

UNIVERSIDAD DE GUAYAQUIL

**FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES**

**AUMENTO DE LA DISPONIBILIDAD DE BASES DE DATOS
MYSQL UTILIZANDO UNA CONFIGURACIÓN
DE BASE EN MODO DE ESPERA E
INSTALACIÓN DE AGENTES
PARA AUTOMATIZAR
CONEXIONES**

TESIS DE GRADO

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

AUTOR: POLL MIGUEL CEDEÑO TELLO

TUTOR: ING. RAYNAUD QUIROZ

GUAYAQUIL – ECUADOR

2011

Guayaquil, 11 de Octubre del 2010

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo de investigación, “Aumento de la disponibilidad de bases de datos MySQL utilizando una configuración de base en modo de espera e instalación de agentes para automatizar conexiones “elaborado por el Sr. Poll Miguel Cedeño Tello, egresado de la Carrera de Ingeniería en Sistemas Computacionales, Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil, previo a la obtención del Título de Ingeniero en Sistemas, me permito declarar que luego de haber orientado, estudiado y revisado, la Apruebo en todas sus partes.

Atentamente

.....
Ing. Raynaud Quiroz
TUTOR

DEDICATORIA

Agradezco a Dios que me ha dado la fuerza y capacidad para llegar hasta aquí, así como a mi familia que me ha estado apoyando en toda la trayectoria de mi carrera tanto en las buenas como en las malas.

TRIBUNAL DE GRADO

Ing. Fernando Abad Montero
DECANO DE LA FACULTAD
CIENCIAS MATEMÁTICAS Y FÍSICAS

Ing. Juan Chanabá Alcócer
DIRECTOR

Ing. Raynaud Quiroz
TUTOR

Inga. Ana María Arellano
PROFESOR DEL ÁREA - TRIBUNAL

Ing. Juan Carlos Cedeño
PROFESOR DEL ÁREA – TRIBUNAL

Dr. José Júpiter
SECRETARIO

UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

CARRERA DE INGENIERÍA EN SISTEMAS

COMPUTACIONALES

AUMENTO DE LA DISPONIBILIDAD DE BASES DE DATOS

MYSQL UTILIZANDO UNA CONFIGURACIÓN

DE BASE EN MODO DE ESPERA E

INSTALACIÓN DE AGENTES

PARA AUTOMATIZAR

CONEXIONES

MySQL es una de las bases de datos libres y de código abierto más usadas en muchos ambientes laborales. Tiene un sin número de herramientas útiles y para cada lugar de trabajo. Este proyecto busca mejorar una de las áreas de MySQL que es la replicación. El problema se debe a que el método de replicación es asíncrono esto significa que no es inmediato y que se debe esperar cierta cantidad de tiempo para que la información se replique, y este tiempo va a depender de varios factores como pueden ser hardware, software, configuraciones; por estos motivos se puede dar el caso de que al momento de replicar la información no llegue o llegue incompleta. La solución que se plantea consiste en extraer del cliente las consultas que se envían a la base de datos y guardarlos en archivos; para lograr esto se procedió con la edición del código fuente del ODBC, agregando también una función la cual va a estar indicando si las consultas se están ejecutando correctamente; caso contrario si alguna no se puede ejecutar se escribirá un mensaje de error en un registro que llevará el nombre del DSN. Se procederá a la instalación de agentes tanto en los clientes como en un servidor esclavo. El agente que se encontrará en el servidor esclavo estará censando la comunicación con el maestro, si se llegara a perder la comunicación se realizarán las configuraciones necesarias para que el servidor funcione como maestro para así poder recibir las consultas enviadas por todos los clientes. El agente en el cliente va a estar encargado de leer el registro con el nombre del DSN; al detectar algún mensaje de error procederá a realizar procesos para replicar la información almacenada, para que así no exista pérdida de esta.

UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

CARRERA DE INGENIERÍA EN SISTEMAS

COMPUTACIONALES

AUMENTO DE LA DISPONIBILIDAD DE BASES DE DATOS

MYSQL UTILIZANDO UNA CONFIGURACIÓN

DE BASE EN MODO DE ESPERA E

INSTALACIÓN DE AGENTES

PARA AUTOMATIZAR

CONEXIONES

MySQL is one of the free data base and open source used in many workplaces. It has different useful tools for each Workplace. This project seeks to improve one of the area of MySQL that is the replication. The problem is that the method of replication is asynchronous this means that is not immediate and it needs to expect a certain amount of time in order to that this information can be replicated and this time will depend on several factors such as hardware, software, configurations, for these reasons it may be the case when comes the moment to replicate the information, this information does not arrive or arrive incomplete. The solution that arises is to extract customer queries that are send to the database and save in files; to achieve this we proceeded with the release of the ODBC code source, also adding a function which will be indicate if queries are running correctly, otherwise if you cannot run it will write an error message in a register held by the name of the DSN. It will install both agents clients like a slave server. The agent may be found on the slave communication registers you with the teacher, if you were to lose the communication was made necessary settings for the server to work as a teacher in order to receive the queries sent by all clients. The agent on the client will be responsible for reading the record with the name of the DSN, to detect any error messages shall carry out processes to replicate the information stored, so that no loss of it.

CERTIFICADO DE ACEPTACIÓN DEL TUTOR

En mi calidad de Tutor del Primer Curso de Fin de Carrera, nombrado por el Departamento de Graduación y la Dirección de la Carrera de Ingeniería en Sistemas Computacionales de la Universidad de Guayaquil,

CERTIFICO:

Que he analizado el Proyecto de Grado presentado por el/la egresado(a) Poll Miguel Cedeño Tello, como requisito previo para optar por el título de Ingeniero cuyo problema es:

AUMENTO DE LA DISPONIBILIDAD DE BASES DE DATOS MYSQL UTILIZANDO UNA CONFIGURACIÓN DE BASE EN MODO DE ESPERA E INSTALACIÓN DE AGENTES PARA AUTOMATIZAR CONEXIONES

considero aprobado el trabajo en su totalidad.

Presentado por:

Poll Miguel Cedeño Tello

0922634670

Apellidos y Nombres completos

Cédula de ciudadanía N°

Tutor: Ing. Raynaud Quiroz

Guayaquil, _____ de _____

ÍNDICE GENERAL

CARÁTULA	i
CARTA DE ACEPTACIÓN DEL TUTOR	ii
DEDICATORIA	iii
TRIBUNAL	iv
RESUMEN	v
ABSTRACT	vi
CERTIFICADO DE APROBACIÓN	vii
ÍNDICE GENERAL	viii
ÍNDICE DE CUADROS	x
ÍNDICE DE GRÁFICOS	xi
INTRODUCCIÓN	1
CAPÍTULO 1.- EL PROBLEMA	
PLANTEAMIENTO DEL PROBLEMA	2
Ubicación del Problema en un contexto	2
Situación Conflicto	3
Causas del problema, Consecuencias	4
Delimitación del Problema	6
Formulación del Problema	6
Evaluación del Problema	7
OBJETIVOS DE LA INVESTIGACIÓN	9
JUSTIFICACIÓN E IMPORTANCIA DE LA INVESTIGACIÓN	10
GLOSARIO	11
CAPÍTULO II.- MARCO TEÓRICO	
FUNDAMENTACIÓN TEÓRICA	
Antecedentes del estudio	13

Fundamentación teórica	14
Fundamentación legal	97
Hipótesis preguntas a contestarse	98
Variables de la Investigación	99
Definiciones Conceptuales	99
GLOSARIO	100
CAPÍTULO III.- METODOLOGÍA	
DISEÑO DE LA INVESTIGACIÓN	105
Modalidad de la Investigación	105
Tipo de Investigación	105
Población y Muestra	107
Operacionalización de las Variables	108
Instrumento de Recolección de Datos	108
Instrumentos de la Investigación	110
Procedimientos de la investigación	111
Recolección de la Información	114
PROCESAMIENTO Y ANÁLISIS	114
CAPÍTULO IV.- MARCO ADMINISTRATIVO	
CRONOGRAMA	115
PRESUPUESTO	118
REFERENCIAS BIBLIOGRÁFICAS	119
ANEXOS	120

ÍNDICE DE CUADROS

CUADRO NO. 1 DISEÑO DE INSTALACIÓN DE RPM MYSQL PARA LINUX	45
CUADRO NO. 2 RPM's NECESARIOS PARA ALGUNOS SISTEMAS OPERATIVOS	45
CUADRO NO. 3 VALOR DEL CPU PARA EL CUAL FUE CREADO EL RPM	46
CUADRO NO. 4 DISEÑO DE INSTALACIÓN DE MYSQL USANDO UN PAQUETE BINARIO (.TAR)	50
CUADRO NO. 5 PARÁMETROS DE CONEXIÓN DEL ODBC	88
CUADRO NO. 6 POBLACIÓN DE ADMINISTRADORES DE BASES DE DATOS Y USUARIOS QUE TRABAJAN CON ELLAS	107
CUADRO NO. 7 MATRIZ DE OPERACIONALIZACIÓN DE VARIABLES	108
CUADRO NO. 8 DETALLE DE EGRESOS DEL PROYECTO	118

ÍNDICE DE GRÁFICOS

	Pág.
Gráfico 1 Suma de Verificación MD5	47
Gráfico 2 Directorio de Instalación de MySQL	52
Gráfico 3 Comandos previos a expandir distribución MySQL con gunzip	53
Gráfico 4 Distribución MySQL expendida con gunzip. Instalación satisfactoria.	54
Gráfico 5 Cambiando atributos de directorio.	56
Gráfico 6 Ejecutando mysql_install_db para instalar tablas del sistema.	57
Gráfico 7 Ingresando a MySQL por primera vez	58
Gráfico 8 Asignando contraseña al usuario root.	59
Gráfico 9 Iniciando el servidor MySQL	60
Gráfico 10 Deteniendo el servidor MySQL	61
Gráfico 11 Representación ambiente de Replicación	87
Gráfico 12 Configuración del conector ODBC	89

Gráfico 13 Configuración de CMAKE	92
Gráfico 14 Selección del compilador	93
Gráfico 15 Fuentes del ODBC en Visual Studio	94
Gráfico 16 Batch Build para compilar fuentes	95
Gráfico 17 Driver ODBC listo para usarse	96

INTRODUCCIÓN

El presente proyecto tiene como finalidad dar a conocer un procedimiento para poder aumentar la disponibilidad de una base de datos, en este caso MySQL, que es actualmente una de las más utilizadas, sobre todo en lo que respecta a código abierto (Open Source). Pero MySQL también ofrece una gran variedad de opciones y herramientas para mejorar nuestro sistema de información, no solamente desarrollado por su equipo, sino también por muchas personas alrededor del mundo.

En este caso la herramienta que se ha tomado en cuenta es la Conectividad Abierta de Base de Batos ODBC (por sus siglas en inglés Open DataBase Connectivity).

Una de las opciones que se toma en cuenta es la replicación, ya que partiendo de ella podemos realizar varias configuraciones teniendo una base en modo de espera lista para cualquier contingencia que se pueda presentar, así como también opciones que a pesar de que están presentes no han sido tomadas en cuenta, pero pueden servirnos en algunos casos, ya que lo más importante para una institución hoy en día ha llegado a ser la información, y ofrecer sus servicios las veinticuatro horas del días los siete días de la semana.

CAPÍTULO I

EL PROBLEMA

1.1. PLANTEAMIENTO DEL PROBLEMA EN UN CONTEXTO

¿Cómo aumentar la disponibilidad tanto de la información así como de bases de datos MySQL en ambientes cliente – servidor usando la información generada por los clientes en un esquema de replicación?

1.2. UBICACIÓN DEL PROBLEMA EN UN CONTEXTO

Desde hace algunos años muchas empresas han optado por el uso de software libre y el código abierto, así como el uso de bases de datos que se encuentran dentro de estas tendencias; MySQL es el caso de una de ellas. MySQL permite diferentes tipos de configuraciones adaptables a las necesidades de cada usuario. Las empresas de hoy en día no pueden permitirse estar fuera de servicio por ningún motivo ya que el riesgo de pérdida de información y de clientes es grande, esta parte por la competencia; por eso es importante lograr que el cliente se haga fiel a la empresa, tanto ofreciendo un producto de calidad, como dando un buen servicio. Garantizar la continuidad de las operaciones es hoy en día una característica de calidad relevante para cualquier empresa dado el impacto que puede provocar una interrupción de las mismas, tanto desde el punto de vista económico (pérdidas tangibles) como de imagen (pérdidas intangibles).

1.3. SITUACIÓN CONFLICTO O NUDOS CRÍTICOS

En las empresas existen muchos riesgos que pueden hacer peligrar la continuidad del negocio así como la integridad de los datos, estos riesgos pueden ser de hardware por ejemplo falla en algún dispositivo, de software por ejemplo los

daños que puede causar un programa malicioso (virus), y humano por ejemplo el ataque de un hacker al servidor.

Otro factor importante a tomar en cuenta es que en algunos casos los dueños de las empresas delegan un bajo presupuesto a las soluciones tecnológicas que se van a implementar sin darse cuenta de los problemas a futuro que pueden ocurrir.

1.4. CAUSAS Y CONSECUENCIAS DEL PROBLEMA

Los riesgos de hardware pueden deberse a un incorrecto mantenimiento de los equipos, malas instalaciones eléctricas, humedad en las cercanías de los equipos, mala instalación de las redes internas de la empresa, todos estos riesgos son debidos también por la falta de mantenimiento de los equipos, así como descuido de las instalaciones de la institución.

Los riesgos por parte de software vienen de diferentes fuentes, ya sea por virus, software espía u otras circunstancias, la falta de controles en los computadores del personal es uno de los principales causantes, así como también la falta de inversión en aplicaciones como antivirus, firewall, etc., así como también en capacitación de personal para concientizarlos en un buen uso de los equipos.

Los riesgos humanos a parte de envolver parte de los riesgos anteriores, también tiene que ver con la posibilidad de ataque a un servidor por una persona interna o externa a la empresa, ya sea por obtener algún beneficio o solo por descubrir vulnerabilidades.

Otro punto a considerar es la falta o actualización de los planes de contingencia.

Si bien todas las organizaciones son conscientes que mantener actualizados sus Planes de Contingencia, es un requisito indispensable para garantizar la

efectividad de cualquier Solución de Continuidad Operativa, en los hechos esta tarea rara vez se realiza en forma adecuada. Esto puede ocurrir por diversos factores, como por ejemplo:

- Dificultad para detectar oportunamente las situaciones que provocan cambios en los Planes debido, entre otras cosas, a que las mismas son de muy diferente origen y naturaleza (cambios en la estructura organizativa, plataforma tecnológica, unidades de negocio, requerimientos de los accionistas, etc.).
- Falta de recursos, conocimientos y experiencia para formular los procesos o tareas a ser modificados.
- Falta de coordinación entre los equipos de respuesta involucrados en los procesos a modificar (sobre todo si se encuentran geográficamente dispersos).
- Falta de definición clara sobre quién debe aprobar los cambios realizados en los planes.

Todos estos riesgos tarde o temprano van a afectar el desenvolvimiento normal de las actividades de la empresa, logrando dejarla fuera de servicio por corto o largo tiempo dependiendo de los daños.

1.5. DELIMITACIÓN DEL PROBLEMA

El fallo de un servidor ya sea a nivel de hardware o de software, causando pérdida de información, puede afectar a cualquier tipo de institución, y las

configuraciones dadas en este documento para evitar este problema son una selección de las muchas opciones disponibles hoy en día.

El prototipo a presentar será evaluado en un ambiente cliente – servidor, no será usado en ambientes web; se usaran configuraciones de replicación maestro – esclavo en los servidores de bases de datos, y funcionará si las tablas de la base tienen identificadores únicos o clave primaria, es decir donde no pueda existir duplicidad de datos.

El uso de agentes para replicar la información generada en los clientes podrá ser usado, siempre y cuando, la aplicación use como método de conexión sea a través de ODBC.

1.6. FORMULACIÓN DEL PROBLEMA

Teniendo en cuenta los lugares donde se origina y en donde se almacena la información:

¿Es posible aumentar la disponibilidad de la información?, ¿Se lo puede realizar en un tiempo menor al que se tiene en el presente?, ¿Cómo poder tener un respaldo extra de la información generada por los clientes?

1.7. EVALUACIÓN DEL PROBLEMA

Los aspectos generales de evaluación a seguir son:

Delimitado: Este problema está enfocado hacia el control de la conexión entre el cliente y el servidor de base de datos, y la facilidad de las configuraciones de replicación de bases de datos para obtener los beneficios de una base en modo de espera.

Evidente: Las empresas están en la necesidad de ofrecer un servicio continuo a sus clientes, así como la disponibilidad de la información, ya que la competencia hoy en día no perdona errores.

Original: Este proyecto utilizará la última versión de MySQL ya que esta permite la replicación semi-síncrona, esta versión es la primera liberada desde la compra de esta base de datos por una compañía privada. También la solución propuesta no existe en la actualidad en las herramientas que ofrece MySQL.

Factible: Al usar herramientas de software libre y código abierto la implementación para las empresas no se verá afectada a nivel de software, en el aspecto monetario, permitiéndole también estar al día en las tendencias actuales.

Identifica los productos esperados: Se obtendrán dos agentes de control de conexiones uno para el servidor y uno para el cliente. El agente del servidor sólo será probado sobre Linux, para Windows. También para el cliente se tendrá una versión editada del ODBC de MySQL.

Concreto: Este proyecto se centra en mejorar la disponibilidad de la información, mediante configuraciones, la elaboración de agentes, y la factibilidad que se tiene al usar estos.

1.8. OBJETIVOS

1.8.1. OBJETIVO GENERAL

Analizar el uso de agentes para aumentar la disponibilidad de bases de datos MySQL, partiendo de un ambiente de replicación y usando la información generada por los clientes.

1.8.2. OBJETIVO ESPECÍFICO

- Determinar el método a utilizar para extraer las consultas (queries) generadas en el cliente.
- Elaborar un agente que sirva para replicar al servidor la información que se guardará en los clientes.
- Elaborar un agente que ejecute los comandos necesarios para que como contingencia el servidor esclavo funcione como un servidor maestro.

1.9. JUSTIFICACIÓN E IMPORTANCIA

La importancia en la implementación de este proyecto ayudará a disminuir los riesgos de pérdida de información, a nivel de base de datos MySQL ya que esta maneja una replicación asíncrona, tanto en sistemas operativos abiertos (Unix), como cerrados (Windows). Así como también el uso de herramientas, la posibilidad de aplicar el mismo concepto en otras como por ejemplo JDBC (por sus siglas en inglés Java DataBase Connectivity).

Al lograr demostrar varios de los métodos que se tienen para evitar pérdida de información, muchas instituciones podrían considerar el uso de herramientas de software libre.

GLOSARIO

ODBC (Open DataBase Connectivity).- Es un estándar de acceso a bases de datos desarrollado por SQL Access Group en 1992, el objetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué sistema de gestión de bases de datos (DBMS) almacene los datos, ODBC logra esto al insertar una capa intermedia (CLI) denominada nivel de Interfaz de Cliente SQL, entre la aplicación y el DBMS, el propósito de esta capa es traducir las consultas de datos de la aplicación en comandos que el DBMS entienda.

JDBC (Java Database Connectivity).- Es un derivado inspirado en el mismo, una interfaz de programación de aplicaciones que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java independientemente del sistema operativo donde se ejecute o de la base de datos a la cual se accede utilizando el dialecto SQL del modelo de base de datos que se utilice.

SQL (Structured Query Language).- El lenguaje de consulta estructurado es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en éstas.

Open Source.- Código abierto es el término con el que se conoce al software distribuido y desarrollado libremente. El código abierto tiene un punto de vista más orientado a los beneficios prácticos de compartir el código que a las cuestiones morales y/o filosóficas las cuales destacan en el llamado software libre.

Software Libre.- El software libre (en inglés free software, aunque esta denominación también se confunde a veces con "gratis" por la ambigüedad del término en el idioma inglés) es la denominación del software que respeta la libertad de los usuarios sobre su producto adquirido y, por tanto, una vez obtenido puede ser usado, copiado, estudiado, modificado y redistribuido libremente.

GNU.- La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente sus siglas del inglés GNU GPL, es una licencia creada por la Free Software Foundation en 1989 (la primera versión), y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

CAPÍTULO II

MARCO TEÓRICO

2.1. ANTECEDENTES DEL ESTUDIO

Hoy en día existe mucha documentación acerca de la replicación, así como diferentes tipos de configuraciones posibles de usar incluyendo la semi-síncrona presentada a inicios de este año, a pesar de esto es utilizada por pocas personas ya sea por falta de conocimiento, por no querer cambiar las configuraciones que tienen en sus servidores.

El estudio actual no es solamente acerca de la replicación, se trata de mejorar la disponibilidad de la información usando configuraciones típicas de las bases de datos en modo de espera así como teniendo respaldos que pueden ser generados en los clientes.

2.2. FUNDAMENTACIÓN TEÓRICA

MYSQL

Es un sistema de gestión de base de datos relacional, multi hilo, multi usuario que sirve de interfaz entre los usuarios y la base de datos; diseñado para entorno de producción críticos con alta carga de trabajo.

El software tiene una doble licencia. Los usuarios pueden elegir entre un entorno de código abierto (Open Source) o adquirir una licencia comercial.

Elpaís.COM “**La empresa de software estadounidense Oracle ha comprado hoy Sun Microsystems**”.

En la actualidad Oracle adquirió Sun Microsystems y con ella adquirió a MySQL. Con esto se pensaba que MySQL iba a dejar de ser de código abierto pero hasta los momentos sigue siéndolo, incluyendo la versión liberada recientemente MySQL 5.5.

<http://www.elblogsalmon.com/empresas/oracle-compra-sun-microsystems>

(Página donde se encuentra el artículo).

Equipo de desarrollo de MySQL **“Características de MySQL”**.

Entre sus características tenemos:

- Escrito en C y en C++
- Probado con un amplio rango de compiladores diferentes
- Funciona en diferentes plataformas.
- Usa Licencia Pública General GNU (por sus siglas en inglés General Public License) Automake, Autoconf, y Libtool para portabilidad.
- APIs disponibles para C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, y Tcl.
- Uso completo de multi hilos mediante hilos del kernel. Pueden usarse fácilmente múltiple 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 interfaz 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 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 en aplicaciones autónomas. Dichas aplicaciones pueden usarse por sí mismas o en entornos donde no hay red disponible.
- Diversos tipos de columnas

- Registros de longitud fija y longitud variable
- Sentencias y funciones
- 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 usa MySQL Server con bases de datos que contienen 50 millones de registros. También se conoce de usuarios que usan MySQL Server con 60.000 tablas y cerca de 5.000.000.000.000 de registros.
 - Se permiten hasta 64 índices por tabla. Cada índice puede consistir desde 1 hasta 16 columnas o partes de columnas. El máximo ancho de límite son 1000 bytes.
- Conectividad
 - Los clientes pueden conectar con el servidor MySQL usando sockets TCP/IP en cualquier plataforma, tanto en Windows como en Linux, para la conexión.
 - A partir MySQL 5.0, los servidores Windows soportan conexiones con memoria compartida.
 - La interfaz para el conector ODBC (MyODBC “MySQL ODBC”) proporciona soporte para programas clientes que usen este tipo de conexiones. Por ejemplo, puede usar MS Access para conectar al servidor. Los clientes pueden ejecutarse en Windows o Unix. El código fuente de MyODBC está disponible.

- 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.
- 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.”; también incluye “myisamchk”, una utilidad de línea de comandos muy rápida para efectuar estas operaciones en tablas MyISAM.
 - Todos los programas MySQL pueden invocarse con las opciones “--help” o “-?” para obtener asistencia en línea.

<http://dev.mysql.com/doc/refman/5.0/en/features.html> (Página donde se encuentra el artículo)

SISTEMAS OPERATIVOS SOPORTADOS POR MYSQL

Equipo de desarrollo de MySQL “Sistemas operativos soportados por MySQL”.

Se ha utilizado GNU Autoconfig, de modo que es posible portar MySQL a todos los sistemas modernos que tengan un compilador de C++ y una implementación funcional de subprocessos (hilos) POSIX. (El soporte de subprocessos es necesario para el servidor. Para compilar únicamente el código del cliente, no se requiere más que el compilador de C++).

MySQL ha sido compilado correctamente en las siguientes combinaciones de sistemas operativos y paquetes de subprocessos. Nótese que, para varios sistemas operativos, el soporte nativo de subprocessos funciona solamente en las versiones más recientes.

- AIX 4.x, 5.x con subprocessos nativos.
- Amiga.
- Digital Unix 4.x con subprocessos nativos.
- FreeBSD 2.x con el paquete MIT-pthreads (paquete para soporte de multi-hilos).
- FreeBSD 3.x and 4.x con subprocessos nativos.
- FreeBSD 4.x con LinuxThreads.
- HP-UX 10.20 con el paquete DCE threads o MIT-pthreads.
- HP-UX 11.x con subprocessos nativos.
- Linux 2.0+ con LinuxThreads 0.7.1+ o glibc 2.0.7+ para varias arquitecturas de CPU.
- NetBSD 1.3/1.4 Intel y NetBSD 1.3 Alpha (requiere GNU make).
- Novell NetWare 6.0.
- OpenBSD 2.5 con subprocessos nativos. OpenBSD < 2.5 con el paquete MIT-pthreads.
- OS/2 Warp 3, FixPack 29 y OS/2 Warp 4, FixPack 4.
- SCO OpenServer 5.0.X con una versión del paquete FSU Pthreads recientemente portada.

- SCO UnixWare 7.1.x.
- SCO Openserver 6.0.x.
- SGI Irix 6.x con subprocesos nativos.
- Solaris 2.5 y posteriores con subprocesos nativos en SPARC y x86.
- SunOS 4.x con el paquete MIT-pthreads package.
- Tru64 Unix.
- Windows 9x, Me, NT, 2000, XP, y 2003.

<http://dev.mysql.com/doc/refman/5.0/en/supported-os.html> (Página donde se encuentra el artículo.)

MOTORES DE ALMACENAMIENTO MYSQL Y TIPOS DE TABLAS

Equipo de desarrollo de MySQL “**Motores de almacenamiento MySQL y tipos de tablas**”.

MySQL soporta varios motores de almacenamiento que tratan con distintos tipos de tablas. Los motores de almacenamiento de MySQL incluyen algunos que tratan con tablas transaccionales y otros que no lo hacen:

- MyISAM trata tablas no transaccionales. Proporciona almacenamiento y recuperación de datos rápida, así como posibilidad de búsquedas fulltext. MyISAM se soporta en todas las configuraciones MySQL, y es el motor de almacenamiento por defecto a no ser que tenga una configuración distinta a la que viene por defecto con MySQL.
- El motor de almacenamiento MEMORY proporciona tablas en memoria. El motor de almacenamiento MERGE permite una colección de tablas MyISAM idénticas ser tratadas como una simple tabla. Como MyISAM, los motores de almacenamiento MEMORY y MERGE tratan tablas no transaccionales y ambos se incluyen en MySQL por defecto.

Nota: El motor de almacenamiento MEMORY anteriormente se conocía como HEAP.

- Los motores de almacenamiento InnoDB y BDB proporcionan tablas transaccionales. BDB se incluye en la distribución binaria MySQL-Max en aquellos sistemas operativos que la soportan. InnoDB también se incluye por defecto en todas las distribuciones binarias de MySQL 5.0. En distribuciones fuente, puede activar o desactivar estos motores de almacenamiento configurando MySQL a su gusto.
- El motor de almacenamiento EXAMPLE es un motor de almacenamiento "tonto" que no hace nada. Puede crear tablas con este motor, pero no puede almacenar datos ni recuperarlos. El objetivo es que sirva como ejemplo en el código MySQL para ilustrar cómo escribir un motor de almacenamiento. Como tal, su interés primario es para desarrolladores.
- NDB Cluster es el motor de almacenamiento usado por MySQL Cluster para implementar tablas que se particionan en varias máquinas. Está disponible en distribuciones binarias MySQL-Max 5.0. Este motor de almacenamiento está disponible para Linux, Solaris, y Mac OS X. Los desarrolladores de MySQL añadirán soporte para este motor de almacenamiento en otras plataformas, incluyendo Windows en próximas versiones.
- El motor de almacenamiento ARCHIVE se usa para guardar grandes cantidades de datos sin índices con una huella muy pequeña.
- El motor de almacenamiento CSV guarda datos en ficheros de texto usando formato de valores separados por comas.

- El motor de almacenamiento FEDERATED se añadió en MySQL 5.0.3. Este motor guarda datos en una base de datos remota. En esta versión sólo funciona con MySQL a través de la API MySQL C Client. En futuras versiones, será capaz de conectar con otras fuentes de datos usando otros drivers o métodos de conexión clientes.
- El motor de almacenamiento InnoDB se diseñó para obtener el máximo rendimiento al procesar grandes volúmenes de datos. Probablemente ningún otro motor de bases de datos relacionales en disco iguale su eficiencia en el uso de CPU. A pesar de estar totalmente integrado con el servidor MySQL, el motor de almacenamiento InnoDB mantiene su propio pool de almacenamiento intermedio para tener un cache de datos e índices en la memoria principal.

<http://dev.mysql.com/doc/refman/5.0/en/storage-engines.html> (Página donde se encuentra el artículo).

Al crear una tabla se puede especificar qué tipo de tabla deseamos usando la opción ENGINE por ejemplo:

```
CREATE TABLE prueba (campo INT) ENGINE = INNODB;
```

Si omite la opción “ENGINE”, se usa el motor de almacenamiento por defecto, que es MyISAM en versiones anteriores a la 5.5 y en versiones iguales o posteriores es INNODB. Puede cambiarlo usando las opciones de arranque “--default-storage-engine” o “--default-table-type”, o cambiando la variable de sistema “storage_engine” o “table_type”.

Para convertir una tabla de un tipo a otro, use un comando ALTER TABLE que indique el nuevo tipo:

```
ALTER TABLE prueba ENGINE = MYISAM;
```

Si se intenta usar un motor de almacenamiento que no está compilado o que está desactivado, MySQL crea una tabla de tipo MyISAM. Este comportamiento es conveniente cuando quiere copiar tablas entre servidores MySQL que soportan distintos motores.

MySQL siempre crea un fichero .frm para guardar la definición de tabla y columnas. El índice y datos de la tabla pueden estar almacenados en uno o más ficheros, en función del tipo de tabla. El servidor crea el fichero .frm por encima del nivel de almacenamiento del motor. Los motores de almacenamiento individuales crean los ficheros adicionales necesarios para las tablas que administran.

Una base de datos puede contener tablas de distintos tipos.

Las tablas transaccionales (TSTs) tienen varias ventajas sobre las no transaccionales (NTSTs):

- Más seguras. Incluso si MySQL cae o tiene problemas de hardware, puede recuperar los datos, mediante recuperación automática o desde una copia de seguridad más el log de transacciones.
- Puede combinar varios comandos y aceptarlos todos al mismo tiempo con el comando COMMIT (si autocommit está desactivado).
- Puede ejecutar ROLLBACK para ignorar los cambios (si autocommit está desactivado).

- Si falla una actualización, todos los cambios se deshacen. (Con tablas no transaccionales, todos los cambios son permanentes.)
- Motores de almacenamiento transaccionales pueden proporcionar mejor concurrencia para tablas que tienen varias actualizaciones concurrentes con lecturas.

En MySQL 5.0, InnoDB usa valores de configuración por defecto si no los especifica.

Tablas no transaccionales tienen varias ventajas al no tener una sobrecarga transaccional:

- Más rápidas
- Menor requerimiento de espacio.
- Menos memoria para actualizaciones

Puede combinar tablas transaccionales y no transaccionales en el mismo comando para obtener lo mejor de ambos mundos. Sin embargo, en una transacción con autocommit desactivado, los cambios de tablas no transaccionales son permanentes inmediatamente y no pueden deshacerse.

INSTALACIÓN DE MYSQL EN WINDOWS CON UN INSTALADOR AUTOMÁTICO

Siempre que sea posible debería emplearse la distribución binaria. Es más simple que las otras y no se necesita ninguna herramienta adicional para poner en funcionamiento el servidor MySQL.

Para ejecutar MySQL para Windows, se necesita lo siguiente:

Un sistema operativo Windows de 32 bits, tal como 9x, Me, NT, 2000, XP, Vista, 7 o Windows Server 2003 o superiores.

Se recomienda fuertemente el uso de un sistema operativo Windows basado en NT (NT, 2000, XP, 2003, Vista, 7) puesto que éstos permiten ejecutar el servidor MySQL como un servicio.

Soporte para protocolo TCP/IP.

Una copia de la distribución binaria de MySQL para Windows.

Una herramienta capaz de leer ficheros .zip, para descomprimir el fichero de distribución (en caso que se decida usar este).

Suficiente espacio en disco duro para descomprimir, instalar, y crear las bases de datos de acuerdo a sus requisitos. Generalmente se recomienda un mínimo de 200 megabytes.

También podrían necesitarse los siguientes ítems opcionales:

Si se planea conectarse al servidor MySQL a través de ODBC, se deberá contar con un driver Connector/ODBC.

Si se necesitan tablas con un tamaño superior a 4GB, debe instalarse MySQL en un sistema de ficheros NTFS o posterior. Al crear las tablas no debe olvidarse el uso de MAX_ROWS y AVG_ROW_LENGTH.

La base de datos del presente proyecto va a estar funcionando sobre Linux pero se va a usar instalación en Windows para la realización de pruebas por eso el

método de instalación a explicar va a ser el recomendado, que es con un instalador automático.

Los usuarios nuevos pueden emplear el asistente de instalación y el asistente de configuración para instalar MySQL en Windows. Éstos están diseñados para instalar y configurar de tal forma que los usuarios nuevos pueden comenzar a utilizar MySQL inmediatamente.

Los asistentes de instalación y configuración se encuentran disponibles en los paquetes Essentials y Complete, y están recomendados para la mayoría de las instalaciones estándar de MySQL. Las excepciones incluyen a usuarios que necesitan implementar múltiples instancias de MySQL en un único servidor y a usuarios avanzados que desean un control completo de la configuración del servidor.

El asistente de instalación es un instalador para el servidor MySQL que emplea las últimas tecnologías de instalador para Microsoft Windows. El Asistente de Instalación de MySQL, en combinación con el asistente de configuración, le permite a un usuario instalar y configurar un servidor MySQL que esté listo para el uso inmediatamente a continuación de la instalación.

El procedimiento para ejecutar el asistente de instalación depende del contenido del paquete descargado. Si existe un fichero setup.exe o .msi, al hacerles doble click comenzará la instalación.

Hay disponibles tres tipos de instalación: típica, completa, y personalizada.

La instalación típica instala el servidor MySQL, el cliente de línea de comandos mysql, y las utilidades de línea de comandos. Los clientes y utilidades incluyen mysqldump, myisamchk, y otras herramientas que ayudan a administrar el servidor MySQL.

La instalación completa instala todos los componentes incluidos en el paquete. El paquete completo incluye componentes como el servidor incrustado (embedded), el conjunto de pruebas de rendimiento (benchmarks), scripts de mantenimiento, y documentación.

La instalación personalizada otorga un control completo sobre los paquetes que se desea instalar y el directorio de instalación que se utilizará.

Si se escoge la instalación típica o la completa, al hacer click sobre el botón Siguiente se avanza a la pantalla de confirmación para verificar las opciones y comenzar la instalación. Si se escoge la instalación personalizada, al hacer click sobre el botón Siguiente.

Si se desea cambiar el directorio de instalación o los componentes que se instalarán, se deberá elegir el tipo de instalación personalizada.

Todos los componentes disponibles se encuentran en un diagrama de árbol en el lado izquierdo del cuadro de diálogo de instalación personalizada. Los componentes que no serán instalados tienen un icono X rojo; los componentes que se instalarán tienen un icono gris. Para indicar si un componente se instalará o no, debe hacerse click en su icono y elegir una opción de la lista desplegable que aparece.

Se puede cambiar el directorio de instalación por defecto haciendo click en el botón Cambiar... a la derecha del directorio de instalación que se muestra.

Después de elegir los componentes a instalar y el directorio de instalación, hacer click en el botón Siguiente hará avanzar al cuadro de diálogo de confirmación.

Una vez que se elige un tipo de instalación y los componentes a instalar, se avanza al cuadro de diálogo de confirmación. Se muestran el tipo y el directorio de instalación se para ser confirmados.

Para instalar MySQL una vez que se está conforme con la configuración, debe hacerse click en el botón Instalar. Para cambiar la configuración, debe hacerse click en el botón Retroceder. Para abandonar el asistente de instalación sin terminar la instalación de MySQL, debe hacerse click en el botón Cancelar.

Una vez que se hace click en el botón Instalar, el asistente de instalación MySQL comienza el proceso de instalación y realiza ciertos cambios en el sistema, que se describen en la siguiente sección.

Cambios al Registro

El asistente de instalación crea una clave de registro, durante una situación típica de instalación, localizada en:

HKEY_LOCAL_MACHINE\SOFTWARE\MySQL AB.

El asistente de instalación crea una clave cuyo nombre es el número de versión principal (número de la serie) del servidor que se encuentra instalado. Contiene dos valores de cadena, Location y Version. La cadena Location contiene el directorio de instalación. En una instalación corriente, contiene C:\Archivos de

Programa\MySQL\MySQL Server\. La cadena Version contiene el número de entrega (release). Por ejemplo, para una instalación de MySQL Server 5.1.2 la clave contiene el valor 5.1.2.

Estas claves del registro son útiles para ayudar a herramientas externas a identificar la ubicación en la que se instaló el servidor MySQL, evitando un rastreo completo del disco para descubrirla. Las claves del registro no son necesarias para la ejecución del servidor y no se crean cuando se usa el fichero Zip noinstall.

Cambios en el menú Inicio

El asistente de instalación crea una nueva entrada en el menú Inicio de Windows, bajo una opción cuyo nombre es el número de versión principal (número de la serie) del servidor que se encuentra instalado. Por ejemplo, si se instala MySQL 5.1, se crea una sección MySQL Server 5.1 en el menú Inicio.

Se crean las siguientes entradas dentro de la nueva sección del menú Inicio:

MySQL Command Line Client: Es un atajo al cliente de línea de comandos mysql y está configurado para iniciar sesión como usuario root. El atajo pregunta por una contraseña perteneciente a un usuario root cuando se conecta.

MySQL Server Instance Config Wizard: Es un atajo al asistente de configuración. Utilice este atajo para configurar un servidor recientemente instalado o reconfigurar uno existente.

MySQL Documentation: Es un vínculo a la documentación del servidor MySQL, que se almacena localmente en el directorio de instalación. Esta opción no está disponible cuando el servidor fue instalado con el paquete Essentials.

Cambios en el sistema de ficheros

El asistente de instalación, por defecto instala el servidor en C:\Archivos de Programa\MySQL\MySQL Server. Ésta es la nueva ubicación donde se recomienda instalar MySQL, en sustitución de la antigua ubicación por defecto, c:\mysql.

Por defecto, todas las aplicaciones MySQL se almacenan en un directorio común localizado en C:\Archivos de Programa\MySQL. Una instalación típica en el ordenador de un desarrollador podría verse así:

C:\Archivos de Programa\MySQL\MySQL Server 5.0

C:\ Archivos de Programa \MySQL\MySQL Administrator 1.0

C:\ Archivos de Programa \MySQL\MySQL Query Browser 1.0

Esta proximidad entre los distintos directorios facilita la administración y el mantenimiento de todas las aplicaciones MySQL instaladas en un sistema en particular.

El asistente de configuración MySQL automatiza el proceso de configurar el servidor bajo Windows. Crea un fichero my.ini personalizado, realizando una serie de preguntas y aplicando las respuestas a una plantilla para generar un fichero my.ini apropiado para la instalación en curso.

El asistente de configuración MySQL se desde la versión 5.0 de MySQL, y por el momento sólo está disponible para usuarios de Windows.

El asistente de configuración generalmente se ejecuta a continuación del asistente de instalación, ni bien éste finaliza. También puede iniciarse haciendo click en la entrada MySQL Server Instance Config Wizard de la sección MySQL del menú Inicio.

Adicionalmente, es posible dirigirse al directorio bin de la instalación MySQL y ejecutar directamente el fichero MySQLInstanceConfig.exe

Si el asistente de configuración detecta un fichero my.ini preexistente, se tiene la opción de reconfigurar el servidor o quitar la instancia borrando el fichero my.ini y deteniendo y quitando el servicio MySQL.

Para reconfigurar un servidor existente, debe escogerse la opción Re-configure Instance y hacer click en el botón Next. El fichero my.ini actual será renombrado como mytimestamp.ini.bak, donde timestamp es la fecha y hora en que el fichero my.ini existente se creó. Para quitar la instancia del servidor actual, debe seleccionarse la opción Remove Instance y hacer click en el botón Next.

Si se selecciona la opción Remove Instance, se continúa con una ventana de confirmación. Al hacer click en el botón Execute, el asistente de configuración detendrá y quitará el servicio MySQL, tras lo cual borrará el fichero my.ini. Los ficheros del servidor, incluyendo el directorio data, no se eliminarán.

Si se opta por Re-configure Instance, se continúa hacia el cuadro de diálogo Configuration Type donde puede elegirse el tipo de instalación a configurar.

Cuando se inicia el asistente de configuración MySQL para una instalación nueva o se escoge la opción Re-configure Instance para una configuración existente, se avanza hacia el cuadro de diálogo Configuration Type.

Hay disponibles dos tipos de configuración: Configuración detallada (Detailed Configuration) y Configuración estándar (Standard Configuration). La Configuración estándar está orientada a usuarios nuevos que deseen comenzar rápidamente con MySQL sin tener que tomar varias decisiones relativas a la configuración del servidor. La Configuración detallada está dirigida a usuarios avanzados que deseen un control más preciso sobre la configuración del servidor. Si se trata de un usuario nuevo de MySQL y necesita un servidor configurado para un ordenador de desarrollo con un único usuario, la Configuración estándar debería cubrir sus necesidades. Al elegir la Configuración estándar el asistente de configuración establece todas las opciones de configuración automáticamente, a excepción de Opciones de servicio (Service options) y Opciones de seguridad (Security options).

La Configuración estándar establece opciones que pueden ser incompatibles con sistemas donde existen instalaciones de MySQL previas. Si se posee una instalación anterior además de la que se está configurando, se recomienda optar por la Configuración detallada (Detailed configuration).

Hay tres tipos de servidor distintos para elegir, y el tipo que se escoja afectará a las decisiones que el asistente de configuración MySQL tomará en relación al uso de memoria, disco y procesador.

Developer machine (Ordenador de desarrollo): Esta opción se aplica a ordenadores de escritorio donde MySQL está orientado a un uso personal solamente. Se asume que se estarán ejecutando varias otras aplicaciones, por lo que el servidor MySQL se configura para utilizar una cantidad mínima de recursos del sistema.

Server machine (Servidor): Esta opción se aplica a servidores donde MySQL se ejecuta junto con otras aplicaciones de servidor como son FTP, correo electrónico, y servidores web. MySQL se configura para utilizar una cantidad moderada de recursos del sistema.

Dedicated MySQL Server Machine (Servidor MySQL dedicado): Esta opción se aplica a ordenadores donde solamente se ejecuta el servidor MySQL. Se asume que no hay otras aplicaciones ejecutándose. El servidor MySQL se configura para utilizar todos los recursos disponibles en el sistema.

El cuadro de diálogo Uso de la base de datos (Database usage) permite indicar los gestores de tablas que se planea utilizar al crear tablas de MySQL. La opción que se escoja determinará si el motor de almacenamiento InnoDB estará disponible y qué porcentaje de los recursos de servidor estarán disponibles para InnoDB

Base de datos polifuncional (Multifunctional database): Esta opción habilita tanto el motor de almacenamiento InnoDB como MyISAM y reparte los recursos uniformemente entre ambos. Se recomienda para usuarios que emplearán los dos motores de almacenamiento en forma habitual.

Base de datos transaccional exclusiva (Transactional database only): Esta opción habilita tanto el motor de almacenamiento InnoDB como MyISAM, pero destina más recursos del servidor al motor InnoDB. Se recomienda para usuarios que emplearán InnoDB casi exclusivamente, y harán un uso mínimo de MyISAM.

Base de datos no-transaccional exclusiva (Non-transactional database only): Esta opción deshabilita completamente el motor de almacenamiento InnoDB y destina todos los recursos del servidor al motor MyISAM. Recomendado para usuarios que no utilizarán InnoDB.

Algunos usuarios pueden querer ubicar los ficheros InnoDB en una ubicación diferente al directorio de datos del servidor MySQL. Esto puede ser deseable si el sistema tiene disponible un dispositivo de almacenamiento con mayor capacidad o mayor rendimiento, como un sistema conjunto redundante de discos independientes RAID (por sus siglas en inglés Redundant Array of Independent Disks).

Para modificar la ubicación por defecto de los ficheros InnoDB, debe elegirse una nueva unidad de disco en la lista desplegable de letras de unidades y elegir una nueva ruta en la lista desplegable de rutas. Haciendo click en el botón ... podrá crearse una ruta personalizada.

Si se está modificando la configuración de un servidor preexistente, debe hacerse click en el botón Modify antes de cambiar la ruta. En dicho caso habrá que desplazar manualmente los ficheros InnoDB existentes hacia la nueva ubicación antes de iniciar el servidor.

Es importante establecer un límite para las conexiones simultáneas que se podrán establecer con el servidor MySQL, para evitar que éste se quede sin recursos. El cuadro de diálogo Conexiones simultáneas (Concurrent connections) permite indicar el uso que se planea darle al servidor, y establecer en consecuencia el límite de conexiones simultáneas. También es posible introducir manualmente el límite.

Soporte de decisiones (Decision support (DSS)/OLAP): Debe escogerse esta opción si el servidor no necesitará una gran cantidad de conexiones simultáneas. El número máximo de conexiones se establece en 100, asumiéndose un promedio de 20 conexiones simultáneas.

Proceso de transacciones en línea (Online transaction processing (OLTP)): Debe escogerse esta opción si el servidor necesitará un gran número de conexiones simultáneas. El número máximo de conexiones se establece en 500.

Configuración manual (Manual setting): Debe escogerse esta opción para establecer manualmente el número máximo de conexiones simultáneas que admitirá el servidor. El número deseado puede elegirse de una lista desplegable o teclearse si no figura en ella.

El cuadro de diálogo Opciones de red (Networking options) permite activar o desactivar el protocolo TCP/IP y modificar el número de puerto por el que se accederá al servidor MySQL.

El protocolo TCP/IP está activado por defecto. Para desactivarlo debe quitarse la marca de la casilla al lado de la opción Activar TCP/IP (Enable TCP/IP networking).

Por defecto se utiliza el puerto 3306 para acceder a MySQL. Para modificar este valor, el número deseado puede elegirse de una lista desplegable o teclearse si no figura en la lista. Si el puerto indicado ya se encuentra en uso, se solicitará la confirmación de la elección.

El servidor MySQL soporta múltiples conjuntos de caracteres, y es posible establecer uno por defecto, que se aplicará a todas las tablas, columnas y bases de datos, a menos que se sustituya. Debe emplearse el cuadro de diálogo Character set para cambiar en el servidor el conjunto de caracteres por defecto.

Juego de caracteres estándar (Standard character set): Esta opción establecerá a Latin1 como el juego de caracteres por defecto en el servidor. Latin1 se usa para el inglés y muchos idiomas de Europa Occidental.

Soporte multilingüe mejorado (Best support for multilingualism): Esta opción establece a UTF8 como el conjunto de caracteres por defecto en el servidor. UTF8 puede almacenar caracteres de muchos idiomas diferentes en un único juego.

Selección manual del conjunto de caracteres por defecto / colación (Manual selected default character set / collation): Esta opción se emplea cuando se desea elegir manualmente el juego de caracteres por defecto del servidor, a través de una lista desplegable.

En plataformas basadas en Windows NT, el servidor MySQL puede instalarse como un servicio. De ese modo, se iniciará automáticamente durante el inicio del sistema, e incluso será reiniciado automáticamente por Windows en caso de producirse un fallo en el servicio.

El asistente de configuración MySQL instala por defecto el servidor MySQL como un servicio, utilizando el nombre de servicio MySQL. Si se desea evitar la instalación del servicio, debe vaciarse la casilla al lado de la opción Instalar como servicio Windows (Install as Windows service). Se puede modificar el nombre del servicio eligiendo un nuevo nombre o tecleándolo en la lista desplegable provista.

Para instalar el servidor MySQL como un servicio pero que no se ejecute al iniciarse Windows, debe vaciarse la casilla al lado de la opción Ejecutar el servidor MySQL automáticamente (Launch the MySQL server automatically).

Se recomienda fuertemente que se establezca una contraseña para el usuario root del servidor MySQL. El asistente de configuración la solicita por defecto. Si no se desea establecer una contraseña, debe vaciarse la casilla al lado de la opción Modificar configuración de seguridad (Modify security settings).

Para establecer la contraseña del usuario root, se debe introducir tanto en el cuadro de texto Nueva contraseña de root (New root password) como en Confirmar (Confirm). Si se está reconfigurando un servidor existente, también será necesario introducir la contraseña en vigencia dentro del cuadro de texto Contraseña de root actual (Current root password).

Para evitar que el usuario root inicie sesión desde cualquier punto de la red, debe marcarse la casilla al lado de la opción Root sólo puede conectarse en modo local (Root may only connect from localhost) . Esto fortalece la seguridad de la cuenta de root.

Para crear una cuenta de usuario anónimo, debe marcarse la casilla al lado de la opción Crear una cuenta de anónimo (Create An Anonymous Account). No se recomienda crear un usuario anónimo porque puede disminuir la seguridad del servidor y ocasionar dificultades de inicio de sesión y de permisos.

El último cuadro de diálogo del asistente de configuración es el de Confirmación (Confirmation dialog). Para concretar el proceso de configuración, debe hacerse click en el botón Ejecutar (Execute). Para volver a un cuadro de diálogo anterior, debe hacerse click en el botón Atrás (Back). Para abandonar el asistente de configuración sin cambiar la configuración del servidor, debe hacerse click en el botón Cancelar (Cancel).

Después de hacer click en el botón Ejecutar (Execute), el asistente de configuración llevará a cabo una serie de tareas cuyo avance se mostrará en la pantalla a medida que cada etapa termine.

El asistente de configuración determina en primer lugar las opciones del fichero de configuración, basándose en las preferencias del usuario, y empleando una plantilla confeccionada por desarrolladores e ingenieros de MySQL AB. Esta plantilla se llama my-template.ini y se localiza en el directorio de instalación del servidor.

Luego, el asistente de configuración guarda dichas opciones en el fichero my.ini. La ubicación final de este fichero se muestra al lado de la tarea Guardar fichero de configuración (Write configuration file).

Si se optó por crear un servicio de Windows para el servidor MySQL, el asistente de configuración creará e iniciará el servicio. Si se está reconfigurando un servicio existente, el asistente de configuración reiniciará el servicio para que tomen efecto los cambios realizados.

Si se optó por establecer una contraseña para el usuario root, el asistente de configuración se conectará al servidor, establecerá la nueva contraseña para root, y aplicará cualquier otra opción de seguridad que se haya seleccionado.

Después de que el asistente de configuración MySQL haya completado sus tareas, se mostrará un resumen. Haciendo click en el botón Terminar (Finish) se abandonará el asistente.

Con esto ya tenemos instalado y listo para trabajar nuestro servidor MySQL el servicio se lo puede iniciar y detener desde la ventana de Administración de Equipos escogiendo la opción de Servicios y buscando MySQL en el listado de servicios.

INSTALACIÓN DE MYSQL EN LINUX DESDE UN RPM

La manera recomendada de instalar MySQL en Linux es utilizando Administrador de paquetes de RedHat RPM (por sus siglas en inglés RedHat Packet Manager). Los RPMs de MySQL están generados en SuSE Linux 7.3,

pero deberían funcionar con cualquier versión de Linux que soporte RPM y el uso de glibc.

MySQL AB proporciona RPMs específicos para algunas plataformas; la diferencia entre un RPM específico para una plataforma y uno genérico es que el primero es generado sobre la misma plataforma a donde está destinado, y emplea enlazado dinámico, en tanto que el RPM genérico está enlazado estáticamente con LinuxThreads.

La parte "AB" del nombre de la compañía es el acrónimo del sueco "aktiebolag", o "stock company", o "sociedad anónima".

Las distribuciones RPM de MySQL a menudo están proporcionadas por otros proveedores. Hay que tener en cuenta que pueden diferir, en características y prestaciones, de aquellas generadas por MySQL AB, y que las instrucciones de instalación en este manual no se les aplican necesariamente. Se deberían consultar las instrucciones del proveedor.

En la mayoría de los casos, sólo será necesario instalar los paquetes MySQL-server y MySQL-client para conseguir una instalación de MySQL en funcionamiento. Los otros paquetes no se necesitan para una instalación estándar.

A partir de MySQL 5.1.24 no se incluye soporte para el motor de almacenamiento NDBCLUSTER motor de almacenamiento para MySQL Cluster. Los usuarios que deberán actualizar a MySQL Cluster NDB 6.2 o MySQL Cluster NDB 6.3.

Si se desea actualizar a una versión superior y la instalación la ha realizado a través de varios RPM's es conveniente que actualice todos los paquetes para no tener inconvenientes.

El diseño de instalación de MySQL para Linux RPM es el siguiente:

CUADRO NO. 1

DISEÑO DE INSTALACIÓN DE RPM MYSQL PARA LINUX

Directorio	Contenido del directorio
/usr/bin	Programas cliente y scripts.
/usr/sbin	El servidor mysqld.
/var/lib/mysql	Los archivos de registro, base de datos.
/usr/share/info	Manual en formato de información.
/usr/share/man	Manual para Unix.
/usr/include/mysql	Incluya (encabezado) archivos.
/usr/lib/mysql	Bibliotecas.
/usr/share/mysql	Archivos de ayuda varios, mensajes de error, ejemplos de los archivos de configuración, SQL para la instalación de bases de datos.
/usr/share/sql-bench	Puntos de referencia.

Elaboración: Equipo de desarrollo de MySQL.

Fuente: <http://dev.mysql.com/doc/refman/5.0/en/installation-layouts.html>

El siguiente cuadro muestra algunos RPM's necesarios para algunos sistemas operativos:

CUADRO NO. 2

RPM's NECESARIOS PARA ALGUNOS SISTEMAS OPERATIVOS

Paquete	Sistemas Operativos
glibc23	Independiente de la plataforma, debería funcionar en cualquier distribución de Linux que soporte glibc2.3
rhel3, rhel4	Red Hat Enterprise Linux 3 o 4
sles9, sles10	Suse Linux Enterprise Server 9 o 10

Elaboración: Equipo de desarrollo de MySQL.

Fuente: <http://dev.mysql.com/doc/refman/5.0/en/linux-installation-rpm.html>

El siguiente cuadro muestra el valor del CPU para el cual fue creado el RPM.

CUADRO NO. 3

VALOR DEL CPU PARA EL CUAL FUE CREADO EL RPM

Paquete	Sistemas Operativos
i386	procesador x86, 386 y hasta
i586	procesador x86, Pentium y superiores
x86_64	procesador 64 bits – x86
ia64	Procesador Itanium (IA-64)

Elaboración: Equipo de desarrollo de MySQL.

Fuente: <http://dev.mysql.com/doc/refman/5.0/en/linux-installation-rpm.html>

Para ver todos los archivos en un paquete RPM (por ejemplo, un MySQL-server RPM), ejecute un comando similar a éste:

```
shell> rpm -qpl MySQL-server- VERSION .glibc23.i386.rpm
```

Para realizar una instalación estándar mínima, instale los RPM's del servidor y del cliente

```
shell> rpm -i MySQL-server- VERSION .glibc23.i386.rpm
```

```
shell> rpm -i MySQL-client- VERSION .glibc23.i386.rpm
```

Para instalar sólo los programas cliente, instalar sólo el cliente RPM:

```
shell> rpm -i MySQL-client- VERSION .glibc23.i386.rpm
```

Si no se está seguro de que un paquete ha sido correctamente descargado se puede usar la suma de verificación MD5 por ejemplo:

```
shell> md5sum mysql-standard-5.1.53-linux-i686.tar.gz
```

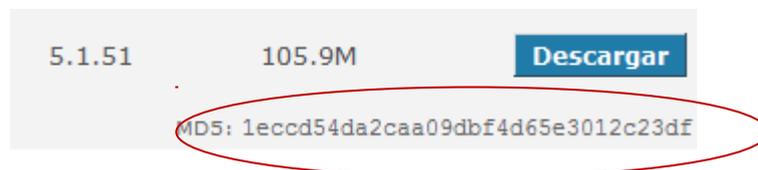
Teniendo como salida lo siguiente:

```
aaab65abbec64d5e907dcd41b8699945  mysql-standard-5.1.53-  
linux-i686.tar.gz  aaab65abbec64d5e907dcd41b8699945  mysql-  
standard-5.1.53-linux-i686.tar.gz
```

Por ejemplo:

Gráfico 1

Suma de Verificación MD5



Elaboración: Poll Miguel Cedeño Tello

Fuente: <http://dev.mysql.com/downloads/mysql/>

Se debe verificar que la suma de verificación resultante (la cadena de dígitos hexadecimales) concuerde con la que se muestra en la página de descarga inmediatamente debajo del nombre del paquete correspondiente.

Hay que tomar en cuenta que algunos sistemas operativos no soportan el comando “md5sum”; en algunos se llama solamente MD5 y otros no lo tienen.

Ya instalado MySQL hay que tomar en cuenta que los datos se guardarán en el directorio `/var/lib/mysql`. También se crea una cuenta de inicio de sesión de un usuario llamado “mysql”, si no existe a fin de ejecutar el servidor MySQL, y crea las entradas adecuadas en `/etc/init.d/` para iniciar el servidor automáticamente al arrancar el sistema.

Si desea instalar el RPM MySQL en las antiguas distribuciones de Linux que no son compatibles con scripts de inicio en `/etc/init.d` (directamente o a través de un enlace simbólico), debe crear un enlace simbólico que apunta a la ubicación en las secuencias de comandos de inicialización en realidad son instalado. Por ejemplo, en caso de que la ubicación es `/etc/rc.d/init.d` , utilice estos comandos antes de instalar el RPM para crear `/etc/init.d` como un vínculo simbólico que apunte allí:

```
shell> cd/etc
shell> ln -s rc.d/init.d .
```

Sin embargo, todas las principales distribuciones de Linux actuales deben apoyar la nueva disposición de directorios que utiliza “`/etc/init.d` ,” porque es necesaria el cumplimiento de LSB (Linux Standard Base).

Desde MySQL 5.1.49, durante una instalación de actualización utilizando los paquetes RPM, si el servidor MySQL se está ejecutando cuando se produce la actualización, el servidor MySQL se para, la actualización se produce, y el servidor MySQL se reinicia. Si el servidor MySQL no está en ejecución cuando

se produce la actualización de RPM, el servidor MySQL no se inicia al final de la instalación.

Para la gestión de usuarios no locales (LDAP, NIS, etc), las herramientas administrativas pueden requerir autenticación adicional como por ejemplo una contraseña, y si no se proporciona esta se producirá un error al momento de la instalación.

INSTALACIÓN DE MYSQL EN LINUX DESDE BINARIOS GENÉRICOS

Esta sección cubre la instalación de las distribuciones binarias de MySQL que se proveen para varias plataformas en forma de ficheros comprimidos .tar (archivos con un .tar.gz extensión).

Oracle ofrece un conjunto de distribuciones binarias de MySQL. Además de los binarios proporcionados en el paquete de formatos específicos de plataforma, que ofrecen distribuciones binarias para una serie de plataformas en forma de ficheros comprimidos .tar (.tar.gz archivos).

En Unix, un archivo binario de distribución .tar se instala de desempaquetarlo en el lugar de instalación que escoja (normalmente /usr/local/mysql) y crea los siguientes directorios en dicha ubicación:

CUADRO NO. 4

DISEÑO DE INSTALACIÓN DE MYSQL USANDO UN PAQUETE BINARIO (.TAR)

Egresos	Dólares
bin	Programas clientes y el mysqld servidor
data	Los archivos de registro, bases de datos
docs	Manual en formato de Información
man	Páginas de manual Unix
include	Incluye archivos de cabecera
lib	Bibliotecas
scripts	Mysql_install_db
share/mysql	admite archivos varios, incluidos los mensajes de error, ejemplos de archivos de configuración, para la instalación de bases de datos SQL
sql-bench	Puntos de Referencia

Elaboración: Equipo de desarrollo de MySQL.

Fuente: <http://dev.mysql.com/doc/refman/5.0/en/installation-layouts.html>

Las versiones de depuración de la [mysqld](#) binarios están disponibles como [mysqld-debug](#).

Los archivos binarios .tar tienen nombres con la forma mysql- VERSION - OS .tar.gz, donde VERSION es un número (por ejemplo, 5.1.53), y el OS indica el tipo de sistema operativo al que se destina a la distribución (por ejemplo, pc-linux-i686).

Además de estos paquetes genéricos, también se puede encontrar paquetes binarios en formatos específicos de plataforma para las plataformas seleccionadas.

Para la instalación se necesita las siguientes herramientas para instalar un archivo tar distribución binaria de MySQL:

GNU gunzip para descomprimir la distribución.

Un tar para expandir la distribución. GNU tar se sabe que funciona. Algunos sistemas operativos vienen con una versión pre instalada de tar que se sabe que tiene problemas. Por ejemplo, el tar que viene siempre con las primeras versiones de Mac OS X, SunOS 4.x, Solaris 8, Solaris 9, Solaris 10 y OpenSolaris, y HP-UX se sabe que tienen problemas con los nombres de archivo largos.

En Mac OS X, puede utilizar el programa gnutar preinstalado.

En Solaris 10 y OpenSolaris puede utilizar el gtar preinstalado.

En otros sistemas que tengan un tar deficiente, se debería instalar antes GNU tar.

Otra consideración a tomar es que si existía una versión anterior de MySQL instalada, debemos asegurarnos de que ha sido eliminada completamente del sistema para no tener inconvenientes al momento de instalar una nueva versión; incluyendo los archivos de configuración ubicados en `/etc/my.cnf` o el `/etc/mysql`.

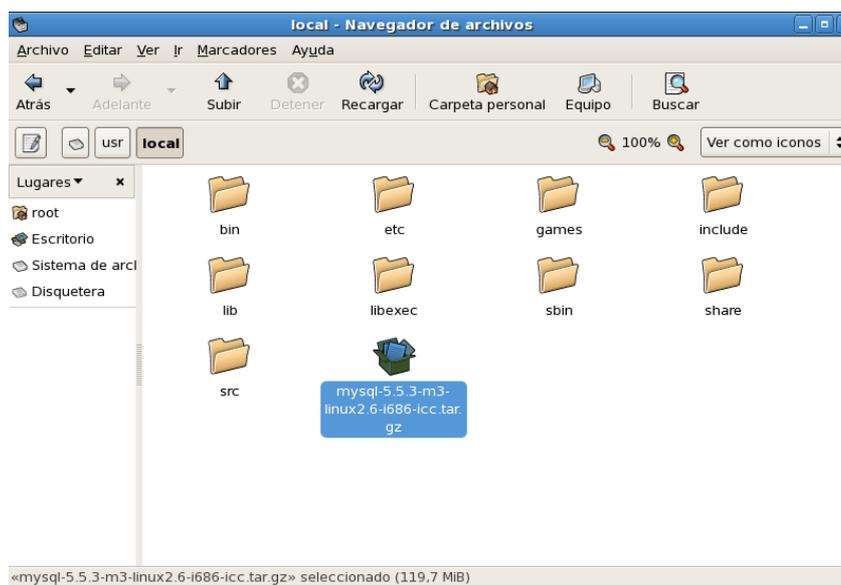
A continuación se presentará paso a paso la instalación de MySQL usando una distribución binaria, como ejemplo de la instalación se mostrará la instalación usada en el presente proyecto, usando el sistema operativo Centos 5.4.

Tomar en cuenta que el presente proceso se lo realiza como usuario root en el sistema operativo.

Primero copiamos dentro de `/usr/local` el archivo binario que decidamos usar para nuestra instalación. En este caso `mysql-5.5.3-m3-linux2.6-i686-icc.tar`

Gráfico 2

Directorio de Instalación de MySQL



Elaboración: Poll Miguel Cedeño Tello
Fuente: Centos 5.4

Primero ingresemos a la carpeta de instalación

```
shell>cd /usr/local
```

Si el sistema aún no tiene un usuario y un grupo para correr [mysqld](#), debe crear una cuenta.

```
shell> groupadd mysql
```

```
shell> useradd -g mysql mysql
```

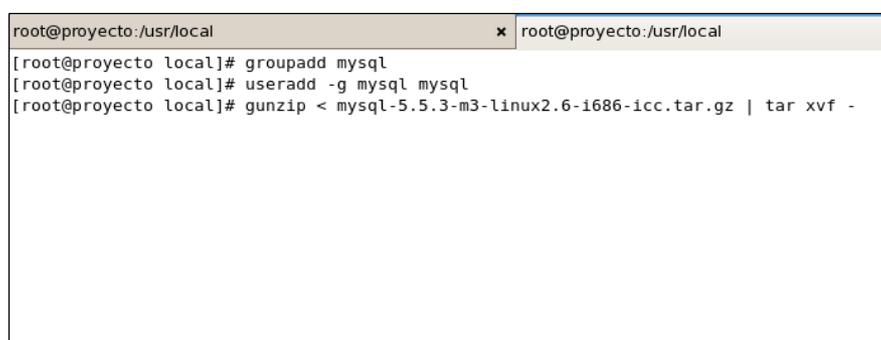
Estos comandos agregan el mysql y el grupo mysql usuario. La sintaxis para useradd y groupadd puede variar ligeramente entre las diferentes versiones de Unix, o pueden tener diferentes nombres, tales como adduser y addgroup.

Expandir la distribución, lo que crea el directorio de instalación. A continuación, cree un vínculo simbólico a ese directorio

```
shell>gunzip < /path/to/mysql-VERSION-OS.tar.gz | tar xvf -
```

Gráfico 3

Comandos previos a expandir distribución MySQL con gunzip



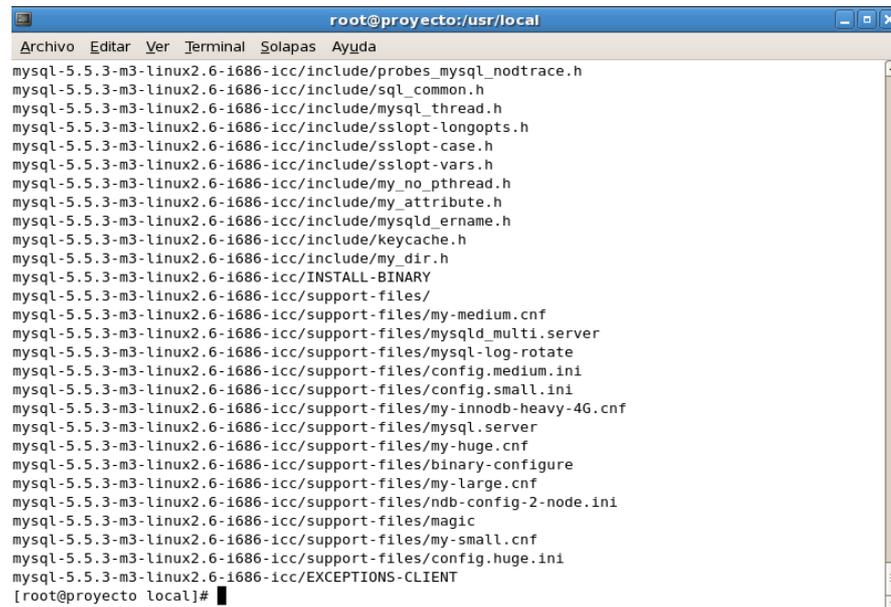
```
root@proyecto:./usr/local x root@proyecto:./usr/local
[root@proyecto local]# groupadd mysql
[root@proyecto local]# useradd -g mysql mysql
[root@proyecto local]# gunzip < mysql-5.5.3-m3-linux2.6-1686-icc.tar.gz | tar xvf -
```

Elaboración: Poll Miguel Cedeño Tello

Fuente: Centos 5.4

Gráfico 4

Distribución MySQL expandida con gunzip. Instalación satisfactoria.



```

root@proyecto:usr/local
Archivo  Editar  Ver  Terminal  Solapas  Ayuda
mysql-5.5.3-m3-linux2.6-i686-icc/include/probes_mysql_nodtrace.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/sql_common.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/mysql_thread.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/sslopt-longopts.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/sslopt-case.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/sslopt-vars.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/my_no_pthread.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/my_attribute.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/mysql_d_ename.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/keycache.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/my_dir.h
mysql-5.5.3-m3-linux2.6-i686-icc/INSTALL-BINARY
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/my-medium.cnf
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/mysqld_multi.server
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/mysql-log-rotate
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/config.medium.ini
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/config.small.ini
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/my-innodb-heavy-4G.cnf
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/mysql.server
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/my-huge.cnf
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/binary-configure
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/my-large.cnf
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/ndb-config-2-node.ini
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/magic
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/my-small.cnf
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/config.huge.ini
mysql-5.5.3-m3-linux2.6-i686-icc/EXCEPTIONS-CLIENT
[root@proyecto local]#

```

Elaboración: Poll Miguel Cedeño Tello

Fuente: Centos 5.4

A continuación, se crea un vínculo simbólico a ese directorio:

```
shell> ln -s full-path-to-mysql-VERSION-OS mysql
```

El comando tar crea un directorio denominado mysql- VERSION - OS . El comando ln crea un vínculo simbólico a ese directorio. Esto le permite referirse con mayor facilidad en el directorio de instalación /usr/local/mysql.

Posicionarse en el directorio de instalación:

```
shell> cd mysql
```

Usted encontrará varios archivos y subdirectorios en el mysql de la guía. Lo más importante para la instalación son los subdirectorios bin y scripts:

El directorio bin contiene los programas cliente y el servidor. Se debe agregar la ruta completa de este directorio al directorio de variable de entorno para que el shell encuentre los programas de MySQL apropiadamente.

El directorio scripts contiene el archivo [mysql_install_db](#) utilizado para iniciar el mysql de base de datos que contiene las tablas que almacenan los permisos de acceso al servidor.

Si ha descomprimido la distribución como el usuario mysql, no se necesitan otras medidas. Si ha descomprimido la distribución como usuario root, el contenido será propiedad de root. Cambie la propiedad a mysql ejecutando los siguientes comandos como root del directorio de instalación:

```
shell> chown -R mysql .
```

```
shell> chgrp -R mysql .
```

El primer comando cambia el atributo de propietario de los archivos al usuario mysql. El segundo cambia el atributo de grupo al grupo mysql.

Si usted no ha instalado MySQL antes, debe crear el directorio de datos MySQL e inicializar las tablas de permisos:

```
shell> scripts/mysql_install_db --user=mysql
```

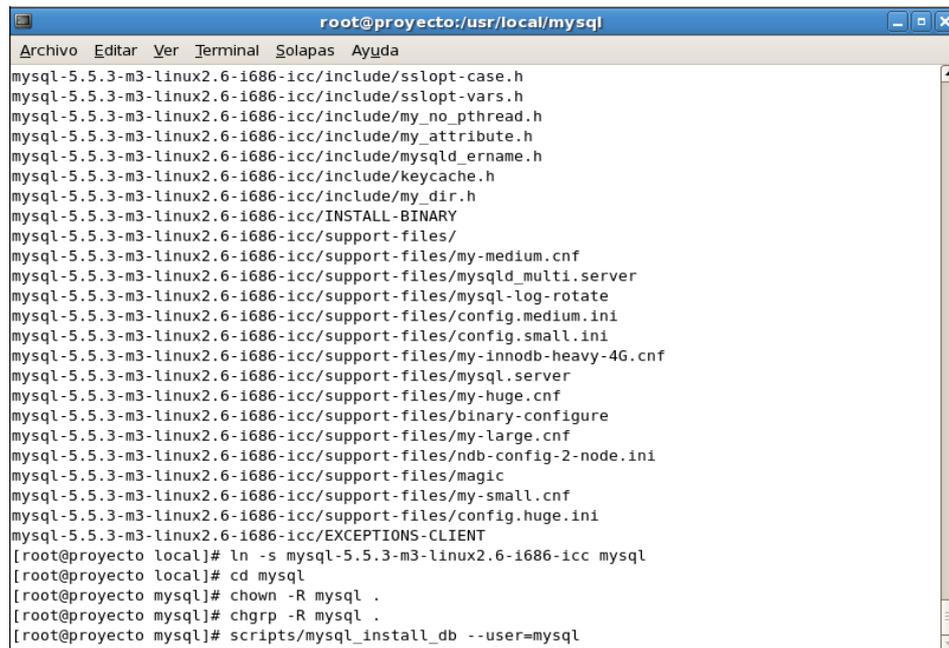
Si ejecuta el comando como root, incluir la opción [--user](#). Si se ejecuta al mismo tiempo iniciar la sesión como ese usuario, y puede omitir [--user](#).

El comando debe crear el directorio de datos y su contenido con mysql como propietario.

Después de crear o actualizar las tablas de permisos, es necesario reiniciar el servidor manualmente.

Gráfico 5

Cambiando atributos de directorio.

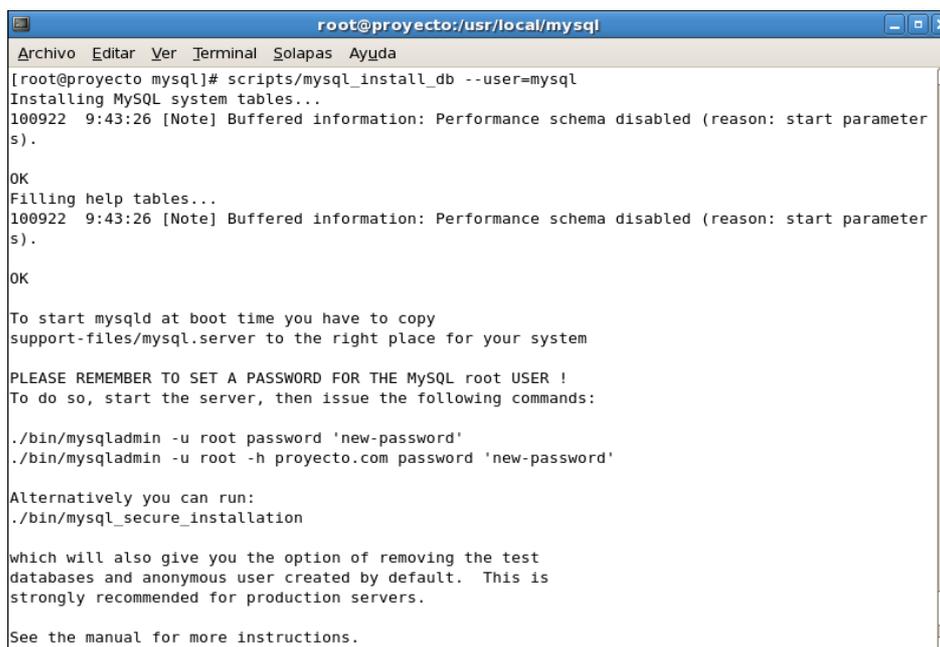


```
root@proyecto:/usr/local/mysql
Archivo Editar Ver Terminal Solapas Ayuda
mysql-5.5.3-m3-linux2.6-i686-icc/include/sslopt-case.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/sslopt-vars.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/my_no_pthread.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/my_attribute.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/mysql_d_ername.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/keycache.h
mysql-5.5.3-m3-linux2.6-i686-icc/include/my_dir.h
mysql-5.5.3-m3-linux2.6-i686-icc/INSTALL-BINARY
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/my-medium.cnf
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/mysql_multi.server
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/mysql-log-rotate
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/config.medium.ini
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/config.small.ini
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/my-innodb-heavy-4G.cnf
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/mysql.server
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/my-huge.cnf
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/binary-configure
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/my-large.cnf
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/ndb-config-2-node.ini
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/magic
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/my-small.cnf
mysql-5.5.3-m3-linux2.6-i686-icc/support-files/config.huge.ini
mysql-5.5.3-m3-linux2.6-i686-icc/EXCEPTIONS-CLIENT
[root@proyecto local]# ln -s mysql-5.5.3-m3-linux2.6-i686-icc mysql
[root@proyecto local]# cd mysql
[root@proyecto mysql]# chown -R mysql .
[root@proyecto mysql]# chgrp -R mysql .
[root@proyecto mysql]# scripts/mysql_install_db --user=mysql
```

Elaboración: Poll Miguel Cedeño Tello
Fuente: Centos 5.4

Gráfico 6

Ejecutando `mysql_install_db` para instalar tablas del sistema.



```

root@proyecto:usr/local/mysql
Archivo Editar Ver Terminal Solapas Ayuda
[root@proyecto mysql]# scripts/mysql_install_db --user=mysql
Installing MySQL system tables...
100922 9:43:26 [Note] Buffered information: Performance schema disabled (reason: start parameter
s).
OK
Filling help tables...
100922 9:43:26 [Note] Buffered information: Performance schema disabled (reason: start parameter
s).
OK

To start mysqld at boot time you have to copy
support-files/mysql.server to the right place for your system

PLEASE REMEMBER TO SET A PASSWORD FOR THE MySQL root USER !
To do so, start the server, then issue the following commands:

./bin/mysqladmin -u root password 'new-password'
./bin/mysqladmin -u root -h proyecto.com password 'new-password'

Alternatively you can run:
./bin/mysql_secure_installation

which will also give you the option of removing the test
databases and anonymous user created by default. This is
strongly recommended for production servers.

See the manual for more instructions.

```

Elaboración: Poll Miguel Cedeño Tello

Fuente: Centos 5.4

La mayor parte de la instalación de MySQL puede ser propiedad de root si quiere. La excepción es que el directorio de datos debe ser propiedad de mysql. Para lograr esto, ejecute los siguientes comandos como root del directorio de instalación:

```
shell> chown -R root .
```

```
shell> chown -R mysql data
```

Realizados todos los pasos ya podemos comenzar a probar el correcto funcionamiento del servidor.

```
shell> bin/mysqld_safe --user=mysql &
```

Si ejecuta el comando como root, debe utilizar el [--user](#) la opción como se muestra. El valor de la opción es el nombre de la cuenta de entrada que creó en el

primer paso a fin de ejecutar el servidor. Si se ejecuta el comando habiendo iniciado sesión como mysql, puede omitir el `--user`.

Si el comando falla inmediatamente y se imprime `mysqld ended`, puede encontrar información en el `host_name.err` archivo en el directorio de datos.

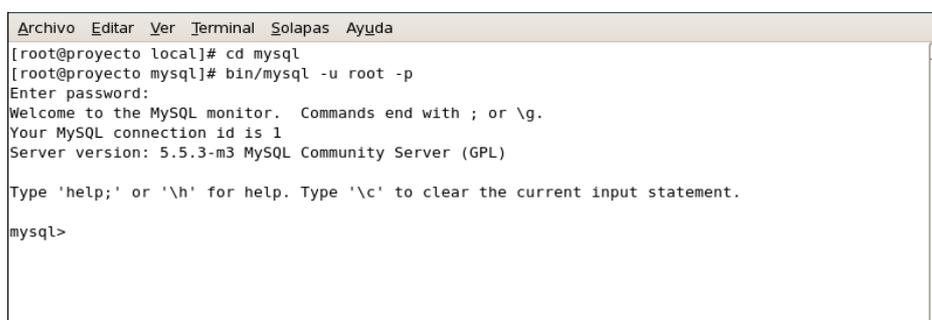
Ahora procederemos a ingresar al servidor MySQL.

```
shell> bin/mysql -u root -p
```

Ya que en este momento no le hemos asignado ninguna contraseña aún al usuario root al momento que nos pida una contraseña simplemente presionamos ENTER.

Gráfico 7

Ingresando a MySQL por primera vez



```
Archivo Editar Ver Terminal Solapas Ayuda
[root@proyecto local]# cd mysql
[root@proyecto mysql]# bin/mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.3-m3 MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql>
```

Elaboración: Poll Miguel Cedeño Tello

Fuente: Centos 5.4

Lo primero que debemos hacer es crear una contraseña para el usuario root.

```
mysql> SET PASSWORD FOR 'root'@'localhost' = PASSWORD('
newpwd ');
```

```
mysql> SET PASSWORD FOR 'root'@'127.0.0.1' = PASSWORD('
newpwd ');
```

```
mysql> SET PASSWORD FOR 'root'@' host_name ' = PASSWORD('
newpwd ');
```

O lo podemos realizar con una única instrucción:

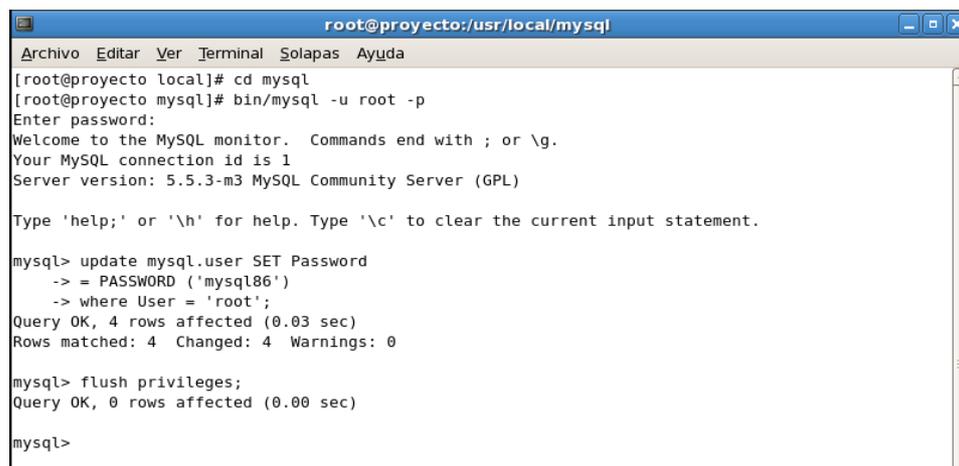
```
mysql> UPDATE mysql.user SET Password = PASSWORD('newpwd')
-> WHERE User = 'root';
```

```
mysql> FLUSH PRIVILEGES;
```

La declaración [FLUSH](#) hace que el servidor vuelva a leer las tablas de permisos. Sin ella, el cambio de contraseña pasa desapercibido por el servidor hasta que lo reinicie.

Gráfico 8

Asignando contraseña al usuario root



```
root@proyecto:/usr/local/mysql
Archivo Editar Ver Terminal Solapas Ayuda
[root@proyecto local]# cd mysql
[root@proyecto mysql]# bin/mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.5.3-m3 MySQL Community Server (GPL)

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> update mysql.user SET Password
-> = PASSWORD ('mysql86')
-> where User = 'root';
Query OK, 4 rows affected (0.03 sec)
Rows matched: 4 Changed: 4 Warnings: 0

mysql> flush privileges;
Query OK, 0 rows affected (0.00 sec)

mysql>
```

Elaboración: Poll Miguel Cedeño Tello

Fuente: Centos 5.4

Hecho esto para salir de la consola de MySQL ejecutamos el siguiente comando.

```
mysql> exit;
```

INICIAR Y DETENER EL MYSQL AUTOMÁTICAMENTE.

Ya teniendo MySQL instalado debemos conocer cómo iniciar y detener el servicio.

Para iniciar el servicio ejecutamos el siguiente comando estando dentro del directorio /usr/local/mysql

```
shell> support-files/mysql.server start
```

La ejecución del comando nos deberá devolver lo siguiente:

Gráfico 9

Iniciando el servidor MySQL

```
[root@proyecto mysql]# support-files/mysql.server start
Starting MySQL                                     [ OK ]
[root@proyecto mysql]#
```

Elaboración: Poll Miguel Cedeño Tello

Fuente: Centos 5.4

Para detener el servicio ejecutamos el siguiente comando

```
shell> support-files/mysql.server stop
```

La ejecución del comando nos deberá devolver lo siguiente:

Gráfico 10

Deteniendo el servidor MySQL

```
[root@proyecto mysql]# support-files/mysql.server stop
Shutting down MySQL..                             [ OK ]
[root@proyecto mysql]# █
```

Elaboración: Poll Miguel Cedeño Tello

Fuente: Centos 5.4

Para iniciar y detener MySQL automáticamente, se necesita agregar los comandos de inicio y detención en los sitios apropiados de los ficheros `/etc/rc*`. Si se emplea el paquete RPM de servidor para Linux (MySQL-server-VERSION.rpm), el script `mysql.server` se instala en el directorio `/etc/init.d` con el nombre `mysql`. No se necesita instalarlo manualmente. Si se instala MySQL desde una distribución de código fuente o mediante una distribución binaria que no instala `mysql.server` automáticamente, se lo puede instalar manualmente. El script se encuentra en el directorio `support-files` dentro del directorio de instalación de MySQL o en el árbol de código fuente de MySQL.

Para instalar `mysql.server` manualmente, se los debe copiar en el directorio `/etc/init.d` con el nombre `mysql`, y luego hacerlo ejecutable. Esto se hace posicionándose dentro del directorio donde está `mysql.server` y ejecutando estos comandos:

```
shell> cd support-files
shell> cp mysql.server /etc/init.d/mysql
shell> chmod +x /etc/init.d/mysql
```

Luego de instalar el script, los comandos necesarios para activarlo en el arranque del sistema dependen del sistema operativo. En Linux, puede utilizarse

`chkconfig`:

```
shell> chkconfig --add mysql
```

En algunos sistemas Linux, el siguiente comando también parece ser necesario

para habilitar completamente al script `mysql`:

```
shell> chkconfig --level 345 mysql on
```

Para terminar de este mismo directorio (`/usr/local/mysql/support-files`) copiamos uno de los archivos de ejemplo de `my.cnf` que nos va a servir para realizar configuraciones del servidor.

```
shell> cp my-medium.cnf /etc/my.cnf
```

AÑADIR CUENTAS DE USUARIO A MYSQL

Podemos crear usuarios en MySQL de dos formas: usando el comando GRANT,

o manipulando la tabla de permisos.

La tabla de permisos en MySQL generalmente lleva el nombre de 'user'.

El método preferido es usar comandos GRANT, ya que son más concisos y

menos propensos a errores.

Para lograr realizar estos ejemplos deberemos ingresar a MySQL con un usuario

que tenga privilegios como ejemplo root.

Los siguientes comandos usan GRANT para inicializar nuevas cuentas:

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'usuario'@'localhost'
```

```
-> IDENTIFIED BY 'contraseña' WITH GRANT OPTION;
```

```
mysql> GRANT ALL PRIVILEGES ON *.* TO 'usuario'@'%'
```

```
-> IDENTIFIED BY 'contraseña' WITH GRANT OPTION;
```

```
mysql> GRANT RELOAD,PROCESS ON *.* TO 'admin'@'localhost';
```

```
mysql> GRANT USAGE ON *.* TO 'prueba'@'localhost';
```

Las cuentas creadas con estos comandos GRANT tienen las siguientes

propiedades:

Dos cuentas tienen el nombre de 'usuario' y una 'contraseña'. Ambas cuentas

son cuentas de superusuario es decir tienen plenos permisos para hacer cualquier

cosa. La cuenta ('usuario'@'localhost') puede usarse sólo cuando se conecte desde

el equipo local. La otra cuenta ('usuario'@'%') puede usarse para conectarse

desde cualquier otro equipo. Es necesario tener ambas cuentas para que 'usuario'

sea capaz de conectarse desde cualquier lugar.

Una cuenta tiene un nombre de usuario de 'admin' y no tiene contraseña. Esta

cuenta puede usarse sólo desde el equipo local. Tiene los privilegios

administrativos RELOAD y PROCESS. Éstos permiten al usuario 'admin'

ejecutar los comandos mysqladmin reload, mysqladmin refresh, y mysqladmin

flush-xxx, así como mysqladmin processlist. No se dan permisos para acceder a

ninguna base de datos, pero se puede añadir tal privilegio posteriormente mediante un comando GRANT adicional.

Una cuenta tiene un nombre de usuario de ‘prueba’ sin contraseña. Esta cuenta puede usarse sólo desde el equipo local. No tiene ningún privilegio. El permiso USAGE en el comando GRANT permite crear una cuenta sin darle ningún privilegio. Tiene el efecto de inicializar todos los privilegios globales a ‘N’. Se asume que se usará para otorgar privilegios específicos posteriormente.

Como alternativa a GRANT, se pueden crear cuentas mediante el comando INSERT y después diciendo al servidor que recargue las tablas de permisos

usando el comando FLUSH PRIVILEGES; por ejemplo:

```
mysql> INSERT INTO user
-> VALUES('localhost','usuario',PASSWORD('contraseña'),
-> 'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y',
-> 'Y','Y','Y','Y');
mysql> INSERT INTO user
-> VALUES('%','usuario',PASSWORD('contraseña'),
-> 'Y','Y','Y','Y','Y','Y','Y','Y','Y','Y',
-> 'Y','Y','Y','Y');
mysql> INSERT INTO user SET Host='localhost',User='admin',
-> Reload_priv='Y', Process_priv='Y';
mysql> INSERT INTO user (Host,User>Password)
-> VALUES('localhost','prueba','');
mysql> FLUSH PRIVILEGES;
```

La razón para usar la función PASSWORD() con INSERT es encriptar las contraseñas. El comando GRANT encripta la contraseña, así que PASSWORD() no es necesario.

El valor ‘Y’ activa permisos para las cuentas. Para la cuenta ‘admin’, puede emplear la sintaxis más clara extendida INSERT usando SET.

En el comando INSERT para la cuenta ‘prueba’, sólo las columnas Host, User, y Password en el registro de la tabla user tienen valores asignados. Ninguna de las columnas de permisos se asignan explícitamente, así que MySQL les asigna a

todas el valor por defecto de 'N'. Esto es equivalente al funcionamiento de GRANT USAGE.

Para inicializar una cuenta de superusuario, sólo es necesario crear una entrada en la tabla user con las columnas de permisos inicializadas a 'Y'. Los privilegios de la tabla user son globales, así que no se necesitan registros en ninguna de las otras tablas de permisos.

Los siguientes ejemplos crean tres cuentas y les dan acceso a bases de datos específicas.

Para crear las cuentas con GRANT, use los siguientes comandos:

```
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
-> ON base.*
-> TO 'usuario'@'localhost'
-> IDENTIFIED BY 'contraseña';
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
-> ON base.*
-> TO 'usuario'@'dominio'
-> IDENTIFIED BY 'contraseña';
mysql> GRANT SELECT,INSERT,UPDATE,DELETE,CREATE,DROP
-> ON base_2.*
-> TO 'usuario'@'dominio_2'
-> IDENTIFIED BY 'contraseña';
```

Las tres cuentas pueden usarse de la siguiente manera:

La primera cuenta puede acceder a la base de datos 'base', pero sólo desde el equipo local.

La segunda cuenta puede acceder la base de datos 'base', pero sólo desde el equipo 'dominio'.

La tercera cuenta puede acceder la base de datos 'base_2', pero sólo desde el equipo "dominio_2".

Al igual que los ejemplos anteriores, para inicializar las cuentas "usuario" sin

usar GRANT, use los comandos INSERT; por ejemplo:

```
mysql> INSERT INTO user (Host,User,Password)
-> VALUES('localhost','usuario',
-> PASSWORD('contraseña'));
mysql> INSERT INTO user (Host,User,Password)
-> VALUES('dominio','usuario',
-> PASSWORD('contraseña'));
```

```

mysql> INSERT INTO user (Host,User,Password)
-> VALUES('dominio_2','usuario',
-> PASSWORD('contraseña'));
mysql> INSERT INTO db
-> (Host,Db,User,Select_priv,Insert_priv,
-> Update_priv>Delete_priv>Create_priv,Drop_priv)
-> VALUES('localhost','base','usuario',
-> 'Y','Y','Y','Y','Y','Y');
mysql> INSERT INTO db
-> (Host,Db,User,Select_priv,Insert_priv,
-> Update_priv>Delete_priv>Create_priv,Drop_priv)
-> VALUES('dominio','base','usuario',
-> 'Y','Y','Y','Y','Y','Y');
mysql> INSERT INTO db
-> (Host,Db,User,Select_priv,Insert_priv,
-> Update_priv>Delete_priv>Create_priv,Drop_priv)
-> VALUES('dominio_2','base_2','usuario',
-> 'Y','Y','Y','Y','Y','Y');
mysql> FLUSH PRIVILEGES;

```

Los primeros tres comandos INSERT añaden registros en la tabla user que permiten al usuario 'usuario' conectar desde los equipos con la contraseña dada, pero no otorga privilegios globales (todos los privilegios se inicializan al valor por defecto 'N'). Los siguientes tres comandos INSERT añaden registros en la tabla db que otorgan privilegios a 'usuario' para las bases de datos 'base' y 'base_2', pero sólo cuando se accede desde los equipos apropiados. Como siempre, cuando modifique las tablas de permisos directamente, debe decirle al servidor que las recargue con FLUSH PRIVILEGES para que los cambios en los permisos tengan efecto.

Si quiere dar a un usuario específico acceso desde todas las máquinas dentro de un dominio dado (por ejemplo, mydomain.com), puede realizar un comando GRANT que use el carácter comodín '%' en la parte del equipo del nombre de

cuenta:

```

mysql> GRANT ...
-> ON *.*
-> TO 'usuario'@'%dominio.com'

```

```
-> IDENTIFIED BY 'contraseña';
```

Para hacer lo mismo modificando las tablas de permisos directamente, haga lo

siguiente:

```
mysql> INSERT INTO user (Host,User>Password,...)
```

```
-> VALUES('%dominio.com','usuario',
```

```
-> PASSWORD('contraseña'),...);
```

```
mysql> FLUSH PRIVILEGES;
```

REPLICACIÓN EN MYSQL

Las características de MySQL soportan replicación asíncrona unidireccional: un servidor actúa como maestro y uno o más actúan como esclavos. Esto contrasta con la replicación síncrona que es una característica de MySQL Cluster. El servidor maestro escribe actualizaciones en el fichero de log binario, y mantiene un índice de los ficheros para rastrear las rotaciones de logs. Estos logs sirven como registros de actualizaciones para enviar a los servidores esclavos. Cuando un esclavo se conecta al maestro, informa al maestro de la posición hasta la que el esclavo ha leído los logs en la última actualización satisfactoria. El esclavo recibe cualquier actualización que haya tenido lugar desde entonces, y se bloquea y espera para que el master le envíe nuevas actualizaciones.

Un esclavo servidor puede servir como maestro si quiere preparar una cadena de replications de replicación.

Tenga en cuenta que cuando usa replicación, todas las actualizaciones de las tablas que se replican deben realizarse en el servidor maestro. De otro modo, debe ser cuidadoso para evitar conflictos entre actualizaciones que hacen los usuarios a las tablas en el maestro y las actualizaciones que hacen en las tablas de los esclavos.

La replicación unidireccional tiene beneficios para la robustez, velocidad, y administración del sistema:

La robustez se incrementa con un escenario maestro/esclavo. En caso de problemas con el maestro, puede cambiar al esclavo como copia de seguridad. Puede conseguirse un mejor tiempo de respuesta dividiendo la carga de consultas de clientes a procesar entre los servidores, maestro y esclavo. Se puede enviar consultas SELECT al esclavo para reducir la carga de proceso de consultas del maestro. Sin embargo, las sentencias que modifican datos deben enviarse siempre al maestro, de forma que el maestro y el esclavo no se desincronicen. Esta estrategia de balanceo de carga es efectiva si dominan consultas que no actualizan datos, pero este es el caso más habitual.

Otro beneficio de usar replicación es que puede realizar copias de seguridad usando un servidor esclavo sin molestar al maestro. El maestro continúa procesando actualizaciones mientras se realiza la copia de seguridad. En MySQL 5.5 se puede usar una interfaz para replicación semisíncrona, en la cual el maestro espera a que por lo menos un esclavo tenga la información replicada.

La replicación en MySQL se basa en un servidor maestro que toma nota de todos los cambios en las bases de datos (actualizaciones, borrados, y así) en los logs binarios. Por lo tanto, para usar replicación, debe activar el log binario en el servidor maestro.

Cada servidor esclavo recibe del maestro las actualizaciones guardadas que el maestro ha guardado en su log binario, de forma que el esclavo puede ejecutar las mismas actualizaciones en su copia de los datos.

Es extremadamente importante tener en cuenta que el log binario simplemente es un registro que comienza en un punto fijo en el tiempo en el que activa el log binario. Cualquier esclavo que inicialice necesita copias de las bases de datos del

maestro tal y como estaban en el momento en que activó el log binario en el maestro. Si arranca sus esclavos con bases de datos que no están en el mismo estado que las del maestro cuando arrancó el log binario, es muy posible que fallen sus esclavos.

Una forma de copiar los datos del maestro al esclavo es usar el comando `LOAD DATA FROM MASTER`. Tenga en cuenta que `LOAD DATA FROM MASTER` funciona sólo si todas las tablas en el maestro usan el motor de almacenamiento MyISAM. Además, este comando adquiere un bloqueo de lectura global, así que no se puede actualizar el maestro mientras las tablas se transfieren al esclavo. Cuando implementemos la copia en caliente sin bloqueo, ya no será necesario el bloqueo global de lectura.

Debido a estas limitaciones, recomendamos que en este punto use `LOAD DATA FROM MASTER` sólo si el conjunto de datos en el maestro es relativamente pequeño, o si se puede realizar un bloqueo de lectura prolongado en el maestro. Mientras que la velocidad real de `LOAD DATA FROM MASTER` puede variar de sistema a sistema, una buena regla de estimar cuánto tarda es 1 segundo por 1MB de datos. Esta es una estimación aproximada, pero puede encontrar que es bastante buena si tanto el maestro como el esclavo son equivalentes a Pentium 700MHz y conectados mediante una red a 100Mbps.

Tras inicializar el esclavo con una copia de los datos del maestro, se conecta al maestro y espera a procesar actualizaciones. Si el maestro falla, o el esclavo pierde conectividad con el maestro, el esclavo sigue intentando conectar periódicamente hasta que es capaz de reanudar la espera de actualizaciones. El

intervalo de reintento lo controla la opción `--master-connect-retry`. Por defecto es de 60 segundos.

Cada esclavo registra dónde lo dejó. El servidor maestro tiene conocimiento de cuántos esclavos hay o cuántos están activos en un momento dado.

Las capacidades de replicación de MySQL están implementadas usando tres flujos (uno en el servidor maestro y dos en el esclavo). Cuando se ejecuta un `START SLAVE`, el esclavo crea un flujo de entrada/salida, que conecta al maestro y le pide que envíe los comandos guardados en su log binario. El maestro crea un flujo para enviar los contenidos del log binario al esclavo. Este flujo puede identificarse como el flujo Binlog Dump en la salida de `SHOW PROCESSLIST` en el maestro. El flujo de entrada/salida del esclavo lee lo que el flujo Binlog Dump del maestro envía y copia estos datos en ficheros locales, llamados *logs retardados*, en el directorio de datos del esclavo. El tercer flujo es el flujo SQL, que crea el esclavo para leer los logs retardados y para ejecutar las actualizaciones que contiene.

En la descripción precedente, hay tres flujos por esclavo. Un maestro que tiene varios esclavos crea un flujo para cada esclavo conectado; cada esclavo tiene sus propios flujos de entrada/salida y SQL.

Leer dos comandos y ejecutarlos se divide en dos tareas independientes. La tarea de leer comandos no se ralentiza si la ejecución es lenta. Por ejemplo, si el servidor esclavo no ha estado en ejecución durante un periodo de tiempo, su flujo de entrada/salida puede tratar rápidamente todos los contenidos del log binario del maestro cuando arranca el esclavo, incluso si el flujo SQL se ralentiza mucho.

Si el esclavo para antes que el flujo SQL haya ejecutado todos los comandos

tratados, el flujo de entrada/salida ha tratado todo de forma que hay una copia de los comandos almacenada localmente en los logs retardados, preparados para ejecutarse la siguiente vez que arranque el esclavo. Esto permite que se purguen los logs binarios del maestro, ya que no necesita esperar que los esclavos traten sus contenidos.

El comando `SHOW PROCESSLIST` proporciona información que le dice qué está ocurriendo en el maestro y en el esclavo teniendo en cuenta la replicación. El siguiente ejemplo ilustra cómo los tres flujos se muestran en `SHOW`

`PROCESSLIST`.

En el servidor maestro, la salida de `SHOW PROCESSLIST` tiene este aspecto:

```
mysql> SHOW PROCESSLIST\G
*****1.row*****
Id: 2
User: root
Host: localhost:32931
db: NULL
Command: Binlog Dump
Time: 94
State: Has sent all binlog to slave; waiting for binlog to be
updated
Info: NULL
```

Aquí, el flujo 2 es un flujo de replicación para un esclavo conectado. La información indica que todas las actualizaciones se han enviado al esclavo y que

el maestro está esperando más actualizaciones.

En el servidor esclavo, la salida para `SHOW PROCESSLIST` tiene este aspecto:

```
mysql> SHOW PROCESSLIST\G
*****1.row*****
Id: 10
User: system user
Host:
db: NULL
Command: Connect
Time: 11
State: Waiting for master to send event
```

```

Info: NULL
*****2.row*****
Id: 11
User: system user
Host:
db: NULL
Command: Connect
Time: 11
State: Has read all relay log; waiting for the slave I/O
thread to update it
Info: NULL

```

Esta información indica que el flujo 10 es el flujo de entrada/salida que se comunica con el maestro, y el flujo 11 es el flujo SQL que procesa las actualizaciones almacenadas en los logs retardados. Cuando se ejecuta SHOW

PROCESSLIST ambos flujos estaban en espera de actualizaciones nuevas.

Tenga en cuenta que los valores en la columna Time pueden mostrar la diferencia

de tiempo en la comparación del esclavo con el maestro.

Aquí hay una breve descripción de cómo inicializar una replicación completa de

su servidor MySQL. Asume que quiere replicar todas las bases de datos en el

maestro y no tiene una replicación previamente configurada. Necesita parar el

servidor maestro brevemente para completar los pasos descritos aquí.

Este procedimiento está escrito en términos de inicializar un esclavo único, pero

puede usarlo para inicializar múltiples esclavos.

Mientras este método es el más directo para inicializar un esclavo, no es el único.

Por ejemplo, si tiene una muestra de los datos del maestro, y el maestro tiene su

ID de servidor y el log binario activo, puede preparar un esclavo sin parar el

maestro o incluso sin bloquear actualizaciones para ello.

Hay que tomar en cuenta algunos detalles antes de iniciar la replicación:

Asegúrese de que las versiones de MySQL instalado en el maestro y en el

esclavo son compatibles.

Prepare una cuenta en el maestro que pueda usar el esclavo para conectar. Esta

cuenta debe tener el privilegio REPLICATION SLAVE. Si la cuenta se usa sólo

para replicación (lo que se recomienda), no necesita dar ningún privilegio adicional.

Suponga que su dominio es mydomain.com y que quiere crear una cuenta con un nombre de usuario “repl” que puedan usar los esclavos para acceder al maestro desde cualquier equipo en su dominio usando una contraseña de “slavepass”.

Para crear la cuenta, use el comando GRANT:

```
mysql> GRANT REPLICATION SLAVE ON *.*
      ->TO 'repl'@'%mydomain.com' IDENTIFIED BY 'slavepass';
```

Si quiere usar los comandos LOAD TABLE FROM MASTER o LOAD DATA FROM MASTER desde el servidor esclavo, necesita dar a esta cuenta privilegios adicionales:

- De a la cuenta el privilegio global SUPER y RELOAD.
- De el privilegio SELECT para todas las tablas que quiere cargar.

Cualquier tabla maestra desde la que la cuenta no puede hacer un

SELECT se ignoran por LOAD DATA FROM MASTER.

Si usa sólo tablas MyISAM, vuelque todas las tablas y bloquee los comandos de

escritura ejecutando un comando FLUSH TABLES WITH READ LOCK:

```
mysql> FLUSH TABLES WITH READ LOCK;
```

Deje el cliente en ejecución desde el que lanza el comando FLUSH TABLES para que pueda leer los efectos del bloqueo. (Si sale del cliente, el bloqueo se

libera.) Luego tome una muestra de los datos de su servidor maestro.

La forma más fácil de crear una muestra es usar un programa de archivo para crear una copia de seguridad binaria de las bases de datos en su directorio de datos del maestro. Por ejemplo: use tar en Unix, o PowerArchiver, WinRAR, WinZip, o cualquier software similar en Windows. Para usar tar para crear un archivo que incluya todas las bases de datos, cambie la localización en el directorio de datos del maestro, luego ejecute el comando:

```
shell> tar -cvf /tmp/mysql-snapshot.tar .
```

Si quiere que el archivo sólo incluya una base de datos llamada `this_db`, use este

comando:

```
shell> tar -cvf /tmp/mysql-snapshot.tar ./this_db
```

Luego copie el archivo en el directorio `/tmp` del servidor esclavo. En esa máquina, cambie la localización al directorio de datos del esclavo, y

desempaquete el fichero usando este comando:

```
shell> tar -xvf /tmp/mysql-snapshot.tar
```

Puede no querer replicar la base de datos mysql si el servidor esclavo tiene un conjunto distinto de cuentas de usuario a la existente en el maestro. En tal caso, debe excluirla del archivo. Tampoco necesita incluir ningún fichero de log en el archivo, o los ficheros `master.info` o `relay-log.info` files.

Mientras el bloqueo de `FLUSH TABLES WITH READ LOCK` está en efecto, lee

el valor del nombre y el desplazamiento del log binario actual en el maestro:

```
mysql > SHOW MASTER STATUS;
```

File	Position	Binlog_Do_DB	Binlog_Ignore_DB
mysql-bin.003	73	test	manual,mysql

La columna `File` muestra el nombre del log, mientras que `Position` muestra el desplazamiento. En este ejemplo, el valor del log binario es `mysql-bin.003` y el desplazamiento es `73`. Guarde los valores. Los necesitará más tarde cuando inicialice el servidor. Estos representan las coordenadas de la replicación en que el esclavo debe comenzar a procesar nuevas actualizaciones del maestro.

Una vez que tiene los datos y ha guardado el nombre y desplazamiento del log,

puede reanudar la actividad de escritura en el maestro:

```
mysql> UNLOCK TABLES;
```

Si está usando tablas InnoDB, debería usar la herramienta InnoDB Hot Backup.

Realiza una copia consistente sin bloquear el servidor maestro, y guarda el nombre y desplazamiento del log que se corresponden a la copia para usarlo posteriormente en el esclavo.

Sin la herramienta Hot Backup , la forma más rápida de hacer una copia binaria de los datos de las tablas InnoDB es parar el maestro y copiar los ficheros de datos InnoDB, ficheros de log, y ficheros de definición de tablas (ficheros .frm).

Para guardar los nombres de los ficheros actuales y desplazamientos, debe

ejecutar el siguiente comando antes de parar el servidor:

```
mysql> FLUSH TABLES WITH READ LOCK;
```

```
mysql> SHOW MASTER STATUS;
```

Luego guarde el nombre del log y el desplazamiento desde la salida de SHOW

MASTER STATUS como se mostró antes. Tras guardar el nombre del log y el

desplazamiento, pare el servidor sin bloquear las tablas para asegurarse que el

servidor para con el conjunto de datos correspondiente al fichero de log

correspondiente y desplazamiento:

```
shell> mysqladmin -u root shutdown
```

Si el maestro se ha ejecutado previamente sin habilitar --log-bin , el nombre del

log y las posiciones mostradas por SHOW MASTER STATUS o mysqldump

--master-data están vacíos. En ese caso, los valores que necesita usar

posteriormente cuando especifica el fichero de log del esclavo y la posición son

una cadena vacía (") y 4.

Asegúrese que la sección [mysqld] del fichero my.cnf en el maestro incluye una

opción log-bin. Esta sección debe también tener la opción server-id=master_id,

donde master_id debe ser un entero positivo de 1 a $2^{32} - 1$. Por ejemplo:

```
[mysqld]
```

```
log-bin=mysql-bin
```

```
server-id=1
```

Si estas opciones no están presentes, añádalas y reinicie el servidor.

Pare el servidor que se vaya a usar como esclavo y añada lo siguiente a su fichero

```
my.cnf :
```

```
[mysqld]
```

```
server-id=slave_id
```

El valor `slave_id`, como el valor `master_id`, debe ser un entero positivo de 1 a $2^{32} - 1$. Además, es muy importante que el ID del esclavo sea diferente del ID

del maestro. Por ejemplo:

```
[mysqld]
```

```
server-id=2
```

Si está preparando varios esclavos, cada uno debe tener un valor de `server-id` único que difiera del maestro y de cada uno de los otros esclavos. Piense en los valores de `server-id` como algo similar a las direcciones IP: estos IDs identifican unívocamente cada instancia de servidor en la comunidad de replicación.

Si no especifica un `server-id`, se usa 1 si no ha definido un `master-host`, de otro modo se usa 2. Tenga en cuenta que en caso de omisión de `server-id`, un maestro rechaza conexiones de todos los esclavos, y un esclavo rechaza conectar a un maestro. Por lo tanto, omitir el `server-id` es bueno sólo para copias de seguridad con un log binario.

Si ha hecho una copia de seguridad binaria de los datos del maestro, cópielo en el directorio de datos del esclavo antes de arrancar el esclavo. Asegúrese que los privilegios en los ficheros y directorios son correctos. El usuario que ejecuta el servidor MySQL debe ser capaz de leer y escribir los ficheros, como en el maestro.

Si hizo una copia de seguridad usando `mysqldump`, arranque primero el esclavo realice los siguientes pasos:

Arranque el esclavo. Si ha estado replicando previamente, arranque el esclavo con la opción `--skip-slave-start` para que no intente conectar inmediatamente al maestro.

Si hace una copia de seguridad de los datos del maestro usando `mysqldump`,

cargue el fichero de volcado en el esclavo:

```
shell> mysql -u root -p < dump_file.sql
```

Ejecute los siguientes comandos en el esclavo, reemplazando los valores de

opciones con los valores relevantes para su sistema:

```
mysql> CHANGE MASTER TO
->     MASTER_HOST='master_host_name',
->     MASTER_USER='replication_user_name',
->     MASTER_PASSWORD='replication_password',
->     MASTER_LOG_FILE='recorded_log_file_name',
->     MASTER_LOG_POS=recorded_log_position;
```

Con esto solo nos falta iniciar el flujo en el esclavo:

```
mysql> START SLAVE;
```

MySQL ofrece una variedad de soluciones en lo que respecta a replicación las

cuales son aplicadas para diferentes ambientes por ejemplo:

- Usar replicación para sacar respaldos.
- Usar replicación con diferentes motores de almacenamiento en los esclavos como en los maestros.
- Usar la replicación para escalar.
- Replicación de diferentes bases de datos a diferentes esclavos.
- Mejorar el rendimiento de la replicación
- Intercambiar roles entre Maestro y Esclavo en el caso de algún fallo.
- Replicación Maestro – Maestro
- Configuración de la replicación usando SSL.
- Replicación Semisíncrona

Replicación semi-síncrona.

El problema más grave de la replicación en MySQL es su funcionamiento asíncrono. Cuando se añade o modifica algún dato en el maestro, este ejecuta los datos en local sin esperar a que los esclavos lo hagan. Esto normalmente no supone un gran problema, ya que la replicación, si no hay ningún problema con índices o con la red, es casi instantánea. Pero aún así se pueden dar algunos problemas:

- El maestro ejecuta los datos sin esperar. Durante un tiempo, aunque pequeño, maestro y esclavo tendrán datos diferentes. Contra mas alto sea el valor que demore la replicación, mayor será el problema.
- El maestro no comprueba que los esclavos hayan recibido los logs binarios con los cambios.
- El maestro no comprueba que los esclavos hayan hecho efectivos los cambios en sus bases de datos.

Este es un problema solucionado en MySQL Cluster, donde la replicación es totalmente síncrona. Los nodos no ejecutan (commit) los cambios hasta que estos se hayan escrito correctamente en los grupos de nodos que correspondan. Si esto no es así, se hace una reversión (rollback). Pero en la replicación normal no tenemos tanta suerte.

Una de las novedades de MySQL 5.5 viene a medio solventar el problema. Con esta nueva versión de desarrollo disponemos de replicación semi-síncrona. Su funcionamiento es simple. Ahora el maestro antes de hacer ejecutar espera a que al menos uno de los esclavos reciba los logs binarios. Pero aún así hay que tener en cuenta lo siguiente:

El maestro solamente comprueba que un esclavo haya recibido los logs, pero no que si lo ha podido escribir correctamente o no. Esto es, no importa el estado del SQL Thread.

Podemos tener 1000 esclavos, pero con que solo uno reciba los logs ya se da por bueno.

Si pasado un tiempo ninguno de los esclavos recibe los logs, el maestro cambia a modo asíncrono ejecutando los cambios.

Para implementar la replicación semi-síncrona debemos tener en cuenta lo siguiente:

- Obtener una copia para instalación de MySQL 5.5 o superior.

- La capacidad de MySQL de soportar carga dinámica para la instalación del Plug-in que permite la replicación semi-síncrona.
- La replicación debe estar funcionando.

Para llevar a cabo la replicación semi-síncrona hay que seguir los siguientes pasos: instalar el Plug-in, Activar variables globales, Detener el esclavo, Iniciar el esclavo.

El Plug-in a instalar se encuentra en las distribuciones de MySQL.

En el maestro se debe hacer lo siguiente:

```
mysql> INSTALL PLUGIN rpl_semi_sync_master SONAME
'semisync_master.so';
```

Por cada esclavo hacemos lo siguiente:

```
mysql> INSTALL PLUGIN rpl_semi_sync_slave SONAME
'semisync_slave.so';
```

Si deseamos ver que Plug-in' se encuentran instalados debemos

ejecutar el siguiente comando

```
mysql> SHOW PLUGINS
```

Con el Plug-in instalado debemos tomar en cuenta que normalmente viene

deshabilitado, se lo debe habilitar tanto en el maestro como en el esclavo.

En el maestro:

```
mysql> SET GLOBAL rpl_semi_sync_master_enabled = {0|1};
mysql> SET GLOBAL rpl_semi_sync_master_timeout = N;
```

En el esclavo:

```
mysql> SET GLOBAL rpl_semi_sync_slave_enabled = {0|1};
```

Para estas variables el valor de 1 es es habilitado y el de 0 es deshabilitado.

Si se inicializa esta replicación mientras el esclavo está en replicación también

debe inicializarse el hilo de entrada y salida de este.

En el esclavo:

```
mysql> STOP SLAVE IO_THREAD; START SLAVE IO_THREAD;
```

Al iniciar el servidor las variables de la replicación se las puede establecer como

líneas de opciones de comando en un fichero por ejemplo en el my.cnf.

En el maestro:

```
[mysqld]
```

```

rpl_semi_sync_master_enabled = 1
rpl_semi_sync_master_timeout=1000 # 1 second
En el esclavo
[mysqld]
rpl_semi_sync_slave_enabled = 1

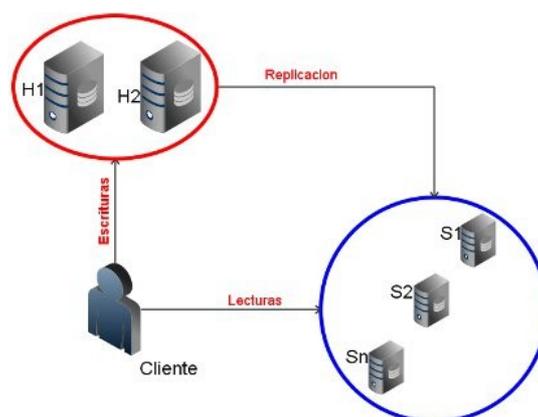
```

Replicación Maestro – Maestro

La arquitectura Maestro/Maestro es muy sencilla tanto de entender como implementar. Cuando vimos anteriormente Maestro/Esclavo, vimos que el Maestro se utilizaba para escrituras, mientras que en lecturas teníamos N servidores. En ese caso la lectura no es problema, hay suficiente hardware procesando peticiones, pero ¿qué pasa con la escritura?. Según el número de usuarios aumente y la carga de escrituras sea mayor, dicho servidor terminará por no dar a basto ralentizando el buen funcionamiento de nuestras aplicaciones. Maestro/Maestro viene a solucionarnos este problema.

Gráfico 11

Representación ambiente de Replicación



Elaboración: Poll Miguel Cedeño Tello
Fuente: Visual C++ 2008 Express Edition

En este caso, H1 y H2 reciben las peticiones de escritura. Los dos deben tener los datos sincronizados, para ello se sigue el siguiente esquema:

- H1 es maestro de H2 (por lo tanto H2 esclavo de H1)
- H2 es maestro de H1 (por lo tanto H1 es esclavo de H2)

De esta forma, todo lo escrito en H1 se replicará a H2 y viceversa.

EL CONECTOR ODBC DE MYSQL

El MySQL Connector / ODBC es el nombre de la familia de controladores ODBC de MySQL (antes llamada controladores MyODBC) que proporcionan acceso a una base de datos MySQL utilizando el estándar de la industria Open Database Connectivity (ODBC).

ODBC (Open Database Connectivity) proporciona una forma para que los programas cliente para acceder a una amplia gama de bases de datos o fuentes de datos. ODBC es un estándar API que permite conexiones a los servidores de base de datos SQL.

Ha sido desarrollado según las especificaciones del grupo de acceso de SQL y define un conjunto de llamadas a funciones, los códigos de error, y tipos de datos que se pueden utilizar para desarrollar aplicaciones de bases de datos independientes. ODBC normalmente se utiliza cuando la independencia de base de datos o el acceso simultáneo a diferentes fuentes de datos se requiere.

CUADRO NO. 5

PARÁMETROS DE CONEXIÓN DEL ODBC

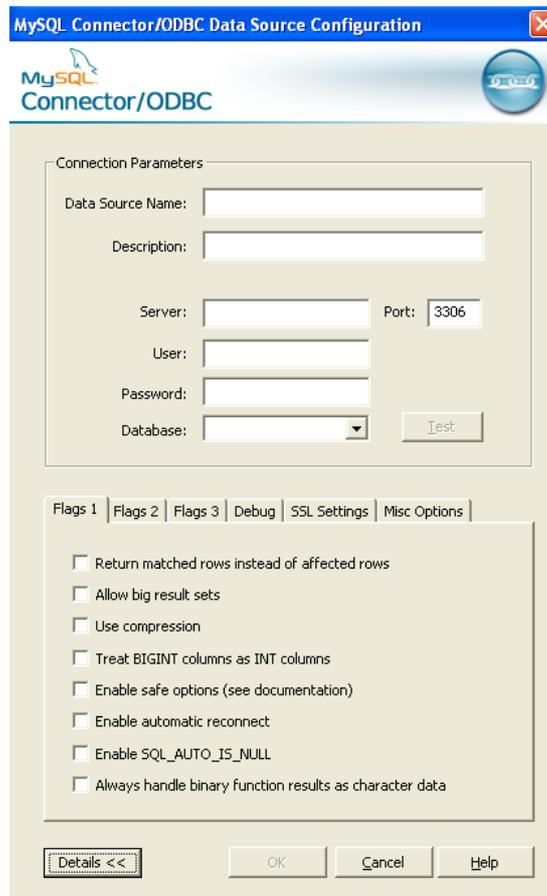
Parámetro	Valor Predeterminado	Funcionalidad
DSN		Nombre único del Data Source

		Name
Description		Descripción, es opcional
Server	localhost	La IP o host del servidor
User		El usuario con que nos vamos a conectar a la base
Password		Contraseña del usuario nos va a servir para conectarnos a la base
Database		La base de datos a la que nos vamos a conectar
Details		Opciones extras que podemos usar para conectarnos a la base, son opcionales

Fuente: Instalador del ODBC MySQL
Autor: Equipo de desarrollo de MySQL

Gráfico 12

Configuración del conector ODBC



Elaboración: Poll Miguel Cedeño Tello
Fuente: Ventana de configuración del ODBC

DSN (Data Source Name)

DSN es una palabra abreviada del inglés (Data Source Name) (en español, Nombre Fuente de datos o Nombre de origen de datos), que representa todo lo relativo a una fuente de datos configurada por el usuario para conectarse a una [Base de datos](#). Es decir, por cada conexión que el usuario quiera establecer con algún(os) fabricante(s), tiene que especificar una serie de información que permitan al [Controlador](#) o [Driver](#) saber con qué fabricante(s) se tiene que conectar y la cadena de conexión que tiene que enviarle a dicho fabricante(s)

para establecer la conexión con la fuente de datos [ODBC](#) accedida por el proveedor en cuestión.

Instalación del ODBC

Puede instalar el Conector / controladores ODBC utilizando dos métodos diferentes, una instalación binaria y una fuente de instalación.

Instalación desde binarios.

La instalación binaria es el método más sencillo y más fácil de la instalación.

Utilizando los métodos de instalación desde el código fuente sólo debería ser necesario en las plataformas donde un paquete de instalación binario no está disponible, o en situaciones en las que desea personalizar o modificar el proceso de instalación o del conector antes de la instalación.

Para Windows existe el instalador MSI mediante el cual solo con darle doble clic podemos comenzar con la instalación.

En cambio en Linux existen los binarios los cuales con los administradores de paquetes como el RPM podemos instalar sin problemas el conector.

Instalación desde el código fuente para Windows.

Para instalar el conector se debe tener las siguientes herramientas

- El código fuente del conector
- Visual Studio 7 o superior
- CMAKE 2.4 o superior (se lo puede descargar de www.cmake.org)
- El API de C de MySQL

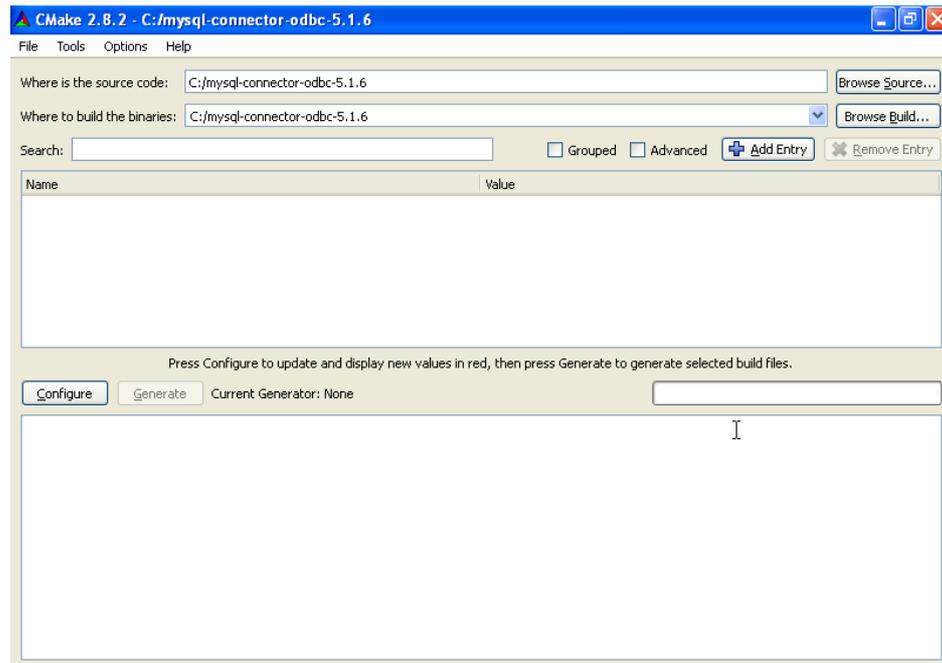
Para antes de compilar el conector debemos haber instalado el API de C, y luego de instalar este también el CMAKE que actualmente funciona a nivel gráfico.

Con el CMAKE se crea el proyecto dependiendo del compilador que tengamos en este caso Visual Studio 2008

En un directorio descomprimimos las fuentes de mysql usando el WinZip o el WinRar.

Ahora procedemos a configurar el CMAKE

Gráfico 13
Configuración de CMAKE

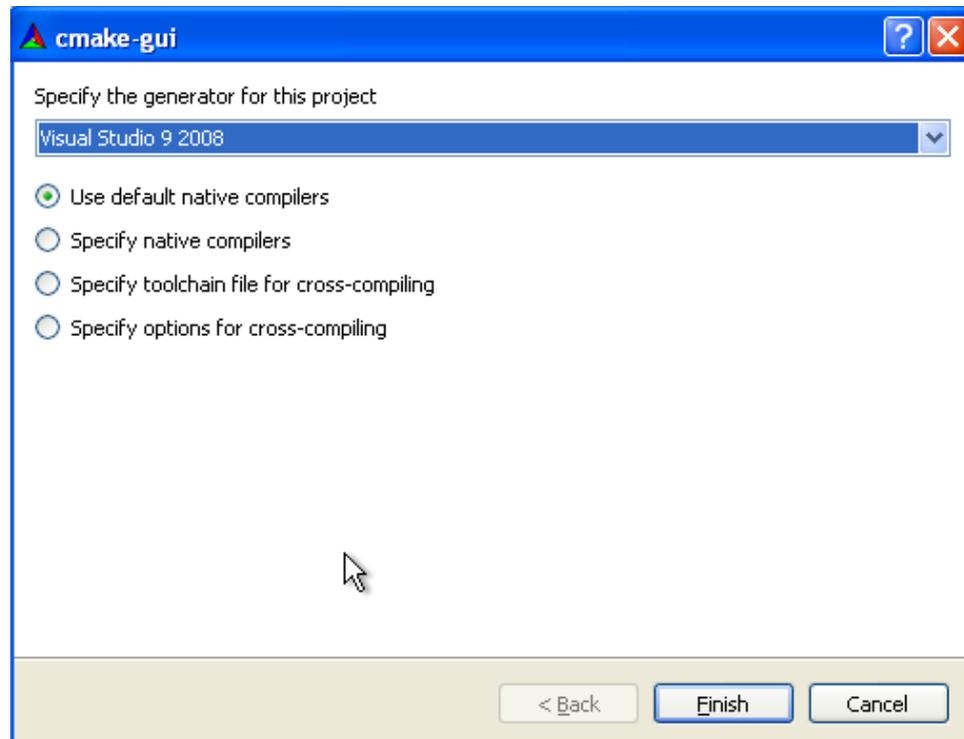


Elaboración: Poll Miguel Cedeño Tello
Fuente: CMAKE 2.8.2 para Windows

En “Where is the source code” ponemos la ubicación de donde descomprimos el código fuente
En “Where to build the binaries” ponemos la ubicación de donde descomprimos el código fuente. Podemos poner otra ubicación pero suele dar problemas al momento de la instalación.
Luego damos clic en el botón configure y nos aparecerá una ventana donde escogeremos el compilador a usar.

Gráfico 14

Selección del compilador



Elaboración: Poll Miguel Cedeño Tello

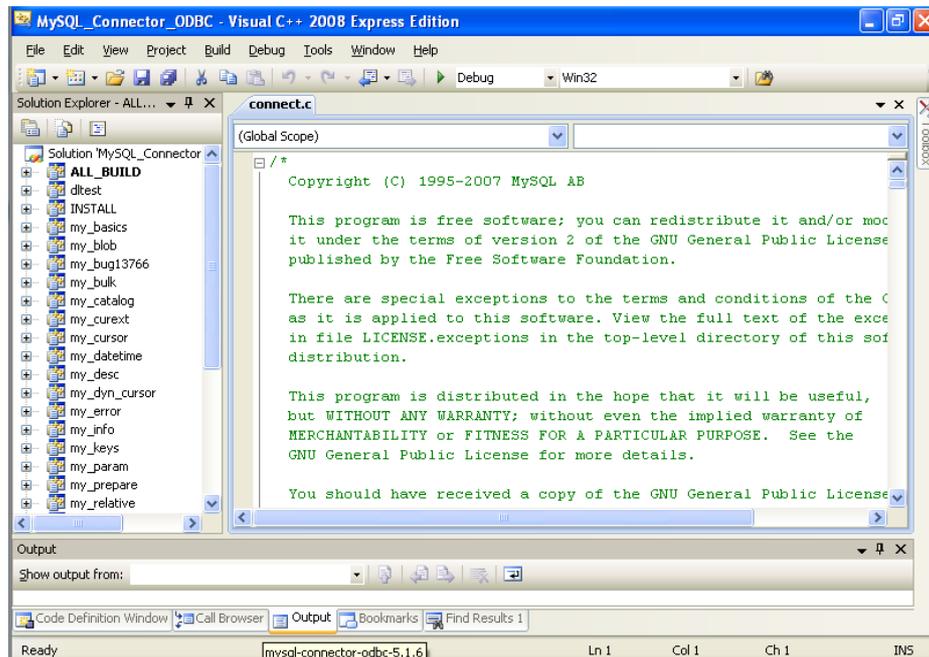
Fuente: CMAKE 2.8.2 para Windows

Luego damos clic en “Finish” en estos momentos el programa buscará automáticamente todas las librerías que necesita; si no encuentra alguna nos dará un mensaje de error.

Si todo está correcto procedemos a dar clic en el botón “Generate” con esto se nos crearán los archivos necesarios para abrir las librerías con el compilador que hayamos seleccionado. Los archivos aparecerán dentro de la carpeta que seleccionamos en “Where to build the binaries”.

Gráfico 15

Fuentes del ODBC en Visual Studio



Elaboración: Poll Miguel Cedeño Tello
Fuente: Visual C++ 2008 Express Edition

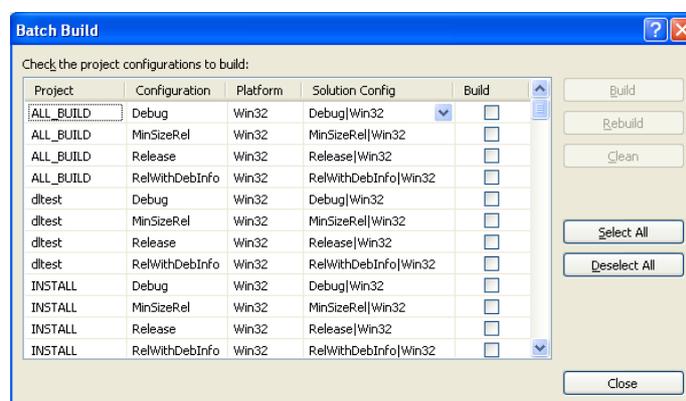
Ahora debemos compilar con las siguientes configuraciones:

- Build Release
- Build RelWithDebInfo

Estas podemos encontrar en el menú Build→Batch Build.

Gráfico 16

Batch Build para compilar fuentes



Elaboración: Poll Miguel Cedeño Tello
Fuente: Visual C++ 2008 Express Edition

Ahora desde en la barra de herramientas del Escritorio de Windows nos vamos a Inicio → Ejecutar y escribimos “cmd” para abrir la terminal de Windows, nos ubicamos en el directorio donde se encuentran los fuentes y escribimos el siguiente comando.

```
cmake -G "Visual Studio 9 2008" -DCMAKE_BUILD_TYPE=Debug
```

En este caso se está usando el compilador Visual Studio 9 2008, si se usa otro solo se cambia el nombre del compilador.

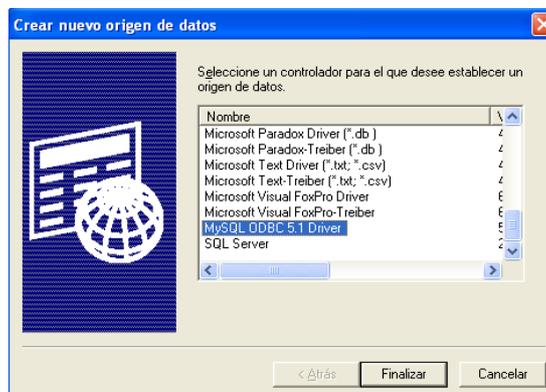
Por último regresamos a la ventana de Batch Build y escogemos la opción Debug.

Con esto generamos los .dll y los .lib necesarios para el uso del ODBC.

Dentro de la carpeta del código fuente encontraremos un archivo de nombre “Install.bat” lo ejecutamos dándole doble click y si todo se ha hecho correctamente nos copiará los drivers a la carpeta system32 de y los registrará en el sistema. Cuando vayamos a crear un DSN aparecerá en el listado con los demás drivers.

Gráfico 17

Driver ODBC listo para usarse



Elaboración: Poll Miguel Cedeño Tello
Fuente: Windows XP

Instalación desde el código fuente para Unix.

Para instalar el conector se debe tener las siguientes herramientas

- UnixODBC (se lo puede descargar desde www.unixodbc.org)
- Un compilador para C en este caso el GNU tool-chain
- El código fuente
- Opcionalmente podemos tener el ODBC gráficamente usando Qt class library (se la puede descargar desde www.trolltech.com)

Ahora solo debemos compilar nos ubicamos dentro del directorio donde tenemos los fuentes y ejecutamos los siguientes comandos.

```
shell> make
```

```
shell> make install
```

Ahora solo queda registrar el ODBC en los registros del sistema operativo esto va a depender de a versión que estemos utilizando.

2.3. FUNDAMENTACIÓN LEGAL

Art. 349.- El sistema de educación superior tiene como finalidad la formación académica y profesional con visión científica y humanística, la investigación científica y tecnológica; la innovación, promoción, desarrollo y difusión de los saberes y las culturas, la construcción de soluciones para los problemas del país, en relación con los objetivos del régimen de desarrollo.

Art. 387.- El Estado destinará los recursos necesarios para la investigación científica, el desarrollo tecnológico, la innovación, la formación científica, la recuperación y desarrollo de saberes ancestrales y la difusión del conocimiento.

Un porcentaje de estos recursos se destinara a financiar proyectos mediante fondos concursales.

2.4. HIPÓTESIS PREGUNTAS A CONTESTARSE

¿Qué tan factible es extraer la información generada desde los clientes?
¿Qué metodología debería usar para obtener la información generada por los clientes?
¿Cómo comunicar el cliente con el servidor para ejecutar las sentencias de los clientes?

2.5. VARIABLES DE LA INVESTIGACIÓN

Configuraciones para tener disponibilidad.- Configuraciones que se han llevado a cabo para asegurar la disponibilidad de información
Replicación.- Configuraciones que se usa para la replicación de información entre bases de datos.
Forma de conexión entre cliente y servidor.- Que herramientas se usan para la comunicación entre los clientes y el servidor.

2.6. DEFINICIONES CONCEPTUALES

Configuraciones para tener disponibilidad.- Es el hardware o software usado en una empresa para mantener la disponibilidad de información.

Replicación.- Las configuraciones que tiene la empresa en el caso de que use algún método de replicación

Forma de conexión entre cliente y servidor.- Por medio de que herramientas se comunican el cliente y el servidor.

GLOSARIO

GNU Automake.- Es una herramienta de programación que produce programas makefiles (Los makefiles son los ficheros de texto que utiliza make para llevar la gestión de la compilación de programas) portables para el uso de make (make es una herramienta de generación o automatización de código, muy usada en los sistemas operativos tipo Unix/Linux. Por defecto lee las instrucciones para generar el

programa u otra acción del fichero makefile.) usado en la compilación de software. Es mantenido por la Fundación de Software Libre como uno de los programas GNU.

Autoconf.- es una herramienta informática para crear shell scripts(Un script (cuya traducción literal es 'guion') o archivo de órdenes o archivo de procesamiento por lotes es un programa usualmente simple, que por lo regular se almacena en un archivo de texto plano) que configuren automáticamente el código fuente de un software para adaptarlo a diversos sistemas tipo UNIX.

Libtool.- Es una herramienta de programación GNU proveniente del sistema de construcción para GNU usada para crear bibliotecas de software portables.

API.- Una interfaz de programación de aplicaciones o API (del inglés Application Programming Interface) es el conjunto de funciones y procedimientos (o métodos, en la programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción. Son usadas generalmente en las bibliotecas (también denominadas comúnmente "librerías").

Kernel.- O núcleo, es un software que constituye la parte más importante del sistema operativo. Es el principal responsable de facilitar a los distintos programas acceso seguro al hardware de la computadora o en forma más básica, es el encargado de gestionar recursos, a través de servicios de llamada al sistema.

B-tree.- En las ciencias de la computación, los árboles-B o B-árboles son estructuras de datos de árbol que se encuentran comúnmente en las implementaciones de bases de datos y sistemas de archivos. Son árboles binarios de búsqueda en los cuales cada

nodo puede poseer más de dos hijos. Los árboles B mantienen los datos ordenados y las inserciones y eliminaciones se realizan en tiempo logarítmico amortizado.

Joins.- La sentencia join en SQL permite combinar registros de dos o más tablas en una base de datos relacional.

POSIX OSIX.- Es el acrónimo de Portable Operating System Interface; la X viene de UNIX como seña de identidad de la API. Una traducción aproximada del acrónimo podría ser "Interfaz de sistema operativo portable".

AIX (Advanced Interactive eXecutive).- Es un sistema operativo UNIX System V propiedad de IBM.

Amiga.- Es un ordenador personal que debido a sus extraordinarias capacidades multimedia, cosechó grandes éxitos en la década de los años 80 y 90.

BSDI.- Es un sistema operativo libre para computadoras basado en las CPU de arquitectura Intel.

FreeBSD.- Es un avanzado sistema operativo para arquitecturas x86 compatibles como Pentium, Athlon y algunos AMD. FreeBSD es un derivado de BSD, la versión de UNIX desarrollada en la Universidad de California, Berkeley.

HP-UX.- Es la versión de Unix desarrollada y mantenida por Hewlett-Packard desde 1983, ejecutable típicamente sobre procesadores HP PA RISC y en sus últimas versiones sobre Intel Itanium (arquitectura Intel de 64 bits).

Linux.- Es un núcleo de sistema operativo libre tipo Unix; es uno de los principales ejemplos de software libre, está licenciado bajo la GPL v2 y está desarrollado por

colaboradores de todo el mundo. El desarrollo del día a día tiene lugar en la Linux Kernel Mailing List.

NetBSD.- Es un sistema operativo de la familia Unix, de código abierto y libre, y, a diciembre de 2008, disponible para más de 56 plataformas hardware. Su diseño y sus características avanzadas lo hacen ideal para multitud de aplicaciones. NetBSD ha surgido como resultado del esfuerzo de un gran número de personas que tienen como meta producir un sistema operativo tipo Unix accesible y libremente distribuible.

Novell NetWare.- Es un Sistema operativo de red. Es una de las plataformas de servicio más fiable para ofrecer acceso seguro y continuado a la red y los recursos de información, sobre todo en cuanto a servidores de archivos.

OpenBSD.- es un sistema operativo libre tipo Unix multiplataforma. Es un descendiente de NetBSD, con un foco especial en la seguridad y la criptografía.

OS/2.- Es un sistema operativo de IBM que intentó suceder a DOS como sistema operativo de las computadoras personales. Se desarrolló inicialmente de manera conjunta entre Microsoft e IBM, hasta que la primera decidió seguir su camino con su Windows 3.0 e IBM se ocupó en solitario de OS/2.

SCO OpenServer.- Sistema operativo propietario, perteneciente a The SCO Group. Está basado en Unix System V.

SGI IRIX.- Fue un sistema operativo para computadoras desarrollado por Silicon Graphics Inc. para ejecutar nativamente arquitecturas de 32 y 64 bits.

Solaris.- Es un sistema operativo de tipo Unix desarrollado desde 1992 inicialmente por Sun Microsystems y actualmente por Oracle Corporation como sucesor de

SunOS. Es un sistema certificado oficialmente como versión de Unix. Funciona en arquitecturas SPARC y x86 para servidores y estaciones de trabajo.

SunOS.- Fue la versión del sistema operativo derivado de Unix y BSD desarrollado por Sun Microsystems para sus estaciones de trabajo y servidores hasta el principio de los años 1990.

Tru64.- Es un sistema operativo de HP para plataformas Alpha, anteriormente conocido como Digital UNIX (1995-98) y antes aun (1992-95) como DEC OSF/1 AXP.

Commit.- Una sentencia COMMIT en SQL finaliza una transacción de base de datos dentro de un sistema gestor de base de datos relacional (RDBMS) y pone visibles todos los cambios a otros usuarios.

Rollback.- En SQL, ROLLBACK es un comando que causa que todos los cambios de datos desde la última sentencia sean descartados por el sistema de gestión de base de datos relacional (RDBMS), para que el estado de los datos sea "rolled back"(devuelto) a la forma en que estaba antes de que aquellos cambios tuvieran lugar.

Benchmark.- El benchmark es una técnica utilizada para medir el rendimiento de un sistema o componente del mismo.

RAID.- En informática, el acrónimo RAID (del inglés Redundant Array of Independent Disks, «conjunto redundante de discos independientes»), anteriormente conocido como Redundant Array of Inexpensive Disks, «conjunto redundante de

discos baratos») hace referencia a un sistema de almacenamiento que usa múltiples discos duros entre los que se distribuyen o replican los datos.

CAPÍTULO III

METODOLOGÍA

DISEÑO DE LA INVESTIGACIÓN

3.1. MODALIDAD DE LA INVESTIGACIÓN

La modalidad de investigación del presente proyecto es bibliográfica, ya que se basa en el estudio de algunos proyectos ya existentes como son las replicaciones y el ODBC para presentar una solución nueva y brindar beneficios principalmente en los respaldos que se puedan hacer de los datos, haciendo un estudio de los posibles casos que se puedan dar para determinar las mejores configuraciones posibles.

3.2. TIPO DE INVESTIGACIÓN

La investigación que se va a realizar es de laboratorio ya que con varios de los datos obtenidos se obtendrán diferentes tipos de escenarios en los cuales se harán pruebas de la solución que se va a presentar para así determinar los mejores escenarios de uso en los que se puede aplicar, ventajas, desventajas y posibles mejoras futuras y puntos a considerar.

3.3. POBLACIÓN Y MUESTRA

Población

Como el estudio realizado será de diferentes escenarios la población que se tomará en cuenta será de ingenieros en sistemas que tengan a su cargo las configuraciones y el cuidado de la información de las bases de datos así como el uso de diferentes tipos de herramientas para tener la disponibilidad de estos. Otro grupo de personas que se tomó en cuenta fue a los usuarios de empresas que usan la base de datos MySQL para determinar si la información que ellos usan está disponible todo el tiempo que la necesitan.

Muestra

La muestra tomada fue de 4 empresas, que tienen diferentes configuraciones tanto de hardware como las herramientas que utilizan, y de un administrador que piensa migrar a MySQL y busca las mejores configuraciones y herramientas que le permitan ofrecer todas las seguridades posibles.

Las empresas tomadas para la muestra son las siguientes:

- Ecuasistemas S.A.
- Copydata S.A.
- Líneas Médicas
- Nextel

CUADRO NO. 6

POBLACIÓN DE ADMINISTRADORES DE BASES DE DATOS Y USUARIOS QUE TRABAJAN CON ELLAS

Población de administradores de bases de datos y usuarios que trabajan con ellas	Cantidad
Empresas	5
Administradores de Bases de datos DBA	10
Usuarios	25

Elaboración: Poll Cedeño Tello

Fuente: Datos Recopilados de la Investigación

3.4. OPERACIONALIZACIÓN DE VARIABLES

CUADRO NO. 7
MATRIZ DE OPERACIONALIZACIÓN DE VARIABLES

Variables	Dimensiones	Indicadores	Técnicas y/o Instrumentos
Configuraciones para tener disponibilidad	Revisión de las configuraciones	Hardware y software que se utiliza	Entrevistas y observación de las configuraciones.
Replicación	Revisión de las configuraciones y procesos que se realizan.	Tipo de replicación que se utiliza	Revisión de reportes y registros del sistema
Forma de conexión entre cliente y servidor conexión entre el cliente y el servidor	Análisis del método de conexión usado.	La herramienta que se usa para la conexión	Entrevista, revisión de configuraciones.

Elaboración: Poll Cedeño Tello

Fuente: Investigación realizada

3.5. INSTRUMENTOS DE RECOLECCIÓN DE DATOS

Para poder realizar el estudio de las configuraciones que tienen en cada uno de los encuestados y así poder tener ambientes de prueba se utilizaron instrumentos de recolección de datos como entrevistas, observación y cuestionarios.

Lo primero que se tuvo que hacer es conseguir citas con los usuarios dentro de cada institución que administren o usen la base de datos. Se pudo conocer por medio de cuestionarios y entrevistas las configuraciones que usan en el lugar de trabajo y el rendimiento que presentan, así como métodos de respaldo y métodos que se usan para tener disponibilidad de la información. Cabe recalcar que a pesar de realizar las entrevistas se hizo énfasis en el respeto a la privacidad tanto de los entrevistados como en las respuestas dadas en los cuestionarios. Por otro lado también se hizo encuestas y entrevistas a los usuarios para conocer el trabajo que realizan así como, su opinión ante el sistema que usan haciendo énfasis en el tiempo de respuesta. El formato de las encuestas realizadas son las siguientes:

Preguntas	SI	NO
¿Se usa en su institución MySQL como motor de base de Datos?		
¿Usa alguna configuración de replicación?		
¿Se toman medidas para compensar la replicación asíncrona?		
¿Usa el ODBC como forma de conexión entre el cliente y el servidor?		

El formato del cuestionario es el siguiente:

- 1.- Describa como tiene configurado(s) el/los servidor(es).
- 2.- Describa las configuraciones usadas para la replicación.
- 3.- Describa qué medidas (programas, configuraciones, etc.) usa para compensar la replicación asíncrona de MySQL.

3.6. INSTRUMENTOS DE LA INVESTIGACIÓN

Para realizar la presente investigación se necesitó de entrevistas y cuestionarios las cuáles fueron hechas a los profesionales que están en contacto con las bases

de datos y sus configuraciones así como los usuarios que son los que notan los cambios en el funcionamiento del sistema principalmente con lo que respecta al tiempo de respuesta. También el principal instrumento para la obtención de la información es el internet, de aquí se investigó los métodos de replicación que se pueden realizar así como las características de una base de datos en modo de espera y la edición del ODBC que está dentro de la solución a plantear para los diferentes casos. Para las pruebas se utilizó una representación de los escenarios obtenidos en la recolección de información así como la implementación de otros que pueden ser usados dependiendo de cada caso.

3.7. PROCEDIMIENTOS DE LA INVESTIGACIÓN

El problema

Hoy en día las empresas o instituciones no pueden permitirse estar fuera de servicio o tener pérdida de la información que está a su disposición ya que hoy en día esta ha llegado a ser el activo más importante. Tomando en cuenta esto los administradores de las bases de datos se preguntan: ¿Que configuración es la más adecuada para su entorno?, ¿se puede mejorar la actualmente usada?, ¿Cómo tener mayor redundancia de la información? Dadas estas interrogantes se obtiene el objetivo del presente proyecto que dado ciertos ambientes se harán pruebas sobre ellos y plantearán diferentes soluciones incluyendo la de obtener la información generada por los clientes. Mejorar las configuraciones de las bases de datos, mejorar los tiempos de respuestas, mejorar la redundancia de la información sin duda es de suma importancia para cualquier institución.

Marco teórico

MySQL es un sistema de gestión de base datos relacional, multi hilo, multi usuario que sirve de interfaz entre los usuarios y la base de datos; diseñado para entornos de producción críticos con alta carga de trabajo.

MySQL posee de varias herramientas que se las puede usar dependiendo de cada entorno de trabajo, son así diferentes motores de almacenamiento, conectores para diferentes lenguajes de programación, permite usar los conceptos de Clúster y replicación, y cada vez presenta nuevas herramientas que facilitarán el trabajo de los que la usen.

Una base de datos en modo de espera es una copia de la base de datos en producción, generalmente esta copia se la realiza por el concepto de replicación, que permite ser usada para la disminución de la carga transaccional de la base principal, hacer un respaldo en caliente, ponerla en producción en caso de que la base principal tenga algún inconveniente.

Otra herramienta que nos presenta MySQL es el conector ODBC que permite la conexión de los usuarios con la base de datos, este nos presenta una variedad de opciones y se puede ser utilizada tanto en entorno Linux como Windows.

En el Ecuador la investigación científica está respaldada en la constitución

Si tenemos a nuestra disposición todos estos conceptos debemos de preguntarnos

¿Cuáles son las configuraciones necesarias para tener las características de una base en modo de espera?, ¿Qué cambios se le debe hacer al ODBC para obtener la información generada por los clientes o usuarios?

Entre los términos usados tenemos base de datos en modo de espera, replicación semi-síncrona, agentes que van a ser las herramientas a desarrollar que se instalarán en los clientes y en los servidores; ODBC (por sus siglas en inglés

Open DataBase Connectivity) que proporciona una forma para que los programas cliente para acceder a una amplia gama de bases de datos o fuentes de datos.

Metodología

El tipo de investigación que se va a usar es de laboratorio ya que parte de la solución es que se procederá a la edición del ODBC para extraer la información generada por los clientes, y el respaldo de esta información hacia los servidores por medio de las herramientas a crear.

La población que se escogió para este proyecto son los administradores de bases de datos y los usuarios que son los que usan los sistemas.

La muestra fue la selección de 4 empresas que presentaban diferentes tipos de ambientes así como el de administrador que está analizando la opción de migrar a MySQL.

La recolección de datos se la realizó por medio de encuestas y entrevistas realizadas, obteniendo así sus opiniones de las configuraciones que tienen en este momento y así como la opción de obtener la información desde los usuarios. Ya obtenida la información esta pasará a ser revisada para sacar conclusiones y proceder a realizar las pruebas del caso

3.8. RECOLECCIÓN DE INFORMACIÓN

El levantamiento de información se lo hizo mediante entrevistas, encuestas y observación a la población descrita anteriormente.

Las encuestas y entrevistas se las hizo tanto a administradores como a usuarios y

la observación se la hizo en poco casos la mayoría a usuarios.

Otra parte de la información recolectada se la hizo por internet en la investigación realizada para obtener información de los diferentes tipos de configuraciones así como de diferentes conceptos y herramientas a usar.

3.9. PROCESAMIENTO Y ANÁLISIS

A las posibles respuestas de las preguntas realizadas en las encuestas se les asignó un valor para determinar los puntos más importantes considerados por los encuestados; y así tomar en cuenta los diferentes criterios de ellos. Esta revisión se la hizo manualmente después de haber realizado todas las encuestas.

CAPÍTULO IV

MARCO ADMINISTRATIVO

4.1. CRONOGRAMA

	ACTIVIDADES	TIEMPO	FECHA INICIO	FECHA FIN	PREDECESORAS
1	Entrega de temas de proyecto para revisión	24 horas	05/04/2010 19:00	10/04/2010 23:00	
2	Inicio del desarrollo del proyecto	108,5 días	12/04/2010 19:00	16/08/2010 21:00	
3	Replanteamiento del tema y aprobación del mismo	36 horas	12/04/2010 19:00	21/04/2010 23:00	1
4	Formulación del problema	15 días	22/04/2010 19:00	08/05/2010 23:00	
5	Investigación sobre el tema	24 horas	22/04/2010 19:00	28/04/2010 23:00	3
6	Delimitación del problema	8 horas	29/04/2010 19:00	30/04/2010 23:00	5
7	Definir objetivos a alcanzar	8 horas	01/05/2010 19:00	03/05/2010 23:00	6
8	Determinar objetivos a alcanzar	8 horas	04/05/2010 19:00	05/05/2010 23:00	7
9	Determinar los recursos disponibles	4 horas	06/05/2010 19:00	06/05/2010 23:00	8
10	Documentación	8 horas	07/05/2010	08/05/2010	9

			19:00	23:00	
11	Definición de la hipótesis	9 días	10/05/2010 19:00	19/05/2010 23:00	
12	Investigación de estudios anteriores realizados sobre el tema	12 horas	10/05/2010 19:00	12/05/2010 23:00	10
13	Definir el objeto del problema escogido para la investigación	8 horas	13/05/2010 19:00	14/05/2010 23:00	12
	ACTIVIDADES	TIEMPO	FECHA INICIO	FECHA FIN	PREDECESORAS
14	Definir las variables del problema	4 horas	15/05/2010 19:00	15/05/2010 23:00	13
15	Definición de términos a usar	4 horas	17/05/2010 19:00	17/05/2010 23:00	14
16	Documentación	8 horas	18/05/2010 19:00	19/05/2010 23:00	15
17	Desarrollo de la hipótesis	21,5 días	20/05/2010 19:00	14/06/2010 21:00	
18	Estudio de las variables	12 horas	20/05/2010 19:00	22/05/2010 23:00	16
19	Estudio de casos relacionado o investigados	8 horas	24/05/2010 19:00	25/05/2010 23:00	18
20	Determinar forma de comunicación entre servidores	6 horas	26/05/2010 19:00	27/05/2010 21:00	19
21	Determinar forma de comunicación entre servidor y cliente	8 horas	27/05/2010 21:00	29/05/2010 21:00	20
22	Definir el esquema de trabajo del agente en el servidor	4 horas	29/05/2010 21:00	31/05/2010 21:00	21
23	Definir el esquema de trabajo del agente en el cliente	12 horas	31/05/2010 21:00	03/06/2010 21:00	22
24	Plan de mapeo de datos	12 horas	03/06/2010 21:00	07/06/2010 21:00	23
25	Plan de pruebas	12 horas	07/06/2010 21:00	10/06/2010 21:00	24
26	Documentación	12 horas	10/06/2010 21:00	14/06/2010 21:00	25
27	Elección de Solución Propuesta y Justificación	40 días	14/06/2010 21:00	30/07/2010 21:00	
28	Determina la solución a usar	12 horas	14/06/2010 21:00	17/06/2010 21:00	26
29	Justificación	37 días	17/06/2010 21:00	30/07/2010 21:00	
30	Instalación de las bases	6 horas	17/06/2010	18/06/2010	28

			21:00	23:00	
31	Creación de las estructuras	4 horas	19/06/2010 19:00	19/06/2010 23:00	30
32	Inicio de la replicación	8 horas	21/06/2010 19:00	22/06/2010 23:00	31
	ACTIVIDADES	TIEMPO	FECHA INICIO	FECHA FIN	PREDECESORAS
33	Instalación de los clientes	8 horas	23/06/2010 19:00	24/06/2010 23:00	32
34	Instalación de aplicativo en clientes	4 horas	25/06/2010 19:00	25/06/2010 23:00	33
35	Ingreso de datos	2 horas	26/06/2010 19:00	26/06/2010 21:00	34
36	Creación del agente para el servidor	24 horas	26/06/2010 21:00	03/07/2010 21:00	35
37	Creación del agente para los clientes	52 horas	03/07/2010 21:00	19/07/2010 21:00	36
38	Ejecución de planes	24 horas	19/07/2010 21:00	26/07/2010 21:00	37
39	Documentación	16 horas	26/07/2010 21:00	30/07/2010 21:00	38
40	Conclusiones	14 días	30/07/2010 21:00	16/08/2010 21:00	
41	Revisión de todo lo realizado	24 horas	30/07/2010 21:00	06/08/2010 21:00	39
42	Documentación	32 horas	06/08/2010 21:00	16/08/2010 21:00	41
43	Revisión de documentación	24 horas	23/08/2010 19:00	28/08/2010 23:00	
44	Revisión y sustentación	4 horas	30/08/2010 19:00	30/08/2010 23:00	

4.2. PRESUPUESTO

CUADRO NO. 8

DETALLE DE EGRESOS DEL PROYECTO

Egresos	Dólares
Suministros de oficina y computación	\$ 150
Fotocopias	10
Computadora y servicios de internet	1000
Transporte	40
Refrigerio	50
Empastado, anillado de tesis de grado	50
Total.....	\$ 1300

Elaboración: Poll Cedeño Tello

Fuente: Investigación realizada

REFERENCIAS BIBLIOGRÁFICAS

Miguel Ángel Nieto. Marzo – 2010. Administración de MySQL

<http://www.slideshare.net/miguelangelnieto/mysql-administracion>

MySQL 5.5 Manual de referencia. 2010

<http://dev.mysql.com/doc/refman/5.5/en/>

Replicación en MySQL. 2010

<http://dev.mysql.com/doc/refman/5.5/en/replication.html>

Conector ODBC de MySQL. 2010

<http://dev.mysql.com/doc/refman/5.1/en/connector-odbc.html>

Manual de base de datos en modo de espera (Standby database). 16 de Marzo del 2010

<http://www.dataprix.com/blogs/il-masacratore/oracle10g-manual-standby-database-planteamiento-inicial>

ANEXOS

Dentro de los archivos del código fuente del ODBC podemos encontrar el procedimiento para compilarlo tanto en Linux como en Windows.

CAPÍTULO V

CONCLUSIONES Y RECOMENDACIONES

5.1. CONCLUSIONES

Se puede tener una variedad de configuraciones que se pueden hacer sin la necesidad de que nos tome una gran cantidad de tiempo o que tengamos que detener el servidor para aplicar los cambios. Dadas las configuraciones realizadas depende de cada persona incluirla en su ambiente de trabajo.

También hay que tomar en cuenta las opciones presentadas no son las únicas a usar si tiene la posibilidad también se podría usar lo que es Clúster que en MySQL este si funciona de manera síncrona al contrario de la replicación que se espera que en un futuro sea síncrona.

No necesitamos de muchos servidores, el número mínimo es de dos, para implementar algunos de estos ejemplos y además no hay que olvidar realizar y probar respaldos periódicamente, que son también de ayuda.

Las herramientas de código abierto son de mucha ayuda para personalizarla a nuestras necesidades pero lo que si hay que tomar en cuenta es que hay que tener mucha paciencia al momento de querer editar el código ya que se debe entender la estructura de cómo ha sido programado.

Muchas personas (por ejemplo en internet) indican que no es muy factible usar el ODBC como hemos podido ver nos ofrece algunas opciones las cuales permiten mejorar el uso del sistema en los clientes como por ejemplo usar la opción de auto-reconectar para no tener que salir obligatoriamente de una aplicación para

volver a conectarme a la base de datos, así como la opción agregada recientemente que permite la obtención de un respaldo más. Además nos permite la administración de nuestra base de datos a través de MS Access.

5.2. RECOMENDACIONES

Como se lo ha mencionado anteriormente cada configuración depende de cada ambiente en donde tengamos nuestra base de datos. Por ejemplo: Si trabajamos en un ambiente, con un servidor maestro y una réplica, donde las escrituras al servidor son mayores que las lecturas, en este caso podemos usar una replicación maestro – maestro, pero si tenemos un ambiente en donde las lecturas son casi iguales o mayores que las escrituras podemos usar a la replicación como base de sólo lectura, claro que hay que tomar en cuenta los permisos que se le deben dar a los usuarios.

El usuario que se crea para la replicación debemos dejarlo solamente para esta función.

Si vamos a usar la presente función agregada al ODBC se recomienda no usar “Log queries to myodbc.sql”, ya que podría disminuir el rendimiento de computadores con pocos recursos de hardware, también debemos poner correctamente los parámetros en el archivo de configuración del agente del cliente, ya que este no va a usar el ODBC para enviar las sentencias a la base. Otro punto a considerar de esta opción es que nada asegura que la

información generada por los clientes no sea modificada por estos y si se la va a replicar a los servidores se debe tomar en cuenta que si se hizo un cambio en el servidor los clientes no pueden verlo.

BIBLIOGRAFÍA

Miguel Angel Nieto. 2010. Replicación semi-síncrona con MySQL 5.5

<http://miguelangelnieto.net/?action=view&url=replicaci%C3%B3n-semi-s%C3%ADncrona-con-mysql-55>

Miguel Angel Nieto. 2010. Arquitectura Maestro/Maestro en MySQL

<http://miguelangelnieto.net/?action=view&url=arquitectura-maestro-maestro-en-mysql>

Manual de C. 2005

<http://www.fismat.umich.mx/mn1/manual/manual.html>

Usando la replicación de MySQL para aumentar la disponibilidad

<http://www.phpbuilder.com/columns/tanoviceanu20000912.php3>

Trabajando con replicación multi-maestro de MySQL

<http://www.databasejournal.com/features/mysql/article.php/3896061/Working-with-MySQL-Multi-master-Replication---Keeping-a-True-Hot-Standby.htm>

Conceptos básicos sobre ODBC

<http://www.pecesama.net/php/odbc.php>

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES

**AUMENTO DE LA DISPONIBILIDAD DE BASES DE DATOS
MYSQL UTILIZANDO UNA CONFIGURACIÓN
DE BASE EN MODO DE ESPERA E
INSTALACIÓN DE AGENTES
PARA AUTOMATIZAR
CONEXIONES**

TESIS DE GRADO

Previa a la obtención del Título de:

INGENIERO EN SISTEMAS COMPUTACIONALES

AUTOR: POLL MIGUEL CEDEÑO TELLO

TUTOR: ING. RAYNAUD QUIROZ

GUAYAQUIL – ECUADOR

2011

Guayaquil, 11 de Octubre del 2010

APROBACIÓN DEL TUTOR

En mi calidad de Tutor del trabajo de investigación, "Aumento de la disponibilidad de bases de datos MySQL utilizando una configuración de base en modo de espera e instalación de agentes para automatizar conexiones" elaborado por el Sr. Poll Miguel Cedeño Tello, egresado de la Carrera de Ingeniería en Sistemas Computacionales, Facultad de Ciencias Matemáticas y Físicas de la Universidad de Guayaquil, previo a la obtención del Título de Ingeniero en Sistemas, me permito declarar que luego de haber orientado, estudiado y revisado, la Apruebo en todas sus partes.

Atentamente

.....
Ing. Raynaud Quiroz
TUTOR

DEDICATORIA

Agradezco a Dios que me ha dado la fuerza y capacidad para llegar hasta aquí, así como a mi familia que me ha estado apoyando en toda la trayectoria de mi carrera tanto en las buenas como en las malas.

TRIBUNAL DE GRADO

Ing. Fernando Abad Montero
DECANO DE LA FACULTAD
CIENCIAS MATEMÁTICAS Y FÍSICAS

Ing. Juan Chanabá Alcócer
DIRECTOR

Ing. Raynaud Quiroz
TUTOR

Inga. Ana María Arellano
PROFESOR DEL ÁREA - TRIBUNAL

Ing. Juan Carlos Cedeño
PROFESOR DEL ÁREA – TRIBUNAL

Dr. José Júpiter
SECRETARIO

UNIVERSIDAD DE GUAYAQUIL

FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS

CARRERA DE INGENIERÍA EN SISTEMAS

COMPUTACIONALES

**AUMENTO DE LA DISPONIBILIDAD DE BASES DE DATOS
MYSQL UTILIZANDO UNA CONFIGURACIÓN
DE BASE EN MODO DE ESPERA E
INSTALACIÓN DE AGENTES
PARA AUTOMATIZAR
CONEXIONES**

MySQL es una de las bases de datos libres y de código abierto más usadas en muchos ambientes laborales. Tiene un sin número de herramientas útiles y para cada lugar de trabajo. Este proyecto busca mejorar una de las áreas de MySQL que es la replicación. El problema se debe a que el método de replicación es asíncrono esto significa que no es inmediato y que se debe esperar cierta cantidad de tiempo para que la información se replique, y este tiempo va a depender de varios factores como pueden ser hardware, software, configuraciones; por estos motivos se puede dar el caso de que al momento de replicar la información no llegue o llegue incompleta. La solución que se plantea consiste en extraer del cliente las consultas que se envían a la base de datos y guardarlos en archivos; para lograr esto se procedió con la edición del código fuente del ODBC, agregando también una función la cual va a estar indicando si las consultas se están ejecutando correctamente; caso contrario si alguna no se puede ejecutar se escribirá un mensaje de error en un registro que llevará el nombre del DSN. Se procederá a la instalación de agentes tanto en los clientes como en un servidor esclavo. El agente que se encontrará en el servidor esclavo estará censando la comunicación con el maestro, si se llegara a perder la comunicación se realizarán las configuraciones necesarias para que el servidor funcione como maestro para así poder recibir las consultas enviadas por todos los clientes. El agente en el cliente va a estar encargado de leer el registro con el nombre del DSN; al detectar algún mensaje de error procederá a realizar procesos para replicar la información almacenada, para que así no exista pérdida de esta.

UNIVERSIDAD DE GUAYAQUIL
FACULTAD DE CIENCIAS MATEMÁTICAS Y FÍSICAS
CARRERA DE INGENIERÍA EN SISTEMAS
COMPUTACIONALES

**AUMENTO DE LA DISPONIBILIDAD DE BASES DE DATOS
MYSQL UTILIZANDO UNA CONFIGURACIÓN
DE BASE EN MODO DE ESPERA E
INSTALACIÓN DE AGENTES
PARA AUTOMATIZAR
CONEXIONES**

MySQL is one of the free data base and open source used in many workplaces. It has different useful tools for each Workplace. This project seeks to improve one of the area of MySQL that is the replication. The problem is that the method of replication is asynchronous this means that is not immediate and it needs to expect a certain amount of time in order to that this information can be replicated and this time will depend on several factors such as hardware, software, configurations, for these reasons it may be the case when comes the moment to replicate the information, this information does not arrive or arrive incomplete. The solution that arises is to extract customer queries that are send to the database and save in files; to achieve this we proceeded with the release of the ODBC code source, also adding a function which will be indicate if queries are running correctly, otherwise if you cannot run it will write an error message in a register held by the name of the DSN. It will install both agents clients like a slave server. The agent may be found on the slave communication registers you with the teacher, if you were to lose the communication was made necessary settings for the server to work as a teacher in order to receive the queries sent by all clients. The agent on the client will be responsible for reading the record with the name of the DSN, to detect any error messages shall carry out processes to replicate the information stored, so that no loss of it.

CERTIFICADO DE ACEPTACIÓN DEL TUTOR

En mi calidad de Tutor del Primer Curso de Fin de Carrera, nombrado por el Departamento de Graduación y la Dirección de la Carrera de Ingeniería en Sistemas Computacionales de la Universidad de Guayaquil,

CERTIFICO:

Que he analizado el Proyecto de Grado presentado por el/la egresado(a) Poll Miguel Cedeño Tello, como requisito previo para optar por el título de Ingeniero cuyo problema es:

AUMENTO DE LA DISPONIBILIDAD DE BASES DE DATOS MYSQL UTILIZANDO UNA CONFIGURACIÓN DE BASE EN MODO DE ESPERA E INSTALACIÓN DE AGENTES PARA AUTOMATIZAR CONEXIONES

considero aprobado el trabajo en su totalidad.

Presentado por:

Poll Miguel Cedeño Tello

0922634670

Apellidos y Nombres completos

Cédula de ciudadanía N°

Tutor: Ing. Raynaud Quiroz

Guayaquil, _____ de _____

ÍNDICE GENERAL

CARÁTULA	i
CARTA DE ACEPTACIÓN DEL TUTOR	ii
DEDICATORIA	iii
TRIBUNAL	iv

RESUMEN	v
ABSTRACT	vi
CERTIFICADO DE APROBACIÓN	vii
ÍNDICE GENERAL	viii
ÍNDICE DE GRÁFICOS	ix
RESOLUCIÓN DEL PROYECTO	1
MANUAL DE USUARIO DEL ODBC	12
MANUAL TÉCNICO DEL ODBC	15
MANUAL DE USUARIO DEL AGENTE	30
MANUAL TÉCNICO DEL AGENTE	33

ÍNDICE DE GRÁFICOS

	Pág.
Gráfico 1 Estado del Maestro “Show Master Status”	3
Gráfico 2 Base de datos ‘data’ creada	5

Gráfico 3 Base de datos 'data' replicada	6
Gráfico 4 Plug-in's instalados	7
Gráfico 5 Replicación semi-síncrona maestro.	9
Gráfico 6 Replicación semi-síncrona esclavo	9
Gráfico 7 Registro de la base de datos Maestro	10
Gráfico 8 Configurando el conector ODBC	12
Gráfico 9 Activando la opción "Log queries by PID and DSN"	13
Gráfico 10 Archivos donde se guardan las transacciones y el log del DSN.	14
Gráfico 11 Abriendo el código fuente del ODBC	16
Gráfico 12 Sección DEBUG en ODBC	16
Gráfico 13 Checkbox para las sentencias creado	18
Gráfico 14 Registro agregado en Windows	24
Gráfico 15 Archivos Creados	29

RESOLUCIÓN DEL PROYECTO

INICIANDO REPLICACIÓN EN MYSQL

Se debe tomar en cuenta que tanto el maestro como el esclavo se puedan comunicar por el puerto 3306, que es el puerto por defecto con el que trabaja MySQL. También

se asumirá que ha sacado respaldo del maestro y los ha ejecutado en el esclavo en el caso de que tenga información ingresada.

También que el servidor usado como de replicación exista en el archivo `/etc/hosts`.

La replicación presente es la llamada maestro – esclavo que es la básica, y hay que recordar que la replicación es asíncrona. El tiempo de replicación por defecto es de 60 segundos.

La IP asignada al maestro va a ser 192.168.100.10 y la del esclavo 192.168.100.20

Los siguientes pasos serán para el servidor maestro.

En el archivo `/etc/my.cnf` debemos tener los siguientes valores:

```
log-bin = proyecto-bin
```

```
server-id=1
```

```
bin-address=0.0.0.0 #Para permitir conexiones remotas
```

```
skip-name-resolve #Si no se usa dominio
```

Ahora procedemos a reiniciar el servidor MySQL para que lea las opciones dadas.

Ingresemos a MySQL y creamos una cuenta para que el servidor de replicación se pueda conectar a la base de datos.

```
mysql> grant replication slave on *.* to
'mysqlrepl'@'%.proyecto.com'
-> identified by 'replica86';
```

Con esto hemos creado un usuario de nombre “mysqlrepl” con contraseña “replica86” el cual solo va a servir para la replicación.

A continuación ejecutamos los siguientes comandos:

```
mysql> flush tables with read lock;
```

```
mysql> show master status;
```

En este momento vamos a tener la siguiente salida.

Gráfico 1. Estado del Maestro “Show Master Status”

```
mysql> show master status
-> ;
+-----+-----+-----+-----+
| File                | Position | Binlog_Do_DB | Binlog_Ignore_DB |
+-----+-----+-----+-----+
| proyecto-bin.000013 |      509 |              |                  |
+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

Aquí copiamos los valores de File y Position

Ejecutamos el comando:

```
shell> unlock tables;
```

Para continuar con la escritura de datos en el maestro. Estos son los pasos necesarios ahora procedemos con el esclavo.

En el archivo /etc/my.cnf del esclavo debemos tener lo siguiente.

```
[mysqld]
```

```
server-id=2
```

Reiniciamos el servicio de mysql.

Ahora desde el cliente de mysql del cliente ejecutamos el siguiente comando.

```
mysql> change master to
```

```
->MASTER_HOST='proyecto.com',  
->MASTER_USER='mysqlrepl',  
->MASTER_PASSWORD='replica86',  
->MASTER_LOG_FILE='proyecto_bin.000013'  
->MASTER_LOG_POS=509;
```

Con esto indicamos los parámetros de conexión al maestro.

Por último iniciamos el esclavo con:

```
mysql> start slave;
```

Para probar que todo esté funcionando crearemos una base de datos e importaremos un respaldo. En mysql procedemos a crear una base de datos

```
mysql> create database data;
```

Ahora desde una terminal de Linux importamos el archivo de creación. Para este ejemplo nos vamos a ubicar dentro del directorio de instalación de mysql (/usr/local/mysql)

```
# bin/mysql -u root -p data < /super.sql
```

Donde /super.sql es donde se encuentra el respaldo. Si todo ha funcionado correctamente no nos debe presentar ningún error. Ahora ingresamos a MySQL para comprobar que se importaron correctamente los datos.

```
mysql> use data;
```

```
mysql> show tables;
```

Con estos comandos procedemos a ver la lista de tablas creadas dentro de la base de datos data. Como la replicación ya ha sido iniciada encontraremos la base de datos


```

root@replica:/usr/local/mysql
Archivo Editar Ver Terminal Solapas Ayuda
You can turn off this feature to get a quicker startup with -A

Database changed
mysql> show tables;
+-----+
| Tables_in_data |
+-----+
| bodega          |
| contenedor      |
| documto         |
| documtoerror    |
| documtop        |
| documtoprint    |
| documtoreport   |
| dpto            |
| empresa        |
| funcionarios    |
| ingbarcode      |
| ingbarcodereport |
| log             |
| mdeal           |
| nueva          |
| prestamos       |
| tipodoc         |
| user           |
+-----+
18 rows in set (0.00 sec)

mysql> █

```

REPLICACIÓN SEMI-SÍCRONA

Ahora vamos a iniciar con la replicación semi-síncrona a diferencia de la replicación normal esta tiene un tiempo para replicar por defecto de 10 segundos.

Lo primero que se debe hacer es instalar el Plug-in.

En el maestro:

```
mysql> install plugin rpl_semi_sync_master soname
'semisync_master.so';
```

Si este comando nos devuelve algún tipo de error debemos revisar que el Plug-in se encuentre en el siguiente directorio `/usr/local/mysql/lib/plugin`.

Ahora en el esclavo también lo instalamos pero el que corresponde al esclavo.

```
mysql> install plugin rpl_semi_sync_slave soname
'semisync_slave.so';
```

Para ver los Plug-in's instalados ejecutamos el siguiente comando;

```
mysql> show plugins;
```

En el listado que se muestra a continuación podemos ver instalado el Plug-in resiente.

Gráfico 4. Plug-in's instalados

```

mysql> show plugins;
+-----+-----+-----+-----+-----+
| Name          | Status | Type          | Library      | License |
+-----+-----+-----+-----+-----+
| binlog        | ACTIVE | STORAGE ENGINE | NULL         | GPL     |
| CSV           | ACTIVE | STORAGE ENGINE | NULL         | GPL     |
| MEMORY        | ACTIVE | STORAGE ENGINE | NULL         | GPL     |
| MRG_MYISAM    | ACTIVE | STORAGE ENGINE | NULL         | GPL     |
| MyISAM        | ACTIVE | STORAGE ENGINE | NULL         | GPL     |
| partition     | ACTIVE | STORAGE ENGINE | NULL         | GPL     |
| ARCHIVE       | ACTIVE | STORAGE ENGINE | NULL         | GPL     |
| BLACKHOLE     | ACTIVE | STORAGE ENGINE | NULL         | GPL     |
| FEDERATED     | DISABLED | STORAGE ENGINE | NULL         | GPL     |
| InnoDB        | ACTIVE | STORAGE ENGINE | NULL         | GPL     |
| INNODB_TRX    | ACTIVE | INFORMATION SCHEMA | NULL         | GPL     |
| INNODB_LOCKS  | ACTIVE | INFORMATION SCHEMA | NULL         | GPL     |
| INNODB_LOCK_WAITS | ACTIVE | INFORMATION SCHEMA | NULL         | GPL     |
| INNODB_CMP    | ACTIVE | INFORMATION SCHEMA | NULL         | GPL     |
| INNODB_CMP_RESET | ACTIVE | INFORMATION SCHEMA | NULL         | GPL     |
| INNODB_CMPMEM | ACTIVE | INFORMATION SCHEMA | NULL         | GPL     |
| INNODB_CMPMEM_RESET | ACTIVE | INFORMATION SCHEMA | NULL         | GPL     |
| PERFORMANCE_SCHEMA | ACTIVE | STORAGE ENGINE | NULL         | GPL     |
| rpl_semi_sync_master | ACTIVE | REPLICATION    | semisync_master.so | GPL     |
+-----+-----+-----+-----+-----+
19 rows in set (0.00 sec)

mysql>

```

Ahora se debe habilitar las variables globales de esta replicación ya que por defecto están deshabilitadas.

En el maestro se ejecuta lo siguiente;

```
mysql> set global rpl_semi_sync_master_enabled = 1;
```

```
mysql> set global rpl_semi_sync_master_timeout = 1000;
```

Con esto le decimos que esta activada la replicación semi-síncrona y que el tiempo va a ser de 1 segundo.

Ahora en el esclavo ejecutamos:

```
mysql> set global rpl_semi_sync_slave_enabled = 1;
```

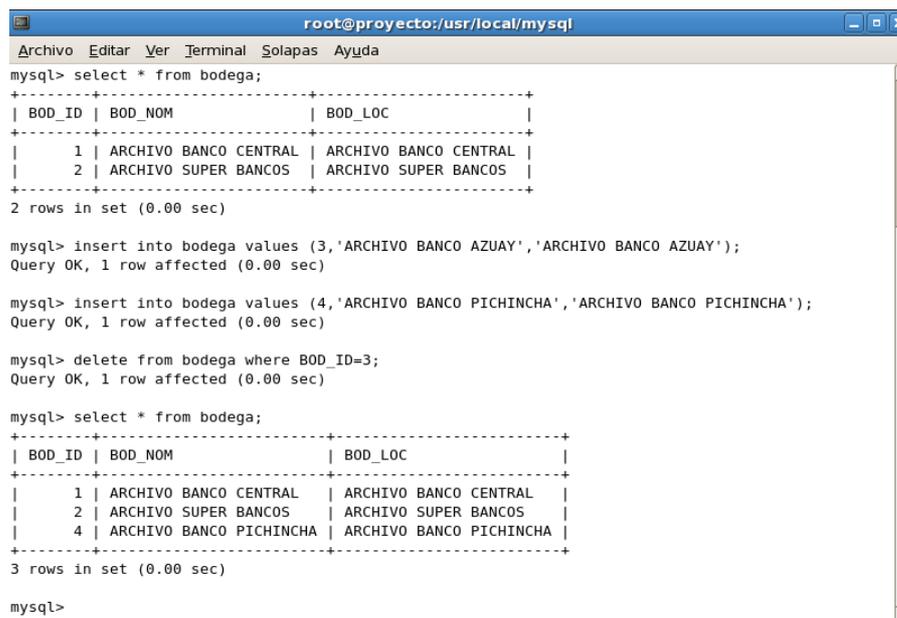
Al igual que en el maestro se debe indicar que la replicación semi-síncrona está activada.

Ahora debemos re-inicializar el hilo de entrada y salida del esclavo.

```
mysql> stop slave io_thread; start slave io_thread;
```

Ahora procederemos a hacer una prueba para comprobar el funcionamiento como estamos en un ambiente de prueba la replicación debe hacerse casi inmediata, por esto vamos a hacer 3 sentencias en el maestro.;

Gráfico 5. Replicación semi-síncrona maestro



```

root@proyecto:/usr/local/mysql
Archivo Editar Ver Terminal Solapas Ayuda
mysql> select * from bodega;
+-----+-----+-----+
| BOD_ID | BOD_NOM          | BOD_LOC          |
+-----+-----+-----+
|      1 | ARCHIVO BANCO CENTRAL | ARCHIVO BANCO CENTRAL |
|      2 | ARCHIVO SUPER BANCOS  | ARCHIVO SUPER BANCOS  |
+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> insert into bodega values (3,'ARCHIVO BANCO AZUAY','ARCHIVO BANCO AZUAY');
Query OK, 1 row affected (0.00 sec)

mysql> insert into bodega values (4,'ARCHIVO BANCO PICHINCHA','ARCHIVO BANCO PICHINCHA');
Query OK, 1 row affected (0.00 sec)

mysql> delete from bodega where BOD_ID=3;
Query OK, 1 row affected (0.00 sec)

mysql> select * from bodega;
+-----+-----+-----+
| BOD_ID | BOD_NOM          | BOD_LOC          |
+-----+-----+-----+
|      1 | ARCHIVO BANCO CENTRAL | ARCHIVO BANCO CENTRAL |
|      2 | ARCHIVO SUPER BANCOS  | ARCHIVO SUPER BANCOS  |
|      4 | ARCHIVO BANCO PICHINCHA | ARCHIVO BANCO PICHINCHA |
+-----+-----+-----+
3 rows in set (0.00 sec)

mysql>

```

Gráfico 6. Replicación semi-síncrona esclavo

```

root@replica:/usr/local/mysql
Archivo Editar Ver Terminal Solapas Ayuda
| nueva |
| prestamos |
| tipodoc |
| user |
+-----+
18 rows in set (0.00 sec)

mysql> install plugin rpl_semi_sync_slave soname 'semisync_slave.so';
Query OK, 0 rows affected (0.10 sec)

mysql> set global rpl_semi_sync_slave_enabled = 1;
Query OK, 0 rows affected (0.01 sec)

mysql> stop slave io_thread; start slave io_thread;
Query OK, 0 rows affected (0.01 sec)

Query OK, 0 rows affected (0.00 sec)

mysql> select * from bodega;
+-----+
| BOD_ID | BOD_NOM | BOD_LOC |
+-----+
| 1 | ARCHIVO BANCO CENTRAL | ARCHIVO BANCO CENTRAL |
| 2 | ARCHIVO SUPER BANCOS | ARCHIVO SUPER BANCOS |
| 4 | ARCHIVO BANCO PICHINCHA | ARCHIVO BANCO PICHINCHA |
+-----+
3 rows in set (0.00 sec)

mysql>

```

Claro que no es sencillo apreciarlo mediante imágenes, así que procedemos a mostrar a mostrar el archivo donde se guardan los cambios, este se puede encontrar de la carpeta `/usr/local/mysql/data` en este caso el nombre del archivo es `proyecto.com.err` donde `proyecto` es el nombre del maestro.

Gráfico 7. Registro de la base de datos Maestro

```

InnoDB: mutexes and rw locks use InnoDB's own implementation
100922 14:55:58 InnoDB: highest supported file format is Barracuda.
100922 14:55:58 InnoDB Plugin 1.0.6 started; log sequence number 44438
100922 14:55:58 [Note] Event Scheduler: Loaded 0 events
100922 14:55:58 [Note] /usr/local/mysql/bin/mysqld: ready for connections.
Version: '5.5.3-m3-log' socket: '/tmp/mysql.sock' port: 3306 MySQL
Community Server (GPL)
100922 15:11:15 [Note] Start binlog_dump to slave_server(2), pos(proyecto-
bin.000004, 456)
100923 11:13:09 [Note] Start binlog_dump to slave_server(2), pos(proyecto-
bin.000013, 343)
100923 12:22:14 [Note] Start binlog_dump to slave_server(2), pos(proyecto-
bin.000013, 509)
100923 13:21:26 [Note] Semi-sync replication initialized for transactions.
100923 13:21:26 [Note] Semi-sync replication enabled on the master.
100923 13:39:52 [Note] Stop asynchronous binlog_dump to slave (server_id: 2)
100923 13:39:52 [Note] Start binlog_dump to slave_server(2), pos(proyecto-
bin.000013, 5725037)
100923 13:39:52 [Note] Start semi-sync binlog_dump to slave (server_id: 2),
pos(proyecto-bin.000013, 5725037)

```

En las dos últimas líneas podemos apreciar que la replicación semi-síncrona fue iniciada con el servidor esclavo.

Si en algún momento queremos reiniciar el servidor y no perder las configuraciones realizadas debemos agregar en el archivo `/etc/my.cnf`

En el maestro

```
[mysqld]
```

```
rpl_semi_sync_master_enabled = 1
```

```
rpl_semi_sync_master_timeout = 1000
```

En el esclavo

```
[mysqld]
```

```
rpl_semi_sync_slave_enabled = 1
```

MANUAL DE USUARIO DEL ODBC

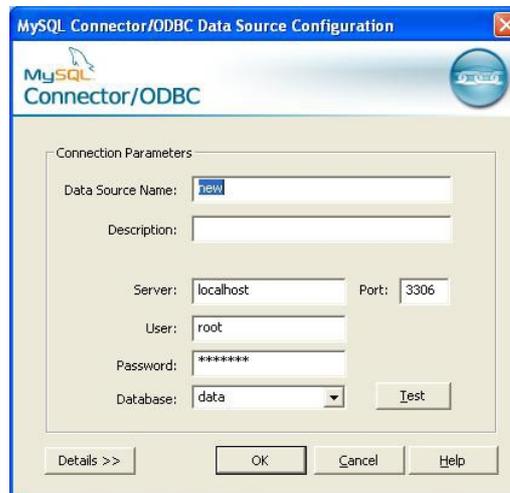
En esta sección se va a proceder a explicar el correcto uso del ODBC modificado en este proyecto. Sólo nos vamos a concentrar en lo que se le ha agregado.

Entre los archivos que tenemos del conector ODBC de MySQL vamos a encontrar un archivo llamado “install.bat” ejecutamos este archivo el cual va a proceder a la instalación del driver de conexión.

La instalación creará un directorio llamado log_queries en el disco c:\ en donde se guardarán los registros de todas las transacciones realizadas por el cliente al servidor.

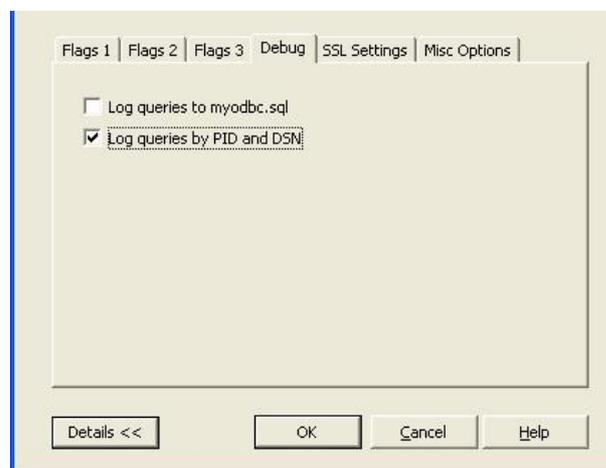
Ahora procedemos a crear un DSN normalmente y damos clic en el botón Details:

Gráfico 8. Configurando el conector ODBC



Se nos desplegará un anexo a la ventana conteniendo varias pestañas. Seleccionamos la pestaña “Debug” y damos click para activar la casilla “Log queries by PID and DSN”, finalmente damos click en el botón “OK” para guardar las configuraciones realizadas.

Gráfico 9. Activando la opción “Log queries by PID and DSN”



Dentro del directorio `c:\log_queries` crearemos un nuevo directorio (nueva carpeta) con el mismo nombre del DSN que creamos anteriormente; aquí se guardarán todas las transacciones generadas en archivos con extensión `.sql`, y sus nombres van a estar

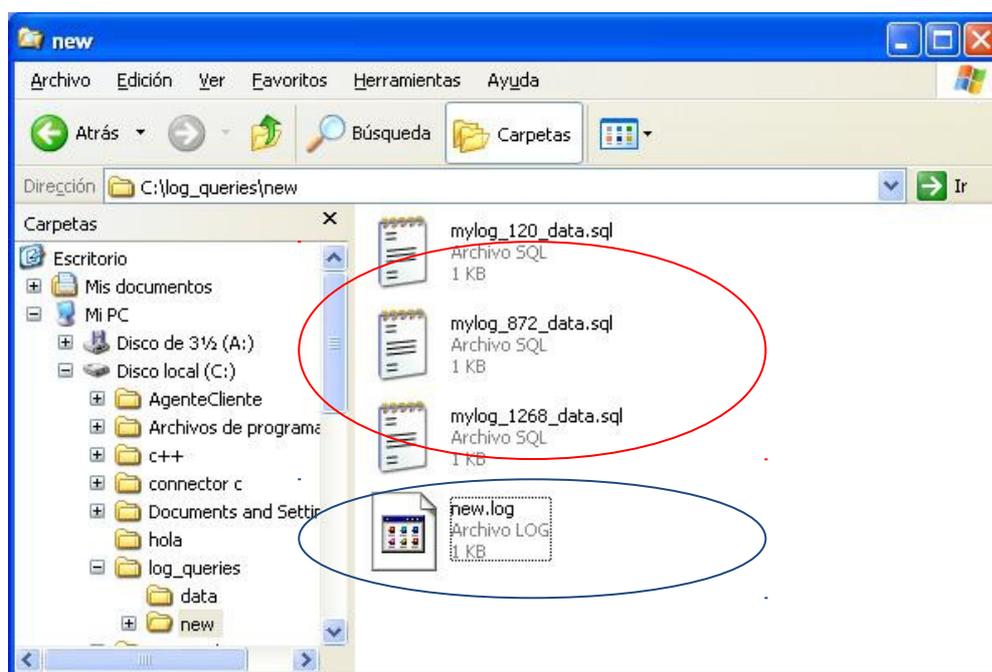
datos de la siguiente manera:

“mylog_NombreDeLaBaseDeDatos_PIDdelPrograma”.

El PID de un programa es el identificador del proceso con que se ejecuta el programa.

Dentro de la carpeta también se creará un archivo con el nombre del DSN y de extensión .log. Este archivo nos indicará si las transacciones están siendo ejecutadas correctamente grabando el estado de “TRANSACCION OK”, y en caso de que haya habido un error se grabará el estado “ERROR”.

Gráfico 10. Archivos donde se guardan las transacciones y el log del DSN.



MANUAL TÉCNICO DEL ODBC

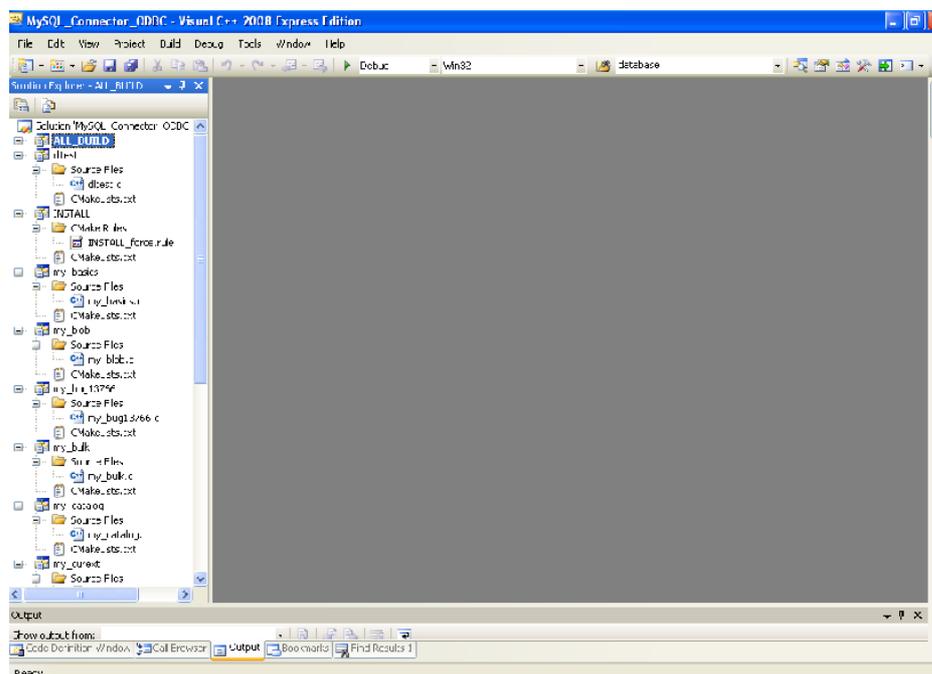
Lo que se plantea hacer como parte de la solución es usar el ODBC para extraer la información que envían los usuarios hacia el servidor. Sólo se explicará lo que se hizo para este proyecto ya que el conector ODBC es un proyecto grande que fue iniciado desde hace varios años.

El compilador que se va a usar es el de Visual Studio 2008 Express Edition.

Luego de haber usado el CMAKE para hacer las configuraciones necesarias procedemos a abrir el archivo.

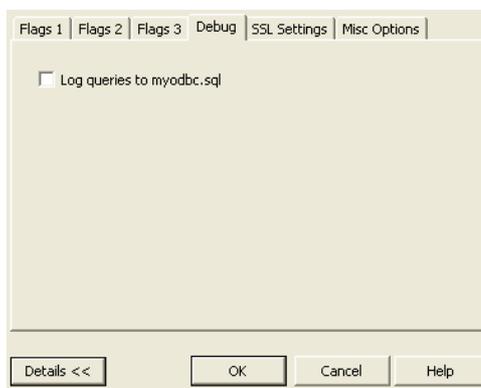
Abrimos el archivo “MySQL_Connector_ODBC.sln” aquí veremos del lado izquierdo dentro de las raíces del árbol las librerías y los archivos en C necesarios para compilar el conector.

Gráfico 11. Abriendo el código fuente del ODBC



Ahora lo primero vamos a poner la opción que vamos a crear dentro la sección Debug en Details del ODBC ya que se relaciona con lo que vamos a hacer.

Gráfico 12. Sección DEBUG en ODBC



Primero en el archivo “C:\mysql-connector-odbc-5.1.6\setupgui\windows\resource.h” agregamos una variable para el checkbox siguiendo el formato de variables creadas.

```
#define IDC_CHECK_save_queries_by_pid_db 10040
```

Ya que los archivos que se van a guardar deben diferenciarse por la aplicación que se usa y por la base de datos, para que no interfiera en las sentencias generada por otras aplicaciones.

Ahora procedemos a crear el checkbox dentro de la pestaña Debug, para esto editamos el archivo "C:\mysql-connector-odbc-5.1.6\setupgui\windows\odbcdialogparams.rc"

```
IDD_TAB4 DIALOGEX 0, 0, 209, 151
```

```
STYLE DS_SETFONT | DS_FIXEDSYS | WS_CHILD
```

```
FONT 8, "MS Shell Dlg", 400, 0, 0x1
```

```
BEGIN
```

```
CONTROL "Log queries to  
myodbc.sql", IDC_CHECK_save_queries,
```

```
"Button", BS_AUTOCHECKBOX |
```

```
WS_TABSTOP, 12, 12, 148, 10
```

```
CONTROL "Log queries by PID and
```

```
DataBase", IDC_CHECK_save_queries_by_pid_db,
```

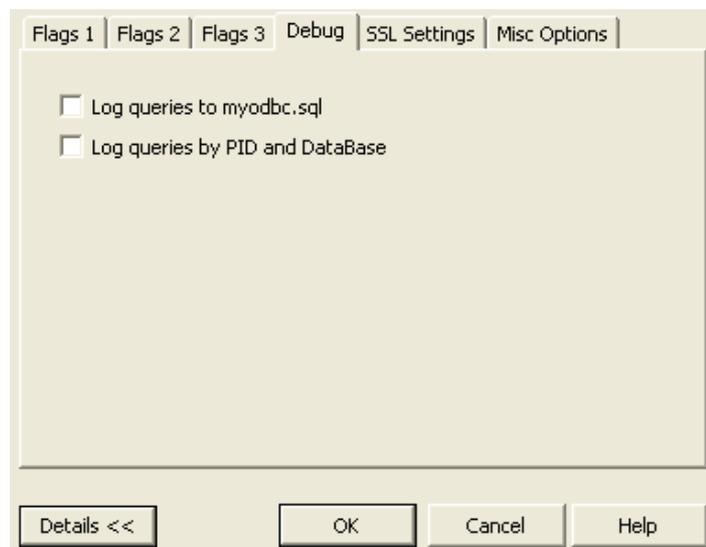
```
"Button", BS_AUTOCHECKBOX |
```

```
WS_TABSTOP, 12, 26, 148, 10
```

```
END
```

Con esto tenemos listo el checkbox si queremos compilamos para visualizarlo, hay que tomar en cuenta que en estos momentos no hace nada.

Gráfico 13. Checkbox para las sentencias creado.



Dentro del archivo “C:\mysql-connector-odbc-5.1.6\util\ installer.h” en la estructura

DataSource creamos la variable:

```
BOOL save_queries_by_pid_db;
```

Esta nos servirá para obtener y escribir el valor en los registros.

En el mismo archivo creamos la macro:

```
#define FLAG_LOG_QUERIES_BY_PID_DB (FLAG_SAFE << 7)
```

Esta se usará para el control del checkbox.

Ahora hay que proceder a grabar en los registros del sistema si se activa esta función, para esto primero creamos el nombre del registro editando el archivo “C:\mysql-connector-odbc-5.1.6\util\ installer.c”

```
static SQLWCHAR W_LOG_QUERY[]=
    {'L','O','G','_','Q','U','E','R','Y',0};
static SQLWCHAR W_LOG_QUERIES_BY_PID_DB[]=
    {'L','O','G','_','Q','U','E','R','I','E','S','_','B','Y','_','P','I','D','_','D','B',0};
static SQLWCHAR W_NO_CACHE[]=
    {'N','O','_','C','A','C','H','E',0};
```

Con el nombre que se le asigna a la variable es el con el cual se va a guardar en los registros.

Ahora en el mismo archivo procedemos a agregar a la cadena que escribe en los registros nuestra variable creada.

```
static const
SQLWCHAR *dsnparams[]= {W_DSN, W_DRIVER, W_DESCRIPTION,
W_SERVER,
W_UID, W_PWD, W_DATABASE, W_SOCKET,
W_INITSTMT,
```

```
W_PORT,          W_OPTION,          W_CHARSET,
W_REPLICATION_SERVER, W_SSLKEY,
                W_SSLCERT, W_SSLCA, W_SSLCAPATH,
W_SSLCIPHER,
                W_SSLVERIFY, W_READTIMEOUT,
W_WRITETIMEOUT,
                W_FOUND_ROWS, W_BIG_PACKETS,
W_NO_PROMPT,
                W_DYNAMIC_CURSOR,          W_NO_SCHEMA,
W_NO_DEFAULT_CURSOR,          W_NO_LOCALE,
W_PAD_SPACE,          W_FULL_COLUMN_NAMES,
W_COMPRESSED_PROTO,
W_IGNORE_SPACE, W_NAMED_PIPE, W_NO_BIGINT,
W_NO_CATALOG,          W_USE_MYCNF,
W_SAFE, W_NO_TRANSACTIONS,          W_LOG_QUERY,
W_LOG_QUERIES_BY_PID_DB, W_NO_CACHE,
W_FORWARD_CURSOR,          W_AUTO_RECONNECT,
W_AUTO_IS_NULL, W_ZERO_DATE_TO_MIN,
W_MIN_DATE_TO_ZERO,
                W_MULTI_STATEMENTS, W_COLUMN_SIZE_S32,
W_NO_BINARY_RESULT,
W_DFLT_BIGINT_BIND_STR, W_ACTIVE_SERVER};
```

En el mismo archivo dentro de la función

```
void ds_map_param(DataSource *ds, const SQLWCHAR *param,
SQLWCHAR ***strdest, unsigned int **intdest, BOOL **booldest)
```

Agregamos lo siguiente:

```
else if (!sqlwcharcasecmp(W_LOG_QUERY, param))
    *booldest= &ds->save_queries;
else if (!sqlwcharcasecmp(W_LOG_QUERIES_BY_PID_DB, param))
    *booldest= &ds->save_queries_by_pid_db;
else if (!sqlwcharcasecmp(W_NO_CACHE, param))
    *booldest= &ds->dont_cache_result;
```

En la función

```
int ds_add(DataSource *ds)
```

Agregamos:

```
if (ds_add_intprop(ds->name, W_LOG_QUERY, ds->save_queries))
goto error;
if (ds_add_intprop(ds->name, W_LOG_QUERIES_BY_PID_DB, ds-
>save_queries_by_pid_db)) goto error;
if (ds_add_intprop(ds->name, W_NO_CACHE, ds-
>dont_cache_result)) goto error;
```

En la función

```
void syncTabs(HWND hwnd, DataSource &params)
```

Agregamos

```
ds->save_queries=          (options & FLAG_LOG_QUERY) > 0;
ds->save_queries_by_pid_db=          (options          &
FLAG_LOG_QUERIES_BY_PID_DB) > 0;
ds->dont_cache_result=      (options & FLAG_NO_CACHE) > 0;
```

En la función

```
ulong ds_get_options(DataSource *ds)
```

Agregamos

```
if (ds->save_queries)
    options|= FLAG_LOG_QUERY;
if (ds->save_queries_by_pid_db)
    options|= FLAG_LOG_QUERIES_BY_PID_DB;
if (ds->dont_cache_result)
    options|= FLAG_NO_CACHE;
```

Ahora en el archivo

“C:\mysql-connector-odbc-5.1.6\setupgui\windows\odbcdialogparams.cpp”

En la función:

```
void syncTabsData(HWND hwnd, DataSource &params)
```

Agregamos:

```
/* debug*/
GET_BOOL(4,save_queries);
GET_BOOL(4,save_queries_by_pid_db);
```

Esto nos permite obtener el valor de esta variable, el número 4 es por el valor que tiene la pestaña Debug

En la función

```
void syncTabs(HWND hwnd, DataSource &params)
```

Agregamos

```
SET_BOOL(DEBUG_TAB, save_queries_by_pid_db);
```

En el archivo

```
C:\mysql-connector-odbc-5.1.6\installer\myodbc3i.c
```

En la función

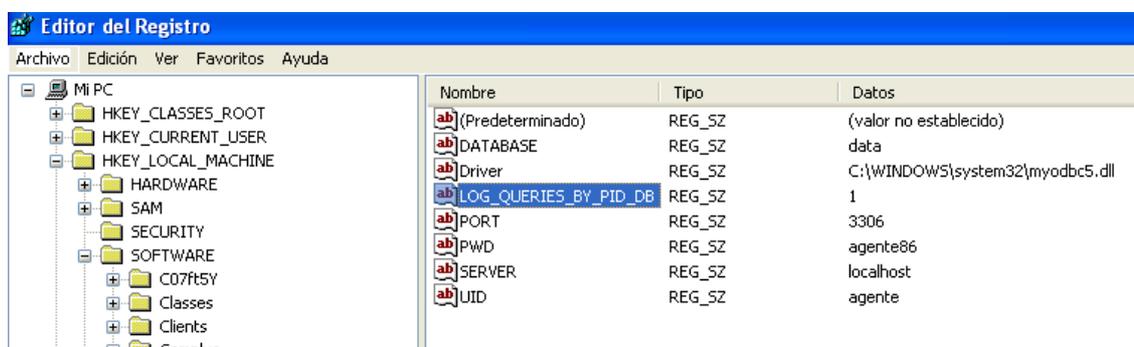
```
int list_datasource_details(DataSource *ds)
```

Agregamos:

```
if (ds->save_queries_by_pid_db)
    printf("\tLOG_QUERIES_BY_PID_DB\n");
```

Estos son los cambios necesarios para guardar y extraer de los registros el valor del checkbox al momento de activar esta opción.

Gráfico 14. Registro agregado en Windows



Ahora hay que escribir lo que van a hacer estas funciones

En la archivo “C:\mysql-connector-odbc-5.1.6\driver\driver.h” definimos las siguientes macros.

```
#if defined(__APPLE__)
#define DRIVER_QUERY_LOGFILE "/tmp/myodbc.sql"
#define DRIVER_BACKUP_FOLDER "/tmp/log_queries/"
#elif defined(_UNIX_)
#define DRIVER_QUERY_LOGFILE "/tmp/myodbc.sql"
#define DRIVER_BACKUP_FOLDER "/tmp/log_queries/"
#else
#define DRIVER_QUERY_LOGFILE "c:\\myodbc.sql"
#define DRIVER_BACKUP_FOLDER "c:\\log_queries\\"
#endif
```

Esta es la carpeta donde se guardarán las sentencias.

En el archivo “C:\mysql-connector-odbc-5.1.6\driver\myutil.h” agregamos:

```
#define MYLOG_BACKUP_QUERY(A,B) {if((A)->dbc->ds-
>save_queries_by_pid_db) \
backup_query_print((A)->dbc,(A)->dbc->my_query_log,(char*)
B);}

#define MYLOG_DBC_BACKUP_QUERY(A,B) {if((A)->ds-
>save_queries_by_pid_db) \
backup_query_print((A),(A)->my_query_log,(char*) B);}

```

Cada vez que agregamos hagamos un llamado a estas macros se llamará a la función que procederá a escribir en los archivos las sentencias generadas por el usuario. Por esta razón en este mismo archivo se debe declarar aquella función

```
void backup_query_print(DBC *dbc, FILE *my_log_file, char
*query);
```

Donde DBC es un puntero que nos permita acceder a la estructura que tiene los valores de conexión a la base de datos.

Ahora en el archivo “C:\mysql-connector-odbc-5.1.6\driver\myutil.h\utility.c” procedemos a escribir la función:

```
void backup_query_print(DBC *dbc, FILE *my_log_file, char
*query)
{
int pid;

char archive_name[70] = DRIVER_BACKUP_FOLDER;

char c_pid[6];

pid = getpid();

itoa(pid, c_pid, 10);

strcat(archive_name, (char*)dbc->name);

#ifdef WIN32

    strcat(archive_name, "\\");

#else

    strcat(archive_name, "/");
```

```

#end if

strcat(archive_name, "mylog_");
strcat(archive_name,c_pid);
strcat(archive_name,"_");
strcat(archive_name,(char*)dbc->database);
strcat(archive_name,".sql");

    if ( (my_log_file = fopen(archive_name, "a+")) )
        fprintf(my_log_file,"%s;\n",query);
    fclose(my_log_file);
    my_log_file= 0;
}

```

Lo que se procede a hacer en esta función es terminar de crear el nombre del archivo para posteriormente pasar a guardar las sentencias, claro que no se puede guardar en una variable global porque se puede usar un solo DSN para que varias aplicaciones accedan a la base de datos lo cual podría hacer una incorrecta combinación de los archivos.

Ahora procedemos a hacer las llamadas necesarias a la función. El archivo que principalmente se usa para ejecutar las sentencias es “C:\mysql-connector-odbc-5.1.6\driver\myutil.h\execute.c”.

En la función:

```
SQLRETURN do_query(STMT FAR *stmt,char *query)
```

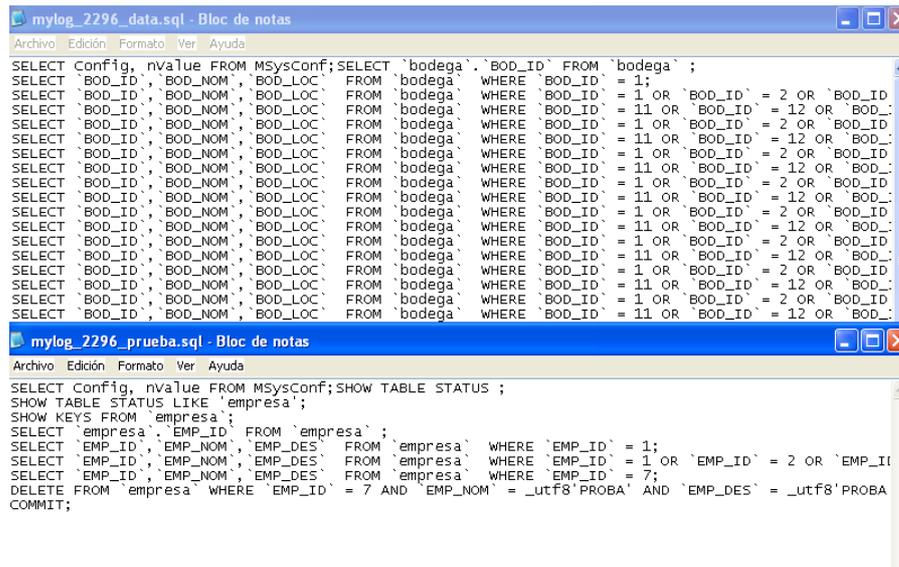
Agregamos

MYLOG_BACKUP_QUERY(stmt,query);

Para mandar a escribir las sentencias.

Ahora procedemos a compilar y a instalar con el archivo Install.bat

Gráfico 15. Archivos Creados



Como podemos darnos cuenta los archivos tienen la numeración “2296” este es el PID del programa en este caso es el mismo porque se ha ingresado a diferentes bases por medio del mismo programa por eso también se escribe el nombre de la base en el archivo.

Para la instalación del agente debemos primero copiar la carpeta AgenteCliente en el disco C:\, luego ejecutamos el archivo “instalador.bat” con esto el Agente iniciará cada vez que se inicie el sistema operativo.

Si se quiere cambiar la ubicación del Agente se debe editar los archivos “instalador.bat” y “ejecutable.bat”.

En el directorio del Agente tenemos tres carpetas más:

- AgenteClienteFuentes: Aquí se encuentra el código fuente del Agente.
- AgenteClienteClases: Aquí se encuentran las clases compiladas y listas para usarse así como también el JDBC para la conexión con la base en caso que se lo necesite.
- Configuraciones: Aquí se encuentra el archivo “Parámetros.txt” en el cual pondremos las configuraciones para el correcto funcionamiento del Agente.

Por cada DSN que tengamos en un cliente deberemos llenar los siguientes parámetros en el archivo “Parámetros.txt”:

dsn = El nombre del DSN que se esté usando.

ipservidor = La IP del servidor escrita en el momento que se creó el DSN.

puerto = El número de puerto que se está usando MySQL (3306 es el valor por omisión).

usuario = El usuario que se usa para conectarse a la base (puede ser el mismo usado al momento de crear el DSN).

contraseña = La contraseña usada por el usuario antes descrito.

baseDatos = Nombre de la base de datos a la que se conecta el DSN.

rutaLogs = La ruta donde se están guardando los archivos .sql generados por el ODBC (en este caso c:\log_queries\).

ipReplicacion = La IP del servidor de replicación que se usará para replicar la información dentro de los archivos .sql en caso de que el servidor principal falle.

replicaciónAutomatica = Si se desea que se haga la replicación inmediatamente cuando se encuentre un error en la comunicación, se deberá poner el valor de 1 caso contrario se pondrá 0.

tiempoLectura = Tiempo de lectura en el cual se va a indicar cada cuantos segundos se debe leer el Log del driver ODBC.

intentosComunicacion = Cantidad de veces que va a intentar comunicarse el agente hacia el servidor esclavo antes de replicar.

tiempoIntento = El tiempo que se va a tomar entre intentos de comunicación, este tiempo se va a medir en segundos.

cambiarIpDsn = Si se desea cambiar la IP de conexión del DSN el valor será de 1, caso contrario el valor debe ser de 0.

rutaBat = Ruta del archivo .bat para cambiar la IP del DSN.

rutaDestinoLogs = Ruta en donde se guardarán los registros leídos después de haber sido replicados.

Si alguna línea empieza con el símbolo '#' no será tomada en cuenta.

MANUAL TÉCNICO DEL AGENTE

En este manual se hablará de cada una de las clases que se usan en el agente, así como del código usado y su explicación.

Clase ProcesosReplicacion

Esta clase será la encargada de replicar la información en los archivos *.sql siempre y cuando exista comunicación con el servidor.

El código es el siguiente:

```
import java.sql.*;
import java.io.File;
import java.io.IOException;
import java.io.BufferedReader;
import java.io.FileReader;
public class ProcesosReplicacion extends LecturaConfiguraciones
{
```

```
    protected int indiceDsn;
```

La variable indiceDsn se va a usar para acceder a los arreglos de la clase LecturaConfiguraciones ya que podemos tener uno o más DSN's y poder tomar la información correcta del mismo DSN, podremos notar estos arreglos más adelante

```
    LecturaConfiguraciones lecturaCL;
```

```

public ProcesosReplicacion(int indiceDsn)
{
    this.indiceDsn = indiceDsn;
    lecturaCL = new LecturaConfiguraciones();
    try
    {
        Class.forName("com.mysql.jdbc.Driver").newInstance();
    } catch (Exception ex) {}
    Cargamos el controlador del jdbc para conectarnos a la base si es
    necesario

}
public void probarComunicacionBasePrincipal()
{
    String cadenaConexion;
    try
    {
        cadenaConexion = "jdbc:mysql://" +
super.ipServidor[this.indiceDsn]
        + "/" + super.baseDatos[this.indiceDsn] + "?"
        + "user=" + super.usuario[this.indiceDsn] + "&"
        + "password=" + super.contraseña[indiceDsn];
        DriverManager.getConnection(cadenaConexion);
        Intentamos conectarnos a la base si no nos podemos conectar nos
        vamos por el error.
    } catch (SQLException e)
    {
        probarComunicacionBaseSecundaria();
    }
    Si no podemos conectarnos con el servidor probamos la comunicación
    con el servidor de replicación.
}

public void probarComunicacionBaseSecundaria()
{
    String cadenaConexion;
    try
    {

```

```

                cadenaConexion = "jdbc:mysql://" +
super.ipReplicacion[this.indiceDsn]
                + "/" + super.baseDatos[this.indiceDsn] + "?"
                + "user=" + super.usuario[this.indiceDsn] + "&"
                + "password=" + super.contraseña[indiceDsn];
                DriverManager.getConnection(cadenaConexion);

```

Ahora probamos la comunicación con la base de replicación si se logra conectar el agente procede a ir a la función para replicar las sentencias en los archivos .sql.

```

    }catch(SQLException e){}
}
void ReplicarQueries()
{
    Connection conn = null;
    Statement instruccion;
    ResultSet tabla;
    String linea, directorioLog;
    File directorio;
    directorioLog = lecturaCL.rutaLogs[indiceDsn] +
lecturaCL.dsn[indiceDsn] + File.separator +
lecturaCL.dsn[indiceDsn] + ".log";
    directorio = new File(directorioLog);
    String [] ficheros = directorio.list();
    Tomamos todos los archivos que se encuentren dentro del directorio y

    los leemos uno por uno para empezar la replicación.

    for (int i =0;i < ficheros.length;i++)
    try
    {
        BufferedReader br = new BufferedReader(new
        FileReader(directorioLog + ficheros[i]));
        while((linea=br.readLine())!=null)
        {
            try
            {
                cadenaConexion = "jdbc:mysql://" +
super.ipReplicacion[this.indiceDsn]
                + "/" + super.baseDatos[this.indiceDsn] + "?"
                + "user=" + super.usuario[this.indiceDsn] + "&"
                + "password=" + super.contraseña[indiceDsn];

```

```

                                conn
DriverManager.getConnection(cadenaConexion);
                                =
                                instruccion = conn.createStatement();
                                tabla = instruccion.executeQuery(linea);
                                }
                                catch(SQLException ex){}
                                Leemos y ejecutamos línea por línea hasta haber terminado
                                todos los archivos.
                                }
                                }
                                catch(IOException e){}
                                }
                                }
}

```

Clases LecturaLogs

Esta clase va a leer el log que va a ser generado por cada DSN.

```

import java.io.File;
import java.io.IOException;
import java.io.RandomAccessFile;
public class LecturaLogs extends Thread
{
    protected Thread hiloLectura;
    protected int indiceDsn;
    LecturaConfiguraciones lecturaCL;
    public LecturaLogs(int indiceDsn)
    {
        this.indiceDsn = indiceDsn;
        lecturaCL = new LecturaConfiguraciones();
        hiloLectura = new Thread(this);
        hiloLectura.setName(Integer.toString(indiceDsn));
        hiloLectura.start();
    }
    public void leerLog()
    {
        RandomAccessFile raf;
        String leerLinea, escribirLinea;
        String directorioLog;
        long lineaPosInicial = 0, lineaPosFinal = 0;
        Las variables declaradas hasta este momento van a servir para leer las
        líneas del log y modificarlas una vez leídas
    }
}

```

```

directorioLog      =      lecturaCL.rutaLogs[indiceDsn]      +
lecturaCL.dsn[indiceDsn]      +      File.separator      +
lecturaCL.dsn[indiceDsn] + ".log";
try
{
    raf = new RandomAccessFile(directorioLog,"rw");
    while((leerLinea=raf.readLine())!=null)
    {
        if(leerLinea.startsWith("ERROR"))
        {
            ProcesosReplicacion      rq      =      new
            ProcesosReplicacion(indiceDsn);
            rq.probarComunicacionBasePrincipal(1);
        }
        Si se detecta algún error llamamos a la clase
        ProcesosReplicacion para que haga su trabajo

        if(!leerLinea.startsWith("--"))
        {
            Si la línea empieza con "--" significa que ya ha sido
            leída y no se la toma en cuenta caso contrario se la lee y
            se le agrega "--" para no volverla a leer.

            lineaPosFinal      =      raf.getFilePointer()      +
            lecturaCL.saltoLinea.length();
            lineaPosInicial      =      raf.getFilePointer()      -
            leerLinea.length()
            - lecturaCL.saltoLinea.length();

            escribirLinea      =      "--"      +      leerLinea      +
            lecturaCL.saltoLinea;

            while((leerLinea=raf.readLine())!=null)
            escribirLinea = escribirLinea + leerLinea +
            lecturaCL.saltoLinea;
            raf.seek(lineaPosInicial);
            raf.writeBytes(escribirLinea);
            raf.seek(lineaPosFinal);
        }
    }
    raf.close();
}

```

```

        }
        catch(IOException e){}
    }
    public void run()
    {
        synchronized(this)
        {
            while(true)
            {
                leerLog();
                try
                {
                    Thread.sleep(3*1000);
                }
                catch(InterruptedException ex){}
            }
        }
    }
}

```

Aquí procedemos a leer los archivos indefinidamente hasta que se termine la ejecución del agente.

Clase LecturaConfiguraciones

Esta clase es la que va a leer las configuraciones iniciales del archivo

“Parámetros.txt”

```

import java.io.FileReader;
import java.io.BufferedReader;
import java.io.IOException;

public class LecturaConfiguraciones
{
    protected String archivoParametros;
    protected String saltoLinea;
    protected String dsn[];
    protected String ipServidor[];
    protected String puerto[];
    protected String usuario[];
    protected String contraseña[];
}

```

```
protected String baseDatos[];
protected String rutaLogs[];
protected String ipReplicacion[];
protected String replicacionAutomatica[];
protected int cantidad_dsn;
```

```
public LecturaConfiguraciones()
{
```

```
    String linea;
    BufferedReader br;
    cantidad_dsn = 0;
    archivoParametros = System.getProperty("archivo_parametros");
```

Es el parámetro que se le envía al momento de ejecutar la clase desde línea de comando, va a guardar la dirección donde se encuentran las configuraciones

```
    saltoLinea = System.getProperty("line.separator");
```

```
    try
    {
        br = new BufferedReader(new FileReader(archivoParametros));
        while((linea=br.readLine())!=null)
        {
            if(!linea.startsWith("#"))
            {
                if(linea.startsWith("dsn="))
                    cantidad_dsn++;
            }
        }
    }
    catch(IOException e){}
```

```
    dsn = new String[cantidad_dsn];
    ipServidor = new String[cantidad_dsn];
    puerto = new String[cantidad_dsn];
    usuario = new String[cantidad_dsn];
    contraseña = new String[cantidad_dsn];
    baseDatos = new String[cantidad_dsn];
    rutaLogs = new String[cantidad_dsn];
    ipReplicacion = new String[cantidad_dsn];
    replicacionAutomatica = new String[cantidad_dsn];
```

Se van a crear arreglos de variables dependiendo de cuántas configuraciones de DSN's se van a encontrar en el archivo "Parámetros.txt"

```

try
{
    int i = -1;
    br = new BufferedReader(new FileReader(archivoParametros));
    while((linea=br.readLine())!=null)
    {
        if(!linea.startsWith("#"))
        {
            if(linea.startsWith("dsn="))
            {
                i++;
                dsn[i] = linea.substring(linea.indexOf("=") + 1);
            }
            if(linea.startsWith("ipServidor=")&&(i!=-1))
                ipServidor[i] = linea.substring(linea.indexOf("=") + 1);
            if(linea.startsWith("puerto=")&&(i!=-1))
                puerto[i] = linea.substring(linea.indexOf("=") + 1);
            if(linea.startsWith("usuario=")&&(i!=-1))
                usuario[i] = linea.substring(linea.indexOf("=") + 1);
            if(linea.startsWith("contraseña=")&&(i!=-1))
                contraseña[i] = linea.substring(linea.indexOf("=") + 1);
            if(linea.startsWith("baseDatos=")&&(i!=-1))
                baseDatos[i] = linea.substring(linea.indexOf("=") + 1);
            if(linea.startsWith("rutaLogs=")&&(i!=-1))
                rutaLogs[i] = linea.substring(linea.indexOf("=") + 1);
            if(linea.startsWith("ipReplicacion=")&&(i!=-1))
                ipReplicacion[i] = linea.substring(linea.indexOf("=") +
1);
            if(linea.startsWith("replicacionAutomatica=")&&(i!=-
1))
                replicacionAutomatica[i] =
linea.substring(linea.indexOf("=") + 1);
        }
    }
    br.close();
}
catch(IOException e){ }
}

```

Este último proceso va a indicar que cada vez que se encuentre una línea del parámetro dsn comience a guardarlo en el número x de los arreglos.

Clase AgenteCliente

Esta clase va a llevar la función main y es la que inicia todo el proceso creando e iniciando los objetos de las otras clases que se van a usar. Su código es el siguiente:

```
public class AgenteCliente
{
    public static void main(String [] arg)
    {
        LecturaConfiguraciones      lecturaC      =      new
LecturaConfiguraciones();
```

Creamos e iniciamos el objeto de la clase LecturaConfiguraciones para leer los parámetros del archivo “Parámetros.txt”, así como inicializar parámetros globales.

```
        for(int i=0;i<lecturaC.cantidad_dsn;i++)
        {
            System.out.println(i);
            LecturaLogs lecturaL = new LecturaLogs(i);
            lecturaL.start();
        }
    }
}
```

También creamos un objeto de la clase LecturaLogs por cada configuración de DSN encontrado e inicializamos los hilos para que empiecen a leer los logs.