



ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO
FACULTAD DE INFORMÁTICA Y ELECTRÓNICA
ESCUELA DE INGENIERÍA EN SISTEMAS

**“ANÁLISIS COMPARATIVO ENTRE LOS MOTORES DE BASE
DE DATOS POSTGRESQL Y FIREBIRD APLICADO AL
GOBIERNO AUTÓNOMO DESCENTRALIZADO MUNICIPAL DE
SAN JOSÉ DE CHIMBO”**

**Tesis de Grado Previa Obtención del Título de:
INGENIERO EN SISTEMAS INFORMÁTICOS**

AUTOR: VÍCTOR ALFONSO COLCHA CHIMBORAZO
TUTOR: DR. JULIO ROBERTO SANTILLÁN CASTILLO

Riobamba - Ecuador

2015

AGRADECIMIENTO

Agradeciendo primeramente a Dios nuestro creador por haberme brindado salud, y guiado en el camino del bien para sobresalir en toda esta etapa de mi vida, a mi familia que siempre han estado conmigo apoyándome incondicionalmente en las buenas y en las malas ya que ellos son el pilar fundamental de mi superación. En especial a mis padres que admiro y respeto, al igual que mis hermanos y hermanas, agradezco también a mi directo y miembro de tesis por la confianza que tuvieron conmigo, por el apoyo incondicional, la paciencia, los consejos y guía de esta Tesis, agradezco a la Escuela de Ingeniería en Sistemas de la ESPOCH por abrirme las puertas y darme la oportunidad de estudiar en sus majestuosas aulas llenas de conocimiento y experiencias, a mis mejores amigo/as y compañero/as que siempre me apoyaron durante mi vida estudiantil.

Víctor Alfonso Colcha Chimborazo

DEDICATORIA

Quiero dedicar especialmente este trabajo a mi madre ya que gracias a su sacrificio sus consejos, su dedicación, sus enseñanzas, el amor hacia mí, junto con mi padre me ha podido sacar adelante, a mis hermanos y hermanas, por darme la motivación y apoyarme en cada decisión, a mis amigos y amigas que hemos compartido ideas y experiencias en nuestra etapa estudiantil.

Víctor Alfonso Colcha Chimborazo

FIRMAS RESPONSABLES Y NOTAS

NOMBRES	FIRMA	FECHA
ING. GONZALO SAMANIEGO DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRÓNICA.	_____	_____
DR. JULIO SANTILLÁN DIRECTOR DE LA ESCUELA DE INGENIERÍA EN SISTEMAS.	_____	_____
DR. JULIO SANTILLÁN DIRECTOR DE TESIS.	_____	_____
ING. GERMANIA VELOZ MIEMBRO DE TESIS.	_____	_____
COORDINADOR SISBIB ESPOCH.	_____	_____

NOTA: _____

RESPONSABILIDAD DEL AUTOR

“Yo Víctor Alfonso Colcha Chimborazo, soy el responsable de las ideas, doctrinas y resultados expuestos en esta: Tesis y el patrimonio intelectual de la misma pertenecen a la Escuela Superior Politécnica de Chimborazo”.

Víctor Alfonso Colcha Chimborazo

ÍNDICE GENERAL

ÍNDICE DE ABREVIATURAS

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

ÍNDICE DE ANEXOS

INTRODUCCIÓN

CAPITULO I

MARCO REFERENCIAL	1
1.1 Antecedentes	1
1.2 Justificación	3
<i>1.2.1 Justificación Teórica</i>	3
<i>1.2.2 Justificación Práctica</i>	3
1.3 OBJETIVOS	4
<i>1.3.1 Objetivo General</i>	4
<i>1.3.2 Objetivos Específicos</i>	4
1.4 HIPÓTESIS	4
1.5 Métodos y Técnica	5
<i>1.5.1 Métodos</i>	5
<i>1.5.2 Técnicas y Fuentes de Información</i>	5

CAPITULO II

MARCO TEÓRICO	6
2.1 Introducción a los Sistemas de Bases de Datos	6
2.2 Base de Datos	6
<i>2.2.1 ¿Por qué Base de Datos?</i>	6
<i>2.2.2 Definición de Base de Datos</i>	7
<i>2.2.3 Componentes</i>	7
2.3 Sistema de Gestión de Base de Datos	8
<i>2.3.1 ¿Por qué el Sistema de Gestión de Base de Datos?</i>	8
<i>2.3.2 Definición de Sistema de Gestión de Base de Datos</i>	8
<i>2.3.3 Características</i>	8
<i>2.3.4 Arquitectura</i>	10
<i>2.3.5 Componentes</i>	10
<i>2.3.6 Funciones</i>	12

2.4	PostgreSQL.....	13
2.4.1	¿Qué es PostgreSQL?.....	13
2.4.2	Breve Historia de PostgreSQL.....	14
2.4.3	Características	15
2.4.4	Ventajas y Desventajas.....	16
2.4.5	Componentes	18
2.4.6	Tipos de Datos.....	19
2.4.7	Herramientas de Administración	21
2.4.8	Límites de PostgreSQL.....	22
2.4.9	Empresa que lo Utilizan	23
2.5	Firebird	24
2.5.1	¿Qué es Firebird?	24
2.5.2	Breve Historia de Firebird.....	24
2.5.3	Características	25
2.5.4	Ventajas y Desventajas.....	26
2.5.5	Arquitectura	27
2.5.6	Tipos de Datos.....	29
2.5.7	Tipos de Servidores	30
2.5.8	Herramientas de Administración	31
2.5.9	Límites de Firebird	31
2.5.10	Empresa que lo Utilizan	32
CAPITULO III		
	ANÁLISIS COMPARATIVO DE LOS SGBD POSTGRESQL Y FIREBIRD	33
3.1	Introducción.....	33
3.2	Sistema de Gestión de Base de Datos a Comparar.....	33
3.2.1	PostgreSQL.....	34
3.2.2	Firebird	34
3.2.3	Otros Criterios	34
3.3	Descripción del Problema	34
3.4	Recursos Hardware y Software	34
3.5	Prototipo de Prueba	35
3.6	Definición de los Parámetros de Medición.....	37
3.6.1	Criterio de Medición del Rendimiento	37
3.6.2	Desempeño.....	38
3.6.3	Consumo de Recursos.....	38
3.6.4	Parámetros Adicionales	39
3.7	Parámetros e Indicadores	39

3.8	Escenarios de Prueba	40
3.9	Modelo para Evaluar los Resultados.....	42
3.9.1	<i>Criterios de Evaluación de Resultados</i>	42
3.9.2	<i>Calificación Porcentual Parámetro de Desempeño</i>	44
3.9.3	<i>Calificación Porcentual Consumo de Recursos</i>	44
3.9.4	<i>Calificación Porcentual de Datos Finales</i>	45
3.9.5	<i>Fórmulas y Reglas</i>	46
3.10	Proceso de Análisis y Ejecución de Pruebas	48
3.10.1	<i>Pruebas de Rendimiento: Desempeño</i>	48
3.10.2	<i>Pruebas de Rendimiento: Consumo de Recursos</i>	56
3.10.3	<i>Prueba Adicional: Robustez de los SGBD</i>	57
3.11	Resultados y Cuadros Comparativos	58
3.11.1	<i>Tiempo de Respuesta en Ingreso de Datos</i>	58
3.11.2	<i>Tiempo de Respuesta en Actualizar Datos</i>	61
3.11.3	<i>Tiempo de Respuesta Listar Datos Diversa Complejidad</i>	64
3.11.4	<i>Comparación Listar Datos de Diversa Complejidad para los dos SGBD</i>	69
3.11.5	<i>Tiempo de Respuesta en Buscar Datos de Diversa Complejidad</i>	75
3.11.6	<i>Comparación Buscar Datos de Diversa Complejidad para dos SGBD</i>	80
3.11.7	<i>Uso de la Memoria RAM</i>	86
3.11.8	<i>Uso del Procesador</i>	89
3.11.9	<i>Prueba Adicional</i>	91
3.12	Análisis de Resultados.....	93
3.12.1	<i>Resultados Desempeño</i>	93
3.12.2	<i>Resultados Consumo de Recursos</i>	98
3.13	Resultados y Evaluación General.....	101
3.14	Comprobación de Hipótesis.....	103
3.14.1	<i>Análisis de los Datos</i>	103
3.14.2	<i>Cálculo del Chi-Cuadrado (X²)</i>	104
3.14.3	<i>Nivel de Significancia y Reglas de Decisión</i>	105
3.14.4	<i>Criterio de decisión</i>	106
CAPITULO IV		
	IMPLEMENTACIÓN DEL SISTEMA SYSGAR	107
4.1	Metodología de Desarrollo de Software XP	107
4.2	Planificación del Proyecto.....	108
4.2.1	<i>Descripción del Sistema</i>	108
4.2.2	<i>Especificación de Requerimientos</i>	108
4.2.3	<i>Usuarios del Sistema</i>	111

4.2.4	<i>Historias de Usuario</i>	111
4.2.5	<i>Plan de Interacciones</i>	112
4.3	Diseño del Sistema	115
4.3.1	<i>Base de Datos</i>	116
4.3.2	<i>Diccionario de Datos</i>	116
4.3.3	<i>Diseño de Interfaz</i>	117
4.4	Codificación	118
4.5	Pruebas	123

CONCLUSIONES

RECOMENDACIONES

RESUMEN

SUMMARY

BIBLIOGRAFÍA

ANEXOS

ÍNDICE DE ABREVIATURAS Y ACRÓNIMOS

ANSI	Instituto Nacional Estadounidense de Estándares.
BD	Base de Datos.
DBA	Administrador de Base de Datos.
ESPOCH	Escuela Superior Politécnica de Chimborazo.
HW	Hardware
MS	Milisegundos
PGDG	Grupo de Desarrollo Global PostgreSQL.
PDF	Portable Document Format
PHP	Personal Home Page
RAM	Memoria de Acceso Aleatorio.
SPARC	Comité de Requisitos y Planificación de Estándares.
SW	Software
SGBD	Sistema de Gestión de Base de Dato.
SQL	Lenguaje de Consulta Estructurado.
SYSGAR	Sistema de Garantías.
XML	Extensible Markup Language
XP	Extreme Programming

ÍNDICE DE TABLAS

TABLA II.I CARACTERÍSTICAS DE LOS SGBD.....	9
TABLA II.II BREVE HISTORIA DE POSTGRESQL.....	14
TABLA II.III TIPOS DE DATOS DE POSTGRESQL.....	20
TABLA II.IV LÍMITES DE POSTGRESQL	23
TABLA II.V BREVE HISTORIA DE FIREBIRD	24
TABLA II.VI TIPOS DE DATOS DE FIREBIRD	29
TABLA II.VII LÍMITES DE FIREBIRD.....	31
TABLA III.I RECURSOS DE HARDWARE Y SOFTWARE.....	35
TABLA III.II PARÁMETROS E INDICADORES	40
TABLA III.III GRADO DE COMPLEJIDAD Y SUS CARACTERÍSTICAS	42
TABLA III.IV PONDERACIÓN DE DATOS	43
TABLA III.V CALIFICACIÓN PORCENTUAL PARÁMETRO DE DESEMPEÑO. 44	
TABLA III.VI CALIFICACIÓN PORCENTUAL CONSUMO DE RECURSOS.....	44
TABLA III.VII CALIFICACIÓN PORCENTUAL DE DATOS FINALES	45
TABLA III.VIII REGLAS PARA ASIGNACIÓN DEL MEJOR PUNTAJE.....	47
TABLA III.IX PRUEBA DE DESEMPEÑO INGRESO DE DATOS.....	48
TABLA III.X PRUEBA DE DESEMPEÑO ACTUALIZACIÓN DE DATOS	49
TABLA III.XI PRUEBA LISTAR DATOS DE COMPLEJIDAD SIMPLE	50
TABLA III.XII PRUEBA LISTAR DATOS DE COMPLEJIDAD MEDIA	51
TABLA III.XIII PRUEBA LISTAR DATOS DE COMPLEJIDAD ALTA	52
TABLA III.XIV PRUEBA BUSCAR DATOS DE COMPLEJIDAD SIMPLE.....	53
TABLA III.XV PRUEBA BUSCAR DATOS DE COMPLEJIDAD MEDIA	54
TABLA III.XVI PRUEBA BUSCAR DATOS DE COMPLEJIDAD ALTA.....	55
TABLA III.XVII PRUEBA DE USO DE LA MEMORIA RAM Y PROCESADOR ..	56
TABLA III.XVIII PRUEBA ADICIONAL.....	57
TABLA III.XIX COMPARACIÓN - INGRESO DE DATOS PARA LOS SGBD.....	60
TABLA III.XX PUNTAJE DE INGRESO DE DATOS PARA LOS DOS SGBD.....	61
TABLA III.XXI COMPARACIÓN - ACTUALIZAR DATOS DE LOS DOS SGBD .	63
TABLA III.XXII PUNTAJE ACTUALIZAR DATOS EN LOS SGBD.....	64
TABLA III.XXIII POSTGRESQL – LISTAR DATOS DIVERSA COMPLEJIDAD	66
TABLA III.XXIV FIREBIRD - LISTAR DATOS DE DIVERSA COMPLEJIDAD .	68

TABLA III.XXV LISTAR DATOS DE COMPLEJIDAD SIMPLE DE LOS SGBD...	70
TABLA III.XXVI PUNTAJE PARA LISTAR DATOS DE LOS SGBD	71
TABLA III.XXVII LISTAR DATOS DE COMPLEJIDAD MEDIA DE LOS SGBD	71
TABLA III.XXVIII PUNTAJE - LISTAR DATOS DE COMPLEJIDAD MEDIA	72
TABLA III.XXIX COMPARACIÓN-LISTAR DATOS DE COMPLEJIDAD ALTA	73
TABLA III.XXX PUNTAJE- LISTAR DATOS DE COMPLEJIDAD ALTA	74
TABLA III.XXXI COMPARACIÓN- BUSCAR DE DIVERSA COMPLEJIDAD.....	76
TABLA III.XXXII BÚSQUEDA-CUADRO COMPARATIVO	79
TABLA III.XXXIII RESULTADO COMPARATIVO DIVERSA COMPLEJIDAD..	80
TABLA III.XXXIV ASIGNACIÓN DEL PUNTAJE BÚSQUEDA SIMPLE	81
TABLA III.XXXV CONSULTA BÚSQUEDA DE COMPLEJIDAD MEDIA.....	82
TABLA III.XXXVI ASIGNACIÓN DEL PUNTAJE BÚSQUEDA MEDIA	83
TABLA III.XXXVII COMPARATIVO BUSCAR DE COMPLEJIDAD ALTA	84
TABLA III.XXXVIII ASIGNACIÓN DEL PUNTAJE BÚSQUEDA	85
TABLA III.XXXIX COMPARACIÓN-USO DE LA MEMORIA RAM	87
TABLA III.XL ASIGNACIÓN DEL PUNTAJE USO DE LA MEMORIA RAM	88
TABLA III.XLI COMPARACIÓN- USO DEL PROCESADOR.....	90
TABLA III.XLII ASIGNACIÓN DEL PUNTAJE-USO DEL PROCESADOR	91
TABLA III.XLIII PRUEBA ADICIONAL DE SELECCIÓN.....	92
TABLA III.XLIV ANÁLISIS DE RESULTADOS INSERCIÓN.....	93
TABLA III.XLV ANÁLISIS DE RESULTADOS ACTUALIZACIÓN	94
TABLA III.XLVI PUNTAJE LISTAR DATOS	95
TABLA III.XLVII RESULTADO LISTAR DATOS.....	96
TABLA III.XLVIII RESULTADO BUSCAR DATOS.....	97
TABLA III.XLIX PUNTAJE BUSCAR DATOS.....	97
TABLA III.L USO DE MEMORIA RAM DE LOS SGBD	98
TABLA III.LI USO DEL PROCESADOR	99
TABLA III.LII CALIFICACIÓN DE LOS PARÁMETROS E INDICADORES.....	101
TABLA III.LIII DATOS OBSERVADOS.....	104
TABLA III.LIV CALCULO CHI-CUADRADO.....	105
TABLA IV.I REQUERIMIENTOS DEL SISTEMA	109
TABLA IV.II REQUERIMIENTO NO FUNCIONALES.....	110
TABLA IV.III HISTORIA DE USUARIO RENOVACIÓN DE GARANTÍA.....	112
TABLA IV.IV HISTORIA DE USUARIO - INTERACCIÓN 1	112

TABLA IV.V HISTORIA DE USUARIO - INTERACCIÓN 2	113
TABLA IV.VI HISTORIA DE USUARIO - INTERACCIÓN 3.....	113
TABLA IV.VII HISTORIA DE USUARIO - INTERACCIÓN 4	114
TABLA IV.VIII HISTORIA DE USUARIO - INTERACCIÓN 5.....	114
TABLA IV.IX HISTORIA DE USUARIO - INTERACCIÓN 6.....	115
TABLA IV.X HISTORIA DE USUARIO - INTERACCIÓN 7	115
TABLA IV.XI PRUEBA HISTORIA DE USUARIOS No 7.....	124

ÍNDICE DE FIGURAS

FIGURA II.1 ARQUITECTURA DE LOS SGBD.....	10
FIGURA II.2 COMPONENTES DE POSTGRESQL.....	18
FIGURA II.3 ARQUITECTURA DE FIREBIRD	28
FIGURA III.1 BASE DE DATOS	36
FIGURA III.2 POSTGRESQL - DATOS ESTADÍSTICOS DE INSERCIÓN.....	59
FIGURA III.3 FIREBIRD - DATOS ESTADÍSTICOS DE INSERCIÓN.....	59
FIGURA III.4 COMPARACIÓN DE INGRESO DE DATOS DE DOS SGBD	60
FIGURA III.5 POSTGRESQL - DATOS ESTADÍSTICOS DE ACTUALIZACIÓN. 62	
FIGURA III.6 FIREBIRD - DATOS ESTADÍSTICOS ACTUALIZAR.....	62
FIGURA III.7 COMPARACIÓN DE DATOS DE ACTUALIZAR EN SGBD	63
FIGURA III.8 POSTGRESQL – LISTAR COMPLEJIDAD SIMPLE	65
FIGURA III.9 POSTGRESQL – LISTAR DATOS COMPLEJIDAD MEDIA.....	65
FIGURA III.10 POSTGRESQL – LISTAR DATOS COMPLEJIDAD ALTA	65
FIGURA III.11 POSTGRESQL- COMPARACIÓN LISTA DE DATOS.....	66
FIGURA III.12 FIREBIRD – LISTAR DATOS COMPLEJIDAD SIMPLE.....	67
FIGURA III.13 FIREBIRD –LISTAR DATOS COMPLEJIDAD MEDIA.....	67
FIGURA III.14 FIREBIRD – LISTAR DATOS COMPLEJIDAD ALTA	68
FIGURA III.15 FIREBIRD - COMPARACIÓN DE LISTAR DATOS	69
FIGURA III.16 COMPARACIÓN LISTAR DATOS COMPLEJIDAD SIMPLE.....	70
FIGURA III.17 LISTAR DATOS DE LA COMPLEJIDAD MEDIA	72
FIGURA III.18 LISTAR DATOS DE COMPLEJIDAD ALTA	74
FIGURA III.19 POSTGRESQL- BUSCAR DATOS COMPLEJIDAD SIMPLE.....	75
FIGURA III.20 POSTGRESQL- BUSCAR DATOS COMPLEJIDAD MEDIA.....	76
FIGURA III.21 POSTGRESQL- BUSCAR DATOS COMPLEJIDAD ALTA	76
FIGURA III.22 POSTGRESQL - COMPARACIÓN BUSCAR DATOS.....	77
FIGURA III.23 FIREBIRD DATOS ESTADÍSTICOS.....	78
FIGURA III.24 FIREBIRD- BUSCAR DATOS COMPLEJIDAD MEDIA	78
FIGURA III.25 FIREBIRD- BUSCAR DATOS COMPLEJIDAD ALTA	78
FIGURA III.26 FIREBIRD- BUSCAR DATOS DE DIVERSA COMPLEJIDAD.....	79
FIGURA III.27 RESULTADO BUSCAR DATOS COMPLEJIDAD SIMPLE	81
FIGURA III.28 BÚSQUEDA DE COMPLEJIDAD MEDIA	83

FIGURA III.29	BÚSQUEDA DE COMPLEJIDAD ALTA	85
FIGURA III.30	USOS DE LA MEMORIA RAM	86
FIGURA III.31	FIREBIRD USO DE MEMORIA RAM	87
FIGURA III.32	RESULTADOS DEL USO DE MEMORIA RAM	88
FIGURA III.33	POSTGRESQL - USO DEL PROCESADOR	89
FIGURA III.34	FIREBIRD USO DEL PROCESADOR	89
FIGURA III.35	RESULTADOS USO DEL PROCESADOR	90
FIGURA III.36	TIEMPO DE RESPUESTA PRUEBA ADICIONAL	92
FIGURA III.37	RESULTADO DE ANÁLISIS INSERTAR	94
FIGURA III.38	RESULTADO ACTUALIZAR DATOS	95
FIGURA III.39	RESULTADO LISTAR DATOS	96
FIGURA III.40	TIEMPO DE RESPUESTA PARA BUSCAR DATOS	98
FIGURA III.41	MEMORIA RAM DE LOS SGBD	99
FIGURA III.42	PROCESADOR	100
FIGURA III.43	RESULTADOS PARCIALES DEL RENDIMIENTO	102
FIGURA III.44	RENDIMIENTO GLOBAL	102
FIGURA III.45	TABLA CHI- CUADRADO	106
FIGURA IV.1	BASE DE DATOS SYSGAR	116
FIGURA IV.2	DICCIONARIO DE DATOS DE LA TABLA CONTRATO	117
FIGURA IV.3	PANTALLA PRINCIPAL SYSGAR	118
FIGURA IV.4	ESTRUCTURA DE SYSGAR	119
FIGURA IV.5	CÓDIGO DE LA CAPA DE ACCESO A DATOS	120
FIGURA IV.6	CAPA LÓGICA DE NEGOCIOS	120
FIGURA IV.7	MODELOS DE CLASES	121
FIGURA IV.8	MODELO DEFUNCIONES	121
FIGURA IV.9	MODELO DEL CONTROLADOR	122
FIGURA IV.10	MODELO VISTA	123

ÍNDICE DE ANEXOS

ANEXO 1:	SENTENCIAS SQL PRUEBA DE DESEMPEÑO	135
ANEXO 2:	SENTENCIAS SQL PRUEBA DE CONSUMO DE RECURSOS	141
ANEXO 3:	LECTURAS REALIZADAS POR CADA INDICADOR	145
ANEXO 4:	INTERFACES DE LAS PRUEBAS EN LOS SGBD	149
ANEXO 5:	HISTORIAS DE USUARIO	158
ANEXO 6:	DICCIONARIO DE DATOS	165
ANEXO 7:	MANUAL DE USUARIO	167

INTRODUCCIÓN

Hoy en día los sistemas de gestión de base de datos libres, se han ido involucrando más en el manejo de los datos de muchas empresas, instituciones y organizaciones en general, ya que han ido evolucionado e integrando nuevas características que se asemejan a los SGBD propietarios considerados de un nivel alto.

El objetivo principal de esta investigación es realizar un análisis comparativo entre los motores de base de datos PostgreSQL y Firebird, en un ambiente de igual características, seleccionando al SGBD que presente mejor desempeño y bajo consumo de recursos para ser implementado en la creación de la base de datos aplicado al sistema de garantías (SYSGAR) para Gobierno Autónomo Descentralizado Municipal de San José de Chimbo.

Se busca también probar si PostgreSQL es el que posee mejor rendimiento que Firebird.

El documento investigativo está conformado por cuatro capítulos distribuidos de la siguiente manera: El **Capítulo I** contiene al marco referencial y está compuesto por los antecedentes, justificación teórica, justificación práctica, objetivo general, objetivos específicos, hipótesis y, las técnicas y fuentes de investigación utilizadas.

En el **Capítulo II**, se contempla al marco teórico que hacer referencia al estudio de los conceptos principales de los motores de gestión de datos PostgreSQL y Firebird e información relacionada a esta investigación.

El **Capítulo III** como base principal de esta investigación contiene el análisis, procesos y resultados de las pruebas que intervinieron para la comparación entre los dos gestores de base de datos en estudio, en el cual intervienen herramientas de gestión, prototipo de la base de datos, parámetros e indicadores de medición, procedimientos almacenados, etc. Con toda la información obtenida se comprobaba al final la hipótesis planteada

Por último se tiene el **Capítulo IV** este contempla la implementación del sistema de garantías (SYSGAR) para el GAD del Cantón Chimbo, mediante la aplicación de la metodología Extreme Programming (XP) la cual esta compuestas de cuatro fases la planificación diseño, codificación y pruebas.

CAPÍTULO I

MARCO REFERENCIAL

1.1 Antecedentes

En los últimos años han ido apareciendo diferentes motores de base de datos, cada uno de ellos poseen las mismas o diferentes herramientas, pero todas llegan al mismo fin en la administración de los datos (información).

Entre los motores de base de datos más conocidos tenemos Oracle que es el líder en el área de base de datos, SQL server, PostgreSQL, MySQL y Firebird en la cual cada uno posee sus propias fortalezas.

En la actualidad los sistemas de gestión de base de datos libres, se han consolidado como alternativa, técnicamente viable y económicamente sostenible, convirtiéndose como una nueva opción para reducir costos.

PostgreSQL es un motor de administración de base de datos relacional orientado a objetos con sus siglas en inglés (ORDBMS), muy difundido e implementado en entornos de software libre, soporta distintos tipos de datos: además del soporte para los tipos base, también soporta datos de tipo fecha, monetarios, elementos gráficos, datos sobre redes (MAC, IP...), cadenas de bits, etc.

También permite la creación de tipos propios, incluye herencia entre tablas, por lo que a este gestor de bases de datos se le incluye entre los gestores objeto-relacionales, cumple con ciertos estándares como por ejemplo el SQL92 y SQL99, y por el volumen

de funcionalidades avanzadas tiene a soporta, por lo cual lo sitúa al mismo o a un nivel superior referentes a los SGBD comerciales.

En cambio Firebird es un motor de administración de base de datos relacional con sus siglas en inglés (RDBMS), que se basó en la versión 6 de Interbase, posee soporte completo de procedimientos almacenados y triggers, las transacciones son totalmente ACID compliant, arquitectura multi generacional, poca o ninguna necesidad de un DBA especializado, ofreciendo así una ventaja enorme por su alto rendimiento y adaptación de almacenamiento de información.

Actualmente no se cuenta con un estudio comparativo sobre el rendimiento (eficiencia en el consumo de recursos y tiempos de respuesta) de un ORDBMS como es PostgreSQL y un RDBMS que es Firebird, lo cual dificulta a los desarrolladores en la creación de sistemas informáticos óptimos

Este estudio se enfocará a la comparación de los SGBD libres PostgreSQL y Firebird.

En otro aspecto el Gobierno Autónomo Descentralizado Municipal de San José de Chimbo, son personas jurídicas de derecho público con autonomía política, administrativa y financiera, que formula y ejecuta los planes de desarrollo y ordenamiento territorial.

Esto quiere decir va a promover el desarrollo económico y sustentable; aplicando políticas ambientales, fortaleciendo los consejos de seguridad y protección integral, patrocinando la cultura, artes, actividades deportivas y recreativas a través de alianzas estratégicas con instituciones públicas y privadas.

En la actualidad el GAD de Chimbo genera proyectos para la comunidad, los cuales requieren de Garantías para su ejecución de acuerdo a la ley de Contratación Pública, en el capítulo III, sobre las Garantías, a las que el GAD debe sujetarse.

De dichas garantías, la entidad cuenta con un proceso de almacenamiento de información manual ya que lo manejan solo en archivos Excel.

Este proceso actual que se lleva a cabo retarda las actividades y requiere de un gran esfuerzo, para mantener actualizada la información así como la integridad de los datos. Por lo tanto requieren contar con sistema que se maneje dicha información.

1.2 Justificación

1.2.1 Justificación Teórica

El resultado de la investigación permitirá a los desarrolladores conocer el SGBD más óptimo para el desarrollo de los sistemas informáticos. Ya que PostgreSQL como Firebird son motores de base de datos de licencia libre y que están a nivel de lo SGBD comerciales, el uso de los mismos ahorrará costos de producción y sobre todo de implantación, brindando así beneficios tanto para el usuario como para el desarrollador.

A demás con el estudio se determinará un soporte informativo para poder recomendar a empresas, instituciones u otras, la buena administración que proporcionan los SGBD expuestos. De acuerdo al desempeño que ofrezca cada uno de ellos y el que brinde menor consumo de recurso garantiza que el gestor presentará un alto grado de eficacia y eficaz al incorporarlo en el manejo de los datos de un sistema.

1.2.2 Justificación Práctica

En la parte aplicativa estará compuesta por dos etapas, el desarrollo de la parte comparativa y la creación del sistema.

En el desarrollo de la primera etapa este permitirá obtener resultados de dicha comparación mediante la implementación de parámetros de medición, para determinar el consumo de recursos y tiempos de respuesta al trabajar con los SGBD, en escalas de datos ingresadas previamente en la base datos, que irán desde pequeñas cantidades hasta un promedio máximo que se requiera.

Una vez obtenido el resultado deseado del proceso anterior, se creará un sistema que se denominará SYSGAR este permitirá la administración de datos referentes a la institución, por la cual su desarrollo está compuesta por las siguientes etapas: Análisis, Diseño e Implementación. Generalizando así que el sistema SYSGAR contendrá módulos que permitirán al usuario(s) ingresar, modificar, y eliminar la información referentes a sus actividades entre ellas sobre las pólizas de seguros que mantiene el

Municipio de San José de Chimbo, buscando obtener así información actualizada que sirva a la entidad en el cumplimiento de sus objetivos y metas.

1.3 OBJETIVOS

1.3.1 Objetivo General

Realizar un análisis comparativo entre los motores de base de datos PostgreSQL y Firebird aplicado al sistema SYSGAR en el Gobierno Autónomo Descentralizado Municipal de San José de Chimbo

1.3.2 Objetivos Específicos

- Estudiar los beneficios, características y funcionamiento de los SGBD PostgreSQL 9.2.3 y Firebird 2.5.2 versiones estables.
- Establecer parámetros y herramientas de medición de rendimiento para los SGBD expuestos.
- Implementar la base de datos para el SYSGAR y determinar el SGBD de mejor rendimiento y menor consumo de recursos.
- Desarrollar el sistema Informático SYSGAR para el Gobierno Autónomo Descentralizado Municipal de San José de Chimbo con el SGBD seleccionado.

1.4 HIPÓTESIS

El SGBD PostgreSQL es el que posee mejor rendimiento que el SGBD Firebird para la administración de información del Gobierno Autónomo Descentralizado Municipal de San José de Chimbo

1.5 Métodos y Técnica

1.5.1 Métodos

El método que se ha utilizado como guía para la realización de la presente investigación es el método científico, el cual contempla los siguientes puntos:

- El planteamiento del problema que en este caso es el análisis comparativo entre los SGBD PostgreSQL y Firebird.
- El apoyo del proceso previo a la formulación de la Hipótesis.
- Levantamiento de información necesaria.
- Análisis e interpretación de Resultados.
- Proceso de comprobación de la Hipótesis.

1.5.2 Técnicas y Fuentes de Información

Técnicas

Para la recopilación de la información necesaria que sustente este trabajo de investigación, se ha establecido como técnicas las siguientes:

- Revisión de Documentos
- Pruebas
- Observación

Fuentes de Información

- Para lo que tiene que ver en cuanto a fuentes de información se utilizarán libros, sitios web oficiales de las tecnologías, pepear, tesis, etc.

CAPÍTULO II

MARCO TEÓRICO

2.1 Introducción a los Sistemas de Bases de Datos

En este capítulo se describe los principales temas, necesarios para adquirir el conocimiento general y comprender la investigación expuesta. Para ello sobre los SGBD PostgreSQL y Firebird se plantean definiciones, historia, características, tipos, ventajas y desventajas, etc.

Previamente se detalla algunos conceptos importantes referentes al área de las bases de datos.

2.2 Base de Datos

2.2.1 *¿Por qué Base de Datos?*

Imaginemos si nos hacemos una pregunta. ¿Cómo pueden compañías e instituciones o demás entidades procesar pequeñas o enormes cantidad de datos? o ¿Cómo pueden guardar y después recuperar rápidamente solo los datos que desean conocer quienes toman decisiones, justo cuando quieran verlos? La respuesta es que usan base de datos. Las bases de datos son “estructuras especializadas que permiten a sistemas computarizados guardar, manejar y recuperar datos” con gran rapidez. Prácticamente en forma general todos los sistemas modernos se apoyan en base de datos, por los cual una buena comprensión de cómo se crean estas estructuras y como se usan es vital para cualquier profesional de sistemas de información.[1]

Una base de datos proporciona a los usuarios el acceso a datos, que pueden visualizar, ingresar o actualizar, en concordancia con los derechos de acceso que se le hayan otorgado. Se convierte más útil a medida que la cantidad de datos aumente.

2.2.2 Definición de Base de Datos

“Fondo común de información almacenada en una computadora para que cualquier persona o programa autorizado pueda acceder a ella, independiente de su procedencia y del uso que haga”. [2]

Se define como “el conjunto de datos lógicamente relacionados entre sí y una descripción de estos datos, diseñado para satisfacer las necesidades de información” de una empresa, organizaciones, instituciones, etc. [3]

Entonces se puede definir que una base de datos es un conjunto de archivos (datos) interrelacionados desde un punto de vista lógico, junto con una descripción de estos datos (metadatos), para luego ser encontrados y manipulados.

2.2.3 Componentes

Los componentes [4] de una base de datos son:

- **Los datos.** El componente fundamental de la base de datos, son los datos que están interrelacionados entre sí, formando un solo conjunto con un mínimo de redundancias.
- **El software.** Los datos, para que puedan ser utilizados por diferentes usuarios y aplicaciones, deben estar estructurados y almacenados de forma independiente de las aplicaciones. Para ello se utiliza un software o conjunto de programas que actúa de interfaz entre los datos y la aplicación. A este software se le denomina Sistema de Gestión de Base de Datos (SGBD).
- **Usuarios.** Otro componente de una base de datos son los usuarios: existen tres tipos diferentes de usuarios como son los **Programadores** que construyen los programas de aplicaciones. **Usuarios Ingenuos** que son poco experimentados y utilizan las aplicaciones escritas previamente. Por último los **Usuarios Casuales** que utilizan consultas formuladas en un lenguaje de consulta de base de datos.

- **Administrador de base de datos (DBA).** Son la persona o grupo de personas encargadas del control del sistema.

2.3 Sistema de Gestión de Base de Datos

2.3.1 ¿Por qué el Sistema de Gestión de Base de Datos?

Luego de la base de datos, rápidamente surgió la necesidad de contar con un sistema de administración para controlar tanto los datos como los usuarios y otras funciones adicionales.

La administración de bases de datos se realiza con un sistema llamado SGBD. Este es un conjunto de servicios (aplicaciones de software) para administrar DB la cual permite un fácil acceso y manipulación a los datos, acceso a la información por múltiples usuarios, etc.

2.3.2 Definición de Sistema de Gestión de Base de Datos

Un sistema de administración de base de datos (DBMS, por sus siglas en inglés): “Es un conjunto de programas que maneja la estructura de la base de datos y controla el acceso a los datos guardados en ella”. [1]

“Conjunto de elementos software con capacidad para definir, mantener y utilizar una base de datos”. [2]

Software que controla la organización, almacenamiento, recuperación, seguridad e integridad de los datos en una base de datos. Acepta solicitudes de la aplicación y ordena al sistema operativo transferir los datos apropiados. Su función principal es servir como interfaz entre la base de datos, el usuario y las aplicaciones.

2.3.3 Características

En la Tabla II.I se presenta las características [25] que son deseables en los SGBD y que capacidades deben ofrecer. El DBA debe aprovechar estas capacidades para lograr

diversos objetivos relacionados con el diseño, la administración y el empleo de una base grande multiusuarios.

TABLA 2.I CARACTERÍSTICAS DE LOS SGBD

Característica	Descripción
Control de redundancia	Debe controlar los datos repetidos
Restricción de los accesos no autorizados	El SGBD debe contar con un mecanismo de seguridad y de autorización que permita al Administrador de la BD crear cuentas y especificar restricciones para ellas.
Almacenamiento persistente de objetos y estructuras de datos de programas.	Los SGBD tradicionales poseen problemas de compatibilidad con los lenguajes de programación (Orientados a objetos). Ante esta necesidad cubren los SGBDOO.
Inferencias en la base de datos mediante reglas de deducción	En un SGBD tradicional se tendría que escribir un programa por procedimientos explícito. Pero si cambian las reglas del mini mundo, casi siempre es más fácil modificar las reglas de deducción declaradas que volver a codificar los programas por procedimientos.
Suministro de múltiples interfaces con los usuarios	El SGBD debe ofrecer diferentes interfaces por los tipos de usuarios expertos e inexpertos.
Representación de vínculos complejos entre los datos	El SGBD debe representar diversos vínculos complejos de los datos y también obtener y actualizar con rapidez y eficiencia datos que estén mutuamente relacionados.
Cumplimiento de las restricciones de integridad	El SGBD tiene que proporcionar recursos para definir dichas restricciones y hacer que se cumplan.
Respaldo y recuperación	Todo SGBD debe contar con mecanismos de recuperación en fallos de HW o de SW. Para ello está el subsistema de respaldo y recuperación del SGBD.

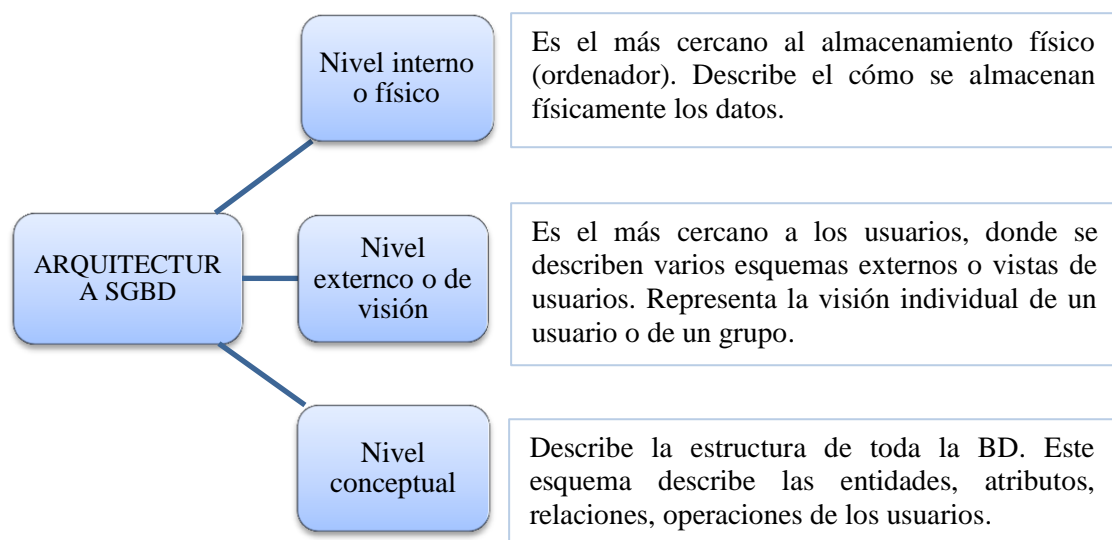
Fuente: Víctor Colcha

Las características mencionadas anteriormente representan la forma básica que debe brindar un buen SGBD.

2.3.4 Arquitectura

“El comité ANSI-SPARC propuso una arquitectura de tres niveles para los SGBD”. La cual sea convertido en un estándar para los SGBD por ser la más utilizada. Cuyo objetivo principal es el de separar los programas de aplicaciones de la base de datos física en este esquema una base de datos se define en tres niveles de abstracción distintos.[8]

FIGURA 2.1 ARQUITECTURA DE LOS SGBD



Fuente: Victor Colcha

Los SGBD necesitan que les demos una descripción o definición de la BD. Esta descripción recibe el nombre de esquema de la BD y los SGBD la tendrán continuamente a su alcance.

2.3.5 Componentes

Un sistema de base de datos se encuentra dividido en módulos cada uno de los cuales controla una parte de la responsabilidad total de sistema.

En la mayoría de los casos, el sistema operativo proporciona únicamente los servicios más básicos y el sistema de la base de datos debe partir de esa base y controlar además el manejo correcto de los datos. Así el diseño de un sistema de base de datos debe incluir la interfaz entre el sistema de base de datos y el sistema operativo.

Los componentes funcionales de un sistema de base de datos, son: [3]

➤ **Gestor de archivos.**

Gestiona la asignación de espacio en la memoria del disco y de las estructuras de datos usadas para representar datos.

➤ **Manejador de base de datos.**

Sirve de interfaz entre los datos y los programas de aplicación.

➤ **Procesador de consultas.**

Traduce las proposiciones en lenguajes de consulta a instrucciones de bajo nivel. Además convierte la solicitud del usuario en una forma más eficiente.

➤ **Compilador de DDL.**

Convierte las proposiciones DDL en un conjunto de tablas que contienen metadatos, estas se almacenan en el diccionario de datos.

➤ **Archivo de datos.**

En él se encuentran almacenados físicamente los datos de una organización.

➤ **Diccionario de datos.**

Contiene la información referente a la estructura de la base de datos.

➤ **Índices.**

Permiten un rápido acceso a registros que contienen valores específicos.

2.3.6 Funciones

Un SGBD realiza varias funciones importantes que garantizan la integridad y consistencia de los datos en la base de datos, casi todas esas funciones son transparentes a los usuarios finales y se describe a continuación. [1]

➤ **Administración de un diccionario de datos.**

- El SGBD guarda definiciones de los elementos de datos y sus relaciones (metadatos) en un diccionario de datos.
- El SGBD emplea el diccionario de datos para buscar las estructuras y relaciones requeridas de componentes de datos, relevando así al usuario de tener que codificar esas complejas relaciones en cada programa.

➤ **Administración de almacenamiento de datos.**

El SGBD crea y maneja las complejas estructuras requeridas para el almacenamiento de estos datos, liberando así al usuario de la difícil tarea de definir y programar las características físicas de datos.

➤ **Transformación y presentación de datos.**

El SGBD transforma los datos introducidos para apegarse a las estructuras de datos requeridas. También libera al usuario del trabajo de hacer una distinción entre el formato de datos lógico y el formato de datos físico.

➤ **Administración de seguridad.**

El SGBD crea un sistema de seguridad que hace cumplir la seguridad del usuario y la privacidad de datos. Las reglas de seguridad determinan cuáles usuarios pueden tener acceso a la base de datos, a qué elementos puede tener acceso el usuario y cuáles operaciones (leer, agregar, eliminar, o modificar) puede realizar el usuario.

➤ **Control de accesos a múltiples usuarios.**

Para dar integridad y consistencia de datos, el SGBD utiliza algoritmos complejos para asegurar que múltiples usuarios tengan acceso concurrentemente a la base de datos sin comprometer la integridad de esta.

➤ **Administración de respaldo y recuperación**

El SGBD suministra respaldo y recuperación de datos para garantizar la seguridad e integridad de los datos.

➤ **Administración de integridad de datos.**

El SGBD promueve y hace cumplir las reglas de integridad, reduciendo al mínimo la redundancia y maximizando la consistencia de los datos. Las relaciones de datos.

➤ **Lenguajes de acceso a base de datos e interfaz de programación de aplicaciones.**

El SGBD proporciona acceso a los datos por medio de un lenguaje de consulta. Un lenguaje de consulta no es un lenguaje de procedimiento; es un lenguaje que permite al usuario especificar qué debe hacerse sin tener que especificar cómo debe hacerse.

➤ **Interfaz de comunicación de una base de datos.**

El SGBD de la generación actual aceptan peticiones de un usuario final hechas a través de múltiples y diferentes ambientes de red.

2.4 PostgreSQL

2.4.1 ¿Qué es PostgreSQL?

Es un “sistema de gestión de bases de datos objeto-relacional que se ha desarrollado en varias formas desde 1977”. “Es considerado como un sistema de base de datos de código abierto más avanzada y experimentado de todo el mundo.” [5]

PostgreSQL es un motor administración de base de datos orientado a objetos o relacional extendido, de software libre se fue publicado en la licencia BSD. PostgreSQL no es manejado por una empresa, institución o persona, sino que es administrado por un grupo de desarrolladores que trabajan de forma desinteresada, libre y/o apoyada por organizaciones comerciales. Este grupo es denominado el PGDG.

2.4.2 Breve Historia de PostgreSQL

Este proyecto PostgreSQL como se conoce hoy en día comenzó en 1996, aunque las bases principales y el trabajo en la que se basa tiene sus principios en la década de los 70. A continuación en la Tabla II.II se ve una corta descripción de la historia de PostgreSQL. [17]

TABLA 2.II BREVE HISTORIA DE POSTGRESQL

Versiones	Años	Hecho histórico
Ingre	1977-1985	Un profesor de la Universidad de Berkeley, Michael Stonebraker empezó a desarrollar sus ideas sobre bases de datos relacionales.
PostgreSQL	1986-1994	Michael Stonebraker lideró un nuevo proyecto llamado Postgres (después de Ingres), el cual en 1988 ya se contaba con una versión utilizable.
Postgres95	1994-1995	Andrew Yu y Jolly Chen, que fueron estudiantes en Berkeley, empezaron a desarrollarla con el código de Postgres (versión 4.2) y llamaron ese proyecto Postgres95. Hicieron una limpieza general del código, arreglaron errores en el mismo, e implementaron otras mejoras
PostgreSQL 6	1996-1999	PostgreSQL Global Development Group. Se integra en el mundo Open Source.
PostgreSQL 7	1999-2004	Adopción en el mundo comercial, se le califico como la quinta SGBD más popular en USA

Continuará: ...

Continúa:

PostgreSQL 8	2005-2009	Pasó el test de Coverty Inspected encontrando solo 20 errores en 775000 líneas de código y trajo nuevas mejoras en la usabilidad, características centradas en el desarrollo y rendimiento.
PostgreSQL 9x	2010 en adelante	Aparece la Replicación síncrona: replicación sin pérdida de datos. En una primicia mundial, los usuarios incluso pueden controlar la durabilidad de cada transacción, y todos los niveles de durabilidad pueden co-existir en la misma aplicación. En la actualidad la última versión es la PostgreSQL 9.4

Fuente: Víctor Colcha

2.4.3 Características

Las características [26] más importantes y que son soportadas por el SGBD PostgreSQL se detalla a continuación. Estas fueron divididas en tres niveles:

a) Nivel General:

- Es una base de datos 100% ACID y controla la integridad referencial
- Posee Tablespaces y Nested transactions (savepoints), replicación asincrónica/sincrónica y realiza copias de seguridad en caliente (Online/hot backups).
- Posee Unicode, Juegos de caracteres internacionales y Regionalización por columna. También contiene Múltiples métodos de autenticación
- Permite el acceso encriptado vía SSL, licencia BSD, grande y diversa documentación
- Disponible para Linux y UNIX en todas sus variantes (AIX, BSD, HP-UX, SGI IRIX, Mac OS X, Solaris, Tru64) y Windows 32/64bit.

b) Nivel Programación

- Permite crear funciones/procedimientos almacenados en numerosos lenguajes de programación, entre otros PL/pgSQL (similar al PL/SQL de Oracle), PL/Perl, PL/Python y PL/Tcl
- Contiene numerosos tipos de datos y posibilidad de definir nuevos tipos. Además de los tipos estándares en cualquier base de datos, tenemos disponibles, entre otros, tipos geométricos, de direcciones de red, de cadenas binarias, UUID, XML, matrices, etc.
- Soporta el almacenamiento de objetos binarios grandes (gráficos, videos, sonido).
- Contiene APIs para programar en C/C++, Java, .Net, Perl, Python, Ruby, Tcl, ODBC, PHP, Lisp, Scheme, Qt y muchos otros.

c) Nivel SQL:

- Contiene los estándares SQL92,SQL99,SQL2003,SQL2008
- Permite crear las claves primarias (PK) y claves foráneas(FK), control de Check, Unique y el control Not null constraints.
- Controla la restricción de unicidad.
- Permite en las columnas el auto-incremento de la numeración, creación de índices compuestos, únicos, parciales y funcionales en los métodos de almacenamiento disponibles en PostgreSQL, B-tree, R-tree, hash ó GiST.
- Permite crear Sub consultas, consultas recursivas, relación de tablas (join), vistas, disparadores (triggers), por columna, condicionales, reglas y proporciona la herencia de tablas.
- Permite realizar Eventos LISTEN/NOTIFY.

2.4.4 Ventajas y Desventajas

Entre las principales ventajas y desventajas de SGBD PostgreSQL tenemos [12].

VENTAJAS

- En PostgreSQL la velocidad de respuesta pueda parecer deficiente en bases de datos muy pequeñas, esa velocidad se mantiene al aumentar el volumen de información de la base de datos, la cual no sucede con otros tipos de programas, donde bruscamente cambian su estado poniéndose lentos.
- Instalación ilimitada: No se puede demandar a una empresa por instalarlo en más ordenadores de los que la licencia permite, ya que no hay costo asociado a la licencia de software.
- Los Ahorros de costo son considerables: PostgreSQL fue diseñado para tener un mantenimiento eficiente y de ajustes menores que los productos de distribuidores comerciales, conservando todas sus potentes características de estabilidad y rendimiento.
- PostgreSQL posee Estabilidad y confiabilidad, por lo que no se han presentado caídas o problemas en las base de datos.
- Extensible: El código fuente está disponible de forma gratuita, para que quien necesite extender o personalizar el programa pueda hacerlo sin costes.
- Multiplataforma: Está disponible en casi cualquier sistema operativo, con 34 plataformas en la última versión estable, además de una versión nativa para Windows.
- Desarrollado para ambientes con altos volúmenes de datos:
- Posee grandes y diversas herramientas de interfaz para diseño y administración de bases de datos.
- Este soporta diferentes tipos de datos, condiciones, funciones y comandos de tipo estándar SQL92/SQL99 y tipos extendidos propios que posee PostgreSQL.
- Puede operar sobre distintas plataformas, incluyendo Linux, Windows, Unix, Solaris y MacOS X.
- Buen sistema de seguridad mediante la gestión de usuarios, grupos de usuarios y contraseñas.
- Gran capacidad de almacenamiento.
- Buena escalabilidad ya que es capaz de ajustarse al número de CPU y a la cantidad de memoria disponible de forma óptima, soportando una mayor cantidad de peticiones simultáneas a la base de datos de forma correcta.

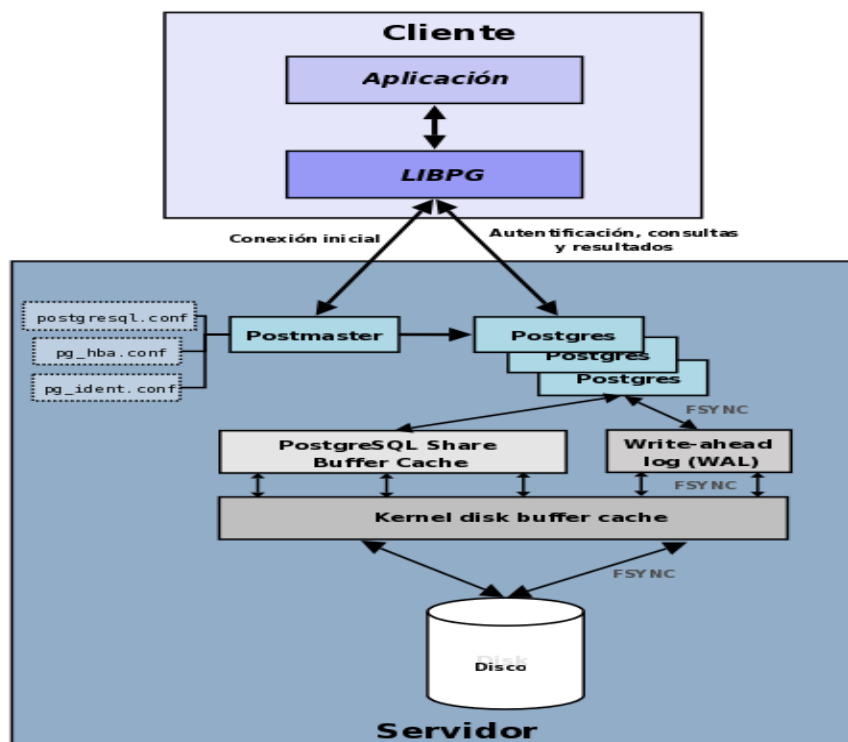
DESVENTAJAS

- En comparación con MySQL es más lento en inserciones y actualizaciones, ya que cuenta con cabeceras de intersección que no tiene MySQL.
- Soporte en línea: Hay foros oficiales, pero no hay una ayuda obligatoria.
- Consume más recursos que MySQL.
- La sintaxis de algunos de sus comandos o sentencias no es nada intuitiva.

2.4.5 Componentes

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. A continuación se muestra un gráfico que ilustra de manera general los componentes más importantes ver Figura II.2. [19]

FIGURA 2.2 COMPONENTES DE POSTGRESQL



Fuente: Víctor Colcha

- **Aplicación cliente:** Esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP o sockets locales.
- **Demonio postmaster:** Es el encargado de escuchar por un puerto/socket por conexiones entrantes de clientes. También es el encargado de crear los procesos hijos que se encargaran de autenticar estas peticiones, gestionar las consultas y mandar los resultados a las aplicaciones clientes
- **Ficheros de configuración:** Los 3 ficheros principales de configuración utilizados por PostgreSQL, postgresql.conf, pg_hba.conf y pg_ident.conf
- **Procesos hijos postgres:** Procesos hijos que se encargan de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- **PostgreSQL share buffer cache:** Memoria compartida usada por PostgreSQL para almacenar datos en caché.
- **Write-Ahead Log (WAL):** Componente del sistema encargado de asegurar la integridad de los datos.
- **Kernel disk buffer cache:** Caché de disco del sistema operativo
- **Disco:** Disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione.

2.4.6 Tipos de Datos

PostgreSQL posee un conjunto de tipos de datos nativos disponibles para los usuarios. Los usuarios pueden agregar nuevos tipos a PostgreSQL usando el comando CREATE TYPE. La siguiente Tabla II.III muestra todos los datos incorporados en los tipos de datos [27] de uso general.

TABLA 2.III TIPOS DE DATOS DE POSTGRESQL

TIPOS NUMÉRICOS			
Nombre	Tamaño	Descripción	Rango
Smallint	2 bytes	Entero de pequeño rango	-32768 a +32767
Integer	4 bytes	Entero (utilizado con regularidad)	-2147483648 a +2147483647
Bigint	8 bytes	Entero de amplio rango	-9223372036854775808 a 9223372036854775807
Decimal	Variable	Precisión especificada por el usuario, exacta	Sin limite
Numeric	Variable	Precisión especificada por el usuario, exacta	Sin limite
Real	4 bytes	Precisión variable inexacta	6 dígitos decimales de precisión
Double	8 bytes	Precisión variable inexacta	15 dígitos decimales de precisión
Serial	4 bytes	Entero auto incremento	1 a 2147483647
Bigserial	8 bytes	Entero grande auto	1 a 9223372036854775807
TIPO MONEDA			
Nombre	Tamaño	Descripción	Rango
money	8 bytes	Moneda	-92233720368547758.08 a +92233720368547758.07
TIPOS CARÁCTER			
Nombre		Descripción	
Character varying(<i>n</i>), varchar(<i>n</i>)		De longitud variable con límite	
Carácter(<i>n</i>), char(<i>n</i>)		De longitud fija, acolchado en blanco	
Text		De longitud ilimitada variables	
TIPO DATO BINARIO			
Nombre	Tamaño	Descripción	
Bytea	1 o 4 bytes	Cadena binaria de longitud binaria	
TIPO FECHA/HORA			
Nombre		Tamaño	Descripción
timestamp [(<i>p</i>)] [without time zone]		8 bytes	Ambas fecha y hora (sin zona horaria)
timestamp [(<i>p</i>)] with time zone		8 bytes	Ambas fecha y hora con zona horaria.
Date		4 bytes	fecha (sin hora)
time [(<i>p</i>)] [without time zone]		8 bytes	Hora del día (sin fecha)
time [(<i>p</i>)] with time zone		12 bytes	Hora del día con zona horaria

Fuente: Victor Colcha

2.4.7 *Herramientas de Administración*

- **PSQL:** Esta herramienta ayuda a conectarse a la DB, ejecutar sentencias SQL y gestionar una base de datos, incluyendo la creación de la DB, incluyendo las nuevas tablas con la actualización o inserción de datos, mediante comandos SQL.[7]
- **pgAdmin III:** Esta herramienta es completamente de modo gráfico para bases de datos de PostgreSQL.[15] es la herramienta libre de administración por excelencia. Entre sus características importantes tenemos el soporte completo para UNICODE, edición rápida de consultas y datos multihilo, y soporte para todos los tipos de objetos de PostgreSQL.
- **phpPgAdmin:** Es una nueva forma de administrar basada en la web para manejar DB de PostgreSQL. Es una aplicación escrita en PHP, que se instala en un servidor web, y proporciona un modo gráfico en navegador aplicaciones web para la gestión de los servidores de bases de datos. [15]
- **DBTools Manager Professional** [13]

El DBManager es la más potente aplicación para la gestión de datos. Con soporte incorporado para MySQL, PostgreSQL, Interbase / Firebird, SQLite, tablas DBF, MSAccess, MSSQL Server, Sybase, Oracle y motores de bases de datos ODBC, sino que también le trae nuevas características que la convierten en la aplicación más avanzada.

Se presenta en dos ediciones para que puedas elegir la que se ajuste a sus necesidades: Standard y Enterprise. La edición estándar es totalmente funcional y algunas características no están aún disponibles en ningún otro software libre para la gestión de bases de datos. El Enterprise Edition es un producto comercial, aunque barato, si se compara con otros productos comerciales disponibles en el mercado.

Características de gran alcance.

- a. Editor de consultas con el Editor de consultas, depurador, planificador, con múltiples resultados set
- b. Gestión de bases de datos, tablas (incluyendo índices, claves foráneas, disparadores, comprobar las limitaciones), dominios, etc.
- c. Muchos asistentes para importación y exportación de datos hacia y desde una variedad de fuentes, incluyendo: MSAccess, MSEXcel, Paradox, FoxPro, ODBC
- d. Tablas DBF, texto y archivos XML.
- e. Diseño diagrama
- f. Forma y Constructores de informe.

Algunas de estas características sólo están disponibles en la versión Enterprise Edition.

- **Otras herramientas de gestión:** A más de las nombradas existen varias herramientas para PostgreSQL, muchas integradas en el proyecto 'pgfoundry'. La página web de GBorg también tiene muchos proyectos relacionados con PostgreSQL, el sitio central de proyectos se encuentra en la página oficial <http://projects.postgresql.org>, y gran cantidad de herramientas que dan soporte a este SGBD en la página web GUI de PostgreSQL.

Un monitor rizador de sesiones llamado 'pgmonitor' específicamente para el monitoreo del SGBD PostgreSQL se encuentra en la página web <http://gborg.postgresql.org/project/pgmonitor>. [15]

2.4.8 Límites de PostgreSQL

PostgreSQL posee los siguientes límites [27]

TABLA 2.IV LÍMITES DE POSTGRESQL

Límites que presenta	Valor
Tamaño máximo de la base de datos	Ilimitado
Tamaño máximo de la tabla	32 TB
Tamaño máximo de la fila	1.6 TB
Tamaño máximo del campo	1 GB
Número máximo de la filas por tabla	Ilimitado
Número máximo de columnas por tabla	250 - 1600 (depende del tipo)
Número máximo de índices por tabla	Ilimitado

Fuente: Víctor Colcha

2.4.9 Empresa que lo Utilizan

- Yahoo!
- Geni.com
- OpenStreetMap
- Afilias
- Sony Online
- BASF
- Reddit
- Skype
- Sun xVM
- MusicBrainZ
- International Space Station
- Heroku
- Instagram

2.5 Firebird

2.5.1 ¿Qué es Firebird?

Es un sistema de administración de base de datos relacional (o RDBMS) (Lenguaje consultas: SQL) de código abierto, “basado en la versión 6 de Interbase, cuyo código fue liberado por Borland en 2000”. [9]

2.5.2 Breve Historia de Firebird

A una semana de ser liberado el Interbase 6.0 por Borland el 25 de julio del 2000, el proyecto Firebird fue creado en SourceForge. “Firebird fue desarrollado para Linux, Windows y Mac OS X el 11 de marzo del 2002, con los puertos de Solaris, FreeBSD 4, HP-UX fue desarrollado en los siguientes 2 meses”. El trabajo en portar el código base de C a C++ empezó en el año 2000. En la Tabla II.V se describe brevemente la historia de Firebird. [9]

TABLA 2.V BREVE HISTORIA DE FIREBIRD

Versiones	Años	Hecho histórico
Firebird 1.5	23 de febrero del 2004	Fue la primera versión estable de la nueva base de código, esta tenía como característica un optimizador de consultas, SQL-92 expresiones condicionales, SQL: 1999 puntos de guardado y soporte para bloqueo explícito
Firebird 2.0	12 de noviembre del 2006	En esta versión se fue añadiendo soporte para arquitectura de 64-bit, tablas anidadas en DRON clauses, y bloqueos temporales programables y transacción de bloqueo.
Firebird 2.1.3,		Esta versión añadió nuevas características incluyendo la acción de desencadenar procedimientos, consultas recursivas, y soporte para SQL: 2003MERGE estados

Continuará: ...

Continúa: ...

Firebird 2.5	Inicio del 2010	Es la actual versión estable, nuevas características incluyen la mejora: multithreading, sintaxis para expresiones regulares y la habilidad de consulta de base de datos remota
versión 3.0	Actualidad	Soporte procedimientos almacenados en otros lenguajes como el Java y C++. SQL y funciones que restringen resultados de la búsqueda

Fuente: Víctor Colcha

2.5.3 Características

Firebird tiene enormes características y la potencialidad de un RDBMS. Se pueden administrar DB desde unos pocos KB hasta varios Gigabytes con un gran rendimiento y casi sin mantenimiento.

Sus características [28] principales son:

- Las transacciones son totalmente ACID compliant, controla la integridad referencial.
- Completo lenguaje para Procedimientos Almacenados y Triggers (PSQL).
- Soporte para funciones externas (UDFs) con poca o ninguna necesidad de DBAs especializados.
- Prácticamente no necesita configuración, ¡sólo instalar y empezar a usarla!
- Una gran comunidad y muchas páginas donde conseguir buen soporte gratuito.
- Opción a usar la versión embebida de un solo fichero ideal para crear CDROM con catálogos, versiones de evaluación o monousuario de aplicaciones.
- Docenas de herramientas de terceros, incluyendo herramientas visuales de administración, replicación, etc.
- Escritura segura, recuperación rápida sin necesidad de logs de transacciones.
- Muchas formas de acceder a tus bases de datos: nativo/API, driver dbExpress, ODBC.
- Soporte variado para los S.O. incluyendo Windows, Linux, Solaris, MacOS.
- Permite el respaldos incrementales y está disponible para arquitecturas de 64bits
- Completa implementación de cursores en PSQL

Otras características

- Ejecutable pequeño, con requerimientos de hardware bajos, con una arquitectura Cliente/Servidor sobre protocolo TCP/IP y otros (embedded).
- Es medianamente escalable, posee una Buena seguridad basada en usuarios/roles.
- Requisitos de administración bajos, siendo considerada como una base de datos libre de mantenimiento, al margen de la realización de copias de seguridad y restauraciones periódicas.
- Pleno soporte del estándar SQL-92, tanto de sintaxis como de tipos de datos.
- Capacidad de almacenar elementos BLOB (binary large objects). [10]

2.5.4 Ventajas y Desventajas

Ventajas [10]

- Se minimizan los daños a la base de datos cuando falla uno de los programas clientes, pues lo peor que puede pasar es que se quede abierta una o más transacciones, las cuales posteriormente se pueden recuperar o eliminar.
- Se aprovecha al máximo las capacidades del hardware del servidor, debido a que el proceso de consulta de datos es ejecutado por el servidor SQL, el cual corre sobre el servidor de hardware.
- Se optimiza el tráfico en la red, debido a que cuando el servidor SQL devuelve los datos de una consulta al programa cliente, solamente envía el conjunto de datos producto de la consulta, y no el total de los datos de la tabla, como sucede cuando se utiliza un servidor de archivos.
- Estas son únicamente las ventajas a nivel de acceso y transmisión de datos, ya que a nivel de seguridad e integridad tiene muchas más ventajas.

Otras ventajas [11]

- Ofrece una gran variedad de medios para acceder a los datos (nativo/API) driver dbExpress, ODBC, OLEDB, .NET provider, etc.

- Otras peculiaridades resaltables son la creación de copias de seguridad y la funcionalidad en arquitecturas de 64 bits.
- Firebird administra la concurrencia en las actualizaciones en los datos, así como el manejo de las transacciones.
- Firebird ofrece una concurrencia excelente, alto rendimiento y poderoso lenguaje de procedimientos almacenados y disparadores.

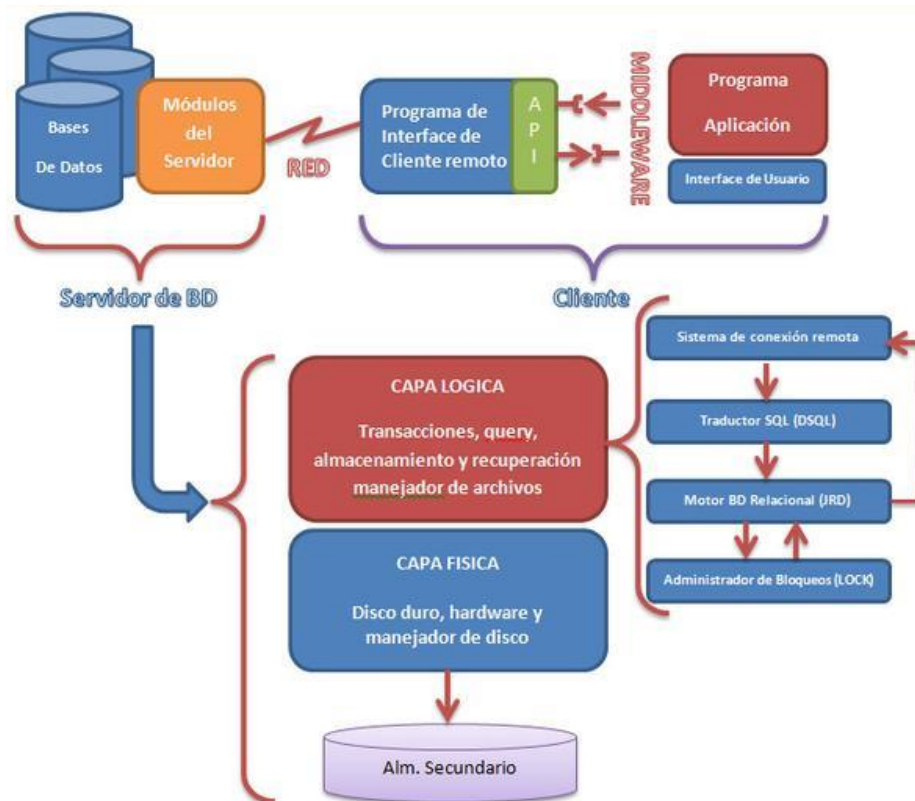
Desventajas

- Si alguna conexión de red o programa cliente tiene un problema, puede dañar el archivo de datos, ya que lo está abriendo directamente.[12]
- Las consultas a datos pueden llegar a ser muy lentas, ya que para seleccionar los registros de una consulta, cada programa cliente debe de pedir todo el archivo al servidor de archivos y toda esta información debe viajar por la red. [10]
- No se están aprovechando los recursos del servidor, solamente aquellos que optimizan el acceso a los archivos, dado que la máquina que ejecuta la consulta es la máquina cliente, además de que se crean cuellos de botella en la red, como se explica en el punto anterior. [10]

2.5.5 *Arquitectura*

A continuación se presenta una en la Figura II.3 obtenida de los escritos del libro "The Firebird Book" de Helen Borrie, donde se visualiza la arquitectura de la SMBD Firebird: [18]

FIGURA 2.3 ARQUITECTURA DE FIREBIRD



Fuente: <http://sabd15n1.wikispaces.com/Firebird>

Etapa cliente:

En esta etapa se encuentran las interfaces de los usuarios que proporcionan al cliente interactuar el sistema, dependiendo del tipo de sistema, requerirá de un software intermedio que realice una conexión de la aplicación con la base de datos.

Como Firebird está basado en el modelo cliente servidor este hace un enlace con la interface de cliente remoto, El servidor acepta peticiones TCP/IP de los clientes, por defecto sobre el puerto 3050. Además puede comunicarse usando IPX. Para que los equipos clientes puedan conectarse al servidor es necesario instalar unas herramientas cliente, generalmente una librería, que en Windows consiste en el fichero gds32.dll/fbclient.dll.

Etapa Servidor BD:

En esta sección se obtiene las solicitudes realizadas por el usuario y es enviada a por medio de la red al servidor de la base de datos, mediante un protocolo para la comunicación.

Capa Lógica:

En Firebird, todas las operaciones se realizan a partir de SQL, un sublenguaje para acceso a sistemas de manejo de bases de datos relacionales. Es en esta capa que se interpretan los query y las operaciones y son traducidas por el manejador de archivo para encontrar o devolver una página determinada. [6]

2.5.6 Tipos de Datos

En la tabla II.VI se describe los tipos de datos que posee el SGBD Firebird. [7]

TABLA 2.VI TIPOS DE DATOS DE FIREBIRD

Tipo	Descripción
BIGINT	Integer, 64 bits (-2 x 10 ⁶³ hasta 2 x 10 ⁶³ -1)
CHAR(n)	Cadena de (n) caracteres.
DATE	Entero de 32 bits. 01-01-100 a 31-12-9999
DECIMAL	Decimal (Precisión: 1-18, escala: 1-18) DECIMAL(8,3)
DOUBLE PRECISION	Punto Flotante de 64 bits 2.225 x 10 ⁻³⁰⁸ hasta 1.797 x 10 ³⁰⁸
FLOAT	Punto Flotante de 32 bits 1.175 x 10 ⁻³⁸ hasta 3.402 x 10 ³⁸
INTEGER	Entero de 32 bits, con signo -2.147.483.648 hasta 2.147. 483.647
NUMERIC	Igual que DECIMAL (Precision [,scale])
SMALLINT	Entero de 16 bits (-32.768 hasta 32.767)
TIME	Entero de 32 bits 0:00:00 hasta 23:59:59.9999
TIMESTAMP	Entero 64 bits.

Continuará: ...

Continúa: ...

VARCHAR(n)	Cadena de (n) caracteres (0 a 32.765 bytes).
BLOB	Variable.

Fuente: Víctor Colcha

2.5.7 Tipos de Servidores

Esta recopilación de los casos más comunes es una sugerencia y sirve como guía, un punto de partida en su elección [14]

SuperServer

- Bases de datos pequeñas o bases de datos con poco acceso Servidores pequeños.
- Entornos donde la caché compartida es más deseable que la escalabilidad de SuperClassic.

ClassicServer

- Entornos donde la estabilidad es la prioridad principal
- Servidores multi-procesadores.
- Grandes bases de datos con cientos de usuarios.

SuperClassic

- Servidores multi-procesadores.
- Grandes bases de datos con cientos de usuarios.
- Entornos donde la caché de dicada es más deseable que la caché compartida de SuperServer.
- Entornos donde ClassicServer ya no escala bien.

Firebird tiene otro tipo de datos que posee también variante: Embedded Server (Servidor embebido), pero esta es una bestia completamente distinta y no está orientada a sus instalaciones cliente-servidor típicas. Esta versión está enfocada a aplicaciones que van a trabajar en modo Local y no necesitan la tecnología cliente servidor para su funcionamiento.

2.5.8 Herramientas de Administración

- **Gfix** - Gestión y reparación de errores

Es una herramienta de Firebird en línea de comandos para la solución de problemas de administración como la reparación de los datos, eliminar temporales, etc

- **Gbak** - Copias de Seguridad

Es una herramienta de Firebird en línea de comandos para copia de seguridad y restauración de una base de datos completa.

- **IBExpert Developer Studio**

Es una completa herramienta de edición para el desarrollo en SQL compatible con Firebird e InterBase, necesaria para desarrollar de forma rápida y eficiente. [30]

2.5.9 Límites de Firebird

Los límites de Firebird se describen en la Tabla II.VII. [29]

TABLA 2.VII LÍMITES DE FIREBIRD

ITEM	Firebird 2.X
Tamaño máximo de DB	Prácticamente ilimitada.
Tamaño máximo DB en un solo fichero	El límite viene definido por el sistema de archivos (4Gb o 2 Gb) en algunas plataformas.
Máximo número de ficheros de base de datos	64.535
Máximo número de tablas	64.535
Máximo tamaño de una tabla	32 Tb
Máximo tamaño de un fichero externo de tabla	2 Gb
Máximo número de filas por tabla	Mayor a 16 Billones (Esto no está comprobado, este dato es teórico).
Máximo tamaño de fila	64.535 bytes (64 KB)
Máximo tamaño de página de datos	16 Kb Nota: El tamaño puede ser de 1024,

Continuará: ...

Continúa:

	2048, 4096, 8192 o 16384.
Máximo número de columnas por tabla	Depende del tipo de dato usado.
Máximo número de índices por tabla	65.535
Máximo tamaño de índice principal	4096 cuando el tamaño de la página es 16 Kb.
Máximo número de índices por tabla	4.398.046.511.104

Fuente: Víctor Colcha

2.5.10 Empresa que lo Utilizan

Algunas de las empresas [16] que utilizan el SGBD Firebird se detallan a continuación:

Bas-X (Australia). Es un proveedor de tecnologías de información empresarial a vendedores independientes, particularmente y grupos de administración. Bas-X es un líder verdadero en el desarrollo basado en Firebird: dos de sus clientes tienen bases de datos de Firebird con más de 450Gb, y otras más tienen bases de datos de más de 200Gb.

Watermark Technologies (Reino Unido). Da servicio a empresas en los sectores de Finanzas y de Gobierno. Watermark Technologies produce software que utiliza Firebird para administración de documentos que incluye un OCR indexado para búsquedas textuales. Es utilizado por consultores financieros, compañías de seguros y similares. Tienen varias bases de datos de Firebird de más de 300 Gb actualmente con sus clientes

Profitmed (Rusia). Es uno de los distribuidores más grandes de Rusia. Tienen bases de datos relativamente pequeñas (sólo cerca de 60Gb y creciendo 2Gb al mes), pero decidimos mencionarlo ya que tienen una carga extremadamente alta en términos de conexiones simultáneas, sirviendo a cientos de pequeños puntos de ventas y farmacias en toda Rusia

CAPÍTULO III

ANÁLISIS COMPARATIVO DE LOS SGBD POSTGRESQL Y FIREBIRD

3.1 Introducción

En este capítulo se presenta un análisis comparativo de los sistemas de gestión de base de datos PostgreSQL y Firebird con el fin de conocer el rendimiento que posee cada uno de ellos, donde se utilizará parámetros e indicadores para su evaluación, aplicándolos en escenarios de pruebas establecidos.

Al final de las pruebas realizadas y con los resultados obtenidos se escogerá el SGBD que posea mejor rendimiento y se implementará en la creación de la base de datos para el sistema de garantías del Gobierno Autónomo Descentralizado Municipal del Cantón Chimbo.

3.2 Sistema de Gestión de Base de Datos a Comparar

Los SGBD a comparar son PostgreSQL 9.2.3 y Firebird 2.5.2 de versiones estable, los mismos posee grandes e importantes características por lo cual a continuación se define las razones y criterios personales de selección de estos dos administradores de base de datos.

3.2.1 PostgreSQL

- SGBD de software libre.
- El más utilizado del mundo.
- Modelo de datos Objeto-Relación o Relacional Extendido.
- Es un SGBD extremadamente robusto que se iguala a ORACLE, el SGBD líder del software propietario.

3.2.2 Firebird

- DBMS de software libre.
- El menos conocido en nuestro medio pero con enormes capacidades potenciales.
- El proceso de manipulación de datos lo realiza en corto tiempo superando a MYSQL. [7]

3.2.3 Otros Criterios

- Conocimiento previo en el manejo de los SGBD PostgreSQL.

3.3 Descripción del Problema

Para la creación de cualquier sistema informático óptimo unos los requerimientos importantes es la selección de un buen SGBD de alto rendimiento, el cual debe proporcionar excelente administración de datos, tiempos de ejecución y respuesta de la información cortos, robustez, seguridad, etc.

Complementando a lo anterior en este contexto el problema se enfoca en medir el rendimiento entre los SGBD PostgreSQL y Firebird, para ello se someterán a distintas pruebas mediante criterios de medición, como los tiempos respuesta de los datos, uso de memoria RAM y procesador.

3.4 Recursos Hardware y Software

Todas las pruebas presentes se realizarán bajo las mismas condiciones de hardware como de software, tiempo, estructuración básica, disponibilidad de recursos y

herramientas de gestión, esto con la finalidad de no proporcionar ninguna clase de ventaja o desventaja a cualquiera de los dos sistemas de administración de bases de datos sometidas a la prueba, tal como se observa en la Tabla III.I.

TABLA 3.I RECURSOS DE HARDWARE Y SOFTWARE

Recursos	
Hardware	Software
Procesador: Ínter (R) Core(TM) i3	Sistema Operativo Windows 8 Enterprise
Velocidad de 2.40GHZ	Gestor de base de datos PostgreSQL 9.2
Memoria RAM de 2 GB	Gestor de base de datos Firebird 2.5
Disco Duro de 320 GB	Herramienta de gestión
Herramientas de Gestión	
POSTGRESQL	PgAdminIII
FIREBIRD	IBExpert
Mixto	DB Tool Manager Professional
Otros	Administrador de Tareas de Windows IBM SPSS Statistics versión 19

Fuente: Víctor Colcha

3.5 Prototipo de Prueba

Para el prototipo de la pruebas se tomó como referencia una porción de la base de datos creada previamente para el sistema de garantías el cual es la parte aplicativa de esta investigación. En donde se implantara a los dos SGBD PostgreSQL y Firebird.

Dicha base de datos creada para el Gobierno Autónomo Descentralizado Municipal del Cantón chimbo se definen a continuación:

Contrato Representa los datos del documento firmado para la ejecución de la obra.

Obra Es el objeto de trabajo a ejecutarse.

Seguro Es la garantía de la obra a ejecutarse

Compania Representa a la compañía de seguros que se encarga de asegurar el cumplimiento de la obra.

Entidad Representa a la institución en este caso es el GAD municipal de Chimbo

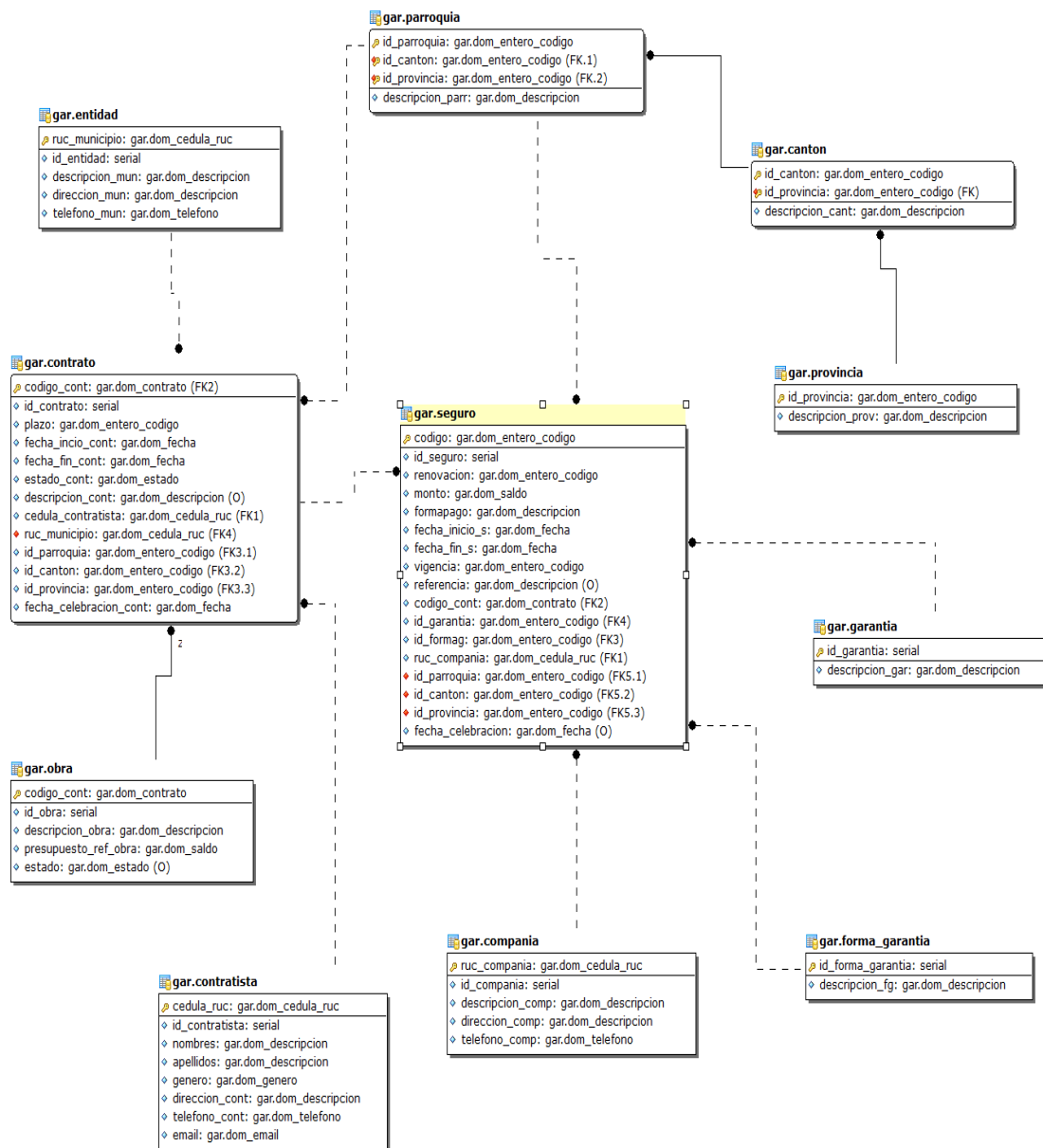
FormaPago es la forma con que se registra la garantía

Garantia Es el tipo de garantía.

La ubicación geográfica representa las tablas Parroquia, Cantón y Provincia

En la Figura III.1 se visualizan la relación entre las distintas tablas.

FIGURA 3.1 BASE DE DATOS



Fuente: Víctor Colcha

3.6 Definición de los Parámetros de Medición

El parámetro de estudio principal en la presente investigación es el Rendimiento el mismo que se define a continuación:

RENDIMIENTO

En el área informática se manifiesta como un criterio de medición o cuantificador de la velocidad y resultado con que realiza una tarea o proceso. [19]

3.6.1 Criterio de Medición del Rendimiento

En numerosas aplicaciones es necesario procesar grandes volúmenes de datos que poseen una estructura bien definida. Los SGBD han demostrado ser de gran utilidad pasando a ser una tecnología consolidada de amplia y necesaria aceptación. El tiempo de respuesta en un SGBD no es un factor crítico cuando se está manipulando pequeños volúmenes de información, pero si son de grandes proporciones será importante el desempeño en el SGBD.

Un aspecto primordial referente al rendimiento es saber cómo está respondiendo el sistema operativo a las actividades que realiza las aplicaciones en cuanto al HW. En este punto es importante contar con un sistema que funcione apropiadamente aún más allá de la carga de trabajo máxima esperado. [31]

En base a la definición y al criterio de medición plateada se ha definido dos parámetros para el análisis del rendimiento los mismos que están directamente relacionados con el parámetro original donde solo se ha tomado nombres diferentes:

- **Desempeño:** Es tiempo que se empleado para realizar una tarea o actividad. [32]
- **Consumo de recurso:** Es el uso del hardware que se requiere para un proceso activo. [32]

3.6.2 Desempeño

Las pruebas de desempeño están diseñadas para medir tiempos de respuesta. Esta prueba ayudará a medir y dar a conocer la rapidez de manipulación de datos que generan los dos SGBD.

Por lo cual se ha definido los siguientes indicadores en base a las necesidades del manejo de la información, más adelante en la sección 3.9.1.se detalla lo mencionado.

Tiempo de respuesta para el registro de datos.- Tiempo que se demora en almacenar la información de acuerdo a una determinada cantidad de datos proporcionados.

Tiempo de respuesta para la actualización de datos.- Tiempo que demora en editar o modificar la información de acuerdo a una determinada cantidad de datos proporcionados

Tiempo de respuesta al listar datos sin condiciones.- Tiempo que se demora en listar la información de acuerdo a un volumen de datos determinado. Cabe mencionar que esta consulta están relacionadas las tablas pero no tienen condición de búsqueda

Tiempo de respuesta al buscar datos con condiciones.- Tiempo que se demora en buscar la información de acuerdo a un volumen de datos determinado. Cabe mencionar que esta consulta están relacionadas las tablas y si poseen condiciones de búsqueda

3.6.3 Consumo de Recursos

En este parámetro consiste en realizar tareas de trabajo diario como consultas, actualizaciones, inserciones y eliminaciones de registros simultáneamente en una sola acción, el SGBD consumirá recursos mientras realiza sus actividades. Lo que se va a medir es la capacidad de realizar la tarea y el comportamiento del sistema operativo en cuanto el nivel de consumo de recurso del procesador y memoria RAM que son los indicadores de este parámetro, más adelante en la sección 3.9.1 se detallara el origen de estos indicadores.

3.6.4 Parámetros Adicionales

En base a los objetivos planteados en esta investigación estos parámetros no serán tomados en cuenta en la calificación final, pero servirán para definir las conclusiones y recomendaciones.

Integridad de datos

La integridad de datos, es la calidad, precisión y validez de la información. [33]

Durante la ejecución de las pruebas es posible que los datos sufran algún tipo de anomalías o se pierdan, ocasionado por problemas físicos en la estructura de la tabla o posiblemente por problemas del SGBD al momento de procesar las tablas con millones de registros. Por esta razón se procederá después de cada prueba a realizar consulta a los registros almacenados comprobando la integridad y existencia.

Robustez

Es la capacidad de los productos software de reaccionar apropiadamente ante condiciones excepcionales. La robustez es la habilidad del sistema para manejar elegantemente entradas inválidas. [33]

La robustez se va a analizar mediante una prueba adicional, que consiste en demostrar cómo responde los SGBD en consultas de selección ante la estructura de la sentencia SQL mal definida.

Cabe mencionar que uno de los SGBD no supero ni las pruebas iniciales, y no se pudo culminar con todas las pruebas, por tal motivo no será tomada en cuenta en la calificación final, esto servirá solo para las conclusiones y recomendaciones.

3.7 Parámetros e Indicadores

Resumiendo del ítem 3.6 en la Tabla III.II se describen los parámetros e indicadores implementados para medir el rendimiento de los SGBD PostgreSQL y Firebird.

TABLA 3.II PARÁMETROS E INDICADORES

Parámetros	Indicadores	Descripción
Desempeño	Tiempo de respuesta de ingreso.	Tiempo que se demora en registrar la cantidad de datos especificados.
	Tiempo de respuesta de actualización.	Tiempo que se demora en actualizar la cantidad de datos especificados.
	Tiempo de respuesta de consulta para listar datos	Tiempo que se demora en seleccionar la cantidad de datos especificado
	Tiempo de respuesta de consultas para búsqueda de datos	Tiempo que se demora en buscar y listar los datos en cantidades especificadas
Consumo de recursos	Uso de la Memoria RAM	Cantidad de memoria necesaria para operaciones de sentencias SQL.
	Uso del procesador	Porcentaje de uso del procesador para operaciones de sentencias SQL.

Fuente: Víctor Colcha

3.8 Escenarios de Prueba

Un aspecto importante de los SGBD y demanda de muchas empresas es su rendimiento que posee al administrar enormes volúmenes de datos, donde los factores principales son el tiempo de respuesta, robustez, seguridad e integridad. Por lo cual es necesario observar el comportamiento del SGBD en estudio PostgreSQL y Firebird enfocándonos al desempeño desde diferentes perspectivas como el tipo y complejidad de las operaciones, así como también el volumen de datos que pueden soportar.

A demás en estas pruebas permitirá determinar los alcances del sistema operativo, de manera que DBA decida donde concentrar esfuerzos para maximizar el desempeño del SGBD, ya sea con el mejoramiento de las características del HW, configurando los elementos del servicio de la base de datos para que su uso sea optimo en velocidad y en uso de memoria, o bien estructurando las estrategias de programación sobre la base de datos.

Como uno de los objetivos de estas pruebas es pretender analizar el efecto que produce la variación del número de registros al insertar, actualizar y de recuperación de los

mismos, en los tiempos de respuesta que proporciona los SGBD en estudio, al igual el consumo de recursos que demanda.

Escenarios

Para poder cumplir dicho objetivo se define 5 escenarios con diferentes volúmenes de datos, los cuales varían entre los 1000, 10000, 100000, 1000000 hasta 10000000 de registros. Aplicando a las operaciones más críticas de un SGBD como la de inserción, actualización y selección de acuerdo a la experiencia del autor más el criterio de evolución de resultado detallados más adelante en la sección 3.9.1.

Operaciones Insertar y Actualizar

Para estas dos operaciones se han creado procedimientos almacenados para cada SGBD, el mismo que en la operación insertar generara el número de registros que se ingrese como parámetro, mientras que en la operación actualizar va a modificar la misma porción de datos del parámetro ingresado, de acuerdo a los escenarios propuestos.

Operación de Consulta (selección de datos)

Para esta operación se ha estructurado en dos partes: el listado de datos y la búsqueda de los mismos.

En la operación de listado de datos se va a visualizar el número de registros insertados como parámetro, mientras que en la búsqueda de datos se va a visualizar el número de registros buscados de acuerdo al valor ingresado como parámetro.

Hay que tener en cuenta de que cuando listamos los datos después de hacer alguna operación de inserción actualización o eliminación de datos estamos comprobando la integridad de los datos, es decir si sufrió algún cambio la información que desea manejar o que estaba manejando.

Grado de complejidad de las sentencias SQL

Para la prueba de selección a implementar se han definido 3 grados de complejidad: simple, mediana y alta.

La Tabla III.III se detalla las características que posee un grado de complejidad para las sentencias SQL en donde se aplicara en las diferentes pruebas.

TABLA 3.III GRADO DE COMPLEJIDAD Y SUS CARACTERÍSTICAS

SENTENCIA SQL	
Grado de complejidad	Características
Simple	<ul style="list-style-type: none">• Tiempos en consultas y ordenamiento de datos.• 3 tablas.
Mediana	<ul style="list-style-type: none">• Tiempos en consultas y ordenamiento de datos.• Capacidad de subconsultas.• 5 tablas.
Alta	<ul style="list-style-type: none">• Tiempos en consultas y ordenamiento de datos.• Capacidad de subconsultas.• Combinación de resultados por medio de la sentencia unión.• 9 tablas.

Fuente: Víctor Colcha

3.9 Modelo para Evaluar los Resultados

Los motores de base de datos tanto a PostgreSQL como Firebird serán evaluados de forma comparativa mediante datos estadísticos, que se obtuvo de los resultados de las pruebas realizadas a los escenarios en cada parámetro planteado. Para comparar el nivel de cumplimiento de cada SGBD se crearán tablas de calificación, donde se pretende exponer dicha comparación.

3.9.1 Criterios de Evaluación de Resultados

Las empresas, instituciones, organizaciones en general, desarrolladores y distintos usuarios sin darse cuenta la necesidad de ellos es conocer dónde y cómo está su información rápidamente.

Esta necesidad con lleva a saber que la administración de la información es un factor muy importante de las personas en todo el mundo. Al igual que los medios para el acceso a ellas en software y hardware. [34]

De aquí nace el criterio de medición para evaluar las distintas pruebas y saber su importancia.

Como primer indicador influyente en cualquier tipo de necesidades es el tiempo por lo que se le asignó al parámetro de desempeño una porcentaje alto del 60 %. Donde se dividirán en porciones iguales para poder administrar toda la información.

Como segundo indicador se ha planteado el consumo del hardware ya que este influye en la velocidad de obtención de la información, principalmente al procesador que es encargado de la administración de los procesos o servicios y la memoria RAM que se encarga de brindar la velocidad requerida para tales procesos. Por lo cual se ha asignado al parámetro consumo de recurso un total de 40 % que serán divididas en porciones igual en el uso del procesador y el uso de la memoria RAM

La evaluación de los dos SGBD se lo hará en base al criterio de evaluación de resultados, importancia de los indicadores de cada parámetro y bajo la experiencia del autor de la tesis, con lo cual se obtendrá resultados cuantitativos y cualitativos.

La Tabla III. IV muestra la ponderación de los datos que ha sido asignado que serán como base para el cumplimiento de objetivos de esta investigación.

TABLA 3.IV PONDERACIÓN DE DATOS

PRUEBAS TÉCNICAS		Valor Porcentual	
Parámetros	Indicadores	Unitario	Total
Desempeño	Tiempos de respuesta para ingresar de datos	15 %	60 %
	Tiempos de respuesta para actualizar de datos	15 %	
	Tiempo de respuesta de consulta para listar datos	15 %	
	Tiempo de respuesta de consulta para buscar datos	15 %	
Consumo de recursos	Uso de la memoria RAM	20 %	40 %
	Uso del Procesador	20 %	
TOTAL		100 %	

Fuente: Víctor Colcha

3.9.2 Calificación Porcentual Parámetro de Desempeño

En base al porcentaje asignado al parámetro de desempeño, para el análisis de los cuatro indicadores detallados de la Tabla III.IV se utiliza la siguiente calificación como se describe en la Tabla III.V.

TABLA 3.V CALIFICACIÓN PORCENTUAL PARÁMETRO DE DESEMPEÑO

Rango de calificación	Valor cuantitativo	Valor cualitativo	Descripción
>12% o <=15%	5	Excelente	Todas las expectativas cumplidas.
>9% o <=12%	4	Muy bueno	Mayoría de las expectativas cumplidas.
>6% o <=9%	3	Buena	Expectativa media
>3% o <=6%	2	Regular	Pocas de las expectativas cumplidas.
>=0% o <=3%	1	Malo	Ninguna expectativa cumplida.

Fuente: Víctor Colcha

3.9.3 Calificación Porcentual Consumo de Recursos

En base al porcentaje asignado al parámetro consumo de recursos, para el análisis de los dos indicadores detallados de la Tabla III.IV se utiliza la siguiente calificación como se describe en la Tabla III.VI.

TABLA 3.VI CALIFICACIÓN PORCENTUAL CONSUMO DE RECURSOS

Rango de calificación	Valor cuantitativo	Valor cualitativo	Descripción
>16% o <=20%	8	Excelente	Todas las expectativas cumplidas.
>12% o <=16%	6	Muy buena	Mayoría de las expectativas

Continúa: ...

			cumplidas.
>8% o <=12%	4	Buena	Mediana expectativa cumplida
>4% o <=8%	2	Regular	Pocas de las expectativas cumplidas.
>=0% o <=4%	0	Malo	Ninguna expectativa cumplida.

Fuente: Víctor Colcha

3.9.4 Calificación Porcentual de Datos Finales

En base a la suma de los dos porcentajes asignados para cada parámetro el desempeño y consumo de recursos detallados en la Tabla III.IV se utiliza la siguiente calificación como se describe en la Tabla III.V.

TABLA 3.VII CALIFICACIÓN PORCENTUAL DE DATOS FINALES

Rango de calificación	Valor cuantitativo	Valor cualitativo	Descripción
>80% o <=100%	5	Excelente	Todas las expectativas cumplidas.
>60% o <=80%	4	Muy buena	Mayoría de las expectativas cumplidas.
>40% o <=60%	3	Buena	Mediana expectativa cumplida
>20% o <=40%	2	Regular	Pocas de las expectativas cumplidas.
>=0% o <=20%	1	Malo	Ninguna expectativa cumplida.

Fuente: Víctor Colcha

3.9.5 Fórmulas y Reglas

FÓRMULAS

Para el cálculo del promedio se utiliza la siguiente la siguiente fórmula.

$$Pp = \frac{\sum_{i=1}^n vip}{n} = \frac{v_i + v_{i+1} + \dots + v_{i+k}}{n} \quad (1)$$

$$Pf = \frac{\sum_i^i vif}{n} = \frac{v_i + v_{i+1} + \dots + v_{i+k}}{n} \quad (2)$$

$$Ptotal = \frac{\sum_i^n id}{nid} = \frac{id_1 + id_2 + id_3 + id_4 + id_5 + id_6}{nid} \quad (3)$$

Dónde:

vip= Valor de las pruebas individuales del SGBD PostgreSQL

vif= Valor de las pruebas individuales del SGBD Firebird

n= Número total de pruebas

Pp= Valor promedio de las pruebas individuales del SGBD PostgreSQL

Pf= Valor promedio de las pruebas individuales del SGBD Firebird

Pf= Valor promedio de las pruebas individuales del SGBD Firebird

Ptotal= Promedio total de los indicadores

id= Indicador

nid= Número total de indicadores

REGLAS

El indicador tiempo especialmente el individual y la capacidad con la que se trabajan todos los procesos, son factores importantes al momento de determinar el óptimo desempeño, robustez y consumo de recurso global de los SGBD.

Es por esta razón donde nace esta forma de calificar, la cual consiste en la asignación de un puntaje que se presenta en cada prueba individual, tanto en PostgreSQL como en Firebird. Con la cual se verificara con mayor exactitud los datos (Tiempo de respuesta y capacidad del recurso) en dichas pruebas, donde el mejor tiempo o capacidad será el menor tiempo de respuesta y por lo tanto el que optimice mejor ese recurso

Para evaluar (asignar un puntaje) los tiempos y consumo de recursos se lo va a realizar de acuerdo a la Tabla III.VIII. [19]

TABLA 3.VIII REGLAS PARA ASIGNACIÓN DEL MEJOR PUNTAJE

Puntaje	Regla	Eqv.	Puntos	Detalle
Excelente	$T1 < T2$	T1	10	Menor tiempo de respuesta, cumple con todo lo esperado.
Muy Buena	$T2 \leq T1 + 20\%T1$	T2	8	Mayoría de las expectativas cumplidas
Buena	$T2 \leq T1 + 60\%T1$	T2	6	Mediana expectativa cumplida
Regular	$T2 \leq T1 + 100\%T1$	T2	4	Pocas de las expectativas cumplidas
Malo	$T2 > T1 + 100\%T1$	T2	2	Ninguna expectativa cumplida

Fuente: Víctor Colcha

3.10 Proceso de Análisis y Ejecución de Pruebas

3.10.1 Pruebas de Rendimiento: Desempeño

✓ Prueba de Desempeño : Ingreso de Datos

TABLA 3.IX PRUEBA DE DESEMPEÑO INGRESO DE DATOS

PRUEBA No.1	INGRESO
--------------------	----------------

OBJETIVO: Obtener el tiempo de respuesta de los SGBD PostgreSQL y Firebird al registrar una determinada cantidad de datos.

PARÁMETROS: ✓ Desempeño: Tiempo de ingreso ✓ Parámetro adicional integridad de datos LECTURAS A OBTENER: 10	HERRAMIENTAS DE LOS SGBD: ✓ PROPIAS: PgAdmin III IBExpert ✓ EXTERNA: DBTools Manager Professional
--	--

ESCENARIO DE PRUEBAS:	1000	10000	100000	1000000	10000000
------------------------------	------	-------	--------	---------	----------

PROCEDIMIENTO:

- ✓ Creación de los procedimientos almacenados **REGISTRAR** con sentencias SQL similares para cada SGBD.
- ✓ Ejecución de los procedimientos almacenados en las herramientas propias de cada SGBD.
- ✓ Obtención de los tiempos de respuesta que emiten cada SGBD.
- ✓ Visualizar el registro mediante una consulta de datos en la herramienta externa.

CÓDIGO:

1. Código Registro de datos en PostgreSQL ver en el anexo 1, literal “a”
2. Código Registro de datos en Firebird ver en el anexo 1, literal “o”

EJECUCIÓN DE LA PRUEBA: La ejecución de esta prueba para PostgreSQL se lo ha realizado en PG ADMIN III y para FIREBIRD en IBEXPERT, se puede apreciar esta ejecución en el anexo 4 literal “a” y “b” respectivamente.

OBSERVACIONES: Mediante la realización de una consulta con la herramienta externa se pudo medir la integridad y existencia de los datos.

✓ **Prueba de Desempeño: Actualizar Datos**

TABLA 3.X PRUEBA DE DESEMPEÑO ACTUALIZACIÓN DE DATOS

PRUEBA No.2	ACTUALIZACIÓN
--------------------	----------------------

OBJETIVO: Obtener el tiempo de respuesta de los SGBD PostgreSQL y Firebird al modificar una determinada cantidad de datos

<p>PARÁMETROS:</p> <ul style="list-style-type: none"> ✓ Desempeño: Tiempo de actualización ✓ Parámetro adicional integridad de datos <p>LECTURAS A OBTENER: 10</p>	<p>HERRAMIENTAS DE LOS SGBD:</p> <ul style="list-style-type: none"> ✓ PROPIAS: PgAdmin III IBExpert ✓ EXTERNA: DBTools Manager Professional
--	--

ESCENARIOS DE PRUEBAS:	1000	10000	100000	1000000	10000000
-------------------------------	-------------	--------------	---------------	----------------	-----------------

PROCEDIMIENTO:

- ✓ Creación de los procedimientos almacenados **ACTUALIZAR** con sentencias SQL similares para cada SGBD.
- ✓ Ejecución de los procedimientos almacenados en las herramientas propias de cada SGBD.
- ✓ Obtención de los tiempos de respuesta que emiten cada SGBD.
- ✓ Visualizar el registro mediante una consulta de datos en la herramienta externa.

CÓDIGO:

1. Código Actualización de datos en PostgreSQL ver en el anexo 1, literal “b”
2. Código Actualización de datos en Firebird ver en el anexo 1, literal “o”

EJECUCIÓN DE LA PRUEBA: La ejecución de esta prueba para PostgreSQL se lo ha realizado en PG ADMIN III y para FIREBIRD en IBEXPERT se puede apreciar esta ejecución en el anexo 4 literal “c” y “d” respectivamente.

OBSERVACIONES: Mediante la realización de una consulta con la herramienta externa se pudo medir la integridad y existencia de los datos.

Fuente: Víctor Colcha

✓ **Prueba de Desempeño : Listar en Complejidad Simple**

TABLA 3.XI PRUEBA LISTAR DATOS DE COMPLEJIDAD SIMPLE

PRUEBA No.3	LISTAR	COMPLEJIDAD SIMPLE
--------------------	---------------	---------------------------

OBJETIVO: Obtener el tiempo de respuesta de los SGBD PostgreSQL y Firebird al listar una determinada cantidad de datos mediante una consulta de complejidad simple.

<p>PARÁMETROS:</p> <ul style="list-style-type: none"> ✓ Desempeño: Tiempo de selección ✓ Parámetro adicional integridad de datos <p>LECTURAS A OBTENER: 10</p>	<p>HERRAMIENTA DE LOS SGBD:</p> <ul style="list-style-type: none"> ✓ EXTERNA: DBTools Manager Professional
--	---

ESCENARIOS DE PRUEBAS:	1000	10000	100000	1000000	10000000
-------------------------------	-------------	--------------	---------------	----------------	-----------------

PROCEDIMIENTO:

- ✓ Creación de los procedimientos almacenados para la consulta de **COMPLEJIDAD SIMPLE** con sentencias SQL similares para cada SGBD.
- ✓ Ejecución de los procedimientos almacenados en la herramienta externa para cada SGBD.
- ✓ Obtención de los tiempos de respuesta que emiten cada SGBD.

CÓDIGO:

1. Código listar datos de complejidad simple en PostgreSQL, ver anexo 1, literal “c”
2. Código listar datos de complejidad simple, en Firebird, ver anexo 1, literal “d”

EJECUCIÓN DE LA PRUEBA: La ejecución de esta prueba para PostgreSQL y FIREBIRD se lo ha realizado en DBTOOLS MANAGER PROFESSIONAL, se puede apreciar esta ejecución en el anexo 4 literal “e” y “f” respectivamente.

OBSERVACIONES: Mediante la realización de la consulta se obtuvo la información esperada probando así también la integridad de los datos.

Fuente: Víctor Colcha

✓ **Prueba de Desempeño : Listar Complejidad Media**

TABLA 3.XII PRUEBA LISTAR DATOS DE COMPLEJIDAD MEDIA

PRUEBA No.4	LISTAR	COMPLEJIDAD MEDIA
--------------------	---------------	--------------------------

OBJETIVO: Obtener el tiempo de respuesta de los SGBD PostgreSQL y Firebird al listar una determinada cantidad de datos mediante una consulta de complejidad media

<p>PARÁMETROS:</p> <ul style="list-style-type: none"> ✓ Desempeño: Tiempo de selección ✓ Parámetro adicional integridad de datos <p>LECTURAS A OBTENER: 10</p>	<p>HERRAMIENTA DE LOS SGBD:</p> <ul style="list-style-type: none"> ✓ EXTERNA: DBTools Manager Professional
--	---

ESCENARIOS DE PRUEBAS:	1000	10000	100000	1000000	10000000
-------------------------------	-------------	--------------	---------------	----------------	-----------------

PROCEDIMIENTO:

- ✓ Creación de los procedimientos almacenados para la consulta de **COMPLEJIDAD MEDIA** con sentencias SQL similares para cada SGBD.
- ✓ Ejecución de los procedimientos almacenados en la herramienta externa para cada SGBD.
- ✓ Obtención de los tiempos de respuesta que emiten cada SGBD.

CÓDIGO:

1. Código listar datos de complejidad media PostgreSQL, ver anexo 1, literal “e”
2. Código listar datos de complejidad media en Firebird, ver anexo 1, literal “f”

EJECUCIÓN DE LA PRUEBA: La ejecución de esta prueba para PostgreSQL y Firebird se lo ha realizado en DBTOOLS MANAGER PROFESSIONAL, se puede apreciar esta ejecución en el anexo 4 literal “e” y “f” respectivamente.

OBSERVACIONES: Mediante la realización de la consulta se obtuvo la información esperada probando así también la integridad de los datos.

Fuente: Víctor Colcha

✓ **Prueba de Desempeño : Listar Complejidad Alta**

TABLA 3.XIII PRUEBA LISTAR DATOS DE COMPLEJIDAD ALTA

PRUEBA No.5	LISTAR	COMPLEJIDAD ALTA
--------------------	---------------	-------------------------

OBJETIVO: Obtener el tiempo de respuesta de los SGBD PostgreSQL y Firebird al listar una determinada cantidad de datos mediante una consulta de complejidad alta.

<p>PARÁMETROS:</p> <ul style="list-style-type: none"> ✓ Desempeño: Tiempo de selección ✓ Parámetro adicional integridad de datos <p>LECTURAS A OBTENER: 10</p>	<p>HERRAMIENTAS DE LOS SGBD:</p> <ul style="list-style-type: none"> ✓ EXTERNA: DBTools Manager Professional
--	--

ESCENARIOS DE PRUEBAS:	1000	10000	100000	1000000	10000000
-------------------------------	-------------	--------------	---------------	----------------	-----------------

PROCEDIMIENTO:

- ✓ Creación de los procedimientos almacenados para la consulta de **COMPLEJIDAD ALTA** con sentencias SQL similares para cada SGBD.
- ✓ Ejecución de los procedimientos almacenados en la herramienta externa para cada SGBD.
- ✓ Obtención de los tiempos de respuesta que emiten cada SGBD.

CÓDIGO:

1. Código listar datos de complejidad alta en PostgreSQL, ver anexo 1, literal “g”
2. Código listar datos de complejidad alta en Firebird, ver anexo 1, literal “h”

EJECUCIÓN DE LA PRUEBA: La ejecución de esta prueba para PostgreSQL y Firebird se lo ha realizado en DBTOOLS MANAGER PROFESSIONAL, se puede apreciar esta ejecución en el anexo 4 literal “e” y “f” respectivamente.

OBSERVACIONES: Mediante la realización de la consulta se obtuvo la información esperada probando así también la integridad de los datos.

Fuente: Víctor Colcha

✓ **Prueba de Desempeño: Búsqueda en Complejidad Simple**

TABLA 3.XIV PRUEBA BUSCAR DATOS DE COMPLEJIDAD SIMPLE

PRUEBA No.6	BÚSQUEDA	COMPLEJIDAD SIMPLE
--------------------	-----------------	---------------------------

OBJETIVO: Obtener el tiempo de respuesta de los SGBD PostgreSQL y Firebird al buscar y listar una determinada cantidad de datos mediante una consulta de complejidad simple.

<p>PARÁMETROS:</p> <ul style="list-style-type: none"> ✓ Desempeño: Tiempo de selección ✓ Parámetro adicional integridad de datos <p>LECTURAS A OBTENER: 10</p>	<p>HERRAMIENTAS DE LOS SGBD:</p> <ul style="list-style-type: none"> ✓ PROPIAS: PgAdmin III IBExpert
--	--

ESCENARIOS DE PRUEBAS:	1000	10000	100000	1000000	10000000
-------------------------------	-------------	--------------	---------------	----------------	-----------------

PROCEDIMIENTO:

- ✓ Creación de los procedimientos almacenados para la consulta de **COMPLEJIDAD SIMPLE** con sentencias SQL similares para cada SGBD.
- ✓ Ejecución de los procedimientos almacenados en la herramienta propia para cada SGBD.
- ✓ Obtención de los tiempos de respuesta que emiten cada SGBD.

CÓDIGO:

1. Código buscar datos de complejidad simple en PostgreSQL, ver anexo 1, literal “i”
2. Código buscar datos de complejidad simple en Firebird, ver anexo 1, literal “j”

EJECUCIÓN DE LA PRUEBA: La ejecución de esta prueba para PostgreSQL se lo ha realizado en PG ADMIN III y para FIREBIRD en IBEXPERT, se puede apreciar esta ejecución en el anexo 4 literal “h” y “i” respectivamente.

OBSERVACIONES: Mediante la realización de la consulta se obtuvo la información esperada probando así también la integridad de los datos.

Fuente: Víctor Colcha

✓ **Prueba de Desempeño: Búsqueda en Complejidad Media**

TABLA 3.XV PRUEBA BUSCAR DATOS DE COMPLEJIDAD MEDIA

PRUEBA No.7	BÚSQUEDA	COMPLEJIDAD MEDIA
--------------------	-----------------	--------------------------

OBJETIVO: Obtener el tiempo de respuesta de los SGBD PostgreSQL y Firebird al buscar y listar una determinada cantidad de datos mediante una consulta de complejidad mediana.

<p>PARÁMETROS:</p> <ul style="list-style-type: none"> ✓ Desempeño: Tiempo de selección ✓ Parámetro adicional integridad de datos <p>LECTURAS A OBTENER: 10</p>	<p>HERRAMIENTAS DE LOS SGBD:</p> <ul style="list-style-type: none"> ✓ PROPIAS: PgAdmin III IBExpert
--	--

ESCENARIOS DE PRUEBAS:	1000	10000	100000	1000000	10000000
-------------------------------	-------------	--------------	---------------	----------------	-----------------

PROCEDIMIENTO:

- ✓ Creación de los procedimientos almacenados para la consulta de **COMPLEJIDAD MEDIA** con sentencias SQL similares para cada SGBD.
- ✓ Ejecución de los procedimientos almacenados en la herramienta propia para cada SGBD.
- ✓ Obtención de los tiempos de respuesta que emiten cada SGBD.

CÓDIGO:

1. Código buscar datos de complejidad media en PostgreSQL, ver anexo 1, literal “k”
2. Código buscar datos de complejidad media en Firebird, ver anexo 1, literal “l”

EJECUCIÓN DE LA PRUEBA: La ejecución de esta prueba para PostgreSQL se lo ha realizado en PG ADMIN III y para FIREBIRD en IBEXPERT, se puede apreciar esta ejecución en el anexo 4 literal “h” y “i” respectivamente.

OBSERVACIONES: Mediante la realización de la consulta se obtuvo la información esperada probando así también la integridad de los datos.

Fuente: Víctor Colcha

✓ **Prueba de Desempeños : Búsqueda con Complejidad Alta**

TABLA 3.XVI PRUEBA BUSCAR DATOS DE COMPLEJIDAD ALTA

PRUEBA No.8	BÚSQUEDA	COMPLEJIDAD ALTA
--------------------	-----------------	-------------------------

OBJETIVO: Obtener el tiempo de respuesta de los SGBD PostgreSQL y Firebird al buscar y listar una determinada cantidad de datos mediante una consulta de complejidad alta

<p>PARÁMETROS:</p> <ul style="list-style-type: none"> ✓ Desempeño: Tiempo de selección ✓ Parámetro adicional integridad de datos <p>LECTURAS A OBTENER: 10</p>	<p>HERRAMIENTAS DE LOS SGBD:</p> <ul style="list-style-type: none"> ✓ PROPIAS: PgAdmin III IBExpert
--	--

ESCENARIOS DE PRUEBAS:	1000	10000	100000	1000000	10000000
-------------------------------	-------------	--------------	---------------	----------------	-----------------

PROCEDIMIENTO:

- ✓ Creación de los procedimientos almacenados para la consulta de **COMPLEJIDAD ALTA** con sentencias SQL similares para cada SGBD.
- ✓ Ejecución de los procedimientos almacenados en la herramienta propia para cada SGBD.
- ✓ Obtención de los tiempos de respuesta que emiten cada SGBD.

CÓDIGO:

1. Código buscar datos de complejidad alta en PostgreSQL, ver anexo 1, literal “m”
2. Código buscar datos de complejidad alta en Firebird, ver anexo 1, literal “n”

EJECUCIÓN DE LA PRUEBA: La ejecución de esta prueba para PostgreSQL se lo ha realizado en PG ADMIN III y para FIREBIRD en IBEXPERT se puede apreciar esta ejecución en el anexo 4 literal “h” y “i” respectivamente.

OBSERVACIONES: Mediante la realización de la consulta se obtuvo la información esperada probando así también la integridad de los datos.

Fuente: Víctor Colcha

3.10.2 Pruebas de rendimiento: Consumo de Recursos

✓ Prueba en consumo de recurso : Memoria RAM y el procesador

TABLA 3.XVII PRUEBA DE USO DE LA MEMORIA RAM Y PROCESADOR

PRUEBA No.9	MEMORIA RAM Y PROCESADOR
--------------------	---------------------------------

OBJETIVO: Obtener el uso de los recursos en memoria RAM y Procesador que generan los SGBD PostgreSQL y Firebird al realizar simultáneamente las 4 operaciones SQL.

<p>PARÁMETROS:</p> <ul style="list-style-type: none"> ✓ Consumo de recursos: Uso de RAM Uso Procesador ✓ Parámetro adicional integridad de datos <p>LECTURAS A OBTENER: 10</p>	<p>HERRAMIENTAS:</p> <ul style="list-style-type: none"> ✓ PgAdmin III ✓ IBExpert ✓ DBTools Manager Professional ✓ Administrador de tareas Windows
--	--

ESCENARIOS DE PRUEBAS:	1000	10000	100000	1000000	10000000
-------------------------------	-------------	--------------	---------------	----------------	-----------------

PROCEDIMIENTO:

- ✓ Creación de los procedimientos almacenados de las 4 operaciones para los dos SGBD con sentencias SQL similares.
- ✓ Ejecución de los procedimientos almacenados en las herramientas propias de cada SGBD.
- ✓ Obtención de los valores del consumo de memoria RAM y del procesador que genera los SGBD.
- ✓ Visualizar el registro mediante una consulta de datos en la herramienta externa.

CÓDIGO:

1. Código para el consumo de recursos en PostgreSQL, ver en el anexo 2, literal “a”
2. Código para el consumo de recursos en Firebird ver en el anexo 2, literal “b”

EJECUCIÓN DE LA PRUEBA: La ejecución de esta prueba para POSTGRESQL se lo ha realizado en PG ADMIN III y para FIREBIRD en IBEXPERT se puede apreciar esta ejecución en el anexo 4 literal “j” y “k” respectivamente.

Continuará: ...

Continúa: ...

OBSERVACIONES: Mediante la realización de una consulta con la herramienta externa se pudo medir la integridad y existencia de los datos.

Fuente: Víctor Colcha

3.10.3 Prueba Adicional: Robustez de los SGBD

TABLA 3.XVIII PRUEBA ADICIONAL

PRUEBA ADICIONAL	ROBUSTEZ DE LOS SGBD
------------------	----------------------

OBJETIVO: Obtener el tiempo de respuesta y comprobar la robustez que posee los SGBD PostgreSQL y Firebird al implementar código inconsistente mediante una consulta de complejidad simple.

PARÁMETROS:	HERRAMIENTAS:
✓ Parámetro adicional Robustez	✓ PgAdmin III
✓ Parámetro adicional integridad de datos	✓ IBExpert
LECTURAS A OBTENER: 3	

ESCENARIOS DE PRUEBAS:	1000	10000	100000	1000000	
------------------------	------	-------	--------	---------	--

PROCEDIMIENTO:

- ✓ Creación de una sola **SENTENCIA SQL** para la consulta de complejidad simple que servirá para los dos SGBD PostgreSQL y Firebird.
- ✓ Ejecución de la sentencia SQL en los diferentes SGBD.
- ✓ Obtención de los tiempos de respuesta que emiten cada SGBD.

CÓDIGO: El Código de los dos SGDB se encuentra en el anexo 2.1 literales “a” y “b”

✓ **EJECUCIÓN DE LA PRUEBA:** La ejecución de esta prueba para PostgreSQL se lo ha realizado en PG ADMIN III y para FIREBIRD en IBEXPERT y su comprobación el DBTools Manager.

OBSERVACIONES: La utilización la sentencia SQL dependió de los tiempos de respuestas de los datos obtenidos en todas las pruebas.

NOTA: Esta prueba no se involucra en la calificación final, solo servirá para definir las conclusiones y recomendaciones de esta investigación.

Fuente: Víctor Colcha

3.11 Resultados y Cuadros Comparativos

Para las distintas pruebas se tomaron 10 lecturas, por cada volumen de datos en los dos SGBD. La tabulación de las lecturas se lo hizo mediante la herramienta SPSS ESTADISTIC, para cada volumen de datos se obtuvieron los siguientes detalles:

N: Número de lecturas

Mínimo: Valor más pequeño.

Máximo: Valor más grande.

Media: Muestra el promedio aritmético; la suma dividida por el número de casos.

E.T. media: Error típico de la media, o sea la desviación típica de la distribución muestral de la media. Se obtiene dividiendo la desviación típica por la raíz cuadrada del número de casos. Es una medida de cuánto puede variar el valor de la media de una muestra a otra, extraídas éstas de la misma distribución.

Desviación típica: Medida de dispersión en torno a la media. Raíz cuadrada de la varianza. Mide el grado en que las puntuaciones de la variable se alejan de su media. Conjuntamente se procederá a su respectiva comparación de valores, su forma gráfica y la asignación del puntaje donde se aplicara las reglas de la tabla III.VII, la cual servirán para el análisis de resultados.

A continuación se definen las pruebas realizadas por cada uno de los indicadores

3.11.1 *Tiempo de Respuesta en Ingreso de Datos*

SGBD POSTGRESQL

En la Figura III.2 muestra los datos estadísticos tabulados por la herramienta SPSS referente a PostgreSQL, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “a”** del anexo 3.

FIGURA 3.2 POSTGRESQL - DATOS ESTADÍSTICOS DE INSERCIÓN

SGBD POSTGRESQL: DATOS ESTADÍSTICOS DE INSERCIÓN

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	4125	8345	5366,30	475,557	1503,842
10000	10	31627	55753	40208,40	2004,846	6339,878
100000	10	403858	465293	418585,00	5343,546	16897,777
1000000	10	4142122	4174688	4151583,00	2962,633	9368,668
10000000	10	53514547	59435323	54262029,10	577522,097	1826285,226
N válido (según lista)	10					

Fuente: Víctor Colcha

SGBD FIREBIRD

En la Figura III.3 muestra los datos estadísticos tabulados por la herramienta SPSS referente a Firebird, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “a”** del anexo 3.

FIGURA 3.3 FIREBIRD - DATOS ESTADÍSTICOS DE INSERCIÓN

SGBD FIREBIRD: DATOS ESTADÍSTICOS DE INSERCIÓN

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	578	781	663,80	16,814	53,170
10000	10	29141	32009	30485,00	287,510	909,185
100000	10	318515	346797	330278,40	2926,291	9253,746
1000000	10	4065478	4108571	4085220,50	5271,559	16670,134
10000000	10	32546754	32939032	32678050,10	31559,880	99801,104
N válido (según lista)	10					

Fuente: Víctor Colcha

Cuadro comparativo

En la Tabla III.XIX se observa la comparación de los dos SGBD con el indicador ingresar datos representados en milisegundos

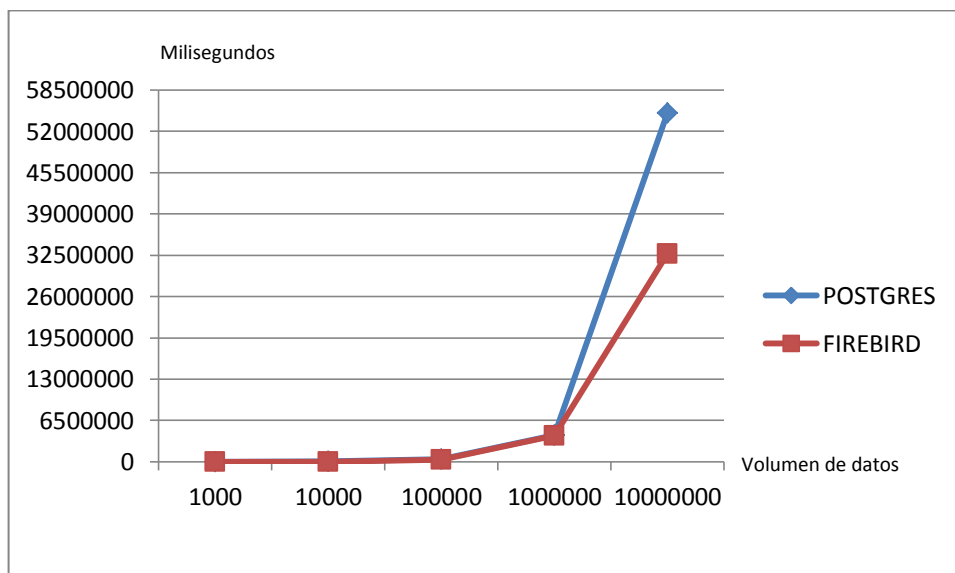
TABLA 3.XIX COMPARACIÓN - INGRESO DE DATOS PARA LOS SGBD

Inserción de datos		
Volumen de datos	PostgreSQL (ms)	Firebird (ms)
1000	5366,3	663,8
10000	40208,4	30485
100000	418585	330278,4
1000000	4151583	4085220,5
10000000	54262029,1	32678050,1

Fuente: Víctor Colcha

En el Figura III.4 se observa el comportamiento de los datos entre los dos SGBD PostgreSQL y Firebird.

FIGURA 3.4 COMPARACIÓN DE INGRESO DE DATOS DE DOS SGBD



Fuente: Víctor Colcha

Asignación del puntaje

Para la asignación del puntaje se hace referencia a la Tabla III.VIII del literal 3.9.4 del modelo para evaluar los resultados, donde se aplica dichas reglas. En la Tabla III.XX se muestra los puntajes con su equivalencia que se asignados para cada SGBD.

TABLA 3.XX PUNTAJE DE INGRESO DE DATOS PARA LOS DOS SGBD

Volumen de datos	POSTGRESQL			FIREBIRD		
	ms	Puntaje		ms	Puntaje	
1000	5366,3	2	Malo	663,8	10	Excelente
10000	40208,4	6	Bueno	30485	10	Excelente
100000	418585	6	Bueno	330278,4	10	Excelente
1000000	4151583	8	Muy buena	4085220,5	10	Excelente
10000000	54262029,1	4	Regular	32678050,1	10	Excelente
PROMEDIO	5.2			10		

Fuente: Víctor Colcha

Interpretación de resultados

Como se apreció en la Tabla III.XIX y observando la Figura III.4 los tiempos de respuesta van creciendo de acuerdo al volumen de datos a ingresar. Mostrando que Firebird globalmente supera aproximadamente con un 50 % de ventaja a PostgreSQL en todos los escenarios de prueba al brindar menor tiempo de respuesta, posesionándose como el mejor en este indicador.

Por lo cual en la Tabla III.XX se muestra por cada volumen de datos el puntaje de calificación asignados con su respectiva equivalencia, superando Firebird con 4.2 puntos de diferencia más rápido que PostgreSQL.

3.11.2 Tiempo de Respuesta en Actualizar Datos

SGBD POSTGRESQL

En la Figura III.5 muestra los datos estadísticos tabulados por la herramienta SPSS referente a PostgreSQL, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “b”** del anexo 3.

FIGURA 3.5 POSTGRESQL - DATOS ESTADÍSTICOS DE ACTUALIZACIÓN

SGBD POSTGRESQL: DATOS ESTADISTICOS DE ACTUALIZACIÓN

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	1906	2735	2115,50	73,673	232,975
10000	10	23076	25689	23849,50	246,976	781,007
100000	10	219356	248407	226932,20	2913,984	9214,828
1000000	10	1610650	2465939	1871904,80	81465,665	257617,054
10000000	10	23159243	24078895	23530934,80	72401,860	228954,786
N válido (según lista)	10					

Fuente: Víctor Colcha

SGBD FIREBIRD

En la Figura III.6 se muestra los datos estadísticos tabulados por la herramienta SPSS referente al SGBD Firebird, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “b”** del anexo 3.

FIGURA 3.6 FIREBIRD - DATOS ESTADÍSTICOS ACTUALIZAR

SGBD FIREBIRD: DATOS ESTADISTICOS DE ACTUALIZACIÓN

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	906	1219	1069,20	29,110	92,055
10000	10	9766	12953	11944,80	264,752	837,218
100000	10	133031	166487	154110,00	3736,634	11816,275
1000000	10	1213361	1253406	1231810,30	3403,712	10763,483
10000000	10	20323423	20463215	20360314,90	14018,861	44331,530
N válido (según lista)	10					

Fuente: Víctor Colcha

Tabla comparativa

En la Tabla III.XXI se observa la comparación de los dos SGBD con el indicador ingresar datos representados en milisegundos.

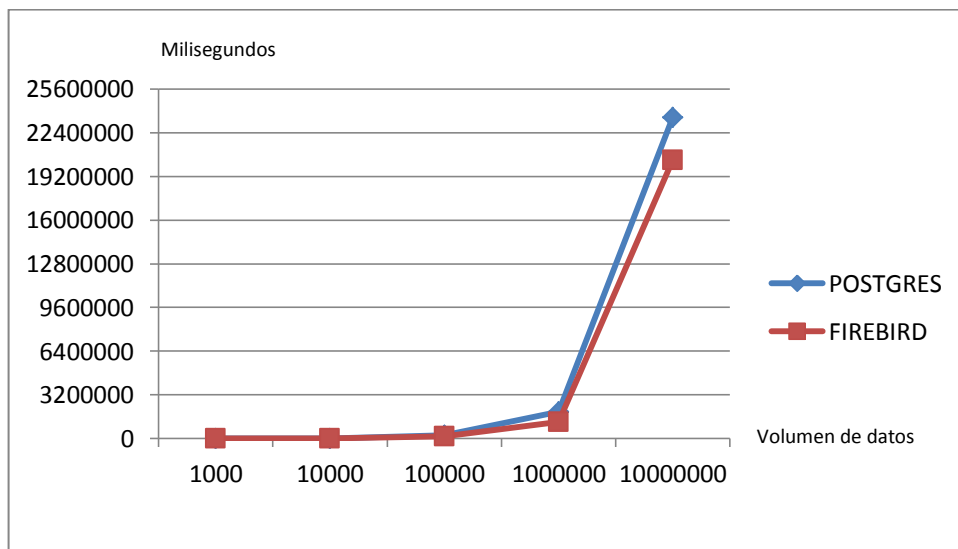
TABLA 3.XXI COMPARACIÓN ACTUALIZAR DATOS DE LOS DOS SGBD

Actualización de datos		
Volumen de datos	PostgreSQL(ms)	Firebird (ms)
1000	2115,5	1069,2
10000	23849,5	11944,8
100000	226932,2	154110
1000000	1871904,8	1231810,3
10000000	23530934,8	20360314,9

Fuente: Autor

En la Figura III.7 se observa el comportamiento de los datos entre los dos SGBD PostgreSQL y Firebird.

FIGURA 3.7 COMPARACIÓN DE DATOS DE ACTUALIZAR EN SGBD



Fuente: Víctor Colcha

Asignación del puntaje

Para la asignación del puntaje se hace referencia a la tabla III.VIII del literal 3.9.4 del modelo para evaluar los resultados donde se aplica dichas reglas. . En la Tabla III.XXII se muestra los puntajes asignados.

TABLA 3.XXII ASIGNACIÓN DE PUNTAJE ACTUALIZAR DATOS EN SGBD

Volumen de datos	POSTGRESQL			FIREBIRD		
	ms	Puntaje		Ms	Puntaje	
1000	2115,5	4	Regular	1069,2	10	Excelente
10000	23849,5	4	Regular	11944,8	10	Excelente
100000	226932,2	6	buena	154110	10	Excelente
1000000	1871904,8	6	buena	1231810,3	10	Excelente
10000000	23530934,8	8	Muy buena	20360314,9	10	Excelente
PROMEDIO		5.6			10	

Fuente: Víctor Colcha

Interpretación de resultados

Como se apreció en la Tabla III.XXI y observando la Figura III.7 los tiempos de respuesta van creciendo de acuerdo al volumen de datos a ser actualizados. Mostrando que Firebird en los escenarios individuales hasta el 1000000 datos supera aproximadamente con un 50 % de ventaja a PostgreSQL, superando también en los 10000000 de datos pero con un nivel de diferencia menor, posesionándose Firebird nuevamente como el mejor en este indicador a al brindar menor tiempo de respuesta.

Por lo cual en la Tabla III.XXII se muestra por cada volumen de datos el puntaje de calificación asignados con su respectiva equivalencia, superando Firebird con 4.4 puntos de diferencia más rápido que PostgreSQL.

3.11.3 Tiempo de Respuesta Listar Datos Diversa Complejidad

SGBD POSTGRESQL

Complejidad Simple: En la Figura III.8 muestra los datos estadísticos tabulados por la herramienta SPSS referente al SGBD PostgreSQL, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “c”** del anexo 3.

FIGURA 3.8 POSTGRESQL – LISTAR COMPLEJIDAD SIMPLE**SGBD POSTGRESQL: DATOS ESTADISTICOS LISTAR COMPLEJIDAD SIMPLE**

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	16	172	49,80	13,935	44,068
10000	10	0	1000	500,00	166,667	527,046
100000	10	2000	6000	2700,00	395,811	1251,666
1000000	10	36000	47000	38900,00	1048,279	3314,949
10000000	10	2884000	3034000	2947300,00	14372,079	45448,506
N válido (según lista)	10					

Fuente: Víctor Colcha

Complejidad Mediana: En la Figura III.9 muestra los datos estadísticos tabulados por la herramienta SPSS referente al SGBD PostgreSQL, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “c”** del anexo 3.

FIGURA 3.9 POSTGRESQL – LISTAR DATOS COMPLEJIDAD MEDIA**SGBD POSTGRESQL: DATOS ESTADISTICOS LISTAR COMPLEJIDAD MEDIA**

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	47	765	146,60	68,837	217,681
10000	10	0	1000	700,00	152,753	483,046
100000	10	6000	9000	6600,00	305,505	966,092
1000000	10	68000	133000	91900,00	6224,414	19683,326
10000000	10	11891000	13902000	12387800,00	198648,702	628182,351
N válido (según lista)	10					

Fuente: Víctor Colcha

Complejidad Alta: En la Figura III.10 muestra los datos estadísticos tabulados por la herramienta SPSS referente al SGBD PostgreSQL, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “d”** del anexo 3.

FIGURA 3.10 POSTGRESQL – LISTAR DATOS COMPLEJIDAD ALTA**SGBD POSTGRESQL: DATOS ESTADISTICOS LISTAR COMPLEJIDAD ALTA**

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	203	1688	367,50	146,782	464,166
10000	10	0	2000	900,00	179,505	567,646
100000	10	20000	29000	24300,00	700,000	2213,594
1000000	10	267000	407000	288100,00	13293,649	42038,210
10000000	10	20743000	22838000	21264100,00	194484,415	615013,722
N válido (según lista)	10					

Fuente: Víctor Colcha

Tabla Comparativa

En la Tabla III.XXIII se observa la comparación del SGBD PostgreSQL del indicador listar datos con diversidad compleja representados en milisegundos.

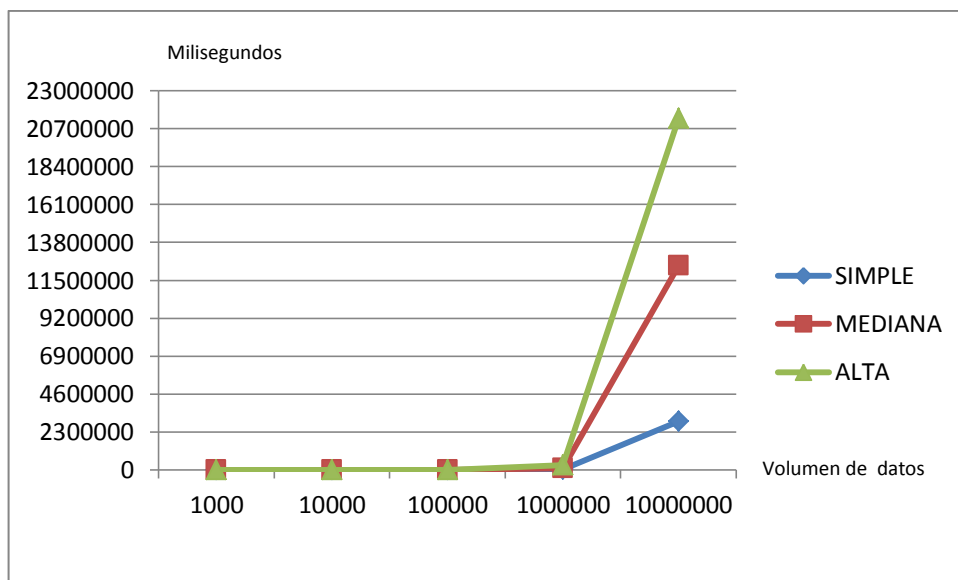
TABLA 3.XXIII POSTGRESQL – LISTAR DATOS DIVERSA COMPLEJIDAD

Consulta listar datos de diversa complejidad			
Volumen de datos	Simple	Mediana	Alta
1000	49,8	146,60	367,5
10000	500	700	900
100000	2700	6600	24300
1000000	38900	91900	288100
10000000	2947300	12387800	21264100

Fuente: Victor Colcha

En la Figura III.11 se observa el comportamiento de los datos entre los dos SGBD PostgreSQL y Firebird.

FIGURA 3.11 POSTGRESQL- COMPARACIÓN LISTA DE DATOS COMPLEJIDAD DIVERSA



Fuente: Victor Colcha

Interpretación de resultados

Como se apreció en la Tabla III.XXIII y observando la Figura II.11 los tiempos de respuesta van creciendo de acuerdo al volumen de datos y del grado de complejidad de

las consultas. Es decir tiene una relación directamente proporcional, mientras más volumen de datos y un alto grado de complejidad los tiempos de respuesta son mayores.

Cabe mencionar que la estructura de las sentencias SQL con distintos grados de complejidad solo se relacionan las tablas no poseen condiciones de comparación y es ejecutada en una herramienta externa llamada DBTools Manager para SGBD PostgreSQL.

SGBD FIREBIRD

Complejidad Simple: En la figura III.12 muestra los datos estadísticos tabulados por la herramienta SPSS referente al SGBD Firebird, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “e”** del anexo 3.

FIGURA 3.12 FIREBIRD – LISTAR DATOS COMPLEJIDAD SIMPLE

SGBD FIREBIRD: DATOS ESTADISTICOS LISTAR COMPLEJIDAD SIMPLE

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	62	156	78,20	8,965	28,350
10000	10	1000	2000	1200,00	133,333	421,637
100000	10	10000	15000	11800,00	442,217	1398,412
1000000	10	174000	230000	202600,00	5844,656	18482,424
10000000	10	3597000	3831000	3693700,00	22284,549	70469,931
N válido (según lista)	10					

Fuente: Víctor Colcha

Complejidad Media: En la Figura III.13 muestra los datos estadísticos tabulados por la herramienta SPSS referente al SGBD Firebird, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “e”** del anexo 3.

FIGURA 3.13 FIREBIRD –LISTAR DATOS COMPLEJIDAD MEDIA

SGBD FIREBIRD: DATOS ESTADISTICOS LISTAR COMPLEJIDAD MEDIA

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	109	175	119,10	6,423	20,311
10000	10	1000	3000	2100,00	233,333	737,865
100000	10	21000	23000	21700,00	213,437	674,949
1000000	10	317000	363000	344100,00	4083,707	12913,817
10000000	10	14701000	15970000	14927400,00	117243,071	370755,145
N válido (según lista)	10					

Fuente: Víctor Colcha

Complejidad Alta: En la figura III.14 muestra los datos estadísticos tabulados por la herramienta SPSS referente al SGBD Firebird, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “f”** del anexo 3.

FIGURA 3.14 FIREBIRD – LISTAR DATOS COMPLEJIDAD ALTA
SGBD FIREBIRD: DATOS ESTADISTICOS LISTAR COMPLEJIDAD ALTA

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	156	188	170,60	2,960	9,359
10000	10	2000	3000	2400,00	163,299	516,398
100000	10	29000	35000	31300,00	597,216	1888,562
1000000	10	578000	743000	630500,00	14822,843	46873,944
10000000	10	29412000	30036000	29570000,00	54603,215	172670,528
N válido (según lista)	10					

Fuente: Víctor Colcha

Cuadro comparativo

En la Tabla III.XXIV se observa la comparación en Firebird del indicador listar datos con diversidad compleja representados en milisegundos

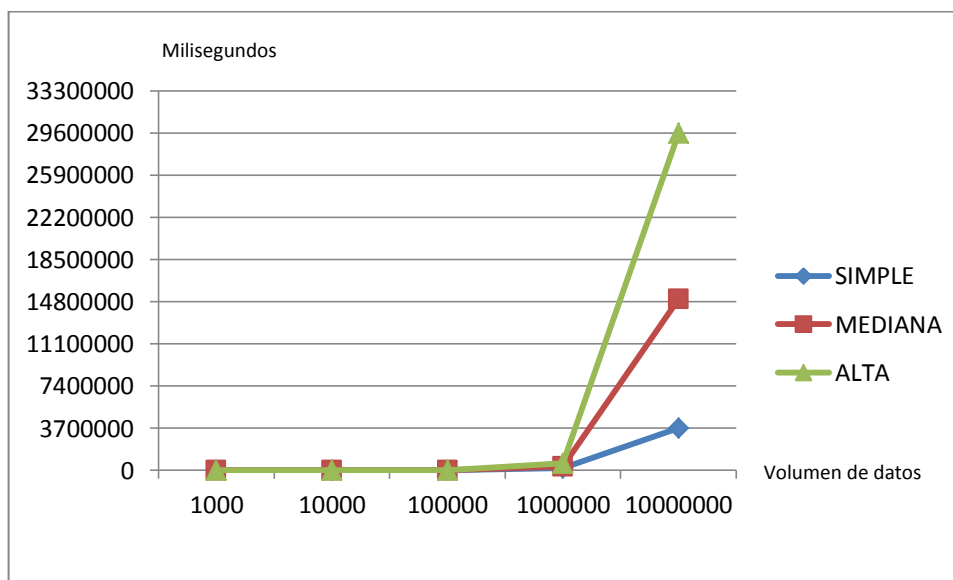
TABLA 3.XXIV FIREBIRD - LISTAR DATOS DE DIVERSA COMPLEJIDAD
Consulta listar datos de diversa complejidad

Volumen de datos	Simple	Mediana	Alta
1000	78,2	119	170
10000	1200	2100	2400
100000	11800	21700	31300
1000000	202600	344100	630500
10000000	3693700	14927400	29570000

Fuente: Autor

En la Figura III.15 se observa el comportamiento de los datos entre los dos SGBD PostgreSQL y Firebird.

FIGURA 3.15 FIREBIRD - COMPARACIÓN DE LISTAR DATOS DE DIVERSA COMPLEJIDAD



Fuente: Víctor Colcha

Interpretación de resultados

Como se mencionó anteriormente en los resultados de PostgreSQL, también se puede apreciar en la tabla III.XXIV y observar en la Figura III.15 los tiempos de respuesta van creciendo de acuerdo al volumen de datos y del grado de complejidad de las consultas. Es decir tiene una relación directamente proporcional, mientras más volumen de datos y un alto grado de complejidad los tiempos de respuesta son mayores.

Cabe mencionar que la estructura de las sentencias SQL con distintos grados de complejidad solo se relacionan las tablas no poseen condiciones de comparación y es ejecutada en una herramienta externa llamada DBTools Manager para el SGBD Firebird.

3.11.4 Comparación Listar Datos de Diversa Complejidad para los dos SGBD

Cuadro comparativo listar datos de complejidad simple con los dos SGBD

En la Tabla III.XXV se observa la comparación de los dos SGBD con el indicador listar datos con complejidad simple representados en milisegundos.

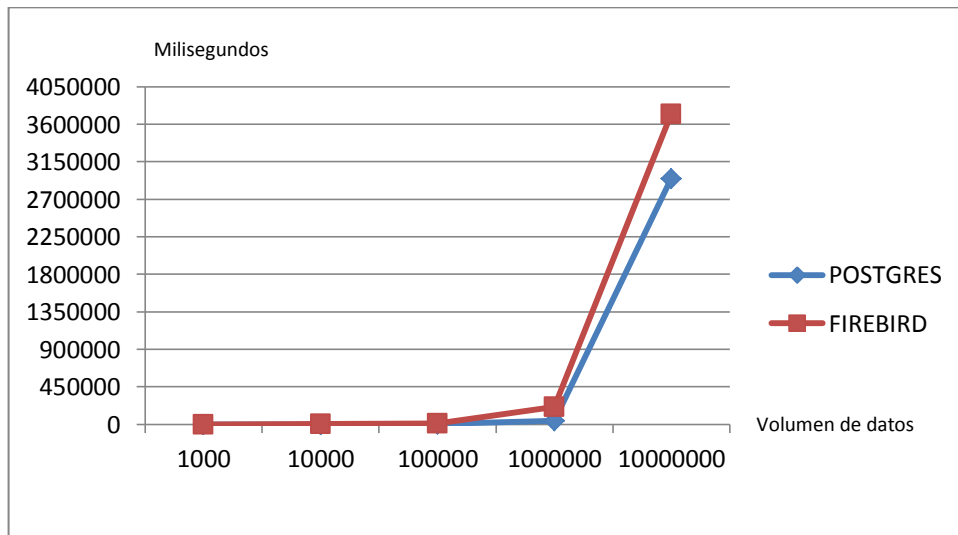
TABLA 3.XXV LISTAR DATOS DE COMPLEJIDAD SIMPLE DE LOS SGBD

Lista de complejidad simple		
Volumen de datos	PostgreSQL (ms)	Firebird (ms)
1000	49,8	78,2
10000	500	1200
100000	2700	11800
1000000	38900	202600
10000000	2947300	3693700

Fuente: Víctor Colcha

En la Figura III.16 se observa el comportamiento de los datos entre los dos SGBD PostgreSQL y Firebird.

FIGURA 3.16 COMPARACIÓN LISTAR DATOS COMPLEJIDAD SIMPLE



Fuente: Víctor Colcha

Asignación del puntaje

Para la asignación del puntaje se hace referencia a la tabla III.VIII del literal 3.9.4 del Modelo para evaluar los resultados donde se aplica dichas reglas.

TABLA 3.XXVI PUNTAJE PARA LISTAR DATOS DE LOS SGBD

Volumen de datos	POSTGRESQL(puntos)			FIREBIRD(puntos)		
	ms	Puntaje		ms	Puntaje	
1000	49,8	10	Excelente	78,2	6	Buena
10000	500	10	Excelente	1200	2	Malo
100000	2700	10	Excelente	11800	2	Malo
1000000	38900	10	Excelente	202600	2	Malo
10000000	2947300	10	Excelente	3693700	6	Buena
PROMEDIO	10			3.6		

Fuente: Víctor Colcha

Interpretación de resultados

Como se aprecia en la tabla III.XXV y observa la Figura III.16 los tiempos de respuesta van creciendo de acuerdo al volumen de datos para ser listados. Mostrando que PostgreSQL globalmente supera aproximadamente con un 64 % de ventaja a Firebird, posesionándose PostgreSQL como el mejor en este grado de complejidad simple al brindar menor tiempo de respuesta.

Por lo cual en la tabla III.XXVI se muestra por cada volumen de datos el puntaje de calificación asignados con su respectiva equivalencia, superando PostgreSQL con 6.4 puntos de diferencia más rápido que Firebird.

Cuadro comparativo de listar datos de complejidad media

En la Tabla III.XXVII se observa la comparación de los dos SGBD con el indicador listar datos de complejidad media representados en milisegundos

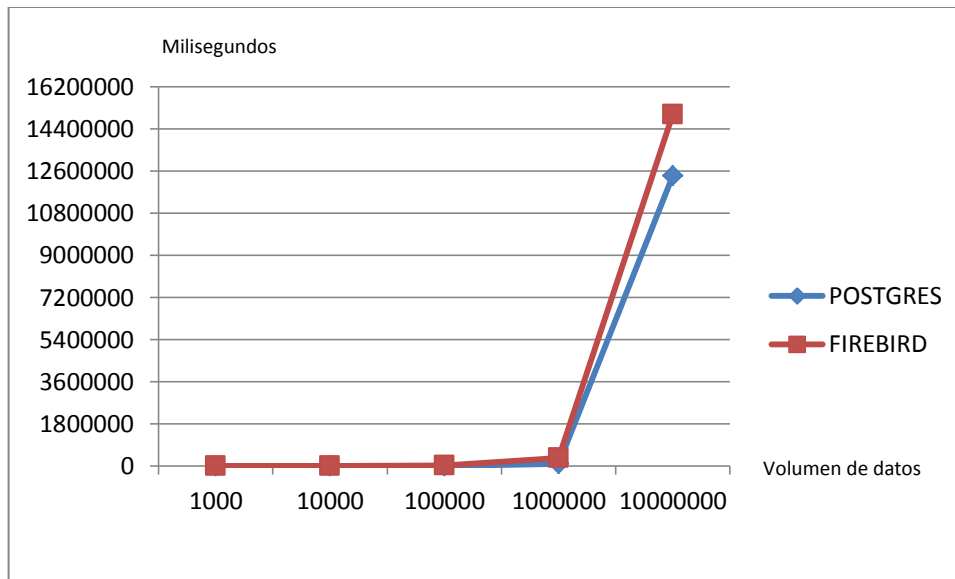
TABLA 3.XXVII LISTAR DATOS DE COMPLEJIDAD MEDIA DE LOS SGBD

Consulta listar de complejidad media		
Volumen de datos	PostgreSQL(ms)	Firebird(ms)
1000	146,6	119,1
10000	700	2100
100000	6600	21700
1000000	91900	344100
10000000	12387800	14927400

Fuente: Víctor Colcha

En la Figura III.17 se observa el comportamiento de los datos entre los dos SGBD PostgreSQL y Firebird.

FIGURA 3.17 LISTAR DATOS DE LA COMPLEJIDAD MEDIA



Fuente: Víctor Colcha

Asignación del puntaje

Para la asignación del puntaje se hace referencia a la tabla III.VIII del literal 3.9.4 del Modelo para evaluar los resultados donde se aplica dichas reglas.

TABLA 3.XXVIII PUNTAJE - LISTAR DATOS DE COMPLEJIDAD MEDIA

Volumen de datos	POSTGRESQL(puntos)		FIREBIRD(puntos)			
	ms	Puntaje	ms	Puntaje		
1000	146,6	6	Excelente	119	10	Buena
10000	700	10	Excelente	2100	2	Malo
100000	6600	10	Excelente	21700	2	Malo
1000000	91900	10	Excelente	344100	2	Malo
10000000	12387800	10	Excelente	14927400	6	Buena
PROMEDIO		9.2			4.4	

Fuente: Víctor Colcha

Interpretación de resultados

Como se aprecia en la tabla III.XXVII y observa la Figura III.17 los tiempos de respuesta de los dos SGBD van creciendo de acuerdo al volumen de datos para ser listados. Mostrando que Firebird en el primer escenario individual de 1000 datos supera aproximadamente con un 40% de ventaja a Firebird, pero PostgreSQL desde los 10000 hasta 10000000 datos supera enormemente con un 70% de ventaja a Firebird, posesionándose PostgreSQL nuevamente como el mejor en este grado de complejidad media a al brindar menor tiempo de respuesta.

Por lo cual en la tabla III.XXVIII se muestra por cada volumen de datos el puntaje de calificación asignados con su respectiva equivalencia, superando PostgreSQL aproximadamente con 4.8 puntos de diferencia más rápido que Firebird.

Cuadro Comparativo de listar datos de complejidad alta

En la Tabla III.XXIX se observa la comparación de los dos SGBD con el indicador listar datos con complejidad alta representados en milisegundos

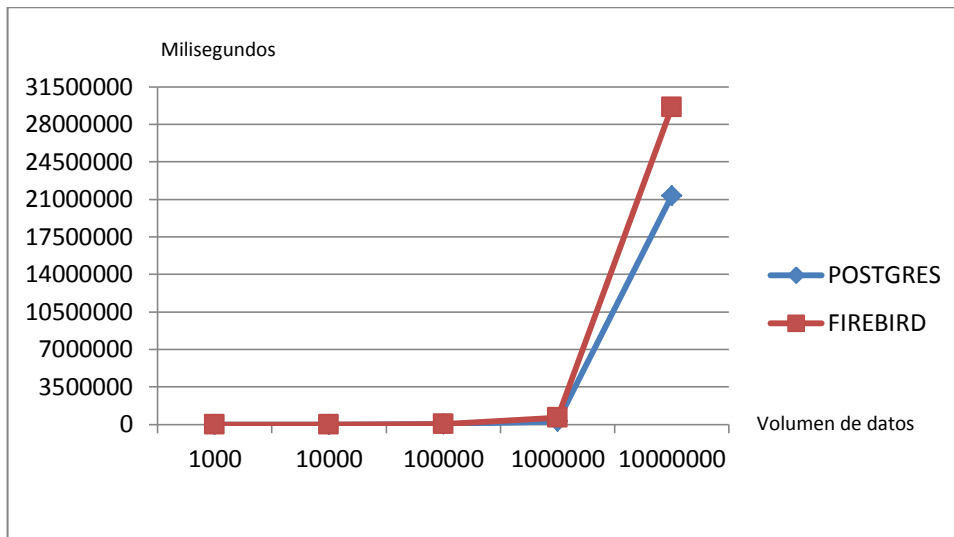
TABLA 3.XXIX COMPARACIÓN-LISTAR DATOS COMPLEJIDAD DE ALTA

Consulta listar de complejidad alta		
Volumen de datos	PostgreSQL (ms)	Firebird(ms)
1000	367,5	170,6
10000	900	2400
100000	24300	31300
1000000	288100	630500
10000000	21264100	29570000

Fuente: Víctor Colcha

En la Figura III.18 se observa el comportamiento de los datos entre los dos SGBD PostgreSQL y Firebird.

FIGURA 3.18 LISTAR DATOS DE COMPLEJIDAD ALTA



Fuente: Víctor Colcha

Asignación del puntaje

Para la asignación del puntaje se hace referencia a la tabla III.VIII del literal 3.9.4 del modelo para evaluar los resultados donde se aplica dichas reglas.

TABLA 3.XXX PUNTAJE- LISTAR DATOS DE COMPLEJIDAD ALTA

Volumen de datos	POSTGRESQL		FIREBIRD	
	ms	Puntaje	ms	Puntaje
1000	367,5	2 Malo	170,6	10 Excelente
10000	900	10 Excelente	2400	2 Malo
100000	24300	10 Excelente	31300	6 Buena
1000000	288100	10 Excelente	630500	2 Malo
10000000	21264100	10 Excelente	29570000	6 Buena
PROMEDIO		8.4		5.2

Fuente: Víctor Colcha

Interpretación de resultados

Como se aprecia en la tabla III.XXIX y observa la Figura III.18 los tiempos de respuesta de los dos SGBD van creciendo de acuerdo al volumen de datos para ser listados. Mostrando que Firebird en el primer escenario individual de 1000 datos supera aproximadamente con un 80 % de ventaja a Firebird, pero PostgreSQL desde los 10000 hasta 10000000 datos supera aproximadamente con un 40 % de ventaja a Firebird,

poseionándose PostgreSQL nuevamente como el mejor en este grado de complejidad alta a al brindar menor tiempo de respuesta.

Por lo cual en la Tabla III.XXX se muestra por cada volumen de datos el puntaje de calificación asignados con su respectiva equivalencia, superando PostgreSQL aproximadamente con 3.2 puntos de diferencia más rápido que Firebird.

3.11.5 Tiempo de respuesta en buscar datos de diversa complejidad.

SGBD POSTGRESQL

Complejidad Simple: En la Figura III.19 muestra los datos estadísticos tabulados por la herramienta SPSS referente al SGBD PostgreSQL, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “g”** del anexo 3.

FIGURA 3.19 POSTGRESQL- BUSCAR DATOS COMPLEJIDAD SIMPLE

SGBD POSTGRESQL: DATOS ESTADISTICOS BUSCAR COMPLEJIDAD SIMPLE

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	16	359	61,00	33,172	104,900
10000	10	109	1141	220,10	102,340	323,626
100000	10	1641	4741	2691,10	272,046	860,285
1000000	10	17548	23220	20204,10	607,197	1920,124
10000000	10	1160232	1621153	1334691,00	39020,747	123394,437
N válido (según lista)	10					

Fuente: Víctor Colcha

Complejidad Media: En la Figura III.20 muestra los datos estadísticos tabulados por la herramienta SPSS referente al SGBD PostgreSQL, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “g”** del anexo 3.

FIGURA 3.20 POSTGRESQL- BUSCAR DATOS COMPLEJIDAD MEDIA**SGBD POSTGRESQL: DATOS ESTADISTICOS BUSCAR COMPLEJIDAD MEDIA**

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	31	610	92,40	57,548	181,982
10000	10	203	1540	386,80	128,355	405,895
100000	10	2329	6032	3464,80	435,479	1377,107
1000000	10	29987	44190	37011,70	1357,158	4291,711
10000000	10	1404493	1787248	1551908,70	30740,783	97210,891
N válido (según lista)	10					

Fuente: Víctor Colcha

Complejidad Alta: En la Figura III.21 muestra los datos estadísticos tabulados por la herramienta SPSS referente al SGBD PostgreSQL, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “h”** del anexo 3.

FIGURA 3.21 POSTGRESQL- BUSCAR DATOS COMPLEJIDAD ALTA**SGBD POSTGRESQL: DATOS ESTADISTICOS BUSCAR COMPLEJIDAD ALTA**

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	140	234	164,10	8,355	26,422
10000	10	1110	1226	1146,30	10,416	32,938
100000	10	12611	18021	14056,20	469,037	1483,225
1000000	10	145071	190792	161294,50	4095,973	12952,605
10000000	10	2341345	2503520	2375348,00	14872,829	47032,015
N válido (según lista)	10					

Fuente: Víctor Colcha

Cuadro comparativo del SGBD PostgreSQL con diversa complejidad

En la Tabla III.XXXI se observa la comparación del SGBD PostgreSQL con el indicador buscar datos con diversa complejidad representados en milisegundos.

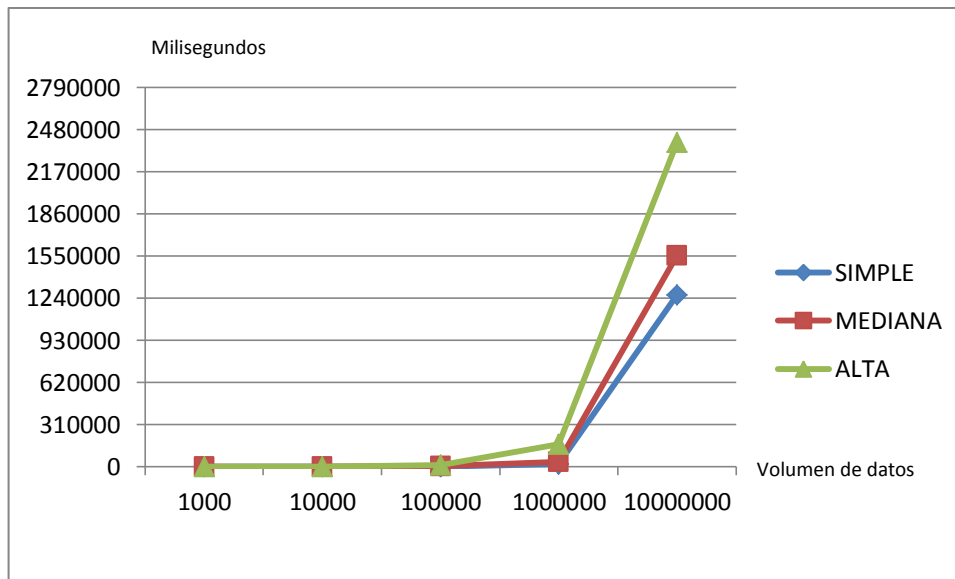
TABLA 3.XXXI COMPARACIÓN BUSCAR DE DIVERSA COMPLEJIDAD

Consulta búsqueda de diversa complejidad			
Volumen de datos	Simple	Mediana	Alta
1000	61	92,4	164,1
10000	220,1	386,8	1146,3
100000	2691,1	3464,8	13986,2
1000000	20204,1	37011,7	161294,5
10000000	1334691	1551908,7	2375348

Fuente: Víctor Colcha

En el grafico número III.22 se observa el comportamiento de los datos entre los dos SGBD PostgreSQL y Firebird.

FIGURA 3.22 POSTGRESQL - COMPARACIÓN BUSCAR DATOS



Fuente: Víctor Colcha

Interpretación de resultados

Se puede apreciar en la tabla III.XXXI y observar en la Figura III.22 los tiempos de respuesta van creciendo de acuerdo al volumen de datos y del grado de complejidad de las consultas. Es decir tiene una relación directamente proporcional, mientras más volumen de datos y un alto grado de complejidad los tiempos de respuesta son mayores.

Cabe mencionar que la estructura de las sentencias SQL con distintos grados de complejidad en este caso ya poseen condiciones de comparación y es ejecutada en la propia herramienta diseñada para el SGBD PostgreSQL. Por ende los resultados de los tiempos ya son de proporciones menores comparados al indicador lista de datos.

SGBD FIREBIRD

Complejidad Simple: En la Figura III.23 se muestra los datos estadísticos tabulados por la herramienta SPSS referente al SGBD Firebird, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “i”** del anexo 3.

FIGURA 3.23 FIREBIRD DATOS ESTADÍSTICOS

SGBD FIREBIRD: DATOS ESTADÍSTICOS BUSCAR COMPLEJIDAD SIMPLE

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	15	78	26,50	6,163	19,489
10000	10	167	519	215,50	33,795	106,869
100000	10	1869	7010	2522,00	499,768	1580,406
1000000	10	25641	43484	28565,80	1758,807	5561,837
10000000	10	643025	700354	666560,80	5656,116	17886,208
N válido (según lista)	10					

Fuente: Víctor Colcha

Complejidad Mediana: En la Figura III.24 se muestra los datos estadísticos tabulados por la herramienta SPSS referente al SGBD Firebird, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “i”** del anexo 3.

FIGURA 3.24 FIREBIRD- BUSCAR DATOS COMPLEJIDAD MEDIA

SGBD FIREBIRD: DATOS ESTADÍSTICOS BUSCAR COMPLEJIDAD MEDIA

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	31	63	34,60	3,159	9,991
10000	10	303	400	320,80	9,360	29,600
100000	10	3465	7766	4023,70	427,126	1350,692
1000000	10	67594	101516	89532,70	2757,911	8721,281
10000000	10	1224531	1564321	1304074,30	29678,259	93850,895
N válido (según lista)	10					

Fuente: Víctor Colcha

Complejidad Alta: En la figura III.25 muestra los datos estadísticos tabulados por la herramienta SPSS referente al SGBD Firebird, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “j”** del anexo 3.

FIGURA 3.25 FIREBIRD- BUSCAR DATOS COMPLEJIDAD ALTA

SGBD FIREBIRD: DATOS ESTADÍSTICOS BUSCAR COMPLEJIDAD ALTA

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	78	125	92,20	4,149	13,122
10000	10	921	1062	971,90	11,912	37,669
100000	10	10109	17500	11636,70	696,406	2202,229
1000000	10	180250	222157	198767,30	3746,802	11848,429
10000000	10	2795531	3053906	2896933,20	25399,695	80320,889
N válido (según lista)	10					

Fuente: Víctor Colcha

Cuadro comparativo

En la Tabla III.XXXII se observa la comparación del SGBD Firebird con el indicador buscar datos con diversa complejidad representados en milisegundos

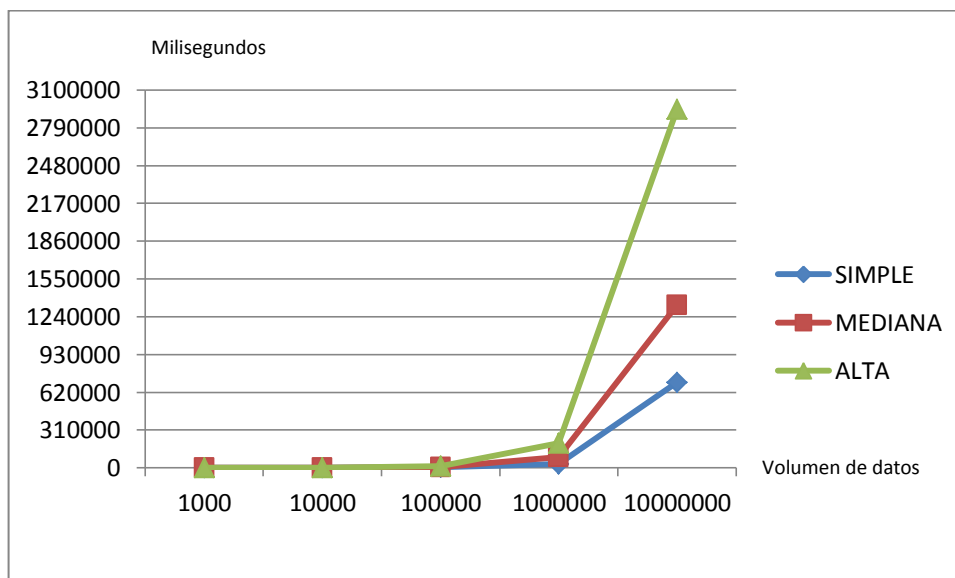
TABLA 3.XXXII BÚSQUEDA-CUADRO COMPARATIVO

Consulta búsqueda de diversa complejidad			
Volumen de datos	Simple	Mediana	Alta
1000	26,5	34,6	92,2
10000	215,5	320,8	971,9
100000	2522	4023,7	11636,7
1000000	28565,8	89532,7	198767,3
10000000	666560,6	1304074,3	2896933,2

Fuente: Autor

En la Figura III.26 se observa el comportamiento de los datos entre los dos SGBD.

FIGURA 3.26 FIREBIRD- BUSCAR DATOS DE DIVERSA COMPLEJIDAD



Fuente: Víctor Colcha

Interpretación de resultados

Como se mencionó anteriormente en los resultados de PostgreSQL de este indicador se puede apreciar en la tabla III.XXXII y observar en la Figura III.26 los tiempos de respuesta van creciendo de acuerdo al volumen de datos y del grado de complejidad de

las consultas. Es decir tiene una relación directamente proporcional, mientras más volumen de datos y un alto grado de complejidad los tiempos de respuesta son mayores.

Cabe mencionar que la estructura de las sentencias SQL con distintos grados de complejidad en este caso ya poseen condiciones de comparación y es ejecutada en la propia herramienta diseñada para el SGBD Firebird. Por ende los resultados de los tiempos ya son de proporciones menores comparados al indicador lista de datos.

3.11.6 Comparación Buscar datos de Diversa Complejidad para dos Dos SGBD

En la Tabla III.XXXIII se observa la comparación de los dos SGBD con el indicador listar datos con complejidad simple representados en milisegundos.

TABLA 3.XXXIII RESULTADOS COMPARATIVOS DIVERSA COMPLEJIDAD

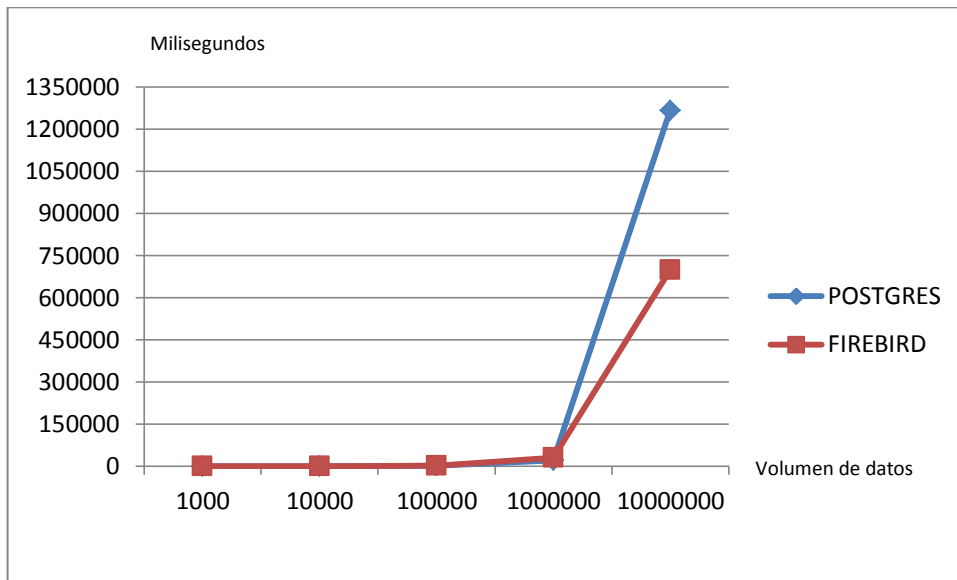
Consulta búsqueda de complejidad simple			
Volumen de datos	PostgreSQL (ms)	Firebird(ms)	Salida de datos
10000	61	26,5	250
100000	220,1	215,5	2500
1000000	2691,1	2522	25000
10000000	20204,1	28565,8	250000
100000000	1334691	666560,6	2500000

Fuente: Víctor Colcha

Grafica estadística

En la Figura III.27 se observa el comportamiento de los datos entre los dos SGBD PostgreSQL y Firebird.

FIGURA 3.27 RESULTADO BUSCAR DATOS COMPLEJIDAD SIMPLE



Fuente: Víctor Colcha

Asignación del puntaje

Para la asignación del puntaje se hace referencia a la tabla III.VIII del literal 3.9.4 del modelo para evaluar los resultados donde se aplica dichas reglas.

TABLA 3.XXXIV ASIGNACIÓN DEL PUNTAJE BÚSQUEDA SIMPLE

Volumen de datos	POSTGRESQL		FIREBIRD	
	ms	Puntaje	ms	Puntaje
1000	61	2 Malo	26,5	10 Excelente
10000	220,1	8 Muy buena	215,5	10 Excelente
100000	2691,1	10 Excelente	2522	8 Muy Buena
1000000	20204,1	10 Excelente	28565,8	6 Buena
10000000	1334691	2 Malo	666560,6	10 Excelente
PROMEDIO		6.4		8.8

Fuente: Víctor Colcha

Interpretación de resultados

Como se aprecia en la Tabla III.XXXIII y observa la Figura 27 los tiempos de respuesta de los dos SGBD van creciendo de acuerdo al volumen de datos en el que van hacer buscados. Mostrando que Firebird en los escenario individual de 1000, 10000 y

10000000 datos supera aproximadamente con un 60 % de ventaja a PostgreSQL, pero en los 100000 y 1000000 datos PostgreSQL supera aproximadamente con un 30 % de ventajas. Posesionándose Firebird como el mejor en este grado de complejidad simple a al brindar menor tiempo de respuesta.

Por lo cual en la Tabla III.XXXIV se muestra por cada volumen de datos el puntaje de calificación asignados con su respectiva equivalencia, superando Firebird aproximadamente con 2.4 puntos de diferencia más rápido que PostgreSQL.

Cuadro comparativo buscar datos de complejidad media

En la Tabla III.XXXV se observa la comparación de los dos SGBD con el indicador buscar datos con complejidad media representados en milisegundos.

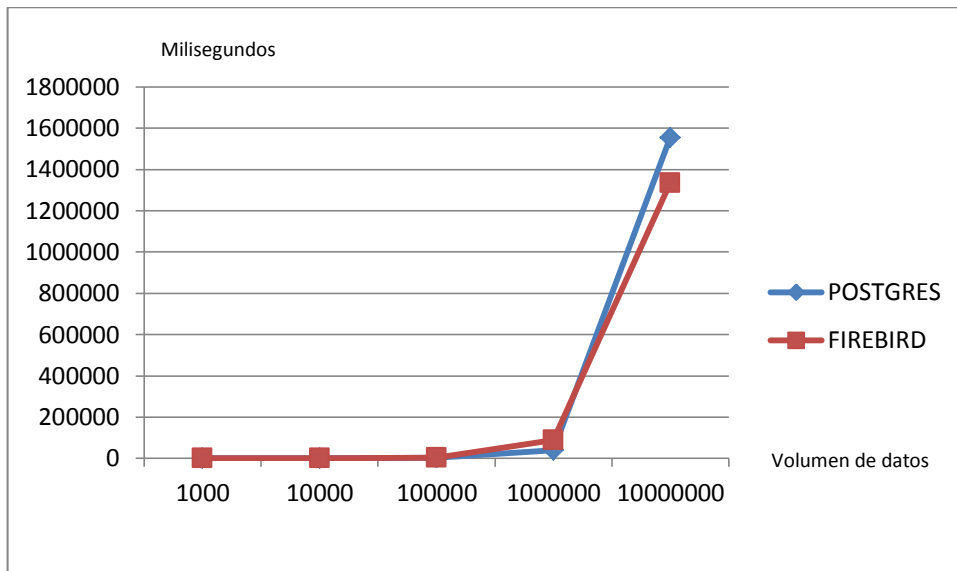
TABLA 3.XXXV CONSULTA BÚSQUEDA DE COMPLEJIDAD MEDIA

Consulta búsqueda de complejidad media			
Volumen de datos	PostgreSQL(ms)	Firebird(ms)	Salida de datos
10000	92,4	34,6	146
100000	386,8	320,8	1564
1000000	3464,8	4023,7	16576
10000000	37011,7	89532,7	174184
100000000	1551908,7	1304074,3	1817656

Fuente: Víctor Colcha

En la Figura III.28 se observa el comportamiento de los datos entre los dos SGBD PostgreSQL y Firebird.

FIGURA 3.28 BÚSQUEDA DE COMPLEJIDAD MEDIA



Fuente: Víctor Colcha

Asignación del puntaje

Para la asignación del puntaje se hace referencia a la tabla III.VIII del literal 3.9.4 del Modelo para evaluar los resultados donde se aplica dichas reglas.

TABLA 3.XXXVI ASIGNACIÓN DEL PUNTAJE BÚSQUEDA MEDIA

Volumen de datos	POSTGRESQL		FIREBIRD			
	ms	Puntaje	ms	Puntaje		
1000	92,4	2	Malo	34,6	10	Excelente
10000	386,8	6	Buena	320,8	10	Excelente
100000	3464,8	10	Excelente	4023,7	8	Muy Buena
1000000	37011,7	10	Excelente	89532,7	2	Malo
10000000	1551908,7	8	Muy Buena	1304074,3	10	Excelente
PROMEDIO	7.2		8			

Fuente: Víctor Colcha

Interpretación de resultados

Como se aprecia en la tabla III.XXXV y observa la Figura III.28 los tiempos de respuesta de los dos SGBD van creciendo de acuerdo al volumen de datos en el que van hacer buscados. Mostrando que Firebird en los escenario individual de 1000, 10000 y

10000000 datos supera aproximadamente con un 46 % de ventaja a PostgreSQL, pero en los 100000 y 1000000 datos PostgreSQL supera aproximadamente con un 50 % de ventajas. Posesionándose Firebird pero con una diferencia corta, nuevamente como el mejor en este grado de complejidad media al brindar menor tiempo de respuesta.

Por lo cual en la tabla III.XXXVI se muestra por cada volumen de datos el puntaje de calificación asignados con su respectiva equivalencia, superando Firebird aproximadamente con 0.8 puntos de diferencia más rápido que PostgreSQL.

Cuadro comparativo de buscar datos de complejidad alta entre los dos SGBD

En la Tabla III.XXXVII se observa la comparación de los dos SGBD con el indicador buscar datos con complejidad alta representados en milisegundos

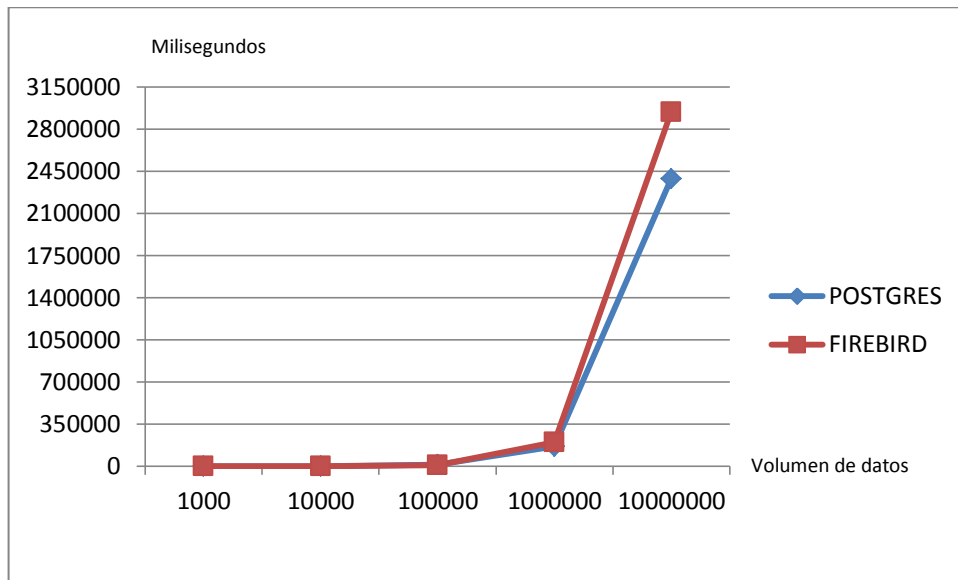
TABLA 3.XXXVII COMPARATIVO BUSCAR DATOS DE COMPLEJIDAD ALTA

Consulta búsqueda de complejidad alta			
Volumen de datos	PostgreSQL (ms)	Firebird (ms)	Salida de datos
10000	164,1	92,2	132
100000	1146,3	971,9	1656
1000000	13986,2	11636,7	19248
10000000	161294,5	198767,3	213984
100000000	2375348	2896933,2	2311872

Fuente: Víctor Colcha

En la Figura III.29 se observa el comportamiento de los datos entre los dos SGBD PostgreSQL y Firebird.

FIGURA 3.29 BÚSQUEDA DE COMPLEJIDAD ALTA DE LOS SGBD



Fuente: Víctor Colcha

Asignación del puntaje

Para la asignación del puntaje se hace referencia a la tabla III.VIII del literal 3.9.4 del Modelo para evaluar los resultados donde se aplica dichas reglas.

TABLA 3.XXXVIII ASIGNACIÓN DEL PUNTAJE BÚSQUEDA ALTA

Volumen de datos	POSTGRESQL			FIREBIRD		
	Ms	Puntaje		ms	Puntaje	
1000	164,1	4	Regular	92,2	10	Excelente
10000	1146,3	8	Muy buena	971,9	10	Excelente
100000	13986,2	8	Muy buena	11636,7	10	Excelente
1000000	161294,5	10	Excelente	198767,3	6	Buena
10000000	2375348	10	Excelente	2896933,2	6	Buena
PROMEDIO		8			8.4	

Fuente: Víctor Colcha

Interpretación de resultados

Como se aprecia en la tabla III.XXXVII y observa la Figura III.29 los tiempos de respuesta de los dos SGBD van creciendo de acuerdo al volumen de datos en el que van hacer buscados. Mostrando que Firebird en los tres primeros escenario individual de

1000, 10000 y 100000 datos supera aproximadamente con un 33.3 % de ventaja a PostgreSQL, pero en los 1000000 y 10000000 datos PostgreSQL supera aproximadamente con un 40 % de ventajas. Posesionándose Firebird pero con una diferencia más corta, nuevamente como el mejor en este grado de complejidad media al brindar menor tiempo de respuesta.

Por lo cual en la tabla III.XXXVIII se muestra por cada volumen de datos el puntaje de calificación asignados con su respectiva equivalencia, superando Firebird aproximadamente con 0.4 puntos de diferencia más rápido que PostgreSQL.

3.11.7 Uso de la Memoria RAM

En la Figura III.30 se muestra los datos estadísticos obtenidos mediante la herramienta SPSS referente a los datos del SGBD PostgreSQL, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “k”** del anexo 3.

FIGURA 3.30 USOS DE LA MEMORIA RAM

SGBD POSTGRESQL: DATOS ESTADISTICOS USO MEMORIA RAM

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	3,4	3,7	3,510	,0314	,0994
10000	10	3,1	5,4	4,890	,2063	,6523
100000	10	18,8	19,7	19,360	,0897	,2836
1000000	10	158,7	164,1	160,050	,4729	1,4954
10000000	10	190,9	205,5	200,250	1,3067	4,1321
N válido (según lista)	10					

Fuente: Víctor Colcha

En la Figura III.31 se muestra los datos estadísticos obtenidos mediante el software SPSS del SGBD Firebird, la existencia de todas las lecturas que se procesaron para este indicador se observa en el **literal “l”** del anexo 3.

FIGURA 3.31 FIREBIRD USO DE MEMORIA RAM**SGBD FIREBIRD: DATOS ESTADISTICOS USO MEMORIA RAM**

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	6,3	8,2	6,830	,1789	,5658
10000	10	16,6	17,9	17,230	,1383	,4373
100000	10	55,1	63,5	60,540	,8455	2,6738
1000000	10	99,8	106,7	103,280	,6863	2,1704
10000000	10	190,2	204,3	197,740	1,4527	4,5938
N válido (según lista)	10					

Fuente: Víctor Colcha

Cuadro comparativo

En la Tabla III.XXXIX se observa la comparación de los dos SGBD con el indicador uso de memoria RAM representados en MB.

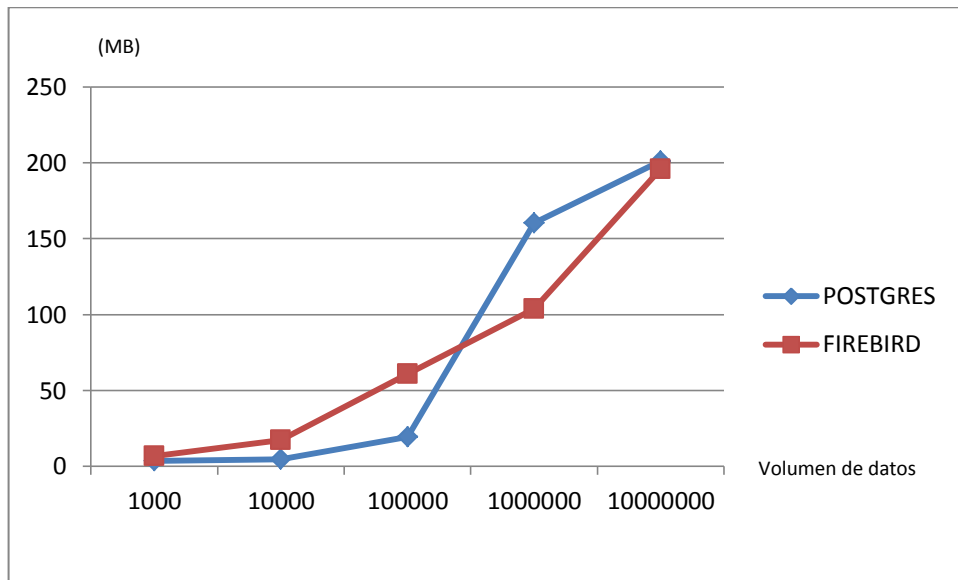
TABLA 3.XXXIX COMPARACIÓN-USO DE LA MEMORIA RAM

Consumo de la memoria RAM		
Volumen de datos	PostgreSQL (MB)	Firebird (MB)
1000	3,510	6,830
10000	4,890	17,230
100000	19,360	60,540
1000000	160,050	103,280
10000000	200,250	197,740

Fuente: Víctor Colcha

En la Figura III.32 se observa el comportamiento de los datos entre los dos SGBD.

FIGURA 3.32 RESULTADOS DEL USO DE MEMORIA RAM



Fuente: Víctor Colcha

Asignación del puntaje

Para la asignación del puntaje se hace referencia a la tabla III.VIII del literal 3.9.4 del Modelo para evaluar los resultados donde se aplica dichas reglas.

TABLA 3.XL ASIGNACIÓN DEL PUNTAJE USO DE LA MEMORIA RAM

Volumen de datos	POSTGRESQL			FIREBIRD		
	MB	Puntaje		MB	Puntaje	
1000	3,510	10	Excelente	6,830	4	Regular
10000	4,890	10	Excelente	17,230	2	Mala
100000	19,360	10	Excelente	60,540	2	Mala
1000000	160,050	6	Buena	103,280	10	Excelente
10000000	200,250	8	Muy buena	197,740	10	Excelente
PROMEDIO		8.8			5.6	

Fuente: Autor

Interpretación de resultados

Como se aprecia en la Tabla III.XXXIX y observa la Figura III.32, el uso de la memoria RAM de los dos SGBD va creciendo de acuerdo al volumen de datos. Mostrando que PostgreSQL en los tres primeros escenario individual de 1000, 10000 y 100000 datos supera aproximadamente con un 73.3% de ventaja a Firebird, pero en los 1000000 y

10000000 datos Firebird supera aproximadamente con un 30% de ventajas. Posesionándose PostgreSQL como el mejor en este indicador al consumir menor memoria RAM y mantener su administración de los datos sin cambio alguno.

Por lo cual en la tabla III.XL se muestra por cada volumen de datos el puntaje de calificación asignados con su respectiva equivalencia, superando PostgreSQL aproximadamente con 3.2 puntos de diferencia más rápido que Firebird.

3.11.8 Uso del Procesador

En la Figura III.33 se muestra los datos procesados mediante el software SPSS de SGBD PostgreSQL, la existencia de todas las lecturas que se procesaron para este indicador se observa en el literal "k" del anexo 3.

FIGURA 3.33 POSTGRESQL - USO DEL PROCESADOR

SGBD POSTGRESQL: DATOS ESTADISTICOS USO PROCESADOR

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	5,1	7,4	6,430	,2357	,7454
10000	10	8,4	10,2	9,910	,1709	,5405
100000	10	8,9	10,9	10,150	,1586	,5017
1000000	10	9,9	10,7	10,500	,0730	,2309
10000000	10	10,0	11,2	10,550	,1186	,3749
N válido (según lista)	10					

Fuente: Autor

En la Figura III.34 se muestra los datos procesados mediante el software SPSS de SGBD Firebird, la existencia de todas las lecturas que se procesaron para este indicador se observa en el literal "l" del anexo 3

FIGURA 3.34 FIREBIRD USO DEL PROCESADOR

SGBD FIREBIRD: DATOS ESTADISTICOS USO PROCESADOR

	N	Mínimo	Máximo	Media		Desv. típ.
	Estadístico	Estadístico	Estadístico	Estadístico	Error típico	Estadístico
1000	10	7,1	10,4	8,710	,3111	,9837
10000	10	9,6	10,6	10,140	,0968	,3062
100000	10	10,0	10,6	10,300	,0715	,2261
1000000	10	9,9	11,1	10,490	,1386	,4383
10000000	10	9,9	11,5	10,570	,1904	,6019
N válido (según lista)	10					

Fuente: Víctor Colcha

Tabla comparativa

En la Tabla III.XLIV se observa la comparación de los dos SGBD con el indicador uso del procesador representado en el porcentaje del proceso

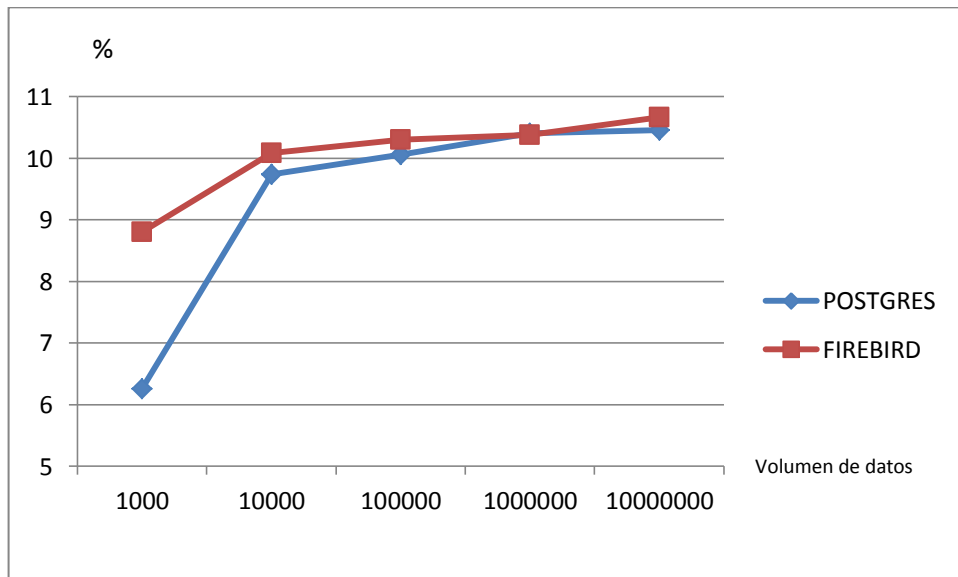
TABLA 3.XLI COMPARACIÓN DEL USO DEL PROCESADOR

Consumo de la procesador		
Volumen de datos	PostgreSQL (%)	Firebird (%)
1000	6,26	8,80
10000	9,74	10,08
100000	10,06	10,30
1000000	10,40	10,38
10000000	10,46	10,66

Fuente: Víctor Colcha

En el Figura III.35 se observa el comportamiento de los datos entre los dos SGBD

FIGURA 3.35 RESULTADOS USO DEL PROCESADOR



Fuente: Autor

Asignación del puntaje

Para la asignación del puntaje se hace referencia a la tabla III.VIII del literal 3.9.4 del Modelo para evaluar los resultados donde se aplica dichas reglas.

TABLA 3.XLII ASIGNACIÓN DEL PUNTAJE - USO DEL PROCESADOR

Volumen de datos	POSTGRESQL			FIREBIRD		
	%	Puntaje		%	Puntaje	
1000	6,430	10	Excelente	8,710	6	Buena
10000	9,910	10	Excelente	10,140	8	Muy Buena
100000	10,150	10	Excelente	10,300	8	Muy Buena
1000000	10,500	8	Muy Buena	10,490	10	Excelente
10000000	10,550	10	Excelente	10,570	8	Muy Buena
PROMEDIO	9.6			8		

Fuente: Víctor Colcha

Interpretación de resultados

Como se aprecia en la Tabla III.XLI y observa la Figura III.35, el uso del procesador de los dos SGBD va creciendo en proporcionalidades muy pequeñas de acuerdo al volumen de datos. Mostrando que Firebird en escenario individual de 1000000 de datos supera aproximadamente con un 20% de ventaja a PostgreSQL. Pero en los de más escenarios PostgreSQL supera aproximadamente con un 25% de ventajas.

Posesionándose PostgreSQL con una diferencia corta como el mejor en este indicador al hacer uso poco uso de procesador y mantener su administración de los datos sin cambio alguno.

Por lo cual en la Tabla III.XLII se muestra por cada volumen de datos el puntaje de calificación asignados con su respectiva equivalencia, superando PostgreSQL con 1.6 puntos de diferencia en menor consumo de este recurso que Firebird.

3.11.9 Prueba Adicional

Esta prueba fue realizada para comprobar la robustez de los SGBD ante situaciones como una mala definición de sentencias SQL. Mediante esta prueba se obtuvieron los siguientes datos como se muestra en la tabla III.XLIII y se describe su equivalencia en tiempo normal.

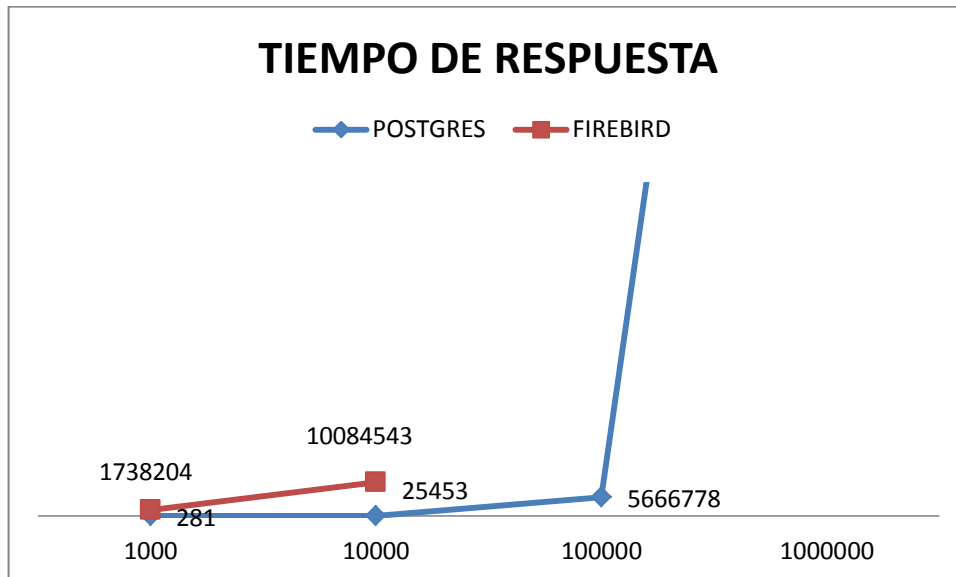
Tabla 3.XLIII PRUEBA ADICIONAL DE SELECCIÓN

Prueba adicional - Consulta de complejidad simple				
Volumen de datos	PostgreSQL		Firebird	
	ms	equivalencia	ms	equivalencia
1000	281	281ms	1738204	21m 58s 204ms
100000	25453	25 s 453 ms	10084543	2h 48m 4s 543ms
1000000	5666778	1.574105 h	Tiempo indeterminado(más 8 horas)	
10000000	42654355	11.848 h	Tiempo indeterminado(más de un día)	
100000000	---prueba no realizada--		---prueba no realizada-----	

Fuente: Víctor Colcha

En la Figura III.36 se visualiza los cambios de tiempos en la recuperación de datos.

FIGURA 3.36 TIEMPO DE RESPUESTA PRUEBA ADICIONAL



Fuente: Víctor Colcha

Interpretación de resultados

Con base al objetivo de esta prueba donde se pretende demostrar si el SGBD soporta sentencias SQL mal definidas se define mediante la visualización de la tabla III.LIII que hace referencia a la Figura III.36 donde se muestra los tiempos de respuesta de acuerdo a los escenarios planteados, el cual PostgreSQL responde con un tiempo inferior a los que presenta Firebird. Cabe mencionar que Firebird presenta volúmenes de tiempos

extremadamente altos referentes al otro gestor, la cual no pudo culminar las pruebas con los demás escenarios ya que el archivo de la base de datos Firebird se dañaba. Por lo tanto Firebird presenta falencias de respuesta mediante un código mal definido.

3.12 Análisis de Resultados

3.12.1 Resultados Desempeño

Una vez que se procesaron los resultados se asignó los puntajes respectivos a sus pruebas. Con la cual se procede al análisis de estos resultados y poder calificarlos

Obteniendo de la Tabla el Promedio de los puntajes de los dos SGBD para cada uno de los indicadores se procedió a aplicar una regla de tres simple para obtener el porcentaje de acuerdo a su ponderación descrita en el modelo de evaluación.

Para la asignación del puntaje se hace referencia a la Tabla III.V del literal 3.9 del modelo para evaluar los resultados donde se aplica dichas reglas.

INSERCIÓN DE DATOS

En la tabla III.XLIV se describe la evaluación para este indicador tomando como referencia la tabla III.XX de donde se obtuvieron los promedios para este indicador.

TABLA 3.XLIV ANÁLISIS DE RESULTADOS INSERCIÓN

SGBD	Puntaje Promedio	Peso Porcentual	Valor cuantitativo	Valor cualitativo	Peso % máximo
PostgreSQL	5.2	7.8%	3	Bueno	15%
Firebird	10	15%	5	Excelente	15%

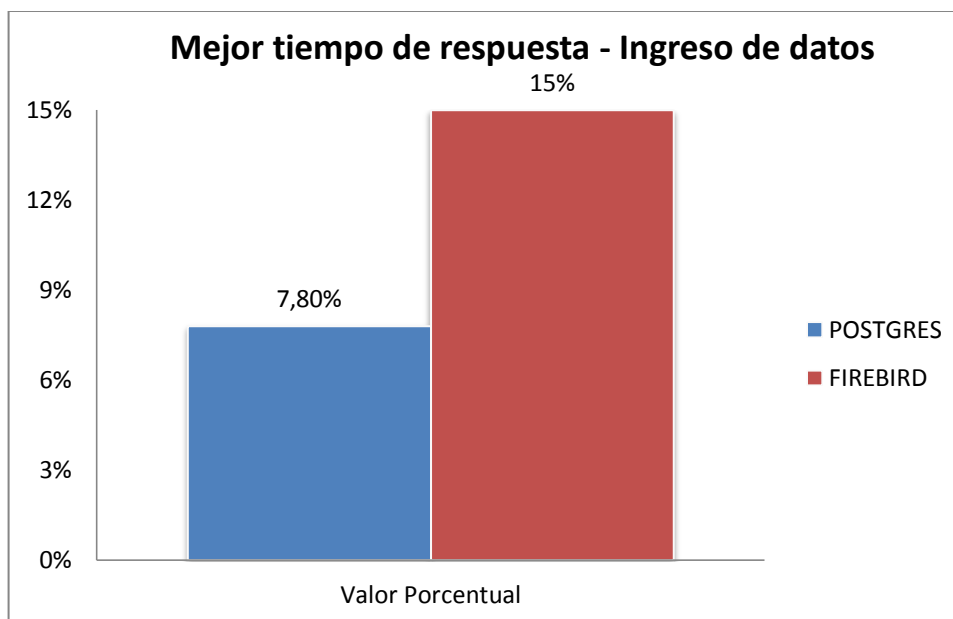
Fuente: Víctor Colcha

Interpretación de resultados

Como se aprecia en la tabla III.XLIV **Firebird** presenta el menor tiempos de respuesta al ingreso de datos en base a los distintos escenarios de prueba, con un puntaje de 10 equivalente a **Excelente** y porcentaje del **15 %** referente a la tabla de calificación para este indicador escritas anteriormente, mientras que **PostgreSQL** obtuvo un puntaje de 5.2 equivalente a **Bueno** y porcentaje de **7.8%** que está por debajo de Firebird.

Por lo cual en la Figura III.37 se observa que Firebird con una diferencia del 7.2% se considera el más rápido en el ingreso de datos entre los dos SGDB expuestos.

FIGURA 3.37 RESULTADO DE ANÁLISIS INSERTAR



Fuente: Autor

ACTUALIZACIÓN DE DATOS

En la Tabla III.XLV se describe la evaluación para este indicador tomando como referencia la Tabla III.XXII de donde se obtuvieron los promedios para este indicador.

TABLA 3.XLV ANÁLISIS DE RESULTADOS ACTUALIZACIÓN

SGBD	Puntaje Promedio	Peso Porcentual	Valor cuantitativo	Valor cualitativo	Peso % máximo
PostgreSQL	5.6	8.4%	3	Bueno	15%
Firebird	10	15%	5	Excelente	15%

Fuente: Autor

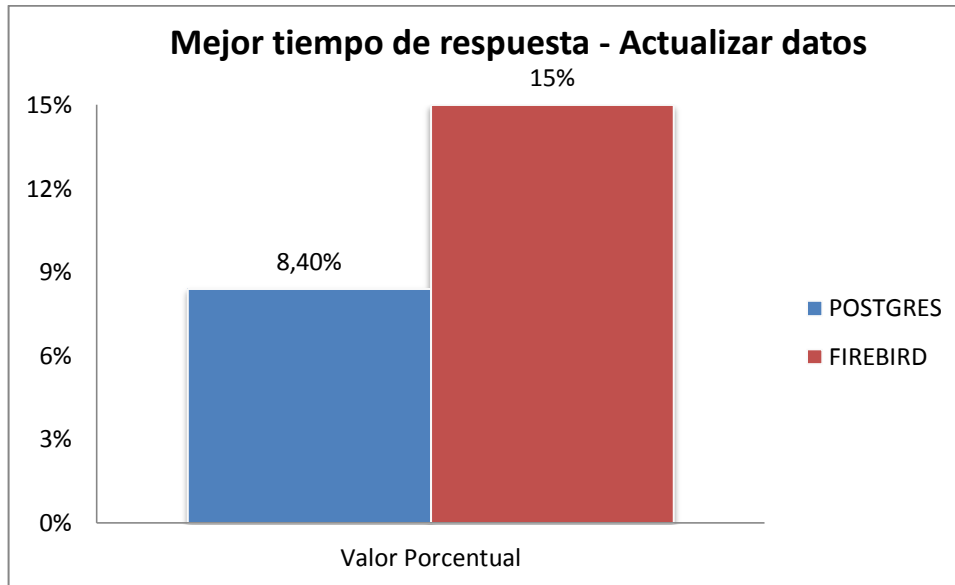
Interpretación de resultados

Como se aprecia en la tabla III.XLV **Firebird** presenta el menor tiempos de respuesta al actualizar los datos en base a los distintos escenarios de prueba, con un puntaje de 10 equivalente a **Excelente** y porcentaje del **15 %** referente a la tabla de calificación para este indicador escritas en sesiones anteriores, mientras que **PostgreSQL** obtuvo un

puntaje de 5.6 equivalente a **Bueno** y porcentaje de **8.4%** que está por debajo de Firebird.

Por lo cual en la Figura III.38 se observa que Firebird con una diferencia del 6.6% se considera el más rápido en la actualización de datos entre los dos SGDB expuestos.

FIGURA 3.38 RESULTADO ACTUALIZAR DATOS



Fuente: Víctor Colcha

CONSULTA: LISTAR DATOS

En la Tabla III.XLVI se describe la evaluación para este indicador tomando como referencia la Tabla III.XLVI, Tabla III.XXVIII y Tabla III.XXX de la prueba listar de diversa complejidad, de donde se obtuvieron los promedios para este indicador.

TABLA 3.XLVI PUNTAJE LISTAR DATOS

	POSTGRESQL	FIREBIRD
Complejidad simple	10	3.6
Complejidad mediana	9.2	4.4
Complejidad alta	8.4	5.2
PROMEDIO	9.2	4.4

Fuente: Víctor Colcha

Obteniendo los promedios de la Tabla III.XLVI se creó los resultados del indicador listar datos donde se observa en la Tabla III.XLVII

TABLA 3.XLVII RESULTADO LISTAR DATOS

SGBD	Puntaje Promedio	Peso Porcentual	Valor cuantitativo	Valor cualitativo	Peso % máximo
POSTGRESQL	9.2	13.8%	5	Excelente	15%
FIREBIRD	4.4	6.6%	3	Bueno	15%

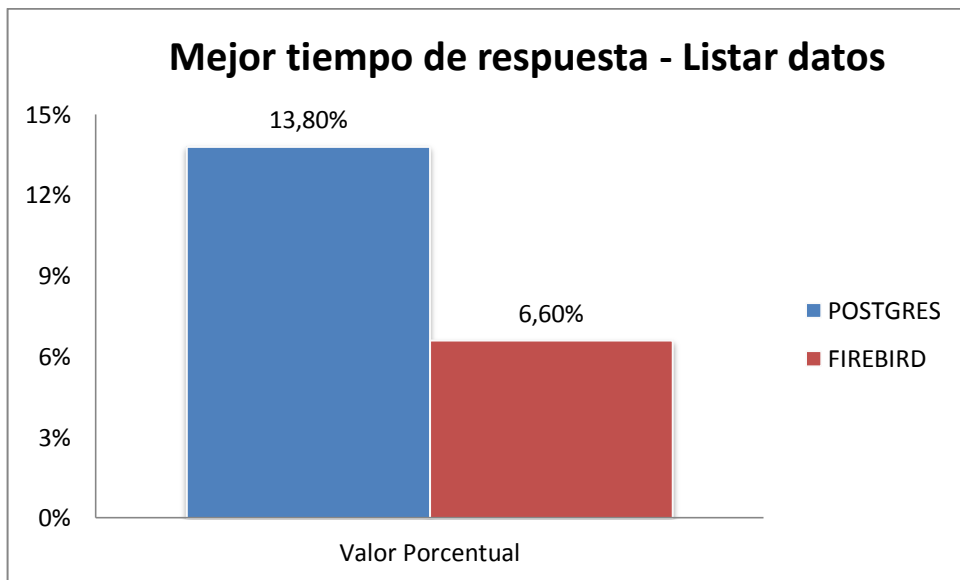
Fuente: Víctor Colcha

Interpretación de datos

Como se aprecia en la tabla III.XLVII **PostgreSQL** presenta el menor tiempos de respuesta al listar los datos en base a los distintos escenarios de prueba y aplicando diferentes grados de complejidad a sus sentencias SQL, con un puntaje de 9.2 equivalente a **Excelente** y porcentaje del **13.8 %** referente a la tabla de calificación para este indicador escritas en sesiones anteriores, mientras que **Firebird** obtuvo un puntaje de 4.4 equivalente a **Bueno** y porcentaje de **6.6%** que está por debajo de PostgreSQL.

Por lo cual en la Figura III.39 se observa que PostgreSQL con una diferencia del 7.2% se considera el más rápido en listar datos entre los dos SGDB expuestos.

FIGURA 3.39 RESULTADO LISTAR DATOS



Fuente: Víctor Colcha

CONSULTA: BÚSQUEDA DATOS

En la tabla III.XLVIII se describe la evaluación para este indicador tomando como referencia a la tabla III.XXXIV, Tabla III.XXXVI y Tabla III.XXXVIII de la prueba listar de diversa complejidad, de donde se obtuvieron los promedios para este indicador.

TABLA 3.XLVIII RESULTADO BUSCAR DATOS

	POSTGRESQL	FIREBIRD
Complejidad simple	6.4	8.8
Complejidad mediana	7.2	8
Complejidad alta	8	8.4
PROMEDIO	7.2	8.4

Fuente: Víctor Colcha

Obteniendo los promedios de la Tabla III.XLVIII se creó los resultados del indicador listar datos donde se observa en la Tabla III.XLIX

TABLA 3.XLIX PUNTAJE BUSCAR DATOS

SGBD	Puntaje Promedio	Peso Porcentual	Valor cuantitativo	Valor cualitativo	Peso % máximo
POSTGRESQL	7.2	10.8%	4	Muy bueno	15%
FIREBIRD	8.4	12.6%	5	Excelente	15%

Fuente: Víctor Colcha

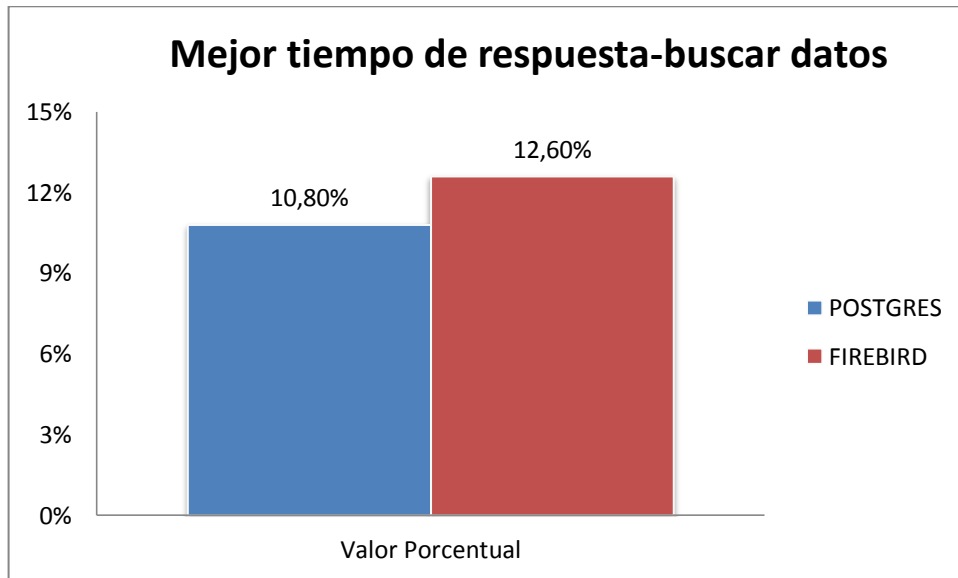
Interpretación de datos

Como se aprecia en la tabla III.XLIX **Firebird** presenta el menor tiempos de respuesta al buscar y mostrar los datos, en base a los distintos escenarios de prueba y aplicando diferentes grados de complejidad a sus sentencias SQL, con un puntaje de 8.4 equivalente a **Excelente** y porcentaje del **12.6 %** referente a la tabla de calificación para este indicador escritas en sesiones anteriores, mientras que **PostgreSQL** obtuvo un

puntaje de 7.2 equivalente a **Muy Bueno** y porcentaje de **10.8%** que se aproxima a los valores de Firebird.

Por lo cual en la Figura III.40 se observa que Firebird con una diferencia del 1.8% se considera el más rápido en buscar y mostrar los datos entre los dos SGDB expuestos.

FIGURA 3.40 TIEMPO DE RESPUESTA PARA BUSCAR DATOS



Fuente: Autor

3.12.2 Resultados Consumo de Recursos

Memoria RAM

En la tabla III.L se describe la evaluación para este indicador tomando como referencia la Tabla III.XL de donde se obtuvieron los promedios para este indicador.

TABLA 3.L USO DE MEMORIA RAM DE LOS SGDB

SGDB	Puntaje Promedio	Peso Porcentual	Valor cuantitativo	Valor cualitativo	Peso % máximo
POSTGRESQL	8.8	17.6%	8	Excelente	20%
FIREBIRD	5.6	11.2%	4	Buen0	20%

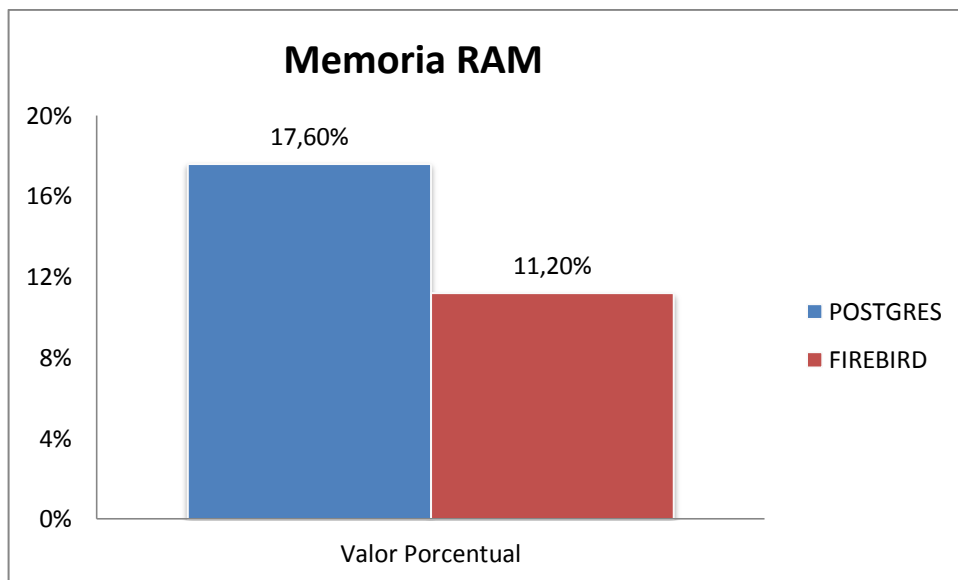
Fuente: Victor Colcha

Interpretación de datos

Como se aprecia en la tabla III.L **PostgreSQL** presenta el menor uso de la memoria RAM en base a los distintos escenarios de prueba y aplicando sentencias SQL de complejidad alta, con un puntaje de 8.8 equivalente a **Excelente** y un porcentaje del **17.6 %** referente a la tabla de calificación para este indicador escritas en sesiones anteriores, mientras que **Firebird** obtuvo un puntaje de 5.6 equivalente a **Bueno** y un porcentaje de **11.26%** que está por debajo de PostgreSQL.

Por lo cual en la Figura III.41 se observa que PostgreSQL con una diferencia del 6.4% se considera el que menos consume la memoria RAM entre los dos SGDB expuestos.

FIGURA 3.41 MEMORIA RAM DE LOS SGDB



Fuente: Víctor Colcha

USO DEL PROCESADOR

En la tabla III.LI se describe la evaluación para este indicador tomando como referencia la Tabla III.XLII de donde se obtuvieron los promedios para este indicador.

TABLA 3.LI USO DEL PROCESADOR

SGDB	Puntaje Promedio	Peso Porcentual	Valor cuantitativo	Valor cualitativo	Peso % máximo
POSTGRESQL	9.6	19.2%	8	Excelente	20%
FIREBIRD	8	16%	6	Muy buena	20%

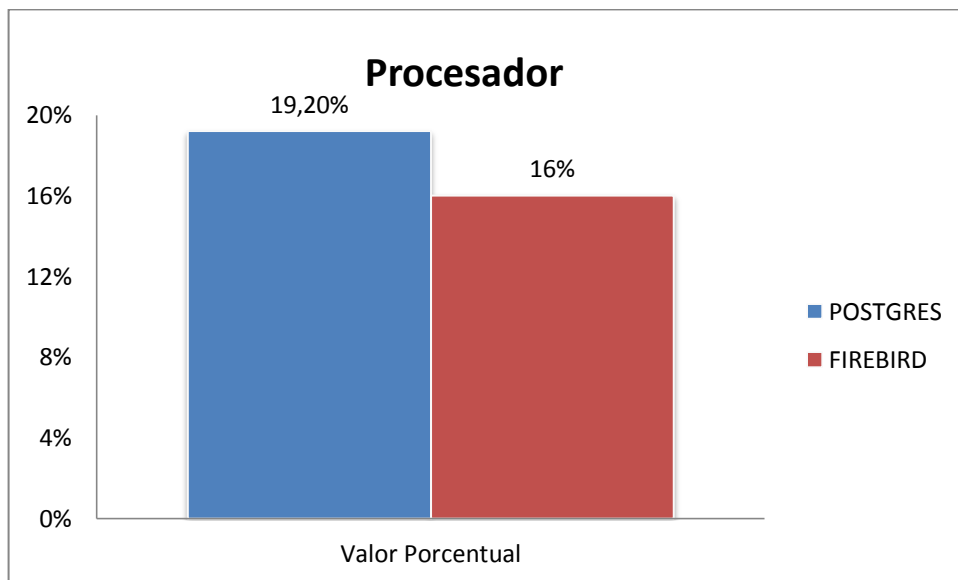
Fuente: Víctor Colcha

Interpretación de datos

Como se aprecia en la tabla III.LI **PostgreSQL** presenta el menor uso del procesador en base a los distintos escenarios de prueba y aplicando sentencias SQL de complejidad alta, con un puntaje de 9.6 equivalente a **Excelente** y porcentaje del **19.2 %** referente a la tabla de calificación para este indicador escritas en sesiones anteriores, mientras que **Firebird** obtuvo un puntaje de 8 equivalente a **Muy Bueno** y porcentaje de **16%** que se aproxima un poco a los valores de PostgreSQL.

Por lo cual en la Figura III.36 se observa que PostgreSQL con una diferencia del 3.2% se considera el que menos consume el procesador entre los dos SGDB expuestos.

FIGURA 3.42 PROCESADOR



Fuente: Víctor Colcha

3.13 Resultados y Evaluación General

Aplicando la fórmula 3 del método de evaluación se procederá a evaluar todos los indicadores en general de los dos SGBD.

Calificación porcentual máxima

Indicador1 (15 %) + Indicador2 (15 %)+ Indicador3 (15 %)+ Indicador4 (15 %) + Indicador5 (20 %)+ Indicador6 (20 %)=100%

Calificación PostgreSQL

7.8% +8.4% + 13.8% + 10.8 % + 17.6% +19.2% = 77,6%

Calificación Firebird

15% + 15% + 6.6% + 12.6% +11.2%+ 16% = 76.4%

En la Tabla III.LII muestra en forma resumida los datos generales por cada indicador de cada parámetro donde se puede apreciar los valores finales para el análisis del rendimiento de los dos SGBD expuesto.

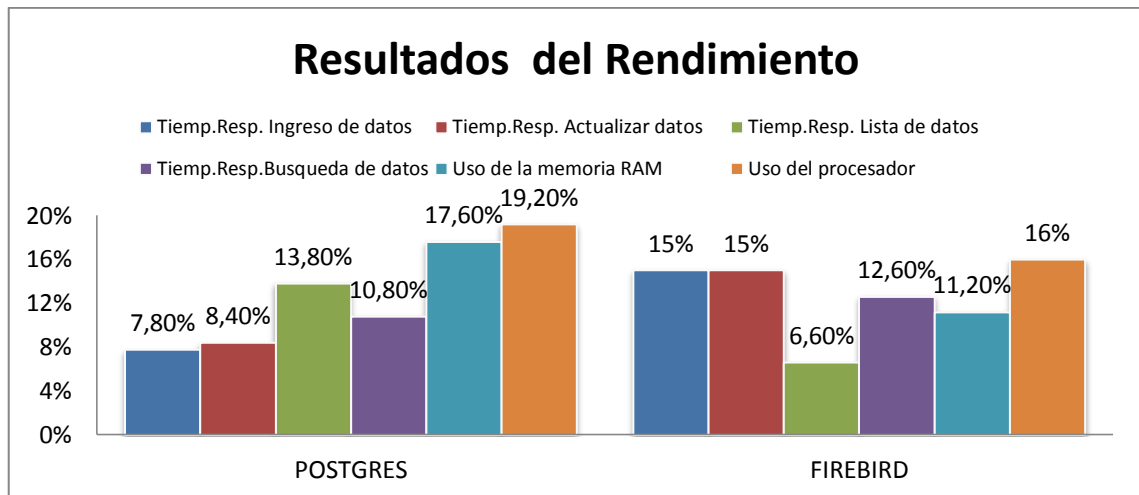
TABLA 3.LII CALIFICACIÓN DE LOS PARÁMETROS E INDICADORES

Rendimiento		Calificación			Porcentaje		
Parámetro	Indicador	PostgreSQL	Firebird	Max	PostgreSQL	Firebird	Max
Desempeño	Ingreso	3	5	5	7.8%	15%	15 %
	Actualizar	3	5	5	8.4%	15%	15 %
	Listar datos	5	3	5	13.8%	6.6%	15 %
	Buscar datos	4	5	5	10.8%	12.6%	15 %
Consumo de recursos	Uso RAM	8	4	8	17.6%	11.2%	20 %
	Uso Procesador	8	6	8	19.2%	16%	20 %
Totales		31	28	36	77.6%	76.4%	100%

Fuente: Víctor Colcha

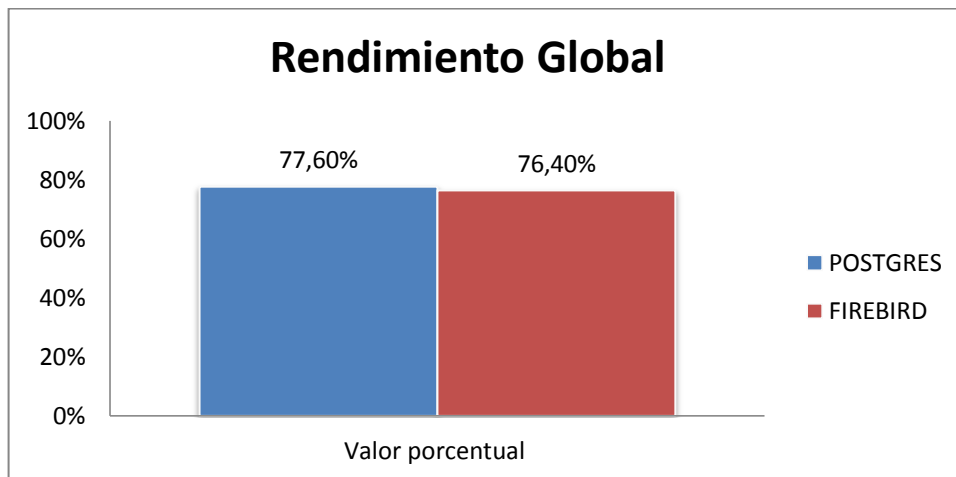
Gráfica estadística

FIGURA 3.43 RESULTADOS PARCIALES DEL RENDIMIENTO



Fuente: Víctor Colcha

FIGURA 3.44 RENDIMIENTO GLOBAL



Fuente: Víctor Colcha

Interpretación de resultados

Por medio del análisis realizado con base a los parámetros e indicadores se ha obtenido como resultado final que **PostgreSQL** con un valor porcentual del **77,60 %** y **Firebird** con su respectivo valor porcentual del **76,40%**. Se considera a ambos SGBD con una equivalencia de **Muy buena** donde cumplen la mayor parte de las expectativas, de acuerdo a la tabla general de evaluación descrita en el modelo de evaluación. Concluyendo así que PostgreSQL 9.2 y Firebird 2.5.2 con una **diferencia** del **1.2 %** y de acuerdo a la tabla III.VII que los dos **SGBD** tienen una calificación de **MUY BUENA** al poseer en forma general casi el mismo rendimiento.

3.14 Comprobación de Hipótesis

En infinidad de ocasiones nos encontraremos con una serie de datos u observaciones que hemos obtenido al analizar una variable aleatoria de patrón desconocido. Esto ocurrirá, por ejemplo, al registrar los tiempos transcurridos entre llamadas sucesivas a un call-center, al registrar los tiempos de fallo de un determinado dispositivo, al contabilizar el número de visitas en una de páginas web, etc

En tales casos, resulta fundamental intentar identificar un patrón conocido (distribución de probabilidad) que nos ayude a explicar el comportamiento de la variable aleatoria.

La distribución Chi Cuadrada (χ^2 cuadrado o X^2) [35] es una de las distribuciones más empleadas en todos los campos como en medicina uno de los ambientes más exactos en el análisis de datos. Su uso más común es cuando se quiere probar si unas mediciones que se hayan efectuado siguen una distribución esperada, por ejemplo la normal o cualquier otra.

Por lo cual luego de haber analizado e interpretado los datos, para la demostración de la hipótesis se ha utilizado la distribución estadística Chi-cuadrado, la cual nos permitió definir la aceptación o rechazo de la hipótesis planteada en esta investigación.

En base a la prueba Chi-cuadrado se plantea una hipótesis nula y la hipótesis investigativa o alternativa.

H₀ = El DBMS PostgreSQL 9.2 es el que no posee mejor rendimiento que el DBMS Firebird 2.5 para la administración de información del Gobierno Autónomo Descentralizado Municipal de San José de Chimbo.

H_i= El DBMS PostgreSQL 9.2 es el que posee mejor rendimiento que el DBMS Firebird 2.5 para la administración de información del Gobierno Autónomo Descentralizado Municipal de San José de Chimbo.

3.14.1 Análisis de los Datos

Para realizar el cálculo del Chi-cuadrado se tomará los valores en base a la calificación porcentual de los resultados finales de cada indicador, detallados en la Tabla III.LII.

Para igualar a un nivel único del 100% a todos los valores porcentuales de cada indicador se aplicó una regla de tres simple, tal proceso se visualiza en la Tabla III.LII. Cabe mencionar que no alterara el significado de calificación, ya que se mantiene el balance obtenido del análisis.

Tabla valores observados

TABLA 3.LIII DATOS OBSERVADOS

Rendimiento		Calificación		
Parámetro	Indicador	PostgreSQL	Firebird	Max
Desempeño	Insertar	52	100	152
	Actualizar	56	100	156
	Lista	92	44	136
	Búsqueda	72	84	156
Consumo	RAM	88	56	144
	Procesador	96	80	176
Totales		456	464	920

Fuente: Víctor Colcha

3.14.2 Cálculo del Chi-Cuadrado (X^2)

A través de Chi-cuadrado [24] se procedió a probar de forma afirmativa o negativa que la distribución de las frecuencias observadas difiere significativamente en relación a la distribución de las frecuencias que deberíamos esperar, en la Tabla III.LIV se visualiza dicho proceso.

$$X^2 = \frac{(f_o - f_e)^2}{f_e}$$

Dónde:

f_o= frecuencia obtenida.

f_e= frecuencia esperada.

f_e= (Total fila*Total columna) /Total general.

(f_o-f_e)² / f_e= Resultado de frecuencia observada menos la frecuencia esperada al cuadrado y todo eso dividido a la frecuencia esperada.

TABLA 3.LIV CALCULO CHI-CUADRADO

Celda	Fo	Fe	(fo-fe)	(fo-fe) ²	(fo-fe) ² / fe
1	52	75,3391304	-23,3391304	544,715009	7,23017383
2	100	76,6608696	23,3391304	544,715009	7,10551566
3	56	77,3217391	-21,3217391	454,61656	5,8795439
4	100	78,6782609	21,3217391	454,61656	5,77817245
5	92	67,4086957	24,5913043	604,73225	8,97113115
6	44	68,5913043	-24,5913043	604,73225	8,81645648
7	72	77,3217391	-5,32173913	28,3209074	0,36627354
8	84	78,6782609	5,32173913	28,3209074	0,35995848
9	88	71,373913	16,626087	276,426767	3,87293838
10	56	72,626087	-16,626087	276,426767	3,80616358
11	96	87,2347826	8,76521739	76,8290359	0,88071562
12	80	88,7652174	-8,76521739	76,8290359	0,86553087
Total (X2)					53,932574

Fuente: Víctor Colcha

3.14.3 Nivel de Significancia y Reglas de Decisión.**Valores críticos de Chi-cuadrado**

La tabla Chi-cuadrado es usada para realizar pruebas de independencia, que nos permite determinar si existe una relación entre dos variables categóricas. La tabla tiene dos entradas:

1. Grado de libertad.

Es un estimador del número de categorías independientes en la prueba de independencia o experimento estadístico. Se encuentran mediante la fórmula siguiente

Grados de libertad = (número filas -1* número de columnas -1)

Grados de libertad = (6-1)*(2-1) = 5

2. Grado de significancia

Nivel de significación (P): Denominado nivel de confianza, se refiere a la probabilidad de que los resultados observados se deban al azar. Este valor es fijado por el investigador, usualmente es el 5% o 10%. Alfa (α) este valor hace referencia al nivel de confianza que deseamos que tengan los cálculos de la prueba; es decir, si queremos tener un nivel de confianza del 95%, el valor de alfa debe ser del 0.05, lo cual corresponde al complemento porcentual de la confianza.

Lo que indica que si se toma $P=0.05$, se está significando que solo en un 5% de las veces en que se realice la medición, el resultado obtenido podría deberse al azar. De lo contrario sería decir que existe un nivel de confianza del 95% que el resultado es real y no debido a la casualidad.

Nivel de confiabilidad = 95% y el grado de significancia será 0.05

En la Figura 3.45 se observa el valor crítico de Chi-cuadrado (tabla)

FIGURA 3.45 TABLA CHI- CUADRADO

TABLA 3-Distribución Chi Cuadrado χ^2

P = Probabilidad de encontrar un valor mayor o igual que el chi cuadrado tabulado, v = Grados de Libertad

v/p	0,001	0,0025	0,005	0,01	0,025	0,05	0,1	0,15	0,2	0,25	0,3	0,35	0,4	0,45	0,5
1	10,8274	9,1404	7,8794	6,6349	5,0239	3,8415	2,7055	2,0722	1,6424	1,3233	1,0742	0,8735	0,7083	0,5707	0,4549
2	13,8150	11,9827	10,5965	9,2104	7,3778	5,9915	4,6052	3,7942	3,2189	2,7726	2,4079	2,0996	1,8326	1,5970	1,3863
3	16,2660	14,3202	12,8381	11,3449	9,3484	7,8147	6,2514	5,3170	4,6416	4,1083	3,6649	3,2831	2,9462	2,6430	2,3660
4	18,4662	16,4238	14,8602	13,2767	11,1433	9,4877	7,7794	6,7449	5,9886	5,3853	4,8784	4,4377	4,0446	3,6871	3,3567
5	20,5147	18,3854	16,7496	15,0863	12,8325	11,0705	9,2363	8,1152	7,2893	6,6257	6,0644	5,5731	5,1319	4,7278	4,3515
6	22,4575	20,2491	18,5475	16,8119	14,4494	12,5916	10,6446	9,4461	8,5581	7,8408	7,2311	6,6948	6,2108	5,7652	5,3481
7	24,3213	22,0402	20,2777	18,4753	16,0128	14,0671	12,0170	10,7479	9,8032	9,0371	8,3834	7,8061	7,2832	6,8000	6,3458
8	26,1239	23,7742	21,9549	20,0902	17,5345	15,5073	13,3616	12,0271	11,0301	10,2189	9,5245	8,9094	8,3505	7,8325	7,3441
9	27,8767	25,4625	23,5893	21,6660	19,0228	16,9190	14,6837	13,2880	12,2421	11,3887	10,6564	10,0060	9,4136	8,8632	8,3428

Fuente: http://labrad.fisica.edu.uy/docs/tabla_chi_cuadrado.pdf

3.14.4 Criterio de decisión:

Para afirmar o negar la hipótesis planteada debemos comparar el valor obtenido de X_2 calculado= 53,932574 con el Chi-cuadrado de la tabla de valores críticos, donde es X_2 crítico= 11,0705. Por consiguiente se observa que X_2 calculado es mayor a X_2 crítico.

X_2 calculado =53,932574 > X_2 crítico =11,0705

Ahora se puede afirmar que H_0 se rechaza y se acepta la H_1 debido a que el valor obtenido de tabla de Chi-cuadrado crítico es menor al calculado. Por lo cual El SGBD PostgreSQL 9.2 es el que posee mejor rendimiento que el SGBD Firebird 2.5 para la administración de información del Gobierno Autónomo Descentralizado Municipal de San José de Chimbo.

CAPITULO IV

IMPLEMENTACIÓN DEL SISTEMA SYSGAR

En este capítulo para el desarrollo del Sistema de Garantías para el GAD del Cantón Chimbo se implementó la metodología de desarrollo de software Programación Extrema [20] (en adelante XP) es una metodología de desarrollo ligera (o ágil) basada en una serie de valores y de prácticas que persigue el objetivo de aumentar la productividad a la hora de desarrollar sistemas.

4.1 Metodología de Desarrollo de Software XP

Según **Kent Beck** quien es considerado como el padre de la metodología XP, menciona que es un proceso ligero, de bajo riesgo, flexible, predecible, científico y divertido de desarrollar software. [36]

En XP se realiza el software que el cliente solicita y necesita, en el momento que lo precisa, alentando a los programadores a responder a los requerimientos cambiantes que plantea el cliente en cualquier momento. Esto es posible porque está diseñado para adaptarse en forma inmediata a los cambios, con bajos costos asociados, en cualquier etapa del ciclo de vida., es decir XP está diseñada para el cambio.

Esta metodología cuenta con cuatro fases, que son: Planificación, Diseño, Codificación y Pruebas, las cuales fueron implementados durante el desarrollo de este capítulo.

4.2 Planificación del Proyecto

El propósito de esta fase es el de llegar a un acuerdo entre los clientes y los programadores para empezar una recopilación de todos los requerimientos del sistema.

En este punto se tendrá que identificar el número de interacciones, historia de usuario, planes de entrega para cumplir el propósito establecido. En otras palabras asumiendo que en este proceso de planificación habrá errores ante la cual esta metodología establece mecanismos de revisión, donde cada tres o cinco iteraciones es normal revisar las historias de los usuarios y renegociar la planificación.

4.2.1 Descripción del Sistema

El Sistema de Garantías también llamado “SYSGAR”, va a permite al usuario(s) ingresar, actualizar o eliminar información sobre las pólizas de seguros que mantiene el Municipio de San José de Chimbo, entre los beneficios principales que brinda el sistemas es la notificación o alerta de las pólizas por vencer a la fecha establecida y poder renovarse la misma, buscando obtener así información actualizada que sirva a la entidad en el cumplimiento de sus objetivos y metas.

Este sistema fue desarrollado con herramientas estables de última tecnología y software libre brindando de esta manera a los usuarios una aplicación confiable, disponible, segura y con interfaz fácil, y eficiente.

4.2.2 Especificación de Requerimientos

En este punto se define los requerimientos de comportamiento funcional y no funcional con el fin de comprender el alcance del sistema de garantías, los mismos que fueron discutidos y acordados en varias reuniones con las autoridades del GAD del Cantón Chimbo.

Requerimientos funcionales

Para poder identificarlos requerimientos se hace uso del prefijo “Req” más un valor numérico, los mismos se detallan en la tabla IV.I

TABLA 4.I REQUERIMIENTOS DEL SISTEMA

ID	Requerimientos
Req1	El sistema permitirá la autenticación de los usuarios.
Req2	El sistema permitirá almacenar, editar y eliminar los datos del usuario
Req3	El sistema permitirá almacenar, editar y eliminar los datos del contratista
Req4	El sistema permitirá almacenar, editar y suspender los datos del contrato
Req5	El sistema permitirá almacenar, editar y eliminar los datos de la compañía de seguros.
Req6	El sistema permitirá almacenar la información de la garantía de la obra en base al artículo de garantías
Req7	El sistema permitirá almacenar la información de las renovaciones
Req8	El sistema permitirá alertar de caducidad de las garantías.
Req9	El sistema permitirá visualizar las pólizas vencidas y por vencer en base al tiempo que dispone en el artículo sobre garantías.
Req10	El sistema permitirá enviar la notificación del vencimiento de la garantía mediante un correo electrónico al contratista, en base al artículo de garantías
Req11	El sistema permitirá estructurar automáticamente los datos de la nueva renovación de la garantía para su posterior envío.
Req12	El sistema permitirá enviar la notificación y datos adjuntos de una nueva renovación de la garantía mediante un correo electrónico a la compañía de seguros

Continuará: ...

Continua:

Req13	El sistema permitirá listar los datos de los contratistas, usuarios y compañía mediante su identificador, respectivamente en cada interfaz
Req14	El sistema permitirá consultar la información de obras y contratos que maneja la institución.
Req15	El sistema permitirá consultar la información de las garantías de la obras

Fuente: Víctor Colcha

Requerimientos no funcionales

TABLA 4.II REQUERIMIENTO NO FUNCIONALES

No funcionales	Descripción
Amigabilidad	Este sistema es intuitivo agradable y fácil de navegar, con esto el usuario final
Escalabilidad	El sistema poseerá una arquitectura distribuida en n-capas.
Disponibilidad	La disponibilidad del sistema será continua con un nivel de servicio para los usuarios de 24 horas al día, ya sea para generar solicitudes, realizar modificaciones o generar reportes, evitando así pérdida de información y procesos rápidos.
Fiabilidad	Este sistema web tiene una probabilidad de operaciones libres de fallos, con la finalidad de que al final el usuario este deliberando la calidad que garantiza al sitio.
Seguridad	Es muy necesario para definir perfiles de usuario y el acceso a los datos consistente del usuario por medio de una autenticación de inicio.
Mantenibilidad	Mantenimiento anual de la base de datos.
Rendimiento	El tiempo que se demora una petición es aceptable conforme a la expectativa del usuario.

Fuente: Víctor Colcha

4.2.3 Usuarios del Sistema

Es la persona que interactúa directamente con el sistema SYSGAR y puede manipular la información de acuerdo a su necesidad

Roles

Existen diferentes roles (actores) y responsabilidades para las distintas tareas y propósitos durante el proceso, a continuación se describe los más importantes:

Administrador: Gestiona todo el sistema SYSGAR (secretario GAD del cantón Chimbo).

Contratista: Visualiza el estado de la información de su contrato de la obra y Garantía

Usuario Dependiente: usuario asignado por el administrador por defecto le permite visualizar la información de los contratos de obras y sus garantías

4.2.4 Historias de Usuario

Permite que clientes (usuarios) describan brevemente las características que el sistema debe poseer, es decir todos los requerimientos funcionales. Está formado de las siguientes componentes:

Número.-Permite indicar el identificador de la historia de usuario.

Nombre de la historia.- Se especifica el nombre de la historia de usuario.

Usuario.- Se especifica nombre del usuario al que pertenece la historia de usuario.

Iteración.- Permite indicar la interacción en la que se va a entregar.

Prioridad de negocio.- Se especifica el nivel de importancia para el usuario.

Programador responsable.- Se especifica el nombre del Programador encargado de historia de usuario.

Riesgo de desarrollo.- Se especifica el nivel de riesgo al momento de desarrollar la historia de usuario.

En La tabla IV.III se visualiza la historia de usuario para el requerimiento “**Req7**”, las demás historias de usuario se encuentran en el anexo 5

TABLA 4.III HISTORIA DE USUARIO RENOVACIÓN DE GARANTÍA

HISTORIA DE USUARIO	
Numero: 7	Usuario: Administrador (secretario GAD)
Nombre de la Historia: Almacena la información de las renovaciones.	Iteración Asignada: 3
Prioridad en Negocio: Alto	Riesgo en desarrollo: Alto
Programador Responsable: Víctor Colcha	
Descripción: El secretario de GAD de Cantón Chimbo almacena la información de las renovaciones que se realiza por cada contrato de las obras en desarrollo.	
Observaciones: Requerimiento primordial para las notificaciones.	

Fuente: Víctor Colcha

4.2.5 Plan de Interacciones

Por medio del plan de interacción se va a medir el tiempo de avance del sistema, es decir la cantidad de trabajo que el equipo de desarrollo realiza en ese tiempo. Para este proyecto la medida del tiempo que se va implementar es por día. A continuación se detallan las interacciones implementadas.

Interacción 1

En esta primera interacción se especifica la autenticación de los usuarios y la gestión del usuario, en la cual se detalla mediante la Tabla IV.IV el número con la historia de usuario y el tiempo que tarda en días el desarrollo de esa tarea.

TABLA 4.IV HISTORIA DE USUARIO - INTERACCIÓN 1

No.	Historia de Usuario	Tiempo(días)
1	Autenticación de los usuarios.	5
2	Almacena, edita y elimina los datos del usuario	7
Total días		12

Fuente: Víctor Colcha

Interacción 2

En la segunda interacción se especifica la gestión del contratista y del contrato de la obra, en la cual se detalla mediante la Tabla IV.V el número con la historia de usuario y el tiempo que tarda en días el desarrollo de esa tarea.

TABLA 4.V HISTORIA DE USUARIO - INTERACCIÓN 2

No.	Historia de Usuario	Tiempo(días)
3	Almacena, edita y elimina los datos del contratista.	6
4	Almacena, edita y suspender los datos del contrato	8
Total días		14

Fuente: Víctor Colcha

Interacción 3

En la tercera interacción se especifica el almacenamiento de la información de la garantía de la obra y las renovaciones respectivas, en la cual se detalla mediante la Tabla IV.VI el número con la historia de usuario y el tiempo que tarda en días el desarrollo de esa tarea.

TABLA 4.VI HISTORIA DE USUARIO - INTERACCIÓN 3

No.	Historia de Usuario	Tiempo(días)
6	Almacenar la información de la garantía de la obra	7
7	Almacena la información de las renovaciones.	8
Total días		15

Fuente: Víctor Colcha

Interacción 4

En la cuarta interacción se especifica la visualización de datos con la alerta de las garantías vencidas y por vencer, y el formato con la información para la nueva renovación, en la cual se detalla mediante la tabla IV.VII el número con la historia de usuario y el tiempo que tarda en días el desarrollo de esa tarea.

TABLA 4.VII HISTORIA DE USUARIO - INTERACCIÓN 4

No.	Historia de Usuario	Tiempo(días)
8	Alerta de caducidad de las garantías.	2
9	Visualiza las pólizas vencidas y por vencer	6
11	Estructura automática de los datos de la nueva renovación de la garantía para su posterior envío.	7
Total días		15

Fuente: Víctor Colcha

Interacción 5

En la quinta interacción se especifica el envío de la notificación por correo electrónico al contratista y a la compañía de seguros, con el archivo adjunto de datos para la renovación, en la cual se detalla mediante la tabla VIII el número con la historia de usuario y el tiempo que tarda en días el desarrollo de esa tarea.

TABLA 4.VIII HISTORIA DE USUARIO - INTERACCIÓN 5

No.	Historia de Usuario	Tiempo(días)
10	Envío de la notificación del vencimiento de la garantía mediante un correo electrónico al contratista.	6
12	Envío de la notificación y datos adjuntos de una nueva renovación de la garantía mediante un correo electrónico a la compañía de seguros.	10
Total días		16

Fuente: Víctor Colcha

Interacción 6

En la sexta interacción se especifica las consultas sobre los contratos de las obras y las respectivas garantías, en la cual se detalla mediante la Tabla IV.IX el número con la historia de usuario y el tiempo que tarda en días el desarrollo de esa tarea.

TABLA 4.IX HISTORIA DE USUARIO - INTERACCIÓN 6

No.	Historia de Usuario	Tiempo(días)
14	Consulta sobre la obra y contrato que maneja la institución.	6
15	Consulta sobre la garantía del contrato de la obra.	6
Total días		12

Fuente: Víctor Colcha

Interacción 7

En la séptima interacción se especifica la gestión de la compañía de seguros y la lista de datos de la misma más la de los contratistas y usuarios, en la cual se detalla mediante la tabla IX el número con la historia de usuario y el tiempo que tarda en días el desarrollo de esa tarea.

TABLA 4.X HISTORIA DE USUARIO - INTERACCIÓN 7

No.	Historia de Usuario	Tiempo(días)
5	Almacena, edita y elimina los datos de la compañía de seguros.	6
13	Lista de los datos de los contratistas, usuarios y compañía mediante su identificador.	8
Total días		14

Fuente: Víctor Colcha

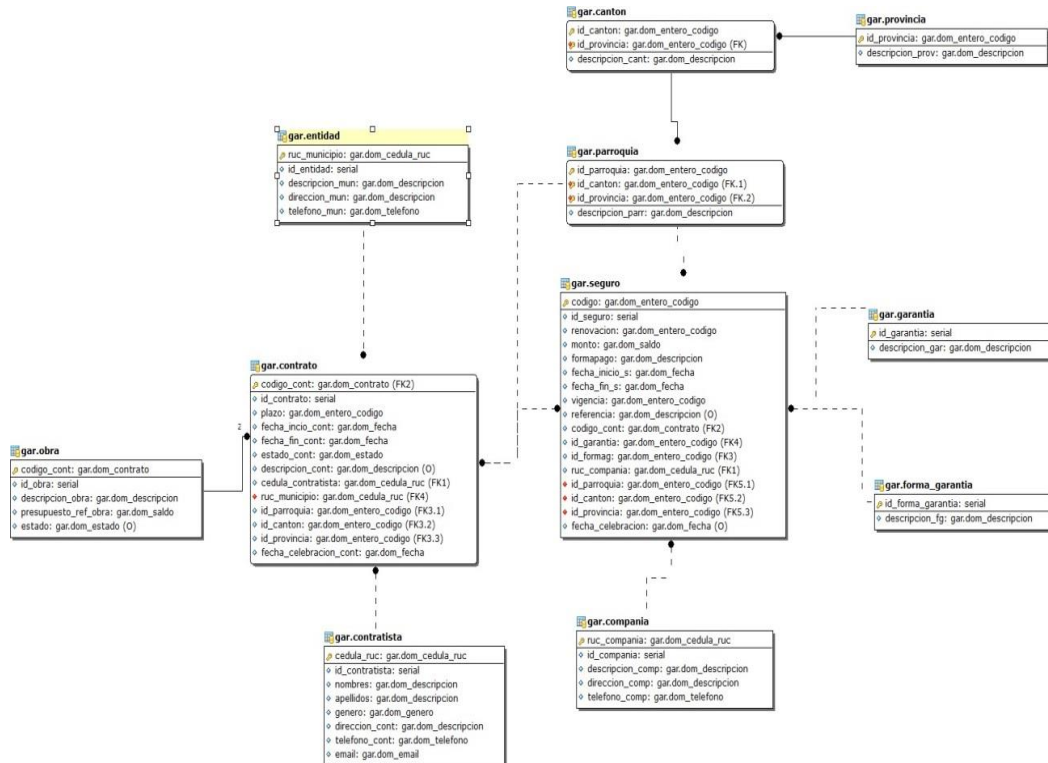
4.3 Diseño del Sistema

El diseño del sistema es una de fases más importante para el desarrollo del software ya que se establecen mecanismos de flexibilidad para cualquier cambio posible, entre sus principales componentes tenemos diagrama de base de datos, diccionario de datos y las interfaces.

4.3.1 Base de Datos

La representación del diagrama de la base de datos se visualizar en la figura IV.1 la cual permitirá un manejo óptimo de información.

FIGURA 4.1 BASE DE DATOS SYSGAR



Fuente: Víctor Colcha

4.3.2 Diccionario de Datos

El diccionario de datos almacena información acerca de la estructura de la base de datos. Es un conjunto de metadatos que contiene las características lógicas y puntuales de los datos que se van a utilizar en el sistema que se programa. En el diccionario de datos se describe el nombre de la tabla, los nombres de los campos o los atributos, tipo de dato, PK o clave primaria, FK o clave foránea, valores no nulos o nulos y una descripción.

En la **Figura IV.2** se detalla el diccionario de datos de la tabla contrato en la cual se puede observar todos los atributos, los tipos de datos y los otros elementos que posee la tabla mencionada, los demás diccionario de datos se pueden encontrar en el anexo xxx.

Tabla Contrato

FIGURA 4.2 DICCIONARIO DE DATOS DE LA TABLA CONTRATO

Field Name	Field Type	Key	Not Null	Default	Description
codigo_cont	gar.dom_contrato (varchar(22))	Primary Key	<input checked="" type="checkbox"/>		pk_contrato
id_contrato	serial		<input checked="" type="checkbox"/>	nextval('gar.contrato_id_cor	
plazo	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		
fecha_inicio_cont	gar.dom_fecha (bigint)		<input checked="" type="checkbox"/>		
fecha_fin_cont	gar.dom_fecha (bigint)		<input checked="" type="checkbox"/>		
estado_cont	gar.dom_estado (varchar(15))		<input checked="" type="checkbox"/>		
descripcion_cont	gar.dom_descripcion (varchar(100))		<input type="checkbox"/>	Null	
cedula_contratista	gar.dom_cedula_ruc (varchar(13))		<input checked="" type="checkbox"/>		fk_contratista
ruc_municipio	gar.dom_cedula_ruc (varchar(13))		<input checked="" type="checkbox"/>		fk_municipio
id_parroquia	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		fk_parroquia
id_canton	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		fk_canton
id_provincia	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		fk_provincia
fecha_celebracion_cc	gar.dom_fecha (bigint)		<input checked="" type="checkbox"/>		

Fuente: Víctor Colcha

4.3.3 Diseño de Interfaz

La Metodología XP define que las interfaces sean fáciles y sencillas, involucrando el menor tiempo posible es el diseño.

Para el desarrollo de las interfaces de usuario o pantallas se determinó un esquema gráfico que posee las siguientes características:

Banner.- También conocido anuncio banner, este se creó con un estilo básico de acuerdo a los estándares de una institución pública, consta del logo del sistema, el de la institución y el nombre del sistema, adicional en la parte izquierda los datos del usuario que se autentico en el mismo.

Menú.- Siguiendo las normas básicas de un sitio web el menú es horizontal, cada ítem del menú posee submenú de acuerdo a la estructura del sistema creado.

Botón notificador.- Se colocó un botón para entrar a las notificaciones en caso que requieran ingresar otra vez

Alertas.- Se colocaron al inicio del sistema SYSGAR dos luces el primero es de color verde que representa la valides de la información y la segunda es de color roja que representa q la información ya caduco.

Fuente.- Para optimizar el desarrollo se crearon hojas de estilos para dar e formato la fuente, tipos, color y el tamaño de letra que se utiliza en la aplicación.

Todas estas características mencionadas se puede visualizan en la Figura IV.3

Para mayor entendimiento del funcionamiento del sistema SYSGAR en el **Anexo xxx** se muestra el manual de usuario del sistema.

FIGURA 4.3 PANTALLA PRINCIPAL SYSGAR



Fuente: Víctor Colcha

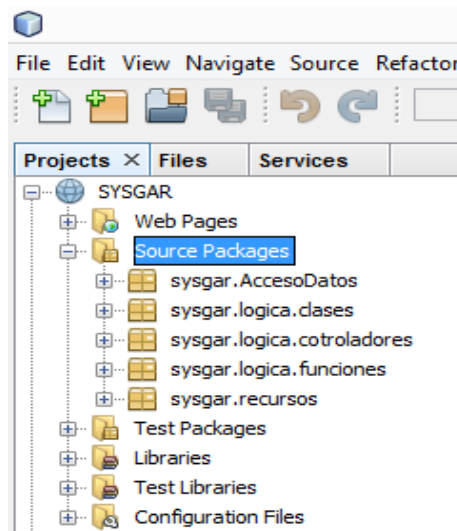
4.4 Codificación

Para la codificación del sistema se empleó el lenguaje de programación Java, el cual es de propósito general, concurrente, orientado a objetos y basado en clases que fue diseñado específicamente para tener tan pocas dependencias de implementación como fuera posible. Su intención es permitir que los desarrolladores de aplicaciones escriban el programa una vez y lo ejecuten en cualquier dispositivo, lo que quiere decir que el código que es ejecutado en una plataforma, no tiene que ser recompilado para correr en otra.

El sistema se elaboró bajo una arquitectura en N-capas a fin de que este sea modular y escalable. Bajo el patrón de diseño MVC permitiendo de esta manera tener un orden en la programación, para ello se desarrolló las capas de: Acceso a datos, Lógica de negocios, Controladores y Vistas, cabe mencionar que el desarrollo de la lógica de negocios también involucró la creación de: Entidades, Parámetros y Servicios.

Cada capa fue creada en el paquete (Java SourcePackage), las mismas que contenían los diferentes archivos .java. Para la codificación del sistema se siguió un estándar propio del autor, para definir aspectos como nombres de: clases, archivos, objetos, atributos, métodos, entidades, parámetros, servicios, controladores. En la Figura IV.5 se visualiza más detalladamente la estructura definida para SYSGAR

FIGURA 4.4 ESTRUCTURA DE SYSGAR



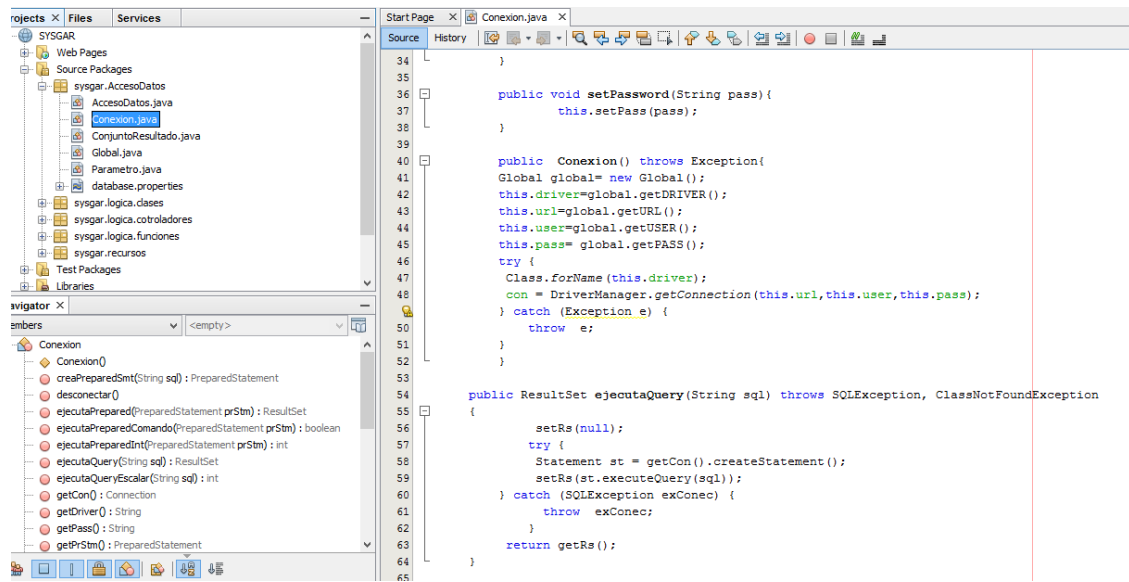
Fuente: Víctor Colcha

A continuación se define cada capa para el desarrollo del sistema de garantías.

Capa de Acceso a Datos

Para la inicialización se definió la capa de acceso a datos, la misma que contiene clases y métodos relacionados con la ejecución de las sentencias SQL de la Base de Datos. La capa de acceso a datos nos ayuda a conectarnos con la base de datos, por ende es de gran utilidad para realizar el módulo de autenticación del sistema, esta capa consta de 5 clases .java y de 1 .properties file los cuales son necesarios para realizar la conexión hacia la base de datos, como se observa en la Figura IV.5 donde se detalla el procedimiento almacenado para la conexión.

FIGURA 4.5 CÓDIGO DE LA CAPA DE ACCESO A DATOS

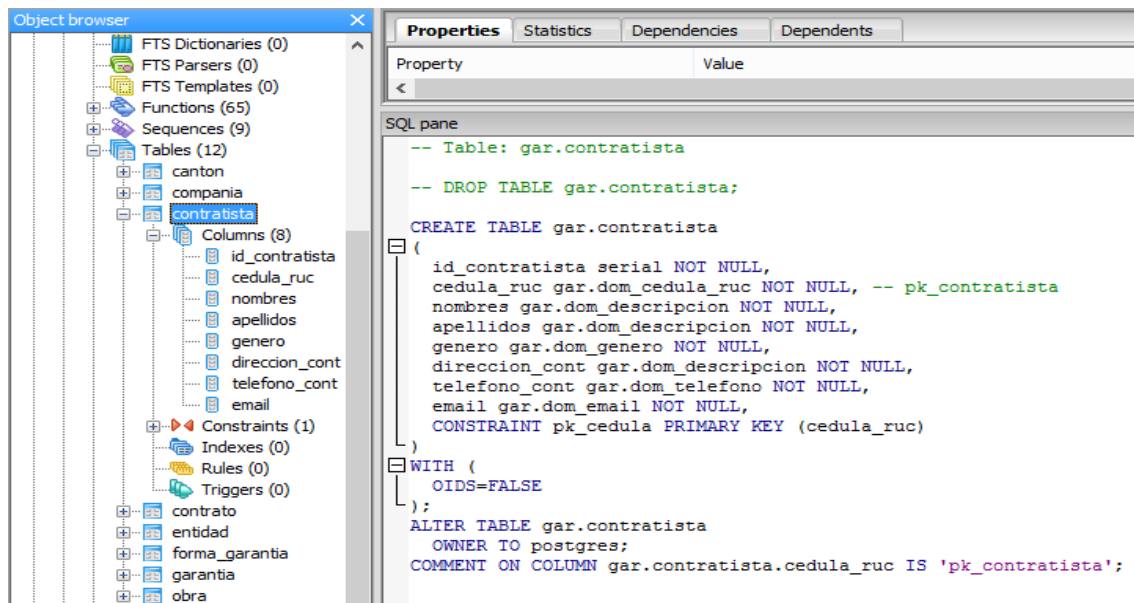


Fuente: Autor

Capa de la lógica de negocios

Luego de realizar la capa de acceso a datos se procedió a realizar la capa lógica de negocios, que en primera instancia se definió las entidades relacionadas las tablas de la base de datos. Es decir por cada tabla y columna una entidad con sus respectivos atributos, en la Figura IV.6 se observa las tablas creadas del SYSGAR. Para para la codificación del sistema en esa capa se dividió en dos elementos las clases y las funciones.

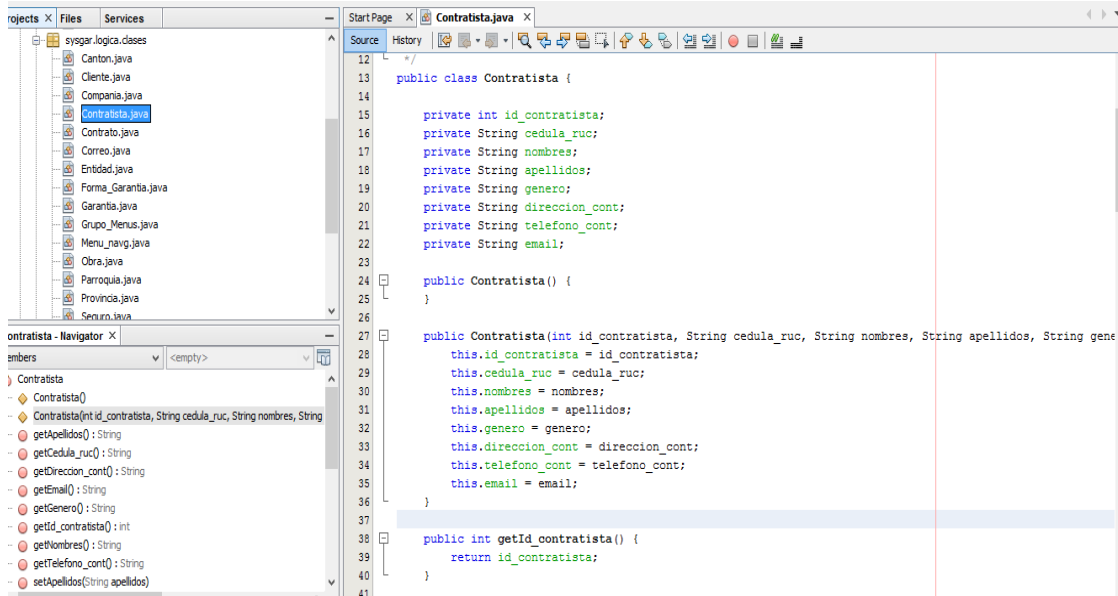
FIGURA 4.6 CAPA LÓGICA DE NEGOCIOS



Fuente: Víctor Colcha

Clases: Como se mencionó anteriormente la clase se creó por cada tabla y columna de la base de datos con sus repetidos atributos. En la Figura IV.7 se puede observar la creación de una clase.

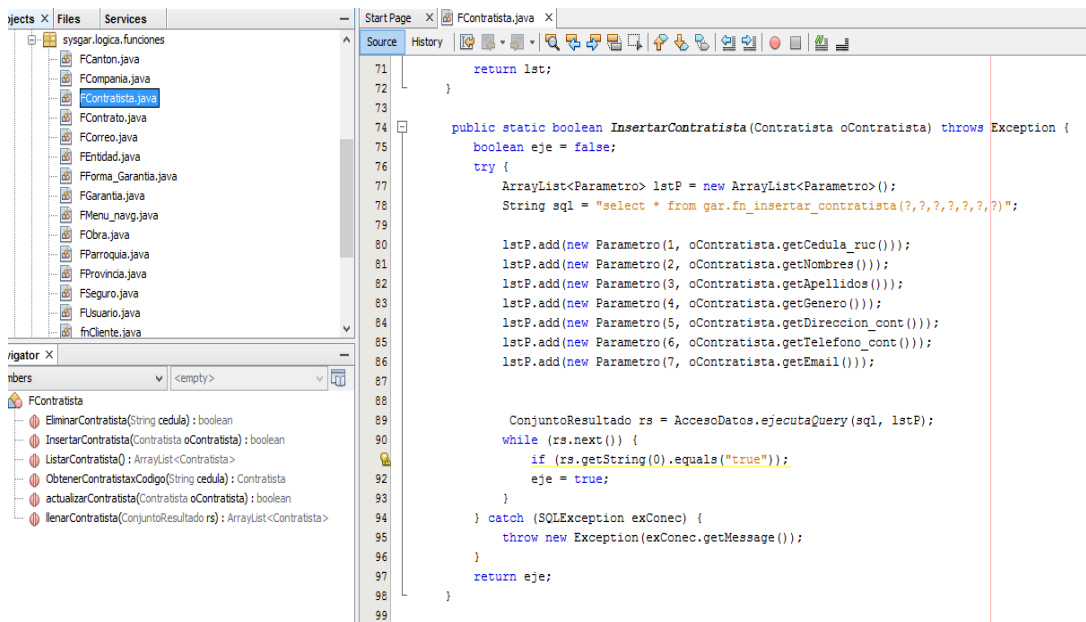
FIGURA 4.7 MODELOS DE CLASES



Fuente: Víctor Colcha

Funciones: En base a las clases se crearon las funciones las cuales van a permitir con una conexión la gestión de los datos de cada tabla, perteneciente a la base de datos del Sistema de Garantías. En la Figura IV.8 se observa un modelo de función.

FIGURA 4.8 MODELO DEFUNCIONES



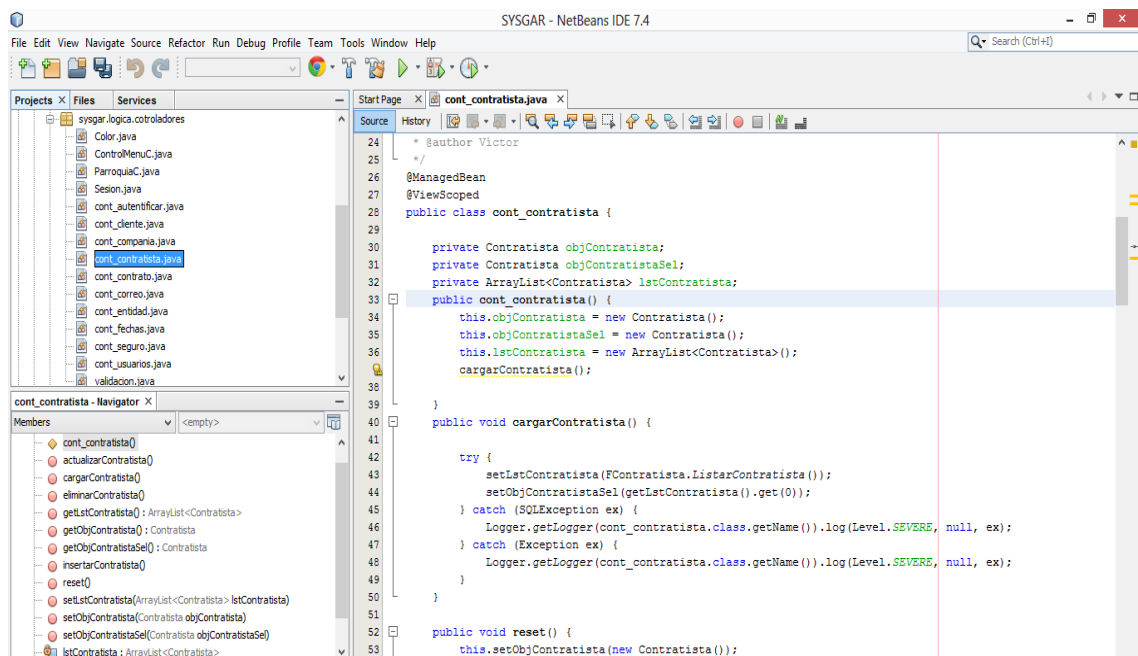
Fuente: Autor

Capa de presentación

Luego de la realización de la capa de lógica de Negocios realizamos, la capa de presentación, en la cual se diseñó clases denominadas controladores y archivos XHTML que con ayuda de otros archivos como CSS, JavaScript permitieron realizar las páginas que actualmente los usuarios navegan. Cabe aclarar que los archivos CSS, JavaScript así como otros forman parte de la capa compartida. En la Figura IV.9 se define las clase controlador y en la Figura IV.10 los archivos XHTML o denominadas vistas.

Controladores: Este es el elemento más importante ya que es el intérprete entre la lógica de negocios y la vista, permitiendo enviar o recibir datos que se encuentran en memoria. En la figura IV.9 se observa un modelo de controlador utilizado en la aplicación.

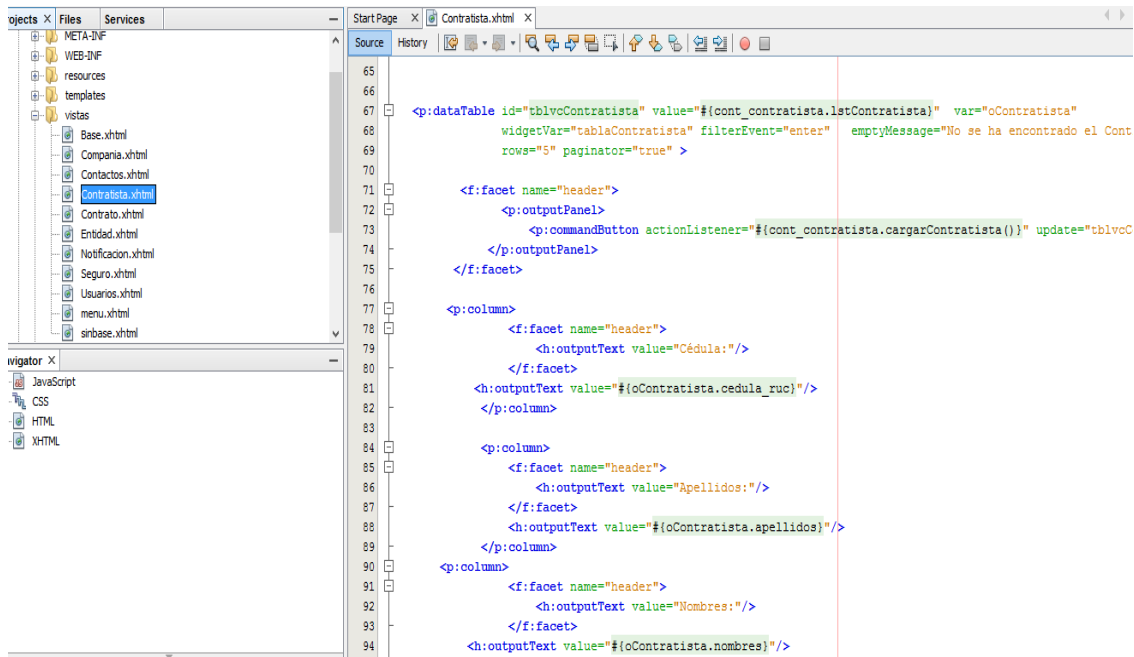
FIGURA 4.9 MODELO DEL CONTROLADOR



Fuente: Víctor Colcha

Vistas: Es la interfaz gráfica del usuario, donde la aplicación con el usuario interactúan para la administración de la información. En la Figura IV.10 se muestra un modelo del código utilizado en la aplicación misma.

FIGURA 4.10 MODELO VISTA



Fuente: Víctor Colcha

4.5 Pruebas

Nadie puede estar seguro de algo si no lo ha probado. En el desarrollo de software, la metodología XP recomienda probar constantemente tanto como sea posible, ya que las pruebas dan la oportunidad de controlar y ver si lo que se está implementado es lo correcto acorde a lo planeado, es decir si se está cumpliendo los requerimientos planteados por el usuario correctamente.

Las pruebas fueron realizadas por cada historia de usuario, a continuación se presenta en Tabla IV.1 la prueba que fue realizada a la historia de usuario No 7. Cada prueba está constituido de los siguientes elementos, formato empleado por el autor.

Numero de prueba: Se especifica el número de la prueba que está realizando.

Historia de usuario: Es nombre de la historia de usuario para la respectiva prueba.

Descripción: Se especifica un relato breve de lo que se desea realizar

Entrada: Son las secuencias de pasos que interviene para cumplir la prueba

Resultado deseado: Es el resultado con la cual se desea concluir la prueba

Prueba: Es el cumplimiento de la prueba

Actividad: Se especifica lo que se está realizando

Autor: Es la persona a quien va dirigida la prueba

Modificador: Es la persona quien rectifica las pruebas

Observaciones: Se especifica los fallos o problemas encontrados para su correcta solución.

Esta prueba de muestra está realizada principalmente porque se especifica uno de las partes importantes que representa el sistema.

Prueba para la historia de usuario No 7: Almacenar las renovaciones

TABLA 4.XI PRUEBA HISTORIA DE USUARIOS No 7

No. 1	HISTORIA DE USUARIO: Almacenamiento de la información de las renovaciones.	
Descripción: El secretario del GAD podrá realizar las renovaciones de la garantía (póliza de seguro).		
Entradas <ul style="list-style-type: none">• El secretario GAD ingresa al sistema SYSGAR como administrador.• El sistema visualiza una notificación en la pantalla principal de las garantías que están por vencer la fecha limite• En el menú se selecciona garantía y luego en renovación.• Se ingresa el código del contrato y se activa la opción renovar• Se despliega una pantalla y registra los datos de la nueva renovación		
Resultado deseado Una vez ingresado los datos acepta y emite un mensaje de renovación realizada.		Prueba: Satisfactoria
Actividad: Prueba de historia de usuario		
Autor : Secretario GAD de Cantón Chimbo	Modificador: Víctor Colcha	
Observaciones: En esta prueba realizada se pudo observar que la estructura de las fechas tenía formatos diferentes, lo cual representaba un problema para el usuario rectificándose inmediatamente.		

Fuente: Víctor Colcha

CONCLUSIONES

- La herramienta **DBTools Manager** permite controlar los dos SGBD, se demostró en la prueba **listar datos** de acuerdo a todos los escenarios planteados, PostgreSQL supera con una diferencia porcentual del **7.2%** con relación a Firebird.
- Mediante el análisis comparativo se demostró que Firebird supera a PostgreSQL con el **8.4 %** de diferencia en **tiempos de respuesta** para la gestión de la información, mientras que con el **9.6%** de diferencia PostgreSQL supera a Firebird en el uso de la memoria **RAM** y **procesador**. Concluyendo que en rendimiento PostgreSQL supera con un valor mínimo del **1.2%** de diferencia a Firebird, calificándolos a ambos con una equivalencia de **MUY BUENA** de acuerdo a la ponderación establecida en esta investigación.
- Para pruebas adicionales con consultas mal definidas en un volumen de 1000 datos, respondieron en un tiempo de respuesta con **281 ms** de PostgreSQL que es menor al de Firebird que obtuvo un tiempo de **21min; 58s; 204 ms**. Mientras que en 100000 datos PostgreSQL logro un tiempo de **1.57 horas** y Firebird presento inconvenientes al no visualizar los datos pasados las **8 horas**. Demostrando así que PostgreSQL es totalmente robusto.
- Se demostró mediante las tablas de lecturas en **tiempos de repuesta** , que el SGBD PostgreSQL la **primera lectura** de datos obtenida varia más de **60%** del promedio totalmente de las demás lecturas, mientras que en Firebird solo vario un **20%**, cabe mencionar que desde la segunda toma se estabilizaron lectura en los dos SGBD.
- Se ha desarrollado el sistema de garantías con el motor de gestión de base de datos PostgreSQL por ser el ganador en el análisis de rendimiento, presentar robustez y controlar la integridad de los datos.

RECOMENDACIONES

- Se recomienda para el desarrollo de futuras aplicaciones y manejo de pequeños o grandes volúmenes de datos como una de las alternativas a PostgreSQL porque ha demostrado robustez, variabilidad en la manipulación de los datos, consumo estable de recurso, control de integridad de datos, etc.
- Si el usuario requiere una alta velocidad en la gestión de datos con pequeños y medianos volúmenes de información en base a lo expuesto en esta investigación, se sugiere que Firebird es una opción fiable ya que proporciona estabilidad en la manipulación de datos, consumo de recursos variable, control de integridad de datos, pero no garantiza la robustez de la misma.
- Es conveniente detectar antes de ejecutar las pruebas de este tipo de análisis detener todos los procesos innecesarios de sistema operativo para obtener la toma de muestras reales y no varié en una o varias interacciones.
- Se recomienda en futuras investigaciones utilizar caracteres especiales y de comparación ya que permite visualizar acciones inesperadas en el proceso de manipulación de la información.

RESUMEN

El propósito de esta Investigación es realizar un análisis comparativo entre los motores de base de datos libres PostgreSQL 9.2.3 y Firebird 2.5.2 de versiones estables con el fin de medir su rendimiento, en donde seleccionaremos el mejor para la utilización en una aplicación web para el control de garantías en los contratos de las obras públicas, aplicado al Gobierno Autónomo Descentralizado (GAD) Municipal del Cantón Chimbo, Provincia de Bolívar.

Por medio del método científico se analizaron los resultados de los datos obtenidos en los distintos escenarios de pruebas. Los parámetros para el análisis es el desempeño que representa los tiempos de respuesta en la gestión de datos y el consumo de recurso que es el uso del hardware. Como herramientas se utilizaron los software PgAdmin III que va a controlar los datos en PostgreSQL, IBExpert para el control de datos en Firebird y una herramienta externa el DBTools Manager Professional que controlara ambas bases de datos. Mientras para el consumo de recursos se utilizó el administrador de tareas de Windows.

Obteniendo como resultados finales en el desempeño más el consumo de recurso a PostgreSQL con el 77.6% y Firebird con un 76.4%. Por lo tanto ambos reciben una equivalencia de muy buena.

Manifestando como conclusión que PostgreSQL supera con un valor mínimo del 1.2% de diferencia a Firebird, por lo que se seleccionó el motor de base de datos PostgreSQL para la creación del sistema de garantías del Gobierno Autónomo Descentralizado Municipal del Cantón Chimbo.

Se recomienda a los técnicos del proceso de contratación pública del GAD Municipal del Cantón Chimbo el uso del Sistema de Garantías para un óptimo desempeño de sus actividades.

Además para el desarrollo de futuras aplicaciones una de las alternativas es PostgreSQL porque ha demostrado robustez y variabilidad en la manipulación de los datos.

Palabras claves: <ANÁLISIS COMPARATIVO> <SISTEMA DE GARANTÍAS> <BASE DE DATOS POSTGRESQL> <BASE DE DATOS FIREBIRD> <DESEMPEÑO DE LA BASE DE DATOS> <CONSUMO DE RECURSOS> <VERSIÓN ESTABLE> <SOFTWARE DBTOOLS MANAGER PROFESSIONAL> <SOFTWARE IBEXPERT> <SOFTWARE PGADMIN III>

SUMMARY

The purpose of this research is a comparative analysis between database engines and free data PostgreSQL 9.2.3 Firebird 2.5.2 stable versions in order to render performance, where would select the best for use in a web application for the guarantees control in the contracts of public works, applied to the Autonomous Decentralized Government (GAD) of Chimbo Canton, in Bolivar Province.

Through the scientific method the results of the data obtained in different test scenarios were analyzed. The parameters for performance analysis that represents the response times in data management and resource consumption that is the use of hardware. PgAdmin III as the software tools that will handle the data in PostgreSQL, IBExpert for Firebird data control and an external tool the DBTools Manager Professional will control both databases were used. While resource consumption for the Windows Task Manager was used.

Obtaining as final results on performance rather resource consumption PostgreSQL with 77.6% and 76.4% Firebird. Therefore both receive a good equivalence.

Manifesting as conclusion that PostgreSQL exceeds a minimum value of 1.2% in difference in Firebird, which the PostgreSQL database engine data was selected, to create the guarantees system of Chimbo Government.

It is recommended that specialist's procurement process GAD Chimbo Cantón the use of the guarantee system for optimal performance of the activities.

In addition to the development of future applications an alternative is PostgreSQL because it has proved variability in robustness and data manipulation.

Keywords: <COMPRARATIVE ANALYSIS> <WARRANTIES SYSTEM> <POSTGRES DATABASE> <FIREBIRD DATABASE> <PERFORMANCE OF THE DATABASE> <CONSUMPTION OF RESOURCES> <STABLE VERSION> <SOFTWARE DBTOOL MANAGER PROFESSIONAL> <IBEXPERT SOFTWARE> <PGADMIN III SOFTWARE>

BIBLIOGRAFÍA

[1] **CORONEL C.**, Bases de Datos, Diseño, Implementación y Administración, 9na ed., México D.F. - México., Cengage Learning Editores, 2011., Pp 5-23.

https://books.google.es/books?id=KINC0Gc_RREC
2014-08-14

[2] **RUIZ N, RODRÍGUEZ J, Y OTROS.**, Introducción a las bases de datos: el modelo relacional., Madrid - España., Editorial Paraninfo., 2005., Pp 2-58.

<https://books.google.es/books?id=2-HwGwc57c0C>
2014-09-12

[3] **KEDAR S.**, Database Management System., Maharashtra -India., Technical. Publications Pune., 2009., Pp 1-40.

<https://books.google.com.ec/books?id=SELyJ4a5nTcC>
2014-10-28

[4] **COBO A.**, Diseño y programación de bases de datos. Madrid - España., Editorial Visión Libros., 2007., Pp 15-22

<https://books.google.com.ec/books?id=anCDr9N-kGsC>
2014-09-12

[5] **DRAKE J. Y WORSLEY J.C.**, Practical PostgreSQL., California - United States of America., O'Reilly Media, Inc., 2002., Pp 3-31, 98- 157.

<https://books.google.es/books?id=fI11AgAAQBAJ>
2015-01-12

[6] **CUERVO M.C.**, Evaluación del rendimiento de los motores de bases de datos Mysql y Firebird., Medellín - Colombia., Rev. Univ. EAFIT., vol. 43, n.º 148, may 2012, Pp. 78-90.

<http://publicaciones.eafit.edu.co/index.php/revista-universidad-eafit/article/viewFile/703/625>
2015-02-09

[7] PILCO, L., Estudio Comparativo entre Sistemas de Gestión de Bases de Datos Libres FIREBIRD y MYSQL en el Desarrollo de Aplicaciones Web. Caso Práctico: Sistema para la Gestión de Equipos de Radio del Escuadrón de Comunicaciones No. 11., (TESIS) Ing. En Sistemas Informáticos., Escuela Superior Politécnica de Chimborazo., Facultad Informática y Electrónica., Escuela Ingeniería en Sistemas., Riobamba - Ecuador., 2012., Pp 40-70.

<http://dspace.esPOCH.edu.ec/handle/123456789/1534>

2014-11-14

[8] SISTEMAS GESTORES DE BASES DE DATOS

<http://www.freelibros.org/base-de-datos/sistemas-gestores-de-bases-de-datos.html>.

2015-01-06

[9] GESTORES DE BASE DE DATOS: GESTOR DE BASE DE DATOS: FIREBIRD

<http://database-firebird.blogspot.com/2011/06/gestor-de-base-de-datos-firebird.html>

2015-02-01

[10] DESARROLLO DE SISTEMAS, INFORMÁTICA Y SOFTWARE: EL GESTOR DE BASE DE DATOS FIREBIRD.

<http://alexechavarriaperez.blogspot.com/2011/05/el-gestor-de-base-de-datos-firebird.html>

2015-02-10

[11] SISTEMAS DE GESTOR DE BASE DE DATOS LIBRES

<https://b1m2.wordpress.com/2011/11/15/sistemas-de-gestor-de-base-de-datos-libres-firebird/>

2015-02-16

[12] SGBD CARACTERÍSTICAS, VENTAJAS, DESVENTAJAS Y REQUERIMIENTOS.

http://www.academia.edu/8199329/SGBD_CARACTERISTICAS_VENTAJAS_DESVENTAJAS_REQUERIMIENTOS.

2015-01-11

[13] DBTOOLS SOFTWARE

<http://www.dbtools.com.br/EN/dbmanagerpro/>

2015-01-12

[14] FIREBIRD

<http://sabd15n1.wikispaces.com/Firebird>

2014-12-01

[15] HERRAMIENTA DE ADMINISTRACIÓN DE BASE DE DATOS PARA POSTGRESQL

https://compumat.uci.cu/sites/default/files/public/p876-ponencia-3297_0.pdf

2015-01-12

[16] FIREBIRD EN LA INDUSTRIA, Scribd.

<https://es.scribd.com/doc/37412397/Firebird-en-La-Industria>

2015-03-13

[17] HISTORIA DE POSTGRESQL HASTA AHORA.

<http://2ndquadrant.com/es/postgresql/postgresql-la-historia-hasta-ahora/>

2015-03-13

[18] ARQUITECTURA DEL SGBD FIREBIRD

<http://sabd15n1.wikispaces.com/Firebird>

2015-02-07

[19] ESTUDIO DEL SISTEMA DE GESTIÓN DE BASE DE DATOS POSTGRESQL

<https://riunet.upv.es/bitstream/handle/10251/17490/Memoria.pdf>

2015-01-12

[20] METODOLOGÍAS RUP, UWE, AUP y XP.

<http://danieldejesustorres.blogspot.com/2013/10/metodologias-rup-uwe-aup-y-xp.html>

2015-03-11

[21] COMPONENTES DE LOS SGBD.

<http://dryvalleycomputer.com/index.php/bases-de-datos/introduccion/48-componentes-de-los-sgbd>

2015-02-11

[22] DOCUMENTATION DATABASE FIREBIRD.

<http://www.firebirdsql.org/manual/es/firebird-database-documentation-es.html>

2015-02-26

[23] AN INTRODUCTION TO IBEXPERT & FIREBIRD

<ftp://bravosoft.com/Databases/IBE%20Personal%20Edition/IBExpertFirebirdGuide.pdf>

2015-02-26

[24] ESTADÍSTICA NO PARAMÉTRICA PRUEBA CHI-CUADRADO

http://www.uoc.edu/in3/emath/docs/Chi_cuadrado.pdf

2015-03-15

[25] CARACTERÍSTICAS DE LOS SGBD (SISTEMAS GESTORES DE BASES DE DATOS)

<http://es.scribd.com/doc/63991597/Caracteristicas-del-SGBD#scribd>

2015-06-15

[26] SOBRE POSTGRESQL

http://www.postgresql.org/es/sobre_postgresql#caracteristicas

2015-02-27

[27] CHAPTER 8. DATA TYPES

<http://www.postgresql.org/docs/9.2/static/datatype.html>

2015-03-01

[28] FIREBIRD: CARACTERÍSTICAS BÁSICAS

http://www.firebird.com.mx/descargas/documentos/tema_2-caracteristicas_basicas.pdf

2015-02-24

[29] LIMITES DE FIREBIRD

<http://firebirdmanual.com/firebird/es/firebird-manual/2/limites-de-firebird/36>

2015-02-24

[30] THE DATABASE EXPERTS

<http://ibexpert.net/ibe/index.php>

2015-02-24

[31] ÉNFASIS INGENIERÍA DE SOFTWARE

<http://es.slideshare.net/tutorialsruby/cuaderno762-2620873>

2015-02-24

[32] MEDIDORES DEL DESEMPEÑO EN SEGURIDAD INFORMÁTICA

<http://www.auditool.org/blog/auditoria-de-ti/1264-los-medidores-del-desempeno-en-la-seguridad-informatica>

2015-03-04

[33] INTEGRIDAD, ROBUSTEZ Y ESTABILIDAD

<http://es.slideshare.net/memodevia/integridad-26651262>

2015-03-14

[34] PENSAMIENTO Y GESTIÓN

<http://www.redalyc.org/articulo.oa?id=64602002>

2015-04-24

[35] LA PRUEBA CHI-CUADRADA (χ^2)

http://dentizta.ccadet.unam.mx/Objetosv2/papime_e/pdfs/b_001.pdf

2015-03-16

[36] METODOLOGÍAS ÁGILES PARA EL DESARROLLO DE SOFTWARE

<http://www.cyta.com.ar/ta0502/v5n2a1.htm>

2015-03-22

ANEXOS

ANEXO 1

SENTENCIAS SQL PARA LA PRUEBA DE DESEMPEÑO

a. **POSTGRESQL:** Procedimiento almacenado ingresar (Un parámetro de entrada).

POSTGRESQL - CÓDIGO INSERTAR

```
begin
v = num_registros / 4;
i=1;
while( i <= v ) loop
INSERT INTO gar.compania(ruc_compania, descripcion_comp, direccion_comp, telefono_comp)
VALUES ('060'||100000 +i || i % 10 || '001', 'Comp. seguros'||i, 'matrix' ||i, '(593-3)|| 1000000+i);
i = i+1;
end loop;
i=1;
while( i <= num_registros ) loop
INSERT INTO gar.obra( descripcion_obra, presupuesto_ref_obra, codigo_cont,estado)
VALUES ('Obra_'||cast(i as varchar), 100.01 + i, 'Contrato_'||cast(i as varchar), 'Activo');
i = i+1;
end loop;
i=1;
while( i <= num_registros ) loop
if (i < 999999) then
INSERT INTO gar.contratista(cedula_ruc, nombres, apellidos, genero, direccion_cont,telefono_cont,
email)
VALUES ('060'||100000 +i || '-' || i % 10, 'nombre_' || i, 'apellido_' || i, 'M', 'direccion_' ||i, '(593-3)||
1000000+i,'nombre_' || i || '@gmail.com' );
INSERT INTO gar.contrato(codigo_cont, plazo, fecha_inicio_cont,fecha_fin_cont, estado_cont,
descripcion_cont, cedula_contratista, ruc_municipio, id_parroquia, id_canton, id_provincia,
fecha_celebracion_cont)
VALUES ('Contrato_'||cast(i as varchar), i+2, 1111+i, 1111+i, 'Activo', 'principal obra', '060'||100000
+i || '-' || i % 10, '0260000680001', 50, 3, 2, 1111);
else
if (i < 999999999) then
INSERT INTO gar.contratista(cedula_ruc, nombres, apellidos, genera, direccion_cont,telefono_cont,
email)
VALUES (100000000 +i || '-' || i % 10, 'nombre_' || i, 'apellido_' || i, 'F', 'direccion_' ||i, '(593-3)||
1000000+i,'nombre_' || i || '@gmail.com' );
INSERT INTO gar.contrato(codigo_cont, plazo, fecha_inicio_cont,fecha_fin_cont, estado_cont,
descripcion_cont, cedula_contratista, ruc_municipio, id_parroquia, id_canton, id_provincia,
fecha_celebracion_cont)
VALUES ('Contrato_'||cast(i as varchar), i+2, 1111+i, 1111+i, 'Activo', 'principal obra', 100000000 +i
|| '-' || i % 10, '0260000680001', 50, 3, 2, 1111);
else
INSERT INTO gar.contratista(cedula_ruc, nombres, apellidos, genero, direccion_cont,telefono_cont,
email)
VALUES (1000000000+i , 'nombre_' || i, 'apellido_' || i, 'M', 'direccion_' ||i, '(593-3)||
1000000+i,'nombre_' || i || '@gmail.com' );
INSERT INTO gar.contrato(codigo_cont, plazo, fecha_inicio_cont,fecha_fin_cont, estado_cont,
descripcion_cont, cedula_contratista, ruc_municipio, id_parroquia, id_canton, id_provincia,
fecha_celebracion_cont)
VALUES ('Contrato_'||cast(i as varchar), i+2, 1111+i, 1111+i, 'Activo', 'principal obra', 1000000000+i
,'0260000680001', 50, 3, 2, 1111);
end if;
end if;
i = i+1;
end loop;
i=1;
y=1;
```



```

r=0;
p=25;
estado = 'Terminado';
while( i <= num_registros +3 ) loop
if( y <= v ) then
INSERT INTO gar.seguro(codigo, renovacion, monto, formapago, fecha_inicio_s, fecha_fin_s,
vigencia, referencia, codigo_cont, id_garantia, id_formag, ruc_compania, id_parroquia, id_canton,
id_provincia, fecha_celebracion)
VALUES (i, r, 10000, p||'%', 111111, 11111, 20, estado, 'Contrato_'||cast(y as varchar), 1, 2,
'060'||100000 +y || y % 10 || '001', 50, 3, 2, 111111);
else
y=0;
r=r+1;
p=p+25;
if(p>74) then
estado = 'Activo';
end if;
end if;
y= y+1;
i = i+1;
end loop;
return true;
end;

```

b. POSTGRESQL: Procedimiento almacenado actualizar (Un parámetro de entrada)

CÓDIGO ACTUALIZAR POSTGRESQL

```

begin
v = num_registros / 4;
i=1;
while( i <= v ) loop
UPDATE gar.compania
SET descripcion_comp='Actualizado Comp. seguros'||i
WHERE ruc_compania='060'||100000 +i || i % 10 || '001';
i = i+1;
end loop;
i=1;
while( i <= num_registros ) loop
UPDATE gar.obra
SET descripcion_obra='Actualizado Obra_'||cast(i as varchar)
WHERE codigo_cont='Contrato_'||cast(i as varchar);
UPDATE gar.contrato
SET descripcion_cont='Actualizar principal obra_'||cast(i as varchar)
WHERE codigo_cont='Contrato_'||cast(i as varchar);
UPDATE gar.seguro
SET id_garantia=2
WHERE codigo=i;
i = i+1;
end loop;
i=1;
while( i <= num_registros ) loop
if( i < 999999 ) then
UPDATE gar.contratista
SET nombres= 'Actualizado nombre_'|| i
WHERE cedula_ruc = '060'||100000 +i || '-' || i % 10;
else
if( i < 999999999 ) then
UPDATE gar.contratista

```

```

SET nombres= 'Actualizado nombre_'|| i
WHERE cedula_ruc =100000000 +i || '-' || i % 10;
else
UPDATE gar.contratista
SET nombres= 'Actualizado nombre_'|| i
WHERE cedula_ruc =1000000000+i;
end if ;
end if;
i = i+1;
end loop;
return true;
end

```

c. POSTGRESQL - Sentencia SQL listar datos de complejidad simple

LISTAR DATOS DE COMPLEJIDAD SIMPLE

```

SELECT
  ob.descripcion_obra as Obra,
  sg.formapago as forma_pago,
  ct.codigo_cont as Contrato,
  sg.renovacion as Renovacion
FROM  gar.obra as ob inner join gar.contrato as ct on (ob.codigo_cont = ct.codigo_cont)
inner join gar.seguro as sg on (sg.codigo_cont = ct.codigo_cont)
GROUP BY  ob.descripcion_obra, ct.codigo_cont, sg.formapago, sg.renovacion, sg.id_seguro
ORDER BY  sg.id_seguro DESC, ct.codigo_cont;

```

d. FIREBIRD- Sentencia SQL listar datos de complejidad simple

Es el mismo código del **literal c** con la diferencia que Firebird no posee esquemas

e. Sentencia SQL listar datos de complejidad media

LISTAR DATOS DE COMPLEJIDAD MEDIA

```

SELECT
sg.renovacion AS Renovacion,
ct.codigo_cont AS Contrato,
ctt.apellidos AS Apellido,
ctt.nombres AS Nombre,
ob.descripcion_obra AS Obra,
cp.descripcion_comp AS Compania,
sg.referencia AS Estado
FROM  gar.obra AS ob INNER JOIN gar.contrato AS ct ON(ob.codigo_cont = ct.codigo_cont)
      INNER JOIN gar.seguro AS sg ON (ct.codigo_cont = sg.codigo_cont)
      INNER JOIN gar.contratista AS ctt ON (ctt.cedula_ruc = ct.cedula_contratista)
      INNER JOIN gar.compania AS cp ON (cp.ruc_compania = sg.ruc_compania)
GROUP BY  sg.renovacion, ct.codigo_cont, ctt.apellidos, ctt.nombres, ob.descripcion_obra,
ct.codigo_cont, cp.descripcion_comp, sg.referencia, sg.id_seguro
ORDER BY  sg.id_seguro DESC, ct.codigo_cont, ctt.apellidos DESC;

```

f. **FIREBIRD** -Sentencia SQL listar datos de complejidad media

Es el mismo código del **literal e**, con la diferencia que FIREBIRD no posee esquemas

g. **POSTGRESQL**- Sentencia SQL listar datos de complejidad alta

LISTAR DATOS DE COMPLEJIDAD ALTA:

```
SELECT
sg.id_seguro AS Id_seguro,
sg.renovacion AS Renovacion,
ct.codigo_cont AS Contrato,
ctt.apellidos AS Apellido,
ctt.nombres AS Nombre,
ctt.cedula_ruc AS Cedula,
ob.descripcion_obra AS Obra,
cp.descripcion_comp AS Compania,
g.descripcion_gar AS Garantia,
c.descripcion_cant AS Canton,
sg.formapago AS Porcentaje_Pago,
sg.referencia AS Estado
FROM gar.obra AS ob INNER JOIN gar.contrato AS ct ON(ob.codigo_cont = ct.codigo_cont)
      INNER JOIN gar.seguro AS sg ON (ct.codigo_cont = sg.codigo_cont)
      INNER JOIN gar.contratista AS ctt ON (ctt.cedula_ruc = ct.cedula_contratista)
      INNER JOIN gar.compania AS cp ON (cp.ruc_compania = sg.ruc_compania)
      INNER JOIN gar.forma_garantia AS fg ON (fg.id_forma_garantia = sg.id_formag)
      INNER JOIN gar.garantia AS g ON(g.id_garantia=sg.id_garantia)
      INNER JOIN gar.parroquia AS p ON (p.id_parroquia= sg.id_parroquia AND p.id_canton
=s.g.id_canton AND p.id_provincia= sg.id_provincia)
      INNER JOIN gar.canton AS c ON (c.id_canton=p.id_canton AND c.id_provincia=
p.id_provincia)
GROUP BY sg.id_seguro, sg.renovacion, ct.codigo_cont, ctt.apellidos, ctt.nombres, ctt.cedula_ruc,
ob.descripcion_obra, cp.descripcion_comp, g.descripcion_gar, sg.referencia, c.descripcion_cant,
sg.formapago, sg.referencia
ORDER BY sg.id_seguro DESC, ct.codigo_cont, ctt.apellidos DESC, ob.descripcion_obra ;
```

h. **FIREBIRD** -Sentencia SQL listar datos de complejidad Alta

Es el mismo código del **literal g** con la diferencia que FIREBIRD no posee esquemas

i. **POSTGRESQL** Sentencia SQL buscar datos de complejidad simple

BUSCAR DATOS DE COMPLEJIDAD SIMPLE

```
SELECT
  ob.descripcion_obra as Obra,
  sg.formapago as forma_pago,
  ct.codigo_cont as Contrato,
  sg.renovacion as Renovacion
FROM gar.obra as ob inner join gar.contrato as ct on (ob.codigo_cont = ct.codigo_cont)
inner join gar.seguro as sg on (sg.codigo_cont = ct.codigo_cont)
WHERE (sg.renovacion =2)
GROUP BY ob.descripcion_obra, ct.codigo_cont, sg.formapago, sg.renovacion, sg.id_seguro
ORDER BY sg.id_seguro DESC, ct.codigo_cont;
```

j. FIREBIRD Sentencia SQL buscar datos de complejidad simple

Es el mismo código del **literal i** con la diferencia que FIREBIRD no posee esquemas

k. Sentencia SQL buscar datos de complejidad media

BUSCAR DATOS DE COMPLEJIDAD MEDIA:

```
SELECT
sg.renovacion AS Renovacion,
ct.codigo_cont AS Contrato,
ctt.apellidos AS Apellido,
ctt.nombres AS Nombre,
ob.descripcion_obra AS Obra,
cp.descripcion_comp AS Compania,
sg.referencia AS Estado
FROM gar.obra AS ob INNER JOIN gar.contrato AS ct ON(ob.codigo_cont = ct.codigo_cont)
INNER JOIN gar.seguro AS sg ON (ct.codigo_cont = sg.codigo_cont)
INNER JOIN gar.contratista AS ctt ON (ctt.cedula_ruc = ct.cedula_contratista)
INNER JOIN gar.compania AS cp ON (cp.ruc_compania = sg.ruc_compania)
WHERE (ct.cedula_contratista IN (SELECT cedula_ruc FROM gar.contratista WHERE apellidos
LIKE'apellido%4' OR apellidos LIKE'apellido%1%') and sg.formapago='25%' and sg.referencia like
'Terminado')
GROUP BY sg.renovacion, ct.codigo_cont, ctt.apellidos, ctt.nombres, ob.descripcion_obra,
ct.codigo_cont, cp.descripcion_comp, sg.referencia, sg.id_seguro
ORDER BY sg.id_seguro DESC, ct.codigo_cont, ctt.apellidos DESC;
```

l. FIREBIRD -Sentencia SQL listar datos de complejidad media

Es el mismo código del **literal k** con la diferencia que FIREBIRD no posee esquemas

m. POSTGRESQL -Sentencia SQL buscar datos de complejidad alta

BUSCAR DATOS DE COMPLEJIDAD MEDIA

```
SELECT
sg.id_seguro AS Id_seguro,
sg.renovacion AS Renovacion,
ct.codigo_cont AS Contrato,
ctt.apellidos AS Apellido,
ctt.nombres AS Nombre,
ctt.cedula_ruc AS Cedula,
ob.descripcion_obra AS Obra,
cp.descripcion_comp AS Compania,
g.descripcion_gar AS Garantia,
c.descripcion_cant AS Canton,
sg.formapago AS Porcentaje_Pago,
sg.referencia AS Estado
FROM gar.obra AS ob INNER JOIN gar.contrato AS ct ON(ob.codigo_cont = ct.codigo_cont)
INNER JOIN gar.seguro AS sg ON (ct.codigo_cont = sg.codigo_cont)
INNER JOIN gar.contratista AS ctt ON (ctt.cedula_ruc = ct.cedula_contratista)
INNER JOIN gar.compania AS cp ON (cp.ruc_compania = sg.ruc_compania)
INNER JOIN gar.forma_garantia AS fg ON (fg.id_forma_garantia = sg.id_formag)
INNER JOIN gar.garantia AS g ON(g.id_garantia=sg.id_garantia)
INNER JOIN gar.parroquia AS p ON (p.id_parroquia= sg.id_parroquia AND p.id_canton
=sg.id_canton AND p.id_provincia= sg.id_provincia)
INNER JOIN gar.canton AS c ON (c.id_canton=p.id_canton AND c.id_provincia=
p.id_provincia)
WHERE (sg.codigo_cont IN (SELECT codigo_cont FROM gar.obra WHERE descripcion_obra LIKE
'Obra%6%' OR (descripcion_obra LIKE 'Obra%8%') ) AND sg.renovacion=0 AND ob.estado
LIKE'Activo' )
GROUP BY sg.id_seguro, sg.renovacion, ct.codigo_cont, ctt.apellidos, ctt.nombres, ctt.cedula_ruc,
```

```

ob.descripcion_obra, cp.descripcion_comp, g.descripcion_gar, sg.referencia, c.descripcion_cant,
sg.formapago, sg.referencia
UNION
SELECT
sg.id_seguro AS Id_seguro,
sg.renovacion AS Renovacion,
ct.codigo_cont AS Contrato,
ctt.apellidos AS Apellido,
ctt.nombres AS Nombre,
ctt.cedula_ruc AS Cedula,
ob.descripcion_obra AS Obra,
cp.descripcion_comp AS Compania,
g.descripcion_gar AS Garantia,
c.descripcion_cant AS Canton,
sg.formapago AS Porcentaje_Pago,
sg.referencia AS Estado
FROM gar.obra AS ob INNER JOIN gar.contrato AS ct ON(ob.codigo_cont = ct.codigo_cont)
INNER JOIN gar.seguro AS sg ON (ct.codigo_cont = sg.codigo_cont)
INNER JOIN gar.contratista AS ctt ON (ctt.cedula_ruc = ct.cedula_contratista)
INNER JOIN gar.compania AS cp ON (cp.ruc_compania = sg.ruc_compania)
INNER JOIN gar.forma_garantia AS fg ON (fg.id_forma_garantia = sg.id_formag)
INNER JOIN gar.garantia AS g ON(g.id_garantia=sg.id_garantia)
INNER JOIN gar.parroquia AS p ON (p.id_parroquia= sg.id_parroquia AND p.id_canton
=sg.id_canton AND p.id_provincia= sg.id_provincia)
INNER JOIN gar.canton AS c ON (c.id_canton=p.id_canton AND c.id_provincia=
p.id_provincia)
WHERE (sg.codigo_cont IN (SELECT codigo_cont FROM gar.contrato WHERE cedula_contratista
LIKE '%-9' OR (cedula_contratista LIKE '%-0') ) and sg.renovacion=3 AND ob.estado LIKE'Activo')
GROUP BY sg.id_seguro, sg.renovacion, ct.codigo_cont, ctt.apellidos, ctt.nombres, ctt.cedula_ruc,
ob.descripcion_obra, cp.descripcion_comp, g.descripcion_gar, sg.referencia, c.descripcion_cant,
sg.formapago, sg.referencia
ORDER BY 1 asc

```

n. **FIREBIRD-** Sentencia SQL listar datos de complejidad simple

Es el mismo código del **literal m** con la diferencia que FIREBIRD no posee esquemas

o. **FIREBIRD-** Sentencia SQL insertar y actualizar

Es el mismo código del **literal a y literal b** con la diferencia que FIREBIRD solo en la sentencia de repetición el orden la estructura es inversa a la de PostgreSQL

ANEXO 2

SENTENCIAS SQL PARA LA PRUEBA DE CONSUMO DE RECURSOS

a. POSTGRESQL- Sentencia SQL de las cuatro operaciones básicas

SENTENCIAS SQL DE LAS CUATRO OPERACIONES BÁSICAS

```
begin
v = num_registros / 4; i=1;
while( i <= v ) loop

INSERT INTO gar.compania(ruc_compania, descripcion_comp, direccion_comp, telefono_comp)
VALUES ('060'||100000 +i || i % 10 || '001', 'Comp. seguros'||i, 'matrix' ||i, '(593-3)'|| 1000000+i);
i = i+1;
end loop;
i=1;
while( i <= num_registros ) loop
INSERT INTO gar.obra( descripcion_obra, presupuesto_ref_obra, codigo_cont,estado)
VALUES ('Obra_'||cast(i as varchar), 100.01 + i, 'Contrato_'||cast(i as varchar), 'Activo');
i = i+1;
end loop;
i=1;
while( i <= num_registros ) loop
if (i < 999999) then
INSERT INTO gar.contratista(cedula_ruc, nombres, apellidos, genero, direccion_cont, telefono_cont,
email)
VALUES ('060'||100000 +i || '-' || i % 10, 'nombre_' || i, 'apellido_' || i, 'M', 'direccion_' ||i, '(593-3)'||
1000000+i,'nombre_' || i || '@gmail.com' );
INSERT INTO gar.contrato(codigo_cont, plazo, fecha_inicio_cont,fecha_fin_cont, estado_cont,
descripcion_cont, cedula_contratista, ruc_municipio, id_parroquia, id_canton, id_provincia,
fecha_celebracion_cont)
VALUES ('Contrato_'||cast(i as varchar), i+2, 11111+i, 11111+i, 'Activo', 'principal obra', '060'||100000 +i
|| '-' || i % 10, '0260000680001', 50, 3, 2, 1111);
else
if (i < 999999999) then
INSERT INTO gar.contratista(cedula_ruc, nombres, apellidos, genero, direccion_cont,telefono_cont,
email)
VALUES (1000000000 +i || '-' || i % 10, 'nombre_' || i, 'apellido_' || i, 'F', 'direccion_' ||i, '(593-3)'||
1000000+i,'nombre_' || i || '@gmail.com' );
INSERT INTO gar.contrato(codigo_cont, plazo, fecha_inicio_cont,fecha_fin_cont, estado_cont,
descripcion_cont, cedula_contratista,
ruc_municipio, id_parroquia, id_canton, id_provincia, fecha_celebracion_cont)
VALUES ('Contrato_'||cast(i as varchar), i+2, 11111+i, 11111+i, 'Activo', 'principal obra', 1000000000 +i
|| '-' || i % 10, '0260000680001', 50, 3, 2, 1111);
else
INSERT INTO gar.contratista(cedula_ruc, nombres, apellidos, genero, direccion_cont,telefono_cont,
email)
VALUES (1000000000+i , 'nombre_' || i, 'apellido_' || i, 'M', 'direccion_' ||i, '(593-3)'||
1000000+i,'nombre_' || i || '@gmail.com' );
INSERT INTO gar.contrato(codigo_cont, plazo, fecha_inicio_cont,fecha_fin_cont, estado_cont,
descripcion_cont, cedula_contratista, ruc_municipio, id_parroquia, id_canton, id_provincia,
fecha_celebracion_cont)
VALUES ('Contrato_'||cast(i as varchar), i+2, 11111+i, 11111+i, 'Activo', 'principal obra', 1000000000+i
, '0260000680001', 50, 3, 2, 1111);
end if ;
end if;
i = i+1;
end loop;
i=1; y=1; r=0; p=25; estado = 'Terminado';
while( i <= num_registros ) loop
```

```

if( y <= v ) then
  INSERT INTO gar.seguro(codigo, renovacion, monto, formapago, fecha_inicio_s, fecha_fin_s,
vigencia, referencia, codigo_cont, id_garantia, id_formag, ruc_compania, id_parroquia, id_canton,
id_provincia, fecha_celebracion)
VALUES (i, r, 10000, p||'%', 111111, 11111, 20, estado, 'Contrato_'||cast(y as varchar), 1, 2, '060'||100000
+y || y % 10 || '001', 50, 3, 2, 111111);
else
y=0; r=r+1; p=p+25;
if(p>74) then
estado = 'Activo';
end if;
end if;
y= y+1; i = i+1;
end loop;
w = num_actualizar / 4;
j=1;
while( j <= w ) loop
UPDATE gar.compania
SET descripcion_comp='Actualizado Comp. seguros'||i
WHERE ruc_compania='060'||100000 +j || j % 10 || '001';
j = j+1;
end loop;
j=1;
while( j <= num_actualizar ) loop
UPDATE gar.obra
SET descripcion_obra='Actualizado Obra_'||cast(j as varchar)
WHERE codigo_cont='Contrato_'||cast(j as varchar);
UPDATE gar.contrato
SET descripcion_cont='Actualizar principal obra_'||cast(j as varchar)
WHERE codigo_cont='Contrato_'||cast(j as varchar);
UPDATE gar.seguro
SET id_garantia=2
WHERE codigo=j;
j = j+1;
end loop;
j=1;
while( j <= num_actualizar ) loop
if( j < 999999 ) then
UPDATE gar.contratista
SET nombres= 'Actualizado nombre_'|| j
WHERE cedula_ruc = '060'||100000 +j || '-' || j % 10;
else
if( j < 999999999 ) then
UPDATE gar.contratista
SET nombres= 'Actualizado nombre_'|| j
WHERE cedula_ruc = 100000000 +j || '-' || j % 10;
else
UPDATE gar.contratista
SET nombres= 'Actualizado nombre_'|| j
WHERE cedula_ruc = 1000000000+j;
end if ;
end if;
j = j+1;
end loop;
-- i=1;
--while( i <= num_registros ) loop
delete from gar.obra where descripcion_obra like '%1';
-- i = i+1;
--end loop;
return query

```

```

SELECT
ctt.nombres as Nombre,
ob.descripcion_obra as Obra,
ct.codigo_cont as Contrato,
ct.descripcion_cont as Des_contrado,
cp.descripcion_comp as Compania,
count(*) as Total
FROM gar.obra AS ob inner join gar.contrato AS ct on(ob.codigo_cont = ct.codigo_cont)
        inner join gar.seguro AS sg on (ct.codigo_cont = sg.codigo_cont)
        inner join gar.contratista AS ctt on (ctt.cedula_ruc = ct.cedula_contratista)
        inner join gar.compania As cp on (cp.ruc_compania = sg.ruc_compania)
WHERE (sg.ruc_compania in(select ruc_compania from gar.compania where ruc_compania
like'%1001') )
GROUP BY ctt.nombres, ob.descripcion_obra, ct.codigo_cont, ct.descripcion_cont,
cp.descripcion_comp, ob.id_obra
ORDER BY ob.id_obra ASC;

```

b. **Firebird**- sentencias SQL de las cuatro operaciones básicas

Es el mismo código del **literal k** con la diferencia que FIREBIRD no posee esquemas

ANEXO 2.1

SENTENCIAS SQL PARA EL PARÁMETRO ADICIONAL

- a. **POSTGRESQL-** sentencias SQL listar mediante el comando de comparación LIKE

LISTAR DATOS DE COMPLEJIDAD SIMPLE

```
SELECT
  ob.descripcion_obra as Obra,
  sg.formapago as forma_pago,
  ct.codigo_cont as Contrato,
  sg.renovacion as Renovacion
FROM gar.obra as ob inner join gar.contrato as ct on (ob.codigo_cont LIKE ct.codigo_cont)
inner join gar.seguro as sg on (sg.codigo_cont LIKE ct.codigo_cont)
GROUP BY ob.descripcion_obra, ct.codigo_cont, sg.formapago, sg.renovacion, sg.id_seguro
ORDER BY sg.id_seguro DESC, ct.codigo_cont;
```

- b. **FIREBIRD-** sentencias SQL listar mediante el comando de comparación LIKE

LISTAR DATOS DE COMPLEJIDAD SIMPLE

```
SELECT
  ob.descripcion_obra as Obra,
  sg.formapago as forma_pago,
  ct.codigo_cont as Contrato,
  sg.renovacion as Renovacion
FROM obra as ob inner join contrato as ct on (ob.codigo_cont LIKE ct.codigo_cont)
inner join seguro as sg on (sg.codigo_cont LIKE ct.codigo_cont)
GROUP BY ob.descripcion_obra, ct.codigo_cont, sg.formapago, sg.renovacion, sg.id_seguro
ORDER BY sg.id_seguro DESC, ct.codigo_cont;
```

ANEXO 3

LECTURAS REALIZADAS POR CADA INDICADOR

En las siguientes figuras se representan las lecturas tomadas por cada indicador de los parámetros de desempeños y consumo de recurso por cada escenario de prueba

a. **POSTGRESQL Y FIREBIRD** - Numero de lecturas realizas para el indicador ingreso de datos.

	p_i_1000	p_i_10000	p_i_100000	p_i_1000000	p_i_10000000	f_i_1000	f_i_10000	f_i_100000	f_i_1000000	f_i_10000000
1	4125	55753	410819	4174688	59435323	578	29141	346797	4093922	32939032
2	7563	31627	465293	4157732	53545765	640	29765	319016	4080141	32546754
3	8345	36658	403858	4149584	53514547	718	31359	318515	4066780	32664344
4	4438	40174	414307	4148453	54003122	781	29359	344172	4106539	32664320
5	6118	37440	415616	4155476	53943102	645	32009	324521	4108571	32654034
6	5432	44323	412254	4144232	53634343	658	30143	329705	4087678	32647898
7	4543	37443	413222	4146544	53636434	648	30543	328332	4100876	32657876
8	4366	40023	417304	4147665	53823443	661	31045	331544	4067898	32689434
9	4311	39232	414944	4149334	53535654	658	31054	330654	4074322	32682083
10	4422	39411	418233	4142122	53548558	651	30432	329528	4065478	32634726

b. **POSTGRESQL Y FIREBIRD** - Numero de lecturas realizas para el indicador actualizar de datos

	p_a_1000	p_a_10000	p_a_100000	p_a_1000000	p_a_10000000	f_a_1000	f_a_10000	f_a_100000	f_a_1000000	f_a_10000000
1	2735	23892	248407	2465939	24078895	1219	12953	133031	1253406	20463215
2	2078	23517	220129	2074154	23453229	906	9766	141125	1213361	20342331
3	2109	24611	221020	2001211	23443022	985	11898	150688	1232448	20323423
4	2172	25689	221032	1935939	23159243	1046	12043	147134	1223543	20381221
5	1906	23611	220282	1709876	23550204	1000	12643	166487	1224773	20392302
6	2105	23325	225643	1610650	23567865	1123	12079	164345	1234565	20377543
7	2074	23076	219356	1807657	23438797	1143	12043	164754	1239347	20399546
8	2056	24009	228743	1705435	23568678	1130	12034	160647	1233574	20323556
9	1995	23278	229067	1707864	23587453	1107	11945	165345	1236543	20334355
10	1925	23487	235643	1700323	23461962	1033	12044	147544	1226543	20325657
11										

c. **POSTGRESQL**- Numero de lecturas realizas para el indicador listar de datos de complejidad simple y media

	p_sshe_1000	p_sshe_10000	p_sshe_100000	p_sshe_1000000	p_sshe_10000000	p_smhe_1000	p_smhe_10000	p_smhe_100000	p_smhe_1000000	p_smhe_10000000
1	172	1000	6000	47000	3034000	765	1000	9000	133000	13902000
2	32	0	2000	39000	2982000	67	0	6000	70000	11891000
3	16	1000	3000	40000	2884000	47	0	6000	68000	12160000
4	47	0	2000	36000	2943000	79	1000	6000	99000	11980000
5	31	0	2000	38000	2899000	72	1000	7000	104000	11991000
6	45	1000	2000	36000	2964000	87	1000	6000	100000	12454000
7	45	0	3000	38000	2962000	85	1000	6000	80000	12043000
8	32	0	2000	36000	2895000	90	0	7000	85000	12943000
9	47	1000	2000	41000	2964000	87	1000	7000	102000	12549000
10	31	1000	3000	38000	2946000	87	1000	6000	78000	11965000
11										

d. **POSTGRESQL**- Numero de lecturas realizas para el indicador listar de datos de complejidad alta.

	p_sahe_1000	p_sahe_10000	p_sahe_100000	p_sahe_1000000	p_sahe_10000000	f
1	1688	2000	29000	407000	22838000	
2	203	1000	25000	267000	20743000	
3	218	1000	24000	281000	21026000	
4	219	0	20000	276000	21071000	
5	234	1000	23000	268000	20917000	
6	220	1000	24000	273000	21430000	
7	203	1000	24000	280000	21654000	
8	214	1000	25000	276000	20843000	
9	246	0	24000	279000	21045000	
10	230	1000	25000	274000	21074000	

e. **FIREBIRD**: Numero de lecturas realizas para el indicador listar de datos de complejidad simple y media.

	f_sshe_1000	f_sshe_10000	f_sshe_100000	f_sshe_1000000	f_sshe_10000000	f_smhe_1000	f_smhe_10000	f_smhe_100000	f_smhe_1000000	f_smhe_10000000	f
1	156	2000	15000	222000	3831000	175	2000	23000	363000	15970000	
2	62	1000	11000	174000	3726000	109	1000	22000	317000	14746000	
3	78	1000	13000	190000	3597000	125	2000	21000	336000	14785000	
4	62	1000	10000	230000	3782000	110	2000	22000	361000	14891000	
5	62	1000	11000	218000	3672000	109	3000	21000	345000	14701000	
6	65	1000	12000	200000	3630000	110	3000	22000	344000	14832000	
7	80	2000	11000	218000	3657000	118	2000	21000	350000	14843000	
8	65	1000	12000	194000	3704000	110	2000	22000	340000	14793000	
9	72	1000	12000	183000	3654000	115	3000	22000	342000	14854000	
10	80	1000	11000	197000	3684000	110	1000	21000	343000	14859000	
11											

f. **FIREBIRD**- Numero de lecturas realizas para el indicador listar de datos de complejidad alta.

	f_sahe_1000	f_sahe_10000	f_sahe_100000	f_sahe_1000000	f_sahe_10000000	f
1	188	3000	35000	743000	30036000	
2	172	2000	29000	612000	29412000	
3	157	2000	33000	643000	29477000	
4	172	3000	32000	578000	29553000	
5	156	2000	32000	598000	29518000	
6	178	2000	30000	601000	29538000	
7	169	3000	31000	632000	29454000	
8	168	2000	30000	648000	29563000	
9	171	2000	29000	653000	29576000	
10	175	3000	32000	597000	29573000	
11						

g. **POSTGRESQL**- Numero de lecturas realizas para el indicador búsqueda de datos de complejidad simple y media

	p_ssh_1000	p_ssh_10000	p_ssh_100000	p_ssh_1000000	p_ssh_10000000	p_smh_1000	p_smh_10000	p_smh_100000	p_smh_1000000	p_smh_10000000
1	359	1141	4741	22627	1621153	610	1540	6032	44190	1787248
2	16	110	1641	23220	1240630	31	203	5375	43190	1556449
3	32	109	2687	22158	1258723	47	250	4625	40080	1464106
4	31	125	1803	17548	1160232	31	234	3531	36111	1559904
5	31	110	3344	18079	1257551	32	274	2329	29987	1404493
6	31	123	2643	19435	1373554	32	272	2445	34453	1546576
7	32	122	2523	19343	1368533	47	274	2565	34654	1548554
8	16	122	2593	19034	1358540	32	275	2575	35544	1546543
9	31	119	2493	20530	1353648	31	276	2596	35943	1546677
10	31	120	2443	20067	1354346	31	270	2575	35965	1558537
11										

h. POSTGRESQL: Numero de lecturas realizas para el indicador búsqueda de datos de complejidad alta.

	p_sah_1000	p_sah_10000	p_sah_100000	p_sah_1000000	p_sah_10000000	f
1	234	1226	18021	190792	2503520	
2	172	1131	12893	161556	2380202	
3	156	1110	14391	175260	2341345	
4	172	1150	13798	150930	2361493	
5	140	1117	12611	145071	2345382	
6	150	1127	13784	154385	2354678	
7	155	1134	13886	158754	2354643	
8	149	1165	13866	158933	2356876	
9	157	1157	13658	158359	2376576	
10	156	1146	13654	158905	2378765	
11						

i. FIREBIRD: Numero de lecturas realizas para el indicador búsqueda de datos de complejidad simple y media

	f_ssh_1000	f_ssh_10000	f_ssh_100000	f_ssh_1000000	f_ssh_10000000	f_smh_1000	f_smh_10000	f_smh_100000	f_smh_1000000	f_smh_10000000
1	78	519	7010	43484	700354	63	400	7766	67594	1564321
2	31	187	1905	32063	643025	32	320	4469	96765	1304391
3	16	187	1969	26578	680501	31	328	3484	101516	1224531
4	15	188	1953	25641	689266	32	313	3469	89266	1281719
5	16	177	1869	26515	662469	31	328	3641	91672	1297015
6	16	167	2109	26435	659353	32	304	3465	88544	1274343
7	31	188	2065	25984	659456	31	305	3465	89644	1266435
8	15	179	2197	25991	653424	31	303	3506	89434	1276754
9	16	187	2003	26034	653235	32	304	3475	90349	1275755
10	31	176	2140	26933	664523	31	303	3497	90543	1275479
11										

j. FIREBIRD: Numero de lecturas realizas para el indicador buscar datos de complejidad alta.

	f_sah_1000	f_sah_10000	f_sah_100000	f_sah_1000000	f_sah_10000000
1	125	1062	17500	222157	3053906
2	94	969	10828	209172	3001407
3	78	953	13031	191578	2795531
4	93	968	10932	210329	2950484
5	94	984	10109	180250	2898016
6	95	964	10435	193657	2843665
7	87	996	11046	194654	2854678
8	82	948	11065	194654	2857544
9	93	921	10765	195657	2856445
10	81	954	10656	195565	2857656
11					

k. **POSTGRESQL**- Numero de lecturas realizas para el indicador usos de memoria RAM y procesador.

	p_ram_1000	p_ram_10000	p_ram_100000	p_ram_1000000	p_ram_10000000	p_p_1000	p_p_10000	p_p_100000	p_p_1000000	p_p_10000000
1	3,6	5,4	19,7	164,1	203,1	5,1	10,1	10,1	9,9	10,4
2	3,7	5,2	19,4	159,3	205,2	6,9	9,9	10,9	10,4	10,0
3	3,4	4,8	19,5	160,0	190,9	7,4	8,4	8,9	10,5	10,1
4	3,5	5,0	18,8	158,7	205,5	6,8	10,1	10,1	10,7	10,6
5	3,4	3,1	19,5	160,1	200,2	5,1	10,2	10,3	10,5	11,2
6	3,4	5,2	19,4	160,1	200,2	6,5	10,1	10,3	10,6	11,1
7	3,5	5,2	19,5	159,3	200,1	6,6	10,2	10,0	10,5	10,5
8	3,5	5,1	19,5	159,6	200,0	6,7	9,9	10,3	10,7	10,6
9	3,6	4,9	19,4	159,3	198,0	6,6	10,1	10,3	10,6	10,5
10	3,5	5,0	18,9	160,0	199,3	6,6	10,1	10,3	10,6	10,5
11										

l. **FIREBIRD**: Numero de lecturas realizas para el indicador usos de memoria RAM y procesador

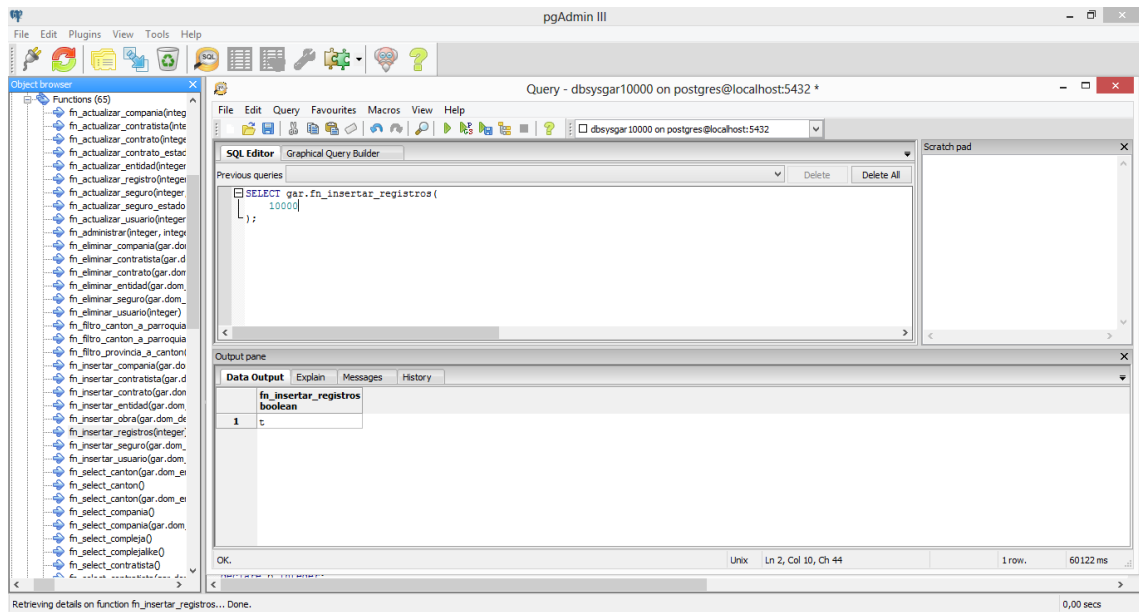
	f_ram_1000	f_ram_10000	f_ram_100000	f_ram_1000000	f_ram_10000000	f_p_1000	f_p_10000	f_p_100000	f_p_1000000	f_p_10000000...
1	6,9	16,6	63,5	106,7	190,2	10,4	9,6	10,6	10,7	11,3
2	8,2	17,9	63,0	102,4	194,5	7,1	10,4	10,3	10,2	10,1
3	6,5	16,9	55,1	99,8	200,0	7,8	10,1	10,4	11,1	11,4
4	6,3	17,5	60,3	104,5	204,3	8,6	10,2	10,0	9,9	9,9
5	6,6	17,8	62,7	106,3	190,4	10,1	10,1	10,2	10,0	10,6
6	6,5	16,8	63,2	104,1	198,2	8,5	10,4	10,5	11,0	11,5
7	6,4	17,4	60,2	102,5	200,0	8,4	10,6	10,3	10,3	10,3
8	6,6	17,1	60,3	103,2	200,3	8,4	10,3	10,6	10,5	10,3
9	7,1	17,4	58,4	101,1	199,2	8,6	9,8	10,1	11,0	10,2
10	7,2	16,9	58,7	102,2	200,3	9,2	9,9	10,0	10,2	10,1

ANEXO 4

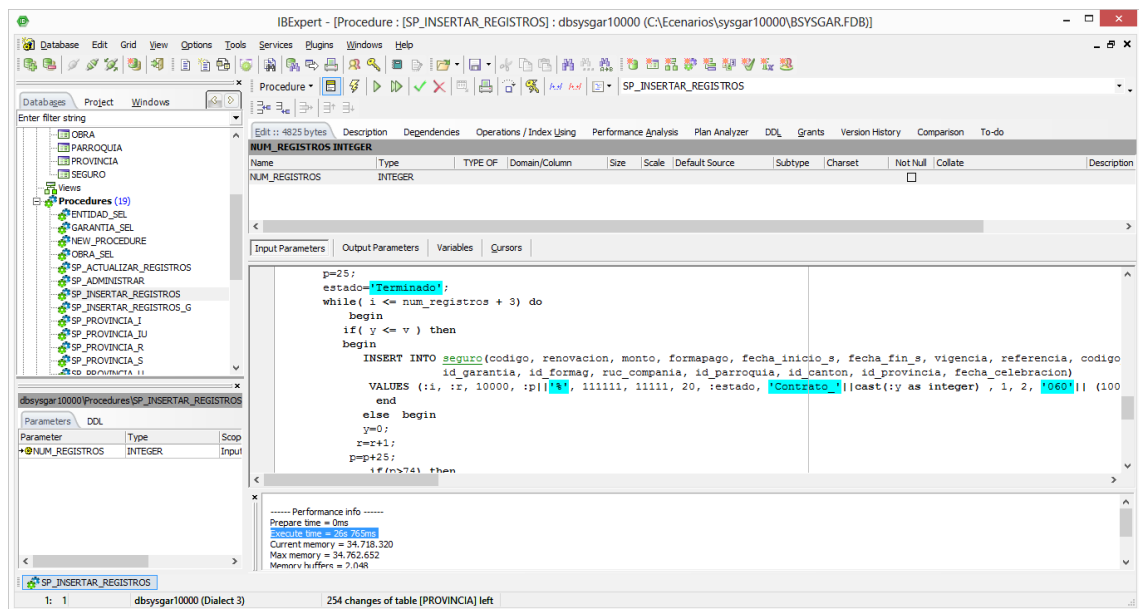
INTERFACES DE LAS PRUEBAS EN LOS SGBD

En las siguientes figuras se representa un ejemplar de las 10 lecturas que se tomaron por cada indicador de los parámetros desempeño y consumo de recurso en los 5 escenarios de prueba, utilizando las herramientas PGADMIN III para PostgreSQL e IBEXPERT para Firebird.

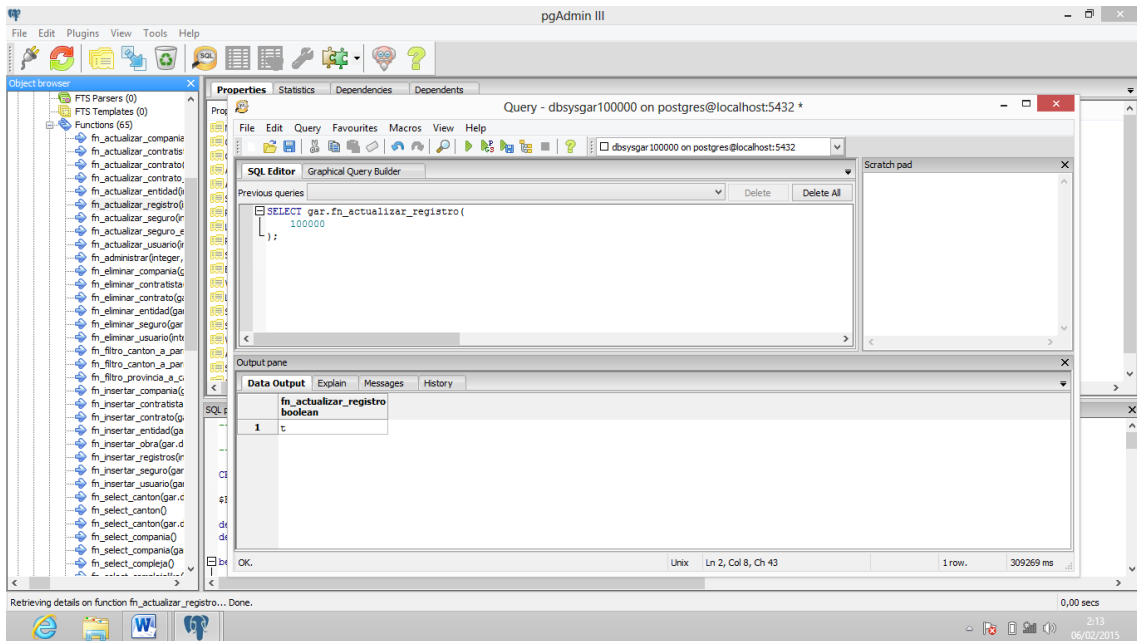
a. POSTGRESQL - Tiempo de repuesta en insertar datos



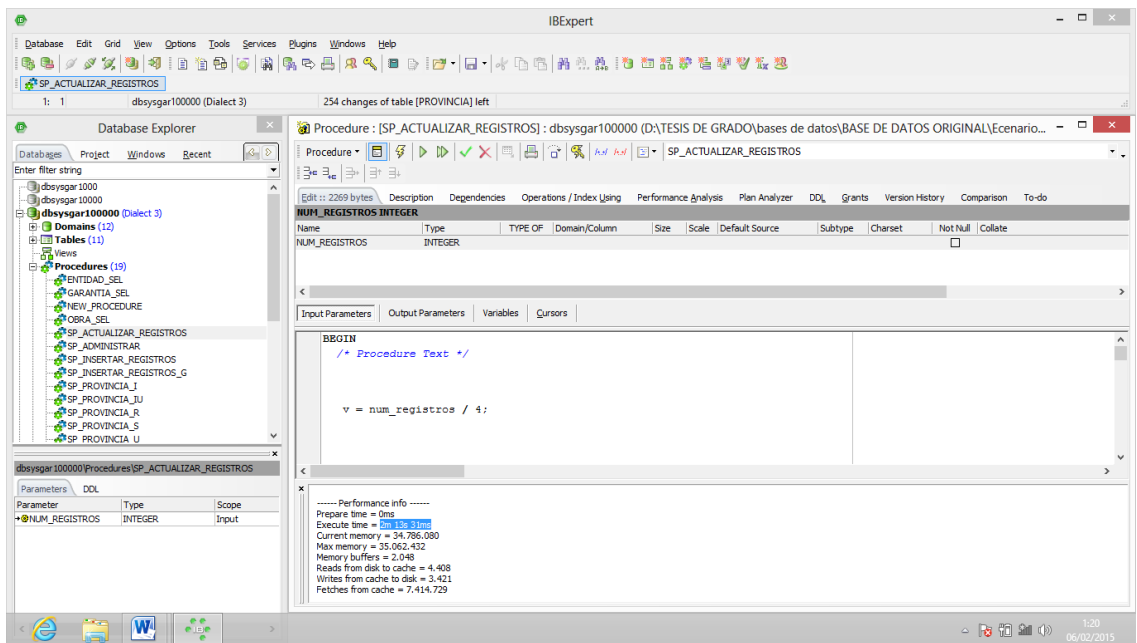
b. FIREBIRD - Tiempo de repuesta en insertar datos



c. POSTGRESQL- Tiempo de repuesta en actualizar datos



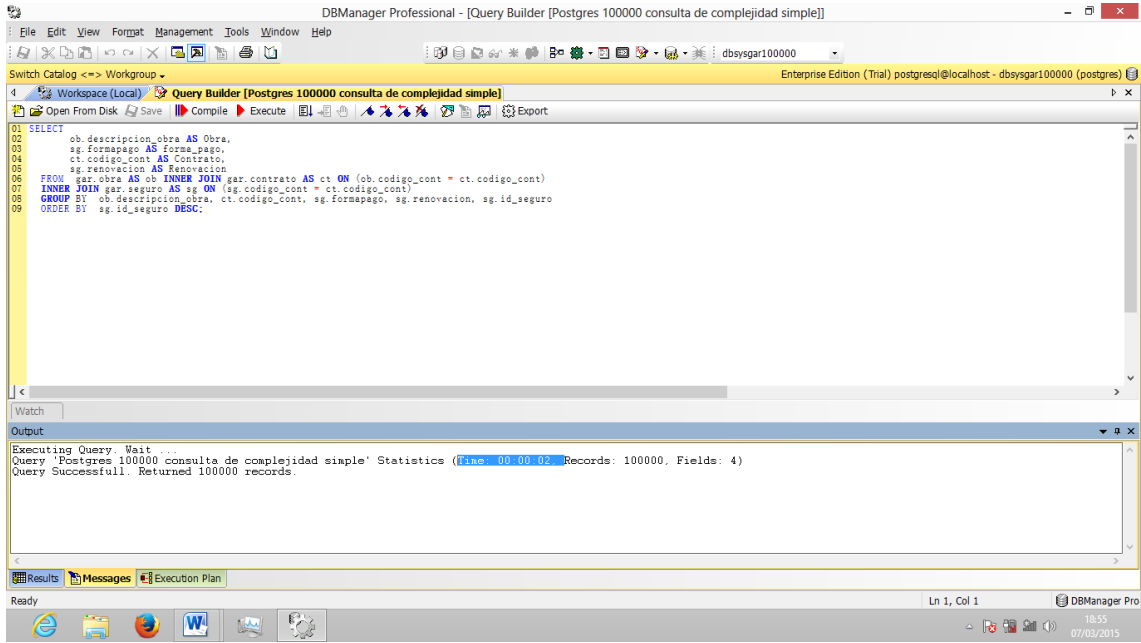
d. FIREBIRD -Tiempo de repuesta en actualizar datos



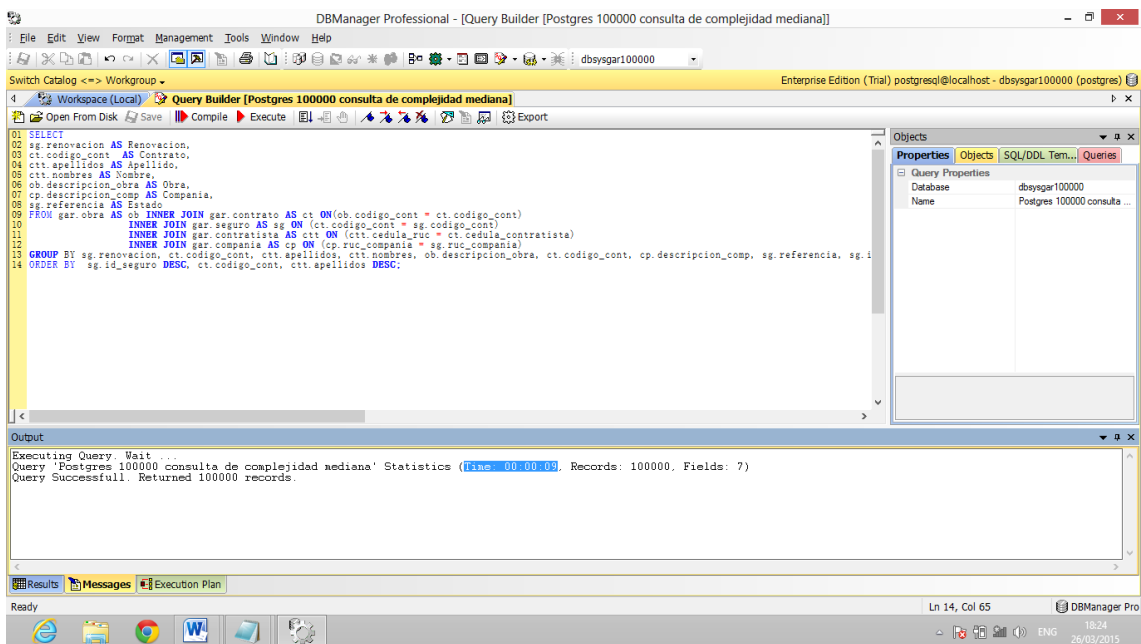
Para el indicar LISTAR DATOS se utilizó DBMANAGER PROFESSIONAL el cual controla los dos SGBD POSTGRESQL Y FIREBIRD

e. **POSTGRESQL-** Tiempo de respuesta en listar

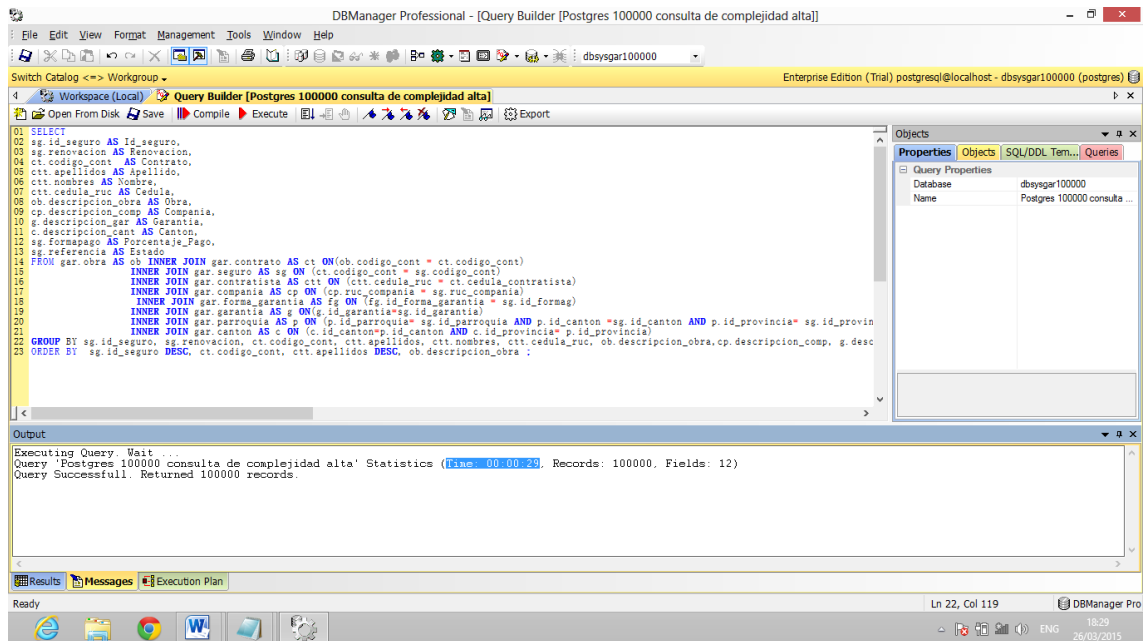
- **Complejidad simple**



- **Complejidad media**

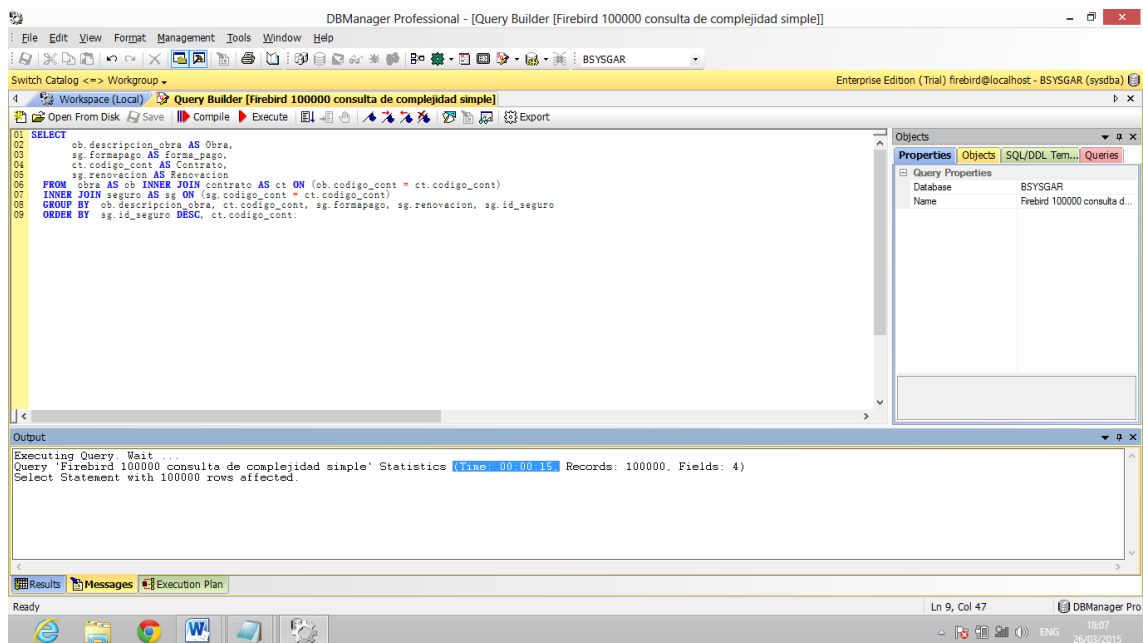


- **Complejidad alta**



f. **FIREBIRD-** Tiempo de repuesta en listar datos

- **Complejidad simple**



• Complejidad media

The screenshot shows the DBManager Professional interface with a SQL query editor. The query is a SELECT statement with several tables and complex joins. The output window shows the execution statistics: Time: 00:00:25, Records: 100000, Fields: 7.

```
01 SELECT
02 sg.renovacion AS Renovacion,
03 ct.codigo_cont AS Contrato,
04 ctt.apellidos AS Apellido,
05 ctt.nombres AS Nombres,
06 ob.descripcion_obra AS Obra,
07 cp.descripcion_comp AS Compania,
08 sg.referencia AS Estado
09 FROM obra AS ob INNER JOIN contrato AS ct ON (ob.codigo_cont = ct.codigo_cont)
10 INNER JOIN seguro AS sg ON (ct.codigo_cont = sg.codigo_cont)
11 INNER JOIN contratista AS ctt ON (ctt.cedula_ruc = ct.cedula_contratista)
12 INNER JOIN compania AS cp ON (cp.ruc_compania = sg.ruc_compania)
13 GROUP BY sg.renovacion, ct.codigo_cont, ctt.apellidos, ctt.nombres, ob.descripcion_obra, ct.descripcion_comp, sg.referencia, sg.i
14 ORDER BY sg.id_seguro DESC, ct.codigo_cont, ctt.apellidos DESC;
```

Output: Executing Query. Wait ...
Query 'Firebird 100000 consulta de complejidad mediana' Statistics (Time: 00:00:25) Records: 100000, Fields: 7)
Select Statement with 100000 rows affected.

g. Complejidad alta

The screenshot shows the DBManager Professional interface with a SQL query editor. The query is a SELECT statement with many tables and complex joins, including geographic information. The output window shows the execution statistics: Time: 00:00:33, Records: 100000, Fields: 12.

```
01 SELECT
02 sg.id_seguro AS Id_seguro,
03 sg.renovacion AS Renovacion,
04 ct.codigo_cont AS Contrato,
05 ctt.apellidos AS Apellido,
06 ctt.nombres AS Nombres,
07 ctt.cedula_ruc AS Cedula,
08 ob.descripcion_obra AS Obra,
09 cp.descripcion_comp AS Compania,
10 g.descripcion_gar AS Garantia,
11 c.descripcion_cant AS Canton,
12 sg.formapago AS Porcentaje_Pago,
13 sg.referencia AS Estado
14 FROM obra AS ob INNER JOIN contrato AS ct ON (ob.codigo_cont = ct.codigo_cont)
15 INNER JOIN seguro AS sg ON (ct.codigo_cont = sg.codigo_cont)
16 INNER JOIN contratista AS ctt ON (ctt.cedula_ruc = ct.cedula_contratista)
17 INNER JOIN compania AS cp ON (cp.ruc_compania = sg.ruc_compania)
18 INNER JOIN forma_garantia AS fg ON (fg.id_forma_garantia = sg.id_forma_gar)
19 INNER JOIN garantia AS g ON (g.id_garantia = fg.id_garantia)
20 INNER JOIN parroquia AS p ON (p.id_parroquia = sg.id_parroquia AND p.id_canton = sg.id_canton AND p.id_provincia = sg.id_provincia)
21 INNER JOIN canton AS c ON (c.id_canton = p.id_canton AND c.id_provincia = p.id_provincia)
22 GROUP BY sg.id_seguro, sg.renovacion, ct.codigo_cont, ctt.apellidos, ctt.nombres, ctt.cedula_ruc, ob.descripcion_obra, cp.descripcion_comp, g.de
23 ORDER BY sg.id_seguro DESC, ct.codigo_cont, ctt.apellidos DESC, ob.descripcion_obra ;
```

Output: Executing Query. Wait ...
Query 'Firebird 100000 consulta de complejidad alta' Statistics (Time: 00:00:33) Records: 100000, Fields: 12)
Select Statement with 100000 rows affected.

h. POSTGRESQL Tiempo de repuesta buscar datos

- **Complejidad simple**

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

```
SELECT
  ob.descripcion_obra AS Obra,
  sg.formapago AS forma_pago,
  ct.codigo_cont AS Contrato,
  sg.renovacion AS Renovacion
FROM gar.obra AS ob INNER JOIN gar.contrato AS ct ON (ob.codigo_cont = ct.codigo_cont)
INNER JOIN gar.seguro AS sg ON (sg.codigo_cont = ct.codigo_cont)
WHERE (sg.renovacion = 2)
GROUP BY ob.descripcion_obra, ct.codigo_cont, sg.formapago, sg.renovacion, sg.id_seguro
ORDER BY sg.id_seguro DESC, ct.codigo_cont;
```

The results panel shows the following data:

obra	forma_pago	contrato	renovacion
character varying(100)	character varying(100)	character varying(22)	integer
1 Obra 25000	75%	Contrato 25000	2
2 Obra 24999	75%	Contrato 24999	2
3 Obra 24998	75%	Contrato 24998	2
4 Obra 24997	75%	Contrato 24997	2
5 Obra 24996	75%	Contrato 24996	2
6 Obra 24995	75%	Contrato 24995	2

- **Complejidad media**

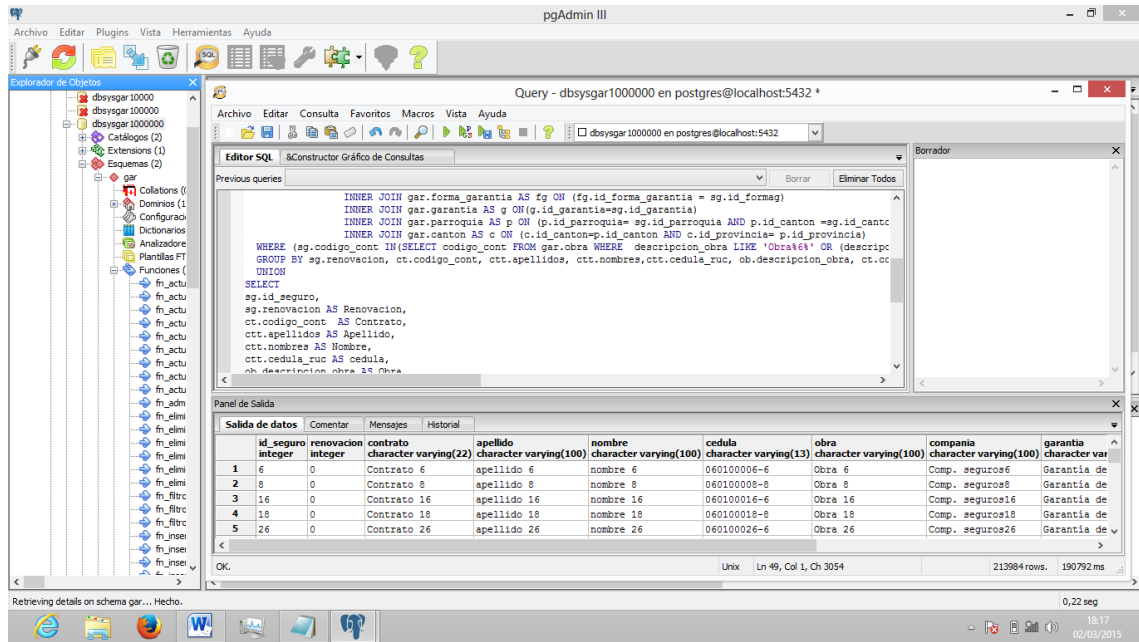
The screenshot shows the pgAdmin III interface with a more complex SQL query:

```
SELECT
  sg.renovacion AS Renovacion,
  ct.codigo_cont AS Contrato,
  ctt.apellidos AS Apellido,
  ctt.nombre AS Nombre,
  ob.descripcion_obra AS Obra,
  cp.descripcion_comp AS Compania,
  sg.referencia AS Estado
FROM gar.obra AS ob INNER JOIN gar.contrato AS ct ON (ob.codigo_cont = ct.codigo_cont)
INNER JOIN gar.seguro AS sg ON (ct.codigo_cont = sg.codigo_cont)
INNER JOIN gar.contratista AS ctt ON (ct.cedula_ruc = ct.cedula_contratista)
INNER JOIN gar.compania AS cp ON (cp.ruc_compania = sg.ruc_compania)
WHERE (ct.cedula_contratista IN (SELECT cedula_ruc FROM gar.bontratista WHERE apellidos LIKE 'apellido%' OR ape
GROUP BY sg.renovacion, ct.codigo_cont, ctt.apellidos, ctt.nombre, ob.descripcion_obra, ct.codigo_cont, cp.des
ORDER BY sg.id_seguro DESC, ct.codigo_cont, ctt.apellidos DESC;
```

The results panel shows the following data:

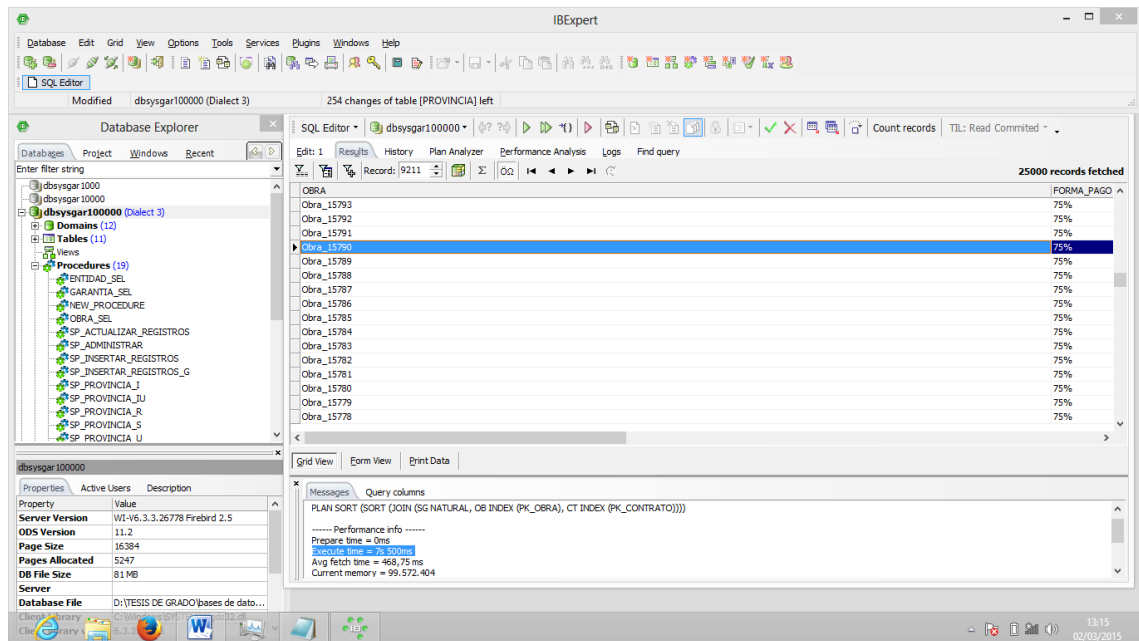
renovacion	contrato	apellido	nombre	obra	compania	estado
integer	character varying(22)	character varying(100)	character varying(100)	character varying(100)	character varying(100)	character varying(100)
1	0	Contrato 24994	apellido 24994	nombre 24994	Obra 24994	Comp. seguros24994 Terminado
2	0	Contrato 24991	apellido 24991	nombre 24991	Obra 24991	Comp. seguros24991 Terminado
3	0	Contrato 24984	apellido 24984	nombre 24984	Obra 24984	Comp. seguros24984 Terminado
4	0	Contrato 24981	apellido 24981	nombre 24981	Obra 24981	Comp. seguros24981 Terminado

- **Complejidad alta**



- i. **FIREBIRD -Tiempo de repuesta en buscar datos**

- **Complejidad simple**



- Complejidad media

The screenshot shows the IBM DB2 SQL Editor interface. The main window displays a complex SQL query with multiple joins and a performance plan. The query is as follows:

```

SELECT
  sg.renovacion AS Renovacion,
  ct.codigo_cont AS Contrato,
  ctt.apellidos AS Apellido,
  ctt.nombres AS Nombre,
  ob.descripcion obra AS Obra,
  cp.descripcion_comp AS Compania,
  sg.referencia AS Estado
FROM obra AS ob INNER JOIN contrato AS ct ON (ob.codigo_cont = ct.codigo_cont)
  INNER JOIN seguro AS sg ON (ct.codigo_cont = sg.codigo_cont)
  INNER JOIN contratista AS ctt ON (ctt.cedula_ruc = ct.cedula_contratista)
  INNER JOIN compania AS cp ON (cp.ruc_compania = sg.ruc_compania)
WHERE (ct.cedula_contratista IN (SELECT cedula_ruc FROM contratista WHERE apellidos LIKE 'Apellido'))
GROUP BY sg.renovacion, ct.codigo_cont, ctt.apellidos, ctt.nombres, ob.descripcion_obra, ct.codigo_cont
ORDER BY sg.id_seguro DESC, ct.codigo_cont , ctt.apellidos DESC;
  
```

The performance plan shows the following details:

```

PLAN SORT (SORT (JOIN (SG NATURAL, CT INDEX (PK_CONTRATO), CP INDEX (PK_COMPANIA), OB INDEX (PK_OBRA), CTT INDEX (PK_CONTRATISTA))))
----- Performance info -----
Prepare time = 0ms
Execute time = 1m 41s 516ms
Avg fetch time = 6.767,73 ms
Current memory = 103.896.356
  
```

- Complejidad alta

The screenshot shows the IBM DB2 SQL Editor interface displaying a table with 213984 records fetched. The table has the following columns:

ID_SEGURO	RENOVACION	CONTRATO	APELLIDO
1.000.000	3	Contrato_250000	apellido_250000
1.000.000	3	Contrato_250000	apellido_250000
1.000.000	3	Contrato_250000	apellido_250000
1.000.000	3	Contrato_250000	apellido_250000
999.930	3	Contrato_249930	apellido_249930
999.939	3	Contrato_249939	apellido_249939
999.940	3	Contrato_249940	apellido_249940
999.949	3	Contrato_249949	apellido_249949
999.950	3	Contrato_249950	apellido_249950
999.959	3	Contrato_249959	apellido_249959
999.960	3	Contrato_249960	apellido_249960
999.969	3	Contrato_249969	apellido_249969
999.970	3	Contrato_249970	apellido_249970
999.979	3	Contrato_249979	apellido_249979
999.980	3	Contrato_249980	apellido_249980

The performance plan shows the following details:

```

(PK_FORMA_GARANTIA), P INDEX (PK_PARROQUIA), C INDEX (PK_CANTON))
----- Performance info -----
Prepare time = 16ms
Execute time = 3m 28.106ms
Avg fetch time = 12.140,60 ms
Current memory = 48.329.972
  
```

j. **POSTGRESQL-** Usos de la memoria RAM (MB) y Procesador

The screenshot shows a Windows desktop environment. On the left, there is a taskbar with various application icons. The main window is a PostgreSQL query editor titled "Query - dbssysgarcarga100000 on postgres@localhost:5432". The query editor displays a SQL query: `SELECT garc_fn_administrar(100000, 100000);`. Below the query editor, the "Output pane" is visible. In the foreground, the Windows Task Manager is open, showing the "Administrador de tareas" window. The "Procesos" tab is selected, and the "Detalles" column is expanded. The following table shows the resource usage of various processes:

Nombre	Estado	CPU	Memoria	Disco	Red
Microsoft Word		10%	37%	49%	0%
Explorador de Windows		0%	23,4 MB	0 MB/s	0 MBps
PostgreSQL Server		10,1%	19,7 MB	0 MB/s	0 MBps
Administrador de ventanas del ...		0,2%	13,9 MB	0 MB/s	0 MBps

The task manager also shows system-wide statistics: CPU Usage - 26%, CPU Speed - 2.400,013 MHz, and Memory Usage - 1,197 GB.

k. **FIREBIRD-** usos de la memoria RAM (MB) y Procesador

The screenshot shows the IBExpert software interface. The main window is titled "Procedure : [SP_ADMINISTRAR] : dbssysgarcarga100000 (D:\TESIS DE GRADO\bases de datos\BASE DE DATOS ORIGINAL\Ecenarios\sys...". The procedure editor displays a PL/pgSQL procedure: `begin v = num_registros / 4; i=1; while(i <= v) begin insert into ... values ...; i=i+1; end`. In the foreground, the Windows Task Manager is open, showing the "Administrador de tareas" window. The "Procesos" tab is selected, and the "Detalles" column is expanded. The following table shows the resource usage of various processes:

Nombre	Estado	CPU	Memoria	Disco	Red
ESET Service		10%	39%	0%	0%
Firebird SQL Server		10,5%	44,5 MB	0 MB/s	0 MBps
Microsoft Word		0%	32,5 MB	0 MB/s	0 MBps
Host de servicio: Sistema local (...)		0%	25,8 MB	0 MB/s	0 MBps

The task manager also shows system-wide statistics: CPU Usage - 25%, CPU Speed - 2.400,011 MHz, and Memory Usage - 1,174 GB.

ANEXO 5
HISTORIAS DE USUARIOS

HISTORIA DE USUARIO	
Numero: 1	Usuario: Administrador (secretario GAD)
Nombre de la Historia: Autenticación de los usuarios.	Iteración Asignada: 1
Prioridad en Negocio: Alto	Riesgo en desarrollo: Alto
Programador Responsable: Víctor Colcha	
Descripción: El secretario de GAD de Cantón Chimbo almacena ingresa al sistema mediante su cuenta de administrador y clave	
Observaciones: el administrador proporciona nos permisos de acceso a los demás usuarios	

HISTORIA DE USUARIO	
Numero: 2	Usuario: Administrador (secretario GAD)
Nombre de la Historia: Almacena, edita y elimina los datos del usuario.	Iteración Asignada: 1
Prioridad en Negocio: Medio	Riesgo en desarrollo: Medio
Programador Responsable: Víctor Colcha	
Descripción: El secretario de GAD de Cantón Chimbo tiene todos los permisos para registrar, editar y eliminar los datos para cada usuario del sistema	
Observaciones: Crea todos los tipos de usuarios	

HISTORIA DE USUARIO	
Numero: 3	Usuario: Administrador (secretario GAD)
Nombre de la Historia: Almacena, edita y elimina los datos del contratista.	Iteración Asignada: 2
Prioridad en Negocio: Alto	Riesgo en desarrollo: Medio
Programador Responsable: Víctor Colcha	
Descripción: El secretario de GAD de Cantón Chimbo tiene todos los permisos para registrar, editar y eliminar los datos para cada contratista del sistema.	
Observaciones:	

HISTORIA DE USUARIO	
Numero: 4	Usuario: Administrador (secretario GAD)
Nombre de la Historia: Almacena, edita y suspender los datos del contrato	Iteración Asignada: 2
Prioridad en Negocio: Alto	Riesgo en desarrollo: Medio
Programador Responsable: Víctor Colcha	
Descripción: El secretario de GAD de Cantón Chimbo tiene todos los permisos para registrar, editar y cambiar de estado de los datos del contrato de obras.	
Observaciones: Requerimiento esencial	

HISTORIA DE USUARIO	
Numero: 5	Usuario: Administrador (secretario GAD)
Nombre de la Historia: Almacena, edita y elimina los datos de la compañía de seguros.	Iteración Asignada: 7
Prioridad en Negocio: Medio	Riesgo en desarrollo: Medio

Programador Responsable: Víctor Colcha
Descripción: El secretario de GAD de Cantón Chimbo tiene todos los permisos para registrar, editar y eliminar los datos de la compañía de seguros
Observaciones:

HISTORIA DE USUARIO	
Numero: 6	Usuario: Administrador (secretario GAD)
Nombre de la Historia: Almacenar la información de la garantía de la obra	Iteración Asignada: 3
Prioridad en Negocio: Alto	Riesgo en desarrollo: Alto
Programador Responsable: Víctor Colcha	
Descripción: El secretario de GAD de Cantón Chimbo almacena la información de la garantía que se realiza por cada contrato de la obra.	
Observaciones: Todo contrato de obra debe tener una garantía	

HISTORIA DE USUARIO	
Numero: 8	Usuario: Administrador (secretario GAD)
Nombre de la Historia: Alerta de caducidad de las garantías.	Iteración Asignada: 4
Prioridad en Negocio: Alto	Riesgo en desarrollo: Alto
Programador Responsable: Víctor Colcha	
Descripción: El secretario de GAD de Cantón Chimbo ingresa al sistema SYSGAR donde se muestra una luz verde en el formulario de consulta alertando la caducidad de garantías	

Observaciones: Las garantías se basan de acuerdo al artículo sobre Garantías

HISTORIA DE USUARIO

Numero: 9

Usuario: Administrador (secretario GAD)

Nombre de la Historia: visualiza las pólizas vencidas y por vencer

Iteración Asignada: 4

Prioridad en Negocio: Alto

Riesgo en desarrollo: Alto

Programador Responsable: Víctor Colcha

Descripción: El secretario de GAD de Cantón Chimbo ingresa al sistema SYSGAR donde este visualiza las pólizas vencidas y por vencer de acuerdo al tiempo que dicta el artículo sobre garantías

Observaciones: Las garantía hace referencia a la póliza ya que es un documento principal para los contratos de obras, tomando en cuenta que si se puede utilizar otro medio de garantía, estos detalles se encuentran en el artículo de la ley de garantías.

HISTORIA DE USUARIO

Numero: 10

Usuario: Administrador (secretario GAD)

Nombre de la Historia: Envío de la notificación del vencimiento de la garantía mediante un correo electrónico al contratista.

Iteración Asignada: 5

Prioridad en Negocio: Alto

Riesgo en desarrollo: Alto

Programador Responsable: Víctor Colcha

Descripción: El secretario de GAD de Cantón Chimbo notifica al contratista el vencimiento de la garantía mediante un correo que permite el sistema.

Observaciones: Notifica para que se acerque a renovar la garantía.

HISTORIA DE USUARIO	
Numero: 11	Usuario: Administrador (secretario GAD)
Nombre de la Historia: Estructura automática de los datos de la nueva renovación de la garantía para su posterior envió.	Iteración Asignada: 4
Prioridad en Negocio: Alto	Riesgo en desarrollo: Alto
Programador Responsable: Víctor Colcha	
Descripción: El secretario de GAD de Cantón Chimbo ingresa al sistema, el cual en las opciones de alerta puede visualizar los datos de la nueva renovación de la garantía creadas automáticamente en un archivo con formato pdf.	
Observaciones: El archivo pdf es creado internamente por sistema solo de las garantías por caducarse	

HISTORIA DE USUARIO	
Numero: 12	Usuario: Administrador (secretario GAD)
Nombre de la Historia: Envío de la notificación y datos adjuntos de una nueva renovación de la garantía mediante un correo electrónico a la compañía de seguros.	Iteración Asignada: 5
Prioridad en Negocio: Alto	Riesgo en desarrollo: Alto
Programador Responsable: Víctor Colcha	
Descripción: El secretario de GAD de Cantón Chimbo notifica con un archivo adjunto a la compañía de seguros la renovación de una nueva garantía por medio de un correo electrónico que permite el sistema	
Observaciones:	

HISTORIA DE USUARIO	
Numero: 13	Usuario: Administrador (secretario GAD)
Nombre de la Historia: Lista de los datos de los contratistas, usuarios y compañía mediante su identificador.	Iteración Asignada: 7
Prioridad en Negocio: Bajo	Riesgo en desarrollo: Bajo
Programador Responsable: Víctor Colcha	
Descripción: El secretario de GAD de Cantón Chimbo ingresa al sistema, donde en cada interfaz del contratista, usuario y compañía posee la opción de buscar por su identificación y lista los datos respectivos.	
Observaciones:	

HISTORIA DE USUARIO	
Numero: 14	Usuario: Administrador (secretario GAD), contratista y usuario dependiente.
Nombre de la Historia: Consulta sobre la obra y contrato que maneja la institución.	Iteración Asignada: 6
Prioridad en Negocio: Media	Riesgo en desarrollo: Alto
Programador Responsable: Víctor Colcha	
Descripción: Con cualquier perfil de usuario que ingrese al sistema le va a permitir poder consultar los datos del contrato de la obra.	
Observaciones: No es necesario que el contrato este vigente	

HISTORIA DE USUARIO	
Numero: 15	Usuario: Administrador (secretario GAD), contratista y usuario dependiente.
Nombre de la Historia: Consulta sobre la garantía del contrato de la obra.	Iteración Asignada: 6
Prioridad en Negocio: Media	Riesgo en desarrollo: Alto
Programador Responsable: Víctor Colcha	
Descripción: Con cualquier perfil de usuario que ingrese al sistema le va a permitir consultar los datos de la garantía del contrato de la obra.	
Observaciones: El contrato debe estar vigente	

ANEXO 6

DICcionario DE DATOS

Tabla **compañía**

Field Name	Field Type	Key	Not Null	Default	Description
ruc_compania	gar.dom_cedula_ruc (varchar(13))	Primary Key	<input checked="" type="checkbox"/>		pk_compania
id_compania	serial		<input checked="" type="checkbox"/>	nextval('gar.compania_id_c')	
descripcion_comp	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		
direccion_comp	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		
telefono_comp	gar.dom_telefono (varchar(20))		<input checked="" type="checkbox"/>		

Tabla **provincia**

Field Name	Field Type	Key	Not Null	Default	Description
id_provincia	gar.dom_entero_codigo (integer)	Primary Key	<input checked="" type="checkbox"/>		pk_provincia
descripcion_prov	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		

Tabla **canton**

Field Name	Field Type	Key	Not Null	Default	Description
id_canton	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		pk
id_provincia	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		pk;fk
descripcion_cant	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		

Tabla **parroquia**

Field Name	Field Type	Key	Not Null	Default	Description
id_parroquia	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		pk
descripcion_parr	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		
id_canton	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		pk ; fk
id_provincia	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		pk ; fk

Tabla **contratista**

Field Name	Field Type	Key	Not Null	Default	Description
cedula_ruc	gar.dom_cedula_ruc (varchar(13))	Primary Key	<input checked="" type="checkbox"/>		pk_contratista
id_contratista	serial		<input checked="" type="checkbox"/>	nextval('gar.contratista_id_c')	
nombres	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		
apellidos	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		
genero	gar.dom_genero (char(1))		<input checked="" type="checkbox"/>		
direccion_cont	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		
telefono_cont	gar.dom_telefono (varchar(20))		<input checked="" type="checkbox"/>		
email	gar.dom_email (varchar(50))		<input checked="" type="checkbox"/>		

Tabla **entidad(municipio)**

Field Name	Field Type	Key	Not Null	Default	Description
ruc_municipio	gar.dom_cedula_ruc (varchar(13))	Primary Key	<input checked="" type="checkbox"/>		pk_municipio
id_entidad	serial		<input checked="" type="checkbox"/>	nextval('gar.entidad_id_entk')	
descripcion_mun	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		
direccion_mun	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		
telefono_mun	gar.dom_telefono (varchar(20))		<input checked="" type="checkbox"/>		

Tabla **formagarantia**

Field Name	Field Type	Key	Not Null	Default	Description
id_forma_garantia	serial	Primary Key	<input checked="" type="checkbox"/>	nextval('gar.forma_garantia')	pk_forma_garantia
descripcion_fg	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		

Tabla garantia

Field Name	Field Type	Key	Not Null	Default	Description
id_garantia	serial	Primary Key	<input checked="" type="checkbox"/>	nextval('gar.garantia_id_gar')	pk_garantia
descripcion_gar	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		

Tabla seguro

Field Name	Field Type	Key	Not Null	Default	Description
codigo	gar.dom_entero_codigo (integer)	Primary Key	<input checked="" type="checkbox"/>		pk_seguro
id_seguro	serial		<input checked="" type="checkbox"/>	nextval('gar.seguro_id_seguro')	
renovacion	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		
monto	gar.dom_saldo (double precision)		<input checked="" type="checkbox"/>		
formapago	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		
fecha_inicio_s	gar.dom_fecha (bigint)		<input checked="" type="checkbox"/>		
fecha_fin_s	gar.dom_fecha (bigint)		<input checked="" type="checkbox"/>		
vigencia	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		
referencia	gar.dom_descripcion (varchar(100))		<input type="checkbox"/>	Null	
codigo_cont	gar.dom_contrato (varchar(22))		<input checked="" type="checkbox"/>		fk_contrato
id_garantia	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		fk_garantia
id_formag	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		fk_forma_garantia
ruc_compania	gar.dom_cedula_ruc (varchar(13))		<input checked="" type="checkbox"/>		fk_compania
id_parroquia	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		fk_parroquia
id_canton	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		fk_canton
id_provincia	gar.dom_entero_codigo (integer)		<input checked="" type="checkbox"/>		fk_provincia
fecha_celebracion	gar.dom_fecha (bigint)		<input type="checkbox"/>	Null	

Tabla obra

Field Name	Field Type	Key	Not Null	Default	Description
codigo_cont	gar.dom_contrato (varchar(22))	Primary Key	<input checked="" type="checkbox"/>		pk_obra
id_obra	serial		<input checked="" type="checkbox"/>	nextval('gar.obra_id_obra_s')	
descripcion_obra	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		
presupuesto_ref_obra	gar.dom_saldo (double precision)		<input checked="" type="checkbox"/>		
estado	gar.dom_estado (varchar(10))		<input type="checkbox"/>	Null	

Tabla usuario

Field Name	Field Type	Key	Not Null	Default	Description
cedula	gar.dom_cedula_ruc (varchar(13))	Unique	<input type="checkbox"/>	Null	
id_usuario	serial	Primary Key	<input checked="" type="checkbox"/>	nextval('gar.usuario_id_usuario')	pk_usuario
nombres	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		
apellidos	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		
genero	gar.dom_genero (char(1))		<input checked="" type="checkbox"/>		
cuenta	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		
clave	gar.dom_descripcion (varchar(100))		<input checked="" type="checkbox"/>		
tipo	gar.dom_tipo (varchar(13))		<input checked="" type="checkbox"/>		
fecha_acceso	gar.dom_fecha (bigint)		<input checked="" type="checkbox"/>		

ANEXO 7

MANUAL DE USUARIO DEL SISTEMA DE GARANTÍAS

Introducción de sistema

Este manual está enfocado a la guía en el manejo del sistema de garantías, conocido también como SYSGAR. Donde se proporciona a los usuarios toda la información posible de la manipulación y control de la información de acuerdo a las herramientas proporcionadas en el sistema, con el fin de realizar sus actividades con eficacia y eficiencia.

Funcionamiento general

SYSGAR está enfocado principalmente al control de garantías de las obras que administra el GAD Municipal de Cantón Chimbo. Básicamente a las alertas y notificaciones de las garantías que se van a caducar la fecha de vencimiento, para su posterior renovación.

Objetivos

Bridar una información clara, precisa y básica para el control y manipulación del sistema de garantías.

Autenticación de los usuarios

En la figura 1 se visualiza la ventana de acceso al sistema la cual le pedirá su **CUENTA** y **CLAVE** pre registrada por el administrador en caso de un usuario, mientras que los contratista su ingreso será de acuerdo a su registro establecido solo para consultas.



Figura 1 Autenticación de los usuarios

Pantalla Principal

Una vez que se autentifico le aparecerá la estructura principal del sistema, tal como se muestra en la Figura 2. La cual consta del encabezado que visualiza el logo y nombre del sistema, posee un menú que está dividido en varios módulos. En la parte central se visualizara lo más importante del sistema las alertas de caducidad y vencimiento de las garantías. Cabe mencionar que el tiempo establecido está basado en la ley de garantías.

The screenshot shows the main interface of the SYSGAR system. At the top, there is a header with the logo 'GOR SYSTEM' and 'SISTEMA DE GARANTIAS'. Below the header is a navigation menu with options like 'Inicio', 'Garantía', 'Entidades', 'Cliente', 'Consultas', and 'Ayuda'. The main content area displays a notification titled 'A INGRESADO AL SISTEMA DE GARANTIAS' and 'NOTIFICACION DE POLIZAR POR VENCER'. It contains a table with the following data:

Días	Contrato	Lugar	Poliza	Monto	Inicio	Fin	Contraista	
1	CONTRATO001	Guamuju	878787	67.7	01-may-2015	19-may-2015	Colcha Victor	Notificar

Below this table, there is another notification titled 'NOTIFICACION DE POLIZAR VENCIDAS' with a table containing one row:

Días	Contrato	Lugar	Poliza	Monto	Inicio	Fin	Contraista	
4	CONTRATO0012	Belavista	45644	59787.8	01-may-2015	14-may-2015	Sanchez Mariene	Notificar

The date 'lunes, 18 de mayo de 2015' is visible at the bottom right of the page.

Figura 2 Pantalla Principal SYSGAR

Menú del sistema

El menú representa los módulos establecidos para la administración del sistema, para una buena organización a la hora de realizar las tareas. En la Figura 3 se muestra como se estableció

This screenshot shows the same SYSGAR system main screen as Figure 2, but with the 'Garantía' menu item in the top navigation bar expanded. The expanded menu shows two sub-items: 'Contrato' and 'Renovacion'. The rest of the page content, including the notification tables, remains the same as in Figure 2.

Figura 3 Pantalla Principal SYSGAR

Alertas en el inicio al sistema de garantías

En la Figura 4 observaremos las alertas que posee este sistema referente a las garantías de los obras para su cumplimiento.

El color verde representa las garantías que están por vencer, el tiempo establecido de acuerdo a la ley de garantías.

El color rojo representa las garantías que están ya se vencieron.

A INGRESADO AL SISTEMA DE GARANTIAS								
NOTIFICACION DE POLIZAR POR VENCER								
Dias	Contrato	Lugar	Poliza	Monto	Inicio	Fin	Contratista	
1	CONTRATO001	Guanaju	878787	67.7	01-may-2015	18-may-2015	Colcha Victor	Notificar

NOTIFICACION DE POLIZAR VENCIDAS								
Dias	Contrato	Lugar	Poliza	Monto	Inicio	Fin	Contratista	
4	CONTRATO0012	Belavista	45844	56787.8	01-may-2015	14-may-2015	Sanchez Mariene	Notificar

Figura 4 Pantalla Principal SYSGAR

Notificación

En Figura 5 se muestra un botón **NOTIFICACIÓN** el cual se emitirá por correo electrónico una notificación de vencimiento de la garantía a los contratistas mientras que a la compañía se enviara un archivo adjunto de los datos. Tomando en cuenta que solo serán en las notificaciones por vencer.

NOTIFICACION DE POLIZAR POR VENCER								
Dias	Contrato	Lugar	Poliza	Monto	Inicio	Fin	Contratista	
1	CONTRATO001	Guanaju	878787	67.7	01-may-2015	19-may-2015	Colcha Victor	Notificar

Figura 5 Pantalla Principal SYSGAR

Módulo de Garantía

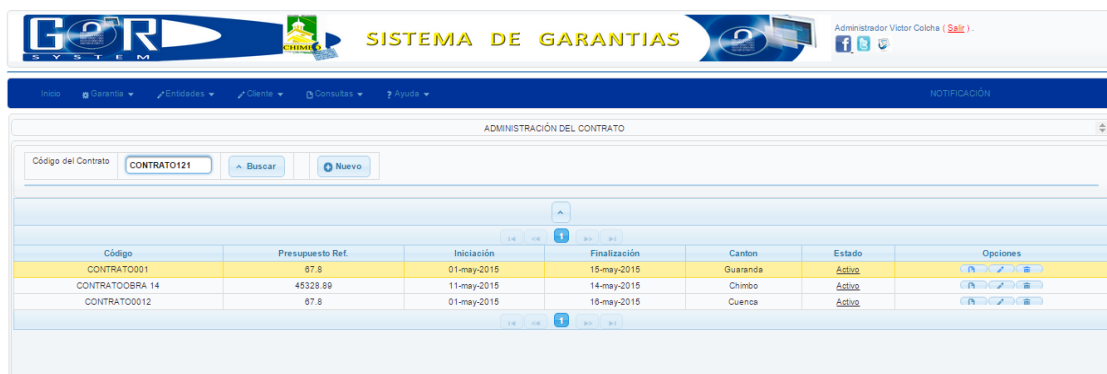
Este se encuentra en la parte inicial del menú y consta de la administración del **CONTRATO** y la administración de la **RENOVACIÓN** con se visualiza en la Figura 6.



Figura 6 Modulo garantía

Administración del Contrato

En la figura 7 se muestra la administración del contrato de cada obra.

The image shows a web application interface for 'ADMINISTRACIÓN DEL CONTRATO'. It features a search bar with 'CONTRATO121' and a 'Nuevo' button. Below is a table with columns for 'Código', 'Presupuesto Ref.', 'Iniciación', 'Finalización', 'Canton', 'Estado', and 'Opciones'. The table contains three rows of contract data.

Código	Presupuesto Ref.	Iniciación	Finalización	Canton	Estado	Opciones
CONTRAT001	87.8	01-may-2015	15-may-2015	Guaranda	Activo	[Icons]
CONTRAT00BRA 14	49328.89	11-may-2015	14-may-2015	Chimbo	Activo	[Icons]
CONTRAT00012	87.8	01-may-2015	16-may-2015	Cuenca	Activo	[Icons]

Figura 7 Modulo garantía

Nuevo contrato.- En la Figura 7 se muestra los datos de ingreso para le contrato

CODIGO DEL CONTRATO:
contrato principal 4343

DATOS DEL PROCESO DE CONTRATACION:
 Obra: Presupuesto: #.00 Fecha Actual:

FECHA DE DURACIÓN:
 Inicio: Fin: Plazo:

UBICACION Y REFERENCIAS ADICIONALES:
 Provincia: Cantón: Parroquia:

INVOLUCRADOS:
 Referencias: Contratista: Contratista:
 Entidad:

Figura 8 Modulo garantía

Opciones varias para la administración del contrato

En el sistema en todos los módulos posee las opciones de listar, actualizar y eliminar los datos

Código	Presupuesto Ref.	Iniciación	Finalización	Cantón	Estado	Opciones
CONTRATO001	87.8	01-may-2015	18-may-2015	Guaranda	Activo	<input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/>
CONTRATOOBRA 14	45328.89	11-may-2015	14-may-2015	Chimbo	Activo	<input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/>
CONTRATO0012	87.8	01-may-2015	16-may-2015	Cuenca	Activo	<input type="button" value="Actualizar"/> <input type="button" value="Eliminar"/>

Figura 9 Modulo garantía

Administración de las garantías

Para Garantizar una obra primero se debe buscar el contrato como se visualiza en la figura 10.

Buscar Contrato
 Seleccione el Contrato

Código	Obra
CONTRATO001	Reconstrucción del area verde del parque central del cantón Chimbo
CONTRATOOBRA 14	Reconstrucción de las calles principales de Cantón Chimbo
CONTRATO0012	Siembra de plantas ornamentales en el parque del Cantón Chimbo

Figura 10 buscar contrato

Garantizar Obra

Una vez obtenida los datos del contrato de la obra como se muestra en la figura 11, se activa la opción GARANTÍA en donde llenamos los datos de la garantía como se visualiza en la Figura 12.

The screenshot shows a web application interface for 'ADMINISTRACIÓN DE LA GARANTÍA'. At the top, there is a navigation bar with 'Inicio', 'Garantía', 'Entidades', 'Cliente', 'Consultas', and 'Ayuda'. Below this, a search bar contains 'CONTRATO OBRA 14' and buttons for 'Buscar contrato', 'Garantía', and 'Renovar'. The main content area displays a table with the following data:

Código del Contrato	Obra	Presupuesto	Plazo
CONTRATO OBRA 14	Reconstrucción de las calles principales de Cantón Chimbo	45328.89	3

Below the table, there are tabs for 'Renovación', 'Garantía', 'Poliza', 'Monto', 'Inicio', 'Fin', 'Contrato', and 'Opciones'. The 'Garantía' tab is active, showing the message 'No se ha encontrado el Seguro'.

Figura 11 garantiza la obra

Ingreso de la garantía

The screenshot shows a 'Nuevos Datos' form for entering guarantee information. The form is divided into several sections:

- DOCUMENTO:** Forma de Garantía (Póliza de seguro), Código Póliza de seguro (534343).
- MONTO Y PAGO:** Monto (43453), Forma de pago (40%), Renovación (0).
- FECHA DE DURACION:** Inicio (18/05/15), Fin (29/05/15), Plazo (11).
- GARANTIA Y ASEGURADORA:** Garantía (Garantía de fiel cumplimiento), Compañía (Aseguradora del sur), Fecha Actual (18/05/15).
- UBICACION:** Provincia (Azúay), Cantón (Cuenca), Parroquia (El Sagrario).

Buttons for 'Aceptar' and 'Cancelar' are located at the bottom of the form.

Figura 12 garantiza la obra

Renovar la Garantía

Una vez que caduca la fecha de la garantía se procede hacer la renovación de esta, en base al artículo III de la ley de garantías. Por lo cual el botón de GARANTÍA una vez registrado se desactiva y automáticamente se activa el de las RENOVACIONES como se llena los datos. Se puede apreciar en la Figura 13

Figura 13 Renovación de la Garantía

Renovaciones Varias: Cabe mencionar que se puede hacer varias renovaciones de la garantía hasta el momento que de por terminado el contrato de la obra. En la Figura 14 se puede observar el contador de renovaciones realizadas

Renovación	Garantía	Póliza	Monto	Inicio	Fin	Contrato	Opciones
0	Garantía por anticipo	878787	87.7	01-may-2015	18-may-2015	CONTRATO001	[Iconos]

Figura 14 Renovación de la Garantía

Validación de datos

En cada operación insertar actualizar eliminar existen validaciones que permitirán un control óptimo de información. En la Figura 15 se observa las validaciones de insertar datos del contrato.

- El código de proceso del contrato es obligatorio
- La descripción de la obra es requerida
- El Presupuesto es requerido
- Debe Seleccionar la provincia
- Debe Seleccionar el cantón
- Debe Seleccionar la Parroquia
- Debe Seleccionar un contratista
- Debe Seleccionar una Entidad

Figura 14 Renovación de la Garantía

