



**ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO**

**FACULTAD DE INFORMÁTICA Y ELECTRÓNICA**

**ESCUELA DE INGENIERÍA EN SISTEMAS**

**“ESTUDIO COMPARATIVO ENTRE SISTEMAS DE GESTIÓN DE BASES DE  
DATOS LIBRES FIREBIRD Y MYSQL EN EL DESARROLLO DE  
APLICACIONES WEB. CASO PRACTICO: SISTEMA PARA LA GESTIÓN DE  
EQUIPOS DE RADIO DEL ESCUADRÓN DE COMUNICACIONES N° 11”**

**TESIS DE GRADO**

**Previa a la obtención del título de**

**INGENIERO EN SISTEMAS INFORMÁTICOS**

**Presentado por:**

**LOURDES VERÓNICA PILCO GUACHI**

**RIOBAMBA - ECUADOR**

**2011**

## **AGRADECIMIENTO**

En primer lugar quiero agradecer a Dios por la inteligencia y sabiduría que me dio al nacer. Gracias por darme las fuerzas necesarias en los momentos en que más las necesité y bendecirme con la posibilidad de caminar a tu lado durante toda mi vida.

A cada uno de los que son parte de mi familia, a mi PADRE que está en alguna parte cerca de Dios y que ha estado conmigo todo el tiempo, a mi MADRE, a mis hermanos; por siempre haberme dado su fuerza y apoyo incondicional.

Agradezco sinceramente a mi director de tesis Dr. Julio Santillán sus conocimientos, sus consejos, su paciencia y su motivación han sido fundamentales para mi formación. A todos los que contribuyeron de una u otra forma apoyándome desinteresadamente.

De igual manera agradecer a mis profesores a quienes les debo gran parte de mis conocimientos, gracias a su paciencia y enseñanza; y como olvidarme de mi querida Institución la cual abre sus puertas a jóvenes como nosotros, preparándonos y formándonos como personas y profesionales de excelencia.

## **DEDICATORIA**

Dedico esta Tesis:

A Dios que me da fortaleza espiritual en los momentos difíciles y ha estado conmigo en cada paso que doy cuidándome y dándome fuerzas para continuar.

A mis padres pilares fundamentales en mi vida, quienes han velado por mi bienestar y educación, por sus sabios consejos y por estar a mi lado en los momentos difíciles.

A mi madre con mucho amor y cariño le dedico todo mi esfuerzo y trabajo puesto para la realización de esta tesis.

A mi padre, quien desde el cielo me guía y estoy segura que está orgulloso de mí.

A mis hermanos quienes me han acompañado y han estado siempre dispuestos a ayudarme.

A mis sobrinitos quienes me han contagiado la alegría de vivir.

**NOMBRE**

**FIRMA**

**FECHA**

Ing. Iván Menes

\_\_\_\_\_

\_\_\_\_\_

**DECANO FACULTAD INFORMÁTICA  
Y ELECTRÓNICA**

Ing. Raúl Rosero

\_\_\_\_\_

\_\_\_\_\_

**DIRECTOR DE LA ESCUELA  
DE INGENIERÍA EN SISTEMAS**

Dr. Julio Santillán

\_\_\_\_\_

\_\_\_\_\_

**DIRECTOR DE TESIS**

Ing. Wladimir Castro

\_\_\_\_\_

\_\_\_\_\_

**MIEMBRO DEL TRIBUNAL**

Lcdo. Carlos Rodríguez

\_\_\_\_\_

\_\_\_\_\_

**DIRECTOR CENTRO DE  
DOCUMENTACIÓN**

**NOTA DE LA TESIS**

\_\_\_\_\_

“Yo, LOURDES VERÓNICA PILCO GUACHI, soy responsable de las ideas y resultados expuestos en esta tesis; y el patrimonio intelectual de la Tesis de Grado pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO”.

---

Lourdes Verónica Pilco Guachi

# ÍNDICE GENERAL

PORTADA

AGRADECIMIENTO

DEDICATORIA

ÍNDICE GENERAL

ÍNDICE DE ABREVIATURAS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

CAPÍTULO I

MARCO REFERENCIAL ..... 17

1.1. ANTECEDENTES..... 17

1.2. JUSTIFICACIÓN DEL PROYECTO DE TESIS..... 19

1.3. OBJETIVOS ..... 21

1.4. HIPÓTESIS..... 22

CAPÍTULO II

MARCO TEÓRICO..... 23

2.1. INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS ..... 23

2.1.1. Bases de Datos ..... 23

2.1.2. Componentes principales de una Base de Datos..... 24

2.1.3. El Sistema Gestor de Bases de Datos..... 25

2.1.4. Funciones de los Sistemas de Gestión de Base de Datos..... 26

2.1.5. Ventajas de los Sistemas de Bases de Datos ..... 31

2.1.6. Componentes de un Sistema de Gestión de Base de Datos..... 35

2.1.7. Arquitectura de un Sistema de Base de Datos..... 36

2.1.8. SGBD libres ..... 38

2.2. BASES DE DATOS EN FIREBIRD ..... 42

2.2.1. Introducción ..... 42

2.2.2. Historia..... 42

2.2.3. Características ..... 46

2.2.4.	Tipos de servidores Firebird.....	50
2.2.5.	Tipos de datos .....	53
2.2.6.	Instalación y puesta en marcha.....	57
2.2.6.1	Instalando el motor.....	57
2.2.6.2	Probar la instalación.....	58
2.2.7.	El cliente de Firebird en modo texto .....	60
2.2.7.1	Conectar a la base de datos de ejemplo.....	61
2.2.7.2	Crear una base de datos usando isql.....	62
2.2.8.	Mi primera base de datos .....	64
2.2.9.	Herramientas .....	65
2.2.10.	Soporte Firebird .....	68
2.2.11.	Conexión y uso de bases de datos Firebird en lenguaje PHP.....	68
2.3.	BASES DE DATOS EN MYSQL .....	70
2.3.1.	Introducción .....	70
2.3.2.	Historia.....	71
2.3.3.	Características .....	75
2.3.4.	Tipos de datos .....	78
2.3.5.	Motores de almacenamiento .....	87
2.3.6.	Instalación y puesta en marcha.....	92
2.3.6.1	Configuración.....	93
2.3.7.	Conectarse y desconectarse al servidor.....	96
2.3.8.	Administración de una Base de Datos.....	97
2.3.8.1	Creación de una Base de Datos.....	97
2.3.8.2	Creación de tablas .....	98
2.3.8.3	Ingreso de datos a las tablas .....	99
2.3.8.4	Extraer información de una tabla .....	100
2.3.9.	Programas cliente y utilidades MySQL .....	101
2.3.10.	MySQL GUI Tools .....	103
2.3.11.	Soporte MySQL .....	106
2.3.12.	Conexión y uso de bases de datos MySQL en lenguaje PHP .....	107
<b>CAPÍTULO III</b>		
	<b>ESTUDIO COMPARATIVO DE MYSQL Y FIREBIRD .....</b>	<b>110</b>
3.1.	Introducción .....	110

3.2.	Descripción del problema .....	111
3.3.	Recursos Hardware y Software utilizados .....	112
3.4.	Diseño de la Base de Datos.....	112
3.5.	Definición de los parámetros de evaluación .....	113
3.6.	Determinación de los escenarios de prueba .....	115
3.7.	Diseño de las pruebas.....	116
3.8.	Ejecución de las pruebas en los SGBD .....	122
3.8.1.	MySQL.....	122
3.8.2.	Firebird.....	124
3.9.	Resultados de las pruebas.....	127
3.9.1.	Comparación del tiempo de ejecución en la inserción de datos .....	127
3.9.2.	Consultas de diversa complejidad.....	128
3.9.3.	Comparación del tiempo de respuesta en consultas simples.....	130
3.9.4.	Comparación del tiempo de respuesta en consultas de complejidad media.....	131
3.9.5.	Comparación del tiempo de respuesta en consultas de complejidad alta.....	132
3.9.6.	Comparación del tiempo de respuesta en la eliminación .....	133
3.9.7.	Comparación del tiempo de respuesta para la actualización .....	134
3.9.8.	Prueba de Interrupción eléctrica.....	135
3.10.	Modelo de puntuación.....	136
3.11.	Resultados finales y análisis de los datos obtenidos .....	137
3.12.	Principales características comparadas de MySQL y Firebird.....	141
3.12.1.	Comparativa general .....	141
3.12.2.	Licenciamiento.....	142
3.12.3.	Características generales .....	143
3.12.4.	Comparativa de características generales.....	145
3.12.5.	Límites .....	146
3.12.6.	Características técnicas .....	148
3.12.7.	Comparativa de tipos de datos .....	149
3.12.8.	Herramientas de administración.....	150
3.12.9.	Soporte .....	151
3.12.10.	Otros.....	151
3.13.	Comparativa de características principales de MySQL y Firebird.....	153
3.14.	Comparación general de los SGBD MySQL y Firebird .....	153

3.15.	Comprobación de la hipótesis .....	154
-------	------------------------------------	-----

## CAPÍTULO IV

SISTEMA PARA LA GESTIÓN DE EQUIPOS DE RADIO DEL ESCUADRÓN DE COMUNICACIONES N° 11 .....	157
---	-----

4.1.	Introducción .....	157
------	--------------------	-----

4.2.	Descripción del problema .....	158
------	--------------------------------	-----

4.3.	Metodología de desarrollo.....	158
------	--------------------------------	-----

4.4.	Planificación.....	159
------	--------------------	-----

4.4.1.	Historias de usuario.....	159
--------	---------------------------	-----

4.4.2.	Plan de entregas.....	162
--------	-----------------------	-----

4.4.3.	Iteraciones .....	163
--------	-------------------	-----

4.4.4.	Análisis de riesgos.....	171
--------	--------------------------	-----

4.4.5.	Estudio de la Factibilidad.....	173
--------	---------------------------------	-----

4.5.	Cronograma de trabajo.....	177
------	----------------------------	-----

4.6.	Diseño .....	177
------	--------------	-----

4.6.1.	Arquitectura del sistema.....	177
--------	-------------------------------	-----

4.6.2.	Esquema de la base de datos .....	178
--------	-----------------------------------	-----

4.6.3.	Diagrama de componentes .....	178
--------	-------------------------------	-----

4.6.4.	Diagrama de despliegue .....	179
--------	------------------------------	-----

4.7.	Implementación.....	179
------	---------------------	-----

4.7.1.	Introducción .....	179
--------	--------------------	-----

4.7.2.	Implementación de las iteraciones .....	179
--------	---	-----

4.7.3.	Integración.....	181
--------	------------------	-----

4.7.4.	Prototipos interfaces de usuario .....	182
--------	--	-----

4.8.	Pruebas .....	189
------	---------------	-----

## CONCLUSIONES

## RECOMENDACIONES

## RESUMEN

## SUMMARY

## GLOSARIO DE TÉRMINOS

## ANEXOS

## BIBLIOGRAFÍA

## ÍNDICE DE ABREVIATURAS Y ACRÓNIMOS

**ACID.** Atomicity, Consistency, Isolation and Durability

**ANSI/SPARC.** American National Standard Institute - Standards Planning and Requirements Committee

**API.** Application programmign Interface (Interfaz de programación de aplicaciones)

**BD.** Base de Datos

**BLOB.** Binary Large Object

**DBA.** Data Base Administrator (Administrador de la base de datos)

**DDL.** Data Definition Language (Lenguaje de definición de datos)

**DML.** Data Manipulation Language (Lenguaje de manipulación de datos)

**GNU.** General Public License

**IEEE.** Institute of Electrical and Electronics Engineers

**IPL.** Licencia Pública InterBase

**ISQL.** Interactive SQL

**JDBC.** Java DataBase Connectivity

**MB.** MegaBytes

**MPL.** Mozilla Public Licencia

**ODBC.** Open DataBase Connectivity

**OLEDB.** Object Linking and Embedding for Databases

**PHP.** Hypertext Preprocessor

**SGBD.** Sistema de Gestión de Base de Datos

**SQL.** Structured Query Language (Lenguaje de consulta estructurado)

**SMP.** Symmetric Multi-Processing

**XP.** Programación Extrema

## ÍNDICE DE FIGURAS

Figura II.1 Esquema del funcionamiento y utilidad de un SGBD.....	26
Figura II.2 Componentes de un SGBD .....	35
Figura II.3 Niveles de abstracción de la arquitectura ANSI .....	38
Figura II.4 Seleccionando tipo de servidor .....	57
Figura II.5 Selección de tareas adicionales .....	58
Figura II.6 Comprobando que Firebird esté ejecutándose .....	59
Figura II.7 Firebird Server Control .....	60
Figura II.8 Acceso al cliente Firebird desde el menú Inicio .....	60
Figura II.9 Cliente Firebird en modo texto .....	61
Figura II.10 IBExpert.....	65
Figura II.11 Configurando MySQL .....	94
Figura II.12 Asignando contraseña al usuario root .....	95
Figura II.13 Cliente MySQL en modo texto .....	96
Figura II.14 MySQL Administrator .....	104
Figura II.15 MySQL Query Browser.....	105
Figura III.16 Implementación de las operaciones en MySQL .....	122
Figura III.17 MySQL Command Line Client – Inserción.....	123
Figura III.18 MySQL Command Line Client – Actualización .....	123
Figura III.19 MySQL Query Browser.....	124
Figura III.20 Implementación de las operaciones en Firebird.....	125
Figura III.21 IBExpert – Inserción.....	125
Figura III.22 IBExpert - Eliminación.....	126
Figura III.23 IBExpert – Selección .....	126
Figura III.24 Tiempo de respuesta en la inserción de datos.....	127
Figura III.25 Tiempo de respuesta en consultas de diversa complejidad MySQL.....	128
Figura III.26 Tiempo de respuesta en consultas de diversa complejidad Firebird.....	129
Figura III.27 Tiempo de respuesta en consultas simples .....	130
Figura III.28 Tiempo de respuesta en consultas de complejidad media .....	132
Figura III.29 Tiempo de respuesta en consultas de complejidad alta .....	133
Figura III.30 Tiempo de respuesta para la eliminación.....	134
Figura III.31 Tiempo de respuesta en la actualización.....	135
Figura III.32 Resultados finales por parámetros .....	139
Figura III.33 Licenciamiento .....	142
Figura III.34 Características generales.....	146
Figura III.35 Límites .....	147
Figura III.36 Características técnicas.....	149
Figura III.37 Soporte.....	151

Figura III.38 Otros .....	152
Figura III.39 Promedio total.....	156
Figura IV.40 Cronograma .....	177
Figura IV.41 Arquitectura del sistema.....	177
Figura IV.42 Base de datos .....	178
Figura IV.43 Diagrama de componentes .....	178
Figura IV.44 Diagrama de despliegue .....	179
Figura IV.45 Validación de usuarios .....	182
Figura IV.46 Opciones del usuario administrador .....	183
Figura IV.47 Ingresar usuarios.....	183
Figura IV.48 Buscando usuario a eliminar .....	184
Figura IV.49 Eliminar usuarios.....	184
Figura IV.50 Buscando usuario a actualizar .....	185
Figura IV. 51 Actualizar usuarios .....	185
Figura IV. 52 Listar usuarios .....	186
Figura IV.53 Ingresar equipos/materiales .....	186
Figura IV.54 Eliminar equipos/materiales .....	187
Figura IV.55 Préstamos de equipos/materiales .....	187
Figura IV.56 Presentación de mensaje.....	188
Figura IV.57 Presentación de mensaje.....	188
Figura IV.58 Consulta de usuarios.....	189
Figura IV.59 Insertando datos.....	190
Figura IV.60 Verificando datos en la BD .....	191
Figura IV.61 Mensaje de error .....	191
Figura IV.62 Eliminando datos .....	192
Figura IV.63 Verificando la eliminación en la BD .....	192
Figura IV.64 Modificando datos .....	193
Figura IV.65 Verificando actualización en la BD.....	194
Figura IV.66 Realizando préstamos.....	195
Figura IV.67 Confirmación de préstamo realizado.....	195
Figura IV.68 Verificando préstamos en la BD.....	196
Figura IV.69 Mensaje de error .....	196
Figura IV.70 Realizando otro préstamo .....	197
Figura IV.71 Verificando datos en la BD .....	197
Figura IV.72 Consultando préstamos.....	198
Figura IV.73 Reporte en formato pdf de todos los usuarios .....	199

## ÍNDICE DE TABLAS

Tabla II.I. Tipos de datos numéricos en Firebird .....	54
Tabla II.II. Tipos de datos fecha y hora en Firebird.....	55
Tabla II.III. Tipos de datos cadena de caracteres en Firebird .....	56
Tabla II.IV. Tipos de datos especiales en Firebird.....	57
Tabla II.V. Tipos de datos numéricos en MySQL .....	81
Tabla II.VI. Tipos de datos fecha y hora en MySQL.....	83
Tabla II.VII. Tipos de datos cadenas de caracteres en MySQL.....	86
Tabla II.VIII. Motores de almacenamiento.....	92
Tabla II.IX. Atributos de columna .....	98
Tabla II.X. Restricciones de tabla.....	99
Tabla III.XI. Parámetros .....	114
Tabla III.XII. Prueba N° 01.....	116
Tabla III.XIII. Prueba N° 02 .....	118
Tabla III.XIV. Prueba N° 03 .....	119
Tabla III.XV. Prueba N° 04.....	119
Tabla III.XVI. Prueba N° 05 .....	120
Tabla III.XVII. Prueba N° 06.....	121
Tabla III.XVIII. Prueba N° 07.....	121
Tabla III.XIX. Tiempo de respuesta en la inserción de datos .....	127
Tabla III.XX. Tiempo de respuesta en consultas de diversa complejidad MySQL .....	128
Tabla III.XXI. Tiempo de respuesta en consultas de diversa complejidad Firebird.....	129
Tabla III.XXII. Tiempo de respuesta en consultas simples .....	130
Tabla III.XXIII. Tiempo de respuesta en consultas de complejidad media.....	131
Tabla III.XXIV. Tiempo de respuesta en consultas de complejidad alta.....	132
Tabla III.XXV. Tiempo de respuesta en la eliminación .....	133
Tabla III.XXVI. Tiempo de respuesta en la actualización.....	134
Tabla III.XXVII. Parámetros con su respectivo valor porcentual.....	136
Tabla III.XXVIII. Escala cuali-cuantitativa para el tiempo de ejecución.....	137
Tabla III.XXIX. Resultados del tiempo de ejecución .....	137
Tabla III.XXX. Evaluación del parámetro tiempo de ejecución.....	138
Tabla III.XXXI. Evaluación del parámetro integridad en los datos.....	138
Tabla III.XXXII. Evaluación del parámetro tolerancia a fallos.....	138
Tabla III.XXXIII. Evaluación de los parámetros.....	138
Tabla III.XXXIV. Sumatoria de los puntos obtenidos por MySQL y Firebird.....	139
Tabla III.XXXV. Comparativa general de las bases de datos.....	141
Tabla III.XXXVI. Peso de Licenciamiento.....	142
Tabla III.XXXVII. Sistemas Operativos soportados .....	143

Tabla III.XXXVIII. Peso de Sistemas Operativos soportados.....	143
Tabla III.XXXIX. Interfaces y conectores soportados.....	144
Tabla III.XL. Peso de Interfaces y Conectores soportados .....	144
Tabla III.XLI. Características fundamentales .....	145
Tabla III.XLII. Peso de características fundamentales .....	145
Tabla III.XLIII. Características generales.....	145
Tabla III.XLIV. Límites.....	146
Tabla III.XLV. Peso de Límites.....	147
Tabla III.XLVI. Características técnicas.....	148
Tabla III.XLVII. Peso Características técnicas.....	148
Tabla III.XLVIII. Peso Tipos de datos.....	149
Tabla III.XLIX. Herramientas de administración .....	150
Tabla III.L. Peso de Herramientas de Administración.....	150
Tabla III.LI. Peso soporte.....	151
Tabla III.LII. Otros.....	152
Tabla III.LIII. Peso otros.....	152
Tabla III.LIV. Evaluación de las características principales.....	153
Tabla III.LV. Evaluación final de las características principales.....	153
Tabla III.LVI. Comparativa general.....	154
Tabla III.LVII. Resultados para los SGBD por parámetros .....	155
Tabla III.LVIII. Promedio total.....	155
Tabla IV.LIX. Plan de entrega N°1 .....	162
Tabla IV.LX. Plan de entrega N°2 .....	162
Tabla IV.LXI. Cronograma de cada historia de usuario .....	163
Tabla IV.LXII. Identificación de riesgos .....	172
Tabla IV.LXIII. Categorización de riesgos.....	172
Tabla IV.LXIV. Gestión de riesgos .....	172
Tabla IV.LXV. Hardware disponible.....	174
Tabla IV.LXVI. Software disponible.....	174
Tabla IV.LXVII. Software requerido.....	174
Tabla IV.LXVIII. Recurso humano requerido.....	175
Tabla IV.LXIX. Costos de desarrollo .....	176
Tabla IV.LXX. Datos de entrada para el ingreso de registros .....	190
Tabla IV.LXXI. Datos de entrada para eliminación de registros.....	191
Tabla IV.LXXII. Datos de entrada para actualización de registros .....	193
Tabla IV.LXXIII. Datos de entrada para realizar préstamos .....	194
Tabla IV.LXXIV. Datos de entrada para consultar préstamos.....	198

## INTRODUCCIÓN

Hoy en día, todas las empresas ya sean grandes, medianas o pequeñas y sin importar su tipo (académicas, comerciales, industriales, etc.) requieren almacenar su información, que es el activo más importante de una organización y uno de los factores más críticos para la determinación del éxito o fracaso.

En el presente contexto, aparecen los Sistemas de Gestión de Base de Datos (SGBD) que se han convertido en las principales herramientas para el almacenamiento de la información. El objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos. Con el advenimiento de Internet, el software libre entre ellos los SGBD se han consolidado como alternativa, técnicamente viable y económicamente sostenible al software comercial, es por ello que se enfocará en dos de los SGBD libres como son MySQL y Firebird.

Primeramente se dará una visión general sobre los conceptos relacionados con las bases de datos y los sistemas gestores de base de datos. El segundo capítulo empezará con los aspectos involucrados sobre base de datos, conceptos, ventajas y componentes principales para posteriormente mencionar conceptos, ventajas y funciones de los SGBD; terminando con un estudio de los SGBD libres más utilizados.

Una vez sentadas estas bases sobre conceptos generales de BD y SGBD, se estará en disposición de profundizar dos de los SGBD libres como son MySQL y Firebird. Así se

empezará con un estudio detallado de Firebird, sus características principales, su uso, funcionamiento, administración y otras particularidades; de igual manera se examinará al motor de base de datos MySQL. También se verán herramientas de administración gráficas para estos dos sistemas gestores de bases de datos, que permitirán mejorar la productividad en el trabajo con ellos.

A continuación, se abordará el método de acceso a estos dos sistemas gestores de bases de datos desde el lenguaje de programación PHP. En cada caso se lo realizará desde una perspectiva totalmente práctica, además se introducirán algunos conceptos relacionados.

Posteriormente se abordará el tercer capítulo, que es el objetivo principal del estudio y que tiene como finalidad determinar cuál de los dos sistemas en comparación es el mejor. Para la toma de esta decisión se realizarán pruebas de inserción, selección, actualización y eliminación a cada SGBD con diferentes cantidades de registros, que serán evaluados de acuerdo a ciertos parámetros y que están dentro de los criterios importantes al momento de seleccionar un SGBD. Además de otros aspectos y características relevantes que no se verán involucrados durante la ejecución de las pruebas pero que si se tiene en cuenta para la elaboración del estudio comparativo.

Finalmente, se entrará al desarrollo del capítulo IV, en el cual se implementará el Sistema para la Gestión de Equipos de Radio del Escuadrón de Comunicaciones N° 11 de la BCB “GALAPAGOS”, utilizando el Sistema de Gestión de Bases de Datos que resultase ganador en el capítulo anterior. Proporcionando así un sistema que será de gran utilidad y que facilite el trabajo diario que se lleva a cabo en esta entidad.

# **CAPÍTULO I**

## **MARCO REFERENCIAL**

### **1.1. ANTECEDENTES**

Las bases de datos son el método preferido para el almacenamiento estructurado de datos. Desde las grandes aplicaciones multiusuario, hasta los teléfonos móviles y las agendas electrónicas utilizan tecnología de bases de datos para asegurar la integridad de los datos y facilitar la labor tanto de usuarios como de los programadores que las desarrollaron.

Desde la realización del primer modelo de datos, pasando por la administración del sistema gestor, hasta llegar al desarrollo de la aplicación, los conceptos y la tecnología asociados son muchos y muy heterogéneos. Sin embargo, es imprescindible conocer los aspectos clave de cada uno de estos temas para tener éxito en cualquier proyecto que implique trabajar con bases de datos.

En la actualidad el software libre entre ellos los sistemas gestores de bases de datos libres, se han consolidado como alternativa, técnicamente viable y económicamente

sostenible, convirtiéndose como una buena opción para ofrecer servicios a un costo reducido. El uso de una base de datos libre, ahorra costos de producción y sobre todo de implementación, brindando beneficios tanto para los productores como para el usuario final. Este estudio se enfocará en la comparación de los Sistemas de Gestión de Bases de Datos libres Firebird y MySQL, Firebird que a pesar que ofrece grandes características no es muy conocido al menos en nuestro entorno a diferencia de MySQL que actualmente es el más difundido y utilizado por los desarrolladores en todo el mundo.

Firebird es un servidor de base de datos multiplataforma que ha sido desarrollado a partir del código fuente de Interbase 6.0 que años atrás Borland Software Corp. entregó al público del software libre bajo la Licencia Pública InterBase el 25 de Julio de 2000.

Es un motor de base de datos de gran capacidad, está a la altura de Oracle, SQL Server, MySQL, cuenta con las características que debe tener una buena base de datos: generadores, triggers, procedimientos almacenados, vistas, roles, funciones, seguridad y aún así, es totalmente desconocida. Además posee varias herramientas de terceros, incluyendo herramientas visuales de administración, replicación, etc. Todo esto lo convierte en una de las mejores alternativas en la actualidad a los motores de base de datos comerciales. Firebird es muy práctico en todo sentido; sencillo de instalar, fácil de usar, requiere poca administración, con requerimientos de hardware bajos y es gratuito.

MySQL es un sistema de gestión de bases de datos relacional. Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos

para cubrir las necesidades de cualquier tipo de organización, desde pequeños establecimientos comerciales a grandes empresas y organismos administrativos. MySQL compite con sistemas RDBMS propietarios como Oracle, Sql Server y Db2.

MySQL incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger los datos. Puede desarrollar sus propias aplicaciones de bases de datos en la mayor parte de lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos, MySQL utiliza el lenguaje de consulta estructurado (SQL).

Por otra parte el Escuadrón de Comunicaciones N° 11 de la Brigada de Caballería Blindada “Galápagos” maneja gran cantidad de información correspondiente a los equipos de radio que se encuentran en bodega, prestados, dañados o en reparación. La administración de esta información actualmente se viene manejando de forma manual, existiendo la posibilidad de que estos datos no sean administrados de forma correcta.

## **1.2. JUSTIFICACIÓN DEL PROYECTO DE TESIS**

Hoy en día la tecnología Open Source ha tomado fuerza y se ha convertido en una alternativa para la solución de problemas empresariales técnicamente viables y económicamente sostenibles al software comercial, una de estas alternativas se ve reflejado en la existencia de Sistemas de Gestión de Bases de Datos libres, este es el caso de Firebird y MySQL.

Por una parte Firebird es un servidor de bases de datos multiplataforma que cuenta con características que debe tener una buena bases de datos como: soporte a procedimientos almacenados, funciones, vistas, triggers, seguridad basada en roles/usuarios, etc.; además consume escasos recursos hardware mucho menos recursos que otros motores de base de datos más difundidos. A pesar de todas estas características que ofrece es poco utilizado y por la mayor parte desconocida, al menos en nuestro entorno, debido a su escasa capacidad publicitaria.

En cambio MySQL es uno de los motores de bases de datos más conocidos y difundidos en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración. Además de poseer todas las características de un Gestor de Bases de Datos.

De ahí nace la importancia del estudio comparativo de estos dos Sistemas de Gestión de Bases de Datos libres, Firebird un motor de bases de datos poco conocido al contrario de MySQL uno de los más utilizados en el mundo; Y sabiendo que Firebird posee gran capacidad con todas las características que debe tener una buena base de datos para la administración de la información de una empresa, siendo así una buena alternativa al igual que MySQL. Con el estudio se determinará un soporte para recomendar a una empresa la utilización de uno de estos SGBD libres de acuerdo al desempeño que ofrezca cada uno de ellos y el que permita mayor optimización de recursos , para lo cual

se tomarán en cuenta factores como: volúmenes de información que soporta, integridad de la información, velocidad entre otros.

El Escuadrón de Comunicaciones N° 11 de la BCB “Galápagos” como se mencionó anteriormente viene manejando gran cantidad de datos que son administrados de forma manual, por lo que requieren de un sistema con bases de datos que permita mejorar la administración y control de la información correspondiente a los equipos de radio existentes en la entidad.

Dicho sistema se encargará de almacenar en una base de datos toda la información de los equipos de radio que se encuentran en: bodega, prestados, en reparación o dañados; permitiendo así llevar un eficiente control de inventario y ofreciendo el almacenamiento confiable de sus datos. Brindando así una respuesta rápida a las necesidades operativas de dicha entidad ya que permitirá optimizar el tiempo de operación de la persona encargada y ofrecer una atención más rápida al personal que utiliza este servicio.

### **1.3. OBJETIVOS**

#### **1.3.1. Objetivo General**

Realizar un estudio comparativo entre Sistemas de Gestión de Bases de Datos libres Firebird y MySQL en el desarrollo de aplicaciones web aplicado al Sistema para la Gestión de Equipos de Radio del Escuadrón de Comunicaciones N° 11 de la Brigada de Caballería Blindada “GALÁPAGOS”.

### **1.3.2. Objetivos Específicos**

- Estudiar las características que ofrecen los Sistemas de Gestión de Bases de Datos libres Firebird y MySQL.
- Determinar los parámetros a ser evaluados entre los Sistemas de Gestión de Bases de Datos libres Firebird y MySQL.
- Seleccionar el Sistema Gestor de Bases de Datos más óptimo para utilizarlo en el desarrollo del Sistema a implementar.
- Desarrollar un Sistema para la Gestión de Equipos de Radio del Escuadrón de Comunicaciones N° 11 de la Brigada de Caballería Blindada “GALAPAGOS”.

### **1.4. HIPÓTESIS**

Con el estudio comparativo entre Sistemas de Gestión de Bases de Datos libres Firebird y MySQL en el desarrollo de aplicaciones web se podrá seleccionar el Sistema Gestor de Bases de Datos que permita la optimización de recursos en el desarrollo de la base de datos del Sistema para la Gestión de los Equipos de Radio del Escuadrón de Comunicaciones N° 11.

## **CAPÍTULO II**

### **MARCO TEÓRICO**

#### **2.1. INTRODUCCIÓN A LOS SISTEMAS DE BASES DE DATOS**

##### **2.1.1. Bases de Datos**

Una base de datos es un conjunto de datos almacenados de forma organizada y estructurada en un soporte de información que es manejado por un ordenador. La información se guarda en archivos independientes integrados en la base, y puede ser compartida por distintos usuarios que la utilicen para fines diferentes en instantes de tiempo que pueden coincidir.

El término de bases de datos fue escuchado por primera vez en 1963, en un simposio celebrado en California, USA. Una base de datos se puede definir como un conjunto de información relacionada que se encuentra agrupada o estructurada.

Desde el punto de vista informático, la base de datos es un sistema formado por un conjunto de datos almacenados en discos que permiten el acceso directo a ellos y un conjunto de programas que manipulen ese conjunto de datos.

El Glosario IEEE de Ingeniería del Software (IEEE<sup>1</sup> 1990) define el término base de datos de la siguiente forma: “Base de datos. Una colección de datos interrelacionados almacenados conjuntamente en uno o más ficheros de computadora”.

### **2.1.2. Componentes principales de una Base de Datos**

**Datos.** Los datos son la Base de Datos propiamente dicha.

**Hardware.** El hardware se refiere a los dispositivos de almacenamiento en donde reside la base de datos, así como a los dispositivos periféricos (unidad de control, canales de comunicación, etc.) necesarios para su uso.

**Software.** Está constituido por un conjunto de programas que se conoce como Sistema Manejador de Base de Datos (DBMS: DataBase Management System). Este sistema maneja todas las solicitudes formuladas por los usuarios a la base de datos.

**Usuarios.** Existen tres clases de usuarios relacionados con una Base de Datos:

1. El programador de aplicaciones, quien crea programas de aplicación que utilizan la base de datos.
2. El usuario final, quien accesa la Base de Datos por medio de un lenguaje de consulta o de programas de aplicación.
3. El Administrador de la Base de Datos (DBA: Data Base Administrator), es quien se encarga del control general del Sistema de Base de Datos. El DBA es la persona que asegura integridad, consistencia, redundancia, seguridad, y se encarga de realizar el mantenimiento diario o periódico de los datos<sup>2</sup>.

---

<sup>1</sup> IEEE. Es la mayor asociación internacional formada por profesionales de las nuevas tecnologías, dedicada a la estandarización.

<sup>2</sup> <http://www.monografias.com/trabajos12/basdat/basdat.shtml>

### 2.1.3. El Sistema Gestor de Bases de Datos

Un SGBD es una colección de programas cuyo objetivo es servir de interfaz entre la base de datos, el usuario y las aplicaciones. Un SGBD se compone de un lenguaje de definición de datos (DDL), de un lenguaje de manipulación de datos (DML) y de un lenguaje de consulta. Un SGBD permite definir los datos a distintos niveles de abstracción y manipular dichos datos, garantizando la seguridad e integridad de los mismos<sup>3</sup>. Algunos ejemplos de SGBD son Oracle, PostgreSQL, Firebird, MySQL, etc.

Un SGBD debe permitir:

- Definir una base de datos: Consiste en especificar los tipos de datos, estructuras y restricciones para los datos que se almacenarán.
- Construir la base de datos: Es el proceso de almacenar/guardar los datos en algún medio de almacenamiento controlado por el mismo SGBD.
- Manipular la base de datos: Incluye funciones como realizar consultas, actualizarla, generar informes.

Otras definiciones de SGBD:

Un Sistema de Gestión de Bases de Datos consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. El objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos.

(KORTH<sup>4</sup>, 1995)

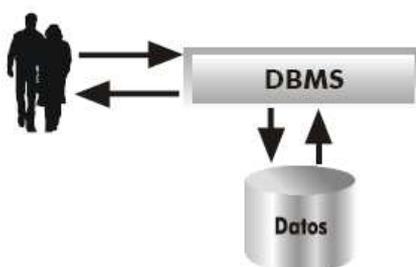
---

<sup>3</sup> <http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/>

<sup>4</sup> KORTH. Lehigh University.

El concepto de Sistema Gestor de Bases de Datos se define en el Glosario IEEE de Ingeniería del Software de la siguiente forma: “Sistema Gestor de Bases de Datos. Un sistema informático compuesto por hardware, software o ambos, que proporciona una técnica sistemática para la creación, el almacenamiento, el procesamiento y la consulta de la información almacenada en base de datos. Un SGBD actúa como un intermediario entre las aplicaciones y los datos, o bien entre los datos y la base de datos”.

En estos sistemas se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos, garantizando además la seguridad de los mismos.



**Figura II.1** Esquema del funcionamiento y utilidad de un SGBD

#### **2.1.4. Funciones de los Sistemas de Gestión de Base de Datos**

Edgar Frank Codd<sup>5</sup>, el creador del modelo relacional, ha establecido una lista con los ocho servicios que debe ofrecer todo SGBD.

1. Un SGBD debe proporcionar a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos. Esta es la función fundamental de un SGBD y por supuesto, el SGBD debe ocultar al usuario la estructura física interna (la organización de los ficheros y las estructuras de almacenamiento).

---

<sup>5</sup> Edgar Frank Codd (1923-2003) creador del modelo de base de datos relacional en 1970.

2. Un SGBD debe proporcionar un catálogo en el que se almacenen las descripciones de los datos y que sea accesible por los usuarios. Este catálogo es lo que se denomina diccionario de datos y contiene información que describe los datos de la base de datos (metadatos). Normalmente, un diccionario de datos almacena:

- Nombre, tipo y tamaño de los datos.
- Nombre de las relaciones entre los datos.
- Restricciones de integridad sobre los datos.
- Nombre de los usuarios autorizados a acceder a la base de datos.
- Esquemas externo, conceptual e interno, y correspondencia entre los esquemas.
- Estadísticas de utilización, tales como la frecuencia de las transacciones y el número de accesos realizados a los objetos de la base de datos.

Algunos de los beneficios que reporta el diccionario de datos son los siguientes:

- La información sobre los datos se puede almacenar de un modo centralizado. Esto ayuda a mantener el control sobre los datos, como un recurso que son.
- El significado de los datos se puede definir, lo que ayudará a los usuarios a entender el propósito de los mismos.
- La comunicación se simplifica ya que se almacena el significado exacto. El diccionario de datos también puede identificar al usuario o usuarios que poseen los datos o que los acceden.
- Las redundancias y las inconsistencias se pueden identificar más fácilmente ya que los datos están centralizados.
- Se puede tener un historial de los cambios realizados sobre la base de datos.

- El impacto que puede producir un cambio se puede determinar antes de que sea implementado, ya que el diccionario de datos mantiene información sobre cada tipo de dato, todas sus relaciones y todos sus usuarios.
- Se puede hacer respetar la seguridad.
- Se puede garantizar la integridad.
- Se puede proporcionar información para auditorías.

3. Un SGBD debe proporcionar un mecanismo que garantice que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna. Una transacción es un conjunto de acciones que cambian el contenido de la base de datos. Una transacción en el sistema informático de la empresa inmobiliaria sería dar de alta a un empleado o eliminar un inmueble. Una transacción un poco más complicada sería eliminar un empleado y reasignar sus inmuebles a otro empleado. En este caso hay que realizar varios cambios sobre la base de datos. Si la transacción falla durante su realización, por ejemplo porque falla el hardware, la base de datos quedará en un estado inconsistente. Algunos de los cambios se habrán hecho y otros no, por lo tanto, los cambios realizados deberán ser deshechos para devolver la base de datos a un estado consistente.

4. Un SGBD debe proporcionar un mecanismo que asegure que la base de datos se actualice correctamente cuando varios usuarios la están actualizando concurrentemente. Uno de los principales objetivos de los SGBD es el permitir que varios usuarios tengan acceso concurrente a los datos que comparten. El acceso concurrente es relativamente fácil de gestionar si todos los usuarios se dedican a leer datos, ya que no pueden

interferir unos con otros. Sin embargo, cuando dos o más usuarios están accediendo a la base de datos y al menos uno de ellos está actualizando datos, pueden interferir de modo que se produzcan inconsistencias en la base de datos. El SGBD se debe encargar de que estas interferencias no se produzcan en el acceso simultáneo.

5. Un SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos en caso de que ocurra algún suceso que la dañe. Como se ha comentado antes, cuando el sistema falla en medio de una transacción, la base de datos se debe devolver a un estado consistente. Este fallo puede ser a causa de un fallo en algún dispositivo hardware o un error del software, que hagan que el SGBD aborte, o puede ser a causa de que el usuario detecte un error durante la transacción y la aborte antes de que finalice. En todos estos casos, el SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos llevándola a un estado consistente.

6. Un SGBD debe proporcionar un mecanismo que garantice que sólo los usuarios autorizados pueden acceder a la base de datos. La protección debe ser contra accesos no autorizados, tanto intencionados como accidentales.

7. Un SGBD debe ser capaz de integrarse con algún software de comunicación. Muchos usuarios acceden a la base de datos desde terminales. En ocasiones estos terminales se encuentran conectados directamente a la máquina sobre la que funciona el SGBD. En otras ocasiones los terminales están en lugares remotos, por lo que la comunicación con la máquina que alberga al SGBD se debe hacer a través de una red. En cualquiera de los dos casos, el SGBD recibe peticiones en forma de mensajes y

responde de modo similar. Todas estas transmisiones de mensajes las maneja el gestor de comunicaciones de datos. Aunque este gestor no forma parte del SGBD, es necesario que el SGBD se pueda integrar con él para que el sistema sea comercialmente viable.

8. Un SGBD debe proporcionar los medios necesarios para garantizar que tanto los datos de la base de datos, como los cambios que se realizan sobre estos datos, sigan ciertas reglas. La integridad de la base de datos requiere la validez y consistencia de los datos almacenados. Se puede considerar como otro modo de proteger la base de datos, pero además de tener que ver con la seguridad, tiene otras implicaciones. La integridad se ocupa de la calidad de los datos. Normalmente se expresa mediante restricciones, que son una serie de reglas que la base de datos no puede violar. Por ejemplo, se puede establecer la restricción de que cada empleado no puede tener asignados más de diez inmuebles. En este caso sería deseable que el SGBD controlara que no se sobrepase este límite cada vez que se asigne un inmueble a un empleado.

Además, de estos ocho servicios, es razonable esperar que los SGBD proporcionen un par de servicios más:

- Un SGBD debe permitir que se mantenga la independencia entre los programas y la estructura de la base de datos. La independencia de datos se alcanza mediante las vistas o subesquemas. La independencia de datos física es más fácil de alcanzar, de hecho hay varios tipos de cambios que se pueden realizar sobre la estructura física de la base de datos sin afectar a las vistas. Sin embargo, lograr una completa independencia de datos lógica es más difícil. Añadir una nueva entidad, un atributo o una relación puede ser sencillo, pero no es tan sencillo eliminarlos.

- Un SGBD debe proporcionar una serie de herramientas que permitan administrar la base de datos de modo efectivo. Algunas herramientas trabajan a nivel externo, por lo que habrán sido producidas por el administrador de la base de datos. Las herramientas que trabajan a nivel interno deben ser proporcionadas por el distribuidor del SGBD. Algunas de ellas son:
  - Herramientas para importar y exportar datos.
  - Herramientas para monitorizar el uso y el funcionamiento de la base de datos.
  - Programas de análisis estadístico para examinar las prestaciones o las estadísticas de utilización.
  - Herramientas para reorganización de índices.
  - Herramientas para aprovechar el espacio dejado en el almacenamiento físico por los registros borrados y que consoliden el espacio liberado para reutilizarlo cuando sea necesario<sup>6</sup>.

### **2.1.5. Ventajas de los Sistemas de Bases de Datos**

Los sistemas de bases de datos presentan numerosas ventajas que se pueden dividir en dos grupos: las que se deben a la integración de datos y las que se deben a la interface común que proporciona el SGBD.

#### **2.1.5.1. Ventajas por la integración de datos**

**Control sobre la redundancia de datos.** Los sistemas de ficheros almacenan varias copias de los mismos datos en ficheros distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos. En los

---

<sup>6</sup> <http://www.iesromerovargas.net/OASIS2/SGBD/Documentos/T2.pdf>

sistemas de bases de datos todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos, o bien es necesaria para mejorar las prestaciones.

**Consistencia de datos.** Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce esta redundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantienen consistentes. Desgraciadamente, no todos los SGBD de hoy en día se encargan de mantener automáticamente la consistencia.

**Más información sobre la misma cantidad de datos.** Al estar todos los datos integrados, se puede extraer información adicional sobre los mismos.

**Compartición de datos.** En los sistemas de ficheros, los ficheros pertenecen a las personas o a los departamentos que los utilizan. Pero en los sistemas de bases de datos, la base de datos pertenece a la empresa y puede ser compartida por todos los usuarios que estén autorizados. Además, las nuevas aplicaciones que se vayan creando pueden utilizar los datos de la base de datos existente.

**Mantenimiento de estándares.** Gracias a la integración es más fácil respetar los estándares necesarios, tanto los establecidos a nivel de la empresa como los nacionales e internacionales. Estos estándares pueden establecerse sobre el formato de los datos para facilitar su intercambio, pueden ser estándares de documentación, procedimientos de actualización y también reglas de acceso.

### **2.1.5.2. Ventajas por la existencia del SGBD**

**Mejora en la integridad de datos.** La integridad de la base de datos se refiere a la validez y la consistencia de los datos almacenados. Normalmente, la integridad se expresa mediante restricciones o reglas que no se pueden violar. Estas restricciones se pueden aplicar tanto a los datos, como a sus relaciones, y es el SGBD quien se debe encargar de mantenerlas.

**Mejora en la seguridad.** La seguridad de la base de datos es la protección de la base de datos frente a usuarios no autorizados. Sin unas buenas medidas de seguridad, la integración de datos en los sistemas de bases de datos hace que éstos sean más vulnerables que en los sistemas de ficheros. Sin embargo, los SGBD permiten mantener la seguridad mediante el establecimiento de claves para identificar al personal autorizado a utilizar la base de datos. Las autorizaciones se pueden realizar a nivel de operaciones, de modo que un usuario puede estar autorizado a consultar ciertos datos pero no a actualizarlos, por ejemplo.

**Mejora en la accesibilidad a los datos.** Muchos SGBD proporcionan lenguajes de consultas o generadores de informes que permiten al usuario hacer cualquier tipo de consulta sobre los datos, sin que sea necesario que un programador escriba una aplicación que realice tal tarea.

**Mejora en la productividad.** El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación. El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel. Muchos SGBD también proporcionan un

entorno de cuarta generación consistente en un conjunto de herramientas que simplifican, en gran medida, el desarrollo de las aplicaciones que acceden a la base de datos. Gracias a estas herramientas, el programador puede ofrecer una mayor productividad en un tiempo menor.

**Mejora en el mantenimiento gracias a la independencia de datos.** En los sistemas de ficheros, las descripciones de los datos se encuentran inmersas en los programas de aplicación que los manejan. Esto hace que los programas sean dependientes de los datos, de modo que un cambio en su estructura, o un cambio en el modo en que se almacena en disco, requiere cambios importantes en los programas cuyos datos se ven afectados. Sin embargo, los SGBD separan las descripciones de los datos de las aplicaciones. Esto es lo que se conoce como independencia de datos, gracias a la cual se simplifica el mantenimiento de las aplicaciones que acceden a la base de datos.

**Aumento de la concurrencia.** En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que el acceso interfiera entre ellos de modo que se pierda información o, incluso, que se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

**Mejora en los servicios de copias de seguridad y de recuperación ante fallos.** Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos. En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a

realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.

### 2.1.6. Componentes de un Sistema de Gestión de Base de Datos

Un Sistema de Gestión de Base de Datos se divide en módulos que tratan cada una de las responsabilidades del sistema general. Los componentes funcionales de un SGBD (KORTH, 1995) incluyen:

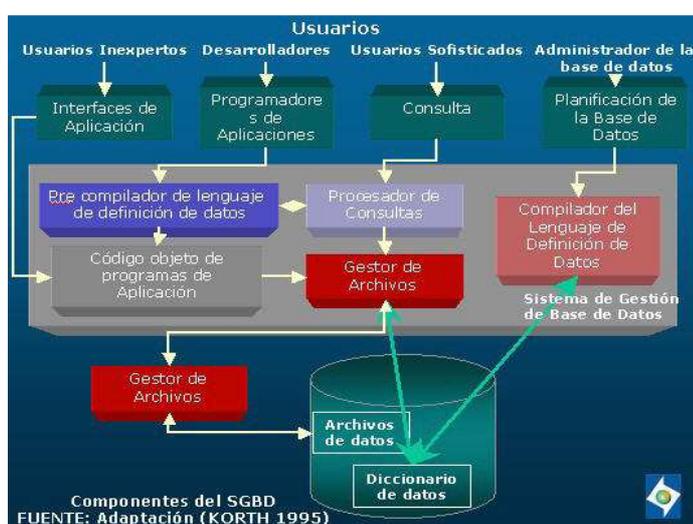


Figura II.2 Componentes de un SGBD

- **Procesador de Consultas.** Traduce sentencias en un lenguaje de consultas a instrucciones de bajo nivel que entiende el gestor de la base de datos.
- **Gestor de la Base de Datos.** Proporciona la interfase entre los datos de bajo nivel almacenados en la base de datos y los programas de aplicación y las consultas que se hacen en el sistema.
- **Gestor de Archivos.** Gestiona la asignación de espacio en la memoria del disco y de las estructuras de datos usadas para representar la información almacenada en disco.

- **Pre compilador del Lenguaje de Manipulación de Datos DML.** Convierte las sentencias en DML incorporadas en un programa de aplicación en llamadas normales a procedimientos en el lenguaje principal.
- **Compilador del Lenguaje de Definición de Datos DDL.** Convierte sentencias en DDL en un conjunto de tablas metadatos o “datos sobre datos”.
- **Gestor del Diccionario de Datos.** Almacena metadatos sobre la estructura de la base de datos.

### 2.1.7. Arquitectura de un Sistema de Base de Datos

Uno de los objetivos de un sistema de base de datos es proporcionar una visión lo más abstracta posible de la información, es decir, ocultar detalles referentes a la forma en que los datos están organizados y almacenados.

La arquitectura más estándar y, por tanto, la más utilizada, es la que hace una división en niveles de la base de datos. La arquitectura de tres niveles<sup>7</sup> propuesta por ANSI/SPARC (Comité de Standarización de Requerimientos y Planificación del Instituto Nacional de Standarización Americano) considera tres niveles según la perspectiva desde la que sea vista la información. El nivel interno es el nivel más bajo de abstracción, donde se describe la información en función del sistema en que se implantará la base de datos. Por encima, se encuentra el conceptual, que representa a alto nivel toda la información de la base de datos independientemente de la máquina en que vaya a utilizarse. El más abstracto de los niveles es el externo, que gestiona la

---

<sup>7</sup> La arquitectura de tres niveles, normalizado en 1975 y mejorado en 1978 también conocido como arquitectura ANSI/SPARC, el objetivo principal de esta arquitectura era el de separar los programas de aplicación de la BD física.

información desde el punto de vista individual de cada usuario, grupo de usuarios, programador o grupo de programadores.

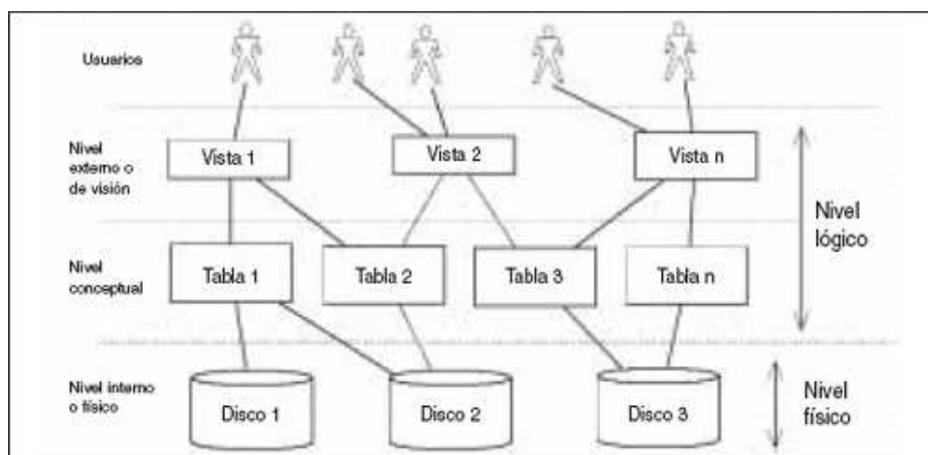
- **Nivel interno o físico:** El más cercano al almacenamiento físico, es decir, tal y como están almacenados en el ordenador. Describe la estructura física de la BD mediante un esquema interno. Este esquema se especifica con un modelo físico y describe los detalles de cómo se almacenan físicamente los datos: los archivos que contienen la información, su organización, los métodos de acceso a los registros, los tipos de registros, la longitud, los campos que los componen, los campos clave, etcétera.

- **Nivel conceptual:** Describe la estructura de toda la BD para un grupo de usuarios mediante un esquema conceptual. Este esquema describe las entidades, atributos, relaciones, operaciones de los usuarios y restricciones, ocultando los detalles de las estructuras físicas de almacenamiento. Representa la información contenida en la BD.

- **Nivel externo:** Es el conjunto de percepciones individuales de la base de datos. Cada visión individual se denomina subesquema o vista. Un subesquema podrá ser compartido por varios usuarios, y cada usuario tendrá la posibilidad de acceder a distintos subesquemas. Al crear un subesquema, es posible mezclar campos de distintos registros, omitir campos, cambiar el orden de los campos, añadir campos que puedan ser calculados a partir de los descritos en el esquema conceptual, etc.

El nivel externo o de visión es el más cercano a los usuarios, es decir, es donde se describen varios esquemas externos o vistas de usuarios. Cada esquema describe la parte de la BD que interesa a un grupo de usuarios en este nivel se representa la visión

individual de un usuario o de un grupo de usuarios. En la Figura II.3 se representan los niveles de abstracción de la arquitectura ANSI.



**Figura II.3** Niveles de abstracción de la arquitectura ANSI

### 2.1.8. SGBD libres

En la actualidad el mundo del software libre ofrece gran variedad de soluciones en el ámbito de las bases de datos. Entre las bases de datos de código libre se pueden mencionar a MySQL que es por definición la base de datos de código libre más popular del mundo después de todo. También está PostgreSQL (la base de datos de código libre más avanzada del mundo), así como Firebird (la base de datos relacional del nuevo milenio). Sin embargo, todavía hay más. BerkeleyDB, si bien no es directamente comparable, merece una mención, mientras que Computer Associates ha lanzado recientemente Ingres, e IBM ha hecho lo mismo con Cloudscape (bajo el nombre de Derby). A continuación se detalla algunos de estos productos:

#### MySQL

MySQL es conocida por ser la base de datos más popular, impulsada por el boom de Internet, es utilizada ampliamente para aplicaciones web. En principio rápida y fácil de

utilizar, pero con funcionalidad limitada, MySQL ha trabajado en dotar de nuevas funcionalidades al producto, y aunque todavía carece de presencia en el ámbito empresarial más exigente, su popularidad, y su gran base de usuarios significan que hay muchísimas herramientas y aplicaciones que trabajan con MySQL, con toneladas de soporte extra oficial que complementa el soporte oficial de MySQL.

### **Firebird**

Inprise (ahora Borland) fue una de las primeras compañías comerciales en lanzar una base de datos bajo los términos del software libre cuando liberaron a principio del 2000 su producto Interbase bajo una variante de la licencia Mozilla. Borland ha lanzado desde entonces varias versiones de su producto comercial, pero Firebird ha continuado desarrollándose en paralelo basándose en la versión del código fuente de Interbase 6.0.

Firebird tiene una comunidad de usuarios pequeña pero muy activa, y hasta hace poco era el factor olvidado en los debates MySQL vs PostgreSQL. Firebird ha tenido siempre muchas más características que MySQL, y también, al contrario que PostgreSQL, siempre ha funcionado sin problemas bajo plataformas Windows, Linux y otras variantes de Unix. Firebird por otra parte también ha sido un poco maltratada por Borland desde su lanzamiento, en el sentido que no lanzaron todos sus componentes, y los scripts de compilación ¡ni siquiera funcionaban!

Firebird tiene una base de usuarios leales, es un buen producto para trabajar, y si consigue aumentar su velocidad de desarrollo, así como mejorar su marketing, será un competidor muy fuerte.

## **PostgreSQL**

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos de software libre, publicado bajo la licencia BSD. PostgreSQL es basado en el proyecto POSTGRES, de la universidad de Berkeley. El director de este proyecto es el profesor Michael Stonebraker.

Fue el pionero en muchos de los conceptos existentes en el sistema objeto-relacional, incluido, más tarde en otros sistemas de gestión comerciales. PostgreSQL es un sistema objeto-relacional, ya que incluye características de la orientación a objetos, como puede ser la herencia, tipos de datos, funciones, restricciones, disparadores, reglas e integridad transaccional.

## **Ingres**

Comercializada en 1980, la base de datos Ingres es un SGBD potente (llamado de clase empresa). Ingres se volvió libre en 2004, su licencia es la GNU GPL. Ingres es conocida por su fiabilidad y su escalabilidad, posee gran cantidad de características, sin embargo la documentación disponible en línea es bastante pobre en comparación con las otras.

## **Berkeley DB**

Berkeley DB es de código abierto y funciona en todos los principales sistemas operativos, incluyendo Linux embebido, Linux, Unix, Microsoft Windows, Mac OS X, QNX y VxWorks. Berkeley DB proporciona los datos básicos de gestión de la funcionalidad, escalabilidad, potencia y flexibilidad de las empresas las bases de datos relacionales, pero sin la sobrecarga de una capa de procesamiento de consultas.

Berkeley DB es un motor relacional de almacenamiento empujado muy ligero, pensado para aplicaciones que no requieren una instalación de un DBMS completo. Los usuarios de MySQL tienen la opción de utilizar el motor de datos de Berkeley DB (al principio era la única manera de garantizar capacidades transaccionales en MySQL).

### **Derby**

Apache Derby es un sistema gestor de base de datos relacional escrito en Java que puede ser embebido en aplicaciones Java y utilizado para procesos de transacciones online. Inicialmente distribuido como IBM Cloudscape, Apache Derby es un proyecto open source licenciado bajo la Apache 2.0 License.

Lo más importante de la situación actual es que las bases de datos libres han alcanzado un nivel de madurez que garantiza su estabilidad y su continuidad a largo plazo, convirtiéndolas en alternativas perfectamente viables a las bases de datos privadas, al menos, en la mayoría de los usos tradicionales que se dan a una base de datos.

El objetivo de este estudio es, precisamente, mostrar la situación actual en el mercado de estas aplicaciones, valorarlas, y señalar tendencias de futuro; específicamente se enfocará en MySQL y Firebird, MySQL el motor de base de datos más utilizado en el mundo con grandes características y Firebird un motor de base de datos que al igual que MySQL posee muchas características que un buen SGBD debe presentar, pero que no es muy conocido al menos en nuestro medio.

## **2.2. BASES DE DATOS EN FIREBIRD**

### **2.2.1. Introducción**

Firebird es un sistema de administración de base de datos relacional (RDBMS) (Lenguaje consultas: SQL) de código abierto. Es multiplataforma y funciona en una variedad de plataformas. El proyecto se desarrolla activamente y sus nuevas versiones van apareciendo para el mundo del software libre. Firebird ofrece alto rendimiento y un poderoso lenguaje de procedimientos almacenados y disparadores.

### **2.2.2. Historia**

Firebird deriva del código fuente de Interbase 6.0 de Borland. Es open source y no hay licencias duales. Tanto para uso comercial como para aplicaciones open source, es totalmente libre. La tecnología de Firebird lleva 20 años funcionando, esto hace que sea un producto muy maduro y estable. Interbase 6.0 fue liberado el 25 de julio del 2000 por Inprise Corp (ahora conocida como Borland Software Corp) bajo la licencia IPL (Licencia Pública InterBase) derivada de MPL (Mozilla Public Licencia) es más permisiva que GPL y similar a BSD.

A principios del año 2000, la compañía Borland anunció que el código de Interbase sería liberado en la versión 6.0 y comenzó las negociaciones para que una empresa separada se encargara del nuevo producto. Cuando los responsables de esta nueva empresa y Borland no llegaron a un acuerdo de separación, Interbase permaneció como un producto de Borland y el código fuente de Interbase 6 se liberó bajo una variante de la “Mozilla Public License” a mediados del 2000.

Con la división de Interbase en Borland, la compañía liberó una versión propietaria de Interbase 6 y luego 6.5. Borland liberó varias actualizaciones para la versión libre antes de anunciar que ya no participaría activamente en el desarrollo de este proyecto. De aquí nació una nueva rama de desarrollo libre basada en el código abierto de Interbase 6 que daría vida a Firebird.

El motor de bases de datos Firebird ha sido desarrollado por un equipo independiente de desarrolladores voluntarios a partir del código fuente de Interbase 6.0. El desarrollo de Firebird lleva aparejado la aparición de versiones que incluyen nuevas características y posibilidades. Así se comenzó con la versión 1.0 (simplemente portar el código de Interbase 6.0 en C), la versión 1.5 (conversión de Firebird a C++).

La versión 1.5 ha sido construida a partir del código portado del original en C a C++, un proceso iniciado por Mike Nordell en el año 2000. La limpieza total del código y la corrección de errores ha continuado, completada por un nuevo gestor de memoria y mejoras en el lenguaje. No menos importantes han sido los cambios experimentados por el optimizador de consultas SQL durante el proceso de desarrollo de la v. 1.5, con mejoras y correcciones de la mano de Arno Brinkman y otros, cuyo resultado ha sido una mejora en la velocidad de entre un 30 y un 60 por ciento como mínimo.

El desarrollo del código Firebird 2 arranca inicialmente en el desarrollo de Firebird 1, con el traspaso del código C de Firebird 1 a C++ y la primera gran limpieza en el código. Firebird 1.5 es la primera versión del código Firebird 2. Ello supone haber cubierto una etapa muy importante para los desarrolladores y el propio proyecto

Firebird, pero no es un fin en sí mismo. Cubierta la etapa de la liberación de Firebird 1.5, el viaje hacia Firebird 2 prosigue con importantes modificaciones.

Firebird 2.0 se liberó oficialmente durante la Cuarta Conferencia Internacional de Firebird en Praga, República Checa. Firebird 2.0 trae una gran colección de muy esperadas mejoras que aumentan significativamente la performance, seguridad y el soporte de idiomas internacionales, y además brinda algunas deseables nuevas características en el lenguaje SQL. Además, Firebird 2.0 también ofrece un código de plataforma mucho más robusto para la rearquitectura planeada para Firebird 3.0. Entre las novedades de Firebird 2.0 se incluyen:

- Nuevo backup incremental.
- El tamaño de las tablas ya no es limitado a 30 Gb.
- Soporte para arquitecturas de 64 bits (Intel EM64T y AMD64).
- Interface tipo plugin para juegos de caracteres internacionales.
- Soporte de tablas derivadas, como se define en SQL200x, incluyendo anidado multi-nivel y joining de "subqueries".

El 18 de abril de 2008 fue liberada la versión 2.1 de Firebird que incluye nuevas características, entre las más importantes se tienen:

- Triggers (Disparadores) de base de datos
- Tablas temporales globales
- Consultas recursivas en DSQL
- Cláusula RETURNING en instrucciones INSERT, UPDATE y DELETE
- Función LIST

- Muchas nuevas funciones básicas, como RAND, REPLACE, FLOOR, ABS, etc.
- Uso de dominios en PSQL
- COLLATE en procedimientos almacenados y en parámetros

La versión 2.5.0 RC1 fue liberada el 26 de diciembre de 2009. Y el 4 de octubre de 2010 fue el lanzamiento de la versión final de Firebird 2.5, entre algunas de sus nuevas características se encuentran:

- Nueva arquitectura SuperClassic. Permite obtener un mejor aprovechamiento de los entornos con hardware multi-core y multi-CPU, optimizando el uso de recursos cuando hay bases de datos con un gran número de usuarios y bases de datos enormes.
- Auditoría. El rastreo de auditoría del sistema y el seguimiento de sesiones de usuario, vía API de Servicios, le permitirá realizar monitoreo y análisis, casi en tiempo real de todo lo que sucede en una bases de datos.
- Consultas entre bases de datos. Consultar una base de datos Firebird desde otra e intercambien información entre sí.
- Gestión de usuarios mejorada. La administración de usuarios (seguridad) se hace más accesible y flexible, por medio de peticiones SQL enviadas desde la base de datos donde están trabajando los usuarios<sup>8</sup>.

Y así hasta llegar a la última prevista, versión 3.0. Firebird seguirá avanzando en sus nuevas mejoras y características para versiones posteriores que estarán disponibles para el público del software libre.

---

<sup>8</sup> <http://firebirdmanual.com/firebird/es/firebird-manual/2/novedades-firebird-2-5/45>

### 2.2.3. Características

Firebird tiene todas las características y la potencia de un RDBMS. Se pueden manejar bases de datos desde unos pocos KB hasta varios Gigabytes con buen rendimiento y casi sin mantenimiento.

Sus características principales son:

- Es multiplataforma, y actualmente puede ejecutarse en los sistemas operativos: Microsoft Windows, Linux, HP-UX, FreeBSD, Mac OS, Solaris, etc.
- Ejecutable pequeño, con requerimientos de hardware bajos.
- Arquitectura Cliente/Servidor sobre protocolo TCP/IP y otros (embedded).
- Buena seguridad basada en usuarios/roles.
- Completo soporte para Procedimientos Almacenados y Triggers (Disparadores)
- Las transacciones son totalmente ACID (Atomicity, Consistency, Isolation and Durability)
- Integridad Referencial
- Arquitectura Multi Generacional. Una de las características claves de Firebird es su arquitectura multigeneracional, que permite el desarrollo y soporte de aplicaciones híbridas OLTP y OLAP. Esto hace a Firebird capaz de servir simultáneamente como un almacén de datos analítico y operacional, porque las lecturas no bloquean a las escrituras cuando acceden a los mismos datos en la mayoría de las situaciones.
- Bajo consumo de recursos
- Completo lenguaje para la escritura de Procedimientos Almacenados y Triggers (PSQL)
- Capacidad de almacenar elementos BLOB (Binary Large Objects).
- Soporte para Funciones Externas (UDFs User-Defined Functions)

- Poca o ninguna necesidad de DBAs especializados.
- Prácticamente no necesita configuración. Solo instalar y comenzar a usarla!
- Una gran comunidad y muchos sitios donde puedes encontrar excelente soporte gratuito.
- Diferentes arquitecturas, entre ellas el Firebird incrustado (embedded server) que permite ejecutar aplicaciones monousuario en ordenadores sin instalar el software Firebird. Ideal para crear catálogos en CDROM, versiones mono usuario, de evaluación o portátiles de las aplicaciones.
- Docenas de herramientas de terceros. Como herramientas de administración gráficas, herramientas de replicación, etc.
- Escritura segura. Recuperación rápida, ¡sin necesidad de logs de transacciones!
- Muchas formas de acceder a las base de datos: nativo/API, drivers dbExpress, ODBC, OLEDB, .Net provider, driver JDBC nativo tipo 4, módulo para Python, PHP, Perl, etc.
- Copias de seguridad incrementales
- Disponible para arquitecturas de 64 bits
- Implementación completa de cursores en PSQL
- Tablas de Monitoreo. Monitoreo de base de datos en tiempo de ejecución accediendo a tablas de sistema virtuales MON\$ que permiten el monitoreo de la base de datos vía consultas SQL. Por ejemplo, la tabla MON\$DATABASE provee información como la estructura en disco, dialecto, OIT y OAT (Oldest Interesting Transaction, Oldest Active Transaction), etc. Se puede utilizar la información obtenida de estas tablas para cancelar una consulta.
- Disparadores a nivel de Conexión y Transacción

- Tablas Temporales. Algunas veces se requieren guardar datos de forma temporal en alguna tabla, bien sea para que estos sean eliminados una vez termine la transacción o para cuando termine la conexión. La implementación cumple con el estándar SQL. Las tablas se crean bajo demanda para uso por conexión o por transacción. Las tablas temporales globales en versiones anteriores a Firebird 2.1 se deben programar de alguna manera manualmente, con Firebird 2.1 se pueden crear las llamadas tablas temporales, ya sean de tipo *ON COMMIT PRESERVE ROWS* (que mantienen los datos hasta que termine la conexión) o *ON COMMIT DELETE ROWS* (que mantiene los datos hasta que termine la transacción). Crear este tipo de tablas es muy sencillo, es igual que crear cualquier otra tabla común y corriente (persistente).
- Pleno soporte del estándar SQL-92, tanto de sintaxis como de tipos de datos.
- Conectividad. Puede acceder a Firebird SQL a través de cualquier lenguaje de programación<sup>9</sup>.

El servidor Firebird viene en tres versiones: SuperServer, Classic y Embedded. Actualmente, Classic es la versión recomendada para máquinas con SMP y algunas otras situaciones específicas.

SuperServer comparte su caché para todas las conexiones y usa un hilo de ejecución para cada conexión. Ésta se suele usar en Windows. Classic inicia un proceso de servidor independiente para cada conexión que se haga. La versión embedded es un servidor Firebird con todas sus características, empaquetado en unos pocos ficheros.

---

<sup>9</sup> [http://www.firebird.com.mx/descargas/documentos/tema\\_2-caracteristicas\\_basicas.pdf](http://www.firebird.com.mx/descargas/documentos/tema_2-caracteristicas_basicas.pdf)  
<http://es.wikipedia.org/wiki/Firebird>

Firebird viene con un completo paquete de utilidades de línea de comandos que permiten crear bases de datos, generar estadísticas, ejecutar comandos y scripts SQL, hacer y recuperar copias de seguridad, etc. Si se prefiere usar herramientas visuales, hay varias opciones donde elegir, incluyendo gratuitas.

En Windows, se puede ejecutar Firebird como servicio o como aplicación. El instalador puede crear un icono en el panel de control que permite controlar el servidor (iniciarlo, pararlo, etc).

Firebird tiene establecido un sistema de seguridad basado en usuarios y contraseñas. Por defecto, cuando se hace una instalación, se tiene el usuario SYSDBA con contraseña 'masterkey'. Esta información de usuarios se almacena en un fichero de base de datos Firebird: security2.fdb (para versiones de Firebird 2 o superiores).

Firebird está orientado a todo tipo de empresas pequeñas, medianas y grandes, y en la actualidad es usado en diferentes aplicaciones y sistemas como puntos de venta, hospitales, monitoreo, portales, call centers, industria automovilística, entre muchos. Un buen ejemplo de esto en Softool'06, donde Avarada (un ERP Ruso) se ejecuta con Firebird 2.0 Classic server y un promedio de 100 conexiones simultáneas contra una base de datos de 120Gb con 700 millones de registros. El servidor es una máquina SMP (2 CPUs - Dell PowerEdge 2950) con 6GB de RAM.

La alta compatibilidad con los estándares de la industria en muchos frentes hace a Firebird la opción obvia para desarrollar aplicaciones interoperables para ambientes

homogéneos e híbridos. La mezcla de características: alto rendimiento, tamaño de distribución pequeño, escalabilidad, instalación sencilla hacen a Firebird una opción muy atractiva para todos los tipos de desarrolladores de software y distribuidores. Es utilizada por aproximadamente 1 millón de desarrolladores de software en todo el mundo.

#### **2.2.4. Tipos de servidores Firebird**

Firebird se distribuye en los tipos de servidores: ClassicServer, SuperServer y Embedded<sup>10</sup>.

##### **2.2.4.1. SUPERSERVER**

Se crea un hilo por cada usuario que se conecta al servidor, se reserva un número de MB por cada base de datos a la que se accede en cada momento, por tanto no se necesita más RAM en el servidor conforme se da servicio a más usuarios, pero a cambio de eso, un único proceso (fbserver) es el encargado de dar ese servicio a todos los hilos que atienden a los usuarios, esto hace que en servidores que cuentan con más de un microprocesador solo se utilice uno.

Resumiendo, esto quiere decir que se puede tener muchos usuarios pero la velocidad de proceso se la dividen entre ellos usando solo un procesador del servidor.

La versión 3 de FireBird, eliminará esta limitación en el número de procesadores.

---

<sup>10</sup> <http://oscarzeladapd.blogspot.com/2008/01/que-es-firebird.html>

#### **2.2.4.2. CLASSIC SERVER**

Crea un proceso y reserva un número de MB por cada conexión-usuario que se conecta, y de este modo se aprovechan todos los procesadores que se tienen, pero la memoria RAM necesaria aumenta conforme se añaden más usuarios al sistema, esto debido a que el servidor Classic lanza un proceso por cada conexión. Solo se recomienda instalar de este modo si: Tiene un servidor multiprocesador y suficiente memoria.

La principal diferencia entre el SuperServer o el Classic Server consiste en que el SuperServer maneja hilos de ejecución individuales para cada conexión. Por lo tanto para un número reducido de conexiones el recomendado sería el Classic porque consumirá menor cantidad de recursos. El servidor Classic es la versión recomendada para máquinas con SMP porque el SuperServer no tiene soporte para este tipo de arquitectura.

Los propios desarrolladores de Firebird recomiendan lo siguiente a la hora de decidirse por uno de estos servidores:

- En plataformas Windows seleccionar el SuperServer.
- En la plataforma Linux simplemente elegir cualquiera, según las conexiones estimadas. En la mayoría de las situaciones no se notará diferencias en la ejecución.

#### **2.2.4.3. EMBEDDED SERVER (SERVIDOR EMBEBIDO)**

La versión Embedded es una interesante variación del servidor. Es un servidor Firebird con todas sus características, empaquetado en unos pocos ficheros. Firebird embebido permite utilizar un archivo de base de datos local (de Firebird) sin necesidad de instalar

el servidor de base de datos. Esto permite poder distribuir aplicaciones sin necesidad de instalar componente extras, solo se debe copiar el archivo con la base de datos y el driver para la conexión embebida (generalmente llamado fbclient.dll o fbembed.dll) y listo, ya se tiene una aplicación con base de datos corriendo.

Es muy fácil de usar, el servidor no necesita instalación. Es ideal para CDROM de catálogos, demos o aplicaciones de escritorio monousuario.

Las desventajas que tiene esta forma de operar con respecto al servidor son:

- No permite conexiones remotas.
- Solo un usuario/aplicación pueden estar accediendo a la base de datos (porque la misma queda bloqueada cuando es abierta).
- No realiza la autenticación de usuarios (es decir, cualquier aplicación puede acceder a la base de datos sin necesidad de poseer usuario y passwords válidos).

#### **2.2.4.4. ARQUITECTURA SUPERCLASSIC**

Firebird 2.5 introduce una nueva arquitectura, llamada “SuperClassic”, para obtener un mejor aprovechamiento de los entornos con hardware multi-core y multi-CPU, optimizando el uso de recursos cuando hay bases de datos con un gran número de usuarios y bases de datos enormes.

Firebird 3.0 será completamente en ambiente SMP. Superclásico es el primer paso en esa dirección.

### 2.2.5. Tipos de datos

Las columnas de una tabla almacenan datos de un solo tipo, que debe ser especificado en la declaración de la tabla. Los tipos de datos soportados por el servidor Firebird son los siguientes<sup>11</sup>:

a. Tipos de datos numéricos:

BIGINT, INTEGER y SMALLINT

FLOAT y DOUBLE PRECISION

NUMERIC y DECIMAL

b. Tipos fecha y hora

DATE, TIME y TIMESTAMP

c. Tipos carácter

CHAR, VARCHAR

d. Tipos especiales

BLOB

El tipo BLOB (Binary Large Object) permite almacenar datos que no pueden ser almacenados fácilmente en los tipos SQL estandar. Suele usarse para objetos de un tamaño variable e indeterminado, como imágenes, videos, capítulos de un libro, etc.

#### **Tipos de datos numéricos.**

Firebird soporta tipos numéricos decimales y de punto flotante. Entre los decimales se encuentran con los que tienen parte fraccionaria 0 (SMALLINT, INTEGER y BIGINT) y los que tienen parte fraccionaria de tamaño fijo (NUMERIC y DECIMAL).

---

<sup>11</sup> [http://www.firebird.com.mx/descargas/documentos/tema\\_2-caracteristicas\\_basicas.pdf](http://www.firebird.com.mx/descargas/documentos/tema_2-caracteristicas_basicas.pdf)

Los tipos de punto flotante soportados son FLOAT y DOUBLE PRECISION. Firebird no soporta tipos de datos numéricos sin signo.

**Tabla III.** Tipos de datos numéricos en Firebird

TIPO	TAMAÑO	RANGO/PRECISIÓN	DESCRIPCIÓN
SMALLINT	16 bits	-32,768 a 32,767	Entero corto con signo
INTEGER	32 bits	-2,147,483,648 a 2,147,483,647	Entero con signo
BIGINT	64 bits	$-2^{63}$ a $2^{63}-1$	Entero largo con signo
NUMERIC (precisión, escala)	Variable (16, 32 o 64 bits)	Precisión: 1 a 18, especifica el número exacto de dígitos de precisión. Escala: 0 a 18, especifica el número de lugares decimales.	Por ejemplo NUMERIC(10,4) sería un número con formato P P P P P P . E E E E (123456.7890)
DECIMAL (precisión, escala)	Variable (16, 32 o 64 bits)	Precisión: 1 a 18, especifica el número mínimo de dígitos de precisión. Escala: 0 a 18, especifica el número de lugares decimales.	Por ejemplo DECIMAL(10,4) sería un número con formato P P P P P P . E E E E (123456.7890)
FLOAT	32 bits	$3.4 \times 10^{-38}$ a $3.4 \times 10^{38}$	IEEE simple precisión: Número real 7 dígitos
DOUBLE PRECISION	64 bits	$1.7 \times 10^{-308}$ a $1.7 \times 10^{308}$	IEEE doble precisión: Número real 15 dígitos

Como se ha comentado anteriormente, se puede trabajar con datos decimales con la parte fraccionaria de tamaño fijo. Para ello se utilizan los tipos NUMERIC y DECIMAL. Estos tipos se utilizan cuando se desea trabajar con decimales con una precisión fija. Por ejemplo, si se quiere manejar el precio de los artículos en un almacén, siempre se usa un número con 4 decimales. Si se lo implementa con un número real en punto flotante se tendrá problemas a la hora del redondeo con operaciones como multiplicaciones o divisiones. Si se usa en su lugar, por ejemplo, un NUMERIC, se acabarán estos problemas.

En SQL estándar se definen estos dos tipos de datos: NUMERIC y DECIMAL. Conceptualmente son idénticos con la única diferencia que dada una precisión, en NUMERIC sólo se admiten números hasta esa precisión, mientras que en DECIMAL se admiten números cuya precisión es de al menos la indicada.

Para comprobar lo anterior se desea almacenar varios números:

- NUMERIC (4,2): se podrá almacenar el 7.2345 (se guarda como 7.23), el 23.45 pero no podría almacenarse el 753.22 (al necesitar una precisión de 5).
- DECIMAL(4,2): se podrían almacenar todos los números indicados para NUMERIC además del 753.22, ya que se ha indicado una precisión de al menos 4 cifras.

Internamente estos tipos de datos se almacenan como un entero del tamaño necesario para la precisión indicada.

### **Tipos fecha y hora.**

**Tabla II.II.** Tipos de datos fecha y hora en Firebird

<b>NOMBRE</b>	<b>TAMAÑO</b>	<b>RANGO/PRECISIÓN</b>	<b>DESCRIPCIÓN</b>
DATE	32 bits con signo	01-01-100 a 31-12-9999	Fecha Ejm: 12/10/1977
TIME	32 bits sin signo	0:00:00 a 23:59:59.9999	Hora
TIMESTAMP	64 bits (2x32 bits)	1 jan 100 CE to 28 feb 32768 CE	Incluye la hora y la fecha en dos estructuras de 32 bits

Al estar todos los tipos fecha almacenados como números, es posible, realizar operaciones aritméticas. Así si a un tipo DATE le suma o resta un entero, se obtendrá una nueva fecha con el incremento o decremento en días correspondiente. Si se resta dos fechas, se obtendrá el número de días entre ellas. Si se resta dos horas, se obtendrá el número de segundos entre ellas.

### **Tipos cadena de caracteres.**

Firebird permite trabajar con cadenas de longitud fija y de longitud variable. En las cadenas de caracteres de tamaño fijo se almacenan exactamente todos los caracteres. En una cadena de longitud variable se almacenan los caracteres que realmente se usan más un entero con la longitud de la cadena usada. Así, si se tiene la cadena 'hola', y se quiere como una cadena de longitud 10, si uso cadenas de longitud fija se almacenaría físicamente como 'hola ' mientras que con cadenas de longitud variable sería 'hola' junto con longitud 4.

Firebird define los siguientes tipos para cadenas de caracteres:

**Tabla II.III.** Tipos de datos cadena de caracteres en Firebird

<b>NOMBRE</b>	<b>TAMAÑO</b>	<b>RANGO/PRECISIÓN</b>	<b>DESCRIPCIÓN</b>
CHAR(n)	N caracteres	0 to 32.767 bytes	Cadena de n caracteres de longitud fija.
VARCHAR(n)	N caracteres	0 to 32.767 bytes	Cadena de n caracteres de longitud variable.

Las cadenas no pueden ocupar más de 32767 bytes. Esta limitación hay que tenerla muy en cuenta ya que, según el conjunto de caracteres, se utilizan uno, dos o tres bytes para almacenar un carácter. Así se podrá almacenar como máximo 32767, 16383 o 8191 caracteres dependiendo del conjunto de caracteres utilizado.

### **Tipos especiales.**

Aquí se encuentra el tipo BLOB. Un BLOB (Binary Large Object) es una estructura compleja para almacenar datos de objetos de tamaño variable, desde unos pocos bytes a gigabytes. Firebird los almacena como punteros. En el registro aparece la referencia a otro lugar donde se almacena realmente el contenido.

En un BLOB se puede almacenar cualquier tipo de información: una imagen de mapa de bits, una imagen vectorial, un video, un capítulo de un libro. Esto hace que el cliente sea el encargado de procesar la información almacenada y que un campo BLOB no pueda ser indexado.

**Tabla II.IV.** Tipos de datos especiales en Firebird

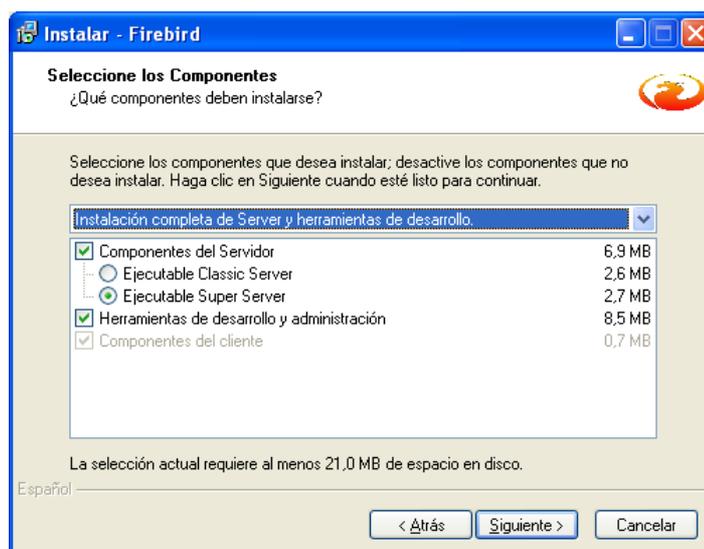
NOMBRE	TAMAÑO	DESCRIPCIÓN
BLOB	<32 GB	Almacenamiento para variables indeterminadas. Textos, Imágenes, Archivos Binarios, etc.

### 2.2.6. Instalación y puesta en marcha

La instalación y puesta en marcha de Firebird es muy simple. Se incluye un programa instalador al estilo *experto* que guiará paso a paso en la instalación; sólo seleccionar qué partes instalar y adónde.

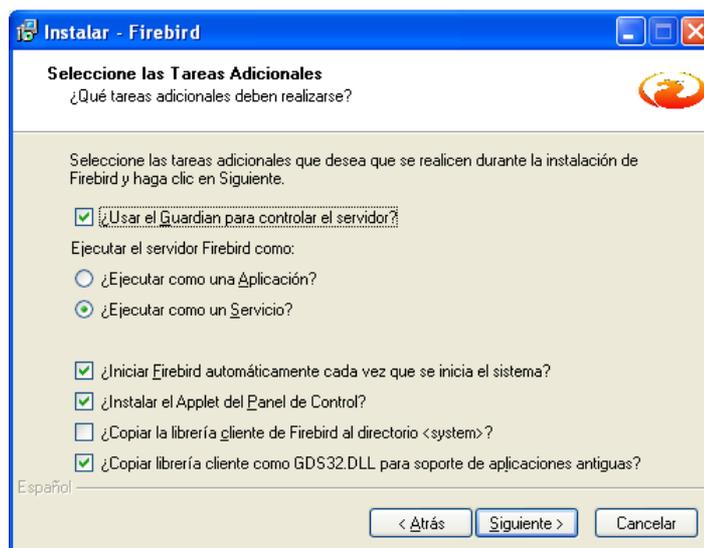
#### 2.2.6.1 Instalando el motor

La instalación es simple. Se va a ver algunas de las pantallas de instalación:



**Figura II.4** Seleccionando tipo de servidor

La Figura II.4 muestra los tipos de servidores posibles a escoger en la instalación, o el SuperServer o el ClassicServer. La diferencia es que SuperServer, crea un único proceso para todas las conexiones mientras que ClassicServer, crea tantos procesos como conexiones haya. Escoger según sea las necesidades. Se escogerá el SuperServer.



**Figura II.5** Selección de tareas adicionales

En la Figura II.5 se seleccionará la forma de ejecución del motor. Se puede escoger entre “*ejecutarlo como una aplicación*” o “*ejecutarlo como un servicio*”. Esto depende del sistema operativo. Si es Windows 9x o ME (no soportan servicios), escoger “*Run as an Application*”, pero si es Windows NT, 2000, 2003 o XP, escoger “*Run as a Service*”. Las demás opciones dejar como muestra la imagen. Y con esto se termina la instalación.

### 2.2.6.2 Probar la instalación

Si todo funciona como fue diseñado, el proceso del servidor Firebird estará corriendo en su servidor después de completada la instalación. Arrancará automáticamente cada vez que reinicie su servidor.

- **Haciendo Ping al servidor**

Usualmente, lo primero que hay que hacer una vez que la instalación está completa es hacer ping al servidor. Esto le dará una comprobación de que su máquina cliente puede ver el equipo servidor de su red. Por ejemplo, si la dirección IP en el dominio en que es visible para su cliente es 192.13.14.1, abra una ventana del shell y escriba el comando.

**ping 192.13.14.1**

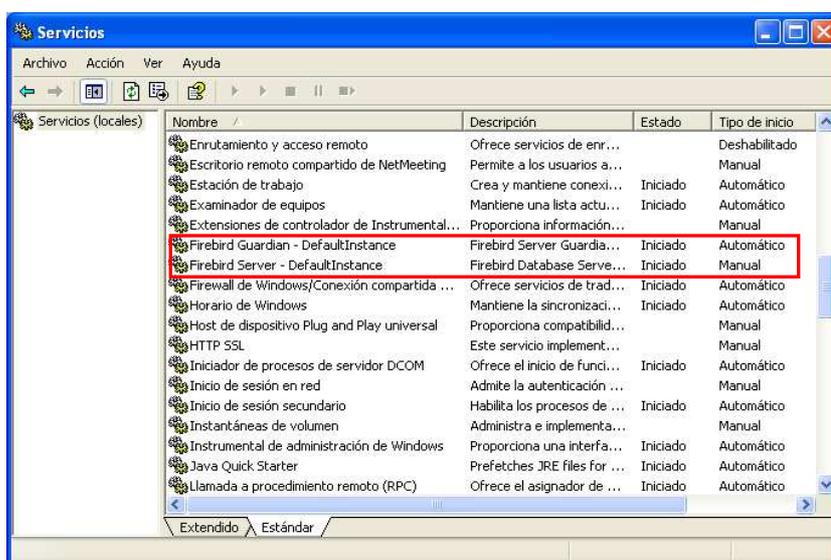
Substituya esta dirección IP por la dirección IP en que su servidor transmite.

Si se está conectando al servidor desde un cliente local esto es, un cliente corriendo en la misma máquina que el servidor, puede hacer ping al servidor virtual de loopback TCP/IP:

**ping localhost** o bien **ping 127.0.0.1**

- **Comprobar que el servidor Firebird está ejecutándose**

Luego de la instalación, el servidor Firebird debería estar corriendo como un servicio en Windows NT, 2000 o XP. Abra el Panel de Control/Herramientas administrativas/Servicios (Windows XP). La Figura II.6 muestra el applet de Servicios en Windows XP.



**Figura II.6** Comprobando que Firebird esté ejecutándose

- **Applets del panel de control para Windows**

Desde la versión 1.03, se incluye un applet para el panel de control en la distribución de Firebird. Aunque el applet no es esencial, provee una forma conveniente para arrancar y detener el servidor.



**Figura II.7** Firebird Server Control

Desafortunadamente, el applet incluido en el kit solamente funciona en Windows NT, 2000 y XP.

### 2.2.7. El cliente de Firebird en modo texto

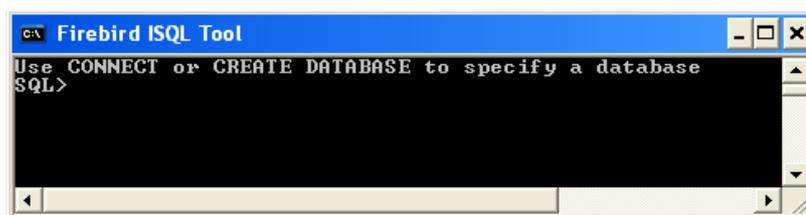
Con la distribución de Firebird se incluye una utilidad para ejecutar sentencias SQL contra el servidor, y mostrar sus resultados. Esta utilidad es de línea de comandos y se encuentra en el archivo *isql.exe* del subdirectorio bin de la instalación de Firebird.

Para acceder al cliente Firebird en modo texto puede realizarse de distintas formas:

- A través del punto del menú que se ha creado durante la instalación y que se puede visualizar en la Figura II.8



**Figura II.8** Acceso al cliente Firebird desde el menú Inicio



**Figura II.9** Cliente Firebird en modo texto

- O ingresando desde la línea de comandos, una vez dentro ir hasta la carpeta bin de la instalación de Firebird. Y ejecutar el programa escribiendo isql.

```
C:\Archivos de programa\Firebird\Firebird_2_1\bin>isql
```

Isql responderá con un indicador (prompt) especial 'SQL>' como se ve en la Figura II.9 Así pues se podrá empezar a trabajar con Firebird. Desde el indicador ejecutar las sentencias SQL.

### **2.2.7.1 Conectar a la base de datos de ejemplo**

Una vez abierta la ventana de isql (Figura II.9), se podrá conectar con la base de datos de ejemplo que viene con Firebird, ésta se encuentra en el subdirectorio examples de su instalación, para conectar escribir:

```
Connect "localhost:c:\Archivos de programa\Firebird\Firebird_2_1\examples\empbuild\employee.fdb" user "SYSDBA" password "masterkey";
```

Nótese que puede cortar la sentencia en dos o más líneas simplemente presionando <Enter>, isql indicará que una línea continua una sentencia anterior cambiando el indicador por un 'CON>'. Las sentencias se terminan con punto y coma (;).

Si la conexión se puede establecer, isql responderá con un mensaje indicando que está ahora conectado a la base de datos.

Database: "c:\Archivos de programa\Firebird\Firebird\_2\_1\examples\empbuild\employee.fdb", User: SYSDBA

A partir de ese momento se pueden ejecutar sentencias SQL sobre esa base de datos, o usar algunos comandos específicos de isql como *Show Tables*.

### **2.2.7.2 Crear una base de datos usando isql**

Para crear una base de datos en forma interactiva usando la interfaz de comandos de isql, se debe trabajar en el servidor. Posicione una terminal de comandos en el subdirectorio bin y arranque isql como sigue:

```
C:\Archivos de programa\Firebird\Firebird_2_1\bin>isql
```

```
Use CONNECT or CREATE DATABASE to specify a database
```

#### **La sentencia CREATE DATABASE**

Ahora se podrá crear una base de datos interactivamente. Por ejemplo se desea crear una base de datos llamada prueba.fdb y almacenarla en un directorio llamado datos en el disco local D:

```
SQL>CREATE DATABASE 'D:\datos\prueba.fdb' page_size 8192 user 'SYSDBA'  
password 'masterkey';
```

La base de datos será creada y luego de unos breves instantes, el prompt SQL volverá a aparecer. Ahora está conectado a la nueva base de datos y se puede proceder a crear algunos objetos de prueba en ella.

Para verificar que realmente hay una base de datos ahí, escribir esta consulta:

```
SQL>SELECT * FROM RDB$RELATIONS;
```

¡La pantalla se llenará con una gran cantidad de datos! Esta consulta selecciona todas las filas de la tabla de sistema adonde Firebird almacena los metadatos para las tablas. Una base de datos “vacía” no está vacía, contiene una base de datos que será completada con metadatos a medida que se cree objetos en ella.

**Creación de tablas.** Las tablas son las estructuras básicas que tienen que estar presentes en nuestra base de datos. Para crear tablas se utilizará la sentencia: create table. La sintaxis general es la siguiente:

```
CREATE TABLE nombre_tabla (nombre_campo1 tipo,....., nombre_campon tipo);
```

Existen condiciones para los campos como por ejemplo:

- NOT NULL: el registro no se introducirá en la tabla si este campo no tiene algún valor.
- DEFAULT: el campo tomará el valor por defecto especificado si no se indica ninguno. El valor puede ser una constante, la palabra 'NULL' (se asigna valor nulo), o la palabra 'USER' (se asigna el nombre del usuario autenticado actualmente). También se puede usar la palabra 'TODAY' para asignar la fecha actual, si el campo es de tipo DATE.

- **COLLATE:** válida para campos de caracteres, indica que se usará la secuencia de ordenamiento especificada.
- **COMPUTED BY:** el valor del campo se calculará aplicando una fórmula.

### **2.2.8. Mi primera base de datos**

Para facilitar las cosas, se va usar un programa IBExpert que ayudará en la creación de una base de datos y la gestión de la misma.

**Creación de una base de datos.** Para crear una base de datos ir al menú Database/Create Database, aparecerá una ventana y se ingresarán los datos requeridos.

**Server:** remoto o local

**Server name:** nombre del servidor de base de datos.

**Protocol:** protocolo a seguir para la comunicación con el servidor, normalmente TCP/IP

**Database:** path completo y nombre de la base de datos.

**Client Library Name:** dejar tal cual, es la librería donde está el API del motor.

**Username y password:** usuario y contraseña. Si no se ha modificado es sysdba y masterkey.

**Page\_size:** tamaño de la página

**Charset:** el ISO8859\_1 es el español

**SQL Dialect:** escoger el dialecto 3

Aceptado lo anterior, terminar de configurar los parámetros que tendrá la base de datos y registrarla.

**Creación de tablas y campos.** Para crear una tabla, dar clic con el botón derecho sobre Tables y seleccionar New table, se mostrará una pantalla desde donde se va a realizar todas las acciones pertinentes.



**Figura II.10** IBExpert

Aquí se definirá los campos que tendrá la tabla, así como el tipo y otras características. Una vez introducidos los campos y las características, aceptar los cambios realizados mediante el 'rayo amarillo'. Así se crearán tantas tablas como se necesiten.

### 2.2.9. Herramientas

Firebird no incluye una herramienta de administración visual; contiene un completo paquete de utilidades de línea de comando que permiten realizar las tareas que ofrece este SGDB. No todos los usuarios están familiarizados con la línea de comandos a la hora de trabajar con una base de datos, es por eso que existen en el mercado una gran cantidad de herramientas visuales de administración de bases de datos para Firebird sean estas comerciales o no, que proporcionan una interfaz gráfica de usuario para llevar a cabo las operaciones más comunes con este SGBD.

Existen varios programas clientes Firebird que permiten conectar con el servidor para acceder a las bases de datos o realizar tareas administrativas, a continuación se mencionan algunas de ellas:

- **Gbak.** Es una utilidad de Firebird en línea de comandos para respaldar y recuperar las bases de datos, se encuentra en el subdirectorío bin de la instalación. Las bases de datos de Firebird se pueden respaldar mientras hay usuarios conectados al sistema haciendo su trabajo normal. El respaldo será tomado de una instantánea del estado de la base de datos al momento de comienzo del mismo.
- **Isql.** Es una utilidad que viene en todas las versiones de Firebird, se encuentra en el directorío bin del mismo. Con esta utilidad se puede realizar cualquier proceso sql, crear bases de datos, modificarlas, ejecutar cualquier tipo de sentencia insert, select, update, delete, incluso se puede llamar con parámetros para ejecutar scripts sql, es una gran ayuda en muchas ocasiones. Su uso es muy simple, se debe ir al directorío donde está o escribir la ruta completa y su nombre, como cualquier otra utilidad.
- **Gfix.** Herramienta de Firebird en línea de comandos para la solución de problemas de administración como la reparación de los datos, eliminar temporales, etc.
- **GSec.** Es una utilidad modo texto para gestionar la seguridad en el servidor. Esta utilidad trabaja a partir de la base de datos de seguridad, security2.fdb, para especificar nombres de usuarios y passwords para acceder al servidor Firebird. La base de datos de seguridad, se encuentra en el directorío de instalación de Firebird, y cuando se solicita la conexión con una base de datos del servidor, el motor de Firebird comprueba que el nombre de usuario y password se correspondan con alguno de los registros de la misma.

- **Gstat.** Es una herramienta por la que se puede administrar a los usuarios existentes en el sistema. Mediante ella se puede dar de alta nuevos usuarios, borrar existentes, cambiar la contraseña, etc.
- **FBServer.** Es un proceso asociado a Firebird, este fichero suele ubicarse en C:\Archivos de programa\Firebird\Firebird\_2\_1\bin\fbserver.exe
- **FBGuard.** El proceso y archivo fbguard.exe es el Guardian, que está asociado a Firebird SQL Server. Este fichero suele ubicarse en C:\Archivos de programa\Firebird\Firebird\_2\_1\bin\fbguard.exe

Además se pueden encontrar gran cantidad de herramientas de terceros desarrolladas para utilizar con Firebird, entre ellas se tienen herramientas de administración, utilidades para realizar respaldos y restauración de las base de datos, herramientas de replicación, migración, reparación, etc.

- **Herramientas de administración gráficas.** Existen varias herramientas de administración, que permiten a desarrolladores y usuarios definir, administrar y manejar la base de datos Firebird. Entre las principales están IBExpert, FlameRobin, IBView, DB Workbench, Firebird Maestro, etc.
- **Utilidades de Backup/Restore.** Utilidades para respaldar y restaurar las bases de datos. Algunas de ellas son: DBak, ibBackup, GBak ScriptRunner.
- **Applets para el panel de Control.** Applets de Panel de Control para iniciar/parar Firebird.
- **Componentes de desarrollo.** Firebird PHP Generator, FIBPlus, IBOjects.
- **Herramientas de desarrollo.** Herramientas usadas para diseñar bases de datos Firebird e Interbase. Por ejemplo DeZign For Databases, Data Architect, etc.

- **Componentes de interfaz.** Componentes ODBC, ADO, OLEDB, Python, etc.
- **Herramientas de migración.** Herramientas que permiten a los desarrolladores migrar de las versiones más viejas de Interbase a las más nuevas versiones (Firebird), o herramientas que permiten la migración de una base de datos a otra.
- **Herramientas de reparación.** Herramientas que pueden usarse para reparar bases de datos Firebird e Interbase. Por ejemplo IBFirstAid y IBBackupSurgeon.
- **Herramientas de replicación.** Herramientas de replicación para bases de datos Firebird e Interbase. Algunas de ellas son: Replication Master, FBReplicator, etc.
- **Librerías UDF.** Un conjunto de Funciones Definidas por el Usuario para usar con Firebird e Interbase. Ejm: FUDlibrary, xLibUDF.

#### **2.2.10. Soporte Firebird**

Ninguna gran empresa presta soporte para Firebird, no hay quién gestione el desarrollo y gane dinero por ello. El patrocinador más grande hoy en día es IBPhoenix, y su principal negocio es Firebird. Además hay otras compañías, pero ninguna de ellas obtienen ganancias por vender licencias de Firebird (Firebird es totalmente gratis). Esas empresas venden soporte pero el retorno económico es mucho menor.

Por lo tanto, no hay nadie que invierta su dinero en mercadeo y haga un despliegue alrededor de este proyecto. Problemas similares plagan a otros proyectos de código abierto como PostgreSQL, por ejemplo.

#### **2.2.11. Conexión y uso de bases de datos Firebird en lenguaje PHP**

Los SGBD proporcionan un sistema de almacenamiento y consulta de los datos a los que se acceden desde las aplicaciones que desarrollemos. Así pues se va a estudiar las

principales funciones que permiten acceder al servidor de base de datos Firebird desde uno de los lenguajes de programación más utilizados como lo es PHP.

**ibase\_connect.** Abre una conexión con una base de datos Firebird/InterBase

```
int ibase_connect (string hostname, string usuario, string password)
```

Devuelve un identificador de enlace positivo si se tiene éxito o falso de lo contrario.

**ibase\_query.** Ejecuta una consulta en una base de datos Firebird/InterBase

```
int ibase_query(int identificador_de_enlace, string sentencia)
```

ibase\_query() envía una sentencia a la base activa en el servidor asociado al identificador de enlace, devuelve TRUE o FALSE para indicar si la sentencia se ha ejecutado correctamente o no.

**ibase\_fetch\_object.** Extrae una fila de resultado como un objeto.

```
object mysql_fetch_object(int id_resultado)
```

Devuelve un objeto con las propiedades que corresponden a la última fila extraída, o falso sino quedan más filas.

**ibase\_free\_result.** Libera un conjunto de resultados.

```
int ibase_free_result (int id_resultado)
```

**ibase\_close.** Cierra el enlace con la base de datos Firebird.

```
int ibase_close ([int identificador_de_enlace])
```

Devuelve: verdadero si tiene éxito, falso si existe error.

## **2.3. BASES DE DATOS EN MYSQL**

### **2.3.1. Introducción**

MySQL en los últimos años ha tenido un crecimiento vertiginoso. Es la base de datos de código abierto más popular del mundo. Código abierto significa que todo el mundo puede acceder al código fuente, es decir, al código de programación de MySQL, esto significa que también todos pueden contribuir con ideas, elementos, mejoras o sugerir optimizaciones. Y así es que MySQL ha pasado de ser una pequeña base de datos a una completa herramienta. Su rápido desarrollo se debe en gran medida a la contribución de mucha gente al proyecto, así como la dedicación del equipo de MySQL.

A diferencia de los proyectos propietarios, en los que el código fuente es desarrollado por un número reducido de personas y se protege atentamente, los proyectos de código abierto no excluyen a nadie interesado en aportar ideas, si disponen de los conocimientos necesarios.

Lo que en un tiempo se consideró como un sencillo juguete para uso en sitios Web, se ha convertido en la actualidad en una solución viable y de misión crítica para la administración de datos.

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario. Se trata de un programa capaz de almacenar una enorme cantidad de datos de gran variedad y de distribuirlos para cubrir las necesidades de cualquier tipo de organización. MySQL fue creada por la empresa sueca MySQL AB, que mantiene el copyright del código fuente del servidor SQL, así como también de la marca. MySQL se desarrolla como software libre en un esquema de licenciamiento dual.

Por un lado se ofrece bajo la GNU GPL (GNU General Public License) para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C.

Antes MySQL se consideraba como la opción ideal de sitios web; sin embargo, ahora incorpora muchas de las funciones necesarias para otros entornos y conserva su gran velocidad. MySQL es una base de datos robusta que se la puede comparar con una base de datos comercial, es incluso más veloz en el procesamiento de las transacciones y dispone de un sistema de permisos elegante y potente, y ahora, además, incluye un motor de almacenamiento InnoDB compatible con ACID, además dispone de store procedures, triggers, vistas.

MySQL es un caso particular, pues se trata de un programa de licencia open-source y gratuito pero que, sin embargo, está mantenido por una empresa (MySql AB). El código fuente de MySQL está sólo relativamente abierto y disponible para modificaciones, puesto que es la empresa MySQL AB la que contrata y coordina los trabajos de mantenimiento del producto. No obstante, los trabajadores contratados, procedentes de todo el mundo, son usuarios del producto que realizan sus encargos a través de Internet.

### **2.3.2. Historia**

El origen de MySQL se remonta a la década de los ochenta. Michael Widenius<sup>12</sup>, también conocido como Monty, un joven programador que realizaba complejas

---

<sup>12</sup> Fundador y creador del motor de base de datos de código abierto MySQL.

aplicaciones en lenguaje BASIC, al no encontrar un sistema de almacenamiento de archivos que le resultara satisfactorio, pensó en construir el suyo propio.

MySQL surgió cuando Michael Widenius comenzó a usar el gestor mSQL para conectar las tablas, usando rutinas de bajo nivel ISAM. Tras unas primeras pruebas, llegó a la conclusión de que mSQL no era lo bastante flexible ni rápido para lo que necesitaba, por lo que tuvo que desarrollar nuevas funciones. Esto lo llevaría a crear una interfaz SQL denominada MySQL, con una interfaz totalmente compatible a mSQL.

Años después, en 1995, y en colaboración con David Axmark, Widenius desarrolló un producto que básicamente era el resultado de sus investigaciones, más dos aportaciones nuevas: el uso del lenguaje SQL y la accesibilidad a través de Internet. Así nació MySQL y también la empresa MySQL AB.

Es así que el desarrollo de MySQL y la aparición de versiones que incluyen nuevas características y mejoras de las versiones anteriores se fueron dando. Así se comenzó y en enero de 1998 se lanzó la primera versión de MySQL para Windows 95 y NT, la versión 3.23 beta fue lanzada en agosto del 2000 y su liberación oficial se dio en enero del 2001.

El desarrollo de la versión 4 de MySQL empezó desde el 2001 y el lanzamiento de su versión beta se dio en agosto del 2002. MySQL 4.0 fue liberada oficialmente en marzo del 2003.

La mayoría del trabajo de desarrollo en la versión 4.0 se ha enfocado en tres áreas: mejorar las características y la eficiencia existentes, agregar nuevas características, y cambiar la arquitectura del software de MySQL para estar preparados con el futuro crecimiento. Entre las mejoras más importantes y las nuevas características que tiene MySQL 4.0 están: Se ha agregado InnoDB a la lista de tipos de tablas soportados en una instalación típica, o estándar. MySQL 4.0 soporta cinco tipos de tablas: MyISAM, ISAM, HEAP, BDB (Base de datos Berkeley), e InnoDB.

Una vez liberada la versión 4.0 el desarrollo se enfoca hacia la versión 4.1 con características planeadas para incluirse como: subconsultas, ayuda para todos los comandos desde el cliente MySQL, respaldos en línea sin decrementar la eficiencia del servidor, implementación del comando RENAME DATABASE, entre otras características. Oficialmente se liberó la versión 4.1 en octubre del 2004.

La primera versión beta 5.0 fue lanzada en marzo del 2004. Y en octubre del 2005 fue liberada oficialmente la versión 5.0 de MySQL, esta versión mejora las posibilidades de uso en aplicaciones comerciales ya que permite, contrariamente a las versiones anteriores, el uso de triggers, stored procedures y views. Funcionalidades que son imprescindibles para la implementación de aplicaciones profesionales con bases de datos. Otra de las novedades son los "Information Schemas" que permiten realizar consultas con el comando SELECT de SQL sobre meta datos. Por ejemplo, es posible obtener informaciones sobre diferentes propiedades de tablas, índices, etc. con bastante facilidad.

La evolución de MySQL ha sido realmente notable, y la comunidad de fuente abierta, con sus aportaciones y sugerencias hace que resulte una de las aplicaciones más dinámicas. Pero, sin duda, la mayor novedad relativa a MySQL en enero de 2008 fue la adquisición de esta reconocida base de datos que operaba en modo fuente abierta desde sus inicios por parte del fabricante Sun Microsystems<sup>13</sup>. Los primeros comunicados de prensa relativos a la adquisición hacían hincapié en que Sun mantendría la modalidad de fuente abierta para el producto, algo que ha hecho incluso con algunos de sus propios productos, como el sistema operativo Solaris. Esto hace esperar que MySQL reciba un fuerte impulso, tanto tecnológico como comercial y un mayor soporte del producto.

En enero del 2009 Sun Microsystems anunció la disponibilidad de MySQL 5.1, una nueva e importante versión de esta base de datos. Con esta versión se pretende mejorar la eficiencia y simplificar la administración de aplicaciones de bases de datos de gran escala. Fue considerada finalmente lista para producción y liberada.

En abril del 2009 la empresa Sun Microsystems fue adquirida por Oracle Corporation que continua con el desarrollo de MySQL como software libre en un esquema de licenciamiento dual.

Bien conocida por su amplia presencia en servidores Web a lo largo de todo el mundo, MySQL cuenta con diversas versiones que cubren desde la más básica y gratuita, pero dotada de un notable conjunto de funciones, hasta versiones apropiadas para soporte de aplicaciones empresariales críticas.

---

<sup>13</sup> Empresa informática recientemente adquirida por Oracle Corporation.

A sus conocidas cualidades de rendimiento, solidez y fiabilidad MySQL ha ido añadiendo un arsenal de herramientas para facilitar su gestión. Entre las novedades más destacables cabe citar el nuevo sistema de monitorización, DBA Advisors, que ofrece una valiosa información sobre el funcionamiento y mejora de la base de datos. DBA Advisors sugiere tanto ajustes para mejorar el rendimiento, como alertas ante eventuales problemas.

### **2.3.3. Características**

MySQL es muy rápido, seguro y fácil de usar y ha desarrollado un conjunto de características muy prácticas, en estrecha cooperación con otros usuarios. MySQL fue desarrollado para manejar grandes bases de datos mucho más rápido que las soluciones existentes y ha sido usado exitosamente en ambientes de producción con altas demandas, por varios años. Aunque está bajo un desarrollo constante, MySQL siempre ofrece conjunto de funciones muy poderoso y eficiente. La conectividad, velocidad y seguridad hace de MySQL una suite poderosa para acceder a bases de datos en Internet.

Entre las principales características de este gestor de bases de datos se encuentran las siguientes:

- Disponible en gran cantidad de plataformas y sistemas.
- Escrito en C y C++
- Dispone de API's en gran cantidad de lenguajes (C, C++, Java, PHP, Perl, Python, etc).
- Aprovecha la potencia de sistemas multiprocesador, gracias a su implementación multihilo.

- Soporta gran cantidad de tipos de datos para las columnas.
- Gestión de usuarios y passwords, manteniendo un muy buen nivel de seguridad en los datos.
- Gran portabilidad entre sistemas.
- Transacciones. A partir de las últimas versiones ya hay soporte para transacciones. En la versión 5.0.22, transacciones con los motores de almacenamiento InnoDB, BDB y Cluster.
- Puntos de recuperación (savepoints) con InnoDB
- Integridad referencial. MySQL tiene este soporte por defecto a partir de la versión 4.0 con el tipo de tabla InnoDB.
- Soporte para vistas, procedimientos almacenados, disparadores (triggers) que fueron implementados en las últimas versiones.
- Information Schema (Esquema de información). Para un fácil acceso a los metadatos.
- Cursores a partir de las últimas versiones.
- Embedded database library.
- Soporte completo para Unicode.
- Utiliza SQL, el lenguaje elegido para todos los Sistemas de Base de Datos modernos.
- Bajo costo en requerimientos para la elaboración de bases de datos, ya que debido a su bajo consumo puede ser ejecutado en una máquina con escasos recursos sin ningún problema.

- Conectividad y seguridad. MySQL está completamente preparado para el trabajo en red y las bases de datos pueden ser accedidas desde cualquier lugar de Internet. Dispone de control de acceso.
- Existen multitud de herramientas para trabajar con MySQL, como clientes gráficos.
- Fácil de instalar y configurar.
- Fácil de usar. Es un Sistema de Base de Datos de alto rendimiento pero relativamente simple y es mucho menos complejo de configurar y administrar que sistemas más grandes<sup>14</sup>.

Una de las características distintivas que son implementada únicamente por MySQL es:

- Múltiples motores de almacenamiento (MyISAM, Merge, InnoDB, BDB, Memory/heap, MySQL Cluster, Federated, Archive, CSV, Blackhole y Example en 5.x), permitiendo al usuario escoger la que sea más adecuada para cada tabla de la base de datos.

La mayoría de las personas que utilizan MySQL saben que MyISAM e InnoDB son los dos motores de almacenamientos más comunes en MySQL. También es sabido, que la mayoría no toma en cuenta el motor de almacenamiento al crear una tabla y acepta el que viene por default en la base de datos. Comparando MyISAM vs InnoDB, ninguno se destaca como la solución para la mayoría de los casos. Cada uno tiene sus pros y sus contras, por lo tanto al momento de decidir que motor de almacenamiento a utilizar dependerá mucho del escenario donde se aplique.

---

<sup>14</sup> <http://es.wikipedia.org/wiki/MySQL>

Los valores centrales de MySQL son:

- La mejor y más usada base de datos en el mundo.
- Disponible y accesible para todos.
- Fácil de usar.
- Se está perfeccionando continuamente mientras permanece rápida y segura.

#### **2.3.4. Tipos de datos**

MySQL soporta un número de tipos de columnas divididos en varias categorías: tipos numéricos, tipos de fecha y hora, y tipos de cadenas de caracteres<sup>15</sup>.

#### **TIPOS NUMÉRICOS**

Los tipos de datos numéricos comprenden dos categorías, los enteros y los números con punto flotante. Entre los tipos enteros se encuentran: tinyint, smallint, mediumint, int, bigint y en los de punto flotante: float, double, decimal.

#### **BIT[(M)]**

En un tipo de datos bit. M indica el número de bits por valor, de 1 a 64. El valor por defecto es 1 si se omite M.

Este tipo de datos se añadió en MySQL 5.0.3 para MyISAM, una extensión en 5.0.5 para MEMORY, InnoDB, y BDB. Antes de 5.0.3, BIT es un sinónimo de TINYINT(1).

#### **TINYINT[(M)]**

Un entero muy pequeño. El rango con signo es de -128 a 127. El rango sin signo es de 0 a 255.

---

<sup>15</sup> <http://dev.mysql.com/doc/refman/5.0/es/index.html>  
[http://www.ebookee.com/La-Biblia-de-MySQL\\_129482.html](http://www.ebookee.com/La-Biblia-de-MySQL_129482.html)

### **BOOL, BOOLEAN**

Son sinónimos para TINYINT(1). Un valor de cero se considera falso. Valores distintos a cero se consideran ciertos.

### **SMALLINT[(M)]**

Un entero pequeño. El rango con signo es de -32768 a 32767. El rango sin signo es de 0 a 65535.

### **MEDIUMINT[(M)]**

Entero de tamaño medio. El rango con signo es de -8388608 a 8388607. El rango sin signo es de 0 a 16777215.

### **INT[(M)]**

Un entero de tamaño normal. El rango con signo es de -2147483648 a 2147483647. El rango sin signo es de 0 a 4294967295.

### **INTEGER[(M)]**

Es un sinónimo de INT.

### **BIGINT[(M)]**

Un entero grande. El rango con signo es de -9223372036854775808 a 9223372036854775807. El rango sin signo es de 0 a 18446744073709551615.

### **FLOAT(p)**

Número con coma flotante. p representa la precisión. Puede ir de 0 a 24 para números de coma flotante de precisión sencilla y de 25 a 53 para números de coma flotante con doble precisión. Estos tipos son como los tipos FLOAT y DOUBLE descritos a continuación. FLOAT(p) tiene el mismo rango que los tipos correspondientes FLOAT y DOUBLE, pero la anchura de muestra y el número de decimales no están definidos.

Esta sintaxis se proporciona para compatibilidad con ODBC.

### **FLOAT[(M,D)]**

Un número de coma flotante pequeño (de precisión simple). Los valores permitidos son de  $-3.402823466E+38$  a  $-1.175494351E-38$ , 0, y de  $1.175494351E-38$  a  $3.402823466E+38$ . Si se especifica **UNSIGNED**, los valores negativos no se permiten. M es el número total de dígitos y D es el número de dígitos después del punto decimal. .  
FLOAT sin argumentos o **FLOAT(p)** (donde p está en el rango de 0 a 24) es un número de coma flotante con precisión simple.

### **DOUBLE[(M,B)]**

Número de coma flotante de tamaño normal (precisión doble). Los valores permitidos son de  $-1.7976931348623157E+308$  a  $-2.2250738585072014E-308$ , 0, y de  $2.2250738585072014E-308$  a  $1.7976931348623157E+308$ . Si se especifica **UNSIGNED**, no se permiten valores negativos. M es el número total de dígitos y B es el número de bits de precisión. **DOUBLE** sin parámetros o **FLOAT(p)** (donde p está en el rango de 25 a 53) es un número de coma flotante con doble precisión. Un número de coma flotante con precisión sencilla tiene una precisión de 7 decimales aproximadamente; un número con coma flotante de doble precisión tiene una precisión aproximada de 15 decimales.

### **DOUBLE PRECISION[(M,D)], REAL[(M,D)]**

Son sinónimos de **DOUBLE**. Excepción: Si el modo del servidor SQL incluye la opción **REAL\_AS\_FLOAT**, **REAL** es un sinónimo para **FLOAT** en lugar de **DOUBLE**.

### **DECIMAL[(M[,D])]**

Número de punto fijo exacto y *empaquetado*. M es el número total de dígitos y D es el número de decimales. El punto decimal y (para números negativos) el signo '-' no se tiene en cuenta en M. Si D es 0, los valores no tienen punto decimal o parte fraccional.

El máximo número de dígitos (M) para DECIMAL es 64. El máximo número de decimales soportados (D) es 30. Si UNSIGNED se especifica, no se permiten valores negativos.

Si se omite D, el valor por defecto es 0. Si se omite M, el valor por defecto es 10.

Todos los cálculos básicos (+, -, \*, /) con columnas DECIMAL se hacen con precisión de 64 dígitos decimales.

**DEC[(M[,D])], NUMERIC[(M[,D])], FIXED[(M[,D])]**

Son sinónimos para DECIMAL. El sinónimo FIXED está disponible por compatibilidad con otros servidores.

**Tabla II.V.** Tipos de datos numéricos en MySQL

TIPO	TAMAÑO	RANGO/PRECISIÓN (Con signo/Sin signo)	DESCRIPCIÓN
TINYINT	1 byte	-128 a 127 0 a 255	Entero muy pequeño
SMALLINT	2 bytes	-32768 a 32767 0 a 65535	Entero pequeño
MEDIUMINT	3 bytes	-8388608 a 8388607 0 a 16777215	Entero mediano
INT INTEGER	4 bytes	-2147483648 a 2147483647 0 a 4294967295	Entero normal
BIGINT	8 bytes	-9223372036854775808 a 9223372036854775807 0 a 18446744073709551615	Entero grande
BIT	aproximadamente (M+7)/8 bytes	1 a 64	
BOOL BOOLEAN	1 byte	0 o 1	
FLOAT(p) p: precisión	4 bytes si 0 ≤ p ≤ 24, 8 bytes si 25 ≤ p ≤ 53		
FLOAT	4 bytes	-3.402823466E+38 a -1.175494351E-38, 0, y desde 1.175494351E-38 a 3.402823466E+38	
DOUBLE[(M[,B])]	8 bytes	-1.7976931348623157E+308 a -2.2250738585072014E-308, 0	Punto flotante de tamaño normal

		y desde 2.2250738585072014E-308 a 1.7976931348623157E+308	
REAL DOUBLE PRECISION	4 bytes ó 8 bytes		
DECIMAL DEC NUMERIC FIXED	M+2 bytes sí D > 0, M+1 bytes sí D = 0, D+2 si M < D	El rango máximo es el mismo que para el tipo DOUBLE	Cadena de caracteres representando un número flotante (es decir se almacena como una cadena de caracteres y no en formato binario)

### **TIPOS DE FECHA Y HORA**

Entre los tipos de fecha y hora están los siguientes:

#### **DATE**

Almacena una fecha. El rango soportado es de '1000-01-01' a '9999-12-31'. MySQL muestra valores DATE en formato 'YYYY-MM-DD', pero permite asignar valores a columnas DATE usando cadenas de caracteres o números.

#### **DATETIME**

Combinación de fecha y hora. El rango soportado es de '1000-01-01 00:00:00' a '9999-12-31 23:59:59'. MySQL muestra valores DATETIME en formato 'YYYY-MM-DD HH:MM:SS', pero permite asignar valores a las columnas DATETIME usando cadenas de caracteres o números.

#### **TIMESTAMP[(M)]**

Combinación de fecha y hora. El rango es de '1970-01-01 00:00:00' hasta el año 2037.

Una columna TIMESTAMP es útil para registrar la fecha y hora de una operación INSERT o UPDATE. La primera columna TIMESTAMP en una tabla se rellena automáticamente con la fecha y hora de la operación más reciente si no le asigna un

valor. Puede asignar a cualquier columna **TIMESTAMP** la fecha y hora actual asignándole un valor **NULL**.

En MySQL 5.0, **TIMESTAMP** se retorna como una cadena de caracteres en el formato 'YYYY-MM-DD HH:MM:SS' cuya anchura de muestra son 19 caracteres. Si quiere obtener el valor como un número, debe añadir +0 a la columna timestamp.

### **TIME**

Almacena una hora. El rango es de '-838:59:59' a '838:59:59'. MySQL muestra los valores **TIME** en formato 'HH:MM:SS', pero permite asignar valores a columnas **TIME** usando números o cadenas de caracteres.

### **YEAR[(2|4)]**

Almacena un año en formato de dos o cuatro dígitos. El valor por defecto está en formato de cuatro dígitos. En formato de cuatro dígitos, los valores permitidos son de 1901 a 2155, y 0000. En formato de dos dígitos, los valores permitidos son de 70 a 69, representando los años de 1970 a 2069. MySQL muestra los valores **YEAR** en formato **YYYY** pero permite asignar valores a columnas **YEAR** usando cadenas de caracteres o números.

**Tabla II.VI.** Tipos de datos fecha y hora en MySQL

<b>TIPO</b>	<b>TAMAÑO</b>	<b>RANGO/PRECISIÓN</b>	<b>DESCRIPCIÓN</b>
DATE	3 bytes	1000-01-01 a 9999-12-31	Fecha
DATETIME	8 bytes	1000-01-01 00:00:00 a 9999-12-31 23:59:59	Fecha y hora
TIMESTAMP	4 bytes	1970-01-01 00:00:00 hasta el año 2037	Fecha y hora
TIME	3 bytes	-838:59:59 a 838:59:59	Hora
YEAR	1 byte	El rango es de 1901 a 2155 para cuatro dígitos, para dos dígitos es de 70 a 69	Almacena un año en formato de dos o cuatro dígitos

## **TIPOS DE CADENAS DE CARACTERES**

### **CHAR (M)**

Una cadena de caracteres de longitud fija que siempre tiene el número necesario de espacios a la derecha para ajustarla a la longitud especificada al almacenarla. M representa la longitud de la columna. El rango de M es de 0 a 255 caracteres.

### **VARCHAR(M)**

Cadena de caracteres de longitud variable. M representa la longitud de columna máxima. El rango de M es de 0 a 65535 en MySQL 5.0.3 y posterior. (La longitud máxima real de un VARCHAR en MySQL 5.0 se determina por el tamaño de registro máximo y el conjunto de caracteres que use. La longitud máxima efectiva desde MySQL 5.0.3 es de 65,532 bytes.)

### **BINARY(M)**

El tipo BINARY es similar al tipo CHAR, pero almacena cadenas de datos binarios en lugar de cadenas de caracteres no binarias. Es decir, que contienen cadenas de bytes en lugar de cadenas de caracteres. Esto significa que no tienen conjunto de caracteres, y la comparación y ordenación se basa en los valores numéricos de los bytes en los valores.

### **VARBINARY(M)**

El tipo VARBINARY es similar al tipo VARCHAR, pero almacena cadenas de caracteres binarias en lugar de cadenas de caracteres no binarias. Es decir, que contienen cadenas de bytes en lugar de cadenas de caracteres. Esto significa que no tienen conjunto de caracteres, y la comparación y ordenación se basa en los valores numéricos de los bytes en los valores.

### **TINYBLOB**

Una columna BLOB con una longitud máxima de 255 ( $2^8 - 1$ ) bytes.

### **TINYTEXT**

Una columna TEXT con longitud máxima de 255 ( $2^8 - 1$ ) caracteres.

### **BLOB[(M)]**

Tipo de dato binario que puede almacenar cualquier tipo de datos o información, desde un archivo de texto con todo su formato (se diferencia en esto de el tipo Text) hasta imágenes, archivos de sonido o video, etc. Al igual que el tipo Text, Blob admite hasta 65535 ( $2^{16} - 1$ ) bytes.

### **TEXT[(M)]**

Tipo de datos no binario que sirve para almacenar texto. Con una longitud máxima hasta 65535 ( $2^{16} - 1$ ) caracteres.

### **MEDIUMBLOB**

Una columna BLOB con longitud de 16.777.215 ( $2^{24} - 1$ ) bytes.

### **MEDIUMTEXT**

Una columna TEXT con longitud máxima de 16.777.215 ( $2^{24} - 1$ ) caracteres.

### **LOBLOB**

Una columna BLOB con longitud máxima de 4.294.967.295 o 4GB ( $2^{32} - 1$ ) bytes. La longitud máxima efectiva (permitida) de las columnas LOBLOB depende del tamaño máximo configurado para los paquetes en el protocolo cliente/servidor y la memoria disponible.

### **LONGTEXT**

Una columna TEXT con longitud máxima de 4.294.967.295 o 4GB ( $2^{32} - 1$ ) caracteres. La longitud máxima efectiva (permitida) de columnas LONGTEXT depende del tamaño máximo de paquete configurado en el protocolo cliente/servidor y la memoria disponible.

**ENUM('value1','value2',...)**

Una enumeración. Un objeto de cadena de caracteres que sólo puede tener un valor, elegido de una lista de valores 'value1', 'value2',..., NULL o el valor de error especial ''.

Una columna ENUM puede tener un máximo de 65535 valores distintos. Los valores ENUM se representan internamente como enteros.

**SET('value1','value2',...)**

Un conjunto. Un objeto de cadena de caracteres que puede tener cero o más valores que deben pertenecer a la lista de valores 'value1', 'value2', ... Una columna SET puede tener un máximo de 64 miembros. Los valores SET se representan internamente como enteros.

**Tabla II.VII.** Tipos de datos cadenas de caracteres en MySQL

TIPO	TAMAÑO	RANGO/PRECISIÓN	DESCRIPCIÓN
CHAR(M)	M bytes	0 a 255	Almacena cadenas de longitud fija
VARCHAR(M)	M+1 byte	0 a 65535	Almacena cadenas de longitud variable
BINARY(M)	M bytes	0 a 255	Similar a CHAR
VARBINARY(M)	M+1 byte	0 a 65535	Similar a VARCHAR
TEXT	Longitud + 2 bytes	Hasta 65535 caracteres	rango de 255 – 65535 caracteres
BLOB	Longitud + 2 bytes	Hasta 65535 bytes	
TINYTEXT	Longitud + 1 bytes	Hasta 255 caracteres	
TINYBLOB	Longitud + 1 bytes	Hasta 255 bytes	
MEDIUMTEXT	Longitud + 3 bytes	Hasta 16.777.215 caracteres	
MEDIUMBLOB	Longitud + 3 bytes	Hasta 16.777.215 bytes	
LONGTEXT	Longitud + 4 bytes	Hasta 4.294.967.295 caracteres	
LOB	Longitud + 4 bytes	Hasta 4.294.967.295 bytes	
ENUM(valor1, valo2, ...)	1 ó 2 bytes, dependiendo del número de valores de ENUM	Hasta 65535 valores distintos	
SET(valor1, valo2, ...)	1, 2, 3, 4, ó 8 bytes, dependiendo del número de valores del conjunto	La lista puede tener un máximo de 64 valores	

### 2.3.5. Motores de almacenamiento

Una peculiaridad de MySQL es que cuando se crea una nueva tabla se puede especificar su tipo. MySQL soporta una serie de tipos de tablas que se distinguen por tener propiedades diferenciadas. Las tres más importantes son MyISAM, InnoDB y MEMORY. Si al crear una tabla no se especifica el tipo, el servidor decide basándose en su configuración<sup>16</sup>.

- MyISAM trata tablas no transaccionales. Proporciona almacenamiento y recuperación de datos rápida, así como posibilidad de búsquedas fulltext (permite buscar subcadenas de texto sobre un campo). 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.

La carencia más importante de MyISAM es que no incluye soporte para *transacciones*. Una transacción consiste en una serie de alteraciones que deben realizarse de forma simultánea en un conjunto de registros para mantener la integridad de los datos.

La principal característica de este tipo de almacenamiento es la gran velocidad que obtiene en las consultas, ya que no tiene que hacer comprobaciones de integridad referencial. Este tipo de tablas está especialmente indicado para sistemas que no tienen un número elevado de inserciones como pueden ser las páginas web.

- INNODB está considerado como uno de los motores más avanzados para el almacenamiento de datos en MySQL. Su característica principal es que soporta

---

<sup>16</sup> <http://dev.mysql.com/doc/refman/5.0/es/index.html>

transacciones de tipo ACID y bloqueo de registros e integridad referencial. InnoDB ofrece una fiabilidad y consistencia muy superior a MyISAM, si bien el mejor rendimiento de uno u otro formato dependerán de la aplicación específica.

- El motor de almacenamiento BDB (BerkeleyDB). Este motor, independiente de MySQL, provee altas prestaciones. Posee un mecanismo de almacenamiento basado en hash de alta eficiencia, lo que facilita el rápido acceso a los datos de forma directa a costa de la lentitud en el acceso secuencial. Las tablas BDB pueden tener una gran probabilidad de sobrevivir a fallos del sistema y ser capaces de realizar COMMIT y ROLLBACK en transacciones.

- El motor de almacenamiento MEMORY crea tablas con contenidos que se almacenan en memoria y no en el disco (lo que permite obtener una velocidad muy alta, ya que el acceso a la memoria es más rápido). Éstas se conocían previamente como HEAP. Usan índices hash que les da un tiempo de acceso rápido a registros individuales. Estas tablas se usan normalmente para almacenar datos temporales. Sin embargo, cuando se apaga el servidor, todos los datos almacenados en las tablas MEMORY se pierden. Las tablas por sí mismas continúan existiendo ya que sus definiciones se almacenan en ficheros .frm en disco, pero están vacías cuando reinicia el servidor.

- El motor de almacenamiento MERGE, también conocido como MRG\_MyISAM, es una colección de tablas MyISAM idénticas que pueden usarse como una. "Idéntica" significa que todas las tablas tienen información de columna e índice idéntica. No puede mezclar tablas en que las columnas se listen en orden distinto, no tengan exactamente

las mismas columnas, o tengan los índices en orden distinto. Sin embargo, alguna o todas las tablas pueden comprimirse con `myisampack`.

Al permitir combinar varias tablas de igual estructura en una única tabla, se puede realizar consultas sobre una tabla que devuelve datos de varias.

Como MyISAM, los motores de almacenamiento MEMORY y MERGE tratan tablas no transaccionales y ambos se incluyen en MySQL por defecto.

- 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, tiene interés principalmente 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 MySQLMax 5.0. Este motor de almacenamiento está disponible para Linux, Solaris, y Mac OS X. Se añadirá soporte para este motor de almacenamiento en otras plataformas, incluyendo Windows en próximas versiones.

Este motor está pensado para aplicaciones que necesitan un gran tráfico de datos, con mucha concurrencia, y con niveles muy altos de disponibilidad y seguridad. Se construye usando un conjunto de ordenadores entre los que no hay compartición de recursos. La característica fundamental es que los datos se almacenan en memoria.

- El motor de almacenamiento ARCHIVE se usa para almacenar grandes cantidades de datos sin índices en muy poco espacio, ya que los comprime alcanzando un nivel de ahorro de espacio considerable. ARCHIVE es una tabla MyISAM, pero comprimida y de sólo lectura.
- El motor de almacenamiento CSV almacena la información utilizando el formato de valores separados por comas (comma-separated values), de forma que cada tabla es un fichero que contiene los datos. No soporta indexado y su fin principal es permitir exportar los datos de forma que puedan ser importados fácilmente por algunas suites ofimáticas.
- El motor de almacenamiento FEDERATED se añadió en MySQL 5.0.3. Es un motor que accede a datos en tablas de bases de datos remotas en lugar de tablas locales.
- BLACKHOLE el sorprendente uso de este motor es no almacenar los datos sino crear un log con la consulta SQL utilizada. Como no almacena ningún dato lógicamente no soporta índices, ni transacciones. Su principal utilidad es mantener un servidor esclavo que mantenga un log del sistema principal.
- FALCON es el futuro motor de almacenamiento para MySQL 6.0, actualmente se encuentra en estado alpha, y está diseñada para entornos de servidores Web de alto volumen.

Naturalmente es posible utilizar distintos motores en una misma base de datos, la decisión correcta depende en gran parte del uso futuro que tendrá la base de datos. Y eso, suele ser difícil saberlo con exactitud durante el diseño.

De todas formas, estos son algunos de los criterios a considerar:

- **Transacciones y Concurrencia.** Si se necesita transacciones y alta concurrencia de usuarios, probablemente InnoDB será la mejor opción. Es decir, que se obtendrá mejores condiciones de rendimiento si las tablas que se vayan a ver involucradas frecuentemente en consultas transacciones y que sufran una alta concurrencia de usuarios se basan en el motor InnoDB.
- Si la concurrencia no es tan alta, pero se sigue necesitando transacciones, tanto InnoDB como BDB pueden ser la solución.
- El motor MyISAM no soporta transacciones, por lo que será la mejor opción cuando no se las necesite.
- Si la aplicación depende de la capacidad para contar rápidamente el número de registros de una tabla se recurre a MyISAM, que siempre sabe cuantos registros tiene sin necesidad de contarlos. Sin embargo, InnoDB debe realizar la cuenta.
- Pero sin embargo, InnoDB será la solución en los casos que se necesite disponer de integridad referencial.
- Se debe tener muy en cuenta que el almacenamiento transaccional es usado si se desea más seguridad, y para el caso que se desee mayor velocidad se utilizará un motor de almacenamiento no transaccional.

**Tabla II.VIII. Motores de almacenamiento**

	<b>MyISAM</b>	<b>BDB</b>	<b>MEMORY</b>	<b>InnoDB</b>	<b>ARCHIVE</b>	<b>NDB</b>	<b>CSV</b>
Transacciones	No	Si	No	Si	No	Si	No
Integridad referencial	No	No	No	Si			No
Granularidad de bloqueo	Tabla	Página (8KB)	Tabla	Registro	Registro	Registro	
Almacenamiento	Ficheros separados	Un fichero por tabla	En memoria	Tablespace	Buffer	En memoria	Fichero de texto
Límites de almacenamiento	Ilimitado	Ilimitado	Limitado	64TB	Ilimitado	Limitado	
Índices	B-Tree Fulltext	B-Tree Hash	B-Tree Hash	B-Tree Hash Clustered	No soporta	B-Tree Hash	No soporta
Soporte geoespacial	Si	No	No	No	No	No	
Compresión de datos	Si	No	No	No	Si	No	
Encriptación de datos	Si	Si	Si	Si	Si	Si	
Costo de almacenamiento	Bajo	Bajo	-	Alto	Muy bajo	Alto	Bajo
Costo de memoria	Bajo	Bajo	Medio	Alto	Bajo	Alto	
Velocidad de inserción	Alto	Alto	Alto	Bajo	Muy alto	Alto	
Cluster DataBase Support	No	No	No	No	No	Si	
Soporte para réplica	Si	Si	Si	Si	Si	Si	
Soporte para claves foráneas	No	No	No	Si	No	No	

Cuando se crea una nueva tabla, puede decirle a MySQL qué tipo de tabla crear añadiendo la opción de tabla ENGINE o TYPE al comando CREATE TABLE, además se puede convertir una tabla de un tipo a otro, usando el comando ALTER TABLE que indique el nuevo tipo: ALTER TABLE nombre\_tabla ENGINE = nuevo\_tipo;

### 2.3.6. Instalación y puesta en marcha

La instalación, configuración y modo de trabajo son comunes a todas las plataformas, distinguiéndose en leves diferencias; ya que la comunidad MySQL intenta publicar sus versiones definitivas para la mayoría de sistemas operativos de forma paralela.

Una vez que se tenga el paquete de instalación de MySQL, se inicia la instalación a través del asistente. De entre las diferentes modalidades de instalación que ofrece

MySQL se encuentran las opciones Typical, Complete y Custom. La instalación Típica está recomendada para los usuarios que desconocen los parámetros de instalación e instala MySQL Server con los valores por defecto. Las otras posibilidades de instalación son Complete, mediante la cual se instalan todas las características del producto, y Custom que deja al usuario la elección sobre qué partes instalar y cuáles no.

Luego de haber realizado la instalación se tendrá la posibilidad de Configurar el Servidor iniciando el asistente de forma automática. Esta opción estará disponible también después de la instalación como MySQL Server Instance Config Wizard y también puede ejecutarse desde la línea de comando del sistema operativo con la orden MySQLInstanceConfig.exe. Con cualquiera de las tres posibilidades iniciar el asistente de configuración MySQL Server Instance.

### **2.3.6.1 Configuración**

Si se ha dejado marcada la casilla *Configurar ahora el servidor MYSQL (Configure the MySQL Server now)* de la última pantalla de instalación, entonces en ese momento empezará el proceso de configuración. Y aparecerá una pantalla que indica si llevar a cabo una instalación *detallada (detailed)* o *estándar (standard)*.

- La *instalación detallada* propone una configuración automática y óptima para el servidor.
- La *instalación estándar* ofrece una configuración para un uso general del producto. Además, la instalación estándar permite seguir el proceso paso a paso y, si procede, hacer cambios en las opciones por defecto que se van ofreciendo.

Se seleccionará la forma de configuración deseada, por nuestra parte se ha optado por la estándar. Al elegir el modo estándar, en primer lugar aparecerá una nueva pantalla con dos características a configurar. Por una parte, se pregunta si instalar MySQL como un servicio de Windows (recomendable efectuar la instalación de esta forma, puesto que así tiene la opción de hacer que se inicie automáticamente cada vez que arranque Windows). Por tanto, si se desea habilitar dicho inicio automático, dejar marcada la casilla Arrancar el servidor MySQL automáticamente (Launch the MySQL Server automatically). No obstante, si se prefiere iniciar el servicio manualmente, entonces desmarcar dicha opción.

La otra opción de configuración de la que dispone es incluir el directorio Bin en el PATH de Windows. En este directorio se encuentran las instrucciones necesarias para ejecutar MySQL desde la línea de comandos. Esta casilla también dejarla marcada.



**Figura II.11** Configurando MySQL

A continuación, asignar una contraseña al usuario que va a crearse. Dicho usuario se denominará root y será el administrador del sistema.



**Figura II.12** Asignando contraseña al usuario root

En esta misma pantalla se encuentran otras dos posibilidades más que el usuario puede configurar:

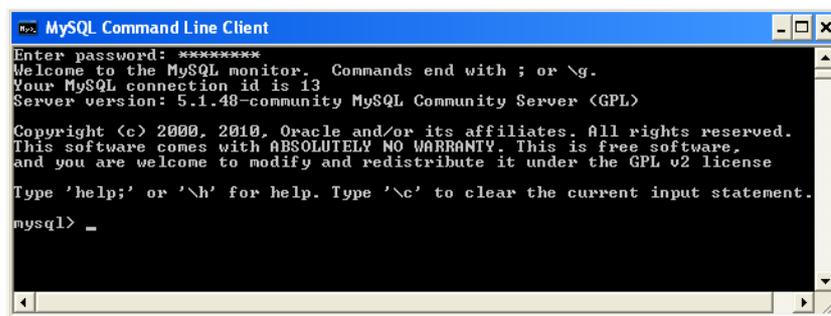
- Crear un usuario anónimo (sin nombre ni contraseña), opción que lógicamente rehusaremos dejando sin marcar la casilla correspondiente.
- Entrar al servidor desde máquinas remotas, a través de la red o incluso desde Internet (*Enable root access from remote machines*). Esta opción deberá activarse sólo si se instala MySQL en un servidor de red al cual se accederá desde los diferentes terminales que se le conectan.

Continuar con la configuración de MySQL y por último pulsar en el botón Execute (Ejecutar) y se irán marcando las casillas según se vayan realizando las distintas tareas. Finalmente, si todo el proceso se ha llevado a cabo de forma adecuada finalizar la configuración y se tendrá instalado y configurado el servidor.

## Acceso al cliente MySQL en modo texto

El cliente MySQL en modo texto es un programa que a través de la línea de comandos, permite trabajar con la base de datos en un entorno interactivo donde el usuario debe escribir las sentencias que desea ver ejecutadas.

El acceso al programa cliente en modo texto se lo puede realizar a través del punto del menú que se ha creado durante la instalación. Se entrará directamente como usuario root, sólo pedirá la contraseña, ingresarla y si la conexión ha sido correcta se podrá empezar a trabajar con MySQL.



**Figura II.13** Cliente MySQL en modo texto

También se puede acceder al programa, desde la línea de comandos escribiendo la siguiente sentencia: `mysql -h servidor -u usuario -p`

Si se escoge la segunda opción, se podrá seleccionar tanto el servidor como el usuario. Sin embargo al seleccionar la primera posibilidad se accederá directamente a la máquina local y como usuario root, que tiene permisos de administrador. Por tanto esta opción no debería estar accesible para los usuarios finales.

### 2.3.7. Conectarse y desconectarse al servidor

Para conectarse al servidor, generalmente se necesita proveer a MySQL un nombre de usuario y un password. Si el servidor corre en una máquina distinta a la que se está

utilizando se debe especificar el nombre del host (máquina). Cuando se conocen todos estos parámetros la conexión se realiza de la siguiente manera:

```
mysql -h servidor -u usuario -p
```

Donde, servidor es el nombre del servidor al que se va a conectar (localhost si se trabaja en modo local) y usuario es el nombre de usuario con el cual se entrará a la base de datos.

Si la conexión se realizó con éxito, MySQL despliega una información de ingreso seguida por el prompt mysql>

Después que se ha realizado la conexión, se puede desconectarse del servidor en cualquier momento escribiendo el comando quit.

## **2.3.8. Administración de una Base de Datos**

### **2.3.8.1 Creación de una Base de Datos**

Para crear una base de datos se debe tener permiso para poder crear base de datos en el servidor MySQL, si se tiene el permiso entonces la sentencia a seguir es:

```
CREATE DATABASE nombre_database;
```

Al crear una base de datos, está no se selecciona para su uso, debe hacerlo explícitamente. Para ello se debe usar el siguiente comando:

```
USE nombre_database;
```

Para usar una base de datos ésta debe seleccionarse cada vez que se inicia una sesión de mysql. Esto puede hacerse a través del comando USE como se indicó anteriormente, o puede indicar la base de datos en la línea de comandos al ejecutar mysql. Simplemente

debe indicar el nombre de la base de datos a continuación de los parámetros que necesite ingresar:

```
mysql -h servidor -u usuario -p nombre_database
```

Para consultar que base de datos actualmente está en uso se utiliza la siguiente consulta:

```
select database();
```

### 2.3.8.2 Creación de tablas

Para crear las tablas que va a contener la base de datos, se realiza de la siguiente forma:

```
CREATE TABLE nombre_tabla(campo_1 tipo(tamaño), campo_2 tipo(tamaño), ...,  
campo_n tipo(tamaño));
```

El campo indica el nombre de la columna y tipo(tamaño) especifica el tipo de dato y el espacio que se va a conservar para cada dato almacenado en esa columna. Ejemplo: código int(5), nombre char(25), fecha date, etc. Cuando se trata de fechas no se especifica el tamaño, puesto que ya está determinado.

**Tabla II.IX.** Atributos de columna

Atributo	Significado
null	Se permiten valores nulos, atributo por omisión si no se especifica lo contrario.
not null	No se permiten valores nulos.
default valor	Valor por omisión que se asigna a la columna.
auto_increment	El valor se asigna automáticamente incrementando en uno el máximo valor registrado hasta ahora. Se aplica sólo a las columnas marcadas como clave primaria.
primary key	Señala al campo como clave primaria, implícitamente también lo declara como not null.

**Tabla II.X.** Restricciones de tabla

<b>Restricción</b>	<b>Significado</b>
primary key	Define la o las columnas que servirán como clave primaria. Las columnas que forman parte de la clave primaria deben de ser not null.
unique	Define las columnas en las que no pueden duplicarse valores. Serán las claves candidatas del modelo relacional.
foreing key (columna) references tabla (columna2)	Define que los valores de columna se permitirán sólo si existen en tabla (columna2). Es decir, columna hace referencia a los registros de tabla, esto asegura que no se realicen referencias a registros que no existen.  Es importante saber que las claves ajenas sólo funcionarán si las tablas son de tipo InnoDB.

### 2.3.8.3 Ingreso de datos a las tablas

Para ingresar información a una tabla se puede hacer básicamente de dos maneras. La primera se utiliza cuando se tiene mucha información a ingresar de una sola vez, entonces es conveniente almacenar esta información en un archivo de texto, es decir, .txt. El fichero de texto debe contener un registro por línea, con cada valor separado por un carácter de tabulación y dispuestos en el orden en el cual se especificaron las columnas en la sentencia CREATE TABLE. Para valores ausentes, puede usar valores NULL, para representar estos valores en el archivo de texto se utiliza \N

Una vez que se tiene este archivo, se procede de la siguiente forma:

```
LOAD DATA LOCAL INFILE 'nombre_archivo.txt' INTO TABLE nombre_tabla;
```

Cuando lo que se desea es agregar nuevos registros de uno a uno se utilizará la sentencia INSERT:

```
INSERT INTO nombre_tabla VALUES ('valor_1','valor_2', ..., 'valor_n');
```

Los datos a ingresar se separan por comas, estos datos indican los valores que van a tomar cada una de las columnas, según el orden en que fueron creadas. En el caso que se quiera ingresar un valor NULL no es necesario las comillas, sólo se coloca NULL.

#### **2.3.8.4 Extraer información de una tabla**

La sentencia SELECT es utilizada para recuperar la información que ésta contenida en una tabla. La sintaxis general de esta sentencia es:

```
SELECT seleccionar_esto
```

```
FROM desde_tabla
```

```
WHERE condiciones;
```

Donde seleccionar\_esto es lo que se quiere ver. Puede ser una lista de columnas, o \* para indicar “todas las columnas”, desde\_tabla indica la tabla donde están los datos a recuperar. La cláusula WHERE es opcional. Si está presente, condiciones representa las condiciones que cada registro debe cumplir para retornar como resultado.

#### **Seleccionar todos los datos**

Es la forma más simple de SELECT, recupera todo lo que hay en la tabla:

```
SELECT * FROM nombre_tabla;
```

#### **Seleccionar registros específicos**

Como se ha visto anteriormente, es fácil recuperar una tabla en su totalidad. Sólo debe omitir la cláusula WHERE en la sentencia SELECT. Pero, generalmente, no se desea ver la tabla completa, especialmente cuando alcanza un gran tamaño. En cambio,

usualmente, se tiene interés en obtener una respuesta para una consulta en particular, en cuyo caso se especifican algunas restricciones para la información que se traerá.

```
SELECT * FROM nombre_tabla WHERE nombre_columna='lo que se desee buscar';
```

### **Seleccionar columnas concretas**

Si no se quieren ver filas completas, solo hace falta indicar las columnas en las que se está interesado, separadas por comas.

```
SELECT nombre_columna_1, nombre_columna_n FROM nombre_tabla;
```

### **2.3.9. Programas cliente y utilidades MySQL**

Hay muchos programas clientes MySQL distintos que conectan con el servidor para acceder a la base de datos o realizar tareas administrativas. Existen también otras utilidades, que no comunican con el servidor, pero que realizan operaciones relacionadas con MySQL<sup>17</sup>.

La siguiente lista describe brevemente los programas clientes y las utilidades MySQL:

- **myisampack.** Utilidad que comprime tablas MyISAM para producir tablas más pequeñas de sólo lectura.
- **mysql.** La herramienta de línea de comando para introducir comandos SQL interactivamente o ejecutarlos desde un fichero en modo batch.
- **mysqlaccess.** Script que verifica los permisos de acceso para una combinación de máquina, usuario y base de datos.

---

<sup>17</sup> <http://dev.mysql.com/doc/refman/5.0/es/index.html>

- **mysqladmin.** Cliente que realiza tareas administrativas, tales como crear y borrar bases de datos, recargar las tablas de permisos, volcar tablas a disco y reabrir ficheros de log. mysqladmin también puede utilizarse para consultar la versión, información de procesos, e información de estado del servidor.
- **mysqlbinlog.** Utilidad para leer comandos de un log binario. El log de comandos ejecutados contenidos en el fichero de log binario puede utilizarse para ayudar en la recuperación de una fallida.
- **MySQL AB** proporciona tres programas con interfaces gráficas para usar con el servidor MySQL:
  - **MySQL Administrator:** Esta herramienta se usa para administrar servidores MySQL, bases de datos, tablas y usuarios.
  - **MySQL Query Browser:** MySQL AB proporciona esta herramienta gráfica para crear, ejecutar y optimizar consultas en bases de datos MySQL.
  - **MySQL Migration Toolkit:** Esta herramienta ayuda en la migración de esquemas y datos de otros sistemas de gestión de bases de datos relacionales a MySQL.
- **mysqlcheck.** Cliente de mantenimiento de tablas que verifica, repara, analiza y optimiza tablas.
- **mysqldump.** El cliente mysqldump puede utilizarse para volcar una base de datos o colección de bases de datos para copia de seguridad o para transferir datos a otro servidor SQL (no necesariamente un servidor MySQL). EL volcado contiene comandos SQL para crear la tabla y/o rellenarla.

- **mysqlhotcopy.** Utilidad que realiza copias de seguridad rápidas de tablas MyISAM o ISAM mientras el servidor está en ejecución. El script está escrito en Perl y bloquea las tablas mientras realiza el respaldo para evitar su modificación.
- **mysqlimport.** Cliente que importa ficheros de texto en sus respectivas tablas usando LOAD DATA INFILE.
- **mysqlshow.** Cliente que muestra información de bases de datos, tablas, columnas, e índices.
- **perror.** Utilidad que muestra el significado de los errores de sistema de MySQL.
- **replace.** Programa que cambia cadenas de caracteres en ficheros o en la entrada estándar.

Cada programa MySQL tienen varias opciones, pero todos ellos proporcionan una opción --help que puede utilizarse para obtener una descripción completa de las distintas opciones del programa. Por ejemplo, mysql --help.

### 2.3.10. MySQL GUI Tools

MySQL GUI Tools es un paquete de herramientas gratuito que incluye tres utilidades gráficas importantes: MySQL Administrator, MySQL Migration Toolkit y MySQL Query Browser.

**MySQL Administrator** es un programa que permite realizar operaciones administrativas, tales como la configuración, monitorización o el inicio y el cierre de los servidores MySQL. También permite administrar los usuarios y las conexiones, realizar copias de seguridad y un gran número de tareas administrativas.

La mayoría de las tareas pueden realizarse mediante una línea de comandos, como `mysqladmin` o `mysql`, aunque MySQL Administrator tiene las siguientes ventajas:

- Interfaz gráfica de usuario.
- Proporciona una visión general mucho mejor mostrando valores cruciales para el rendimiento la fiabilidad y la seguridad de los servidores MySQL.
- Muestra indicadores gráficos del rendimiento, haciendo más sencilla la administración y la configuración del servidor.



**Figura II.14** MySQL Administrator

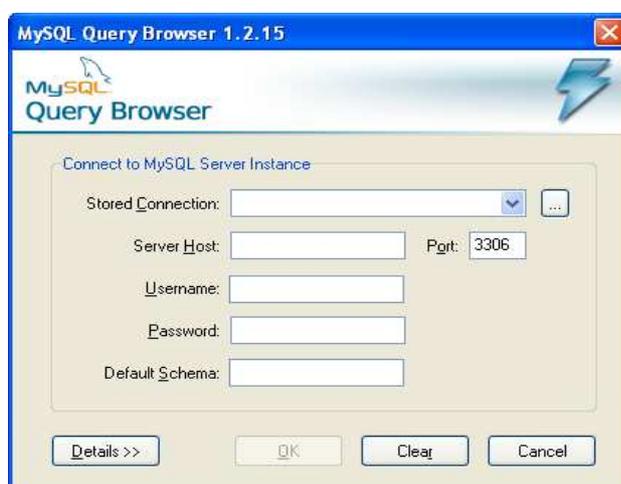
Los datos a introducir son:

- Server Host: Es el nombre del equipo o IP donde está el servidor de base de datos MySQL.
- Port MySql: 3306
- Username: Usuario con el que se conectará a la base de datos.
- Password: Contraseña del usuario.

**MySQL Migration Toolkit** es una herramienta gráfica proporcionada por MySQL AB para posibilitar la migración de los esquemas y datos desde varios sistemas de bases de datos a MySQL.

**MySQL Query Browser** es una herramienta gráfica proporcionada por MySQL AB para la creación, ejecución y optimización de las consultas en un entorno gráfico. Mientras MySQL Administrator está diseñado para gestionar un servidor MySQL, MySQL Query Browser está diseñado para ayudarle a consultar y analizar los datos almacenados en su base de datos MySQL.

También permite ejecutar las consultas desde la línea de comandos utilizando la utilidad mysql e incluso permite consultar y editar los datos de una manera más intuitiva y gráfica.



**Figura II.15** MySQL Query Browser

Los datos a introducir son:

- Server Host: Es el nombre del equipo o IP donde está el servidor de base de datos MySQL.
- Port MySql : 3306
- Username: Usuario con el que no se conectará a la base de datos.
- Password: Contraseña del usuario.
- Default Schema: Nombre de la base de datos con la que se trabajará.

- **Stored Connection:** Permite crear conexiones a la base de datos para no tener que estar introduciendo el nombre del equipo, usuario y contraseña cada vez que se quiera conectar al servidor.

### **2.3.11. Soporte MySQL**

MySQL Network provee soporte y servicios directos de los desarrolladores de MySQL asegurando una respuesta confiable, segura y rápida. En la página web las personas interesadas en la herramienta tienen acceso a listas de correo, foros, documentación en línea, información técnica de temas específicos.

El soporte de MySQL se divide en los siguientes tipos para los clientes con aplicaciones de bases de datos crítica:

**Mysql Network Basic.** Provee consultores independientes para pequeños negocios con servicios que mejoran la productividad. Tiene acceso a software certificado y optimizado, mantenimiento, updates, ingreso a la base de conocimiento de MySQL. Si tiene incidentes, puede realizar un máximo de dos incidencias, acceso web y tiene un tiempo máximo de respuesta de 2 días. Este tipo de soporte cuesta \$595.00 por servidor por año.

**Mysql Network Silver.** Provee los mismos servicios que el básico en lo que se refiere a software, pero en la parte de resolución de problemas, puede tener un número de incidentes ilimitado, acceso web, acceso telefónico 8x5 y un límite de 4 horas en el tiempo de respuesta. El costo es 1995.00 por servidor y por año.

**Mysql Network Gold.** Tiene las mismas particularidades del básico y plata, pero en la resolución de problemas tiene número de incidentes ilimitado, acceso web, acceso telefónico 24x7, máximo tiempo de respuesta a incidencias 2 horas, además, en caso de emergencias, tiene un tiempo máximo de respuesta de 30 minutos, también dispone de resolución de problemas en forma remota. El costo es de 2995.00 por servidor por año.

**Mysql Network Platinum.** Además de las particularidades de software de las anteriores, tiene como opción el desarrollo de proyectos especiales, tiene un número de incidentes ilimitado, acceso web, acceso a llamadas telefónicas 24x7, máximo tiempo de respuesta en incidencias 30 min, en caso de emergencia tiene también 30 minutos de tiempo base. Para soporte de consultaría tiene resolución de problemas remotamente, revisión de código, revisión de performance, revisión de código para manejar las diferentes API's que soporta MySQL y funciones definidas por el usuario. El costo es de 4995.00 por servidor por año.

### **2.3.12. Conexión y uso de bases de datos MySQL en lenguaje PHP**

Uno de los principales objetivos de un SGBD es proporcionar un sistema de almacenamiento y consulta de datos al que se puedan conectar las aplicaciones que se desarrollen. Así pues a continuación se estudia las funciones que permiten acceder a MySQL desde PHP uno de los lenguajes de programación más utilizados.

**mysql\_connect.** Abre una conexión a un servidor MySQL.

```
int mysql_connect ([string hostname [:puerto] [:camino/al/socket] [, string usuario [,string password]])
```

Devuelve un identificador de enlace positivo si tiene éxito, o falso en caso de error. Todos los argumentos son opcionales, y si no hay, se asumen los valores por defecto ('localhost', usuario propietario del proceso del servidor, password vacía).

**mysql\_select\_db.** Selecciona una base de datos MySQL.

```
int mysql_select_db (string base_de_datos [, int identificador_de_enlace])
```

Devuelve: true si tiene éxito, false en caso de error.

mysql\_select\_db() establece la base activa que estará asociada con el identificador de enlace especificado. Si no se especifica un identificador de enlace, se asume el último enlace abierto. Si no hay ningún enlace abierto, la función intentara establecer un enlace como si se llamara a mysql\_connect().

**mysql\_query.** Envía una sentencia SQL a MySQL

```
int mysql_query (string sentencia [, int identificador_de_enlace])
```

mysql\_query() envía una sentencia a la base activa en el servidor asociado al identificador de enlace. Si no es especificado un *identificador\_de\_enlace*, se asumirá el último enlace abierto. Si no hay ningún enlace abierto, la función intenta establecer un enlace como si se llamara a la función mysql\_connect() sin argumentos, y lo utiliza.

Devuelve TRUE o FALSE para indicar si la sentencia se ha ejecutado o no.

**mysql\_fetch\_array.** Extrae la fila de resultado como una matriz asociativa.

```
array mysql_fetch_array (int id_resultado [, int tipo_de_resultado])
```

Devuelve una matriz que corresponde a la sentencia extraída, o falso si no quedan más filas.

`mysql_fetch_array()` es una versión extendida de `mysql_fetch_row()`. Además de guardar los datos en el índice numérico de la matriz, guarda también los datos en los índices asociativos, usando el nombre de campo como clave.

**`mysql_free_result`**. Libera la memoria del resultado.

```
int mysql_free_result (int id_resultado)
```

`mysql_free_result()` solo necesita ser llamada si te preocupa usar demasiado memoria durante la ejecución del script. Toda la memoria del resultado especificado en el parámetro del identificador de resultado será automáticamente liberada.

**`mysql_close`**. Cierra el enlace con MySQL.

```
int mysql_close ([int identificador_de_enlace])
```

Devuelve: verdadero si tiene éxito, falso si existe error.

`mysql_close()` cierra el enlace con la base de datos MySQL que está asociada con el identificador de enlace especificado. Si no se especifica el identificador de enlace, se asume por defecto el último enlace.

## **CAPÍTULO III**

### **ESTUDIO COMPARATIVO DE MYSQL Y FIREBIRD**

#### **3.1. Introducción**

En la actualidad existen gran cantidad de Sistemas de Gestión de Bases de Datos, y en nuestro medio se conoce muy poco de cuál es el motor de base de datos más apropiado para desarrollar una aplicación.

El presente capítulo tiene como objetivo fundamental, analizar y evaluar los Sistemas Gestores de Base de Datos libres MySQL y Firebird para seleccionar cual es el mejor motor de base de datos, para ello se tomarán en cuenta factores como: tiempo de respuesta en realizar los procesos, características que ofrecen los SGBD, volúmenes de información que soporta, entre otros; considerando estos criterios como importantes en la elección de un SGBD. Para determinar el tiempo que tardan los SGBD en realizar los procesos, se someterán a pruebas de inserción, selección, eliminación y actualización de datos con diferentes cantidades de registros, lo que permitirá determinar cuál de estos dos SGBD presenta mejor desempeño en la ejecución de las pruebas mencionadas

anteriormente, ofreciendo el menor tiempo de respuesta, desde diferentes perspectivas tales como almacenamiento y extracción de datos.

La decisión de realizar los comparativos entre estos dos SGDB, es porque hoy en día la tecnología “Open Source” ha tomado fuerza y se ha convertido en una buena opción para el almacenamiento de los datos de cualquier organización, ya sean estas pequeñas medianas o grandes empresas.

Una vez realizada la comparación se procederá a determinar el mejor Gestor de Base de Datos en base a las pruebas realizadas y a las características que se han tomado en cuenta para el estudio comparativo, criterios que no se verán involucrados en la ejecución de las pruebas pero se tienen en cuenta para seleccionar el SGBD.

### **3.2. Descripción del problema**

Un Sistema Gestor de Base de Datos, debe desempeñar a un nivel que sea válido para los usuarios, esto quiere decir que debe responder con tiempos aceptables de respuesta a las solicitudes, satisfaciendo así completamente las necesidades de los usuarios.

En este contexto el problema se enfoca principalmente en determinar el tiempo de respuesta de los SGBD MySQL y Firebird, sometido a diferentes pruebas que permitirán determinar que tan rápido responde a situaciones de carga extrema ya sean en almacenamiento o extracción de datos.

### **3.3. Recursos Hardware y Software utilizados**

Con la finalidad de que las pruebas se realicen en un entorno de total igualdad para ambos SGBD, se utilizarán los mismos recursos físicos y lógicos ejecutándose la prueba en la misma computadora.

- **Recursos Hardware**

La computadora utilizada en las pruebas tiene las siguientes características:

- Procesador: Intel Pentium Dual-Core, 2.50 GHz
- Memoria RAM de 1 GB
- Disco duro de 300 GB

- **Recursos Software**

Para el desarrollo de las pruebas se requiere instalar:

- Gestor de Base de Datos Firebird
- Gestor de Base de Datos MySQL
- MySQL GUI Tools
- IBExpert
- MS Windows XP Service Pack 3

### **3.4. Diseño de la Base de Datos**

Para la realización de las pruebas se creó una base de datos con la misma cantidad de tablas, estructura, tipos de datos, llaves entre otros; en ambos Sistemas de Gestión de Base de Datos, se ingresará el mismo volumen de información y longitud en cada uno de los datos.

A continuación se muestran las tablas de la base de datos que se ha tomado de ejemplo para la realización de las pruebas.

usuario(*cod\_usu*,*ci*,*nombre*,*apellido*,*direccion*,*telefono*,*cod\_ran*)

prestamo(*num\_prest*,*fecha*,*total\_matp*,*cod\_usu*)

inventario(*cod\_mat*,*nom\_mat*,*modelo*,*estado*)

prest\_mat(*num\_p*,*cod\_m*)

reparacion(*cod\_mat*,*fecha*,*falla*,*observacion*)

### **3.5. Definición de los parámetros de evaluación**

Para la realización de las pruebas se optó por tener en cuenta aquellos parámetros que al momento de evaluarlos se les pudiera dar un valor cuantitativo y que estuvieran dentro de los criterios importantes al momento de seleccionar un SGBD. Además existen criterios (ver sección 3.12) que no se vieron involucrados durante la ejecución de las pruebas pero si se tiene en cuenta para la realización del estudio comparativo y la determinación del mejor SGBD.

#### **Tiempo de ejecución**

El tiempo de ejecución que un SGBD ofrece al realizar las operaciones, es un factor importante cuando se está trabajando con un gran volumen de información, a mayor volumen proporcionalmente será importante el desempeño en el SGBD. Las pruebas están diseñadas para medir tiempos de respuesta que presentan MySQL y Firebird durante la ejecución de cada una de las pruebas. A menor tiempo de respuesta mejor es el desempeño de un SGBD.

#### **Integridad en los Datos**

Cuando se habla de integridad, se refiere a la precisión y suficiencia de la información, así como a su validez de acuerdo con los valores y expectativas del negocio.

Durante la ejecución de las pruebas es posible que los datos sufran algún tipo de “alteración” o se pierdan, ocasionado por problemas físicos en la estructura de la tabla o posiblemente por problemas del SGBD al momento de procesar tablas con millones de registros. Es por esta razón que después de cada prueba se realizará una consulta a los registros almacenados comprobando la integridad y existencia. Para este criterio el punto de medición es el número de inconsistencias.

### **Tolerancia a Fallos**

Es muy importante que un Sistema de Gestión de Bases de Datos se recupere de anomalías ya sean de hardware, software o red, tomando el control los sistemas alternos o de respaldo propios de cada SGBD, sin pérdida de datos o transacciones.

Esta prueba consiste en realizar una interrupción eléctrica en el servidor durante la actualización de registros, luego de ello se realizará una verificación de la información para probar la integridad de los datos.

A continuación se determinan los parámetros e indicadores propuestos para la comparación:

**Tabla III.XI. Parámetros**

<b>Parámetros</b>	<b>Indicadores</b>	<b>Observaciones</b>
Tiempo de ejecución	Tiempo de respuesta en la inserción	Tiempo en el cual realiza la inserción de datos, según la cantidad de filas especificado.
	Tiempo de respuesta en la selección	Tiempo en el cual realiza la selección de datos, según la cantidad de filas especificado.
	Tiempo de respuesta en la actualización	Tiempo en el cual realiza la actualización de datos, según la cantidad de filas especificado.
	Tiempo de respuesta en la eliminación	Tiempo en el cual realiza la eliminación de datos, según la cantidad de filas especificado.
Integridad en los datos	Cantidad de registros	Número de registros insertados, seleccionados o actualizados.
Tolerancia a fallos	Cantidad de registros	

### **3.6. Determinación de los escenarios de prueba**

El rendimiento que ofrece un SGBD es de gran importancia en su elección, es así que es necesario conocer los tiempos de respuesta de MySQL y Firebird en realizar los procesos más habituales como son inserciones, consultas, actualizaciones y eliminaciones, desde diferentes puntos de vista como pueden ser la ejecución de consultas con diferentes grados de complejidad y con diferentes cantidades de registros, así como también el volumen de información que pueden soportar.

Para la realización de las pruebas, se han definido 5 escenarios con valores: 1.000, 10.000, 100.000, 500.000, 1.000.000 de registros; en cada uno de los escenarios se realizarán operaciones como: inserción, selección, actualización y eliminación.

Con los escenarios propuestos se analizará el efecto en los tiempos de respuesta, que produce la variación del número de registros en realizar las operaciones antes mencionadas en los SGBD en estudio.

Para las pruebas de inserción de registros se crearon procedimientos almacenados para cada motor de base de datos, el mismo que genera el número de registros que se ingresen como parámetro en el procedimiento almacenado, de acuerdo a los escenarios establecidos anteriormente.

Para las pruebas de consultas se han definido tres tipos con diferentes grados de complejidad:

**Consulta simple.** La consulta simple se encarga de obtener 4 campos de dos tablas de la base de datos, haciendo un join entre las tablas préstamo y usuario, además realiza un ordenamiento.

**Consulta de complejidad media.** Esta consulta se encarga de obtener 6 campos de tres tablas, haciendo 3 join y con dos ordenamientos.

**Consulta de complejidad alta.** Obtiene 8 campos de toda la base de datos, haciendo 4 joins, donde se cumpla una condición y además se realizan tres ordenamientos.

### 3.7. Diseño de las pruebas

Las pruebas a realizar a MySQL y Firebird, están diseñadas para medir los tiempos de respuesta en los escenarios propuestos, a partir de los cuales se podrá determinar el SGBD que ofrezca menor tiempo de respuesta en la ejecución de cada una de las pruebas.

#### 3.7.1. Prueba N° 01: Inserción

**Tabla III.XII.** Prueba N° 01

<b>Prueba N° 01</b>	
PROCESO	Inserción
OBJETIVO	Realizar pruebas de inserción de datos con diferentes cantidades de registros para obtener el tiempo de respuesta en realizar la inserción en cada SGBD.
PARÁMETROS	Tiempo de ejecución Integridad en los datos
RECURSOS NECESARIOS	MySQL Command Line Client IBExpert
PROCEDIMIENTO A EMPLEAR	<ol style="list-style-type: none"> <li>1. Se crearán los procedimientos almacenados para insertar los datos.</li> <li>2. Ejecutar los procedimientos almacenados.</li> <li>3. Se obtendrá el tiempo de duración en realizar las pruebas.</li> <li>4. Se realizará una consulta para verificar la información ingresada</li> </ol>
CÓDIGO	<pre>create procedure sp_insertar(in numreg int) begin declare i int; declare a int; set a=numreg div 2; set i=1; while i &lt;= numreg do</pre>

	<pre>insert into usuario values(i,i,'nombres usuario','apellidos usuario','direccion', '098791556','R001'); set i=i+1; end while; set i=1; while i &lt;= numreg do if (i&lt;=a) then insert into prestamo values(i,'2010-04-02','1',i); else insert into prestamo values(i,'2010-04-02','1',i-a); end if; set i=i+1; end while; set i=1; while i &lt;= numreg do if (i mod 2 = 0) then insert into inventario values(i,'nombre del material o equipo','DSC 001','B'); else insert into inventario values(i,'nombre del material o equipo','DSC 001','P'); end if; set i=i+1; end while; set i=1; while i &lt;= numreg do insert into prest_mat values(i,i); set i=i+1; end while; set i=1; while i &lt;= numreg do if (i&lt;=a) then insert into reparacion values(i,'2010-07-10','falla del material o equipo', 'observaciones'); else insert into reparacion values(i-a,'2010-07-10','falla del material o equipo', 'observaciones'); end if; set i=i+1; end while;</pre>
--	---

	<p>end;</p> <p>Los procedimientos almacenados serán ejecutados pasando como parámetro de entrada el número de registros a insertar, según los escenarios establecidos para las pruebas.</p>
OBSERVACIONES	<p>Se realizó una consulta a los registros insertados comprobando así la integridad y existencia de los datos.</p> <p>Las pruebas de inserción fueron cumplidas con éxito para ambos SGBD</p>

### 3.7.2. Prueba N° 02: Consultas simples

**Tabla III.XIII. Prueba N° 02**

<b>Prueba N° 02</b>	
PROCESO	Selección
OBJETIVO	Realizar las pruebas de selección simple para determinar el tiempo de respuesta en cada SGBD
PARAMETROS	Tiempo de ejecución Integridad en los datos
RECURSOS NECESARIOS	Sistemas Gestores de Bases de Datos MySQL y Firebird MySQL GUI Tools IBExpert
PROCEDIMIENTO EMPLEAR	A <ol style="list-style-type: none"> <li>1. Crear la consulta simple</li> <li>2. Ejecutar la consulta simple</li> <li>3. Obtener el tiempo de respuesta en realizar las pruebas</li> </ol>
CÓDIGO	<pre>select prestamo.num_prest,usuario.nombre,usuario.apellido, prestamo.fecha from usuario inner join prestamo on usuario.cod_usu=prestamo.cod_usu order by prestamo.num_prest;</pre> <p>Se ejecutará la consulta en diferentes cantidades de registros.</p>
OBSERVACIONES	<p>En ambos SGBD se obtuvo la cantidad de registros adecuado y además se verificó que la información sea la solicitada, comprobando así la integridad de los datos.</p> <p>Las pruebas de selección con grado de complejidad simple fueron realizadas con éxito para los dos SGBD.</p>

### 3.7.3. Prueba N° 03: Consulta de complejidad media

**Tabla III.XIV. Prueba N° 03**

<b>Prueba N° 03</b>	
PROCESO	Selección
OBJETIVO	Realizar las pruebas de selección con mediana complejidad para determinar el tiempo de respuesta en cada SGBD
PARAMETROS	Tiempo de ejecución Integridad en los datos
RECURSOS NECESARIOS	Sistemas Gestores de Bases de Datos MySQL y Firebird MySQL GUI Tools IBExpert
PROCEDIMIENTO EMPLEAR	A 1. Crear la consulta de complejidad media 2. Ejecutar la consulta 3. Obtener el tiempo de respuesta en realizar las pruebas
CÓDIGO	<pre>select prest_mat.num_p,usuario.cod_usu,usuario.nombre, usuario.apellido,prest_mat.cod_m,prestamo.fecha from prest_mat inner join prestamo on prest_mat.num_p=prestamo.num_prest inner join usuario on prestamo.cod_usu=usuario.cod_usu order by prest_mat.num_p,prest_mat.cod_m;</pre> <p>Se ejecutará la consulta en diferentes cantidades de registros.</p>
OBSERVACIONES	En los dos SGBD se obtuvo la cantidad de registros adecuado y además se verificó que la información sea la solicitada, comprobando así la integridad de los datos.  Las pruebas de selección con grado de complejidad media fueron realizadas con éxito para los dos SGBD.

### 3.7.4. Prueba N° 04: Consulta de complejidad alta

**Tabla III.XV. Prueba N° 04**

<b>Prueba N° 04</b>	
PROCESO	Selección
OBJETIVO	Realizar las pruebas de selección de complejidad alta para determinar el tiempo de respuesta en cada SGBD
PARAMETROS	Tiempo de ejecución Integridad en los datos

RECURSOS NECESARIOS	Sistemas Gestores de Bases de Datos MySQL y Firebird MySQL GUI Tools IBExpert
PROCEDIMIENTO EMPLEAR	A 1. Crear la consulta compleja 2. Ejecutar la consulta compleja 3. Obtener el tiempo de respuesta en realizar las pruebas
CÓDIGO	select distinct prest_mat.num_p,prestamo.cod_usu,usuario.nombre, usuario.apellido,prest_mat.cod_m,inventario.nom_mat, prestamo.fecha, reparacion.fecha from usuario inner join prestamo on usuario.cod_usu=prestamo.cod_usu inner join prest_mat on prestamo.num_prest=prest_mat.num_p inner join inventario on prest_mat.cod_m=inventario.cod_mat inner join reparacion on inventario.cod_mat=reparacion.cod_mat where inventario.estado='P'  order by prest_mat.num_p,prest_mat.cod_m,prestamo.cod_usu;  Se ejecutará la consulta en diferentes cantidades de registros.
OBSERVACIONES	En ambos SGBD se obtuvo la cantidad de registros adecuado y además se verificó que la información sea la solicitada, comprobando así la integridad de los datos.  Las pruebas de selección compleja fueron realizadas con éxito para los dos SGBD.

### 3.7.5. Prueba N° 05: Eliminación

**Tabla III.XVI.** Prueba N° 05

Prueba N° 05	
PROCESO	Eliminación
OBJETIVO	Conocer cuál de los dos SGBD es más rápido al eliminar la información.
PARAMETROS	Tiempo de ejecución
RECURSOS NECESARIOS	MySQL Command Line Client IBExpert
PROCEDIMIENTO EMPLEAR	A 1. Crear la instrucción para eliminar los registros. 2. Ejecutar la instrucción para eliminar los registros en cada SGBD. 3. Obtener el tiempo de respuesta en realizar las pruebas

	4. Realizar una consulta para verificar que toda la información haya sido eliminada.
OBSERVACIONES	El objetivo de la prueba fue cumplido con éxito para los dos SGBD. Se realizaron las verificaciones pertinentes para verificar que todos los registros hayan sido borrados.

### 3.7.6. Prueba N° 06: Actualización a un campo

**Tabla III.XVII. Prueba N° 06**

Prueba N° 06	
PROCESO	Actualización
OBJETIVO	Conocer cuál de los dos SGBD es más rápido al actualizar un campo.
PARAMETROS	Tiempo de ejecución
RECURSOS NECESARIOS	Sistemas Gestores de Bases de Datos MySQL y Firebird Tabla usuario con diferentes cantidades de registros
PROCEDIMIENTO A EMPLEAR	<ol style="list-style-type: none"> <li>1. Crear la instrucción para la actualización.</li> <li>2. Ejecutar la instrucción para actualizar los registros de la tabla usuario en cada SGBD.</li> <li>3. Obtener el tiempo de respuesta en realizar las pruebas</li> <li>4. Comprobar que toda la información haya sido actualizada.</li> </ol>
CÓDIGO	<pre>update usuario SET direccion='riobamba' where direccion='direccion';</pre> <p>Se ejecutará la consulta en diferentes cantidades de registros.</p>
OBSERVACIONES	El objetivo de la prueba fue cumplido con éxito para ambos SGBD.

### 3.7.7. Prueba N° 07: Interrupción eléctrica

**Tabla III.XVIII. Prueba N° 07**

Prueba N° 07	
OBJETIVO	Conocer en qué estado se encontrarán los registros de las tablas de la base de datos luego de sufrir una interrupción eléctrica mientras se estaba realizando una actualización.
PARAMETROS	Tolerancia a fallos
RECURSOS NECESARIOS	Sistemas Gestores de Bases de Datos MySQL y Firebird Tabla usuario con diferentes cantidades de registros

PROCEDIMIENTO EMPLEAR	A <ol style="list-style-type: none"><li>1. Verificar que al reiniciar la computadora el servicio del SGBD al que estamos haciendo la prueba se ejecute automáticamente.</li><li>2. Ejecutar la instrucción para actualizar la información de un campo de la tabla usuario.</li><li>3. Luego de ejecutar la instrucción, se procederá a realizar la interrupción eléctrica al servidor, esto se lo hará desconectando el cable de poder del CPU.</li><li>4. A continuación se restablecerá el servicio de energía al servidor, dando inicio al sistema y posteriormente de forma automática al SGBD.</li><li>5. Acceder a la base de datos y verificar la información</li></ol>
OBSERVACIONES	Se realizó las pruebas en 100.000, 500.000 y 1.000.000 registros. Se realizaron las pruebas para verificar la integridad de la información y era como si no se hubiera ejecutado la instrucción UPDATE.

### 3.8. Ejecución de las pruebas en los SGBD

#### 3.8.1. MySQL

Primeramente se implementa las operaciones que deberán ser evaluadas, utilizando la herramienta MySQL Administrator (utilidad de MySQL GUI Tools que permite administrar el servidor).

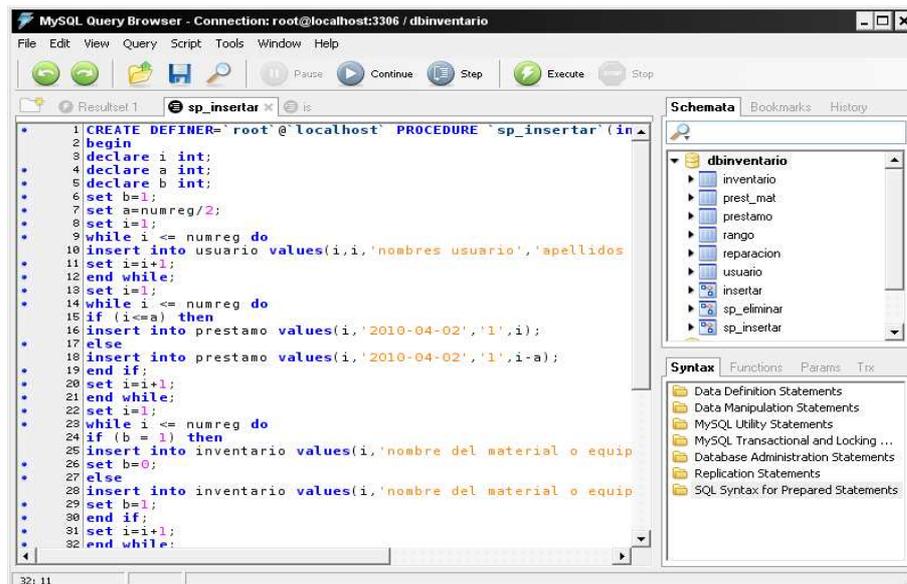
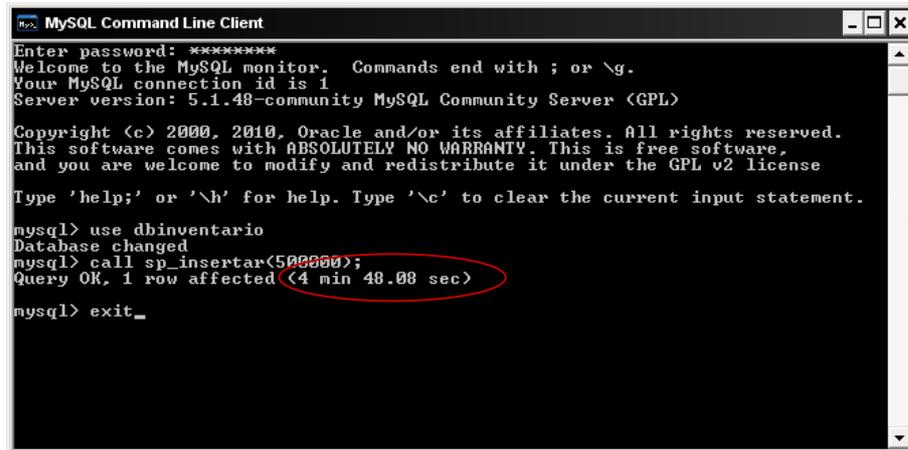


Figura III.16 Implementación de las operaciones en MySQL

Para obtener el tiempo de respuesta en las pruebas de inserción se utiliza la herramienta MySQL Command Line Client (herramienta de línea de comandos que viene incluido en el paquete de instalación de MySQL y que permite ejecutar sentencias SQL), la misma que proporciona el tiempo de ejecución que toma el servidor en realizar el proceso.



```
MySQL Command Line Client
Enter password: *****
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 1
Server version: 5.1.48-community MySQL Community Server (GPL)

Copyright (c) 2000, 2010, Oracle and/or its affiliates. All rights reserved.
This software comes with ABSOLUTELY NO WARRANTY. This is free software,
and you are welcome to modify and redistribute it under the GPL v2 license

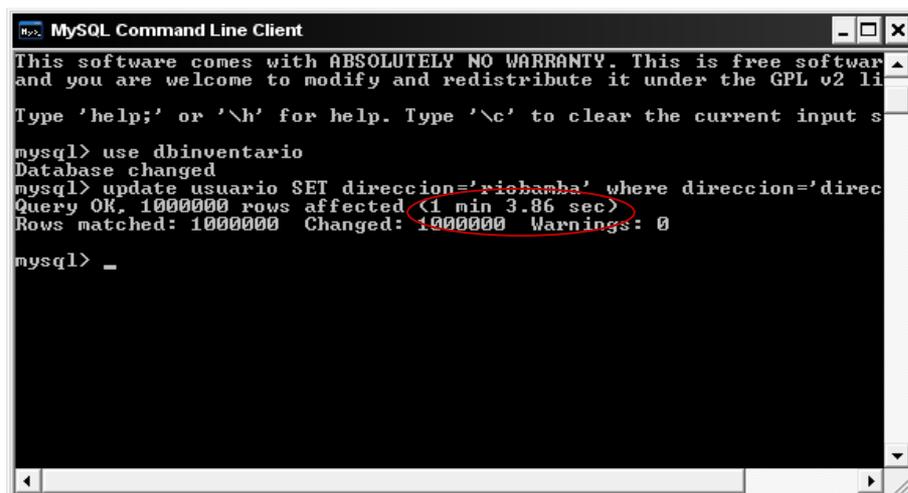
Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> use dbinventario
Database changed
mysql> call sp_insertar(500000);
Query OK, 1 row affected (4 min 48.08 sec)

mysql> exit_
```

**Figura III.17** MySQL Command Line Client – Inserción

Igualmente se utilizará la herramienta MySQL Command Line Client para obtener los tiempos de respuesta en realizar las pruebas de actualización y eliminación.



```
MySQL Command Line Client
This software comes with ABSOLUTELY NO WARRANTY. This is free software
and you are welcome to modify and redistribute it under the GPL v2 li

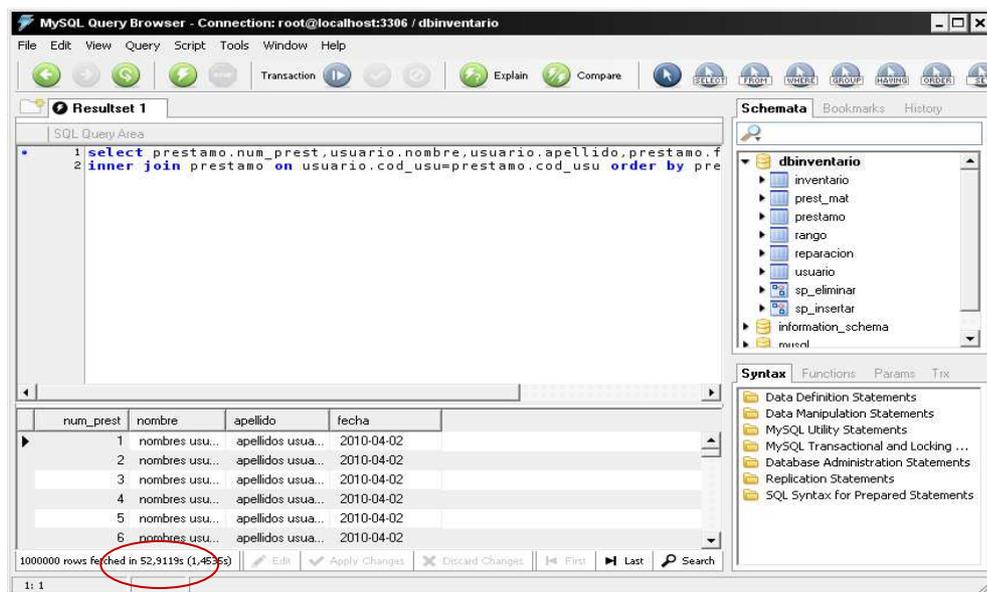
Type 'help;' or '\h' for help. Type '\c' to clear the current input s

mysql> use dbinventario
Database changed
mysql> update usuario SET direccion='risbamba' where direccion='dirc
Query OK, 1000000 rows affected (1 min 3.86 sec)
Rows matched: 1000000  Changed: 1000000  Warnings: 0

mysql> _
```

**Figura III.18** MySQL Command Line Client – Actualización

Para obtener el tiempo de respuesta en las pruebas de selección se utiliza la herramienta MySQL Query Browser (utilidad de MySQL GUI Tools que permite crear y ejecutar consultas MySQL en un entorno gráfico), la misma que proporciona el tiempo que demora en realizar una consulta de MySQL.



**Figura III.19** MySQL Query Browser

### 3.8.2. Firebird

Para obtener los tiempos de respuesta de las pruebas que se realizarán al SGBD Firebird, se utilizará la herramienta IBExpert la misma que proporciona el tiempo de ejecución que toma el servidor en realizar las operaciones a ejecutar.

En IBExpert se diseñan las operaciones que deberán ser evaluadas.

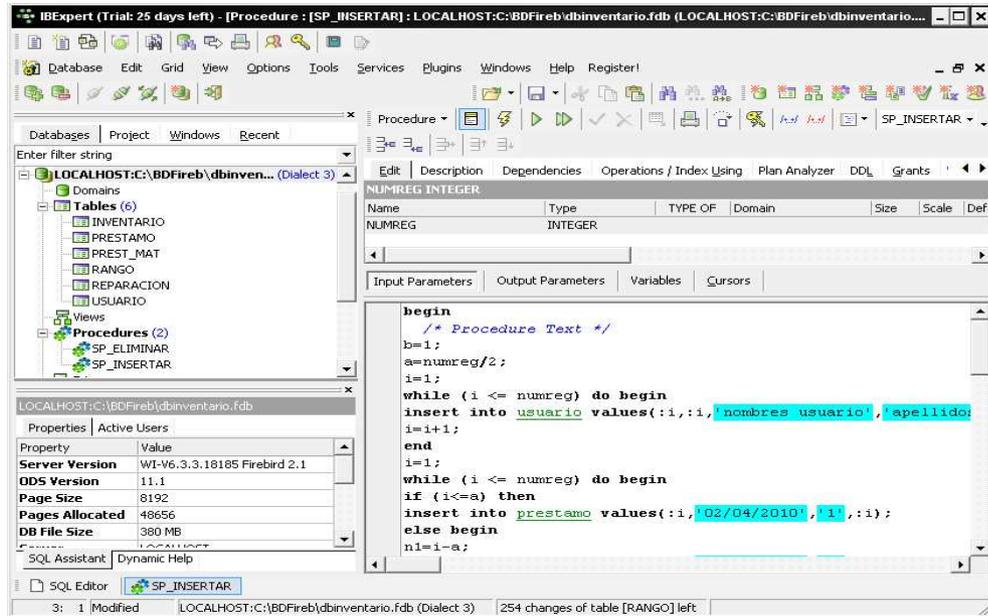


Figura III.20 Implementación de las operaciones en Firebird

Una vez que se han implementado los procesos necesarios, se procede a ejecutarlos. Para las pruebas de inserción se ejecuta el procedimiento almacenado y al término de la ejecución IBExpert presenta en la parte inferior el tiempo en el cual realizó la inserción. De igual manera se procederá para la realización de las pruebas de eliminación.

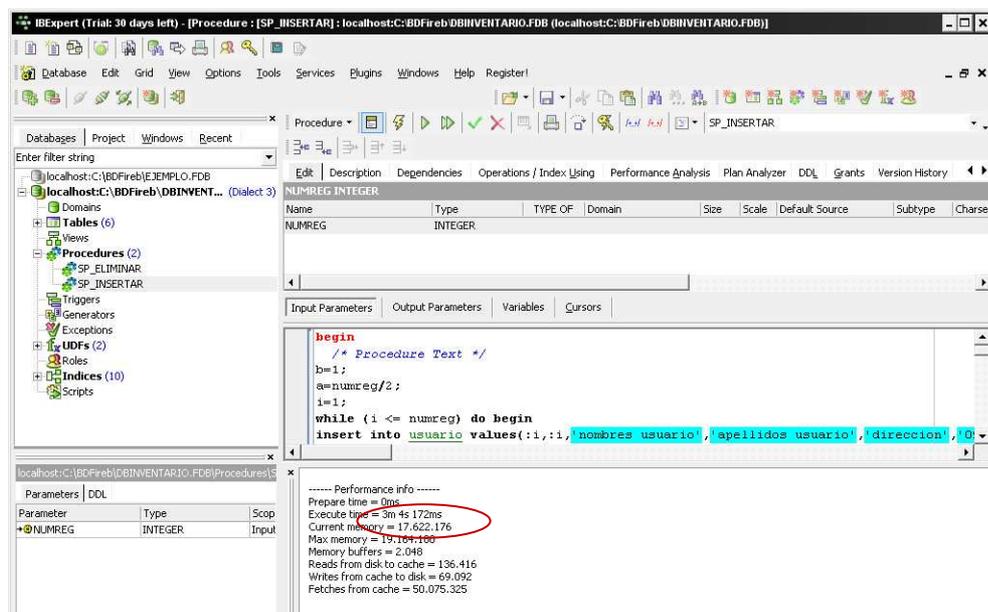


Figura III.21 IBExpert – Inserción

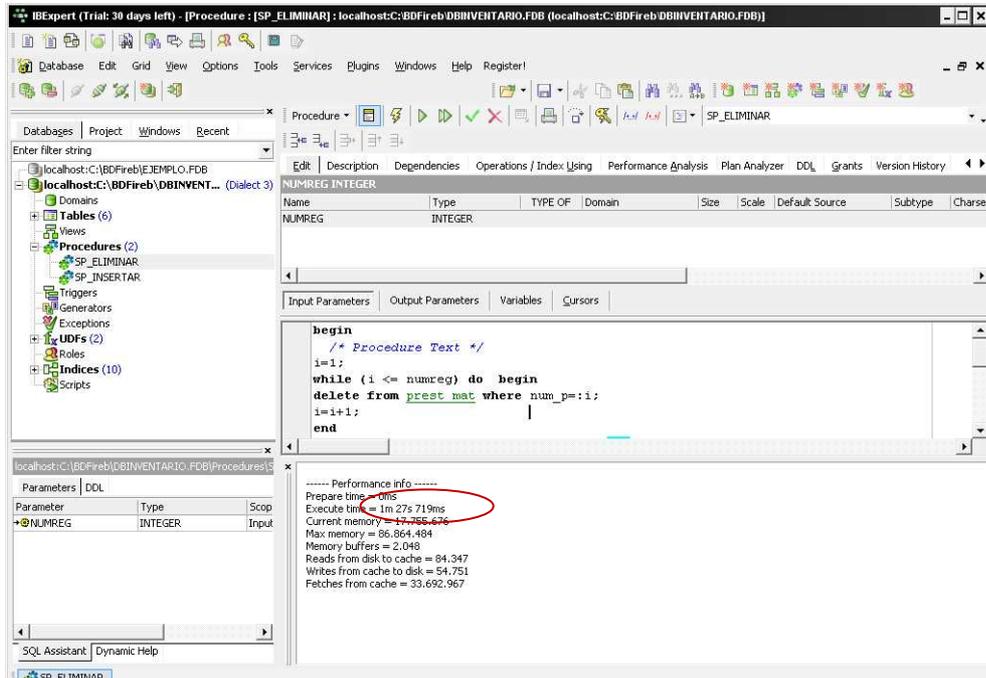


Figura III.22 IBExpert - Eliminación

Para las pruebas de selección abrir un SQL Editor en IBExpert (menú Tools/SQL Editor), se implementa la consulta a realizar y se ejecuta, al término de la ejecución IBExpert muestra el tiempo en el cual realizó el proceso. De igual forma se procede para realizar las pruebas de actualización de datos.

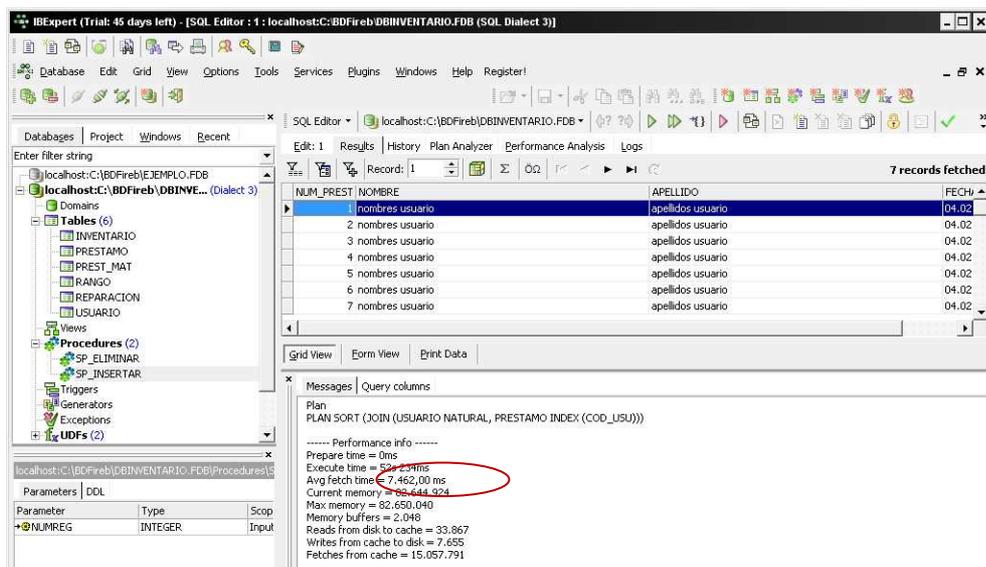


Figura III.23 IBExpert – Selección

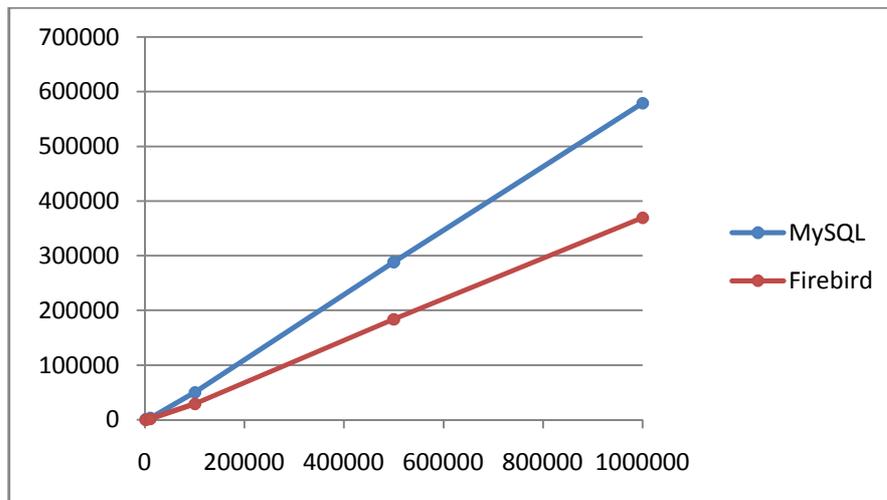
### 3.9. Resultados de las pruebas

A continuación se presenta los resultados obtenidos después de concluir las pruebas satisfactoriamente a cada uno de los SGBD.

#### 3.9.1. Comparación del tiempo de ejecución en la inserción de datos

**Tabla III.XIX.** Tiempo de respuesta en la inserción de datos

Inserción (milisegundos)		
Número de registros	MySQL	Firebird
1.000	520	266
10.000	2720	1765
100.000	50480	29156
500.000	288080	184172
1.000.000	579160	369734
<b>Promedio</b>	184192	117019



**Figura III.24** Tiempo de respuesta en la inserción de datos

En ambos SGBD los tiempos de respuesta van creciendo de acuerdo al número de registros insertados. Pero Firebird presenta una gran ventaja en realizar la inserción, al ofrecer el menor tiempo de respuesta en realizar las pruebas.

### 3.9.2. Consultas de diversa complejidad

Resultados obtenidos en las pruebas de consultas de diversa complejidad, luego de haberlas realizado satisfactoriamente a MySQL y Firebird.

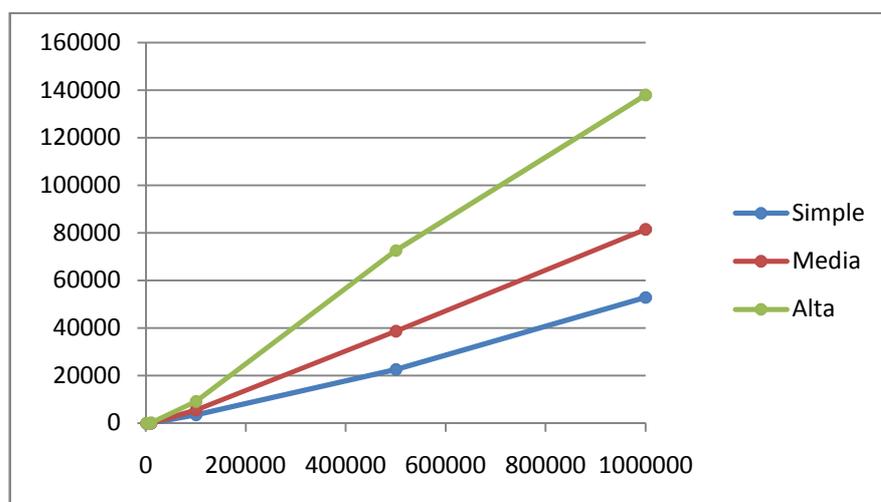
#### 3.9.2.1. Resultados del tiempo de respuesta en consultas de diversa complejidad

##### MySQL

La tabla III.XX muestra los tiempos de respuesta obtenidos durante las pruebas en consultas con diferentes grados de complejidad y con diferentes cantidades de registros en MySQL.

**Tabla III.XX.** Tiempo de respuesta en consultas de diversa complejidad MySQL

MySQL – Consultas de diversa complejidad (milisegundos)			
Número de registros	Simple	Media	Alta
1.000	4	6	12
10.000	38	54	122
100.000	3515	5475	9092
500.000	22511	38608	72578
1.000.000	52911	81404	137901



**Figura III.25** Tiempo de respuesta en consultas de diversa complejidad MySQL

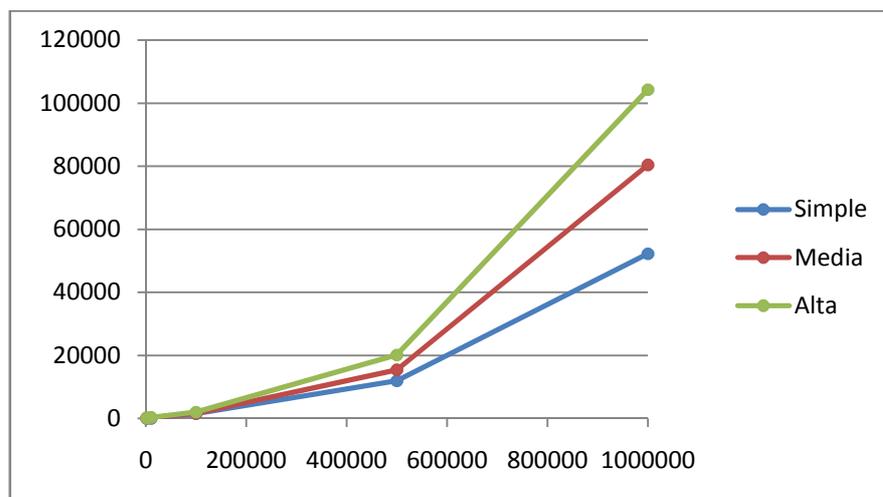
De acuerdo a los datos obtenidos, se puede observar que el tiempo de respuesta en consultas simples, de complejidad media y alta tienen una tendencia creciente de acuerdo al número de registros y de acuerdo a la complejidad de la consulta.

### 3.9.2.2. Resultados del tiempo de respuesta en consultas de diversa complejidad Firebird

En la tabla III.XXI se presentan los tiempos de respuesta obtenidos en la ejecución de las consultas con diferentes grados de complejidad y con diferentes cantidades de registros en Firebird.

**Tabla III.XXI.** Tiempo de respuesta en consultas de diversa complejidad Firebird

Firebird – Consultas de diversa complejidad (milisegundos)			
Número de registros	Simple	Media	Alta
1.000	16	78	96
10.000	63	269	281
100.000	1516	1610	2031
500.000	11968	15359	20141
1.000.000	52234	80352	104296



**Figura III.26** Tiempo de respuesta en consultas de diversa complejidad Firebird

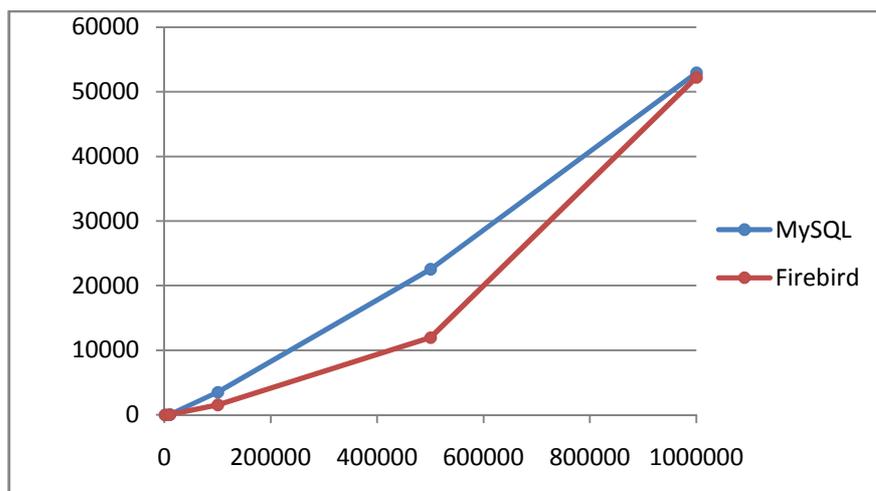
Firebird al igual que MySQL tiene un tiempo de respuesta en las consultas de diversa complejidad (simple, media y alta) que va creciendo de acuerdo al número de registros y a la complejidad de la consulta.

### 3.9.3. Comparación del tiempo de respuesta en consultas simples

La tabla III.XXII muestra los tiempos de respuesta obtenidos durante las pruebas de consultas simples a los SGBD MySQL y Firebird, y con diferentes cantidades de registros.

**Tabla III.XXII.** Tiempo de respuesta en consultas simples

Consultas simples (milisegundos)		
Número de registros	MySQL	Firebird
1.000	4	16
10.000	38	63
100.000	3515	1516
500.000	22511	11968
1.000.000	52911	52234
<b>Promedio</b>	15796	13159



**Figura III.27** Tiempo de respuesta en consultas simples

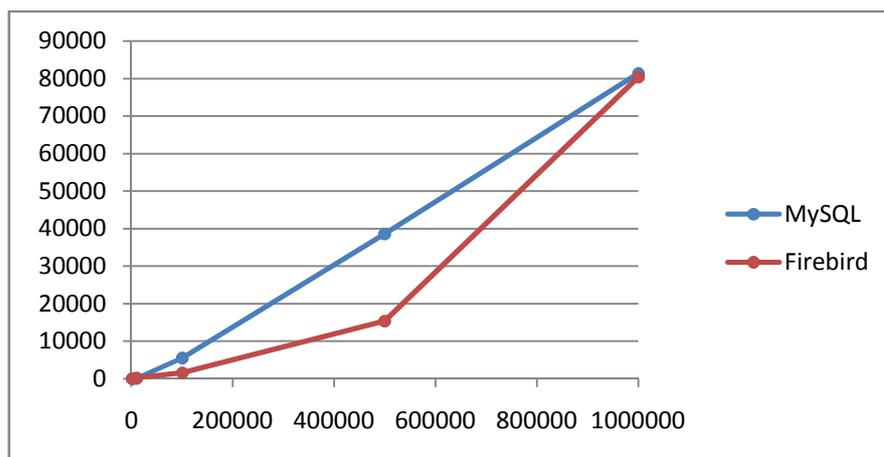
Ambos SGBD presentan un tiempo de respuesta que va creciendo de acuerdo a la cantidad de registros consultados. Firebird en la extracción de 1.000 y 10.000 filas presenta mayor tiempo de respuesta que MySQL pero a partir de 100.000 empieza a tener mejor desempeño, aunque en consultas de grandes cantidades de registros va casi al mismo nivel que MySQL, lo que lo posiciona a Firebird como el mejor en realizar las pruebas de consultas simples, al ofrecer el menor tiempo de respuesta en la mayoría de escenarios en las que se realizó las pruebas.

#### 3.9.4. Comparación del tiempo de respuesta en consultas de complejidad media

En la III.XXIII se muestran los tiempos de respuesta obtenidos durante las pruebas al ejecutar consultas de mediana complejidad en los SGBD MySQL y Firebird, y con diferentes cantidades de registros.

**Tabla III.XXIII.** Tiempo de respuesta en consultas de complejidad media

Consulta de complejidad media (milisegundos)		
Número de registros	MySQL	Firebird
1.000	6	78
10.000	54	269
100.000	5475	1610
500.000	38608	15359
1.000.000	81404	80352
<b>Promedio</b>	25109	19534



**Figura III.28** Tiempo de respuesta en consultas de complejidad media

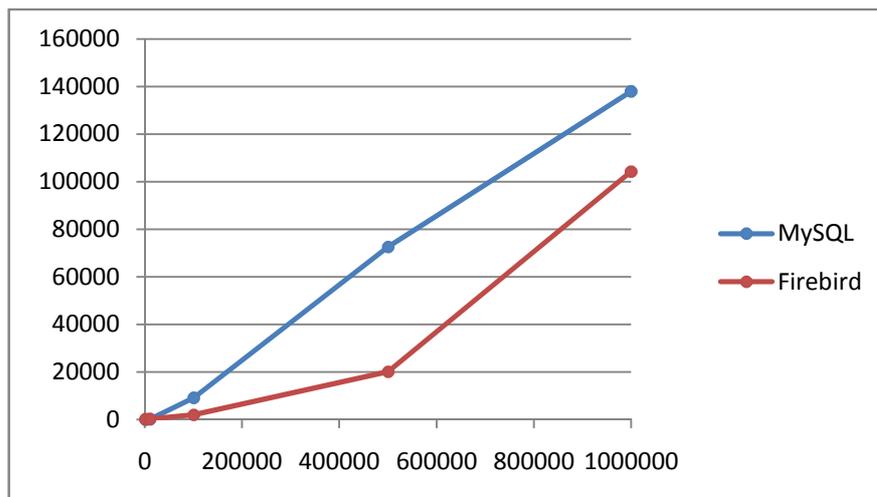
En MySQL y Firebird los tiempos de respuesta van creciendo de acuerdo a la cantidad de registros consultados. Y al igual que en las consultas simples Firebird presenta mejor desempeño al realizar la consulta de complejidad media, al tener el menor tiempo de respuesta en realizar las consultas en la mayoría de escenarios propuestos.

### 3.9.5. Comparación del tiempo de respuesta en consultas de complejidad alta

La tabla III.XXIV presenta los tiempos de respuesta obtenidos durante la ejecución de las pruebas de selección de complejidad alta en los SGBD MySQL y Firebird, y con diferentes cantidades de registros.

**Tabla III.XXIV.** Tiempo de respuesta en consultas de complejidad alta

Consultas de complejidad alta (milisegundos)		
Número de registros	MySQL	Firebird
1.000	12	96
10.000	122	281
100.000	9092	2031
500.000	72578	20141
1.000.000	137901	104296
<b>Promedio</b>	43941	25369



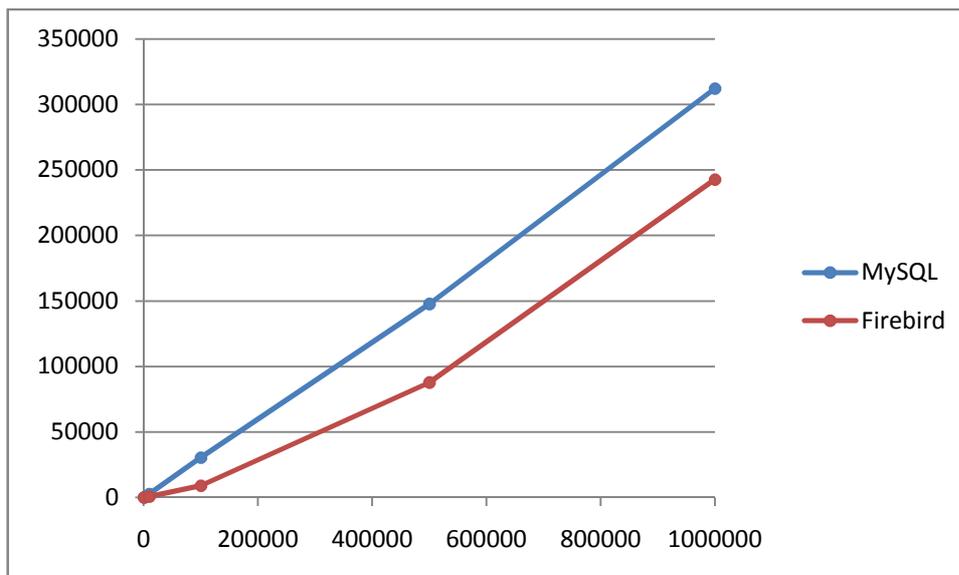
**Figura III.29** Tiempo de respuesta en consultas de complejidad alta

En los dos SGBD los tiempos de respuesta van creciendo de acuerdo al número de registros consultados. Firebird al igual que en las consultas de complejidad simple y media, presenta una gran ventaja al realizar la consulta de complejidad alta, ofreciendo el menor tiempo de respuesta en realizar las pruebas.

### 3.9.6. Comparación del tiempo de respuesta en la eliminación

**Tabla III.XXV.** Tiempo de respuesta en la eliminación

Eliminación (milisegundos)		
Número de registros	MySQL	Firebird
1.000	330	85
10.000	2720	859
100.000	30530	9204
500.000	147580	87729
1.000.000	312170	242781
<b>Promedio</b>	98666	68132



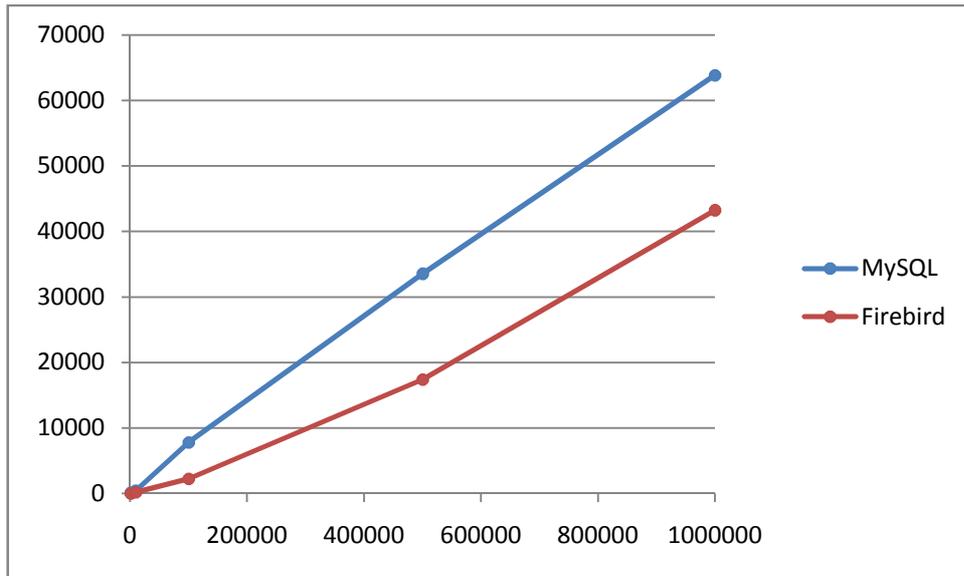
**Figura III.30** Tiempo de respuesta para la eliminación

En MySQL y Firebird los tiempos de respuesta van creciendo de acuerdo al número de registros eliminados. Pero Firebird ofrece el menor tiempo de respuesta en realizar las pruebas, por lo que presenta una gran ventaja en comparación con MySQL.

### 3.9.7. Comparación del tiempo de respuesta para la actualización

**Tabla III.XXVI.** Tiempo de respuesta en la actualización

Actualización (milisegundos)		
Número de registros	MySQL	Firebird
1.000	90	15
10.000	450	234
100.000	7810	2266
500.000	33520	17407
1.000.000	63860	43219
<b>Promedio</b>	21146	12628



**Figura III.31** Tiempo de respuesta en la actualización

En ambos SGBD los tiempos de respuesta van creciendo de acuerdo al número de registros actualizados. Pero Firebird ofrece el menor tiempo de respuesta en realizar las pruebas, por lo que presenta una gran ventaja en comparación con MySQL.

### **3.9.8. Prueba de interrupción eléctrica**

Después de haber realizado la prueba de interrupción eléctrica especificado en el diseño de pruebas, se verificó el estado de la información en ambos SGBD, y se encontró toda la información intacta, como si no se hubiera suspendido abruptamente el servicio de energía y como si la sentencia sql nunca se hubiera actualizado. La integridad de los datos se conservó. Por lo tanto a MySQL y Firebird se los calificará con el valor máximo que es 10.

### 3.10. Modelo de puntuación

Luego de haber realizado las respectivas pruebas se procede a evaluar cada una de ellas, para lo cual se va a calificar con un peso de 10/10 al SGBD que presente mejor desempeño (menor tiempo de respuesta) en realizar las pruebas, para la calificación del otro SGBD se lo va a realizar mediante una regla de proporcionalidad. En cuanto a la integridad de los datos en las pruebas realizadas a MySQL y Firebird no se tuvieron ningún problema por lo tanto a ambos SGBD se los va a calificar con un valor de 10.

Para seleccionar el mejor SGBD de acuerdo a las pruebas realizadas se lo hizo en base a una comparación cuali-cuantitativa definida bajo nuestro propio criterio, en base a la importancia de los parámetros a evaluar.

**Tabla III.XXVII.** Parámetros con su respectivo valor porcentual

<b>Parámetros</b>	<b>Valor porcentual</b>
Tiempo de ejecución	40%
Integridad en los datos	40%
Tolerancia a fallos	20%
<b>Total</b>	<b>100%</b>

Con los valores porcentuales definidos para MySQL y Firebird se obtendrá el SGBD que ofrezca mejor desempeño en la ejecución de las pruebas y por lo tanto el que optimice uno de los recursos más importantes como es *el tiempo de ejecución* en realizar los procesos más habituales en los sistemas.

El parámetro tiempo de ejecución es importante al momento de determinar los recursos utilizados, es por eso que se realizará una comparación del tiempo de respuesta que presenta MySQL y Firebird en realizar las pruebas, en donde el mejor tiempo será el menor tiempo de respuesta y por lo tanto el que optimice más este recurso. Para la evaluación del tiempo de ejecución se lo va a realizar de acuerdo a la tabla III.XXVII.

**Tabla III.XXVIII.** Escala cuali-cuantitativa para el tiempo de ejecución

	<b>Peso</b>	<b>Equivalencia</b>
t	10	Excelente
t + 20%t	8	Muy bueno
t + 60%t	6	Bueno
t +100 %t	4	Regular
> t +100 %t	2	Malo

Dónde: t = menor tiempo de respuesta

### 3.11. Resultados finales y análisis de los datos obtenidos

**Tabla III.XXIX.** Resultados del tiempo de ejecución

<b>Pruebas</b>	<b>MySQL Tiempo de respuesta (milisegundos)</b>	<b>Firebird Tiempo de respuesta (milisegundos)</b>
Inserción	184192	117019
Consulta simple	15796	13159
Consulta de complejidad media	25109	19534
Consulta de complejidad alta	43941	25369
Eliminación	98666	68132
Actualización	21146	12628
<b>P. Total</b>	<b>64808</b>	<b>42640</b>

**Tabla III.XXX.** Evaluación del parámetro tiempo de ejecución

<b>Pruebas</b>	<b>MySQL (Peso)</b>	<b>Firebird (Peso)</b>
Inserción	6	10
Consulta simple	8	10
Consulta de complejidad media	6	10
Consulta de complejidad alta	4	10
Eliminación	6	10
Actualización	4	10
<b>P. Total</b>	<b>5.67</b>	<b>10</b>

**Tabla III.XXXI.** Evaluación del parámetro integridad en los datos

<b>Pruebas</b>	<b>MySQL (Peso)</b>	<b>Firebird (Peso)</b>
Inserción	10	10
Consulta simple	10	10
Consulta de complejidad media	10	10
Consulta de complejidad alta	10	10
Prueba de interrupción eléctrica	10	10
<b>P. Total</b>	<b>10</b>	<b>10</b>

**Tabla III.XXXII.** Evaluación del parámetro tolerancia a fallos

<b>Pruebas</b>	<b>MySQL (Peso)</b>	<b>Firebird (Peso)</b>
Prueba de interrupción eléctrica	10	10
<b>P. Total</b>	<b>10</b>	<b>10</b>

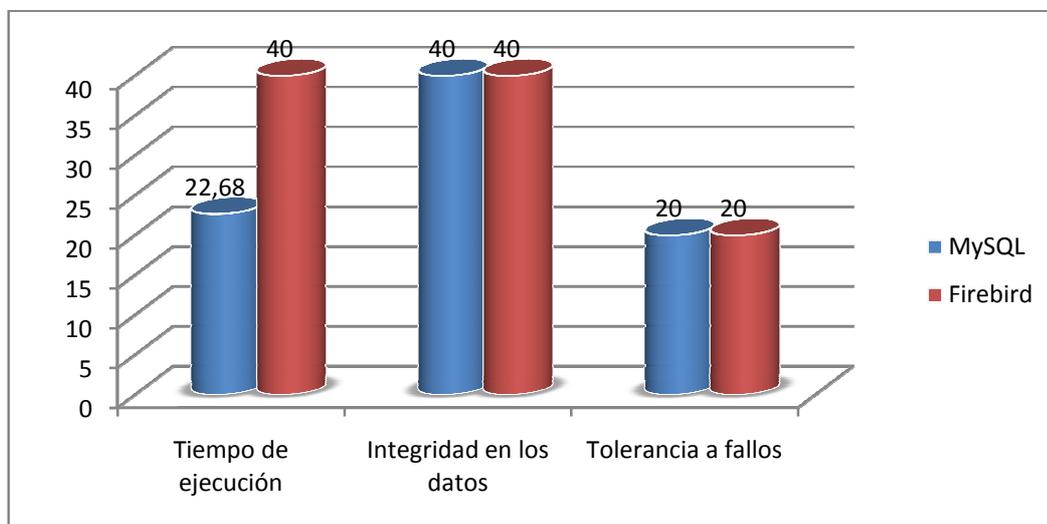
**Tabla III.XXXIII.** Evaluación de los parámetros

	<b>Tiempo de ejecución</b>		<b>Integridad en los datos</b>		<b>Tolerancia a fallos</b>	
	<b>Peso</b>	<b>Equivalencia</b>	<b>Peso</b>	<b>Equivalencia</b>	<b>Peso</b>	<b>Criterio</b>
MySQL	5.67	Bueno	10	Excelente	10	Excelente
Firebird	10	Excelente	10	Excelente	10	Excelente

La tabla III.XXXIV muestra los resultados finales de acuerdo a los valores porcentuales que se le dio a cada uno de los parámetros a evaluar y según el peso que hayan obtenido.

**Tabla III.XXXIV.** Sumatoria de los puntos obtenidos por MySQL y Firebird

<b>Resultados finales</b>		
	MySQL	Firebird
Tiempo de ejecución	22.68	40
Integridad en los datos	40	40
Tolerancia a fallos	20	20
<b>Total</b>	<b>82.68</b>	<b>100</b>



**Figura III.32** Resultados finales por parámetros

### **Análisis de resultados**

- De acuerdo a los datos obtenidos se puede concluir que el SGBD Firebird ofrece un menor tiempo de respuesta en la ejecución de las operaciones ya sean de inserción, selección, eliminación o actualización comparado con MySQL.
- En las pruebas de selección simple y media MySQL mejora el tiempo de respuesta en realizar los procesos, pero no es lo suficiente como para llegar al mismo nivel que Firebird.
- En las consultas de diversa complejidad (simple, media y alta) Firebird al realizar extracción de datos de grandes cantidades de registros empieza a realizarlas casi al mismo nivel que MySQL esto se lo puede observar en la selección de bases de datos con 1.000.000 de registros.
- En las pruebas de eliminación y actualización Firebird siempre presenta mayor ventaja que MySQL, ofreciendo un menor tiempo de respuesta en realizar los procesos en bases de datos que manejen cualquier volumen de información.
- MySQL y Firebird en cuanto a la integridad de los datos en realizar las operaciones de inserción y selección quedan por igual, pues ninguno de ellos presenta alguna inconsistencia en los datos insertados o consultados.
- Los dos SGBD en la prueba de interrupción eléctrica no sufrieron ninguna alteración en sus datos por lo que MySQL y Firebird se los calificó con el mejor puntaje.
- Firebird obtuvo una calificación de 100 puntos, obteniendo el mayor puntaje y por ende siendo mejor SGBD que MySQL.

### 3.12. Principales características comparadas de MySQL y Firebird

Además de las pruebas realizadas, se definirán y tomarán en cuenta características importantes tales como son: posee soporte para procedimientos almacenados, integridad referencial, vistas, transacciones, triggers, interfaces y conectores que soportan, dimensionamiento, entre otros. Además de otros criterios tales como documentación, soporte, licenciamiento, herramientas administrativas. Debido a que son características fundamentales que debe poseer un buen SGBD se ha visto importante tomarlos en cuenta para el estudio comparativo, ya que estas características influyen en el desempeño y permiten obtener un mejor resultado en el rendimiento del SGBD. Se tendrá en cuenta la información recopilada en los capítulos anteriores sobre MySQL y Firebird.

Para la calificación de estas características de los SGBD, se procederá de la siguiente manera. Cada categoría a calificar tiene un número de ítems que se va a valorar de la siguiente manera: 1 pto para el que posea la característica y 0 pto de lo contrario. A cada SGBD se le asignará un peso sobre 10, de acuerdo al número de ítems con los que cumpla en la categoría que se estará evaluando.

#### 3.12.1. Comparativa general

La siguiente tabla muestra una comparación general de las bases de datos.

**Tabla III.XXXV.** Comparativa general de las bases de datos

	<b>MySQL</b>	<b>Firebird</b>
Nombre de la Empresa	MySQL AB	IBPhoenix
1 <sup>er</sup> Release público	1998	2000
Tipo de licenciamiento	GPL o Propietario	IPL

### 3.12.2. Licenciamiento

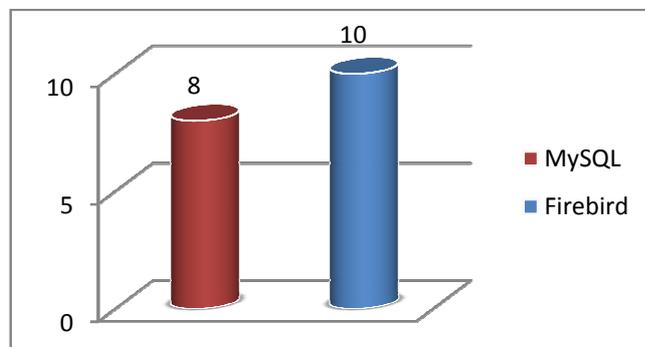
La licencia GPL de MySQL obliga a que la distribución de cualquier producto derivado (aplicación) se haga bajo esa misma licencia. Si un desarrollador desea incorporar MySQL en su producto pero desea distribuirlo bajo otra licencia que no sea la GPL, puede adquirir una licencia comercial de MySQL que le permite hacer justamente eso.

Firebird es open source y no hay licencias duales, tanto para uso comercial como para aplicaciones open source, es totalmente libre.

Por lo tanto de acuerdo a nuestro criterio se va a calificar el licenciamiento de Firebird con un valor de 10 ya que es completamente libre, y a MySQL con un valor de 8 porque presenta cierta restricción al distribuir las aplicaciones de lo contrario se debería adquirir una licencia comercial.

**Tabla III.XXXVI. Peso de Licenciamiento**

LICENCIAMIENTO	
	Peso
MySQL	8
Firebird	10



**Figura III.33 Licenciamiento**

### 3.12.3. Características generales

#### 3.12.3.1. Sistemas Operativos soportados

**Tabla III.XXXVII.** Sistemas Operativos soportados

<b>SISTEMAS OPERATIVOS SOPORTADOS</b>				
	<b>MySQL</b>	<b>Puntaje</b>	<b>Firebird</b>	<b>Puntaje</b>
Windows	Si soporta	1	Si soporta	1
Linux	Si soporta	1	Si soporta	1
Mac OS	Si soporta	1	Si soporta	1
Free BSD	Si soporta	1	Si soporta	1
IBM AIX	Si soporta	1	Si soporta	1
Solaris	Si soporta	1	Si soporta	1
HP-UX	Si soporta	1	Si soporta	1
Otros	Si soporta	1	No	0

**Tabla III.XXXVIII.** Peso de Sistemas Operativos soportados

<b>SISTEMAS OPERATIVOS SOPORTADOS</b>		
	<b>Puntaje</b>	<b>Peso</b>
<b>MySQL</b>	8	10
<b>Firebird</b>	7	8.75

MySQL soporta más sistemas operativos obteniendo el mayor puntaje, Firebird obtiene menor puntaje ya que soporta menos sistemas operativos entre los principales y más utilizados por los desarrolladores.

#### 3.12.3.2. Interfaces (API'S) / Conectores soportados

Uno de los principales objetivos de un SGBD es proporcionar un sistema de almacenamiento y consulta de datos al que se puedan conectar las aplicaciones que desarrollemos. Es por ello que es muy necesario que existan varias formas de acceder a nuestras bases de datos.

**Tabla III.XXXIX.** Interfaces y conectores soportados

<b>INTERFACES Y CONECTORES SOPORTADOS</b>				
	<b>MySQL</b>	<b>Puntaje</b>	<b>Firebird</b>	<b>Puntaje</b>
Interfaces				
C	Si soporta	1	Si soporta	1
C++	Si soporta	1	Si soporta	1
Delphi	Si soporta	1	Si soporta	1
Perl	Si soporta	1	Si soporta	1
Python	Si soporta	1	Si soporta	1
Php	Si soporta	1	Si soporta	1
Java	Si soporta	1	Si soporta	1
Ruby	Si soporta	1	Si soporta	1
Conectores				
.Net	Si soporta	1	Si soporta	1
ODBC	Si soporta	1	Si soporta	1
JDBC	Si soporta	1	Si soporta	1
OLEDB	Si soporta	1	Si soporta	1

**Tabla III.XL.** Peso de Interfaces y Conectores soportados

<b>INTERFACES Y CONECTORES SOPORTADOS</b>		
	<b>Puntaje</b>	<b>Peso</b>
<b>MySQL</b>	12	10
<b>Firebird</b>	12	10

### 3.12.3.3. Características fundamentales

Son las características que debe cumplir un buen sistema gestor de base de datos, por lo que el incumplimiento con alguna de ellas haría que estos sistemas no cumplan con su verdadero objetivo.

**Tabla III.XLI.** Características fundamentales

<b>CARACTERÍSTICAS FUNDAMENTALES</b>				
	<b>MySQL</b>	<b>Puntaje</b>	<b>Firebird</b>	<b>Puntaje</b>
Transacciones	Si soporta	1	Si soporta	1
ACID	Si soporta	1	Si soporta La implementación de transacciones de Firebird cumple con todas estas propiedades.	1
Integridad referencial	Si soporta	1	Si soporta	1

**Tabla III.XLII.** Peso de características fundamentales

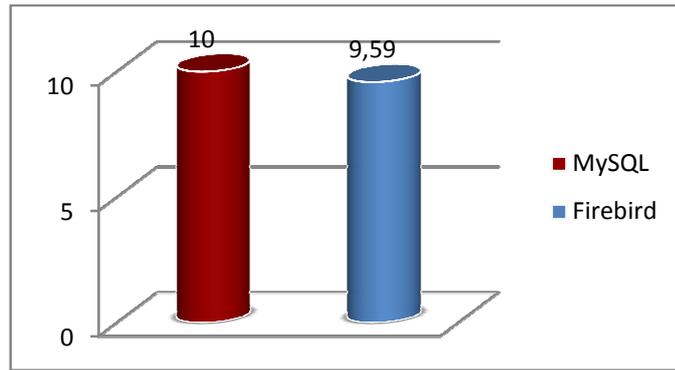
<b>CARACTERÍSTICAS FUNDAMENTALES</b>		
	<b>Puntaje</b>	<b>Peso</b>
<b>MySQL</b>	3	10
<b>Firebird</b>	3	10

MySQL y Firebird obtienen el mismo puntaje debido a que poseen todas las características básicas que un SGBD debe presentar.

### 3.12.4. Comparativa de características generales

**Tabla III.XLIII.** Características generales

<b>CARACTERÍSTICAS GENERALES</b>		
	<b>MySQL</b>	<b>Firebird</b>
Sistemas Operativos soportados	10	8.75
Interfaces (api's) / conectores soportados	10	10
Características fundamentales	10	10
<b>P. Total</b>	10	9.59



**Figura III.34** Características generales

### 3.12.5. Límites .

Los límites de almacenamiento ya sean en la base de datos, tablas, registros, campos son de gran importancia porque indican que un SGBD proporciona mayor capacidad para manejar el volumen de información que se requiera.

En este punto se va a evaluar las características de la siguiente manera: Ilimitado: 10, Limitado: Se asignará el valor de 10 al que posea mayor capacidad y dependiendo de la capacidad del otro se le asignará un valor en comparación con el de mayor capacidad.

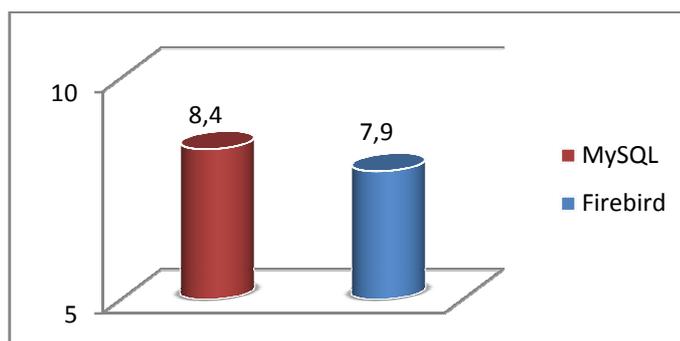
**Tabla III.XLIV.** Límites

LÍMITES				
	Tamaño máximo de una base de datos	Tamaño máximo de una tabla	Tamaño máximo de un registro	Tamaño máximo de un campo
MySQL	Ilimitado	Depende del sistema de archivos (teórico 64 Tb)	64 Kb (Los tipos de datos largos consumen 8 bytes de cabecera y se almacenan por separado)	4 Gb
Firebird	Ilimitado	32 Tb	64 Kb (Los tipos de datos largos consumen 8 bytes de cabecera y se almacenan por separado)	32 Gb

LÍMITES				
	Máximo de registros por tabla	Máximo de campos por tabla	Máximo de índices por tabla	Número de tablas en una BD
MySQL	Depende del sistema de archivos (teórico 64 Tb)	Dependerá de los tipos de datos seleccionados y hasta completar los 64 Kb	Mayor a 64 (no especifica el límite)	Ilimitado
Firebird	>16 Billion	Dependerá de los tipos de datos seleccionados y hasta completar los 64 Kb	65535	64535

**Tabla III.XLV. Peso de Límites**

LÍMITES		
	MySQL	Firebird
Tamaño máximo de una base de datos	10	10
Tamaño máximo de una tabla	10	5
Tamaño máximo de un registro	10	10
Tamaño máximo de un campo	1.25	10
Máximo de registros por tabla	6	8
Máximo de campos por tabla	10	10
Máximo de índices por tabla	10	5
Número de tablas en una BD	10	5
<b>Total</b>	<b>8.4</b>	<b>7.9</b>



**Figura III.35 Límites**

### 3.12.6. Características técnicas

Es muy importante tener en cuenta antes de seleccionar un SGBD, que características técnicas posee, ya que estas facilitarán el desarrollo y administración de las bases de datos. Cada SGBD soporta estas características, pero no todas, lo que los pone en desventaja en comparación con los otros. A continuación se mencionan las más importantes:

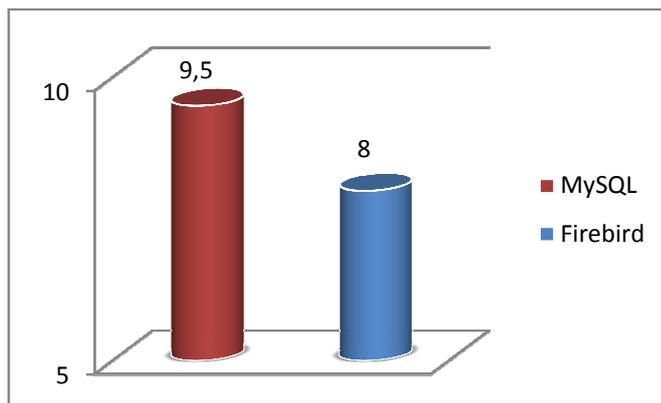
**Tabla III.XLVI.** Características técnicas

<b>CARACTERÍSTICAS TÉCNICAS</b>				
	<b>MySQL</b>	<b>Peso</b>	<b>Firebird</b>	<b>Peso</b>
Vistas	Si soporta	10	Si soporta	10
Triggers	Si soporta	10	Si soporta	10
Roles	Si soporta	10	Si soporta	10
Funciones	Si soporta	10	Si soporta	10
Store Procedures	Si soporta	10	Si soporta	10
Save Points	Si soporta	10	Si soporta	10
Indexado	Si soporta	10	Si soporta	10
Tablespaces	Simulado	5	No soporta	0
Particionamiento de tablas	Si soporta	10	No soporta	0
Motor embebido	Si soporta	10	Si soporta	10
<b>P. Total</b>		9.5		8

En este punto se calificó las características de la siguiente manera: Si posee la característica: 10, Si no posee: 0, Simulado: 5. Al final se obtuvo el valor promedio.

**Tabla III.XLVII.** Peso Características técnicas

<b>CARACTERÍSTICAS TÉCNICAS</b>	
	<b>Peso</b>
<b>MySQL</b>	9.5
<b>Firebird</b>	8



**Figura III.36** Características técnicas

### 3.12.7. Comparativa de tipos de datos

Después del diseño de una base de datos, es necesario crear las tablas correspondientes dentro de la misma. Para cada campo de cada una de las tablas, es necesario determinar el tipo de datos que contiene, para de esa forma ajustar el diseño de la base de datos, y conseguir un almacenamiento óptimo con la menor utilización de espacio.

**Tabla III.XLVIII.** Peso Tipos de datos

TIPOS DE DATOS	
	Peso
MySQL	10
Firebird	7

MySQL soporta los tipos de datos Enum, Set, Datetime entre otros y Firebird no lo soporta. Por lo tanto de acuerdo a lo estudiado sobre los tipos de datos tanto de MySQL y Firebird se va a proceder a calificar a MySQL con un valor de 10 y a Firebird con 6.

### 3.12.8. Herramientas de administración

Otro punto para considerar cuando se escoge un sistema administrador de base de datos es que tan fácil es su administración, para ello se debe tener herramientas como por ejemplo para manipular los componentes del diccionario de datos (tablas, index, seguridad, etc.).

Ambos sistemas de base de datos poseen gran cantidad de herramientas ya sean estas propias o de terceros, gráficas o de consola que facilitan la administración de las base de datos. Existen herramientas ya sean para análisis de consultas, replicación, reparación, backup, etc.

**Tabla III.XLIX. Herramientas de administración**

<b>HERRAMIENTAS DE ADMINISTRACIÓN</b>				
	<b>MySQL</b>	<b>Puntaje</b>	<b>Firebird</b>	<b>Puntaje</b>
Consola	Si	1	Si	1
Administración por GUI	Si	1	Si	1
Herramientas de Respaldo y Restauración	Si	1	Si	1
Herramientas de Replicación	Si	1	Si	1
Herramientas de Reparación	Si	1	Si	1
Herramientas de Migración	Si	1	Si	1

**Tabla III.L. Peso de Herramientas de Administración**

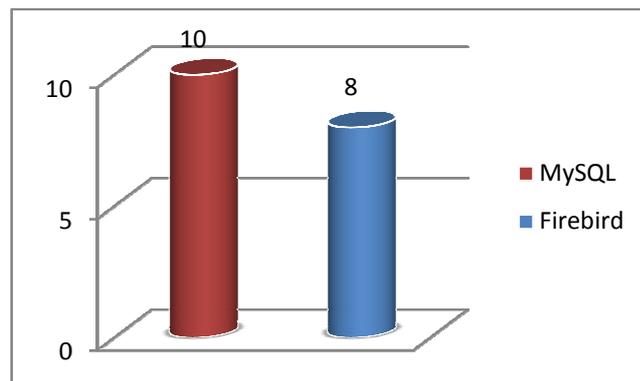
<b>HERRAMIENTAS DE ADMINISTRACIÓN</b>		
	<b>Puntaje</b>	<b>Peso</b>
<b>MySQL</b>	6	10
<b>Firebird</b>	6	10

### 3.12.9. Soporte

En cuanto al soporte técnico ofrecido para estos dos SGBD, se puede decir que MySQL presenta mayor ventaja que el de Firebird, ya que éste último al ser totalmente libre no tiene varias empresas que lo patrocinan porque no obtienen mayores ganancias. Por lo tanto se lo va a calificar a MySQL con un soporte aceptable calificándolo con un valor de 10, y a Firebird con un valor de 8.

**Tabla III.LI.** Peso soporte

	Peso
MySQL	10
Firebird	8



**Figura III.37** Soporte

MySQL ofrece soporte y servicios de sus propios desarrolladores asegurando así una respuesta confiable y segura.

### 3.12.10. Otros

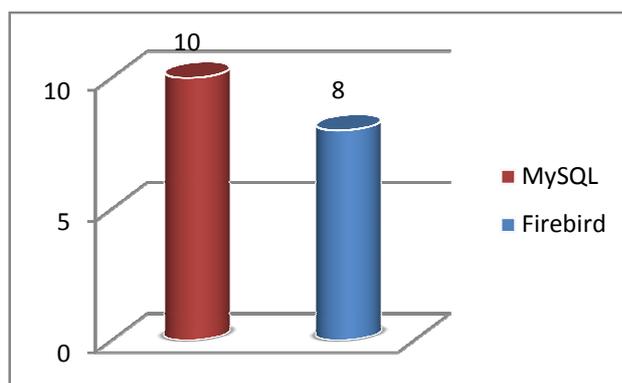
Además existen otros criterios que se deben tener en cuenta, como por ejemplo la documentación que es muy importante y que representa un gran apoyo para el manejo del SGBD.

**Tabla III.LII. Otros**

OTROS				
	MySQL	Puntaje	Firebird	Puntaje
Documentación	Alta	10	Escasa	6
Consumo de recursos	Bajo	10	Bajo	10

**Tabla III.LIII. Peso otros**

OTROS	
	Peso
MySQL	10
Firebird	8



**Figura III.38 Otros**

Como se observa, en éste punto MySQL presenta mayor ventaja en comparación con Firebird, esto debido a que Firebird posee poca documentación y sobre todo en español a diferencia de MySQL que presenta gran cantidad de documentación ya sea en libros, on line, etc.

### 3.13. Comparativa de características principales de MySQL y Firebird

**Tabla III.LIV.** Evaluación de las características principales

COMPARATIVA CARACTERÍSTICAS PRINCIPALES		
	MySQL	Firebird
Licenciamiento	8	10
Características generales	10	9.59
Límites	8.4	7.9
Características técnicas	9.5	8
Tipos de datos	10	7
Herramientas de administración	10	10
Soporte	10	8
Otros	10	8
<b>Total</b>	9.49	8.56

Para la evaluación final de las características que se han tomado en cuenta para determinar el mejor SGBD, se lo va a calificar sobre 100 puntos, de acuerdo al peso promedio obtenido.

**Tabla III.LV.** Evaluación final de las características principales

COMPARATIVA CARACTERÍSTICAS PRINCIPALES				
	MySQL	Porcentaje	Firebird	Porcentaje
Características principales	9.49	94.8	8.56	85.6

### 3.14. Comparación general de los SGBD MySQL y Firebird

A continuación se determina el mejor SGBD en base a las pruebas realizadas y a las características tomadas en cuenta en el apartado anterior, acorde a las calificaciones obtenidas. Debido a que tanto las características como el desempeño que presentan al realizar las pruebas son de gran importancia, se va a obtener un valor promedio entre estas dos para determinar el mejor SGBD.

**Tabla III.LVI.** Comparativa general

	<b>MySQL</b>	<b>Firebird</b>
Pruebas realizadas	82.68	100
Características principales	94.8	85.6
<b>P. Total</b>	<b>88.74</b>	<b>92.8</b>

### **3.15. Comprobación de la hipótesis**

#### **3.15.1. Hipótesis**

Con el estudio comparativo entre Sistemas de Gestión de Bases de Datos libres Firebird y MySQL en el desarrollo de aplicaciones web se podrá seleccionar el Sistema Gestor de Bases de Datos que permita la optimización de recursos en el desarrollo de la base de datos del Sistema para la Gestión de los Equipos de Radio del Escuadrón de Comunicaciones N° 11.

#### **3.15.2. Comprobación**

Una vez realizado el estudio de MySQL y Firebird, basado en la evaluación comparativa, se especifican a continuación los puntajes obtenidos para estos dos motores de base de datos, de tal forma que se obtenga como resultado el mejor.

En cuanto a las características de optimización de recursos de estos dos SGBD que se obtuvo principalmente con respecto al tiempo de respuesta en las pruebas que se realizaron se puede observar claramente en la tabla III.LVII el resultado obtenido para Firebird con un 100% y para MySQL con un 82.68%.

**Tabla III.LVII.** Resultados para los SGBD por parámetros

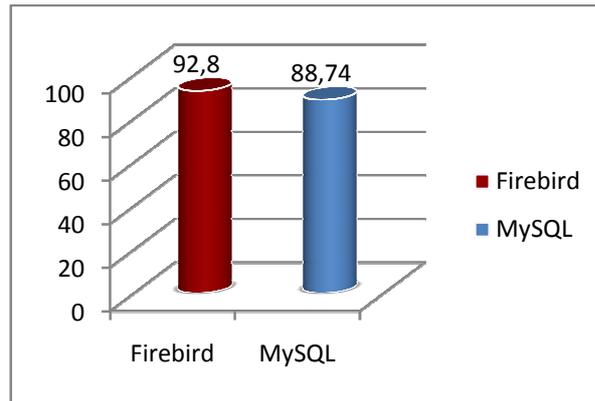
<b>Resultados finales</b>		
	<b>MySQL</b>	<b>Firebird</b>
Tiempo de ejecución	22.68	40
Integridad en los datos	40	40
Tolerancia a fallos	20	20
<b>Total</b>	<b>82.68</b>	<b>100</b>

Finalmente para determinar el mejor gestor de base de datos se lo ha realizado en base a las pruebas y las características que se han tomado en cuenta para el estudio comparativo, siendo superior Firebird ya que además de ser más rápido que MySQL al realizar las operaciones logrando optimizar tiempo, también permite un bajo consumo de recursos y un bajo costo debido a que posee una licencia totalmente libre.

**Tabla III.LVIII** Promedio total

	<b>SISTEMAS GESTORES DE BASE DE DATOS</b>	
	<b>MySQL</b>	<b>Firebird</b>
<b>P. Total</b>	88.74	92.8

La tabla III.LVIII y la figura.III.39 muestran claramente el resultado obtenido, teniendo como ganador a Firebird en comparación con MySQL.



**Figura III.39** Promedio total

### 3.15.3. Conclusiones del estudio comparativo

- De acuerdo a los datos obtenidos en las diferentes pruebas y después de hacer un análisis de las tablas y las gráficas comparativas se puede concluir que Firebird muestra un mejor desempeño debido a que su tiempo de respuesta es menor respecto al motor de base de datos MySQL.
- MySQL obtiene mayor calificación en las características comparadas, ya que soporta más tipos de datos y ofrece soporte de sus propios desarrolladores siendo así más confiable y seguro, además tiene gran cantidad de documentación.
- Debido a que Firebird es totalmente libre, no existen costos en cuanto a la obtención de este motor de base de datos, ni se tiene que pagar si se desea distribuir nuestras aplicaciones, caso contrario sucede con MySQL que a pesar de ser un SGBD libre y para distribuir las aplicaciones nos vemos obligados a usar el mismo tipo de licencia o de lo contrario tendría que obtener una licencia comercial.
- El SGBD Firebird obtuvo un 92.8% sobre un total de 100 puntos, obteniendo de esta manera el mayor puntaje, y MySQL obtuvo un 88.74%
- Firebird es el Sistema Gestor de Bases de Datos ganador ya que además de ofrecer un mejor tiempo de respuesta en las pruebas presenta muy buenas características.

## **CAPÍTULO IV**

### **SISTEMA PARA LA GESTIÓN DE EQUIPOS DE RADIO DEL ESCUADRÓN DE COMUNICACIONES N° 11**

#### **4.1. Introducción**

El gran crecimiento de la tecnología y el desarrollo de las empresas e instituciones han llevado a la necesidad del desarrollo de aplicaciones que faciliten su desenvolvimiento en todas sus áreas, logrando mayor eficiencia y brindando servicios de calidad, para lo cual es necesario el desarrollo de aplicaciones que permitan realizar las tareas diarias de una manera sencilla y se puedan acceder a través de una red.

Es por ello que las autoridades del Escuadrón de Comunicaciones N° 11 de la Brigada de Caballería Blindada “GALÁPAGOS” en su afán de mejorar los servicios de préstamos de equipos y materiales de radio pertenecientes a dicha entidad, ha decidido y visto adecuado el desarrollo de una aplicación que permita automatizar las labores que se llevan a cabo.

#### **4.2. Descripción del problema**

El Escuadrón de Comunicaciones N° 11 de la Brigada de Caballería Blindada “GALÁPAGOS” posee materiales y equipos de radio que de acuerdo a una solicitud son prestados al personal perteneciente a la entidad, actualmente estos procesos se manejan de una manera ineficiente ya que son administrados manualmente. Es por eso que se ha planteado el desarrollo de una aplicación que automatice todos estos procesos y se lleven a cabo de una manera más óptima y eficaz, facilitando las actividades a realizar por parte del encargado y de las personas que manejan y utilizan este servicio.

Este sistema permitirá realizar préstamos de los materiales y equipos de radio solamente al personal perteneciente a la institución, así como también la administración de los usuarios y materiales por parte de un usuario administrador. Además todos los usuarios podrán realizar consultas sobre los préstamos que han realizado, o la actualización de su información.

Como bien se sabe una aplicación debe desarrollarse en base a una metodología, siendo la que mejor se ajusta a las características de la aplicación, se utiliza la metodología XP.

#### **4.3. Metodología de desarrollo**

La metodología usada para el desarrollo del sistema es XP (Programación Extrema). La programación extrema se basa en una serie de reglas y principios de la ingeniería de software tradicional pero con la excepción de que XP da prioridad a las tareas que es lo que precisamente da resultados directos y reducen los procesos largos de recopilación de gran información para generar resultados esperados.

La programación extrema es un modelo de desarrollo muy disciplinado y que se apoya en ciertos valores fundamentales y necesarios acoplar a nuestro grupo de trabajo como son: comunicación, simplicidad, realimentación, coraje.

Debido a que XP se enfoca en la entrega frecuente de resultados es necesario que en poco tiempo se pueda generar resultados de gran importancia, es por eso que para obtener más calidad implicaría el trabajo e incorporación de un gran equipo de trabajo.

#### **4.4. Planificación**

Esta fase contendrá información sobre la planificación del proyecto como son las historias de usuario, las diferentes iteraciones que se llevan a cabo entre las historias y más detalladamente las tareas dentro de las iteraciones.

##### **4.4.1. Historias de usuario**

Permiten conocer los requerimientos y necesidades principales de los usuarios que van hacer uso de la aplicación estas son escritas por los clientes en un lenguaje no técnico y tienen la misma finalidad que los casos de uso.

##### **HU1 Autenticación de usuarios**

“La aplicación debe permitir la autenticación de usuarios para el ingreso a las páginas, las mismas que deben estar de acuerdo a los permisos establecidos para cada uno de ellos”.

## **HU2 Administración de usuarios**

“El sistema debe contener varias opciones que permitan la administración de los usuarios, dichas opciones son la inserción, eliminación y actualización de los datos de los usuarios, entre otra funciones”. Esto lo podrá realizar solamente el usuario administrador.

## **HU3 Administración de materiales/equipos**

“El sistema debe contener varias opciones que permitan la administración de los materiales o equipos de radio, las opciones son la inserción, eliminación y actualización de los datos de los materiales, entre otras funciones”. Esto lo podrá realizar solamente el usuario administrador.

## **HU4 Préstamos de materiales/equipos**

“El sistema debe permitir realizar préstamos de los materiales a los usuarios, para esto se debe verificar que se encuentre disponible”. Esto lo realizará el usuario administrador.

## **HU5 Enviar materiales/equipos a reparación**

“El sistema deberá permitir registrar los materiales que se encuentran en reparación para ello se ingresará el material, fecha, falla en la cual se envió a reparación”. Esto lo podrá realizar solamente el usuario administrador.

#### **HU6 Actualización de datos**

“El sistema deberá permitir actualizar los datos a los usuarios”. Tanto el usuario administrador como todo el personal de dicha entidad podrán modificar su información.

#### **HU7 Presentación de Mensajes**

“Se debe tomar en consideración mensajes de error o de aceptación cuando se realice alguna acción y a su vez considerar acciones inválidas en la aplicación”.

#### **HU8 Presentación de Reportes/Consultas**

“Los usuarios solicitan al sistema la generación de un reporte o una consulta específica para ellos, el sistema aceptará la petición y procede a generar el reporte o consulta, los usuarios visualizan el reporte”.

#### **HU9 Consideraciones Adicionales**

“Se debe considerar aspectos extras que mejoren o a su vez optimicen la funcionalidad, operatividad o navegabilidad de la aplicación a desarrollarse”.

*Seguridad/confiabilidad.* Para mayor seguridad en el sistema se crearán contraseñas para cada usuario, lo que permitirá el acceso únicamente a las personas que tienen autorización para hacer uso del sistema.

*Usabilidad.* El sistema debe ser de fácil uso para los usuarios y presentar interfaces amigables, intuitivas y de fácil interacción con los usuarios.

*Disponibilidad.* El producto debe ser disponible únicamente para los usuarios que se encuentren registrados.

#### 4.4.2. Plan de entregas

Para determinar el tiempo que conlleva el desarrollo y ejecución de los requerimientos de la aplicación especificados en las historias de usuario, se debe plasmar un cronograma tentativo de acuerdo al versionado de la aplicación, así se tiene:

La versión inicial abarca las historias de usuario principales que indican o destacan la funcionalidad principal de la aplicación.

**Tabla IV.LIX.** Plan de entrega N°1

<b>N° de versión:</b> Versión 1.0	<b>Fecha final:</b> 27-01-2011
<b>N° Historia de Usuario</b>	<b>Nombre de Historia de Usuario</b>
HU2	Administración de usuarios
HU1	Autenticación de usuarios
HU3	Administración de materiales/equipos
HU4	Préstamos de materiales/equipos
HU5	Enviar materiales/equipos a reparación
HU9	Consideraciones Adicionales

En la versión completa a más de mejorar las historias de usuario definidas en la versión anterior, se toma en consideración las historias de usuario restantes que den la correcta funcionalidad de la aplicación.

**Tabla IV.LX.** Plan de entrega N°2

<b>N° de versión:</b> Versión 2.0	<b>Fecha final:</b> 28-02-2011
<b>N° Historia de Usuario</b>	<b>Nombre de Historia de Usuario</b>
HU6	Actualización de datos
HU7	Presentación de mensajes
HU8	Presentación de Reportes/Consultas

La última versión, contempla la refinación y verificación del funcionamiento correcto de la aplicación tomando en consideración todas las historias de usuario.

### Cronograma de cada historia de usuario

**Tabla IV.LXI.** Cronograma de cada historia de usuario

<b>Nº H.U</b>	<b>Responsables</b>	<b>Fecha Inicial</b>	<b>Fecha Final</b>
2	Administrador del sistema Desarrolladores Diseñador	27-10-2010	17-11-2010
1	Administrador del sistema Desarrolladores	18-11-2010	24-11-2010
9	Administrador del sistema Desarrolladores Diseñador	25-11-2010	01-12-2010
3	Administrador del sistema Desarrolladores Diseñador	02-12-2010	24-12-2010
4	Administrador del sistema Desarrolladores Diseñador	27-12-2010	18-01-2011
5	Administrador del sistema Desarrolladores Diseñador	19-01-2011	26-01-2011
6	Administrador del sistema Desarrolladores Diseñador	27-01-2011	02-02-2011
7	Administrador del sistema Desarrolladores	03-02-2011	09-02-2011
8	Administrador del sistema Desarrolladores Diseñador	10-02-2011	24-02-2011

#### 4.4.3. Iteraciones

Para el desarrollo de la aplicación fueron necesarias realizar varias iteraciones, en cada una de ellas se realizan nuevas tareas, se estudian las tareas más importantes que estarán contempladas en cada una de las iteraciones realizadas. Cada iteración tomará como tiempo máximo de desarrollo tres semanas, las mismas que se encuentran conformadas como a continuación se indica:

## **Iteración 1**

Empieza con una reunión con el cliente, es decir con los miembros y personal que solicitaron la ejecución de la aplicación en donde ponen en conocimiento las historias de usuario principales con las que va a constar la aplicación.

La primera iteración va a constar de las siguientes historias de usuario: HU2, HU9.

**Tiempo de Duración:** 3 semanas

**Fecha Inicial:** 25-10-2010

**Fecha Final:** 19-11-2010

### **HU2 Administración de usuarios**

El sistema debe contener varias opciones que permitan la administración de los usuarios, dichas opciones son la inserción, eliminación y actualización de los datos de los usuarios, entre otras funciones. Las tareas que se realizarán dentro de esta historia de usuario son:

**Tarea 1** *Determinar las opciones que debe realizar el administrador de la aplicación para la gestión de usuarios.*

**Responsables:** Administrador del sistema, Desarrolladores.

**Tiempo de Duración:** 2 días.

Para poder presentar una propuesta que se rija a los requerimientos especificados por la entidad solicitante, es necesario realizar un estudio previo sobre los parámetros involucrados y los propuestos como solución a las peticiones involucradas de dicha institución.

Las opciones que se proponen para el usuario administrador son la creación, eliminación, actualización de los datos de los usuarios, entre otras.

**Tarea 2** *Implementación de las opciones para el usuario administrador.*

**Responsables:** Administrador del sistema, Desarrolladores.

**Tiempo de Duración:** 3 días.

Una vez que se tengan determinadas las opciones que va a realizar un usuario administrador, se procederá a realizar la respectiva implementación de cada una de ellas.

Para la creación de nuevos usuarios a más de los datos personales se ingresará un login y password los mismos que servirán para la autenticación de usuarios y así permitir que solo usuarios autorizados puedan ingresar a la aplicación.

## **HU9 Consideraciones Adicionales**

Se debe considerar aspectos extras que mejoren o a su vez optimicen la funcionalidad, operatividad o navegabilidad de la aplicación a desarrollarse. Las tareas que se realizarán dentro de esta historia de usuario son:

**Tarea 1** *Diseñar interfaces amigables que sean de fácil uso para los usuarios, además crear contraseñas que permitan acceder solo a usuarios autorizados a la aplicación.*

**Responsables:** Administrador del sistema, Desarrolladores, Diseñador.

**Tiempo de Duración:** 3 días

Se diseñarán interfaces amigables para que los usuarios que hagan uso de la aplicación interactúen con la misma de una forma fácil. Además los botones para cada una de las opciones tendrán una etiqueta que vaya de acuerdo a lo que realice cada opción.

## **Iteración 2**

La iteración 2 va a constar de las siguientes historias de usuario: HU1, HU9.

**Tiempo de Duración:** 2 semanas

**Fecha Inicial:** 18-11-2010

**Fecha Final:** 02-12-2010

### **HU1 Autenticación de usuarios**

La aplicación debe permitir la autenticación de usuarios para el ingreso a las páginas, las mismas que deben estar de acuerdo a los permisos establecidos para cada uno de ellos. Para el cumplimiento de esta historia de usuario, se han realizado las siguientes tareas:

**Tarea 1** *Validación de los datos de los usuarios para el ingreso a la aplicación.*

**Responsables:** Administrador del sistema, Desarrolladores.

**Tiempo de Duración:** 2 días

Los usuarios ingresarán el login y password para autenticarse, estos datos se consultarán en la base de datos y si son correctos se accederá a las opciones que pueda realizar de acuerdo al tipo de usuario con el que se haya validado.

En esta iteración se debe tomar en cuenta la historia de usuario 2 y 9 para que solo los usuarios que existan en nuestra base de datos tengan la autorización para ingresar a la aplicación.

### **Iteración 3**

La tercera iteración va a constar de las siguientes historias de usuario: HU3, HU9.

**Tiempo de Duración:** 3 semanas

**Fecha Inicial:** 02-12-2010

**Fecha Final:** 24-12-2010

#### **HU3 Administración de materiales/equipos**

El sistema debe contener varias opciones que permitan la administración de los materiales o equipos de radio, las opciones son la inserción, eliminación y actualización de los datos de los materiales, entre otras funciones. Para el cumplimiento de esta historia de usuario, se han realizado las siguientes tareas:

**Tarea 1** *Determinar las opciones que debe realizar el administrador de la aplicación para la gestión de los materiales de radio.*

**Responsables:** Administrador del sistema, Desarrolladores.

**Tiempo de Duración:** 2 días

Una vez que el usuario haya ingresado sus datos y este sea un usuario administrador, ingresará a opciones que permitirán la inserción, eliminación, actualización de los datos de los materiales, listar materiales.

**Tarea 2** *Implementación de las opciones para el usuario administrador.*

**Responsables:** Administrador del sistema, Desarrolladores.

**Tiempo de Duración:** 3 días

Una vez que se tengan determinadas las opciones que va a realizar un usuario administrador para la gestión de los materiales o equipos de radio, se procederá a realizar la respectiva implementación de cada una de ellas.

#### **Iteración 4**

La cuarta iteración va a constar de las siguientes historias de usuario: HU4, HU9.

*Tiempo de Duración:* 3 semanas

*Fecha Inicial:* 27-12-2010

*Fecha Final:* 19-01-2011

#### **HU4 Préstamos de materiales/equipos**

El sistema debe permitir realizar préstamos de los materiales a los usuarios, para esto se debe verificar que se encuentre disponible.

**Tarea 1** *Realizar préstamos de los materiales o equipos de radio.*

*Responsables:* Administrador del sistema, Desarrolladores.

*Tiempo de Duración:* 3 días

Los préstamos los puede hacer únicamente un usuario administrador, para ello una vez que se haya validado procederá a realizar un préstamo, para la realización del préstamo se verificará que el usuario que solicita un material exista en la base de datos, al igual que el material pero éste además debe no estar prestado es decir debe estar en bodega, el material no debe estar registrado en reparación o no debe estar dañado.

#### **Iteración 5**

La iteración 5 va a constar de las siguientes historias de usuario: HU5, HU9.

**Tiempo de Duración:** 1 semanas

**Fecha Inicial:** 19-01-2011

**Fecha Final:** 27-01-2011

### **HU5 Enviar materiales/equipos a reparación**

El sistema deberá permitir registrar los materiales que se encuentran en reparación para ello se ingresará el material, fecha, falla en la cual se envió a reparación.

**Tarea 1** *Enviar material a reparación.*

**Responsables:** Administrador del sistema, Desarrolladores.

**Tiempo de Duración:** 2 días

Una vez que el usuario se haya autenticado y este sea un usuario administrador, ingresará a la aplicación. Para saber que un material o equipo de radio esta en reparación, este será registrado en una tabla de la base de datos indicando su falla y la fecha en la que se envió a reparación.

### **Iteración 6**

La iteración 6 va a constar de las siguientes historias de usuario: HU6, HU7.

**Tiempo de Duración:** 2 semanas

**Fecha Inicial:** 27-01-2011

**Fecha Final:** 09-02-2011

### **HU6 Actualización de datos**

El sistema deberá permitir actualizar los datos a los usuarios. Tanto el usuario administrador como todo el personal de dicha entidad podrán modificar su información.

**Tarea 1** *Actualizar los datos.*

**Responsables:** Administrador del sistema, Desarrolladores.

**Tiempo de Duración:** 2 días

Los usuarios podrán actualizar algunos de sus datos, para ello una vez que se haya autenticado ingresará a la aplicación. Para actualizar su información personal primeramente se visualizarán sus datos y ahí se procederá a modificar, no todos los datos pueden ser modificados.

### **HU7 Presentación de Mensajes**

Se debe tomar en consideración mensajes de error o de aceptación cuando se realice alguna acción y a su vez considerar acciones inválidas en la aplicación.

**Tarea 1** *Mostrar mensajes de error o de aceptación.*

**Responsables:** Administrador del sistema, Desarrolladores.

**Tiempo de Duración:** 2 días

Luego de ejecutar alguna acción se deberá mostrar mensajes de aceptación indicando que se ha realizado satisfactoriamente, o de lo contrario se mostrará mensajes de error indicando que ha ocurrido un error.

### **Iteración 7**

La iteración 7 va a constar de las siguientes historias de usuario: HU8, HU9.

**Tiempo de Duración:** 2 semanas

**Fecha Inicial:** 10-02-2011

**Fecha Final:** 28-02-2011

## **HU8 Presentación de Reportes/Consultas**

Los usuarios solicitan al sistema la generación de un reporte o una consulta específica para ellos, el sistema aceptará la petición y procede a generar el reporte o consulta, los usuarios visualizan el reporte.

**Tarea 1** *Generación de consultas.*

**Responsables:** Administrador del sistema, Desarrolladores, Diseñador.

**Tiempo de Duración:** 3 días

Tanto el usuario administrador como el otro tipo de usuario podrán realizar diferentes consultas. Para esto dependiendo de qué tipo de usuario sea, ingresará a la aplicación, y podrá generar las consultas implementadas para él.

### **4.4.4. Análisis de riesgos**

El análisis de riesgos permitirá determinar las amenazas y vulnerabilidades que se pueden presentar en el desarrollo del sistema. Para lo cuál se pondrá en marcha estrategias de Análisis y Gestión de Riesgos adecuada a su situación y entorno que contemple los recursos necesarios para tratar los posibles riesgos de una forma eficaz.

#### **Identificación del Riesgo**

La identificación de los riesgos está orientada a evitar que estos ocurran y en caso de que existieran plantear las posibles soluciones que ayudan a contrarrestar el riesgo. Se han identificado los siguientes riesgos y amenazas que se han considerado que pueden ocurrir en el desarrollo del proyecto.

**Tabla IV.LXII.** Identificación de riesgos

<b>RIESGOS</b>
R1: Incumplimiento de los requerimientos planteados
R2: Incumplimiento del cronograma
R3: Falta de comunicación entre los miembros del equipo de trabajo
R4: Problemas de acceso a las bases de datos
R5: Insuficiente información para el desarrollo del trabajo
R6: Interfaces mal diseñadas
R7: Insatisfacción de los usuarios
R8: Falta de disponibilidad del personal técnico

### Categorización del Riesgo

**Tabla IV.LXIII.** Categorización de riesgos

<b>RIESGOS</b>	<b>PROBABILIDAD</b>		
	<b>ALTA</b>	<b>MEDIA</b>	<b>BAJA</b>
R1			X
R2		X	
R3		X	
R4	X		
R5		X	
R6		X	
R7			X
R8		X	

### Gestión del Riesgo

**Tabla IV.LXIV.** Gestión de riesgos

<b>RIESGO</b>	<b>PLAN DE GESTIÓN</b>
<b>R1</b>	<p><b>Problema:</b> Sistema no cumple con las expectativas ni con los requerimientos planteados.</p> <p><b>Actividades a desarrollar:</b></p> <ul style="list-style-type: none"> <li>• Seguir y evaluar continuamente el avance del proyecto.</li> <li>• Interactuar continuamente con la institución solicitante y los desarrolladores</li> </ul> <p><b>Responsable:</b> Administrador del sistema</p>
<b>R2</b>	<p><b>Problema:</b> Retraso en la entrega del producto final.</p> <p><b>Actividades a desarrollar:</b></p> <ul style="list-style-type: none"> <li>• Planificar entregas parciales del trabajo desarrollado.</li> <li>• Controlar continuamente el avance del proyecto en base a la planificación inicialmente planteada.</li> </ul> <p><b>Responsable:</b> Administrador del sistema</p>

<b>R3</b>	<p><b>Problema:</b> Incapacidad de llegar a consensos.</p> <p><b>Actividades a desarrollar:</b></p> <ul style="list-style-type: none"> <li>• Designar a un miembro que lidere las actividades de todos.</li> <li>• Motivar a que trabajen conjuntamente y que tengan un alto nivel de liderazgo.</li> </ul> <p><b>Responsable:</b> Administrador del sistema</p>
<b>R4</b>	<p><b>Problema:</b> La base de datos no se adapta al lenguaje de programación utilizado, provocando así retrasos en la entrega del proyecto.</p> <p><b>Actividades a desarrollar:</b></p> <ul style="list-style-type: none"> <li>• Investigar más a fondo sobre el motor de base de datos a utilizar y cómo acceder a éste desde el lenguaje de programación php.</li> </ul> <p><b>Responsable:</b> Programadores</p>
<b>R5</b>	<p><b>Problema:</b> Mal enfoque de la solución del problema</p> <p><b>Actividades a desarrollar:</b></p> <ul style="list-style-type: none"> <li>• Planificar reuniones de trabajo para abstraer los verdaderos requerimientos del cliente.</li> </ul> <p><b>Responsable:</b> Administrador del sistema</p>
<b>R6</b>	<p><b>Problema:</b> La aplicación no es usada.</p> <p><b>Actividades a desarrollar:</b></p> <ul style="list-style-type: none"> <li>• Integrar a los potenciales usuarios del sistema a la elaboración de interfaces.</li> </ul> <p><b>Responsable:</b> Diseñador</p>
<b>R7</b>	<p><b>Problema:</b> Desperdicio de los recursos empleados en una aplicación no utilizada.</p> <p><b>Actividades a desarrollar:</b></p> <ul style="list-style-type: none"> <li>• Identificar y evaluar permanentemente el cumplimiento de los requerimientos de los usuarios.</li> </ul> <p><b>Responsable:</b> Programadores</p>
<b>R8</b>	<p><b>Problema:</b> Atraso en los cronogramas establecidos y en la respectiva entrega de los avances.</p> <p><b>Actividades a desarrollar:</b></p> <ul style="list-style-type: none"> <li>• Establecer políticas internas de control.</li> <li>• Fomentar el espíritu de responsabilidad.</li> </ul> <p><b>Responsable:</b> Administrador del sistema</p>

#### 4.4.5. Estudio de la Factibilidad

##### FACTIBILIDAD TÉCNICA

Está enfocada a los requerimientos hardware, software y de recurso humano necesarios para la implementación del sistema. La institución cuenta con los recursos hardware, software y una infraestructura de red necesarios para el desarrollo del sistema, y el cuál

podrá ejecutarse sin ningún inconveniente. A continuación se detalla los recursos existentes y requeridos para el desarrollo del sistema.

### Recurso hardware disponible

El hardware con el que cuenta para el desarrollo de la aplicación es el siguiente:

**Tabla IV.LXV.** Hardware disponible

<b>Equipo</b>	<b>Cantidad</b>	<b>Descripción</b>
Computadoras	2	Procesador Intel Celeron de 1.80 GHZ Memoria RAM de 1 GB Disco duro de 160 GB Monitor, teclado, mouse Tarjeta de red.
Impresora	1	HP Photosmart C5280
Infraestructura de red		

### Recurso software disponible

Para el desarrollo del sistema se dispone del siguiente software:

**Tabla IV.LXVI.** Software disponible

<b>Software</b>	<b>Descripción</b>
Windows XP Professional	Sistema Operativo
Microsoft Office 2007	Herramienta para realizar la documentación

### Recurso software requerido

**Tabla IV.LXVII.** Software requerido

<b>Software</b>	<b>Descripción</b>
Firebird	Motor de base de datos
Php	Lenguaje de programación
Apache	Servidor web

## Recurso Humano

**Tabla IV.LXVIII.** Recurso humano requerido

<b>Función</b>	<b>Tarea</b>
Programadores	Se encargarán del desarrollo y diseño de la arquitectura de la aplicación.
Diseñador	Se encargará del diseño de la interfaz de la aplicación, con la cuál interactuará el usuario.
Administrador del sistema	Encargado de controlar que el desarrollo de la aplicación se lleve a cabo dentro del tiempo establecido.

La propuesta se encuentra técnicamente viable puesto que la institución posee una infraestructura tecnológica adecuada, así como también recursos hardware y software suficiente para que la solución planteada pueda ser ejecutada de manera correcta.

### **FACTIBILIDAD OPERATIVA**

El sistema a desarrollar es deseable, puesto que el mismo presta los beneficios esperados por la institución facilitando así las tareas que se realizan diariamente. Para garantizar el buen funcionamiento de la aplicación será desarrollado de acuerdo a los requerimientos planteados por los usuarios.

### **FACTIBILIDAD LEGAL**

El sistema a desarrollar no incumple con ninguno de los reglamentos y leyes actualmente vigentes en la institución, por lo cual puede decirse que es legalmente factible su implementación.

### **FACTIBILIDAD ECONÓMICA**

Como se conoce en estos días la principal riqueza de las instituciones gira en torno a la automatización de procesos y manejo de la información.

Para justificar la inversión que se tiene que realizar para este proyecto se han de identificar los costos por desarrollo e implantación, así como los beneficios de los que se beneficiará la institución a través del desarrollo del proyecto.

**Tabla IV.LXIX.** Costos de desarrollo

<b>Descripción</b>	<b>Cantidad</b>	<b>Valor</b>	<b>Horas</b>	<b>Días</b>	<b>Total</b>
Programador	2	3\$	4	50	1200
Diseñador	1	2\$	2	40	160
Administrador del sistema	1	3\$	4	60	720
Movilización					50
Suministros de oficina					15
Fotocopias, impresiones					20
Otros (servicio telefónico, imprevistos)					60
Total					2225

El presupuesto es referencial puesto que es un proyecto a desarrollarse como objetivo de tesis de grado.

### **Beneficios**

Los beneficios que se obtendrán con el sistema que se desarrollará son los siguientes:

- La aplicación permitirá realizar una correcta gestión de los materiales/equipos de radio.
- Los usuarios podrán realizar las tareas de manera fácil, ya que se encuentran automatizadas.
- Información más confiable.
- Mejor servicio a los usuarios.
- Disminución del tiempo de trabajo en el desarrollo de sus actividades diarias.

## 4.5. Cronograma de trabajo

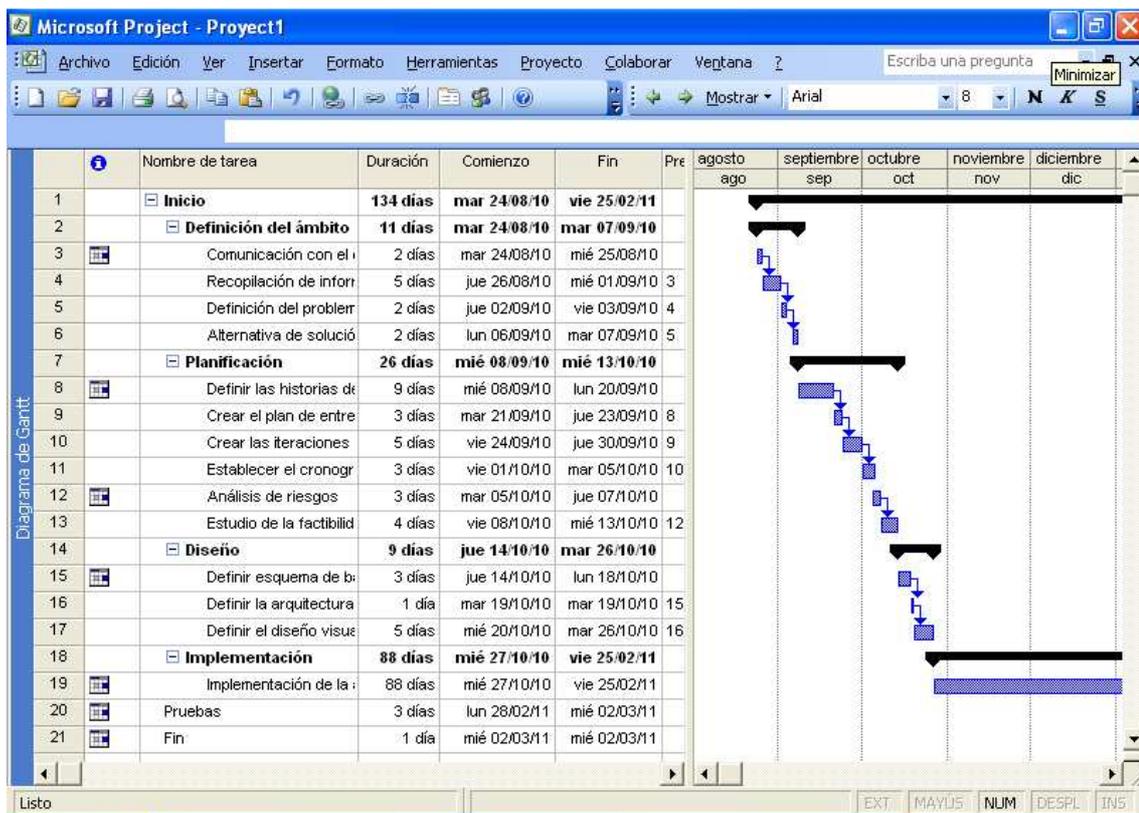


Figura IV.40 Cronograma

## 4.6. Diseño

### 4.6.1. Arquitectura del sistema



Figura IV.41 Arquitectura del sistema

**Capa de Presentación.** La capa de presentación es donde se encuentra la GUI es decir las interfaces de usuario en este caso es una aplicación Web.

**Capa de Acceso a Datos.** Contempla la lógica de acceso a la base de datos, centraliza la funcionalidad de acceso a la base de datos.

**Base de Datos.** Contiene los procedimientos almacenados y acciones solicitadas por el cliente hacia la base de datos.

#### 4.6.2. Esquema de la base de datos

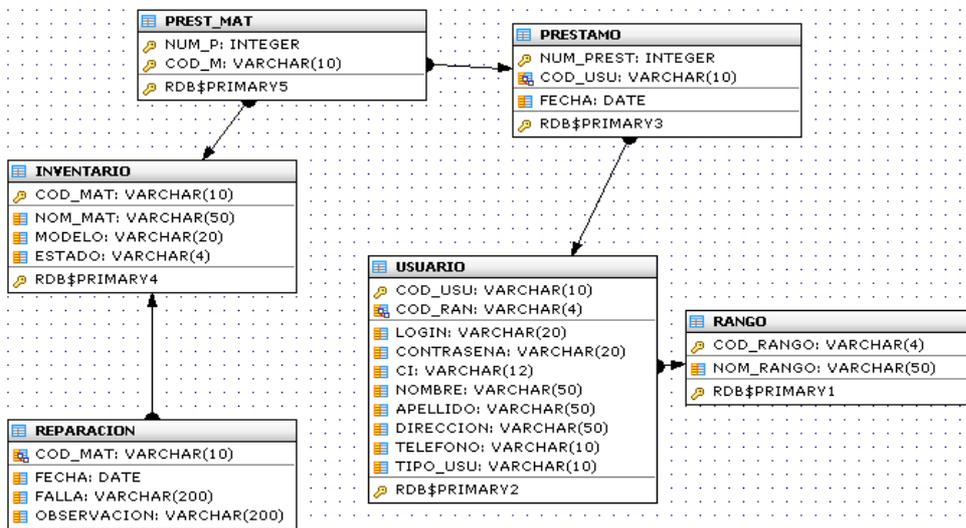


Figura IV.42 Base de datos

#### 4.6.3. Diagrama de componentes

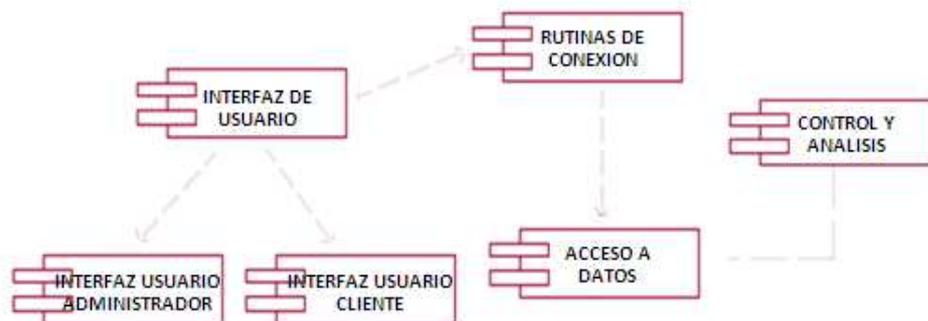


Figura IV.43 Diagrama de componentes

#### 4.6.4. Diagrama de despliegue



**Figura IV.44** Diagrama de despliegue

### 4.7. Implementación

#### 4.7.1. Introducción

Esta fase involucra las versiones del producto, desarrollados por iteraciones y es en donde se debe asegurar que todos sus módulos o iteraciones cumplan con lo establecido en la fase de planificación; y estas pueden ser integradas para obtener el producto final. Además de la creación del código de la aplicación se desarrolla la estructura de la solución y la integración de todas las iteraciones.

#### 4.7.2. Implementación de las iteraciones

Una vez que se han determinado todos los requerimientos se procede a implementar cada una de las iteraciones; en cada iteración se seguirá mejorando la funcionalidad de la aplicación (Código de la aplicación: Ver Anexo 1).

#### **Iteración 1**

Para la creación de la primera versión se comienza con la implementación de la parte inicial de la aplicación, tomando en cuenta consideraciones generales que debe

contemplar la aplicación así como también cada una de las opciones que puede realizar un usuario administrador con respecto a la administración de usuarios (inserción, eliminación, actualización..).

Para la implementación de la primera iteración se tomaron en cuenta las historias de usuario: HU2, HU9 establecidos y analizados anteriormente.

### **Iteración 2**

Continuando con el desarrollo se procedió con la implementación de la validación de usuarios, tomando en cuenta que sólo los usuarios que se encuentren en la base de datos puedan acceder a la aplicación dependiendo de sus permisos. Para la implementación de esta iteración se tomaron en cuenta las historias de usuario: HU1 y HU9.

### **Iteración 3**

En esta iteración se implementa lo referente a la gestión de materiales/equipos de radio como son: inserción, eliminación, actualización de materiales y de algunas otras opciones que sólo el usuario administrador las puede realizar.

Para la implementación de la iteración 3 se tomaron en cuenta las historias de usuario: HU3, HU9 establecidos y analizados anteriormente.

### **Iteración 4**

En esta iteración se implementó lo que es la prestación de los materiales o equipos de radio. Para la implementación de esta iteración se debe tomar muy en cuenta que solamente un usuario administrador puede realizar los préstamos. Se tomaron en cuenta las historias de usuario: HU4, HU9.

### **Iteración 5**

En esta iteración se realiza la implementación del código para registrar un material en la tabla reparación cuando uno de ellos sea enviados a reparación. Para la implementación de la iteración 5 se tomaron en cuenta las historias de usuario: HU5, HU9.

### **Iteración 6**

Esta iteración comenzó con el desarrollo del módulo para el personal de la institución que no sea un usuario administrador, iniciando con la implementación de actualización de datos. Además se realizó la presentación de mensajes cuando se realice alguna acción, estos mensajes pueden ser de aceptación o de error. Se tomaron en cuenta las historias de usuario: HU6, HU7.

### **Iteración 7**

El requerimiento implementado en esta iteración es la generación de consultas necesarias para los usuarios. Para el desarrollo de esta iteración se tomaron en cuenta las historias de usuario: HU8, HU9.

En cada una de las iteraciones se realizaron verificaciones del funcionamiento correcto del sistema y algunas modificaciones en algunos aspectos que lo requerían.

### **4.7.3. Integración**

Se realiza la integración de cada una de las partes de la aplicación implementadas anteriormente para finalmente obtener el módulo final, para lo cual se debe verificar que los componentes funcionen correctamente, evitando así, fallas innecesarias al momento de integrar dichos componentes.

Una vez integrados los módulos se realizaron pruebas continuas para verificar su adecuado funcionamiento, para ello se ingresaron datos en la aplicación y se verificaron que estos sean correctos y se almacenen en la base de datos. Lo cual se cumplió a cabalidad con todos los requerimientos establecidos inicialmente, teniendo que realizar solo pequeñas modificaciones a las interfaces de usuario.

#### **4.7.4. Prototipos interfaces de usuario**

##### **Versión 1.0**

##### **Página de inicio**

Para la creación de la versión inicial se diseñó una pantalla mediante la cual ingresarán a la aplicación dependiendo de los permisos establecidos para cada uno de los usuarios. Se presentan dos niveles de administración, para el usuario administrador y para el resto del personal de la institución al cual se denominó como usuario personal.

##### **Página de validación de usuarios**



El prototipo de la página de validación de usuarios es un formulario con un fondo azul oscuro. En la parte superior, el título "Iniciar sesión" está escrito en blanco. Debajo del título, hay dos campos de entrada de texto con un fondo azul claro y un borde azul oscuro. El primer campo está etiquetado como "Login:" y el segundo como "Password:". Debajo de estos campos, hay un botón rectangular con el texto "Entrar" en azul oscuro. El formulario está centrado en la página.

**Figura IV.45** Validación de usuarios

##### **Página principal del administrador**

Luego que se valide como un usuario administrador, este accede y puede realizar varias opciones como pueden ser: administración de usuarios, administración de equipos/materiales, préstamos de equipos/materiales, consultas.

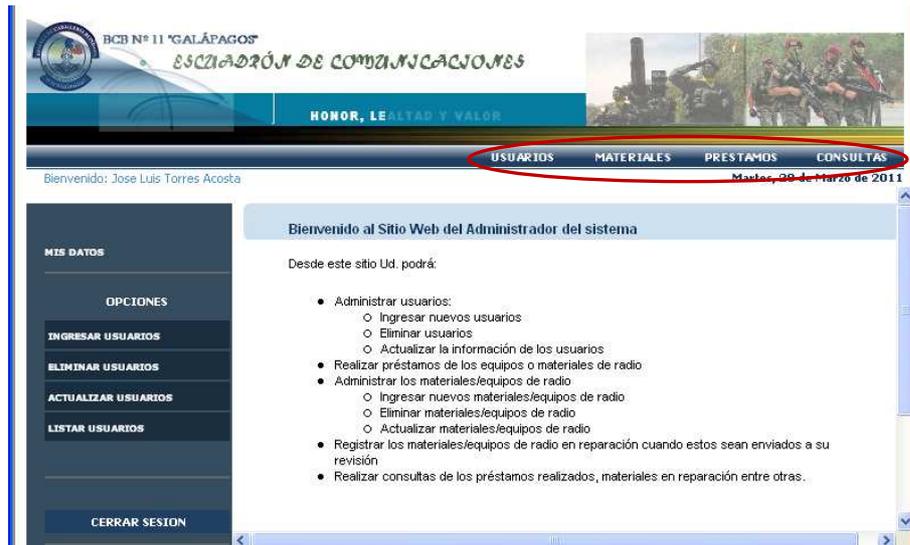


Figura IV.46 Opciones del usuario administrador

A continuación se presentan algunas pantallas diseñadas para las opciones que un usuario administrador puede realizar con respecto a la gestión de usuarios.

### Ingresar usuarios

Permite ingresar nuevos usuarios al sistema, para lo cual se ingresarán los datos solicitados.



Figura IV.47 Ingresar usuarios

## Eliminar usuarios

Para la eliminación de usuarios, primeramente se visualizará una pantalla en la cual se ingresará el número de cédula de identidad o también se podrá realizar la búsqueda por medio del nombre del usuario para posteriormente proceder a eliminar. Esto se aplicará a todas las búsquedas que se realicen.



Figura IV.48 Buscando usuario a eliminar

A continuación se mostrarán los datos del usuario y para eliminarlo hay que confirmar presionando en *Eliminar*.



Figura IV.49 Eliminar usuarios

## Actualizar usuarios

Al igual que en la eliminación de usuarios, en la primera pantalla se ingresará el número de cédula de identidad del usuario y enviar a *Buscar*.

The screenshot shows the 'ACTUALIZAR USUARIOS' (Update Users) form. At the top, the header includes the BCB N° 11 'GALÁPAGOS' logo and the motto 'HONOR, LEALTAD Y VALOR'. A navigation bar contains 'USUARIOS', 'MATERIALES', 'PRESTAMOS', and 'CONSULTAS'. The user is logged in as 'Jose Luis Torres Acosta' on 'Martes, 29 de Marzo de 2011'. On the left, a sidebar menu lists 'MIS DATOS', 'OPCIONES', 'INGRESAR USUARIOS', 'ELIMINAR USUARIOS', 'ACTUALIZAR USUARIOS', 'LISTAR USUARIOS', and 'CERRAR SESION'. The main form area has a title 'ACTUALIZAR USUARIOS' and a prompt 'Ingrese C.I. o el nombre del usuario:'. Below this, there is a text input field for 'N° C.I.' containing '1845798789' and a 'Buscar' button.

Figura IV.50 Buscando usuario a actualizar

The screenshot shows the 'MODIFICAR DATOS DEL USUARIO' (Modify User Data) form. The header and navigation elements are identical to the previous screenshot. The sidebar menu is also the same. The main form area has a title 'MODIFICAR DATOS DEL USUARIO'. It contains several input fields for user information: 'Ci:' (1845798789), 'Nombres:' (Josue Javier), 'Apellidos:' (Davila E), 'Dirección:' (BCB), 'Teléfono:' (084231523), 'Login:' (jdevila), 'Password:' (jjdevila1980), 'Código rango:' (R004), and 'Tipo de usuario:' (personal). A 'Modificar' button is located at the bottom of the form.

Figura IV. 51 Actualizar usuarios

## Listar usuarios

Permite realizar un listado de todos los usuarios existentes en el sistema.

The screenshot shows a web application interface for 'BCB Nº 11 "GALÁPAGOS" ESCUADRÓN DE COMUNICACIONES'. The header includes the organization's name, logo, and the motto 'HONOR, LEALTAD Y VALOR'. A navigation bar contains links for 'USUARIOS', 'MATERIALES', 'PRESTAMOS', and 'CONSULTAS'. The user is logged in as 'Jose Luis Torres Acosta' on 'Martes, 29 de Marzo de 2011'. The main content area is titled 'LISTAR USUARIOS' and displays a table of users. A sidebar on the left contains navigation options: 'MIS DATOS', 'OPCIONES', 'INGRESAR USUARIOS', 'ELIMINAR USUARIOS', 'ACTUALIZAR USUARIOS', 'LISTAR USUARIOS', and 'CERRAR SESION'.

CI	Nombres	Apellidos	Dirección	Teléfono
0632548794	Pedro Manuel	Abad Solis	Los Pinos	03215469
0635462356	Fabrizio Javier	Cepeda L.	La Merced	032156423
0632165488	Rafael Felipe	Lopez Ramirez	Loma de Quito	093654215
1856987122	Juan Andres	Muñoz Paredes	Los Pinos	999999999
0635421366	Daniilo Fabian	Garces Perez	Condominios Chimborazo	084215789
0635262344	Victor Daniel	Izurietta Solis	Los Pinos	098456231
0612456789	Miguel Angel	Barroso Orozco	Los Pinos	097125468
0632145687	Pedro Javier	Olmedo Acosta	Loma de Quito	095147562
1845784126	Esteban Samir	Robles Castelo	Bellavista	032457894
0632569874	Marco Patricio	Yepez Molina	La Ceramica	098456325

Figura IV. 52 Listar usuarios

A continuación se visualiza las opciones para la gestión de equipos/materiales de radio que puede realizar el usuario administrador.

## Ingresar equipos/materiales

En la pantalla de ingreso de equipos y materiales de radio se debe llenar los datos solicitados para posteriormente enviar a guardarlos.

The screenshot shows the 'INGRESAR LOS DATOS DEL EQUIPO O MATERIAL DE RADIO' form in the same web application. The header and navigation bar are identical to the previous screenshot. The main content area is a form with the following fields: 'Código:', 'Nombre:', 'Modelo:', 'Estado:' (with a dropdown menu set to 'Disponible'), and 'FechaE:' (with the value '2011/03/29'). There are 'Ingresar' and 'Cancelar' buttons at the bottom of the form. The sidebar on the left contains navigation options: 'MIS DATOS', 'OPCIONES', 'INGRESAR EQUIPO/MATERIAL', 'ELIMINAR EQUIPO/MATERIAL', 'ACTUALIZAR EQUIPO/MATERIAL', and 'LISTAR EQUIPO/MATERIAL'.

Figura IV.53 Ingresar equipos/materiales

### Eliminar equipos/materiales

Se ingresa el código del equipo o material que se desea eliminar, a continuación busca si existe y lo elimina.

The screenshot shows a web application interface for the 'BCB Nº 11 GALÁPAGOS ESCUADRÓN DE COMUNICACIONES'. The header includes the organization's name and the motto 'HONOR, LEALTAD Y VALOR'. A navigation menu at the top contains 'USUARIOS', 'MATERIALES', 'PRESTAMOS', and 'CONSULTAS'. The user is logged in as 'Jose Luis Torres Acosta' on 'Martes, 29 de Marzo de 2011'. The main content area is titled 'ELIMINAR EQUIPO O MATERIAL DE RADIO'. It features a sidebar with 'OPCIONES' including 'INGRESAR EQUIPO/MATERIAL', 'ELIMINAR EQUIPO/MATERIAL', 'ACTUALIZAR EQUIPO/MATERIAL', 'LISTAR EQUIPO/MATERIAL', 'REPARACION', and 'ENTREGAR DE REPARACION'. The main form prompts the user to 'Ingrese el código o nombre:' and includes a 'Código:' input field and a 'Buscar' button.

Figura IV.54 Eliminar equipos/materiales

### Préstamos de equipos/materiales

En la pantalla de préstamos, se ingresa el número de cédula del usuario que solicita el préstamo actual.

The screenshot shows the 'PRESTAMO DE EQUIPOS/MATERIALES DE RADIO' web interface. The header and navigation menu are identical to the previous figure. The user is logged in as 'Jose Luis Torres Acosta' on 'Martes, 29 de Marzo de 2011'. The main content area is titled 'PRESTAMO DE EQUIPOS/MATERIALES DE RADIO'. The sidebar 'OPCIONES' includes 'REALIZAR PRESTAMOS', 'LISTAR PRESTAMOS', 'ACTUALIZAR PRESTAMO', and 'CERRAR SESION'. The main form prompts the user to provide details for the loan, including 'Número de préstamo:' (with the value '4'), 'Fecha:' (with the value '2011/03/29'), and 'N° C.I.'. There is an 'Aceptar' button at the bottom of the form.

Figura IV.55 Préstamos de equipos/materiales

## Versión 2.0

A la página de inicio y de validación en esta versión no se realizó ningún cambio y a las demás solo se realizaron pequeñas modificaciones a las interfaces de usuario. Esta versión principalmente involucra lo que es la presentación de mensajes de aceptación o de error y la generación de consultas tanto para el usuario administrador como para el personal.

## Presentación de Mensajes

Para cualquier acción que se realice se presentarán mensajes ya sean de aceptación o mensajes de error que indicarán lo que ha sucedido.



Figura IV.56 Presentación de mensaje



El usuario no existe

Figura IV.57 Presentación de mensaje

Además se realizó todo lo referente a consultas ya sea por parte del administrador o el personal. Se visualiza un listado de todos los usuarios de nuestra base de datos.

LISTAR USUARIOS				
CI	Nombres	Apellidos	Dirección	Teléfono
0632548794	Pedro Manuel	Abad Solis	Los Pinos	03215469
0635462356	Fabrizio Javier	Cepeda L	La Merced	032156423
0632165488	Rafael Felipe	Lopez Ramirez	Loma de Quito	093654215
1856987122	Juan Andres	Muñoz Paredes	Los Pinos	999999999
0635421366	Danilo Fabian	Garces Perez	Condominios Chimborazo	084215789
0635262344	Victor Daniel	Izurietta Solis	Los Pinos	098456231
0612456789	Miguel Angel	Barroso Orozco	Los Pinos	097125468
0632145687	Pedro Javier	Olmedo Acosta	Loma de Quito	095147562
1845784126	Esteban Samir	Robles Castelo	Bellavista	032457894

**Figura IV.58** Consulta de usuarios

### **Versión 3.0**

En esta versión se realizó la refinación de la aplicación y la verificación del funcionamiento correcto de la aplicación. A las pantallas se realizaron pequeñas modificaciones para una mejor presentación. Luego de haber corregido las versiones anteriores, se cuenta con una versión final, la cual cumple con todos los requerimientos establecidos inicialmente.

### **4.8. Pruebas**

A continuación se describen las pruebas más importantes que se realizó a la aplicación.

#### **Pruebas de validación y verificación**

Las pruebas de validación ayudarán a verificar si los requerimientos se han cumplido en su totalidad. Además de controlar las transacciones al momento de ingresar a la base de datos.

A continuación los requerimientos con su respectiva comprobación:

- **Ingreso de registros**

**Datos de entrada:**

**Tabla IV.LXX.** Datos de entrada para el ingreso de registros

Dato	Descripción
CI	Número de cédula de identidad (Clave primaria)
Nombres	Nombres del usuario
Apellidos	Apellidos del usuario
Dirección	Dirección actual del usuario
Teléfono	Teléfono del usuario
Login	Login del usuario para la autenticación.
Password	Password del usuario para la autenticación.
Tipo de usuario	Administrador o personal

**Datos de salida:** Usuario ingresado con éxito o mensaje indicando que ha existido un error.

**Descripción:** Una vez que se ingresaron los datos en la aplicación se procedió a verificar si estos se encuentran en la base de datos; y se comprueba que han sido almacenados de forma correcta de acuerdo a restricciones que se le implementaron para cada uno de los campos.

BCB Nº 11 GALÁPAGOS  
ESCUADRÓN DE COMUNICACIONES  
HONOR, LEALTAD Y VALOR

USUARIOS MATERIALES PRESTAMOS CONSULTAS

Bienvenido: Jose Luis Torres Acosta Martes, 29 de Marzo de 2011

**INGRESAR LOS DATOS DEL USUARIO**

CI:

Nombres:

Apellidos:

Dirección:

Teléfono:

Login:

Password:

Código rango:

Tipo de usuario:

MIS DATOS

OPCIONES

INGRESAR USUARIOS

ELIMINAR USUARIOS

ACTUALIZAR USUARIOS

LISTAR USUARIOS

CERRAR SESION

**Figura IV.59** Insertando datos

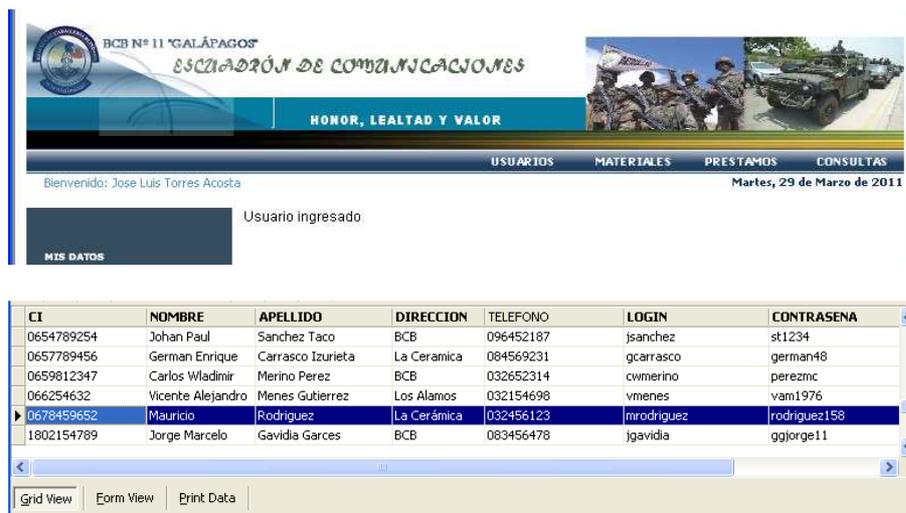


Figura IV.60 Verificando datos en la BD

En el caso que se ingrese datos que ya existen (cédula, login o password) se presenta la siguiente pantalla.



Figura IV.61 Mensaje de error

- **Eliminación de registros**

**Datos de entrada:**

**Tabla IV.LXXI.** Datos de entrada para eliminación de registros

Dato	Descripción
Cédula identidad	Número de cédula de identidad del usuario que se va a eliminar

**Datos de salida:** Usuario eliminado con éxito o no existe el usuario.

**Descripción:** Luego de ingresar el número de cédula de identidad del usuario a eliminar, busca en la base de datos y en el caso de que exista se muestra la información del usuario y procede a eliminarlo. La transacción se realizó correctamente ya que se verificó que el usuario exista para eliminarlo y en el caso que no exista se muestra un mensaje.



Figura IV.62 Eliminando datos

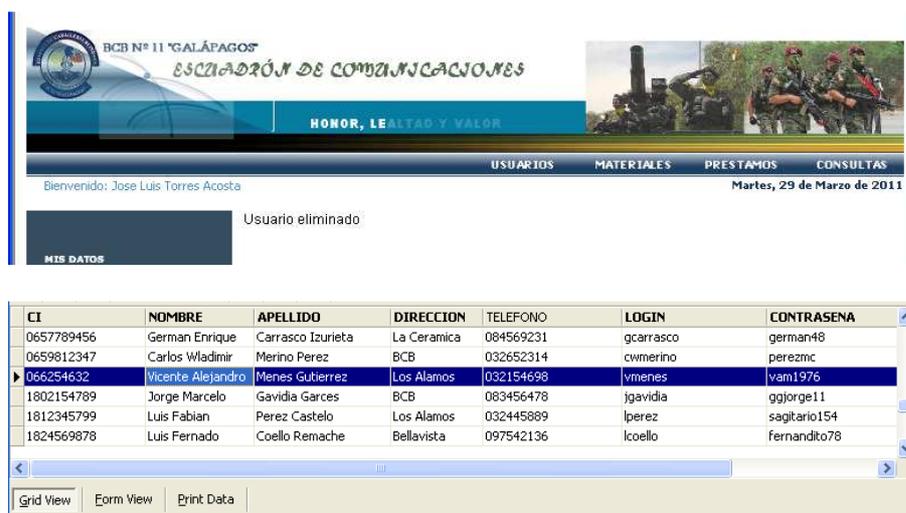


Figura IV.63 Verificando la eliminación en la BD

- Actualización de registros

**Datos de entrada:**

**Tabla IV.LXXII.** Datos de entrada para actualización de registros

Dato	Descripción
Cédula identidad	Número de cédula de identidad del usuario que se va a actualizar

**Datos de salida:** Usuario actualizado con éxito o no existe el usuario.

**Descripción:** Se comprobó que los datos son modificados correctamente en la base de datos.

CI	NOMBRE	APELLIDO	DIRECCION	TELEFONO	LOGIN	CONTRASENA
0654789254	Johan Paul	Sanchez Taco	BCB	096452187	jsanchez	st1234
0657789456	German Enrique	Carrasco Izurieta	La Ceramica	084569231	gcarrasco	german48
0659812347	Carlos Wladimir	Merino Perez	BCB	032652314	cwmerino	perezmc
066254632	Vicente Alejandro	Menes Gutierrez	Los Alamos	032154698	vmenes	vam1976
0678459652	Mauricio	Rodriguez	La Cerámica	032456123	mrodriguez	rodriguez158
1802154789	Jorge Marcelo	Gavida Garces	BCB	083456478	jpgavida	ggjorge11

Dato anterior

Grid View Form View Print Data

**Figura IV.64** Modificando datos



**Figura IV.65** Verificando actualización en la BD

Al igual que en la gestión de usuarios (inserción, eliminación y actualización de usuarios), se realizaron las pruebas en la gestión de materiales/equipos de radio y se verificaron que los datos fueron ingresados correctamente.

- **Préstamos de materiales**

**Datos de entrada:**

**Tabla IV.LXXIII.** Datos de entrada para realizar préstamos

Dato	Descripción
Número de préstamo	Número de préstamo a realizar
Fecha	Fecha en la que se realiza el préstamo.
Cédula identidad	Número de cédula de identidad del usuario que realiza el préstamo
Código de material	Código del equipo/material que se va a prestar.

**Datos de salida:** Préstamo realizado con éxito o mensaje que ha ocurrido un error.

**Descripción:** Luego de ingresar los datos necesarios para realizar un préstamo, verifica en la base de datos y si son correctos se almacenan.

BCB N° 11 "GALÁPAGOS"  
ESCUADRÓN DE COMUNICACIONES  
HONOR, LEALTAD Y VALOR

USUARIOS MATERIALES PRESTAMOS CONSULTAS

Bienvenido: Jose Luis Torres Acosta Martes, 29 de Marzo de 2011

**PRESTAMO DE EQUIPOS/MATERIALES DE RADIO**

MIS DATOS

OPCIONES

REALIZAR PRESTAMOS

LISTAR PRESTAMOS

ACTUALIZAR PRESTAMO

Número de préstamo:

Fecha:

N° C.I.:

BCB N° 11 "GALÁPAGOS"  
ESCUADRÓN DE COMUNICACIONES  
HONOR, LEALTAD Y VALOR

USUARIOS MATERIALES PRESTAMOS CONSULTAS

Bienvenido: Jose Luis Torres Acosta Martes, 29 de Marzo de 2011

**INGRESAR CODIGO DE MATERIAL A PRESTAR**

MIS DATOS

OPCIONES

REALIZAR PRESTAMOS

LISTAR PRESTAMOS

ACTUALIZAR PRESTAMO

N° Préstamo:

Fecha:  N° C.I.:

Código de material:

Figura IV.66 Realizando préstamos

BCB N° 11 "GALÁPAGOS"  
ESCUADRÓN DE COMUNICACIONES  
HONOR, LEALTAD Y VALOR

USUARIOS MATERIALES PRESTAMOS CONSULTAS

Bienvenido: Jose Luis Torres Acosta Martes, 29 de Marzo de 2011

**EQUIPOS/MATERIALES PRESTADOS**

MIS DATOS

OPCIONES

REALIZAR PRESTAMOS

LISTAR PRESTAMOS

ACTUALIZAR PRESTAMO

CODIGO	NOMBRE
M008	fchgfh

Desea retirar otro material?

Figura IV.67 Confirmación de préstamo realizado

The image shows two screenshots of a database application interface. The top screenshot displays a table named 'PRESTAMO' with columns 'NUM\_PREST', 'FECHA', and 'CI\_USU'. The bottom screenshot displays a table named 'PREST\_MAT' with columns 'NUM\_P' and 'COD\_M'.

NUM_PREST	FECHA	CI_USU
1	02.02.2011	0654789254
2	01.01.2010	0321457894
3	14.06.2011	2222222222
4	29.03.2011	0654712355

NUM_P	COD_M
1	M002
1	M003
3	M004
3	M005
4	M008

Figura IV.68 Verificando préstamos en la BD

Si se realiza en el mismo préstamo, el pedido del mismo material se indica un mensaje de error.

The image shows a screenshot of a web application interface. The header includes the logo of 'FCB Nº 11 GALÁPAGOS ESCUADRÓN DE COMUNICACIONES' and the motto 'HONOR, LEALTAD Y VALOR'. The navigation menu includes 'USUARIOS', 'MATERIALES', 'PRESTAMOS', and 'CONSULTAS'. The user is logged in as 'Jose Luis Torres Acosta' on 'Martes, 29 de Marzo de 2011'. The main content area displays an error message: 'MATERIALES/EQUIPOS PRESTADOS' and 'Código de equipo/material no existe o no esta disponible'. Below the message, there is a table with columns 'CODIGO' and 'NOMBRE', showing a row with 'M008' and 'fchgfh'. At the bottom, there is a question 'Desea retirar otro material?' with 'Si' and 'No' buttons.

Figura IV.69 Mensaje de error

Pero si se realiza en el mismo préstamo, el pedido de otro material esto se ejecuta correctamente.

BCB Nº 11 "GALÁPAGOS" ESCUADRÓN DE COMUNICACIONES  
HONOR, LEALTAD Y VALOR

USUARIOS MATERIALES PRESTAMOS CONSULTAS

Bienvenido: Jose Luis Torres Acosta Martes, 29 de Marzo de 2011

**INGRESAR CODIGO DE MATERIAL A PRESTAR**

MIS DATOS

OPCIONES

REALIZAR PRESTAMOS

LISTAR PRESTAMOS

ACTUALIZAR PRESTAMO

CERRAR SESION

Nº Préstamo:

Fecha:  Código de usuario:

Código de material:

BCB Nº 11 "GALÁPAGOS" ESCUADRÓN DE COMUNICACIONES  
HONOR, LEALTAD Y VALOR

USUARIOS MATERIALES PRESTAMOS CONSULTAS

Bienvenido: Jose Luis Torres Acosta Martes, 29 de Marzo de 2011

**MATERIALES/EQUIPOS PRESTADOS**

CODIGO	NOMBRE
M008	fchgfh
M010	nooo

Desea retirar otro material?

Figura IV.70 Realizando otro préstamo

Tabla          Contar registros PRESTAMO

Disparadores Datos Master/Detail View Descripción DDL Permisos Logging Comp

Record: 4

NUM_PREST	FECHA	CI_USU
1	02.02.2011	0654789254
2	01.01.2010	0321457894
3	14.06.2011	2222222222
4	29.03.2011	0654712355

Tabla         Contar registros PREST\_MAT

Campos Restricciones Indices Dependencias Disparadores Datos Master/Detail v

Record: 6

NUM_P	COD_M
1	M002
1	M003
3	M004
3	M005
4	M008
4	M010

Figura IV.71 Verificando datos en la BD

- **Generación de reportes/consultas**

Se realizaron diferentes reportes y consultas como por ejemplo: de préstamos realizados; de equipos/materiales existentes, prestados, disponibles; de préstamos realizados por un cierto usuario, etc. Para lo cual dependiendo del tipo de consulta debe ingresar los datos requeridos.

**Datos de entrada:**

**Tabla IV.LXXIV.** Datos de entrada para consultar préstamos

Dato	Descripción
Cédula identidad	Número de cédula de identidad del usuario del cual se desea ver los préstamos que ha realizado.

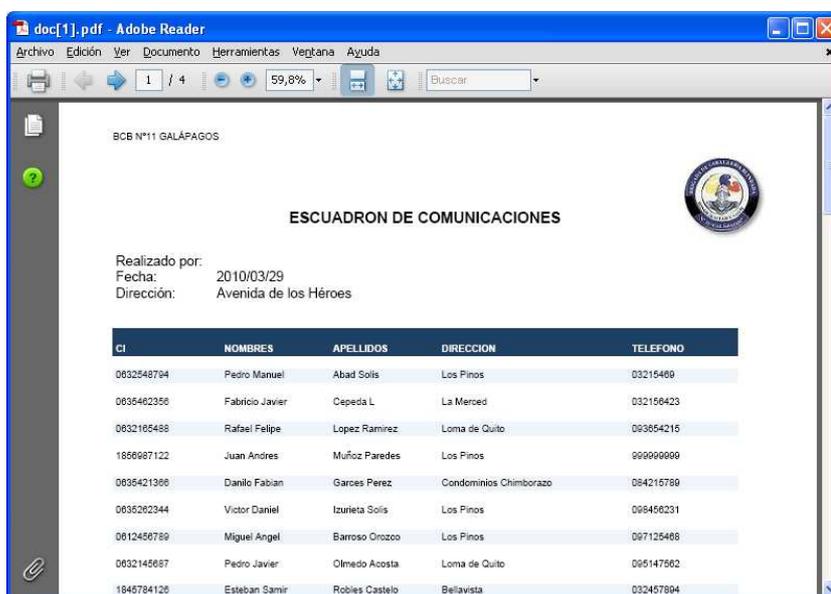
**Datos de salida:** Listado de todos los préstamos de un usuario si es que los tiene.

**Descripción:** Luego de ingresar el número de cédula de identidad, busca en la base de datos y en el caso de que exista se muestra los préstamos que tiene el usuario. La consulta se realizó exitosamente y visualizó los datos solicitados.



**Figura IV.72** Consultando préstamos

Los usuarios tienen la posibilidad de generar un documento en formato pdf de la consulta ejecutada, lo cual se realizó correctamente.



**Figura IV.73** Reporte en formato pdf de todos los usuarios

Todos los requerimientos se han cumplido a cabalidad por tanto las pruebas de validación y verificación se han completado con éxito.

## CONCLUSIONES

- El análisis comparativo ha permitido determinar el SGBD que optimice uno de los recursos más importantes como es el tiempo de ejecución en realizar los procesos en aplicaciones intensivas ya sean en inserción, selección, actualización o eliminación de datos, además proporciona una funcionalidad completa como soporte a procedimientos almacenados, triggers, vistas, etc. características indispensables que debe poseer un buen motor de base de datos.
- De acuerdo a los datos obtenidos en las diferentes pruebas se puede concluir que Firebird es más rápido que MySQL, característica muy importante el tiempo de respuesta en lo que satisfacción al cliente se refiere y en ambientes grandes de información.
- MySQL obtiene mejor calificación en algunas características tales como: soporta más tipos de datos, ofrece soporte de sus propios desarrolladores siendo así más confiable y seguro, además tiene gran cantidad de documentación.
- Debido a que Firebird es totalmente libre, no existen costos en cuanto a la obtención de este motor de base de datos, ni se tiene que pagar si se desea distribuir nuestras aplicaciones, caso contrario sucede con MySQL que a pesar de ser un SGBD libre y para distribuir las aplicaciones nos vemos obligados a usar el mismo tipo de licencia o de lo contrario tendría que obtener una licencia comercial.

- Es así que cada sistema tiene ventajas e inconvenientes, por lo que la elección de un SGBD para gestionar una base de datos depende de las necesidades de la empresa, por ejemplo costos, soporte, dimensionamiento, características técnicas y el factor más importante que es el tiempo de respuesta que ofrece al realizar sus transacciones.
- Para determinar el mejor Sistema Gestor de Bases de Datos se lo realizó en base a las pruebas en las cuáles se obtuvo un 82.68% para MySQL y para Firebird 100%; y en base a las características generales que debe poseer un motor de bases de datos en dónde MySQL obtuvo un 94.8% y Firebird un 85.6%.
- Firebird resultó ser el Sistema Gestor de Bases de Datos ganador con un 92.8% y MySQL obtuvo un 88.74%, esto se determinó de acuerdo a las pruebas realizadas y a las características que se tomaron en cuenta en el estudio comparativo.
- La implementación del sistema para la gestión de equipos de radio del Escuadrón de Comunicaciones N° 11 ayudará al desarrollo de las actividades que realiza la persona encargada ofreciendo al personal que hace uso de este servicio una atención más rápida y eficaz.

## RECOMENDACIONES

- Al momento de elegir un Sistema de Gestión de Base de Datos se debe tener en cuenta las necesidades de la organización como pueden ser volumen de información con el que se va a trabajar, costos, características o el rendimiento con el que deseen que responda frente a los procesos a realizar.
- Se recomienda este estudio como una guía para comparativas mucho más profundas en otras áreas específicas como por ejemplo seguridad entre MySQL y Firebird.
- Se ha comprobado que existen alternativas libres de Sistemas Gestores de Base de Datos que ofrecen las mismas características y buen rendimiento con respecto al software comercial, por lo que MySQL y Firebird se recomienda a empresas que requieren de un sistema con buenas prestaciones, a un bajo costo e inclusive gratis.
- Se recomienda probar nuevas alternativas de software libre para sus desarrollos, porque quizá estas prestan iguales o más funcionalidades que las más conocidas y utilizadas, y sobre todo al alcance de su economía.

## RESUMEN

El objetivo de la investigación fue seleccionar, mediante un análisis comparativo entre los sistemas gestores de bases de datos (SGBD) libres MySQL y Firebird, aquel que permita la mayor optimización de recursos, y posteriormente, utilizarlo como motor de base de datos en el desarrollo de un sistema para la gestión de equipos de radio del Escuadrón de Comunicaciones de la BCB “Galápagos”, de la ciudad de Riobamba.

Se basó en el método científico y se utilizaron Firebird y MySQL, herramientas IBExpert y MySQL GUI Tools, sobre plataforma Windows XP. Los parámetros utilizados para la evaluación en las pruebas realizadas fueron: tiempo de ejecución, integridad en los datos, tolerancia a fallos; además de otros criterios tales como: características, límites que no se vieron involucrados en las mismas pero que son de importancia para la elección de un SGBD.

Se obtuvo un 82.68% para MySQL y para Firebird 100%; en la evaluación de los otros criterios MySQL obtuvo un 94.8% y Firebird un 85.6%. En consecuencia Firebird con el 92.8% ofrece más ventajas al optimizar recursos siendo uno de los más importantes el tiempo de ejecución en realizar los procesos; además de proporcionar funcionalidades importantes como soporte a procedimientos almacenados, triggers, vistas características indispensables que debe poseer un motor de bases de datos.

La aplicación se desarrolló con Firebird y el lenguaje de programación PHP. El sistema sirve de ayuda en la administración de todos los procesos que actualmente se los realiza, mejorando la calidad de servicio.

Se recomienda usar Firebird por sus características y rendimiento especialmente en entornos web en donde los requerimientos de información son más exigentes.

## SUMMARY

The objective of this research was to select a management data base through a comparative analysis between free solicitor data base system (SGBD) MySQL and Firebird, which will permit us the best optimization of the resources and then used it as a management data base for a system development for a radio equipment management of Communication Squad at BCB “Galápagos” in Riobamba city.

It was based on scientific method, and Firebird and MySQL were used, IBExpert and MySQL GUI Tools on XP Windows Platform. The used parameters for evaluation on given tests were: execution time, data safety, and error tolerance; among other criterion such as characteristics, limits which were not involved in themselves, but they are important for the selection of a SGBD.

It was obtained 82.68% for MySQL and 100% for Firebird; on other criterion evaluation 94.8% for MySQL and 85.6% for Firebird. Therefore Firebird with 92.8% offers more advantages upon optimizing resources being one of the most important time execution at putting in practice the processes; besides providing important functionalities such as supporting to store procedures, triggers, sights, indispensable characteristics that a management data base must have.

The application was developed with Firebird and PHP language programming. The system helps for the administration of all the processes that are used today by improving service quality.

It is advisable to use Firebird for its characteristics and performance specially at web surroundings where requirements are more demanding.

## GLOSARIO DE TÉRMINOS

**ACID.** En bases de datos se denomina ACID a un conjunto de características necesarias para que una serie de instrucciones puedan ser consideradas como una transacción.

**Datos.** Valores almacenados que representan hechos o realidades del mundo real.

**Diccionario de Datos.** Esta característica de la base de datos permite obtener fácilmente la información de los objetos que pertenecen a la base de datos como son las tablas, vistas, procedimientos almacenados, etc.

**Funciones.** Las funciones difieren de los procedimientos almacenados en que el propósito de la función es retornar un valor o información de una llamada a rutina o de una consulta.

**Indexado.** Característica de la base de datos que permite mejorar el rendimiento cuando se accesa a una tabla.

**Información.** Significado de los datos.

**Integridad referencial.** Es una propiedad deseable en las bases de datos. Gracias a la integridad referencial se garantiza que una entidad (fila o registro) siempre se relaciona con otras entidades válidas, es decir, que existen en la base de datos. Implica que en todo momento dichos datos sean correctos, sin repeticiones innecesarias, datos perdidos y relaciones mal resueltas.

**Interbase.** Es un Sistema Gestor de Base de Datos Relacional desarrollado y comercializado por la compañía Borland Software Corporation y actualmente desarrollado por su ex-filial CodeGear. Interbase se destaca por su bajo consumo de recursos, su casi nula necesidad de administración y su arquitectura multi-generacional.

**Iteración.** Es una secuencia de actividades dentro de un plan establecido, con unos criterios claros de evaluación, que se organiza con el propósito de entregar parte de la funcionalidad del producto.

**JDBC.** Es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java.

**Lenguaje de definición de datos.** Es el lenguaje proporcionado por el sistema gestor de base de datos para llevar a cabo tareas de definición de estructuras de la base de datos.

**Lenguaje de manipulación de datos.** Permite llevar a cabo consultas y manipulación de los datos.

**ODBC.** Es un estándar de acceso a bases de datos, el objetivo de ODBC es hacer posible el acceder a cualquier dato desde cualquier aplicación, sin importar qué Sistema Gestor de Bases de Datos almacene los datos.

**OLEDB.** Es una tecnología desarrollada por Microsoft usada para tener acceso a diferentes fuentes de información, o bases de datos, de manera uniforme.

**Particionamiento de tablas.** Se trata de una característica de las bases de datos actuales que permite dividir físicamente una tabla de la base de datos en particiones de acuerdo a la lógica del negocio que se necesite, de forma que, reduce significativamente los tiempos de respuesta de la aplicación.

**PHP.** Es un lenguaje de programación interpretado usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web, y últimamente también para la creación de otro tipo de programas incluyendo aplicaciones con interfaz gráfica.

**Sistema de Ficheros.** Es un conjunto de programas informáticos que permiten al usuario almacenar, consultar y modificar datos. Dichos datos se almacenan en ficheros

diseñados para una determinada aplicación. Cada programa define y maneja sus propios datos.

**SMP.** *Symmetric Multi-Processing*, multiproceso simétrico. Se trata de un tipo de arquitectura de ordenadores en que dos o más procesadores comparten una única memoria central.

**SQL.** Es empleado por el usuario para extraer información de la base de datos. El lenguaje de consulta permite al usuario hacer requisiciones de datos sin tener que escribir un programa, usando instrucciones como el SELECT, JOIN.

**Store Procedures.** Un procedimiento almacenado (*stored procedure*) es un programa o procedimiento el cual es almacenado físicamente en una base de datos. La ventaja de un procedimiento almacenado es que al ser ejecutado, en respuesta a una petición de usuario, es ejecutado directamente en el motor de bases de datos, el cual usualmente corre en un servidor separado. Como tal, posee acceso directo a los datos que necesita manipular y sólo necesita enviar sus resultados de regreso al usuario, deshaciéndose de la sobrecarga resultante de comunicar grandes cantidades de datos salientes y entrantes.

**Transacciones.** Una transacción en un Sistema de Gestión de Bases de Datos, es un conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica. Un SGBD se dice transaccional, si es capaz de mantener la integridad de los datos, haciendo que estas transacciones no puedan finalizar en un estado intermedio. Cuando por alguna causa el sistema debe cancelar la transacción, empieza a deshacer las órdenes ejecutadas hasta dejar la base de datos en su estado inicial (llamado punto de integridad), como si la orden de la transacción nunca se hubiese realizado.

**Triggers.** Los triggers o disparadores son objetos que se asocian con tablas y se almacenan en la base de datos. Su nombre se deriva por el comportamiento que presentan en su funcionamiento, ya que se ejecutan cuando sucede algún evento (INSERT, DELETE, UPDATE) sobre las tablas a las que se encuentra asociado.

**Vistas.** Una vista de base de datos es un resultado de una consulta SQL de una o varias tablas, también se le puede considerar una tabla virtual.

## ANEXOS

### Anexo 1: CÓDIGO

#### Conexión a la Base de Datos

Función para conectarse a la base de datos firebird de la aplicación:

```
function      Conectarse($servidor="localhost:C:/BDFire/BDINVENTARIO.FDB",
$usuario="SYSDBA",$password="*****")
{
    $link = ibase_connect($servidor,$usuario,$password);
    if (!$link)
        { exit(); }
    return $link;
}
```

#### Inserción de registros

Una de las partes más importantes de la implementación de un sistema, es la inserción de registros, a continuación se presenta el código utilizado para dicho proceso.

```
<?php
include("conec.php");
$link=Conectarse();
$codigo = $_GET['codigo'];
$login = $_GET['login'];
$contrasena = $_GET['contrasena'];
$ci = $_GET['ci'];
$nombre = $_GET['nombre'];
$apellido = $_GET['apellido'];
$direccion = $_GET['direccion'];
$telefono = $_GET['telefono'];
$rango = $_GET['rango'];
$tipo = $_GET['tipo'];
$sql="insert into USUARIO (COD_USU,LOGIN,CONTRASENA,CI,NOMBRE,
APPELLIDO,DIRECCION,TELEFONO,COD_RAN,TIPO_USU) values
```

```
(' $codigo', '$login', '$contrasena', '$ci', '$nombre', '$apellido', '$direccion', '$telefono', '$rango', '$tipo)";  
ibase_query($link,$Sql);  
?>
```

### **Eliminación de registros**

```
<?php  
include("conec.php");  
$link=Conectarse();  
$codigo = $_POST['codigo'];  
ibase_query($link,"delete from USUARIO where COD_USU='$codigo'");  
?>
```

### **Actualización de registros**

```
<?php  
include("conec.php");  
$link=Conectarse();  
$codigo = $_POST['codigo'];  
$login = $_POST['login'];  
$contrasena = $_POST['contrasena'];  
$ci = $_POST['ci'];  
$nombre = $_POST['nombre'];  
$apellido = $_POST['apellido'];  
$direccion = $_POST['direccion'];  
$telefono = $_POST['telefono'];  
$rango = $_POST['rango'];  
$tipo = $_POST['tipo'];  
$Sql="UPDATE USUARIO SET COD_USU='$codigo',LOGIN='$login',  
CONTRASENA='$contrasena',CI='$ci',NOMBRE='$nombre',APELLIDO='$apellido',  
DIRECCION='$direccion',TELEFONO='$telefono',COD_RAN='$rango',TIPO_USU='  
$tipo' WHERE COD_USU='$codigo';  
ibase_query($link,$Sql); ?>
```

## Generación de consultas

Con la finalidad de permitir a los usuarios visualizar la información deseada, se implementa varias opciones para realizar diferentes consultas, a continuación se muestra el código para listar registros de la base de datos.

```
<?php
include("conec.php");
$link=Conectarse();
$result=ibase_query($link,"select * from USUARIO");
?>
<table border=0 width="100%" cellspacing=1 cellpadding=1>
  <tr>
    <td>Código</td>
    <td>CI</td>
    <td>Nombres</td>
    <td>Apellidos</td>
    <td>Dirección</td>
    <td>Teléfono</td>
  </tr>
  <?php
  while($row = ibase_fetch_object($result)) {
  printf("<tr><td>&nbsp;%s&nbsp;</td><td>&nbsp;%s&nbsp;</td><td>&nbsp;%s&nbsp;</td><td>&nbsp;%s&nbsp;</td><td>&nbsp;%s&nbsp;</td><td>&nbsp;%s&nbsp;</td></tr>",
  $row->COD_USU,$row->CI,$row->NOMBRE,$row->APELLIDO,$row->DIRECCION,$row->TELEFONO);
  }
  ibase_free_result($result); ?>
</table>
```

## **Anexo 2: MANUAL DE USUARIO**

### **PRESENTACIÓN**

El presente manual va enfocado a nivel de usuario. Describe la funcionalidad y el uso del Sistema para la Gestión de Equipos de radio del Escuadrón de Comunicaciones N°11.

Este manual contiene información completa del manejo correcto del sistema, por lo que le recomendamos leer detenida y completamente su contenido. Muestra la manera más eficiente de manejarlo ya que presenta toda la información con el fin de que los diferentes tipos de usuarios puedan tener una completa visión del software antes de iniciar con su funcionamiento.

### **INTRODUCCIÓN**

El Sistema para la Gestión de Equipos de radio, ha sido desarrollado con el objetivo de mejorar la calidad de servicio en la prestación de equipos y materiales de radio al personal perteneciente al Escuadrón de Comunicaciones N°11, además todos los usuarios podrán realizar consultas sobre los préstamos que han realizado, o la actualización de su información.

#### **Objetivos del documento**

- Utilizar como una guía operativa que permita la orientación de los usuarios para el uso del sistema, con el fin de validar su correcto manejo.

- Permitir conocer en forma detallada el manejo de cada una de las opciones con las que cuenta el sistema.

### **REQUISITOS MÍNIMOS INDISPENSABLES**

Usted solamente necesita un computador con buenas características y el acceso a la red de la institución para obtener acceso al Sistema para la Gestión de Equipos de radio. A continuación se mencionan los requisitos mínimos indispensables del equipo:

#### **Hardware:**

- PC con procesador Pentium III o superior.
- 512 MB de memoria RAM.
- Disco duro de 40 Gb
- Tarjeta de Red Fast Ethernet 100Mbps (configurada para acceder a la red institucional)
- Monitor
- Impresora

#### **Software:**

- Sistema operativo Microsoft Windows XP Professional
- Internet Explorer (Browser)

### **COMO USAR EL PROGRAMA / MI PRIMERA SESIÓN**

Para empezar a utilizar el sistema, y por ser una Aplicación Web es necesaria la instalación de un browser para que el usuario pueda navegar y visualizar el contenido y resultado del sistema. El browser o navegador viene incluido en nuestra versión de

Windows, en caso de no existir una se puede utilizar cualquier navegador disponible para su descarga desde Internet, por ejemplo:

- Mozilla Firefox
- Netscape
- Microsoft Internet Explorer, etc.

## **ESTRUCTURA DEL SISTEMA**

### **PÁGINA PRINCIPAL**

Una vez ingresada a la pantalla principal del sistema se deberá autenticarse con el Login y Password para posteriormente ingresar hacer uso del sistema; la aplicación presenta dos niveles de administración que permitirá la gestión de funciones tanto para el usuario administrador como para el personal.



AUTENTICACION DE USUARIOS

Iniciar sesión

Login: joselito

Password: ●●●●●●●●●●

Entrar

Los usuarios iniciarán su respectiva sesión luego de ingresar su login y password correspondientes asignadas por el administrador al momento de crear un usuario.

En el caso de que el nombre de usuario y password ingresados sean incorrectos, el sistema no le permitirá ingresar hasta que sus datos sean los correctos. Si los datos son correctos se presentará una pantalla principal dependiendo del usuario con que se ha logeado.

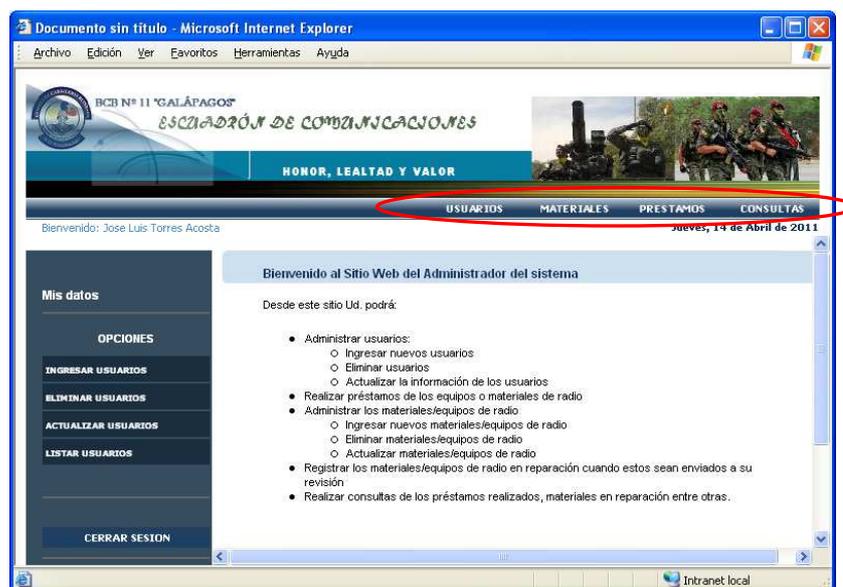
A continuación presentamos las opciones que pueden realizar cada uno de los usuarios que tienen acceso al Sistema para la Gestión de Equipos de Radio:

## ADMINISTRADOR

Es el usuario principal del sistema entre las funcionalidades que este usuario puede realizar se encuentran: La administración de los usuarios que manejarán la aplicación como son el ingreso, eliminación, actualización y listado de usuarios. La administración de los equipos y materiales de radio (ingreso, eliminación, actualización y listado). La realización de los préstamos de equipos y materiales; además de diferentes tipos de consultas que se detallarán más adelante.

## PÁGINA PRINCIPAL DEL ADMINISTRADOR

La pantalla principal del administrador consta de un menú con las funcionalidades que se mencionó anteriormente, además contiene un enlace (Mis datos) con el que puede ver su información personal y actualizarlo en el caso que lo desee.



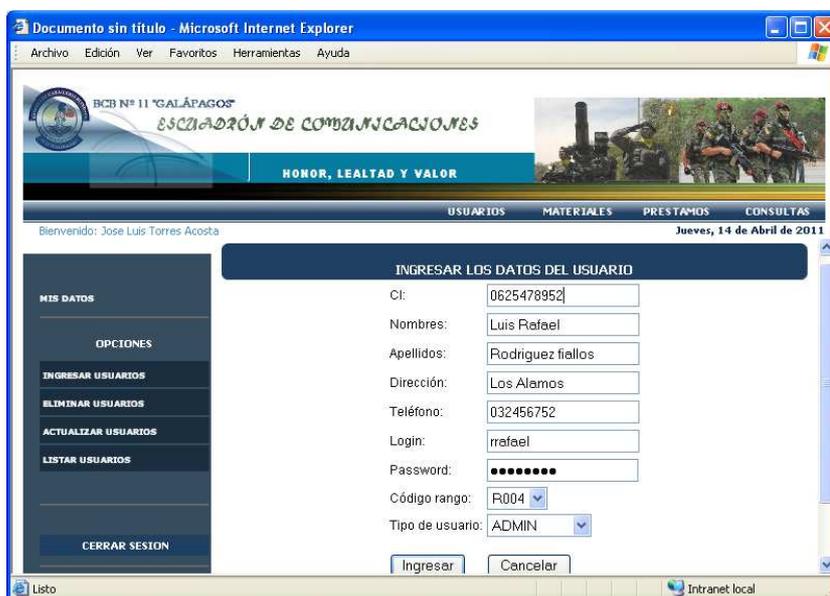
## ADMINISTRACIÓN DE USUARIOS

Para acceder a cada una de las opciones que un usuario administrador puede realizar con respecto a la administración de usuarios se debe dar clic en el enlace con la opción deseada.



### Ingresar usuarios

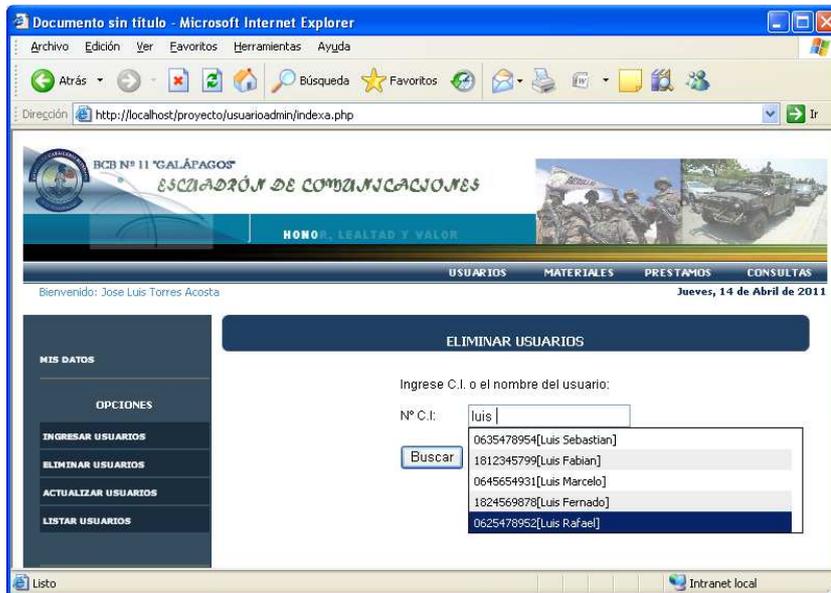
Para el ingreso de un nuevo usuario seleccionar la opción *INGRESAR USUARIOS*, se presentara la página con un formulario y se rellenaran los datos del nuevo usuario. A continuación presionar en el botón *Ingresar*.

Una captura de pantalla de un navegador web que muestra una interfaz de administración de usuarios. El navegador es Microsoft Internet Explorer. La página tiene un encabezado con el logo de "BCB Nº 11 'GALÁPAGOS' ESCUADRA DE COMUNICACIONES" y el lema "HONOR, LEALTAD Y VALOR". Hay una barra de navegación con "USUARIOS", "MATERIALES", "PRESTAMOS" y "CONSULTAS". El usuario está logueado como "Jose Luis Torres Acosta" el día "Jueves, 14 de Abril de 2011". A la izquierda hay un menú con "OPCIONES" y "INGRESAR USUARIOS" seleccionado. El formulario principal se titula "INGRESAR LOS DATOS DEL USUARIO" y contiene los siguientes campos: "Ci:" con el valor "0625478952"; "Nombres:" con "Luis Rafael"; "Apellidos:" con "Rodriguez fiallos"; "Dirección:" con "Los Alamos"; "Teléfono:" con "032456752"; "Login:" con "lrafael"; "Password:" con caracteres ocultos por puntos; "Código rango:" con un menú desplegable que muestra "R004"; "Tipo de usuario:" con un menú desplegable que muestra "ADMIN". Al final del formulario hay dos botones: "Ingresar" y "Cancelar".

Si el usuario no existe y los datos ingresados son correctos el sistema desplegará un mensaje "*Usuario ingresado*" indicando que el usuario fue ingresado con éxito. Cuando se va a ingresar un usuario y este ya existe o los datos ingresados son incorrectos, se indicará mediante un mensaje que ese usuario no puede ingresarse.

## Eliminar usuarios

Para la eliminación de un usuario se seleccionará la opción *ELIMINAR USUARIOS*, en la página que se presenta ingresar la cédula de identidad o el nombre del usuario y enviar a *Buscar*. Esto se aplicará a todas las búsquedas que se realicen.



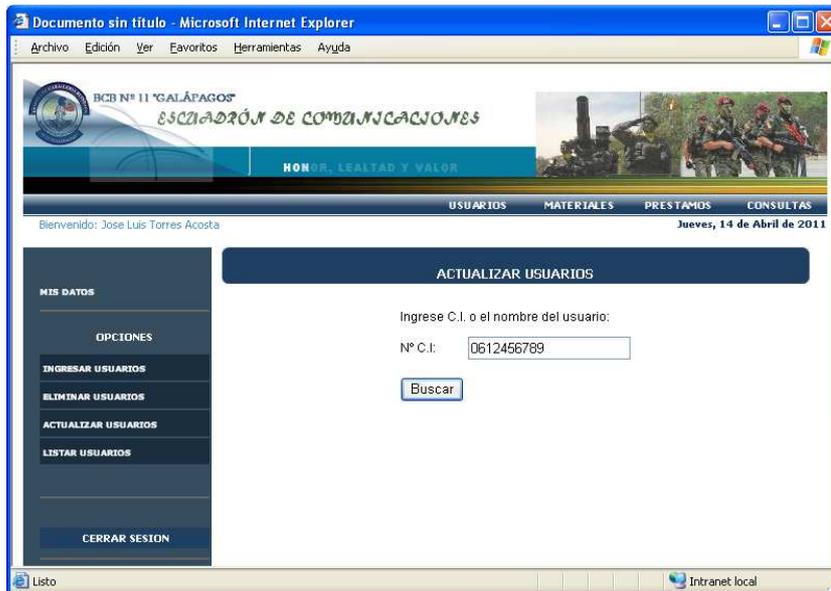
Si el usuario existe se mostrará todos sus datos y para eliminarlo dar clic en el botón *Eliminar*.



La eliminación se confirmará con el mensaje *“Usuario eliminado”*. Si el usuario no existe se indicara mediante un mensaje *“El usuario no existe”*.

## Actualizar usuarios

Si su opción es *ACTUALIZAR USUARIOS* al igual que la opción anterior se debe ingresar la cédula de identidad del usuario y enviar a *Buscar* para realizar las respectivas actualizaciones de los datos en el caso de que existiera.

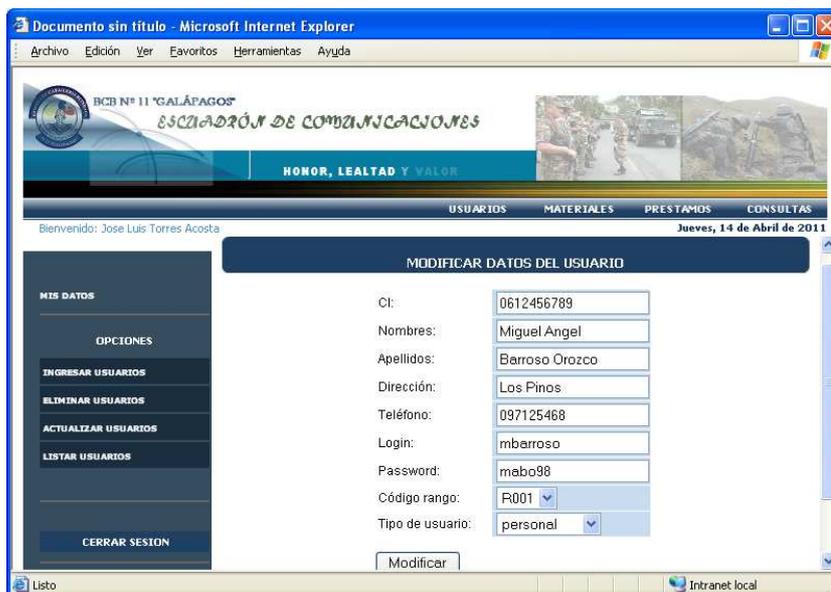


The screenshot shows a web browser window titled 'Documento sin título - Microsoft Internet Explorer'. The page header includes the logo of 'BCB Nº 11 "GALÁPAGOS" ESCUADRA DE COMUNICACIONES' and the motto 'HONOR, LEALTAD Y VALOR'. A navigation bar contains 'USUARIOS', 'MATERIALES', 'PRESTAMOS', and 'CONSULTAS'. The user is logged in as 'Jose Luis Torres Acosta' on 'Jueves, 14 de Abril de 2011'. The main content area is titled 'ACTUALIZAR USUARIOS' and contains a form with the following fields and buttons:

- Text prompt: 'Ingrese C.I. o el nombre del usuario:'
- Text input field: 'N° C.I.: 0612456789'
- Submit button: 'Buscar'

A sidebar on the left lists navigation options: 'MIS DATOS', 'OPCIONES', 'INGRESAR USUARIOS', 'ELIMINAR USUARIOS', 'ACTUALIZAR USUARIOS', 'LISTAR USUARIOS', and 'CERRAR SESION'.

Si el usuario existe se muestran los datos y se editarán los campos que se deseen actualizar. Dar clic en el botón *Modificar* para confirmar. Y si el usuario no existe se indicara el mensaje “*El usuario no existe*”.



The screenshot shows the same web browser window, but the form is now titled 'MODIFICAR DATOS DEL USUARIO'. The form contains the following fields and buttons:

- Text input field: 'Ci: 0612456789'
- Text input field: 'Nombres: Miguel Angel'
- Text input field: 'Apellidos: Barroso Orozco'
- Text input field: 'Dirección: Los Pinos'
- Text input field: 'Teléfono: 097125468'
- Text input field: 'Login: mbarroso'
- Text input field: 'Password: mabo98'
- Text input field: 'Código rango: R001' (with a dropdown arrow)
- Text input field: 'Tipo de usuario: personal' (with a dropdown arrow)
- Submit button: 'Modificar'

The sidebar on the left remains the same as in the previous screenshot.

Si los datos del usuario se actualizaron correctamente se visualizará el siguiente mensaje: “*El usuario ha sido actualizado*”.

## Listar usuarios

Si selecciona la opción *LISTAR USUARIOS* permitirá consultar los datos de todos los usuarios existentes en el sistema.



The screenshot shows a web browser window with the title "Documento sin título - Microsoft Internet Explorer". The page header includes the logo of "BCB N° 11 'GALÁPAGOS' ESCUADRÓN DE COMUNICACIONES" and the motto "HONOR, LEALTAD Y VALOR". The navigation menu contains "USUARIOS", "MATERIALES", "PRESTAMOS", and "CONSULTAS". The user is logged in as "Jose Luis Torres Acosta" on "Jueves, 14 de Abril de 2011". The main content area is titled "LISTAR USUARIOS" and displays a table of user data.

CI	Nombres	Apellidos	Dirección	Teléfono
0632548794	Pedro Manuel	Abad Solis	Los Pinos	03215469
0635462356	Fabrizio Javier	Cepeda L	La Merced	032156423
0632165488	Rafael Felipe	Lopez Ramirez	Loma de Quito	093654215
1856987122	Juan Andres	Muñoz Paredes	Los Pinos	999999999
0635421366	Danilo Fabian	Garces Perez	Condominios Chimborazo	084215789
0635262344	Victor Daniel	Izurieta Solis	Los Pinos	098456231
0612456789	Miguel Angel	Barroso Orozco	Los Pinos	097125468
0632145687	Pedro Javier	Olmedo Acosta	Loma de Quito	095147562
1845784126	Esteban Samir	Robles Castelo	Bellavista	032457894
0632569874	Marco Patricio	Yeppez Molina	La Ceramica	098456325
0635478954	Luis Sebastian	Siza Tenelanda	BCB	098756425

## ADMINISTRACIÓN DE EQUIPOS DE RADIO

Para la gestión de los equipos y materiales de radio seleccionamos en el menú superior la opción *MATERIALES*, una vez seleccionado se presentan las opciones que un usuario administrador puede realizar, se debe dar clic en el enlace con la opción deseada.



The screenshot shows a vertical menu titled "OPCIONES" with the following items: "INGRESAR EQUIPO/MATERIAL", "ELIMINAR EQUIPO/MATERIAL", "ACTUALIZAR EQUIPO/MATERIAL", "LISTAR EQUIPO/MATERIAL", "REPARACION", and "ENTREGAR DE REPARACION".

## Ingresar equipo/material

Al seleccionar *INGRESAR EQUIPO/MATERIAL* aparece una pantalla con un formulario en el que debemos llenar los siguientes datos: código, nombre, modelo del equipo o material de radio, estado y fecha. Presionamos en el botón *Ingresar*. (*estado*: permite almacenar el estado en que se encuentra un equipo/material este puede ser Disponible (cuando se ingresa un nuevo equipo/material), Prestado (cuando se realiza un préstamo), Reparación (cuando se envía a reparación); *fecha*: es la fecha ya sea de cuando se ingresa un nuevo equipo/material, o la fecha de entrega de un equipo/material que ha sido prestado o a estado en reparación, normalmente es la fecha actual).



The screenshot shows a web browser window titled "Documento sin título - Microsoft Internet Explorer". The page header includes the logo of "BCB Nº 11 'GALÁPAGOS' ESCUADRON DE COMUNICACIONES" and the motto "HONOR, LEALTAD Y VALOR". Below the header, there are navigation tabs for "USUARIOS", "MATERIALES", "PRESTAMOS", and "CONSULTAS". The user is logged in as "Jose Luis Torres Acosta" on "Lunes, 29 de Marzo de 2010".

The main content area is titled "INGRESAR LOS DATOS DEL EQUIPO O MATERIAL DE RADIO" and contains the following form fields:

- Código: M011
- Nombre: Material 11
- Modelo: LRU01
- Estado: Disponible (dropdown menu)
- FechaE: 2011/04/14

At the bottom of the form are two buttons: "Ingresar" and "Cancelar". On the left side, there is a sidebar menu with options: "MIS DATOS", "OPCIONES", "INGRESAR EQUIPO/MATERIAL", "ELIMINAR EQUIPO/MATERIAL", "ACTUALIZAR EQUIPO/MATERIAL", "LISTAR EQUIPO/MATERIAL", "REPARACION", "ENTREGAR DE REPARACION", and "CERRAR SESION".

Si el código del equipo de radio no existe y los datos ingresados son correctos el sistema desplegará un mensaje indicando que fue ingresado con éxito. En el caso que ya exista no permitirá su ingreso.

## Eliminar equipo/material

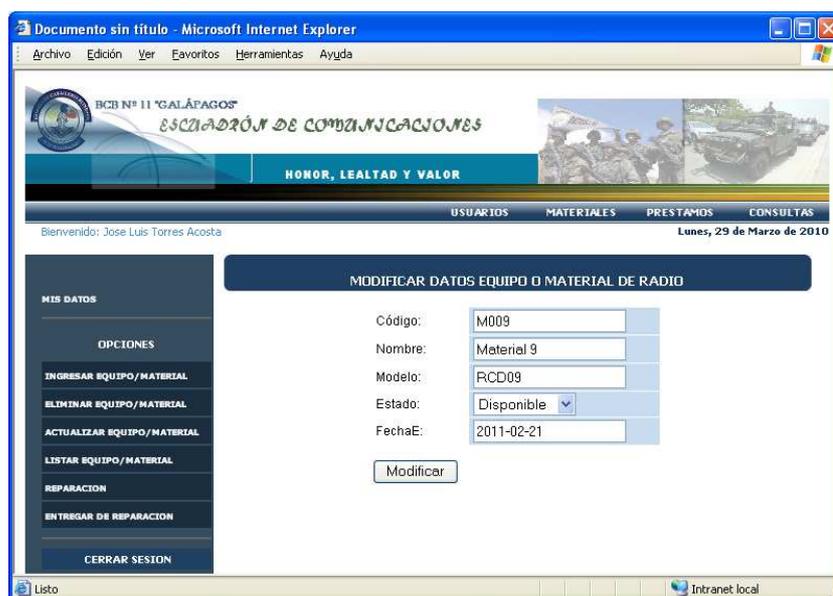
Si escoge la opción *ELIMINAR EQUIPO/MATERIAL* se podrá eliminar un equipo o material de radio, para ello debe ingresar el código y enviar a *Buscar*.



Se visualizarán todos sus datos en el caso de que exista y para eliminarlo pulsamos en el botón *Eliminar*. La eliminación se confirmará con el mensaje “*Se ha realizado la eliminación con éxito*”. Si no existe el código del equipo o material se desplegará un mensaje indicando que “*No existe*”.

### Actualizar equipo/material

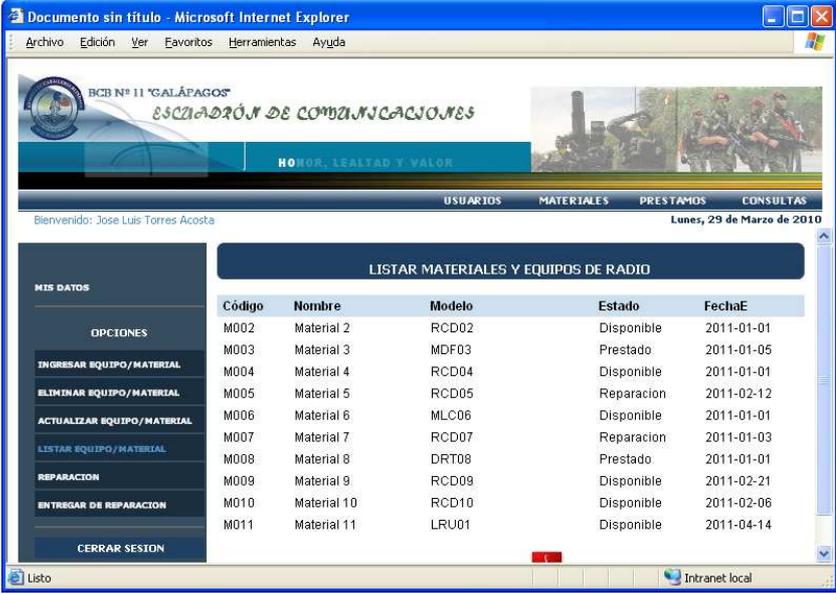
Al igual que en la opción anterior se debe ingresar el código para luego enviar a buscar.



Se visualizarán los datos en un formulario y se editarán los campos que se deseen actualizar. A continuación presionamos en *Modificar*.

## Listar equipo/material

Si selecciona la opción *LISTAR EQUIPO/MATERIAL* permitirá consultar los datos de todos los equipos y materiales de radio existentes.



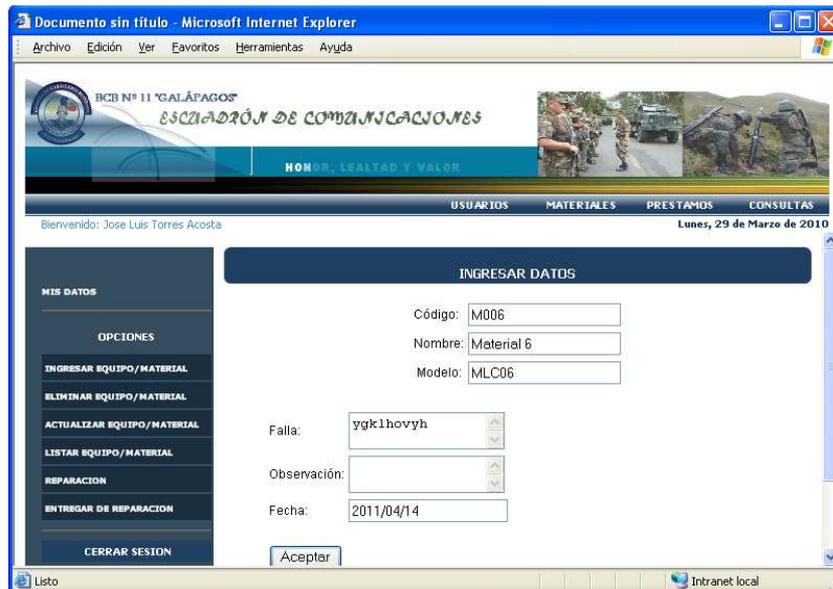
The screenshot shows a web browser window with the title "Documento sin título - Microsoft Internet Explorer". The page header includes the logo of "BCB Nº 11 'GALÁPAGOS' ESCUADRÓN DE COMUNICACIONES" and the motto "HONOR, LEALTAD Y VALOR". The navigation menu contains "USUARIOS", "MATERIALES", "PRESTAMOS", and "CONSULTAS". The main content area is titled "LISTAR MATERIALES Y EQUIPOS DE RADIO" and displays a table with the following data:

Código	Nombre	Modelo	Estado	FechaE
M002	Material 2	RCD02	Disponible	2011-01-01
M003	Material 3	MDF03	Prestado	2011-01-05
M004	Material 4	RCD04	Disponible	2011-01-01
M005	Material 5	RCD05	Reparacion	2011-02-12
M006	Material 6	MLC06	Disponible	2011-01-01
M007	Material 7	RCD07	Reparacion	2011-01-03
M008	Material 8	DRT08	Prestado	2011-01-01
M009	Material 9	RCD09	Disponible	2011-02-21
M010	Material 10	RCD10	Disponibile	2011-02-06
M011	Material 11	LRU01	Disponibile	2011-04-14

## Reparación

Otra de las opciones en la gestión de equipos y materiales de radio que se debe realizar es: cuando un equipo o material se encuentra dañado este es enviado a reparación, para lo cual debemos registrarlo indicando falla, observación y fecha del equipo que se encuentra en reparación.

Para ingresar a esta opción seleccionamos el enlace *REPARACION* se presentará una pantalla en la que debemos ingresar el código del equipo o material y enviar a buscar.



Se visualizarán sus datos en el caso de que exista, además se debe ingresar otros datos como son falla, observación y fecha desde la cual el equipo o material se encuentra en reparación. Presionamos en el botón *Aceptar*.

Si el código del equipo o material no existe, o ya se encuentra en reparación se desplegará un mensaje indicando lo sucedido.

### **Entregar de reparación**

Una vez que el equipo o material sea reparado este debe ser registrado como Disponible para que pueda ser prestado. Para ello seleccionamos la opción *ENTREGAR DE REPARACION* ingresaremos el código para enviar a buscar.

Si el equipo o material se encuentra en reparación se mostrarán el código, fecha desde la cual se encuentra en reparación, falla y observación. Presionamos en *Aceptar* y se actualizará automáticamente los datos con la fecha actual y el estado de material como Disponible, para que este pueda ser prestado.



En el caso que el código no exista o no se encuentre en reparación se visualizará un mensaje indicando lo ocurrido.

## **PRÉSTAMOS DE EQUIPOS DE RADIO**

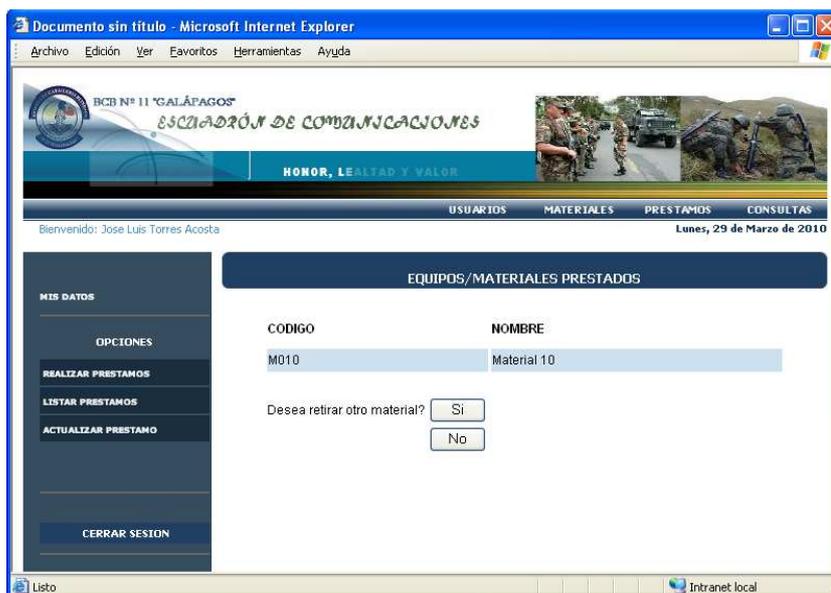
Para realizar préstamos de los equipos y materiales de radio seleccionamos en el menú superior la opción *PRESTAMOS* una vez seleccionado se presentará cada una de las opciones que un usuario administrador puede realizar, se debe dar clic en el enlace con la opción deseada.

### **Realizar préstamos**

Si selecciona la opción *REALIZAR PRESTAMOS*, se presentará la página con un formulario se rellenarán los campos con el número de préstamo, cédula de identidad del usuario, el campo fecha ya se encuentra con la fecha actual. A continuación presionar en el botón *Aceptar*.

Si los datos ingresados son correctos (número de préstamo no existe y el usuario existe) se presentará un formulario en el cual debe ingresar el código del equipo o material para su respectiva prestación. Damos clic en el botón *Prestar*.

Si el equipo o material de radio existe y está disponible, será prestado. Se puede realizar préstamos de otros materiales o equipos si presionamos en el botón *Si*, de lo contrario seleccionamos *No*.



En nuestro caso seleccionamos la opción *Si* para realizar el préstamo de otro equipo/material. Ingresar el código y presionar en el botón *Prestar*.



Si el equipo o material de radio existe y está disponible, será prestado.



## Listar préstamos

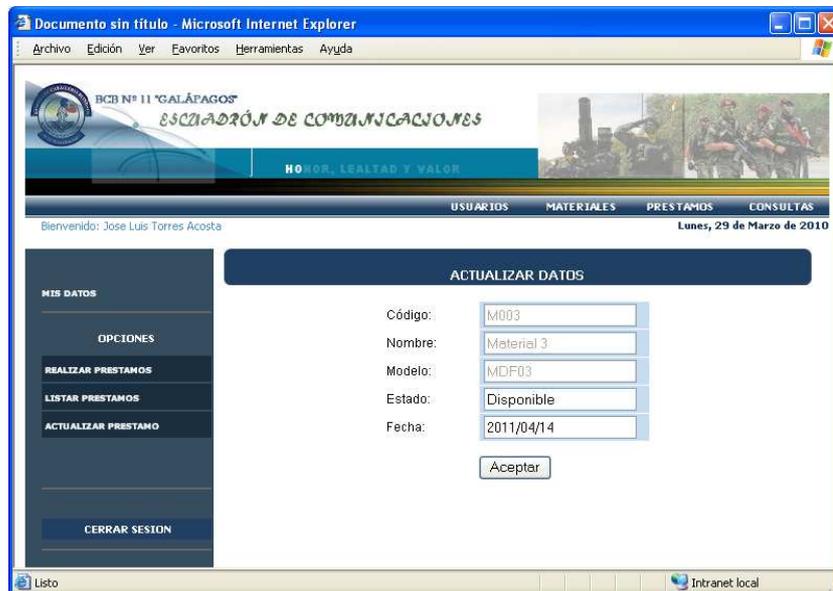
Si selecciona la opción *LISTAR PRESTAMOS* permitirá consultar el número de préstamo, la fecha y los datos de quién realizó dichos préstamos.



## Actualizar préstamos

Cuando un equipo o material de radio es prestado este debe ser devuelto, para ello existe la opción *ACTUALIZAR PRESTAMOS* la misma que nos permite actualizar los datos del material o equipo con la fecha actual y el estado como Disponible. Para lo cual

debemos ingresar el código y enviar a buscar, si existe se mostrarán los datos, presionamos en *Aceptar* para que se actualicen.



## CONSULTAS Y REPORTE

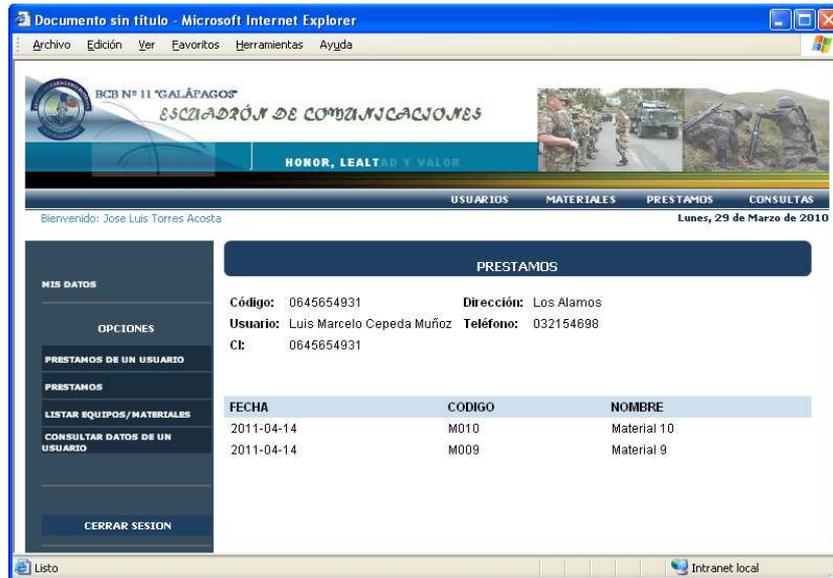
El usuario administrador podrá realizar diferentes consultas como: préstamos realizados por un usuario, consultar datos de un cierto usuario, consultar los equipos o materiales de radio disponibles, en reparación, prestados, listar todos los materiales existentes, además podrá generar reportes de algunas de estas consultas realizadas. Para acceder a estas opciones seleccionamos en el menú superior la opción *CONSULTAS*.



### Préstamos de un usuario

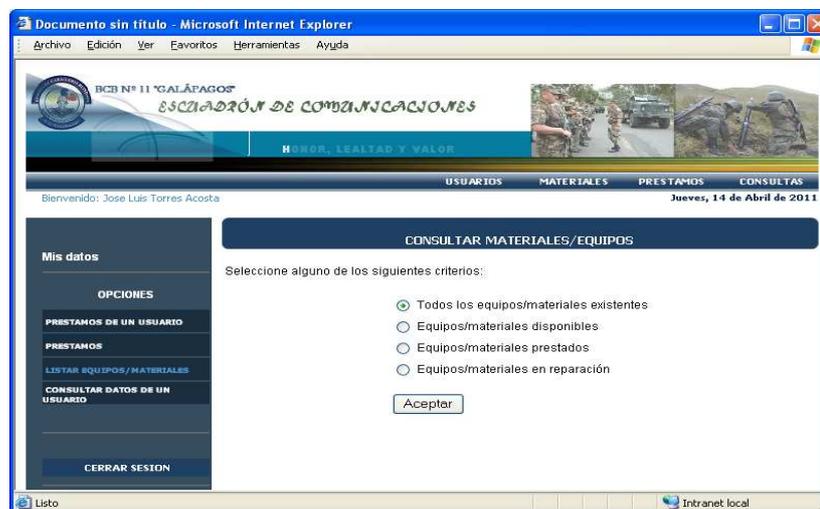
Al seleccionar esta opción se podrá consultar los préstamos realizados por un usuario para lo cual se deberá ingresar el código de usuario y enviar a *Buscar*. Se mostrarán

todos los préstamos realizados por el usuario en el caso de que existan. De lo contrario se visualizará un mensaje indicando que el usuario no tiene ningún préstamo o que el usuario no existe.



## Listar equipos/materiales

Para consultar los equipos o materiales de radio existentes, disponibles para realizar préstamos, en reparación o prestados, seleccionamos la opción *LISTAR EQUIPOS/MATERIALES* se presentará una pantalla en la cual debemos seleccionar alguno de los criterios de consulta y presionar en el botón *Aceptar*.



- Si selecciona, *Todos los equipos/materiales existentes*

Se visualizarán los datos de todos los equipos o materiales de radio existentes como son: código, nombre, modelo, estado y fecha desde la cual se encuentra en inventario.

Código	Nombre	Modelo	Estado	FechaE
M002	Material 2	RCD02	Disponible	2011-01-01
M003	Material 3	MDF03	Prestado	2011-01-05
M004	Material 4	RCD04	Disponible	2011-01-01
M005	Material 5	RCD05	Reparacion	2011-02-12
M006	Material 6	MLC06	Reparacion	2011-01-01
M007	Material 7	RCD07	Reparacion	2011-01-03
M008	Material 8	DRT08	Prestado	2011-01-01
M009	Material 9	RCD09	Prestado	2011-02-21
M010	Material 10	RCD10	Prestado	2011-02-06
M011	Material 11	LRU01	Disponible	2011-04-14

Además se podrá generar un reporte de esta consulta en formato .PDF, para ello al final de la consulta se encuentra un enlace “*Generar PDF*”.

- Si selecciona, *Equipos/materiales disponibles*

Se mostrará un listado con el código, nombre y modelo de los equipos o materiales de radio disponibles para realizar préstamos.

Código	Nombre	Modelo
M002	Material 2	RCD02
M004	Material 4	RCD04
M011	Material 11	LRU01

[Generar PDF](#) 

Al igual que en la opción anterior se podrá generar el reporte de esta consulta en formato .PDF, para ello presionamos en el enlace “*Generar PDF*”.

- Si selecciona, *Equipos/materiales prestados*

Se generará un listado con el código y nombre del equipo o material, además del usuario y la fecha desde la cual el equipo o material se encuentra prestado.



The screenshot shows a web browser window with the title "Documento sin título - Microsoft Internet Explorer". The page header includes the logo of "BCB Nº 11 'GALÁPAGOS' ESCUADRÓN DE COMUNICACIONES" and the motto "HONOR, LEALTAD Y VALOR". The user is logged in as "Jose Luis Torres Acosta" on "Lunes, 29 de Marzo de 2010". The main content area is titled "MATERIALES Y EQUIPOS DE RADIO PRESTADOS" and contains a table with the following data:

Código	Nombre	Fecha	Usuario
M003	Material 3	2011-01-05	Pedro Javier Olmedo Acosta
M008	Material 8	2011-01-01	Edgar Manuel Arnaguaya Lopez
M009	Material 9	2011-02-21	Luis Marcelo Cepeda Muñoz
M010	Material 10	2011-02-06	Luis Marcelo Cepeda Muñoz

Below the table is a link labeled "Generar PDF" with a PDF icon.

Generaremos un reporte de la consulta si seleccionamos “*Generar PDF*”.

- Si selecciona, *Equipos/materiales en reparación*

Se podrá visualizar un listado de los equipos o materiales de radio que se encuentran en reparación. Se mostrará el código, nombre, modelo, falla y fecha desde la cual se encuentra en reparación.

Documento sin título - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

BCB Nº 11 "GALÁPAGOS" ESCUADRÓN DE COMUNICACIONES

HONOR, LEALTAD Y VALOR

Bienvenido: Jose Luis Torres Acosta Lunes, 29 de Marzo de 2010

USUARIOS MATERIALES PRESTAMOS CONSULTAS

MIS DATOS

OPCIONES

PRESTAMOS DE UN USUARIO

PRESTAMOS

LISTAR EQUIPOS/MATERIALES

CONSULTAR DATOS DE UN USUARIO

CERRAR SESION

MATERIALES EN REPARACION

CODIGO	NOMBRE	MODELO	FECHA	FALLA
M005	Material 5	RCD05	2011-04-14	xdgvfv
M007	Material 7	RCD07	2011-04-14	sgvxdvb
M006	Material 6	MLC06	2011-04-14	yglkhovyh

Generar PDF

Listo Jueves, 14 de Abril de 2011

Para generar un reporte de la consulta seleccionamos “Generar PDF”.

### Consultar datos de un usuario

Al seleccionar esta opción se podrá consultar todos los datos de un usuario para ello ingresamos la cédula de identidad del usuario y presionamos en *Buscar*, si existe se mostrará la información de lo contrario se desplegará un mensaje indicando que el usuario no existe.

Documento sin título - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

BCB Nº 11 "GALÁPAGOS" ESCUADRÓN DE COMUNICACIONES

HONOR, LEALTAD Y VALOR

Bienvenido: Jose Luis Torres Acosta Lunes, 29 de Marzo de 2010

USUARIOS MATERIALES PRESTAMOS CONSULTAS

MIS DATOS

OPCIONES

PRESTAMOS DE UN USUARIO

PRESTAMOS

LISTAR EQUIPOS/MATERIALES

CONSULTAR DATOS DE UN USUARIO

CERRAR SESION

DATOS DEL USUARIO

Ci:	0632569874
Nombres:	Marco Patricio
Apellidos:	Yepez Molina
Dirección:	La Cerámica
Teléfono:	098456325
Login:	myepez
Password:	mpy76
Código rango:	R003
Tipo de usuario:	personal

Listo Intranet local

## MIS DATOS

Al dar clic sobre la opción *Mis Datos* podemos ver que se muestra los datos actuales del administrador los mismos que pueden ser cambiados o modificados si editamos los campos y damos clic en *Aceptar*, caso contrario solo podemos visualizarlos.



Documento sin título - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

BCB Nº 11 "GALÁPAGOS" ESCUADRÓN DE COMUNICACIONES  
HONOR, LEALTAD Y VALOR

Bienvenido: Jose Luis Torres Acosta Lunes, 29 de Marzo de 2010

USUARIOS MATERIALES PRESTAMOS CONSULTAS

**MIS DATOS**

CI:	0632578945
Nombres:	Jose Luis
Apellidos:	Torres Acosta
Dirección:	Los Pinos
Teléfono:	032451203
Login:	josecito
Password:	josecito
Código rango:	R003
Tipo de usuario:	ADMIN

OPCIONES

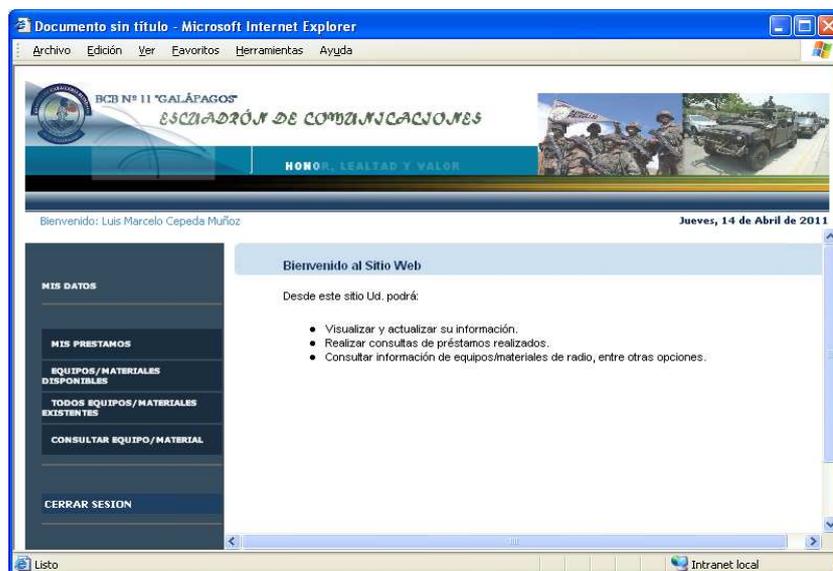
- PRESTAMOS DE UN USUARIO
- PRESTAMOS
- LISTAR EQUIPOS/MATERIALES
- CONSULTAR DATOS DE UN USUARIO
- CERRAR SESION

Listo Intranet local

## PERSONAL

Este tipo de usuario podrá acceder una vez que se haya autenticado correctamente en la página de autenticación de usuarios, para esto deberá ingresar su login y password.

## Página principal



Documento sin título - Microsoft Internet Explorer

Archivo Edición Ver Favoritos Herramientas Ayuda

BCB Nº 11 "GALÁPAGOS" ESCUADRÓN DE COMUNICACIONES  
HONOR, LEALTAD Y VALOR

Bienvenido: Luis Marcelo Cepeda Muñoz Jueves, 14 de Abril de 2011

Bienvenido al Sitio Web

Desde este sitio Ud. podrá:

- Visualizar y actualizar su información.
- Realizar consultas de préstamos realizados.
- Consultar información de equipos/materiales de radio, entre otras opciones.

MIS DATOS

- MIS PRESTAMOS
- EQUIPOS/MATERIALES DISPONIBLES
- TODOS EQUIPOS/MATERIALES EXISTENTES
- CONSULTAR EQUIPO/MATERIAL
- CERRAR SESION

Listo Intranet local

Entre las opciones a las cuales tiene acceso este tipo de usuario se encuentran: consultar mis préstamos, consultar equipos/materiales disponibles, consultar la información de un cierto equipo/material. Además podrá visualizar su información personal a través de la opción *Mis datos* y actualizarla si lo desea. Para acceder a una de ellas debe dar clic en el enlace con la opción deseada.



## Mis datos

Al seleccionar la opción *Mis Datos* se muestra la información del usuario actual los mismos que pueden ser cambiados o modificados si editamos los campos y damos clic en *Aceptar*, caso contrario solo podemos visualizarlos.

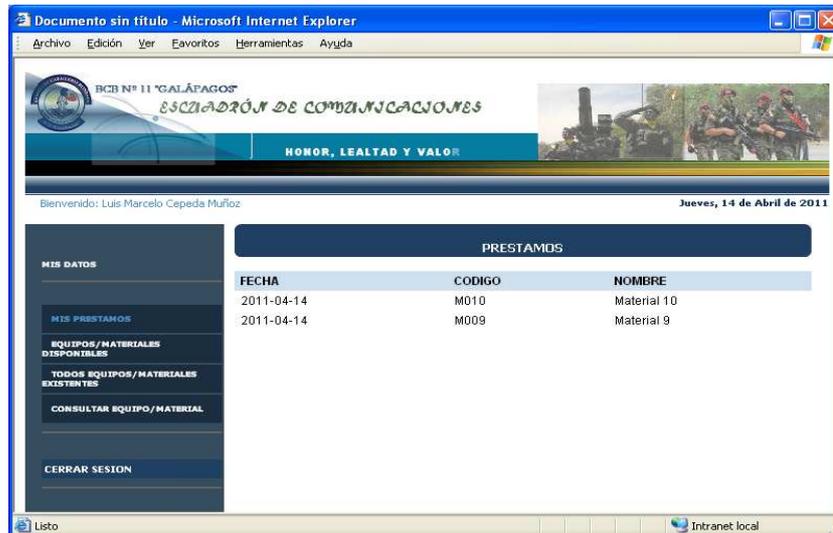
The screenshot shows a web browser window with the title "Documento sin título - Microsoft Internet Explorer". The browser's address bar and menu bar are visible. The page content includes a header for "BCB Nº 11 'GALÁPAGOS' ESCUADRÓN DE COMUNICACIONES" with the motto "HONOR, LEALTAD Y VALOR". Below the header, it says "Bienvenido: Luis Marcelo Cepeda Muñoz" and "Jueves, 14 de Abril de 2011". The main content area is titled "MIS DATOS" and contains a form with the following fields:

Ci:	0645654931
Nombres:	Luis Marcelo
Apellidos:	Cepeda Muñoz
Dirección:	Los Alamos
Teléfono:	032154698
Login:	lmcepeda
Password:	clm1976
Código rango:	R004
Tipo de usuario:	personal

At the bottom of the browser window, the status bar shows "Listo" and "Intranet local".

## Mis préstamos

Esta opción permite consultar los préstamos que actualmente tiene el usuario que ha iniciado sesión. Para ello simplemente seleccionamos la opción *Mis Préstamos* y se visualizará los préstamos que el usuario actual tiene.



The screenshot shows a web browser window with the title 'Documento sin título - Microsoft Internet Explorer'. The page header includes the logo of 'BCB Nº 11 GALÁPAGOS ESCUADRÓN DE COMUNICACIONES' and the motto 'HONOR, LEALTAD Y VALOR'. Below the header, it says 'Bienvenido: Luis Marcelo Cepeda Muñoz' and 'Jueves, 14 de Abril de 2011'. The main content area is titled 'PRESTAMOS' and contains a table with the following data:

FECHA	CODIGO	NOMBRE
2011-04-14	M010	Material 10
2011-04-14	M009	Material 9

On the left side, there is a navigation menu with options: 'MIS DATOS', 'MIS PRESTAMOS', 'EQUIPOS/MATERIALES DISPONIBLES', 'TODOS EQUIPOS/MATERIALES EXISTENTES', 'CONSULTAR EQUIPO/MATERIAL', and 'CERRAR SESION'. The status bar at the bottom shows 'Listo' and 'Intranet local'.

Si el usuario no tiene ningún préstamo se desplegará un mensaje “Ud. no tiene préstamos realizados”.

## Equipos/materiales disponibles

Si selecciona esta opción permitirá visualizar un listado con el código, nombre y modelo de los equipos o materiales de radio disponibles para realizar préstamos.



The screenshot shows the same web browser window as above, but the main content area is titled 'MATERIALES Y EQUIPOS DE RADIO DISPONIBLES'. It contains a table with the following data:

Código	Nombre	Modelo
M002	Material 2	RCD02
M004	Material 4	RCD04
M011	Material 11	LRU01

Below the table, there is a link labeled 'Generar PDF' with a PDF icon. The navigation menu and header are identical to the previous screenshot.

Además podrá generar un reporte de esta consulta en formato .PDF, para ello al final de la consulta se encuentra un enlace “*Generar PDF*”.

### **Todos equipos/materiales existentes**

Al seleccionar esta opción se visualizará un listado con el código, nombre, modelo, estado y fecha.



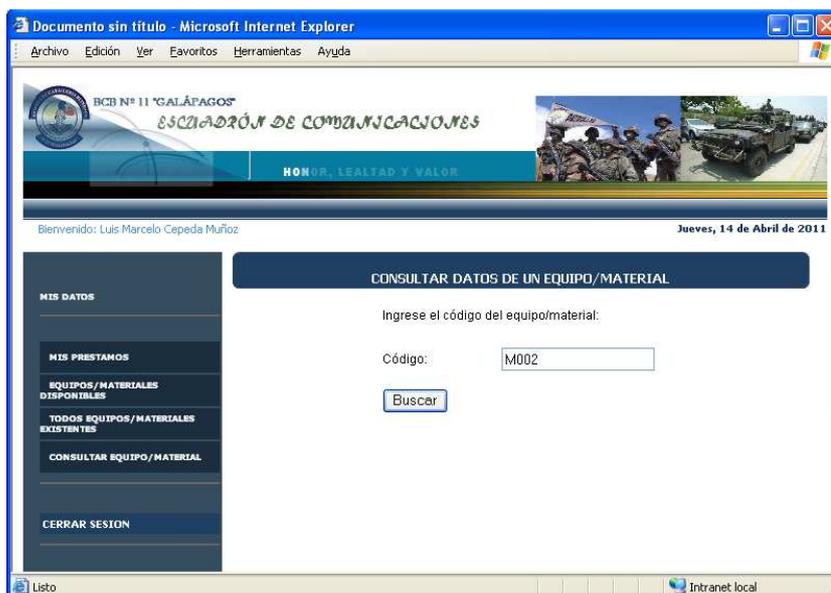
The screenshot shows a web browser window with the title "Documento sin título - Microsoft Internet Explorer". The page header includes the logo of "BCB Nº 11 'GALÁPAGOS' ESCUADRÓN DE COMUNICACIONES" and the motto "HONOR, LEALTAD Y VALOR". The user is logged in as "Luis Marcelo Cepeda Muñoz" on "Jueves, 14 de Abril de 2011". The main content area is titled "LISTAR MATERIALES Y EQUIPOS DE RADIO" and displays a table with the following data:

Código	Nombre	Modelo	Estado	FechaE
M002	Material 2	RCD02	Disponible	2011-01-01
M003	Material 3	MDF03	Prestado	2011-01-05
M004	Material 4	RCD04	Disponible	2011-01-01
M005	Material 5	RCD05	Reparacion	2011-02-12
M006	Material 6	MLC06	Reparacion	2011-01-01
M007	Material 7	RCD07	Reparacion	2011-01-03
M008	Material 8	DRT08	Prestado	2011-01-01
M009	Material 9	RCD09	Prestado	2011-02-21
M010	Material 10	RCD10	Prestado	2011-02-06
M011	Material 11	LRU01	Disponible	2011-04-14

Además se podrá generar un reporte de esta consulta en formato .PDF, para ello al final de la consulta se encuentra un enlace “*Generar PDF*”.

### **Consultar equipo/material**

Para la consulta de algún equipo o material de radio, luego de haber seleccionado la opción *Consultar equipo/material* se presentará una pantalla en la cual se ingresará el código. Presionar en el botón *Buscar*.



En el caso de que exista se mostrarán los datos de lo contrario se presentará un mensaje indicando que el equipo o material de radio no existe.



## CERRAR SESIÓN

Una vez que ha realizado las actividades necesarias con cualquier tipo de usuario que se ha logeado debe proceder a cerrar sesión para evitar que otras personas tengan acceso a su información.

## **BIBLIOGRAFIA**

1. BURBANO PROAÑO, Diego Javier. Análisis comparativo de bases de datos de código abierto vs código cerrado.  
<http://www.monografias.com/trabajos35/comparativa-bases-datos/comparativa-bases-datos.shtml>  
2010/05/06
2. CANTU, Carlos. Conoce Firebird en 2 minutos. Abril 2008.  
[http://www.firebirdnews.org/docs/fb2min\\_es.html](http://www.firebirdnews.org/docs/fb2min_es.html)  
2010/01/06
3. FIREBIRD: puesta en marcha, creación de bases de datos.  
<http://infoback.com.ar/buhardilla/firebird/firebird.htm>  
2010/01/06
4. FIREBIRD: características.  
<http://es.wikipedia.org/wiki/Firebird>  
2010/01/06
5. FIREBIRD DEVELOPMENT USING IBEXPERT  
<http://www.ibexpert.info/ibe/index.php?n=Doc.FirebirdDevelopmentUsingIBExpert>  
2010/02/11
6. FIREBIRD PROJECT: Firebird RDBMS  
<http://www.firebirdsql.org/index.php?op=files>  
2010/01/06
7. FIREBIRD 1.5 - GUIA DE ARRANQUE RAPIDO  
<http://www.firebirdsql.org/pdfmanual/es/Firebird-1.5-Arranque.pdf>  
2010/01/06

8. GILFILLAN, Ian. La biblia de MySQL

[http://www.ebookee.com/La-Biblia-de-MySQL\\_129482.html](http://www.ebookee.com/La-Biblia-de-MySQL_129482.html)

2010/04/14

9. MARTINEZ GARSABALL, Xavier. Inicio rápido a Firebird.

<http://www.clubdevelopers.com/index.php?p=58>

2010/01/06

10. MYSQL: características

<http://es.wikipedia.org/wiki/MySQL>

2010/04/14

11. MYSQL AB. MySQL 5\_0 reference manual.

<http://dev.mysql.com/doc/refman/5.0/es/index.html>

2010/04/14

12. ROMERO VARGAS, Francisco. Sistemas gestores de bases de datos:

funciones, ventajas.

<http://www.iesromerovargas.net/OASIS2/SGBD/Documentos/T2.pdf>

2010/03/15

13. SGBD: arquitectura de los sistemas de bases de datos.

<http://www.mailxmail.com/curso-sistemas-bases-datos/sgbd-arquitectura->

[sistemas-bases-datos-primera-parte](http://www.mailxmail.com/curso-sistemas-bases-datos/sgbd-arquitectura-sistemas-bases-datos-primera-parte)

2010/03/15

14. TEJERO TRUEBA, Vicente. Firebird: características básicas. Septiembre 2008

[http://www.firebird.com.mx/descargas/documentos/tema\\_2-](http://www.firebird.com.mx/descargas/documentos/tema_2-)

[caracteristicas\\_basicas.pdf](http://www.firebird.com.mx/descargas/documentos/tema_2-caracteristicas_basicas.pdf)

2010/01/20

15. TEJERO TRUEBA, Vicente. Firebird: lenguaje de manipulación de datos.  
Enero 2009.  
[http://www.firebird.com.mx/descargas/documentos/tema\\_4\\_-dml.pdf](http://www.firebird.com.mx/descargas/documentos/tema_4_-dml.pdf)  
2010/01/20
16. TELURA TECHNOLOGY. Firebird manual.  
[http://firebirdmanual.com/index\\_es.html](http://firebirdmanual.com/index_es.html)  
2010/01/20
17. ORACLE CORPORATION. MySQL Query Browser.  
<http://dev.mysql.com/doc/query-browser/es/index.html>  
2010/04/14
18. QUIÑONES AZCARATE, Ernesto. Sistemas administradores de bases de datos  
libres para el entorno empresarial.  
[http://www.eqsoft.net/presentas/sistemas\\_administradores\\_de\\_bases\\_de\\_datos\\_libres\\_para\\_el\\_entorno\\_empresarial.pdf](http://www.eqsoft.net/presentas/sistemas_administradores_de_bases_de_datos_libres_para_el_entorno_empresarial.pdf)  
2010/05/06
19. URANGA CRUZ, Raúl. Bases de datos.  
<http://www.monografias.com/trabajos12/basdat/basdat.shtml>  
2010/03/15
20. ZELADA POZO, Oscar. Qué es Firebird, tipos de servidores.  
<http://oscarzeladapd.blogspot.com/2008/01/que-es-firebird.html>  
2010/01/06