



Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas

Comparación entre MySQL vs. PostgreSQL

**Trabajo de graduación previo a la obtención del título de
Ingeniero de Sistemas**

Autor: Jessica D. Cambi Alvarado

Jessica A. Zúñiga García

Director: Ing. Oswaldo Merchán Manzano

Cuenca, Ecuador

2006

Agradecimientos

Todos mis esfuerzos realizados a lo largo de mi estudio universitario, los sacrificios, las motivaciones y logros de mi carrera les dedico con todo mi cariño y con mil gracias a mis padres Román y Rosario, que estuvieron siempre junto a mi y vivieron todas las experiencias, tropiezos y triunfos que alcancé; pero no puedo dejar de lado la incondicional ayuda de mi esposo Polo y de las alegrías que me dio mi hijo en momentos duros y de desesperación que me llevaron a la idea de no desistir y dejarme vencer. A todos ustedes gracias.....

Jessica Zúñiga G.

El presente trabajo no hubiera sido posible, si no fuera por todas las bendiciones que Dios ha puesto en vida, mi madre y mis hermanos Angie y Dorian, que siempre me han apoyado y han cuidado de mí. Gracias por estar siempre dándome ánimos en momentos difíciles y felicitándome en buenos tiempos.

Dios les Bendiga.....

Jessica Cambi A.

Índice de Contenidos

Agradecimientos.....	2
Índice de Contenidos.....	3
Índice de Anexos.....	5
Resumen.....	6
Abstract.....	7
Introducción.....	8
Capítulo 1: Sistemas de Gestión de Bases de Datos MySQL y PostgreSQL.....	9
Introducción.....	9
1.1 Introducción a MySQL y PostgreSQL.....	9
1.2 Historia.....	10
1.3 Metodología y Principios.....	12
1.4 Conclusiones.....	18
Capítulo 2: Versiones y Aplicaciones.....	19
Introducción.....	19
2.1 Versiones Existentes y sus Avances.....	20
2.2 Aplicaciones Existentes en el mercado.....	29
2.3 Conclusiones.....	30
Capítulo 3: Instalación.....	31
Introducción.....	31
3.1 Requerimientos de Instalación.....	31
3.2 Acceso a los Instaladores.....	32
3.3 PostgreSQL 8.1 sobre plataforma Windows.....	32
3.4 MySQL 5.0 sobre plataforma Windows.....	33
3.5 Conclusiones.....	34
Capítulo 4: Base de Datos.....	35
Introducción.....	35
4.1 Análisis de la Base de Datos a ser utilizada (Locutorio ETAPA).....	35
4.2 Preparación de datos.....	37
4.3 Conclusiones.....	40
Capítulo 5: Tests de Rendimiento.....	41
Introducción.....	41
5.1 Migración de datos desde Access.....	41
5.2 Migración de datos desde un Archivo de texto.....	42

5.3 Velocidad en búsqueda y ordenamiento de datos.....	43
5.4 Velocidad en procedimiento en cálculo de columna.....	45
5.5 Capacidad de Subconsultas.....	46
5.6 Combinación de resultados por medio de union.....	47
5.7 Respaldo de datos (backups).....	49
5.8 Seguridad de Password.....	49
5.9 Conclusiones.....	50
 Capítulo 6: Cuadros Comparativos.....	 52
Introducción.....	52
6.1 Recopilación de datos de tests.....	52
6.2 Realización de cuadros estadísticos.....	54
6.3 Ventajas y Desventajas.....	62
6.3.1 Características.....	62
6.3.2 Requerimientos.....	64
6.3.3 Instalaciones.....	64
6.3.4 Resultados de Tests.....	64
6.4 Conclusiones.....	65
 Capítulo 7. Conclusiones y Recomendaciones.....	 66
Introducción.....	66
7.1 Conclusiones Teóricas.....	66
7.2 Conclusiones Metodológicas.....	67
7.3 Conclusiones Pragmáticas.....	69
7.4 Recomendaciones.....	69
 Referencias.....	 70

Índice de Anexos

Contenido de Anexos	71
Anexo 1	73
Anexo 2	79
Anexo 3	81
Anexo 4	84
Anexo 5	85
Anexo 6	86
Anexo 7	87

MySQL 5.0 vs. PostgreSQL 8.1

Los dos sistemas de gestión de bases de datos más populares son sin duda MySQL y PostgreSQL, considerados por muchos, de los mejores que existen en el mercado. El presente trabajo muestra una introducción a estos dos gestores de datos, especialmente las versiones MySQL 5.0 y PostgreSQL 8.1, últimas versiones lanzadas por sus respectivos desarrolladores. Además se muestra la historia, metodologías manejadas, principios, versiones y aplicaciones existentes, instalaciones, así como también el análisis de la base de datos a ser usada, la misma que se trata de un locutorio perteneciente a Etapa con aproximadamente 9000 datos. Para probar las potencialidades que ofrece cada gestor se realiza tests de rendimiento en cuanto se refiere a migración de datos, velocidad en búsqueda de datos, backups, etc. y cuadros comparativos con el objetivo de saber cuáles son las ventajas y desventajas tanto de MySQL como de PostgreSQL, que son magníficos sistemas de gestión de bases de datos. El objetivo de este trabajo es tratar de guiar al lector al momento de seleccionar el más apropiado Sistemas Gestor de base de datos.

MySQL 5.0 vs. PostgreSQL 8.1

The two systems of administration databases more popular are without a doubt MySQL and PostgreSQL, considered by many, of the best that exist in the market. The present work shows an introduction to these two agents of data, especially the last versions rushed by its respective developers MySQL 5.0 and PostgreSQL 8.1. Beside, it shows the history, managed methodologies, principles, installers, versions and existent applications; also, the analysis of the databases from an Etapa's phone booth with 9000 data. To prove the potentialities that each agent offers is carried out benchmark of migration of data, speed in search of data, backups, etc, and comparatives squares with objective of knowing which the advantages and disadvantages of MySQL are like of PostgreSQL, that are magnificent systems of administration of databases. The object of this work is try to guide the reader to the most appropriate election at moment to select an SGDB.

Introducción

En la actualidad la tecnología avanza y desarrolla nuevas herramientas de bases de datos mucho más poderosas y que brindan los mismos beneficios, resultados y características de otras ya existentes en el mercado. Esto hace que un Ingeniero de Sistemas siempre esté investigando acerca de las nuevas propuestas que surgen ya sean estas libres o comerciales, para poder asesorar a las empresas con las alternativas con las que cuenta, para la posible administración de sus datos.

MySQL y PostgreSQL son los dos sistemas de gestión de bases de datos que están actualmente en auge, debido a que son las más conocidas en el mercado. Muchas personas confían en MySQL por ser la más antigua, mientras que otras prefieren experimentar con nuevas herramientas como es PostgreSQL, esta divergencia que se mantiene, es debido a las características y funcionalidades que posee cada una, por lo que nace la pregunta ¿Cuál es la mejor?, y que llevó a la iniciativa de investigar sobre ellas.

“*Comparación entre MySQL vs. PostgreSQL*” es el título que lleva esta monografía fundamentada en escritos y documentos publicados en los sitios www.mysql.com y www.postgresql.org páginas oficiales de los dos sistemas a investigar. Además algunas consultas realizadas en libros sobre MySQL, PostgreSQL y artículos relacionados al tema. El presente trabajo muestra las características de estos dos gestores, su historia, versiones, aplicaciones, instalaciones y desde luego su rendimiento a través de cuadros comparativos para así dar a conocer sus ventajas y desventajas objetivo principal de esta monografía.

En cuanto a la realización de este documento, cabe indicar que se utilizó la metodología de investigación documental ya que se recopiló información sobre estas dos bases de datos tanto en libros, documentos y artículos descargados del Internet. Adicionalmente se usó la metodología experimental por lo que se realizaron pruebas de rendimiento tanto en MySQL como en PostgreSQL, utilizando la misma base de datos, con el fin de demostrar y comprobar cada una de sus potencialidades.

CAPITULO 1

SISTEMAS DE GESTIÓN DE BASES DE DATOS MYSQL Y POSTGRESQL

Común es la pregunta entre las personas que se adentran por primera vez en el mundo de las bases de datos ¿MySQL o PostgreSQL?. En realidad no es una pregunta asociada específicamente a los "novatos", ya que incluso los profesionales dedicados a este campo se realizan muchas veces esta misma pregunta. MySQL y PostgreSQL son dos bases de datos muy conocidas en la actualidad por lo que se presenta una breve introducción, reseña histórica de cada una, versiones y aplicaciones de cada una de ellas.

1.1 Introducción a MySQL y PostgreSQL

En el mundo, MySQL es la base de datos de código abierto más usada. Su ingeniosa arquitectura lo hace extremadamente rápida y fácil de personalizar. La extensiva reutilización del código dentro del software y una aproximación minimalística para producir características funcionalmente ricas, ha dado lugar a un sistema de administración de la base de datos muy buena en velocidad, compactación, estabilidad y facilidad de despliegue. Esta gran aceptación de MySQL es debida, en parte, 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. Su principal objetivo de diseño fue la velocidad, por ello se suprimieron algunas características de los demás SGBDs (Sistema de Gestión de Base de Datos), como las transacciones y las subselects. Consume pocos recursos y se distribuye bajo licencia GPL (a partir de la versión 3.22).

Por otra parte PostgreSQL es un servidor de base de datos relacional libre, liberado bajo la licencia BSD lo que significa que cualquiera puede acceder a su código fuente y modificarlo a su voluntad o necesidad y se puede obtener descargándolo libremente de su página Web para múltiples plataformas.

PostgreSQL se diseñó como una base de datos orientada a objetos, es decir, una ORDBMS. Es decir, que las tablas no son tablas, sino objetos, y las tuplas son instancias de ese objeto y usa un modelo cliente-servidor. Además este gestor de base de datos es un producto Open Source y disponible sin costo, desarrollado originalmente en el Departamento de Ciencias de Computación de UC Berkeley, fue pionero en muchos de los conceptos de objetos y relacionales que ahora están apareciendo en algunas bases de datos comerciales.

1.2 Historia

MySQL fue creada por la empresa MySQL AB fundada en 1995 una de las más grandes empresas de software libre del mundo. Aunque sus oficinas principales están ubicadas en Uppsala, Suecia y Cupertino, California, USA tiene más de 200 empleados en más de 20 países gracias a su estrategia de teletrabajo (trabajo a distancia con preponderancia de la informática y las telecomunicaciones).

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.

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 (aunque bien podrían tratarse de la misma), han dado lugar al nombre de este conocido gestor de bases de datos. El nombre del Delfín de MySQL (logo de MySQL) es "Sakila," que fue escogido por los fundadores de MySQL AB de una lista inmensa de nombres sugeridos por usuarios en un concurso llamado "Denomina el Delfin". El nombre victorioso fue sometido por Ambrose Twebaze, un revelador Abierto de software de Fuente de Swazilandia, Africa. Según Ambrose, el nombre femenino Sakila tiene sus raíces en SiSwati, el idioma local

de Swazilandia. Sakila es también el nombre de un pueblo en Arusha, Tanzania, país de origen cercano de Ambrose, Uganda.

En cuanto a la historia de PostgreSQL tiene su origen desde 1977. Empezó como el Proyecto Ingres de la Universidad California en Berkeley. El líder del proyecto, Michael Stonebraker abandonó Berkeley para comercializar Ingres en 1982. Más tarde Ingres fue desarrollado comercialmente por Relational Technologies/ Ingres Corporation. Tras su retorno a Berkeley en 1985, Stonebraker comenzó un proyecto post-Ingres para resolver los problemas con el modelo de base de datos relacional que habían sido aclarados a comienzos de los años 80. El principal de estos problemas era la incapacidad del modelo relacional de comprender "tipos", es decir, combinaciones de datos simples que conforman una única unidad. Actualmente estos son llamados objetos.

El proyecto resultante, llamado Postgres, era orientado a introducir la menor cantidad posible de funcionalidades para completar el soporte de tipos. Estas funcionalidades incluían la habilidad de definir tipos, pero también la habilidad de describir relaciones las cuales hasta ese momento eran ampliamente utilizadas pero mantenidas completamente por el usuario. En Postgres la base de datos "comprendía" las relaciones y podía obtener información de tablas relacionadas utilizando reglas.

Comenzando en 1986, el equipo liberó una serie de documentos describiendo la base del sistema y en 1988 poseían un prototipo funcional. Un Sistema de Bases de Datos Objeto-Relacional llamado Postgres. Entre los documentos liberados tenemos:

- The Design of Postgres.
- The Postgres Data Model.
- The Design of the Postgres Rules System.
- The Postgres Storage System.

La versión 1 fue liberada a un pequeño grupo de usuarios en junio de 1989, seguido por la versión 2 con un sistema de reglas reescrito en junio de 1990. Para

la versión 3, liberada en 1991, el sistema de reglas fue reescrito nuevamente, pero también agregó soporte para múltiples administradores de almacenamiento y un sistema de consultas mejorado. Hacia 1993, Postgres había crecido inmensamente en popularidad y poseía una demanda asfixiante de nuevas funcionalidades. Tras liberar la versión 4, la cual era una simple versión de limpieza, el proyecto fue abandonado.

A pesar de que el proyecto Postgres hubiese finalizado oficialmente, la licencia BSD bajo la cual Postgres había sido liberado permite a desarrolladores de código abierto el obtener una copia del código para continuar su desarrollo. En 1994, Andrew Yu y Jolly Chen añadieron un intérprete de lenguaje SQL a Postgres. En 1996 se elige un nuevo nombre, PostgreSQL, para reflejar la relación entre el Postgres original y las versiones más recientes con capacidades SQL.

1.3 Metodología y Principios

El diseño de MySQL Server es multi-capa, con módulos independientes. Algunos de los últimos módulos se listan a continuación con una indicación de lo bien testeados que están:

- Replicación (Estable)

Hay grandes grupos de servidores usando replicación en producción, con buenos resultados. Se trabaja para mejorar características de replicación en MySQL 5.x.

- InnoDB tablas (Estable)

El motor de almacenamiento transaccional InnoDB es estable y usado en grandes sistemas de producción con alta carga de trabajo.

- BDB tablas (Estable)

El código Berkeley DB es muy estable, pero todavía se está mejorando con la interfaz del motor de almacenamiento transaccional BDB en MySQL Server.

- Búsquedas Full-text (Estable)

Búsquedas Full-text es ampliamente usada.

- MyODBC 3.51 (Estable)

MyODBC 3.51 usa ODBC SDK 3.51 y es usado en sistemas de producción ampliamente. Algunas cuestiones surgidas parecen ser cuestión de las aplicaciones que lo usan e independientes del controlador ODBC o la base de datos subyacente.

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

- Interioridades y portabilidad
 - Escrito en C y en C++
 - Probado con un amplio rango de compiladores diferentes
 - Funciona en diferentes plataformas.
 - Usa GNU Automake, Autoconf, y Libtool para portabilidad.
 - APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
 - Uso completo de multi-threaded mediante threads del kernel. Pueden usarse fácilmente multiple CPUs si están disponibles.
 - Proporciona sistemas de almacenamiento transaccionales y no transaccionales.
 - Usa tablas en disco B-tree (MyISAM) muy rápidas con compresión de índice.
 - Relativamente sencillo de añadir otro sistema de almacenamiento. Esto es útil si desea añadir una interfase SQL para una base de datos propia.
 - Un sistema de reserva de memoria muy rápido basado en threads.
 - Joins muy rápidos usando un multi-join de un paso optimizado.
 - Tablas hash en memoria, que son usadas como tablas temporales.
 - Las funciones SQL están implementadas usando una librería altamente optimizada y deben ser tan rápidas como sea posible. Normalmente no hay reserva de memoria tras toda la inicialización para consultas.
 - El código MySQL se prueba con Purify (un detector de memoria perdida comercial) así como con Valgrind, una herramienta GPL.

- El servidor está disponible como un programa separado para usar en un entorno de red cliente/servidor. También está disponible como biblioteca y puede ser incrustado (linkado) en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.
- Tipos de columnas
 - Diversos tipos de columnas: enteros con/sin signo de 1, 2, 3, 4, y 8 bytes de longitud, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMESTAMP, YEAR, SET, ENUM, y tipos espaciales OpenGIS. Consulte.
 - Registros de longitud fija y longitud variable.
- Sentencias y funciones
 - Soporte completo para operadores y funciones en las cláusulas de consultas SELECT y WHERE.
 - Soporte completo para las cláusulas SQL GROUP BY y ORDER BY. Soporte de funciones de agrupación (COUNT(), COUNT(DISTINCT ...), AVG(), STD(), SUM(), MAX(), MIN(), y GROUP_CONCAT()).
 - Soporte para LEFT OUTER JOIN y RIGHT OUTER JOIN cumpliendo estándares de sintaxis SQL y ODBC.
 - Soporte para alias en tablas y columnas como lo requiere el estándar SQL.
 - DELETE, INSERT, REPLACE, y UPDATE devuelven el número de filas que han cambiado (han sido afectadas). Es posible devolver el número de filas que serían afectadas usando un flag al conectar con el servidor.
 - El comando específico de MySQL SHOW puede usarse para obtener información acerca de la base de datos, el motor de base de datos, tablas e índices. El comando EXPLAIN puede usarse para determinar cómo el optimizador resuelve una consulta.

- Los nombres de funciones no colisionan con los nombres de tabla columna. Por ejemplo, ABS es un nombre válido de columna. La única restricción es que para una llamada a una función, no se permiten espacios entre el nombre de función y el '(' a continuación.
- Puede mezclar tablas de distintas bases de datos en la misma consulta (como en MySQL 3.22).
- Seguridad
 - Un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.
- Escalabilidad y límites
 - Soporte a grandes bases de datos.
 - Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2). Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes (500 antes de MySQL 4.1.2). Un índice puede usar prefijos de una columna para los tipos de columna CHAR, VARCHAR, BLOB, o TEXT.
- Conectividad
 - Los clientes se pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma. En sistemas Windows de la familia NT (NT,2000,XP, o 2003), los clientes pueden usar named pipes para la conexión. En sistemas Unix, los clientes pueden conectar usando ficheros socket Unix.
 - En MySQL 5.0, los servidores Windows soportan conexiones con memoria compartida si se inicializan con la opción --shared-memory. Los clientes pueden conectar a través de memoria compartida usando la opción --protocol=memory.
 - La interfaz para el conector ODBC (MyODBC) proporciona a MySQL soporte para programas clientes que usen conexiones ODBC (Open Database Connectivity). Por ejemplo, puede usar MS

Access para conectar al servidor MySQL. Los clientes pueden ejecutarse en Windows o Unix. El código fuente de MyODBC está disponible. Todas las funciones para ODBC 2.5 están soportadas, así como muchas otras.

- La interfaz para el conector J MySQL proporciona soporte para clientes Java que usen conexiones JDBC. Estos clientes pueden ejecutarse en Windows o Unix. El código fuente para el conector J está disponible.
- Localización
 - El servidor puede proporcionar mensajes de error a los clientes en muchos idiomas.
 - Soporte completo para distintos conjuntos de caracteres, incluyendo latin1 (ISO-8859-1), german, big5, ujis, y más. Por ejemplo, los caracteres escandinavos 'â', 'ä' y 'ö' están permitidos en nombres de tablas y columnas. El soporte para Unicode está disponible
 - Todos los datos se guardan en el conjunto de caracteres elegido. Todas las comparaciones para columnas normales de cadenas de caracteres son case-insensitive.
 - La ordenación se realiza acorde al conjunto de caracteres elegido (usando colación Sueca por defecto). Es posible cambiarla cuando arranca el servidor MySQL.
- Clientes y herramientas
 - MySQL server tiene soporte para comandos SQL para chequear, optimizar, y reparar tablas. Estos comandos están disponibles a través de la línea de comandos y el cliente mysqlcheck. MySQL también incluye myisamchk, una utilidad de línea de comandos muy rápida para efectuar estas operaciones en tablas MyISAM. Consulte.
 - Todos los programas MySQL pueden invocarse con las opciones --help o -? para obtener asistencia en línea.

Sin embargo PostgreSQL ofrece un control de concurrencia multi-versión, soportando SQL, además de una gran variedad de lenguajes de programación.

Las características positivas que posee este gestor son:

- Posee una gran escalabilidad. 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.
- 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.
- Claves ajenas también denominadas Llaves ajenas o Llaves Foráneas (foreign keys)
- Disparadores (triggers)
- Vistas
- Integridad transaccional
- Acceso concurrente multiversión (no se bloquean las tablas, ni siquiera las filas, cuando un proceso escribe)
- Capacidad de albergar programas en el servidor en varios lenguajes.
- Herencia de tablas
- Tipos de datos y operaciones geométricas

1.4 Conclusiones

- MySQL y PostgreSQL son sistemas de gestión de datos, cuyas licencias son tanto libres como comerciales.
- Tanto la historia de MySQL como de PostgreSQL indican que ambos SGBD surgen debido a la necesidad que hubo de manejar gran cantidad de datos en las empresas.
- Ambos sistemas incluyen magníficas características, ya sean estas de seguridad, transacciones o conectividad, las cuales las llevan a ser las más conocidas en el mercado.

CAPITULO 2

VERSIONES Y APLICACIONES

En el mundo actual podemos encontrarnos con diferentes tipos de versiones y aplicaciones tanto de MySQL y PostgreSQL. Las mismas que son utilizadas por distintas empresas de acuerdo a sus necesidades.

La actual serie en producción de prueba es MySQL 5.1, versiones anteriores son: 5.0, cuya versión estable es la 5.0.18, publicada en agosto del 2005. La serie de producción anterior es la MySQL 4.1, cuya versión estable es 4.1.7, publicada en octubre del 2004. Estatus de producción significa que el futuro del desarrollo 5.0 y 4.1. está limitado sólo a arreglar problemas. Para versiones anteriores a MySQL 4.0 y la serie 3.23, sólo se repara virus críticos.

La versión actual de PostgreSQL es la 8, edición de acceso. Esta basado en la versión de PostgreSQL 8.0 RC3, lanzada en enero del 2005. Postgres 8 Edición de Acceso no es apropiada para el uso de la producción. Es apropiado para la evaluación y pruebas.

Postgres Pervasive 8 Edición de Acceso está disponible en el formulario binario desde febrero del 2005 para siguientes plataformas.

- La SUSE Linux Empresa Servidor 9 en el i386
- SUSE Profesional 9.1 en el i386
- Red Hat Enterprise Linux 3 en el i386
- Fedora Core en el i386
- Microsoft Windows 2000/2003/XP en el i386

2.1 Versiones existentes y sus avances

MySQL

MySQL es un servidor multiusuario con la base de datos relacional de código libre más usada en el mundo. Existen diferentes versiones comenzando en el año 1995 cuando empieza la vida de la compañía MySQL AB. La cual constantemente ha estado liberando tanto versiones de prueba como estables, a continuación se detalla las versiones que han sobresalido en el mercado por sus correcciones, control de virus, mejoras de sus características y sus avances en cuanto a rendimiento:

- Versión 4.0
 - El optimizado de consultas de MySQL 4.0 es más inteligente en el uso de índices para resolver las consultas. Algunas consultas que requieren un ordenamiento extra son ahora significativamente más rápidas.
 - 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.
 - MySQL 4.0 dispone de borrados multi-tablas, es decir, eliminar registros relacionados de múltiples tablas al mismo tiempo. Al especificar múltiples tablas y la cláusula WHERE correcta, MySQL hace todo. 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 ha sido mejorado notablemente. Muchos de los cambios fueron hechos en anticipación del sistema de replicación fail-safe. El proceso mismo de replicación 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.
 - El número de variables de estado en MySQL casi se ha duplicado. Se tiene un panorama mucho más claro y amplio. Muchas de las herramientas de administración de MySQL creadas por terceros han sido actualizadas para utilizar estas nuevas variables.
- Versión 4.1
- Subconsultas.
 - Ayuda para todos los comandos desde el cliente mysql.
 - Nuevo formato para la definición de tablas. Permite agregar nuevos tipos para las columnas, más opciones para claves, y guarda definiciones FOREIGN KEY.
 - SHOW COLUMNS FROM nombreTabla (usado en el cliente mysql) ya no debe abrir la tabla, únicamente el archivo de definición de la tabla.
 - Replicación fail-safe.
 - Respaldos en línea sin decrementar la eficiencia del servidor.
 - Soporte más estable para OpenSSL (El soporte SSL en la versión 4.0 no está 100% probado).
 - Implementación del comando RENAME DATABASE.

- Versión 5.0

- Bit Data Type: Puede ser usado para almacenar números en notación binaria. En MySQL 5.0.3, un tipo de datos BIT está disponible para almacenar valores de un bit. (Antes de 5.0.3, MySQL interpreta BIT como TINYINT(1).) En MySQL 5.0.3, BIT lo soporta sólo tablas MyISAM. MySQL 5.0.5 extiende soporte de BIT para MEMORY, InnoDB, y BDB.
- Cursores: Soporte elemental para cursores de un lado del servidor.
- Diccionario de Datos (Esquema de Información): La introducción del esquema de información de la base de datos en MySQL 5.0 proporciona un sencillo estándar como medio para acceder al metadata del servidor de MySQL, esto es, datos acerca de la base de datos (esquemas) en el servidor y los objetos que contienen estos.
- Instance Manager: Puede ser usado para empezar a parar el servidor de MySQL, aún desde un host remoto.
- Precisión Matemática: MySQL 5.0 introduce los criterios más estrictos para la aceptación o el rechazo de datos, e implementó una biblioteca nueva para el punto fijo aritmético. Estos han contribuido a un grado mucho más alto de la certeza para operaciones matemáticas y control más grande sobre valores inválidos.
- Motores de Almacenamiento: Los motores de almacenamiento agregados en MySQL 5.0 incluyen ARCHIVE y FEDERATED.
- Rutinas Almacenadas: MySQL 5.0 implementó procedimientos y funciones almacenadas.
- Modo Estricto y Manejo de Errores Estándar
- Triggers: MySQL 5.0 agrega soporte limitado para triggers (disparadores).

- Tipo de datos VARCHAR: Soporte nativo VARCHAR. La longitud máxima de VARCHAR es 65,532 bytes ahora, y no se cortan espacios en blanco consecutivos.
 - Vistas: MySQL 5.0 agregó soporte para las denominadas vistas.
- Versión 5.1
- Particionamiento: Esta capacidad permite distribuir porciones de tablas individuales a través del filesystem, de acuerdo a reglas que pueden ser puestas cuando la tabla es creada. En efecto, diferentes porciones de la tabla son almacenados en diferentes ubicaciones, pero desde el punto de vista de usuario, la tabla particionada es todavía una sola tabla.
 - Plugin API: MySQL 5.1 agrega soporte para un muy flexible plugin API que permite la carga y descarga de varios componentes del runtime. Aunque el trabajo en esto aún no está terminado, analizadores plugin full-text son el primer paso en esta dirección. Esto permite a los usuarios implementar sus propios filtros de entrada en el texto indexado, permitiendo capacidad de búsqueda full-text en datos arbitrarios como archivos PDF u otros formatos de documentos.
 - The Instante Manager (IM) ahora con una funcionalidad adicional:
 - SHOW instance_name LOG FILES provee una lista de todos los archivos del log usados por la instancia. (Author: Petr Chardin)
 - SHOW instance_name LOG {ERROR | SLOW | GENERAL} size retribuye una parte de lo especificado al archivo del log. (Author: Petr Chardin)
 - SET instance_name.option_name=option_value pone una opción para especificar un valor y escribir en el archivo de configuración. (Author: Petr Chardin)

PostgreSQL

En 1996, Postgres95 desaparece de la academia y comienza una nueva vida de código abierto en el mundo cuando un grupo de desarrolladores fuera de Berkeley ven la promesa de un sistema y deciden continuar su desarrollo.

Contribuyendo con una enorme cantidad de tiempo, destreza, labor, y experiencia técnica, este grupo de desarrolladores transformó radicalmente al Postgres. Sobre los próximos 8 años, ellos trajeron consistencia y uniformidad al código base, crearon y detallaron pruebas de regresión para medidas de calidad, comenzaron con una lista de direcciones para reportes de virus, reparando cada uno de ellos, agregando increíbles nuevas características, y perfeccionando el sistema para rellenar varias grietas tales como documentación para desarrolladores y usuarios.

Con el comienzo de esta nueva vida en el mundo del código abierto, con muchas nuevas características, este sistema de Bases de Datos toma el nombre de PostgreSQL el cual año a año va mejorando y liberando sus diferentes versiones:

- Versión 6.0 – 7.0

- Comenzó con la versión 6.0, dando crédito a muchos años de desarrollo. El PGDG creció gradualmente, y a lo largo y con la ayuda de cientos de otros desarrolladores de todo el mundo, cambiaron y mejoraron el sistema en casi todas las áreas.
- Los próximos 4 años siguientes (versiones 6.0 - 7.0), tuvieron un mejoramiento enorme y nuevas características tales como:
- Control de concurrencia (MVCC). Nivel de seguridad de las tablas fue reemplazado con una sofisticada multiversión de sistema de control de concurrencia, el cual permitía leer y continuar leyendo

constantemente datos durante las actividades de escritura y habilitar en línea backups mientras la base de datos esta corriendo.

- Importantes Características de SQL. Muchos grandes SQL fueron hechos incluyendo subselects, valores por defecto, restricciones, llaves primarias, llaves foreanas, identificadores, cadenas de tipo de relación, tipos de datos binarios y hexadecimales , enteros de entrada hacia otros.
- Instrucción de tipos. Nuevos tipos nativos donde se agregaba incluyendo un ancho del rango de fecha/tiempo y tipos geométricos.
- Mayor velocidad e incremento de ejecución en el orden de un porcentaje de 20 – 40 % de lo que se halla hecho.

- Versión 7.0 – 7.4

- Los siguientes 4 años apareció Write-Ahead Log (WAL), temas de SQL, preparación de consultas, relaciones, consultas complejas, sintaxis de SQL92, TOAST, soporta IPv6, temas de información estándar de SQL, índices, autovacuna, Perl/Python/TCL lenguajes de procedimientos, soporte de mejoramiento de SSL, y optimizar el análisis, información estática de la base de datos, agregar seguridades, funciones de las tablas, y mejoramiento de velocidad entre varias cosas.

- Versión 8.0

- Esperando tener un gran debut dentro del mercado de las bases de datos, incorporando los tan conocidos tablespaces. Java almaceno algunos procedimientos, puntos de recuperación, repetición asincrónica, y save points (puntos de guarda). Con esto viene una característica de un puerto nativo de Windows.

- Con la versión 8.0 se pueden realizar las siguientes tareas:
 - Instalar un número ilimitado de veces sin temor de sobrepasar la cantidad de licencias, la principal preocupación de muchos proveedores de bases de datos comerciales
 - Obtener velocidad y rendimiento excepcionales
 - Obtener fiabilidad a toda prueba
 - Utilizar su flexibilidad para extenderse según se requiera
 - Manejar un diseño altamente escalable
 - Tener muchas opciones de soporte
 - Operar con mínimos requerimientos de administración
 - Obtener bajo Costo Total de Operación (TCO)
 - Obtener adaptación a Estándares ANSI

- Versión 8.1

- Contiene más de 120 nuevas características y las mejoras que contribuirán a más de 200 desarrolladores. Algunas de las principales son:
- Roles: PostgreSQL ahora soporta roles de bases de datos, los que simplifican el manejo de grandes cantidades de usuarios con esquemas complejos de privilegios superpuestos.
- Parámetros de entrada/salida (IN/OUT): Las funciones ahora soportan parámetros de entrada (IN), salida (OUT) y entrada/salida (INOUT), lo cual mejora sustancialmente el soporte de lógica compleja y aplicaciones J2EE y .NET.
- Compromiso en Dos Fases (Two-Phase Commit, 2PC): esta característica, muy necesaria para aplicaciones WAN y centros de cómputo heterogéneos, permite transacciones conformes con ACID a través de servidores distribuidos.

- Rendimiento Multiprocesador (SMP) mejorado: el gestor de memoria ha sido mejorado para 8.1, de manera que escala casi linealmente con el número de procesadores. Esto conlleva mejoras significativas de rendimiento en sistemas de 8 vías, 16 vías, dual-core y multi-core.
- Recorridos de Mapas de Bits: los índices son convertidos a mapas de bits en memoria cuando es apropiado, otorgando hasta veinte veces más rendimiento en consultas complejas para tablas muy grandes. Esto también ayuda a simplificar la administración de bases de datos reduciendo significativamente la necesidad de índices multi-columna.
- Particionamiento de Tablas: El optimizador de consultas es capaz de evitar recorrer secciones completas de tablas grandes, a través de una técnica conocida como Exclusión por Restricciones. Similar a las características de Particionado de Tablas de otros sistemas gestores de datos, esta característica mejora tanto el rendimiento como la gestión de datos para tablas de varios gigabytes.
- Bloqueos Compartidos de Registros: El modelo de bloqueos «mejor que a nivel de registro» de PostgreSQL ahora soporta niveles de concurrencia aún mayores, a través de la adición de candados compartidos a nivel de registro para llaves foráneas. Estos candados compartidos mejorarán el rendimiento de inserción y actualización para muchas aplicaciones OLTP de gran concurrencia.
- «PostgreSQL 8.1 ofrece una mejora de rendimiento enorme, a todos los niveles, en nuestros servidores Opteron de doble procesador», según Merlin Moncure, Administrador de Bases de Datos para Reliable Computer Solutions. «Más específicamente, estamos viendo alrededor de un 20% de reducción en los tiempos de ejecución de consultas simples, y una reducción adicional de un 20% en el nivel de carga de CPU, para una reducción total de carga del servidor de entre 20 y 40%».

- Además de las características principales listadas en el comunicado de prensa, hay más de 120 características nuevas y mejoras contribuidas por más de 200 desarrolladores para la versión 8.1. Destacan entre ellas:
- GiST: El Árbol de Búsqueda Generalizado (Generalized Search Tree, GiST) de PostgreSQL, mecanismo extensible de indexamiento, ha sido mejorado de manera que soporta concurrencia de alta velocidad, la recuperación y rendimiento de actualizaciones, que antes estaba disponible sólo para los índices B-Tree. GiST es el mecanismo central para el soporte de indexamiento total de texto (TSearch2), geo-espacial (GIS) y de estructuras de árbol. Con esta mejora, los tipos de datos complejos tendrán buen rendimiento aún en grandes aplicaciones de alta disponibilidad.
- Refactorización de COPY: COPY ha sido refactorizado, lo cual ha acelerado más de un 30% las operaciones de carga masiva de datos. Junto con las mejoras de carga CSV, esto permite cargas de bases de datos grandes aún más rápidamente.
- Memoria Compartida de 64 bits: el gestor de buffer ha sido mejorado, y ahora soporta la utilización de hasta 2 terabytes de RAM en plataformas de 64 bits, preparando a PostgreSQL para los servidores de datos del futuro.
- Autovacuum Integrado: el sistema automatizado de mantenimiento de PostgreSQL ha sido mejorado e integrado en el proceso servidor, haciendo los servidores PostgreSQL más simples de configurar y administrar.
- Agregación más rápida: Las funciones de agregación han sido mejoradas, para hacer las consultas de reportes aún más rápidas. Los desarrolladores reescribieron el manejo de memoria para las funciones de agregación, y además agregaron optimizaciones de índice para MIN() y MAX().

- Funciones de Administración: se agregaron nuevas funciones para obtener información del servidor y para realizar tareas de administración con mayor facilidad desde la línea de mandatos PSQL.
- Funciones de Compatibilidad: se han agregado las funciones lastval(), greatest() y least(), para facilitar el porte de aplicaciones desde MySQL y Oracle.

2.2 Aplicaciones existentes en el mercado

MySQL

MySQL soporta muchos lenguajes distintos de programación como C, C++, Eiffel, Java, Perl, PHP, Python y TCL. También tiene la opción de protección mediante contraseña, la cual es flexible y segura, por lo que MySQL tiene un sin número de aplicaciones.

Una de ellas es el desarrollo de páginas Web interactivas realizadas en PHP con base de datos MySQL. Las mismas que son utilizadas por las empresas de nuestro medio para realizar pedidos vía Internet y para la demostración de catálogos.

PostgreSQL

Por ser una nueva herramienta para el manejo de base de datos no muy conocida, y muy pocos son los usuarios que se inclinan por la utilización de la misma, se cita en este estudio unas de las aplicaciones existentes en América Latina como es:

La aplicación desarrollada por el Gobierno de la ciudad de Buenos Aires en Argentina, a través de la Dirección General de Sistemas GIS. El sistema

ofrece una vista general del plano de la ciudad, las divisiones territoriales, dirección de calles, recorridos a pie, datos útiles y, por supuesto, fotografías satelitales.

Lo interesante de esta aplicación es que está construida íntegramente sobre herramientas de código libre: MapServer, PostgreSQL, PostGIS y Maplab. Dicho mapa esta disponible en la siguiente dirección:
<http://mapa.buenosaires.gov.ar/sig/index.phtml>

2.3 Conclusiones

- Las últimas versiones de los SGBD analizados son la 8.1 para PostgreSQL y para MySQL la 5.1, siendo las más utilizadas ya sea sobre plataforma Windows o Linux.
- La aplicación más significativa de Postgres es la realización de un mapa de la ciudad de Buenos Aires en Argentina, y en cuanto a MySQL las principales aplicaciones son páginas de Internet interactivas.

CAPITULO 3

INSTALACIÓN

Una vez especificado ciertos temas con respecto a estos sistemas de gestión de bases de datos, se realiza la instalación de cada uno de ellos sobre la plataforma Windows, conociendo los requerimientos técnicos y de software que exigen, así como también los accesos a instaladores de las versiones 5.0 y 8.1 que son desde las páginas oficiales tanto de MySQL como de PostgreSQL respectivamente. Dicha instalación se realiza sobre una computadora portátil con las siguientes características: Intel Celeron Pentium 4, 512 de RAM, Disco Duro de 30 G. y procesador de 1.3 GHz.

3.1 Requerimientos de instalación

MySQL

Para funcionar MySQL sobre Windows, se necesita lo siguiente:

- Un sistema operativo de 32-bits, tal como NT, 2000, XP, o un servidor 2003 de Windows. Un sistema operativo NT basado en Windows permite que el servidor de MySQL pueda funcionar como servicio. Se recomienda el uso de un sistema operativo NT basado en Windows.
- Ayuda del protocolo de TCP/IP.
- Bastante espacio para desempaquetar, instalar, y para crear las bases de datos de acuerdo con los requisitos de cada usuario (un mínimo de 200 megabytes se recomienda generalmente.)

Requisitos opcionales:

- Si se planea conectar con el servidor de MySQL vía ODBC, se necesita el siguiente drive “Connector/ODBC”.
- Si las tablas que se van a manejar son de un tamaño más grande que 4GB, es necesario instalar MySQL en un NTFS o un filesystem más nuevo.

PostgreSQL

La instalación de PostgreSQL sobre Windows es sencilla, debido a que no tiene requerimientos específicos ni de software ni de hardware, si se trabaja sobre Windows NT, 2000 o XP, son necesarios aproximadamente 70 MBytes de disco duro, cualquier ordenador capaz de soportar estos sistemas operativos es suficiente.

3.2 Acceso a instaladores

Una copia de la distribución de MySQL para Windows, se puede descargar desde la página oficial www.mysql.com, en donde se encuentran las diferentes versiones existentes tanto para plataformas Linux como Windows, al igual se tiene acceso a varias herramientas y manuales que están disponibles para la administración de este gestor de bases de datos. Para PostgreSQL los instaladores se pueden descargar también desde la página principal de Postgres www.postgresql.org. En esta página a igual que MySQL existe una variedad de herramientas y desde luego la sección de descargas, en la que se encuentra la versión 8.1 tanto para Linux (.tar.gz) como para Windows (.zip).

3.3 PostgreSQL 8.1 sobre plataforma Windows

Una vez conocidos los requerimientos de instalación y luego de haber obtenido el programa se procede a la instalación, en la cual el primer paso consiste en descomprimir y elegir el Paquete “PostgreSQL 8.1”. A continuación el setup nos presenta un Wizard en el que hay que ir configurando las opciones que se indican, como son la configuración correcta del idioma, instrucciones sobre el modelo de licencias y las herramientas a instalar, así como la elección de la base de datos la misma que se configuró como un servicio (por comodidad) y el establecimiento de un usuario y contraseña. Dicho usuario si no existe se crea y por el contrario si existe, no debe estar en el grupo de administradores, tomando en cuenta que el Account name no exista.

También definimos los parámetros de configuración (sobre todo el superusuario y contraseña). Elegimos el nombre usado como lenguaje de Script para los procedimientos almacenados el “PL/pgsql” y por último seleccionamos los módulos opcionales (dejando la configuración por defecto). Todo el procedimiento de la instalación se realizó siguiendo las instrucciones que se detallan en la página <http://pginstaller.projects.postgresql.org>. Complementariamente para poder realizar la migración de datos se instaló la herramienta *Access2PostgreSQL Pro versión 1.1.2*. que permite migrar datos desde Access hacia PostgreSQL. “Access2PostgreSQL” se puede descargar gratis desde la siguiente página: <http://www.data-conversions.net/access-to-postgresql-pro-converter.html>.

3.4 MySQL 5.0 sobre plataforma Windows

Para realizar el estudio entre estos dos gestores de bases de datos, se tenía planificado trabajar con las últimas versiones, al encontrarse con el inconveniente de que la versión MySQL 5.1 para Windows no se encuentra liberada, se ha decidido trabajar con la versión 5.0.18 que es la inmediata inferior a la 5.1 la misma que se encuentra disponible y corregida.

Para el uso de MySQL fue necesario instalar las siguientes herramientas con las especificaciones detalladas a continuación:

MySQL essential 5.0.18-win32

Tipo de Setup	Custom
Configuración del servidor	Server Machine
Tipo de Base de Datos	Base de Datos Multi-funcional
Red TCP/IP (Networking)	# de puerto 3306

Además se instaló como un servicio de Windows.

MySQL Administrador 1.1.7

Tipo de Setup	Completo
---------------	-------	----------

MySQL Query Browser 1.1.19

Tipo de Setup Completo

MySQL Migration 1.0.22

Tipo de Setup Completo

Estas son las especificaciones con las que fue configurado MySQL de acuerdo al uso que se le va a dar estudio, como es la migración de datos entre otras pruebas.

3.5 Conclusiones

- Los instaladores tanto para MySQL como PostgreSQL se encuentran disponibles en las páginas oficiales en Internet, en las que puede conseguir documentación y herramientas para el manejo de cada una de ellas.
- Instalar estos dos SGBD es una tarea simple de realizar; pero no por ser tan fácil se debe dejar de tomar en cuenta ciertas opciones de configuración que van surgiendo a lo largo del proceso, opciones que se pueden revisar en los documentos electrónicos existentes tanto para MySQL como PostgreSQL.
- Comparar los dos gestores de bases de datos.

CAPITULO 4

BASE DE DATOS

Para realizar las pruebas en los dos gestores, se hace un análisis previo de la base de datos que se va a manipular. Dicha base se trata de un Locutorio de Etapa con más de 9000 datos, los suficientes para probar las capacidades de los SGBD puestos a estudio, así como también la preparación de estos datos previa a la migración a MySQL y PostgreSQL.

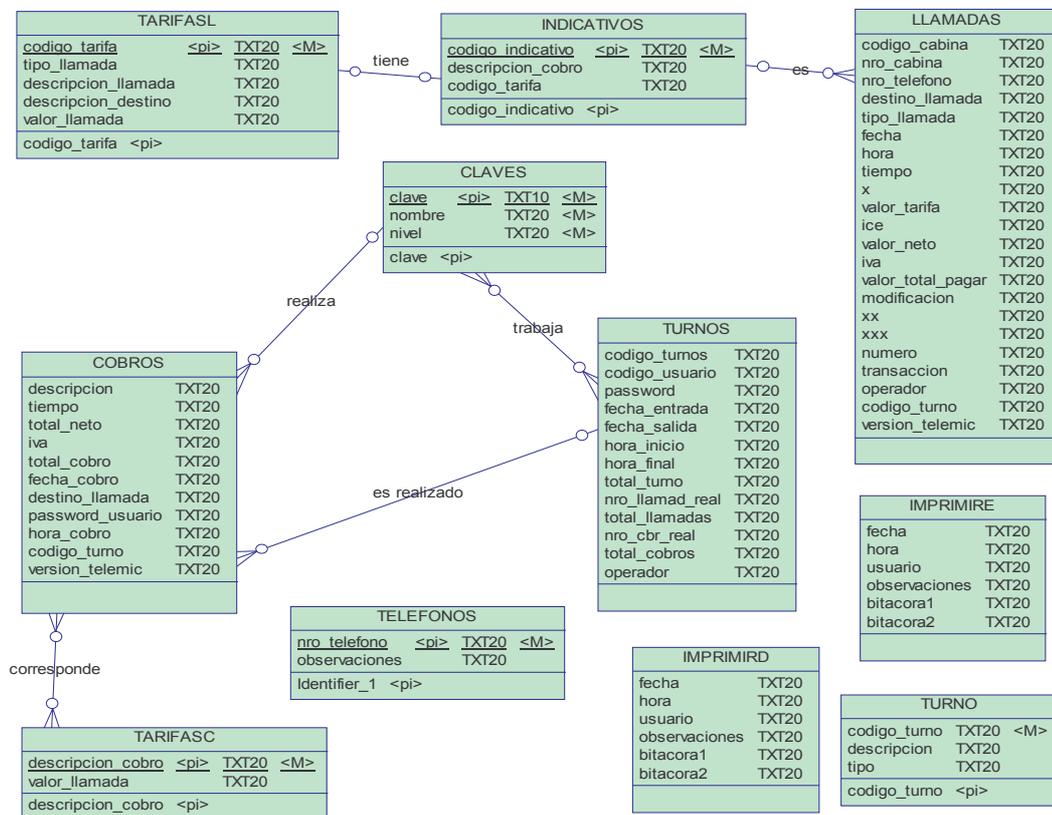
4.1 Análisis de la base de datos a ser utilizada (Locutorio Etapa).

La base de datos con la cual se va a realizar las pruebas pertenece al Locutorio de Etapa ubicado en la Av. España y Av. Huayna Cápac (sector Chola Cuencana) con aproximadamente 9000 datos disponibles en un total de 21 tablas que conforman dicha base. La información presentada corresponde a un mes de trabajo. A continuación se menciona el nombre de cada tabla y su detalle en cuanto se refiere a tipos de datos y su tamaño.

TABLA	DESCRIPCION	REGISTROS
C1-C10	Tablas temporales con la misma estructura de la tabla Llamadas	0
CLAVES	Tabla de operadores con sus respectivos passwords	4
COBROS	Tabla que indica los valores de cobros de servicios de fax	42
IMPRIMIRD	Tabla con datos de auditoria	153
IMPRIMIRE	Tabla temporal con la estructura de la tabla ImprimirD	0
INDICATIVOS	Tabla con datos de destinos de llamada	1680
LLAMADAS	Tabla con datos de las llamadas realizadas	6978
TARIFASC	Tabla con tarifas de servicios	11
TARIFASL	Tabla con tarifas de llamadas	16
TELEFONOS	Tabla de cobros pendientes a números que no han cancelado	5
TURNOS	Tabla temporal del turno actual de operación	19
TURNOS	Tabla con datos de los turnos de los operadores	45
	Total datos:	8953

Para una mejor comprensión de los datos se construyó el modelo conceptual, el mismo que muestra todas las tablas que intervienen en la base de datos del Locutorio de Etapa. Las tablas C1 hasta C10 no se modelaron por cuanto se indicó anteriormente son tablas temporales con la misma estructura de la tabla de “Llamadas” y que son utilizadas en caso de que ocurra algún problema con ésta cuyos datos serán respaldados en las tablas C.

MODELO DE DATOS CONCEPTUAL “LOCUTORIO DE ETAPA”



Nota.- Este modelo fue tomado directamente de la base de datos utilizada en el Locutorio de Etapa sin modificación alguna, por lo que presenta algunas inconveniencias. Se diseñaron las entidades en la herramienta Power Designer para una mejor comprensión.

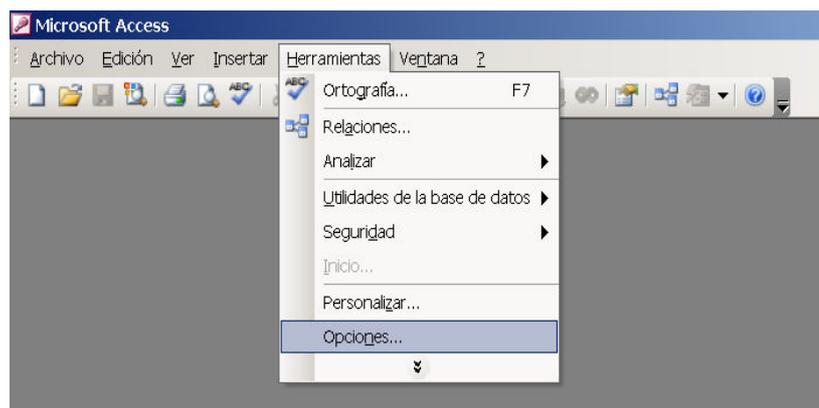
4.2 Preparación de datos

Para la migración básicamente se pueden dividir en métodos directos e indirectos. Los métodos directos abren los archivos .DBF o .MDB, se conectan a la base de datos y transfieren sus contenidos a través de esa conexión. Los métodos indirectos generan uno o más archivos temporales, en base a los que posteriormente se crearán las tablas de la base de datos y/o insertaran los registros. Previo a la migración de los datos de una base a otra, y si es a través de un método indirecto, se deberá preparar los datos. Para los tests propuestos en el estudio será necesario realizar los siguientes pasos en la base de Microsoft Access.

Migración Access – MySQL

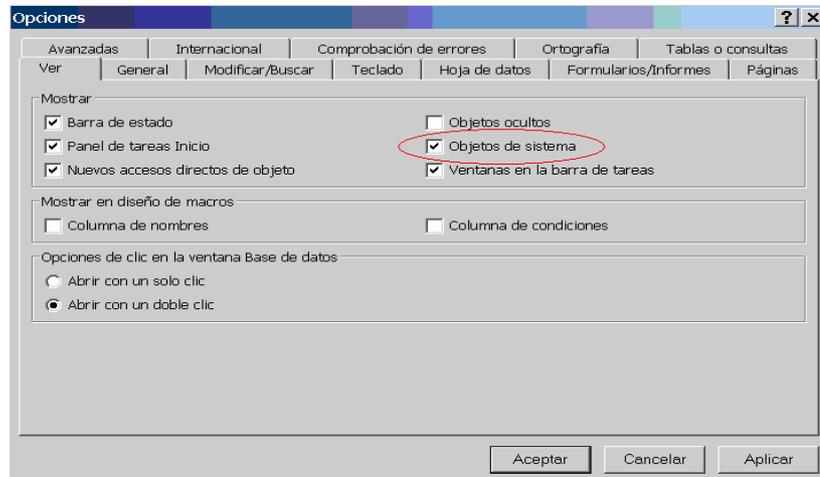
Paso 1

Se deberá abrir la base y se dará un click sobre el menú de *Herramientas*, a continuación se seleccionará *Opciones*.



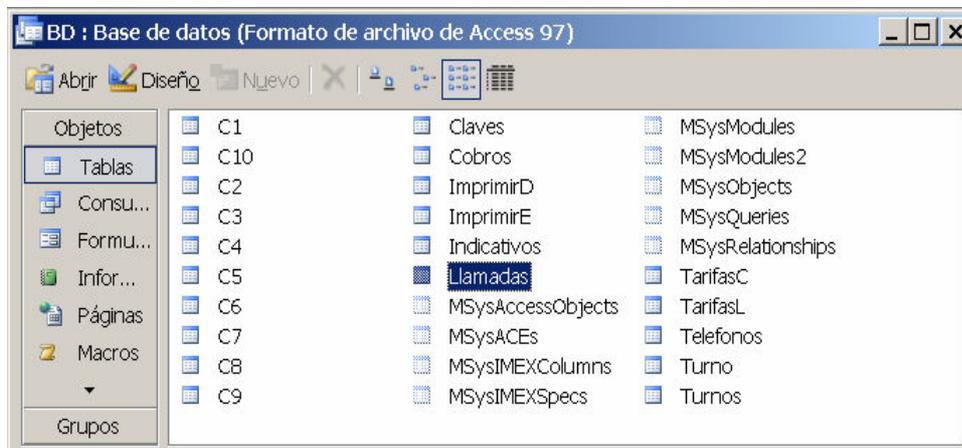
Paso 2

En el diálogo de Opciones se tendrá que señalar el checkbox perteneciente a *Objetos del Sistema* y luego se presionará el botón de Aplicar y Aceptar.



Paso 3

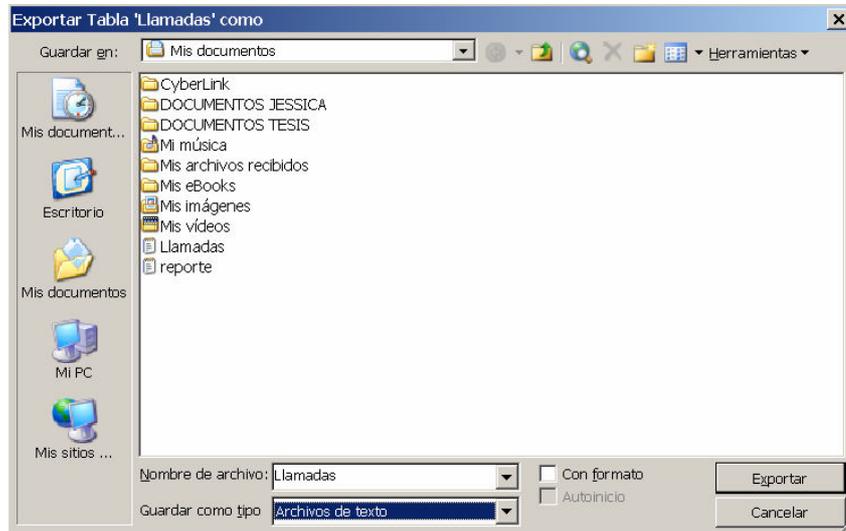
Una vez terminado con el paso 2, nuestra base se mostrará de la siguiente manera y estará lista para ser migrada a MySQL.



Migración Access – Archivo Plano

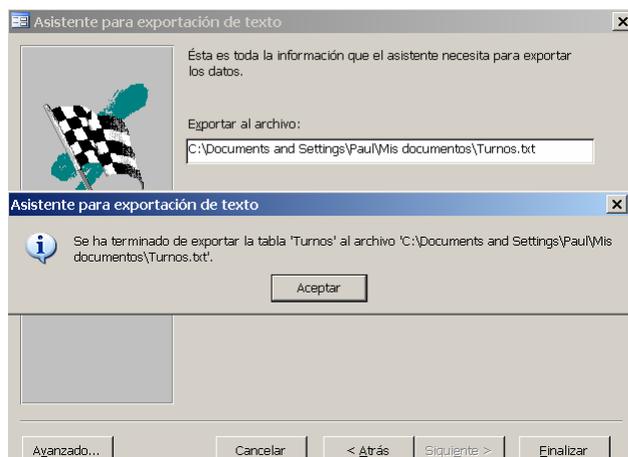
Paso 1

Por cada tabla se usa la opción "Archivo / Guardar como o exportar... / En un archivo o una base de datos externo", en la ventana de diálogo, en la opción "Guardar como tipo:" seleccionar "Archivos de texto"



Paso 2

Luego al presionar el botón de Exportar se nos presentará un asistente en el cual se deberá ir especificando un archivo con el siguiente formato de archivo: Delimitado Delimitador de campo: , Cualificador de texto: " Orden de la fecha: DMA . y estará listo nuestro archivo plano.



Para el caso de la migración Access – PostgreSQL se trabajará gcon el archivo plano.

4.3 Conclusiones

- La base del Locutorio de Etapa nos proporciona un aproximado de 6000 datos de llamadas reales, con los que se realizará el benchMark (comparador de rendimiento) entre los dos gestores de bases.
- Para la preparación de los datos a migrar, existen diferentes métodos tanto directos como indirectos, para realizar los tests del presente estudio se ha decidido trabajar con métodos indirectos ya que se utilizarán herramientas propias de los SGBD como son el Migration Toolkit y Access2PostgreSQL.

CAPITULO 5

TESTS DE RENDIMIENTO

Los tests permiten la mejor manera, de medir el rendimiento completo de un sistema en el mundo real. En el presente capítulo se realizarán diferentes pruebas con los dos SGBD (MySQL, PostgreSQL) y se utilizará la base del Locutorio de Etapa, así como también herramientas propias de cada gestor de datos. Además se utilizará un programa benchmark (*CS FIRE Monitor versión 2.5.0*), a través del cual se ha de medir el rendimiento del computador durante los diferentes tests.

5.1 Migración de datos desde Access

Para migrar desde Microsoft Access se empleó dos herramientas propias de cada gestor de datos, ambos se utilizan para migrar datos desde Access hacia MySQL o hacia PostgreSQL.

MySQL	PostgreSQL
MySQL Migration Toolkit versión 1.0.22	Access2PostgreSQL Pro versión 1.1.2

Como se indica en el cuadro anterior Migration Toolkit es una herramienta gráfica proporcionada por MySQL AB para migrar esquemas y datos desde varios sistemas de base de datos relacionales hacia MySQL, con este ingenio se pudo migrar los datos que se encuentran en la base de Access hacia MySQL. A igual con PostgreSQL se utilizó Access2PostgreSQL, un software de migración de datos para convertir una base de datos de Access a una base de datos en PostgreSQL. Con cada una de ellas se realizaron pruebas, sobre rendimiento del CPU y también se midió cuanta memoria ocupa cada una al momento del proceso de la migración.

Los resultados obtenidos en este testing de migrar datos desde Access a MySQL y a PostgreSQL fueron los siguientes:

MySQL	CPU Utilizado	Memoria Utilizada MB.	PostgreSQL	CPU Utilizado	Memoria Utilizada MB.
Iniciación	2%	177.172	Sesiones	2%	246.41
Conexión a Servidores	25%	213.558	Fuente	2%	247.027
Ingeniería Inversa	73%	227.004	Destino	2%	251.348
Creación de Objetos	11%	224.707	Configuración	2%	252.328
Transferencia datos BD	79%	240.769	Ejecución	91%	263.305

Ver procedimiento de migración tanto de Migration Toolkit como de Access2PostgreSQL en Anexo 1.

El cálculo de la memoria utilizada se realizó de la siguiente manera, las imágenes que se muestran en Anexo 1 indican los MB de memoria disponible en relación con un total de memoria física de 503.371 MB existentes, que, al restarle la memoria disponible se obtiene el total de MB utilizados realmente. Este cálculo se utiliza para todos los resultados de tests realizados en el presente capítulo.

5.2 Migración de datos desde un archivo de texto

En la migración de los datos desde un archivo de texto se uso comandos propios de cada gestor como son “load data” para MySQL y “copy” para PostgreSQL, y, para poder manipular los datos, el software que se menciona a continuación.

MySQL	PostgreSQL
MySQL Query Browser versión 1.1.19	PgAdmin III Query versión 1.4.1

Para realizar la migración de los datos desde un archivo plano, es necesario como se indicó anteriormente preparar los datos en MS Access, luego una vez dispuesto, tanto en MySQL como en PostgreSQL se tienen que crear cada una de las tablas con la misma estructura que en Access para así poder completar con la inserción de los datos. Para mayor entendimiento, *ver forma de migrar datos desde un archivo plano en Anexo 2.*

El rendimiento que mostró cada uno de los gestores al momento de pasar los datos desde el archivo plano se menciona en el subsiguiente cuadro.

	MySQL		PostgreSQL	
	CPU Utilizado	Memoria Utilizada	CPU Utilizado	Memoria Utilizada
Inicio	4%	212.808 MB.	4%	190.293 MB.
Transmisión de datos	22%	196.391 MB.	22%	209.164 MB.

5.3 Velocidad en búsqueda y ordenamiento de datos

Para medir la velocidad de búsqueda de los datos se utilizó varias sentencias SQL aplicadas a la tabla de Llamadas perteneciente al Locutorio de Etapa, la cual contiene la mayor cantidad de datos (6789 filas), dichas sentencias fueron ejecutadas en el mismo software que en el subcapítulo anterior “Migración de datos desde un archivo de texto” y para medir el rendimiento del sistema, el software mencionado anteriormente “CS Fire Monitor”. La primera consulta que se realizó para probar el rendimiento tanto de MySQL como de PostgreSQL fue seleccionar todas las llamadas Internacionales que se realizaron en el Locutorio ordenadas por fecha en forma descendente, los resultados obtenidos fueron los siguientes:

	PostgreSQL	MySQL
CPU Utilizado	19%	17%
Memoria Utilizada	277.785 MB	200.926 MB
Tiempo	781 ms	250.6 ms

La sintaxis SQL para la búsqueda fue la siguiente en los dos sistemas de gestión de datos:

```
SELECT * FROM LLAMADAS
WHERE TIPO_LLAMADA LIKE 'I%'
ORDER BY FECHA DESC;
```

La segunda consulta que se verificó fue obtener cuanto dinero se recaudó en total por cada uno de los tipos de llamadas efectuadas. Como se observa en el ulterior cuadro MySQL consume muy pocos recursos, especialmente en memoria PostgreSQL carga más memoria.

	PostgreSQL	MySQL
CPU Utilizado	8%	27%
Memoria Utilizada	210.148 MB	200.844 MB
Tiempo	110 ms	356.2 ms

La sintaxis SQL empleada fue distinta para los dos sistemas de gestión de base de datos, debido a que MySQL no es muy estricta en cuanto a la sintaxis que utiliza, por el contrario PostgreSQL emplea el mismo lenguaje SQL que Oracle lo que le hace un poco más estricto en su sintaxis, a continuación se indica las sentencias utilizadas para cada uno de ellos.

MySQL SELECT TIPO_LLAMADA, SUM(VALOR_NETO) TOTAL
 FROM LLAMADAS
 GROUP BY TIPO_LLAMADA;

PostgreSQL SELECT TIPO_LLAMADA,
 SUM (TO_NUMBER (VALOR_NETO,'999999990.0000')) AS
 TOTAL
 FROM PRUEBA.LLAMADAS
 GROUP BY TIPO_LLAMADA;

Una tercera búsqueda que se realizó para examinar tanto MySQL como PostgreSQL fue una selección de todas las cabinas en donde se realizaron llamadas que tenían como destino el país Argentina. De la misma forma que en los tests anteriores se presenta los resultados obtenidos en el siguiente cuadro.

	PostgreSQL	MySQL
CPU Utilizado	4%	7%
Memoria Utilizada	236.531 MB	196.332 MB
Tiempo	32 ms	53 ms

La sentencia SQL aplicada para la obtención de estos resultados a igual que en la búsqueda anterior varía en su sintaxis, por ejemplo al momento de nombrar alias PostgreSQL requiere de la palabra “AS” para identificar a un alias, por el contrario MySQL entiende que el nombre que se encuentra a continuación del nombre de la columna de la tabla es un alias. Las consultas SQL empleadas son:

```
MySQL      SELECT NRO_CABINA CABINA,
           DESTINO_LLAMADA DESTINO,
           COUNT (DESTINO_LLAMADA) NRO_LLAMADAS
           FROM LLAMADAS
           WHERE DESTINO_LLAMADA = 'ARGENTINA'
           GROUP BY CABINA, DESTINO;
```

```
PostgreSQL SELECT NRO_CABINA AS CABINA,
           DESTINO_LLAMADA AS DESTINO,
           COUNT (DESTINO_LLAMADA) AS NRO_LLAMADAS
           FROM PRUEBA.LLAMADAS
           WHERE TRIM (DESTINO_LLAMADA) = 'ARGENTINA'
           GROUP BY CABINA, DESTINO;
```

Ver Test de Búsqueda de datos en Anexo 3.

5.4 Velocidad en procedimiento de cálculo de columna

Para medir la velocidad en procedimiento de cálculo de columna se desarrolló la consulta tanto en el Query Browser de MySQL como de PostgreSQL. Dicha consulta fue seleccionar el destino de las llamadas con el respectivo número de llamadas realizadas, el total de tiempo entre todas las llamadas, el valor de los impuestos que se pagó y el valor que se cobró por dichas llamadas. Los resultados que originó esta consulta se muestra en el subsiguiente cuadro.

	PostgreSQL	MySQL
CPU Utilizado	11%	5%
Memoria Utilizada	213.578 MB	231.824 MB
Tiempo	547 ms	558.7ms

La sentencia SQL que se utilizó en los dos gestores de bases de datos fueron las siguientes.

MySQL

```
SELECT * FROM (SELECT CONCAT ('nro llamadas a: ',
DESTINO_LLAMADA), COUNT (VALOR_PAGAR) AS
TOTAL, SUM (TIEMPO) AS TIEMPO, SUM (ICE + IVA) AS
IMPUESTOS, SUM (VALOR_NETO) AS VALOR FROM
LLAMADAS GROUP BY DESTINO_LLAMADA) TABLA;
```

PostgreSQL

```
SELECT * FROM (SELECT ('nro llamadas a: '||
DESTINO_LLAMADA ), COUNT(VALOR_PAGAR) AS
TOTAL, SUM(TO_NUMBER(TIEMPO,'999990.0000')) AS
TIEMPO, SUM(TO_NUMBER(ICE,'99990.0000') +
TO_NUMBER(IVA,'999990.0000')) AS IMPUESTOS,
SUM(TO_NUMBER(VALOR_NETO,'999990.0000'))AS VALOR
FROM PRUEBA.LLAMADAS GROUP BY
DESTINO_LLAMADA) TABLA;
```

Ver Test de Cálculo de Columna en Anexo 4.

5.5 Capacidad de subconsultas

La característica de subconsulta de SQL permite utilizar los resultados de una consulta como parte de otra, para la prueba de subconsultas se empleó una consulta en la cual se lista todos los números de teléfono, el destino de la llamada,

el tiempo y el tipo de llamada en donde el tipo de llamada sea “Nacional” y el tiempo sea mayor a 5 minutos.

En este test sólo se obtuvieron resultados por parte del SGBD PostgreSQL, debido a que MySQL no soporta subconsultas, por tal motivo no se pudo realizar las pruebas respectivas.

	PostgreSQL
CPU Utilizado	5%
Memoria Utilizada	189.051 MB
Tiempo	93 ms

La sentencia SQL utilizada fue:

```
PostgreSQL SELECT NRO_TELEFONO, DESTINO_LLAMADA, TIEMPO,
TIPO_LLAMADA FROM PRUEBA.LLAMADAS WHERE
TIPO_LLAMANA IN (SELECT TIPO_LLAMADA FROM
PRUEBA.LLAMADAS WHERE TIPO_LLAMADA =
'NACIONAL') AND TO_NUMBER (TIEMPO,'9990.000') > 5;
```

Ver Test de capacidad de subconsultas Anexo 5.

5.6 Combinación de resultados por medio de la sentencia union

El siguiente test fue desarrollado gracias a la característica UNION de la sentencia SELECT que permite esta capacidad y que fue probada a través de la siguiente consulta, en la cual se pidió que se mostrara el destino de las llamadas, el número total de llamadas realizadas, y el total recaudado por ese destino de llamadas y ordenado por el tipo de la llamada. La consulta indicada originó el resultado que se muestra a continuación:

	PostgreSQL	MySQL
CPU Utilizado	10%	50%
Memoria Utilizada	192.609 MB	172.949 MB
Tiempo	173 ms	560.1ms

La sintaxis SQL empleada como en los test anteriores varía en cuanto a pequeñas condiciones de sintaxis que maneja cada gestor de base de datos. Dichas sentencias SQL se mencionan a continuación.

MySQL SELECT * FROM (SELECT CONCAT ('nro de llamadas a: ', DESTINO_LLAMADA), TIPO_LLAMADA, COUNT (VALOR_PAGAR) AS TOTAL FROM LLAMADAS GROUP BY DESTINO_LLAMADA UNION SELECT CONCAT ('TOTALES DE ', DESTINO_LLAMADA), TIPO_LLAMADA, SUM (VALOR_PAGAR) AS TOTAL FROM LLAMADAS GROUP BY DESTINO_LLAMADA) RESULTADO ORDER BY TIPO_LLAMADA;

PostgreSQL SELECT * FROM (SELECT ('nro de llamadas a: ' || DESTINO_LLAMADA), TIPO_LLAMADA, COUNT (VALOR_PAGAR) AS TOTAL FROM PRUEBA.LLAMADAS GROUP BY DESTINO_LLAMADA, TIPO_LLAMADA UNION SELECT ('TOTALES DE ' || DESTINO_LLAMADA), TIPO_LLAMADA, SUM (TO_NUMBER (VALOR_PAGAR,'999990.0000')) AS TOTAL FROM PRUEBA.LLAMADAS GROUP BY DESTINO_LLAMADA, TIPO_LLAMADA) RESULTADO ORDER BY TIPO_LLAMADA;

Ver Test de combinación de resultados con UNION en Anexo 6.

5.7 Respaldo de datos (BACKUPS)

Para realizar los respaldos se puede emplear dos maneras, una es utilizando las opciones que nos brinda cada uno de los administradores gráficos de los SGBD, y la otra es mediante comandos de línea como son: `pg_dump [option...] [dbname]` para PostgreSQL y `mysqldump` para MySQL.

En este caso se ha decidido utilizar el siguiente software para realizar los tests referente a respaldo de datos o los tan conocidos como Backups.

MySQL	PostgreSQL
MySQL Query Browser versión 1.1.19	PgAdmin III Query versión 1.4.1

Los resultados obtenidos en este test fue mediante dos mediciones, uno al empezar y el otro al finalizar el proceso de los backups.

	MySQL		PostgreSQL	
	CPU Utilizado	Memoria Utilizada	CPU Utilizado	Memoria Utilizada
Inicio	22%	174.805 MB	2%	174.226 MB
Backups datos	79%	200144 MB	34%	194.035 MB

Ver Test de Backups de tablas Anexo 7.

5.8 Seguridad de password

Inicialmente el mecanismo utilizado por MySQL para cifrar las contraseñas en el momento de almacenarlas no ofrecía garantías de seguridad, ya que mediante fuerza bruta se podía obtener la contraseña en relativamente poco tiempo. El problema se encontraba en que las contraseñas, que se almacenan en una tabla utilizando `PASSWORD ()`, son cifradas de una forma en que facilitan su descifrado por fuerza bruta a gran velocidad. Esto podía ser utilizado por un atacante malévolo, con acceso a la base de datos que contiene las contraseñas, a identificar en periodos de tiempo relativamente cortos cualquier contraseña.

Debe indicarse, no obstante, que MySQL ofrecía otros mecanismos de cifrado de contraseña mucho más fuertes. Actualmente en las últimas versiones de MySQL se utiliza un sistema de privilegios y contraseñas que es muy flexible y seguro, y que permite verificación basada en el host. Las contraseñas son seguras porque todo el tráfico de contraseñas está encriptado cuando se conecta con un servidor.

PostgreSQL ofrece encriptación a varios niveles, y proporciona flexibilidad, protegiendo los datos de robos, de administradores poco escrupulosos, y de las redes inseguras. La encriptación de passwords se encuentra almacenado como MD5 hashes, así el administrador no puede descifrar el password actual asignado a un usuario. Si la encriptación MD5 es usada para autenticación del cliente, ésta nunca estará temporalmente presente en el servidor debido a que el cliente MD5 lo encripta antes de ser enviada por la red.

El método de autenticación MD5 es una doble-encriptación de la contraseña en el cliente antes de enviarlo al servidor. Primero MD5 encripta dicha contraseña basándose en el nombre del usuario, y luego encripta en base salt aleatoria enviada por el servidor cuando la conexión de la base de datos está hecha. Esto es el valor de la doble-encriptación que se envía encima de la red al servidor. La doble-encriptación no sólo previene la contraseña de descubrirse, sino también impide a otra conexión usar la misma contraseña encriptada para conectar al servidor de la base de datos en un momento más tarde.

5.9 Conclusiones

- Existen varias pruebas que se pueden realizar con el fin de probar las potencialidades tanto de MySQL como de PostgreSQL, por lo que es necesario escoger las más adecuadas y que puedan ocupar todas las características de éstos.
- La capacidad para subconsultas sólo se puede probar en PostgreSQL ya que MySQL no soporta las subconsultas. Esto hace que PostgreSQL tenga un nivel más alto en cuanto a consultas y subconsultas select se refiere.

- Las copias de seguridad se pueden realizar en los dos gestores de datos ya que éstos presentan comandos propios como son `pg_dump` y `mysqldump`.
- Ambos poseen métodos para la encriptación de contraseñas y para dar seguridad a los datos existentes dentro del Sistema de Gestión. En el caso de MySQL se trabaja bajo verificaciones basadas en el host y por parte de PostgreSQL se lo realiza a través del método MD5.

CAPITULO 6

CUADROS COMPARATIVOS

Antes de sumergirnos en las conclusiones y recomendaciones, es una buena idea presentar una perspectiva global de lo expresado y analizado hasta el momento. Este capítulo contiene un repaso sobre las características, mejoras y pruebas realizadas a los sistemas de gestión de bases de datos, los mismos que se encuentran detallados por medio de cuadros comparativos y gráficos estadísticos.

6.1 Recopilación de resultados de los tests

Migración de datos desde Access

MySQL	CPU Utilizado	Memoria Utilizada MB.	PostgreSQL	CPU Utilizado	Memoria Utilizada MB.
Iniciación	2%	177.172	Sesiones	2%	246.41
Conexión a Servidores	25%	213.558	Fuente	2%	247.027
Ingeniería Inversa	73%	227.004	Destino	2%	251.348
Creación de Objetos	11%	224.707	Configuración	2%	252.328
Transferencia datos BD	79%	240.769	Ejecución	91%	263.305

Migración de datos desde un archivo de texto

	MySQL		PostgreSQL	
	CPU Utilizado	Memoria Utilizada	CPU Utilizado	Memoria Utilizada
Inicio	4%	212.808 MB.	4%	190.293 MB.
Transmisión de datos	22%	196.391 MB.	22%	209.164 MB.

Velocidad en búsqueda y ordenamiento de datos

	PostgreSQL	MySQL
CPU Utilizado	19%	17%
Memoria Utilizada	277.785 MB	200.926 MB
Tiempo	781 ms	250.6 ms

	PostgreSQL	MySQL
CPU Utilizado	8%	27%
Memoria Utilizada	210.148 MB	200.844 MB
Tiempo	110 ms	356.2 ms

	PostgreSQL	MySQL
CPU Utilizado	4%	7%
Memoria Utilizada	236.531 MB	196.332 MB
Tiempo	32 ms	53 ms

Velocidad en procedimiento de cálculo de columna

	PostgreSQL	MySQL
CPU Utilizado	11%	5%
Memoria Utilizada	213.578 MB	231.824 MB
Tiempo	547 ms	558.7ms

Capacidad de subconsultas

	PostgreSQL
CPU Utilizado	5%
Memoria Utilizada	189.051 MB
Tiempo	93 ms

Combinación de resultados por medio de la sentencia union

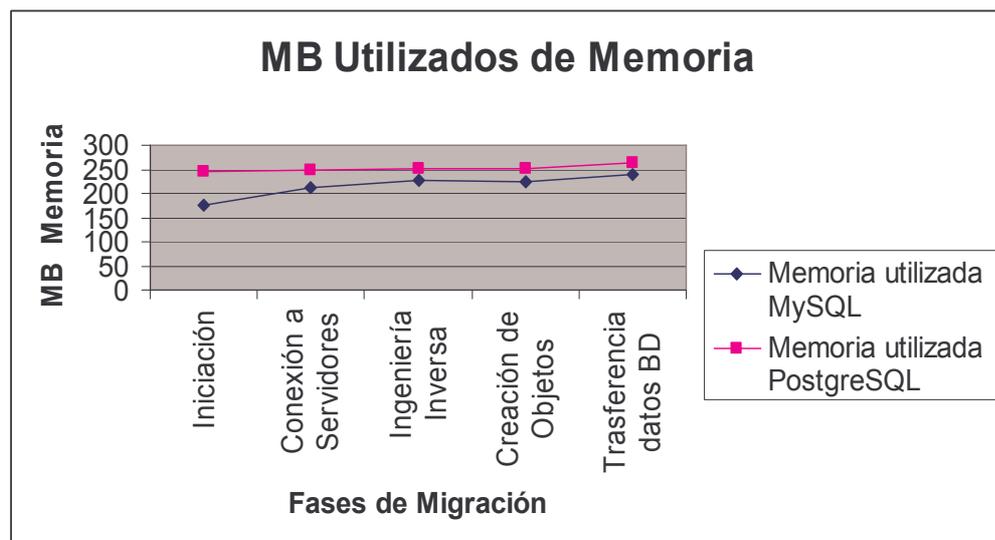
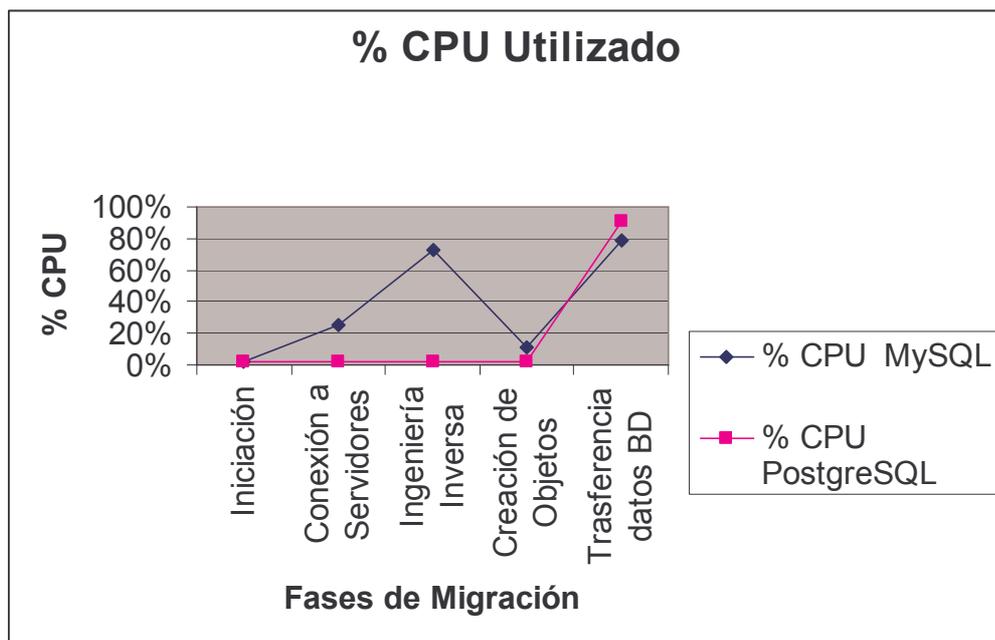
	PostgreSQL	MySQL
CPU Utilizado	10%	50%
Memoria Utilizada	192.609 MB	172.949 MB
Tiempo	173 ms	560.1ms

Respaldo de datos (BACKUPS)

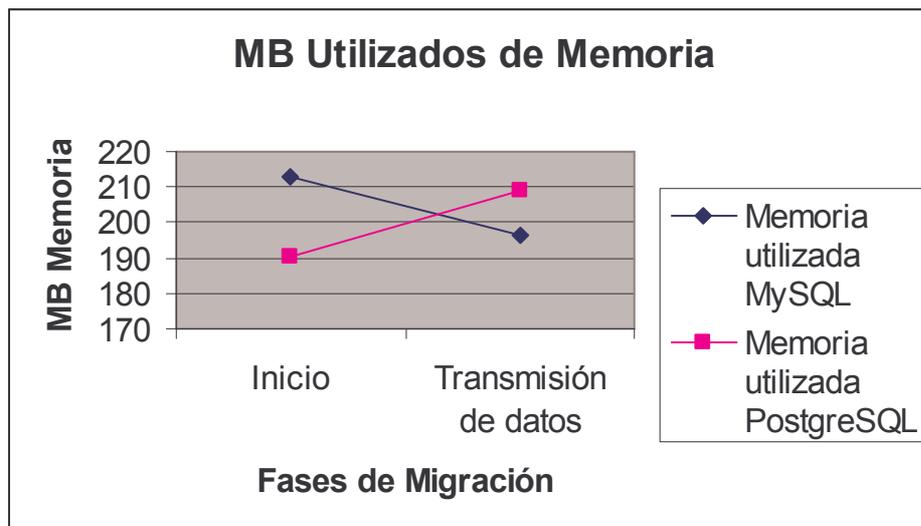
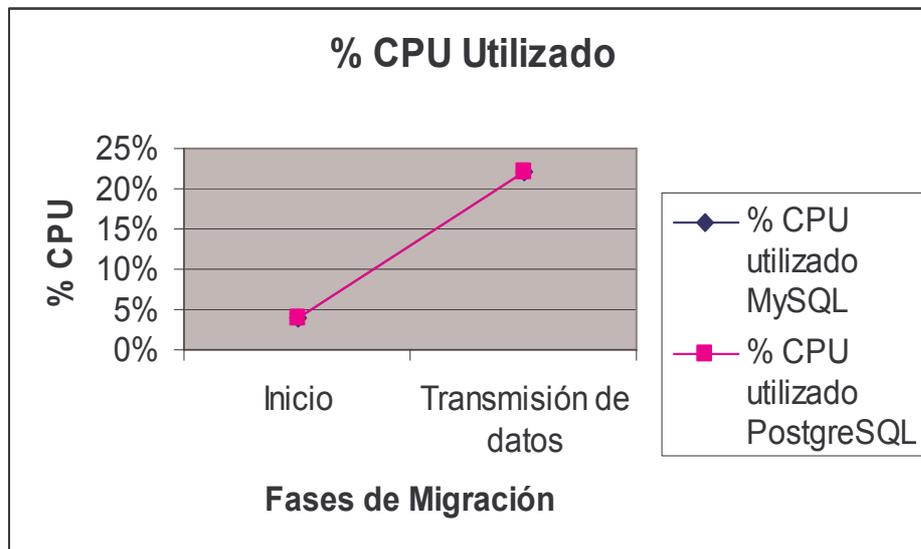
	MySQL		PostgreSQL	
	CPU Utilizado	Memoria Utilizada	CPU Utilizado	Memoria Utilizada
Inicio	22%	174.805 MB	2%	174.226 MB
Backups datos	79%	200144 MB	34%	194.035 MB

6.2 Realización de cuadros estadísticos

Migración de datos desde Access.- En esta prueba se observa claramente como PostgreSQL consume muchos más recursos tanto de CPU como de Memoria; pero más claramente de memoria, ya que en algunas fases MySQL consume más recursos de CPU. En la última fase de migración en la que hace referencia a la “Transmisión de Datos” PostgreSQL ocupa más recursos que MySQL, como se presenta en los siguientes cuadros este último gestor ocupa 91% de CPU y 263.305 MB de memoria.

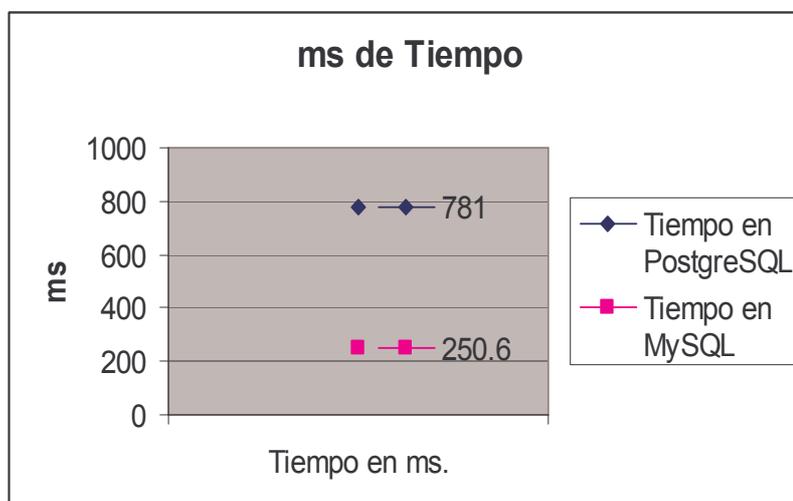
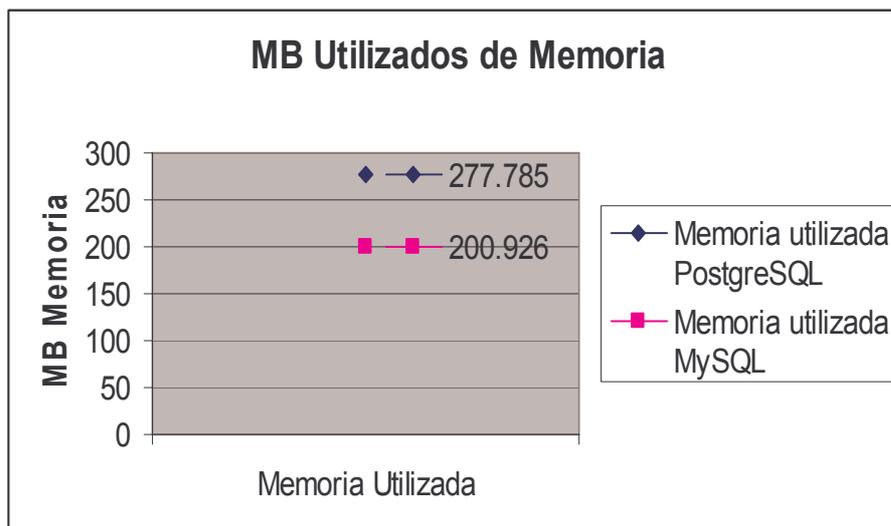
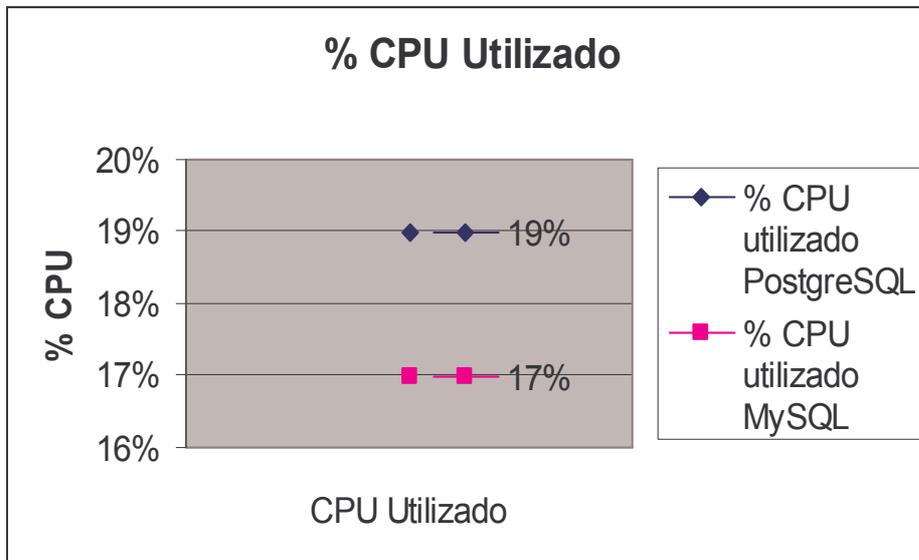


Migración de datos desde un archivo de texto.- Para este caso los dos sistemas de gestión de datos ocuparon el mismo porcentaje de CPU al momento de realizar la migración del archivo plano hacia cada uno de ellos, en cuanto a la memoria, en primera instancia MySQL superó a PostgreSQL, pero en la parte más crítica la de la transmisión de datos PostgreSQL ocupó un 6% más de memoria que MySQL. Los resultados son plasmados en los siguientes cuadros.

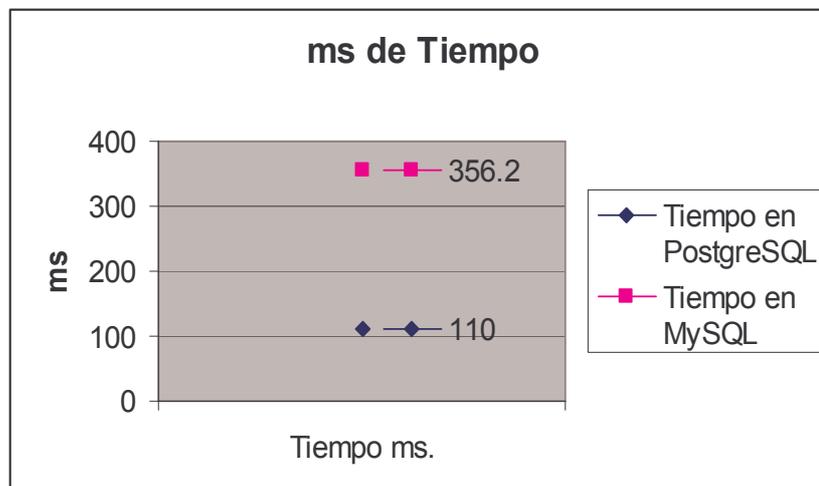
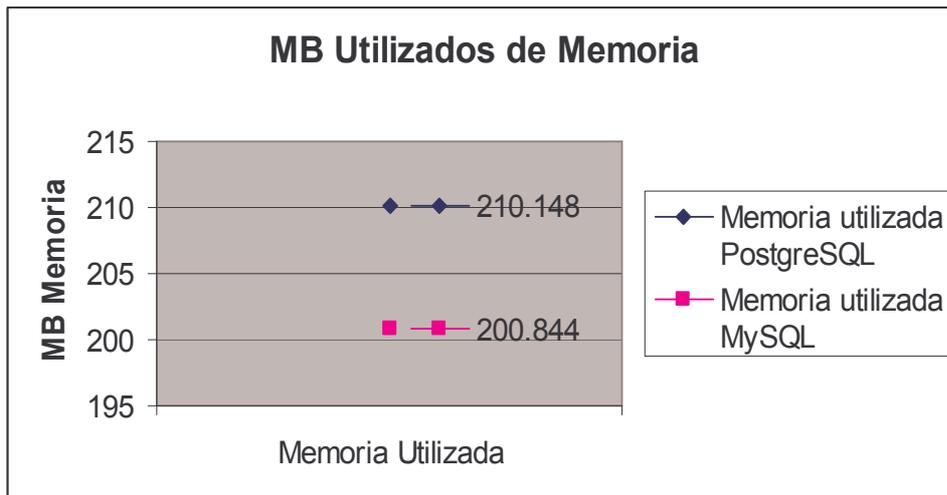
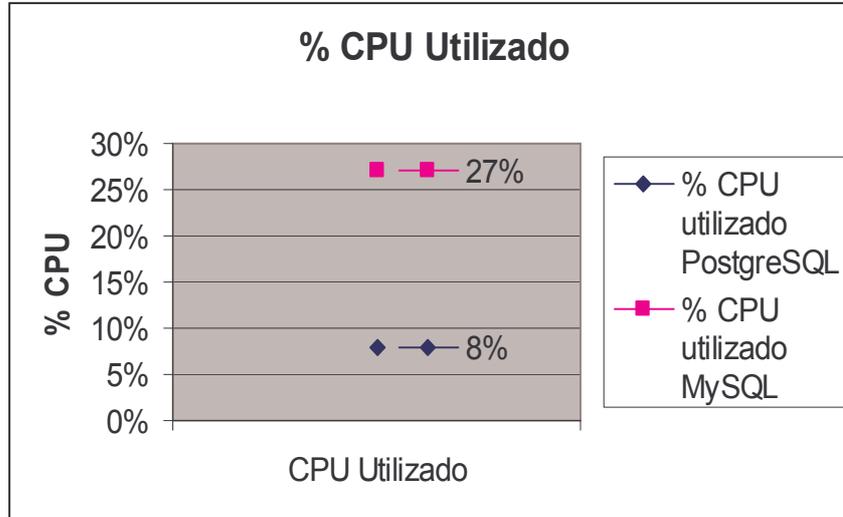


Velocidad en búsqueda y ordenamiento de datos.- En la velocidad de búsqueda y ordenamiento de datos PostgreSQL resultó ser la más rápida, a pesar de que ocupa más recursos en memoria. En esta prueba se utilizaron tres sentencias SQL en las que se demostró como se indicó anteriormente la capacidad de velocidad que posee PostgreSQL. Para mayor agudeza los cuadros que a continuación se muestran contienen los diferentes resultados que produjeron las tres sentencias.

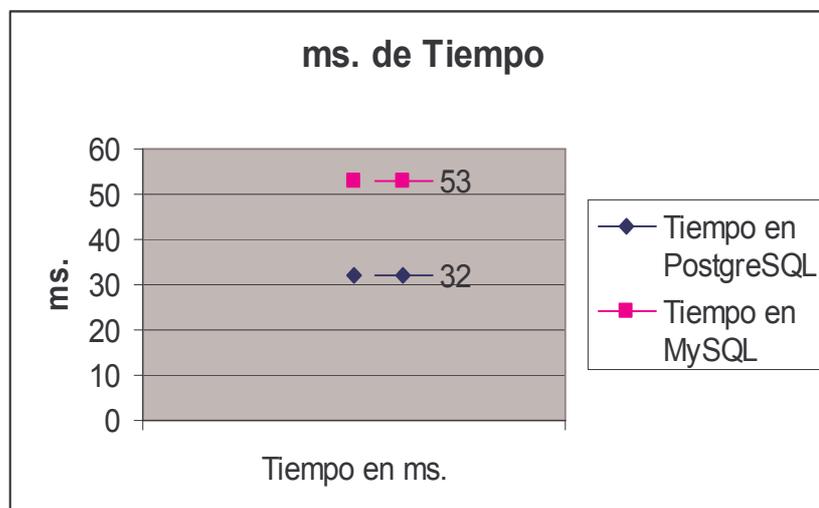
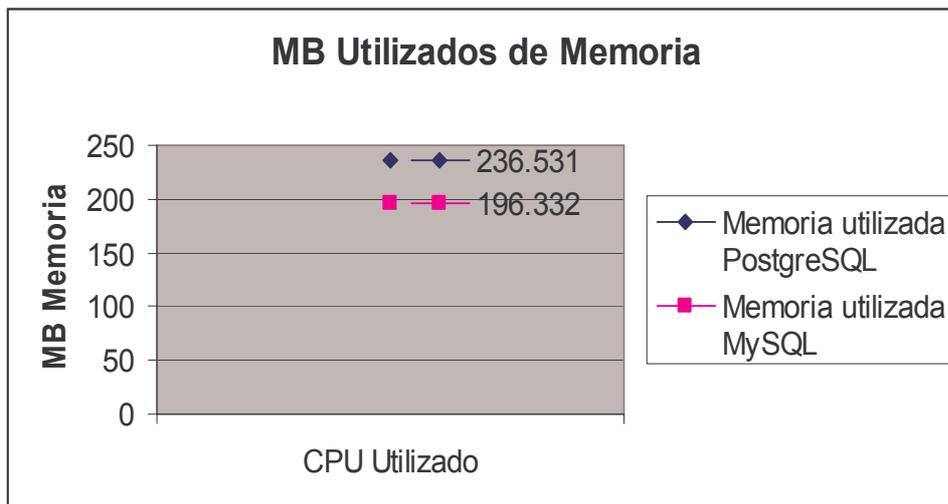
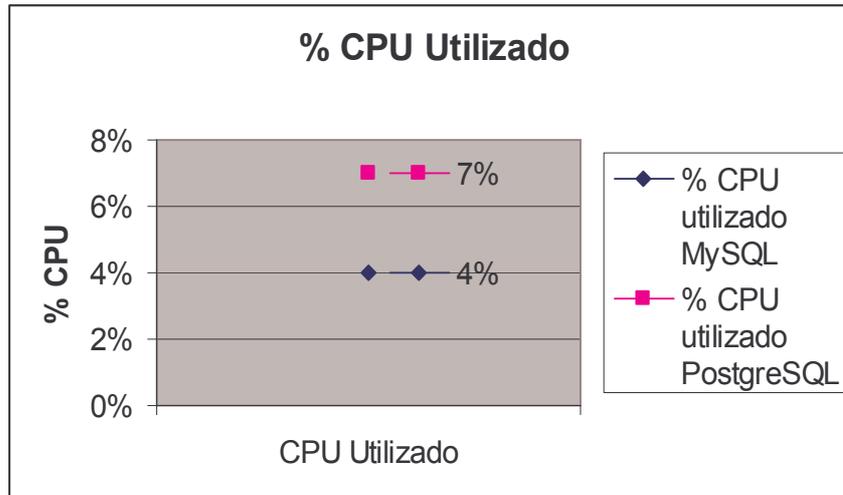
a)



b)

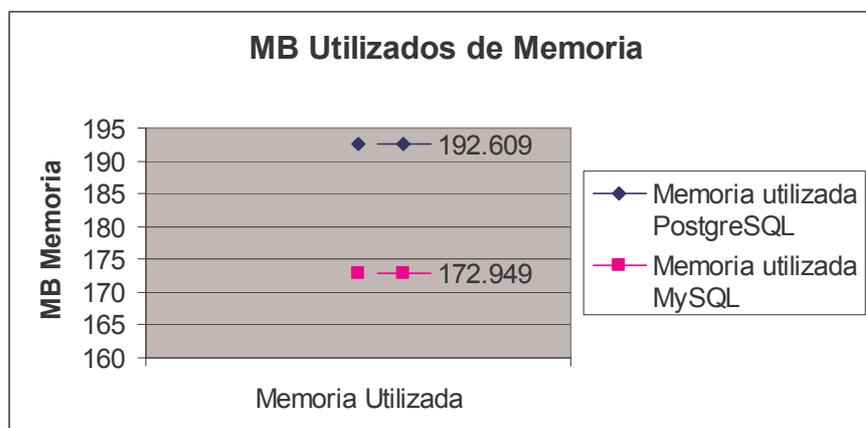
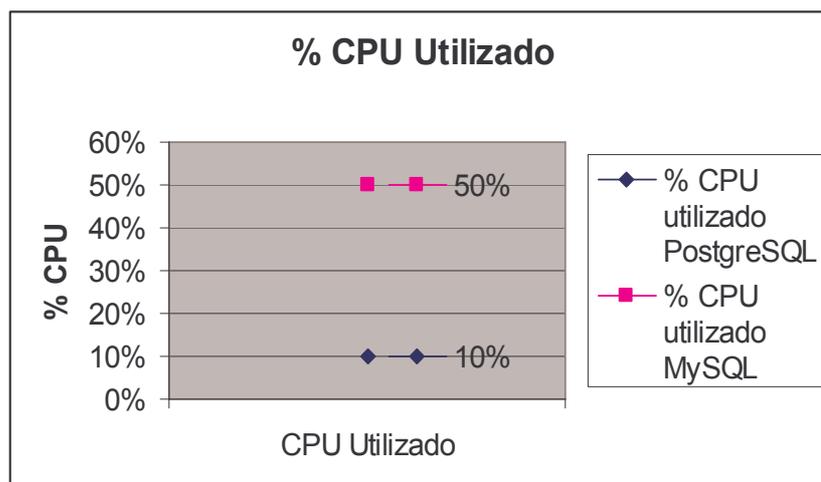


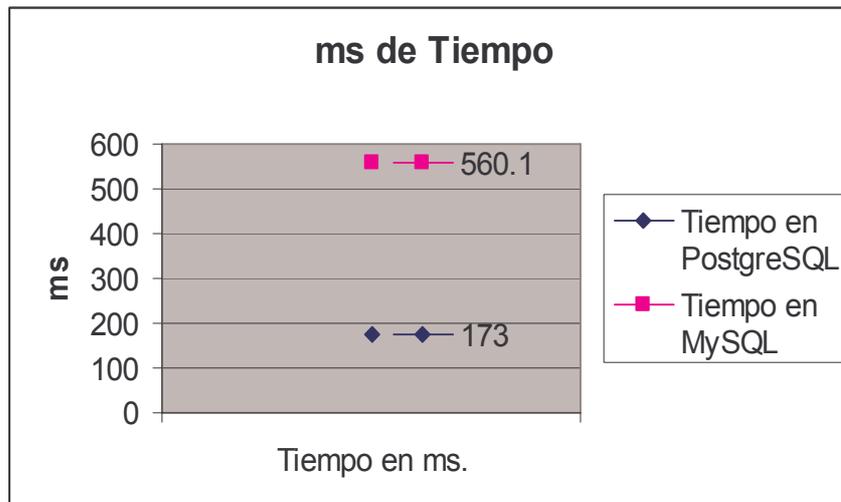
c)



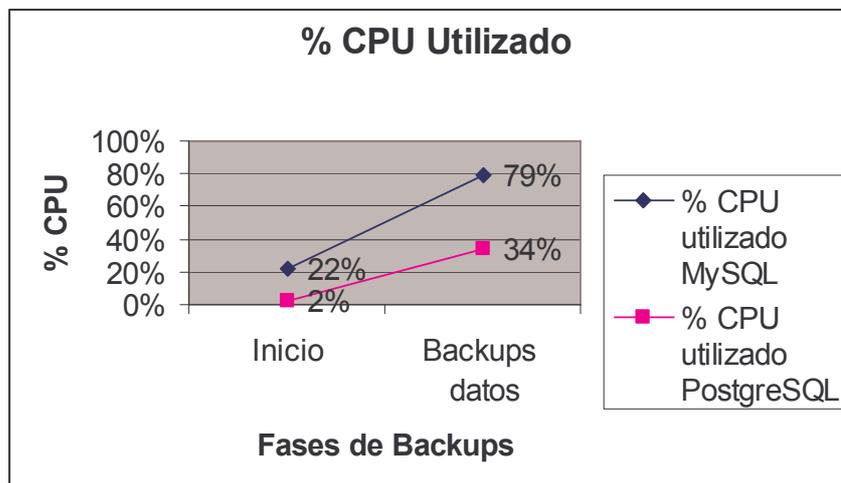
Capacidad de subconsultas.- Para este test no se realizó un cuadro estadístico debido a que MySQL no soporta subconsultas y PostgreSQL no tiene contra quien compararse, pero se puede observar sus resultados como se muestran en el capítulo anterior en donde ocupa un 5% de CPU, 189.051 MB de memoria y le tomó generar la consulta en 93 segundos.

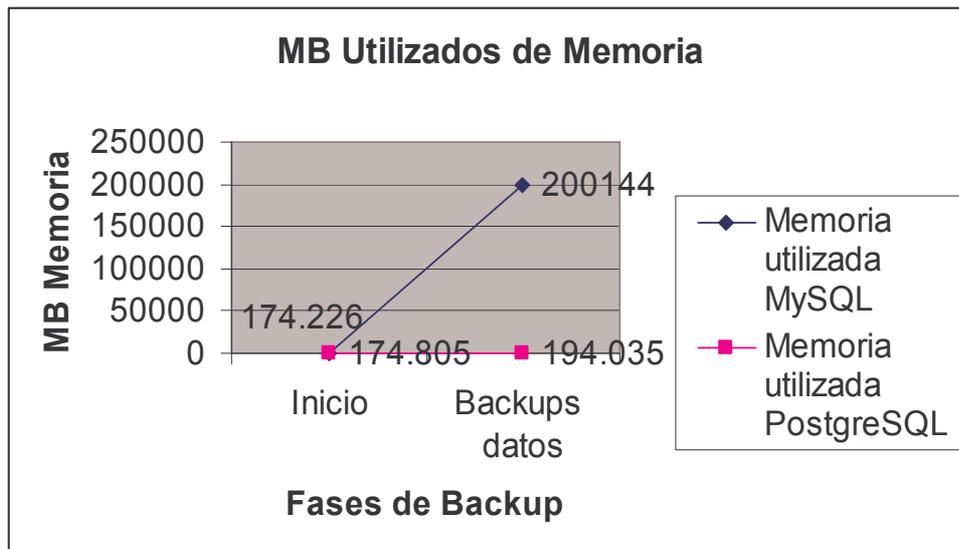
Combinación de resultados por medio de la sentencia “UNION”.- A igual que en los cuadros estadísticos anteriores PostgreSQL ocupa más cantidad de memoria aunque el porcentaje de CPU utilizado sea menor. En los siguientes cuadros se puede apreciar cuanto han ocupado cada gestor del CPU, cuanto en memoria y además el tiempo que les tomó realizar la consulta con la sentencia “UNION”.





Respaldo de datos (BACKUPS).- Tanto en MySQL como en PostgreSQL se realizaron las copias de seguridad de las bases de datos demostrando que MySQL en respaldo de datos consume muchos más recursos tanto de CPU como de memoria que PostgreSQL. Los resultados obtenidos se presentan en los cuadros siguientes.





6.3 Ventajas y Desventajas

A continuación se muestran las ventajas y desventajas que poseen tanto MySQL como PostgreSQL según los resultados obtenidos de las características que poseen, de los requerimientos de instalación y de los resultados de tests obtenidos para poder tener un poco más claro las funcionalidades que ofrecen cada una de ellas, así como sus pros y sus contras.

6.3.1 Características

MySQL

Ventajas

1. Sin lugar a duda, lo mejor de MySQL es su velocidad a la hora de realizar las operaciones, lo que le hace uno de los gestores que ofrecen mayor rendimiento.
2. Su bajo consumo lo hacen apto para ser ejecutado en una máquina con escasos recursos sin ningún problema.
3. 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.

5. 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).

Desventajas

1. Carece de soporte para transacciones, rollback's y subconsultas.
2. El hecho de que no maneje la integridad referencial, hace de este gestor una solución pobre para muchos campos de aplicación, sobre todo para aquellos programadores que provienen de otros gestores que sí que poseen esta característica.
3. No es viable para su uso con grandes bases de datos, a las que se acceda continuamente, ya que no implementa una buena escalabilidad.

PostgreSQL

Ventajas

1. Posee una gran escalabilidad. 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).
2. 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.
3. 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.

Desventajas

1. Consume gran cantidad de recursos.
2. Tiene un límite de 8K por fila, aunque se puede aumentar a 32K, con una disminución considerable del rendimiento.

6.3.2 Requerimientos

En cuanto a los requerimientos que cada gestor exige, el que mayor ventaja tiene es PostgreSQL, ya que ambos operan sobre Windows NT, 2000 o XP pero MySQL requiere un mínimo de 200 MBytes mientras que PostgreSQL requiere solamente 70 MBytes.

6.3.3 Instalaciones

En lo que se refiere en cuanto a instalaciones podemos mencionar los siguientes puntos:

MySQL

Ventajas:

1. Sus instaladores y manuales son fáciles de conseguir se encuentran disponibles en la página oficial de MySQL.
2. El instalador presenta un wizard lo cual facilita, su instalación ya que le permite seleccionar el tipo de servidor a implementarse y opciones como número de usuarios entre otras.

Desventaja:

1. Los paquetes deben ser instalados de manera separada, es decir, un paquete para poder trabajar con MySQL 5.0.18-win32 y otro que es el MySQL administrator que permite administrar la base de datos de una manera gráfica.

6.3.4 Resultados de test

MySQL

Ventajas

1. El consumo de la memoria es mínimo.
2. Mayor velocidad para presentar resultados con pocos datos.
3. Posee una sintaxis flexible.

Desventajas

1. Carece de soporte para subconsultas.
2. Disminuye su velocidad cuando la cantidad de datos es extensa.

PostgreSQL

Ventajas

1. Implementa el uso de subconsultas haciendo el funcionamiento mucho más eficaz.
2. Mayor velocidad en presentar resultados con grandes bases de datos, es decir, mantiene una velocidad similar a la de MySQL.

Desventajas

1. Consume muchos más recursos y carga más el sistema.
2. Sintaxis estricta.

6.4 Conclusiones

- Según los resultados obtenidos de los tests, PostgreSQL consume mayor cantidad de recursos especialmente de memoria, la misma que fue medida en MBytes, y para bases de datos grandes su velocidad es mejor que la de MySQL.
- MySQL mejora su velocidad cuando trabaja con poca cantidad de datos y consume pocos recursos tanto de CPU como de Memoria. Su sintaxis es flexible en relación a la de PostgreSQL que trata de parecerse a las grandes bases de datos existentes en el mercado como es Oracle.

CAPITULO 7

CONCLUSIONES Y RECOMENDACIONES

Existen varios artículos sobre los benchmarks que hacen referencia sobre cual es el mejor Sistema de Gestión de Bases de Datos, lo que se ha llevado acabo en este documento es una comparación en cuanto a características y rendimientos mediante tests, tanto de MySQL al igual que PostgreSQL, los mismos que fueron realizados en lo posible en igualdad de condiciones cosa que por otra parte, muchas veces resulta complicado debido a las distintas implementaciones que poseen estos gestores.

7.1 Conclusiones Teóricas

Las conclusiones obtenidas mediante la investigación sobre las características y funcionalidad de estos SGBDs son:

- Son sistemas de gestión de datos, cuyas licencias son libres MySQL (GPL) y PostgreSQL (BSD).
- Tanto la historia de MySQL en 1995, como de PostgreSQL en 1996 indican que ambos SGBD surgen debido a la necesidad que hubo de manejar gran cantidad de datos relacionados y a la motivación de los desarrollares de mejorar la calidad del software mediante la corrección de errores y la implementación de mejoras con lo cual se da paso a los diferentes versiones.
- Las últimas versiones de los SGBD son la 8.1 para PostgreSQL y para MySQL la 5.1 (versión alfa), sobre plataforma Windows o Linux.
- Los instaladores tanto para MySQL como PostgreSQL se encuentran disponibles en las páginas oficiales en Internet, en las que puede conseguir documentación y herramientas para el manejo de cada una de ellas.
- Instalar estos dos SGBD es una tarea simple de realizar; por la comodidad que cada uno posee su wizard para el momento de la instalación, pero no por ser tan fácil se debe dejar de tomar en cuenta ciertas opciones de

configuración que van surgiendo a lo largo del proceso, opciones que pueden variar de acuerdo a las aplicaciones con las cuales se vaya a trabajar.

- Además estos poderosos sistemas gestores de bases de datos como lo son MySQL y PostgreSQL poseen varias herramientas adicionales las cuales permiten realizar diferentes tareas, como son las migraciones, consultas y la administración de la bases de datos de manera sencilla y mediante un ambiente gráfico.

7.2 Conclusiones Metodológicas

La metodología que se utilizó fue a través de benchmarks entre los sistemas gestores de bases de datos, además se trató de que ambos trabajen bajo las mismas condiciones para obtener resultados reales.

Los tests trabajaron con los siguientes datos:

- La base del Locutorio de Etapa nos proporciona un aproximado de 9000 datos distribuidos en 21 tablas, lo cual nos permite realizar las prácticas necesarias en nuestros SGBD.
- Una de los pasos que fue necesario antes de realizar la migración, fue una previa preparación de los datos que se encontraban almacenados en una base de datos perteneciente a Microsoft Access, existen diferentes métodos tanto directos como indirectos, para realizar los tests del presente estudio se decidió trabajar con métodos indirectos ya que se utilizarán herramientas propias de los SGBD como son el Migration Toolkit y Access2PostgreSQL, entre otras.

De los tests realizados se obtuvo el siguiente cuadro, en el cual se puede observar el comportamiento de cada uno de los gestores de bases de datos, frente a los tests propuestos con lo que se obtuvieron los siguientes resultados:

Test	MySQL	PostgreSQL
Migrar Access	✓ Migration Toolkits	✓ Access2postgresql Pro
Migrar Archivo Plano	✓ MySQL Query Browser	✓ PgAdmin III Query
Búsqueda y Ordenamiento de datos	✓ 250.6 ms	✓ 781 ms
Cálculo de columna	✓ 558.7 ms	✓ 547 ms
Subconsultas	X No	✓ 93 ms
Sentencia UNION	✓ 560.1 ms	✓ 173 ms
Backups	✓ MySQL Administrator	✓ PgAdmin III
Seguridad de Password	✓ Basadas en Host	✓ Método MD5 Hash

En cuanto a lo que se refiere a los recursos ocupados por parte del computador se utilizo la herramienta benchmark CS FIRE Monitor versión 2.5.0, con lo que se obtuvo los siguientes resultados:

Recursos	MySQL 5.0	PostgreSQL 8.1
CPU %	50 %	50 %
Memoria(MB)	Menor	Mayor
Velocidad	Lento	Rápido
Instaladores	MySQL 5.0.18	PostgreSQL 8.1
www	www.mysql.com	www.postgresql.org
Sistema Operativo	Linux, Windows, FreeBSD, MacOS X, Solaris, HP UX, AIX, and other	Windows XP, más de 2 docenas de sistemas operativos Unix-como (Linux, all BSDs, HP-UX, AIX, OS X, Unixware, Netware...)
Lenguajes	Php, cgi, perl, Eiffel, Java,	PL/pgSQL, PL/Tcl, PL/Perl, PL/Python PL/PHP, PL/Java
Sintaxis	Flexible	Estricta
Puerto	3306	5432

7.3 Conclusiones Pragmáticas

El presente estudio tenía como finalidad, dar a conocer muchas de las ventajas y desventajas existentes entre estos dos Sistemas Gestores de Bases de Datos, es posible tener un sin número de aplicaciones dentro de los mercados. Una aplicación existente y muy interesante en el mercado es la implementada por el gobierno de Argentina la cual está construida íntegramente sobre herramientas de código libre: MapServer, PostgreSQL, PostGIS y Maplab. Dicho mapa esta disponible en la siguiente dirección:
<http://mapa.buenosaires.gov.ar/sig/index.phtml>

7.4 Recomendaciones

Para concluir lo único que podemos acotar es que siendo PostgreSQL un nuevo gestor de bases de datos y el menos conocido, el desempeño de PostgreSQL es comparable con el de otras bases de datos comerciales y de código abierto. Es más rápida para algunas cosas, más lenta para otras. El desempeño es usualmente +/-10% comparado con otras bases de datos.

Luego de haber indicado varias ventajas y desventajas mediante los diferentes cuadros comparativos, tanto de MySQL como de PostgreSQL, y siendo ambos magníficos gestores de bases de datos, solo queda la decisión por parte de la persona encargada de decidir cual es indicado para manejar sus datos.

Si desea aprender el funcionamiento de las bases de datos es aconsejable empezar por aprender MySQL por tener una sintaxis más flexible, y trabajar con lenguajes utilizados en nuestro medio. Pero si lo que busca es un poderoso gestor de bases de bases de datos que probablemente le iguala a Oracle es aconsejable PostgreSQL.

Referencias

<http://www.pervasive-postgres.com/postgresql/mysql.asp>

Fecha: 20 de diciembre de 2005

PAGINA OFICIAL DE POSTGRESQL

<http://www.postgresql.org/>

Fecha: 19 de diciembre de 2005

<http://www.sindominio.net/ayuda/postgresql/tutorial-postgresql-1.html>

Autor: Manuel Soriano

Fecha: 18 de diciembre de 2005

PAGINA OFICIAL DE MYSQL

<http://www.mysql.com/>

Autor: MySQL AB. All rights reserved

Fecha: 19 de diciembre de 2005

ARTICULOS TECNICOS

<http://techdocs.PostgreSQL.org/>

Fecha: 15 de Enero del 2006

COMUNIDAD DE DESARROLLADORES

<http://www.postgresql.org/files/documentation>

Fecha: 28 de noviembre del 2005

TUTORIALES

<http://www.intermedia.net/support/sql/sqltut.shtm>

http://ourworld.compuserve.com/homepages/graeme_birchall/HTM_COOK.HTM

<http://sqlcourse.com>.

Fecha: Enero del 2006

SANCHEZ, Jorge , Guía rápida de MySQL,

www.jorgesanchez.net

Fecha: 2004

INTRODUCCION A MYSQL

www.mysql-hispano.org

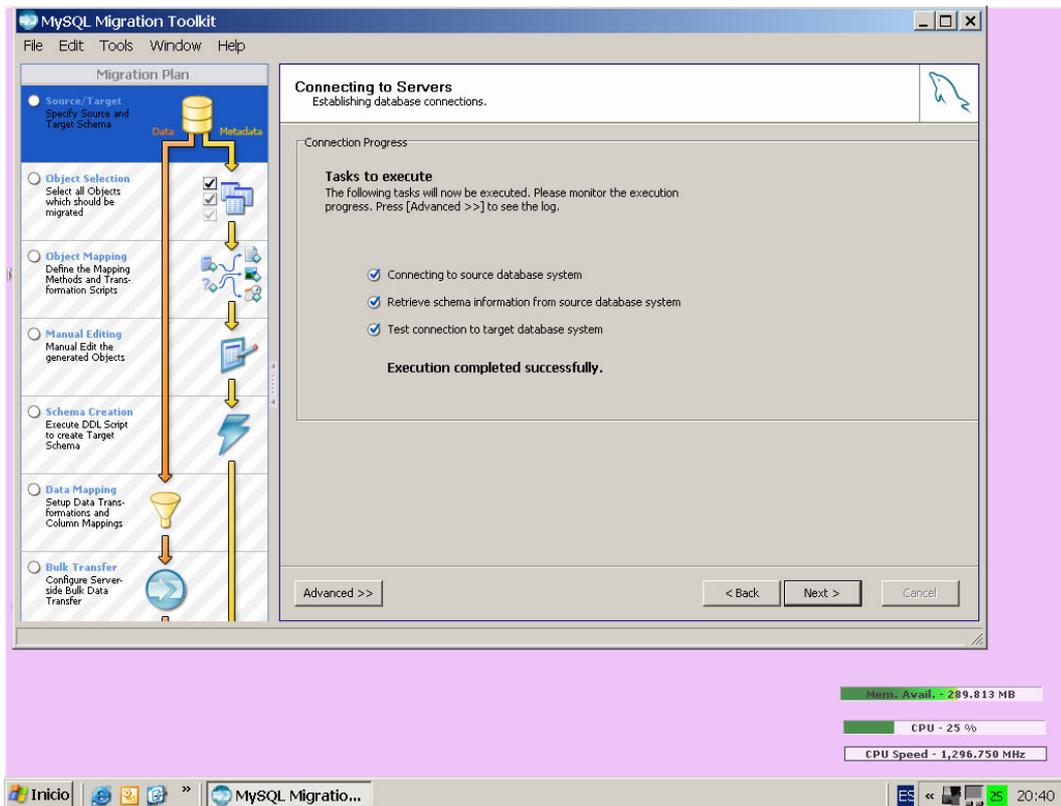
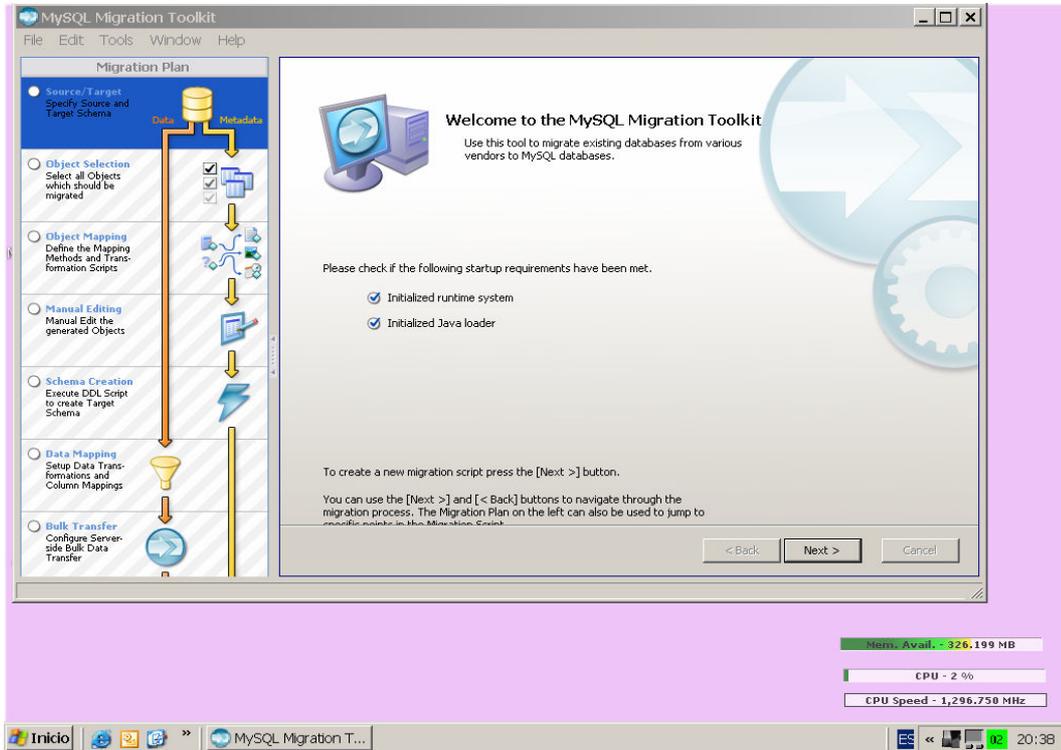
Fecha: 29 de Mayo del 2003

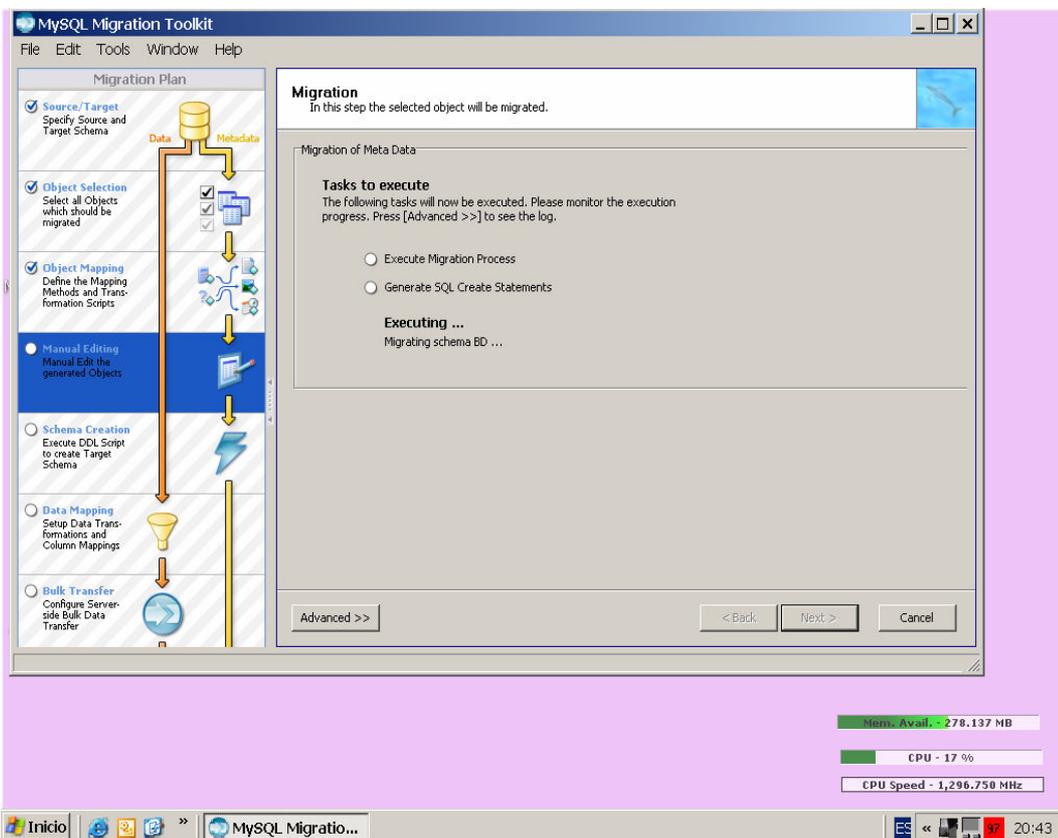
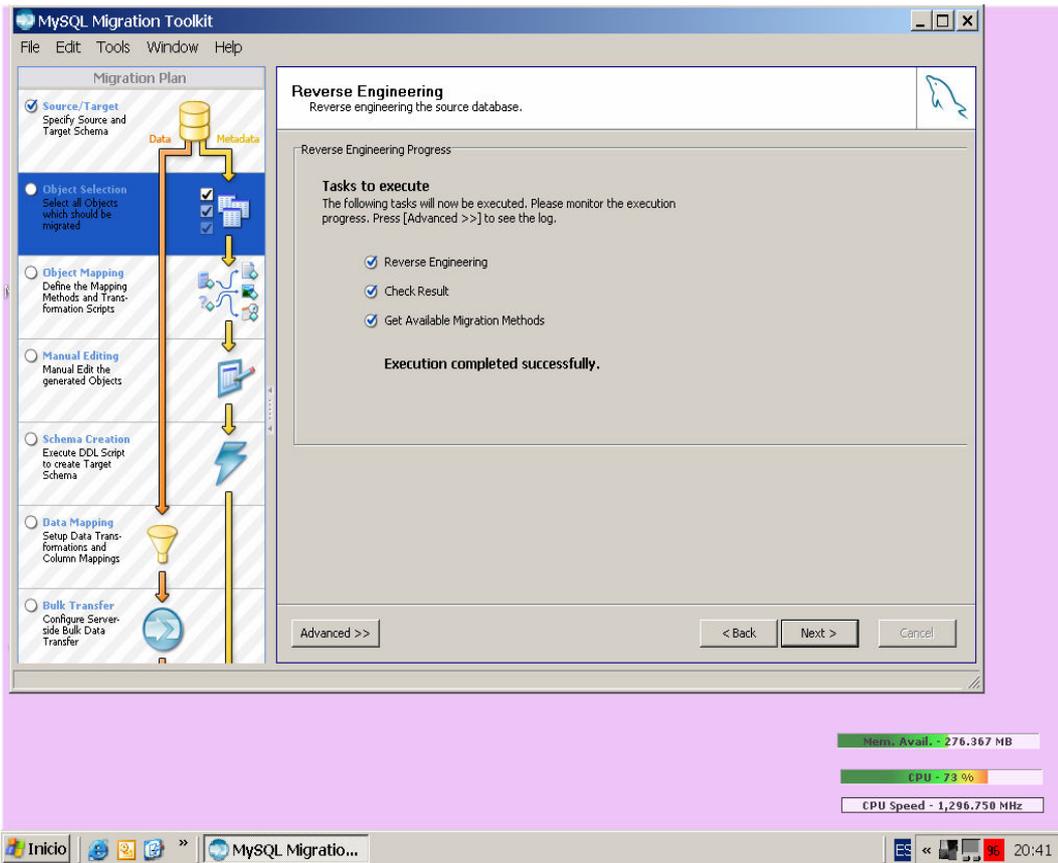
ANEXOS

Los anexos que se incluyen en este documento son los siguientes:

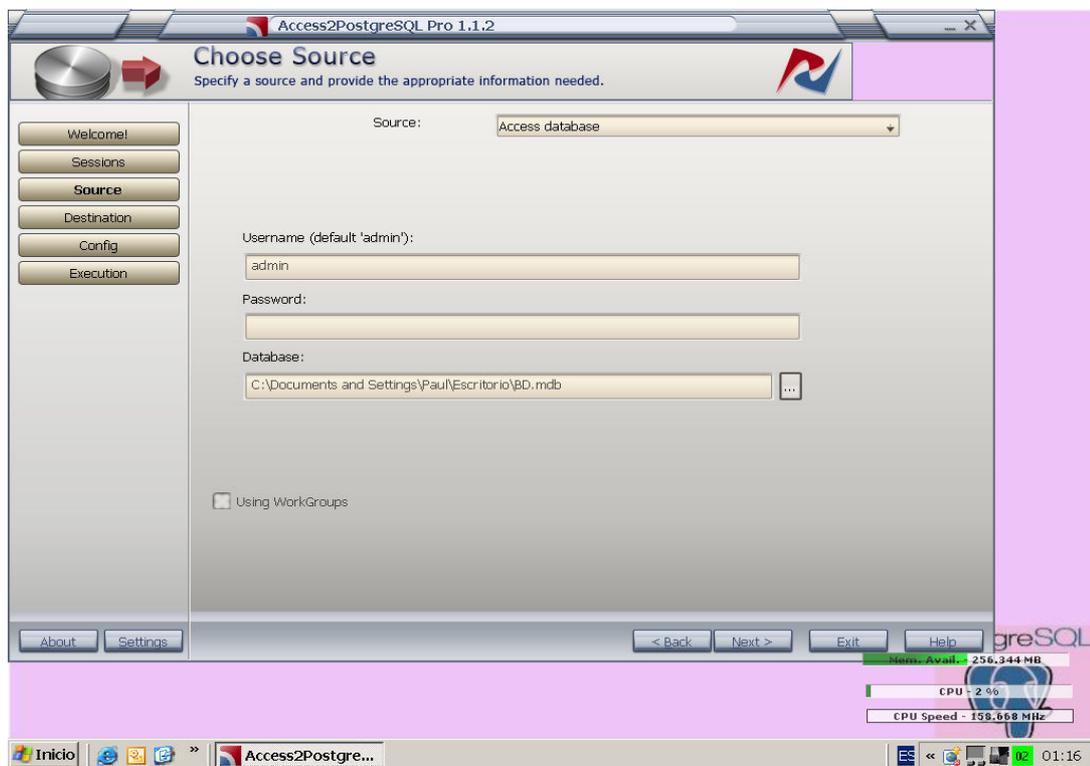
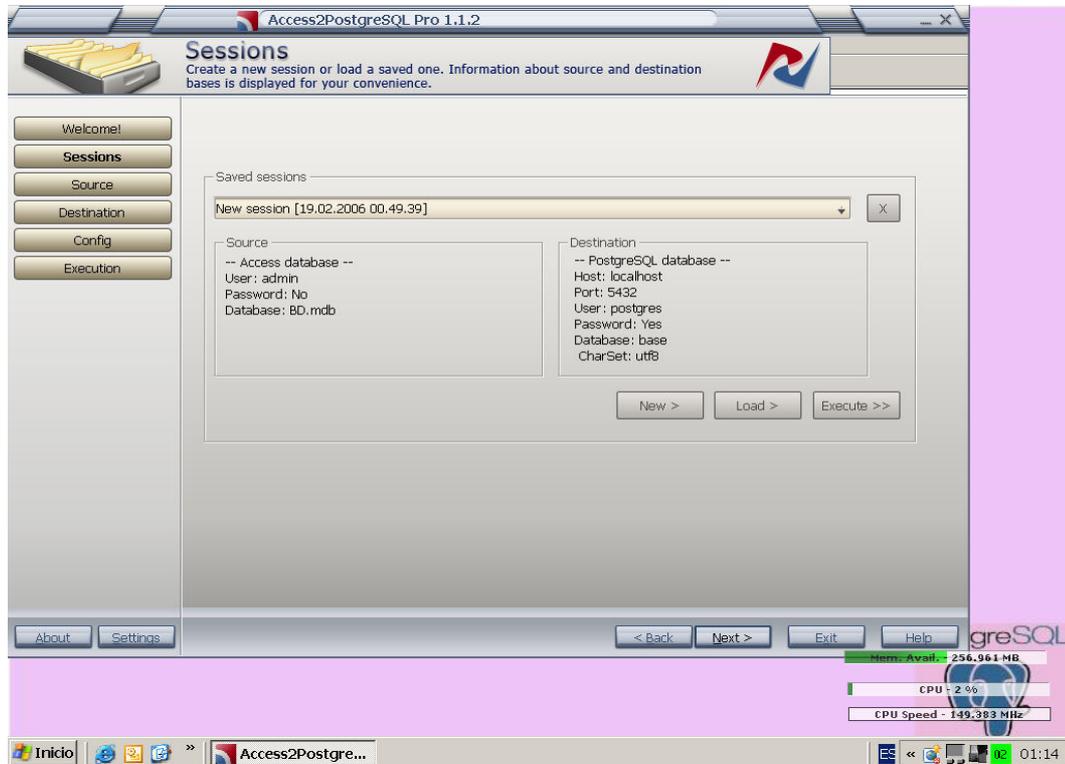
- Anexo 1: Migración de datos desde Access a MySQL y a PostgreSQL.
- Anexo 2: Migración desde un archivo plano hacia MySQL y PostgreSQL.
- Anexo 3: Velocidad en búsqueda y ordenamiento de datos.
- Anexo 4: Velocidad en procedimiento de cálculo de columna.
- Anexo 5: Capacidad para subconsultas.
- Anexo 6: Capacidad para combinación de resultados (sentencia UNION).
- Anexo 7: Respaldo de datos (BACKUPS).

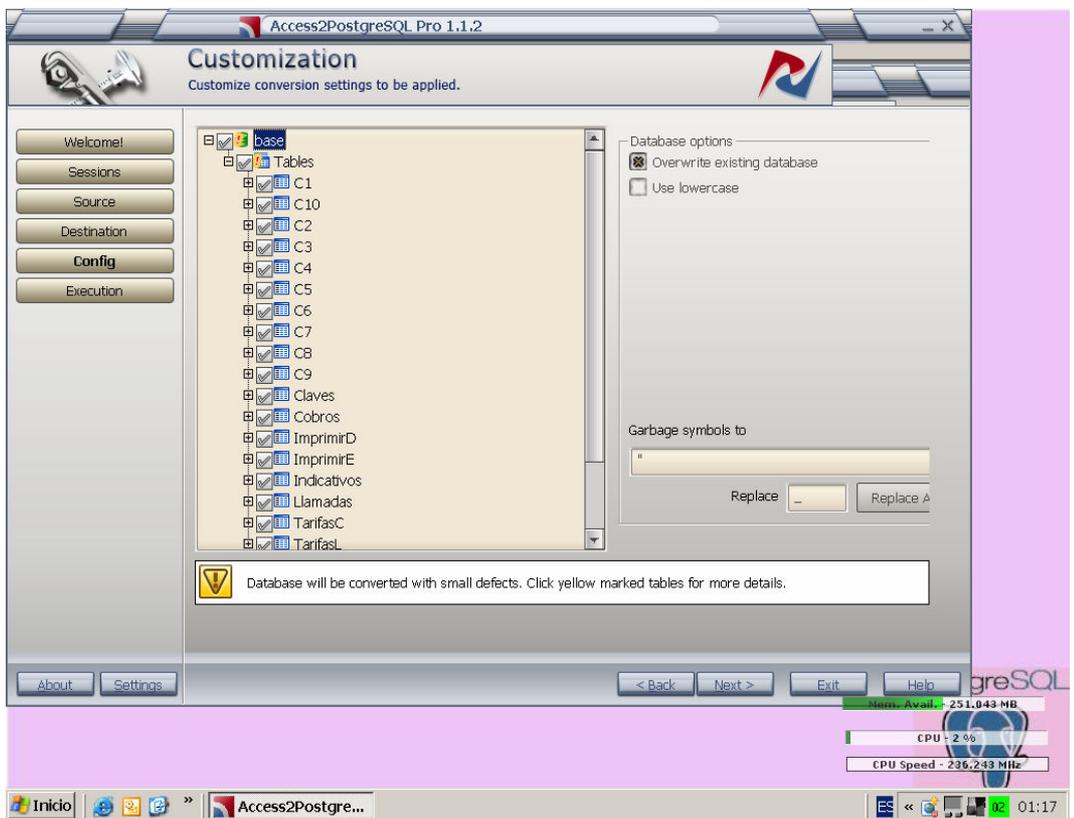
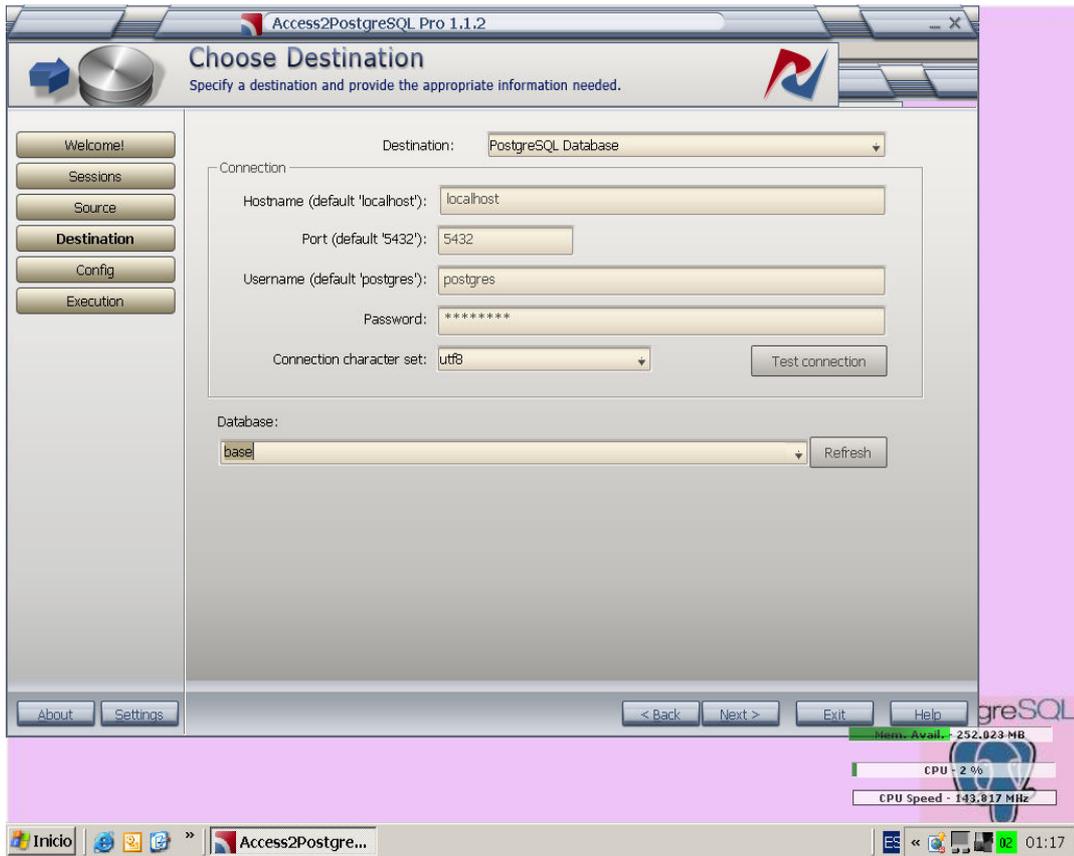
Anexo 1: Migración de datos desde Access a MySQL mediante la herramienta Migration Toolkit.





Migración desde Access a PostgreSQL utilizando la herramienta Access2PostgreSQL Pro.





Access2PostgreSQL Pro 1.1.2

Execution

At last you can save the current session configuration. Click COMMIT to execute the copying process.

Welcome!
Sessions
Source
Destination
Config
Execution

Total progress:
Current operation:

Operation log:
Creating table 'Cobros'.
Copying data for table 'Cobros'.
Creating table 'ImprimirD'.
Copying data for table 'ImprimirD'.
Creating table 'ImprimirE'.
Copying data for table 'ImprimirE'.
Creating table 'Indicativos'.
Copying data for table 'Indicativos'.
Creating table 'Llamadas'.
Copying data for table 'Llamadas'.

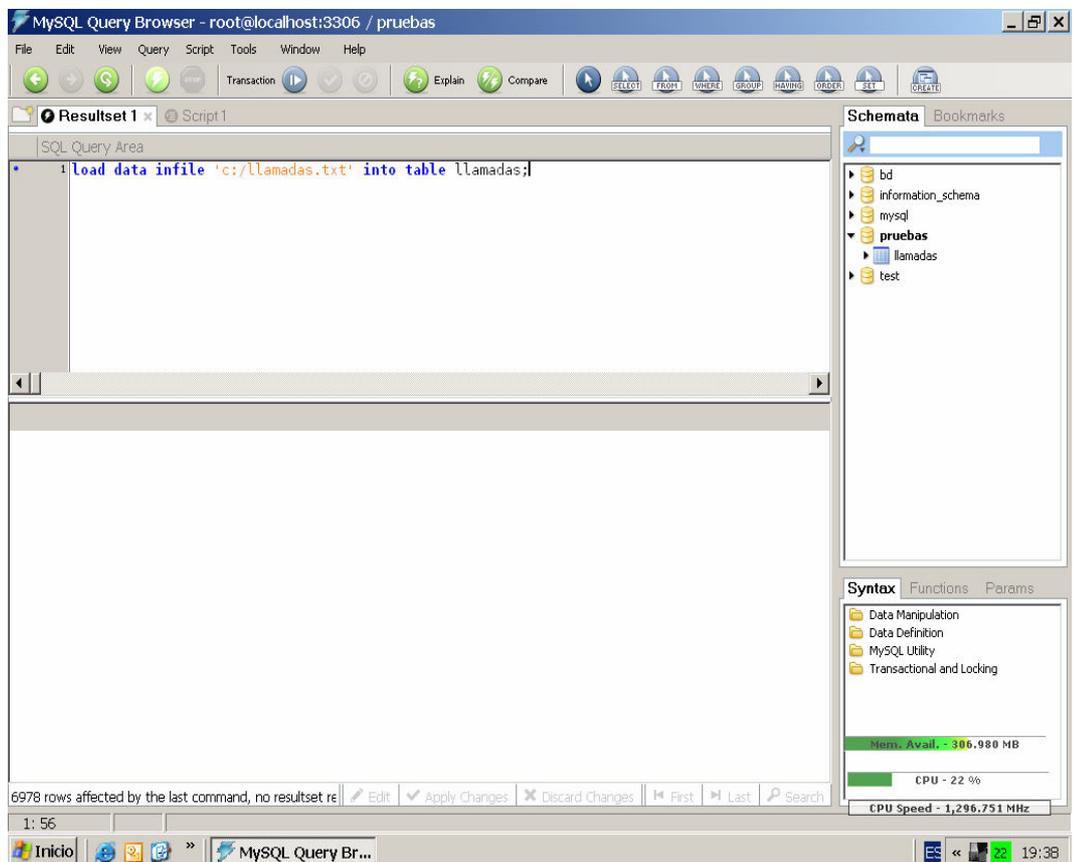
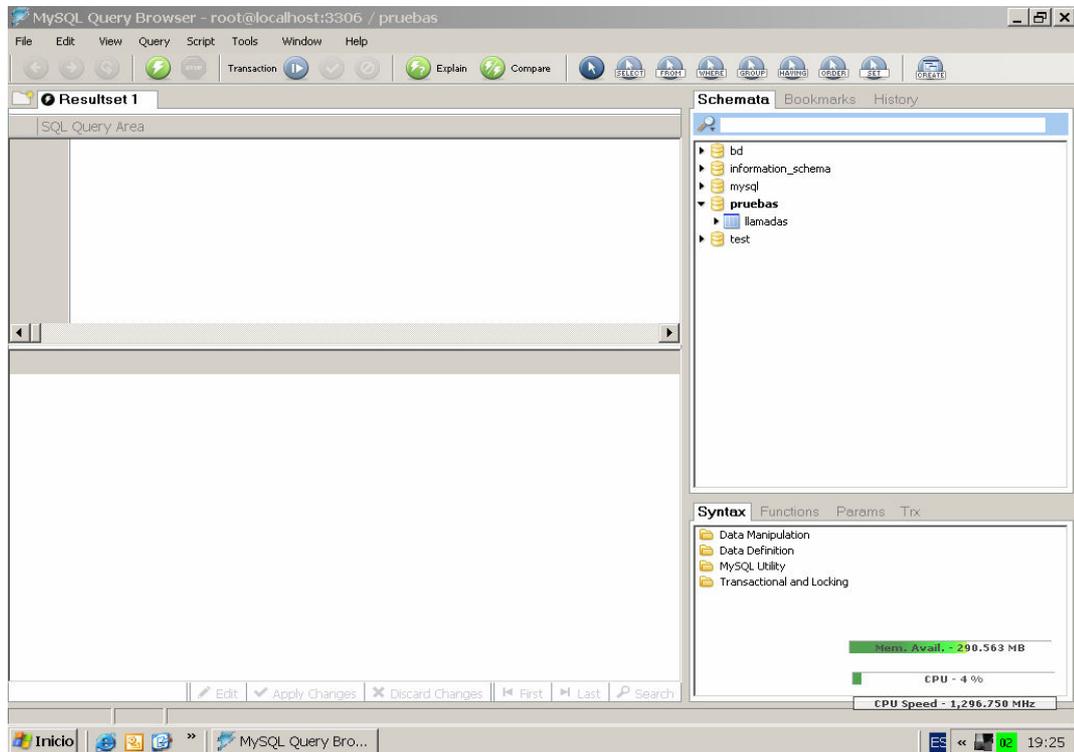
Session control:
Session
New session [19.02.2006 01.15.25] Store
 Store settings automatically on Commit

About Settings < Back Commit Stop Help

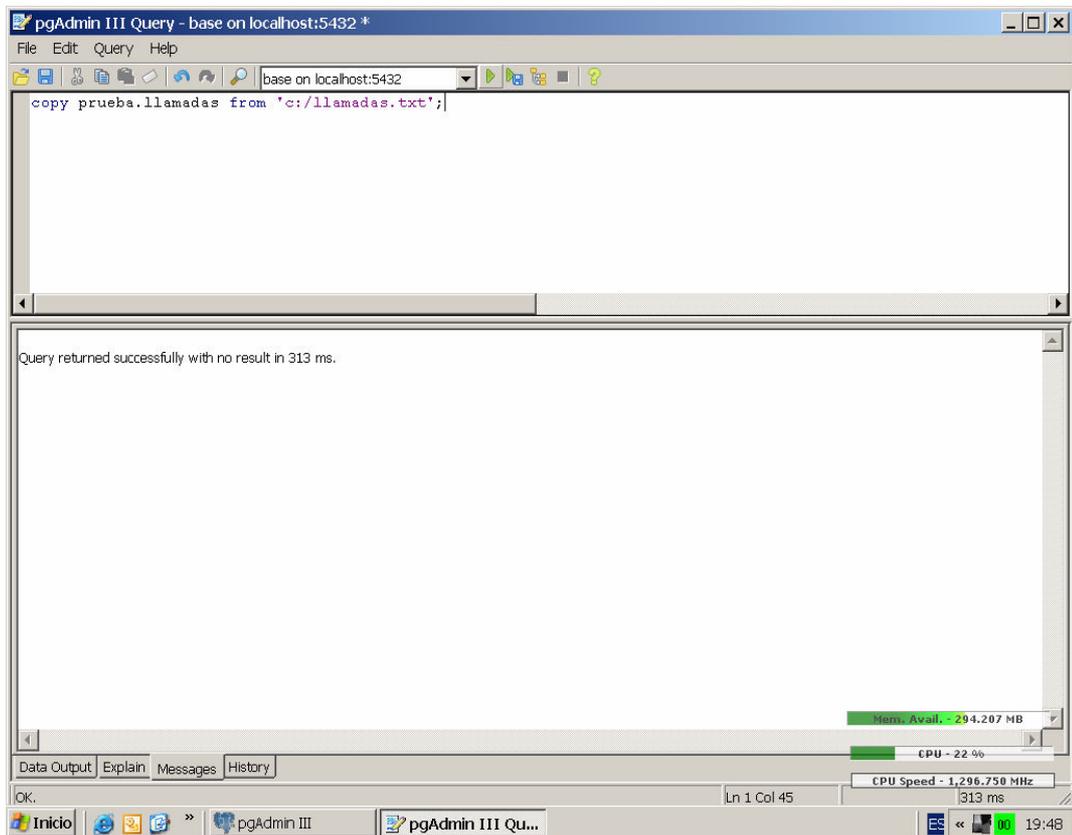
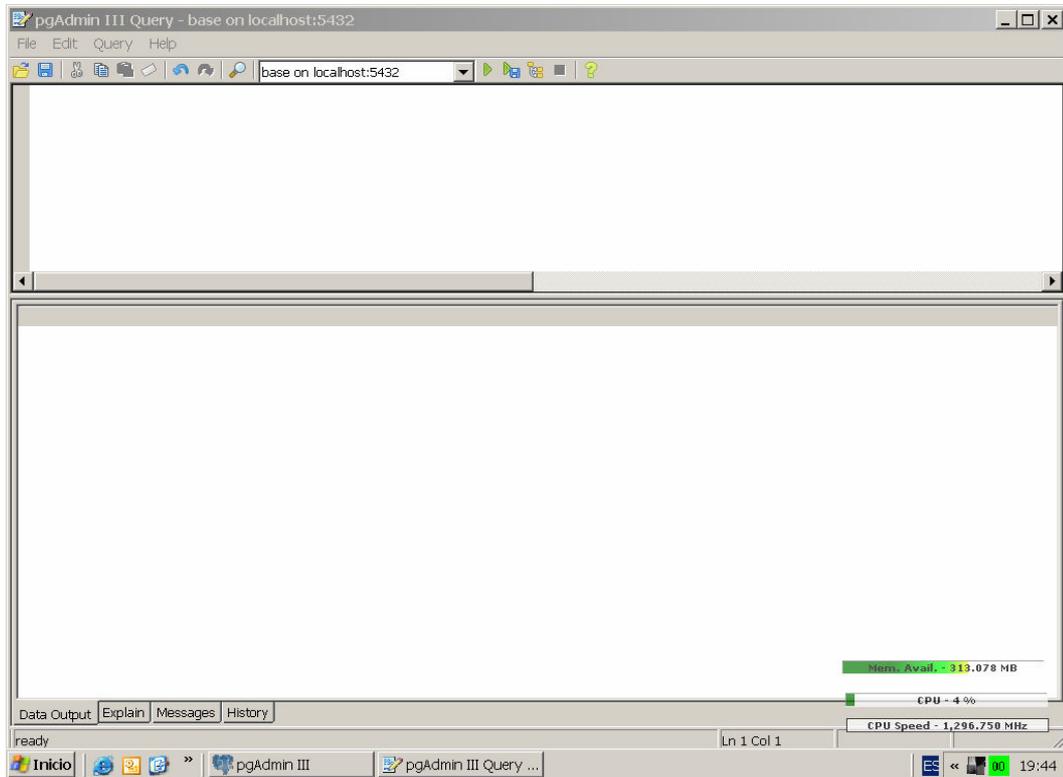
Mem. Avail. - 240.066 MB
CPU - 9.1%
CPU Speed - 1,296.750 MHz

Inicio Access2Postgre... 01:19

Anexo 2: Migración desde un archivo plano hacia MySQL.



Migración desde un archivo plano hacia PostgreSQL.



Anexo 3: Velocidad en búsqueda y ordenamiento de datos para MySQL.

The screenshot shows the MySQL Query Browser interface. The SQL Query Area contains the following query:

```

1 SELECT * FROM LLAMADAS
2 WHERE TIPO_LLAMADA LIKE 'I%'
3 ORDER BY FECHA DESC;
    
```

The result set is displayed as a table with the following columns: telefono_cabina, nro_cabina, and nro_telefono. The status bar indicates that 482 rows were fetched in 0.0689s (0.1817s).

telefono_cabina	nro_cabina	nro_telefono
072849092	06	005173504710
072849092	06	005173504710
072847067	02	005173501618
072849094	08	005173504834
072847067	02	005142556653
072849091	01	00573112745509
072849091	01	0034679644166
072849091	01	0051749725170

System performance indicators at the bottom right show: Mem. Avail. - 302.445 MB, CPU - 17 %, and CPU Speed - 765.072 MHz.

The screenshot shows the MySQL Query Browser interface. The SQL Query Area contains the following query:

```

1 SELECT TIPO_LLAMADA, SUM(VALOR_NETO) TOTAL
2 FROM LLAMADAS
3 GROUP BY TIPO_LLAMADA;
    
```

The result set is displayed as a table with the following columns: TIPO_LLAMADA and TOTAL. The status bar indicates that 7 rows were fetched in 0.0081s (0.3481s).

TIPO_LLAMADA	TOTAL
0	0
CELULAR	1241.747399999...
INTERNACIONAL	399.0621
LOCAL	167.8352
NACIONAL	493.7667000000...
REGIONAL	191.2188
SERVICIOS	3.7415

System performance indicators at the bottom right show: Mem. Avail. - 302.527 MB, CPU - 27 %, and CPU Speed - 438.505 MHz.

MySQL Query Browser - root@localhost:3306 / pruebas

File Edit View Query Script Tools Window Help

Transaction Explain Compare

Resultset 1 Resultset 2

SQL Query Area

```

1 SELECT NRO_CABINA CABINA, DESTINO_LLAMADA DESTINO, COUNT(DESTINO_LLAMADA) NRO_LLAMADAS
2 FROM LLAMADAS
3 WHERE DESTINO_LLAMADA = 'ARGENTINA'
4 GROUP BY CABINA, DESTINO;

```

Schemata Bookmarks

- bd
- information_schema
- mysql
- pruebas
 - llamadas
 - test

Syntax Functions Params

- Data Manipulation
- Data Definition
- MySQL Utility
- Transactional and Locking

CABINA	DESTINO	NRO_LLAM...
01	ARGENTINA	3
05	ARGENTINA	1
06	ARGENTINA	1

3 rows fetched in 0.0099s (0.0441s) | Edit | Apply Changes | Discard Changes | First | Last | Search

1: 1

Mem. Avail. - 307.039 MB

CPU - 7 %

CPU Speed - 408.233 MHz

Inicio MySQL Query Br...

20:03

Velocidad en búsqueda y ordenamiento de datos para PostgreSQL.

pgAdmin III Query - base on localhost:5432 *

File Edit Query Help

base on localhost:5432

```

SELECT * FROM PRUEBA.LLAMADAS
WHERE TIPO_LLAMADA LIKE 'I%'
ORDER BY FECHA DESC;

```

R...	telefono...	nro_cabin...	nro_telef...	destino_ll...	tipo_llam...	fecha (text)	hora (text)	tiempo (t...	x (text)	valor_tari...	ice (text)	valor_net...
1	072849095	07	00517399...	PERU ...	INTERNA...	22/05/2005	16:04:42	1.10	X	0.263	0.0442	0.2946
2	072849095	07	00517399...	PERU ...	INTERNA...	22/05/2005	16:02:19	1.23	X	0.263	0.0485	0.3235
3	072849091	01	00517497...	PERU ...	INTERNA...	22/05/2005	14:29:44	5.50	X	0.263	0.2170	1.4465
4	072849091	01	00346796...	ESPAÑA ...	INTERNA...	22/05/2005	14:21:23	1.10	X	0.287	0.0474	0.3157
5	072849091	01	00573112...	COLOMBI...	INTERNA...	22/05/2005	11:08:34	5.08	X	0.263	0.2004	1.3360
6	072847067	02	00514255...	PERU ...	INTERNA...	22/05/2005	10:28:52	3.32	X	0.263	0.1310	0.8732
7	072849094	08	00517350...	PERU ...	INTERNA...	22/05/2005	09:54:42	2.48	X	0.263	0.0978	0.6522
8	072847067	02	00517350...	PERU ...	INTERNACIONAL	05/2005	09:55:39	0.78	X	0.263	0.0308	0.2051
9	072849092	06	00517350...	PERU ...	INTERNA...	22/05/2005	09:54:10	2.12	X	0.263	0.0836	0.5576
10	072849092	06	00517350...	PERU ...	INTERNA...	22/05/2005	09:53:36	0.02	X	0.263	0.0000	0.00
11	072849091	01	00517256...	PERU ...	INTERNA...	21/05/2005	17:56:59	1.63	X	0.263	0.0651	0.4340
12	072849091	01	00514496...	PERU ...	INTERNA...	21/05/2005	17:47:25	0.02	X	0.263	0.0000	0.00
13	072847067	02	00525555...	MEXICO ...	INTERNA...	21/05/2005	17:31:04	0.17	X	0.442	0.0113	0.0751

Data Output Explain Messages History

OK. Ln 1 Col 22 482 rows. 640+141 ms

Mem. Avail. - 225.586 MB

CPU - 19 %

CPU Speed - 345.333 MHz

Inicio pgAdmin III pgAdmin III Qu...

20:21

pgAdmin III Query - base on localhost:5432 *

File Edit Query Help

base on localhost:5432

```
SELECT TIPO_LLAMADA, SUM(TO_NUMBER(VAOR_NETO,'999999990.0000')) AS TOTAL
FROM PRUEBA.LLAMADAS
GROUP BY TIPO_LLAMADA;
```

R...	tipo_llam...	total (nu...
1	CELULAR	1241.7474
2	INTERNA...	399.0621
3	0	0.0000
4	NACIONAL	493.7667
5	LOCAL	167.8352
6	REGIONAL	191.2188
7	SERVICIOS	3.7415

Data Output Explain Messages History

OK. Ln 4 Col 1 7 rows. 110+0 ms

Mem. Avail. - 293.223 MB
CPU - 8 %
CPU Speed - 482.993 MHz

Inicio pgAdmin III pgAdmin III Qu... 11:32

pgAdmin III Query - base on localhost:5432 *

File Edit Query Help

base on localhost:5432

```
SELECT NRO_CABINA AS CABINA, DESTINO_LLAMADA AS DESTINO, COUNT(DESTINO_LLAMADA) AS NRO_LLAMADAS
FROM PRUEBA.LLAMADAS
WHERE TRIM(DESTINO_LLAMADA) = 'ARGENTINA'
GROUP BY CABINA, DESTINO;
```

R...	cabina (t...	destino (t...	nro_llam...
1	01	ARGENTI...	3
2	05	ARGENTI...	1
3	08	ARGENTI...	1

Data Output Explain Messages History

OK. Ln 4 Col 26 3 rows. 32+0 ms

Mem. Avail. - 266.840 MB
CPU - 4 %
CPU Speed - 270.561 MHz

Inicio pgAdmin III pgAdmin III Qu... 20:26

Anexo 4: Velocidad en procedimiento de cálculo de columna para MySQL.

MySQL Query Browser - root@localhost:3306 / pruebas

SQL Query Area

```

1 SELECT * FROM(
2 SELECT CONCAT('nro llamadas a: ', DESTINO_LLAMADA ),
3 COUNT(VALOR_PAGAR) AS TOTAL, SUM(TIEMPO) AS TIEMPO,
4 SUM(ICE + IVA) AS IMPUESTOS, SUM(VALOR_NETO)AS VALOR
5 FROM LLAMADAS
6 GROUP BY DESTINO_LLAMADA) TABLA;

```

CONCAT('nro llamadas a: ', DESTINO_LLAMA...	TOTAL	TIEMPO	IMPUESTOS	VALOR
nro llamadas a:	1	0.05	0	
nro llamadas a: ALAMOR	6	25.19	0.8169	3.2676
nro llamadas a: ALAUSI	27	40.43	1.8597	6.8884
nro llamadas a: ALOAG	2	1.78	0.0817	0.2754
nro llamadas a: AMALUZA	1	2.1	0.068	0.2423
nro llamadas a: AMBATO	55	146.12	6.7162	24.220
nro llamadas a: ARENILLAS	17	49.1	1.5932	5.519
nro llamadas a: ARGENTINA	5	11.4	1.363	5.0464
nro llamadas a: ATACAMES	1	5.28	0.2423	0.9809
nro llamadas a: ATUNTAQUI	3	7.29	0.3347	1.1592

183 rows fetched in 0.0738s (0.4849s)

System Status: Mem. Avail. - 271.547 MB, CPU - 5%, CPU Speed - 536.803 MHz

Velocidad en procedimiento de cálculo de columna para PostgreSQL.

pgAdmin III Query - base on localhost:5432 *

base on localhost:5432

```

SELECT * FROM(
SELECT ('nro llamadas a: '|| DESTINO_LLAMADA ),
COUNT(VALOR_PAGAR) AS TOTAL, SUM(TO_NUMBER(TIEMPO,'999990.0000')) AS TIEMPO,
SUM(TO_NUMBER(ICE,'99990.0000') + TO_NUMBER(IVA,'999990.0000')) AS IMPUESTOS,
SUM(TO_NUMBER(VALOR_NETO,'999990.0000'))AS VALOR
FROM PRUEBA.LLAMADAS
GROUP BY DESTINO_LLAMADA) TABLA;

```

R...	?column?...	total (nt8)	tiempo (n...	impuesto...	valor (nu...
1	nro llama...	1	5.77	0.2648	0.9809
2	nro llama...	1	5.55	0.2556	0.9469
3	nro llama...	3	11.39	0.5228	1.9363
4	nro llama...	1	0.17	0.0078	0.0289
5	nro llama...	7	9.64	0.3129	1.1592
6	nro llama...	9	54.75	1.7748	6.5736
7	nro llama...	3	11.46	1.3625	5.0464
8	nro llama...	12	22.01	0.7140	2.6448
9	nro llama...	27	40.43	1.8597	6.8884
10	nro llama...	22	61.98	2.8488	10.5519
11	nro llama...	3	24.59	1.1288	4.1803
12	nro llama...	3	0.09	0.0041	0.0153
13	nro llama...	2	27.20	0.8822	3.2676
14	nro llama...	4	19.19	0.8808	3.2623
15	nro llama...	1	1.62	0.0743	0.2754
16	nro llama...	1	2.35	0.0761	0.2820
17	nro llama...	2	3.16	0.1024	0.3792

Data Output Explain Messages History

System Status: Mem. Avail. - 289.793 MB, CPU - 11%, CPU Speed - 185.236 MHz

184 rows, 531+16 ms

Anexo 5: La capacidad para subconsultas se verificó únicamente en PostgreSQL, ya que en MySQL no se pueden realizar subconsultas.

The screenshot shows the pgAdmin III Query tool interface. The query editor contains the following SQL code:

```
select nro_telefono, destino_llamada, tiempo, TIPO_LLAMADA
from prueba.llamadas
where TIPO_llamada in(
select TIPO_llamada from prueba.llamadas
where TIPO_llamada = 'NACIONAL') and
to_number(tiempo,'9990.000') > 5;
```

The results pane displays 14 rows of data:

R...	nro_telef...	destino_ll...	tiempo (t...	tipo_llam...
1	042880948	GUAYAQU...	5.73	NACIONAL
2	042361095	GUAYAQU...	10.52	NACIONAL
3	042652071	GUAYAQU...	7.08	NACIONAL
4	052770098	SAN CAM...	8.12	NACIONAL
5	052399485	LEONIDA...	5.88	NACIONAL
6	022773229	VALLE HE...	5.55	NACIONAL
7	042941445	SANTA E...	7.47	NACIONAL
8	042807634	DURAN ...	5.30	NACIONAL
9	032847124	AMBATO ...	13.33	NACIONAL
10	022408219	QUITO ...	8.68	NACIONAL
11	042845254	GUAYAQU...	8.80	NACIONAL
12	032940835	RIOBAMB...	5.27	NACIONAL
13	022691300	GUAMANI...	7.52	NACIONAL
14	052637485	PORTOVI...	7.57	NACIONAL

The status bar at the bottom indicates: OK, Ln 7 Col 1, 162 rows, 62+31 ms. System performance indicators show: Mem. Avail. - 314.320 MB, CPU - 5%, and CPU Speed - 240.654 MHz. The taskbar shows the pgAdmin III application is active.

Anexo 6: Capacidad para combinación de resultados por medio de la característica UNION para MySQL.

The screenshot shows the MySQL Query Browser interface. The SQL Query Area contains the following query:

```

1 SELECT * FROM(
2 SELECT CONCAT('nro de llamadas a: ', DESTINO_LLAMADA ),
3 TIPO_LLAMADA ,COUNT(VALOR_PAGAR) AS TOTAL
4 FROM LLAMADAS
5 GROUP BY DESTINO_LLAMADA
6 union
7 select CONCAT('TOTALES DE ', DESTINO_LLAMADA ),
8 TIPO_LLAMADA , SUM(VALOR_PAGAR) AS TOTAL
9 FROM LLAMADAS
10 GROUP BY DESTINO_LLAMADA) RESULTADO
11 ORDER BY TIPO_LLAMADA;
    
```

The results table shows the following data:

CONCAT(nro de llamadas a: ' DESTINO_LLA...	TIPO_LLAMADA	TOTAL
nro de llamadas a: BLOQUEADO	0	18
nro de llamadas a: Desconocido	0	101
TOTALES DE BLOQUEADO	0	1.43
TOTALES DE Desconocido	0	53.99
nro de llamadas a:	CELULAR	1
nro de llamadas a: CELULAR-ALLEGRO	CELULAR	55
nro de llamadas a: CELULAR-BELLSOUTH	CELULAR	754
nro de llamadas a: CELULAR-MOVISTAR	CELULAR	506
nro de llamadas a: CELULAR-PORTA	CELULAR	2120
TOTALES DE	CELULAR	0

366 rows fetched in 0.0045s (0.5556s)

Capacidad para combinación de resultados por medio de la característica UNION para PostgreSQL.

The screenshot shows the pgAdmin III Query tool interface. The SQL Query Area contains the following query:

```

SELECT * FROM(
SELECT ('nro de llamadas a: ' || DESTINO_LLAMADA ),
TIPO_LLAMADA ,COUNT(VALOR_PAGAR) AS TOTAL
FROM PRUEBA.LLAMADAS
GROUP BY DESTINO_LLAMADA, TIPO_LLAMADA
union
select ('TOTALES DE ' || DESTINO_LLAMADA ),
TIPO_LLAMADA , SUM(TO_NUMBER(VALOR_PAGAR,'999990.0000')) AS TOTAL
FROM PRUEBA.LLAMADAS
GROUP BY DESTINO_LLAMADA, TIPO_LLAMADA) RESULTADO
ORDER BY TIPO_LLAMADA;
    
```

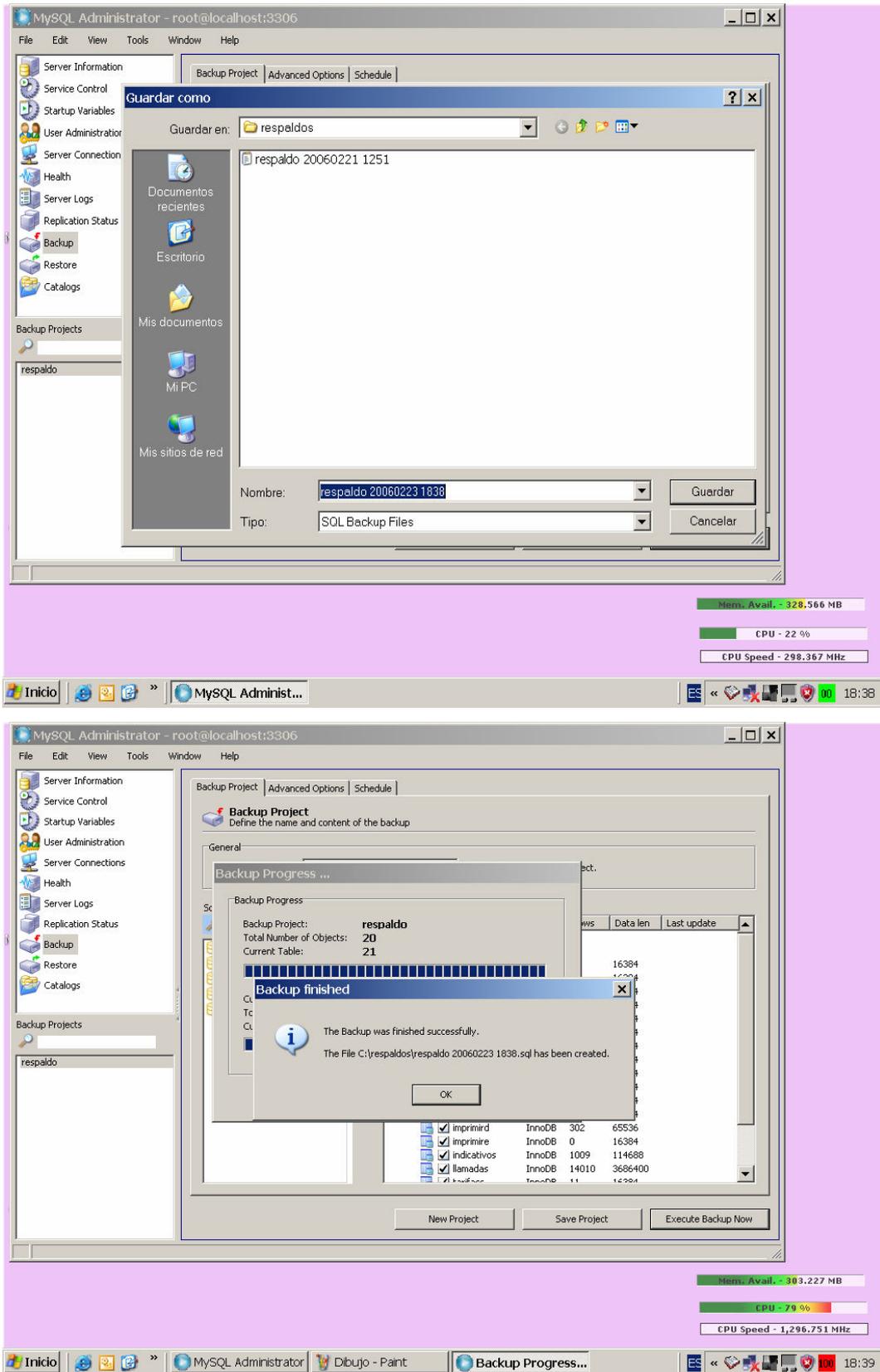
The results table shows the following data:

R...	?column?...	tipo_llam...	total (nu...
1	TOTALES...	0	0.00
2	nro de lla...	0	11
3	TOTALES...	0	0.00
4	nro de lla...	0	12
5	TOTALES...	CELULAR	268.96
6	TOTALES...	CELULAR	265.81
7	TOTALES...	CELULAR	0.13
8	TOTALES...	CELULAR	14.48
9	TOTALES...	CELULAR	997.24
10	TOTALES...	CELULAR	0.00
11	nro de lla...	CELULAR	51
12	TOTAL FS...	CELULAR	31.10

Data Output Explain Messages History

OK. Ln 11 Col 23 390 rows. 141+32 ms

Anexo 7: Respaldo de datos para MySQL (Backups).



Respaldo de datos para PostgreSQL (Backups).

