Las empresas invierten cientos de miles de dólares en modificaciones de equipo y programación especial para acelerar los sistemas que no pueden hacerlo con las óptimas pero lentas herramientas de cuarta generación.

Si bien es preciso estudiar otros aspectos del rendimiento en el nivel de sistema, como la memoria, el hardware, entre otros, la experiencia muestra que la ganancia de rendimiento procedente de estas áreas por lo general aumenta. SQL Server suele administrar automáticamente los recursos de hardware disponibles, lo que reduce la necesidad de realizar optimizaciones manuales extensas en todo el sistema.

Obviamente, cuando las ofertas son muchas y las funciones son complejas no es fácil tomar decisiones. En una época donde los recursos de las empresas deben ser utilizados al máximo, siempre es recomendable analizar los pros y contras de cada software, pensando en la función que cada usuario pueda asignarle a las diferentes tecnologías que ofrece el mercado.

⁷ www.sdgconsulting.net/spain/events/master/masterworkshop.html

6. BASES DE DATOS DISTRIBUIDAS

6.1 DEFINICIÓN

⁸Un sistema de base de datos distribuidas consiste en una colección de sitios, conectados por medio de algún tipo de red de comunicación, en el cual cada sitio es un sistema de base de datos completo por derecho propio, pero los sitios han acordado trabajar juntos, a fin de que un usuario de cualquier sitio pueda acceder a los datos desde cualquier lugar de la red, exactamente como si los datos estuvieran guardados en el propio sitio del usuario.

Las bases de datos distribuidas, es en realidad un tipo de datos **virtual** cuyas partes componentes están almacenadas en varias bases de datos “reales” distintas que se encuentran en varios sitios distintos (de hecho, es la unión lógica de esas bases de datos reales).

La distribución de datos a través de las distintas sedes o departamentos de una organización permite que éstos residan donde han sido generados o donde son más necesarios, pero continúan siendo accesibles desde otros lugares o departamentos diferentes. El hecho de guardar varias copias de la base de datos en diferentes sitios permite que puedan continuar las operaciones sobre la base de datos aunque algún sitio se vea afectado por algún desastre natural, como una inundación, un incendio o un terremoto. Se han desarrollado los sistemas de

⁸ DATE, C. J. Introducción a los Sistemas de Bases de Datos. 7 ed. México: Prentice Hall, 2001.

bases de datos **distribuidas** para manejar datos distribuidos geográfica o administrativamente a lo largo de múltiples sistemas de bases de datos.

En un sistema distribuido de bases de datos se almacena la base de datos en varias computadoras. Varios medios de comunicación, como las redes de alta velocidad o las líneas telefónicas, son los que pueden poner en contacto las distintas computadoras de un sistema distribuido. No comparten ni memoria ni discos. Las computadoras pueden variar en tamaño y función, pudiendo abarcar desde estaciones de trabajo a grandes sistemas.

Los **emplazamientos o nodos** son las computadoras que forman parte de un sistema distribuido. Para enfatizar la distribución física de estos sistemas se usa principalmente el término **emplazamiento**.

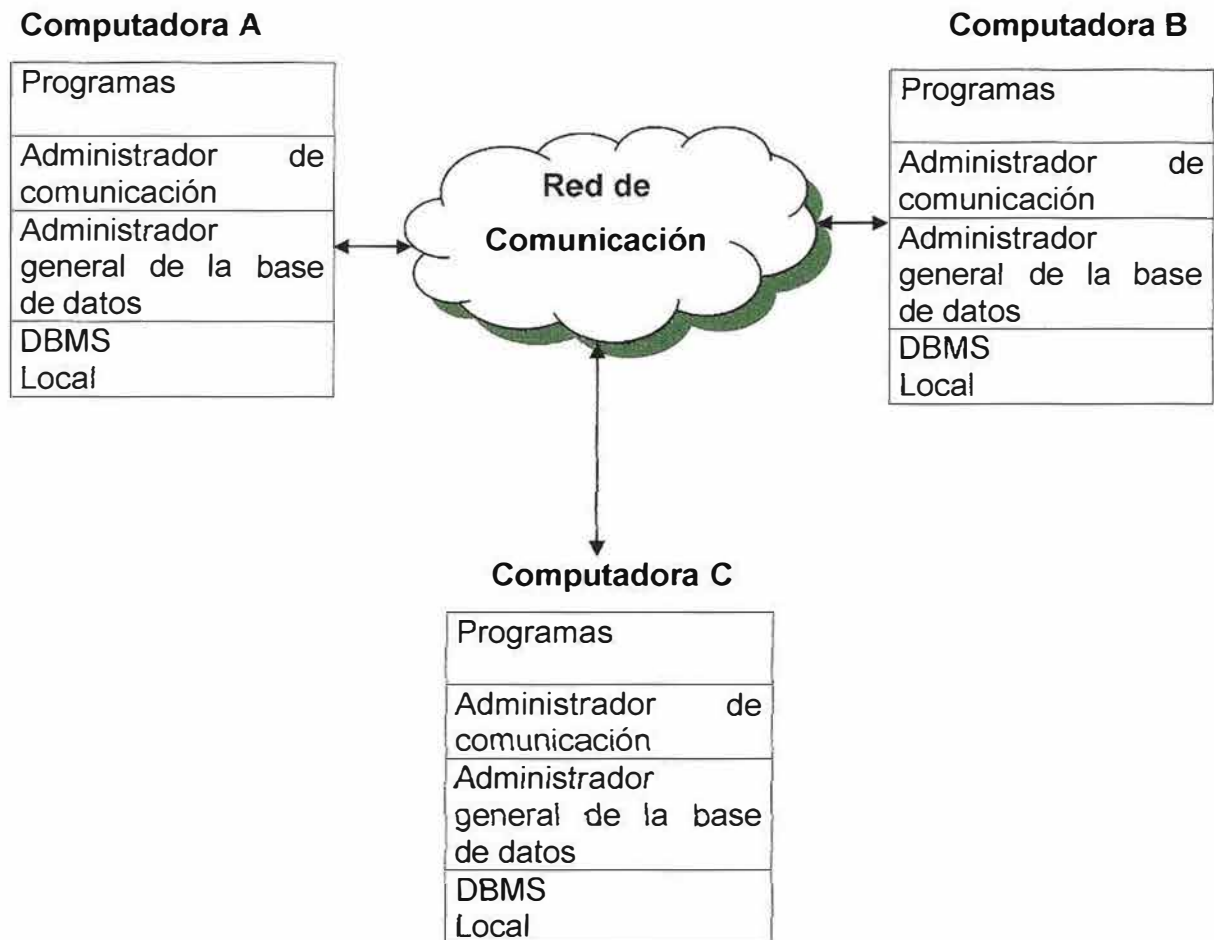
Las bases de datos distribuidas normalmente se encuentran en varios lugares geográficos distintos, se administra de forma separadas y poseen una interconexión más lenta. Se dan dos tipos de transacciones, las **locales** y las **globales**. Una transacción local es aquella que accede a los datos del único emplazamiento en el cual se inició la transacción. Una transacción global es aquella que accede a los datos situados en un emplazamiento diferente de aquel en el que se inició la transacción o a datos de varios emplazamientos distintos.

Cuando distintos usuarios en diferentes zonas geográficas van a compartir una gran base de datos, las compañías pueden dividir la base de datos en bases más pequeñas llamadas **fragmentos**, las cuales se pueden almacenar en lugares distintos. Una base de datos distribuida divide una base en fragmentos y los almacena en sitios diferentes. Uno de los objetivos del sistema distribuido es proteger a los programas de cambios cuando un fragmento se mueve de un sitio a otro.

En una base de datos distribuida, cada sitio cuenta con una computadora y un DBMS para administrar su propia base local. Los fragmentos de una base de datos distribuida se conectan entre sí por medio de una red de comunicación.

Cuando surge una petición de consulta de algún sitio en particular, el administrador general de la base analiza la petición de determinar qué fragmentos de la base se requieren. Un programa en un sitio puede invocar programas en otros sitios remotos para acceder a sus fragmentos locales con el DBMS local. La siguiente figura muestra la configuración de una base de datos distribuida:

Figura 1. Configuración de una base de datos distribuida

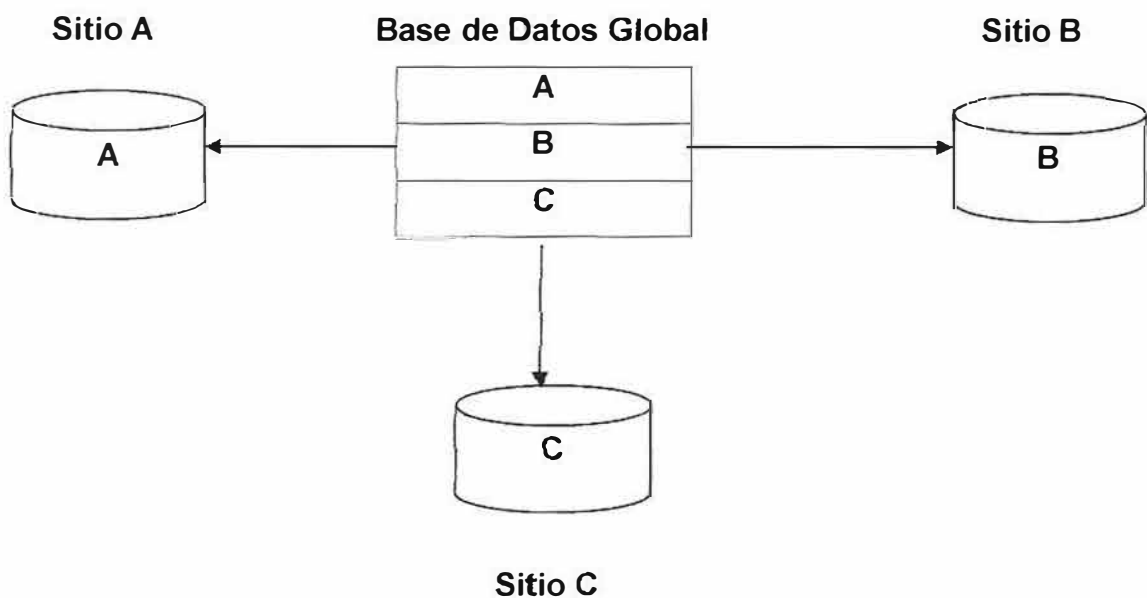


6.2 TIPOS DE BASES DE DATOS DISTRIBUIDAS

Hay tres tipos distintos de distribución de una base de datos:

- 🌐 **Base Distribuida Particionada.** Con este tipo de distribución, una base global se parte en tres (A, B y C), cada una de las cuales se almacena en un sitio distinto.

Figura 2. Base de datos distribuida particionada

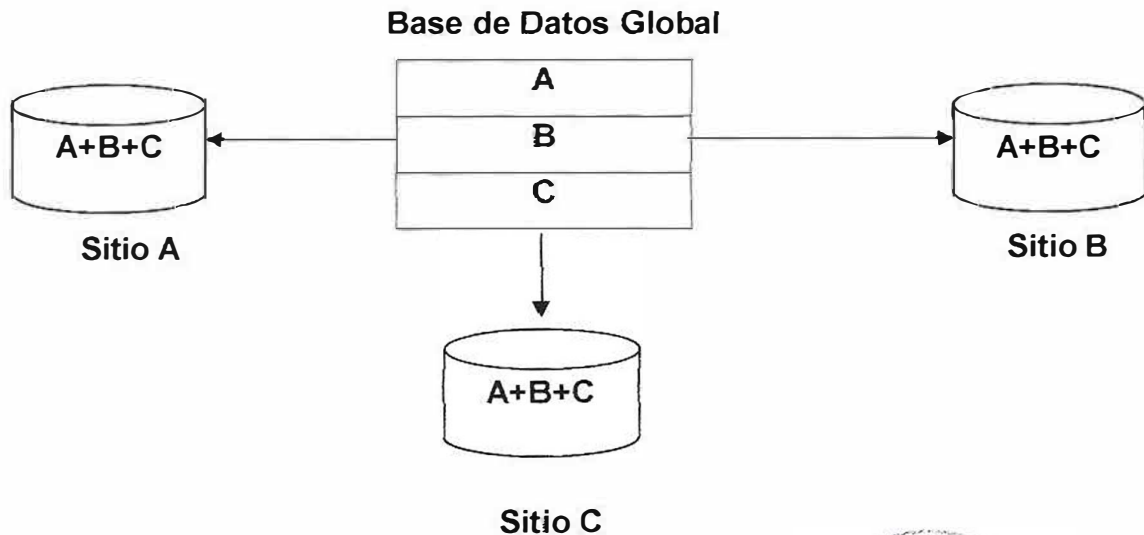


- 🌐 **Base de Datos Multiplicada y Distribuida.** Toda la base global se duplica en cada sitio. El objetivo de las réplicas es reducir los costos de comunicación y aumentar el desempeño del sistema eliminando la necesidad de la transmisión de datos entre ubicaciones distintas. Este tipo de distribución es rara vez implantado debido al costo global de las réplicas de la base (redundancia extrema) y a problemas en la actualización de los datos redundantes en varios sitios. Si se replica o particiona una relación k , se guardará una copia de la

misma en dos o más nodos. En el caso más extremo se tendrá una *réplica completa*, en la que se guarda una copia en cada emplazamiento del sistema. La replica presenta un cierto numero de ventajas e inconvenientes.

- Ⓒ **Disponibilidad.** Si falla uno de los emplazamientos que contienen la relación k , ésta se podrá encontrar en otro emplazamiento. Por lo que el sistema puede seguir procesando las consultas que impliquen a k , a pesar del fallo.
- Ⓒ **Aumento del paralelismo.** En el caso de que la mayor parte de los accesos a la relación k sólo den por resultado la lectura de la misma, varios emplazamientos pueden procesar en paralelo las consultas que impliquen a k . Cuantas más réplicas de k haya, mayor será la posibilidad de que el dato buscado se halle en el emplazamiento en que se esté ejecutando la transacción. Por tanto, la replica de los datos minimiza el tráfico de datos entre los emplazamientos.

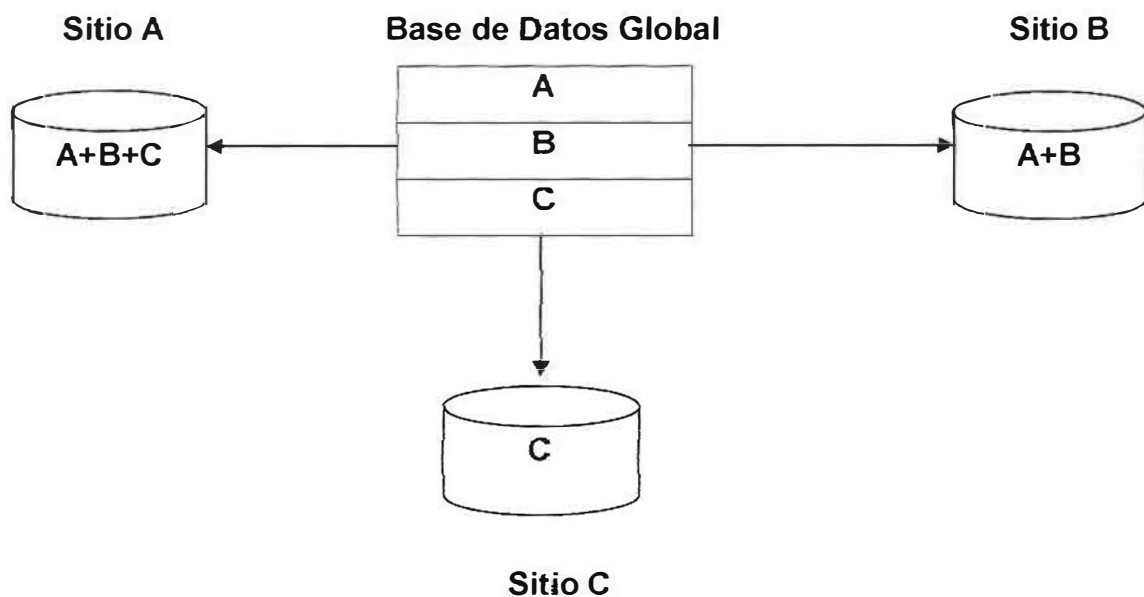
Figura 3. Aumento de paralelismo



9 Distribución de la Base de Datos combinando Réplicas y Particiones.

Cada nodo contiene un subconjunto único de la base y una selección de fragmentos réplica de otros sitios. En la práctica, esta distribución es la que se emplea con mayor frecuencia. Un archivo se parte basándose en su patrón de uso, es decir, los datos se guardarán en los lugares donde tengan mayor probabilidad de acceso. Sin embargo, cuando se requiere cierto fragmento en más de un sitio, el fragmento puede estar almacenado redundantemente en cada uno de esos sitios.

Figura 4. Base de datos combinando réplicas y particiones.



6.3 DIVISIÓN PARA ALMACENAMIENTO

Una base de datos se divide, para almacenamiento, de una de las siguientes maneras:

⁹ PRESSMAN, Roger. Ingeniería de software : un enfoque práctico. 2 ed. Bogotá: Mc Graw Hill, 1988.

- ④ **Partición Horizontal:** Se escogen las columnas deseadas de un archivo de la base.
- ④ **Partición Vertical:** Se escogen registros basándose en el valor de algunos campos de la base.
- ④ **Partición Horizontal y Vertical (Arborescente):** Se escogen columnas deseadas de un archivo de la base y registros basándose en el valor de algunos campos de la base.

Las bases de datos distribuidas constituyen el método preferido en los sistemas de tiempo real, debido a que la multitarea es muy común y que los datos se procesan frecuentemente en paralelo. Si la base de datos es distribuida y no centralizada, las tareas individuales pueden acceder a sus datos más rápidamente, fiablemente y con menos inconvenientes. El uso de una base de datos distribuida para aplicaciones de tiempo real divide el “tráfico” de entrada / salida y acorta las colas de tareas en espera, para acceder a una base de datos. Además, un fallo de una base de datos raramente causará el fallo del sistema entero si se construyen con redundancia.

6.4 VENTAJAS DE LAS BASES DE DATOS DISTRIBUIDAS

Un sistema de bases de datos distribuidas ofrece las siguientes ventajas:

- ④ **Costo reducido en la comunicación de datos:** Una base distribuida reduce el tráfico en las intercomunicaciones ya que la mayoría de las preguntas o consultas se pueden contestar sin necesidad de acceder a los archivos de lugares distantes.

- ④ **Autonomía local:** En un sistema distribuido, los datos obtenidos en algún lugar en particular son generalmente procesados sólo por usuarios locales. Así, la distribución del almacenamiento de la base de datos permite que los grupos individuales de usuarios tengan control sobre sus propios datos. Mientras tanto, los usuarios en cualquier otro lugar pueden también acceder a los datos almacenados en otras localidades por una red de comunicación.
- ④ **Confiabilidad:** Cuando un computador de un nodo particular se “cae”, el procesamiento de la base puede continuar en otros nodos porque cada nodo está equipado con su propia computadora y fragmentos locales de la base de datos. No se requiere la participación de ningún programador para transferir datos de un nodo a otro.
- ④ **Mayor accesibilidad:** Es posible acceder a una cuenta desde un sitio diferente a donde fue creada y viceversa, por medio de la red de comunicación. Por ejemplo, si usuario se encuentra en San Francisco puede acceder a una cuenta de la base de datos de los Ángeles.
- ④ **Eficiencia de procesamiento:** Los datos se mantienen cerca del punto donde usan más frecuentemente.

6.5 DESVENTAJAS DE LAS BASES DE DATOS DISTRIBUIDAS

Las eficiencias del rendimiento conseguido mediante el uso de una base de datos distribuida, deben ser sopesadas frente a cualquier otro problema potencial asociado con la partición y replicación de los datos. Aunque la redundancia de los datos mejora el tiempo de respuesta, suministrando múltiples fuentes de información, los requerimientos de replicación para los archivos distribuidos

también producen problemas logísticos y de sobrecarga, puesto que todas las copias de los archivos deben ser actualizadas.

Los problemas comunes de una base distribuida son:

- **Control global de la concurrencia:** El control de concurrencia está basado en el bloqueo comprende la sincronización de las bases de datos de forma que todas las copias tengan la *misma y correcta información disponible* para los accesos. Para evitar interferencia debido a la concurrencia y enfilar la ejecución de los servicios presentes se utilizan varios métodos: Bloqueo y Estampado de tiempo, y Copia Primaria.

En el método de bloqueo y estampado de tiempo, se inician las siguientes tareas: 1) la base de datos es “bloqueada”, de forma que se asegure el control de concurrencia; no se permite E/S; 2) se realiza la actualización requerida; 3) la base de datos es desbloqueada; 4) los archivos son validados para asegurar que todas las actualizaciones se han hecho correctamente, y 5) se reconoce que la actualización se ha terminado. Todas las tareas de bloqueo están monitorizadas por un reloj maestro (es decir, estampado de tiempo). Los retardos implicados en estos procedimientos y los problemas para evitar actualizaciones inconsistentes e interbloqueos, se reducen por el amplio uso de las bases de datos distribuidas.

En el método de copia primaria, para un objeto dado A, el sitio que mantiene la copia primaria de A manejará todas las operaciones de bloqueo que involucren a A (las copias primarias de objetos diferentes estarán generalmente en sitios diferentes). Bajo este método, el conjunto de todas las copias de un objeto puede ser considerado como un solo objeto para efectos de bloqueo, y la cantidad total de mensajes se reducirá de $5n$ a $2n + 3$ (una solicitud de bloqueo, una garantía de bloqueo, n actualizaciones, n notificaciones y una

solicitud de desbloqueo). Pero observe de nuevo que esta solución implica una gran pérdida de autonomía; ahora, una transacción puede fallar si no hay una copia primaria disponible, aún cuando la transacción sea de sólo lectura y tenga una copia local disponible.

Un problema con el bloqueo que se presenta en un sistema distribuido es que puede conducir a un bloqueo mortal global. El bloqueo mortal global es aquél que involucra dos o más sitios.

El problema con un bloqueo mortal es que *ningún sitio puede detectarlo usando solamente información interna de ese sitio*. En otras palabras, no hay ciclos en el grafo de espera local, pero aparecerá un ciclo si se combinan esos dos grafos locales para que formen un grafo global. De aquí se desprende que la detección del bloqueo mortal global incurre en más sobrecarga de comunicaciones, debido a que requiere que los grafos locales individuales estén reunidos de alguna forma.

Se han desarrollado algunas técnicas para aumentar la velocidad de actualización y resolver el problema de concurrencia. Una de éstas, llamada el *protocolo del escritor exclusivo*, mantiene la consistencia de los archivos replicados permitiendo que sólo una tarea de escritura actualice un archivo en exclusiva. Por tanto, elimina la gran sobrecarga de los procedimientos de bloqueo y estampado. De cualquier modo, estas estrategias requieren de complicados protocolos de control los que pueden deteriorar el desempeño del sistema al generar tráfico extra entre las distintas localidades.

- ④ **Propagación de la actualización de los datos duplicados:** En un sistema distribuido, algunos datos pueden estar duplicados en varios sitios. Por tanto, la actualización de cualquier dato en cualquier sitio se debe generalizar a todos los lugares donde exista la duplicidad. Una dificultad inmediata es que

un sitio que mantiene una copia del objeto podría no estar disponible (debido a una falla del sitio o de la red) en el momento de la actualización.

Un esquema común para atacar los problemas que se presentan en la propagación de los datos es el llamado esquema de copia primaria, que funciona de la siguiente forma:

- ④ A una copia de cada objeto replicado se le designa como copia primaria. Todas las demás son copias secundarias.
 - ④ Las copias primarias de diferentes objetos están en diferentes sitios (por lo tanto, este es nuevamente un esquema distribuido).
 - ④ Decimos que las operaciones de actualización quedaron lógicamente terminadas tan pronto como se actualizó la copia primaria. Entonces, el sitio que mantiene esa copia es responsable de la propagación de la actualización hacia las copias secundarias en algún tiempo subsecuente.
-
- ④ **Optimización de los procesos de consulta:** Cuando una petición de consulta entra en un sistema distribuido, existen diversas maneras en que el sistema puede dividir la consulta en subpreguntas para recuperar datos de diferentes localidades. El tiempo de respuesta varía, dependiendo de la ruta de procesamiento escogida. Así, el sistema debe optimizar la ruta de procesamiento determinando la sucesión de accesos a sitios distintos tales que el número de registros correctos por transferir de un lugar a otro sea tan pequeño como sea posible.

El objetivo de minimizar la utilización de la red implica que el propio proceso de optimización de consultas debe ser distribuido al igual que el proceso de ejecución de la consulta. En otras palabras, el proceso de optimización general consistirá típicamente en un paso de optimización global seguido por pasos de optimización local en cada sitio afectado.

- ⌚ **Administración del catálogo:** La mayoría de los sistemas distribuidos no alcanzan la completa transparencia de la ubicación. Los usuarios no tienen que saber dónde están almacenados físicamente los datos, sino que deben ser capaces de comportarse como si todos los datos estuvieran almacenados en su sitio local. Por esto requieren de la participación de los usuarios que deben transferir archivos para su procesamiento local.

En un sistema distribuido, el catálogo del sistema incluirá no solo datos usuales de este con relación a las varrels base, vistas, autorizaciones, etc., sino toda la información de control necesaria para permitir que el sistema proporcione la independencia de ubicación, fragmentación y replicación necesaria. ¿Dónde y cómo puede ser almacenado el propio catalogo?

- ⌚ **Centralizado:** El catálogo total es almacenado exactamente una vez en un sitio central.
 - ⌚ **Completamente replicado:** El catálogo total es almacenado por completo es cada uno de los sitios.
 - ⌚ **Dividido:** Cada sitio mantiene su propio catálogo de los objetos que están almacenados en ese sitio. El catálogo total es la unión de todos los catálogos locales *disjuntos*.
 - ⌚ **Combinación de uno y tres:** Cada sitio mantiene su propio catálogo local; además, un único sitio central mantiene una copia unificada de todos esos catálogos locales.
-
- ⌚ **Control de la recuperación:** El control de recuperación en sistemas distribuidos está basado típicamente en el protocolo de confirmación de dos fases. La confirmación de dos fases es necesaria en *cualquier* ambiente en el cual una transacción única puede interactuar con varios administradores de recursos autónomos, aunque es particularmente importante en un sistema

distribuido debido a que los administradores de los recursos en cuestión están operados en sitios distintos y por lo tanto, son *muy* autónomos. Surgen los siguientes puntos:

- ④ El objetivo de la “no dependencia de un sitio central” indica que la función coordinadora no debe ser asignada a un sitio distinguido en la red, sino que en su lugar, debe ser realizada por sitios diferentes para transacciones diferentes.
- ④ El proceso de confirmación de dos fases requiere que el coordinador se comunique con cada uno de los sitios participantes, lo cual significa más mensajes y más sobrecarga.
- ④ Si el sitio Y actúa como participante en un proceso de confirmación de dos fases coordinado por el sitio X, entonces el sitio Y *debe* hacer lo que le indica el sitio X (confirmar o deshacer, según se aplique) y esto es una pérdida tal vez menor de autonomía local.

Hay varias mejoras posibles para el algoritmo básico que buscan mejorar el rendimiento:

- ④ En primer lugar, si el agente de algún sitio participante específico de sólo lectura, ese participante puede responder “ignórame” en la fase 1 y el coordinador puede ignorar a ese participante en la fase 2.
- ④ Segundo, si *todos* los participantes responden “ignórame” en la fase 1, es posible omitir completamente la fase 2.
- ④ Tercero, hay dos variantes importantes sobre el esquema básico, que son *confirmar de manera presupuesta* y *deshacer de manera presupuesta*, tienen el efecto de reducir el número de mensajes involucrados en el caso de una operación exitosa o en el caso de falla.

6.6 OBJETIVOS DE LAS BASES DE DATOS DISTRIBUIDAS

- ④ **Autonomía local.** La autonomía local significa que todas las operaciones en un sitio dado están controladas por ese sitio; ningún sitio X debe depender de algún otro sitio Y para su operación satisfactoria. Implica que los datos locales son poseídos y administrados localmente con contabilidad local: todos los datos pertenecen “realmente” a alguna base de datos local, aun cuando estén accesibles desde otros sitios remotos. Por tanto, la seguridad, integridad y representación de almacenamiento de los datos locales permanecen bajo el control y jurisdicción del sitio local.

El objetivo de autonomía local no es totalmente alcanzable; existen varias situaciones en las que un sitio X dado **debe** transferir un determinado grado de control a algún otro sitio Y. Los sitios deben ser autónomos en el mayor grado posible.

- ④ **No dependencia de un sitio central.** La autonomía local implica que todos los sitios deben ser tratados como iguales. Por lo tanto, no debe haber particularmente ninguna dependencia de un sitio “maestro” central para algún servicio central tal que todo el sistema dependa de ese sitio central. Si el primer objetivo se logra, el segundo también **forzosamente**. Pero “la no dependencia de un sitio central” es necesaria por sí misma, aunque no se logre la autonomía local completa. Es un objetivo independiente.

La dependencia de un sitio central es indeseable porque el sitio central puede ser un cuello de botella y el sistema sería *vulnerable*; es decir, si el sitio central falla, también fallará todo el sistema (problema del “único punto de falla”).

- ④ **Operación Continua.** Una ventaja de los sistemas distribuidos es que deben proporcionar mayor *confiabilidad* y mayor *disponibilidad*. Confiabilidad, la probabilidad de que el sistema esté listo y funcionando en cualquier momento dado mejora, ya que los sistemas distribuidos pueden continuar operando, en un nivel reducido, cuando hay una falla en algún componente independiente. Disponibilidad, la probabilidad de que el sistema esté listo y funcionando continuamente a lo largo de un período especificado mejora, por la misma razón que la confiabilidad y por la posibilidad de replicación de datos.

Se aplican cuando ocurre un apagado no planeado, una falla de algún tipo en un punto del sistema, son difíciles de evitar. Por el contrario, los apagados planeados *nunca* deberían ser requeridos, nunca debería ser necesario apagar el sistema para realizar alguna tarea.

- ④ **Independencia de ubicación.** La idea principal de la independencia o transparencia de ubicación es que los usuarios deben ser capaces de comportarse como si todos los datos estuvieran almacenados en su propio sitio local. Éste objetivo simplifica los programas de usuario y las actividades terminales; y permite que los datos emigren de un sitio a otro sin invalidar ninguno de estos programas o actividades. Esta capacidad de emigración permite mover los datos por la red en respuesta a los diferentes requerimientos de desempeño.

- ④ **Independencia de fragmentación.** Un sistema soporta la fragmentación de datos cuando una varrel dada puede ser dividida en partes o *fragmentos*, para efectos de almacenamiento físico. La fragmentación es necesaria por razones de rendimiento: los datos pueden estar almacenados en la ubicación donde son usados más frecuentemente para que la mayoría de las operaciones sean locales y se reduzca el tráfico en la red.

Existen básicamente dos tipos de fragmentación, *horizontal* y *vertical*, que corresponden a las operaciones relacionales de restricción y proyección, respectivamente. Un fragmento puede ser derivado mediante cualquier combinación arbitraria de restricciones y proyecciones; arbitraria con excepción de que:

- En el caso de la **restricción**, las restricciones deben constituir una descomposición *ortogonal*, las restricciones en esa división deben estar todas disjuntas, ninguna tupla puede aparecer en más de una de ellas.
- En el caso de la **proyección**, las proyecciones deben constituir una descomposición *sin pérdida*, la descomposición en un conjunto de proyecciones tales que la varrel dada pueda ser reconstruida juntando de nuevo las proyecciones y ninguna de esas proyecciones sea redundante en el proceso de reconstrucción.

El efecto neto de estas dos reglas es que todos los fragmentos de una varrel dada serán *independientes*, ya que ninguno de ellos puede ser derivado a partir de los otros. La reconstrucción de la varrel original es lograda por medio de las operaciones juntar en fragmentos verticales y unir en los horizontales.

La facilidad de fragmentación y reconstrucción son dos de las razones principales por las que los sistemas distribuidos son relacionales; el modelo relacional proporciona las operaciones necesarias para estas tareas.

La idea principal de la independencia o transparencia de fragmentación es que los usuarios deben ser capaces de comportarse como si los datos estuvieran sin fragmentación alguna. Éste objetivo simplifica los programas de usuario y las actividades terminales; y permite que los datos sean fragmentados y los fragmentos sean redistribuidos en cualquier momento

en respuesta a los diferentes requerimientos de rendimiento, sin invalidar ninguno de esos programas o actividades.

La independencia de fragmentación implica que a los usuarios se les presentará una vista de los datos en la cual los fragmentos estarán re combinados lógicamente por medio de juntas y de uniones adecuadas. Es responsabilidad del *optimizador del sistema* determinar cuáles fragmentos necesitan ser accedidos físicamente para satisfacer cualquier solicitud de usuario dada.

- 🌀 **Independencia de replicación.** Un sistema soporta la replicación de datos cuando una varrel almacenada dada o un fragmento dado de una varrel guardada puede ser representada por muchas copias distintas, o *réplicas* guardadas en muchos sitios distintos.

Las réplicas son necesarias porque puede significar un mejor rendimiento, ya que las aplicaciones pueden operar sobre copias locales en vez de tener que comunicarse con sitios remotos; y una mejor disponibilidad, un objeto replicado dado permanece disponible para su procesamiento mientras esté disponible al menos una copia. La principal desventaja es que al actualizar un objeto replicado dado es necesario actualizar *todas las copias* de ese objeto: el problema de la propagación de la actualización.

La idea principal de la independencia o transparencia de replicación es que los usuarios deben ser capaces de comportarse como si los datos no estuvieran replicados. Éste objetivo simplifica los programas de usuario y las actividades terminales; y permite que las réplicas sean creadas y destruidas en cualquier momento en respuesta a los distintos requerimientos, sin invalidar ninguno de esos programas o actividades.

La independencia de replicación implica que es responsabilidad del optimizador del sistema determinar cuáles réplicas necesitan ser accedidas físicamente para satisfacer cualquier solicitud de usuario dada.

🕒 **Procesamiento de consultas distribuidas.** La optimización es más importante en un sistema distribuido que en uno centralizado. El punto básico es que habrá muchas formas posibles de mover los datos en el sistema para satisfacer la solicitud, y es crucialmente importante que se encuentre una estrategia eficiente razón por la cual los sistemas distribuidos siempre son relacionales. Es probable que un sistema relacional supere a uno que no lo es por varios órdenes de magnitud posibles.

🕒 **Administración de transacciones distribuidas.** Hay dos aspectos principales en la administración de transacciones: el control de la recuperación y de la concurrencia. En los sistemas distribuidos, una sola transacción puede involucrar la ejecución de código o actualizaciones en muchos sitios. Un agente es el proceso realizado en nombre de una transacción dada en un sitio dado. Cada transacción consiste en varios agentes. El sistema necesita saber cuándo dos agentes son parte de la misma transacción, no se debe permitir que dos agentes que son parte de la misma transacción caigan en bloqueo mortal entre ellos.

En lo que se refiere al control de recuperación, para asegurarse de que una transacción dada sea atómica (todo o nada) en el ambiente distribuido, el sistema debe asegurarse de que el conjunto de agentes para esa transacción sea confirmado o deshecho al unísono, este efecto puede lograrse por medio del protocolo de confirmación de dos fases. En lo que se refiere al control de concurrencia, en la mayoría de los sistemas distribuidos está basado generalmente en el bloqueo.

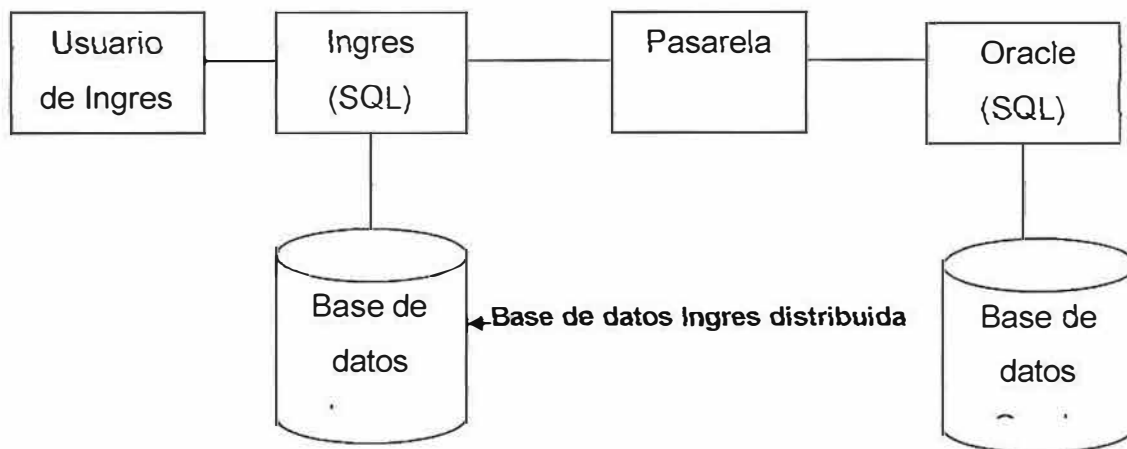
- ④ **Independencia de hardware.** Por lo general, las instalaciones de computadoras involucran en la realidad una gran diversidad de máquinas diferentes (IBM, ICL, HP, PC y estaciones de trabajo de diversos tipo, etc.) y existe una necesidad real de poder integrar los datos en todos estos sistemas y presentar al usuario una “imagen de sistema único”. Por lo tanto, es necesario tener la posibilidad de ejecutar el mismo DBMS en diferentes plataformas de hardware y hacer que esas máquinas diferentes participen como socios igualitarios en un sistema distribuido.
- ④ **Independencia de sistema operativo.** Este objetivo es resultado del anterior. Es necesario tener la posibilidad de ejecutar el mismo DBMS en diferentes plataformas de hardware y de sistema operativo (incluyendo diferentes sistemas operativos en el mismo hardware) y tener una versión MVS, una versión UNIX y una versión NT participando en el mismo sistema distribuido.
- ④ **Independencia de red.** Si el sistema tiene la posibilidad de soportar muchos sitios distintos con hardware distintos y sistemas operativos distintos, es obviamente necesario tener la posibilidad de soportar también una variedad de redes de comunicación distintas.
- ④ **Independencia de DBMS.** En los sistemas, en realidad se necesita que todos los ejemplares del DBMS en sitios diferentes soporten la misma interfaz, aunque no tienen que ser necesariamente copias del mismo software DBMS. Por ejemplo, si Ingres y Oracle soportan el estándar de SQL oficial, sería posible tener un sitio Ingres y un sitio Oracle comunicándose entre sí en el contexto de un sistema distribuido. En otras palabras, sería posible que el sistema distribuido fuera heterogéneo, al menos en cierto grado.

El soporte para la heterogeneidad es definitivamente necesario. El hecho es que por lo general las instalaciones de computación en la realidad emplean no

sólo muchas máquinas diferentes y muchos sistemas operativos diferentes, sino que muy frecuentemente, también DBMSs diferentes; y sería muy bueno si esos DBMS diferentes pudieran participar de alguna forma en un sistema distribuido. En otras palabras, el sistema distribuido ideal debe proporcionar independencia de DBMSs.

Para comunicar sistemas con DBMSs diferentes se utiliza un programa especial por lo general llamado pasarela cuyo objetivo es que éstos DBMSs se parezcan. En la siguiente gráfica se muestra una pasarela hipotética proporcionada por Ingres para Oracle.

Figura 5. Una pasarela hipotética proporcionada por Ingres para Oracle.



6.7 MODELO DE TRANSACCIONES DISTRIBUIDAS

Una **transacción** es una **unidad** de la ejecución de un programa que accede y posiblemente actualiza varios elementos de datos. El acceso a los diferentes elementos de datos en los sistemas distribuidos suele realizarse mediante transacciones, que deben conservar las propiedades ACID (Atomicidad, Consistencia, Aislamiento y Durabilidad). Hay que tener en cuenta dos tipos de

transacciones: las transacciones **locales** son las que tienen acceso y actualizan datos sólo en una base de datos locales; las transacciones **globales** son las que tienen acceso y actualizan datos en varias bases de datos locales. Las propiedades ACID de las transacciones globales, es mucho más complicada, dado que pueden estar participando en la ejecución varios emplazamientos. El fallo de uno de estos emplazamientos puede dar lugar a cálculos erróneos.

④ **Arquitectura del sistema.** Cada emplazamiento tiene su propio **gestor de transacciones locales**, cuya función es asegurar las propiedades ACID de las transacciones que se ejecutan en ese emplazamiento. Los diferentes gestores de transacciones colaboran para ejecutar las transacciones globales. Para comprender el modo en que se puede implementar un gestor de este tipo se puede definir un modelo abstracto del sistema de transacciones. Cada emplazamiento del sistema contiene dos subsistemas:

- ④ El gestor de transacciones administra la ejecución de las transacciones (o subtransacciones) que tienen acceso a datos guardados en un emplazamiento local. Las transacciones de este tipo pueden ser transacciones locales (es decir, transacciones que sólo se ejecutan en un emplazamiento) o parte de transacciones globales (es decir, transacciones que se ejecutan en varios emplazamientos).
- ④ El coordinador de transacciones coordina la ejecución de las diferentes transacciones (tanto locales como globales) iniciadas en ese emplazamiento.

La estructura de los gestores de transacciones es parecida en muchos aspectos a la utilizada en el caso de los sistemas centralizados. Cada gestor de transacciones es responsable de conservar un registro histórico con fines de recuperación y participar en un esquema adecuado de control de

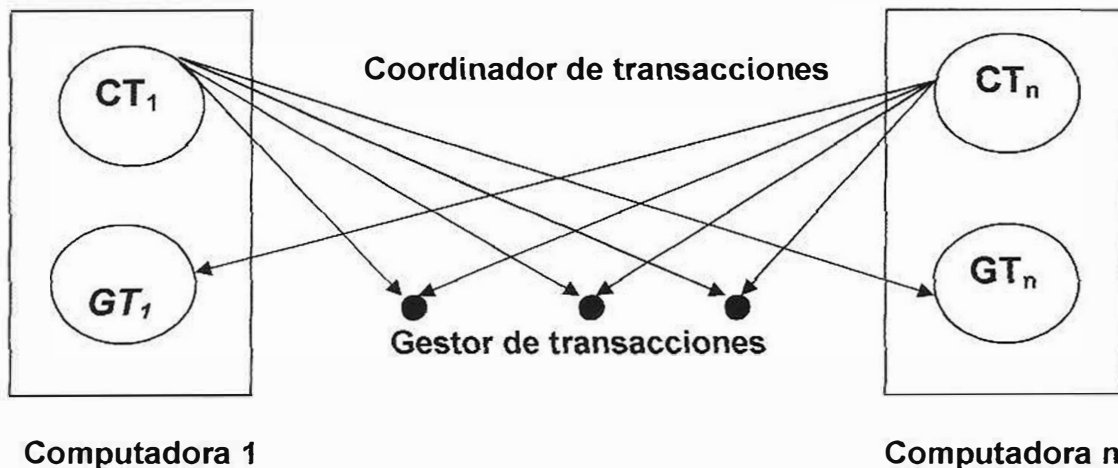
conurrencia para coordinar la ejecución concurrente de las transacciones en ejecución en ese emplazamiento.

Hay que modificar los esquemas de recuperación y de concurrencia para acomodar la distribución de las transacciones.

El subsistema coordinador de transacciones no se necesita en los entornos centralizados, dado que las transacciones tienen acceso a los datos en un único emplazamiento. El coordinador de transacciones, como su propio nombre indica, es responsable de coordinar la ejecución de todas las transacciones iniciadas en ese emplazamiento. Para cada transacción de ese tipo, el coordinador es responsable de:

Iniciar la ejecución de la transacción, dividir la transacción en una serie de subtransacciones y distribuir estas subtransacciones para su ejecución en los emplazamientos adecuados; y coordinar la terminación de la transacción, lo que puede hacer que la transacción se comprometa en todos los emplazamientos o se aborte en todos ellos.

Figura 6. Arquitectura del sistema global.



- ② **Modos de fallo del sistema.** Los sistemas distribuidos pueden sufrir los mismos tipos de fallos que los sistemas centralizados (por ejemplo, errores de

software, errores de hardware o fallos graves de disco). Además, hay otros tipos principales de fallos: Fallos de un emplazamiento, pérdidas de mensajes, fallos de un enlace de comunicaciones y división de la red.

Los sistemas utilizan protocolos de control de la transmisión, como TCP/IP, para tratar la pérdida o deterioro de los mensajes posibles en los sistemas distribuidos.

Los emplazamientos del sistema pueden estar conectados físicamente en gran número de maneras: **red completamente conectada, red en anillo, red conectada parcialmente, red en árbol, red en estrella.**

6.8 SISTEMAS CLIENTE - SERVIDOR

Los sistemas cliente-servidor son considerados como un caso especial de los sistemas distribuidos en general en el cual a) algunos sitios son **clientes** y algunos son **servidores**, b) todos los datos residen en los sitios servidores, c) todas las aplicaciones son ejecutadas en los sitios clientes y d) “se ven las costuras” (no existe independencia total de ubicación).

El termino “cliente-servidor” es una **arquitectura** o división logica de responsabilidades. El cliente es la aplicación conocida también como **interfaz o parte frontal** y el servidor es el DBMS conocido también como **servicio de fondo o parte dorsal**.

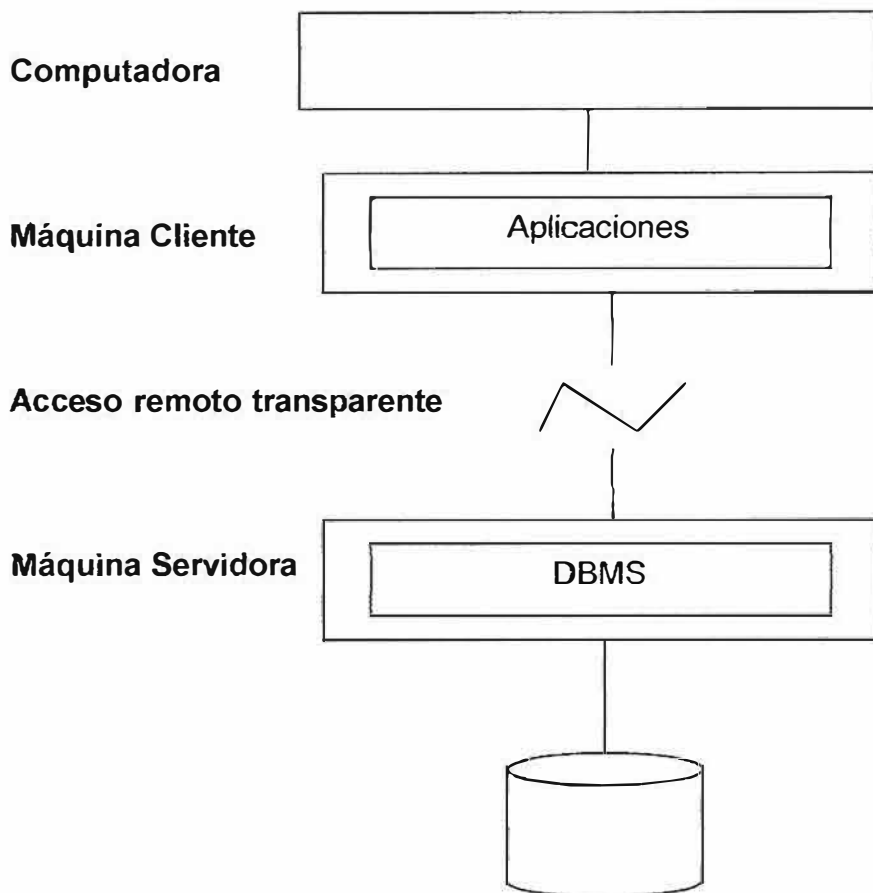
Son posibles muchas variaciones del tema básico:

Varios clientes pueden compartir el mismo servidor (éste es el caso normal).

Un solo cliente puede acceder a varios servidores. Ésta posibilidad se divide a su vez en dos casos:

- ④ El cliente está limitado a acceder a un solo servidor a la vez; es decir, cada solicitud individual de base de datos debe ser dirigida hacia un solo servidor. No es posible combinar, dentro de una sola solicitud, datos de dos o mas servidores diferentes. Además, el usuario debe saber cuál servidor en particular almacena cuáles partes de datos.
- ④ El cliente puede acceder a muchos servidores simultáneamente; es decir, una sola solicitud de base de datos pueden combinar datos de varios servidores, lo que significa que los servidores ven al cliente como si fueran en realidad, sólo un servidor, y el usuario no tiene que saber qué servidor almacena qué partes de datos.

Figura 7. Un sistema Cliente – Servidor



7. BASES DE DATOS EN LA WEB

7.1 INTEGRACIÓN DE BASES DE DATOS EN LA WEB

¹⁰En la actualidad, muchas instituciones se han dado cuenta de la importancia que tiene el Web, ya que con ello pueden lograr una mejor comunicación con personas o instituciones situadas en cualquier lugar del mundo.

Gracias a la conexión con la red mundial Internet, poco a poco, cada individuo o institución va teniendo acceso a mayor cantidad de información de las diversas ramas de la ciencia con distintos formatos de almacenamiento. La mayor parte de información es presentada de forma estática a través de documentos HTML, lo cual limita el acceso a los distintos tipos de almacenamiento en que ésta pueda encontrarse.

Pero, en la actualidad surge la posibilidad de utilizar aplicaciones que permitan acceder a información de forma dinámica, tal como a bases de datos, con diversos contenidos y formatos.

Una de las ventajas de utilizar el Web para este fin, es que no hay restricciones en el sistema operativo que se debe usar. Además, no hay necesidad de cambiar el formato o estructura de la información dentro de las bases de datos.

¹⁰ www.intelibizsa.com/datawarehouse/

7.2 ¿CÓMO FUNCIONA LA INTEGRACIÓN DE BASES DE DATOS EN LA WEB?

Para requerir acceso desde el Web hasta una base de datos no sólo se necesita de un browser (navegador) del Web y de un Servidor Web, sino también de un software de procesamiento (aplicación CGI), el cual es el programa que es llamado directamente desde un documento HTML en el cliente. Dicho programa lee la entrada de datos desde que provienen del cliente y toma cierta información de variables de ambiente. El método usado para el paso de datos está determinado por la llamada CGI(Common Gateway Interface).

Una vez se reciben los datos de entrada (sentencias SQL o piezas de ellas), el software de procesamiento los prepara para enviarlos a la interfaz en forma de SQL, y luego ésta procesa los resultados que se extraen de la base de datos.

La interfaz contiene las especificaciones de la base de datos necesarias para traducir las solicitudes enviadas desde el cliente, a un formato que sea reconocido por dicha base. Además, contiene toda la información, estructuras, variables y llamadas a funciones, necesarias para comunicarse con la base de datos.

El software de acceso usualmente es el software distribuido con la base de datos, el cual permite el acceso a la misma, a través de solicitudes con formato. Luego, el software de acceso recibe los resultados de la base de datos, aún los mensajes de error, y los pasa hacia la interfaz, y ésta a su vez, los pasa hasta el software de procesamiento.

Cualquier otro software (servidor HTTP, software de redes, etc.) agrega enlaces adicionales a este proceso de extracción de la información, ya que el software de

procesamiento pasa los resultados hacia el servidor Web, y éste hasta el browser del Web (ya sea directamente o a través de una red).

7.3 CATEGORIZACIÓN DE INTERFACES WEB/DBMS

Tradicionalmente en el Web se han utilizado documentos HTML estáticos para los cuales se creaban las posibles respuestas ante peticiones del cliente. Este método requiere de un gran desarrollo de aplicaciones y de mantenimiento de las mismas. Al interactuar con las bases de datos, este proceso se complica aún más.

Como la necesidad de acceder a bases de datos desde el Web se ha incrementado, han sido creadas también interfaces que manipulan sus escritos para procesar la información, utilizando sentencias SQL para requerir datos a la base.

Los productos iniciales son simplemente modelos del ambiente cliente/servidor, con una capa adicional para crear resultados HTML que pueden ser vistos a través del Web, por medio de un procesamiento de los datos de la forma que fueron introducidos por el cliente.

Además, al usar estas interfaces se puede crear el programa principal de la aplicación. Como se puede ver, estas herramientas permiten construir poderosas aplicaciones en el Web, pero se requiere que programadores experimentados logren un desarrollo a gran escala. Debido a que el mantenimiento de las mismas es significativamente más complejo y extenso.

Una de las estrategias más famosas para la creación de aplicaciones de interacción con el Web, es la de descargar del Web, aplicaciones o componentes funcionales que se ejecutarán dentro del browser. Con ellas se realizará un

procesamiento complejo del lado del cliente, lo cual requiere un gran esfuerzo para crear las piezas de la aplicación. Estas estrategias poseen dos características principales: garantizan la seguridad tanto en los sistemas de distribución como en la comunicación que se establece con tales aplicaciones, a través de Internet.

Han aparecido bibliotecas que incluyen motores propios de servidor que corren de forma conjunta con el Servidor Web, lo cual facilita el desarrollo de nuevas aplicaciones.

Una aplicación que posibilita interconectar al Web con una base de datos tiene muchas ventajas, además de que las funciones que cumplen actualmente los Servidores Web y las herramientas de desarrollo de aplicaciones Web, hacen más fácil que nunca la construcción de aplicaciones más robustas. Tal vez el mayor beneficio del desarrollo de estas aplicaciones en el Web sea la habilidad de que sean para múltiples plataformas, sin el costo de distribuir múltiples versiones del software.

Cada una de las interfaces para comunicar al Web con bases de datos, ha sido creada basándose en una tecnología de integración especial, a través de procesos de interconexión especiales, que serán descritos en el siguiente apartado.

7.4 TECNOLOGÍAS PARA LA INTEGRACIÓN DE BASES DE DATOS EN LA WEB

- ④ El Common Gateway Interface (CGI)
- ④ Interfaz de Programación de Aplicaciones (API)
- ④ Interfaz de Programación de Aplicaciones del Servidor Internet (ISAPI)
- ④ Java, JDBC y JavaScript

- Aplicaciones Java
- Conectividad de Bases de Datos de Java (JDBC)
- JavaScript

Cuando se utiliza una interfaz para lograr la integración del Web con cierta base de datos, se puede verificar que los procesos seguidos varían, dependiendo de la tecnología que se esté utilizando, lógicamente. Entre estas tecnologías se tienen las siguientes:

• El Common Gateway Interface (CGI)

Actualmente, ésta es la solución que más se está utilizando para la creación de interfaces Web/DBMS. Fue probada por primera vez en el servidor NCSA. Entre las ventajas de la programación CGI, se tiene:

- Su sencillez, ya que es muy fácil de entender.
- Es un lenguaje de programación independiente, ya que los escritos CGI pueden elaborarse en varios lenguajes.
- Es un estándar para usarse en todos los servidores Web.
- Puede funcionar bajo una arquitectura independiente, ya que ha sido creado para trabajar con cualquier arquitectura de servidor Web.
- No pone en peligro al servidor, ya que la aplicación CGI se encuentra funcionando de forma independiente, en cuanto al cumplimiento de todas las tareas que éste se encuentre realizando, o al acceso del estado interno del mismo.

Pero el CGI presenta cierta desventaja en su eficiencia, debido a que el Servidor Web tiene que cargar el programa CGI y conectar y desconectar con la base de datos cada vez que se recibe una requisición (petición, requerimiento). Además,

no existe un registro del estado del servidor, sino que todo hay que hacerlo manualmente.

Interfaz de Programación de Aplicaciones (API)

Es un conjunto de rutinas, protocolos y herramientas para construir aplicaciones de interfaz. Una buena API hace más fácil el trabajo de desarrollo de un programa, ya que debe proveer todos los bloques para construirlo. El programador lo único que hace es poner todos los bloques juntos.

API está diseñado especialmente para los programadores, ya que garantiza que todos los programas que utilizan API, tendrán interfaces similares. Asimismo, esto le facilita al usuario aprender la lógica de nuevos programas. Cuando se realiza una requisición, el servidor llamará al API, brindando la ventaja de disponer de una mayor cantidad de servicios.

Interfaz de Programación de Aplicaciones del Servidor Internet (ISAPI)

Es la interfaz propuesta por Microsoft como una alternativa más rápida que el CGI, y ya está incluida en el Servidor Microsoft Internet Information (IIS).

Así como en el CGI, los programas escritos usando ISAPI habilitan un usuario remoto (distante, lejano) para ejecutar un programa, busca información dentro de una base de datos, o intercambia información con otro software localizado en el servidor.

Los programas escritos usando la interfaz ISAPI, son cargados por el servidor Web cuando éste se inicia. Estos programas se vuelven residentes en memoria, por lo que se ejecutan mucho más rápido que las aplicaciones CGI, debido a que requieren menos tiempo de uso de CPU al no iniciar procesos separados.

Uno de los programas ISAPI más usados es el HTTPODBC.DLL que se usa para enviar y/o devolver información hacia y desde las bases de datos, a través de ODBC. Además, ISAPI permite realizar primeramente un procesamiento previo de la solicitud y uno posterior de la respuesta, con lo cual manipula la solicitud/respuesta HTTP. Los filtros ISAPI pueden utilizarse para aplicaciones tales como autenticación, acceso o apertura de sesión.

Java, JDBC y JavaScript

Java es tanto un compilador como un lenguaje intérprete. Java ofrece un ambiente de programación muy sencillo, robusto, dinámico, de propósito general, orientado a objetos y múltiples plataformas, creado por Sun Microsystems.

El código fuente de Java es convertido en instrucciones binarias simples, y compilado con un formato universal. El Compilador realiza todas las actividades de un procesador real en un ambiente virtual más seguro. Es decir, ejecuta instrucciones, crea y manipula información, carga y hace referencia a bloques de código nuevos.

El Intérprete, que es pequeño y muy útil, es capaz de ser implantado en cualquier forma que se desee para un sistema operativo particular. Este puede correr como una aplicación independiente, o como una parte de otro software, tal como el Web Browser. El concepto de Java es diferente al de CGI, ya que el CGI se ejecuta en el servidor, mientras que Java se ejecuta en el cliente.

Aplicaciones Java

Los programadores pueden desarrollar pequeñas aplicaciones, las cuales permiten tener sitios Web con una gran funcionalidad en cuanto a: animación,

actualización en vivo, interacción bidireccional y más. Al integrarse en una página Web, las aplicaciones de Java tienen acceso a:

- Recreación de gráficos expertos.
- Interacción en tiempo real con los usuarios.
- Actualización en vivo de la información.
- Interacción instantánea con los servidores a través de la red.

Las aplicaciones de Java pueden obtenerse en cualquier servidor con esta tecnología y funcionan de forma segura bajo cualquier plataforma o arquitectura de CPU, permitiendo introducirlas en páginas HTML.

Las aplicaciones son programas independientes.

Java está diseñado para proveer la máxima seguridad posible en redes públicas, con múltiples formas de seguridad ante virus, posibles invasiones o accesos incorrectos, archivos basura, etc. Java es como una versión de C++, en la cual se puede causar cualquier daño. Es funcional como C y modular como C++.

• Conectividad de Bases de Datos de Java (JDBC)

Se considera el primer producto estándar de Java con DBMS, creado y ofrecido por primera vez en marzo de 1996. Crea una interfaz con un nivel de programación que le permite comunicarse con las bases de datos mediante un concepto similar al de componentes ODBC, el cual se ha convertido en el estándar que se utiliza en computadoras personales o en redes locales.

El estándar de JDBC está basado en un nivel de interfaz con instrucciones SQL X/Open, que es básicamente lo mismo que en ODBC. Las clases de objetos para iniciar la transacción con la base de datos, están escritas completamente en Java, lo cual permite mantener la seguridad, robustez y portabilidad de este ambiente.



El puente JDBC-ODBC manipula la traducción de llamadas JDBC a aquellas que puedan ser entendidas por el cliente ODBC a un nivel de lenguaje C.

JavaScript

Es un lenguaje de escritos compacto, basado en objetos, para el desarrollo de aplicaciones Internet Cliente/Servidor. Es un lenguaje muy poderoso y especialmente diseñado para la creación de escritos, que se alojan dentro de un documento HTML. Dicho lenguaje es propiedad de Netscape.

Es un API programable que permite crear escritos de eventos, objetos y acciones, bajo cualquier plataforma. Gracias a que JavaScript es parte de la conexión en vivo, se puede usar para crear interacciones entre documentos HTML, Plug-ins (aplicaciones que corren dentro del browser de la Web) y Java.

Mediante el uso de JavaScript se pueden enviar respuestas ante una variedad de eventos, objetos y acciones, permitiendo cambiar imágenes o activar sonidos ante determinados eventos, tales como entrar o salir de una página, presionar el ratón, etc.

Las sentencias JavaScript que reconocen y responden ante eventos, pueden ser introducidas directamente en una página Web. Por ejemplo, se puede escribir una función JavaScript que verifique la correcta entrada de datos a una forma, sin necesidad de transmisión de datos a través de la red. Así, una página HTML con código JavaScript puede interpretar el texto introducido y alertar al usuario si el dato es inválido.

Para la interconexión de una base de datos con el Web, se pueden construir escritos CGI en lenguajes de programación adecuados, tales como C o Perl.



Estos escritos se mantendrán cargados en memoria a la espera de requisiciones o llamadas del servidor, y realizando las conexiones con la base de datos.

Algo un poco más complejo es crear un servidor personal, tal como APACHE, y agregarle los servicios que se consideren necesarios, a través de módulos. Para ello, se puede obtener el código fuente de los servidores ya existentes y que se encuentren disponibles en Internet, tal como el anteriormente mencionado, luego compilarlo y modificarlo según conveniencias.

7.5 CUADRO DE RESUMEN INTERFACES WEB/DBMS

Tabla 1. Interfaces WEB/DBMS

DBMS	PLATAFORMA	
	UNIX	WINDOWS
Ingres	DbCGI	
MINISIS	WebQuery	Interfaz WWW MINISIS
Progress	DbCGI	
Micro CDS/ISIS	IsisWWW IQuery WWWIsis	IsisWWW IQuery WWWIsis
Access		Cold Fusion DbCGI db-Connector dbWeb Extensiones ODBC para Perl-Win32 Internet Database Connector WebDBC West Wind Web Connection X-Works

Btrieve		Cold Fusion DbCGI db-Connector dbWeb Extensiones ODBC para Perl-Win32 Internet Database Connector WebDBC
dBase		Cold Fusion DbCGI db-Connector dbWeb Extensiones ODBC para Perl-Win32 Internet Database Connector WebDBC
MiniSQL	Biblioteca API de MiniSQL W3-mSQL WDB	
FoxPro		Cold Fusion DbCGI db-Connector dbWeb Extensiones ODBC para Perl-Win32 Fox Web Internet Database Connector WebDBC West Wind Web Connection X-Works

Informix		Cold Fusion DbCGI db-Connector dbWeb Extensiones ODBC para Perl-Win32 Internet Database Connector WebDBC
Oracle	DbCGI db-Connector Oraperl WDB WebDBC	Cold Fusion DbCGI db-Connector dbWeb Extensiones ODBC para Perl-Win32 Internet Database Connector WebDBC X-Works
Paradox		Cold Fusion DbCGI db-Connector dbWeb Extensiones ODBC para Perl-Win32 Internet Database Connector WebDBC
SQL Server		Cold Fusion DbCGI db-Connector dbWeb Extensiones ODBC para Perl-Win32 Internet Database Connector WebDBC West Wind Web Connection X-Works

<p>Sybase</p>	<p>DbCGI WDB WebDBC Web.sql</p>	<p>Cold Fusion DbCGI db-Connector dbWeb Extensiones ODBC para Perl-Win32 Internet Database Connector WebDBC Web.sql X-Works</p>
----------------------	---	---

8. DATAWAREHOUSE

¹¹Es una colección de datos orientados por temas, integrados, variados en el tiempo y no volátiles, organizados para soportar la administración del proceso de toma de decisiones.

Data warehouse o bodega de datos es una colección de información corporativa derivada directamente de los sistemas operacionales (DB) y de algunos datos externos. Su propósito es soportar la toma de decisiones en un negocio (no las operaciones del negocio).

Repositorio completo de datos de la empresa, donde se almacenan datos estratégicos, tácticos y operativos, al objeto de obtener información táctica y estratégica.

Para construir una bodega de datos se necesitan herramientas para ayudar a la migración y a la transformación de los datos hacia la bodega de datos. Ya construida la bodega de datos se necesitan de herramientas para manejar grandes volúmenes de información.

Es una réplica de su información, estructurada de manera tal que se pueda analizar y visualizar. El DW le facilita al analista, manager o ejecutivo generar el conocimiento necesario para la toma de mejores y más rápidas decisiones.

¹¹ www.uca.edu.sv/investigacion/bdweb/intbdweb.html

Un sistema de DataWarehousing permite acceder y representar grandes volúmenes de datos. La manera más común de generar la información es a través de un sistema OLAP (*on-line analytical processing*). Los productos OLAP ofrecen una capacidad de análisis muy avanzada, tales como el multidimensional y el estadístico, análisis de *Data Mining* (*scoring* y detección de fraudes) y la creación de ficheros de Meta Información. Entre las muchas razones que motivan a las empresas a implementar DW se encuentran:

- ⌚ Realización de consultas a bases de datos y reportes desde servers/discos dedicados para esta tarea, sin que tomen recursos de servers/discos dedicados a transacciones comerciales.
- ⌚ Uso de una tecnología que acelere las consultas a las bases de datos y la extracción de reportes.
- ⌚ Obtención de información a partir de una multiplicidad de datos sin necesidad de poseer profundos conocimientos técnicos.
- ⌚ Proveer un repositorio de datos "limpios" contra los cuales es posible realizar reportes.

8.1 ¿CÓMO IMPLEMENTAR UN DATAWAREHOUSE CON ÉXITO?

Éstos son los principales temas sobre el desarrollo e implementación de proyectos de Datawarehouse, Sistemas para la Toma de Decisiones y Análisis de Información Multidimensional:

- ⌚ **Visión:** Obtenga una visión completa sobre la Organización, sus "puntos claves", el rol y perfil de los responsables de cada área involucrada en el proyecto, las fuentes de reportes y "juegue con las alternativas de datos".

- ④ **Descubrimiento:** Tome los "detalles ocultos" de los circuitos operativos del negocio, analice las ventajas funcionales de cada área, encuentre la proyección del negocio.
- ④ **Arquitectura:** Diseñe la opción tecnológica que ofrezca la mejor alternativa para la ecuación *costo final / tiempo de implementación / cantidad de usuarios / negocio a futuro*.
- ④ **Construcción:** Construya la solución de información para su empresa sobre los productos standard del mercado.
- ④ **Implementación:** Logre la participación de los principales actores ("Decisores") y la distribución de los beneficios del proyecto para el resto de la organización.
- ④ **Auditoria e Iteración:** Verifique el funcionamiento operativo post-producción, implante procedimientos para la actualización funcional de acuerdo a los cambios del negocio.

8.2 ¿CÓMO TRABAJA UNA BODEGA DE DATOS?

Las bodegas de datos son una base de datos históricos y operativos de la compañía que están disponibles para el usuario. Contrario a muchos sistemas, se establece de acuerdo con la lógica del negocio, mas que con la lógica de los sistemas. Le permite a los usuarios cavar y dar vueltas entre toda esa información importante de los clientes, para buscar relaciones y efectuar consultas.

El proceso mediante el cual los usuarios se sientan enfrente de una montaña de hechos y datos para descubrir tendencias que sugieran nuevas oportunidades de negocios se llama "minería de datos" (data mining).

La empresa debe comprometerse a mantener actualizados los datos que están alojados en la bodega, asegurarse que todos los datos son validos, exactos y oportunos.

8.3 ¿QUÉ HAY QUE HACER PARA CONFORMAR LA BODEGA DE DATOS?

Son tres los aspectos que intervienen en el establecimiento y el uso de la bodega de datos: el diseño, el mantenimiento, y el uso de la misma.

🕒 **El Diseño.** Requiere de personas capacitadas en la correcta estructuración de la bodega. De esta decisión depende la velocidad de búsqueda y la calidad y oportunidad obtenida en las respuestas a nuestras inquietudes. Una decisión mal tomada en este aspecto puede significar demoras de meses en vez de horas o minutos para la obtención de las respuestas requeridas.

🕒 **El Mantenimiento.** El mantenimiento de la información se convierte en parte fundamental, una vez que la bodega de datos forma parte integral de los negocios. El mayor problema que se vive con los proyectos de bodegas de datos es la obsolescencia de su información.

🕒 Se **actualiza** la información para el proyecto piloto, pero no se establecen mecanismos de actualización permanente que garantice la oportunidad siempre. Por último, pero no menos importante, está la capacidad para buscar relaciones y analizar la información. Si bien es cierto que la bodega de datos

agiliza esta tarea, es deber del "minero" saber cómo preguntar y cómo interpretar y poner en práctica los resultados que obtiene.

8.4 ¿CUÁL ES EL RETORNO DE LA INVERSIÓN?

Los beneficios y recompensas son abundantes para una compañía que configure y mantenga adecuadamente la bodega de datos. Ahorros en costos al igual que el aumento en ingresos encabeza la lista de los beneficios tangibles. Adiciónale a esto el análisis de las bases de datos de mercadeo para efectuar venta cruzada de productos, identificar y mantener a los clientes que generen la mayor utilidad, mientras se mantiene un mejor conocimiento de quienes son los clientes de la empresa.

Por ejemplo, una empresa de servicios telefónicos puede utilizar una bodega de datos para determinar cuales servicios pueden interesarle a cada uno de sus clientes. De esta manera evita gastos en correo masivo e impersonal. Una compañía no se debe olvidar que el objetivo de cualquier proyecto para una bodega de datos es reducir los costos operativos y generar ingresos.

Esto es una inversión y se debe poder esperar un retorno cuantificable a esta inversión en el tiempo. En conclusión, una bodega de datos bien implementada y mantenida redundará en beneficios para la compañía. Cualquier falla en su implementación, garantizará que el dinero invertido se ha despilfarrado.

8.5 ARQUITECTURA DATAWAREHOUSE

OLTP: Bases de datos transaccionales, propias o incorporadas.

OLAP: BBDD Datawarehouse de análisis.

PROCESOS DE CONSOLIDACIÓN

- ④ Cambio de tecnología BBDD.
- ④ Sumarizan datos disgregados.
- ④ Transforman datos.
- ④ Consolidan datos de aplicaciones no integradas.

CONSISTENCIA DE CONSOLIDACIÓN

- ④ Comprobar la validez de los datos en el entorno operacional: Datos que no se usan y datos que no se mantienen.
- ④ Inconsistencia entre distintas aplicaciones dentro del sistema: Datos no igualmente mantenidos y codificaciones diferentes.

MIDDLE-WARE

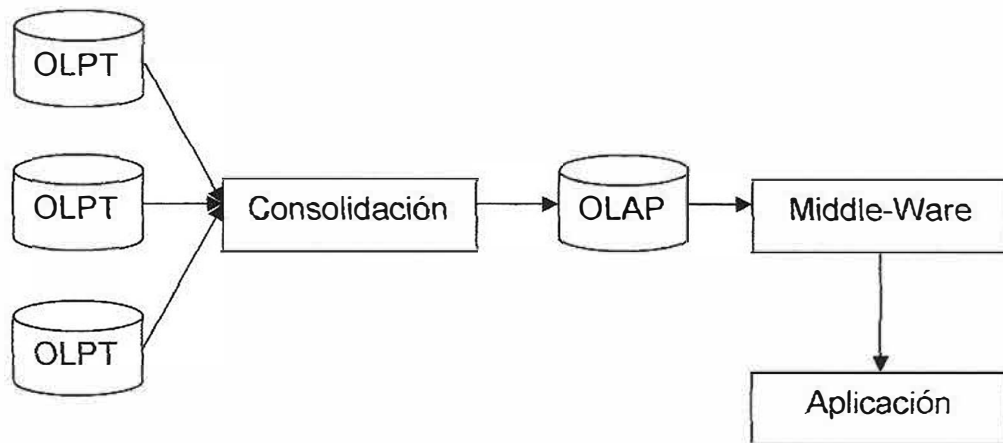
- ④ Gestiona comunicaciones con el datawarehouse.
- ④ Coordina la concurrencia.
- ④ Controla procesos batch.

APLICACIONES

- ④ Sistemas de presentación.
- ④ Sistemas interrogativos.
- ④ Sistemas de simulación.
- ④ Sistemas funcionales.
- ④ Sistemas expertos.
- ④ DSS

Figura 8. Arquitectura Datawarehouse.

Diagrama de Flujo de Datos



9. CRM- CUSTOMER RELATIONSHIP MANAGEMENT

El CRM es el conjunto de estrategias de ventas, marketing, comunicación y tecnología diseñadas con el propósito de establecer relaciones duraderas con todos los clientes, identificando y satisfaciendo sus necesidades.

Según Cesar Chiva de Agustín, "el CRM es una visión integral de la empresa sobre cómo debe relacionarse con los clientes, cuál es el canal que debe emplear, la herramienta tecnológica que debe utilizar para poder tener un trato masivo y simultáneo con cientos o miles de sus clientes. Asimismo el CRM "balancea" la organización empresarial hacia el cliente: cambia el foco desde la "operación" para centrarse en la figura del comprador de sus servicios".¹²

La gestión de clientes no es un descubrimiento de las nuevas teorías sobre Management empresarial. Sin embargo, en los últimos años se ha ampliado su peso como el verdadero motor del negocio de la empresa. La revolución tecnológica de los últimos años, el aumento de la competencia, la aparición de nuevos canales en la comunicación con los clientes (internet, correo electrónico, teléfono,...) ha provocado el aumento de esta visión empresarial.

El CRM presenta un cambio en el enfoque de la gestión de clientes al integrar dinámicamente las Ventas, el Marketing y el Servicio de Atención al cliente. Asimismo se integran en los sistemas de la empresa toda la información que

¹² CHIVA DE AGUSTÍN, César. Informe CRM. España: Febrero del 2003.

afecta al negocio: las bases de datos de clientes, los sistemas corporativos, los sistemas de soporte al negocio, etc.

9.1 IMPLANTACIÓN INTERNA

Algunas empresas con recursos (financieros y humanos) suficientes se plantean implantar desde cero una solución a medida que cumpla todas las necesidades de su gestión de clientes. Esta implantación suele tener un coste elevado: junto a la inversión en el Hardware es necesaria la adquisición de un software estándar, o herramienta de desarrollo en su defecto. Por último además del coste interno de personal dedicado de forma total o parcial a un proyecto de implantación CRM es necesario contratar externamente a una empresa -normalmente una consultora de servicios informáticos- que posea el conocimiento funcional y técnico de la solución a instalar. Uno de los factores a considerar asimismo es la integración del sistema de gestión de clientes con los sistemas preexistentes en la empresa. De nada serviría un sistema de interacción utilizado en un "contact-center" que no dispusiese de datos sobre la facturación, el cobro, status del pedido, recepción de quejas, etc.

9.2 ASP DE PRODUCTOS CRM

Los sistemas ASP proveen aplicaciones preempaquetadas de todo tipo, también CRM. Habitualmente siguen un régimen de alquiler de la solución que incluye el hardware, la licencia así como el mantenimiento, nuevas versiones, etc. Esta alternativa se utiliza en empresas que no tienen una definición de procesos demasiado cerrada y que necesita tener un sistema en un plazo inmediato.

9.3 OUTSOURCING DEL CONTACT CENTER

- Ⓜ Mayor eficiencia y control de costes (mano de obra)
- Ⓜ Disponen de base de clientes que les permiten ofrecer tecnología madura y comprobada
- Ⓜ Empresa cliente cede el control
- Ⓜ No ofrecen experiencia en consultoría e integración de sistemas

9.4 HOSTING DE SOLUCIONES CRM

- Ⓜ Administración y operación remota de la tecnología ofrecida por varios proveedores e integrada.
- Ⓜ Acceso rápido a una arquitectura preintegrada.
- Ⓜ Estructura de costes predecible.
- Ⓜ El proveedor de hosting debe tener experiencia en consultoría e integración de sistemas.

9.5 CARACTERÍSTICAS

- Ⓜ Fácil de implantar
- Ⓜ Rápida obtención de resultados (análisis del ROI)
- Ⓜ Flexible
- Ⓜ Grandes prestaciones
- Ⓜ Adaptable a cualquier negocio
- Ⓜ Progresivo
- Ⓜ Escalable
- Ⓜ Sistema del cliente, no del implantador

9.6 TIPOS DE CRM

9.6.1 CRM Analítico

- Herramienta para la explotación y análisis de la información sobre el cliente.
Business Intelligence:

1. DataWarehouse (almacén central de los datos de la empresa) y
2. DataMining (analiza información para descubrir tendencias, escenarios, etc.)

- Detección de patrones de comportamiento
- Permite diseñar acciones comerciales diferenciadas

9.6.2 CRM Operacional

- Responsable de la gestión de las diferentes funciones de ventas, marketing y servicio al cliente y de su integración con sistemas existentes.
- No hay compartimentos estanco con información no compartida

9.6.3 CRM Colaborativo

Se gestionan los diferentes canales de relación con los clientes:

- FRONT OFFICE
- Web
- E-mail
- Fax
- Teléfono
- Interacción directa

9.7 FUNCIONALIDAD CRM

15 parámetros que teóricamente debe proporcionar una solución CRM completa:

1. **Gestión de oportunidades:** OMS (Opportunity Management Service).
2. **Sistema de configuración de ventas:** Permite la configuración de materiales, precios, promociones, servicios, opciones de financiación y marketing.
3. **Partner Relationship Management**
4. **Sistema de Venta Interactiva:** Capacidad de establecer relaciones comerciales sin necesidad de un operador. Incluye técnicas como cara-a-cara, web colaborativa y cliente self-service,
5. **Compensación e Incentivos:** Corresponde a la capacidad de gestionar los incentivos, planes de compensación y cuotas para transacciones susceptibles de generar comisiones de venta. Implica la existencia de herramientas de reporting y de utilidades de simulación de ventas.
6. **Gestión de contenidos:** Esta herramienta permite el acceso a los contenidos de cualquier naturaleza (textos, gráficos, animaciones, vídeos,...)
7. **E-service:** Corresponde a herramientas que permiten a clientes, partners y clientes potenciales para realizar consultas self-service e interacciones con la empresa vía internet, intranet o extranets.
8. **Gestión de llamadas:** Funcionalidad central de las aplicaciones de CSS (Customer Service Support). Se usa para llamadas y transacciones entrantes, y para la gestión de las transacciones del inicio al fin de la interacción.
9. **Field Service:** Incluye herramientas de previsiones de ventas, gestión de contratos, garantías, servicio de componentes, gestión de infraestructura, inventario, seguimiento de defectos (gestión de calidad) y reporting.

10. **Personalización:** Capacidad de personalizar el entorno de contacto del cliente en función de múltiples parámetros (preferencias, datos relevantes de cliente, comportamientos, etc).
11. **Software analítico:** Capacidad de elaborar estadísticas e informes.
12. **Gestión de Marketing:** Gestión de campañas, generación de perfiles, target groups y capacidades de segmentación.
13. **Escalabilidad**
14. **Parametrización**
15. **Rapidez de implantación**

9.8 ANÁLISIS DE LA INTEGRACIÓN

Las compañías estarán obligadas a invertir una cantidad significativa de recursos para enlazar los diferentes módulos de su sistema, o construir funcionalidades adicionales para soluciones puntuales con el fin que los procesos CRM puedan quedar consistentes con todas las aplicaciones. Los costes de consultoría pueden superar ampliamente los costes de las licencias de software.

META GROUP estima que en los proyectos CRM los costes de integración llegan a cerca del 60% de media sobre el coste total de implantación.

Claramente, estos costes aumentarán dependiendo de la selección del paquete y de los requerimientos de implantación. De esta manera cuando el coste total de propiedad (TCO) se incrementa, el ROI se decrementa.

9.9 EVOLUCIÓN CRONOLÓGICA DE UN SISTEMA INTEGRADO CON CRM Y DATAWAREHOUSE

Aunque no todos los modelos de empresa requieren todos los componentes de cada evolución, sí requieren al menos algunos de ellos. En cualquier caso Data Warehouse es la herramienta de reporting multidimensional de las plataformas presentes y futuras.

9.9.1 Fase I. Core Del Negocio

Implantación de los sistemas de operatividad del negocio de la compañía en su BACK-OFFICE.

9.9.2 Fase II. Gestión De Clientes

Implantación de la solución CRM

9.9.3 Fase III. Data Warehouse

Implantación de Business Warehouse: Reporting, análisis complejos, análisis de incidencias, análisis de cliente, marketing (campañas, segmentaciones, etc.)

9.9.4 Fase IV

Nuevas herramientas de gestión empresarial de reporting y análisis adaptadas a las necesidades futuras de la empresa

10. PRM (PARTNER RELATIONSHIP MANAGEMENT)

La estrategia de negocio Partner Relationship Management (Manejo de Relaciones con Aliados de negocio), denominado por el acrónimo PRM, involucra de manera homogénea a cada componente empresarial. Muchos de los expertos en el tema incluyen al PRM dentro como un componente fundamental de una estrategia de CRM, específicamente en el área de backend, que incluye todos los procesos que no están a la vista del cliente pero que forman un papel determinante en su nivel de satisfacción y atención total.

Como buena referencia podemos tomar la definición del "padre" de PRM, Robert Thompson, fundador y presidente de Front Line Solutions Inc., empresa especializada en el desarrollo de sistemas centrados en PRM. "PRM es una estrategia de negocio para seleccionar y manejar a los compañeros de negocio para optimizar las relaciones de largo término con la empresa. Esto significa seleccionar a los mejores partners trabajar con ellos para ayudarles a atender a los clientes mutuos (clientes finales) y asegurarse que los partners y dichos clientes sean entes satisfechos."¹³

Argumentando a Robert Thompson, el PRM es un modelo estratégico de manejo de procesos con los diferentes "compañeros" de negocio. Desglosando un poco el origen del PRM (el CRM) podemos asegurar que, al igual que CRM, PRM no es una implantación de un sistema informático. Es una realidad que tiene soporte en

¹³ GREENBERG, Paul. CRM at the Speed of Light. McGraw Hill, p. 149

un sistema óptimo de información, pero no debe ser el punto central de la implantación.

Por medio del PRM, las empresas pueden "habilitar sus business partners" con las siguientes opciones:

- a. Acceso a documentación online
- b. Datos de clientes (historial, detalle de transacciones, cuenta personal, etc)
- c. Sistema de manejo de inventarios
- d. Ordenes de pedidos
- e. Facturaciones y cuentas por cobrar

Se puede partir de las siguientes tres premisas para gestionar un proyecto PRM:

1. **Análisis de sensibilidad financiero.** A través de este estudio se podrán modificar las variables de la cuenta de resultados y se puede observar cuáles serían los beneficios.
2. **Estudio de viabilidad informática.** Esto representa analizar cómo sería el proceso de homologación de sistemas. Es muy común que diferentes empresas cuenten con múltiples paquetes de gestión (SAP, Siebel, PeopleSoft, etc.) para gestionar sus procesos internos. Los líderes de proyecto deben establecer ante qué sistema se estará trabajando y cuales serían los costes de integración (que a final de cuentas siempre es lo más caro en la implantación).
3. **Marketing ante los BP's.** Finalmente los Business Partners serán aquellos que se enfrenten día a día con el sistema y que ingresen información. Se debe convencer a dichos usuarios (intermediarios) que los beneficios pueden resultar impensables, sobre todo si hablamos del factor crítico de todo negocio: fidelización de clientes

Evidentemente en los puntos anteriores no se encuentra todo el proceso para la implementación de un PRM, pero sí resultan directrices importantes para la gestión de dicho proyecto.

Finalmente, y a manera de conclusión, debemos destacar que cada sistema PRM puede tener variados objetivos, sin embargo uno de los puntos de mayor relevancia debe ser la expansión de mercado a través de la homologación de atención y servicio al cliente.

Si se decide a comenzar la aventura de generar mayor rentabilidad de sus clientes finales a través de sus business partners, los beneficios del proyecto pueden resultar de gran valor para la empresa.

11. BASES DE DATOS CENTRALIZADAS

11.1 GENERALIDADES

Una base de datos centralizada es aquella donde todos los recursos de la base de datos se encuentran en un servidor y para acceder a la información es necesario estar conectado por algún tipo de red que permita por medio de una aplicación y desde otra computadora de menos capacidad realizar operaciones centralizadas (el servidor proporciona todos los recursos necesarios para el procesamiento de la información).

Un sistema de bases de datos centralizado (SBDC) es un sistema en el cual múltiples usuarios pueden manipular datos, pero por otra parte los datos que están manipulando se encuentran exactamente en la computadora central, lo que ocurre es que por medio de una aplicación o módulo el usuario puede obtener la información que se encuentra almacenada en la base de datos central o que se encuentra en el servidor.

Esto viene a hacer que las tablas y todas las aplicaciones de la base de datos están almacenados en el computador central (servidor), que es el que brinda por medio del sistema operativo que el usuario piense que está manipulando la información desde su sitio de trabajo, lo que él no sabe que lo que está utilizando es una aplicación o módulo que permite la manipulación de la información.

11.2 CARACTERÍSTICAS

- ⌚ **Autonomía central:** Las operaciones se realizan directamente en la base de datos, por medio de aplicaciones o módulos que son controladas por el servidor.
- ⌚ **Dependencia con respecto a la localización:** Debe ser necesario que los usuarios sepan donde están almacenados físicamente los datos, para que el usuario tenga conocimiento de que está utilizando una aplicación.
- ⌚ **Dependencia con respecto a la fragmentación:** La fragmentación es indeseable por razones de desempeño, los datos, deben almacenarse en la computadora central, aunque esto ocasione mayores tráficos en la red.
- ⌚ **Dependencia de replicas:** La información solo se almacena en un sitio específico, por lo que es más sencillo detectar fallos.

11.3 VENTAJAS

- ⌚ La principal ventaja se refiere al control y manejo de los datos, dado que estos residen en un solo sitio lo que permite un control más estrictos de estos en cuanto a seguridad se refiere.
- ⌚ La información se encuentra localizada en un solo sitio, lo que permite mayores posibilidades de detectar fallos.
- ⌚ Los datos no están replicados, el control de concurrencia y los mecanismos de recuperación son mucho más sencillos que en un sistema distribuido.
- ⌚ La centralización produce una disminución en la complejidad del diseño y en la implementación del sistema.

11.4 DESVENTAJAS

- ⓐ Los datos se localizan en un lugar remoto, por tanto, el acceso es mas lento; el procedimiento es lento debido que un solo equipo interviene en el procedimiento de una carga de trabajo.
- ⓐ Los costos de operación son muy altos, es difícil incorporar nuevas estaciones de trabajo.
- ⓐ Los datos permanecen en un solo sitio, esto no permite un control por parte de los usuarios que están conectados en la red.
- ⓐ No existe autonomía por parte de los usuarios, el administrador tiene el control de todos los recursos del sistema.
- ⓐ Al momento de acceder la información el nivel de concurrencia es bastante alto, ya que la información se encuentra en un solo sitio.

12. S.D.K SOFTWARE DEVELOPER KIT

Es una herramienta de desarrollo de software que permite a los programadores expandir la forma de crear aplicaciones de una manera sencilla, rápida y con facilidad para conectarse a diferentes clases de bases de datos.

El kit de desarrollo de software le dará a las instituciones académicas y a los aficionados a la informática las herramientas necesarias para expandir la plataforma del sistema de almacenamiento de información.

Se puede crear y compartir programas nuevos y datos en un entorno comercial dentro de la comunidad de programadores, lo que acelerará el desarrollo de las diferentes aplicaciones.

13. SOFTWARE DE FUENTES PÚBLICAS (OPEN SOURCE SOFTWARE)

Este término Open Source Software describe el software que está disponible en forma de fuentes para quien desee usarlo. La actividad de compartir los resultados de trabajo en el computador siempre ha estado presente en la subcultura que rodea estas máquinas, particularmente en el ámbito académico. Recientemente esta actividad ha adquirido popularidad.

13.1 CARACTERÍSTICAS

- ④ Libre redistribución
- ④ Código fuente disponible
- ④ Permitir obras derivadas
- ④ Restricciones sobre versiones modificadas
- ④ No discriminación de personas o grupos
- ④ No discriminación de uso
- ④ Distribución de la licencia
- ④ No específica a formar parte de un producto
- ④ No restringir software distribuido con el producto
- ④ No restringir tecnología de distribución o instalación

13.2 TIPOS DE LICENCIAS DE FUENTE PÚBLICA

Muchas veces se confunde el software de fuente pública con software de dominio público. Sin embargo, hay diferencias entre las opciones que se han usado para distribuirlo. Todas ellas se basan en las leyes y convenciones internacionales de derecho de autor, que especifican ciertos derechos del autor (o del dueño de los derechos) de una obra literaria o artística (lo que definitivamente es una curiosa categoría bajo la cual clasificar software). En síntesis, esta legislación reserva para el autor (o dueño de los derechos de autor) el uso exclusivo de la obra del caso. El dueño de los derechos puede **licenciar** el uso (es decir, comprometerse a no llevar a los tribunales al usuario) bajo ciertas condiciones. El caso típico es que el usuario deba pagar por el derecho de uso del software, posiblemente con ciertas restricciones como usarlo en una sola máquina, no sacar copias, uso educacional únicamente, entre otras. De esto vive la industria tradicional de software.

Sin embargo, desde los inicios de la computación han existido grupos que intercambiaban software, aprendiendo los unos de los otros y dejando un legado creciente creado en forma conjunta. Inicialmente se crearon partiendo de grupos de usuarios de las distintas plataformas, como fue el caso de Decus, grupo de usuarios de minicomputadores de DEC. Con la aparición de las redes, y luego de Internet estos grupos aislados comenzaron a establecer contactos mas fluidos. Ayudó en esto la aparición de Unix, y luego los estándares POSIX asociados, que permitieron trasladar programas fuentes de una máquina a otra. Aparecieron así variadas opciones para distribuir software. Todas las opciones de distribución incluyen cláusulas en el sentido que el software se distribuye sin garantías de ninguna especie. Entre las condiciones más importantes se cuentan:

- **Software de dominio público:** El público en general es dueño de los derechos. En la práctica, cualquier persona puede tomar software de dominio público y hacer con él lo que desee: Usarlo para crear nuevo software o distribuirlo a su antojo, entre otras.
- **Licencia BSD: (Berkeley Software Distribution).** Particularmente se destacan las modificaciones que se introdujeron al sistema Unix en el curso de una serie de proyectos de investigación. Básicamente deja al receptor del software plena libertad en su uso, para cualquier fin, siempre que mantenga las notas de copyright de los autores.
- **GNU General Public Licence (GPL):** La GPL especifica que toda modificación que se haga a software bajo GPL debe entregarse con al menos la opción de obtener fuentes completas a quien reciba el software. Además, queda prohibido imponer restricciones adicionales a la distribución de parte de éste. Toda modificación a partir del software de GNU estará siempre disponible para todos.

En realidad, hay una gran variedad de licencias de fuente realmente pública, pero en general se pueden asimilar a una de estas tres grandes categorías. Lo que tienen en común es no impedir y fomentar la distribución de los fuentes del software. Otras licencias reservan los derechos de todo software desarrollado a partir de las fuentes entregadas al dueño de los derechos.

13.3 DESARROLLO DE SOFTWARE DE FUENTES PÚBLICAS

Eric S. Raymond en su serie de artículos clásicos *The Cathedral and the Bazaar* (La Catedral y el Bazar), compara el esquema de desarrollo de software en ambiente cerrado con una catedral y el desarrollo abierto, en el cual cualquiera

puede participar, con un bazar, en el cual describe cómo y porqué funciona el sistema de desarrollo que instituyó Linus Torvalds, que consiste en distribuir, delegar y desarrollar software abierto.

Sus conclusiones en general apuntan a que el participar en un proyecto abierto, en el cual se puede ganar el reconocimiento público de sus pares en todo el mundo es un fuerte incentivo. El que hayan varias personas trabajando simultáneamente sobre un mismo tema sirve para explorar diversas alternativas, eligiendo la que da mejores resultados en la práctica una vez que estén relativamente completas.

Estas conclusiones llevaron a que Netscape anunciara públicamente a comienzos de 1998 que abriría el código de Netscape Navigator. Ese mismo año, IBM anunció que soportaría Apache como servidor web como parte de WebSphere. Desde hacía tiempo, empresas como Digital (hoy parte de Compaq) y Sun han estado apoyando al desarrollo de Linux en forma más o menos abierta. Más recientemente Apple, IBM, SGI, y Hewlett-Packard han comenzado a portar Linux a sus máquinas. El caso de SGI es importante, existen indicios de que estudian seriamente migrar sus ofertas en el ámbito Unix a Linux, y en particular apoyar fuertemente el desarrollo de Linux para sus supercomputadores. Para estas empresas eso significa contar con un rango mucho mayor de desarrolladores talentosos, y les descarga parte del costo de desarrollo de software, que en sí no les produce beneficios (su producto es, finalmente, el hardware). Hoy en día, tanto IBM como SGI y Compaq tienen proyectos importantes sobre Linux, con apoyo oficial de la empresa. Por otro lado, Sun adquirió la empresa Star Division, quienes desarrollaron StarOffice, el producto de oficina más popular en Linux en este momento. Inicialmente distribuían StarOffice gratuitamente, para liberar el código fuente bajo GPL el 13 de octubre del 2000.

Otro ejemplo es el caso del desarrollo de la colección de compiladores del proyecto GNU, conocidos colectivamente como GCC. Originalmente, este proyecto estaba férreamente controlado, con versiones liberadas muy a lo lejos.

En 1997 la empresa Cygnus, cuyo principal negocio consiste en portar este compilador a nuevas CPUs y que es uno de los principales proveedores de herramientas para desarrollo de aplicaciones empujadas, lanzó el proyecto **Egcs** para continuar el desarrollo (que a la sazón estaba claramente estancado) en el espíritu del proyecto Linux. Desde entonces, con versiones casi semanales de prueba, el desarrollo se ha acelerado enormemente. Más recientemente a los interesados se les sugiere sincronizar su copia de los fuentes usando CVS (un sistema de control de versiones distribuido) directamente a través de la red desde el repositorio central. Muchos otros proyectos están haciendo esto, entre los más importantes se cuenta el paquete SAMBA, que permite a una máquina Unix actuar como servidor y cliente en una red Windows. Incluso han aparecido sitios que mantienen los repositorios de fuentes de varios paquetes, el más importante de éstos es SourceForge.

13.4 EMPRESAS ALREDEDOR DE LINUX

Durante 1999 se produjo una verdadera locura en las bolsas por la alta demanda de las acciones de empresas relacionadas con Linux. Un análisis calmado muestra que este efecto fue en parte resultado de una estampida de parte de los inversionistas por no quedarse atrás en la siguiente ola de innovaciones tecnológicas, y que los precios siderales que alcanzaron algunas acciones simplemente no tenían justificación racional. Recientemente muchas de las acciones de empresas en el rubro Internet sufrieron fuertes caídas, pero las empresas alrededor de Linux en general se han mantenido.

Linux (y en términos más generales, software de libre distribución) es un negocio, incluso un buen negocio. Las distribuciones de Linux más populares (entre las que se cuentan Red Hat, SuSE, Mandrake, y Conectiva) son mantenidas por compañías que han experimentado una expansión fenomenal. Y estas compañías

prosperan a pesar que es perfectamente posible obtener copias de su productos bajándolas a través de la red, sin pagarles un centavo (incluso les significa un costo el ponerlos a la disposición del público). Esto se compensa con que es mucho más conveniente para el usuario comprar el paquete, que incluye manuales impresos y posiblemente otras ventajas, e instalar de allí. Algunos simplemente adquieren el paquete con la intención de retribuir en parte el esfuerzo que ha hecho la empresa. Estas empresas por lo demás ofrecen otros servicios, como soporte técnico y capacitación; y últimamente se ha puesto de moda ofrecer cursos y pruebas de certificación. A cambio de abrir sus productos (incluso a la competencia), ganan dado que los usuarios, reportan problemas, y, como tienen acceso al código fuente, hasta pueden ofrecer correcciones.

13.5 SEGURIDAD

Los terminos “libre y publico” son interpretados por la comunidad como inseguridad. El razonamiento es que los crackers, al tener acceso a los fuentes, pueden encontrar debilidades con mayor facilidad. Sin embargo, ese mismo acceso permite que uno tenga la posibilidad de verificar que no hay debilidades ocultas. Es poca la gente suficientemente paranoica como para hacer estas verificaciones por sí mismas (o contratar a alguien para que las haga por ellos), pero al existir la posibilidad de hacerlo hay personas que lo harán. Claro que no hay garantías de que este proceso lleve a descubrir los problemas, el que ande buscando modificar un programa para ajustarlo a su conveniencia revisará sólo lo necesario para sus fines, y no necesariamente se fijará en los problema de seguridad. También es posible que los que descubran los problemas sean malintencionados.

Hay herramientas que permiten detectar posibles problemas de seguridad en forma automática, como por ejemplo ITS4. Estas herramientas analizan el código fuente de una aplicación para detectar errores comunes. Hay bastantes

interesados en mejorar la seguridad del sistema, entre ellos en primera línea los proveedores del mismo.

Es una de las verdades comúnmente aceptadas en la comunidad de seguridad que lo que llaman **seguridad a través de la oscuridad** (o sea, tratar de hacer seguro un sistema ocultando su funcionamiento) es inútil: Tarde o temprano el secreto se revelará, y la seguridad desaparece con él. Es así como recomiendan confiar sólo en sistemas que han sido sometidos a escrutinio y análisis públicos.

14. MYSQL

14.1 BREVE HISTORIA DE MYSQL

MySQL surgió como un intento de conectar el gestor mSQL a las tablas propias de MySQL AB, usando sus propias rutinas a bajo nivel. Tras unas primeras pruebas, vieron que mSQL no era lo bastante flexible para lo que necesitaban, por lo que tuvieron que desarrollar nuevas funciones. Esto resultó en una interfaz SQL a su base de datos, con una interfaz totalmente compatible a mSQL.

Se comenta en el manual que no se sabe con certeza de donde proviene su nombre. Por un lado dicen que sus librerías han llevado el prefijo 'my' durante los diez últimos años. Por otro lado, la hija de uno de los desarrolladores se llama My. No saben cuál de estas dos causas, han dado lugar al nombre de este conocido gestor de bases de datos.

14.2 ¿QUÉ ES MYSQL?

MySQL es un sistema de gestión de bases de datos no relacional muy rápido, licenciado bajo la GPL de la GNU. Su diseño multihilo le permite soportar una gran carga de forma muy eficiente. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca.

MySQL es un servidor de bases de datos multiusuario y multihilo, usado sobre todo en Internet en conjunción con PHP. Con solo pocas líneas de código PHP se pueden administrar datos guardados en una base MySQL, agregando, eliminando o actualizando campos y registros.

Aunque MySQL es software libre (licencia GPL) y es mantenido por la compañía sueca MySQL AB, MySQL AB distribuye una versión comercial de MySQL, que no se diferencia de la versión libre más que en el soporte técnico que se ofrece y la posibilidad de integrar este gestor en un software propietario, ya que de no ser así, se vulneraría la licencia GPL.

GNU General Public Licence (GPL): La GPL especifica que toda modificación que se haga a software bajo GPL debe entregarse con al menos la opción de obtener fuentes completas a quien reciba el software. Además, queda prohibido imponer restricciones adicionales a la distribución de parte de éste. Toda modificación a partir del software de GNU estará siempre disponible para todos.

Este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debido a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración.

Esta base de datos sueca, es seguramente la base de datos para *Linux* más popular de todas. Eso lo demuestra que viene incluida en casi todas las distribuciones de *Linux*. Además está disponible para casi todas las plataformas hardware y sistemas operativo (incluidos *Windows NT/2000, 98/95/ME*).

MySQL es conocida sobretodo por su velocidad y escasos recursos que consume. Es muy fácil de instalar y administrar y tiene una gran comunidad de usuarios. Buscar información para realizar cualquier cosa con *MySQL* es fácil. Es muy indicada para iniciarse en el mundo de las bases de datos, puesto que dispone de una infinidad de utilidades, tutoriales y documentación que la inmensa comunidad de usuarios de *MySQL* se ha encargado de realizar desinteresadamente.

El gran propulsor de la base de datos *MySQL* ha sido sin duda el lenguaje interpretado para *Web PHP*. *MySQL* ha crecido al mismo paso que ha crecido la comunidad de *PHP*. Cuando llegó la popularización del *Linux* y de *PHP* como plataforma para aplicaciones *Web* de bajo costo se encontró que la única base de datos gratuita decente del momento era *MySQL*. Hoy muchas *Webs* populares usan *MySQL*, como *Slashdot* o *Freshmeat*. Tiene también contribuciones para añadir soporte de *Full-text-indexing* en las tablas, un valor añadido importante hoy en día.

Las propiedades *ACID* de las *RDBMS*, *MySQL* no las cumple. No tiene integridad referencial, con lo que la base de datos puede llegar a tener datos inconsistentes (por ejemplo empleados pertenecientes a departamentos inexistentes). En el *CREATE TABLE* aunque le pongas *FOREIGN KEY*, lo ignorará por completo. Carece también de transacciones, *triggers* o procedimientos almacenados.

De hecho el *MySQL* no es una *RDBMS*, sino un sistema de ficheros con una interfaz *SQL*. Realiza los bloqueos de escritura al nivel de tabla, castigando la concurrencia de inserciones en la base de datos.

MySQL guarda los *Blobs* (binarios) en la misma tabla que guarda los datos normales, o sea como un campo más en la misma fila. No es el mejor método para guardar un binario, ya que hace caer el rendimiento del acceso a los otros datos.



14.3 PRINCIPALES CARACTERÍSTICAS

Las principales características de este gestor de bases de datos son las siguientes:

- ⌚ Mayor velocidad tanto al conectar con el servidor como al servir selects y demás, como a la hora de realizar las operaciones, lo que lo hace uno de los gestores que ofrecen mayor rendimiento y robustez.
- ⌚ Escrito en C y C++. Usa GNU autoconf para portabilidad.
- ⌚ Gran portabilidad entre sistemas.
- ⌚ Clientes C, C++, Java, Perl, TCL, etc
- ⌚ Multiproceso, es decir puede usar varias CPU si éstas están disponibles.
- ⌚ Puede trabajar en distintas plataformas y S.O. distintos.
- ⌚ Todas las claves viajan encriptadas en la red.
- ⌚ Registros de longitud fija y variable.
- ⌚ Todas las columnas pueden tener valores por defecto.
- ⌚ Utilidad (Isamchk) para chequear, optimizar y reparar tablas.
- ⌚ Todos los datos están grabados en formato ISO8859_1.
- ⌚ Los clientes usan TCP o UNIX Socket para conectarse al servidor.
- ⌚ El servidor soporta mensajes de error en distintas lenguas.
- ⌚ Todos los comandos tienen help o -? para las ayudas.
- ⌚ Soporta gran cantidad de tipos de datos para las columnas como enteros de 1, 2, 3, 4, y 8 bytes, coma flotante, doble precisión, carácter, fechas, enumerados, etc.
- ⌚ Según benchmarks disponibles en Internet, hasta 80 veces más rápida que Oracle en las mismas condiciones.
- ⌚ Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, etc).
- ⌚ Soporta hasta 32 índices por tabla.

- ④ Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- ④ Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
- ④ Las utilidades de administración de este gestor son envidiables para muchos de los gestores comerciales existentes, debido a su gran facilidad de configuración e instalación.
- ④ Tiene una probabilidad muy reducida de corromper los datos, incluso en los casos en los que los errores no se produzcan en el propio gestor, sino en el sistema en el que está.
- ④ El conjunto de aplicaciones Apache-PHP-MySQL es uno de los más utilizados en Internet en servicios de foro (Barrapunto.com) y de buscadores de aplicaciones (Freshmeat.net).
- ④ Licencia GPL a partir de la versión 3.23.19.
- ④ Mayor rendimiento.
- ④ Mejores utilidades de administración (backup, recuperación de errores, etc).
- ④ Aunque se cuelgue, no suele perder información ni corromper los datos.
- ④ Mejor integración con PHP.
- ④ No hay límites en el tamaño de los registros.
- ④ Mejor control de acceso, en el sentido de qué usuarios tienen acceso a qué tablas y con qué permisos.
- ④ MySQL se comporta mejor que Postgres a la hora de modificar o añadir campos a una tabla "en caliente".
- ④ MySQL es muy utilizada en servidores, como complemento a PHP, otro lenguaje de código abierto. Y, por supuesto, muy conocida en el mundo Linux, entre programadores "liberales".
- ④ Se basa en el lenguaje SQL (Structured Query Language, Lenguaje Estructurado de Consulta).
- ④ Es compatible con muchos lenguajes, aunque el dúo perfecto lo forma con PHP.

- ⌚ Ideal para bases de datos en Internet.
- ⌚ Rápida, fiable, segura.

14.3.1 Características (version 4.0 en adelante)

Inicialmente, MySQL carecía de elementos considerados esenciales en las bases de datos relacionales, tales como integridad referencial y transacciones. A pesar de ello, atrajo a los desarrolladores de páginas Web con contenido dinámico, justamente por su simplicidad; aquellos elementos faltantes fueron llenados por la vía de las aplicaciones que la utilizan.

Poco a poco los elementos faltantes en MySQL están siendo incorporados tanto por desarrollos internos, como por desarrolladores de software libre. Entre las características disponibles en las últimas versiones se puede destacar:

- ⌚ Amplio subconjunto del lenguaje **SQL**. Algunas extensiones son incluidas igualmente.
- ⌚ Disponibilidad en gran cantidad de plataformas y sistemas.
- ⌚ Diferentes opciones de almacenamiento según si se desea velocidad en las operaciones o el mayor número de operaciones disponibles.
- ⌚ Transacciones y claves foráneas.
- ⌚ Conectividad segura.
- ⌚ Replicación.
- ⌚ Búsqueda e indexación de campos de texto.
- ⌚ Se ha agregado InnoDB a la lista de tipos de tablas soportados en una instalación típica, o estándar.

La licencia GPL de MySQL obliga a distribuir cualquier producto derivado (aplicación) bajo esa misma licencia. Si un desarrollador desea incorporar MySQL

en su producto pero no desea distribuirlo bajo licencia GPL, puede adquirir la licencia comercial de MySQL que le permite hacer justamente eso.

14.3.2 Mejoras desde la version 3.23

Comenzando por el “**optimizador de consultas**”, ahora es más inteligente en el uso de índices para resolver las posibles dudas que puedan surgir mientras se utiliza.

En 4.0, las opciones de índices de texto completo han sido movidas al archivo de configuración de MySQL, así que solamente se tienen que hacer las adecuaciones necesarias y reiniciar MySQL para que los cambios tengan efecto. Muchos fallos en las búsquedas de texto completo han sido corregidos también.

Cambios al código de la caché de claves han producido un significativo aumento en el tiempo de ejecución durante algunas consultas basadas en índices. Esto es especialmente útil en servidores que tienen demasiada carga.

Para eliminar registros relacionados de múltiples tablas al mismo tiempo, ahora MySQL 4.0 dispone de **borrados multi-tablas**. Al especificar múltiples tablas y la cláusula WHERE correcta, MySQL hará sin problemas lo que esperas. También se pueden agregar opciones **ORDER BY** y **LIMIT** a las consultas DELETE, para obtener un mejor control sobre cuántos registros son eliminados y el orden en el que son eliminados dichos registros.

El sistema de replicación de MySQL es multi-hilo en los servidores esclavos. Si el servidor principal llega a fallar, es ahora mucho más probable que cada esclavo tenga los datos necesarios para hacer por sí mismo una recuperación de los datos y trabajar como si fuera el servidor maestro. Los registros de replicación ahora

contienen los marcadores de transacción necesarios para asegurarse que las transacciones son replicadas apropiadamente. Muchos de estos cambios fueron realizados antes del sistema de replicación **fail-safe**.

14.3.3 InnoDB: Transacciones ACID y más

La modularidad proporcionada por MySQL es una ventaja muy importante sobre muchos otros sistemas de administración de bases de datos: en concreto, se puede elegir el tipo de una tabla al momento mismo de crearla.

En la versión 4.0 MySQL ha agregado **InnoDB** a la lista de tipos de tablas soportados en una instalación típica, o estándar. MySQL 4.0 soporta cinco tipos de tablas: **MyISAM**, **ISAM**, **HEAP**, **BDB (Base de datos Berkeley)**, e **InnoDB**. BDB e InnoDB son ambos tipos de tablas transaccionales. Se puede utilizar la sentencia estándar **BEGIN WORK** seguida de varias consultas y finalizar con un **COMMIT** o **ROLLBACK** para completar la transacción. O, se pueden correr en modo **AUTOCOMMIT**, así que cada consulta es efectivamente una transacción separada.

InnoDB es un motor de bases de datos muy completo que ha sido embebido dentro de MySQL. InnoDB proporciona las siguientes características:

- ⌚ Recuperación automática ante fallas. Si MySQL se da de baja de una forma anormal, InnoDB automáticamente completará las transacciones que quedaron incompletas.
- ⌚ **Integridad referencial**. Ahora se pueden definir llaves foráneas entre tablas InnoDB relacionadas para asegurarse de que un registro no puede ser eliminado de una tabla si aún está siendo referenciado por otra tabla.

- Ⓢ **Bloqueo a nivel de filas.** Al usar tablas MyISAM, y tener consultas muy grandes que requieren de mucho tiempo, simplemente no se podían ejecutar más consultas hasta que terminarían las consultas que estaban en ejecución. En cambio, las tablas InnoDB usan bloqueo a nivel de filas para mejorar de manera impresionante el rendimiento.
- Ⓢ **SELECTs sin bloqueo.** Como si el bloqueo a nivel de filas no fuera suficiente, el motor InnoDB usa una técnica conocida como **multi-versioning** (similar a PostgreSQL) que elimina la necesidad de hacer bloqueos en consultas SELECT muy simples. Ya no será necesario molestarse porque una simple consulta de sólo lectura está siendo bloqueada por otra consulta que está haciendo cambios en una misma tabla.
- Ⓢ Crear tablas InnoDB es tan simple como crear otro tipo de tablas. Sólo se tiene que especificar **Type = InnoDB** al final de la sentencia CREATE TABLE. Ejemplo:

```
CREATE TABLE algunaTabla(  
    campoX  INTEGER UNSIGNED AUTO_INCREMENT PRIMARY KEY,  
    campoY  INTEGER NOT NULL,  
    campoZ  VARCHAR(255) NOT NULL  
    ) Type = InnoDB;
```

14.4 ¿QUÉ ES LO QUE LE FALTA?

MySQL surgió como una necesidad de un grupo de personas sobre un gestor de bases de datos rápido, por lo que sus desarrolladores fueron implementando únicamente lo que precisaban, intentando hacerlo funcionar de forma óptima. Es por ello que, aunque MySQL se incluye en el grupo de sistemas de bases de datos relacionales, carece de algunas de sus principales características:

- ⌚ Subconsultas: tal vez ésta sea una de las características que más se echan en falta, aunque gran parte de las veces que se necesitan, es posible reescribirlas de manera que no sean necesarias.
- ⌚ SELECT INTO TABLE: Esta característica propia de Oracle, todavía no está implementada.
- ⌚ Triggers y Procedures: Se tiene pensado incluir el uso de procedures almacenados en la base de datos, pero no el de triggers, ya que los triggers reducen de forma significativa el rendimiento de la base de datos, incluso en aquellas consultas que no los activan.
- ⌚ Transacciones: en las últimas versiones ya hay soporte para transacciones, aunque no por default si no que hay que activarlo de una forma especial.
- ⌚ No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.

Los desarrolladores comentan en la documentación que todas estas carencias no les resultan un problema, ya que era lo que ellos necesitaban. De hecho, MySQL fue diseñada con estas características, debido a que lo que buscaban era un gestor de bases de datos con una gran rapidez de respuesta. Pero ha sido con la distribución de MySQL por Internet, cuando más y más gente les están pidiendo estas funcionalidades, por lo que serán incluidas en futuras versiones del gestor.

14.5 TAREAS DE UN ADMINISTRADOR DE BASE DE DATOS

Un administrador de BD desempeña las siguientes tareas:

- Software
 - Instalación, desinstalación y actualización de software SGBDR y clientes
 - Instalación y desinstalación de parches
 - Pruebas de productos

- Documentación sobre bugs y versiones de productos
 - Creación de informes de recomendación de software
 - Conocimiento de las aplicaciones de los usuarios
- Creación de BD
- Configuración parámetros de funcionamiento de la BD
- Gestión de
 - Usuarios
 - Creación, Baja
 - Gestión de privilegios
 - Recursos
 - Asignación de tablespaces, espacio en HD
 - Actualización de parámetros del sistema
- Backup y recuperación
- Cargas de datos
 - Crear y modificar programas de carga
 - Ejecutar programas de carga
- Mantenimiento de la BD. Altas, Bajas y Modificaciones de Objetos de la BD (vistas, índices, tablas, etc.)
- Monitorización
 - Accesos de los usuarios
 - Rendimiento de los procesos
 - Crecimiento de ficheros
 - Creación de estadísticas. Crear y modificar programas de estadísticas
- Optimización, mejoras de rendimiento

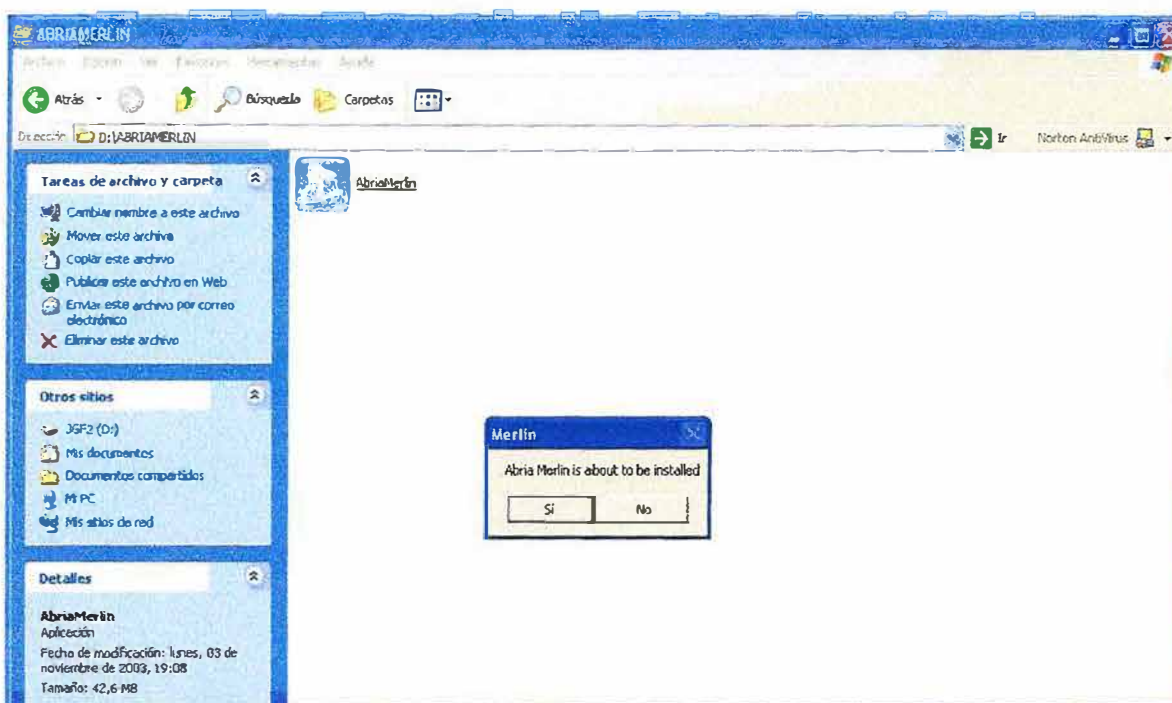
Para administrar la base de datos, normalmente nos proporcionan un interfaz Web bastante potente y amigable. Esto se puede hacer de diferentes maneras, ya que existen distintos programas para administrar bases de datos, como el MySQL-Front.

14.6 MYSQL-FRONT

14.6.1 Instalación y configuración del Abria Merlin

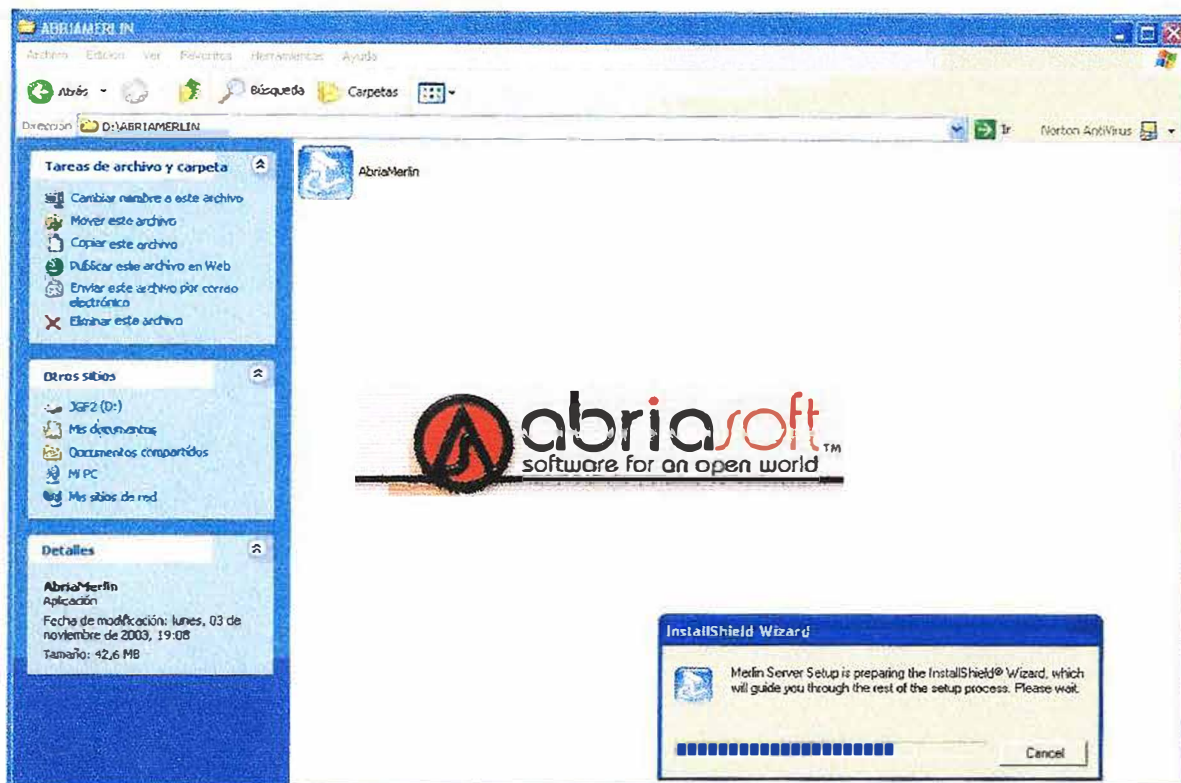
Lo primero que debemos hacer para que el MySQL funcione de manera optima es instalar el Abria Merlin, una vez descargado de Internet, lo ejecutamos haciendo clic sobre el icono con la extensión .exe y escogiendo la opción **SI** en el cuadro de dialogo que aparece, confirmando el proceso de instalación.

Figura 9. Proceso de Instalación Abria Merlin (Paso 1)



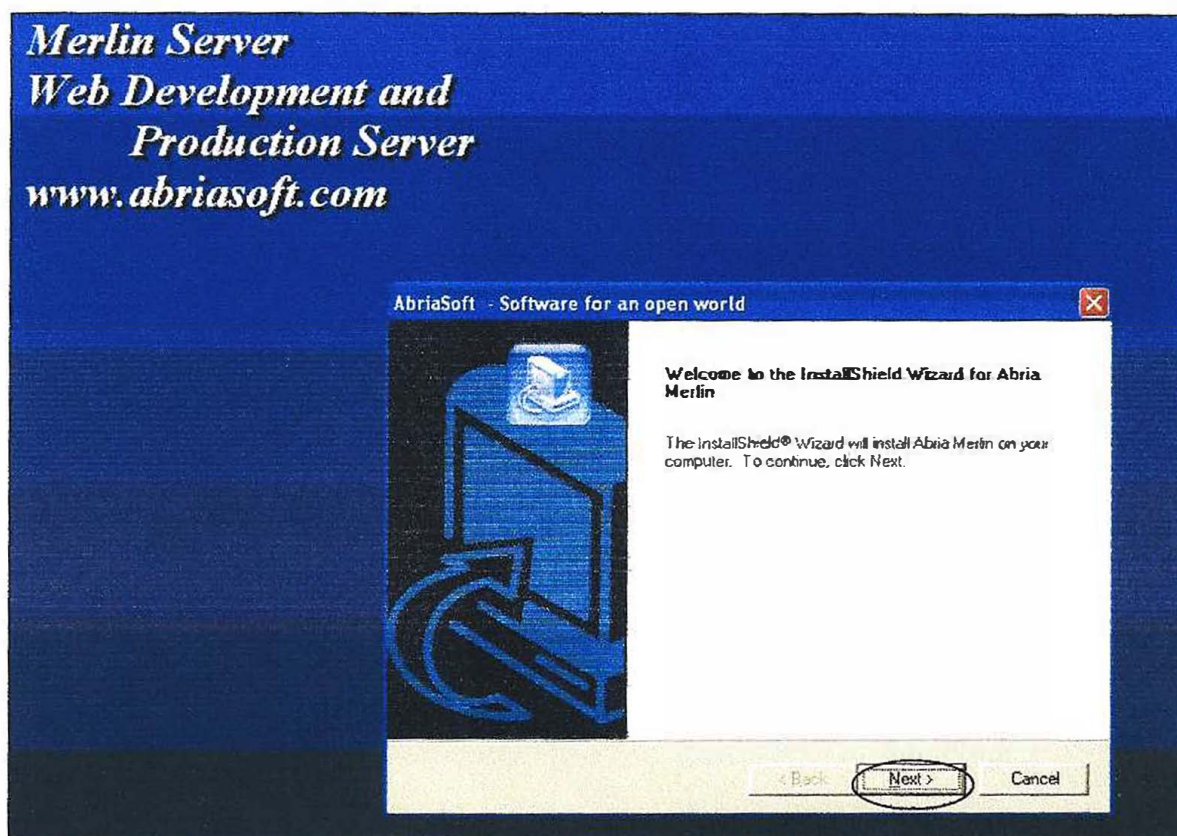
Una vez confirmado el proceso de instalación, se extraerán los archivos necesarios para llevar a cabo dicha instalación.

Figura 10. Proceso de Instalación Abria Merlin (Paso 2)



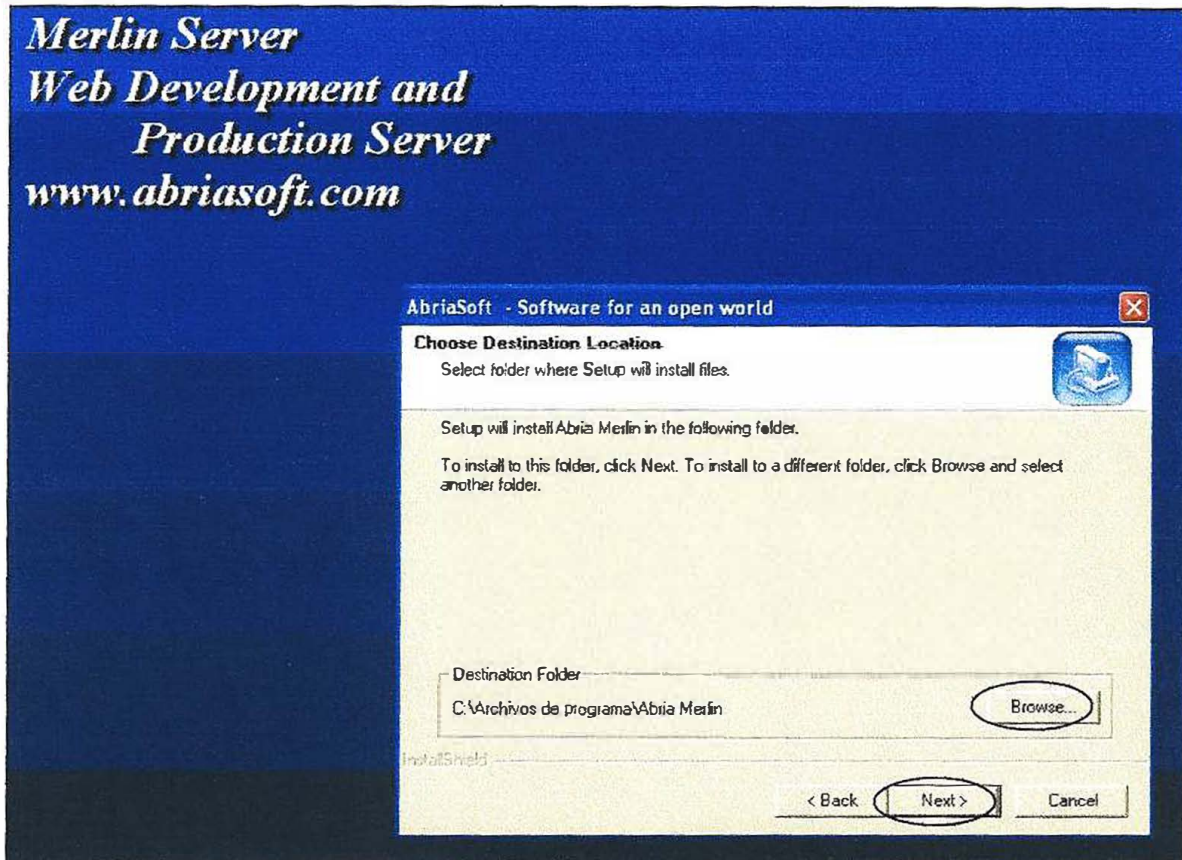
Aparecerá el siguiente cuadro de dialogo, dando la bienvenida al proceso de Instalación de Abria Merlin, hacemos clic en **<Next>**, aceptamos los términos del contrato haciendo clic en **<Yes>**.

Figura 11. Proceso de Instalación Abria Merlin (Paso 3)



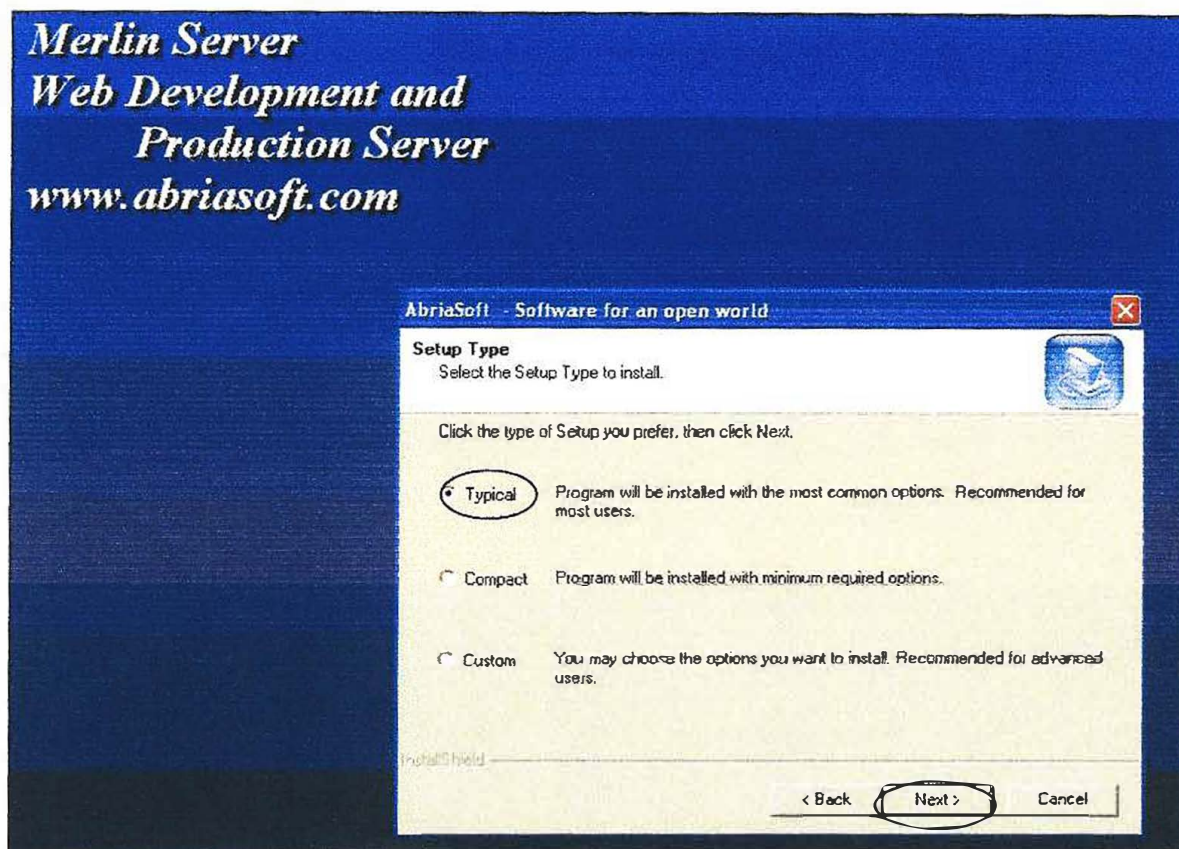
Escogemos la ubicación donde deseamos que queden almacenados los archivos del programa en el disco duro (Por lo general es C:\Archivos de programa\Abria Merlin), si queremos instalarlos en una ubicación diferente, hacemos clic en la opción **<Browse...>** y luego damos clic en **<Next>**.

Figura 12. Proceso de Instalación Abria Merlin (Paso 4)



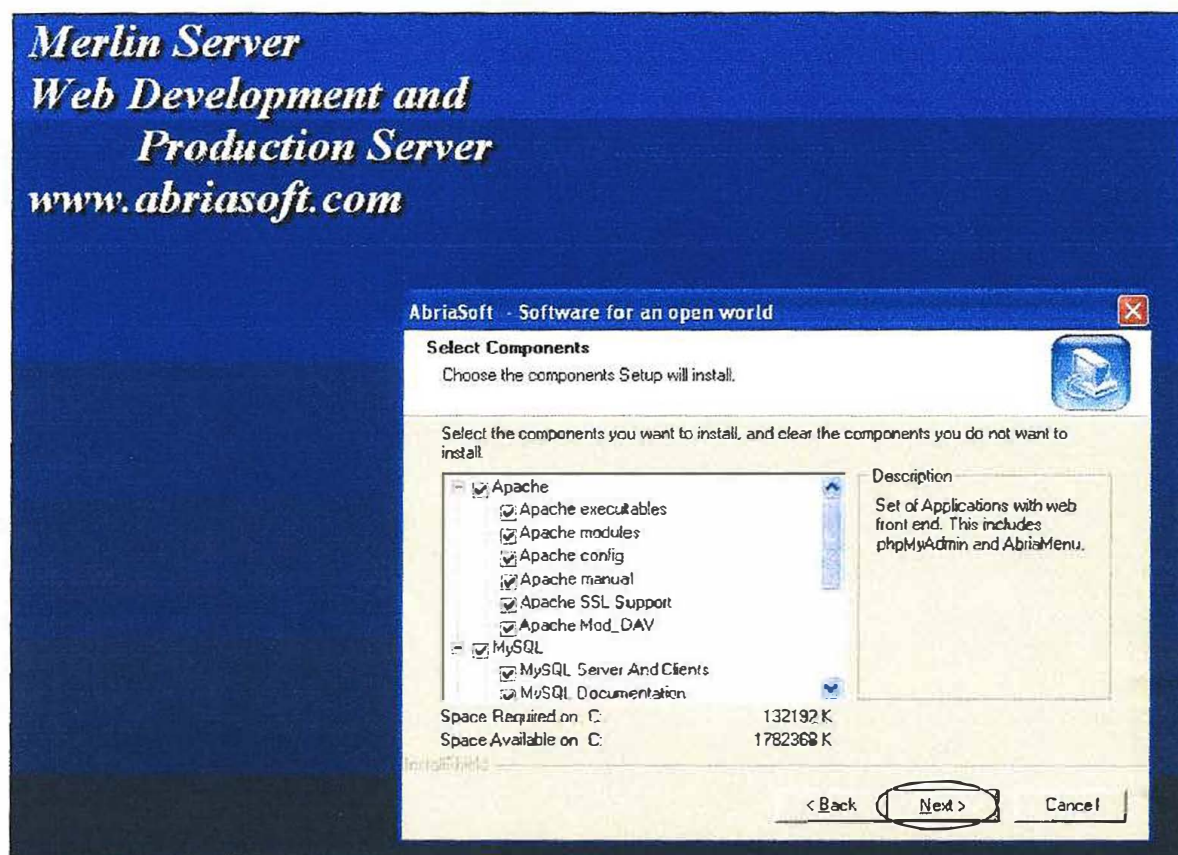
Escogemos el tipo de instalación que queremos (Typical, Compact o Custom). Con la opción **Typical** el programa será instalado con las opciones más comunes recomendada para los usuarios novatos. Con la opción **Compact** el programa será instalado con las opciones mínimas requeridas. Con la opción **Custom** podemos escoger las opciones que deseamos instalar recomendada para los usuarios avanzados.

Figura 13. Proceso de Instalación Abria Merlin (Paso 5)



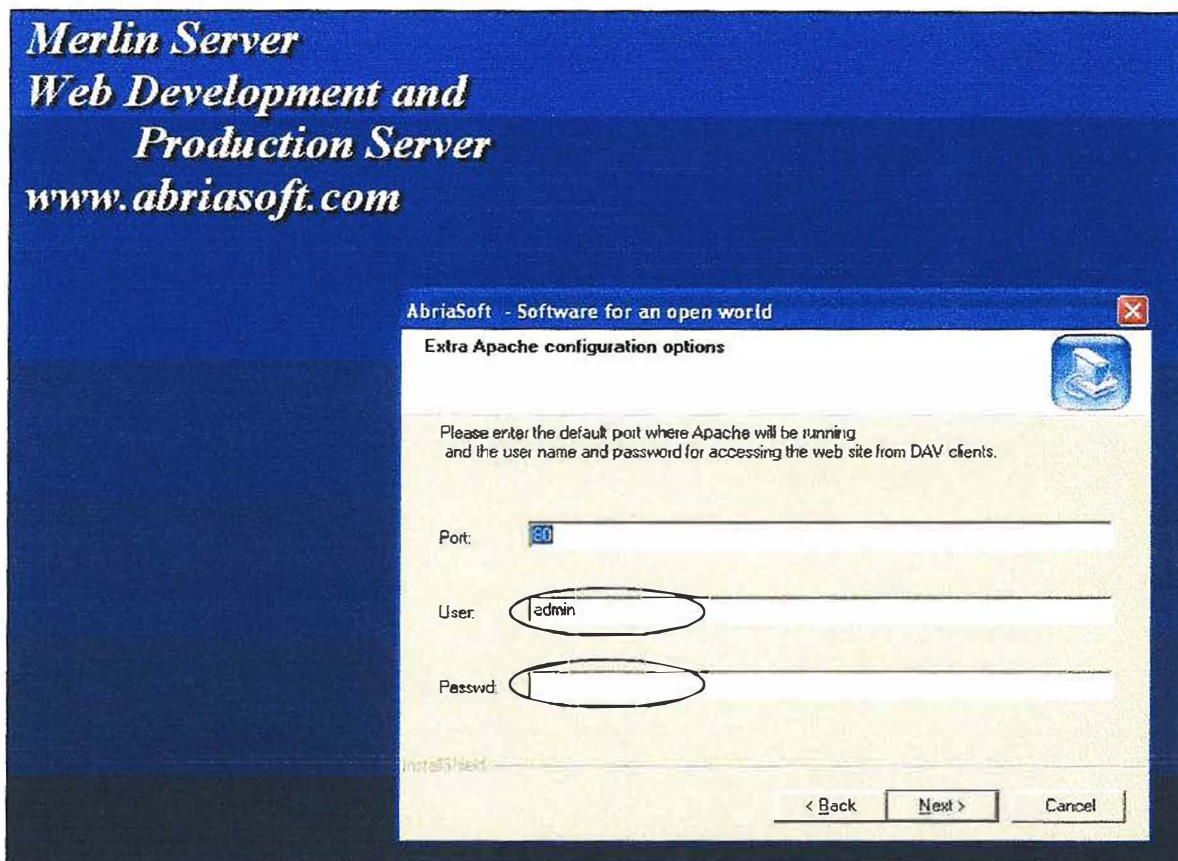
La ultima opción permite seleccionar los componentes que deseamos instalar de este paquete (El servidor apache, MySQL, Php,...), esta opción es para usuarios avanzados, seguidamente hacemos clic en **<Next>**.

Figura 14. Proceso de Instalación Abria Merlin (Paso 6)



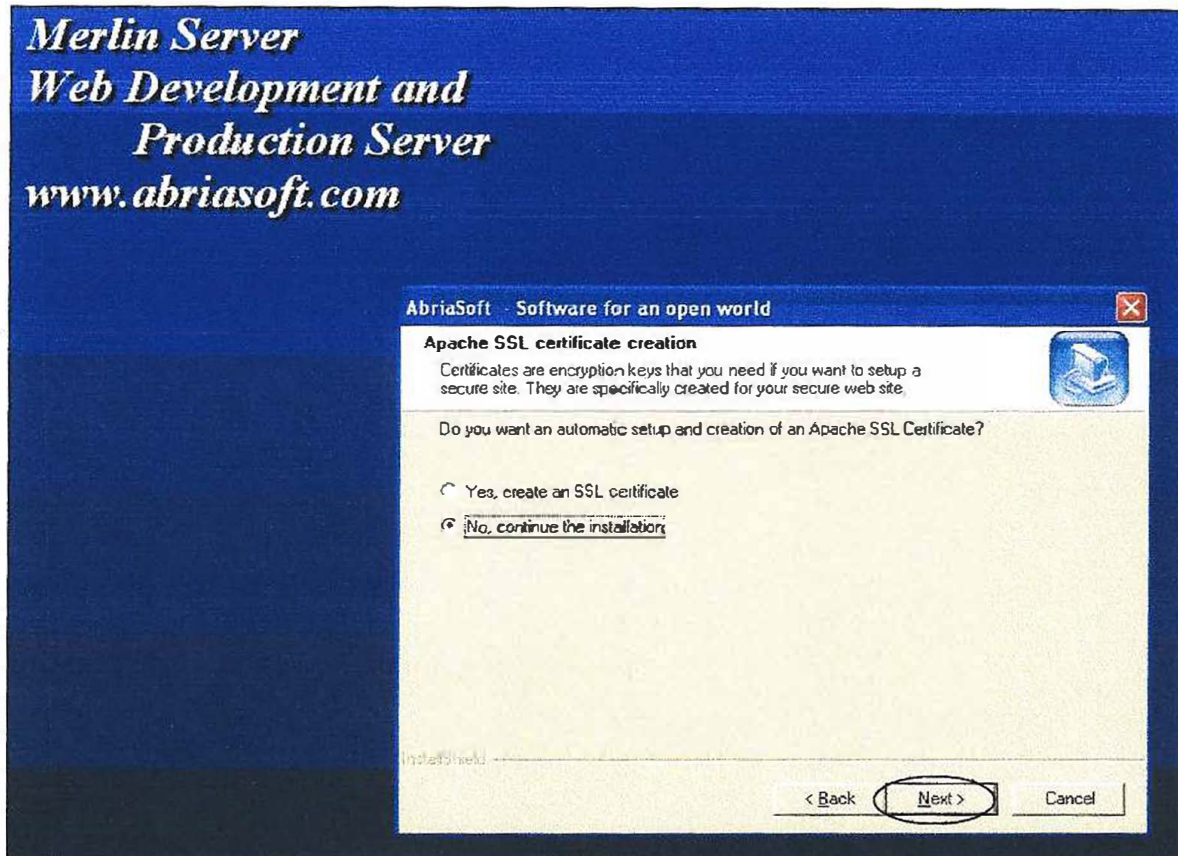
Asignamos un nombre de usuario y contraseña para acceder al sitio web desde DAV clients y hacemos clic en **<Next>**.

Figura 15. Proceso de Instalación Abria Merlin (Paso 7)



Escogemos la primera opción, si nuestra base de datos va a estar ubicada en un sitio web, esto para proporcionar seguridad sobre los datos. En caso contrario, escogemos la segunda opción, luego de esto asignamos una clave de administrador para el MySQL, la cual nos va a permitir acceder a la base de dato, esta es esencial para la seguridad de nuestro servidor.

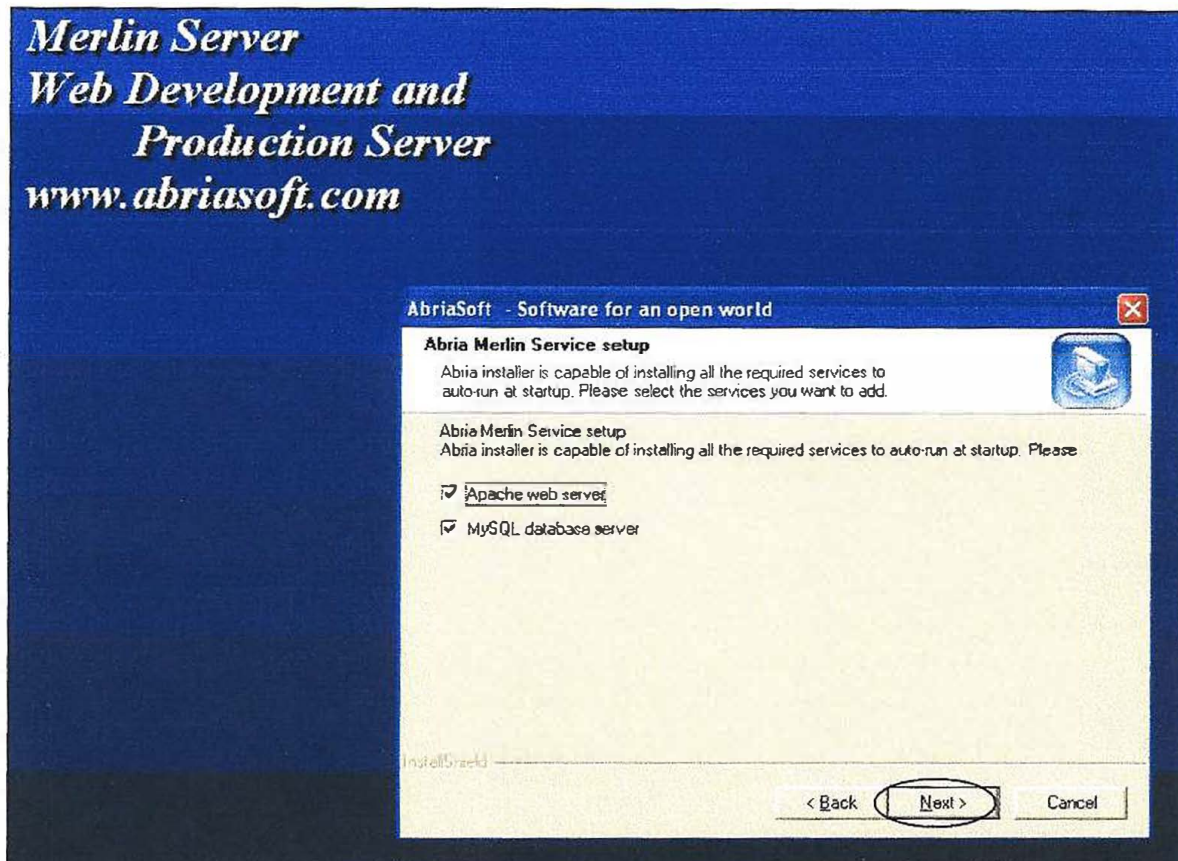
Figura 16. Proceso de Instalación Abria Merlin (Paso 8)



Escogemos las opciones Apache web server y MySQL database Server, la primera debido a que el administrador de MySQL (PhpMyAdmin), necesita de un servidor web para poder funcionar puesto que su estructura esta conformada por código Php y la segunda es nuestro servidor de base de datos.

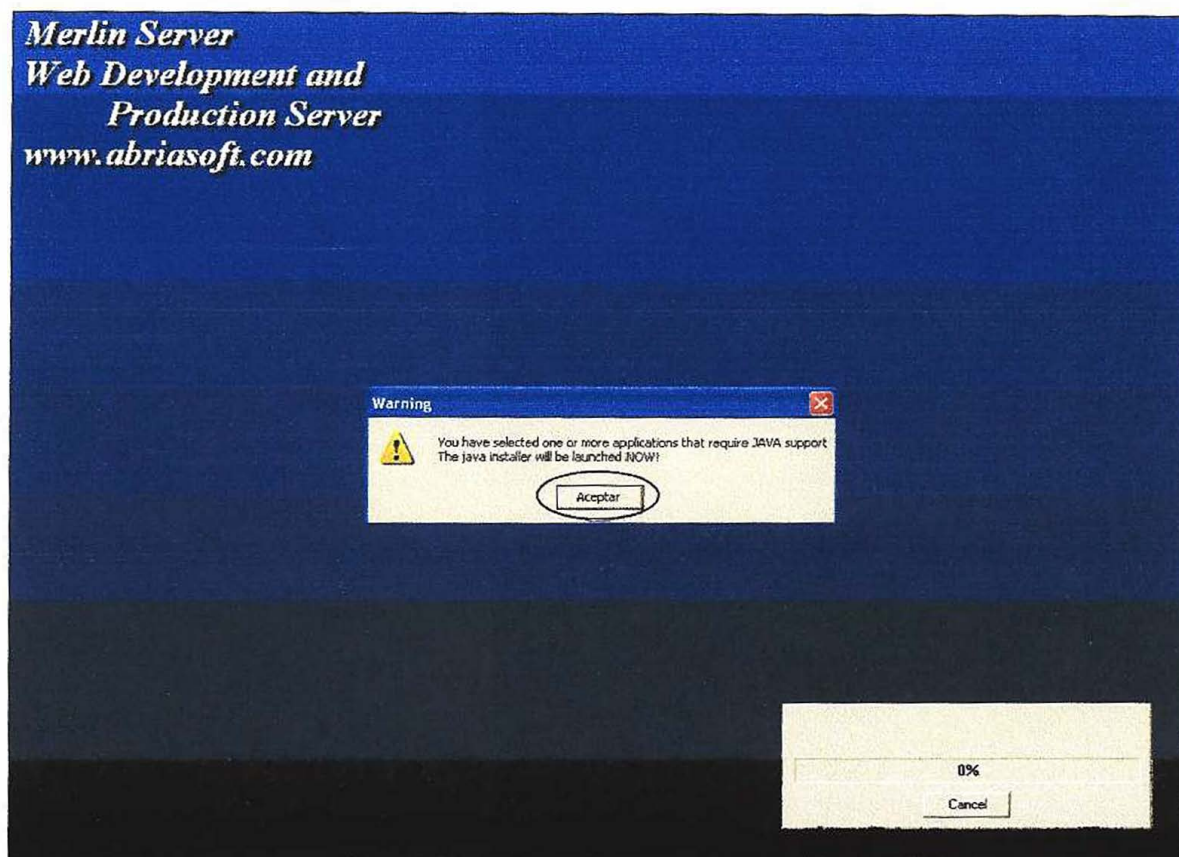


Figura 17. Proceso de Instalación Abria Merlin (Paso 9)



Algunas aplicaciones de este paquete requieren soporte JAVA, por tal razón debe ser instalado junto con el mismo, sino está instalado en su equipo deberá instalarlo también viene en el paquete de Abria Merlin.

Figura 18. Proceso de Instalación Abria Merlin (Paso 10)



Después de haber finalizado la instalación del JAVA, la instalación del Abria Merlin continuara su curso.

Figura 19. Proceso de Instalación Abria Merlin (Paso 11)

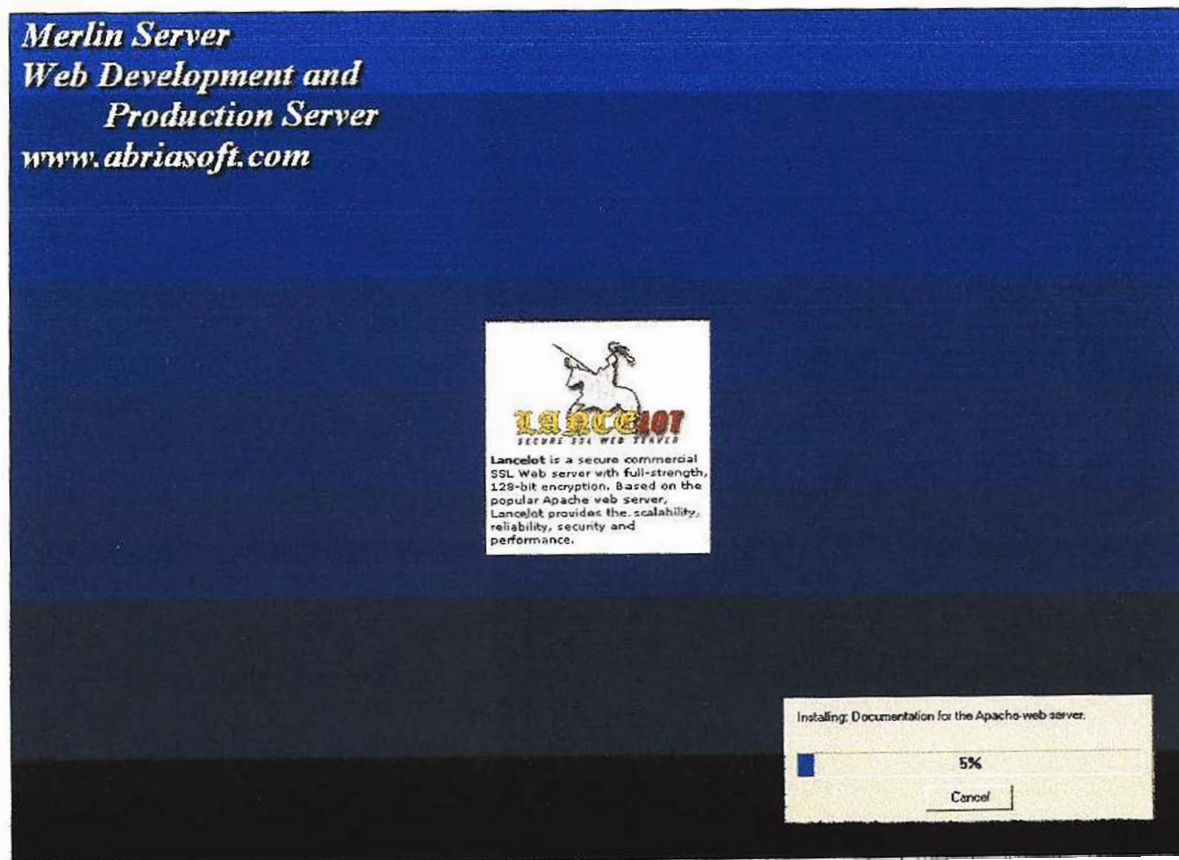


Figura 20. Proceso de Instalación Abria Merlin (Paso 12)

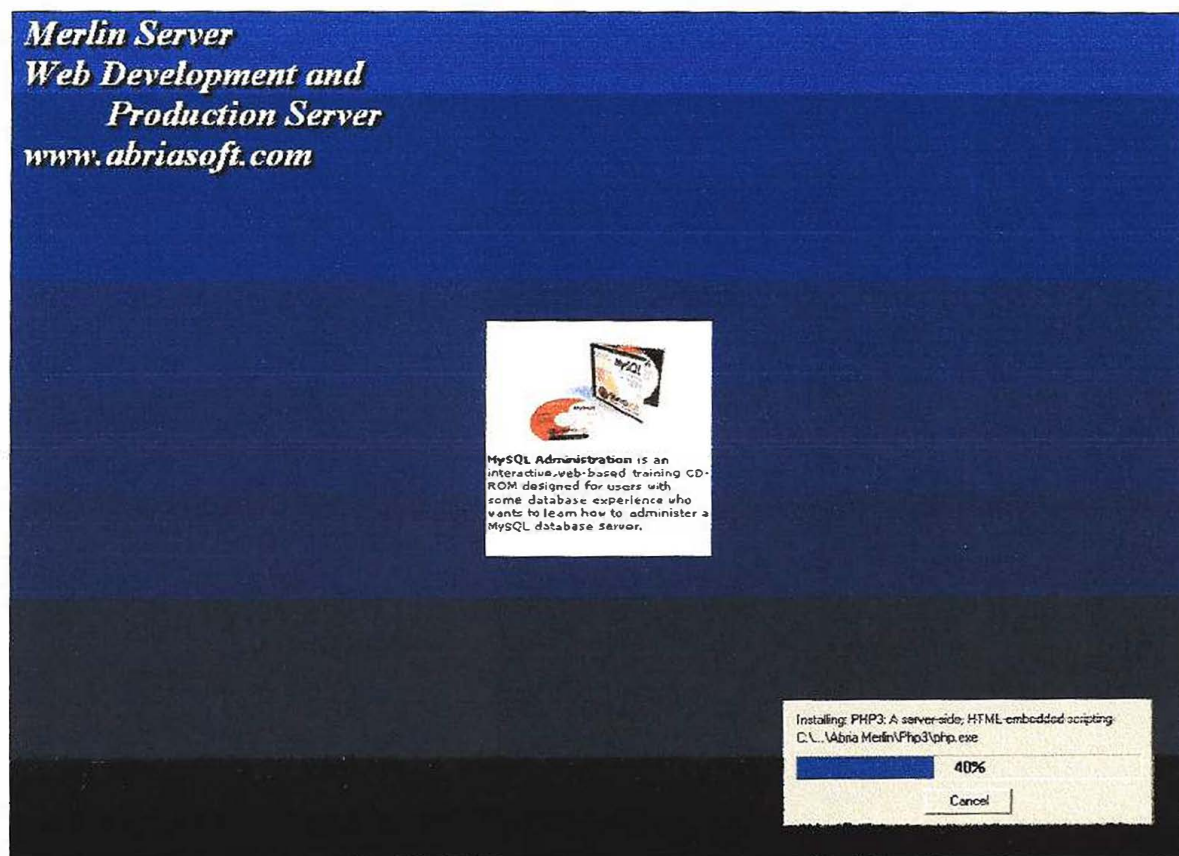
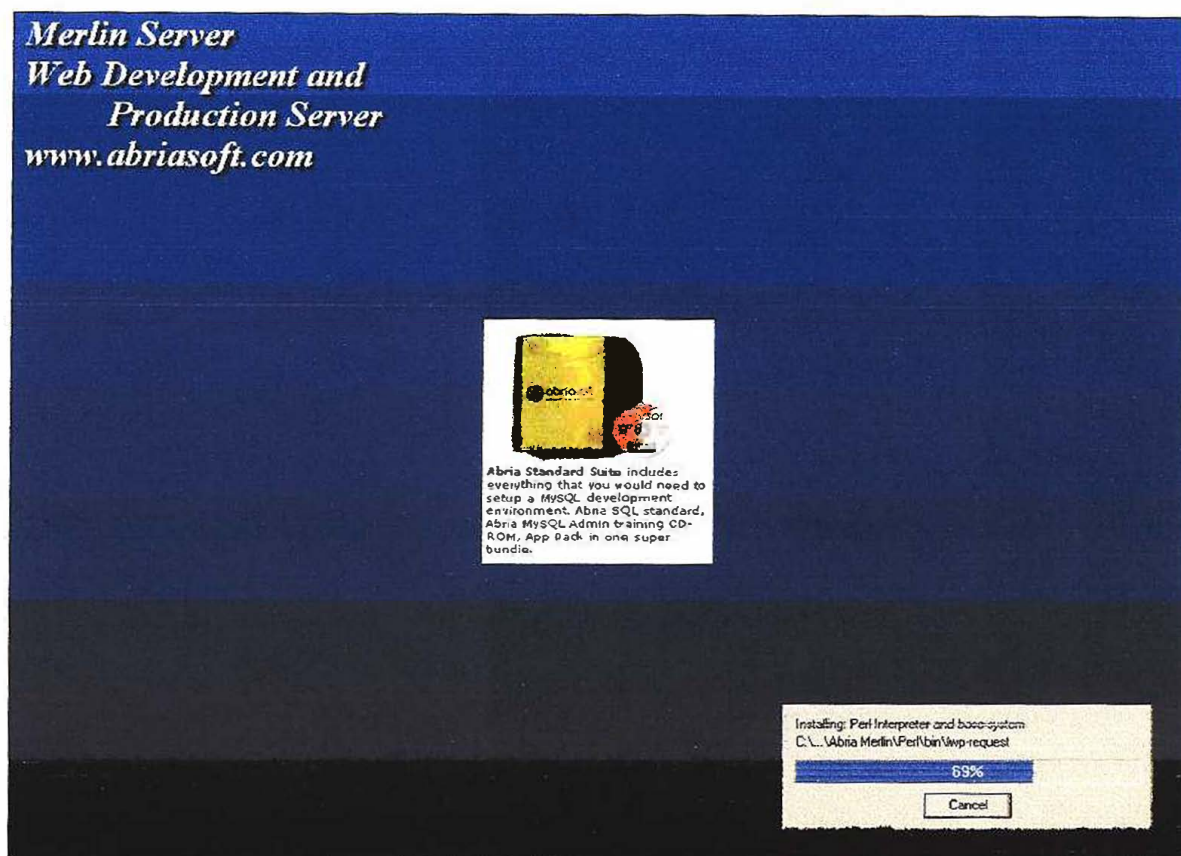
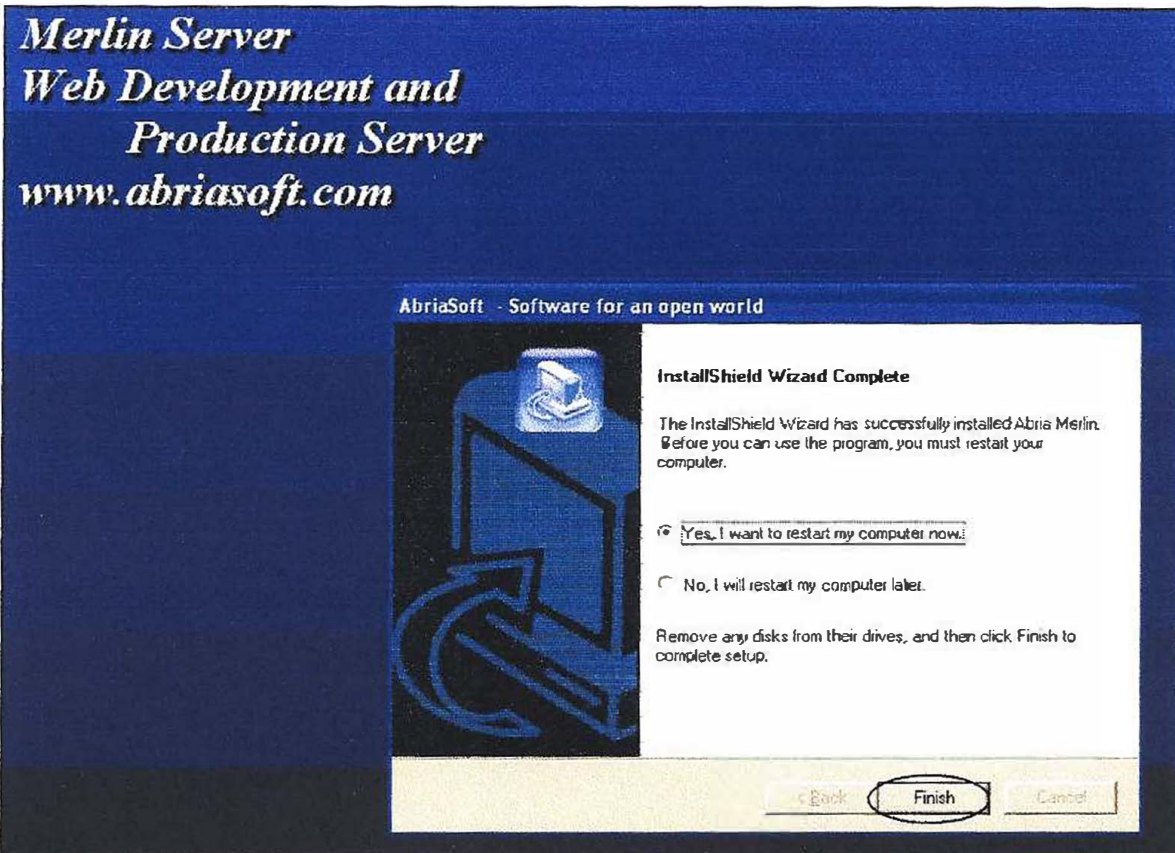


Figura 21. Proceso de Instalación Abria Merlin (Paso 13)



Por ultimo se recibe un mensaje de culminación de la instalación. Se debe reiniciar el equipo para que los cambios surjan efecto y pulsamos la tecla **Finish**.

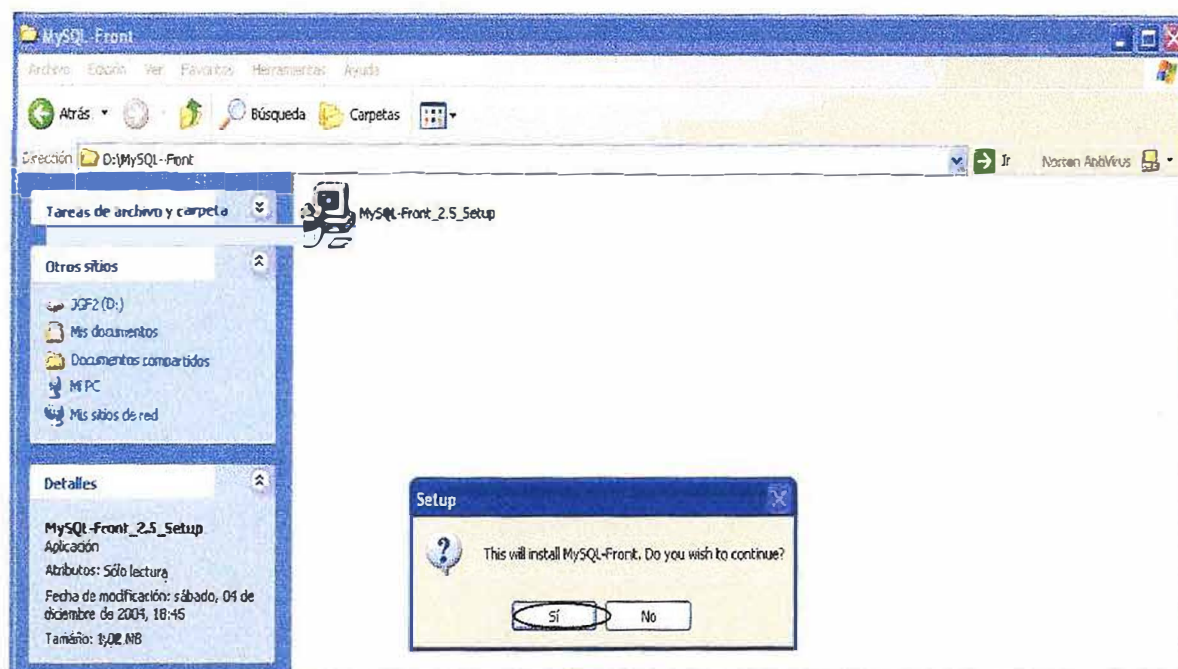
Figura 22. Proceso de Instalación Abria Merlin (Paso 14)



14.6.2 Instalación y configuración del MySQL-Front

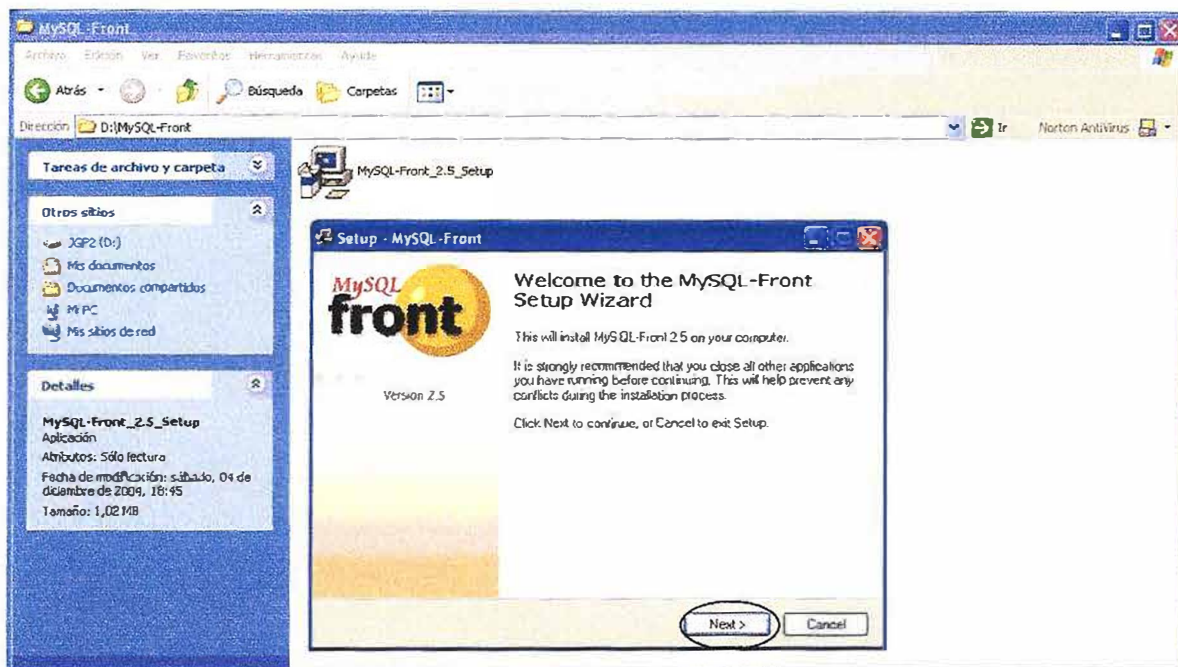
Ahora empezamos el proceso de instalación del MySQL-Front, lo ejecutamos haciendo clic sobre él y escogiendo la opción **SI** en el cuadro de dialogo que aparece, confirmando empezar el proceso de instalación.

Figura 23. Proceso de Instalación de MySQL Front (Paso 1)



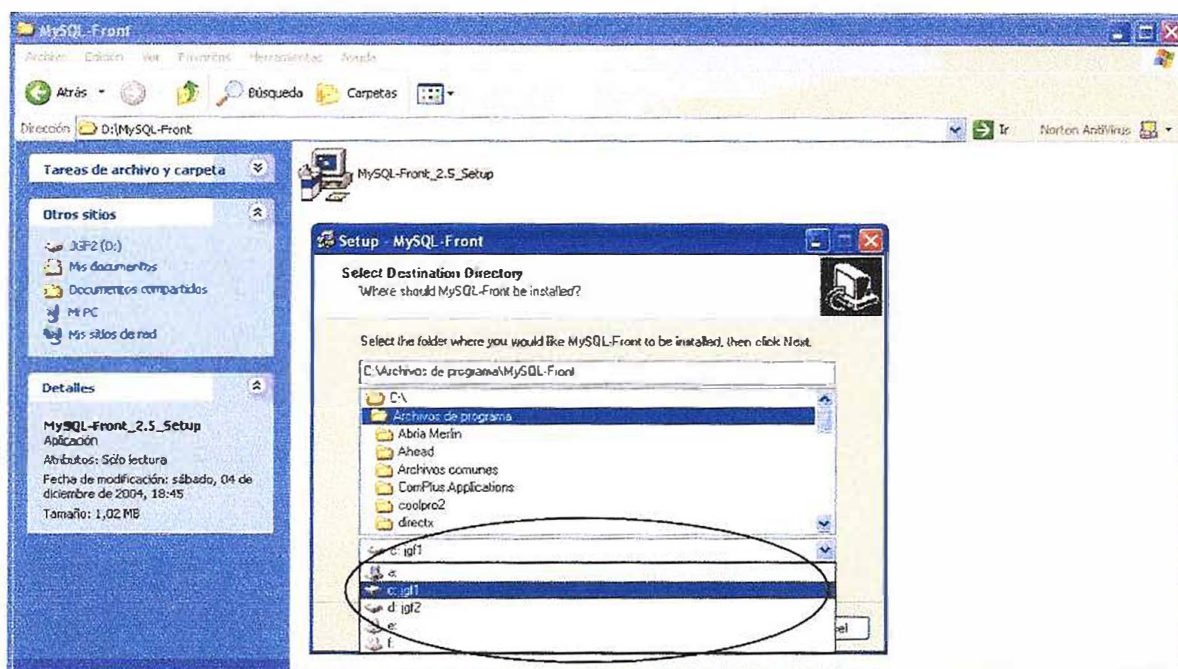
Aparecerá el siguiente cuadro de dialogo, dando la bienvenida a MySQL-Front, hacemos clic en **<Next>**, aceptamos los términos del contrato haciendo clic en **<Yes>**.

Figura 24. Proceso de Instalación de MySQL Front (Paso 2)



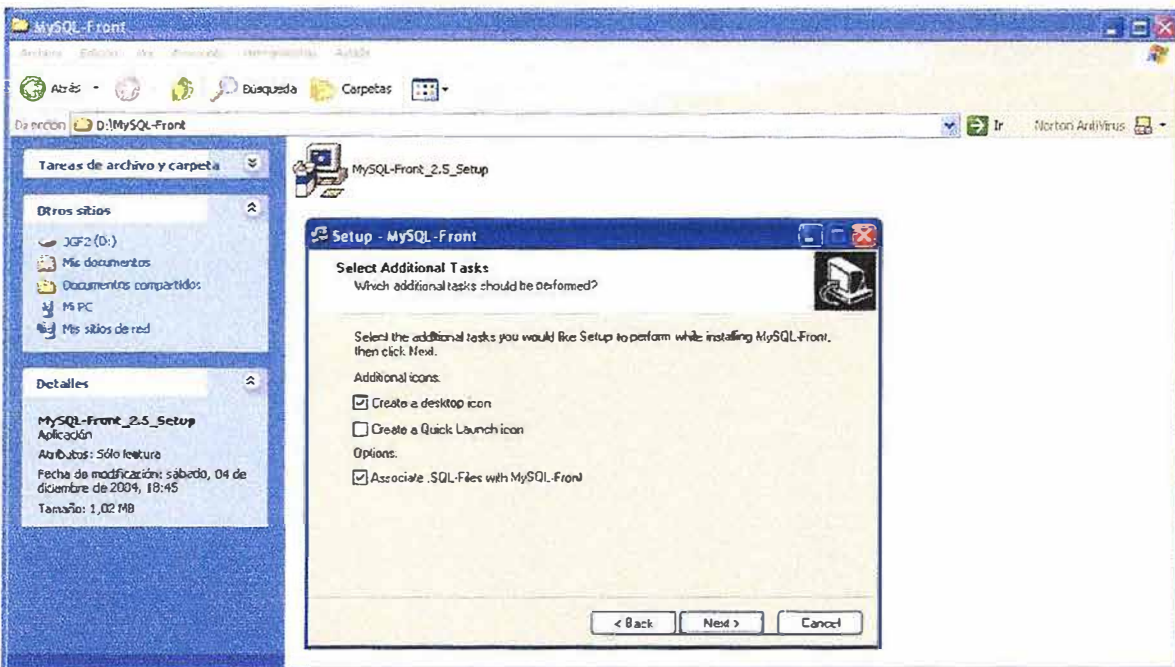
Escogemos la ubicación donde deseamos que queden almacenados los archivos del programa en el disco duro (Por lo general es C:\Archivos de programa\MySQL-Front), si queremos instalarlos en una ubicación diferente, hacemos clic en la caja de opciones ubicada en la parte inferior del cuadro de dialogo que aparece y luego damos clic en **<Next>**.

Figura 25. Proceso de Instalación de MySQL Front (Paso 3)



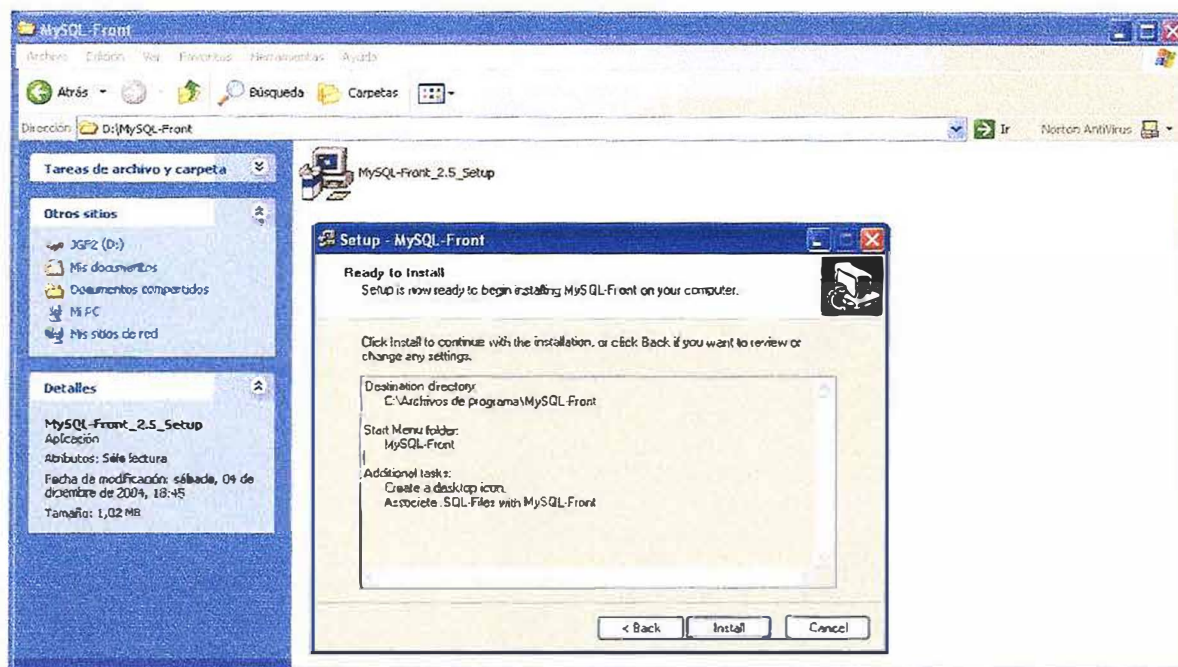
Aparecerá el siguiente cuadro de dialogo con las opciones crear icono en el escritorio, crear icono en la barra de inicio rápido y asociar archivos SQL con MySQL-Front respectivamente, la ultima opción es indispensable, pues son sobre archivos SQL sobre los que se va a trabajar, las otras dos son opcionales y luego damos clic en **<Next>**..

Figura 26. Proceso de Instalación de MySQL Front (Paso 4)



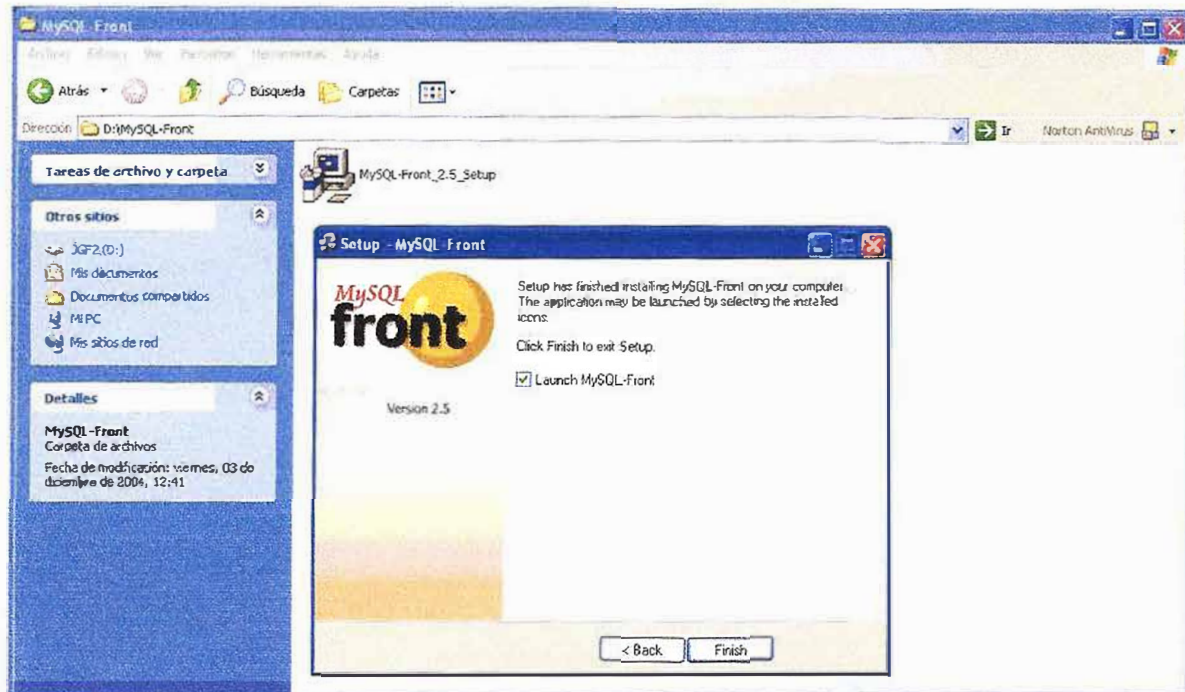
Luego confirmamos la instalación haciendo clic sobre la opción **<Install>**.

Figura 27. Proceso de Instalación de MySQL Front (Paso 5)



Por ultimo se recibe un mensaje de confirmación-culminación de la instalación.

Figura 28. Proceso de Instalación de MySQL Front (Paso 6)



Con **MySQL-Front** se pueden realizar acciones básicas como añadir, borrar o modificar tablas, campos, registros, y además:

- Ⓢ Ver variables del servidor
- Ⓢ Ejecutar y matar procesos
- Ⓢ Ejecutar SQL-scripts
- Ⓢ Exportar tablas a SQL-scripts o a otras bases de datos
- Ⓢ Replicar bases de datos
- Ⓢ Guardar datos en formato HTML o CSV (ideal para Excel)
- Ⓢ Escribir *queries* en SQL
- Ⓢ Importar datos de ODBC
- Ⓢ Realizar diagnóstico de tablas (optimización, reparación, etc.)
- Ⓢ Ver propiedades avanzadas de tablas (tipo, comentario, key_length, etc.)

Para funcionar correctamente necesita:

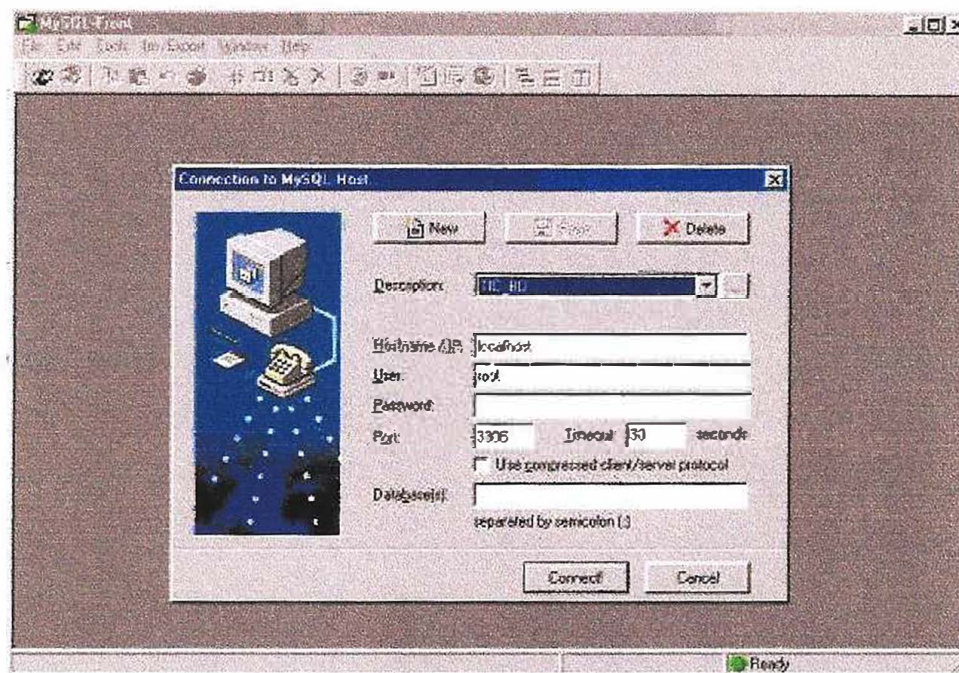
- ④ Microsoft C Runtime Library (msvcrt.dll)
- ④ Protocolo TCP/IP
- ④ ODBC (para la importación de datos)
- ④ La librería Cliente para MySQL Server incluida (libmysql.dll)

Una vez instalado el My SQL-Front, hacemos lo siguiente:

14.6.3 Conectarse a la Base de Datos

- ④ Click en **New**
- ④ Digitar el nombre de la conexión
- ④ Digitar nombre del host (generalmente es **localhost**)
- ④ Digitar el nombre del usuario (generalmente es **root**)
- ④ Digitar el password correspondiente
- ④ Digitar nombre de la base de datos a la que desea conectarse (si existe)
- ④ Click en **<connect>**

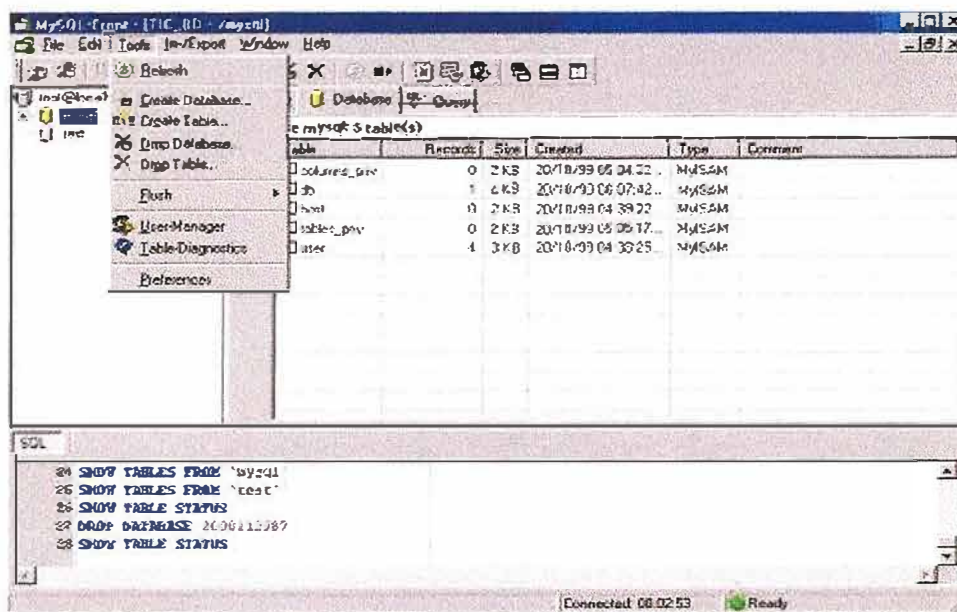
Figura 29. Conectarse a la Base de Datos



14.6.4 Creación de Base De Datos y/o Tablas

Una vez conectado, la opción **Tools** de la barra de herramientas permite crear y eliminar BD, así como también crear y eliminar Tablas.

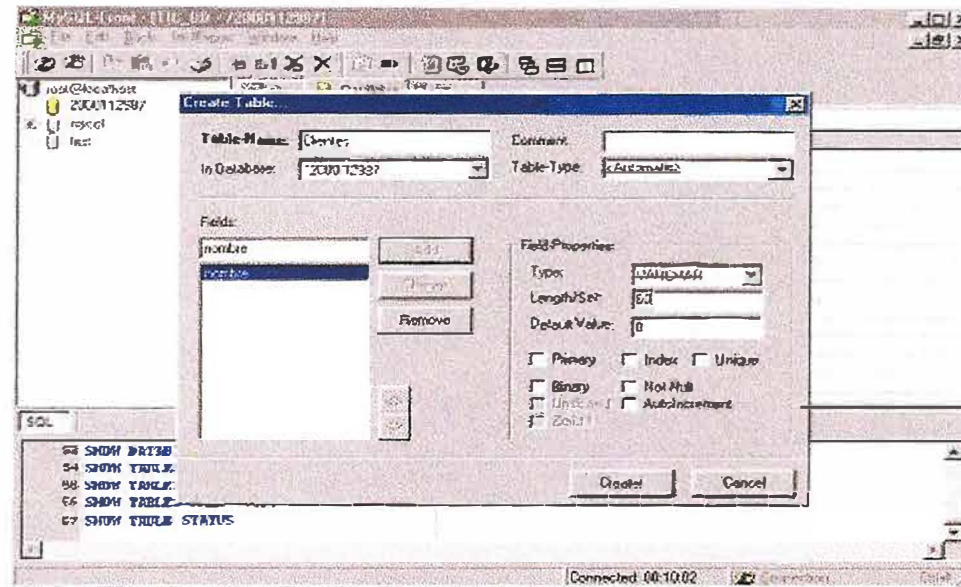
Figura 30. Creación de Base de Datos



14.6.5 Creación de Tablas

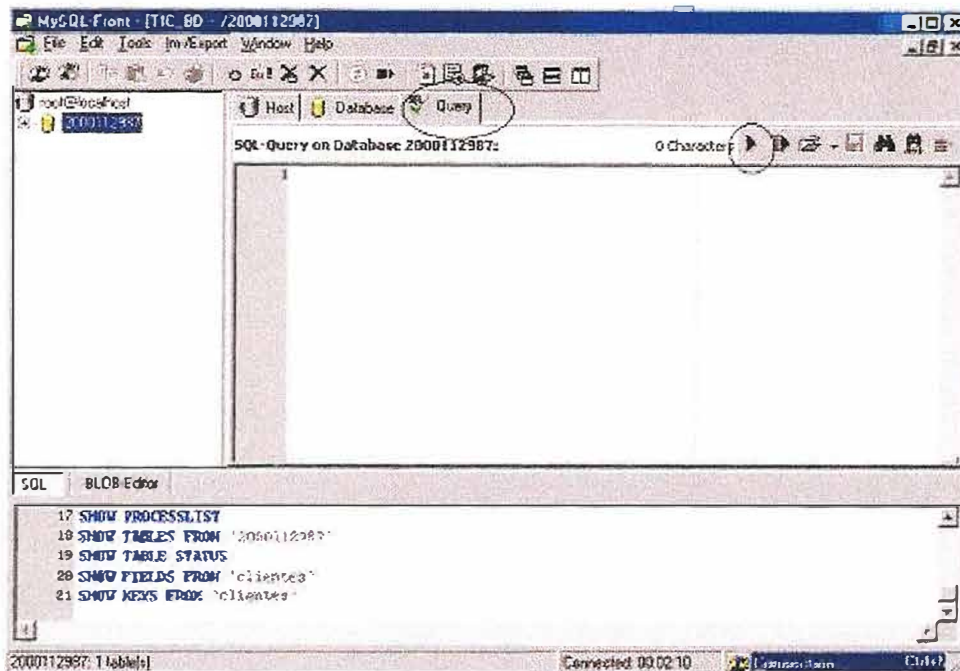
Una vez creada la BD, procedemos a crear las tablas con sus campos y propiedades correspondientes (Tipo, longitud, PK, not null,...), adicionando **<Add>** uno a uno los campos y por ultimo click en **<create!>**

Figura 31. Creación de tablas y entidades



Otra manera de crear tablas en una base de datos es transcribiendo los **scripts** de la Base de Datos, en el espacio en blanco que aparece al seleccionar la pestaña **Query** y ejecutando con la punta de flecha encerrada en el círculo.

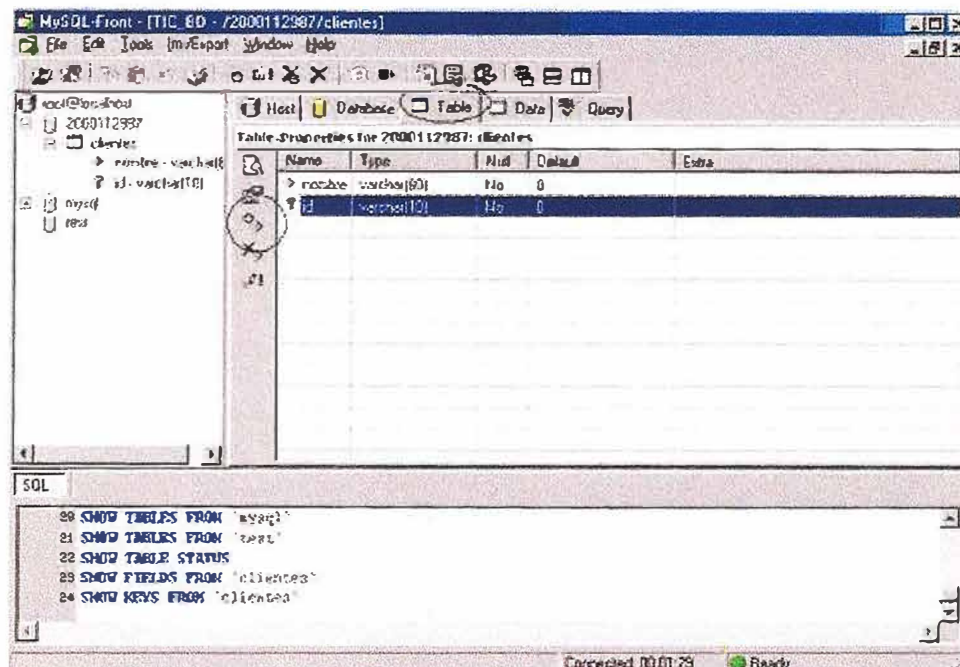
Figura 32. Creación de tablas por Query



14.6.6 Visualizar e Ingresar más Campos y Datos

Si se quieren ver los campos de la tabla se selecciona la pestaña **Table** (encerrada en el círculo) y si se necesita adicionar más campos a la tabla se da click en la **cruz ubicada al lado izquierdo** de la ventana (encerrada en el círculo).

Figura 33. Ingresar más campos



Ahora bien, si se quiere ingresar datos a la tabla, se da click en la pestaña **Data** ubicada al lado de la opción Table y luego se presiona <F5> para actualizar.

14.7 PHPMYADMIN: ADMINISTRACION MYSQL A TRAVÉS DE LA WEB

S.O.: Win95/98/ME/NT/2000/XP/Linux

Web: <http://www.phpmyadmin.net/>.

Para que nos funcione debemos tener configurado el soporte php e instalado MySQL. Una de las mejores utilidades, es la capacidad de exportar e importar información de un modo sencillo y remoto. Podemos visualizar los scripts de inicialización de la base de datos con lo que podemos reconstruirla fácilmente.

PhpMyAdmin es un programa de libre distribución en PHP, creado por una comunidad sin ánimo de lucro. Es una herramienta muy completa que permite acceder a todas las funciones típicas de la base de datos MySQL a través de una interfaz web muy intuitiva.

La aplicación es un conjunto de archivos escritos en PHP que podemos copiar en un directorio de nuestro servidor web, de modo que, cuando accedemos a esos archivos, nos muestran unas páginas donde podemos encontrar las bases de datos a las que tenemos acceso en nuestro servidor de bases de datos y todas sus tablas. La herramienta nos permite crear tablas, insertar datos en las tablas existentes, navegar por los registros de las tablas, editarlos y borrarlos, borrar tablas y un largo etcétera, incluso ejecutar sentencias SQL y hacer un backup de la base de datos.

14.7.1 Página de PhpMyAdmin

La página de inicio del proyecto es <http://www.phpmyadmin.net/>. Desde allí podemos descargar los ficheros de la última versión de la aplicación, que posteriormente debemos colocar en nuestro servidor web. También podemos encontrar a phpMyAdmin dentro de la red Sourceforge.net, que es un sitio que recoge multitud de proyectos "Open Source" (código abierto).

Hay varias versiones disponibles, pero es recomendable escoger la que nos aconsejen como la última versión estable (The last stable versión). En el momento de escribir este artículo era la 2.5.6, por lo que hemos de descargar el archivo phpMyAdmin-2.5.6.zip

Una vez descargado, hay que descomprimirlo en el directorio raíz del servidor web. Se generará la carpeta **phpMyAdmin-2.5.6** quedando los directorios de la

siguiente manera: c:\inetpub\wwwroot\phpMyAdmin-2.5.6. Si se desea, puede renombrar la carpeta **phpMyAdmin-2.5.6** a **phpMyAdmin** (c:\inetpub\wwwroot\phpMyAdmin).

Los archivos que hemos descargado son de la versión 4 de PHP, aunque también ofrecen la posibilidad de bajarse los archivos que guardan compatibilidad con la versión 3 de PHP, para que aquellos que no dispongan del motor de PHP más actual.

La página de inicio del programa también nos ofrece la posibilidad de ver un demo online, aunque nos avisan de que el servidor donde se aloja puede estar caído.
<http://www.phpmyadmin.net/phpMyAdmin/>

14.7.2 Instalación de PhpMyAdmin

PhpMyAdmin es una herramienta escrita en PHP con el fin de facilitar la tarea de administración de una base de datos MySQL desde el navegador, con el cual es posible, entre muchas otras cosas, crear/borrar/alternar tablas, borrar/editar/agregar campos y ejecutar sentencias SQL.

Se trata de una herramienta escrita íntegramente en lenguaje PHP, realizada con el fin de facilitar las tareas de administración. Permite administrar en forma completa un servidor MySQL (requiere de una cuenta super usuario) como así también bases de datos simples de un usuario en particular.

14.7.3 Características

 crear y eliminar bases de datos

- ⌚ crear, copiar, eliminar, renombrar y alternar tablas
- ⌚ mantenimiento de tablas
- ⌚ eliminar, editar y agregar campos
- ⌚ ejecutar sentencias SQL
- ⌚ administrar campos claves
- ⌚ cargar archivos de textos a las tablas
- ⌚ exportar los datos a diferentes formatos
- ⌚ administrar usuarios MySQL y privilegios
- ⌚ administrar múltiples servidores
- ⌚ crear archivos PDF del diseño de la base de datos
- ⌚ búsquedas globales
- ⌚ comunicar en 47 idiomas diferentes

14.7.4 Requisitos

Es necesario contar con un servidor web con soporte de **PHP**, un motor de bases de datos **MySQL** y, por supuesto, un navegador web. Ante cualquier duda sobre cómo instalarlos, se recomienda visualizar los artículos de nuestro sitio.

14.7.5 Configuración

Abrimos el archivo **config.inc.php** con algún editor de texto como Edit Plus o simplemente el **Notepad de Windows**. Deberemos configurar las siguientes instancias:

\$cfg['PmaAbsoluteUri'] = 'http://localhost/phpMyAdmin';

Hay que asignar la ruta completa para acceder al phpMyAdmin. Si trabajamos en forma local: **http://localhost/phpMyAdmin** y en caso de trabajar en un servidor **http://www.servidor.com/phpMyAdmin**.

\$cfg['Servers'][\$i]['host'] = 'localhost';

El host o número de IP del servidor de la base de datos MySQL.

\$cfg['Servers'][\$i]['user'] = 'usuario';

El nombre del usuario de la base de datos. Si trabajamos en forma local generalmente pondremos el usuario **root** (super usuario) para poder administrar usuarios y bases de datos desde el navegador. En caso de utilizar un servidor externo, seguramente sólo tendremos privilegios de usuario común con una sola base de datos.

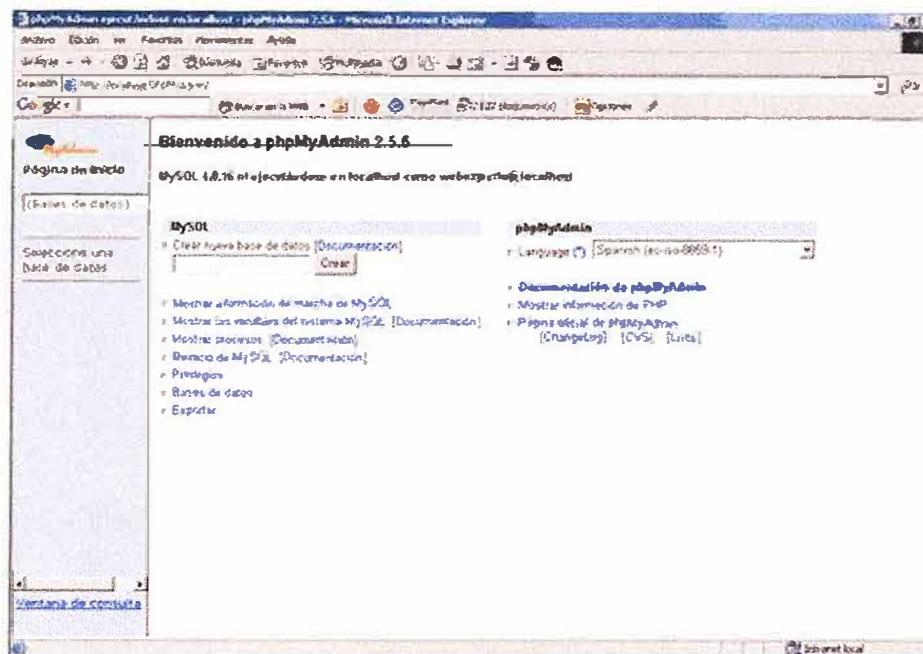
\$cfg['Servers'][\$i]['password'] = 'contraseña';

La contraseña del usuario especificado en la línea anterior.

14.7.6 Utilizando PhpMyAdmin

Configurando las líneas señaladas anteriormente ya estamos habilitados para utilizar la herramienta. En el navegador ingresamos a la dirección correspondiente <http://localhost/phpMyAdmin>

Figura 34. Página inicial de PhpMyAdmin



14.7.6.1 Creación de bases de datos

En la página principal completamos el campo del formulario con el nombre de la base de datos deseada (**pruebas**) y apretamos sobre el botón **Crear**.

Figura 35. Crear base de datos

MySQL

Crear nueva base de datos [Documentación]

14.7.6.2 Creación de tablas

En la página siguiente, creamos una tabla (**noticias**) y la cantidad de campos (**3**).

Figura 36. Crear tabla

Base de datos *pruebas* ejecutándose en *localhost*

Base de datos pruebas se creó.

consulta SQL : [\[Editar\]](#) [\[Crear código PHP\]](#)
CREATE DATABASE `pruebas`;

Estructura

SQL

Exportar

Buscar

No se han encontrado tablas en la base de datos.

- Crear nueva tabla en la base de datos pruebas :

Nombre :

Campos :

14.7.6.3 Configuración de los campos

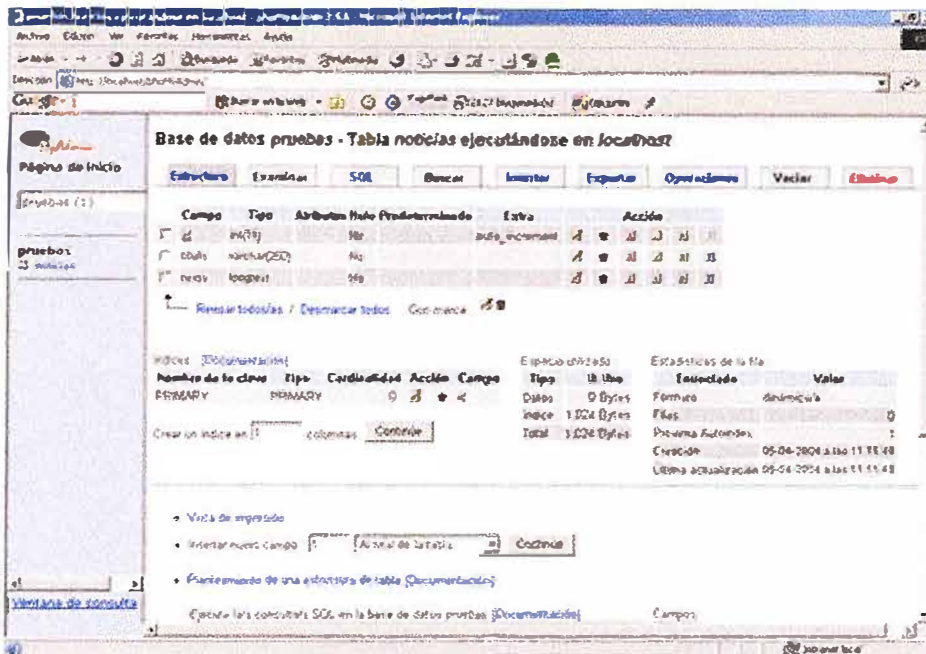
Luego que definimos cuántos campos vamos a utilizar, debemos configurarlos:

Figura 37. Configurar campos

Campo	Tipo	Longitud	Extra	Primaria
id	INT	11	auto_increment	Sí
titulo	VARCHAR	250		
texto	LONG TEXT			

Completado los valores, clickeamos sobre **Grabar**.

Figura 38. Grabar campos



14.7.6.4 Adicionar datos

Para agregar datos hay que clickear sobre **Insertar**, completamos el formulario (el campo id no es necesario completarlo ya que lo declaramos como autonumérico) y clickeamos sobre **Continuar**.

Figura 39. Insertar datos


Campo	Tipo	Función	Nulo	Valor
id	int(11)			
titulo	varchar(250)			Instalación y uso del phpMyAdmin
	longtext			
Artículo dedicado a la instalación, configuración e instalación del phpMyAdmin...				


14.7.6.5 Visualización de datos

Los datos de la tabla pueden ser visualizados clickeando sobre **Examinar**, con el cual veremos un listado de los diferentes registros contenidos en la tabla. Allí mismo, se permite la opción de Editar y Eliminar los mismos.

Figura 40. Ver datos

id	titulo	texto
1	Instalación y uso del phpMyAdmin	Artículo dedicado a la instalación, configuración ...

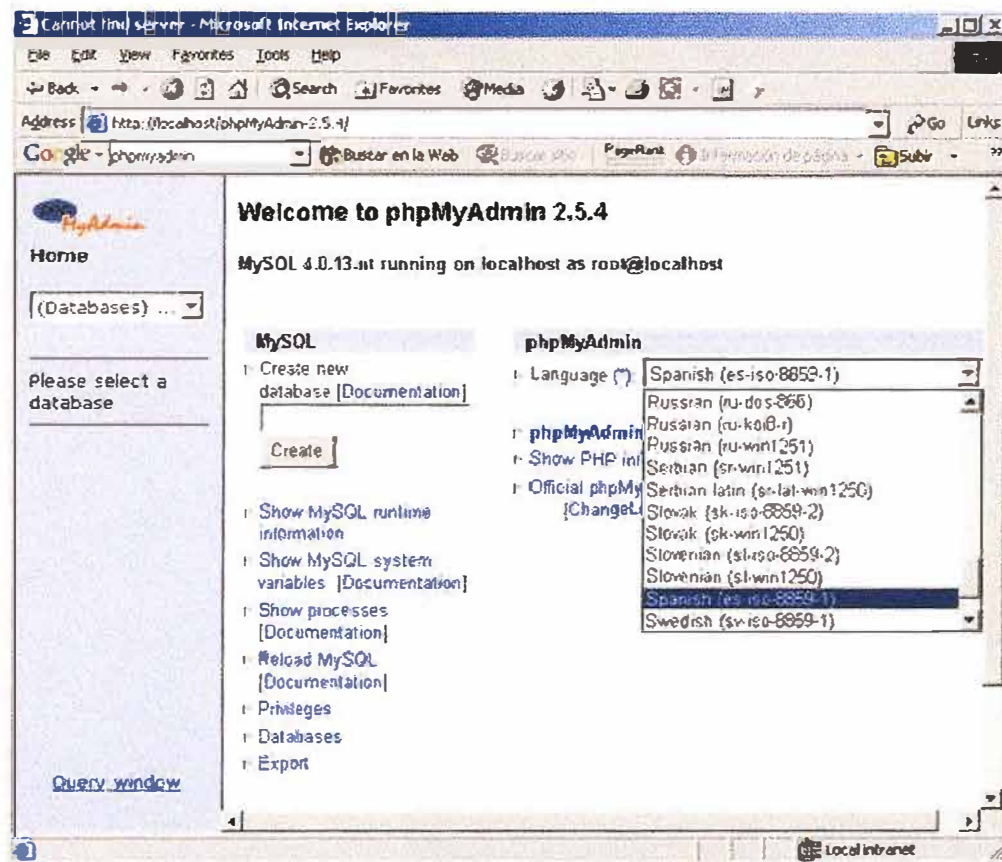
Con marca: 

 eliminar

 modificar

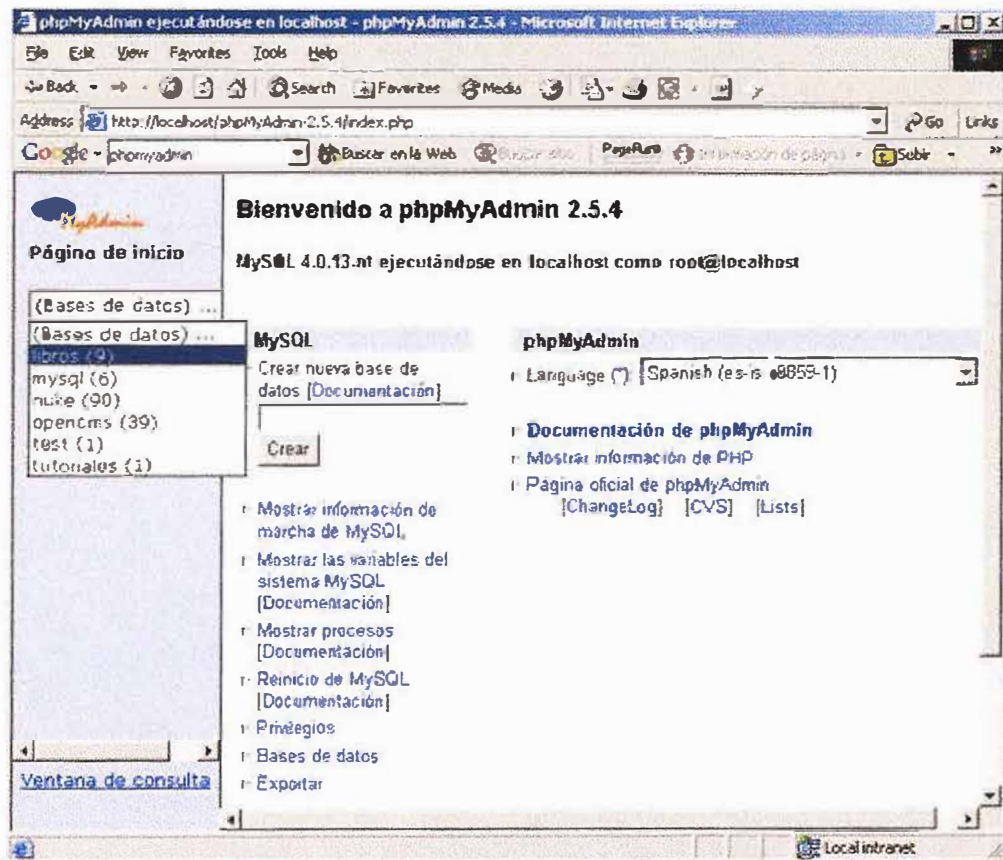
PhpMyAdmin es una excelente herramienta de administración de bases de datos MySQL, que permite su utilización a del navegador web. Además de ser muy intuitivo y fácil de utilizar, es realmente completo y eficaz.

Figura 41. Welcome to PhpMyAdmin 2.5.4 (1)



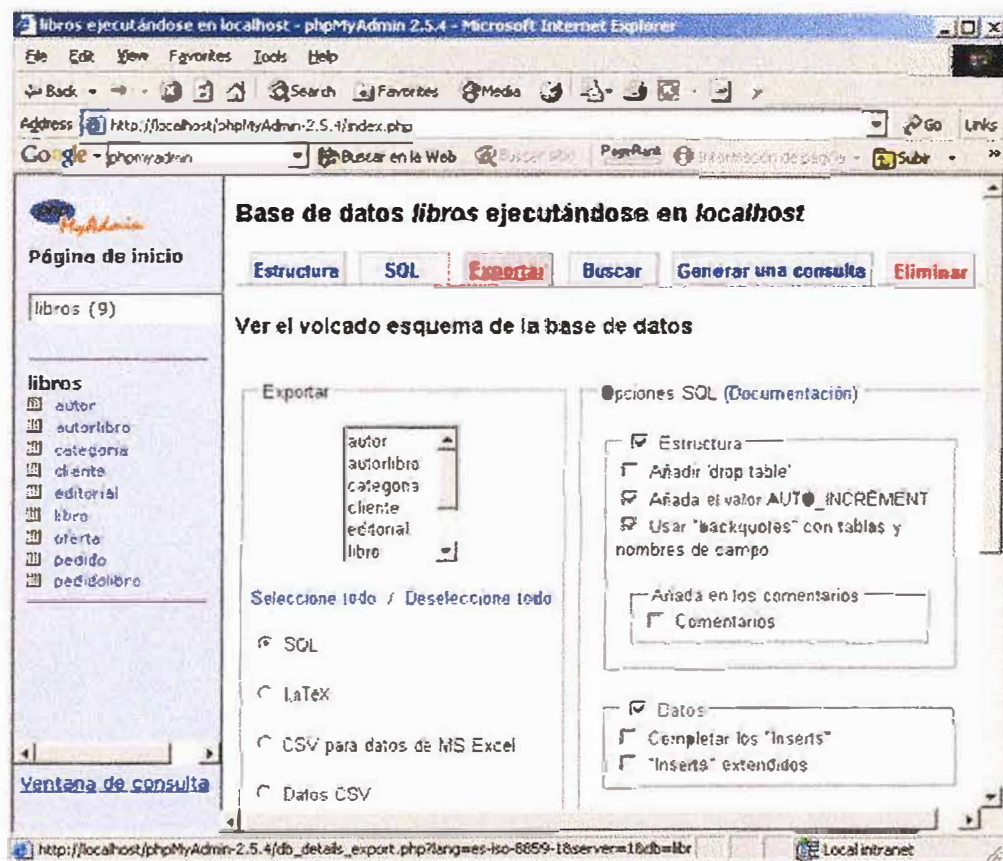
Y la base de datos con la que trabajar. Verificar en resto de opciones fundamentalmente la parte de privilegios.

Figura 42. Welcome to PhpMyAdmin 2.5.4 (2)



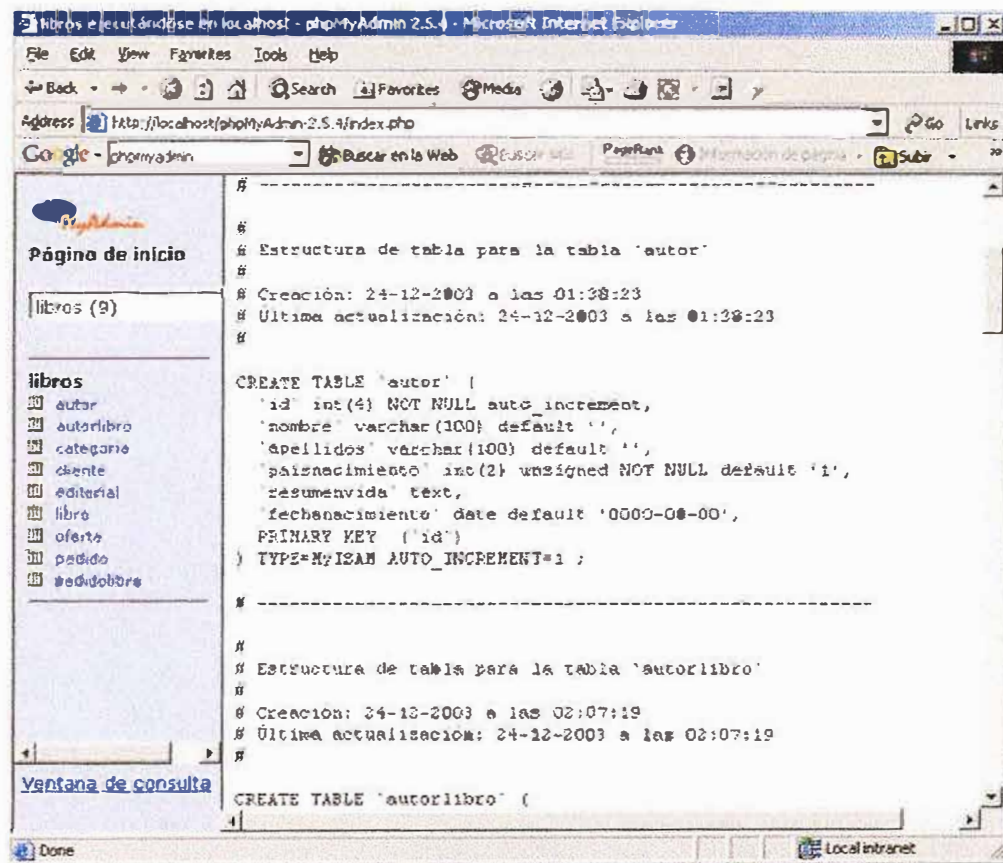
Una de las mejores utilidades, es la capacidad de exportar e importar información de un modo sencillo y remoto.

Figura 43. Exportar información



Podemos visualizar los Scripts de inicialización de la base de datos con lo que podemos reconstruirla facilmente.

Figura 44. Ver los script de la base de datos



14.8 MYSQLADMIN

S.O.: Win95/98/ME/NT/2000/XP/Linux

Web: mysqlAdmin.net

Utilidad por realizar los funcionamientos administrativos, como crear o eliminar bases de datos, actualizar tablas, eliminar tablas, y abrir archivos. También se utiliza para recuperar la versión, procesos e información de estado del servidor.

Utilidad para realizar funcionamientos administrativos. La sintaxis es:

shell> mysqladmin [OPCIONES] command [command-option] command...

Usted puede obtener una lista de opciones de tu versión del mysqladmin en el ejecutable mysqladmin--help.

El mysqladmin actual apoya los órdenes siguientes:

- ④ Create databasename: Crea una nueva base de datos.
- ④ Drop databasename: Elimina una base de datos y todas sus tablas.
- ④ Extended-status: Da un mensaje de estado extendido del servidor.
- ④ Flush-hosts
- ④ IFlush-logs
- ④ Flush-tables: Actualiza el contenido de todas las tablas.
- ④ Flush-privileges: Actualizan las tablas.
- ④ Kill id, id,...
- ④ Password new-password: Cambia la contraseña antigua por una nueva contraseña.
- ④ Ping: Verifican si el mysqld está funcionando.
- ④ Processlist: Muestran lista de las actividades realizadas en el servidor.
- ④ Reload: Actualiza tablas.
- ④ Refresh: Actualiza el contenido de todas las tablas y cierre y abre archivos.
- ④ Shutdown: Detiene el servidor.
- ④ Slave-start: Comienza la replicación de slave.
- ④ Slave-stop: Detiene la replicación de slave.
- ④ Status: Da un mensaje corto del estado del servidor.
- ④ Variables: Imprimen las variables disponibles.
- ④ Versión: Obtiene información de la versión del servidor.

Todos los comandos pueden acortarse por su único prefijo. Por ejemplo:

```
shell> mysqladmin proc stat
```

Tabla 2. Resultado de ejecución del comando

Id	User	Host	db	Command	Time	State	Info
6	monty	localhost		Processlist	0		

Uptime: 10077 Threads: 1 Questions: 9 Slow queries: 0 Opens: 6 Flush tables: 1
Open tables: 2 Memory in use: 1092K Max memory used: 1116K

Los estados del mysqladmin resultan según las columnas siguientes:

- ⌚ Uptime: Segundos que el servidor de MySQL ha sido puesto en marcha.
- ⌚ Threads: Numero de threads activos (los clientes).
- ⌚ Questions: Numero de preguntas de los clientes desde que el mysqld fue iniciado.
- ⌚ Slow queries: Preguntas que han tomado más de segundos del long_query_time.
- ⌚ Opens: cuántos tablas mysqld estan abiertas.
- ⌚ Flush tables: Numero de actualizaciones y recargas de órdenes.
- ⌚ Open tables: Numero de tablas que están abiertas en ese momento.
- ⌚ Memory in use: Memoria de uso asignada directamente por el código del mysqld (sólo disponible cuando MySQL se compila con --with--debug).
- ⌚ Max memory used: Memoria máxima asignada directamente por el código del mysqld (sólo disponible cuando MySQL se compila con --with--debug).

Si usted hace el cierre del mysqladmin en un socket (es decir, en un computador dónde el mysqld está ejecutando), mysqladmin esperará hasta que el pid-archivo



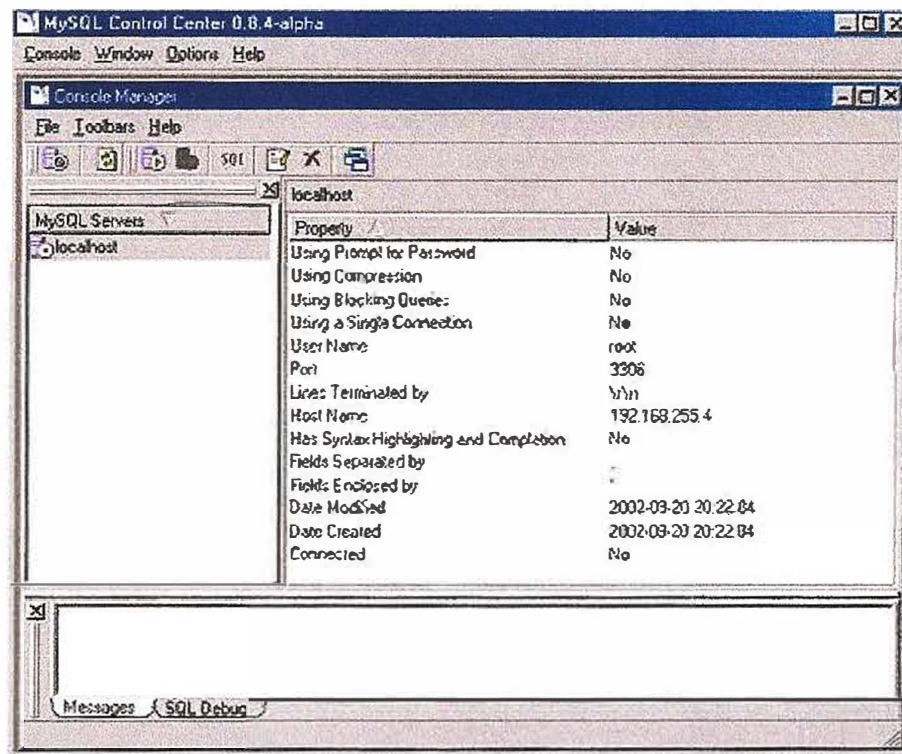
de MySQL se quite para asegurar que el servidor del mysqld se ha detenido propiamente.

14.9 MYSQL CONTROL CENTER

Es una consola de Administración de Mysql creada por Mysql AB desde la que se pueden administrar las bases de datos, los usuarios y el servidor de bases de datos.

MyCC es una potente consola de Administración para Mysql, que permite realizar todas las tareas de administración y trabajo de Mysql mediante un interface gráfica.

Figura 45. MySQL Control Center – Console Manager

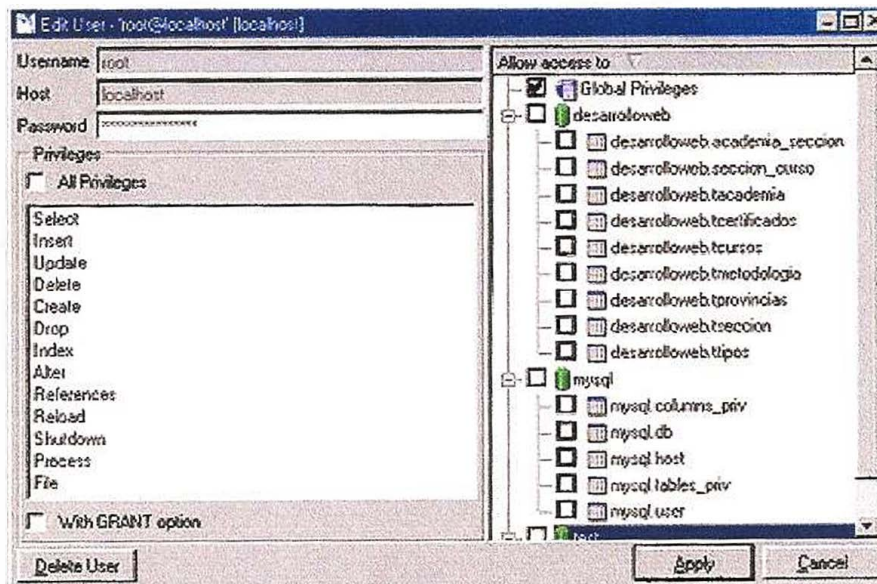


Existen actualmente versiones disponibles para Linux y para Windows. Para poder trabajar con un servidor de Mysql, primero lo tendrás que dar de alta mediante el icono de **Resgistrar servidor**. Aparece un cuadro de diálogo en el cual se pueden configurar todos los parámetros de acceso al servidor, como son nombre del servidor, nombre de usuario, contraseña, puerto etc. Una vez conectado a un servidor se podrá acceder a las tareas de administración de los usuarios de Mysql, administrador del gestor, y administración de las tablas.

14.9.1 Administrador de Usuarios

Desde el se pueden añadir, borrar y modificar las propiedades de los usuarios de Mysql. Al hacer click sobre un usuario aparece la ventana de configuración del usuario.

Figura 46. Administración de usuarios

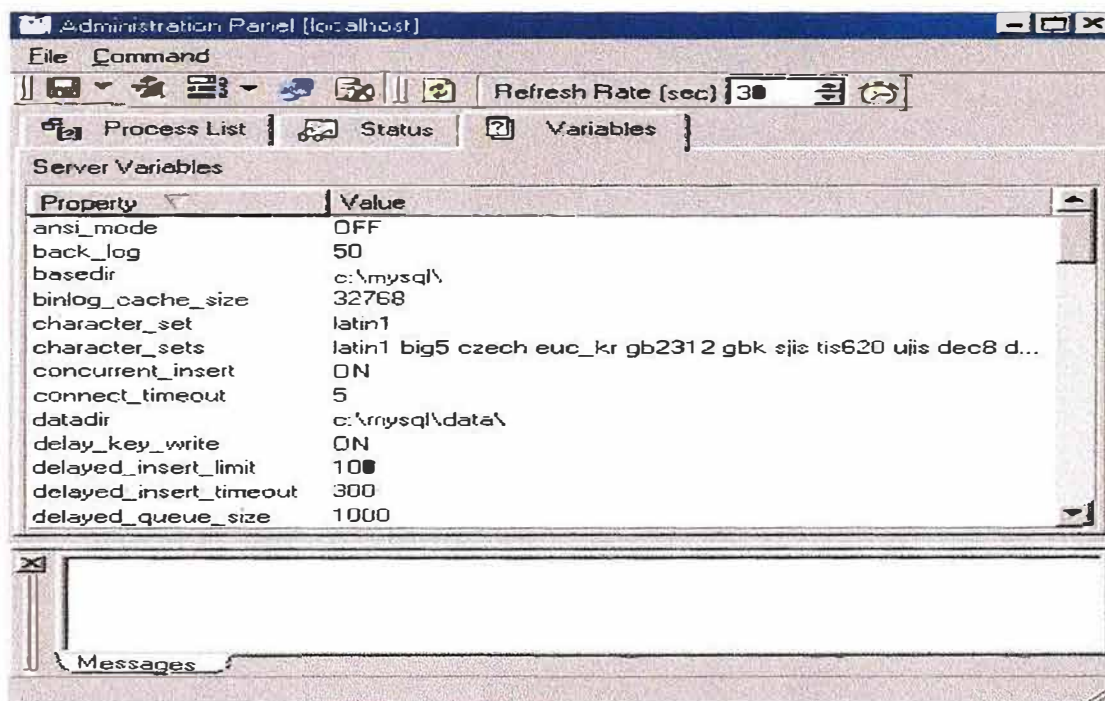


En ella se pueden configurar el nombre de usuario, la contraseña. Respecto a los permisos, permite establecer un sistema de permisos general o individualizado para cada una de las tablas. Además permite darle permiso al usuario para que se lo ceda a otros (GRANT OPTIONS). Uno de los puntos fuertes respecto a la seguridad es que al definir un usuario, hay que especificar el servidor desde el cual accederá, limitando de esta forma, los riesgos de acceso indebido.

14.9.2 Administración del Servidor

La ventana de Administración del servidor consta de tres pestañas desde la cual se pueden acceder a la lista de procesos, las variables de estado del servidor y las variables de configuración.

Figura 47. Administración del servidor

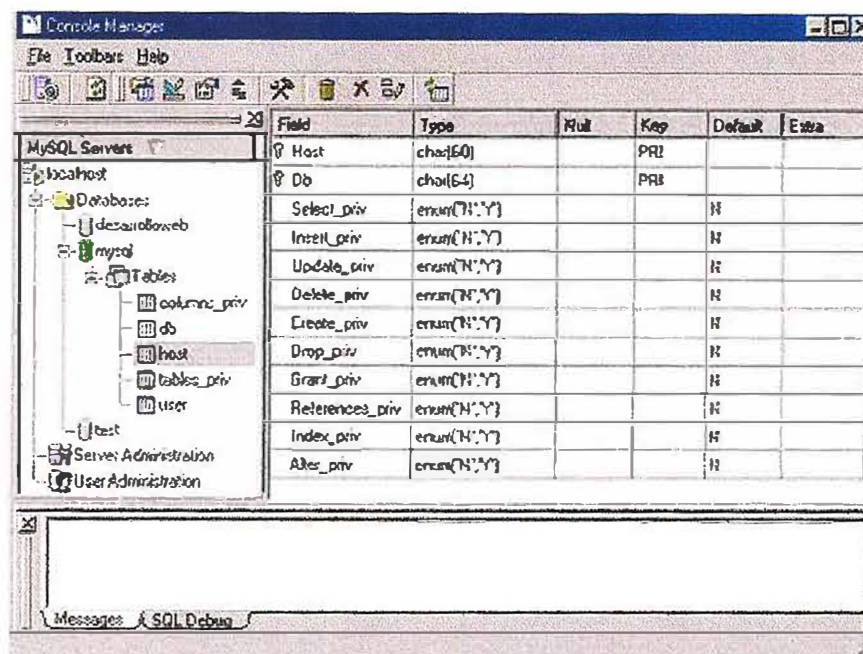


Desde esta ventana además se puede hacer un Ping al servidor, detenerlo, o guardar el contenido de las variables de estado. Desde la pestaña de procesos se puede ver el estado de cada uno de los procesos activos, viendo sus propiedades y con la opción de detenerlos mediante un Kill. Desde la pestaña de estado, se puede ver el contenido de cada una de las variables de estado, y se puede personalizar para mostrar simplemente las que te interesen. Desde la pestaña de variables se puede acceder al valor de cada una de las variables de configuración del servidor de Mysql.

14.9.3 Ventana de Base de Datos

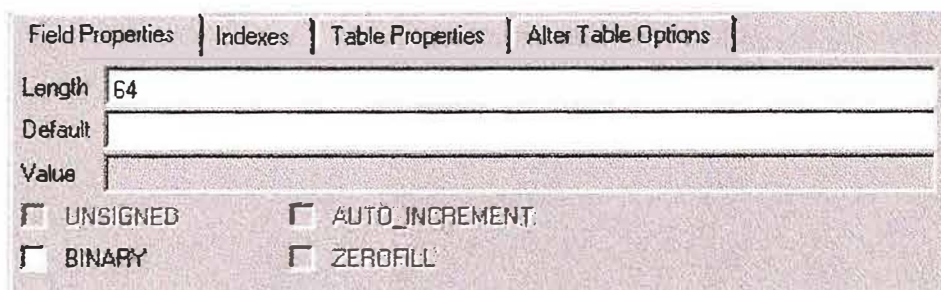
Al seleccionar una base de datos dentro de la sección de bases de datos, aparece la ventana de base de datos, en ella se puede ver información relacionada con la base de datos, como puede ser número de tablas, el número de consultas por segundo, el tiempo que lleva en funcionamiento etc. Al seleccionar una base de datos, se pueden ver los nombres de las tablas que contiene, el número de registros que contiene. Las tablas se pueden eliminar, vaciar y renombrar. Cuando se selecciona una tabla, se pueden ver el nombre de los campos, los tipos de datos y los parámetros opcionales de cada uno de ellos.

Figura 48. Ventana de Base de datos



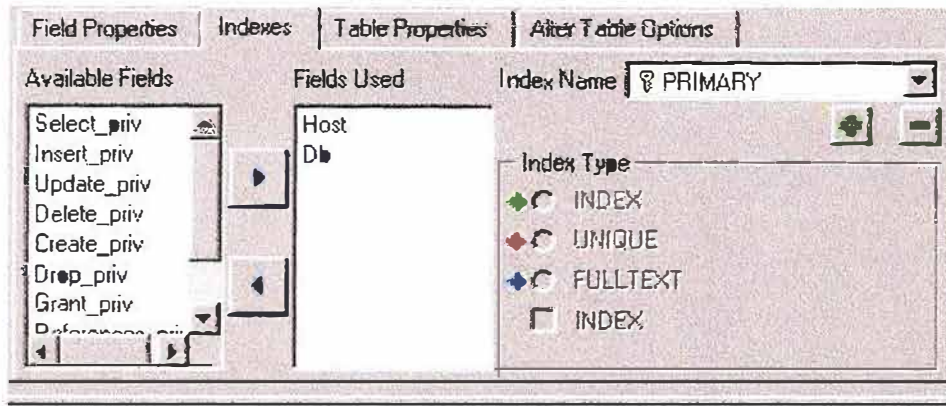
Esta ventana permite crear nuevos campos, eliminarlos o modificar sus propiedades. Además mediante el icono de herramientas, se puede analizar la tabla, optimizarla y repararla. Al hacer doble clic sobre una tabla se muestran en la ventana los datos que contiene. Haciendo clic sobre cualquiera de los campos, se accede a la estructura de la misma. Para cada campo se puede elegir el tipo de datos, el tamaño, si es clave o no, el valor predeterminado del mismo.

Figura 49. Field Properties



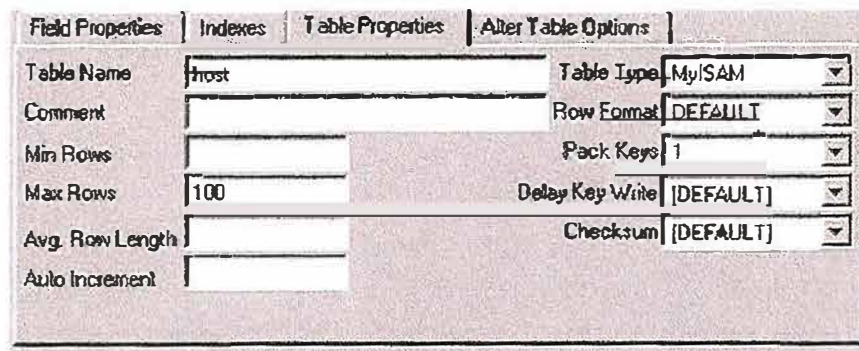
Dentro de la tabla, existe un cuadro para configurar los índices que tiene.

Figura 50. Indexes



Desde ella se pueden seleccionar los campos que lo contienen, así como el tipo de índice que se desea crear. Desde la pestaña de propiedades de la tabla se acceden a los atributos de la misma, como pueden ser el tipo de tabla, el modo de escritura, el tamaño máximo de cada fila etc.

Figura 51. Table Properties



14.9.4 Copias de Seguridad en MySQL

El gestor de Bases de datos MySQL incluye varias herramientas para la realización de copias de seguridad de la base de datos. Mediante ellas podremos poner a salvo nuestros datos, para que, en el caso de que se pierdan, poderlos recuperar.

A la hora de hacer una copia de seguridad, lo primero que se hay que tener en cuenta es la integridad de los datos que se estén guardando. En todos los casos es necesario que haya integridad en los datos de una tabla, con esto quiero decir que todos los datos de la tabla deberán estar escritos en la misma, Mysql dispone de diversas "caches" en las que se almacenan datos temporalmente con el objetivo de mejorar en rendimiento, de forma que por ejemplo, una vez hecha una modificación en una tabla, puede ser que los datos no se guarden inmediatamente en disco, hasta que termine, por ejemplo, una consulta que se estaba ejecutando. Por esto, es necesario "forzar" a Mysql a escribir todos los datos en el disco, mediante la sentencia ***Flush Tables***.

Además es necesario que no se escriba en las tablas mientras se esta haciendo la copia de seguridad de la base de datos, que se consigue con el comando "lock tables", seguido del nombre de la tabla. Puede haber bases de datos en las que sea necesario bloquear todas las tablas al mismo tiempo antes de hacer la copia de seguridad.

Existen varias opciones para realizar la copia de seguridad de una base de datos de Mysql:

- 🕒 En primer lugar, se podría utilizar alguna herramienta comercial que gestione todo el proceso, esto tiene como ventaja la simplicidad del método, y como

inconveniente, que no suelen ser gratis, sino que hay que pagar licencia por la utilización de los mismos. Estas herramientas no se cubrirán en este reportaje.

- ④ En segundo lugar, y a partir de la versión 3.23.25 y posteriores, existe la posibilidad de realizar una copia de seguridad a través de la sentencia sql "backup table".
- ④ Como tercera opción, es posible realizar copias de seguridad a través de las herramientas que nos proporciona el propio gestor de base de datos, como pueden ser mysqldump ó mysqlhotcopy.

15.9.4.1 Backup de los datos (backup table)

Este comando nos permite hacer una copia de los ficheros de las tablas de las cuales queremos hacer un backup, actualmente solo funciona con tablas de tipo MyIsam, y copia tanto los ficheros .frm que contienen la definición de la tabla, como los ficheros .myd, que contienen los datos.

Antes de ejecutarse, guarda todos los cambios que pudiera haber en memoria de la tabla, de forma que quede de una manera consistente. Asimismo, durante la ejecución del comando, bloquea la tabla sobre la que se está haciendo la copia de seguridad para que los datos sean consistentes en la tabla. Hay que tener en cuenta que este comando va bloqueando una a una las tablas, según va haciendo la copia de seguridad. Esto podría provocar inconsistencia de datos a nivel de base de datos, si es necesario hacer un backup de todas las tablas en un instante, en cuyo caso habría que utilizar el comando "lock tables" para bloquear todas las tablas antes de comenzar la copia de seguridad.

Al finalizar el comando devuelve una tabla, que contiene los campos:

Tabla 3. Resultado comando Backup table

Columna	Valores
Table	Nombre de la tabla
Op	Siempre pone "backup"
Msg_type	Puede contener status, error, info o warning.
Msg_text	Mensaje descriptivo del resultado de la operación

Y en la que hay un registro por cada tabla que sobre la que se ha hecho backup.

Para realizar una copia de una tabla llamada pedidos, a la carpeta Backups:

BACKUP TABLE pedidos TO '/backups/'

15.9.4.2 Recuperación de datos

Para complementar la sentencia "**Backup table**", Mysql dispone de la sentencia "**Restore table**", que permite restaurar una tabla a partir de una copia de seguridad que se ha realizado con el comando "Backup Table".

Esta opción solo se puede utilizar si la tabla que se pretende restaurar no se encuentra en la base de datos, ya que en caso afirmativo mostrará un mensaje de error. Al igual que el comando de Backup, esta opción sólo esta disponible para las tablas de tipo Mysql, y a partir de la versión 3.23.25.

Como el comando de backup, no copia los ficheros de índices, el comando para recuperar los datos, vuelve a reindexar todos los campos que contienen índices, creando los ficheros correspondientes. Al igual que el comando para hacer copias de seguridad, invocar al comando "Restore table" devuelve una tabla, con un

registro por cada tabla sobre la que se ha hecho la base de datos, y que contiene la siguiente información:

Tabla 4. Resultado comando Restore table

Columna	Valores
Table	Nombre de la tabla
Op	Siempre pone "restore"
Msg_type	Puede contener status, error, info o warning.
Msg_text	Mensaje descriptivo del resultado de la operación

Para restaurar una tabla llamada pedidos de la carpeta Backups a la base de datos:

```
RESTORE TABLE pedidos FROM `/backups/`
```

15.9.4.3 Backup MySQL con mysqldump

El comando **mysqldump** del sistema gestor de base de datos MySQL sirve para hacer copias de seguridad.

Este comando permite hacer la copia de seguridad de una o múltiples bases de datos. Además permite que estas copias de seguridad se puedan restaurar en distintos tipos de gestores de bases de datos, sin la necesidad de que se trate de un gestor de mysql. Esto lo consigue creando unos ficheros, que contienen todas las sentencias sql necesarias para poder restaurar la tabla, que incluyen desde la sentencia de creación de la tabla, hasta una sentencia insert por cada uno de los registros que forman parte de la misma.

El comando dispone de una amplia variedad de opciones que nos permitirá realizar la copia de la forma más conveniente para el propósito de la misma. Para poder restaurar la copia de seguridad, bastará con ejecutar todas las sentencias sql que se encuentran dentro del fichero, bien desde la línea de comandos de mysql, o desde la pantalla de creación de sentencias sql de cualquier entorno gráfico como puede ser el Mysql Control Center.

Las limitaciones de la restauración dependerán de las opciones que se han especificado a la hora de hacer la copia de seguridad, por ejemplo, si se incluye la opción `--add-drop-table` al hacer la copia de seguridad, se podrán restaurar tablas que existen actualmente en el servidor (borrándolas primero). Por lo que es necesario estudiar primero los procedimientos que se utilizarán tanto en la copia como en la restauración, para que todo salga correcto.

Algunas de las opciones que tiene son:

--add-locks

Añade LOCK TABLES antes, y UNLOCK TABLE despues de la copia de cada tabla.

--add-drop-table

Añade un drop table antes de cada sentencia create

-A,

--all-databases

Copia todas las bases de datos. Es lo mismo que utilizar `--databases` seleccionando todas.

-a,

--all

Incluye todas las opciones de creación específicas de Mysql.

--allow-keywords

Permite la creación de nombres de columnas que son palabras clave, esto se realiza poniendo de prefijo a cada nombre de columna, el nombre de la tabla.

-c,**--complete-insert**

Utiliza inserts incluyendo los nombres de columna en cada sentencia (incrementa bastante el tamaño del fichero).

-C,**--compress**

Comprime la información entre el cliente y el servidor, si ambos soportan compresión.

-B,**--databases**

Para copiar varias bases de datos. En este caso, no se especifican tablas. El nombre de los argumentos se refiere a los nombres de las bases de datos. Se incluirá USE db_name en la salida antes de cada base de datos.

--delayed

Inserta las filas con el comando INSERT DELAYED.

-e,**--extended-insert**

Utiliza la sintaxis de INSERT multilinea. (Proporciona sentencias de insert más compactas y rápidas.)

-#,**--debug[=option_string]**

Utilización de la traza del programa (para depuración).

--help

Muestra mensaje de ayuda y termina.

--fields-terminated-by=...
--fields-enclosed-by=...
--fields-optionally-enclosed-by=...
--fields-escaped-by=...
--lines-terminated-by=...

Estas opciones se utilizan con la opción -T y tienen el mismo significado que la correspondiente cláusula LOAD DATA INFILE.

-F, **--flush-logs**
Escribe en disco todos los logs antes de comenzar con la copia.

-f, **--force,**
Continúa aunque se produzca un error de SQL durante la copia.

-h, **--host=..**
Copia los datos del servidor de Mysql especificado. El servidor por defecto es localhost.

-l, **--lock-tables.**
Bloquea todas las tablas antes de comenzar con la copia. Las tablas se bloquean con READ LOCAL para permitir inserts concurrentes en caso de las tablas MyISAM. Cuando se realiza la copia de múltiples bases de datos, --lock-tables bloqueará la copia de cada base de datos por separado. De forma que esta opción no garantiza que las tablas serán consistentes lógicamente entre distintas bases de datos. Las tablas en diferentes bases de datos se copiarán en estados completamente distintos.

-K, **--disable-keys**
Se incluirá en la salida `/*!40000 ALTER TABLE tb_name DISABLE KEYS */;` y `/*!40000 ALTER TABLE tb_name ENABLE KEYS */;` Esto hará que carga de datos

en un servidor MySQL 4.0 se realice más rápido debido a que los índices se crearán después de que todos los datos hayan sido restaurados.

-n, --no-create-db

No se incluirá en la salida CREATE DATABASE /*!32312 IF NOT EXISTS*/ db_name; Esta línea se incluye si la opción --databases o --all-databases fue seleccionada.

-t, --no-create-info

No incluirá la información de creación de la tabla (sentencia CREATE TABLE).

-d, --no-data

No incluirá ninguna información sobre los registros de la tabla. Esta opción sirve para crear una copia de sólo la estructura de la base de datos.

--opt

Lo mismo que --quick --add-drop-table --add-locks --extended-insert --lock-tables. Esta opción le debería permitir realizar la copia de seguridad de la base de datos de la forma más rápida y efectiva.

-pyour_pass, --password[=your_pass]

Contraseña utilizada cuando se conecta con el servidor. Si no se especifica, `=your_pass', mysqldump preguntará la contraseña.

-P, --port=...

Puerto utilizado para las conexiones TCP/IP.

--protocol=(TCP | SOCKET | PIPE | MEMORY)

Especifica el protocolo de conexión que se utilizará.

-q, **--quick**

No almacena en el buffer la sentencia, la copia directamente a la salida. Utiliza `mysql_use_result()` para realizarlo.

-Q, **--quote-names**

Entre comilla las tablas y nombres de columna con los caracteres.

-r, **--result-file=...**

Redirecciona la salida al fichero especificado. Esta opción se debería utilizar en MSDOS, porque previene la conversión de nueva línea ``n'` en nueva línea y retorno de carro ``n\r'`.

--single-transaction

Utiliza el comando `BEGIN` antes de realizar la copia desde el servidor. Es muy útil con las tables InnoDB y el nivel de transacción `READ_COMMITTED`, porque en este modo realizará la copia de seguridad en un estado consistente sin necesidad de bloquear las aplicaciones. Consultar el manual para más detalles.

-S **/path/to/socket,** **--socket=/path/to/socket**

El fichero de sockets que se especifica al conectar al localhost (que es el host predeterminado).

--tables

sobreescribe la opción `--databases (-B)`.

-T, **--tab=path-to-some-directory**

Crea un fichero `table_name.sql`, que contiene la sentencia de creación de SQL, y un fichero `table_name.txt`, que contiene los datos de cada tabla. El formato del fichero ``.txt'` se realiza de acuerdo con las opciones `--fields-xxx` y `--lines--xxx options`. Nota: Esta opción sólo funciona si el comando `mysqldump` se ejecuta en

la misma máquina que el demonio mysqld, el usuario deberá tener permisos para crear y escribir el fichero en la ubicación especificada

```
-u nombre_usuario, --user=nombre_usuario
```

El nombre de usuario que se utilizará cuando se conecte con el servidor, el valor predeterminado es el del usuario actual.

-v, --verbose

Va mostrando información sobre las acciones que se van realizando (más lento).

-w, **--where='cláusula** **where'**

Sirve para realizar la copia de determinados registros

```
-X,                                     --xml
```

Realiza la copia de seguridad en un documento xml

-x, --first-slave

Bloquea todas las tablas de todas las bases de datos.

Ejemplo:

Para realizar la copia se seguridad de la base de datos mibase al fichero copia_seguridad.sql

```
mysqldump --opt mibase > copia_seguridad.sql
```

15. ORACLE

15.1 HISTORIA DE ORACLE

El manejador de Base de datos ORACLE, surgió a final de los años 70 y principio de los años 80. Lawrence J. Ellison, Robert N. Miner y Edward Oates fundaron System Development Laboratories en 1977 con el objetivo de desarrollar programas de bases de datos para equipos informáticos. Ellison había trabajado en una compañía que desarrolló el primer mainframe compatible con los equipos IBM (International Business Machines Corporation). Ya en la nueva compañía, Ellison observó que IBM desarrollaba bases de datos llamadas bases de datos relacionales e intentó desarrollar sus propias versiones. System Development Laboratories derrotó a IBM en el mercado en 1979 con Oracle RDBMS (sistema de administración de base de datos relacional), el primer producto comercial que utilizó SQL (Structured Query Language, lenguaje de consulta estructurado). En la actualidad, todos los productos para bases de datos utilizan SQL como patrón.

Oracle conocida entonces como Relational Software, tenía poco más de 25 empleados en aquel tiempo y solo unos pocos clientes importantes. Sin embargo, cuando se completó el estudio, Oracle fue declarada vencedora. George afirmó que el SGBD Oracle era técnicamente el mejor producto del mercado. Estas declaraciones fueron hecha en una época en la que muy poca gente conocía el significado del término "Relacional", y los que lo conocían (o creían conocerlo) no tenían muchas cosas favorables que decir de él.

La compañía de Oracle Corporation estaba trabajando entonces para perfeccionar su joven producto, para comprender los tipos de características y funcionalidad que podría hacerlo útil y productivo en el mundo de los negocios. El esfuerzo

contribuyó a su refinamiento. Algunas de las características de Oracle, tales como las salidas de SQL*FORMS fueron el resultado de dicho esfuerzo.

Oracle Corporation, es una de las mayores compañías de software del mundo y principal distribuidora de sistemas de gestión de bases de datos. Los programas de bases de datos de Oracle permiten a las grandes empresas tener acceso y almacenar grandes cantidades de información, desde informes de ventas y listas de clientes hasta reservas aéreas, en diferentes tipos de sistemas operativos. Oracle también fabrica herramientas para programadores de software, así como aplicaciones para gestión de información en servidores. La sede central de la compañía se encuentra en Redwood Shores, California.

En 1982 System Development Laboratories cambió su nombre a Oracle, en honor a su exitoso producto para bases de datos. Oracle se convirtió enseguida en el líder del sector de gestión de bases de datos y firmó contratos con algunas de las compañías más importantes de Estados Unidos y de todo el mundo. Durante la década de 1980 las ventas se doblaron año tras año, pasando de 2,5 millones de dólares estadounidenses en 1982 a 916 en 1990.

En 1991 la compañía tuvo pérdidas por primera vez y se vio forzada a despedir a cientos de empleados y a reorganizar su sistema de administración. En 1992 remontó la crisis y volvió a ser rentable.

Posteriormente, Oracle comenzó a desarrollar hardware y software para controlar enormes bases de datos de vídeo, audio y texto, y a distribuir datos a través de redes globales. La compañía adquirió nCube Corporation, una empresa con sede en California, pionera en supercomputadoras, que utilizaba una técnica llamada procesamiento en paralelo. Esta compra permitió a Oracle combinar su software con hardware capaz de gestionar las enormes cantidades de datos necesarias para administrar bases de datos multimedia.

En 1996 Oracle comenzó a promover y desarrollar una plataforma de software para un equipo de bajo coste diseñado para acceder a Internet y funcionar de forma independiente al sistema operativo Windows de Microsoft Corporation. Esta iniciativa se enfrentó a la competencia de Sun Microsystems, Inc., IBM y otros fabricantes de equipos informáticos. Al año siguiente Oracle creó la división Network Computer Inc. (en la actualidad Liberate Technologies) para sacar al mercado su equipo de bajo coste.

15.2 ¿QUÉ ES ORACLE?

Oracle es una de las Bases de Datos Relacionales más extendidas en las empresas del mundo, y con mayor prestigio, basado en los 25 años de experiencia en implantaciones de esta base de datos. Es un manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware para garantizar su aprovechamiento al máximo en ambientes cargados de información. Es el conjunto de datos que proporciona la capacidad de almacenar y acude a estos de forma consecuente con un modelo definido como relacional. Además es una suite de productos que ofrece una gran variedad de herramientas.

Oracle es el mayor y más usado Sistema Manejador de Base de Dato Relacional (RDBMS) en el mundo. La Corporación Oracle ofrece este RDBMS como un producto incorporado a la línea de producción. Funciona en plataformas Windows, así como en entornos UNIX - LINUX, ofreciendo un sistema de acceso Cliente/Servidor. Actualmente Oracle se instala como Base de Datos de Gestión de empresa, y habitualmente con desarrollo de aplicaciones Developer 2000, así como para dar servicio de datos en ámbitos Internet/Extranet/Internet. Es este último aspecto, en el que la Corporación Oracle más está trabajando, ofreciendo

herramientas con soluciones e-business y e-commerce, soportados por esta base de datos.

Entre las características de Oracle destacamos su **escalabilidad** y alta **disponibilidad**, aportando un sistema de administración completo para gestionar todas las situaciones críticas de una Base de Datos de estas características: Sistema de seguridad basados en usuarios, grupos y roles, alertas, backups y restauración de datos, auditorías, etc. Oracle, desde su versión Oracle 9i, hace especial hincapié en la seguridad y robustez de su base de datos, óptimo rendimiento y mejoras en sus mecanismos de concurrencia (número elevado de usuarios accediendo a la base de datos).

Es el conjunto de datos que proporciona la capacidad de almacenar y acude a estos de forma consecuente con un modelo definido como **relacional**. Además es una suite de productos que ofrece una gran variedad de herramientas. Además incluye cuatro generaciones de desarrollo de aplicación, herramientas de reportes y utilitarios. Oracle corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo. Soporta unos 17 idiomas, corre automáticamente en más de 80 arquitectura de hardware y software distintos sin tener la necesidad de cambiar una sola línea de código. Esto es porque más el 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos. Su evolución va desde las versiones 5, 6, 7,8 y 9.

15.3 VERSIONES DE ORACLE

- 🕒 **Oracle 5 y Oracle 6:** fueron las dos primeras versiones de Oracle, quedando aun rezagadas por las versiones sucesoras.
- 🕒 **Oracle 7:** La base de datos relacional componentes de Oracle Universal Server. Posee además las versiones 7.1, 7.1.2, y 7.1.3.

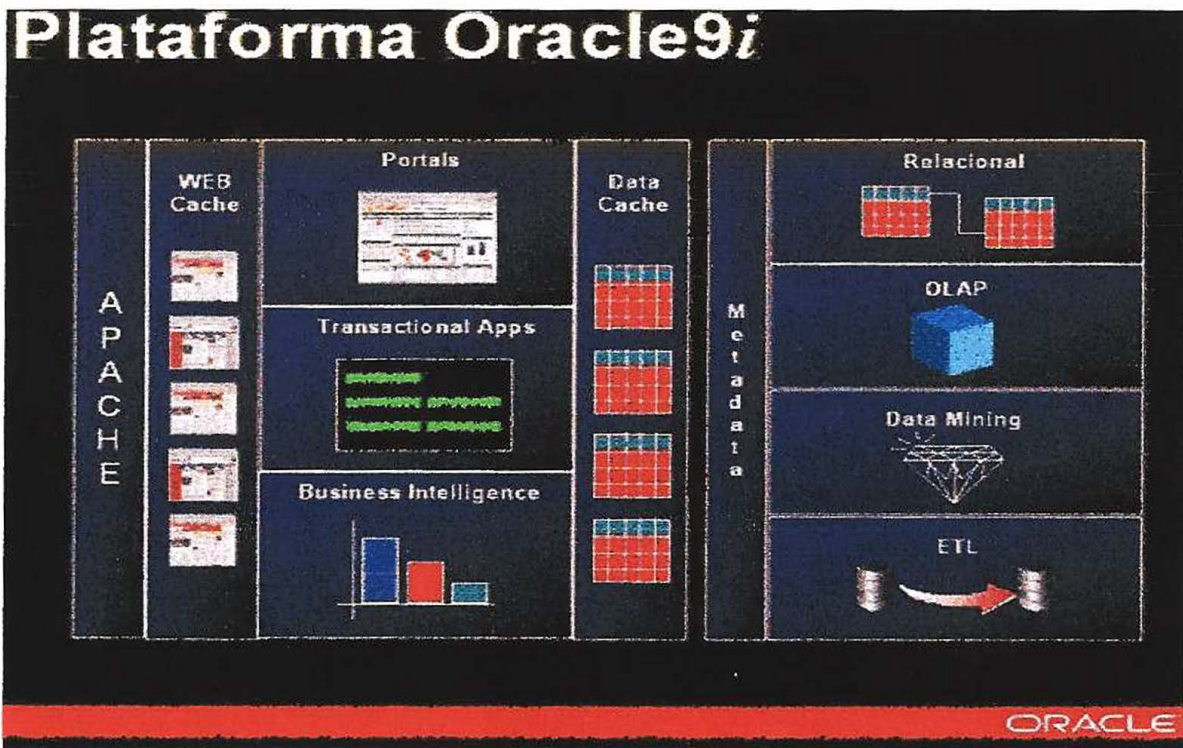
- ④ **Oracle 7 Parallel:** Ofrece a los usuarios un método seguro y administrable para incrementar la performance de sus bases de datos existentes introduciendo operaciones en paralelo y sincrónicas dentro de sus ambientes informáticos.
- ④ **Oracle 8:** Incluye mejoras de rendimiento y de utilización de recursos. Independiente de que se necesite dar soporte a decenas de miles de usuarios y cientos de terabytes de datos, o se disponga de un sistema mucho más pequeño, pero igualmente crítico, todos se benefician del rendimiento de Oracle8. Este soporta aplicaciones de procesamiento de transacciones on line (OLTP) y de data warehousing mayores y más exigentes.
- ④ **Oracle Fail Safe:** Protege al sistema de caída de entornos Cluster Windows NT. Este producto es para que el sistema operativo Windows NT disponga de una mayor oferta de soluciones en entornos cluster.
- ④ **Oracle Universal Server:** Con soporte completo de Web, mensajería y datos multimedia, Oracle Universal Server es el eslabón fundamental en el camino de transición de la industria de la tecnología a la informática de red, donde la complejidad de software es transferida desde la PC de escritorio a poderosos servidores y redes.
- ④ **Oracle 9i** es una base de datos adaptable a sus necesidades. Oracle 9i con su base de datos viene en 3 versiones, y ofrece además una gran cantidad de herramientas que ayudan a desarrollar el manejo de la información.
- ④ **Oracle 9i Standard Edition** es fácil de usar y tiene un muy buen precio especialmente diseñada para grupos de trabajo como departamentos y aplicaciones de Internet/Intranet. Incluye una serie de herramientas para su fácil distribución, replicación y manejo por el Web. Incluye todo lo necesario para construir aplicaciones críticas de negocios.
- ④ **Oracle9i Enterprise Edition** nos da eficiencia, confiabilidad, y alta seguridad en aplicaciones de alto volumen como: on-line transaction processing (OLTP), búsqueda intensiva de información en base a preguntas (query-intensive data warehouses), y aplicaciones que demandan el uso intensivo de

base de datos por Internet. Esta edición cumple con requerimientos de datos de corporativos y sus aplicaciones.

- **Oracle9i Personal Edition** _ soporta desarrollo de un solo usuario que requiere ser 100% compatible con Oracle9i Standard Edition and Oracle9i Enterprise Edition. . Con toda la funcionalidad en una sola computadora Workstation

La siguiente figura nos muestra la plataforma de Oracle 9i:

Figura 52. Plataforma Oracle 9i



15.4 EL KERNEL DE ORACLE

El Kernel es el corazón del RDBMS Oracle, el cual maneja las siguientes tareas:

- Manejar el almacenamiento y definición de los datos.

- Ⓢ Suministrar y limitar el acceso a los datos y la concurrencia de los usuarios.
- Ⓢ Permitir los backup y la recuperación de los datos.
- Ⓢ Interpretar el SQL y PL/SQL.

El Kernel Oracle es cargado a la memoria al inicio de las operaciones y es usado por cada base de datos existente en el equipo.

15.5 ALGUNAS DE LAS HERRAMIENTAS DE ORACLE

- Ⓢ **SQLForms**: es la herramienta de Oracle que permite, de un modo sencillo y eficiente, diseñar pantallas para el ingreso, modificaciones, bajas y consultas de registros. El usuario podrá, una vez definida la forma, trabajar con ella sin necesidad de generar códigos, dado que Oracle trae incorporado un conjunto de procedimientos y funciones asociados a las teclas de funciones, como por ejemplo la tecla [F7], que se usa para iniciar una consulta.
- Ⓢ **SQLReport** de Oracle realiza de forma flexible, sencilla y eficiente la creación de reportes, informes o listados permitiendo, entre otras facilidades, la visualización previa por pantalla con una gran variedad en estilos de presentación.
- Ⓢ **SQLMenu**: es el producto de ORACLE destinado a producir árboles de menús que permiten al usuario desplazarse fácilmente a través de su aplicación, facilitando a su vez, la integración con los diferentes módulos de ORACLE.

15.6 VENTAJAS

- Ⓢ Oracle posee igual interacción en todas las plataformas (Windows, Unix, Macintosh y Mainframes). Esto porque más del 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de Sistemas Operativos.
- Ⓢ Oracle soporta bases de datos de todos los tamaños, desde severas cantidades de bytes y gigabytes en tamaño.

- ☉ Oracle provee salvar con seguridad de error lo visto en el monitor y la información de acceso y uso.
- ☉ Oracle soporta un verdadero ambiente cliente servidor. Este establece un proceso entre bases de datos del servidor y el cliente para la aplicación de programas.

15.7 COSTOS

Herramienta que se consigue por no menos de US\$ 1000 y hay que pagar US\$ 195 por usuario que la use (mínimo 5 usuarios)

Oracle 9i Standard Edition: Desde US\$3384 para 10 usuarios

Oracle9i Enterprise Edition: Desde precios de usuarios a precios por procesador con usuarios ilimitados

Oracle9i Personal Edition: Desde una sola licencia

15.8 INSTALACIÓN

15.8.1 Proceso de Pre-Instalación

El primer paso es determinar el tamaño del software de instalación. Esto no incluye el espacio requerido para la producción del sistema o el espacio para el desarrollo de aplicaciones del o los sistemas Oracle.

15.8.2 Productos y Tamaño

Oracle RDBMS 11.6 MB

SQL*PLUS 1.6 MB

SQL*FORMS 2.4 MB

SQL*MENU 1.8 MB

SQL*REPOT WRITER 2.1 MB

Pro*C, Pro*Fortran, Pro*Cobol 1.3 MB

NLS 2.1 MB

Archivos de Oracle RDBMS 2.8 MB

SQL*Net 2.8 MB

Oracle Demo Database 5.9 MB

El próximo paso es determinar la localización del producto y las aplicaciones que soportan el nuevo RDBMS Oracle, así como el espacio a ser reservado para los propios objetos de la base de datos.

Oracle soporta dos tipos de almacenamiento, por carácter (RAW) o por bloques (Files System), generalmente es recomendable que los sean colocados en Raw Device. El **Raw Device** es un dispositivo de caracteres disponibles en algunos sistemas operativos el cual es asignado directamente a Oracle.

Oracle corre más rápidamente con Raw Device que con Files System, por varias razones:

- ⌚ E I/O (Input/Output) es realizado directamente en el disco por Oracle, independientemente del sistema operativo.
- ⌚ El buffer cache del sistema operativo es dejado a un lado.
- ⌚ Los buffers del sistema operativo y de oracle son independiente entre sí.

Con la intención de evitar la contención de los discos, se debe considerar la instalación de Oracle en dispositivos separados, especialmente si se tienen varios discos, y más esencialmente, si se poseen más de una controladora de disco. La planeación debe realizarse teniendo en cuenta los siguientes criterios:

- ⌚ Los Files System y sus dispositivos asignados.
- ⌚ El swapping y paginamiento en Oracle, deberán estar en los dispositivos más rápidos.
- ⌚ Los tablespaces para tables e índices en dispositivos separados.

- ⓐ Los Log Files en un dispositivo separado al del tablespace de RDBMS Oracle.

15.8.3 Requerimientos

Estación de Trabajo:

- ⓐ Procesador Pentium 166 MHz o Superior.
- ⓐ Minimo 128 Mb RAM*, 1 Gb Disco Duro
- ⓐ Entre 298 y 587 Mb disponible de espacio en disco, dependiendo de la opcion que escoja
- ⓐ Sistema Operativo: Windows NT 4.0, Windows 2000, Windows98 or Windows95
- ⓐ Requiere 256 Mb RAM si está usando utilidades JAVA (Compare, Merge, Version History Viewer, Version Event Viewer, Dependency Manager)
- ⓐ Para **Windows**, Podrias necesitar una cuenta de Servidor.

Servidor:

- ⓐ Minimo 64 Mb RAM, de los cuales 32 Mb estarán disponibles para el System Global Area (SGA)
- ⓐ Para cada Repositorio: Aproximadamente 140 Mb en el SYSTEM tablespace para los paquetes del repositorio, procedimientos and vistas entre 20 y 325 Mb en otras tablespaces, para los datos del repositorio.

15.8.4 Instalación de Oracle

- ⓐ Instalar desde la cuenta de Administrador
- ⓐ Comprobar los permisos desde el Administrador de Usuarios
- ⓐ Tal vez Oracle intente detener algunos servicios. Tal vez haya que detenerlos manualmente (**Panel de Control - Servicios**)

- ⓐ Si hay problemas en acceder a los iconos del panel de control, también se puede llegar a ellos mediante:

C:\WINNT\System32*.cpl

- ⓐ Si de todas formas hay problemas en arrancar y tirar servicios, también se puede actuar sobre ellos mediante:

C:\>net Stara

C:\>net start servicio

C:\>net stop servicio

15.8.5 Desinstalación de Oracle

- ⓐ Desinstalar con Oracle Installer (Software Asset Manager)
- ⓐ Oracle intentará detener los servicios. Tal vez haya que detenerlos manualmente (***Panel de Control - Servicios***)
- ⓐ Si la desinstalación falla: Eliminar los servicios de Oracle del Panel de Control - Servicios. Por ejemplo:

c:\orant\bin\oradim80 -delete -sid grop

Para eliminar otros servicios, por ejemplo:

c:\orant\bin\oradim80 -delete -srv OracleStartGROP

c:\orant\bin\oradim80 -delete -srv OracleWebAssistant

Si no se pueden eliminar, al menos deshabilitarlos

La ventana de servicios se ha de cerrar y volver a abrir para ver los cambios (en ciertos casos reiniciar el equipo ¿?)

Borrar c:\orant (a ser posible antes de esto se deben haber eliminado los servicios)

Entrar en regedit (*Inicio-Ejecutar-regedit*) y borrar

MiPC\HKey_local_machine\Software\Oracle

15.8.6 Arranque de una BD Oracle

Tabla 5. Arranque de una Base de datos Oracle

	Arrancar una instancia de BD	Montar una BD	Abrir una BD
Fase	Nomount	Mount	Open
3			startup
2+1		startup mount	alter database open;
1+1+1	startup nomount	alter database mount;	alter database open;
Al finalizar la fase, el significado del estado es	Decir que la la instancia está arrancada significa que existen procesos y recursos que van a ser asociados a los ficheros de esa BD	Decir que la base de datos está montada significa que los procesos y recursos de la instancia se asocian a los ficheros de la BD física	Decir que la base de datos está abierta significa que está disponible a los usuarios, y que se han abierto todos los archivos log.
Pasos que se realizan	<ul style="list-style-type: none"> ④ Se lee el archivo init<sid>.ora ④ Se identifican los controlfiles ctl<sid>.ora ④ Se crea e inicializa la SGA usando init<sid>.ora ④ Se arrancan los procesos batch de 	<ul style="list-style-type: none"> ④ Se localizan los datafiles (*.dat, *.dbf) ④ Se abren y leen los control files (archivos de control, ctl<sid>.ora) ④ Se "monta" la BD (se obtiene <i>instance lock</i>) ④ Se determina el modo de apertura de la BD (Normal o <i>crash recover</i>) 	<ul style="list-style-type: none"> ④ Se abren y bloquean los datafiles (*.dat, *.dbf) ④ Si es la primera instancia, se obtiene el <i>startup lock</i> ④ Abre los online redo logs ④ Si es la primera instancia, se realiza <i>crash</i>

	background		<i>recovery</i> si es necesario
Ejemplos de operaciones a realizar en este estado		<p>Cualquier sentencia <i>alter database</i> puede ser ejecutada si la base de datos está montada, y algunas de ellas, sólo si está montada. Montada y no abierta permite:</p> <ul style="list-style-type: none"> Ⓢ Acciones de archivo redo log Ⓢ Cambio del modo de archivo (Alternar entre los modos <i>archivelog</i> y <i>noarchivelog</i>) Ⓢ Recuperación Ⓢ Renombrado de archivos Ⓢ Con <i>media recovery</i>, recuperar archivos dañados Ⓢ Poner un datafile online o offline Ⓢ Mover datafile y redo log files 	<ul style="list-style-type: none"> Ⓢ Accesos normales de los usuarios
Razones por las que puede fallar cada fase	<ul style="list-style-type: none"> Ⓢ El ejecutable del sgbd está corrupto Ⓢ Init<sid>.ora es léxicamente o sintácticamente incorrecto, hay errores en la especificación de sus parámetros Ⓢ Las configuraciones especificadas en 	<ul style="list-style-type: none"> Ⓢ No existen los control files o están corruptos (archivos de control, ctl<sid>.ora). En ese caso se deben recuperar de un backup o recrearlos mediante el comando create controlfile Ⓢ Los ficheros de control no están sincronizados. La solución es usar el más reciente. 	<ul style="list-style-type: none"> Ⓢ Alguno de los datafiles no existen como ficheros del sistema operativo Ⓢ Otro tipo de inconsistencias en la información contenida en el diccionario de datos Ⓢ No se encuentra alguno de los archivos de redo log

	init<sid>.ora no son posibles (errores semánticos)		esperados (log<sid>.ora)
--	--	--	-----------------------------

15.8.7 Seguridad

Oracle pone al alcance del Administrador de Bases de Datos (**DBA**) varios niveles de seguridad:

🔒 Seguridad de cuentas para la validación de usuarios.

Para acceder a los datos en una BD Oracle, se debe tener acceso a una cuenta en esa BD, donde cada cuenta debe tener una palabra clave o *password* asociada. Los *passwords* son establecidos cuando se crea un usuario y pueden ser alterados por el DBA o por el usuario mismo. La BD almacena una versión encriptada del *password* en una tabla del diccionario llamada **dba_users**.

🔒 Seguridad en el acceso a los objetos de la base de datos.

El acceso a los objetos de la BD se realiza por medio de los **privilegios**. Estos permiten que determinados comandos sean utilizados contra determinados objetos de la BD., es decir restringe ciertas acciones. Esto se especifica con el comando **GRANT**, *conceder*. Los privilegios se pueden agrupar formando lo que se conoce por *roles*. La utilización de los roles simplifica la administración de los privilegios cuando tenemos muchos usuarios y pueden ser protegidos con *passwords*, y pueden activarse y desactivarse dinámicamente, con lo que constituyen una capa más de seguridad en el sistema.

🔒 Seguridad a nivel de sistema para la gestión de privilegios globales.

Los roles se pueden utilizar para gestionar los comandos de sistema disponibles para los usuarios. Estos incluyen comandos como *CREATE TABLE* o *SELECT ANY TABLE*. Todos los usuarios que quieran acceder a la BD deben tener el rol *CONNECT*; aquellos que necesiten crear segmentos necesitarán el rol *RESOURCE*. Un usuario con el rol *DBA* tiene derecho para ver y manejar todos los datos de la BD. En Oracle **CONNECT, RESOURCE y DBA son roles de sistema**. Así, un usuario puede tener concedido el privilegio *CREATE TABLE*, pero no el *ALTER TABLE*.

15.8.7.1 Implementación de Seguridad

El primer paso en la seguridad de la BD es asegurar la plataforma en la que reside. Una vez que esto ha sido identificado, se debe considerar la seguridad del sistema operativo.

15.8.7.2 Creación de Usuarios

El objetivo de la creación de usuarios es establecer una cuenta segura y útil, que tenga los privilegios adecuados y los valores por defecto apropiados. En Oracle se puede especificar todo lo necesario para abrir una cuenta con el comando **CREATE USER**. Los parámetros que se le pueden pasar son:

Tabla 6. Parámetros para la creación de usuarios

Parámetro	Significado
Username	Nombre del Usuario (Esquema)
Password	Palabra clave de la cuenta. Puede ser asociada

	directamente a una cuenta del sistema operativo.
Default Tablespace	<u>Espacio de tablas por defecto en el que los objetos de este usuario serán creados.</u> Esto no da al usuario derechos de crear objetos.
Temporary Tablespace	El espacio de tablas en el que <u>se almacenarán</u> los segmentos temporales de las ordenaciones.
Quota	Espacio máximo que puede ocupar en un espacio de tablas.
Profile	Asigna un perfil al usuario. Los perfiles se utilizan para restringir el uso de recursos como el tiempo de CPU.

Uso del comando CREATE USER para crear una cuenta para un usuario:

SVRMGR> create user nombre_usuario

2> identified by zerep

3> default tablespace users

4> temporary tablespace temp;

Si no se especifica un perfil, se aplica el perfil por defecto de la BD, que se llama **DEFAULT** y tiene asignados todos los límites a **UNLIMITED**. Por otra parte si no se especifica una **Quota** el usuario no puede crear objetos.

15.8.7.3 Eliminación de Usuarios

Los usuarios pueden ser eliminados de la BD utilizando el comando **DROP USER**. Este comando tiene un único parámetro, **CASCADE**, el cual permite borrar todos los objetos del usuario antes de eliminar el usuario. Ejemplo para eliminar un usuario:

SVRMGR> drop user perez cascade;

Si se crea otro usuario con el mismo nombre no hereda los objetos del anterior usuario con ese nombre, debido a que Oracle asigna a cada cuenta un número además del nombre, y utiliza ese número para determinar el propietario de todos los objetos que crea esa cuenta, y utiliza el nombre solo para la comunicación con los usuarios.

15.8.7.4 Privilegios del Sistema

Los roles de sistema se utilizan para distribuir la disponibilidad de los comandos del sistema utilizados para gestionar la BD. Los privilegios más comunes están en la siguiente tabla. En ella se distinguen entre privilegios de **manejo de objetos** y de **gestión de la BD**. La palabra clave **ANY** significa que ese usuario tiene el privilegio para todos los esquemas en la BD.

Tabla 7. Privilegios del sistema

Privilegio	Capacidades
Manejo de Objetos	...
CREATE ANY INDEX	Crear cualquier índice.
CREATE [PUBLIC] SYNONYM	Crear sinónimos [públicos].
CREATE [ANY] TABLE	Crear tablas. El usuario debe tener cuota en el espacio de tablas, o ha de tener asignado el privilegio UNLIMITED TABLESPACE.
CREATE [ANY] VIEW	Crear vistas.
ALTER ANY INDEX	Alterar cualquier índice.
ALTER ANY TABLE	Alterar cualquier tabla
DROP ANY INDEX	Borrar cualquier índice.
DROP ANY SYNONYM	Borrar cualquier sinónimo.

DROP PUBLIC SYNONYM	Borrar sinónimos públicos.
DROP ANY VIEW	Borrar cualquier vista.
DROP ANY TABLE	Borrar cualquier tabla.
SELECT ANY TABLE	Efectuar selecciones de cualquier tabla o <u>vista</u> .
INSERT ANY TABLE	Insertar en cualquier tabla o vista.
DELETE ANY TABLE	Borrar filas de cualquier tabla o vista, y también truncar.
ALTER SESSION	Alterar los parámetros de <u>la sesión</u> .
CREATE SESSION	Conectarse a la BD.
Gestión de la BD	...
CREATE PROFILE	Crear perfiles de usuario.
CREATE ROLE	Crear roles.
CREATE ROLLBACK SEGMENT	Creación de segmentos de rollback.
CREATE TABLESPACE	Crear espacios de tablas.
CREATE USER	Crear usuarios.
ALTER PROFILE	Alterar perfiles existentes.
ALTER ANY ROLE	Alterar cualquier rol.
ALTER ROLLBACK SEGMENT	Alterar segmentos de rollback.
ALTER TABLESPACE	<u>Alterar espacios de tablas.</u>
ALTER USER	Alterar usuarios.
DROP PROFILE	Borrar un perfil existente.
DROP ANY ROLE	Borrar cualquier rol.
DROP ROLLBACK SEGMENT	Borrar un segmento de rollback existente.
DROP TABLESPACE	Borrar un espacio de tablas.
DROP USER	Borrar un usuario. Añadir CASCADE si el usuario posee objetos.
ALTER DATABASE	Permite una sentencia ALTER DATABASE.
GRANT ANY PRIVILEGE	Otorgar cualquiera de estos privilegios.
GRANT ANY ROLE	<u>Otorgar cualquier rol a un usuario.</u>

UNLIMITED TABLESPACE	Puede usar una cantidad de almacenamiento ilimitada.
DROP PROFILE	Borrar un perfil existente.

Los privilegios se pueden agrupar en **roles**, para así satisfacer a distintos tipos de usuarios. En la instalación se crea un rol llamado **OSOPER** que sirve para los operarios de la máquina donde está la BD y permite realizar copias de seguridad **en frío** y **en caliente**. Los privilegios de OSOPER son **STARTUP, SHUTDOWN, ALTER DATABASE OPEN/MOUNT, ALTER DATABASE BACKUP, ARCHIVE LOG, RECOVER y RESTRICTED SESSION**.

Se pueden crear nuevos roles. Por ejemplo, podemos crear un rol llamado *creadorCuentas* que sólo pueda crear usuarios y no pueda realizar ninguna otra operación de DBA. Las sentencias que permiten hacer esto son las siguientes:

SVRMGR> create role creadorCuentas;

Statement processed.

SVRMGR> grant create session, create user to creadorCuentas;

Statement processed.

Oracle incluye otros tres roles de sistema: **CONNECT, RESOURCE y DBA**, cuyos privilegios son:

Tabla 8. Privilegios de los roles CONNECT, RESOURCE y DBA

Rol	Privilegios
CONNECT	alter session, create <u>session</u> , create cluster, create table, create view, create synonym, create sequence, create database link
RESOURCE	create cluster, create table, create procedure, create sequence, create trigger
DBA	todos los privilegios de sistema con la opción <i>with admin option</i>

15.8.7.5 Perfiles de Usuario

Los perfiles se utilizan para limitar la cantidad de recursos del sistema y de la BD disponibles para un usuario. Si no se definen perfiles para un usuario se utiliza el perfil por defecto, que especifica recursos ilimitados.

Los recursos que pueden ser limitados via perfil son los siguientes:

Tabla 9. Recursos limitados via perfil

Recurso	Descripción
SESSIONES_PER_USER	El número de sesiones concurrentes que un usuario puede tener en una instancia.
CPU_PER_SESSION	El tiempo de CPU, en centenas de segundos, que una sesión puede utilizar.
CONNECT_TIME	El número de minutos que una sesión puede permanecer activa.
IDLE_TIME	El número de minutos que una sesión puede permanecer sin que sea utilizada de manera activa.
LOGICAL_READS_PER_SESSION	El número de bloques de datos que se pueden leer en una sesión.
LOGICAL_READS_PER_CALL	El número de bloques de datos que se pueden leer en una operación.
PRIVATE_SGA	La cantidad de espacio privado que una sesión puede reservar en la zona de SQL compartido de la SGA.
COMPOSITE_LIMIT	El número de total de recursos por sesión, en unidades de servicio. Esto resulta de un cálculo ponderado de CPU_PER_SESSION, CONNECT_TIME, LOGICAL_READS_PER_SESSION y PRIVATE_SGA, cuyos pesos se pueden variar con el comando ALTER RESOURCE COST.

Los perfiles se pueden crear via el comando **CREATE PROFILE**, y se pueden modificar con la sentencia **ALTER PROFILE**.

15.8.7.6 Gestionando Privilegios

Los privilegios dan acceso a los usuarios a los datos que no poseen. Permiso para realizar una acción, asignable a un usuario o un rol. Pero los privilegios se pueden manejar de manera explícita en algunas circunstancias. Los privilegios se crean via el comando **GRANT** y son registrados en el ***diccionario de datos***.

Los privilegios que pueden otorgarse sobre objetos son los siguientes:

Tabla 10. Privilegios otorgados sobre objetos

Privilegio	Capacidades Otorgadas
SELECT	Puede consultar a un objeto.
INSERT	Puede insertar filas en una tabla o vista. Puede especificarse las columnas donde se permite insertar dentro de la tabla o vista.
UPDATE	Puede actualizar filas en una tabla o vista. Puede especificarse las columnas donde se permite actualizar dentro de la tabla o vista.
DELETE	Puede borrar filas dentro de la tabla o vista.
ALTER	Puede alterar la tabla.
INDEX	Puede crear índices de una tabla.
REFERENCES	Puede crear claves ajenas que referencie a esta tabla.
EXECUTE	Puede ejecutar un procedimiento, paquete o función.

Haciendo un privilegio **PUBLIC** lo hace disponible a todos los usuarios de la BD.

15.8.7.7 Listar Privilegios Otorgados

La información de los privilegios otorgados se almacena en el diccionario de datos. Estos datos son accesibles a través de las siguientes vistas del diccionario de datos:

Tabla 11. Vistas del diccionario de datos

Vista	Contenidos
DBA_ROLES	Nombres de los roles y su estado del password.
DBA_ROLES_PRIVS	Usuarios a los que han sido otorgados roles.
DBA_SYS_PRIVS	Usuarios a los que han sido <u>otorgados privilegios</u> del sistema.
DBA_TAB_PRIVS	Usuarios a los que han sido otorgados privilegios sobre objetos.
DBA_COL_PRIVS	Usuarios a los que han sido otorgados privilegios sobre columnas de tablas.
ROLE_ROLE_PRIVS	Roles que han sido otorgados a otros roles.
ROLE_SYS_PRIVS	Privilegios de sistema que han sido otorgados a roles.
ROLE_TAB_PRIVS	Privilegios de tabla que han sido otorgados a roles.

15.9 ENFOQUE RELACIONAL

Oracle es un manejador de base de datos relacional que hace uso de los recursos del sistema informático en todas las arquitecturas de hardware, para garantizar su aprovechamiento al máximo en ambientes cargados de información.

Es el mayor y más usado Sistema Manejador de Base de Dato Relacional (RDBMS) en el mundo. La Corporación Oracle ofrece este RDBMS como un producto incorporado a la línea de producción. Además incluye cuatro generaciones de desarrollo de aplicación, herramientas de reportes y utilitarios.

Oracle corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo. Soporta unos 17 idiomas, corre automáticamente en más de 80 arquitectura de hardware y software distinto sin tener la necesidad de cambiar una sola línea de código. Esto es porque más el 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos.

Esta basado en el concepto de *relación*. Una relación es un conjunto de n -tuplas. Una tupla, al contrario que un segmento, puede representar tanto entidades como interrelaciones. Los lenguajes matemáticos sobre los que se asienta el modelo relacional, el álgebra y el cálculo relacionales, aportan un sistema de acceso y consultas orientado al conjunto. La repercusión del modelo en los DBMSs comerciales actuales ha sido enorme, estando hoy en día la gran mayoría de los gestores de bases de datos basados en mayor o menor medida en el modelo relacional.

Un modelo relacional posee tres grandes aspectos:

- ⌚ Estructuras: Definición de objetos que contengan datos y que son accesibles a los usuarios.
- ⌚ Operaciones: Definir acciones que manipulen datos u objetos.
- ⌚ Reglas: Leyes para gobernar la información, como y quien manipular.

Una base de datos relacional simplifica y definida como un modelo de información es estrictamente visualizable por los usuarios mediante tablas. Una tabla esta compuesta por una matriz bidimensional de filas y columnas. En cualquier ocasión la información es cambiada en una base de datos relacional, cualquier información es el resultado de una consulta presentad por el usuario en el formato filas/columnas.

16. ADMINISTRACIÓN SGBD Y CONFIGURACIÓN

16.1 COMANDOS INTERACTIVOS ORACLE Y MYSQL

Estos dos RDBMS utilizan para su interacción con el usuario SQL. Estos son los comandos para realizar las tareas mas frecuentes.

- ⌚ Guardar cambios efectuados sobre la base de datos: Commit;
- ⌚ Deshacer o no guardar cambios efectuados sobre la base de datos: Rollback;
- ⌚ Limpiar pantalla: Shift+Supr

16.2 COMANDOS DDL (LENGUAJE DE DEFINICIÓN DE DATOS)

Hace referencia a la creación de la estructura de la base de datos

Creación de tabla:

```
Create Table Nombre_de_Tabla (  
campo1          tipo_de_dato (precisión),  
campo2          tipo_de_dato (precisión),  
...  
campon          tipo_de_dato (precisión)  
);
```

Tipos de Datos:

VarChar2 : Tipo de dato alfanumérico.
Number : Tipo de dato numérico real.

Date : Tipo de dato fecha.

Eliminación de Tabla:

Drop Table **Nombre_de_Tabla** Cascade Constraints;

Creación de llave primaria:

Alter Table **Nombre_de_Tabla** Add Constraint **Nombre_de_Tabla_PK** Primary
Key (campos_que_conforman_la_llave_separados_por_comas);

Eliminación de llave primaria:

Alter Table **Nombre_de_Tabla** Drop Primary Key

Creación de llave foránea:

Alter Table **Nombre_de_Tabla_Muchos** Add Constraint
NomTabMuc_NomTabUno_FK Foreign Key
(campos_que_conforman_la_llave_separados_por_comas)
References **Nombre_de_Tabla_Uno**
(campos_que_conforman_la_llave_separados_por_comas);

Donde, **Nombre_de_Tabla_Muchos** es el nombre de la tabla que posee de su lado de la relacion la cardinalidad MUCHOS, y **Nombre_de_Tabla_Uno** es el nombre de la tabla que posee de su lado de la relacion la cardinalidad UNO.

Eliminación de llave foranea:

Alter Table **Nombre_de_Tabla** Drop Foreign Key

Creación de restricción de Not Null para campos de una tabla:

Alter Table **Nombre_de_Tabla** Modify **Nombre_del_campo** Not Null;

Modificación de la estructura de una tabla para añadirle un campo:

```
Alter Table Nombre_de_Tabla Add Nombre_del_campo tipo_de_dato  
(precisión);
```

Modificación de la estructura de una tabla para modificarle el tipo de dato a un campo:

```
Alter Table Nombre_de_Tabla Modify Nombre_del_campo tipo_de_dato  
(precisión);
```

16.2.1 Aplicación de Comandos DDL: (Tabla Estudiantes)

Paso 1: Elimina una estructura ya existente con ese nombre.

```
Drop Table estudiantes Cascade Constraints;
```

Paso 2: Crea la tabla.

```
Create Table estudiantes(  
Id          VarChar2(12),  
Nom         VarChar2(60),  
Tel         VarChar2(30),  
Dir         VarChar2(60),  
Ciu         VarChar2(3),  
Saldo       Number(10,2)  
);
```

Paso 3: Revisa la estructura de la tabla.

```
Describe estudiantes <Enter>
```

Paso 4: Añade la llave primaria.

```
Alter Table estudiantes Add Constraint estudiantes_PK Primary Key (Id);
```

Paso 5:

Alter Table estudiantes Modify Nom Not Null;

Paso 6: Revisa la estructura de la tabla para visualizar los cambios.

Describe estudiantes <Enter>

Paso 7: Añade un campo faltante en la tabla.

Alter Table estudiantes Add Grupo VarChar2(1);

Paso 8: Revisa la estructura de la tabla para visualizar el cambio.

Describe estudiantes <Enter>

16.3 COMANDOS DML (LENGUAJE DE MANIPULACIÓN DE DATOS)

Hace referencia a la manipulación de los datos de la base de datos. Aquí debemos destacar el uso del Algebra Relacional con el objetivo de facilitar la consulta de información entre tablas.

Visualización de tablas existentes en una base de datos:

Select * from tab;

Conteo, agrupamiento, sumatoria:

Group by

Condicionamiento de resultados:

Having

Ordenar campos:

Order by

Requiere campos numéricos:

Sum avg

Visualiza los máximos valores de una condición:

Max()

Visualiza los mínimos valores de una condición:

Min()

Inserción de datos en campos de una tabla:

Insert Into **Nombre_de_Tabla**

(campo1, campo2,..., campon)

Values

(Valor_campo1, Valor_campo2,..., Valor_campon);

Nota: Este estilo de Insert requiere del cambio constante de los valores para los campos encerrados en los paréntesis.

Insert Into **Nombre_de_Tabla**

(campo1, campo2,..., campon)

Values

(&Valor_campo1, &Valor_campo2,..., &Valor_campon);

Nota: Esta variante permite la inserción de datos en forma interactiva, en donde al terminar la inserción de un registro, solo basta escribir en el prompt / <Enter> y en comando se activa nuevamente para la inserción de otro registro.

Recuperación o visualización de datos de una tabla:

Select campo1, campo2, ..., campon

From **Nombre_de_Tabla**

Where condicion_que_involucra_valores_de_los_campos;

Nota: Recuerda que el Select tiene variantes dependiendo del tipo de consulta que se diseñe; esto es, varía según operadores del Algebra Relacional.

Algebra Relacional: Los operadores relacionales más utilizados son:

- Ⓐ **Juntar:** Regresa una relación que contiene todas las tuplas posibles que son una combinación de dos tuplas de cada una de las relaciones especificadas, tales que las dos tuplas que contribuyen a cualquier combinación dada tengan un valor común para los atributos comunes de las dos relaciones.

```
Select Nombre_campo_tabla1, Nombre_campo_tabla2
```

```
From tabla1, tabla2
```

```
Where tabla1, Nombre_campo_tabla1 = tabla2, Nombre_campo_tabla2;
```

- Ⓐ **Restringir o seleccionar:** Regresa una relación que contiene todas las tuplas de una relación definida que satisfacen una condición especificada.

```
Select * from tabla where Nombre_campo = 'condition';
```

- Ⓐ **Proyectar:** Regresa una relación que contiene todas las tuplas o subtuplas que quedan en una relación definida luego de quitar los atributos especificados.

```
Select Nombre_campo1, Nombre_campo2 from tabla;
```

- Ⓐ **Producto:** Regresa una relación que contiene todas las tuplas posibles que son una combinación de dos tuplas, una de cada una de dos relaciones definidas.

```
Select * from tabla1, tabla2, tablan;
```

Eliminación de todos los datos en campos de una tabla:

Delete; una vez ubicados sobre la tabla, claro está ò

```
Delete from Nombre_de_Tabla;
```

Eliminación de filas de una tabla:

Delete From **Nombre_de_Tabla**

Where condicion_que_involucra_valores_de_los_campos;

Nota: La palabra clave From es opcional. Si no se le indica una condición al Delete, todas las tuplas de la tabla serán borradas.

Actualización de datos en campos de una tabla:

Update **Nombre_de_Tabla**

Set campo1 = Nuevo_valor1, campo2 = Nuevo_valor2, ..., campon = Nuevo_valorn

Where condicion_que_involucra_valores_de_los_campos;

Nota: Si no se le indica una condición al Update, todas las tuplas de la tabla serán actualizadas.

16.3.1 Aplicación de Comandos DML: (Tabla Estudiantes)

Paso 1: Inserta datos.

Insert Into estudiantes

(Id, Nom, Grupo, Dir, Tel, Ciu, Saldo)

Values

('9010','LENIS H','A','SU CASA','212121','BQL',1000);

Insert Into estudiantes

(Id, Nom, Grupo, Dir, Tel, Ciu, Saldo)

Values

```
('1011','OLGA R','A','SU CASA','323232','BQL',1500);
```

Insert Into estudiantes

(Id, Nom, Grupo, Dir, Tel, Ciu, Saldo)

Values

```
('1012','BERTHY R','A','SU CASA','141414','BQL',3500);
```

Commit;

Paso 2: Realiza consultas variadas.

1. Consulta la cantidad de registros insertados

```
Select Count(*) From estudiantes;
```

2. Visualiza el nombre, el grupo y la ciudad.

```
Select Nom, Grupo, Ciu From estudiantes;
```

3. Visualiza el nombre, el grupo y la cantidad de dinero.

```
Select Nom, Grupo, Saldo From estudiantes;
```

4. Cuenta la cantidad de chicas que hay por grupo.

```
Select Grupo, Count(*) From estudiantes
```

```
Group By Grupo;
```

5. Cuenta la cantidad de dinero que hay por grupo.

Select Grupo, Sum(Saldo) From estudiantes
Group By Grupo;

6. Ver qué grupos tienen más dinero acumulado que una cifra establecida.

Select Grupo, Sum(Saldo) From estudiantes
Group By Grupo
Having Sum(Saldo) > &Valor;

7. Cuenta la cantidad de dinero que hay por grupo y por ciudad.

Select Grupo, Ciu, Sum(*) From estudiantes
Group By Grupo, Ciu;

8. Visualiza los máximos valores de saldo que hay por grupo.

Select Grupo, Max(Saldo) From estudiantes
Group By Grupo;

9. Visualiza los nombres y grupos de aquellas chicas que poseen los máximos valores de saldo por grupo.

Select Nom, Grupo, Saldo From estudiantes
Where (Grupo, Saldo) In (Select Grupo, Max(Saldo)
From estudiantes Group By Grupo);

10. Visualiza los máximos valores de saldo que hay por grupo.

Select Grupo, Max(Saldo) From estudiantes
Group By Grupo;

11. Visualiza los mínimos valores de saldo que hay por grupo.

```
Select Grupo, Min(Saldo) From estudiantes  
Group By Grupo;
```

12. Visualiza los nombres y grupos de aquellas chicas que poseen los mínimos valores de saldo por grupo.

```
Select Nom, Grupo, Saldo From estudiantes  
Where (Grupo, Saldo) In (Select Grupo, Min(Saldo)  
From estudiantes Group By Grupo);
```

13. Visualiza los nombres y grupos de aquellas chicas que poseen los mínimos y máximos valores de saldo por grupo.

```
Select Nom, Grupo, Saldo From estudiantes  
Where (Grupo, Saldo) In (Select Grupo, Min(Saldo)  
From estudiantes Group By Grupo)  
Union  
Select Nom, Grupo, Saldo From estudiantes  
Where (Grupo, Saldo) In (Select Grupo, Max(Saldo)  
From estudiantes Group By Grupo);
```

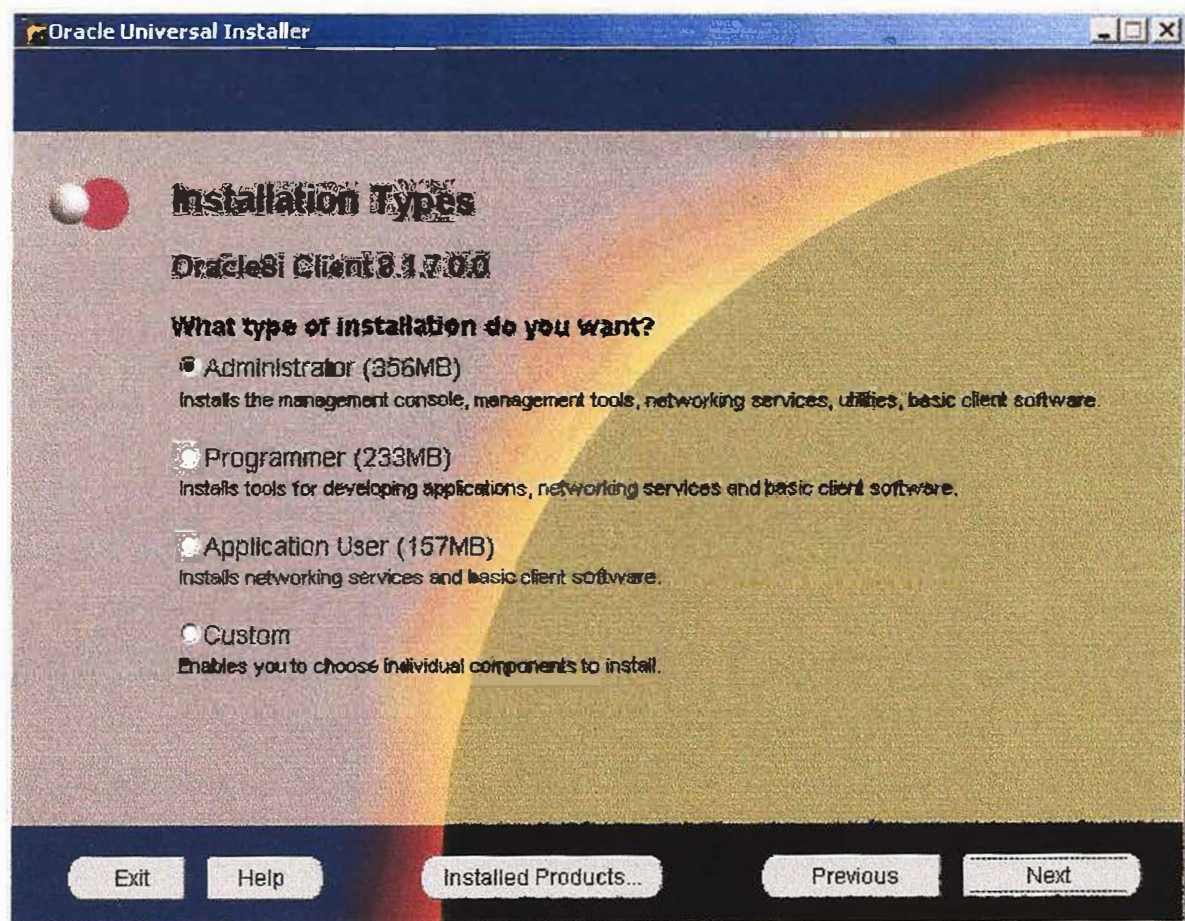
16.4 DBA STUDIO – ADMINISTRADOR DE ORACLE

La versión 8 de la base de datos Oracle incluye una herramienta de administración gráfica que es mucho más intuitiva y cómoda de utilizar. Se emplea en forma alternativa a los comandos de línea de texto que se usan para efectuar administración. Por lo tanto, no es necesario disponer de esta herramienta en

forma obligatoria porque siempre es posible administrar una base de datos desde la línea de comandos, sin embargo, es mucho más recomendable por la facilidad de uso y rapidez para efectuar la mayoría de los comandos.

Cuando se instalan las aplicaciones cliente de la base de datos, aparece la pantalla siguiente, con todas las opciones que se indican.

Figura 53. Pantalla de instalación del cliente de Oracle



El DBA Studio es una herramienta integrada que permite administrar instancias de bases de datos, esquemas, seguridad y almacenamiento desde un view tree que los incorpora a todos ellos, haciendo más fácil el intercambio entre las tareas y la tener una vista global exacta de la configuración de la base de datos.

Podemos ejecutar el DBA Studio con o sin conectar a un Oracle Management Server (OMS). El DBA Studio combina múltiples herramientas de bases de datos en una aplicación y se utiliza para administrar:

- **Instancias:** startup, shutdown e inicialización.
- **Schemas o Esquemas:** tablas, índices y objetos.
- **Seguridad:** usuarios accounts, roles y privilegios.
- **Storage o Almacén:** tablespaces, datafiles y segmentos de rollback.
- **Replication:** multimaster y snapshot replication.
- **Oracle8i JVM (JServer):** namespaces, CORBA y componentes EJB y sus permisos.

Al abrir el DBA Studio aparecen los siguientes pantallazos:

Figura 54. Primer pantallazo con Login to the Oracle Management Server seleccionado

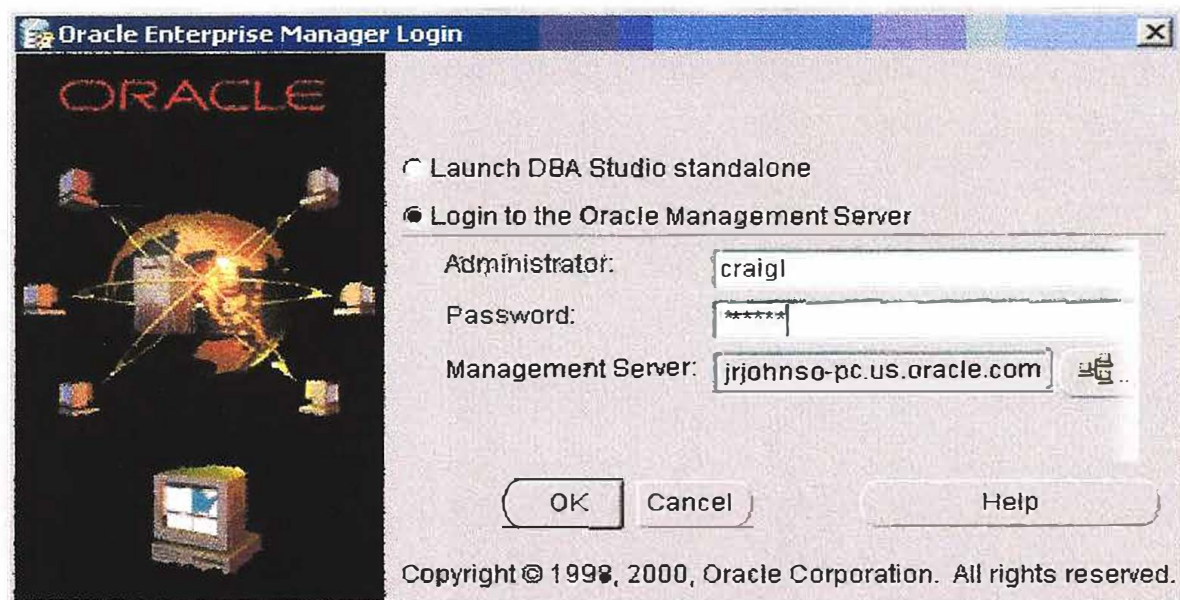


Figura 55. Primer pantallazo con Launch DBA Studio standalone seleccionada

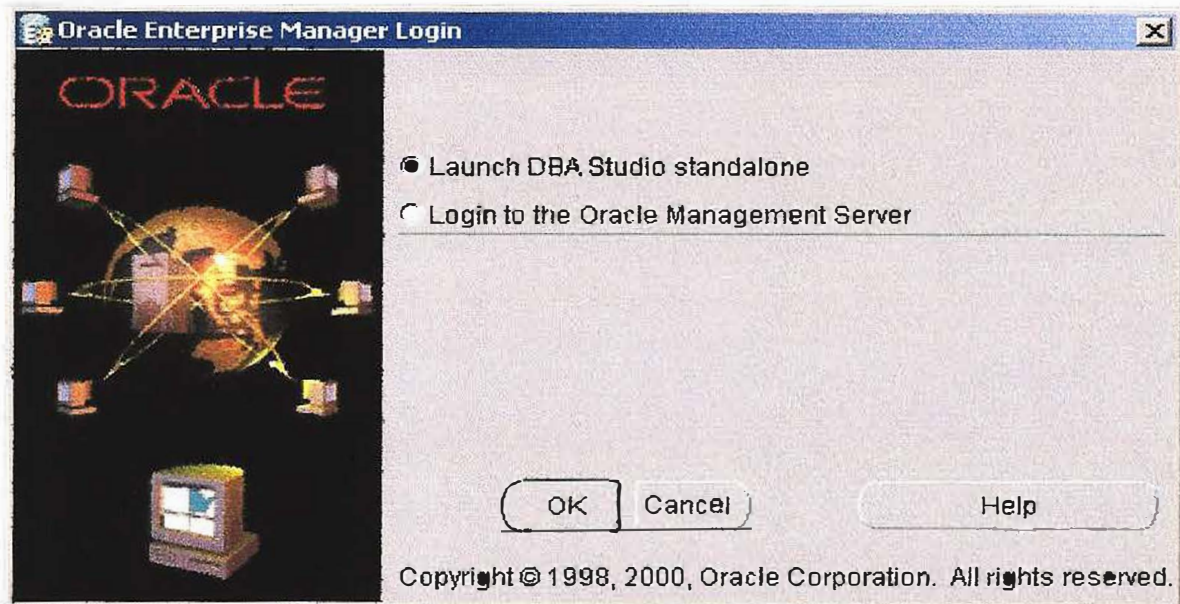
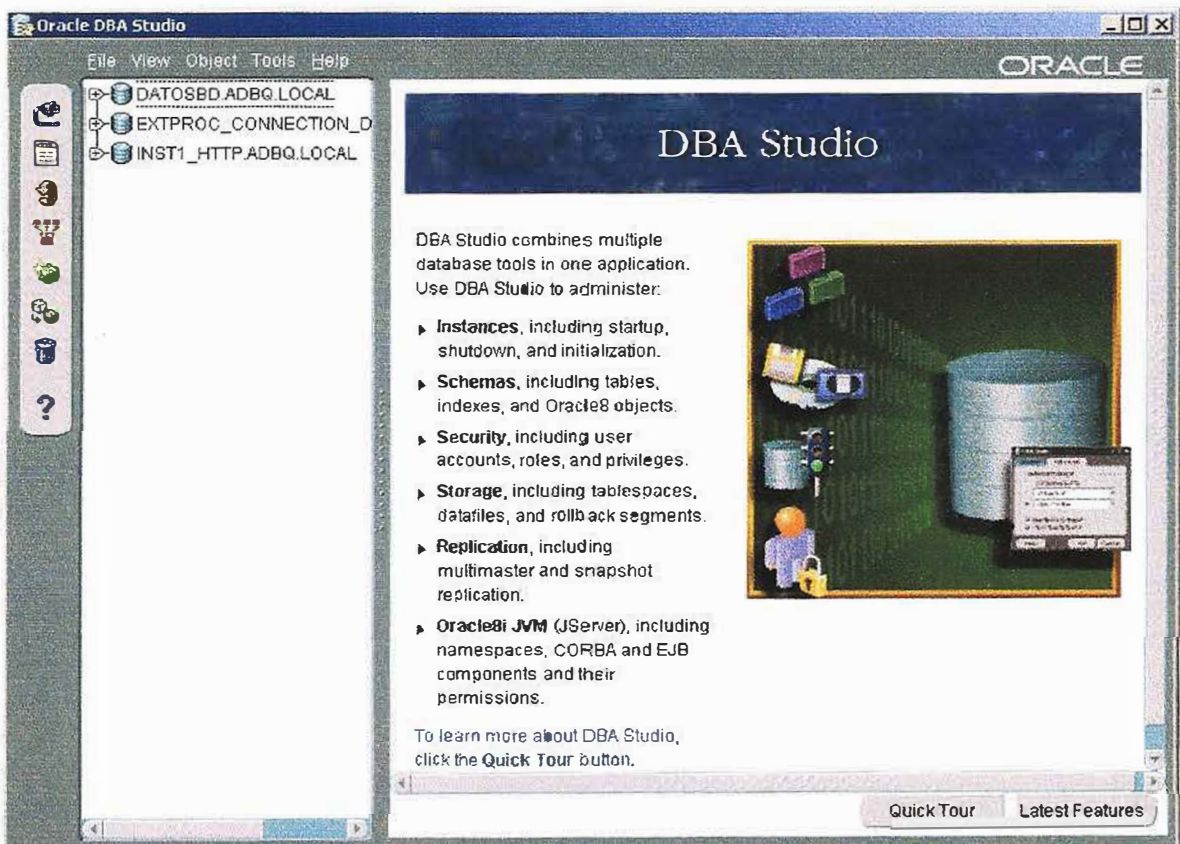


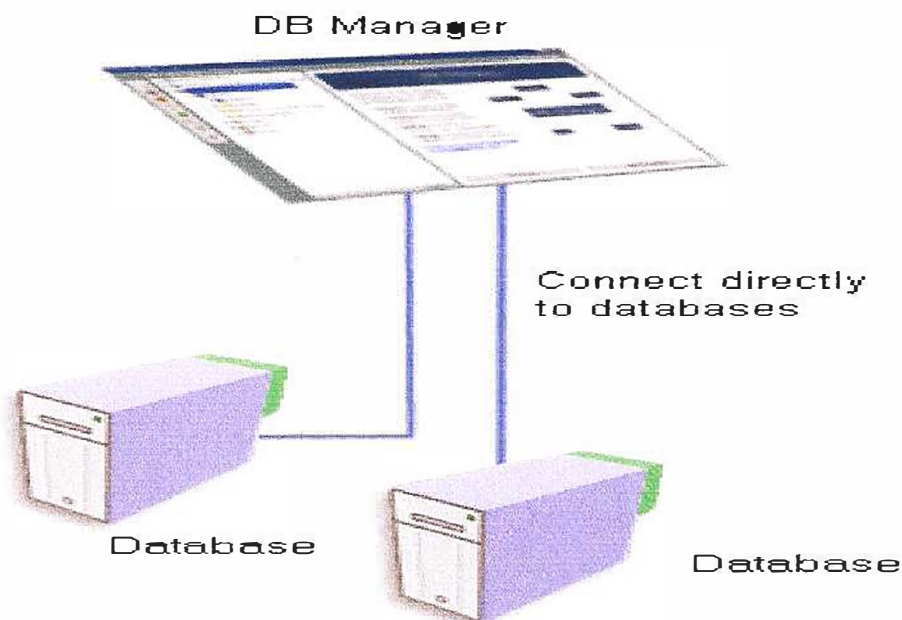
Figura 56. Oracle DBA Studio



16.4.1 Conexión Directa

Es apropiado realizar una **conexión directa** para manejar bases de datos sin arrancar el OMS o cuando la Console/OMS no está disponible. La conexión directa a las bases de datos sin el OMS/Console le permite realizar las tareas básicas sin acceso a los rasgos de OMS.

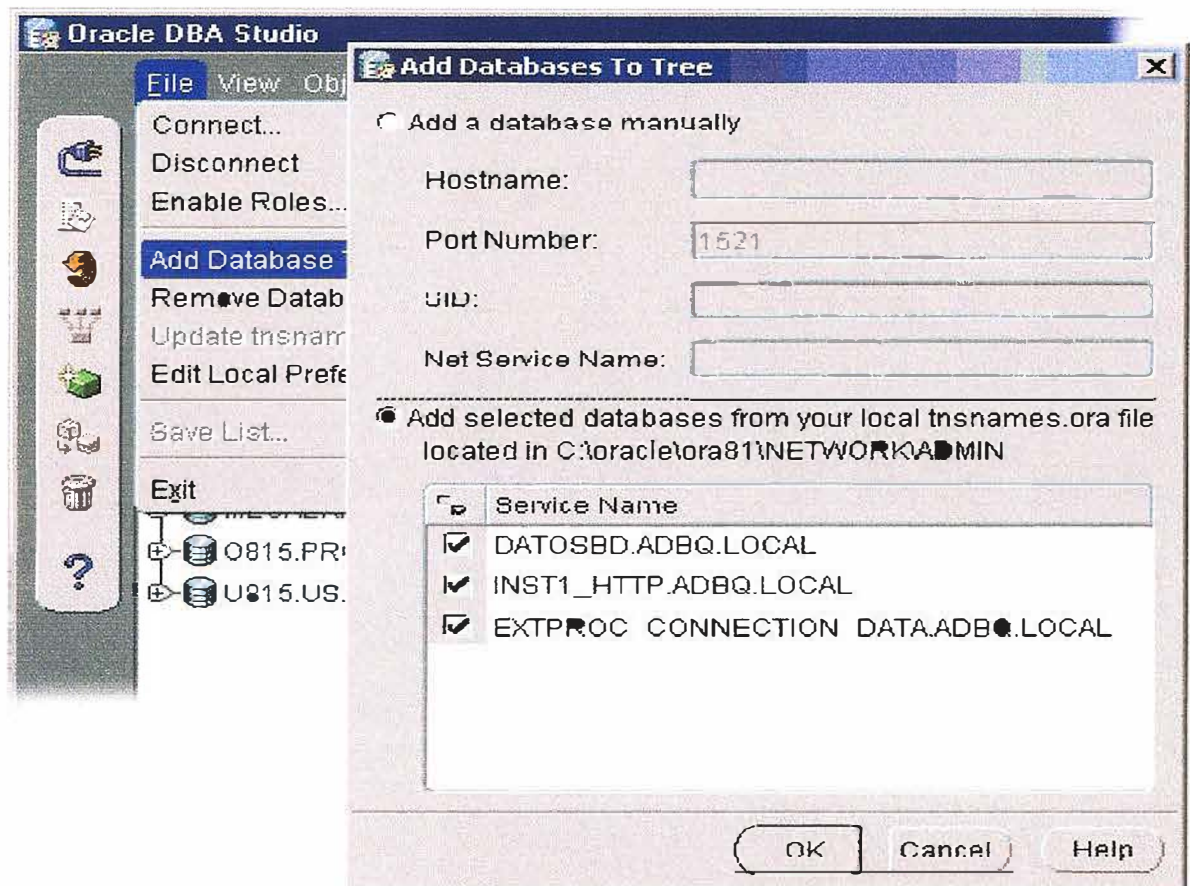
Figura 57. Conexión Directa



Adicionar y remover desde el árbol: Cuando el DBA Studio se conecta directamente con la base de datos, guarda información sobre la conexión de base de datos en el archivo local `tnsnames.ora`, y las bases de datos se adicionan o quitan del view tree del DBA Studio. Podemos transmitir información de conexión de bases de datos desde la Consola al DBA Studio cuando este conectado al OMS y conectar fácil y directamente el DBA Studio a las bases de datos, sin una conexión de OMS.

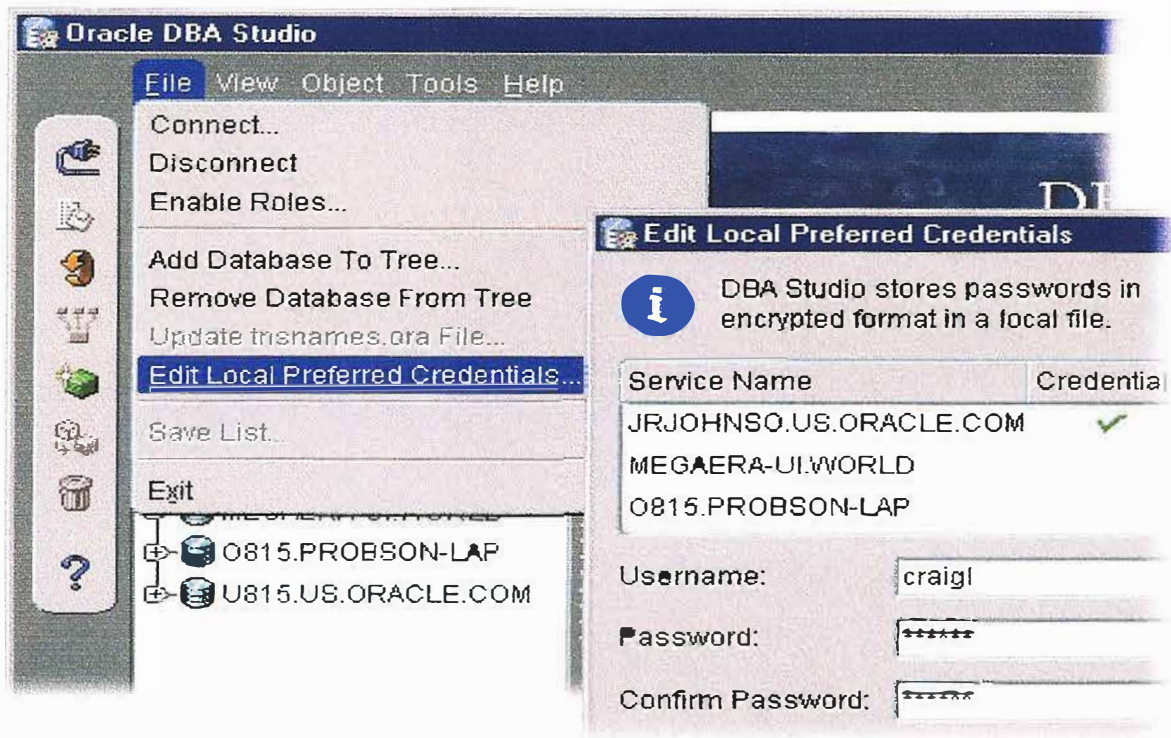
También podemos actualizar la Preferred Credentials local con los nombres de usuarios de OMS y contraseñas encriptadas, actualizando el archivo tnsnames.ora con la información del OMS.

Figura 58. Add Database To Tree



La opción **Edit Local Preferred Credentials** permite almacenar la información del nombre de usuario y la encriptación de su contraseña localmente para realizar la conexión directa. Esta opción se encuentra en el menú File. El DBA Studio le da la opción de mantener usernames y contraseñas, separado del OMS, para que establecer conexiones directas de la base de datos sea fácil. Las contraseñas guardadas localmente son encriptadas.

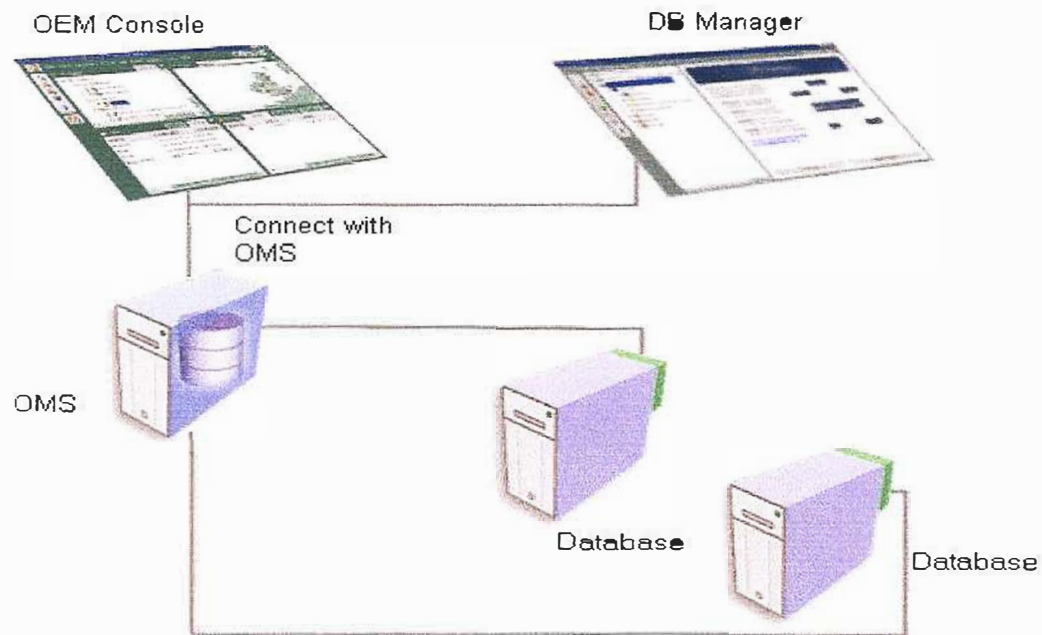
Figura 59. Edit Local Preferred Credentials



16.4.2 Conexión con OMS

En la **conexión con OMS** se pueden realizar backup, administrar o analizar los datos; usar la lista de las bases de datos, cuando se ejecuta el DBA Studio con un browser cliente.

Figura 60. Conexión con OMS



Adicionar y remover en consola: Cuando el DBA Studio DBA se conecta con OMS, se maneja la información con la EM Console, adiciona y quita desde el view tree de Console, y se visualiza dentro del DBA Studio.

Con una conexión OMS, la información del nombre de usuario y contraseña configurados en la EM Console es utilizada cuando se conecta a la base de datos.

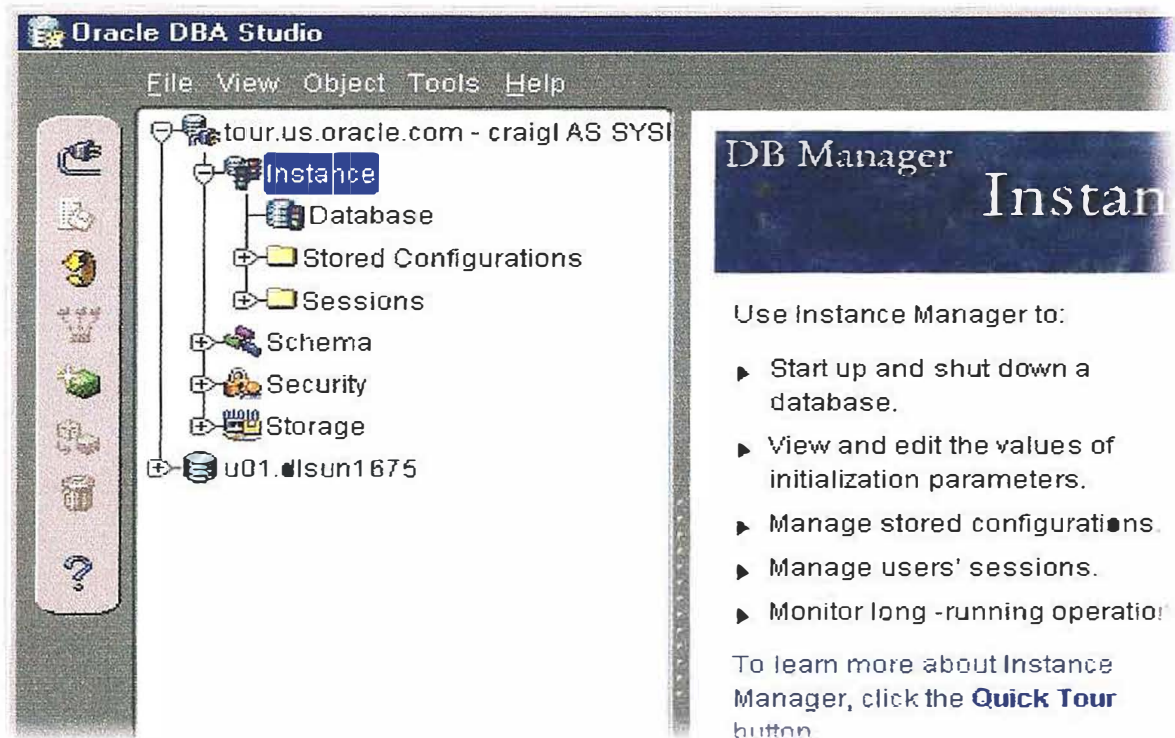
16.4.3 Administrador de Instancias (Instance)

El Administrador de Instancias en el DBA Studio permite:

- ⓐ Poner en marcha y cerrar una base de datos.
- ⓑ Visualizar y editar los valores de los parámetros de inicialización.

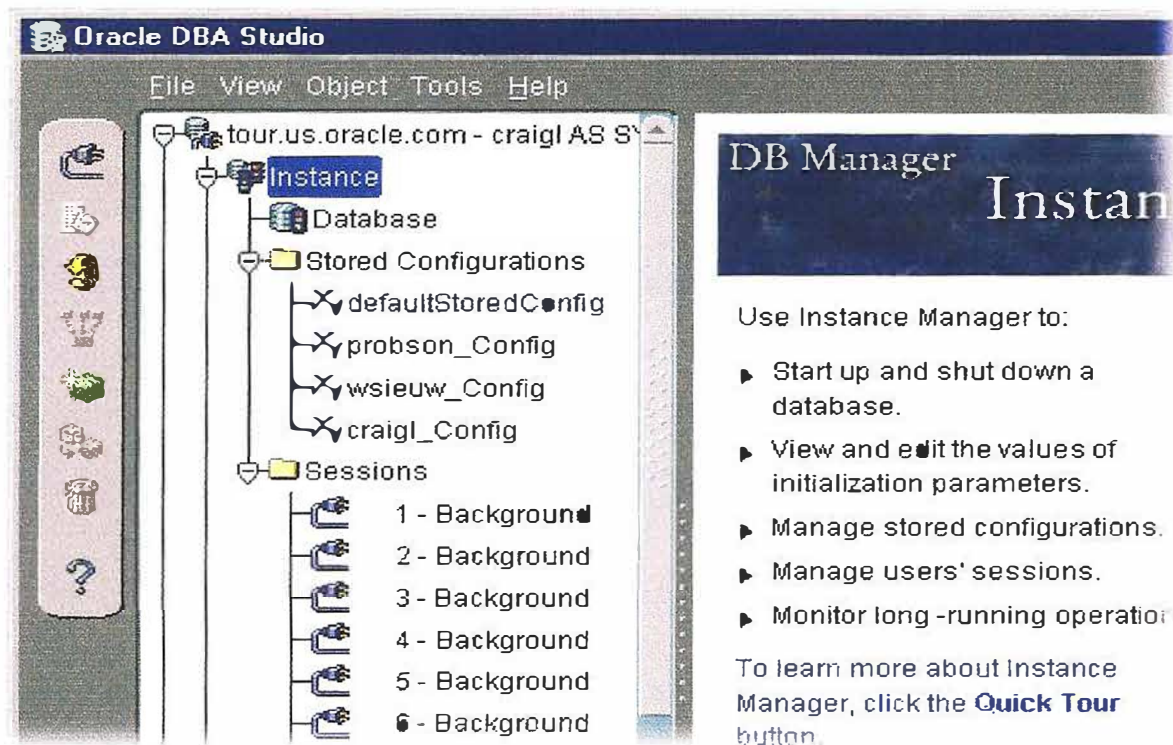
- ④ Visualizar la sesión actual SQL y el Explain Plan.
- ④ Manejar configuraciones guardadas.
- ④ Manejar la CPU y los recursos paralelos en la base de datos.
- ④ Manejar las sesiones de usuarios.
- ④ Monitorear las operaciones long-running.

Figura 61. Instance



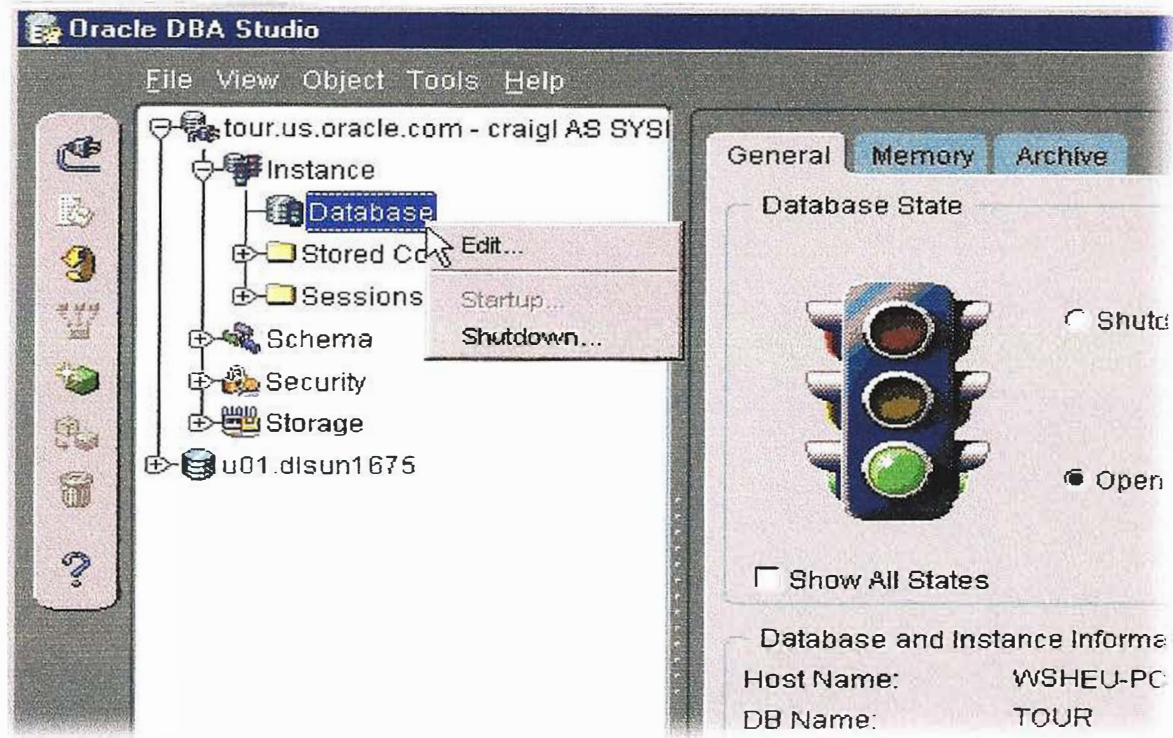
Desplegando el árbol **Instance** veremos iconos y folders para manejar la Base de datos (Database), Sesiones (Sessiones), Resource Consumer Groups, Resource Plans y si se conecta a un OMS, Stored Configurations. Cuando seleccionamos uno de los objetos de la instancia, aparecen sus propiedades del lado derecho de la pantalla.

Figura 62. Árbol Instance



Cuando seleccionamos el icono **Database**, despliega información sobre la base de datos, incluso el estado y límites de los recursos de la base de datos. Podemos utilizar la opción Database State o el menú Database para arrancar, cerrar, o montar la base de datos seleccionada. Si nos conectamos a una base de datos remota, debemos tener una copia local del archivo INIT <SID> .ORA para arrancar la base de datos.

Figura 63. Database

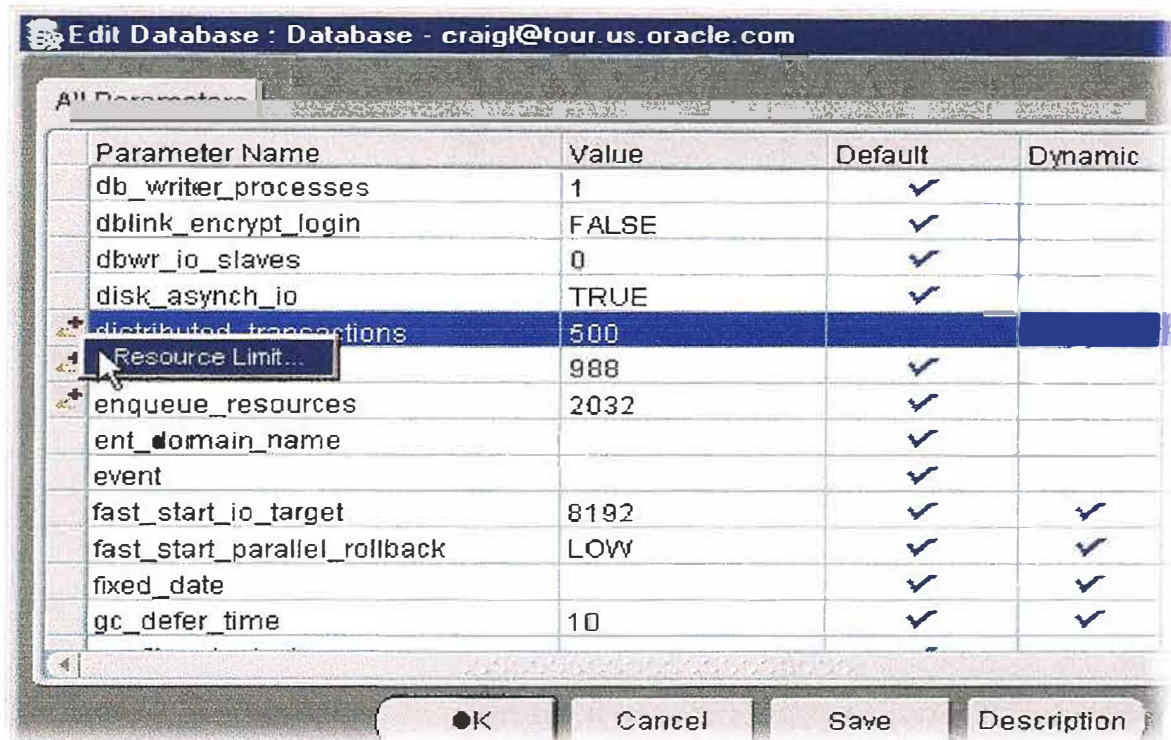


La Instancia lista los parámetros de la base de datos de dos maneras:

- ④ **All Parameters:** Podemos acceder a él seleccionando el icono Database y el boton All Initialization Parameters.
- ④ **Initialization Parameters:** Seleccionando Stored Configuration.

Con el clic derecho del mouse en el signo mas para ver los limites de recurso de parámetro.

Figura 64. All Parameters



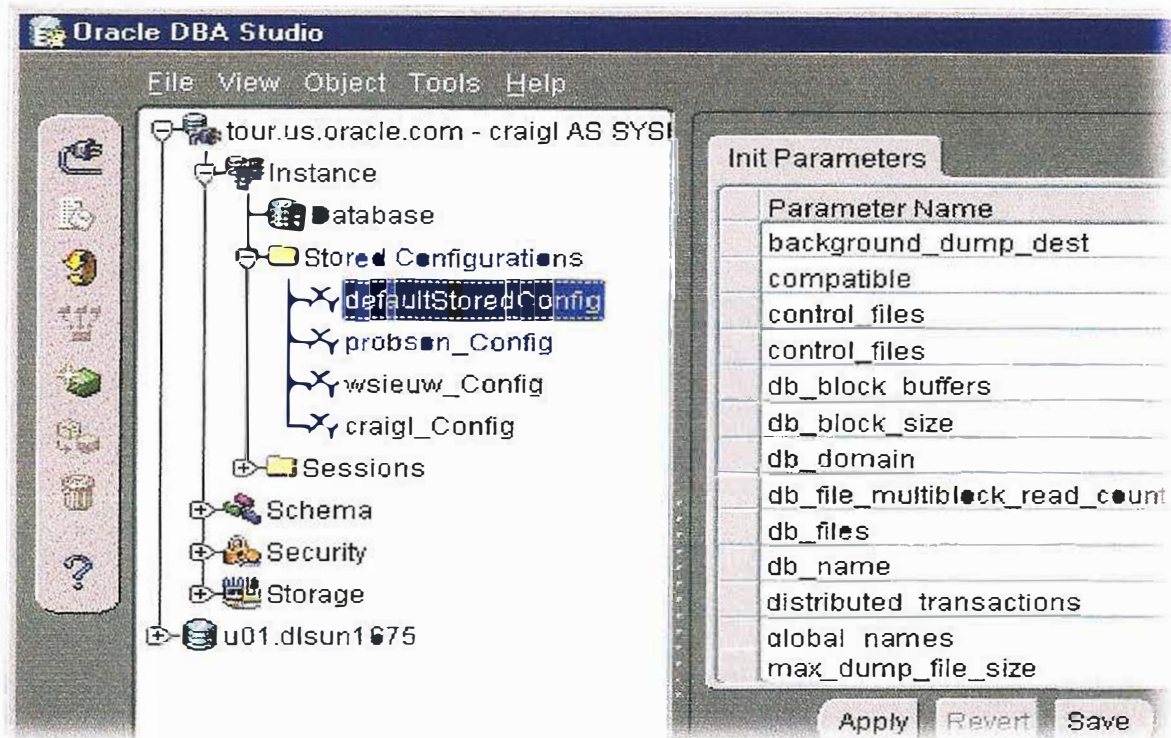
Parameter Name	Value	Default	Dynamic
db_writer_processes	1	✓	
dblink_encrypt_login	FALSE	✓	
dbwr_io_slaves	0	✓	
disk_asynch_io	TRUE	✓	
distributed_transactions	500		
Resource Limit...	988	✓	
enqueue_resources	2032	✓	
event_domain_name		✓	
event		✓	
fast_start_io_target	8192	✓	✓
fast_start_parallel_rollback	LOW	✓	✓
fixed_date		✓	✓
gc_defer_time	10	✓	✓

Buttons: OK, Cancel, Save, Description

El icono **Stored Configurations** permite crear configuraciones de startup de múltiples bases de datos sin la necesidad de guardar localmente archivos de parámetro de inicialización (INIT <SID> .ORA). La opción Stored Configurations está en el almacén de Oracle Enterprise Manager y no como archivos externos, además sólo es visible con una conexión OMS.

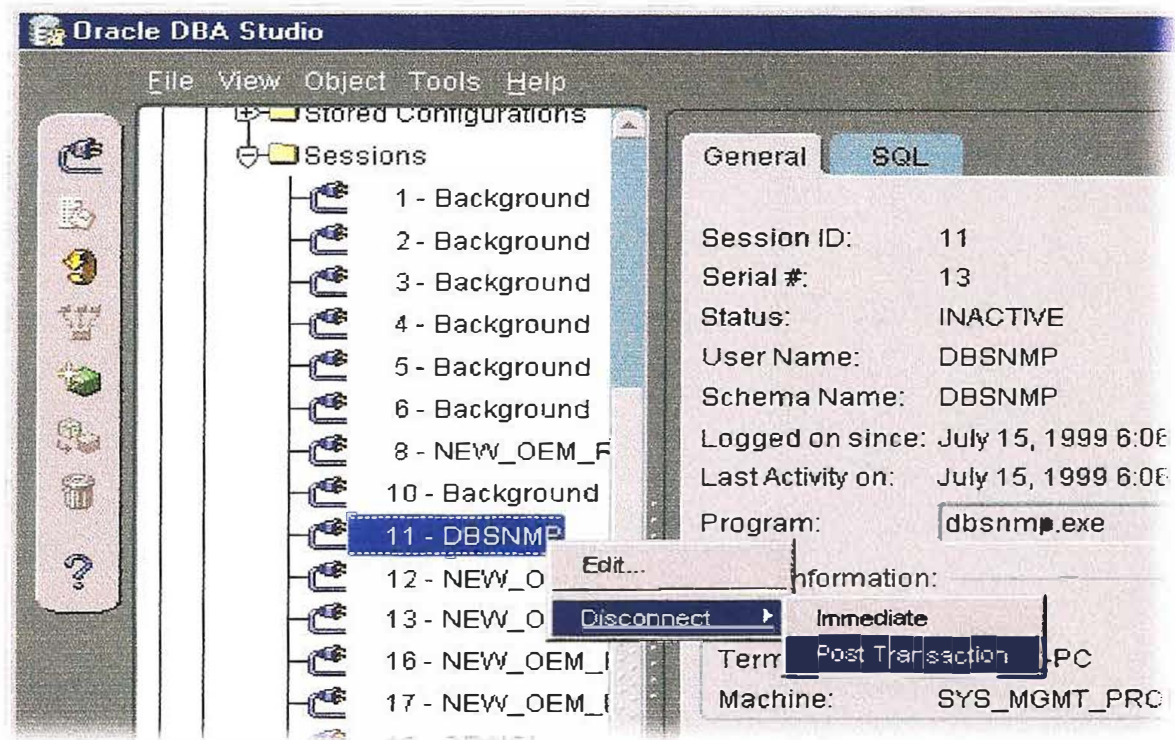
Podemos crear, actualizar y borrar configuraciones guardadas, adicionar y borrar parámetros, y exportar una configuración a un archivo.

Figura 65. Stored Configurations



La pestaña **Sessions** se utiliza para listar los usuarios conectados a la base de datos. Seleccione una sesión para ver los detalles. Podemos desconectar cualquier sesión, inmediatamente o después de que la transacción actual esté completa, y restringe acceso a los usuarios con la opción RESTRICTED SESSION.

Figura 66. Sessions



Las pestañas **Resource Consumer Groups** y **Resource Plans** facilitan la asignación y manejo de recursos de la CPU y controla su funcionamiento de forma paralela.

- El Resource Consumer Groups es utilizado para agrupar sesiones de usuario con requisitos similares.
- El Resource Plans asigna recursos entre los grupos consumidores.

Figura 67. Resource Consumer Groups

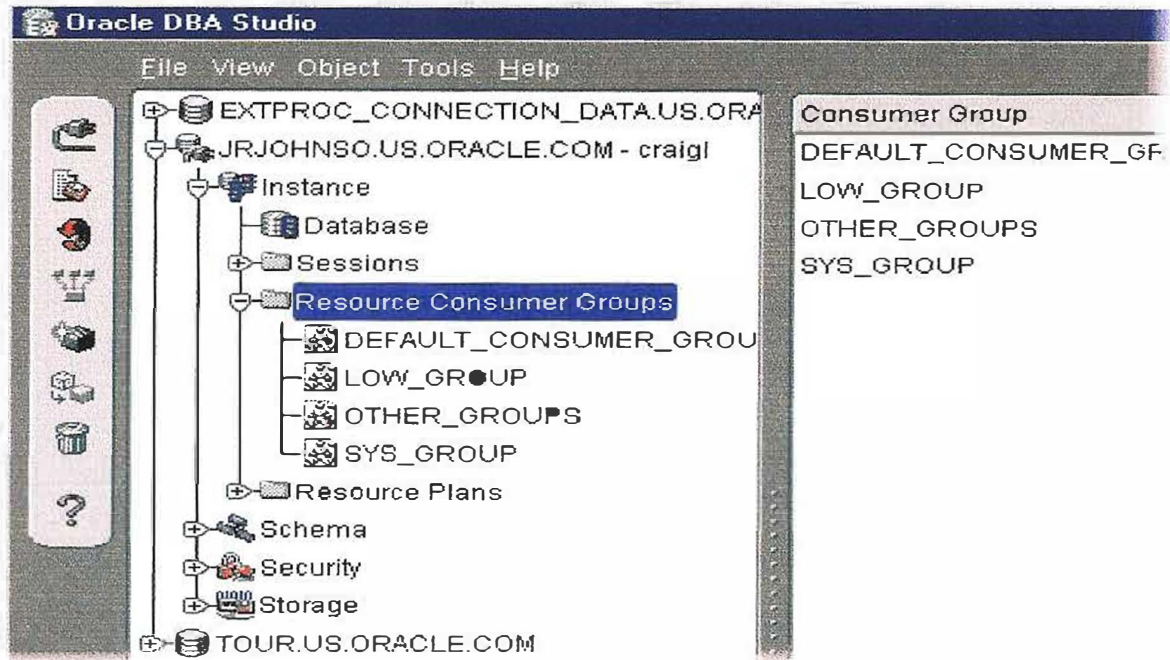
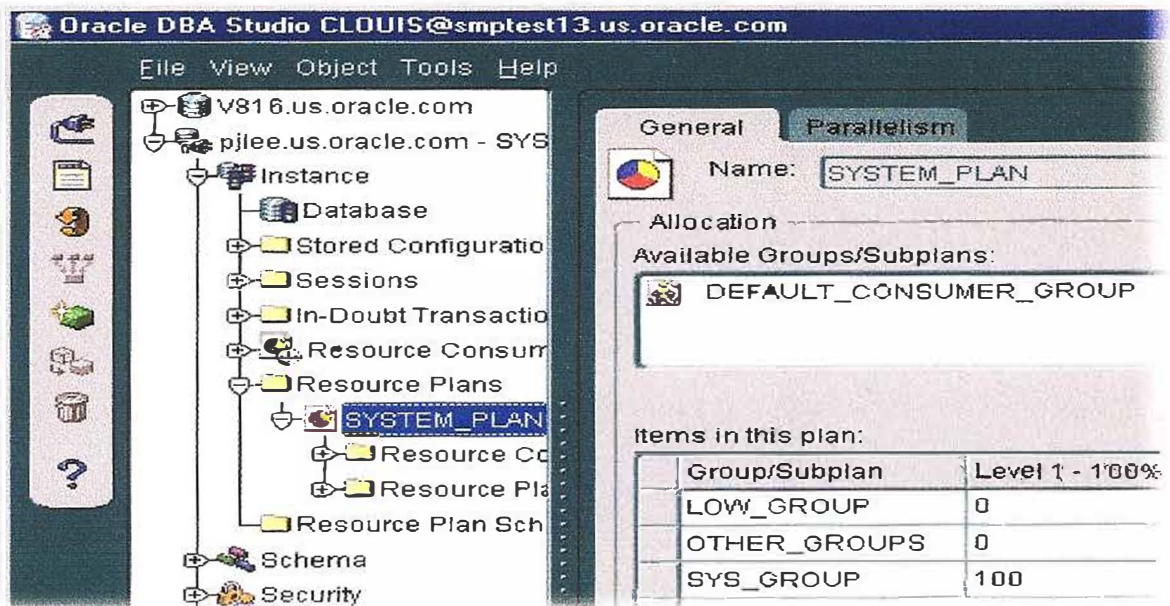


Figura 68. Resource Plans

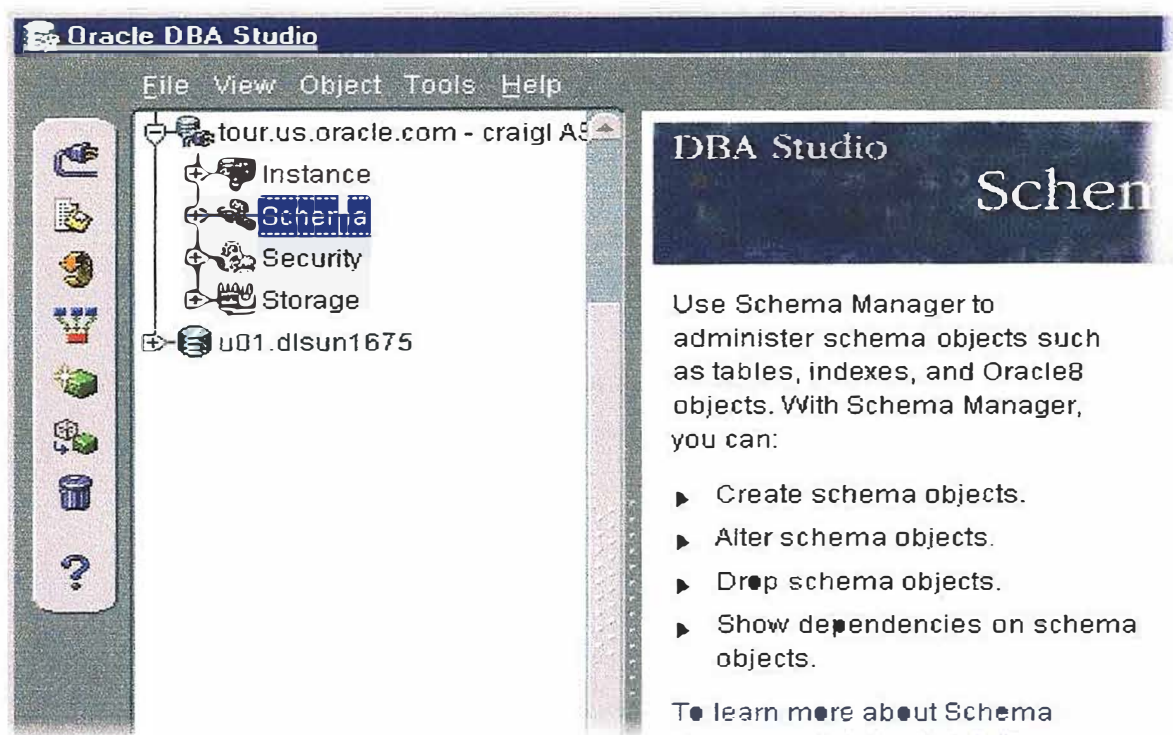


16.4.4 Administrador de Esquemas (Schema)

Un esquema es una colección de objetos asociados dentro de una base de datos. El esquema de la base de datos (*schema*) consta de objetos como tablas, clusters, índices, vistas, procedimientos almacenados, triggers, secuencias y otros.

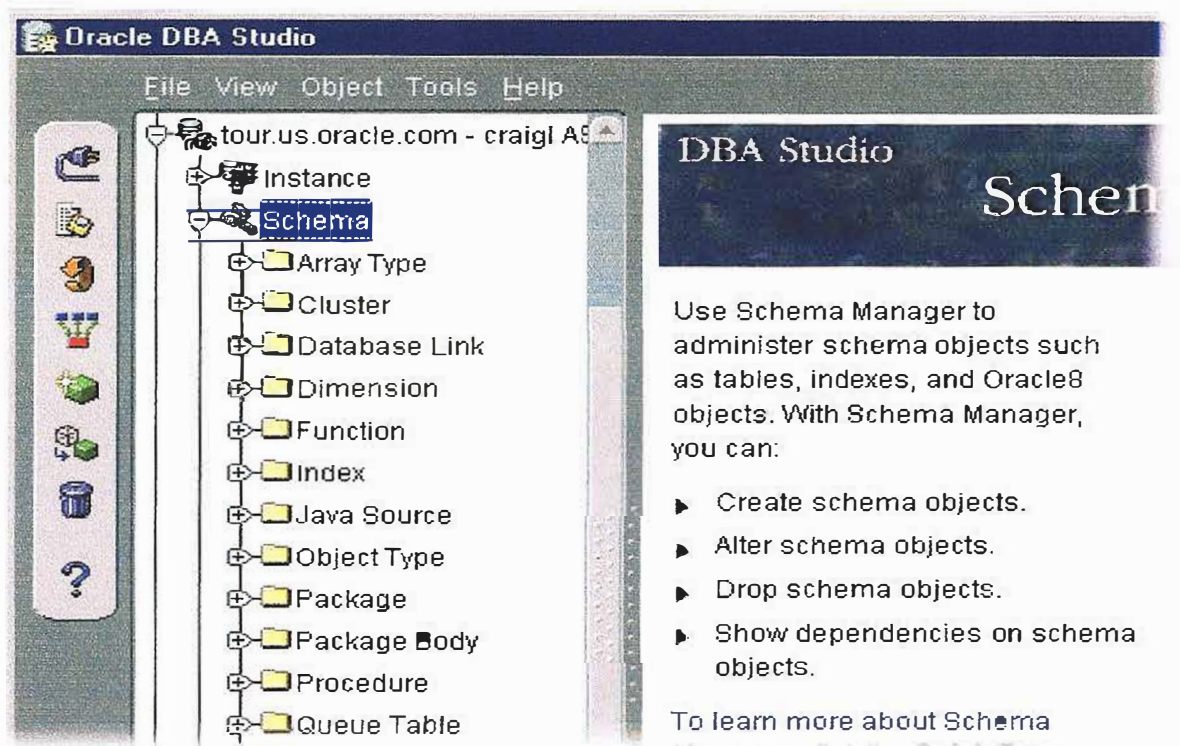
El administrador de Esquemas permite crear, modificar, o eliminar objetos como clusters, índices, snapshots, tablas y vistas. También apoya características Oracle8 como tablas particionadas e índices, tablas organizadas por índices, objetos Oracle8, procedimientos de almacenamiento Java, entre otras.

Figura 69. Schema



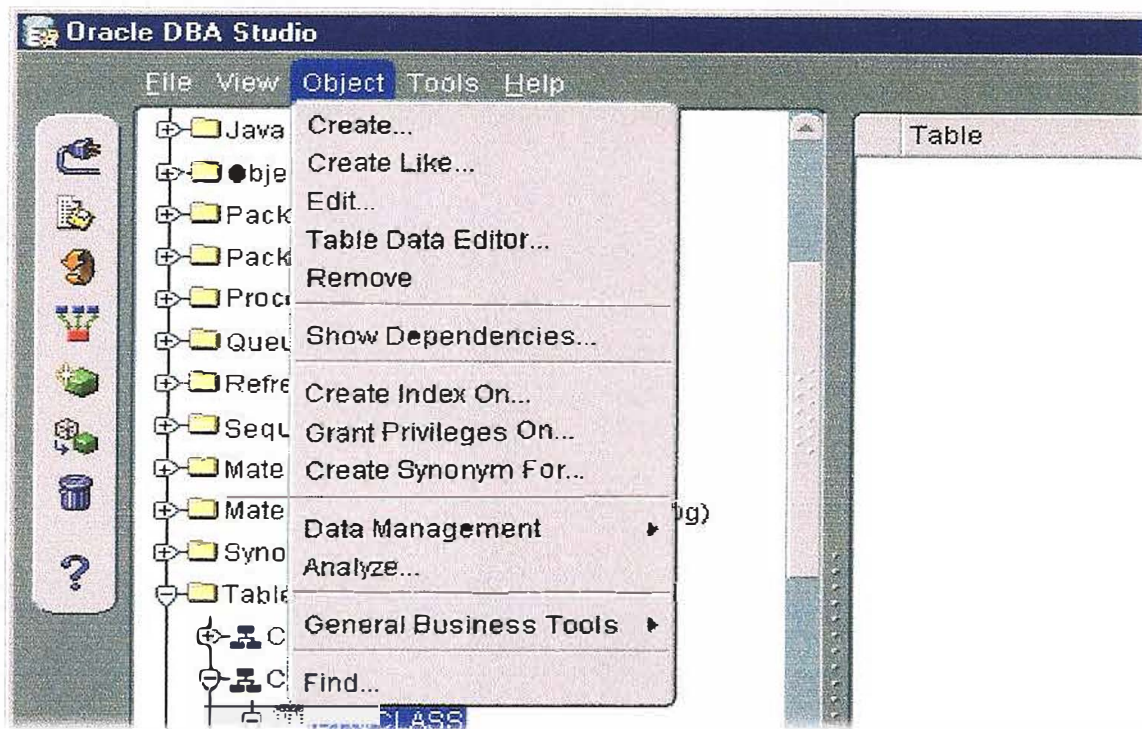
Desplegando el icono **Schema** aparecen varios folders listados alfabéticamente uno para cada tipo de objeto del esquema.

Figura 70. Folders Schema



El menú **Object** se utiliza para identificar y ejecutar operaciones para la mayoría de los objetos del esquema como crear o quitar un objeto, crear copias de un objeto, Dar privilegios a un usuario o rol en un objeto, crear alias a los objetos, ver los objetos que dependen de una tabla y aquéllos que dependen de él.

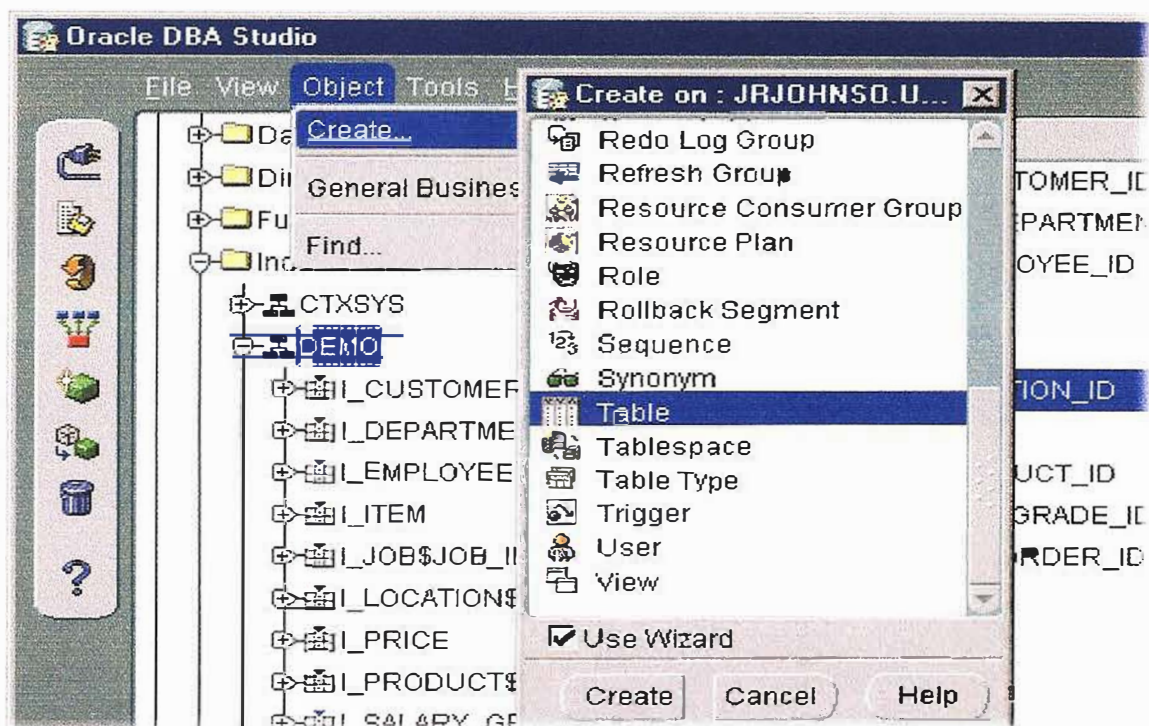
Figura 71. Menu Object



Para crear un objeto, como una tabla, definimos la tabla especificando los parámetros en un diálogo, o si usando un Wizard, como el Create Table Wizard, a través de un proceso que él mismo le indicará.

Al seleccionar un objeto y utilizar el comando Create Like, se crea un "clon" del objeto con todos sus atributos, es idéntico excepto el nombre. Usted puede editar los parámetros en la nueva hoja de propiedades del nuevo objeto. El DBA Studio ahora soporta la creación de tablas particionadas.

Figura 72. Creando una tabla (Paso 1)



Editor de datos de tablas: Cuando un objeto de la tabla se selecciona y pulsamos el botón derecho del mouse activamos el Table Data Editor que se utiliza para ver, actualizar y borrar el contenido de una tabla. Para ver el código de SQL generado para modificar el objeto debemos pulsar el boton View SQL.

Figura 73. Creando una tabla (Paso 2)

Create Table - CRAIGL@wsheu-pc.us.oracle.com

General Constraints Cluster Columns Partitions Storage Options

Name: BONUS2
Schema: SCOTT
Tablespace: SYSTEM

Table: ☒ Standard ☐ Organized Using Index (IOT)
☒ Define Columns ☐ Define Query ☐ Object Table

Name	Schema	Datatype	Size	Scale	P
ENAME	<None>	VARCHAR2	10		
JOB	<None>	VARCHAR2	9		
SAL	<None>	NUMBER	0	0	

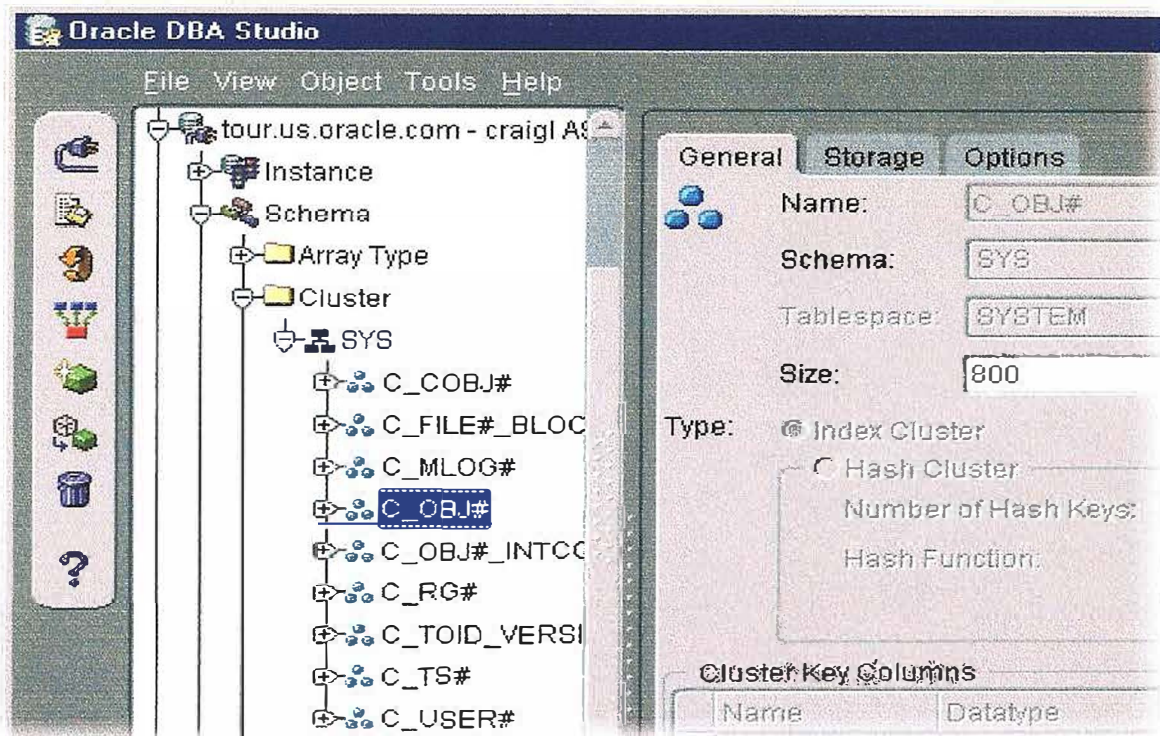
Create Cancel Hide SQL Help

SQL Text

```
CREATE TABLE "SCOTT"."BONUS2"("ENAME" VARCHAR2(10), "JOB" VARCHAR2(9), "SAL" NUMBER, "COMM" NUMBER) TABLESPACE "SYSTEM" PCTFREE 1 PCTUSED 40 INITRANS 1 MAXTRANS 255 STORAGE ( INITIAL 10K NEXT 10K
```

Cuando se selecciona un objeto, como un Cluster, los detalles aparecen al lado derecho de la pantalla. Podemos modificar los parámetros, como el nombre, el tipo, los atributos claves, además de los del Storage y Options.

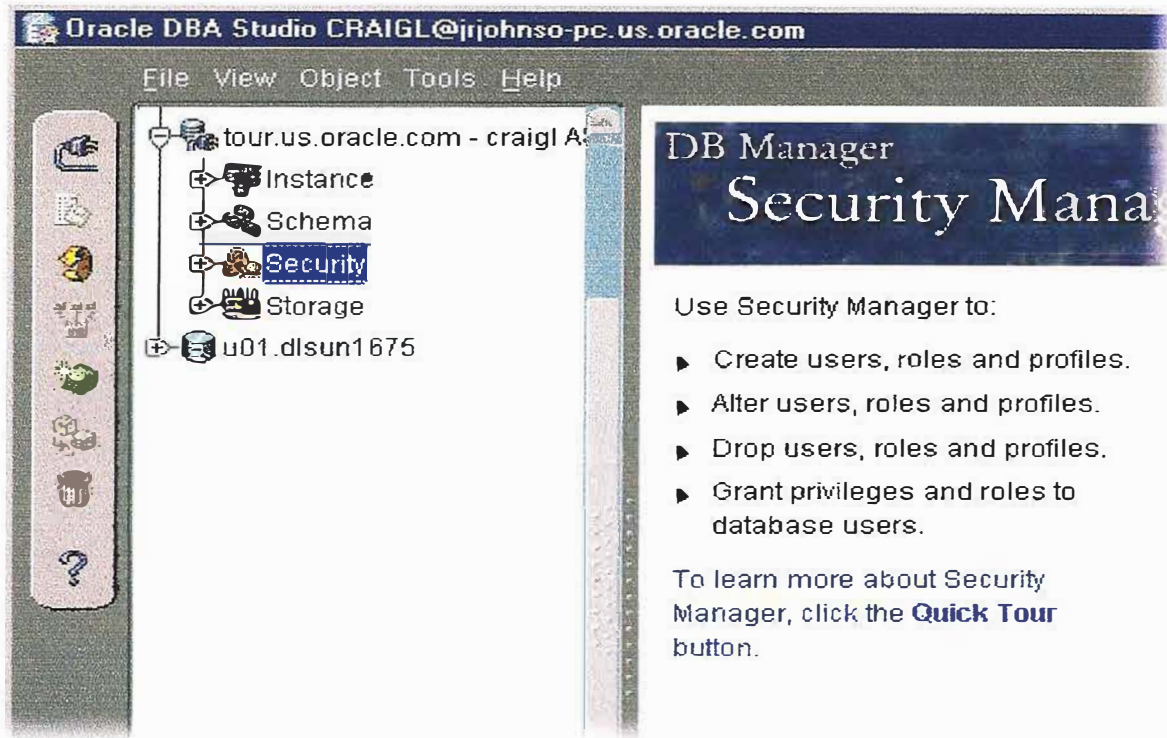
Figura 74. Editando un objeto



16.4.5 Administrador de Seguridad (Security)

El administrador de Seguridad permite crear, modificar y eliminar usuarios, roles y perfiles, y privilegios y roles a los usuarios de la base de datos. Además permite dar privilegios y roles a usuarios múltiples al mismo tiempo.

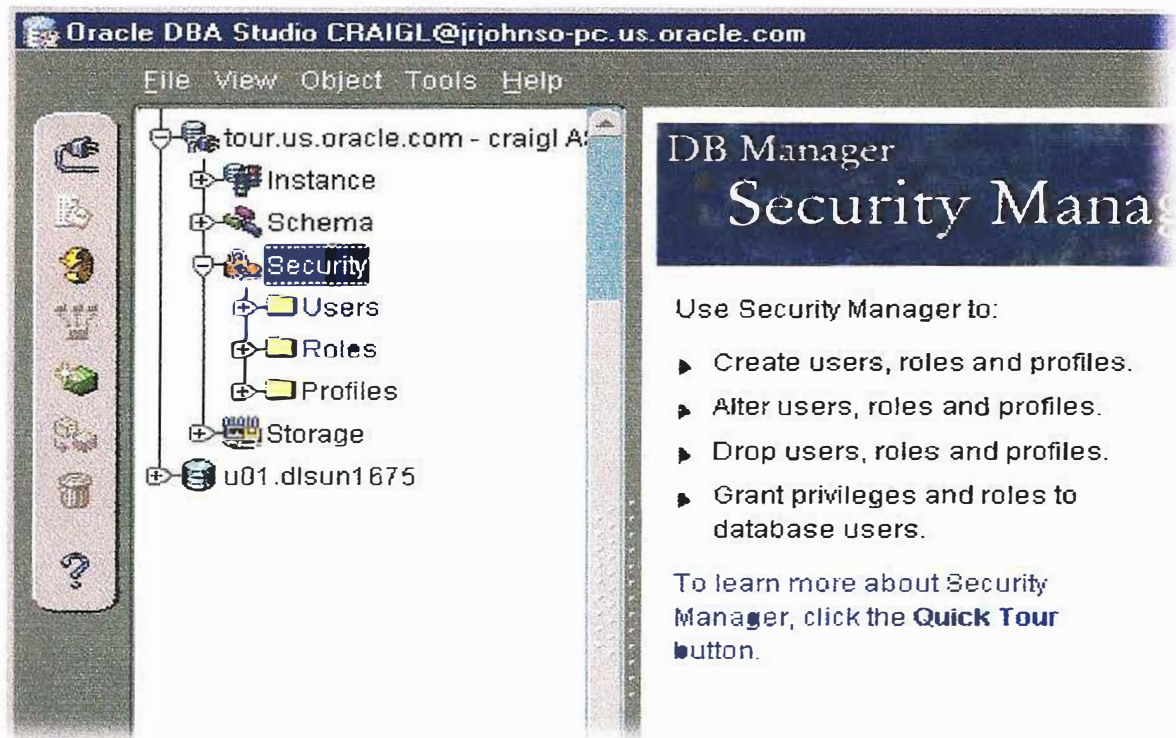
Figura 75. Security



Al desplegar el icono Security aparecen tres folders denominados:



- Ⓢ **Users:** Administra las cuentas de los usuarios
- Ⓢ **Roles:** Coloca los privilegios en roles grupales
- Ⓢ **Profiles:** Maneja los recursos limitados de la base de datos en los perfiles del usuarios.





Figura 76. Folders de Security



Los tres folders contienen listas de todos los objetos de la base de datos que pueden ser administrados por el DBA Studio Security Management. Al desplegarse un usuario se puede visualizar los roles asignados, perfiles y privilegios. También pueden desplegarse los roles y perfiles para visualizar los privilegios. Cada uno de estos objetos es identificado por un único icono.

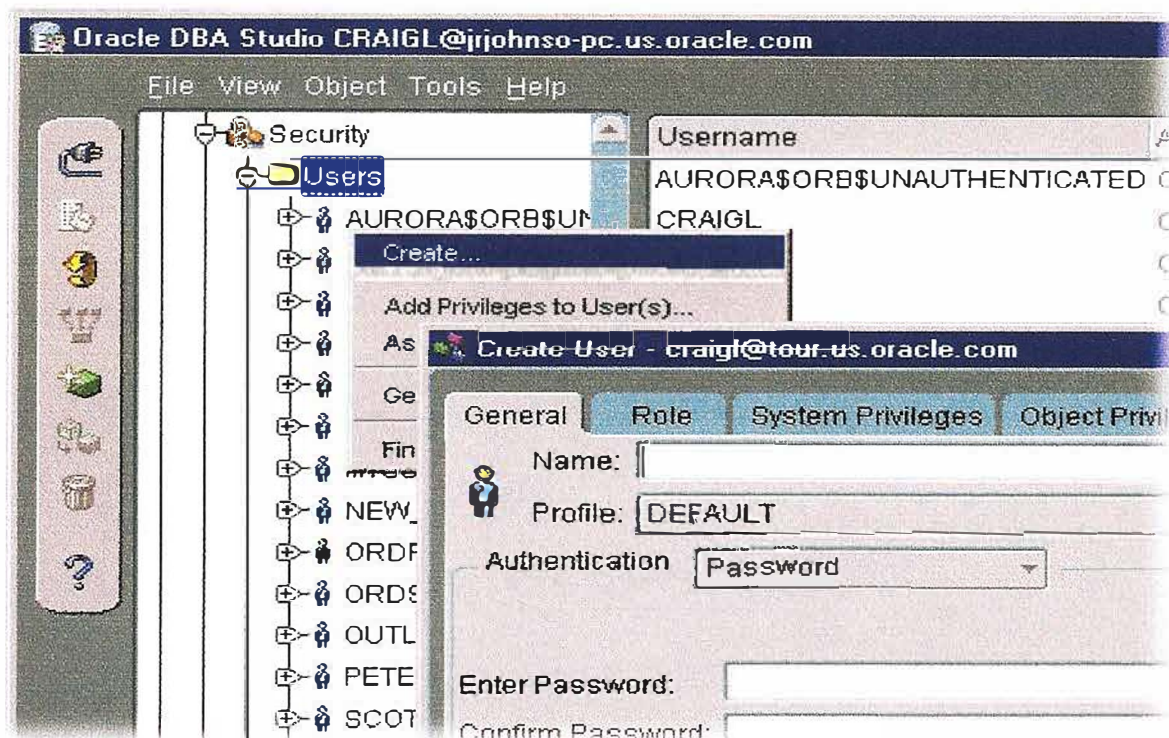
Figura 77. Iconos de los objetos

Icono	Objeto
	User
	Role or subrole

	Profile
	System privilege
	Object privilege
	Key - indica que el objeto fue transferido con la opción option/grant.

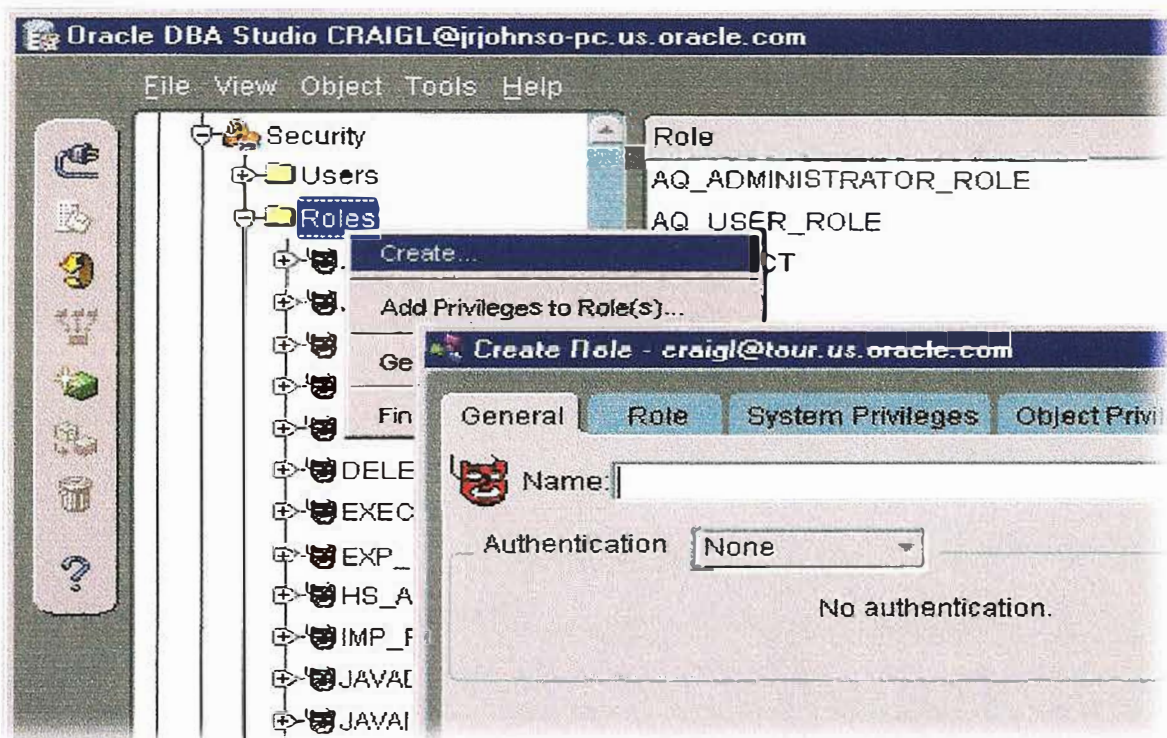
El Administrador de Seguridad permite realizar las siguientes tareas: crear un usuario, crear un usuario a partir de otro existente, modificar las propiedades de un usuario incluyendo el estado de cuentas y perfil, adicionar y quitar privilegios y roles a un usuario.

Figura 78. Crear usuario



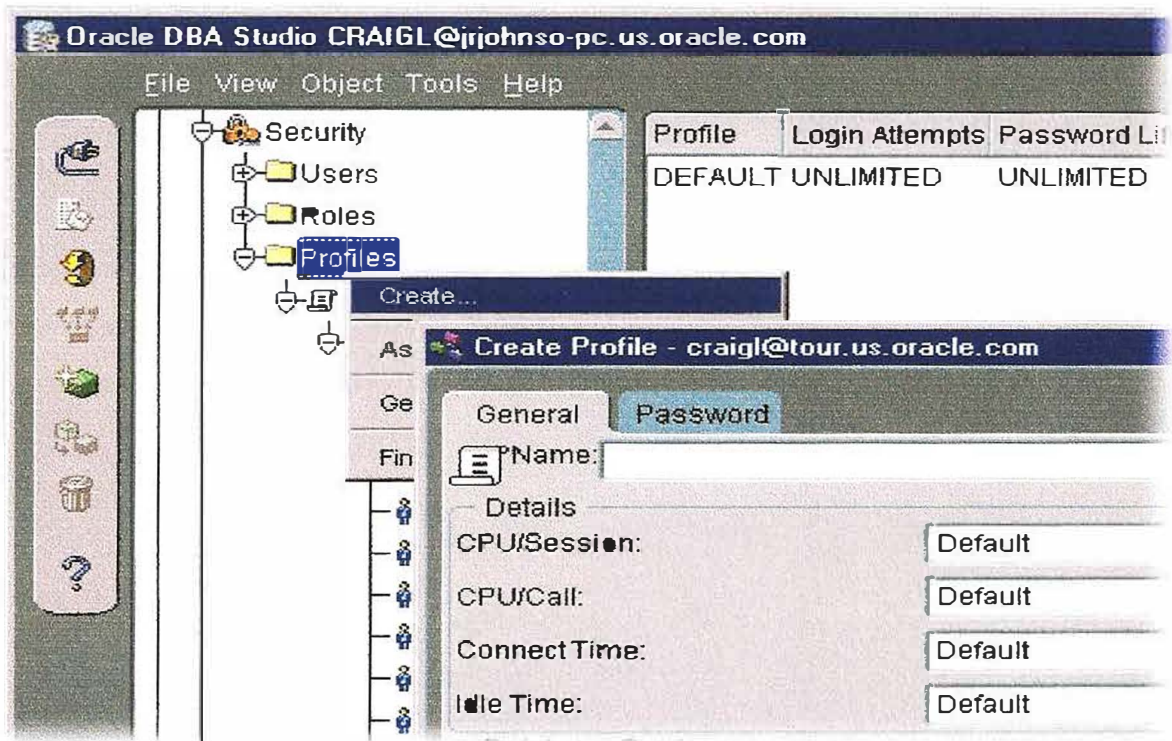
Los **roles** son utilizados para asignar privilegios a los usuarios y que les permiten acceder a diferentes objetos y operaciones. Las operaciones hechas a los roles incluyen la creación, creación de un rol a partir de otro existente, modificación de propiedades, adicionar y quitar privilegios y roles de un rol.

Figura 79. Crear rol



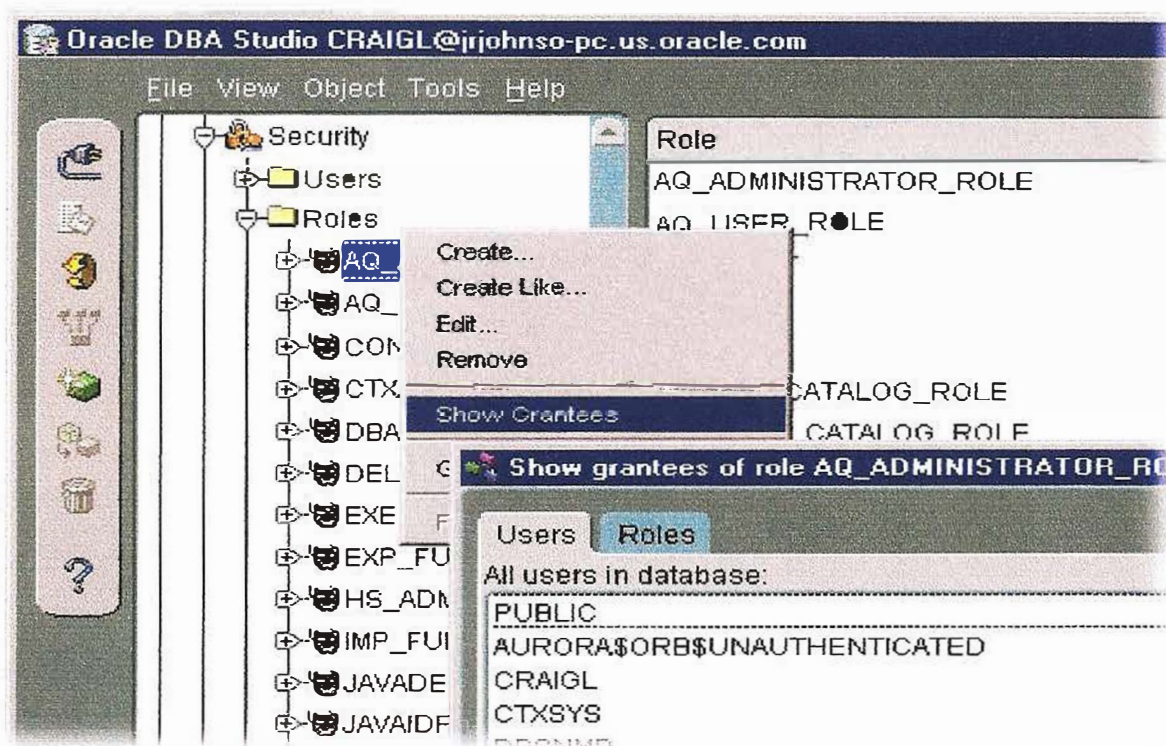
Los **perfiles** denotan la cantidad de recursos del sistema que se permite consumir a un usuario de una base de datos. El administrador puede realizar las tareas con los perfiles; crear, quitar, modificar sus propiedades, mostrar las personas a cargo del perfil y dependencias, asignar un perfil a los usuarios.

Figura 80. Crear perfil



El **Show Grantees** simplifica la administración de usuarios, haciendo click derecho en cualquier privilegio del sistema, de un objeto, o rol podemos visualizar a qué usuarios el para mostrar qué usuarios fueron otorgados qué privilegio o rol. Se puede conceder o quitar roles o privilegios a múltiples usuarios.

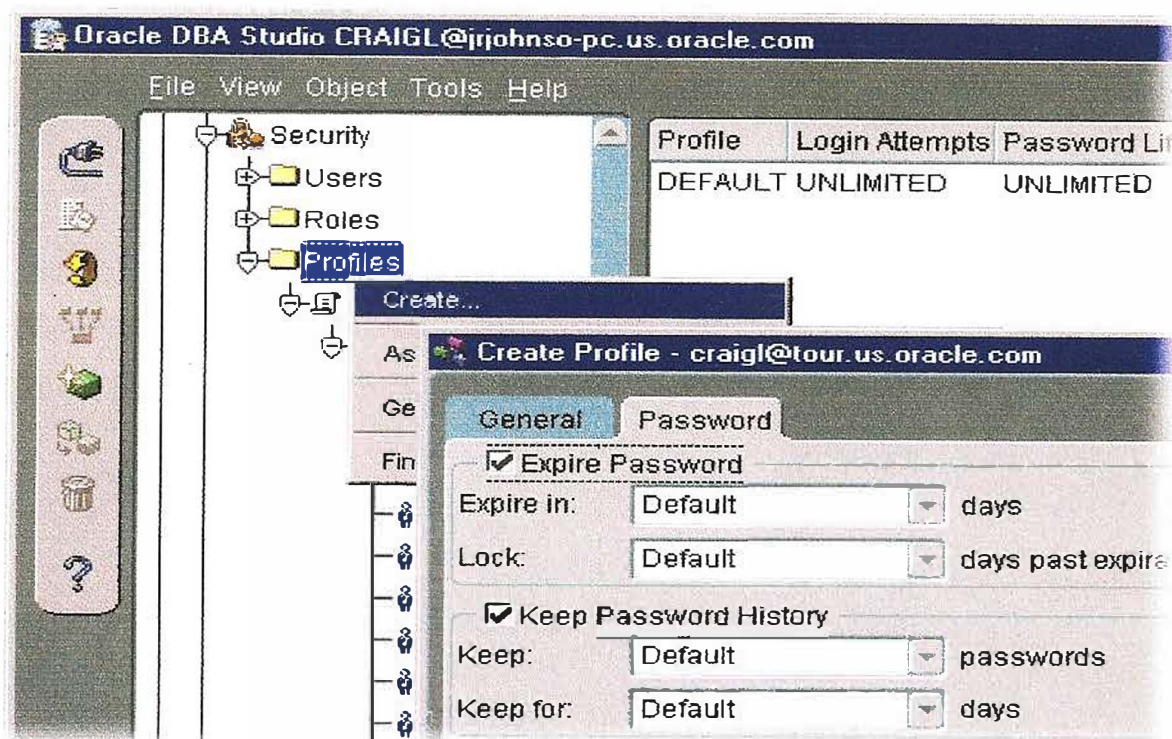
Figura 81. Show Grantees



El administrador de seguridad soporta contraseñas aumentando así la seguridad del sistema, por medio de cuentas con claves, tiempo de vida, expiración, historia, complejidad, verificaciones y export/import de contraseñas.



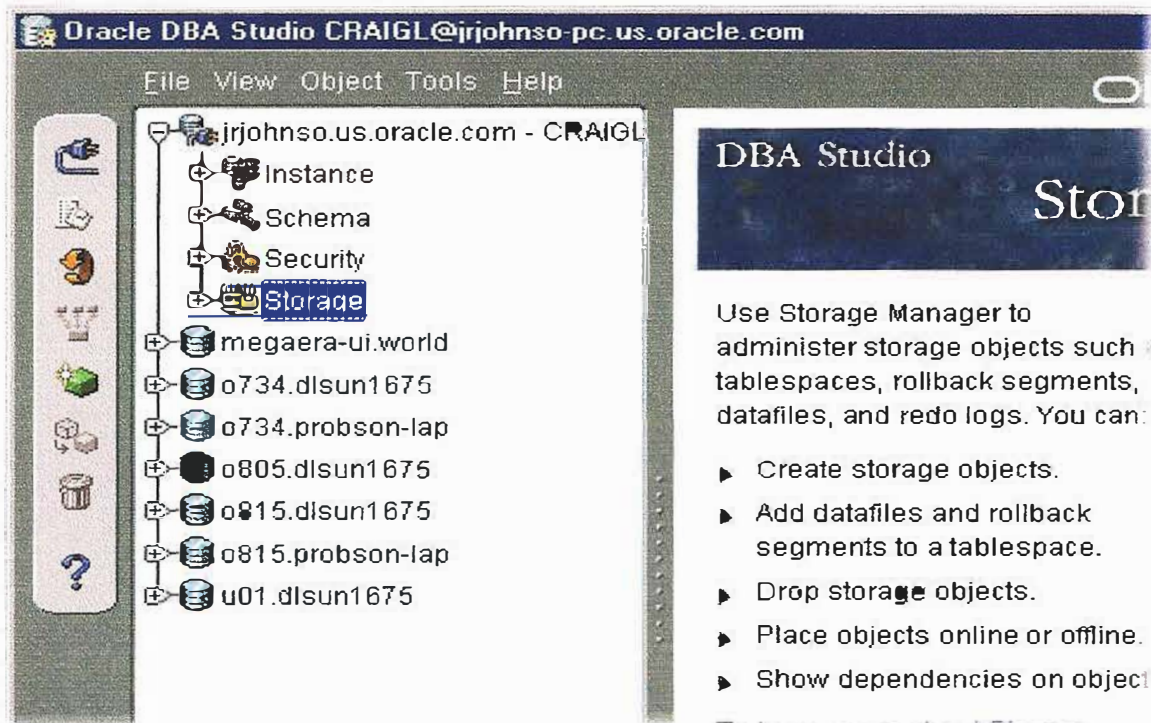
Figura 82. Show Grantees



16.4.6 Administrador de Almacén (Storage)

Con el Administrador de almacén usted puede gestionar las tablespaces, segmentos de rollback, datafiles, y redo logs, también puede ver los controlfiles y sus atributos, así como una lista de todos los archivos logs.

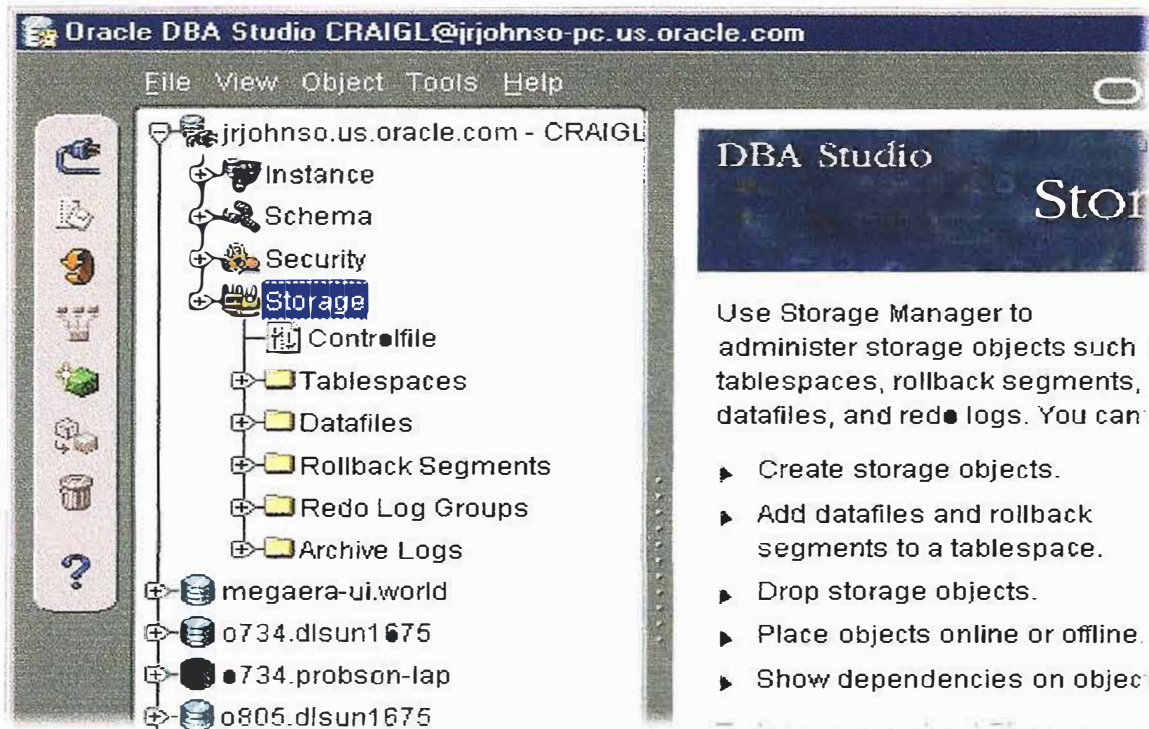
Figura 83. Storage



Desplegando la lista Storage aparecen el Controlfile, y cinco folders de gestión de almacenamiento denominados: Tablespaces, Datafiles, Rollback Segments, Redo Logs Groups y Archive Logs.

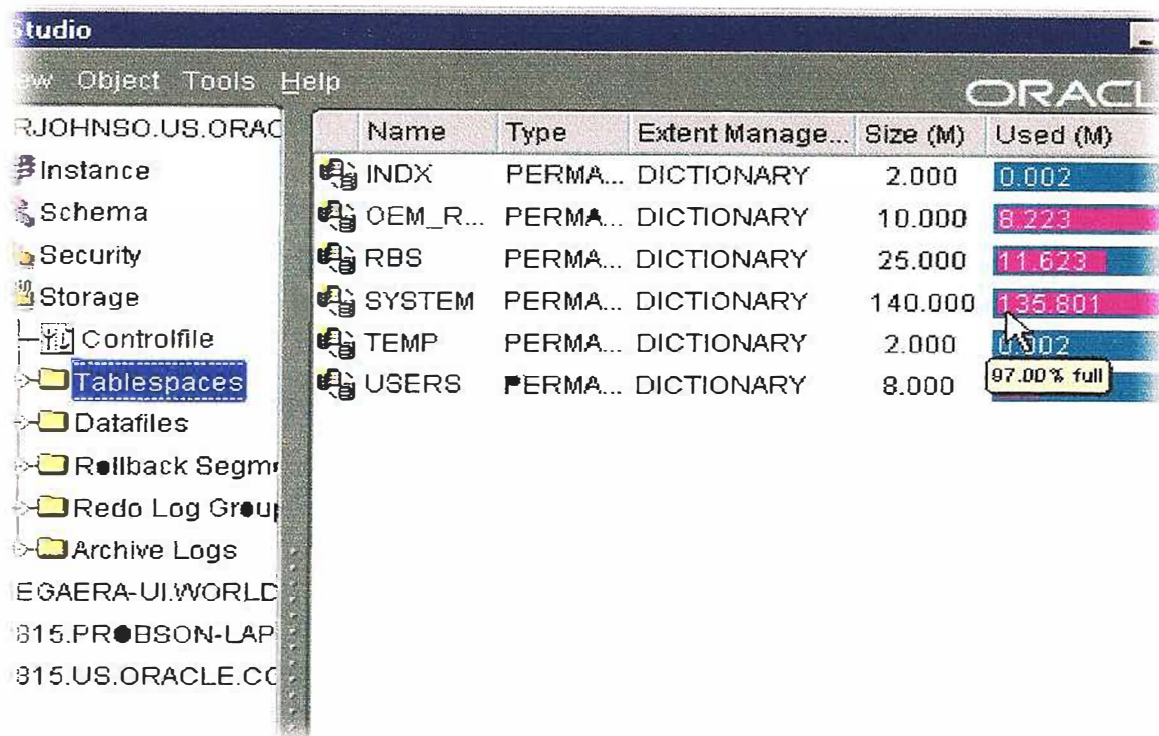
Los **controlfiles** son archivos que contienen información que se utiliza cuando se levanta una instancia, tal como la información de dónde se encuentran ubicados los datafiles y los archivos redo log. Estos archivos de control deben encontrarse siempre protegidos.

Figura 84. Listado del Storage



Los cinco folders contienen iconos ordenados alfabéticamente para el almacenamiento de objetos en la base de datos. Podemos desplegar un tablespace y mostrar los datafiles y los segmentos rollback.

Figura 85. Tablespaces

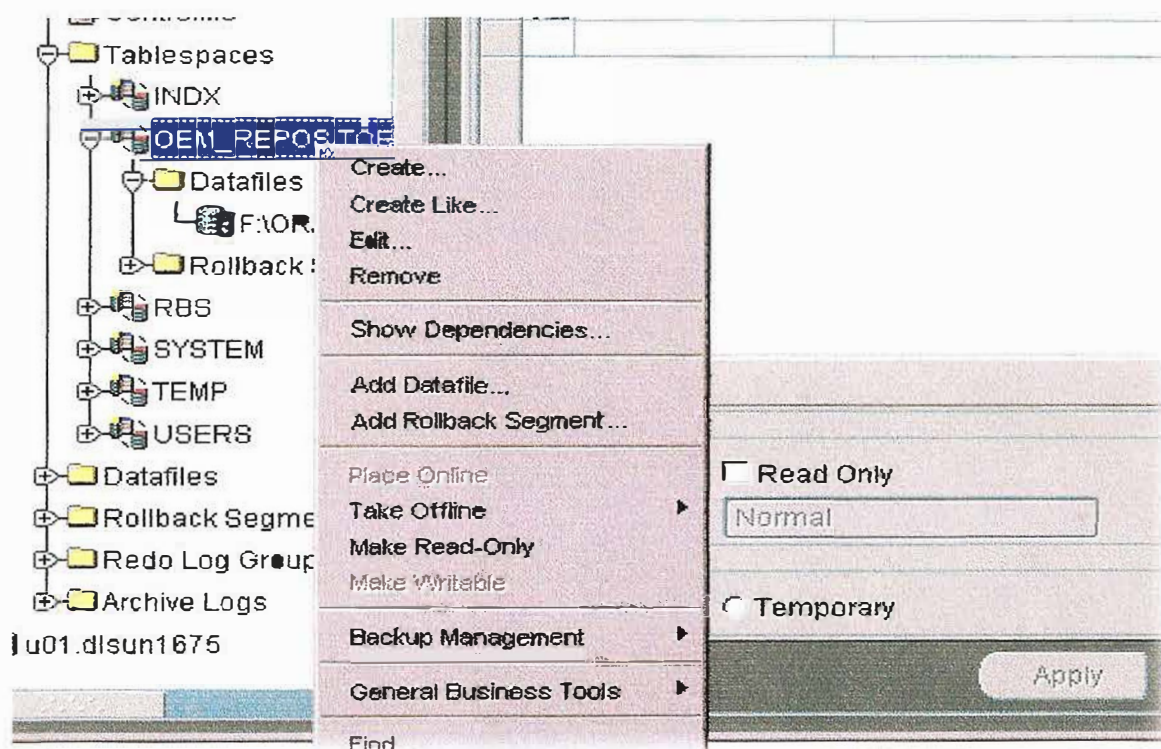


Name	Type	Extent Manage...	Size (M)	Used (M)
INDX	PERMA...	DICTIONARY	2.000	0.002
OEM_R...	PERMA...	DICTIONARY	10.000	8.223
RBS	PERMA...	DICTIONARY	25.000	11.623
SYSTEM	PERMA...	DICTIONARY	140.000	135.801
TEMP	PERMA...	DICTIONARY	2.000	0.002
USERS	PERMA...	DICTIONARY	8.000	97.00% full

Los **tablespaces** son utilizados para separar la información en grupos y así simplificar la administración de los datos y están constituidos por uno o mas datafiles.

Las operaciones que pueden realizarse sobre los tablespaces son crear, editar, borrar un tablespace; adicionar un datafile, un segmento de rollback, tomar un tablespace offline, hacer un tablespace de solo lectura o regrabable, poner parámetros de almacenamiento a cada objeto.

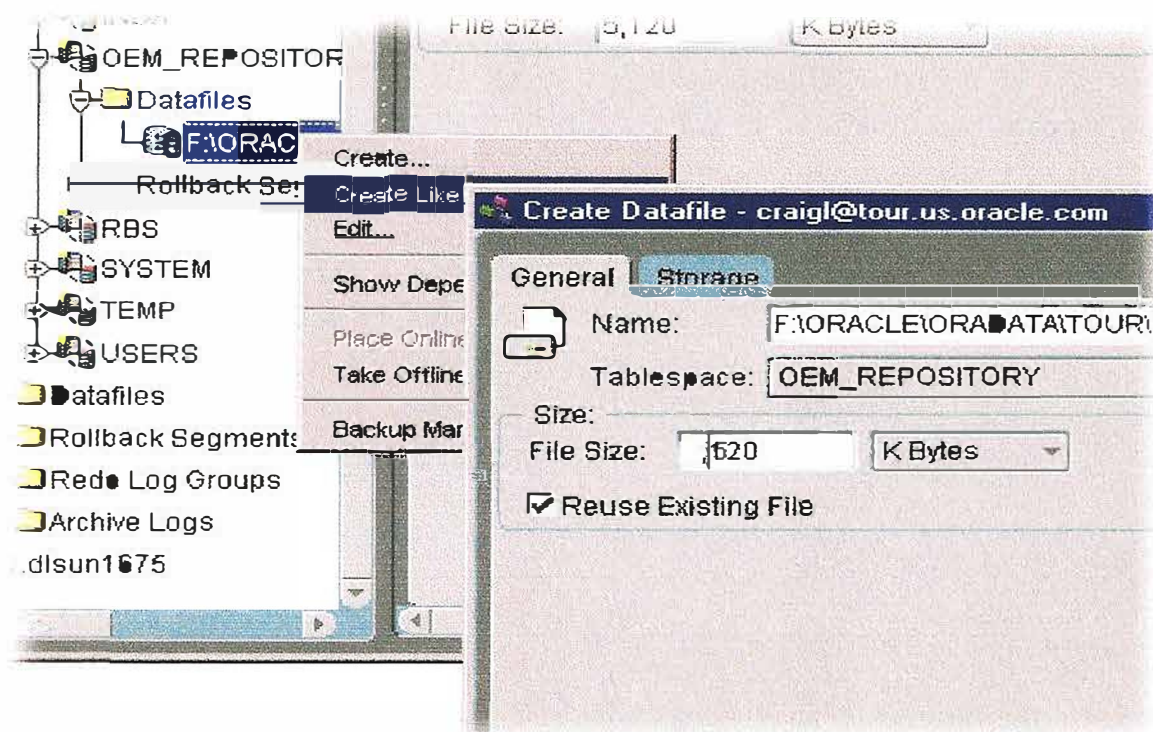
Figura 86. Operaciones con Tablespaces



Los **datafiles** almacenan toda la información ingresada en una base de datos. Se pueden tener sólo uno o cientos de ellos. Muchos objetos (tablas, índices) pueden compartir varios datafiles. El número máximo de datafiles que pueden ser configurados está limitado por el parámetro de sistema MAXDATAFILES.

Las operaciones que pueden realizarse sobre los datafile son crear un nuevo datafile, crear uno basado en otro seleccionado, editar un datafile, tomar un datafile offline.

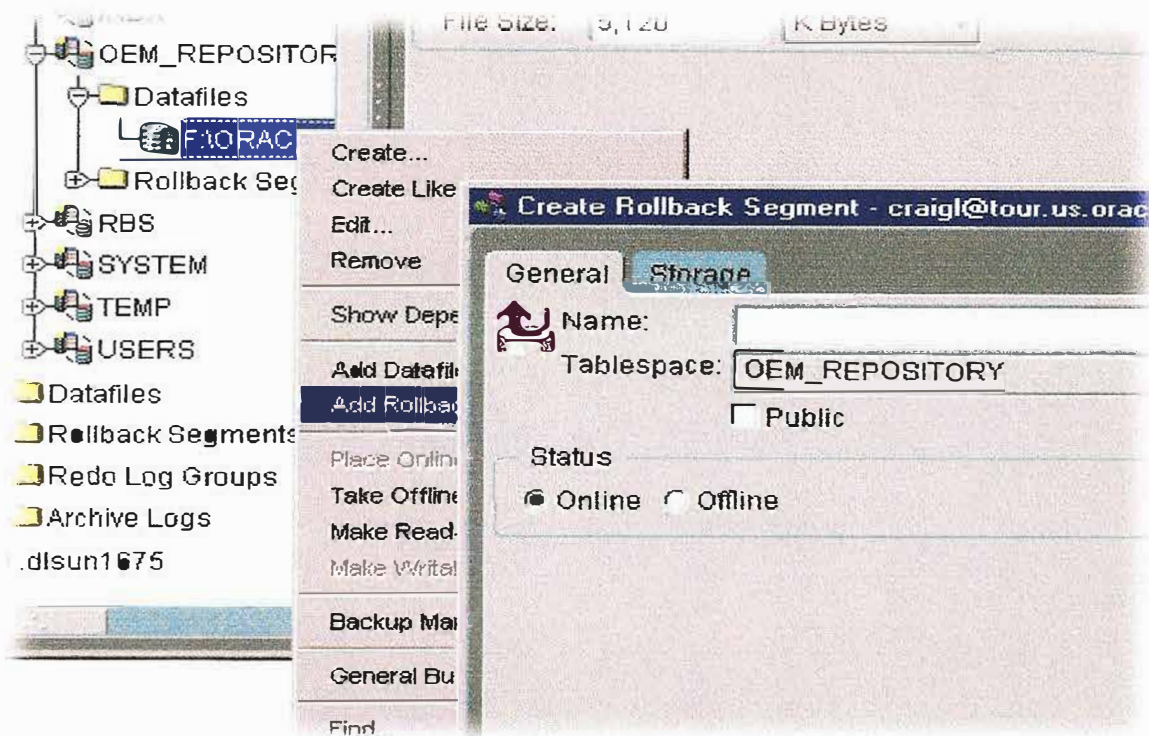
Figura 87. Operaciones con Datafiles



Los **segmentos de rollback** son áreas lógicas de la base de datos que contienen información de las transacciones que se encuentran en curso y que aún no han sido confirmadas o deshechas.

Las operaciones que pueden realizarse sobre los segmentos de rollback son crear, modificar, borrar, retirar un segmento de rollback y tomar un segmento de rollback offline.

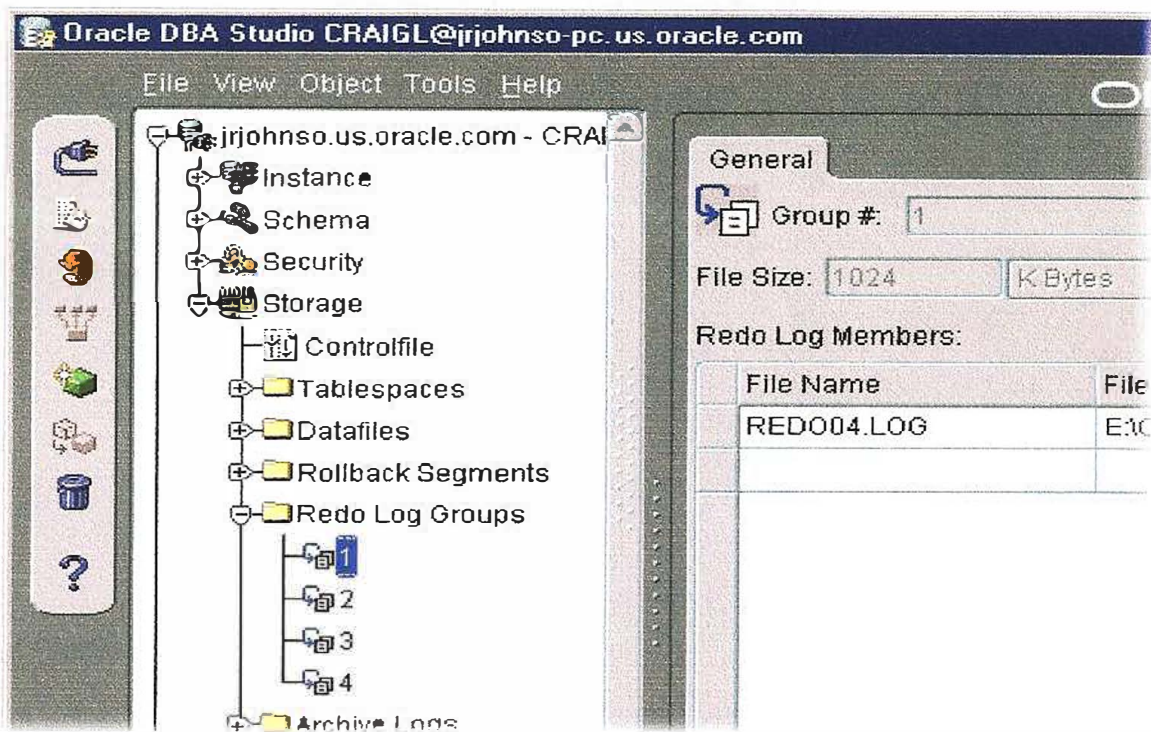
Figura 88. Operaciones con Segmentos de Rollback



Los archivos del tipo **redo log** almacenan información que se utiliza para la recuperación de una base de datos en caso de falla. Estos archivos almacenan la historia de cambios efectuados sobre la base de datos y son particularmente útiles cuando se necesita corroborar si los cambios que la base de datos ya ha confirmado se han efectuado realmente en los datafiles.

Las operaciones que pueden realizarse sobre los redo log group son cambiar los redo log group actuales, activar un punto en un redo log group, crear un nuevo redo log group, renombrar, quitar, o agregue nuevos redo log members.

Figura 89. Operaciones con Redo Log Group



17. MICROSOFT ACCESS

Access es un sistema interactivo de administración de bases de datos para Windows, miembro del Paquete Office, que permite crear, administrar y utilizar ***“BASES DE DATOS”***.

Access, como miembro del paquete Office, de la familia Microsoft, es, uno de los sistemas más populares para el tratamiento de información en archivos de formato “Base de Datos”.

Potente y muy utilizada herramienta de manipulación de bases de datos creada por Microsoft la cual permite a los usuarios expertos y novatos crear bases de datos de una manera fácil e ilustrativa. También tiene asistentes capaces de crear formularios y consultas. Se apoya en el lenguaje de programación Visual Basic y es orientada a objetos. Es fácil de conectar a través del ODBC de Windows para su uso en otras plataformas.

17.1 CARACTERÍSTICAS PRINCIPALES

Entre sus principales características se encuentran:

- ☉ Access es gráfico, por lo que aprovecha al máximo la potencia gráfica de Windows, ofreciendo métodos usuales de acceso a los datos y proporcionando métodos simples y directos de trabajar con la información.
- ☉ Access facilita la administración de datos, ya que sus posibilidades de consulta y conexión le ayudan a encontrar rápidamente la información deseada, cualquiera que sea su formato o lugar de almacenamiento.

- Con Access es posible producir formularios e informes sofisticados y efectivos, así como gráficos y combinaciones de informes en un solo documento.
- Access permite lograr un considerable aumento en la productividad mediante el uso de los asistentes y las macros. Estos permiten automatizar fácilmente muchas tareas sin necesidad de programar.
- Compatibilidad con XML.

17.2 INSTALACIÓN

Tabla 12. Requisitos para instalación de Microsoft Access

COMPONENTES	REQUISITOS
Equipo y Procesador	Se recomienda un equipo con un procesador Intel Pentium de 233 (MHz) o superior. Se recomienda la utilización de Pentium III.
Memoria	Se recomienda 128 megabytes (MB) de RAM como mínimo.
Disco Duro	El espacio de disco duro que se utiliza depende de la configuración realizada, las diversas opciones de instalación requieren más o menos espacio. A continuación hay una lista con los requisitos de espacio de cada edición de Office 2003. Para la instalación opcional de caché de archivos (recomendado) se necesitan 250 MB más de espacio libre en el disco duro
Sistema Operativo	Microsoft Windows (R) 2000 con Service Pack 3 (SP3) o posterior o Windows XP o posterior, Windows 98.
Pantalla	Monitor con resolución Super VGA (800 x 600) o superior

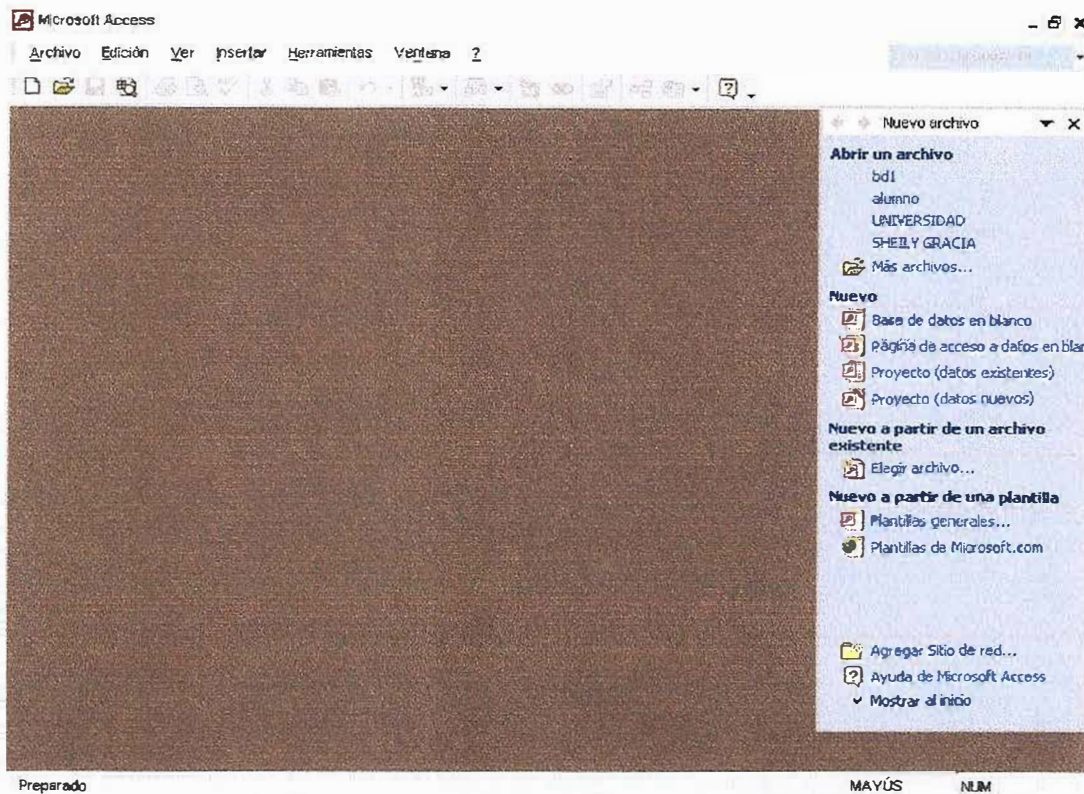
17.2.1 Proceso de Instalación

En el Panel de control de Windows, haga doble clic en el icono **Agregar o quitar programas**. Siga los pasos indicados por el propio proceso de instalación. Puede ahorrarse espacio en el disco eligiendo la opción de instalación

“Completa/Personalizada” y seleccionando sólo los componentes que desee instalar.

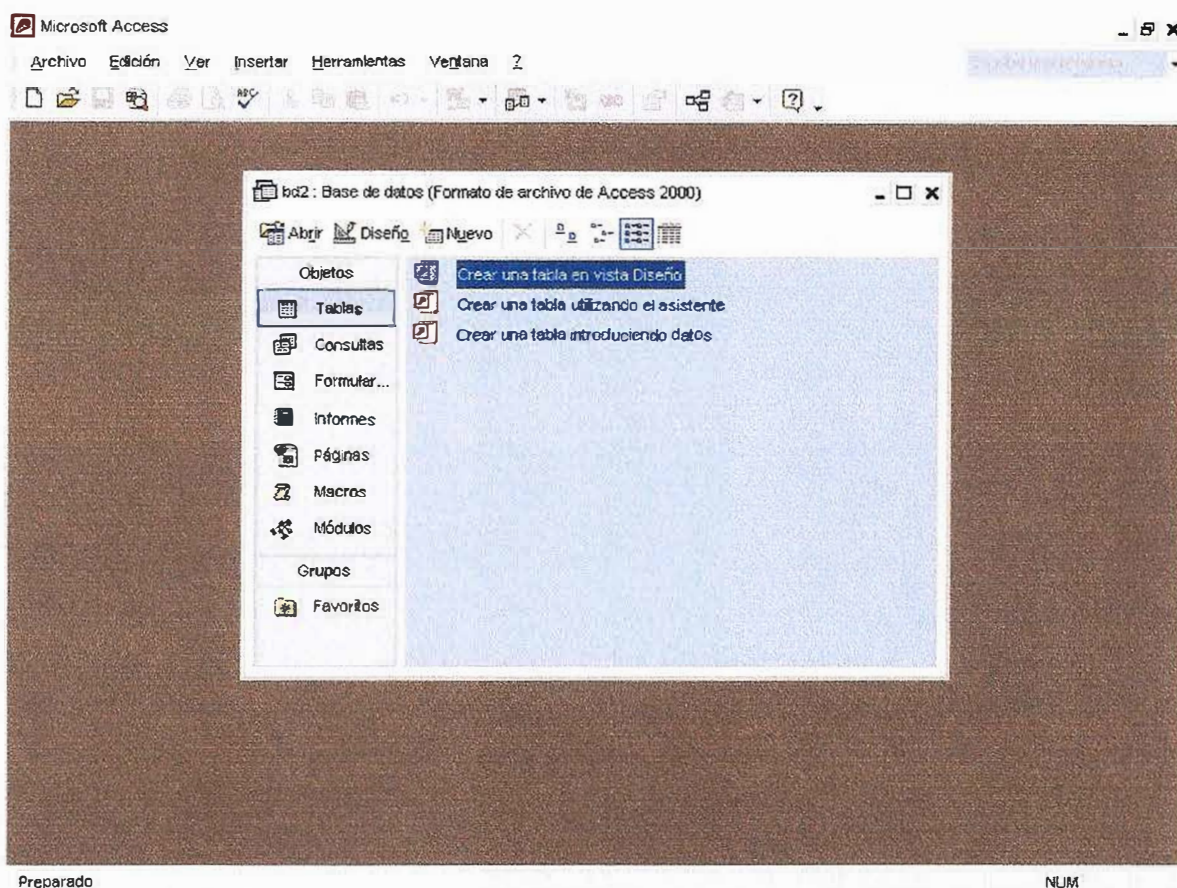
17.3 CREACIÓN DE BASES DE DATOS

Figura 90. Pantalla Inicial de Access



Access, inicialmente, solicitará información, sobre el nombre a asignar a su base de datos (sugiriendo “BD1”), asignará la extensión **.MDB**, y la carpeta donde se guardará. Una vez definido esto, mostrará la siguiente pantalla:

Figura 91. Creación de Base de datos



La ventana, muestra, a la izquierda **SIETE OBJETOS**, de los cuales, los más habitualmente utilizados son **TABLAS, FORMULARIOS E INFORMES**, Y en la parte de la derecha, dependiendo del objeto seleccionado a la izquierda, Access nos permitirá crear o modificarlos. En nuestro caso el objeto seleccionado es la **Tabla**, principal elemento de cualquier base de datos ya que todos los demás objetos se crean a partir de éstas.

17.3.1 Creación de Tablas

Para crear las tablas, que son los objetos más importantes de cualquier base de datos podemos realizarlo de dos formas:

- ④ Uno utilizando un asistente que va guiando paso por paso la creación de la tabla.
- ④ Otro empezando la tabla en blanco.


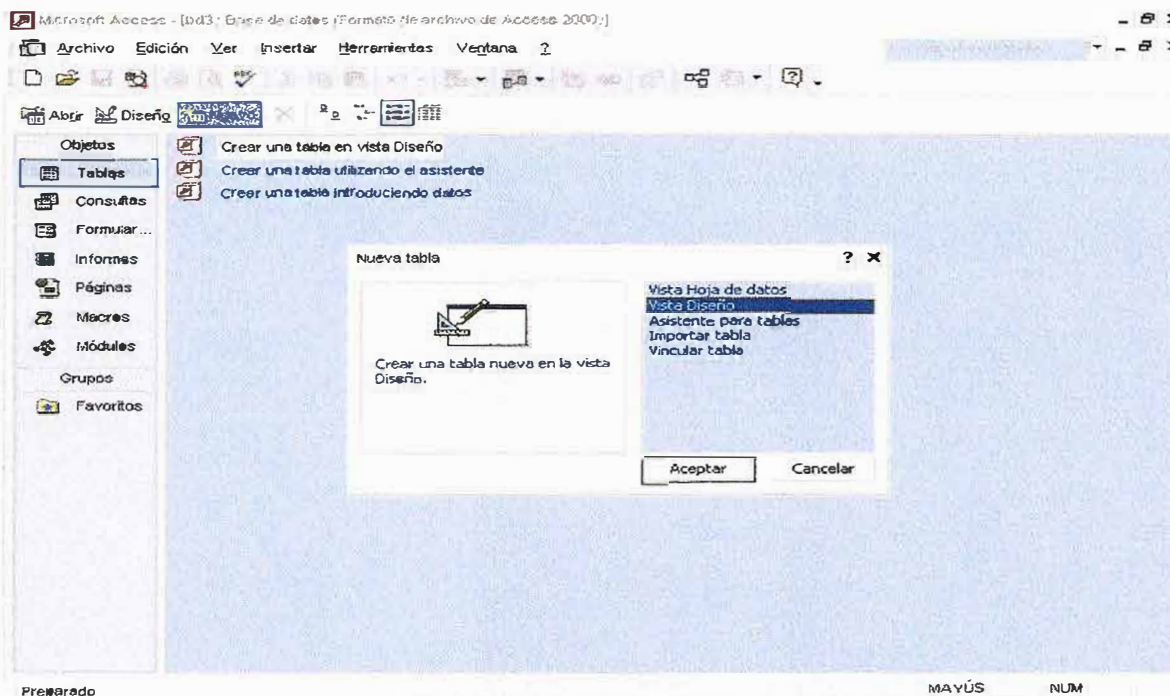
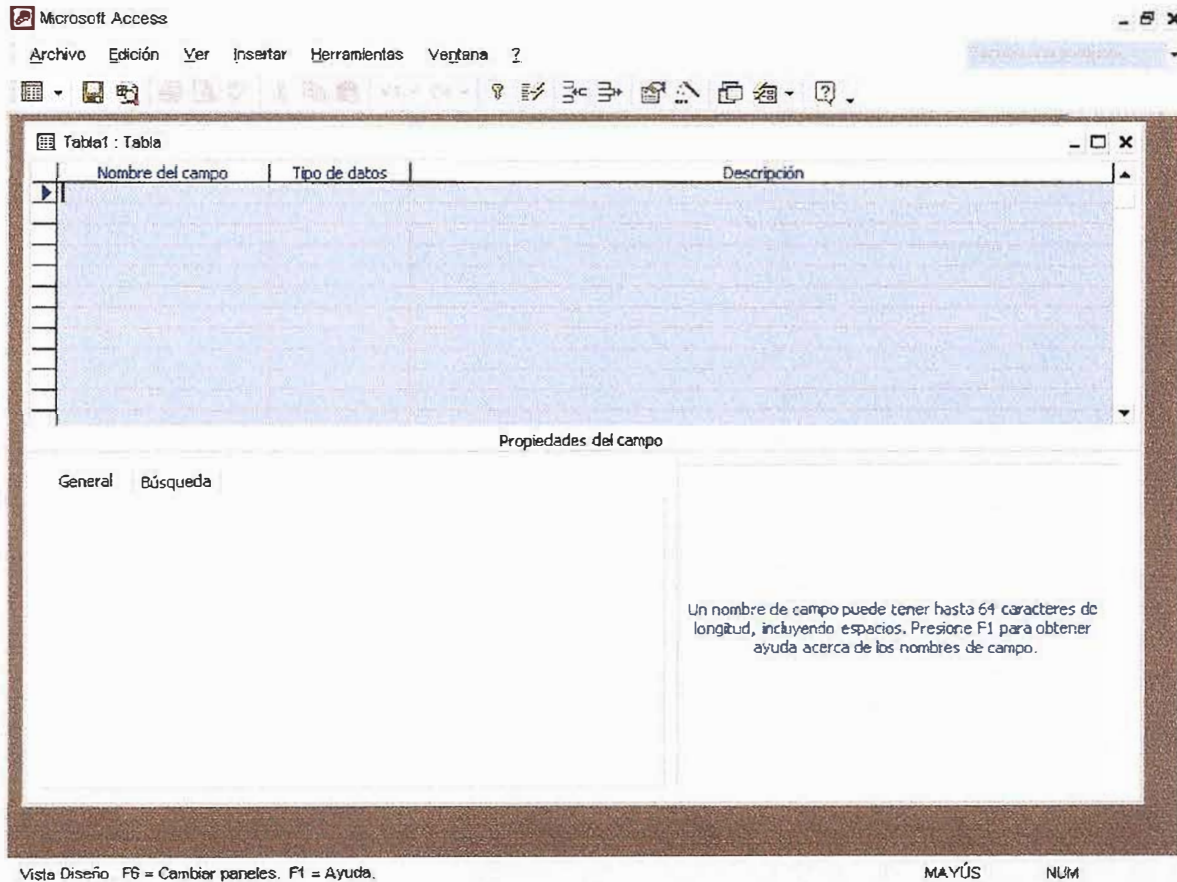
Tomando la segunda forma: Seleccionamos la opción **Crear una tabla en vista diseño** de la ventana de la base de datos. Hacemos clic sobre el botón Nuevo  de la ventana de la base de datos. Seleccionar **Vista Diseño**, y por ultimo clic sobre el botón **Aceptar**.

Figura 92. Creación de tablas en Vista Diseño



Aparecerá la ventana de diseño de la tabla:

Figura 93. Diseño de la tabla



Cada fila se utiliza para la creación de los campos de la tabla, indicando, nombre, tipo de dato, descripción, y propiedades.

17.3.2 Insertar Datos

Para visualizar los campos de una tabla se selecciona esta en la ventana de base de datos y se da click en el icono **Diseño** encerrada en el círculo.

Figura 94. Click en el icono Diseño

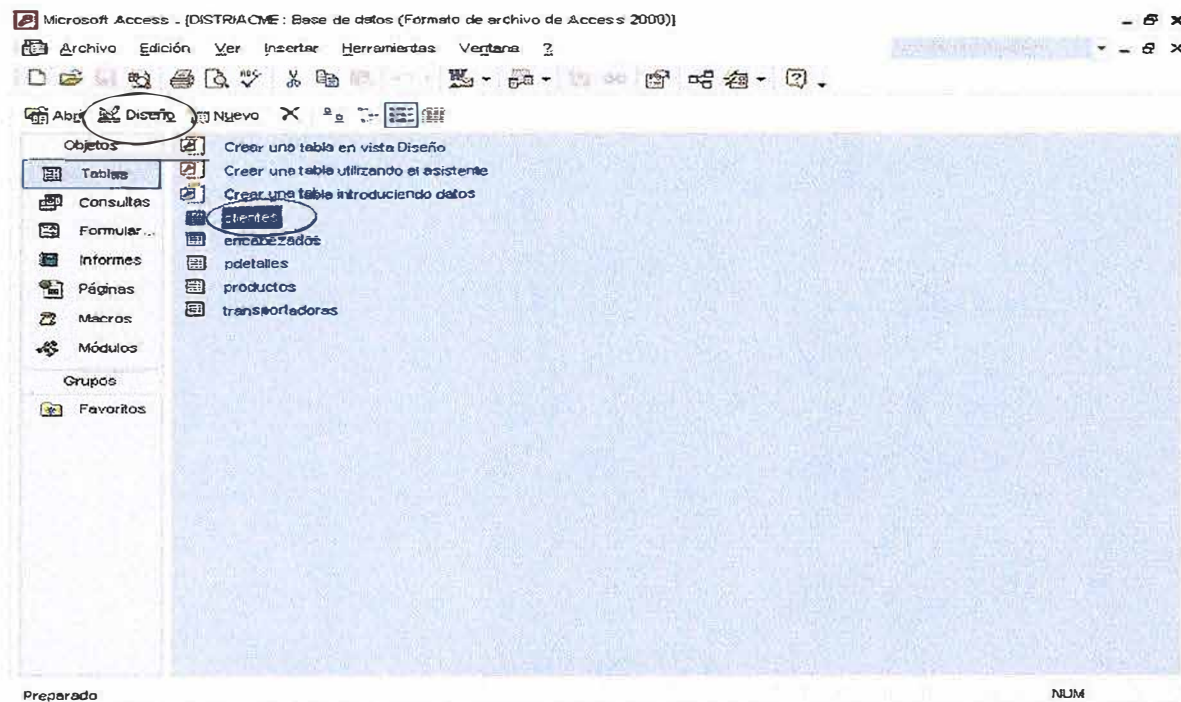
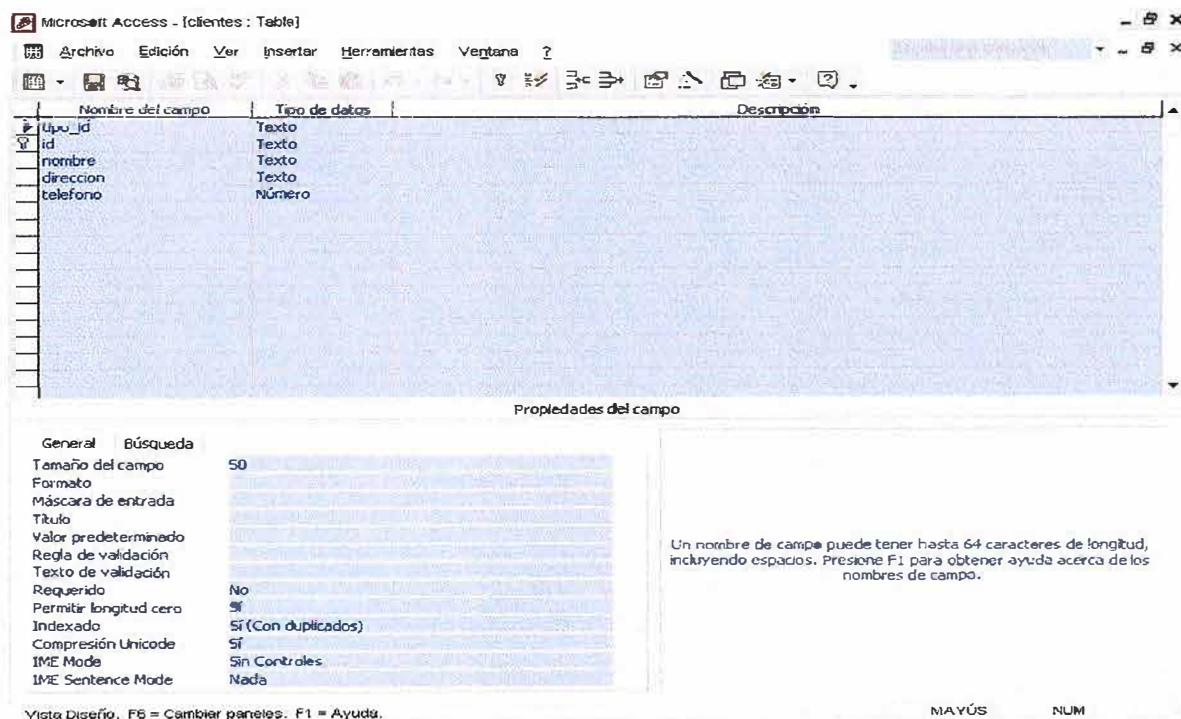


Figura 95. Visualizar los campos de la tabla



Ahora bien, si lo que se quiere es ingresar datos a la tabla, se da click en el icono **Abrir** de la ventana de la base de datos.

Figura 96. Click en el icono Abrir

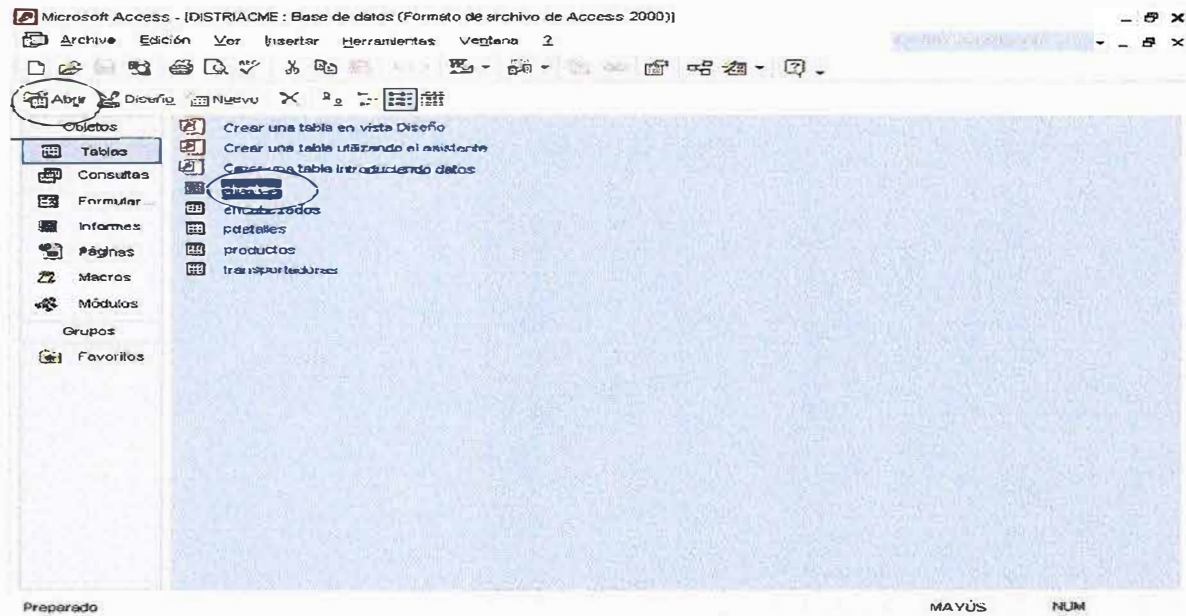
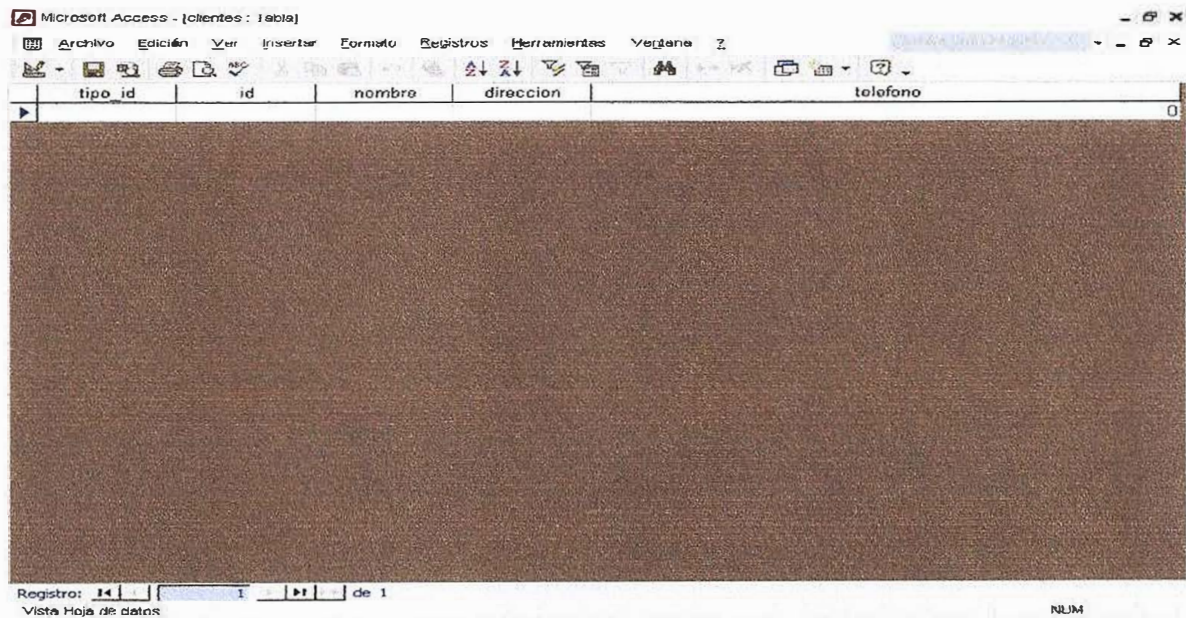
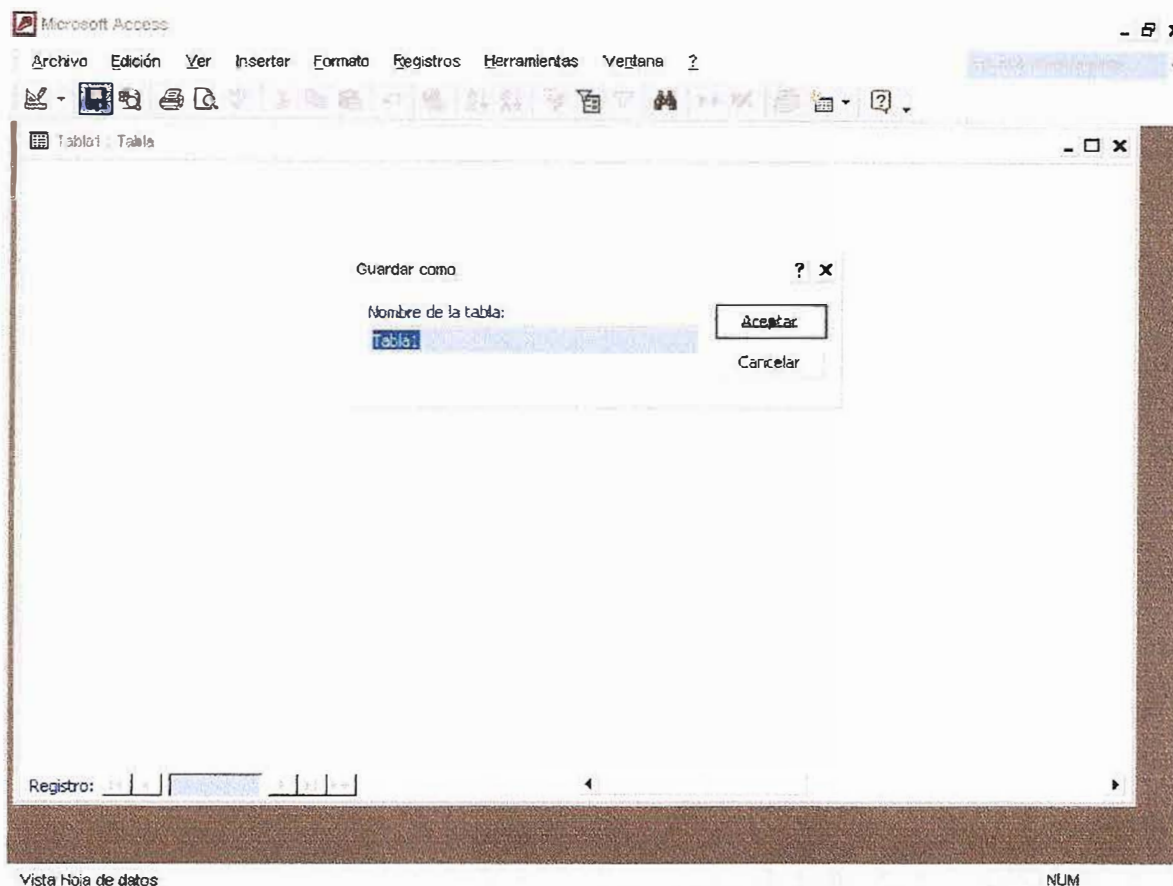


Figura 97. Ingresar datos en la tabla



Una vez creada la tabla con sus respectivos campos, tipo de dato y propiedades procedemos a guardarla:

Figura 98. Guardar la tabla



17.4 FILTROS

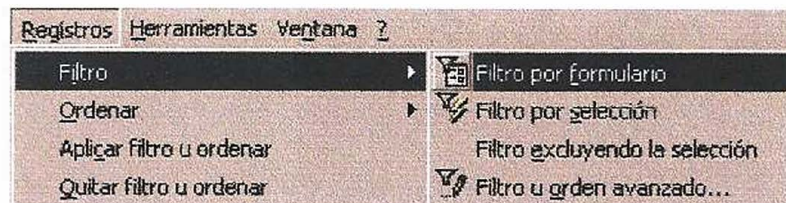
Un filtro es una selección de algunos elementos de una tabla o de un formulario. Aplicando un filtro es posible ver dentro de una tabla sólo aquellos registros que cumplen unas condiciones. El resto de los registros no desaparecen, sólo están ocultos.

Access tiene tres tipos de filtros:

- Ⓒ Filtro por selección
- Ⓒ Filtro por formulario
- Ⓒ Filtro u orden avanzado

Para aplicar un filtro se selecciona el menú Registros|Filtro:

Figura 99. Menu Registro/Filtro



17.4.1 Filtro por Selección

Este filtro se aplica seleccionando dentro de la tabla el elemento que va a ser la condición de filtrado. Cuando la tabla o el formulario no es muy grande es sencillo de utilizar.

Si se quiere a aplicar un filtro para mostrar sólo aquellos clientes que se apelliden de determinada manera, lo primero es situarse sobre cualquier registro que tenga ese apellido. Después se pulsa el botón de **Filtro por selección**. De la tabla aparecerán sólo aquellos registros en los cuales el campo apellidos sea el que usted escogió, el resto desaparecerán.

Para volver a ver todos los registros de la tabla bastará con pulsar sobre el botón de **Quitar filtro**.

17.4.2 Filtro por Formulario

Es un filtro de mucha utilidad para tablas grandes y con un gran número de registros. Tras pulsar sobre el botón de ***filtro por formulario***, el menú y la barra de herramientas variarán.

Pulsando en la primera fila de cada columna aparecerá **una flecha** en la zona derecha de la celda. Pulsando sobre esta flecha se despliega la lista de todos los valores archivados en ese campo. Si se prefiere se puede escribir directamente el valor exacto que se está buscando o la expresión cuyo resultado se desea utilizar como criterio.

17.4.3 Filtro u Orden Avanzado

Este filtro permite determinar con mayor minucia las características del filtro. Es muy similar a una consulta, y se almacena en Consultas. Para aplicar este filtro es necesario seleccionar el menú **Registros|Filtro** u orden avanzado ya que no hay ningún botón en la barra de herramientas con esta función.

Al igual que en el filtro por formulario, el menú y la barra de herramientas son propios de los filtros. El resto de la pantalla se divide en dos. En la parte superior se encuentra la tabla sobre la que se está aplicando el filtro, con la lista de campos recuadrada. En la parte inferior es donde se concretan las características del filtro.

Para crear el filtro hay que detallar el campo, el orden a seguir y los criterios:

Ⓔ **Campo:** para incluir el campo o los campos con los que se va a crear el filtro, se pueden emplear tres procedimientos distintos:

1. Al hacer clic sobre la casilla campo aparece una flechita dentro, que despliega la lista de todos los campos de la tabla.
2. Seleccionando la tabla en la zona superior y haciendo doble clic sobre el campo.
3. Seleccionando la tabla en la zona superior y arrastrando el campo hacia la zona inferior.

Se pueden utilizar tantos campos como se consideren necesario para crear el filtro. Para seleccionar todos los campos en una celda se selecciona el *, el filtrado se realizará en todos los campos de la tabla.

Ⓕ **Orden:** Establece el orden que se seguirá cuando se encuentre más de un registro que cumpla las condiciones del filtro. El orden puede ser ascendente, descendente o simplemente no seguir un orden.

Ⓖ **Criterios:** En esta casilla se escribe el criterio que debe seguir el filtro. Se puede escribir:

Un texto: se teclea y el programa lo pondrá automáticamente entre comillas.

Una expresión: utilizando los operadores "Entre...Y" o los operadores de comparación (<, >, <>, <=, >=) .

Para volver a ver todos los registros de la tabla se vuelve a pulsar el botón **Quitar Filtro**. Si se piensa que en un futuro puede ser práctico volver a ver sólo estos datos, se puede almacenar el filtro para aplicarlo. Pero en la tabla, se estarán guardando todos los registros por mucho que se guarde el filtro. Porque aunque el resto de los registros no se vean, siguen estando ahí.

18. POSTGRESQL 7.1

18.1 HISTORIA DE POSTGRESQL

PostGreSQL (llamado también Postgres95) fue derivado del proyecto Postgres, A sus espaldas, este proyecto lleva más de una década de desarrollo, siendo hoy en día, el sistema libre más avanzado con diferencia, soportando la gran mayoría de las transacciones SQL, control concurrente, teniendo a su disposición varios "language bindings" como por ejemplo C, C++, Java, Python, PHP y muchos más.

La implementación de Postgres DBMS comenzó en 1986, y no hubo una versión operativa hasta 1987. La versión 1.0 fue liberada en Junio de 1989 a unos pocos usuarios, tras la cual se liberó la versión 2.0 en Junio de 1990 debido a unas críticas sobre el sistema de reglas, que obligó a su reimplementación. La versión 3.0 apareció en el año 1991, e incluyó una serie de mejoras como una mayor eficiencia en el ejecutor de peticiones. El resto de versiones liberadas a partir de entonces, se centraron en la portabilidad del sistema. El proyecto se dio por finalizado en con la versión 4.2, debido al gran auge que estaba teniendo, lo cual causó la imposibilidad de mantenimiento por parte de los desarrolladores.

En 1994, Andrew Yu y Jolly Chen añadieron un intérprete de SQL a este gestor. Postgres95 fue liberado a Internet como un proyecto libre (OpenSource). Estaba escrito totalmente en C, y la primera versión fue un 25% más pequeña que Postgres, y entre un 30 y un 50% más rápida. A parte de la corrección de algunos bugs, se mejoró el motor interno, se añadió un nuevo programa monitor, y se

compiló usando la utilidad GNU Make y el compilador Gcc sin necesidad de parchearlo.

En 1996, los desarrolladores decidieron cambiarle el nombre al DBMS, y lo llamaron PostGreSQL (versión 6.0) para reflejar la relación entre Postgres y las versiones recientes de SQL. Se crearon nuevas mejoras y modificaciones, que repercutieron en un 20-40% más de eficiencia, así como la incorporación del estándar SQL92.

18.2 ¿QUÉ ES POSTGRESQL?

El PostgreSQL es un sistema manejador de base de datos relacional orientada a objetos (ORDBMS) basado en el proyecto POSTGRES, de la universidad de Berkeley - California en el Departamento de Computer Science. El director de este proyecto es el profesor Michael Stonebraker, y fue patrocinado por Defense Advanced Research Projects Agency (DARPA), el Army Research Office (ARO), el National Science Foundation (NSF), y ESL, Inc

PostGreSQL es una derivación libre (OpenSource) de este proyecto, y utiliza el lenguaje SQL92/SQL99. Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional actual, incluido en otros sistemas de gestión comerciales. PostGreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional. A pesar de esto, PostGreSQL no es un sistema de gestión de bases de datos puramente orientado a objetos.

PostgreSQL tiene **Licencia BSD (Berkeley Software Distribution)**: particularmente se destacan las modificaciones que se introdujeron al sistema

Unix en el curso de una serie de proyectos de investigación. Básicamente deja al receptor del software plena libertad en su uso, para cualquier fin, siempre que mantenga las notas de copyright de los autores.

Si *MySQL* es la base de datos número uno en popularidad en el mundo *Linux*, *PostgreSQL* es el número dos.

PostgreSQL se diseñó como una base de datos orientada a objetos, esto significa, que las tablas no son tablas, sino objetos, y las *tuplas* son instancias de ese objeto. Se puede crear nuevos tipos de datos, hacer herencias entre objetos.

Ofrece muchos rasgos modernos como son:

- ⌚ Queries complejos
- ⌚ Llaves foráneas
- ⌚ Triggers
- ⌚ Vistas
- ⌚ Integridad transaccional
- ⌚ Control de coincidencia multiversión

También puede ser ampliado en el usuario desde muchos puntos de vista, por ejemplo añadiendo nuevos:

- ⌚ Tipos de datos
- ⌚ Funciones
- ⌚ Operadores
- ⌚ Funciones agregadas
- ⌚ Métodos de indexación
- ⌚ Lenguaje procedimental

PostgreSQL tiene todo de lo que carece *MySQL*, pero también le falta todo lo que tiene *MySQL*. *PostgreSQL* tiene transacciones, integridad referencial, vistas, y multitud de funcionalidades, pero es lento y pesado.

Con la aparición de las versiones 7.X, los de *PostgreSQL* argumentaron que empezaba una nueva era: más rápido, más fiable, etc. En la práctica continúan más o menos igual.

Han incorporado una maravilla llamada *MVCC* (*multiversion concurrency control*) con lo que los bloqueos de escritura actúan sólo en la sesión del cliente, no en las de los demás clientes. También tiene soporte de *Full-Text-indexing* a través de un *trigger* incluido en la distribución. El límite es de 32k por fila, hasta la versión 7.1 es de 8k.

Detrás de todo eso siguen existiendo errores: el *daemon* se sigue muriendo de vez en cuando, siguen no siendo *multithreading* (cada conexión es un proceso), los procesos siguen dejando lagos de memoria y conexiones abiertas, además de la gran cantidad de memoria que requieren.

Tampoco acaban de arreglar el tema de los permisos de acceso: un usuario con permiso de conexión a una base de datos, puede crear tablas en ella. Este error ya ha acarreado problemas de tipo *DoS* (*Denied of Service*) en servidores *Web*, pero siguen sin arreglarlo.

Todo esto hace de *PostgreSQL* una *RDBMS* no viable para un servidor de *Internet* tipo *Slashdot*, pero si para un servidor con pocas conexiones pero que requiera de muchas funcionalidades como un *e-commerce*. *PostgreSQL* está diseñado para realizar pocas operaciones a la vez, pero complejas, no muchas operaciones sencillas.

Apache crea nuevas instancias cada vez que recibe nuevos clientes. Cada instancia de *apache* puede soportar varios clientes *http* (150 por defecto). Pero el *apache*, para cada cliente que pide una página dinámica, que requiere un acceso a la base de datos, tiene que abrir conexión con el *PostgreSQL*. Al no ser *Multithreading*, el *PostgreSQL* crea una nueva instancia. Pero las consultas son lentas y se acumulan rápidamente las instancias del *postmaster*. Llegando rápidamente al límite de procesos (32 por defecto) produciéndose un error en la página generada. Podemos subir el límite de instancias de *PostgreSQL* a 1024. Pero entonces nos falla el límite de los *files handles* abiertos (los *threads* comparten los *files handles* dentro un proceso, pero los procesos no). El límite de *handles* por usuario (el *daemon* del *PostgreSQL*, corre bajo el usuario '*postgres*') esta limitado en el *kernel*, con lo que tienes que recompilar el *kernel*. Después te encuentras que falla el límite de semáforos del *PostgreSQL*, teniendo que recompilar el *PostgreSQL*. Además tienes que controlar los ficheros de configuración del *apache* para bajar el número de clientes atendidos por instancia y así controlar el número de instancias de *PostgreSQL* que se puedan generar debido a la gran cantidad de recursos que ocupan.

Debido a la licencia liberal, el *PostgreSQL* puede ser usado, modificado, y distribuido gratuitamente para cualquier objetivo, sea privado, comercial, o académico.

18.3 CARACTERÍSTICAS DE POSTGRESQL

- ④ Implementación del estándar SQL92/SQL99.
- ④ Soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc. También permite la creación de tipos propios.

- ⌚ Incorpora una estructura de datos array.
- ⌚ Incorpora funciones de diversa índole: manejo de fechas, geométricas, orientadas a operaciones con redes, etc.
- ⌚ Permite la declaración de funciones propias, así como la definición de disparadores.
- ⌚ Soporta el uso de índices, reglas y vistas.
- ⌚ Incluye herencia entre tablas (aunque no entre objetos, ya que no existen), por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales.
- ⌚ Permite la gestión de diferentes usuarios, como también los permisos asignados a cada uno de ellos.
- ⌚ Posee una gran escalabilidad. Gracias a su arquitectura de diseño es capaz de ajustarse al número de CPUs y a la cantidad de memoria que posee el sistema de forma óptima, haciéndole capaz de soportar una mayor cantidad de peticiones simultáneas de manera correcta (en algunos benchmarks se dice que ha llegado a soportar el triple de carga de lo que soporta MySQL).
- ⌚ Implementa el uso de rollback's, subconsultas y transacciones, haciendo su funcionamiento mucho más eficaz, y ofreciendo soluciones en campos en las que MySQL no podría.
- ⌚ Tiene la capacidad de comprobar la integridad referencial, así como también la de almacenar procedimientos en la propia base de datos, equiparándolo con los gestores de bases de datos de alto nivel, como puede ser Oracle.
- ⌚ Desde la versión 7.0 implementa llaves foráneas con integridad referencial.
- ⌚ Tiene mejor soporte para triggers y procedimientos en el servidor.
- ⌚ Soporta un subconjunto de SQL92 mayor que el que soporta MySQL. Además, tiene ciertas características orientadas a objetos.

18.4 ¿QUÉ ES LO QUE LE FALTA?

PostgreSQL es un magnífico gestor de bases de datos, capaz de competir con muchos gestores comerciales, pero carece de:

- ⌚ Un conjunto de herramientas que permitan una fácil gestión de los usuarios y de las bases de datos que contenga el sistema.
- ⌚ La velocidad de respuesta que ofrece este gestor con bases de datos relativamente pequeñas puede parecer un poco deficiente, aunque esta misma velocidad la mantiene al gestionar bases de datos realmente grandes, cosa que resulta loable.
- ⌚ Consume gran cantidad de recursos y carga más el sistema.
- ⌚ Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.
- ⌚ Es de 2 a 3 veces más lento que MySQL.
- ⌚ Menos funciones en PHP.

19. INTERBASE 6.0

Borland siguiendo la política de liberalizar las herramientas de desarrollo, liberalizó *Interbase*. *Interbase* es una buena base de datos con 16 años de experiencia en el sector de las bases de datos comerciales.

Interbase ha sido la primera base de datos *Open Source* en ser compatible *SQL92* a nivel de entrada (*MySQL*, *PostgreSQL*, *mSQL*, no lo son). Eso significa que puedes adaptar cualquier aplicación que funcione sobre otra base de datos compatible *SQL92* como *Oracle*, *DB2* o *Informix* y traspasarla fácilmente a *Interbase*, o al revés: empezar con *Interbase* al principio y pasarla si algún día hace falta a alguna *RDBMS* mayor. Eso tan simple e importante, es imposible hacerlo con *MySQL* o *PostgreSQL*.

Casi todos los gestores de Backup soportan *Interbase* (como *ArcServe*). Tiene también gestores remotos para Windows de muy alta calidad, como *Marathon*.

Todas estas herramientas hacen de *Interbase 6.0* un producto muy profesional preparado para cualquier proyecto medio que necesite de una base de datos realmente fiable. Estas herramientas se echan de menos en *MySQL* y *PostgreSQL* a la hora de intentar integrar estas bases de datos en una empresa.

De todas formas, *Borland* no ha liberalizado las herramientas más avanzadas como el *IBReplicator* para la *clusterización* de *Interbase*.

Interbase tiene la mayoría de funcionalidades de una base de datos comercial: *triggers*, tratamiento especial de *blobs*, *backup On-line*, gran escalabilidad, bases

de datos de solo lectura (para ponerlas en *CD-ROM*), *integridad referencial en cascada* o el *autotunning*.

Interbase tiene todo lo que una base de datos moderna tiene que tener. Además no consume demasiados recursos y es igual de rápida que un *MySQL* y con muchas de las funcionalidades del *PostgreSQL*.

La instalación y administración es parecida e igual de simple que la del *MySQL*, más que el *PostgreSQL*. Dispone, además, de soporte directo para *PHP* como *MySQL* y *PostgreSQL*. Un detalle a tener en cuenta, es que el *daemon* del *Interbase* se esconde detrás el *Inetd*, a diferencia de las otras comparadas.

La desventaja que tiene el *Interbase* frente a las dos anteriores es la poca comunidad de la que dispone de momento. En otras palabras, que con *Interbase* te tienes que espabilar más. Hay poca información aun en la red. La *Web* de *Interbase* aparece muy comercial y un poco liada. Si *Borland* quiere que *Interbase* cuaje en el mundo *Linux* tendrá que abrir la *Web* a las comunidades de desarrolladores.

19.1 ADMINISTRACIÓN

InterBase nos ofrece herramientas tanto bajo *Windows* como en modo texto (basadas en línea de comandos) para la correcta administración de las bases de datos y de los servidores.

Se pueden administrar las bases de datos, tanto si se está trabajando en modo local (*Local InterBase*) como con *InterBase Server* por medio del *Server Manager*, desde una aplicación *Windows* que puede correr en el servidor o en un *PC* cliente.

También se puede utilizar la utilidad en modo texto ubicado en el servidor. Aunque el servidor sea una máquina con sistema operativo UNIX, se pueden realizar todas las operaciones de mantenimiento de las bases de datos desde un cliente cuyo sistema operativo sea Windows, por medio de las herramientas de administración tipo Windows que acompañan a InterBase.

Ambas aplicaciones, tanto la que trabaja bajo Windows como la otra, habilitan al administrador para lo siguiente:

- Control de la seguridad en el servidor.
- Hacer copias de seguridad y restaurar las bases de datos.
- Hacer el mantenimiento general de la base de datos.
- Control del estado de la base de datos y las estadísticas del "Lock Manager".

19.2 INSTALACIÓN

19.2.1 Instalación en Windows NT y Windows 95/98:

Antes de comenzar, es conveniente desinstalar cualquier versión anterior de InterBase que esté en el servidor antes de instalar InterBase 5.x. Si el servidor de InterBase es el InterBase Server para Windows NT 4.0 o anterior, entonces es obligatorio desinstalarlo, porque sólo debe existir una copia del fichero gds32.dll en la máquina mientras el servidor o el cliente de InterBase 5.x esté ejecutándose. Si ese fichero existe en el servidor, se deberá de borrar antes de instalar.

No se puede instalar InterBase en una unidad de red mapeada, es decir que solo se puede instalar en unidades de disco que pertenezcan al servidor.

Testear la conexión ODBC mientras se está realizando la instalación, puede dar como resultado que la instalación no se realice de forma correcta.

Es necesario entrar como Administrador en Windows NT para poder instalar o desinstalar InterBase en Windows NT.

Instalación en Windows: El sistema operativo debe ser Windows NT 4.0, Windows 95 ó Windows 98 con unos 20 Mb (19.458Kb) de disco duro libre.

Las opciones son:

- Ⓢ **InterBase 5.1.1: Client and Server:** Es la instalación completa de InterBase. A partir de aquí se nos mostrarán las opciones de instalar el cliente, el servidor, los drivers ODBC de InterBase, las bases de datos de ejemplo, la documentación de ayuda en línea, etc.
- Ⓢ **InterBase 5.1.1: Client Only:** Es una opción similar a la anterior, exceptuando la instalación del servidor. Es la destinada a las máquinas cliente.
- Ⓢ **Interclient 1.1:** Instala todo lo necesario para el desarrollo de sistemas InterBase con JAVA, como el driver JDBC para InterBase, las clases de desarrollo de aplicaciones Java, y el Kit de desarrollo de web-server.
- Ⓢ **Adobe Acrobat Reader 3.0:** Es el visualizador de los documentos de ayuda en línea de InterBase (ya no vienen en formato HLP).

Si escogemos la primera opción, nos aparecen una serie de pantallas en las que se nos explica con detalle las condiciones bajo las que hemos adquirido el software, sobre todo en lo que respecta a las licencias de conexión para clientes, etc., y algunos de los requerimientos del sistema para la instalación de InterBase. La primera pantalla en la que se nos pide una entrada de datos es la **License Certificate**, estas licencias vienen en la caja de InterBase, y son las que nos permiten instalar el servidor.

Las licencias para clientes, deben introducirse posteriormente con la herramienta de Registro de licencias (InterBase License Registration Tool). La licencia de evaluación que ofrecemos con el curso, se activa introduciendo la palabra *eval* en los dos campos.

Una vez introducidos estos datos, nos aparece la pantalla en la que podemos seleccionar los componentes que vamos a instalar. Donde nos permite escoger los componentes a instalar, la unidad y la ruta de instalación, con la restricción de que debe ser una unidad local, no puede ser una unidad de red.

Tras esta pantalla nos aparece otra en la que podemos configurar InterBase para que sea iniciado automáticamente por el sistema al arrancar o solo a petición del usuario.

Si escogemos la opción Windows Startup, el servidor se pondrá en marcha al iniciar la sesión de trabajo, y si escogemos la opción Manually Startup, entonces podemos iniciar el Servidor por medio del panel de control en la opción Servicios (en Windows NT), haciendo doble click sobre el icono de la aplicación, o sobre el InterBase guardián.

Luego se nos ofrece la opción, InterBase Guardián en el caso de que el servidor se caiga por alguna razón lo reinicie o no. A partir de aquí comienza la instalación en sí, con la copia de ficheros al disco duro, la configuración del ODBC, etc. Una vez que ha terminado la instalación, podemos escoger entre lanzar o no el servicio InterBase Guardián. Tras el proceso de instalación se crearán los iconos de InterBase y se actualizará el menú inicio, y después se abrirá el fichero README.txt, donde encontraremos más información acerca del producto.

La instalación en Windows NT es similar a la de Windows 95 o Windows 98 no resulta complicada, todas las opciones y cuadros de diálogo son bastante claros y

las decisiones en el proceso de instalación se toman de una en una, lo que favorece la instalación por usuarios poco experimentados.

Es interesante recalcar la figura del InterBase Guardián, que se encarga de controlar e iniciar al Server en caso de que por alguna razón se caiga o deje de funcionar, ya que en versiones anteriores de InterBase no existía.

La **instalación del cliente en Windows** es idéntica al proceso de instalación del servidor. Si acaso las únicas diferencias lógicas que nos encontraremos son las referentes a la ausencia de referencias al servidor en ésta instalación. Por lo demás es igual a la instalación del servidor.

Para realizar la **instalación de interclient en Windows**, es necesario contar con Windows NT 4.0 o Windows 95 y unos 2 Mb de espacio libre en disco duro para la instalación completa.

Una vez sobrepasadas las pantallas de bienvenida, avisos legales, etc., nos encontraremos con la pantalla que nos ofrece las diversas modalidades de instalación para la versión Interclient de InterBase.

Las cuatro posibilidades que se nos ofrecen son las siguientes:

- ④ **Development Environment:** Se instalará todo el entorno de desarrollo, es decir tanto las clases Interclient para el desarrollo local de JDBC como el Interserver para acceso local o remoto a un servidor de InterBase.
- ④ **Development and Web Deploy Environment:** Se instalará todo lo anterior, y además las clases Interclient se copiarán en el directorio raíz del servidor Web.
- ④ **Deploy to Web Server Only:** Instala el Interserver para acceder de forma local o remota a un servidor de InterBase, y copia las clases Interclient al servidor Web.

- ☛ **Client Development Only:** Instala las clases Interclient para el desarrollo de aplicaciones Java en modo local contra un servidor remoto que tenga una instalación de Interclient realizada.

Una vez realizada la elección, pasamos a una ventana en la que se nos dará a elegir cuáles de los componentes que conforman la totalidad del paquete queremos instalar.

Caso de que la opción escogida incluya el apartado de desarrollo de web, en uno de los cuadros de diálogo se nos preguntará por el directorio en el que se encuentra nuestro servidor web.

El resto del proceso es similar a la instalación del InterBase Server, excepto en la pantalla de configuración del Interserver, que en este caso nos presenta la forma en la que se iniciará el servicio de Interserver.

Este es prácticamente el último cuadro de diálogo al que accederemos, siempre que elijamos configurar el Interserver durante la instalación.

A partir de aquí se nos preguntará si queremos ver el fichero README.TXT, y la pantalla de despedida donde se nos indica que el proceso de instalación ha concluido.

Para desinstalar InterBase adecuadamente, es necesario utilizar la opción del panel de control de Windows para agregar/quitar programas.

Respecto a la desinstalación de InterBase del sistema, cabe realizar las siguientes observaciones:

- ☛ Para poder realizar la desinstalación de InterBase del sistema, no pueden estar corriendo en ese momento ni el InterBase Guardián, ni el InterBase Server, ni

el InterServer. Primero debemos parar estas aplicaciones, ya sea desde el administrador de servicios (en Windows NT) o pulsando con el botón derecho del ratón sobre el icono del programa en la barra de tareas del sistema, para posteriormente poder iniciar el proceso de desinstalación.

- ⓐ Tanto par instalar como para desinstalar InterBase en Windows NT, es necesario entrar en el sistema como administrador.
- ⓐ El proceso de desinstalación nunca borra el fichero `isc4.gdb`, ni los ficheros creados por InterBase Server, como `InterBase.log`, o los `*.evn`, `*.lck`.
- ⓐ El proceso de desinstalación sólo elimina los componentes instalados en la última instalación.
- ⓐ El driver ODBC tampoco es eliminado por el proceso de desinstalación, por lo que debe ser eliminado por el usuario a través de la opción del panel de control.

19.2.2 Instalación en UNIX:

Antes de iniciar la instalación de InterBase, hay que tener en cuenta los siguientes puntos:

- ⓐ InterBase 5.x utiliza ODS 9.0, por lo que las bases de datos creadas con versiones anteriores de InterBase, deben ser adaptadas haciendo backup con la versión anterior, y restore con la versión 5.x.
- ⓐ Si está migrando de InterBase 4 a InterBase 5 y no quiere perder toda la información referente a opciones de configuración personalizada del producto, puede salvar el fichero `isc4.gdb` con `gbak` y los ficheros `isc_license.dat` e `isc_config` en el directorio `/var/tmp` con el comando `cp`. Ahora bien, si utiliza `pkgrm` para desinstalar InterBase 4.0 para Solaris, estos ficheros se salvan automáticamente en `/usr/tmp`.

- ☉ También se pueden conservar las versiones anteriores de InterBase tirando abajo el servidor y posteriormente renombrando el directorio en el que se encuentre.
- ☉ Para instalar InterBase en HP-UX es necesaria la versión HP-UX 10.20, y debe estar instalado el HP DCE/9000.
- ☉ Para Solaris es necesario tener las versiones 2.5x o 2.6x.

19.2.2.1 Instalación en algunas plataformas UNIX

Para instalar InterBase, primero es necesario montar el CD-ROM. Una vez realizado esto, depende de la plataforma UNIX en la que nos encontremos, el proceso a seguir varía un poco.

Por ejemplo, para instalar **InterBase para Solaris**, es necesario utilizar la utilidad `pkgadd`, con la siguiente instrucción:

```
# pkgadd -d /cdrom/InterBase_SOS_V500
```

El proceso de instalación es realmente sencillo, y una vez terminada ésta, podemos restaurar los ficheros de configuración volviéndolos a copiar al directorio en el que estaban. Después iniciáramos el Server con la instrucción:

```
# echo " /usr/InterBase/bin/ibmgr -start -forever" | su Internase
```

A partir de aquí ya podríamos con `gbak` recuperar la base de datos `isc4.gdb`.

El proceso para HP-UX sería diferente ya que para empezar los ficheros para esta plataforma vienen en formato `tar`, con lo cual lo primero sería colocarnos en el directorio `/usr` del CD-ROM y ejecutar:

```
# tar -xvf /cdrom/InterBase/InterBase50_HUS.tar
```

y posteriormente iniciar la instalación con el comando siguiente

./InterBase/install

La instalación en LINUX es muy similar a la que se realiza en HP-UX, ya que los ficheros vienen en formato tar. La instrucción para descomprimir es:

tar -xzf IB_4.0_Linux.tar.gz

y posteriormente se inicia la instalación con el comando:

sh ./install

19.2.2.2 Desinstalación en UNIX

En Solaris existe una utilidad, pkgrm que realiza la desinstalación. En el resto de plataformas como HP-UX o LINUX es necesario borrar el directorio de instalación de InterBase y todas las referencias al mismo.

También es necesario editar el fichero /etc/services para eliminar la entrada de gds_db.

20. LAS COMPARATIVAS

Primero compararemos 2 bases de datos, una propietaria (Oracle 8) y otra open source (MySQL). Para lograr establecer las diferencias y similitudes en sus características principales.

Tabla 13. Características Principales

CARACTERISTICAS		MYSQL 3.23.37	ORACLE 8
TIPO DE DATOS	Caracteres, numéricos y de tipo dato/fecha	Compatible con los tipos SQL'92, incluyendo todos los tipos numéricos SQL'92.	Compatible con los tipos SQL'92. Algunos tipos SQL'92 son mapeados dentro de los tipos de Oracle. No permite tipo Boolean ni de equivalencia.
CARACTERÍSTICAS DEL LENGUAJE SQL	Acorde con los standares de SQL.	Compatible enteramente con SQL'92. MySQL tiene algunas diferencias si lo comparamos con ANSI SQL92	Buena compatibilidad con SQL'89, enteramente compatible con el nivel declarado SQL'92.
	Posibilidad de subqueries (queries anidados)	No son posibles.	Son permitidos los subqueries relacionados y no relacionados. Los queries pueden ser anidados hasta más de 255 niveles.
	Subqueries en FROM clause	No	Si

DECLARACIÓN DE CONSTRAINTS	Llaves Primarias	Si	Si
	Llaves Unicas	Si	Si
	Llaves Foráneas	Llaves foráneas no permitidas y no tienen efecto en las bases de datos.	Si. Y soporta el borrado en cascada
	Check	La cláusula <i>Check</i> no esta permitida, no tiene efecto en la base de datos.	Si
PROGRAMACIÓN ABSTRACTA	Views: presentación de datos contenidos en una o más tablas (u otras views). Un view toma la salida de un query y lo trata como una tabla por lo tanto puede se usado como una tabla virtual.	No	Si
	Views actualizables: es una vista la cual puede ser usada en DML para la modificación de datos.	No	Si
	Sinónimos: es un alias para alguna tabla, views u otro objeto en una base de datos.	No	Si
ACCESO MULTIUSUARIO	Máximo numero de usuarios concurrentes	No Definido	Por encima de los mil. Habilidad para servir a gran escala en conexiones en paralelo a la base de datos.
	Máximo numero de usuarios concurrentes escribiendo en la base de datos.	No Definido	No tiene limite
EXPORTAR DATOS	Herramientas para exportar datos (en formatos universales)	Todos los datos pueden ser exportados (dumped) a otra base de datos.	Depende de los resultados de los SQL query

		Existe una utilidad especial - <code>mysqldump</code> la cual exporta solo tablas seleccionadas o la base de datos entera.	
	Herramientas para importar datos	Importa archivos texto, html, dbf.	Alta velocidad y cargador de datos flexible.
PORTABILIDAD DEL DBMS Y ESCALABILIDAD	Hardware y sistemas de plataformas soportado	Plataformas Unix, Windows y OS/2.	Más de 200 plataformas, incluyendo Unix, Windows, mid-range y mainframe platforms.
	Soporte para sistemas SMP (parallel query execution, etc.) Procesamiento en paralelo en sistemas de memoria - también conocida como multiprocesamiento simétrico (SMP), en el cual múltiples procesadores use un solo recurso de memoria.	MySQL es multi-threaded server por lo que puede usar muchos procesadores. Cada conexión crea un camino separado.	Oracle puede usar sistemas multiprocesadores SMP.
LIMITE DE TAMAÑO DE DATOS	Máximo numero de filas y columnas en una tabla	El máximo numero de columnas en una tabla es 3398	El máximo número de columnas en una tabla es 1000. Por encima de 32 columnas en index key.
	Máximo numero de tablas e índice por tablas	Numero de tablas no limitada, una tabla puede tener por encima de 16 indexes (El tamaño de la llave es de 256 bytes max.).	Ilimitado.

En esta parte, primero se hizo una comparación de las características de las bases de datos y luego una comparación de los rendimientos de las mismas.

20.1 COMPARATIVA DE CARACTERÍSTICAS TÉCNICAS

Ninguna de las bases de datos que vamos a analizar es perfecta. Cada una tiene sus ventajas y desventajas. En la siguiente tabla podemos ver un resumen de las características técnicas que cumple cada *RDBMS*. Ponemos sólo las más relevantes.

Tabla 14. Características Técnicas

	MySQL 3.23.37	PostgreSQL 7.1	Interbase 6.0
Cumple SQL 92	N	N	S
Integridad Referencial	S	S	S
Procedimientos Almacenados	N	S	S
Vistas (views)	N	S	S
Vistas actualizables	N	N	N
Bases de datos de sólo lectura	N	N	S
Secuencias (generadores de números)	S	S	S
Subselects	N	S	S
Roles (perfiles de usuarios)	N	N	S
Rules (reglas)	N	S	N
Locks a nivel de fila	N (nivel tabla)	S	S
Concurrencia Multi Versión	N	S	N
Tratamiento de Blobs	N	S	S
Control acceso de usuarios	Muy bien	Medio	Bien
Transacciones	N	S	S
Triggers	N	S	S
Multithread	S	N	S
Limite tamaño registro	N	S	S

Orientación a objetos	N	S	N
Drivers JDBC	S	S	S
Drivers ODBC	S	S	S
Soporte nativo PHP	S	S	S
Creación de nuevos tipos de datos	N	S	N
Backup OnLine	S	S	S
Indexación de texto	Bien	Muy Bien	Mal
Soporte Perl, C, Python, TCL, Java, Delphi	S	S	S
Herencia	N	S	N

Hay otras características que sólo se aprenden mediante la experiencia. Son los temas referentes a la facilidad de *administración*, *backups*, *instalación*, etc. En cualquier caso, son difíciles de medir y varían ampliamente las opiniones entre los autores.

Tabla 15. Otras características

	MySQL 3.23.37	PostgreSQL 7.1	Interbase 6.0
Facilidad de instalación	Muy bien	Bien	Bien
Facilidad administración	Muy bien	Bien	Bien
Documentación Web oficial	Muy bien	Medio	Medio
Documentación no oficial (news, webs, tutoriales)	Muy bien	Muy bien	Medio
Configuración PHP	Muy bien	Bien	Bien
Potencia	Medio	Muy bien	Bien
Velocidad	Muy bien	Medio	Muy bien
Utilidad Web oficial	Bien	Bien	Mal
Grandes Bases de datos	Medio	Medio	Bien
Escalable	Bien	Medio	Muy bien
Utilidades Backups	Medio	Medio	Bien
Clusterizable	N	N	Si (Op)
Administración remota desde	Medio	Medio	Muy Bien

Windows			
Estable (daemon no se muere)	Bien	Medio	Muy Bien
Multiplataforma	Muy Bien	Muy Bien	Bien
Búsqueda de texto (Full-text-indexing)	Muy Bien	Bien	Medio
Consumo recursos (CPU y memoria)	Muy Bien	Medio	Bien
Adecuado para almacenar binarios	Mal	Muy bien	Bien

20.2 COMPARATIVA RENDIMIENTOS

El ejemplo y las tablas siguientes fueron establecida por Marc Vaquer Crusat, en su web personal. Pero el análisis fue realizado por los autores del presente proyecto.

Para comparar los rendimientos el autor preparó una serie de tests donde poder controlar los tiempos que tardaba cada base de datos en pasarlos. La máquina de test es un *Pentium III 800Mhz* con *256Mb RAM* y un disco *IDE* de *20Gb*.

Para comprobar los rendimientos atacó las bases de datos desde dos puntos: desde una entrada de un fichero *batch* desde línea de comandos, y desde una apache con soporte *PHP* para generación de páginas dinámicas.

Las tareas preparatorias fueron las siguientes: Crear una base de datos llamada *Empresas* en todas las *RDBMS*.

Crear en todas las bases de datos las siguientes tablas:

Tabla 16. Script creación de tablas

```
Create table departamentos (  
    Id integer primary key,  
    Nombre varchar(10)  
);  
create table empleados (  
    Id integer primary key,  
    Nombre varchar(30),  
    Nif varchar(9),  
    Departamento integer,  
    Foreign key FK_EMP_DEP (ID) references departamentos(Id)  
);
```

Instala el *PHP 4.0.5* como módulo en el *Apache* con soporte para las cuatro bases de datos.

20.2.1 Ataque Desde Línea De Comandos

Atacar las RDBMS redireccionando 3 ficheros de entrada a los clientes de línea de comandos. Los ficheros son:

- ④ Fichero con 100 *inserts* de departamentos y 1000 empleados por cada departamento. Total: 100.100 *Inserts*.
- ④ Simulamos un entorno normal de producción: 70% de *Selects*, 10% *inserts*, 10% *updates*, 10% *deletes*. Generamos un fichero con 100.000 instrucciones

repartidas en esos porcentajes. Los *selects* son variados con *Max()*, *Count()*, *joins*, pero con resultados cortos.

- © Simulamos un entorno *internet*: 99% *selects*, 0.3% *insert,s* 0.3% *updates*, 0.3% *deletes*. Los *selects* son complejos y con resultados largos. Total de 1.000 instrucciones.

Tabla 17. Resultados entrada fichero de texto

	MySQL 3.23.37	PostgreSQL 7.1	Interbase 6.0
Inserción 100 mil tuplas	10 seg.	12min 38seg	42seg
100.000 variados	17seg	6min 30seg	1min 07seg
1000 selects complejos	55seg	1min 26seg	2min 23seg

Según los resultados de las tablas vemos que **MySQL** gana siempre en velocidad. En el ejemplo no hay que comprobar que por cada empleado, existe el departamento.

PostgreSQL pierde mucho en las inserciones debido a que guarda físicamente cada escritura que hace. En las pruebas con **PostgreSQL**, el disco no para de escribir, mientras que con las otras, va escribiendo a ratos. **PostgreSQL** también tiene un modo '*flush*' (usar *buffer*) pero no recomiendan usarlo. En cuanto a los *selects* complejos los trata muy bien.

Interbase da tiempos moderados en todos los campos sin destacar en ninguno.

20.2.2 Ataque Desde Php

Con este test queremos ver la capacidad para servir páginas dinámicas largas de una combinación *Apache+PHP+base de datos*. Las versiones usadas son: *PHP 4.0.5* y *Apache 1.3.12* y el *PHP* está compilado como *módulo dinámico*.

Página de test para Interbase

```
<html>
<head>
</head>
<body>
<table width="100%" bgcolor="#336699">
<tr>
<td><center><font color=#ffffff>
Prueba Interbase 6.0
</center>
</font>
</td>
</tr>
</table>
<?
    $r=rand(10,79);
    $conn=ibase_pconnect("/opt/interbase/examples/spl.gdb","SYSDBA","maste
rkey","");
    $sql = "Select * from departamentos where ID=".$r;
    $st = ibase_prepare($conn,$sql);
    $st2= ibase_execute($st);
    $row = ibase_fetch_object ($st2);
```

```

?>
<table width="100%">
<tr bgcolor="#99ff99">
    <td colspan=2><center>DEPARTAMENTO</center></td>
</tr>
<tr bgcolor="#ddffdd">
    <td><center>ID</center></td>
    <td><center>NOMBRE</center></td>
</tr>
<tr bgcolor="#eeffee">
    <td><center><?=$row->ID;?></center></td>
    <td><center><?=$row->NOMBRE;?></center></td>
</tr>
</table>
<?

```

```

$sql = "Select * from empleados where ";
$sql = $sql."departamento=10 or ";
$sql = $sql."departamento=".$r;
$sql = $sql." order by nombre desc";
$st = ibase_prepare($conn,$sql);
$st2= ibase_execute($st);
$row = ibase_fetch_object ($st2);

```

```

?>
<table width="100%">
<tr bgcolor="#99ff99">
    <td colspan=3><center>EMPLEADOS</center></td>
</tr>
<tr bgcolor="#ddffdd">
    <td><center>ID</center></td>
    <td><center>NOMBRE</center></td>

```



```

        <td><center>NIF</center></td>
    </tr>
    <?    while ($row = ibase_fetch_object ($st2)) { ?>
    <tr bgcolor="#eeffee">
        <td><center><?=$row->ID;?></center></td>
        <td><center><?=$row->NOMBRE;?></center></td>
        <td><center><?=$row->NIF;?></center></td>
    </tr>
    <?    }
    ?>
</table>
</body>
</html>

```

Lo primero fue buscar un programa para hacer *benchmarking* de servidores *web*. Después de probar el *Autobench*, el *WebBench* (para *Windows*), el *http_perf* y el *Apache Benchmark* (*ab*), al final nos decidimos por el *ab*, que es una contribución al proyecto *Apache*. Un ejemplo de uso del *ab* sería:

```
>ab -n 100 -c 10 -v2 http://saba/spl/provap_mysql.php
```

Generamos dos páginas para cada *RDBMS* a analizar. Una utilizando la apertura de conexión (*MySQL_connect*, *pg_connect*, *ibase_connect*, y *odbc_connect*) normal y otra con la apertura de *conexión persistente* (*MySQLp_connect*, *ibasep_connect*, y *odbcp_connect*). El propósito es ver si realmente incrementaba el rendimiento el hecho de usar *conexiones persistentes* o no.

Las páginas de pruebas hacen lo siguiente:

- Generar un número aleatorio entre 1 y 100.

- ⌚ Obtener y mostrar el departamento que corresponde a ese número.
- ⌚ Obtener y mostrar todos los empleados (1000) que correspondan a ese departamento, y los del departamento 10 (otros 1000).

Cada página resultante ocupa aproximadamente 240Kb y aprovecha los 100.000 empleados y 100 departamentos insertados previamente.

Las pruebas las realizamos siempre con 1000 *peticiones* y sólo variamos el número de usuarios concurrentes para poder ver la escalabilidad del *RDBMS* analizado.

Los *tests* empiezan con 10 clientes concurrentes hasta los 50, incrementando de 10 en 10. Los resultados los damos en tres variables: tiempo en resolver todas las *peticiones*, *páginas por segundo* y *velocidad de transferencia* en *Kb/seg*. En las tablas, un 20 significa un *test* con 20 *peticiones* concurrentes de páginas normales, un 20p significa un *test* con 20 *peticiones* con *conexión persistente*.

Los tests se realizan con un *script bash* que ejecuta todos los *tests* uno a uno y los envía a un fichero de resultados. Esto afecta a los resultados: vemos que el *MySQL* a partir de 40 conexiones da *timeout* en las pruebas, pero si no ejecutamos los tests de 10, 20 y 30 conexiones anteriormente y ejecutamos el de 40 aisladamente dejando reposar la máquina, lo pasa sin problemas, incluso el de 50. Pero creemos que la gracia de un *test*, es ir incrementando sin parar las conexiones, para ver como escala, no probar puntas de trabajo aisladas. De hecho, desde un estado de reposo, todas aguantan hasta las 75 conexiones concurrentes.

Cuando las bases de datos se colapsan (por falta de memoria, *CPU*, disco, etc), *MySQL* e *Interbase* simplemente dejan de responder (o tardan demasiado en hacerlo) y el *Apache* da un *timeout* que recoge el *Apache Benchmark*. Pero el

PostgreSQL cuando ya no puede dar absorbe más conexiones de un error al generar la página de *Too many clients*. En este caso, la página es generada, pero con el típico error de conexión, y el *Apache Benchmark* no lo puede detectar. Tienes que mirar el tamaño de los ficheros devueltos para detectar cuando el *PostgreSQL* ha dejado de aceptar conexiones. Una página bien generada ocupa 240Kb y una con error unos 2Kb.

En las siguientes tablas se muestran los resultados:

Tabla 18. Segundos en completar las 1000 peticiones

	MySQL	PostgreSQL	Interbase
10p	312,66	752,74	225,30
10	325,22	943,40	212,42
20p	849,47	813,69	262,76
20	947,91	974,51	291,12
30p	1153,88	1620,00	377,75
30	1199,01	1468,79	408,23
40p	Timeout	Error	681,38
40	Timeout	Error	683,29
50p	Timeout	Error	Timeout
50	Timeout	Error	Timeout

Fuente: Marc Vaquer Crusat. Comparando MySQL, PostgreSQL, Interbase y SAP DB.

Tabla 19. Peticiones servidas por segundo

	MySQL	PostgreSQL	Interbase
10p	3,17	1,32	4,39
10	3,04	1,04	4,66

20p	1,17	1,21	3,77
20	1,04	1,01	3,40
30p	0,86	0,61	2,62
30	0,83	0,67	2,43
40p	Timeout	Error	1,45
40	Timeout	Error	1,45
50p	Timeout	Error	Timeout
50	Timeout	Error	Timeout

Fuente: Marc Vaquer Crusat. Comparando MySQL, PostgreSQL, Interbase y SAP DB.

Según las tablas anteriores, **MySQL** escala muy mal. Bajo pocas conexiones respondía perfectamente, cumpliendo expectativas, pero no pudo con el *Interbase* en el tramo final.

PostgreSQL 7.X realmente ha mejorado la velocidad respecto las versiones 6.X. Lo normal en las 6.X es que fueran 3 veces más lentas que el *MySQL*. Ahora bajo pocas conexiones apenas es la mitad, y la iguala en un entorno cargado.

Interbase es el mejor en todos los campos en este terreno. Le pertenecen los mejores tiempos en cada test. Es el único que llega a 50 peticiones concurrentes sin dar errores y además el que mejor escala.

Si **MySQL** se llevó el premio del test por fichero, el **Interbase** se lleva el de base de datos ideal para servidor *Internet*.

Este es el resultado de nuestro análisis del tests consultado. Hemos intentado que sean lo más representativos posibles. No significa que *Interbase* sea la mejor base de datos, sino que en este entorno ha sido la mejor.

21. SQL SERVER

Microsoft SQL Server 7.0 constituye un lanzamiento determinante para los productos de bases de datos de Microsoft, continuando con la base sólida establecida por SQL Server 6.5. Como la mejor base de datos para Windows NT, SQL Server es el RDBMS de elección para una amplia gama de clientes corporativos y Proveedores Independientes de Software (ISVs) que construyen aplicaciones de negocios.

Las necesidades y requerimientos de los clientes han llevado a la creación de innovaciones de producto significativas para facilitar la utilización, escalabilidad, confiabilidad y almacenamiento de datos.

21.1 OBJETIVOS DEL DISEÑO DE SQL SERVER

Los clientes están buscando soluciones para sus problemas de negocios. La mayoría de las "soluciones" de bases de datos solamente traen múltiples niveles de costos y complejidad. La estrategia de Microsoft es la de hacer que SQL Server sea la base de datos más fácil de utilizar para construir, administrar e implementar aplicaciones de negocios. Esto significa tener que poner a disposición un modelo de programación rápido y sencillo para desarrolladores, eliminando la administración de base de datos para operaciones estándar, y suministrando herramientas sofisticadas para operaciones más complejas.

SQL Server 7.0 disminuye el costo total de propiedad a través de características como administración multi-servidor y con una sola consola; ejecución y alerta de trabajos basadas en eventos; seguridad integrada; y scripting administrativo. Esta versión también libera al administrador de base de datos para aspectos más sofisticados del trabajo al automatizar las tareas de rutina. Al combinar estos poderosos servicios de administración con las nuevas características de configuración automática, Microsoft SQL Server 7.0 es la elección ideal de automatización de sucursales y aplicaciones de base de datos insertadas.

Los clientes invierten en sistemas de administración de bases de datos, en forma de aplicaciones escritas para esa base de datos y la educación que implica para la implementación y administración. Esa inversión debe protegerse: a medida que el negocio crece, la base de datos deberá crecer y manejar más datos, transacciones y usuarios. Los clientes también desean proteger las inversiones a medida que escalan aplicaciones de base de datos hacia equipos portátiles y sucursales.

Para cumplir con estas necesidades, Microsoft ofrece un motor de base datos único que escala desde una computadora portátil que ejecuta Windows® 95 o Windows 98, hasta clusters de procesadores múltiples simétricos de terabyte que ejecutan Windows NT Server Enterprise Edition. Todos estos sistemas mantienen la seguridad y confiabilidad que exigen los sistemas de negocios de misión crítica.

Nueva para el lanzamiento de 7.0 es una versión de rastro de baja memoria con capacidades de replicación de multi-sitio. Se ajusta muy bien a las necesidades cada vez mayores del mercado de la computación móvil. Las otras características tales como bloqueo a nivel de línea dinámico, el paralelismo intra-query, query distribuido, y mejoras para las bases de datos muy grandes (VLDB) hacen que el SQL Server 7.0 sea la elección ideal para sistemas OLTP de alta tecnología y sistemas de data warehousing.

Mientras los sistemas de procesamiento siguen siendo un componente clave para las infraestructuras de bases de datos corporativas, las compañías también están invirtiendo bastante en mejorar la comprensión que tienen de sus datos. La estrategia de Microsoft consiste en reducir el costo y la complejidad del data warehousing mientras hace que la tecnología sea más accesible a una mayor cantidad de público.

Microsoft ha establecido un enfoque total a todo el *proceso* de data warehousing (almacenamiento de datos). El objetivo es facilitar la construcción y diseño de soluciones de data warehousing costo efectivas a través de una combinación de tecnologías, servicios y alianzas con los proveedores.

La Microsoft Alliance for Data Warehousing es una coalición que une a los líderes en la industria de almacenamiento de datos y aplicaciones. El Microsoft Data Warehousing Framework constituye un conjunto de interfaces de programación diseñadas para simplificar la integración y administración de soluciones de data warehousing.

Las innovaciones del producto en SQL Server 7.0 mejoran el proceso de data warehousing: Servicios de Transformación de Datos; manejo mejorado de las consultas complejas y bases de datos muy grandes; procesamiento analítico en línea e integrado; y el Microsoft Repository. Otro componente esencial es el soporte extenso para integración de terceros.

Las innovaciones permiten que SQL Server 7.0 sea el líder en varias de las categorías de aplicación de rápido crecimiento en la industria de base de datos. Estas incluyen comercio electrónico, computación móvil, automatización de sucursales, aplicaciones de línea de negocios insertadas y mercados de datos.

Las áreas de liderazgo e innovación en el Microsoft SQL Server 7.0 incluyen:

- La primera base de datos en escalar desde la computadora portátil hasta la empresa utilizando la misma base de código y ofrecer el 100% de compatibilidad de código
- La primera base de datos en soportar la auto-configuración y auto-sintonización
- Primera base de datos con OLAP integrado
- La primera base de datos con Servicios de Transformación de Datos integrado
- El Data Warehousing Framework constituye el primer enfoque comprehensivo al problema de metadatos
- La primera base de datos en proveer administración de multi-servidor para cientos de servidores
- La más amplia gama de opciones de replicación de cualquier base de datos
- La mejor integración con Windows NT Server
- La mejor integración con Microsoft Transaction Server

Diseñada desde su inicio para trabajar en entornos Internet e Intranet, Microsoft SQL Server es capaz de integrar los nuevos desarrollos para estos entornos específicos con los desarrollos heredados de aplicaciones "tradicionales". Es más, cada aplicación que desarrollemos para ser empleada en entornos de red local puede ser utilizada de forma transparente -en parte o en su totalidad- desde entornos Internet, Intranet o Extranet.

- **Plataforma de desarrollo fácil y abierta:** integrada con las mejores tecnologías de Internet como ActiveX, ADC y Microsoft Transaction Server y con las mejores herramientas de gestión y desarrollo para Internet como FrontPage97, Microsoft Office97 y Visual Interdev.
- **Diseñada para INTERNET:** Es el único gestor de base de datos que contiene de forma integrada la posibilidad de generar contenido HTML de forma automática.

- ☉ **La Base de Soluciones Integradas:** La Integración total con BacOffice permite resolver toda las necesidades de infraestructura de la empresa con un sólo paquete.
- ☉ **Potente y Escalable:** Microsoft SQL Server es la única base de datos cuyo rendimiento sobre Internet está publicado, ofreciendo registros espectaculares.
- ☉ **Mínimo coste de Propiedad:** La sencillez de la instalación, y la potencia de sus herramientas de gestión y el menor coste de toda la industria para entornos Internet, hacen de Microsoft SQL Server la mejor opción con el menor coste.

21.2 ARQUITECTURA RDBMS

La Arquitectura RDBMS posee las siguientes características:

- ☉ Arquitectura de servidor simétrico y paralelo con balanceo automático de carga en múltiples procesadores.
- ☉ Kernel multithread real para mejor rendimiento transaccional y escalabilidad.
- ☉ Soporte grandes bases de datos (VLDB) (+1 TB).
- ☉ Completo proceso transaccional interactivo con rollback automático y recuperación de roll-forward.
- ☉ Optimizador de consultas mejorado basado en coste.
- ☉ Checkpointing mejorado para un mejor throughput de datos y tiempo de respuesta.
- ☉ Soporte E/S asíncrono para acceso en paralelo a múltiples dispositivos de disco para un mejor throughput.
- ☉ Bloqueo a nivel fija y página con escalación de bloqueos; resolución automática de deadlocks

21.3 DISPONIBILIDAD, FIABILIDAD Y TOLERANCIA A FALLOS

- Mirroring de dispositivos de base de datos con failover automático para tolerancia a fallos de dispositivos.
- Copias de seguridad online desatendidas garantizando la consistencia de datos para la más alta disponibilidad.
- Contextos de usuario protegidos, que pueden aislar los fallos a un thread de un único usuario.
- Recuperación point-in-time para restaurar bases de datos o transacción logs en un intervalo de tiempo.
- Tolerancia a fallos de servidor, permitiendo failover automático a un servidor de backup o en espera.

22. ESTADO DEL ARTE

La Tecnología de Información y Comunicaciones (TIC) es un elemento esencial en cualquier estrategia de desarrollo, por esta razón el gobierno colombiano ha incluido estos programas en los modelos de desarrollo los cuales mediante planes de acción compartidos entre el sector público y privado tienen el objetivo de masificar el uso y el aprovechamiento de las TICs, y apoyar al progreso de la comunidad en el aspecto político, económico y social.

En nuestro país, se está desarrollando la política de Estado: **“Agenda de Conectividad: el Salto a Internet”**, corresponde a uno de los objetivos en materia de telecomunicaciones del Plan Nacional de Desarrollo 1998-Actual “Cambio para Construir la Paz”.

Este programa estatal busca proveer el uso de las Tecnologías de la Información y así aumentar la competitividad del sector productivo, modernizar las instituciones públicas y de gobierno, y socializar el acceso a la información, siguiendo los lineamientos establecidos en el Plan Nacional de Desarrollo 1998 – Actual “Cambio para Construir la Paz”.

“El DANE por convenio celebrado con la Agenda de Conectividad se ha comprometido a diseñar, desarrollar, implementar y difundir un modelo de medición de las TICs, en Colombia, mediante el cual se produzcan los indicadores que permitan conocer y analizar el estado del arte de estas tecnologías en cuanto a: cobertura, acceso, infraestructura, contenido y aprovechamiento, en los

sectores: Estatal, productivo, educativo y comunitario.”¹⁴ “De acuerdo con la información de la población del país, el 42% de la población del país se encuentra concentrada en Antioquia, Atlántico, Valle del Cauca y Bogotá. Por lo tanto, se hace importante no solo hacer énfasis en estos sitios, sino también promover la masificación de la infraestructura de Tecnologías de Información y sus servicios”¹⁵

En consecuencia, a través de esta política el gobierno nacional busca incluir las Tecnologías de Información y Comunicaciones en el modelo de desarrollo del país, particularmente implementando acciones previstas en las estrategias de socialización de Internet.

Así, los esfuerzos del gobierno nacional relacionados con las tecnologías de información, buscan que nuestro país se ubique en posición de vanguardia en materia de infraestructura, acceso, contenido, desarrollo y aprovechamiento de las TICs.

En la actualidad la política gubernamental para fomentar el uso y el aprovechamiento de las TICs se define en los alcances del programa Agenda de Conectividad: el Salto a Internet.

Los objetivos que se han definido para esta política están enfocados hacia los siguientes sectores:

Comunidad: Fomentar el uso de las Tecnologías de la Información para mejorar la calidad de vida, ofreciendo un acceso equitativo a oportunidades de educación, trabajo, justicia, cultura, recreación, entre otros.

¹⁴ LEGUIZAMON, Manuel. Modelo general de la medición de las Tecnologías de Información y Comunicaciones (TICs). Bogotá: Diciembre del 2002.

¹⁵ Agenda de Conectividad. ¿Cómo se encuentra el país en Tecnologías de la Información?. Bogotá: Diciembre del 2002

Sector Productivo: Fomentar el uso de las Tecnologías de la Información y la Comunicación como soporte del crecimiento y aumento de la competitividad, el acceso a mercados para el sector productivo, y generación de empleo.

Estado: Proveer al Estado la conectividad que facilite la gestión de los organismos gubernamentales y apoye la función de servicio al ciudadano.

Para alcanzar eficientemente estos objetivos la política Agenda de Conectividad: el Salto a Internet, ha definido las siguientes estrategias:

* **Acceso a la Infraestructura de la Información:** Fortalecer la infraestructura nacional de telecomunicaciones y ofrecer acceso a las tecnologías de la información a la mayoría de los colombianos, a costos más asequibles.

* **Uso de TI en los procesos Educativos y Capacitación en el uso de TI:** Fomentar el uso de las tecnologías de la información como herramienta educativa, capacitar y sensibilizar a la población sobre la importancia en el uso de estas tecnologías, fortalecer el recurso humano especializado en el desarrollo y mantenimiento de tecnologías de la información.

* **Uso de TI en las Empresas:** Aumentar la competitividad de las empresas nacionales a través del uso de las tecnologías de la información.

* **Fomento a la Industria Nacional de TI:** Crear ambientes favorables para el desarrollo de la industria de tecnologías de la información.

* **Gobierno en Línea:** Mejorar el funcionamiento, la eficiencia, la transparencia del Estado y fortalecer la función del Estado de servicio al ciudadano a través del uso de tecnologías de la información.

23. ANTECEDENTES

Las Tecnologías de la Información y de las Comunicaciones (TIC), han experimentado un significativo avance en los últimos años en Colombia. Con diferentes intensidades y velocidades, los diversos sectores de la sociedad, de la economía y del sector público han comenzado a incorporar las nuevas tecnologías en sus actividades.

En Colombia organizaciones han realizado estudios de las tecnologías de información y comunicaciones y han llegado a diferentes conclusiones tales como:

La Presidencia de la República, con su programa Agenda de Conectividad: “El Centro para el Desarrollo Internacional de la Universidad de Harvard diseñó la metodología “Readiness for the Networked World”. Esta metodología permite medir el estado de avance de las Tecnologías de la Información y la Comunicación en los países en vía de desarrollo de acuerdo a cinco categorías de variables: a) acceso, b) educación, c) sociedad, d) economía, y e) políticas. Cada una de estas variables contiene varios componentes y, de acuerdo a la metodología, cada uno de esos componentes se los clasifica en cuatro etapas de desarrollo posible. Si un país se clasifica en la primera etapa, tiene un menor nivel de desarrollo en TICs. En el otro extremo, si se encuentra en la etapa cuatro dicho país está en el nivel avanzado de desarrollo en TICs. En las siguientes secciones el informe trata, en su orden, las categorías de la “Guía para los Países en Desarrollo: la Preparación

para el Mundo Interconectado” desarrollada por Center for International Development (CID) de la Universidad de Harvard.”¹⁶

Según una investigación de Acopi —Asociación Colombiana de Medianas y Pequeñas Industrias— de las 45 mil empresas que tienen entre uno y diez empleados, sólo un 40 por ciento tiene acceso a Internet, un 20 por ciento está enfocada hacia el e-business, y la gran mayoría sólo usa sus computadores para procesamiento de texto y software contable. La razón: En Colombia aún muchos pequeños y medianos empresarios piensan que sus compañías han sobrevivido y crecido sin hacer mayores inversiones en equipos y software, lo cual prueba que la tecnología es sólo un gasto y no es tan importante como se dice. Otros consideran que, aunque es posible que la tecnología informática les ayude a ser más productivos, acceder a ella es muy complicado por su precio y, sobre todo, por la complejidad de su implementación y uso.

“El 63 por ciento de las PYMES colombianas no están automatizadas, realizan muchos de sus procesos manualmente y tienen sus bases de datos disgregadas en distintos programas como Excel, Access, FoxPro, etc. Por eso el mismo trabajo se hace hasta cinco veces, lo que representa una gran carga operativa y altas probabilidades de error en el ingreso de la información. Lo peor es que la ineficiencia en el manejo de la información afecta la toma de decisiones. Es el caso del gerente que necesita ver un balance a fin de mes, pero sólo puede hacerlo a los dos o tres meses”¹⁷, afirma Luisa Méndez, Business Development Manager de Oracle.

¹⁶ Agenda de Conectividad. ¿Cómo se encuentra el país en Tecnologías de la Información?. Bogotá: Diciembre del 2002

¹⁷ MENDEZ, Luisa. ¿Pymes sin tecnología? ¡Imposible!. Revista Intercambio. Agosto – Septiembre del 2002.

Grupo CNET e integradores de tecnología informática: “Nuestra experiencia nos muestra que la principal necesidad es resolver los problemas transaccionales, y, después, abrir nuevas oportunidades de mercado, por medio de tecnologías de Internet como portales y marketplaces”¹⁸, asegura Mario Rozo. Para Luisa Méndez, la principal necesidad es la integración de la información. “Las pequeñas y medianas empresas necesitan soluciones tecnológicas fáciles de instalar, de usar y de actualizar, escalables —es decir, que puedan crecer según las necesidades—, de bajo costo —puesto que son muy sensibles al precio— y con elementos de seguridad incorporados”, afirma.

Una adecuada tecnología de la información puede resolver tanto las necesidades de comunicación entre los miembros de la empresa, como el manejo de procesos operacionales como la contabilidad, el manejo de nómina, y los pedidos e inventarios. “Se trata de ganar en productividad por medio de soluciones de conectividad, y por otra parte tener un mejor acceso a la información y un mayor control del negocio por medio de soluciones contables. En resumen, la tecnología debe estar enfocada a facilitar la solución de problemas reales, reducir costos y aumentar la productividad”, anota Guillermo Guzmán, gerente de la Unidad de PYMES de Microsoft.

Hoy más que nunca todas las empresas, cualquiera que sea su tamaño y sector de actividad, deben competir en un entorno global, disponer necesariamente de la tecnología de la información (TI) más moderna y saber aplicarla de forma eficaz y rentable. En este contexto, “escoger las herramientas tecnológicas adecuadas para gestionar la organización interna, optimizar los procesos y planificar la actividad, significa adquirir la capacidad para competir en nuevos escenarios.

¹⁸ ROZO, Mario. ¿Pymes sin tecnología? ¡Imposible!. Revista Intercambio. Agosto – Septiembre del 2002.

Las empresas pueden tener distintas dimensiones, pero todas deben desarrollar actividades similares (comprar, vender, mantener relaciones con los clientes, llevar la contabilidad, gestionar el personal y adaptarse a las modificaciones de las normativas legales y financieras). Por lo tanto, también las pequeñas y medianas empresas deben adoptar tecnologías que soporten plenamente los objetivos de la empresa y que permitan reaccionar de forma rápida y flexible ante los acontecimientos externos, interviniendo en la organización comercial, en la logística de producción y en la toma de decisiones financieras.”¹⁹

¹⁹ SAP.com

24. MARCO CONCEPTUAL

ACID: Conjunto de propiedades cuyo acrónimo significa Atomicidad, Consistencia, Aislamiento y Durabilidad.

API: Interfaz de Programación de Aplicaciones. Conjunto de rutinas, protocolos y herramientas para construir aplicaciones de interfaz.

Atomicidad: Característica de la transacción donde todos los pasos se cumplen. Conformación de ejecución y culminación de los procesos de una transacción.

Atributos: Datos que caracterizan a la entidad.

Bases de Datos: Conjunto de datos persistentes utilizado por las aplicaciones de una empresa. Sistema computarizado para manipular registros.

Bases de Datos Distribuidas: Colección de sitios, conectados por medio de algún tipo de red de comunicación, en el cual cada sitio es un sistema de base de datos completo por derecho propio.

Batch: Nombre genérico que se da a los archivos de proceso por lotes.

Benchmarks: Pruebas de medición del rendimiento de las bases de datos.

Browser: Examinador. Utilidad que nos permite consultar bases de datos o buscar en textos con facilidad.

CAD: Diseño Asistido por Computadora.

CASE: Ingeniería de Software Asistida por Computadora.

Catalogo: Información de control para permitir que el sistema proporcione la independencia de ubicación, fragmentación y replicación necesaria.

CGI: Common Gateway Interface – Interfaz de Entrada Común.

Cliente-Servidor: Sistema de organización de la información en la cual la aplicación central o servidor almacena los ficheros y los pone a disposición de las aplicaciones clientes.

CONACYT: Consejo Nacional de Ciencia y Tecnología.

CPU: Unidad Central de Procesamiento.

Datawarehouse: Colección de información corporativa derivada directamente de los sistemas operacionales y de datos externos.

DB2: Segunda versión del programa DBASE, que utiliza base de datos relacionales.

DBMS: Sistema Administrador de Bases de Datos. Conjunto de archivos interrelacionados de un conjunto de programas que permiten manipular la información almacenada en dichos archivos.

DW: Datawarehouse.

Ejemplar: Cada uno de los objetos.

Emplazamientos: Computadoras que forman parte de un sistema distribuido.

Entidad: Algo distinguible del mundo real.

Fragmentos: División de las bases de datos en bases más pequeñas.

Hipertexto: Sistema de presentación de la información en el cual el texto, las imágenes, los sonidos y las acciones están enlazados mediante una red compleja y no secuencial de asociaciones que posibilitan al usuario a examinar los distintos temas que están tratando, con independencia del orden de presentación de los mismos.

HTML: Lenguaje Descriptor de Hipertexto. Lenguaje estándar de hipertexto que se utiliza para crear paginas world wide web.

HTTP: HyperText Transfer Protocol. Protocolo de Internet que define cómo un servidor de Web debe responder a las solicitudes de los artículos que se le hacen vía *anchor's* y *URL's*.

IMS: Sistema Administrador de Información.

ISAPI: Interfaz de Programación de Aplicaciones del Servidor Internet.

ISS: Servidor Microsoft Internet Information.

JDBC: Conectividad de Bases de Datos de Java.

Multimedia: Combinación de sonidos, graficas, animación, videos y textos dentro de una misma aplicación.

MVS: Multiple Virtual Storage. Nombre que recibe el sistema operativo que utilizan los grandes ordenadores IBM.

Netscape: Programa navegador del Web compatible con Windows, Macintosh y X-Windows.

Nodos: Computadoras que forman parte de un sistema.

ODBC: Nombre de la biblioteca dinámica de Windows cuya misión es la facilitar el acceso de un tipo específico de sistema de gestión de bases de datos.

OLAP: Procesamiento Analítico en Línea.

OLTP: Procesamiento de Transacciones en Línea. Base de datos transaccionales, propias o incorporadas.

OML: Lenguaje de Manipulación de Objetos.

Papel: Categorías a las que pueden pertenecer los objetos.

Perl: Lenguaje de Extracción y Reportes Prácticos. Nombre comercial de un tipo de lenguaje de programación para los entornos UNIX que está siendo distribuido a través de Usenet.

Plug-ins: Aplicaciones que corren dentro del browser de la Web.

Registro: Conjunto de campos que poseen cierta información. También se le conoce como tupla o fila.

SQL: (Structure Query Language – Lenguaje de Consultas Estructurado). Lenguaje que permite realizar consultas en las bases de datos a través de mandatos.

TCP/IP: Protocolo de Control de Transmisión / Protocolo Internet. Conjunto de protocolos diseñados para conectar entre sí redes diferentes y en el cual se basa Internet.

TPC: Transaction Processing Performance Council. Consejo de negociación de ejecución de procesos.

TPC-C: Prueba de rendimiento que modela un sistema de control de inventarios y pérdidas.

TpmC ó TpmC-C: Cantidad de Transacciones por Minuto.

Transacción: Unidad lógica de trabajo que por lo regular comprende varias operaciones en la base de datos. Unidad de la ejecución del programa que accede y actualiza varios elementos de datos.

Transacción Global: Es la transacción que tiene acceso y actualiza datos en varias bases de datos locales.

Transacción Local: Es la transacción que tiene acceso y actualiza datos sólo en una base de datos local.

Tupla: Conjunto de campos que poseen cierta información.

UNIX: Sistema multitarea y multiusuario desarrollado originariamente por AT&T.

25. METODOLOGIA

25.1 TIPO DE ESTUDIO

El tipo de estudio es **Descriptivo** ya que se aplica un método exhaustivo de investigación para la obtención de nuevos conocimientos acerca de los Sistemas de Bases de Datos y su tecnología.

El objetivo de esta investigación es generar un documento que suministre información sobre la instalación, requerimientos y administración de bases de datos open source y propietarias que proporcione una ayuda en el conocimiento y la escogencia de las herramientas tecnológicas en cuanto a la base de datos mas apropiada para sus necesidades de información.

25.2 LINEA DE INVESTIGACION

La línea de investigación es Ingeniería de Software – Sistemas de Bases de Datos.

26. RECURSOS

RECURSOS	DESCRIPCIÓN	CANTIDAD	VALOR UNITARIO	VALOR TOTAL
HUMANOS	PERSONAS REALIZADORAS DEL PROYECTO: JORGE CASTRO, LENIS HERRERA, BERTHA REYES, OLGA RODRÍGUEZ, ARNULFO SALGUEDO. ASESORES: ING. JOHAN MANCERA MORENO, ING. PATTY PEDROZA, ING. WILSON NIETO.			
INSTITUCIONALES	BIBLIOTECA JOSÉ MARTÍ, USB			
	BIBLIOTECA DE COMBARRANQUILLA UNIDAD BOSTON			
	BIBLIOTECA DE COMFAMILIAR DEL ATLÁNTICO			
	INTERNET	100h	\$1.500	\$150.000
MATERIALES	COMPUTADORES			
	DISQUETE - CDS	15	\$1.500	\$22.500

	IMPRESORA (TINTA)	8	\$40.000	\$320.000
	RESMA	2	\$13.000	\$26.000
	LIBROS	1	\$54.000	\$54.000
	COPIAS	300	\$100	\$30.000
	UTILES			\$25.000
FINANCIEROS	PASAJES	200	\$900	\$180.000
			SUBTOTAL	\$807.500
			IMPREVISTO	\$100.000
			TOTAL	\$907.500

BIBLIOGRAFIA

DATE, C. J. Introducción a los Sistemas de Bases de Datos. 7 ed. México: Prentice Hall, 2001.

INSTITUTO COLOMBIANO DE NORMAS TÉCNICAS Y CERTIFICACIÓN. Tesis y otros trabajos de Grado. Bogotá: Icontec, 2002.

PRESSMAN, Roger. Ingeniería de software : un enfoque práctico. 2 ed. Bogotá: Mc Graw Hill, 1988.

SILBERSCHATZ, Abraham; KORTH, Henry y SUDARSHAN, S. Fundamentos de bases de datos. 3 ed. Madrid: Mc Graw Hill, 1998.

TAMAYO Y TAMAYO, Mario. Aprender a Investigar. Módulos 5 y 2. Bogotá: Icfes, 1999.

TESAI, Alice Y. Sistemas de Bases de Datos : administración y uso. Prentice Hall, 1990.

<http://www.acis.org.co/Paginas/publicaciones/modelo.doc>

http://www.arrakis.es/qenda/Articles/ArticleDB/Articulo_DB_9-5-01a.htm

<http://www.cem.itesm.mx/docs/publicaciones/logos/palabra/>

<http://www.clarin.com.ar/suplementos/informatica/2000-07-12/t-00201d.htm>

<http://www.cnet.com.co/noticias.htm>

<http://elrinconcito.com/delphi/Interbase5/Inicio/Inicio.htm>

http://www.encyclopedia-mex.galeon.com/Cad_Practicas.htm

<http://dev.mysql.com/Tech-resources/crash-me.php>

<http://www.intelibizsa.com/datawarehouse/>

<http://www.iberbanda.es>

<http://mysqlfront.sstienemann.de>

<http://www.phpencastellano.com>

<http://www.phpmyadmin.net>

<http://www.postgresql.org/docs>

<http://www.screenart.es/documentacion/postgres/tutorial/intro.htm#AEN34>

<http://www.sdgconsulting.net/spain/events/master/masterworkshop.html>

<http://www.uca.edu.sv/investigacion/bdweb/intbdweb.html>

http://www.unad.edu.co/documentos_ing_e/agenda%20de%20conectividad.pdf

ANEXOS

CRONOGRAMA

FASES	TIEMPO ACTIVIDADES	2003																							
		MESES																							
		ABRIL				MAYO				AGOSTO				SEPTIEMBRE				OCTUBRE				NOVIEMBRE			
		I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV	I	II	III	IV
I	ELECCIÓN DEL TEMA	■																							
	CONSULTAS SOBRE EL TEMA	■	■	■	■	■	■	■	■																
	PLANTEAMIENTO JUSTIFICACIÓN									■	■	■	■												
	OBJETIVOS										■	■	■	■	■	■									
	ALCANCES Y LIMITACIONES													■	■										
	MARCO TEÓRICO													■	■	■	■	■	■						
	MARCO CONCEPTUAL																	■	■						
	RECURSOS																		■	■					
	CRONOGRAMA																		■	■					
	BIBLIOGRAFIA																			■	■				
	II	PROFUNDIZACION DE PROYECTO INICIAL																				■	■	■	■

TITULO:

MODELO DE TECNOLOGIA DE INFORMACION Y COMUNICACIONES EN
SISTEMAS DE BASES DE DATOS APLICADO A LAS PEQUEÑAS Y
MEDIANAS EMPRESAS DEL DISTRITO BARRANQUILLA

AUTORES:

JORGE GUILLERMO FEDERICO CASTRO CERVANTES

LENIS MARIA HERRERA RODRIGUEZ

BERTHA ABIGAIL REYES SALAS

OLGA PATRICIA RODRIGUEZ RUZ

En el 2004, de acuerdo con los postulados teóricos de Roger Pressman expuestos en su libro Ingeniería Del Software Un Enfoque Práctico sobre ingeniería de software, Alice Tesai en su libro Sistemas de Bases de Datos : administración y uso sobre sistemas de bases de datos; Mario Tamayo y Tamayo en su obra Aprender a Investigar sobre metodología, diseño y desarrollo del proceso de investigación, se realizó una investigación sobre los sistemas de bases de datos y benchmarking de las siguientes bases de datos: Oracle, Access, MySql, Interbase y PostgreeSql. Para lo cual se utilizaron ayudas web y pruebas realizadas para comprobar por nosotros mismos la capacidad de cada una de los gestores de bases de datos mencionados anteriormente. Los autores lograron llevar a cabo una investigación exhaustiva acerca de los temas que están relacionados con el benchmarking de las bases de datos como son costos, integridad, instalación, administración, entre otros.

Los autores lograron un profundo entendimiento y claridad sobre las ventajas y desventajas de cada una de las bases de datos dependiendo el tipo de utilización que vaya a tener. Se analizaron y construyeron los cuadros comparativos teniendo en cuenta la objetividad para con cada una de las bases de datos estudiadas.



RESUMEN TRABAJO DE INVESTIGACIÓN

F-GI-01

Fecha vigencia:

Investigación ☒Tesis ☐

Fecha de inicio

Marzo de 2003

Programa:

Ingeniería de Sistemas

Fecha Finalización

Noviembre de 2004

Tutor:

Wilson Nieto

Estudiantes:	Castro Cervantes Jorge Guillermo Federico	Herrera Rodríguez Lenis Maria
	Reyes Salas Bertha Abigail	Rodríguez Ruz Olga Patricia
Título		
MODELO DE TECNOLOGIA DE INFORMACION Y COMUNICACIONES EN SISTEMAS DE BASES DE DATOS APLICADO A LAS PEQUEÑAS Y MEDIANAS EMPRESAS DEL DISTRITO DE BARRANQUILLA		
Palabras Claves		
Bases de datos, Benchmarks, Open source, GNU, Slashdot, Licencia GPL, RDBMS, ORDBMS, Diseños multihilo, Multiproceso, Triggers, Integridad transaccional, Multithreading, Herencia, ODBC, JDBC, SQL, Rollback, Datafiles, DBA Studio, Tablespaces, Controlfiles, Redo logs, Perfiles, Roles, Schema(Eschema), Storage, Instancia.		
Línea De Investigación		
Ingeniería de Software		
Eje Temático		
Base de datos		
Pregunta Problema		
¿Este estudio permitirá crear una base de conocimientos para formular patrones que apoyen la toma de decisiones con respecto a la optimización de los recursos tecnológicos y la información de una Pyme del distrito de Barranquilla?		
Justificación (Importancia e Impacto de la Investigación)		



RESUMEN TRABAJO DE INVESTIGACIÓN

F-GI-01

Fecha vigencia:

Las empresas con mayor aptitud de adaptarse a los cambios comerciales obtendrán grandes beneficios de las tecnologías procedentes de las comunicaciones y el cómputo distribuido, pues estarán capacitadas para ordenar sus sistemas de información y arquitecturas de datos de acuerdo a su área comercial y por lo tanto podrán operar eficaz y eficientemente en la obtención acertada de información importante.

Particularmente, las Bases de Datos son un escenario sólido que permite fortalecer el procesamiento eficiente e integro de la información que es la parte vital de cualquier institución, razón por la cual se debe brindar el conocimiento, a los futuros ingenieros, sobre la buena administración de ésta herramienta que brinda la tecnología informática.

El poseer un esquema de Bases de Datos traería beneficios sustanciales para una empresa que considere sus datos como un activo valioso. Entre estos beneficios encontramos:

- Los datos pueden compartirse
- Es posible reducir la redundancia
- Es posible evitar la inconsistencia
- Es posible brindar manejo de transacciones
- Es posible mantener la integridad
- Es posible hacer cumplir la seguridad

El conocimiento acerca de sistemas de bases de datos se ha convertido en una parte esencial en la formación informática. El propósito de esta línea de investigación es presentar temas especializados aplicados en los diseños de Bases de Datos orientados a los sistemas de información en general comparando bases de datos comerciales (Oracle y Access) y open source (MySQL, Interbase y PostgreSQL) proporcionar ayuda en la escogencia de las herramientas tecnológicas mas apropiadas según la necesidad del usuario.



RESUMEN TRABAJO DE INVESTIGACIÓN

F-GI-01

Fecha vigencia:

Objetivo General

Generar un documento que suministre información sobre la instalación, requerimientos y administración de bases de datos open source y propietarias que proporcione una ayuda en el conocimiento y la escogencia de las herramientas tecnológicas en cuanto a la base de datos mas apropiada para sus necesidades de información.

Objetivos Específicos

- Proporcionar una base sólida en los fundamentos y desarrollos de las tecnologías de Bases de Datos.
- Conocer las ventajas y desventajas que ofrecen los diferentes Sistemas de Bases de Datos (Distribuidas, Data warehousing, en la WEB, Centralizadas, CRM, PRM, SDK, MySQL, Oracle, Access, PostgreSql e Interbase).
- Enriquecer los conocimientos con diversos conceptos que proporcionan los Sistemas de Bases de Datos.
- Comprender de qué se tratan los Sistemas de Bases de Datos y sus beneficios para con las organizaciones que las manejan para la administración de la información.
- Estudiar las características de los sistemas de bases de datos, lo que proporciona al usuario y cómo se comunica con los sistemas en funcionamiento.
- Suministrar información acerca del comportamiento de los motores de Bases de Datos frente a los diversos Lenguajes de Programación y Sistemas Operacionales para lograr una mejor utilización de éstas en pro de la conservación y la operatividad de la información en la organización.
- Proporcionar información sobre la administración, instalación y configuración de Bases de Datos Open Source (MySQL, PostgreSql e Interbase) y Propietarias (Oracle, Access).
- Realizar encuestas en una serie de empresas para conocer acerca de los Sistemas de Bases de Datos que utilizan o implementan para la manipulación de la información.



RESUMEN TRABAJO DE INVESTIGACIÓN

F-GI-01

Fecha vigencia:

Marco Teórico (Enfoque Conceptual: Teorías - Autor Guía)

Uno de los autores utilizados fue Roger Preesman quien en su obra "Ingeniería de Software. Un enfoque práctico" nos muestra la forma de parametrizar la calidad del software contra factores como la corrección, la fiabilidad, la eficiencia, entre otras. También nos muestra los Factores de Calidad ISO 9126 del cual enuncia "...ha sido desarrollado en un intento de identificar los atributos claves de calidad para el software.", como son: funcionalidad, confiabilidad, usabilidad, eficiencia, facilidad de mantenimiento y portabilidad.

También consultamos las páginas corporativas de cada una de los gestores de bases de datos para establecer las características, páginas personales para encontrar un punto de vista más objetivo en cuanto a sus ventajas y desventajas, y programas para hacer benchmarking de servidores web. Otros autores consultados fueron Mario Tamayo y Tamayo para establecer la metodología que utilizaríamos para llevar a cabo el proyecto y realizarlo de la mejor manera posible.

También realizamos pruebas sobre la utilización de las bases de datos y su comportamiento frente a algunos lenguajes de programación y su compatibilidad con estos.

Metodología

El tipo de estudio es Descriptivo ya que se aplica un método exhaustivo de investigación para la obtención de nuevos conocimientos acerca de los gestores de Bases de Datos (Oracle, PostgreSQL, Interbase, Access y MySQL) y su tecnología.

Investigación Cualitativa



Investigación Cuantitativa



Tipo de Estudio

El tipo de estudio es Descriptivo ya que se aplica un método exhaustivo de investigación para la obtención de nuevos conocimientos acerca de los gestores de Bases de Datos (Oracle, PostgreSQL, Interbase, Access y MySQL) y su tecnología.

Población y Muestra

Pequeñas y medianas empresas del Distrito de Barranquilla

Técnica de Recolección de la Información y/o Procedimientos

Para la realización del presente proyecto utilizamos diferentes técnicas de recolección de información como consultas de libros especializados, consultas de páginas web corporativas y personales relacionadas con el tema y entrevistas informales a personas que han trabajado con alguna de las bases de datos estudiadas.

Análisis de Resultados



RESUMEN TRABAJO DE INVESTIGACIÓN

F-GI-01

Fecha vigencia:

Este es el resultado de nuestro análisis del tests consultado para el proyecto investigativo, para mayor información remitirse al documento. Hemos intentado que sean lo más representativos posibles. No tratamos de decir cuál base de datos es la mejor de todas sino establecer cuál se desempeñó de forma óptima en el test.

MySQL escala muy mal. Bajo pocas conexiones respondía perfectamente, cumpliendo expectativas, pero no pudo con el Interbase en el tramo final, MySQL se llevó el premio del test por fichero. MySQL gana siempre en velocidad

PostgreSQL 7.X realmente ha mejorado la velocidad respecto las versiones 6.X. Lo normal en las 6.X es que fueran 3 veces más lentas que el MySQL. Ahora bajo pocas conexiones apenas es la mitad, y la iguala en un entorno cargado. Pierde mucho en las inserciones debido a que guarda físicamente cada escritura que hace.

Interbase es el mejor en todos los campos en este terreno. Le pertenecen los mejores tiempos en cada test. Es el único que llega a 50 peticiones concurrentes sin dar errores y además el que mejor escala. Interbase se lleva el de base de datos ideal para servidor Internet, da tiempos moderados en todos los campos.



RESUMEN TRABAJO DE INVESTIGACIÓN

F-GI-01

Fecha vigencia:

Conclusiones

Las bases de datos hoy en día son una pieza esencial para cualquier proyecto. Una mala elección puede significar un fracaso. Según el benchmark realizado entre Oracle, MySQL, Access, PostgreSQL e Interbase podemos opinar sobre cada una de ellas.

MySQL es fácil de manejar e instalar, existe una infinidad de documentación en la red sobre esta base de datos, es rápido, es compatible con muchos lenguajes de programación aunque el duo perfecto lo hace con PHP, es open source lo que conlleva a que tenga muchos seguidores, facilita la administración de usuarios, ideal para bases de datos en Internet, el shell de comandos muestra una interfaz más amena y los comandos para gestionar la base de datos son más intuitivos. Aunque también tiene sus debilidades como que no posee integridad referencial y no implementa una buena escalabilidad.

PostgreSQL es una base de datos orientada a objetos, implementa herencia, permite la gestión de diferentes usuarios, tiene una gran escalabilidad y posee integridad referencial. Las impresiones en contra son que no se encuentra mucha documentación en la web, se requieren conocimientos avanzados la óptima utilización de éste, consume gran cantidad de recursos y es 2 o 3 veces más lento que MySQL.

Interbase es la solución propicia para los desarrolladores que quieran utilizar Open Source para un proyecto Web pequeño o medio. Es estable, rápido, escalable y tiene las suficientes funcionalidades. Además es compatible SQL 92 a nivel de entrada. El único problema de Interbase es que hay muy poca documentación por la red.

Access es la base de datos de Microsoft, no es muy estable, pero es muy llamativa gracias a su ambiente gráfico y por pertenecer al conglomerado Microsoft, facilita la administración de datos, aumenta la productividad mediante el uso de los asistentes y las macros, estos permiten automatizar fácilmente muchas tareas sin necesidad de programar.

Oracle es para nosotros la base de datos mas estable y segura, nos ofrece muchas herramientas, soporta un ambiente cliente servidor, interactúa en todas las plataformas y soporta bases de datos de todos los tamaños y además posee una excelente documentación en Internet.

En conclusión, todas las bases de datos estudiadas tienen sus ventajas y desventajas es deber de nosotros como usuarios saber aprovecharlas y manejarlas de la mejor manera posible, para lograr un óptimo rendimiento de ellas.

Recomendaciones



RESUMEN TRABAJO DE INVESTIGACIÓN

F-GI-01

Fecha vigencia:

Análisis de la Bibliografía

Número total de fuentes: 24

Libros: 6

Revistas: ninguna

Periódicos: ninguno

Internet: 18

Otros: ninguno

Documentos
especializados: 24

Documentos
generales: 0

Clasificación de fuentes por año de publicación

1988 1

1990 1

1998 1

1999 4

2001 4

2002 4

2003 5

2004 4

Aplicado por: Patty Pedroza