



ESCUELA SUPERIOR POLICTÉCNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA EN SISTEMAS

“ANÁLISIS COMPARATIVO DE HERRAMIENTAS PARA POSTGRESQL, CASO

PRÁCTICO: SISTEMA WEB DE GESTIÓN DE CONVENIOS ESPOCH”

TESIS DE GRADO

Previa obtención del título de:

INGENIERO EN SISTEMAS INFORMÁTICOS

Presentado por:

EDGAR ALONSO CUENCA SARANGO

RIOBAMBA – ECUADOR

2013

AGRADECIMIENTO

Este trabajo agradezco primero a DIOS por darme la vida, a mis padres Víctor y Lucía que me han dado toda la educación y su sabiduría, a mis hermanos Mario, Santiago, Claudio y mi hermana Jimena con los que compartí mi niñez y juventud, a todos y cada uno de mis amigos que me motivaron y ayudaron a seguir adelante y luchar por alcanzar estos anhelados sueños de ser un profesional.

Edgar Alonso Cuenca Sarango

DEDICATORIA

A mi DIOS, y principalmente a mis padres Víctor y Lucía, a mis hermanos Mario, Santiago, Claudio y mi hermana Jimena, y a todos mis compañeros y amigos, a cada uno por su tiempo dado y comprensión de mis logros y caídas que estuvieron para alentarme siempre.

Edgar Alonso Cuenca Sarango

FIRMAS RESPONSABILIDAD Y NOTA

NOMBRE	FIRMA	FECHA
Ing. Iván Menes Camejo DECANO DE LA FACULTAD DE INFORMÁTICA Y ELECTRONICA
Ing. Raúl Rosero DIRECTOR DE LA ESCUELA DE INGENIERIA EN SISTEMAS
Ing. Danilo Pastor DIRECTOR DE TESIS
Ing. Fernando Proaño MIEMBRO DEL TRIBUNAL
Tlgo. Carlos Rodríguez DIRECTOR DEL CENTRO DE DOCUMENTACIÓN
NOTA DE LA TESIS	

RESPONSABILIDAD DEL AUTOR

“Yo Edgar Alonso Cuenca Sarango soy responsable de las ideas, doctrinas y resultados expuestos en esta Tesis de Grado, y el patrimonio intelectual de la misma pertenece a la Escuela Superior Politécnica de Chimborazo.”

FIRMA:

Edgar Alonso Cuenca Sarango

INDICE DE ABREVIATURAS

A

ACO AccesControlObject– Objetosdecontroldeacceso

AM AgileModeling–ModeladoAgile

B

BD BasedeDatos

C

CLR.CommonLanguageRuntime–LenguajeComúndeEjecución

CR ClaseResponsabilidadyColaboración

CSS CascadingStyleSheets–HojadeEstiloenCascada

D

DLL DynamicallyLinkedLibrary-LibreríadeEnlaceDinámico

F

FAQ FrencuentlyAskedQuestion – PreguntasyRespuestasFrecuentes

G

GPL GeneralPublicLicense-Licencia PúblicaGeneral

GUI GraphicalUserInterface– Interfazgráficadeusuario

H

HTTP HypertextTransferProtocol–Protocolode

transferenciadeHipertexto

HW Hardware

I

IDE Integrated Development Environment – Entorno de Desarrollo Integrado

IIS Internet Information Server – Servidor de información de Internet

IMAP Internet Message Access protocol – Protocolo de acceso a mensajes de internet

ISAPI Internet Server API Application Programming Interface –

Interfaz de programación de aplicaciones para Internet Information Server

M

MVC Modelo Vista Controlador

N

NAT Network Address Translation

O

ODBC Open Data Base Connectivity

ODBMS Object Data Base Management System

OO Orientada a Objetos

P

P Promedio máximo

PHP Personal Home Page

POO Programación Orientada a Objetos

PGDG PostgreSQL Global Development Group

R

RAD Rapid Application Development – Desarrollo Rápido de Aplicaciones

S

SOAP Simple Object Access Protocol – Protocolo de Acceso a objetos simples

SQL Structure Query Language – Lenguaje de Consulta Estructurado

SGDB Sistema Gestor de Base de datos

U

URI Uniform Resource Identifier – Identificador uniforme del recurso

URL Uniform Resource Locator – Localizador Uniforme de Recursos

UML Unified Modeling Language - Lenguaje de modelado unificado

UNICODE Universalidad, Uniformidad y Unicidad

V

VR Velocidad de la Red

X

XML Extended Markup Language – Lenguaje de marcado extendido

XP Extreme Programming – Programación extrema

Z

INDICE GENERAL

CAPÍTULO I.....	20
1.1. Antecedentes	20
1.2. Justificación del Proyecto de Tesis	23
1.2.1. Justificación Teórica	23
1.2.2. Justificación Práctica.....	24
1.3. Objetivos	24
1.3.1. Objetivo General	24
1.3.2. Objetivos Específicos.....	25
1.4. Hipótesis.....	25
1.5. Métodos y Técnicas.....	26
1.5.1. Métodos.....	26
1.5.2. Técnicas	26
CAPÍTULO II	28
2.1. Definiciones	28
2.1.1. Base de Datos.....	28
2.1.2. Base de Datos Orientadas a Objetos	29
2.1.3. Componentes principales de una Base de Datos	30
2.1.4. Sistema Gestor de Bases de Datos	31
2.1.5. Funciones de los Sistemas de Gestión de Base de Datos	33
2.1.6. Ventajas de los Sistemas de Bases de Datos	38
2.1.6.1. Ventajas por la integración de datos	38
2.1.6.2. Ventajas por la existencia del SGBD	40
2.1.7. Componentes de un Sistema de Gestión de Base de Datos.....	42
2.1.8. Arquitectura de un Sistema de Base de Datos.....	44
2.2. Productos SGBD disponibles en el mercado.....	46
2.2.1. Sistemas libres.....	46
2.2.2. Sistemas no libres.....	47
2.2.3. Sistemas no libres y gratuitos.....	48
2.3. Herramientas de administración de bases de datos	48
2.3.1. Administración de bases de datos	48

2.3.2.	Postgresql	50
2.3.2.1.	Nombre del producto.....	50
2.3.2.2.	Evolución	51
2.3.2.3.	Ciclo de vida (EOL) y soporte	56
2.3.2.4.	Arquitectura de postgresql	57
2.3.2.5.	Componentes más importantes en un sistema PostgreSQL	58
2.3.2.6.	Características de Postgresql.....	59
2.3.2.7.	Límites de PostgreSQL	63
CAPÍTULO III.....		64
2.1.	Recursos Hardware y Software utilizados	65
2.1.1.	Recursos Hardware	65
2.1.2.	Recursos Software.....	65
2.2.	Determinación de las Herramientas Postgresql a comparar	65
2.3.	Análisis de las Herramientas seleccionadas	69
2.3.1.	PGADMIN III	69
2.3.1.1.	Características	69
3.1.1.2.	Requisitos de Instalación.....	70
3.1.2.	PHP PGADMIN.....	70
3.1.2.1.	Características	71
3.1.2.2.	Requisitos de instalación.....	72
3.1.3.	EMS SQL MANAGER FOR POSTGRESQL.....	72
3.1.3.1.	Características	73
3.1.3.2.	Requisitos de Instalación.....	74
3.2.	Criterios técnicos contemplados para ejecución de pruebas en las Herramientas.....	74
3.3.	Determinación de parámetros de comparación	75
3.3.1.	Parámetro 1: Instalación.....	76
3.3.2.	Parámetro 2: Seguridad	77
3.3.3.	Parámetro 3: Administración	77
3.3.4.	Parámetro 4: Calidad.....	78
3.4.	Descripción de los módulos de pruebas	79
3.4.1.	Módulo de Navegación de interfaces	80
3.4.2.	Módulo de transacciones de alta, media y baja.....	81
3.4.3.	Módulo de funciones.....	81

3.5.	Desarrollo de los módulos de pruebas	81
3.5.1.	GUI PgAdminIII	82
3.5.1.1.	Módulo de Navegación de Interfaces.....	83
3.5.1.2.	Módulo de Funciones y transacciones de alta.....	84
3.5.1.3.	Módulo de funciones y transacciones de baja.....	85
3.5.2.	GUI PHPpgAdmin	86
3.5.2.1.	Módulo de navegación de interfaces.....	87
3.5.2.2.	Módulo de funciones y transacciones de alta.....	87
3.5.2.3.	Módulo de funciones y transacciones de baja.....	88
3.5.3.	EMS SQL Manager for PostgreSQL.....	89
3.5.3.1.	Módulo de navegación de interfaces.....	90
3.5.3.2.	Módulo de funciones y transacciones de alta.....	90
3.5.3.3.	Módulo de funciones y transacciones de baja.....	91
3.6.	Análisis Comparativo.....	92
3.6.1.	Indicador 1: Instalación.....	92
3.6.1.1.	Determinación del Indicador.....	92
3.6.1.2.	Valoraciones.....	95
3.6.1.3.	Calificación	95
3.6.1.4.	Interpretación	101
3.6.1.5.	Descripción Resultados	101
3.6.2.	Indicador 2: Seguridad	103
3.6.2.1.	Determinación del Indicador.....	103
3.6.2.2.	Valoraciones.....	105
3.6.2.3.	Calificación	106
3.6.2.4.	Interpretación	112
3.6.2.5.	Descripción Resultados	113
3.6.3.	Indicador 3: Administración	114
3.6.3.1.	Determinación del Indicador.....	114
3.6.3.2.	Valoraciones.....	117
3.6.3.3.	Calificación	118
3.6.3.4.	Interpretación	124
3.6.3.5.	Descripción Resultados	124
3.6.4.	Indicador 4: Calidad.....	126

3.6.4.1.	Determinación del Indicador	126
3.6.4.2.	Valoraciones.....	131
3.6.4.3.	Calificación	132
3.6.4.4.	Interpretación	137
3.6.4.5.	Descripción Resultados	137
3.7.	Puntajes Alcanzados.....	138
3.8.	Interpretación	139
3.9.	Análisis de Resultados	141
3.10.	Comprobación de Hipótesis	142
3.10.1.	Análisis Comparativo en porcentajes de las herramientas de administración para postgresql, PgAdminIII, PhpPgAdmin y EMS SQL Lite forPostgreSQL.	142
3.10.2.	Conclusión de la Comprobación de la Hipótesis.....	145
CAPÍTULO IV		146
4.1.	Ingeniería de la información	147
4.1.1.	Definición del ámbito	147
4.1.2.	Identificar Requerimientos.....	147
4.1.3.	Estudio de Factibilidad.....	148
4.1.3.1.	Factibilidad Económica.....	148
4.1.3.2.	Factibilidad Técnica	149
4.1.3.3.	Factibilidad Operativa.....	149
4.1.3.4.	Factibilidad Legal.....	150
4.1.4	Planificación Temporal	150
4.2.	Análisis del sistema	150
4.2.1.	Casos de Uso del Sistema	150
4.2.2.	Detalle de los casos de uso identificados	152
4.2.2.1.	Funcionalidad de los casos de uso.....	152
4.2.3.	Diagramas de secuencia	156
4.3.	Diseño	160
4.3.1.	Casos de uso reales	160
4.3.2.	Definición de informes e interfaces de usuario	163
4.3.3.	Diagramas de interacción.....	164
4.3.4.	Diagrama de clases.....	168
4.3.5.	Diagrama de Despliegue	168

4.4. Implementación y Pruebas	169
4.4.1. Definición de estándares de programación	169
4.4.2. Pruebas Unitarias	170
4.4.3. Pruebas de modulo y del sistema	170

CONCLUSIONES

RECOMENDACIONES

RESUMEN

SUMMARY

GLOSARIO

BIBLIOGRAFÍA

ANEXOS

INDICE FIGURAS

FIGURA II.1 ESQUEMA DEL FUNCIONAMIENTO Y UTILIDAD DE UN SGBD.....	32
FIGURA II.2 COMPONENTES DE UN SGBD	43
FIGURA II.3 NIVELES DE ABSTRACCIÓN DE LA ARQUITECTURA ANSI	46
FIGURA II.4 VERSIONES DE POSTGRESQL.....	55
FIGURA II.5 LÍNEAS CÓDIGO	56
FIGURA II.6 CICLO DE VIDA POSTGRESQL.....	57
FIGURA II.7 ARQUITECTURA DE POSTGRESQL.....	58
FIGURA II.8 COMPONENTES POSTGRESQL	58
FIGURA III. 9 VALORIZACIÓN DE LA ENCUESTA.....	67
FIGURA III.10 ESTRUCTURA DETALLADA DE PGADMINIII.....	82
FIGURA III. 11 INTERFAZ PRINCIPAL SISTEMA CON LA BASE DE DATOS CREADA.....	83
FIGURA III. 12 INTERFAZ DE PRESENTACIÓN DE INGRESO INFORMACIÓN A LA BASE DE DATOS	83
FIGURA III. 13 INTERFAZ DE FUNCIONES DE INGRESO Y ACTUALIZACIÓN DE DATOS.....	84
FIGURA III. 14 PGADMINIII CON DATOS PRESENTADOS DESDE LA INTERFAZ.....	84
FIGURA III. 15 INTERFAZ DE FUNCIONES DE ELIMINACIÓN EN LA BASE DE DATOS	85
FIGURA III. 16 INTERFAZ PGADMINIII ELIMINADO DATO DESDE INTERFAZ.....	86
FIGURA III. 17 ESTRUCTURA DETALLADA DE PHPPGADMIN VER ANEXO 1	86
FIGURA III. 18 PHPPGADMIN CON DATOS PRESENTADOS DESDE LA INTERFAZ.....	86
FIGURA III. 19 INTERFAZ DE NAVEGACIÓN DE FUNCIONES PHPPGADMIN	87
FIGURA III. 20 INTERFAZ DE LISTADO DATOS CONTROL ELIMINACIONES.....	87
FIGURA III. 21 PHPPGADMIN DATOS QUE SE PRESENTAN DESDE LA INTERFAZ	88
FIGURA III. 22 INTERFAZ DE PARA REALIZAR CARGAS TRANSACCIONALES.....	88
FIGURA III. 23 ESTRUCTURA DETALLADA DE EMS SQL LITE FOR POSTGRESQL VER ANEXO 1.....	89
FIGURA III. 24 CON DATOS PRESENTADOS EN LA TABLA DESDE INTERFAZ.....	89
FIGURA III. 25 INTERFAZ DE NAVEGACIÓN DE FUNCIONES EMS SQL LITE FOR POSTGRESQL	90
FIGURA III. 26 INTERFAZ DE LISTADO DATOS CONTROL ELIMINACIONES.....	90
FIGURA III. 27 EMS SQL LITE FOR POSTGRESQL DATOS QUE SE PRESENTAN DESDE LA INTERFAZ.....	91
FIGURA III. 28 INTERFAZ DE PARA REALIZAR CARGAS TRANSACCIONALES.....	91
FIGURA III. 29 RESULTADO POR ÍNDICE DEL INDICADOR 1: INSTALACIÓN.....	99
FIGURA III. 30 RESULTADO FINAL DEL INDICADOR 1: INSTALACIÓN	100
FIGURA III.31 RESULTADO POR ÍNDICE DEL INDICADOR 2: SEGURIDAD	110
FIGURA III.32 RESULTADO FINAL DEL INDICADOR 2: SEGURIDAD	112
FIGURA III. 33 RESULTADO POR ÍNDICE DEL INDICADOR 3: ADMINISTRACIÓN.....	122
FIGURA III. 34 RESULTADO FINAL DEL INDICADOR 3: ADMINISTRACIÓN	123
FIGURA III.35 RESULTADO POR ÍNDICE DEL INDICADOR 4: CALIDAD	135
FIGURA III.36 RESULTADO FINAL DEL INDICADOR 4: CALIDAD	136
FIGURA III.37 RESULTADO HERRAMIENTAS Y NO HERRAMIENTAS PARA POSTGRESQL	144
FIGURA IV.38 DIAGRAMA DE CASOS DE USO GENERAL	156
FIGURA IV.39 DIAGRAMA DE SECUENCIA DE AUTENTICACIÓN.....	157
FIGURA IV.40 DIAGRAMA DE SECUENCIA DE CONSULTAS	157
FIGURA IV.41 DIAGRAMA DE SECUENCIA DE GESTIÓN DE CONVENIOS	158
FIGURA IV.42 DIAGRAMA DE SECUENCIA DE CANCELACIÓN DE CONVENIOS.....	159
FIGURA IV.43 DIAGRAMA DE CASOS DE USO DE AUTENTICACIÓN	160
FIGURA IV.44 DIAGRAMA DE CASOS DE USO DE GESTIÓN DE CONVENIOS	161
FIGURA IV.45 DIAGRAMA DE CASOS DE USO DE CANCELACIÓN DE CONVENIOS	162
FIGURA IV.46 DIAGRAMA DE CASOS DE USO DE CONSULTAS	163

FIGURA IV.47 DIAGRAMA DE SECUENCIA DE AUTENTICACIÓN.....	164
FIGURA IV.48 DIAGRAMA DE SECUENCIA DE GESTIÓN DE CONVENIOS.....	165
FIGURA IV.49 DIAGRAMA DE SECUENCIA DE CANCELACIÓN DE CONVENIO.....	166
FIGURA IV.50 DIAGRAMA DE SECUENCIA DE CONSULTAS.....	167
FIGURA IV.51 DIAGRAMA DE CALLES.....	167
FIGURA IV.52 DIAGRAMA DE CLASES.....	168
FIGURA IV.53 DIAGRAMA DE COMPONENTES.....	168
FIGURA IV.54 DIAGRAMA DE NODOS.....	169

INDICE TABLAS

TABLA I DESCRIPCIÓN INDICADOR 1: INSTALACIÓN.....	76
TABLA II DESCRIPCIÓN INDICADOR 2: SEGURIDAD	77
TABLA III DESCRIPCIÓN INDICADOR 3: ADMINISTRACIÓN	78
TABLA IV DESCRIPCIÓN INDICADOR 4: CALIDAD.....	79
TABLA V VALORIZACIÓN PARA EL INDICADOR 1: INSTALACIÓN	93
TABLA VI VALORIZACIÓN DEL ÍNDICE 1.1: TIEMPO DE INSTALACIÓN.....	93
TABLA VII VALORIZACIÓN DEL ÍNDICE 1.2: PASOS DE CONFIGURACIÓN	93
TABLA VIII VALORIZACIÓN DEL ÍNDICE 1.3: NIVEL DE CONTROL DE ACCESO	94
TABLA IX CALIFICACIÓN DEL INDICADOR 1: INSTALACIÓN, SE BASARÁN EN LA CALIFICACIÓN DE LAS TABLAS (6-10)	95
TABLA X VALORES Y PORCENTAJES DEL INDICADOR 1: INSTALACIÓN, CON SUS RESPECTIVOS ÍNDICES....	98
TABLA XI VALORES Y PORCENTAJES FINALES DEL INDICADOR 1: INSTALACIÓN.....	100
TABLA XII REPRESENTACIÓN DEL INDICADOR 1: INSTALACIÓN.....	101
TABLA XIII VALORIZACIÓN PARA EL INDICADOR 2: SEGURIDAD	103
TABLA XIV VALORIZACIÓN DEL ÍNDICE 1.1: TIPO DE AUTENTICACIÓN USADA	103
TABLA XV VALORIZACIÓN DEL ÍNDICE 2.2: TIPOS DE ALGORITMOS DE SEGURIDAD.....	104
TABLA XVI VALORIZACIÓN DEL ÍNDICE 2.3: NIVEL DE SEGURIDAD PARA ESTABLECER SESIÓN	104
TABLA XVII VALORIZACIÓN DEL ÍNDICE 2.4: TIPOS DE SOPORTE DE RESPALDO Y RESTAURACIÓN	105
TABLA XVIII RESULTADOS DEL INDICADOR 2: SEGURIDAD, DESPUÉS DE LA OBTENCIÓN DE LA EJECUCIÓN DE LAS DIFERENTES PRUEBAS VER ANEXO 2.	105
TABLA XIX CALIFICACIÓN DEL INDICADOR 2: SEGURIDAD, SE BASARÁN EN LA CALIFICACIÓN DE LAS TABLAS ()	106
TABLA XX VALORES Y PORCENTAJES DEL INDICADOR 2: SEGURIDAD, CON SUS RESPECTIVOS ÍNDICES..	110
TABLA XXI VALORES Y PORCENTAJES FINALES DEL INDICADOR 2: SEGURIDAD.....	111
TABLA XXII REPRESENTACIÓN DEL INDICADOR 2: SEGURIDAD	112
TABLA XXIII VALORIZACIÓN PARA EL INDICADOR 3: ADMINISTRACIÓN	114
TABLA XXIV VALORIZACIÓN DEL ÍNDICE 3.1: CONEXIÓN LOCAL	115
TABLA XXV VALORIZACIÓN DEL ÍNDICE 3.2: CONEXIÓN REMOTA.....	115
TABLA XXVI VALORIZACIÓN DEL ÍNDICE 3.3: FACILIDAD DISEÑO DE BASE DE DATOS.	115
TABLA XXVII VALORIZACIÓN DEL ÍNDICE 3.4: FACILIDAD DISEÑO DE TABLAS.....	116
TABLA XXVIII VALORIZACIÓN DEL ÍNDICE 3.5: FACILIDAD DISEÑO DE FUNCIONES	116
TABLA XXIX VALORIZACIÓN DEL ÍNDICE 3.6: DISEÑO Y CONSULTAS DE QUERYS	116
TABLA XXX RESULTADOS DEL INDICADOR 3: ADMINISTRACIÓN, DESPUÉS DE LA OBTENCIÓN DE LA EJECUCIÓN DE LAS DIFERENTES PRUEBAS VER ANEXO 2.	117
TABLA XXXI CALIFICACIÓN DEL INDICADOR 3: ADMINISTRACIÓN, SE BASARÁN EN LA CALIFICACIÓN DE LAS TABLAS (6-10)	118
TABLA XXXII VALORES Y PORCENTAJES DEL INDICADOR 3: ADMINISTRACIÓN, CON SUS RESPECTIVOS ÍNDICES.....	121
TABLA XXXIII VALORES Y PORCENTAJES FINALES DEL INDICADOR 3: ADMINISTRACIÓN.....	123
TABLA XXXIV REPRESENTACIÓN DEL INDICADOR 3: ADMINISTRACIÓN.....	124
TABLA XXXV VALORIZACIÓN PARA EL INDICADOR 4: CALIDAD	126
TABLA XXXVI VALORIZACIÓN DEL ÍNDICE 4.1: USO DE RECURSOS HARDWARE	127
TABLA XXXVII VALORIZACIÓN DEL ÍNDICE 4.2: FACILIDAD DE USO	127
TABLA XXXVIII PREGUNTAS Y CONSULTA DEL ÍNDICE 4.2: FACILIDAD DE USO.....	128
TABLA XXXIX VALORIZACIÓN DEL ÍNDICE 4.3COMPATIBILIDAD SOFTWARE.	130

TABLA XL RESULTADOS DEL INDICADOR 4: CALIDAD, DESPUÉS DE LA OBTENCIÓN DE LA EJECUCIÓN DE LA PRUEBA VER ANEXO 2.....	131
TABLA XLI CALIFICACIÓN DEL INDICADOR 4: CALIDAD, SE BASARÁN EN LA CALIFICACIÓN DE LAS TABLAS (6-10).....	132
TABLA XLII VALORES Y PORCENTAJES DEL INDICADOR 4: CALIDAD CON SUS RESPECTIVOS ÍNDICES.....	134
TABLA XLIII VALORES Y PORCENTAJES FINALES DEL INDICADOR 4: CALIDAD.	136
TABLA XLIV REPRESENTACIÓN DEL INDICADOR 4: CALIDAD	137
TABLA XLV CUADRO RESULTADOS DE INDICADOR E ÍNDICE	142
TABLA XLVI RECURSOS HUMANOS.....	149
TABLA XLVII RECURSOS SOFTWARE	149
TABLA XLVIII ACTORES Y SUS FUNCIONES	151
TABLA XLIX CASO DE USO DE AUTENTICACIÓN.....	152
TABLA L CASO USO DE GESTIÓN DE LOS CONVENIOS	153
TABLA LI CASO DE USO DE CANCELA CONVENIO	154
TABLA LII CASO DE USO DE CONSULTAS	155
TABLA LIII CASO DE USO DE AUTENTICACIÓN.....	160
TABLA LIV CASO DE USO DE GESTIÓN DE INFORMACIÓN DE LOS CONVENIOS	160
TABLA LV CASO DE USO DE CANCELACIÓN DE CONVENIOS	161
TABLA LVI CASO DE USO DE CONSULTAS	162

INTRODUCCIÓN

El presente trabajo de investigación de tesis previo a la obtención del título de Ingeniería en Sistemas Informáticos, trata “ANÁLISIS COMPARATIVO DE HERRAMIENTAS PARA POSTGRESQL, CASO PRÁCTICO: SISTEMA WEB DE GESTIÓN DE CONVENIOS ESPOCH”.

Es un estudio y análisis investigativo a fondo de como por medio de diferentes Herramientas Open Source Gráficas de Administración de Bases de Datos conocidos también como Sistemas Gestores de Bases de Datos paraPostgresqly así obtener como resultado laHerramienta Gráfica de Postgresql más adecuada para el desarrollo de aplicaciones web.

Como precedente cuando se trata de desarrollar aplicaciones web en lenguaje Java, la mayoría de desarrolladores implementan sus proyectos en la Base de Datos que tienen mejor conocimiento, sin realizar una investigación previa en búsqueda de Herramientas Gráficas de Bases de Datos que se ajusten a sus necesidades a su vez ofrezca el mejor rendimiento y estabilidad.

En la actualidad el desarrollo de aplicaciones web requieren la mejor optimización de recursos de hardware y software que se encuentran conectados a la red de Internet, Aquí nace la preocupación de la ingeniería de software, una vez recopilados los requisitos del proyecto web, estandarizar la búsqueda delaHerramienta de Base de Datos que se acople a las necesidades del sistema.

El presente trabajo contiene los siguientes capítulos:

El en capítulo 1 se presenta el planteamiento de la investigación, antecedentes, hipótesis, métodos y técnicas, es todo el marco referencial para el desarrollo de la tesis.

En el capítulo 2 se detalla los aspectos teóricos motivo de la investigación, conceptos, terminologías, referenciadas al objeto de estudio.

El capítulo 3 trata del desarrollo del análisis comparativo de las Herramientas Gráficas para postgresql, donde se han seleccionado 3 herramientas de Bases de Datos de postgresql para determinar cual tiene la mejor administración en la gestión de base de datos para postgresql.

Además se han determinado los indicadores e índices que sirven para comparación de las Herramientas Gráficas de Postgresql, finalizando con la demostración de la hipótesis.

En el capítulo 4 se detalla la parte aplicada de la tesis, contiene los requisitos de ingeniería de software, desarrollo rápido de aplicaciones web, estándares de desarrollo, todo referente al Sistema Web de Convenios de la ESPOCH.

CAPÍTULO I

MARCO REFERENCIAL

1.1. Antecedentes

Los desarrolladores de software se dividen en dos tipos: los que usan entornos de desarrollo integrado o IDEs y los que no. Estos últimos prefieren un editor de texto como Emacs o el Blog de Notas, un compilador y depurador. Los pertenecientes al primer tipo, sin embargo, prefieren usar IDEs para ayudarles a la generación del código y a la construcción de proyectos. Tarde o temprano, independientemente del grupo al cual pertenezcan, todos se enfrentan a sus propios problemas.

En la actualidad el desarrollo de aplicaciones web se ha vuelto muy popular a nivel de todas las empresas que deseen ofertas sus productos y servicios.

En nuestro medio al no existir estudios previos de administración de herramientas gráficas para postgresql que ayuden al desarrollador, ellos utilizan empíricamente un

entorno de desarrollo integrado, en estos casos no existen criterios de desarrollo para el aseguramiento de la calidad del software.

Actualmente los sitios web se han convertido en un punto primordial en el funcionamiento de las instituciones públicas, privadas y de empresas de cualquier índole, es por ello que la evolución en el desarrollo web ha avanzado a pasos agigantados.

Uno de los aspectos más importantes en el desarrollo de los sitios web es como se diseña su base de datos en caso de tenerla, la seguridad, y más aún la transmisión de información, es por ello que actualmente los portales necesitan asegurar la información que es enviada y recibida a los clientes.

La principal debilidad de los desarrolladores es como diseñar rápidamente las bases de datos por eso será mucho mejor si es a través de herramientas visuales de forma que puedan dar resultados satisfactorios en corto tiempo.

En este sentido las mejores prácticas de escalabilidad, confianza, seguridad, disponibilidad, en la utilización de administradores de bases de datos en postgresql están relacionadas con la tecnología y la velocidad de cambio de la misma, la cual muchas veces causa serios problemas de productividad en el diseño de las bases de datos.

Las características relacionadas con la administración visual de bases de datos les ofrecen a los desarrolladores la posibilidad de navegar y modificar las estructuras de la base de datos, como también los objetos de esquema y el almacenamiento en la base de datos, además de mantener la seguridad de la base de datos.

Las necesidades de administración en postgresql, pueden aparecer paulatinamente en el tiempo, de forma que se tiene tiempo para digerir esta nueva situación; o es posible que de manera eventual aparezcan nuevos diseños y cargas enormes en las bases de datos, en respuesta a algo que ocurre puntualmente o incluso a menudo sin previo aviso; provisionar de manera ágil el hardware necesario para hacer frente a este tipo de situaciones no es sencillo.

Son innumerables las situaciones en las que es vital contar con un administrador visual de base de datos en postgresql mucho mejor si es Open Source, que se puedan utilizar según las necesidades de cada momento y que sean a efectos prácticos ilimitados.

Adicionalmente, sin usar un administrador GUI de bases de datos, la escalabilidad, alta disponibilidad y reducción de costes de operación son las principales desventajas.

El motor de base de datos que se va a analizar es postgresql, el mismo que utiliza herramientas de administración visual de las bases de datos para su creación, por lo tanto lo que se va a realizar en el presente trabajo de investigación es su análisis comparativo de las herramientas GUI para postgresql Open Source ya que existen varias de estas pero cada una ofrece un modo de administración y se levantara una aplicación web para gestionar los convenios que existen en la Escuela Superior Politécnica de Chimborazo.

1.2. Justificación del Proyecto de Tesis

1.2.1. Justificación Teórica

La clasificación de las herramientas gráficas de postgresql mediante estudios y comparaciones técnicas es parte vital para la selección de la herramienta que permita mejorar la calidad del software, ya que si no se elige la Herramienta de postgresql con la mejor administración para las bases de datos nuestros procesos tendrían tiempos de ejecución y respuesta lentos.

Al comparar las herramientas que ayuden mejor en el diseño de base de datos postgresql el desarrollador podrá seleccionar el entorno de desarrollo más óptimo acorde a sus requerimientos.

El desarrollo de este trabajo está encaminado a realizar pruebas de laboratorio con el fin de mejorar y promover el diseño rápido de bases de datos de nuestras aplicaciones en la web, evaluando y poniendo a prueba todas las capacidades de las distintas herramientas gráficas de postgresql más populares en la actualidad.

La selección de la herramienta gráfica de postgresql óptima ayudará a mejorar los tiempos de desarrollo y peticiones del sistema web a desarrollar.

Para el desarrollo del sistema web se seleccionaría un IDE que trabaje con la herramienta gráfica de administración de bases de datos postgresql ganadora para poder tener la mejor administración.

El alcance que se desea enfocar al sistema web de convenios ESPOCH es que los tiempos de respuestas de las peticiones que realicen los usuarios sean los mínimos

optimizando el código para un excelente desempeño del sistema y sea utilizado más a menudo por su eficiencia.

1.2.2. Justificación Práctica

La razón principal para el desarrollo de este proyecto surge de las necesidades que existen, debido a que estos convenios se manejan de manera manual y rudimentaria lo que con lleva un consumo excesivo de recursos económicos, materiales y de talentos humanos, y además en el tiempo en que se demoran en realizar dicha actividad que influye en el desarrollo de la misma.

Así que con la visión de optimizar y enlazar al máximo las nuevas capacidades implementadas es posible dar una solución informática creando un Sistema Web de Gestión de Convenios (CONVENIOS ESPOCH) permitiendo a los usuarios contar con un ordenamiento alfanumérico y cronológico de los convenios que se dan cada año.

1.3. Objetivos

1.3.1. Objetivo General

Seleccionar una Herramienta GUI de Administración de Bases de datos Postgresql mediante un Análisis Comparativo para el desarrollo de un Sistema Web Gestión de Convenios en la Unidad de Vinculación de la ESPOCH

1.3.2. Objetivos Específicos

- Estudiar 3 Herramientas de Administración GUI de Bases de datos Open Source para PostgreSQL.
- Determinar parámetros de comparación de las herramientas GUI de AdministraciónPostgreSQL.
- Seleccionar la Herramienta de Administración de Base de Datos más adecuada.
- Desarrollar el Sistema Web para la Gestión de Convenios de la ESPOCH apoyada con la Herramienta Visual de Administración de Base de Datos seleccionada.

1.4. Hipótesis

La utilización de Herramientas de Administración de Base de Datos adecuada para el sistema Web permitirá mejorar la Administración de Bases de Datos en Postgresql.

1.5. Métodos y Técnicas

1.5.1. Métodos

El método utilizado como guía para la presente investigación es el Método Científico, el cual contempla los siguientes puntos:

- El planteamiento del problema que es objeto principal de nuestro estudio.
- 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, etc.

Para complementar la investigación se aplicará el método deductivo ya que parte de verdades previamente establecidas como principios generales, para luego aplicarlo a casos individuales y comprobar así su validez en el desarrollo del Sistema Web de Convenios de la ESPOCH.

1.5.2. Técnicas

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

- Revisión de Documentos
- Observación
- Técnica de Comprobación de Hipótesis

CAPÍTULO II

MARCO TEORICO

2.1. Definiciones

2.1.1. Base de Datos

Se define una base de datos como una serie de datos organizados y relacionados entre sí, los cuales son recolectados y explotados por los sistemas de información de una empresa o negocio en particular. La Base de datos es un conjunto de información almacenada en memoria auxiliar que permite acceso directo y un conjunto de programas que manipulan esos datos.

Una base de datos es un “almacén” que nos permite guardar grandes cantidades de información de forma organizada para que luego podamos encontrar y utilizar fácilmente. A continuación se presenta una guía que explicará el concepto y características de las bases de datos.

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

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

Cada base de datos se compone de una o más tablas que guarda un conjunto de datos. Cada tabla tiene una o más columnas y filas. Las columnas guardan una parte de la información sobre cada elemento que queremos guardar en la tabla, cada fila de la tabla conforma un registro. [6]

2.1.2. Base de Datos Orientadas a Objetos

En una base de datos orientada a objetos, la información se representa mediante objetos como los presentes en la programación orientada a objetos. Cuando se integra las características de una base de datos con las de un lenguaje de programación orientado a objetos, el resultado es un sistema gestor de base de datos orientada a objetos (ODBMS, Object Data Base Management System).

Un ODBMS hace que los objetos de la base de datos aparezcan como objetos de un lenguaje de programación en uno o más lenguajes de programación a los que dé soporte.

Un ODBMS extiende los lenguajes con datos persistentes de forma transparente, control de concurrencia, recuperación de datos, consultas asociativas y otras capacidades. [7]

2.1.3. Componentes principales de una Base de Datos

- **Datos.** Los datos son la Base de Datos propiamente dicha.
- **Hardware.** El hardware se refiere a los dispositivos de almacenamiento en donde reside la base de datos, así como a los dispositivos periféricos (unidad de control, canales de comunicación, etc.) necesarios para su uso.
- **Software.** Está constituido por un conjunto de programas que se conoce como Sistema
- **Manejador de Base de Datos (DBMS: DataBase Management System).** Este sistema maneja todas las solicitudes formuladas por los usuarios a la base de datos.
- **Usuarios.** Existen tres clases de usuarios relacionados con una Base de Datos:
 - ✓ El programador de aplicaciones es, quien crea programas de aplicación que utilizan la base de datos.
 - ✓ El usuario final, quien acceda a la Base de Datos por medio de un lenguaje de consulta o de programas de aplicación.
 - ✓ El Administrador de la Base de Datos (DBA: Data Base Administrator), es quien se encarga del control general del Sistema de Base de Datos. El DBA es la persona que asegura integridad, consistencia, redundancia,

seguridad, y se encarga de realizar el mantenimiento diario o periódico de los datos.[3]

2.1.4. Sistema Gestor de Bases de Datos

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

Un SGBD debe permitir:

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

Otras definiciones de SGBD:

- Un Sistema de Gestión de Bases de Datos consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. El objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez

conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos. (KORTH4, 1995)

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

En estos sistemas se proporciona un conjunto coordinado de programas, procedimientos y lenguajes que permiten a los distintos usuarios realizar sus tareas habituales con los datos, garantizando además la seguridad de los mismos.[4]

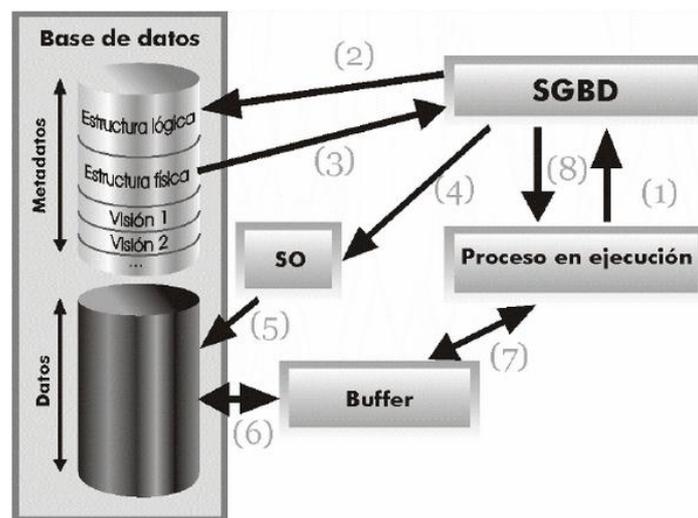


Figura II.1 Esquema del funcionamiento y utilidad de un SGBD

Fuente:<http://es.scribd.com/doc/26992505/27/2-6-5-funcionamiento-del-SGBD>

2.1.5. Funciones de los Sistemas de Gestión de Base de Datos

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

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

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

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

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

- La información sobre los datos se puede almacenar de un modo centralizado. Esto ayuda a mantener el control sobre los datos, como un recurso que son.

- El significado de los datos se puede definir, lo que ayudará a los usuarios a entender el propósito de los mismos.
- La comunicación se simplifica ya que se almacena el significado exacto. El diccionario de datos también puede identificar al usuario o usuarios que poseen los datos o que los acceden.
- Las redundancias y las inconsistencias se pueden identificar más fácilmente ya que los datos están centralizados.
- Se puede tener un historial de los cambios realizados sobre la base de datos.
- El impacto que puede producir un cambio se puede determinar antes de que sea implementado, ya que el diccionario de datos mantiene información sobre cada tipo de dato, todas sus relaciones y todos sus usuarios.
- Se puede respetar la seguridad.
- Se puede garantizar la integridad.
- Se puede proporcionar información para auditorías.

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

cambios realizados deberán ser deshechos para devolver la base de datos a un estado consistente.

4. Un SGBD proporciona un mecanismo que asegure que la base de datos se actualice correctamente cuando varios usuarios la están actualizando concurrentemente. Uno de los principales objetivos de los SGBD es el permitir que varios usuarios tengan acceso concurrente a los datos que comparten. El acceso concurrente es relativamente fácil de gestionar si todos los usuarios se dedican a leer datos, ya que no pueden interferir unos con otros. Sin embargo, cuando dos o más usuarios están accediendo a la base de datos y al menos uno de ellos está actualizando datos, pueden interferir de modo que se produzcan inconsistencias en la base de datos. El SGBD se debe encargar de que estas interferencias no se produzcan en el acceso simultáneo.

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

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

7. Un SGBD debe ser capaz de integrarse con algún software de comunicación. Muchos usuarios acceden a la base de datos desde terminales. En ocasiones estos terminales se encuentran conectados directamente a la máquina sobre la que funciona el SGBD. En otras ocasiones los terminales están en lugares remotos, por lo que la comunicación con la máquina que alberga al SGBD se debe hacer a través de una red. En cualquiera de los dos casos, el SGBD recibe peticiones en forma de mensajes responde de modo similar. Todas estas transmisiones de mensajes las maneja el gestor de comunicaciones de datos. Aunque este gestor no forma parte del SGBD, es necesario que el SGBD se pueda integrar con él para que el sistema sea comercialmente viable.

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

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

- Un SGBD debe permitir que se mantenga la independencia entre los programas y estructura de la base de datos. La independencia de datos se alcanza mediante las vistas o sub-esquemas. La independencia de datos física es más fácil de alcanzar, de hecho hay varios tipos de cambios que se pueden realizar sobre la estructura física de la base de datos sin afectar a las vistas. Sin embargo, lograr una completa independencia de datos lógica es más difícil. Añadir una nueva entidad, un atributo o una relación puede ser sencillo, pero no es tan sencillo eliminarlos.
- Un SGBD debe proporcionar una serie de herramientas que permitan administrar la base de datos de modo efectivo. Algunas herramientas trabajan a nivel externo, por lo que habrán sido producidas por el administrador de la base de datos. Las herramientas que trabajan a nivel interno deben ser proporcionadas por el distribuidor del SGBD. Algunas de ellas son:
 - ✓ Herramientas para importar y exportar datos.
 - ✓ Herramientas para monitorizar el uso y el funcionamiento de la base de datos.
 - ✓ Programas de análisis estadístico para examinar las prestaciones o las estadísticas de utilización.
 - ✓ Herramientas para reorganización de índices.

- ✓ Herramientas para aprovechar el espacio dejado en el almacenamiento físico por los registros borrados y que consoliden el espacio liberado para reutilizar cuando sea necesario.[5]

2.1.6. Ventajas de los Sistemas de Bases de Datos

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

2.1.6.1. Ventajas por la integración de datos

- ✓ **Control sobre la redundancia de datos.** Los sistemas de ficheros almacenan varias copias de los mismos datos en ficheros distintos. Esto hace que se desperdicie espacio de almacenamiento, además de provocar la falta de consistencia de datos. En los sistemas de bases de datos todos estos ficheros están integrados, por lo que no se almacenan varias copias de los mismos datos. Sin embargo, en una base de datos no se puede eliminar la redundancia completamente, ya que en ocasiones es necesaria para modelar las relaciones entre los datos, o bien es necesaria para mejorar las prestaciones.
- ✓ **Consistencia de datos.** Eliminando o controlando las redundancias de datos se reduce en gran medida el riesgo de que haya inconsistencias. Si un dato está almacenado una sola vez, cualquier actualización se debe realizar sólo una vez, y

está disponible para todos los usuarios inmediatamente. Si un dato está duplicado y el sistema conoce estaredundancia, el propio sistema puede encargarse de garantizar que todas las copias se mantienen consistentes. Desgraciadamente, no todos los SGBD de hoy en día se encargan de mantener automáticamente la consistencia.

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

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

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

2.1.6.2. Ventajas por la existencia del SGBD

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

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

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

- ✓ **Mejora en la productividad.** El SGBD proporciona muchas de las funciones estándar que el programador necesita escribir en un sistema de ficheros. A nivel básico, el SGBD proporciona todas las rutinas de manejo de ficheros típicas de los programas de aplicación. El hecho de disponer de estas funciones permite al programador centrarse mejor en la función específica requerida por los usuarios, sin tener que preocuparse de los detalles de implementación de bajo nivel. Muchos SGBD también proporcionan un entorno de cuarta generación consistente en un conjunto de herramientas que simplifican, en gran medida, el desarrollo de las aplicaciones que acceden a la base de datos. Gracias a estas herramientas, el programador puede ofrecer una mayor productividad en un tiempo menor.

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

- ✓ **Aumento de la concurrencia.** En algunos sistemas de ficheros, si hay varios usuarios que pueden acceder simultáneamente a un mismo fichero, es posible que

el acceso interfiera entre ellos de modo que se pierda información o, incluso, que se pierda la integridad. La mayoría de los SGBD gestionan el acceso concurrente a la base de datos y garantizan que no ocurran problemas de este tipo.

- ✓ **Mejora en los servicios de copias de seguridad y de recuperación ante fallos.** Muchos sistemas de ficheros dejan que sea el usuario quien proporcione las medidas necesarias para proteger los datos ante fallos en el sistema o en las aplicaciones. Los usuarios tienen que hacer copias de seguridad cada día, y si se produce algún fallo, utilizar estas copias para restaurarlos. En este caso, todo el trabajo realizado sobre los datos desde que se hizo la última copia de seguridad se pierde y se tiene que volver a realizar. Sin embargo, los SGBD actuales funcionan de modo que se minimiza la cantidad de trabajo perdido cuando se produce un fallo.[6]

2.1.7. Componentes de un Sistema de Gestión de Base de Datos

Un Sistema de Gestión de Base de Datos se divide en módulos que tratan cada una de las responsabilidades del sistema general. Los componentes funcionales de un SGBD incluyen. (Ver Figura III.2)

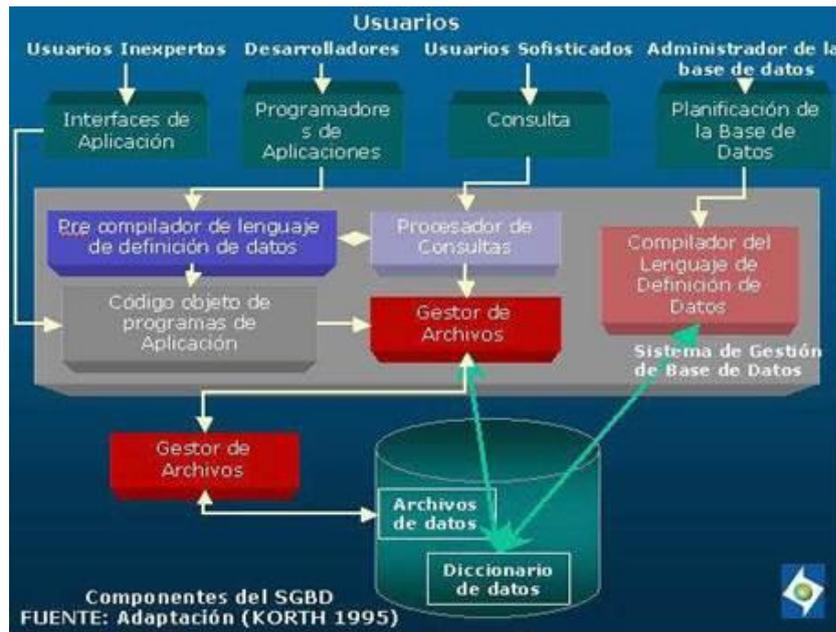


Figura II.2 Componentes de un SGBD

Fuente: <http://basessss.galeon.com/s1.htm>

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

- **Pre compilador del Lenguaje de Manipulación de datos DML.** Convierte las sentencias en DML incorporadas en un programa de aplicación en llamadas normales a procedimientos en el lenguaje principal.
- **Compilador del Lenguaje de Definición de datos DDL.** Convierte en sentencias en DDL en un conjunto de tablas metadatos o “datos sobre datos”.
- **Gestor del diccionario de datos.** Almacena metadatos sobre la estructura de la base de datos.[4]

2.1.8. Arquitectura de un Sistema de Base de Datos

Uno de los objetivos de un sistema de base de datos es proporcionar una visión lo más abstracta posible de la información, es decir, ocultar detalles referentes a la forma en que los datos están organizados y almacenados. La arquitectura más estándar y, por tanto, la más utilizada, es la que hace una división en niveles de la base de datos. La arquitectura de tres niveles propuesta por ANSI/SPARC (Comité de Estandarización de Requerimientos y Planificación del Instituto Nacional de Estandarización Americano) considera tres niveles según la perspectiva desde la que sea vista la información. El nivel interno es el nivel más bajo de abstracción, donde se describe la información en función del sistema en que se implantará la base de datos. Por encima, se encuentra el conceptual, que representa a alto nivel toda la información de la base de datos independientemente de la máquina en que vaya a utilizarse. El más abstracto de los niveles es el externo, que gestiona la información desde el punto de vista individual de cada usuario, grupo de usuarios, programador o grupo de programadores.

- **Nivel Interno o físico.** El más cercano al almacenamiento físico, es decir, tal y como están almacenados en el ordenador. Describe la estructura física de la BD mediante un esquema interno. Este esquema se especifica con un modelo físico y describe los detalles de cómo se almacenan físicamente los datos: los archivos que contienen la información, su organización, los métodos de acceso a los registros, los tipos de registros, la longitud, los campos que los componen, los campos clave, etcétera.
- **Nivel Conceptual.** Describe la estructura de toda BD para un grupo de usuarios mediante un esquema conceptual. Este esquema describe las entidades, atributos, relaciones, operaciones de los usuarios y restricciones, ocultando los detalles de las estructuras físicas de almacenamiento. Representa la información contenida en la BD.
- **Nivel Externo.** Es el conjunto de percepciones individuales de la base de datos. Cada visión individual se denomina sub-esquema o vista. Un sub-esquema podrá ser compartido por varios usuarios, y cada usuario tendrá la posibilidad de acceder a distintos sub-esquemas. Al crear un sub-esquema, es posible mezclar campos de distintos registros, omitir campos, cambiar el orden de los campos, añadir campos que puedan ser calculados a partir de los descritos en el esquema conceptual, etc.

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

individual de un usuario o de un grupo de usuarios. En la Figura III.3 se representan los niveles de abstracción de la arquitectura ANSI.[18]

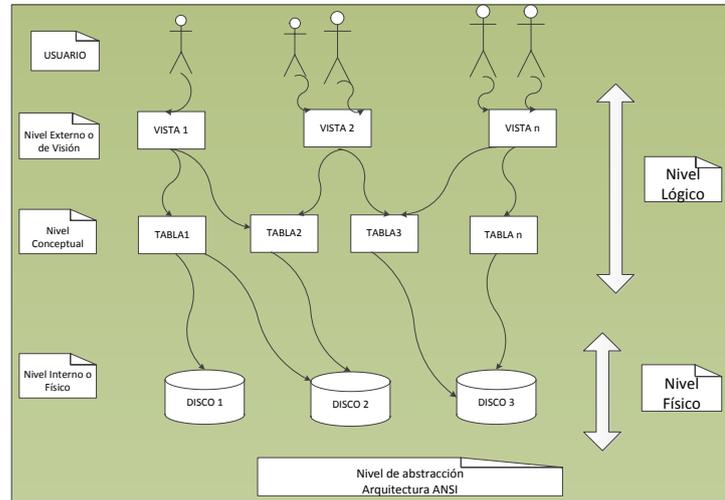


Figura II.3 Niveles de abstracción de la arquitectura ANSI

Fuente: Edgar Cuenca

2.2. Productos SGBD disponibles en el mercado

2.2.1. Sistemas libres

- PostgreSQL (<http://www.postgresql.org> Postgresql) Licencia BSD
- Firebird basada en la versión 6 de InterBase, InitialDeveloper's PUBLIC LICENSE Version 1.0.
- SQLite (<http://www.sqlite.org> SQLite) Licencia Dominio Público
- DB2 Express-C (<http://www.ibm.com/software/data/db2/express/>)
- Apache Derby (<http://db.apache.org/derby/>)
- MariaDB (<http://mariadb.org/>)
- MySQL (<http://dev.mysql.com/>)

- Drizzle (<http://www.drizzle.org/>)

2.2.2. Sistemas no libres

- MySQL: Licencia Dual, depende del uso. No se sabe hasta cuándo permanecerá así, ya que ha sido comprada por Oracle. Sin embargo, existen 2 versiones: una gratuita que sería equivalente a la edición "express" SQL server de Microsoft Windows, y otra más completa de pago.
- Advantage Database
- dBase
- FileMaker
- Fox Pro
- gsBase
- IBM DB2: Universal Database (DB2 UDB)
- IBM Informix
- Interbase de CodeGear, filial de Borland
- MAGIC
- Microsoft Access
- Microsoft SQL Server
- NexusDB
- Open Access
- Oracle
- Paradox
- PervasiveSQL

- Progress (DBMS)
- Sybase ASE
- Sybase ASA
- Sybase IQ
- WindowBase
- IBM IMS Base de Datos Jerárquica
- CA-IDMS

2.2.3. Sistemas no libres y gratuitos

- Microsoft SQL Server Compact Edition Basica
- Sybase ASE Express Edition para Linux (edición gratuita para Linux)
- Oracle Express Edition 10 (solo corre en un servidor, capacidad limitada). [3]

2.3. Herramientas de administración de bases de datos

2.3.1. Administración de bases de datos

Rápidamente surgió la necesidad de contar con un sistema de administración para controlar tanto los datos como los usuarios. La administración de bases de datos se realiza con un sistema llamado DBMS Data Base Management System(Sistema de administración de bases de datos).

El DBMS es un conjunto de servicios aplicaciones de software para administrar bases de datos, que permite:

- un fácil acceso a los datos
- el acceso a la información por parte de múltiples usuarios

- la manipulación de los datos encontrados en la base de datos (insertar, eliminar, editar)
- El DBMS puede dividirse en tres subsistemas:
 - ✓ **El sistema de administración de archivos:**
Para almacenar información en un medio físico
 - ✓ **El DBMS interno:**
Para ubicar la información en orden
 - ✓ **El DBMS externo:**
Representa la interfaz del usuario

Los DBMS principales

Los principales sistemas de administración de bases de datos son:

- Borland Paradox
- Filemaker
- IBM DB2
- Ingres
- Interbase
- Microsoft SQL server
- Microsoft Access
- Microsoft Foxpro
- Oracle
- Sybase
- My SQL

- PostgreSQL
- mSQL
- SQL Server 11. [10]

2.3.2. Postgresql

PostgreSQL es un sistema de gestión de base de datos relacional orientada a objetos y libre, publicado bajo la licencia BSD. Como muchos otros proyectos de código abierto, el desarrollo de PostgreSQL no es manejado por una empresa y/o persona, sino que es dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, altruista, libre y/o apoyada por organizaciones comerciales. Dicha comunidad es denominada el PGDG (PostgreSQL Global DevelopmentGroup).

2.3.2.1.Nombre del producto

El uso de caracteres en mayúscula en el nombre PostgreSQL puede confundir a algunas personas a primera vista. Las distintas pronunciaciones de "SQL" pueden llevar a confusión. Los desarrolladores de PostgreSQL lo pronuncian /po:stgʁɛskju: ɛl/. Es también común oír abreviadamente como simplemente "Postgres", el que fue su nombre original. Debido a su soporte del estándar SQL entre la mayor parte de bases de datos relacionales, la comunidad consideró cambiar el nombre al anterior Postgres. Sin embargo, el PostgreSQLCoreTeam anunció en 2007 que el producto seguiría llamándose PostgreSQL. El nombre hace referencia a los orígenes del proyecto como la base de datos "post-Ingres", y los autores originales también desarrollaron la base de datos Ingres.

2.3.2.2.Evolución

PostgreSQL ha tenido una larga evolución, la cual se inicia en 1982 con el proyecto Ingres en la Universidad de Berkeley. Este proyecto, liderado por Michael Stonebraker, fue uno de los primeros intentos en implementar un motor de base de datos relacional. Después de haber trabajado un largo tiempo en Ingres y de haber tenido una experiencia comercial con él mismo, Michael decidió volver a la Universidad en 1985 para trabajar en un nuevo proyecto sobre la experiencia de Ingres, dicho proyecto fue llamado post-ingres o simplemente POSTGRES.

El proyecto post-ingres pretendía resolver los problemas con el modelo de base de datos relacional que habían sido aclarados a comienzos de los años 1980. El principal de estos problemas era la incapacidad del modelo relacional de comprender "tipos", es decir, combinaciones de datos simples que conforman una única unidad. Actualmente estos son llamados objetos. Se esforzaron en introducir la menor cantidad posible de funcionalidades para completar el soporte de tipos. Estas funcionalidades incluían la habilidad de definir tipos, pero también la habilidad de describir relaciones las cuales hasta ese momento eran ampliamente utilizadas pero mantenidas completamente por el usuario. En Postgresql la base de datos comprendía las relaciones y podía obtener información de tablas relacionadas utilizando reglas. Postgresql usó muchas ideas de Ingres pero no su código.

La siguiente lista muestra los hitos más importantes en la vida del proyecto Postgresql.

- 1986: se publicaron varios papers que describían las bases del sistema.
- 1988: ya se contaba con una versión utilizable.
- 1989: el grupo publicaba la versión 1 para una pequeña comunidad de usuarios.

- 1990: se publicaba la versión 2 la cual tenía prácticamente reescrito el sistema de reglas.
- 1991: publicación de la versión 3, esta añadía la capacidad de múltiples motores de almacenamiento.
- 1993: crecimiento importante de la comunidad de usuarios, la cual demandaba más características.
- 1994: después de la publicación de la versión 4, el proyecto terminó y el grupo se disolvió.

Después de que el proyecto POSTGRES terminara, dos graduados de la universidad, Andrew Yu y JollyChen, comenzaron a trabajar sobre el código de POSTGRES, esto fue posible dado que POSTGRES estaba licenciado bajo la BSD, y lo primero que hicieron fue añadir soporte para el lenguaje SQL a POSTGRES, dado que anteriormente contaba con un intérprete del lenguaje de consultas QUEL (basado en Ingres), creando así el sistema al cual denominaron Postgres95.

Para el año 1996 se unieron al proyecto personas ajenas a la Universidad como Marc Fournier de Hub.Org NetworkingServices, Bruce Momjian y Vadim B. Mikheev quienes proporcionaron el primer servidor de desarrollo no universitario para el esfuerzo de desarrollo de código abierto y comenzaron a trabajar para estabilizar el código de Postgres95.

En el año 1996 decidieron cambiar el nombre de Postgres95 de tal modo que refleje la característica del lenguaje SQL y lo terminaron llamando PostgreSQL, cuya primera versión de código abierto fue lanzada el 1 de agosto de 1996. La primera versión formal

de PostgreSQL (6.0) fue liberada en enero de 1997. Desde entonces, muchos desarrolladores entusiastas de los motores de base de datos se unieron al proyecto, coordinaron vía Internet y entre todos comenzaron a incorporar muchas características al motor.

Aunque la licencia permitía la comercialización de PostgreSQL, el código no se desarrolló en principio con fines comerciales, algo sorprendente considerando las ventajas que PostgreSQL ofrecía. La principal derivación se originó cuando Paula Hawthorn (un miembro del equipo original de Ingres que se pasó a Postgres) y Michael Stonebraker conformaron IllustraInformation Technologies para comercializar Postgres.

En 2000, ex inversionistas de Red Hat crearon la empresa Great Bridge para comercializar PostgreSQL y competir contra proveedores comerciales de bases de datos. Great Bridge auspició a varios desarrolladores de PostgreSQL y donó recursos de retorno a la comunidad, pero a fines de 2001 cerró debido a la dura competencia de compañías como Red Hat y pobres condiciones del mercado.

En 2001, CommandPrompt, Inc. lanzó MammothPostgreSQL, la más antigua distribución comercial de PostgreSQL. Continúa brindando soporte a la comunidad PostgreSQL a través del auspicio de desarrolladores y proyectos, incluyendo PL/Perl, PL/php y el alojamiento de proyectos de comunidades como PostgreSQLBuild Farm.

En enero de 2005, PostgreSQL recibió apoyo del proveedor de base de datos Pervasive Software, conocido por su producto Btrieve que se utilizaba en la plataforma Novell

Netware. Pervasive anunció soporte comercial y participación comunitaria y logró algo de éxito. Sin embargo, en julio de 2006 dejó el mercado de soporte de PostgreSQL.

A mediados de 2005 otras dos compañías anunciaron planes para comercializar PostgreSQL con énfasis en nichos separados de mercados. EnterpriseDB añadió funcionalidades que le permitían a las aplicaciones escritas para trabajar con Oracle ser más fáciles de ejecutar con PostgreSQL. Greenplum contribuyó mejoras directamente orientadas a aplicaciones de Data Warehouse e Inteligencia de negocios, incluyendo el proyecto BizGres.

En octubre de 2005, John Loiacono, vicepresidente ejecutivo de software en Sun Microsystems comentó: "No estamos yendo tras el OEM de Microsoft pero estamos viendo a PostgreSQL ahora", aunque no se dieron especificaciones en ese momento. Para noviembre de 2005, Sun Solaris 10 (lanzamiento 6/06) incluía PostgreSQL.

En agosto de 2007 EnterpriseDB anunció el PostgresResource Center y EnterpriseDBPostgres, diseñados para ser una completamente configurada distribución de PostgreSQL incluyendo muchos módulos contribuidos y agregados. EnterpriseDBPostgres fue renombrado Postgres Plus en marzo de 2008.

El proyecto PostgreSQL continúa haciendo lanzamientos principales anualmente y lanzamientos menores de reparación de bugs, todos disponibles bajo la licencia BSD, y basados en contribuciones de proveedores comerciales, empresas a portantes y programadores de código abierto mayormente.

Existe también una gran comunidad de usuarios, programadores y administradores que colaboran activamente en numerosos aspectos y actividades relacionadas con el proyecto. Informes y soluciones de problemas, test, comprobación del funcionamiento, aportaciones de nuevas ideas, discusiones sobre características y problemas, documentación y fomento de PostgreSQL son solo algunas de las actividades que la comunidad de usuarios realiza.

En los últimos años los trabajos de desarrollo se han concentrado mucho en la velocidad de proceso y en características demandadas en el mundo empresarial. En el gráfico a continuación se puede ver cuando las diferentes versiones de PostgreSQL han visto la luz y las principales características en las que se ha centrado el desarrollo.



Figura II.4 Versiones de PostgreSQL

Fuente: http://www.postgresql.org.es/sobre_postgresql

Durante los años de existencia del Proyecto PostgreSQL, el tamaño del mismo, tanto en número de desarrolladores, como en números de línea de código, funciones y complejidad del mismo ha ido aumentando año a año. En el siguiente gráfico se tiene una gráfica con la evolución del número de líneas de código en cada versión de PostgreSQL.

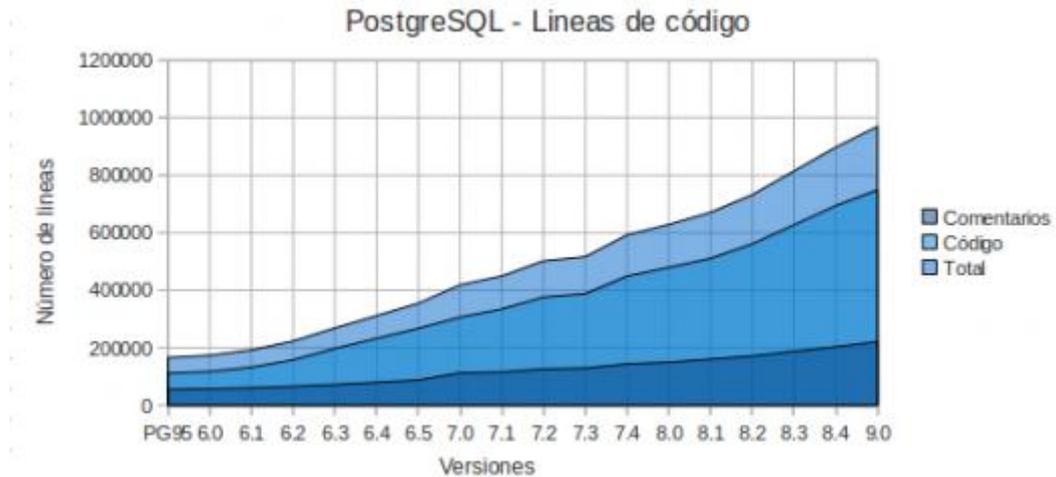


Figura II.5 Líneas Código

Fuente: http://www.postgresql.org.es/sobre_postgresql

Los datos de este gráfico están generados con CLOC. Contabilizado como líneas de código a todas las líneas de código en diferentes lenguajes, más comentarios, menos líneas en blanco. Los ficheros HTML y CSS no se cuentan como código. [15]

2.3.2.3.Ciclo de vida (EOL) y soporte

El Proyecto PostgreSQL tiene como objetivo mantener y soportar cada versión de PostgreSQL durante 5 años desde el momento de su lanzamiento, (Ver Figura II.6). A continuación se tienen un resumen del ciclo de vida de las diferentes versiones de PostgreSQL. [12]

Versión	Versión menor	Soportada	Lanzamiento	Soporte
9.2	9.2.0	Si	Sep 2012	Sep 2017
9.1	9.1.5	Si	Sep 2011	Sep 2016
9.0	9.0.9	Si	Sep 2010	Sep 2015
8.4	8.4.13	Si	Jul 2009	Jul 2014
8.3	8.3.20	Si	Feb 2008	Feb 2013
8.2	8.2.23	No	Dic 2006	Dic 2011
8.1	8.1.23	No	Nov 2005	Nov 2010
8.0	8.0.26	No	Ene 2005	Oct 2010
7.4	7.4.30	No	Nov 2003	Oct 2010
7.3	7.3.21	No	Nov 2002	Nov 2007
7.2	7.2.8	No	Feb 2002	Feb 2007
7.1	7.1.3	No	Abr 2001	Abr 2006
7.0	7.0.3	No	May 2000	May 2005
6.5	6.5.3	No	Jun 1999	Jun 2004
6.4	6.4.2	No	Oct 1998	Oct 2003
6.3	6.3.2	No	Mar 1998	Mar 2003

Figura II.6 Ciclo de vida Postgresql

Fuente: http://www.postgresql.org.es/sobre_postgresql

2.3.2.4.Arquitectura depostgresql

Es un SGDB Objeto Relacional basado en una arquitectura Cliente/Servidor, conocida como “Proceso por Usuario”. En Campos al 2005 se muestra la arquitectura que usa Postgresql: un programa servidor llamado postgres y diversos programas cliente como Pgaccess (un cliente gráfico) y Psql (un cliente en modo texto). El proceso servidor postgres puede atender exclusivamente a un solo cliente, es decir, se requiere de tantos procesos servidores postgres como clientes haya. El proceso postmaster es el encargado de ejecutar un nuevo servidor para cada cliente que solicite una conexión. Se llama sitio al equipo anfitrión (host) que almacena un conjunto de bases de datos Postgresql. En un sitio se ejecuta un solo proceso postmaster y múltiples procesos postgres. Los clientes pueden ejecutarse en el mismo sitio o en equipos remotos conectados por TCP/IP. Es posible restringir el acceso a usuarios o a direcciones IP modificando las opciones del archivo pg_hba.conf, que se encuentra en /etc/postgresql/pg_hba.conf.

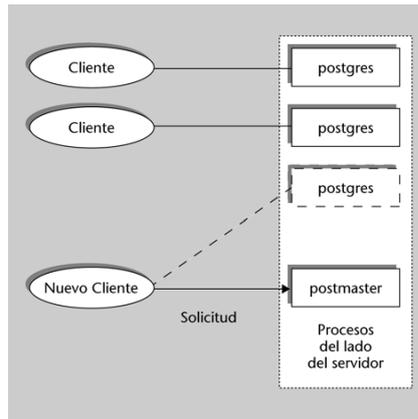


Figura II.7 Arquitectura de PostgreSQL

Fuente: <http://www.dataprix.com/72-arquitectura-postgresql>

2.3.2.5. Componentes más importantes en un sistema PostgreSQL

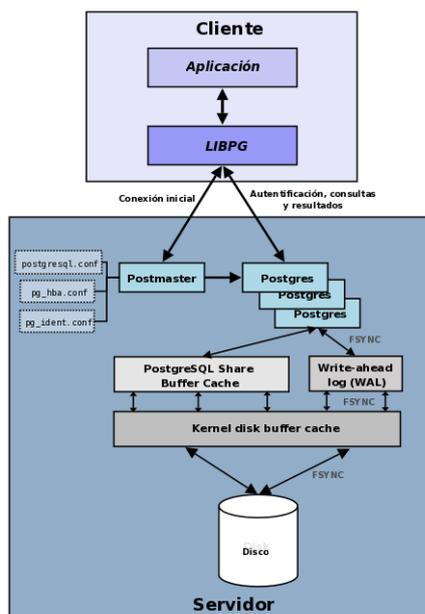


Figura II.8 Componentes Postgresql

Fuente: http://www.postgresql.org.es/sobre_postgresql

Aplicación cliente: Esta es la aplicación cliente que utiliza PostgreSQL como administrador de bases de datos. La conexión puede ocurrir vía TCP/IP ó sockets locales.

Demonio postmaster: Este es el proceso principal de PostgreSQL. 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 (recuperación de tipo REDO)

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.[5]

2.3.2.6. Características de Postgresql

- **Alta concurrencia**

PostgreSQL permite que mientras un proceso escribe en una tabla, otros accedan a la misma tabla sin necesidad de bloqueos. Cada usuario obtiene una visión consistente de lo último a lo que se le hizo commit. Esta estrategia es superior al uso de bloqueos

portable o por filas común en otras bases, eliminando la necesidad del uso de bloqueos explícitos.

- **Amplia variedad de tipos nativos**

PostgreSQL provee nativamente soporte para:

- ✓ Números de precisión arbitraria.
- ✓ Texto de largo ilimitado.
- ✓ Figuras geométricas (con una variedad de funciones asociadas).
- ✓ Direcciones IP (IPv4 e IPv6).
- ✓ Bloques de direcciones estilo CIDR.
- ✓ Direcciones MAC.
- ✓ Arrays.

Adicionalmente los usuarios pueden crear sus propios tipos de datos, los que pueden ser por completo indexables gracias a la infraestructura GiST de PostgreSQL. Algunos ejemplos son los tipos de datos GIS creados por el proyecto PostGIS.

Claves ajenas (Llaves ajenas o Claves Foráneas)

Disparadores (triggers): Un disparador define como una acción específica que se realiza de acuerdo a un evento, cuando éste ocurra dentro de la base de datos.

En PostgreSQL esto significa la ejecución de un procedimiento almacenado basado en una determinada acción sobre una tabla específica. Ahora todos los disparadores se definen por seis características:

1. El nombre del disparador o trigger
2. El momento en que el disparador debe arrancar
3. El evento del disparador deberá activarse sobre...
4. La tabla donde el disparador se activará
5. La frecuencia de la ejecución
6. La función que podría ser llamada

Combinando estas seis características, PostgreSQL permite crear una amplia funcionalidad a través de su sistema de activación de disparadores, como son:

- ✓ Vistas.
 - ✓ Integridad transaccional.
 - ✓ Herencia de tablas.
 - ✓ Tipos de datos y operaciones geométricas.
-
- **Soporte para transacciones distribuidas.** Permite a PostgreSQL integrarse en un sistema distribuido formado por varios recursos (p.ej, una base de datos PostgreSQL, otra Oracle, una cola de mensajes IBM MQ JMS y un ERP SAP) gestionado por un servidor de aplicaciones donde el éxito ("commit") de la transacción global es el resultado del éxito de las transacciones locales.
 - **Funciones**
Bloques de código que se ejecutan en el servidor. Pueden ser escritos en varios lenguajes, con la potencia que cada uno de ellos da, desde las operaciones

básicas de programación, tales como bifurcaciones y bucles, hasta las complejidades de la programación orientada a objetos o la programación funcional.

Los disparadores son funciones enlazadas a operaciones sobre los datos.

- **Lenguajes para Posgresql**

- ✓ PL/PgSQL (similar al PL/SQL de oracle).
- ✓ C.
- ✓ C++.
- ✓ Java PL/Java web.
- ✓ PL/Perl.
- ✓ plPHP.
- ✓ PL/Python.
- ✓ PL/Ruby.
- ✓ PL/sh.
- ✓ PL/Tcl.
- ✓ PL/Scheme.

- **Lenguaje para aplicaciones estadísticas R por medio de PL/R.**

PostgreSQL soporta funciones que retornan "filas", donde la salida puede tratarse como un conjunto de valores que pueden ser tratados igual a una fila retornada por una consulta.

Las funciones pueden ser definidas para ejecutarse con los derechos del usuario ejecutor o con los derechos de un usuario previamente definido. El concepto de funciones, en otros DBMS, son muchas veces referidas como procedimientos almacenados. [9]

2.3.2.7.Límites de PostgreSQL

Algunos de los límites que podría tener PostgreSQL son:

Tabla III. Límites de Postgresql

Límite	Valor
Máximo tamaño base de dato	Ilimitado (Depende del Sistema de almacenamiento)
Máximo tamaño de tabla	32 TB
Máximo tamaño de fila	1.6 GB
Máximo tamaño de campo	1 GB
Máximo número de filas por tabla	Ilimitado
Máximo número de columnas por tabla	250 -1600 (dependiendo del tipo)
Máximo número de índices por tabla	Ilimitado

CAPÍTULO III

ANÁLISIS COMPARATIVO DE LAS HERRAMIENTAS GRÁFICAS PARA POSTGRESQL

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

El presente capítulo tiene como objetivo fundamental, analizar y evaluar los Sistemas Gestores de Base de Datos libres para Postgresql, PgAdminIII, PHPpgAdminyEMS SQL Lite FOR POSTGRESQL para seleccionar cual es el mejor, para ello se tomarán en cuenta factores como, tiempo de respuesta en realizar la instalación, los procesos, características que ofrecen los SGBD, entre otros.

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

2.1. Recursos Hardware y Software utilizados

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

2.1.1. Recursos Hardware

La computadora utilizada en las pruebas tiene las siguientes características:

- ✓ Procesador: Intel Pentium Dual-Core, 2.50 GHz
- ✓ Memoria RAM de 4 GB
- ✓ Disco duro de 300 GB

2.1.2. Recursos Software

Para el desarrollo de las pruebas se requiere instalar:

- ✓ Gestor de Base de Datos Postgresql
- ✓ Postgresql GUI Tools
- ✓ Windows 7

2.2. Determinación de las Herramientas Postgresql a comparar

En la actualidad la existencia de varias Herramientas para Postgresql, se ha realizado una investigación previa de estas, por sitios publicados en el Internet se han elegido las tres Herramientas por las razones que se explican a continuación.

- En el portal de Guía de comunidad para herramientas GUI para postgresql “http://wiki.postgresql.org/wiki/Guía_de_la_Comunidad_para_las_herramientas_GUI_de_PostgreSQL” han realizado una clasificación de todas las

herramientas open source disponibles clasificándolas de la siguiente manera.

[14]

- ✓ pgAdmin III
 - ✓ PGAccess
 - ✓ phpPgAdmin
 - ✓ EMS SQL MANAGER Lite POSTGRESQL
 - ✓ OpenOffice.org
 - ✓ Herramientas gráficas para la Red Hat Database, Administrador de RHDB y Explain Visual
 - ✓ Xpg: Cliente de Java para PostgreSQL
 - ✓ Mergeant
 - ✓ TOra, una herramienta de Oracle con algún soporte para PostgreSQL
 - ✓ KNoda
 - ✓ PGINhaler
 - ✓ SQuirreL
 - ✓ AnySQL Maestro
 - ✓ PostgreSQL PHP Generator
 - ✓ WaveMaker Ajax GUI Design Tool
- En el portal “<http://ibiblio.org/pub/linux/docs/LuCaS/Postgresql-es/web/navegable/Howto/PostgreSQL-COMO-11.html>” describe las Herramientas de Gestión de PostgreSQL de la siguiente manera. [10]
 - ✓ pgAdmin III
 - ✓ phpPgAdmin

- ✓ EMS SQL MANAGER Lite POSTGRESQL
- ✓ PGACCESS
- ✓ ISQL
- ✓ MPMGR

- Una encuesta por medio de la red social G+ operada por Google sobre herramientas para postgresql a estudiantes, y desarrolladores de bases de datos en el sitio, <https://docs.google.com/spreadsheet/viewform?fromEmail=true&formkey=dFVTaVRibkhYZkJqaGhkMXdPQ2VsRIE6MQ>", los resultados arrojados fueron los siguientes.

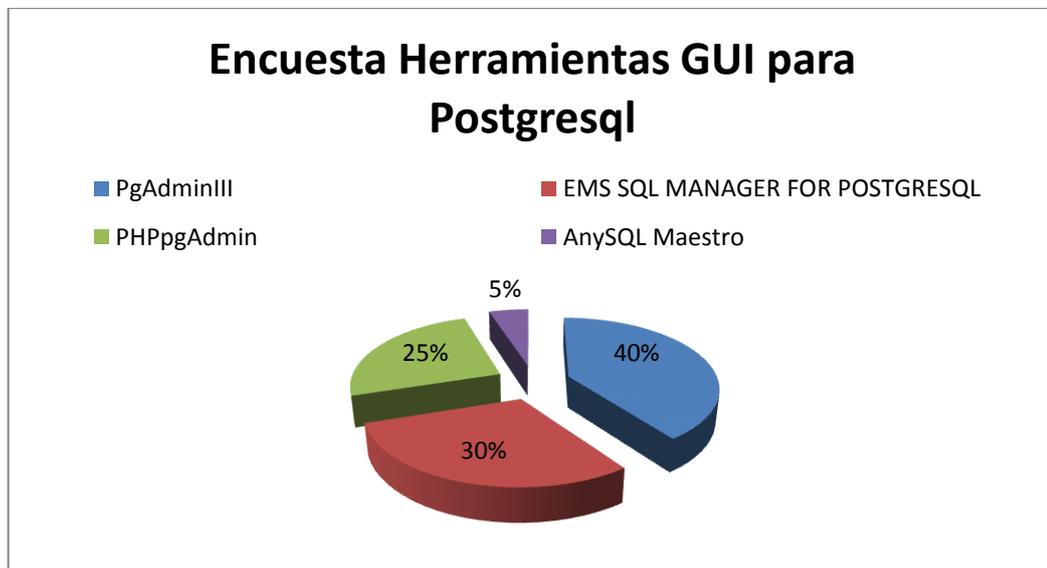


Figura III. 9 Valorización de la encuesta

- ✓ PgAdminIII
- ✓ EMS MANAGER FOR POSTGRESQL
- ✓ PHPPgAdmin
- ✓ AnySQLMaestro

Para la realización de este análisis comparativo se han seleccionado las siguientes PgAdminIII, PHP PGADMIN, EMS MANAGER LITE FOR POSTGRESQL por los siguientes motivos:

- Las tres Herramientas gozan de una popularidad entre los desarrolladores y diseñadores por las publicaciones de diferentes sitios web.
- Los resultados de la encuesta que se realizó por medio del G+, nos dio como veredicto a estas Herramientas más usados por desarrolladores y estudiantes informáticos en nuestro medio.
- Tienen una excelente documentación en línea y comunidad de usuarios.
- La factibilidad de uso es muy buena ya que se basan en el modelo vista controlador.
- La optimización de recursos es fundamental para que exista un mejor desempeño de los sitios web.
- El manejo y la abstracción de base datos con postgresql es realmente bueno.

2.3. Análisis de las Herramientas seleccionadas

2.3.1. PGADMIN III

Es la herramienta Open Source de administración por excelencia para bases de datos PostgreSQL derivados EnterpriseDBPostgres Plus Advanced Server y GreenplumDatabase, incluye: Interfaz administrativa gráfica, Herramienta de consulta SQL, Editor de código procedural, Agente de planificación SQL/shell/batch. PGAdminIII se diseña para responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas.

La interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración. Está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, FreeBSD, Mac OSX y Solaris. [16]

2.3.1.1. Características

- Soporte completo para UNICODE.
- Edición rápida de consultas y datos multihilo
- Soporte para todos los tipos de objetos de PostgreSQL.
- PgAdminIII es empaquetado con el instalador de Windows, puede ser usado como un cliente para administrar un servidor remoto en otro sistema operativo
- Es el centro de administración para las bases de datos PostgreSQL. Incluye una interfaz gráfica de administración, una herramienta para el trabajo con SQL, un editor de código de procedimientos y funciones, y mucho más. PgAdminIII está diseñado para darle respuesta de las necesidades de la mayoría de los usuarios,

desde la escritura de consultas simples en SQL hasta el desarrollo de bases de datos complejas. La interfaz gráfica soporta todas las características presentes de PostgreSQL y se puede hacer la administración fácilmente. Está disponible en más de 30 lenguajes y para varios sistemas operativos.

3.1.1.2.Requisitos de Instalación

- PgAdminIII es una interfaz de administración para gestionar bases de datos PostgreSQL, es multiplataforma y puede funcionar bajo GNU / Linux, FreeBSD, Mac, Windows y Solaris.
- Puede instalarse en un Sistema operativo virtualizado.
- En Linux realizar la descarga de los ficheros necesarios para la instalación de PostgreSQL disponer de conexión a Internet.
- Tarjeta de red y conexión de red, a ser posible con IP fija (no DHCP). [11]

3.1.2. PHP PGADMIN

Es un conjunto de scripts PHP para administrarPostgreSQL a través de la WWW, actualmente se puede crear y eliminar bases de datos, crear, copiar, eliminar y alterar tablas / vistas / secuencias / funciones, editar y añadir campos, ejecutar cualquier sentencia SQL, incluso por lotes, gestionar las claves en campos, cargar archivos de texto en las tablas, crear y leer tablas, gestionar una base de datos única, administrar usuarios postgresql. [17]

3.1.2.1. Características

- Administrar varios servidores
- Soporte para PostgreSQL 7.4.x, 8.0.x, 8.1.x, 8.2.x, 8.3.x, 8.4.x, 9.0.x
- Administrar todos los aspectos de:
 - Usuarios y grupos
 - Bases de datos
 - Esquemas
 - Las tablas, índices, restricciones, triggers, reglas y privilegios
 - Vistas, secuencias y funciones
 - Objetos avanzados
 - Informes
- Fácil manipulación de datos:
 - Examinar tablas, vistas e informes
 - Ejecutar SQL arbitrario
 - Seleccionar, insertar, actualizar y eliminar
 - Volcado de datos de la tabla en una variedad de formatos: SQL, COPY, XML, XHTML, CSV, pestañas, pg_dump
 - Importación de secuencias de comandos SQL, copiar datos, XML, CSV y pestañas
- Apoya la Slony maestro-esclavo motor de replicación
- Excelente soporte de idiomas:
 - Disponible en 27 idiomas

- No hay conflictos de codificación. Editar rusos utilizando una interfaz de japonés.
- Basado en Navegadores Web. [12]

3.1.2.2.Requisitos de instalación

- Requiere un servidor web
- Paquetes de Apache y PHP
- Fácil y rápida instalación

3.1.3. EMS SQL MANAGER FOR POSTGRESQL

EMS SQL Manager forPostgreSQL es una herramienta de alto rendimiento para la administración de bases de datos PostgreSQL y el desarrollo. Funciona con cualquier versión de PostgreSQL hasta la más reciente y soporta las últimas características de PostgreSQL incluyendo la exclusión constriñe, "cuándo", cláusula de disparadores, funciones de tabla, y otros. SQL Manager forPostgreSQL ofrece una amplia gama de herramientas de bases de datos de gran alcance tales como diseñador visual de base de datos para crear la base de datos PostgreSQL en unos pocos clics, Visual QueryBuilder para crear consultas complejas PostgreSQL, editor BLOB potente y muchas características más útiles para la administración de PostgreSQL eficiente. SQL Manager forPostgreSQL tiene una interfaz de estado-of-the-art gráfica de usuario con bien descritos asistente sistema, de modo claro en su uso que ni un principiante no debe confundirse con ella. [1]

3.1.3.1. Características

- Soporte completo de PostgreSQL hasta la versión 9.1
- Soporte de datos UTF8
- Soporte de colaciones
- Soporte de extensiones
- Soporte de vista desencadenantes
- Soporte de tablas sin talar
- Excelentes herramientas visuales y de texto para elaboración de consultas
- Nuevo estado de la técnica interfaz gráfica de usuario
- Rapid base de datos y la navegación
- Avanzadas herramientas de manipulación de datos
- Explorador de base de datos mejorada para facilitar la gestión de todos los objetos de PostgreSQL
- Gestión de seguridad efectiva
- Exportación de datos impresionante y capacidades de importación
- Potente diseñador visual de base de datos
- Fácil de usar asistentes para realizar tareas de mantenimiento de PostgreSQL
- Conexión mediante el reenvío de puerto local a través del túnel SSH
- El acceso al servidor PostgreSQL a través del protocolo HTTP
- Base de datos de gestión del cambio
- Diseñador de informes con claros en el uso de la construcción asistente de informes. [2]

3.1.3.2.Requisitos de Instalación

- 300 megahercios (MHz), 600 MHz o más rápido recomendado
- Microsoft Windows NT4 con Service Pack 4 o posterior, Microsoft Windows 2000, Microsoft Windows 2000 Server, Microsoft Windows XP, Microsoft Windows Server 2003, Microsoft Windows 2008 Server, Microsoft Windows Vista, Microsoft Windows 7
- 64 MB de RAM o más, 128 MB o más recomendado
- 50 MB de espacio disponible en HD para la instalación del programa
- Super VGA (800x600) o de mayor resolución de adaptador de vídeo y monitor;
Super VGA (1024x768) o de mayor resolución recomendado por Microsoft
Mouse o un dispositivo señalador compatible
- Las versiones compatibles del servidor PostgreSQL: desde 7,3 hasta 9,0

3.2. Criterios técnicos contemplados para ejecución de pruebas en las Herramientas

- Uso de un mismo equipo para todas las pruebas en los 3 gestores, con la finalidad de mantener los recursos físicos y lógicos. Las características de hardware del PC son :
 - Procesador : Intel Pentium Dual CPU T6600
 - RAM : 4.0 GB
 - HD: 320 GB. Espacio Libre en disco: 60 GB
 - Sistema Operativo Windows 7 Ultimate 64Bit
 - Reiniciar el equipo antes de cada prueba para liberar la memoria.

3.3. **Determinación de parámetros de comparación**

Los parámetros han sido determinados tomando en cuenta la manera como las herramientas de administración de bases de datos postgresql puede mejorar la comunicación con los diseñadores de bases de datos y así mejorar el entorno de trabajo, de forma fácil, simple y rápida de manera que mejore la productividad y de esta manera determinar la mejor Herramienta para Gestionar Base de Datos.

- **Instalación**

- ✓ Tiempo en Completar Instalación
- ✓ Pasos de Configuración
- ✓ Nivel de Control de Acceso
- ✓ Nivel de ayuda en los procesos de instalación

- **Seguridad**

- ✓ Tipo de autenticación usada
- ✓ Tipos de algoritmos de seguridad
- ✓ Eficiencia en establecer sesión
- ✓ Tipos de soporte de respaldo y restauración

- **Administración**

- ✓ Conexión local
- ✓ Conexión remota
- ✓ Facilidad diseño de Base de Datos
- ✓ Facilidad diseño de Tablas

- ✓ Facilidad diseño de Funciones
- ✓ Diseño y Consultas de Querys

- **Calidad**

- ✓ Uso de recursos hardware
- ✓ Facilidad de Uso
- ✓ Compatibilidad Software

3.3.1. Parámetro 1: Instalación

El Parámetro de Instalación proporciona métricas referentes al proceso que se debe seguir para dar funcionalidad a la Herramienta y su entorno de trabajo.

Tabla I Descripción Indicador 1: Instalación

		Descripción
Indicador	Instalación	La sección de Instalación de la herramienta con sus respectivas configuraciones en el proceso de instalación.
[índice 1.1]	Tiempo en Completar Instalación	Es el tiempo que tarda en instalar la Herramienta.
[índice 1.2]	Pasos de Configuración	Describe el grado de complejidad y pasos que toma en configurar los elementos de trabajo de la Herramienta.
[índice 1.3]	Nivel de Control de Acceso	Este índice evaluará el número de firmas digitales que tiene la herramienta.
[índice 1.4]	Nivel de ayuda en los procesos de instalación.	Este índice evalúa la facilidad de manipulación que tiene la interfaz administrativa.

3.3.2. Parámetro 2: Seguridad

Aquí se describe la Seguridad que posee cada implementación, haciendo énfasis en el tiempo para iniciar una sesión, los paquetes que se envían y la cantidad de algoritmos que ofrece cada implementación analizada.

Tabla II Descripción Indicador 2: Seguridad

Indicador	Seguridad	Descripción
[índice 2.1]	Tipo de autenticación usada	Describe como es la seguridad y la autenticarán los usuarios para cada caso.
[índice 2.2]	Tipos de algoritmos de seguridad	Describe el tiempo que tarda en autenticarse un usuario con cada SGBD.
[índice 2.3]	Nivel de seguridad para establecer sesión	Describe el grado de complejidad que toma en configurar los elementos de trabajo de la Herramienta.
[índice 2.4]	Tipos de soporte de respaldo y restauración	Este índice evaluará cuantos algoritmos de seguridad soporta.
		El índice determinará las formas en que se puede respaldar y guardar la información de un modo seguro.

3.3.3. Parámetro 3: Administración

Nos indica la forma de Administrar, que cuenta la Herramienta para respaldar la información, además del soporte para respaldar datos por medio de la interfaz.

Tabla III Descripción Indicador 3: Administración

		Descripción
Indicador	Administración	Describe la forma de gestionar las bases de datos por medio de las herramientas y manera que se acceden a ellas.
[índice 3.1]	Conexión local	Permitirá evaluar cómo se realiza conexión desde el servidor local.
[índice 3.2]	Conexión remota	Permitirá evaluar cómo se realiza conexión desde el servidor remoto.
[índice 3.3]	Facilidad diseño de Base de Datos.	Permitirá evaluar cómo se realiza el diseño de bases de datos por medio de la interfaz.
[índice 3.4]	Facilidad diseño de Tablas	Permitirá evaluar cómo se realiza el diseño de tablas por medio de la interfaz.
[índice 3.5]	Facilidad diseño de Funciones	Permitirá evaluar cómo se realiza el diseño de funciones por medio de la interfaz.
[índice 3.6]	Diseño y Consultas de Querys	Permitirá evaluar cómo se realiza el diseño y consultas querys por medio de la interfaz.

3.3.4. Parámetro 4: Calidad

La calidad puede ser valorada desde diferentes perspectivas, en el presente estudio el parámetro calidad será evaluado con el fin de determinar el conjunto de características propias de un software, que permitan valorarlo como igual, mejor o peor que el resto de

productos de su especie, a través de las cuales satisfacen necesidades explícitas para las cuales ha sido creado el software.

Tabla IV Descripción Indicador 4: Calidad

Indicador	Calidad	Descripción
[índice 4.1]	Uso de recursos hardware	Describe cómo interactúan las herramientas a los recursos internos y externos al momento de ejecutarlas.
[índice 4.2]	Facilidad de Uso	Es el factor que determina las características mínimas de hardware para iniciar a trabajar con la herramienta para el diseño de base de datos.
[índice 4.3]	Compatibilidad Software	Se refiere al grado en el que la interfaz, el conjunto de componentes y la forma como están distribuidos los sub-paneles en la pantalla principal de la herramienta, facilitan o dificultan su manejo.
		Permite verificar la posibilidad de poder abrir el trabajo en un software, en otro programa que permita realizar tareas similares.

3.4. Descripción de los módulos de pruebas

Los módulos de prueba son escenarios que ayudan a verificar y obtener datos para determinar que Herramienta es mejor ante diferentes configuraciones.

Los módulos que se desarrollaron serán implementados en los 3 Herramientas seleccionadas cada una con su manera particular de diseño y desarrollo siguiendo el mismo modelo de arquitectura de desarrollo llamado MVC (modelo vista, controlador).

En cada Herramientas se probará los mismos escenarios y se obtendrá los resultados mediante un software especial para calcular el rendimiento de hardware y software.

Para el desarrollo de la aplicación web, se tendrá 3 módulos que son:

1. Módulo de Navegación de Interfaces
2. Módulo de transacciones de alta y baja.
3. Módulo de funciones.

A continuación se explica cada uno de los módulos para su posterior desarrollo e implementación.

3.4.1. Módulo de Navegación de interfaces

Conocido también como capa de presentación. El módulo describe la manera en que el usuario interactúa con la aplicación y la información almacenada en la base de datos, con este módulo se verificará la velocidad de interpretación del código jsp para su posterior presentación en forma visual de interfaces.

Este módulo está compuesto por todos los elementos de una interfaz tales como tiempo en el proceso de instalación, menús, botones, navegación por tabulaciones, hipervínculos, código jsp, llamada a datos de bases de datos hacia la interfaz.

3.4.2. Módulo de transacciones de alta,mediay baja

En este módulo se encuentra la manera de acceder a la información desde y hacia una base de datos postgresql.

Con este módulo se probara la administración, seguridad y la capacidad de carga que puede soportar la herramienta en cuanto a manejar grandes cantidades de transacciones al mismo tiempo, así como su integridad en cuanto a la transmisión de la información, es decir si llega a información con errores o fallas en él envío.

3.4.3. Módulo de funciones

En este módulo se encuentra la lógica del negocio, la programación de funciones y procedimientos para que los módulos realicen las tareas dinámicamente, ya sea generando interfaces o realizando transacciones como consultas, inserciones e eliminaciones, sentencias querys.

Con la iteración en este módulo probaremos la capacidad y mejores prácticas de programación y el acceso de la base de datos facilidad en administración que brinda cada Herramienta al momento de llamar a sus funciones para generar las interfaces e información requerida por el usuario para ver la calidad de cada una.

3.5. Desarrollo de los módulos de pruebas

Los distintos módulos dentro de la aplicación están programados bajo el lenguaje de programación jsp y siguiendo la arquitectura MVC Modelo vista controlador que es un patrón de arquitectura de las aplicaciones software.

Todos los módulos siguen el mismo patrón de diseño MVC y cada módulo en diferente Herramienta tiene su manera estándar de realizar las acciones para llamar, crear, generar funciones,

Cada base de datos creada por laHerramienta es implementada en la misma y realizaran idénticas acciones como funciones, para probar los escenarios planteados.

3.5.1. GUIPgAdminIII

El desarrollo de la aplicación web jsp utilizando PgAdminIII siguiendo la arquitectura Modelo Vista controlador, como indica la FiguraIII.10. Los principales componentes de administración de PgAdminIII dentro de su estructura son:

- ✓ Databases
- ✓ Tablespace
- ✓ Group Roles
- ✓ Login Roles

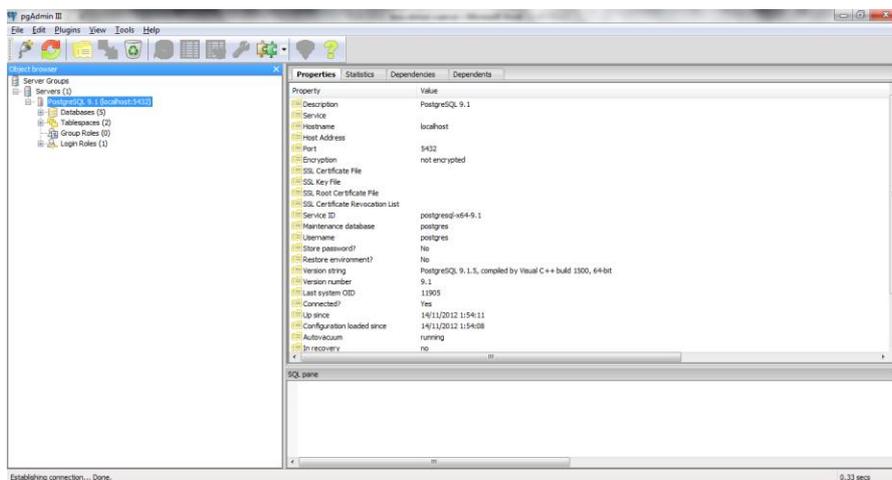


Figura III.10 Estructura Detallada de PgAdminIII

3.5.1.1. Módulo de Navegación de Interfaces

Dentro del módulo de interfaces del usuario, se ha realizado una interfaz web jspamigable utilizando la base de datos creada en padminIII, esta interfaz web cuenta con funciones propias para una navegación intuitiva y rápida, estas funciones son paneles de navegación que nos permitirán interactuar en una página y otra, la interfaz nos permitirá capturar el tiempo que dura el proceso de iteración entre los elementos creados.



Figura III. 11 Interfaz principal sistema con la base de datos creada



Figura III. 12 Interfaz de Presentación de ingreso Información a la Base de Datos

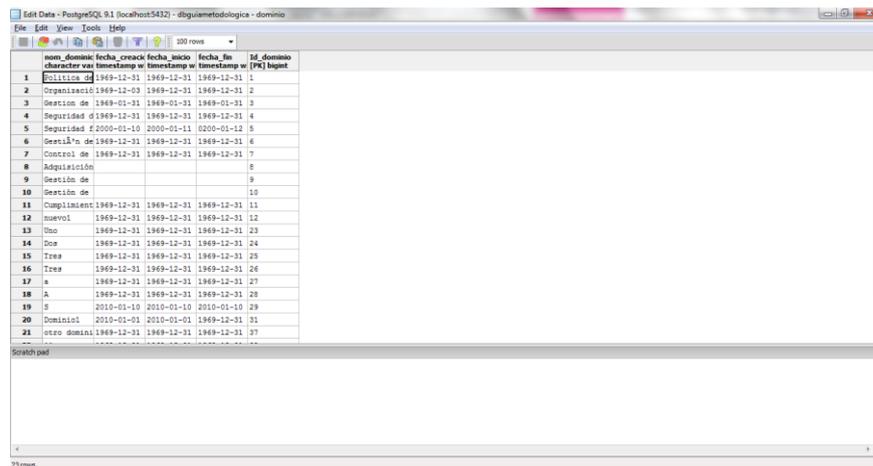
3.5.1.2. Módulo de Funciones y transacciones de alta

Dentro de la arquitectura MVC se encuentra la estructura del modelo que es la manera de acceder a la base de datos, donde se implementa todas las funciones para el ingreso, actualización desde y hacia la base de datos.



NOMBRE DOMINIO	FECHA CREACION	FECHA INICIO	FECHA FINAL	ELIMINAR	ACTUALIZAR	DETALLES
Adquisición,	null	null	null	X		VER
Gestión	null	null	null	X		VER
Gestión	null	null	null	X		VER
nuevo1	1969-12-31	1969-12-31	1969-12-31	X		VER
Política	1969-12-31	1969-12-31	1969-12-31	X		VER
Cumplimiento	1969-12-31	1969-12-31	1969-12-31	X		VER
Uno	1969-12-31	1969-12-31	1969-12-31	X		VER
Organización	1969-12-03	1969-12-31	1969-12-31	X		VER
Dos	1969-12-31	1969-12-31	1969-12-31	X		VER
Tres	1969-12-31	1969-12-31	1969-12-31	X		VER
Tres	1969-12-31	1969-12-31	1969-12-31	X		VER
a	1969-12-31	1969-12-31	1969-12-31	X		VER

Figura III. 13 Interfaz de Funciones de Ingreso y Actualización de Datos



nom_dominio	fecha_creacion	fecha_inicio	fecha_fin	id_dominio	caracteres
Adquisición	1969-12-31	1969-12-31	1969-12-31	1	
Organización	1969-12-03	1969-12-31	1969-12-31	2	
Gestión de	1969-01-31	1969-01-31	1969-01-31	3	
Seguridad d	1969-12-31	1969-12-31	1969-12-31	4	
Seguridad d	2000-01-10	2000-01-10	2000-01-10	5	
Servicio de	1969-12-31	1969-12-31	1969-12-31	6	
Control de	1969-12-31	1969-12-31	1969-12-31	7	
Adquisición				8	
Gestión de				9	
Gestión de				10	
Cumpliment	1969-12-31	1969-12-31	1969-12-31	11	
nuevo1	1969-12-31	1969-12-31	1969-12-31	12	
Uno	1969-12-31	1969-12-31	1969-12-31	13	
Dos	1969-12-31	1969-12-31	1969-12-31	14	
Tres	1969-12-31	1969-12-31	1969-12-31	15	
Tres	1969-12-31	1969-12-31	1969-12-31	16	
a	1969-12-31	1969-12-31	1969-12-31	17	
A	1969-12-31	1969-12-31	1969-12-31	18	
o	2010-01-10	2010-01-10	2010-01-10	19	
Dominio	2010-01-01	2010-01-01	1969-12-31	20	
otro dominio	1969-12-31	1969-12-31	1969-12-31	21	

Figura III. 14 PgAdminIII con datos presentados desde la interfaz

3.5.1.3. Módulo de funciones y transacciones de baja

Dentro de la arquitectura MVC se encuentra la estructura del modelo que es la manera de acceder a la base de datos, donde se implementa todas las funciones para la eliminación, actualización desde y hacia la base de datos.

COD	NOMBRE CONTROL	DOMINIO	ESTADO	NORMA ISO	FECHA INICIO	FECHA FIN	ELIMINAR	ACTUALIZAR
1	Documento	Organización	activo	Cumple	2010-01-10	2010-01-01	X	
2	Revisión	Política	no	Cumple	null	null	X	
3	Compromiso	Organización	activo	Cumple	1969-12-31	1969-12-31	X	
4	Coordinación	Organización	activo	Cumple	null	null	X	
5	Asignación	Organización	activo	Cumple	null	null	X	
6	Proceso	Organización	pasivo	Cumple	null	null	X	
7	Acuerdos	Organización	no	Cumple	null	null	X	
8	Contacto	Organización	activo	Cumple	null	null	X	
9	Contacto	Organización	pasivo	Cumple	null	null	X	
10	Revisión	Organización	no	Cumple	null	null	X	
11	Identificación	Organización	activo	Cumple	null	null	X	
12	Tratamiento	Organización	pasivo	Cumple	null	null	X	
13	Tratamiento	Organización	no	Cumple	null	null	X	

Figura III. 15 Interfaz de Funciones de Eliminación en la Base de Datos

id_control (PK)	nom_control	id_dominio	id_estado	id_norma	fecha_inicio	fecha_fin
1	Revisión	1	3	1		
2	Compromiso	2	1	1	1969-12-31	1969-12-31
3	Coordinación	2	1	1		
4	Asignación	2	1	1		
5	Proceso de	2	2	1		
6	Acuerdos de	2	3	1		
7	Contacto co	2	1	1		
8	Contacto co	2	2	1		
9	Revisión in	2	3	1		
10	Identificac	2	1	1		
11	Tratamiento	2	2	1		
12	Tratamiento	2	3	1		
13	Inventario	3	1	1		
14	Responsable	3	2	1		
15	Uso aceptab	3	3	1		
16	Directrices	3	3	1		
17	Etiquetado	3	2	1		
18	Funciones y	3	1	1		
19	Selección y	3	1	1		
20	Términos y	3	2	1		
21	Responsabil	4	3	1		
22						

Figura III. 16 Interfaz PgAdminIII eliminado dato desde interfaz

3.5.2. GUI PHPPgAdmin

Estructura de datos del Patrón de diseño MVC, se detalla la manera de organización de las bases de datos. Los principales componentes de administración de PHPPgAdmin dentro de su estructura son:

- ✓ Bases de Datos
- ✓ Roles
- ✓ Tablespaces
- ✓ Exportar
- ✓ Reportes

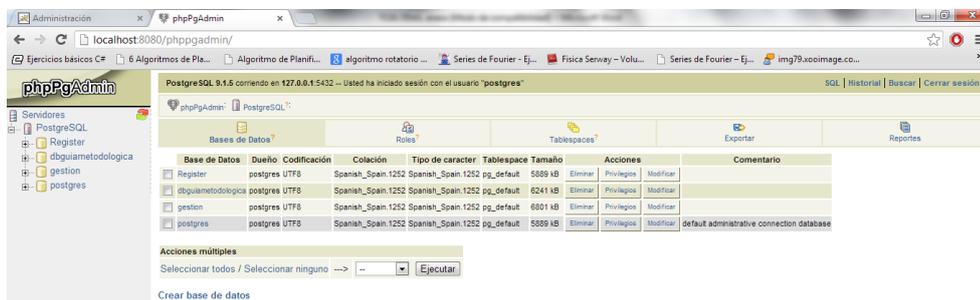


Figura III. 17 Estructura Detallada de PhppgAdmin Ver Anexo 1

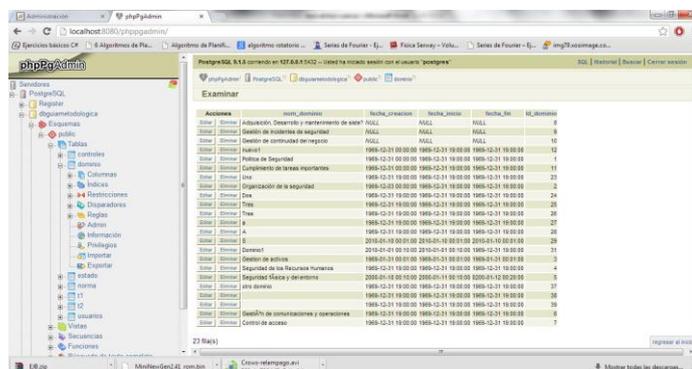


Figura III. 18 PhppgAdmin con datos presentados desde la interfaz

3.5.2.1. Módulo de navegación de interfaces

Se muestra la iteración entre los elementos principales de la interfaz, como son el Área de trabajo de la herramienta con los links, botones y demás elementos que forman la interfaz.

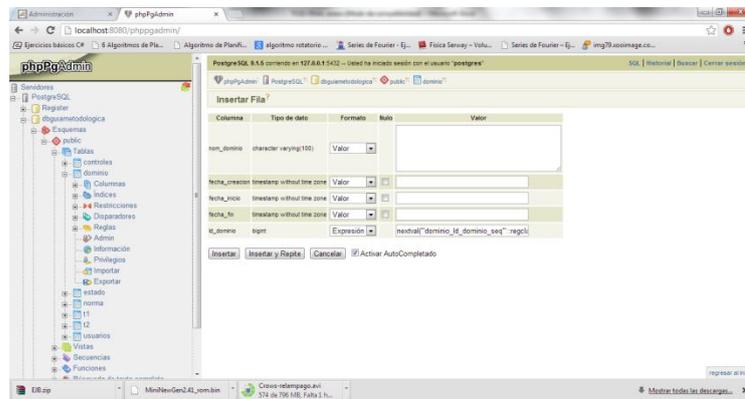


Figura III. 19 Interfaz de Navegación de Funciones phppgadmin.

3.5.2.2. Módulo de funciones y transacciones de alta

Dentro del módulo de interfaz tenemos un modular dedicado al ingreso de la información, para ingreso y edición en la base de datos.



Figura III. 20 Interfaz de listado datos control eliminaciones

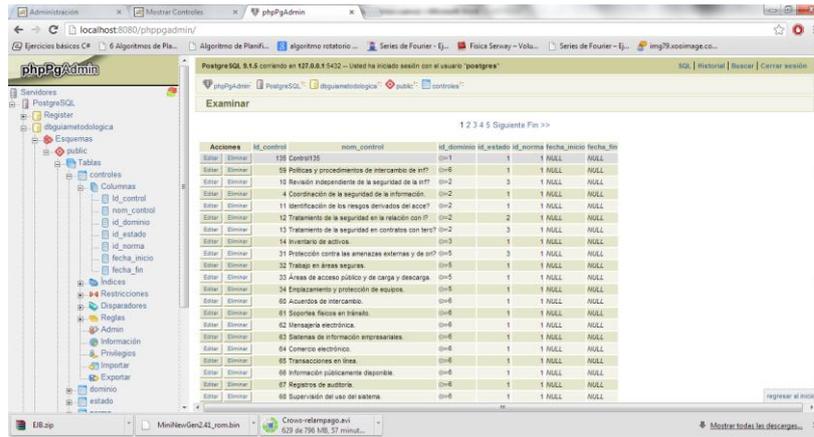


Figura III. 21 Phppgadmin datos que se presentan desde la interfaz

3.5.2.3.Módulo de funciones y transacciones de baja

Se muestra la captura del módulo de funciones, donde se detalla las funciones que realiza para capturar la información sobre las cargas transaccionales que se realiza con la herramienta

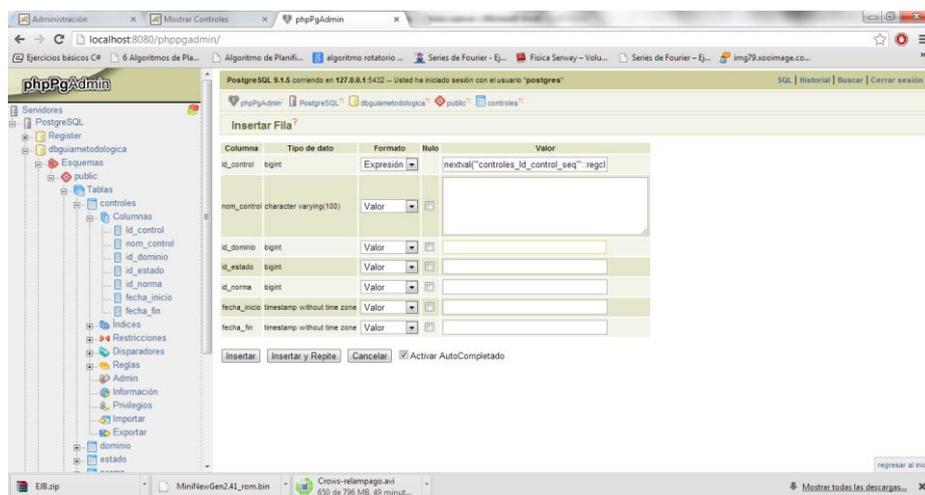


Figura III. 22 Interfaz de para Realizar Cargas Transaccionales.

3.5.3. EMS SQL Manager for PostgreSQL

Patrón de diseño MVC, se detalla la estructura interna del Herramienta, destacando sus principales estructuras que son:

- ✓ Schemas
- ✓ Tablespaces
- ✓ Languages
- ✓ Reports
- ✓ Favorite Queries
- ✓ Projects
- ✓ Local Scripts

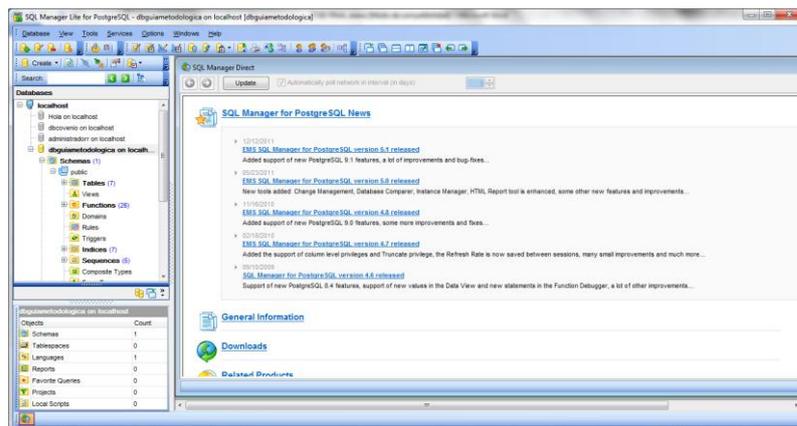


Figura III. 23 Estructura Detallada de EMS SQL LiteforPostgreSQL Ver Anexo 1

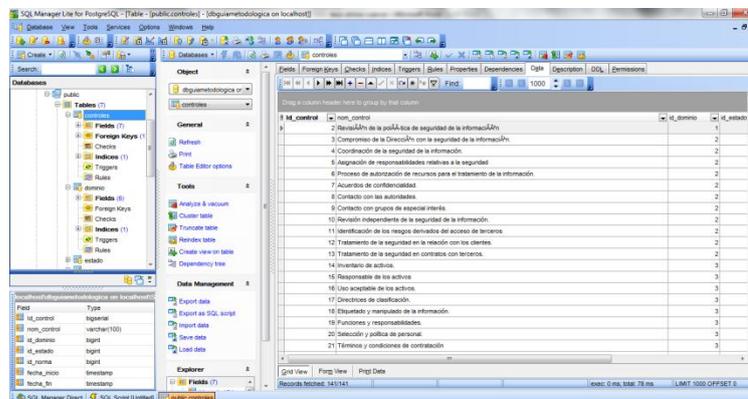


Figura III. 24 con datos presentados en la tabla desde interfaz

3.5.3.1. Módulo de navegación de interfaces

Se muestra la iteración entre los elementos principales de la interfaz, como son el Área de trabajo de la herramienta con los links, botones y demás elementos que forman la interfaz.

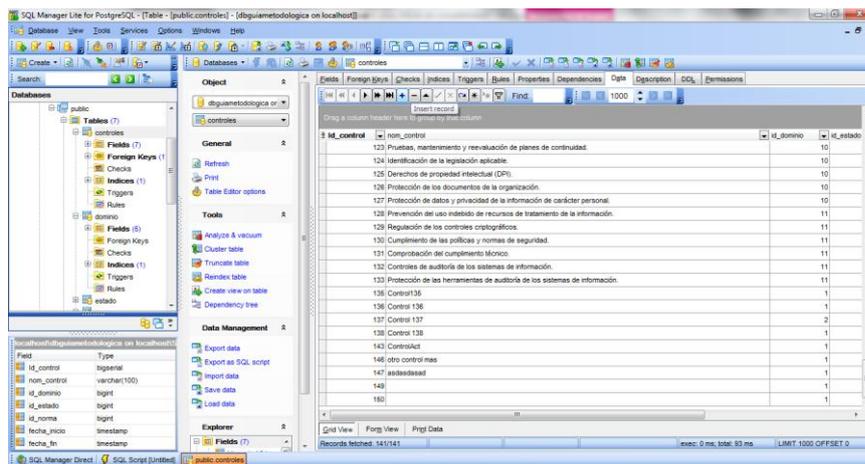


Figura III. 25 Interfaz de Navegación de Funciones EMS SQL LiteforPostgreSQL

3.5.3.2. Módulo de funciones y transacciones de alta

Dentro del módulo de interfaz tenemos un modular dedicado al ingreso de la información, para ingreso y edición en la base de datos.

The screenshot shows the 'LISTADO DE CONTROLES' interface. At the top, there is a header with the logo 'SGSI' and the text 'Sistema de Gestión de Seguridad de la Información'. Below the header is a table with the following columns: COD, NOMBRE CONTROL, DOMINIO, ESTADO, NORMA ISO, FECHA INICIO, FECHA FIN, and ELIMINAR. The table contains 14 rows of data, with the 'ELIMINAR' column containing red 'X' marks. The interface also includes a search bar and a 'Centro Sesion' button.

COD	NOMBRE CONTROL	DOMINIO	ESTADO	NORMA ISO	FECHA INICIO	FECHA FIN	ELIMINAR
2	RevisiAAñ	Politica	no	Cumple	null	null	X
3	Compromiso	Organización	activo	Cumple	1969-12-31	1969-12-31	X
4	Coordinación	Organización	activo	Cumple	null	null	X
5	Asignación	Organización	activo	Cumple	null	null	X
6	Proceso	Organización	pasivo	Cumple	null	null	X
7	Acuerdos	Organización	no	Cumple	null	null	X
8	Contacto	Organización	activo	Cumple	null	null	X
9	Contacto	Organización	pasivo	Cumple	null	null	X
10	Revisión	Organización	no	Cumple	null	null	X
11	Identificación	Organización	activo	Cumple	null	null	X
12	Tratamiento	Organización	pasivo	Cumple	null	null	X
13	Tratamiento	Organización	no	Cumple	null	null	X
14	Inventario	Gestion	activo	Cumple	null	null	X

Figura III. 26 Interfaz de listado datos control eliminaciones

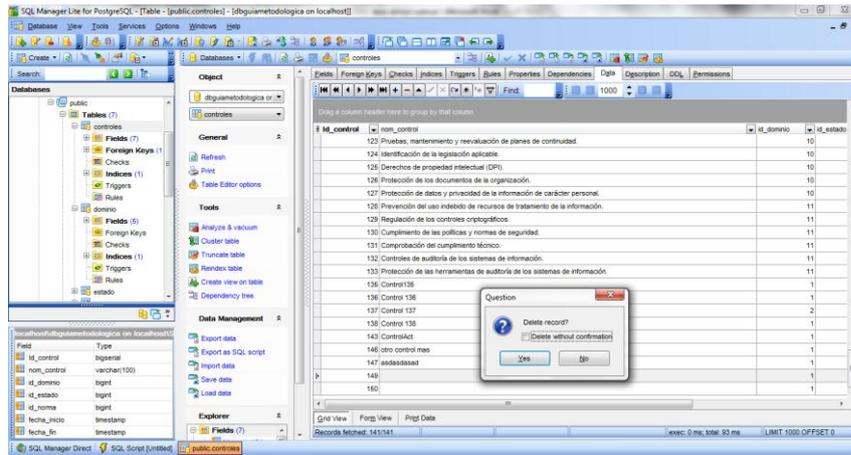


Figura III. 27 EMS SQL Lite for PostgreSQL datos que se presentan desde la interfaz

3.5.3.3. Módulo de funciones y transacciones de baja

Se muestra la captura del módulo de funciones, donde se detalla las funciones que realiza para capturar la información sobre las cargas transaccionales que se realiza con la herramienta.

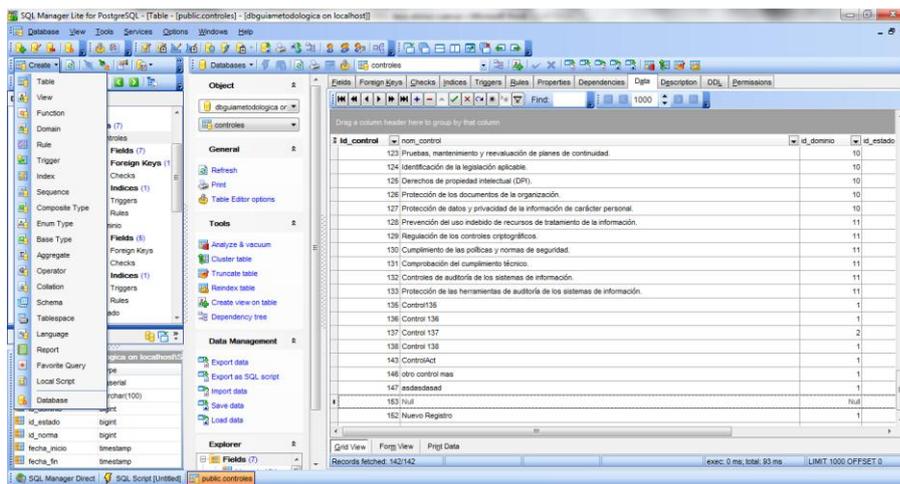


Figura III. 28 Interfaz de para Realizar Cargas Transaccionales.

3.6. Análisis Comparativo

En esta sección se va a mostrar el estudio de las Herramientas para los Sistemas Gestores de Base de Datos libres para Postgresql; PgAdminIII, PHPpgAdmin y EMS SQL Lite forpostgresql a manera de cuadros comparativos, seguidos estos de una interpretación y calificación del criterio evaluado por parte del autor, estos cuadros comparativos se encuentran clasificados de acuerdo a los parámetros de comparación definidos anteriormente.

Para obtener resultados cuantitativos y cualitativos que permitan una selección sustentada de una de las herramientas analizadas, la clasificación de cada uno de los parámetros de comparación se basa en las siguientes escalas que se describen en los cuadros siguientes.

3.6.1. Indicador 1: Instalación

3.6.1.1.Determinación del Indicador

La sección de Instalación posee 4 índices cada uno de los cuales mantuvo una prueba diferente. Sus resultados obtenidos serán evaluados tomando sus valores mínimos, máximos para cada índice, así realizar una escala de valorización y determinar los valores no satisfactorio, poco satisfactorio, satisfactorio y muy satisfactorio respectivamente.

Tabla V Valorización para el Indicador 1: Instalación

Valor Cualitativo	Valor Cuantitativo	Valor Representativo
No Satisfactorio	1	♣
Poco Satisfactorio	2	♣♣
Satisfactorio	3	♣♣♣
Muy Satisfactorio	4	♣♣♣♣

- **Índice 1.1: Tiempo en Completar Instalación:** En este índice se valorizará para cada ver el tiempo de instalación en cada una de las herramientas.

Tabla VI Valorización del Índice 1.1: Tiempo de Instalación

Tiempo Minutos	Valor Cualitativo
17-25+	No Satisfactorio
11-16	Poco Satisfactorio
6-10	Satisfactorio
1-5	Muy Satisfactorio

- **Índice 1.2: Pasos de Configuración:** En este índice se valorizará para cada una de las configuraciones necesarias en la instalación por Herramienta.

Tabla VII Valorización del Índice 1.2: Pasos de Configuración

Pantallas de Pasos configuración	Valor Cualitativo
16-20+	No Satisfactorio
11-15	Poco Satisfactorio
6-10	Satisfactorio
1-5	Muy Satisfactorio

- **Índice 1.3: Nivel de Control de Acceso:** En este índice se valorizará para cada una de las Herramientas el acceso a las bases de datos.

Tabla VIII Valorización del Índice 1.3: Nivel de Control de Acceso

Pantallas de Niveles acceso	Valor Cualitativo
7-8+	No Satisfactorio
5-6	Poco Satisfactorio
3-4	Satisfactorio
0-2	Muy Satisfactorio

- **Índice 1.4: Nivel de ayuda en los procesos de instalación.:** En este índice se valorizará que tan fácil es administrar cada una de las Herramientas.

Tabla IX Valorización del Índice 1.4: Nivel de ayuda en los procesos de instalación

Pantallas proceso de Ayuda	Valor Cualitativo
1-10	No Satisfactorio
11-20	Poco Satisfactorio
21-30	Satisfactorio
31-40	Muy Satisfactorio

3.6.1.2.Valoraciones

Tabla X Resultados del Indicador 1: Instalación, después de la obtención de la ejecución de las diferentes pruebas Ver Anexo 2.

	INSTALACIÓN		
	PgAdminIII	PHPpgAdmin	EMS SQL Lite for PostgreSQL
Tiempo en Completar Instalación	03:46:22 minutos	05:39:74 minutos	01:06:66 minutos
Pasos de Configuración	7	14	8
Nivel de Control de Acceso	1	2	0
Nivel de ayuda en los procesos de instalación.	7	14	8

3.6.1.3.Calificación

Tabla IX Calificación del Indicador 1: Instalación, se basarán en la calificación de las Tablas (6-10)

	INSTALACIÓN						
	PgAdminIII		PHPpgAd min		EMS SQL Lite for PostgreSQL		
	Cuantita tivo	Cualitati vo	Cuantita	Cualitati	Cuantita tivo	Cualitati vo	
Tiempo en Completar Instalación	4	Muy Satisfactorio	3	Satisfactorio	4	Muy Satisfactorio	

Pasos de Configuración	3	Satisfactorio	2	Poco Satisfactorio	3	Satisfactorio
Nivel de Control de Acceso	4	Muy Satisfactorio	4	Muy Satisfactorio	4	Muy Satisfactorio
Nivel de ayuda en los procesos de instalación.	4	Muy Satisfactorio	3	Satisfactorio	4	Muy Satisfactorio

Para sacar los porcentajes de cada índice se utilizará las siguientes fórmulas.

Dónde:

X = Equivale a PgAdminIII.

Y = Equivale a PHPpgAdmin

Z = Equivale a EMS SQL Lite forPostgreSQL.

C1(X)= Puntaje acumulativo del Tiempo en Completar Instalación de PgAdminIII

C2(X)= Puntaje acumulativo de Pasos de Configuración de PgAdminIII

C3(X)= Puntaje acumulativo Nivel de Control de Acceso de PgAdminIII

C4(X)=Puntaje acumulativo Nivel de ayuda en los procesos de instalación de PgAdminIII

C1(Y)= Puntaje acumulativo del Tiempo en Completar Instalación de PHPpgAdmin

C2(Y)= Puntaje acumulativo de Pasos de Configuración de PHPpgAdmin

C3(Y)= Puntaje acumulativo Nivel de Control de Acceso de PHPpgAdmin

C4(Y)=Puntaje acumulativo Nivel de ayuda en los procesos de instalación de PHPpgAdmin

C1(Z)= Puntaje acumulativo del Tiempo en Completar Instalación de EMS SQL Lite forPostgreSQL.

C2(Z)=Puntaje acumulativo de Pasos de Configuración de EMS SQL Lite forPostgreSQL.

C3(Z)=Puntaje acumulativo Nivel de Control de Acceso de EMS SQL Lite forPostgreSQL.

C4 (Z)= Puntaje acumulativo Nivel de ayuda en los procesos de instalación de EMS SQL Lite forPostgreSQL.

Vi= Resultado de las pruebas.

$$\left[\begin{array}{l} \mathbf{C1(X)} = \sum_{i=1}^{n-1} vi ; \mathbf{C2(X)} = \sum_{i=1}^{n-1} vi ; \mathbf{C3(X)} = \sum_{i=1}^{n-1} vi ; \mathbf{C4(X)} = \sum_{i=1}^{n-1} vi \end{array} \right] \quad [1.1]$$
$$\left[\begin{array}{l} \mathbf{C1(Y)} = \sum_{i=1}^{n-1} vi ; \mathbf{C2(Y)} = \sum_{i=1}^{n-1} vi ; \mathbf{C3(Y)} = \sum_{i=1}^{n-1} vi ; \mathbf{C4(Y)} = \sum_{i=1}^{n-1} vi \end{array} \right] \quad [1.2]$$
$$\left[\begin{array}{l} \mathbf{C1(Z)} = \sum_{i=1}^{n-1} vi ; \mathbf{C2(Z)} = \sum_{i=1}^{n-1} vi ; \mathbf{C3(Z)} = \sum_{i=1}^{n-1} vi ; \mathbf{C4(Z)} = \sum_{i=1}^{n-1} vi \end{array} \right] \quad [1.3]$$

Pa(X)= Valor acumulativo de PgAdminIII

Pa (Y)= Valor acumulativo de PHPpgAdmin

Pa (Z)= Valor acumulativo de EMS SQL Lite forPostgreSQL.

$$\left[\begin{array}{l}
 Pa(X) = \sum C1(X) + C2(X) + C3(X) + C4(X) \\
 Pa(Y) = \sum C1(Y) + C2(Y) + C3(Y) + C4(Y) \\
 Pa(Z) = \sum C1(Z) + C2(Z) + C3(Z) + C4(Z)
 \end{array} \right] \quad [1.4]$$

PpT (X)= Porcentaje Parcial Total de PgAdminIII

PpT (Y)= Porcentaje Parcial Total de PHPpgAdmin

PpT (Z)= Porcentaje Parcial Total de EMS SQL Lite forPostgreSQL

Vp= Valor máximos de índices es de 4

$$\left[\begin{array}{l}
 PpT(X) = \left(\frac{Pa(X)}{Vp} \right) * 100 \\
 PpT(Y) = \left(\frac{Pa(Y)}{Vp} \right) * 100 \\
 PpT(Z) = \left(\frac{Pa(Z)}{Vp} \right) * 100
 \end{array} \right] \quad [1.5]$$

Tabla X Valores y Porcentajes del Indicador 1: Instalación, con sus respectivos Índices

	INSTALACIÓN					
	PgAdminIII		PHPpgAdmin		EMS SQL Lite for PostgreSQL	
	Valor (X)	% PpT	Valor (Y)	% PpT	Valor (Z)	% PpT
Tiempo en Completar InstalaciónC1	4	100	3	75	4	100
Pasos de ConfiguraciónC2	3	75	2	50	3	75
Nivel de Control de AccesoC3	4	100	4	100	4	100
Nivel de ayuda en los procesos de instalación.C4	4	100	3	75	4	100

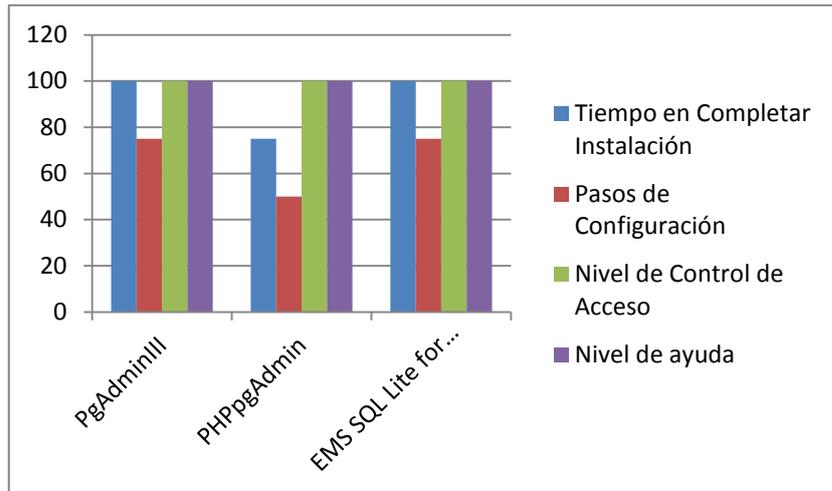


Figura III. 29 Resultado por índice del Indicador 1: Instalación

Para sacar los porcentajes de cada índice se utilizará las siguientes formulas.

Dónde:

PT (X)= Porcentaje Total de PgAdminIII

PT (Y)= Porcentaje Total de PHPpgAdmin

PT (Z)= Porcentaje Total de EMS SQL Lite forPostgreSQL

VA= Valor máximos de índices es de 16

$$\left(\begin{array}{l} PT(X) = \left(\frac{Pa(X)}{VA} \right) * 100 \\ PT(Y) = \left(\frac{Pa(Y)}{VA} \right) * 100 \\ PT(Z) = \left(\frac{Pa(Z)}{VA} \right) * 100 \end{array} \right) \quad [1.6]$$

Tabla XI Valores y Porcentajes Finales del Indicador 1: Instalación.

	Instalación	
	Valor (Pa)	% (PT)
PgAdminIII: (X)	15	93,75
PHPpgAdmin: (Y)	12	75
EMS SQL Lite for PostgreSQL: (Z)	15	93,75

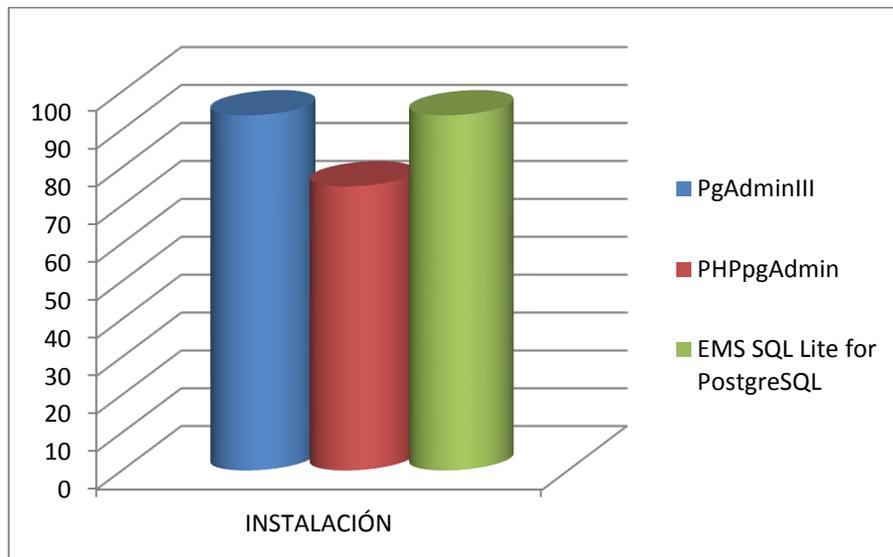


Figura III. 30 Resultado Final del Indicador 1: Instalación

3.6.1.4. Interpretación

Tabla XII Representación del Indicador 1: Instalación

	INSTALACIÓN		
	PGADMINIII	PHPPGADMIN	EMS SQL LITE FOR POSTGRESQL
Tiempo en Completar Instalación	♣♣♣♣	♣♣♣	♣♣♣♣
Pasos de Configuración	♣♣♣	♣♣	♣♣♣
Nivel de Control de Acceso	♣♣♣♣	♣♣♣♣	♣♣♣♣
Nivel de ayuda en los procesos de instalación.	♣♣♣♣	♣♣♣	♣♣♣♣

3.6.1.5. Descripción Resultados

Los resultados expuestos en la tabla XIV “representación del Indicador 1: Instalación” serán explicados por índices.

Tiempo en Completar Instalación: El Tiempo es primordial cuando se trata de realizar tareas administrativas, en el proceso de instalación de los clientes para postgres, estas herramientas mantienen una concordancia entre si ya que son aplicaciones de escritorio con instaladores propios y su proceso de instalación es muy idéntico pero se nota que el tiempo para completar la instalación en PgadminIII es igual al de EMS SQL

Lite para Postgresql es decir el tiempo que se necesita para instalarlos es menor comparándolos con el que necesita PHPpgAdmin ya que su tiempo de instalación es mucho mayor que los dos anteriores.

Pasos de Configuración:Guiándose en los resultados de la tabla de referencia podemos afirmar que PgAdminIII y EMS SQL Lite para Postgresql tienen los mismos pasos de configuración al momento de instalarse en comparación con PHPpgAdmin ya que requiere varias configuraciones adicionales antes de proceder y concluir su configuración necesaria.

Nivel de Control de Acceso:Los mejores puntajes obtienen PgAdminIII y EMS SQL Lite para Postgresql con 4 puntos ya que el control de acceso a las bases de datos lo realiza por medio de una clave de acceso al sistema y así administrar los recursos que esta tenga seguido muy de cerca de PhpPgAdmin con 3 puntos debido a que en el proceso de instalación no solicita controles para su acceso respectivo.

Nivel de ayuda en los procesos de instalación.: Destacando de manera sorprendente nuevamente a PgAdminIII, por mantener una ayuda directa en cada paso al momento de realizar la instalación respectiva, detallado cada paso y proceso de configuración permitiendo una fácil instalación desde su inicio, lo mismo ocurre con EMS SQL Lite para Postgresql al momento de instalarlo, seguido muy de cerca también por PhpPgAdmin con el mismo nivel de ayuda en su instalación.

3.6.2. Indicador 2: Seguridad

3.6.2.1. Determinación del Indicador

La sección de Seguridad posee 4 índices que serán evaluados tomando sus valores mínimos, máximos para cada índice, así realizar una escala de valorización y determinar los valores no satisfactorio, poco satisfactorio, satisfactorio y muy satisfactorio respectivamente.

Tabla XIII Valorización para el Indicador 2: Seguridad

Valor Cualitativo	Valor Cuantitativo	Valor Representativo
No Satisfactorio	1	♣
Poco Satisfactorio	2	♣♣
Satisfactorio	3	♣♣♣
Muy Satisfactorio	4	♣♣♣♣

- **Índice 2.1: Tipo de autenticación usada:** En este índice se valorizará para cada tipo de autenticación sea local o remota dependiendo de cada una de las Herramientas.

Tabla XIV Valorización del Índice 1.1: Tipo de autenticación usada

Modos de Autenticación	Valor Cualitativo
Ninguno	No Satisfactorio
Local	Poco Satisfactorio
Remota	Satisfactorio
Remota, Local	Muy Satisfactorio

- **Índice 2.2: Tipos de algoritmos de seguridad:** En este índice se valorizará para cada una de las configuraciones necesarias en la instalación por Herramienta.

Tabla XV Valorización del Índice 2.2: Tipos de algoritmos de seguridad

Tipos de Algoritmos de Cifrado	Valor Cualitativo
Sin Algoritmo.	No Satisfactorio
MS-CHAP	Poco Satisfactorio
SSL,EAP,MD5	Satisfactorio
EAP,SSL,MD5,CA	Muy Satisfactorio

- **Índice 2.3: Nivel de seguridad para establecer sesión:** En este índice se valorizará para cada una de las Herramientas el acceso a las bases de datos.

Tabla XVI Valorización del Índice 2.3: Nivel de seguridad para establecer sesión

Acceso y certificados de seguridad	Valor Cualitativo
Password	No Satisfactorio
Login+Password	Poco Satisfactorio
SSL	Satisfactorio
SSL,login+password	Muy Satisfactorio

- **Índice 2.4: Tipos de soporte de respaldo y restauración:** En este índice se valorizará para cada Herramienta.

Tabla XVII Valorización del Índice 2.4: Tipos de soporte de respaldo y restauración

Modos de Soporte y Restauración	Valor Cualitativo
Sin Soporte	No Satisfactorio
Script Sql	Poco Satisfactorio
Backup Base de Datos, Restaurar Base de Datos, Script Sql	Satisfactorio
Backup Base de Datos, Restaurar Base de Datos, Backup automático, Script Sql, Respaldo remoto Base de datos.	Muy Satisfactorio

3.6.2.2.Valoraciones

Tabla XVIII Resultados del Indicador 2: Seguridad, después de la obtención de la ejecución de las diferentes pruebas Ver Anexo 2.

	SEGURIDAD		
	PgAdminIII	PHPpgAdmin	EMS SQL Lite for PostgreSQL
Tipo de autenticación usada	Local, Remota	Local	Local, Remota
Tipos de algoritmos de seguridad	EAP,SSL,MD5, CA	MD5	SSL,EAP,MD5

Nivel de seguridad para establecer sesión	SSL, login+password	Login+Passwo rd	SSL, login+password
Tipos de soporte de respaldo y restauración	Backup Base de Datos, Restaurar Base de Datos, Backup automático, Script Sql, Respaldo remoto Base de datos.	Script Sql	Backup Base de Datos, Restaurar Base de Datos, Script Sql

3.6.2.3. Calificación

Tabla XIX Calificación del Indicador 2: Seguridad, se basarán en la calificación de las Tablas ()

	SEGURIDAD					
	PgAdminIII		PHPpgAdmin		EMS SQL Lite for PostgreSQL	
	Cuantitati vo	Cualitativ o	Cuantitati vo	Cualitativ o	Cuantitati vo	Cualitativ o
Tipo de autenticación usada	4	Muy Satisfactorio	2	Poco Satisfactorio	4	Muy Satisfactorio
Tipos de algoritmos de seguridad	4	Muy Satisfactorio	3	Satisfactorio	3	Satisfactorio

Nivel de seguridad para establecer sesión	4	Muy Satisfactorio	2	Poco Satisfactorio	4	Muy Satisfactorio
Tipos de soporte de respaldo y restauración	4	Muy Satisfactorio	2	Poco Satisfactorio	3	Satisfactorio

Para sacar los porcentajes de cada índice se utilizará las siguientes formulas.

Dónde:

X = Equivale a PgAdminIII.

Y = Equivale a PHPpgAdmin

Z = Equivale a EMS SQL Lite forPostgreSQL.

C1(X)= Puntaje acumulativo del Tipo de autenticación usada de PgAdminIII

C2(X)= Puntaje acumulativo de Tipos de algoritmos de seguridad de PgAdminIII

C3(X)= Puntaje acumulativo Nivel de seguridad para establecer sesión de PgAdminIII

C4(X)= Puntaje acumulativo Tipos de soporte de respaldo y restauración de PgAdminIII

C1 (Y)= Puntaje acumulativo del Tiempo Tipo de autenticación usada de PHPpgAdmin

C2 (Y)= Puntaje acumulativo de Tipos de algoritmos de seguridad de PHPpgAdmin

C3 (Y)= Puntaje acumulativo Nivel de seguridad para establecer sesión de PHPpgAdmin

C4 (Y)= Puntaje acumulativo Tipos de soporte de respaldo y restauración dePHPpgAdmin

C1 (Z)= Puntaje acumulativo del Tipo de autenticación usada de EMS SQL Lite forPostgreSQL.

C2 (Z)= Puntaje acumulativo de Tipos de algoritmos de seguridad de EMS SQL Lite forPostgreSQL.

C3 (Z)= Puntaje acumulativo Nivel de seguridad para establecer sesión de EMS SQL Lite forPostgreSQL.

C4 (Z)= Puntaje acumulativo Tipos de soporte de respaldo y restauración.deEMS SQL Lite forPostgreSQL.

Vi= Resultado de las pruebas.

$$\left[C1(X) = \sum_{i=1}^{n=1} vi ; C2(X) = \sum_{i=1}^{n=1} vi ; C3(X) = \sum_{i=1}^{n=1} vi ; C4(X) = \sum_{i=1}^{n=1} vi \right] \quad [2.1]$$

$$\left[C1(Y) = \sum_{i=1}^{n=1} vi ; C2(Y) = \sum_{i=1}^{n=1} vi ; C3(Y) = \sum_{i=1}^{n=1} vi ; C4(Y) = \sum_{i=1}^{n=1} vi \right] \quad [2.2]$$

$$\left[C1(Z) = \sum_{i=1}^{n=1} vi ; C2(Z) = \sum_{i=1}^{n=1} vi ; C3(Z) = \sum_{i=1}^{n=1} vi ; C4(Z) = \sum_{i=1}^{n=1} vi \right] \quad [2.3]$$

Pa(X)= Valor acumulativo de PgAdminIII

Pa (Y)= Valor acumulativo de PHPpgAdmin

Pa (Z)= Valor acumulativo de EMS SQL Lite forPostgreSQL.

$$\left[\begin{array}{l} Pa(X) = \sum C1(X) + C2(X) + C3(X) + C4(X) \\ Pa(Y) = [\sum C1(Y) + C2(Y) + C3(Y) + C4(Y)] \\ Pa(Z) = [\sum C1(Z) + C2(Z) + C3(Z) + C4(Z)] \end{array} \right] \quad [2.4]$$

PpT (X)= Porcentaje Parcial Total de PgAdminIII

PpT (Y)= Porcentaje Parcial Total de PHPpgAdmin

PpT (Z)= Porcentaje Parcial Total de EMS SQL Lite forPostgreSQL

Vp= Valor máximos de índices es de 4

$$\left[\begin{array}{l} PpT(X) = \left(\frac{Pa(X)}{Vp} \right) * 100 \\ PpT(Y) = \left(\frac{Pa(Y)}{Vp} \right) * 100 \\ PpT(Z) = \left(\frac{Pa(Z)}{Vp} \right) * 100 \end{array} \right] \quad [2.5]$$

Tabla XX Valores y Porcentajes del Indicador 2: Seguridad, con sus respectivos Índices

	SEGURIDAD					
	PgAdminIII		PHPpgAdmin		EMS SQL Lite for PostgreSQL	
	Valor (X)	% PpT	Valor (Y)	% PpT	Valor (Z)	% PpT
Tipo de autenticación usada.C1	4	100	2	50	4	100
Tipos de algoritmos de seguridad.C2	4	100	3	75	3	100
Nivel de seguridad para establecer sesión.C3	4	100	2	50	4	100
Tipos de soporte de respaldo y restauraciónC4	4	100	2	50	3	75

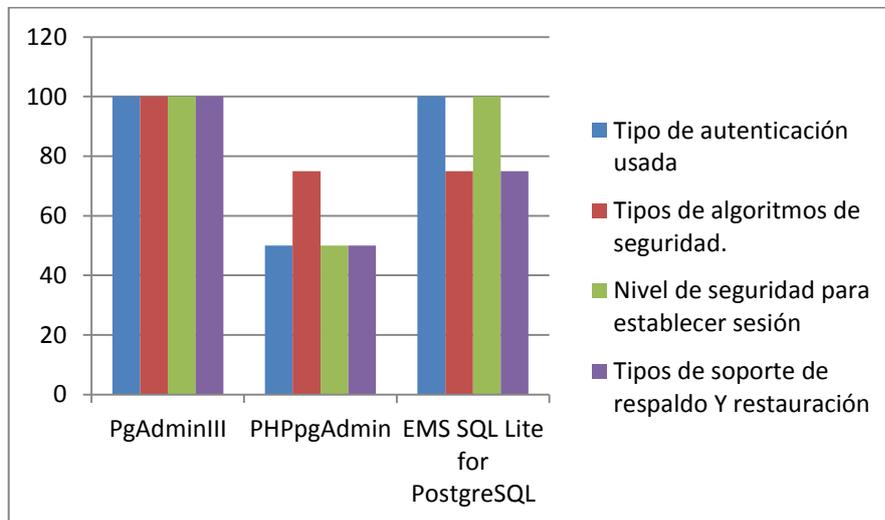


Figura III.31 Resultado por índice del Indicador 2: Seguridad

Para sacar los porcentajes de cada índice se utilizará las siguientes formulas.

Dónde:

PT (X)= Porcentaje Total de PgAdminIII

PT (Y)= Porcentaje Total de PHPpgAdmin

PT (Z)= Porcentaje Total de EMS SQL Lite forPostgreSQL

VA= Valor máximos de índices es de 16

$$\left(\begin{array}{l} \mathbf{PT(X)} = \left(\frac{\mathbf{Pa(X)}}{\mathbf{VA}} \right) * \mathbf{100} \\ \mathbf{PT(Y)} = \left(\frac{\mathbf{Pa(Y)}}{\mathbf{VA}} \right) * \mathbf{100} \\ \mathbf{PT(Z)} = \left(\frac{\mathbf{Pa(Z)}}{\mathbf{VA}} \right) * \mathbf{100} \end{array} \right) \quad [2.6]$$

Tabla XXI Valores y Porcentajes Finales del Indicador 2: Seguridad.

	Seguridad	
	Valor (Pa)	% (PT)
PgAdminIII: (X)	16	100
PHPpgAdmin: (Y)	9	56,25
EMS SQL Lite for PostgreSQL: (Z)	14	87,5

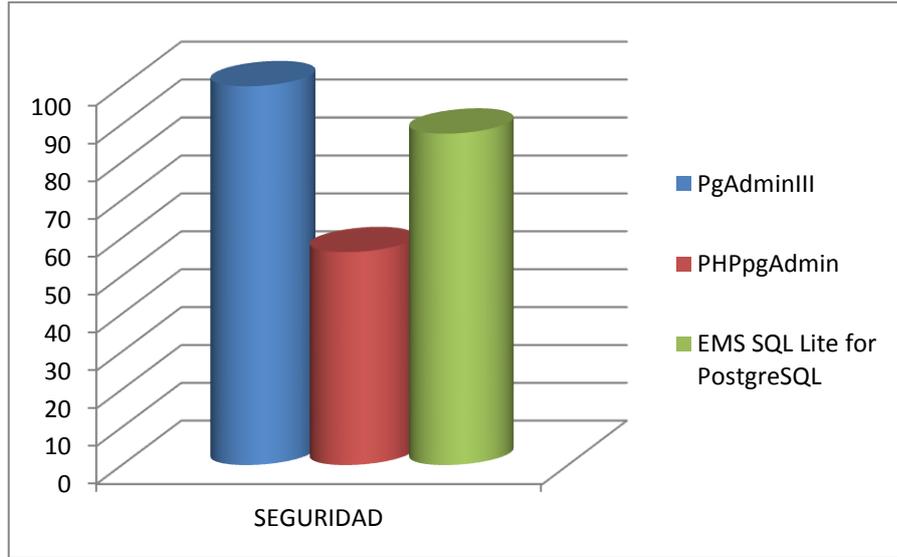


Figura III.32 Resultado Final del Indicador 2: Seguridad

3.6.2.4. Interpretación

Tabla XXII Representación del Indicador 2: Seguridad

	SEGURIDAD		
	PGADMINIII	PHPPGADMIN	EMS SQL LITE FOR POSTGRESQL
Tipo de autenticación usada	♣♣♣♣	♣♣	♣♣♣♣
Tipos de algoritmos de seguridad	♣♣♣♣	♣♣♣	♣♣♣
Nivel de seguridad para establecer sesión.	♣♣♣♣	♣♣	♣♣♣♣
Tipos de soporte de respaldo y restauración	♣♣♣♣	♣♣	♣♣♣

3.6.2.5.Descripción Resultados

Los resultados expuestos en la tabla XXIV “representación del Indicador 2: Seguridad” serán explicados por índices.

Tipo de autenticación usada: Es importante al momento de autenticarse con la base de datos postgresql, por lo tanto se establece que tanto PgAdminIII y EMS SQL Lite para Postgresql tienen similares características de autenticación sea local o remota estas herramientas mantienen una concordancia entre sí, mientras que PhppgAdmin solo tiene un modo de autenticación con respecto a las dos anteriores.

Tipos de algoritmos de seguridad: Guiándose en los resultados de la tabla de referencia podemos afirmar que PgAdminIII posee algoritmos de seguridad superiores con respecto a EMS SQL Lite para Postgresql y PhpPgAdmin además estos dos últimos poseen los mismos algoritmos de seguridad.

Nivel de seguridad para establecer sesión: Los mejores puntajes obtienen PgAdminIII y EMS SQL Lite para Postgresql con 4 puntos ya que su nivel de seguridad es similar al tener certificados de autenticación SSL y siempre solicitar un nombre de usuario y contraseña respectiva antes de ingresar a manipular las bases de datos creadas el control de acceso a las bases de datos lo realiza por medio de una clave de acceso al sistema y así administrar los recursos que esta tenga seguidos por PhpPgAdmin con 2 puntos debido a que este no posee certificados de autenticación para su acceso respectivo.

Tipos de soporte de respaldo y restauración: Destacando, nuevamente a PgAdminIII, por mantener una ayuda directa en cada paso al momento de realizar respaldos de una base de datos lo mismo ocurre a restaurar la misma, EMS SQL Lite para Postgresql posee modos de dar respaldo y restaurar bases de datos ya sea de manera gráfica generando sentencias sql de manera automática o manualmente por medios de sentencias sql, PhppgAdmin queda en último lugar ya que solo posee un modo de realizar respaldos de bases de datos y restaurarlos de manera gráfica.

3.6.3. Indicador 3: Administración

3.6.3.1. Determinación del Indicador

La sección de Administración posee 4 índices que serán evaluados tomando sus valores mínimos, máximos para cada índice. De esta manera se realiza una escala de valorización y determinar los valores no satisfactorio, poco satisfactorio, satisfactorio y muy satisfactorio respectivamente.

Tabla XXIII Valorización para el Indicador 3: Administración

Cuantitativa	0	1	2	3	4
	NO				SI
Cualitativa	Malo	Regular	Bueno	Medianamente Bueno	Muy Bueno
	Baja	Regular	Buena	Medianamente Alta	Muy Alta
	Muy Dificil	Difícil	Medianamente Fácil	Fácil	Muy Fácil
Representativo	X	♣	♣♣	♣♣♣	♣♣♣♣

- **Índice 3.1: Conexión local:** En este índice se valorizará para cada tipo de autenticación sea local o remota dependiendo de cada una de las Herramientas.

Tabla XXIV Valorización del Índice 3.1: Conexión local

Modos de Conexión	Valor Cualitativo
Si	Muy Bueno
No	Malo

- **Índice 3.2: Conexión remota:** En este índice se valorizará para cada una de las configuraciones necesarias para la conexión y acceso a las bases de datos.

Tabla XXV Valorización del Índice 3.2: Conexión remota

Modo de Conexión	Valor Cualitativo
Si	Muy Bueno
No	Malo

- **Índice 3.3: Facilidad diseño de Base de Datos:** En este índice se valorizará para cada una de las Herramientas las maneras de cómo se puede crear bases de datos.

Tabla XXVI Valorización del Índice 3.3: Facilidad diseño de Base de Datos.

Modos de Creación	Valor Cualitativo
EditorSql	Bueno
Interfaz	Medianamente Bueno
Interfaz, EditorSql	Muy Bueno

- **Índice 3.4: Facilidad diseño de Tablas:** En este índice se valorizará para cada Herramienta, como se puede crear tablas dentro de una base de datos específica.

Tabla XXVII Valorización del Índice 3.4: Facilidad diseño de Tablas

Modos de Creación	Valor Cualitativo
Editor Sql	Bueno
Interfaz	Medianamente Bueno
Interfaz, Editor Sql	Muy Bueno

- **Índice 3.5: Facilidad diseño de Funciones:** En este índice se valorizará para cada una de las Herramientas como se crean funciones dentro de las bases de datos.

Tabla XXVIII Valorización del Índice 3.5: Facilidad diseño de Funciones

Modos de Creación	Valor Cualitativo
Editor Sql	Bueno
Interfaz	Medianamente Bueno
Interfaz, Editor Sql	Muy Bueno

- **Índice 3.6: Diseño y Consultas de Querys:** En este índice se valorizará para cada una de las Herramientas en caso de poseer los modos y formas que realizan sentencias sql, en las bases de datos.

Tabla XXIX Valorización del Índice 3.6: Diseño y Consultas de Querys

Modos de Creación	Valor Cualitativo
Editor Sql	Bueno
Interfaz	Medianamente Bueno
Interfaz, Editor Sql	Muy Bueno

3.6.3.2. Valoraciones

Tabla XXX Resultados del Indicador 3: Administración, después de la obtención de la ejecución de las diferentes pruebas Ver Anexo 2.

	ADMINISTRACIÓN		
	PgAdminIII	PHPpgAdmin	EMS SQL Lite for PostgreSQL
Conexión local	SI	SI	SI
Conexión remota	SI	NO	SI
Facilidad diseño de Base de Datos.	Interfaz, Editor Sql	Editor Sql	Interfaz, Editor Sql
Facilidad diseño de Tablas	Interfaz, Editor Sql	Interfaz, Editor Sql	Interfaz, Editor Sql
Facilidad diseño de Funciones	Interfaz, Editor Sql	Editor Sql	Interfaz, Editor Sql
Diseño y Consultas de Querys	Interfaz, Editor Sql	Editor Sql	Interfaz, Editor Sql

3.6.3.3. Calificación

Tabla XXXI Calificación del Indicador 3: Administración, se basarán en la calificación de las Tablas (6-10)

ADMINISTRACIÓN						
PgAdminIII		PHPpgAdmin		EMS SQL Lite for PostgreSQL		
Cuantitativo	Cualitativo	Cuantitativo	Cualitativo	Cuantitativo	Cualitativo	Cualitativo
Conexión local	4	MuyBueno	4	MuyBueno	4	MuyBueno
Conexión remota	4	MuyBueno	0	Malo	4	MuyBueno
Facilidad diseño de Base de Datos.	4	MuyBueno	3	Medianamente Bueno	4	MuyBueno
Facilidad diseño de Tablas	4	MuyBueno	3	Medianamente Bueno	4	MuyBueno
Facilidad diseño de Funciones	4	MuyBueno	3	Medianamente Bueno	4	MuyBueno
Diseño y Consultas de Querys	4	MuyBueno	2	Bueno	4	MuyBueno

Para sacar los porcentajes de cada índice se utilizará las siguientes formulas.

Dónde:

X = Equivale a PgAdminIII.

Y = Equivale a PHPpgAdmin

Z = Equivale a EMS SQL Lite forPostgreSQL.

C1(X)= Puntaje acumulativo deConexión local de PgAdminIII

C2(X)= Puntaje acumulativo de Conexión remota de PgAdminIII

C3(X)= Puntaje acumulativo Facilidad diseño de Base de Datos de PgAdminIII

C4(X)= Puntaje acumulativo Facilidad diseño de Tablas de PgAdminIII

C5(X)= Puntaje acumulativo Facilidad diseño de Funciones de PgAdminIII

C6(X)= Puntaje acumulativo Diseño y Consultas de QuerysdePgAdminIII

C1 (Y)= Puntaje acumulativo de Conexión local de PHPpgAdmin

C2 (Y)= Puntaje acumulativo de Conexión remota de PHPpgAdmin

C3 (Y)= Puntaje acumulativo Facilidad diseño de Base de Datos de PHPpgAdmin

C4 (Y)= Puntaje acumulativo Facilidad diseño de TablasdePHPpgAdmin

C5 (Y)= Puntaje acumulativo Facilidad diseño de Funciones de PHPpgAdmin

C6 (Y)= Puntaje acumulativo Diseño y Consultas de QuerysdePHPpgAdmin

C1 (Z)= Puntaje acumulativo de Conexión local de EMS SQL Lite forPostgreSQL.

C2 (Z)= Puntaje acumulativo de Conexión remota de EMS SQL Lite forPostgreSQL.

C3 (Z)= Puntaje acumulativo Facilidad diseño de Base de Datos de EMS SQL Lite forPostgreSQL.

C4 (Z)= Puntaje acumulativo Facilidad diseño de Tablasde EMS SQL Lite forPostgreSQL.

C5 (Z)= Puntaje acumulativo Facilidad diseño de Funciones de EMS SQL Lite forPostgreSQL.

C6 (Z)= Puntaje acumulativo Diseño y Consultas de Querysde EMS SQL Lite forPostgreSQL.

Vi= Resultado de las pruebas.

$$\left[\begin{array}{l} \mathbf{C1}(X) = \sum_{i=1}^{n=1} vi ; \mathbf{C2}(X) = \sum_{i=1}^{n=1} vi ; \mathbf{C3}(X) = \sum_{i=1}^{n=1} vi ; \mathbf{C4}(X) = \\ \sum_{i=1}^{n=1} vi ; \mathbf{C5}(X) = \sum_{i=1}^{n=4} vi ; \mathbf{C6}(X) = \sum_{i=1}^{n=4} vi \end{array} \right] \quad [3.1]$$

$$\left[\begin{array}{l} \mathbf{C1}(Y) = \sum_{i=1}^{n=1} vi ; \mathbf{C2}(Y) = \sum_{i=1}^{n=1} vi ; \mathbf{C3}(Y) = \sum_{i=1}^{n=1} vi ; \mathbf{C4}(Y) = \\ \sum_{i=1}^{n=1} vi ; \mathbf{C5}(Y) = \sum_{i=1}^{n=1} vi ; \mathbf{C6}(Y) = \sum_{i=1}^{n=1} vi \end{array} \right] \quad [3.2]$$

$$\left[\begin{array}{l} \mathbf{C1}(Z) = \sum_{i=1}^{n=1} vi ; \mathbf{C2}(Z) = \sum_{i=1}^{n=1} vi ; \mathbf{C3}(Z) = \sum_{i=1}^{n=1} vi ; \mathbf{C4}(Z) = \\ \sum_{i=1}^{n=1} vi ; \mathbf{C5}(Z) = \sum_{i=1}^{n=1} vi ; \mathbf{C6}(Z) = \sum_{i=1}^{n=1} vi \end{array} \right] \quad [3.3]$$

Pa(X)= Valor acumulativo de PgAdminIII

Pa (Y)= Valor acumulativo de PHPpgAdmin

Pa (Z)= Valor acumulativo de EMS SQL Lite forPostgreSQL.

$$\left(\begin{array}{l} \mathbf{Pa}(X) = \sum \mathbf{C1}(X) + \mathbf{C2}(X) + \mathbf{C3}(X) + \mathbf{C4}(X) + \mathbf{C5}(X) + \mathbf{C6}(X) \\ \mathbf{Pa}(Y) = \sum \mathbf{C1}(Y) + \mathbf{C2}(Y) + \mathbf{C3}(Y) + \mathbf{C4}(Y) + \mathbf{C5}(Y) + \mathbf{C6}(Y) \\ \mathbf{Pa}(Z) = \sum \mathbf{C1}(Z) + \mathbf{C2}(Z) + \mathbf{C3}(Z) + \mathbf{C4}(Z) + \mathbf{C5}(Z) + \mathbf{C6}(Z) \end{array} \right) \quad [3.4]$$

PpT (X)= Porcentaje Parcial Total de PgAdminIII

PpT (Y)= Porcentaje Parcial Total de PHPpgAdmin

PpT (Z)= Porcentaje Parcial Total de EMS SQL Lite forPostgreSQL

Vp= Valor máximos de índices es de 6

$$\left[\begin{array}{l} PpT(X) = \left(\frac{Pa(X)}{Vp} \right) * 100 \\ PpT(Y) = \left(\frac{Pa(Y)}{Vp} \right) * 100 \\ PpT(Z) = \left(\frac{Pa(Z)}{Vp} \right) * 100 \end{array} \right] \quad [3.5]$$

Tabla XXXII Valores y Porcentajes del Indicador 3: Administración, con sus respectivos Índices

	ADMINISTRACIÓN					
	PgAdminIII		PHPpgAdmin		EMS SQL Lite for PostgreSQL	
	Valor (X)	% PpT	Valor (Y)	% PpT	Valor (Z)	% PpT
Conexión local. C1	4	66,66	4	66,66	4	66,66
Conexión remota C2	4	66,66	0	0	4	66,66
Facilidad diseño de Base de Datos. C3	4	66,66	3	50	4	66,66
Facilidad diseño de Tablas C4	4	66,66	3	50	4	66,66
Facilidad diseño de Funciones. C5	4	66,66	3	50	4	66,66
Diseño y Consultas de QuerysC6	4	66,66	2	33,33	4	66,66

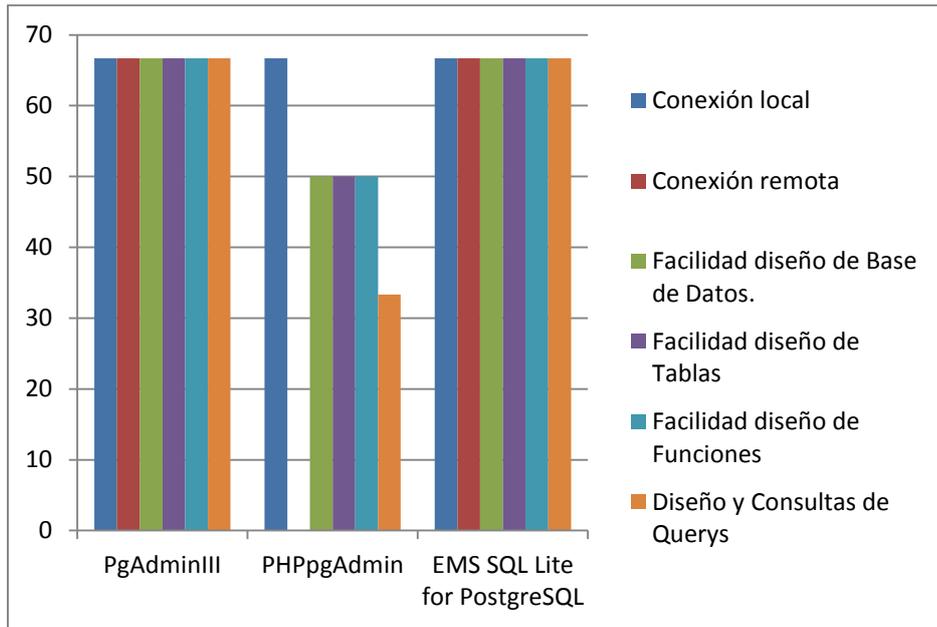


Figura III. 33 Resultado por índice del Indicador 3: Administración

Para sacar los porcentajes de cada índice se utilizará las siguientes formulas.

Dónde:

PT (X)= Porcentaje Total de PgAdminIII

PT (Y)= Porcentaje Total de PHPpgAdmin

PT (Z)= Porcentaje Total de EMS SQL Lite forPostgreSQL

VA= Valor máximos de índices es de 24

$$\left[\begin{aligned} PT(X) &= \left(\frac{Pa(X)}{VA} \right) * 100 \\ PT(Y) &= \left(\frac{Pa(Y)}{VA} \right) * 100 \\ PT(Z) &= \left(\frac{Pa(Z)}{VA} \right) * 100 \end{aligned} \right] \quad [3.6]$$

Tabla XXXIII Valores y Porcentajes Finales del Indicador 3: Administración.

	Administración	
	Valor (Pa)	% (PT)
PgAdminIII: (X)	24	100
PHPpgAdmin: (Y)	15	62,5
EMS SQL Lite for PostgreSQL: (Z)	24	100

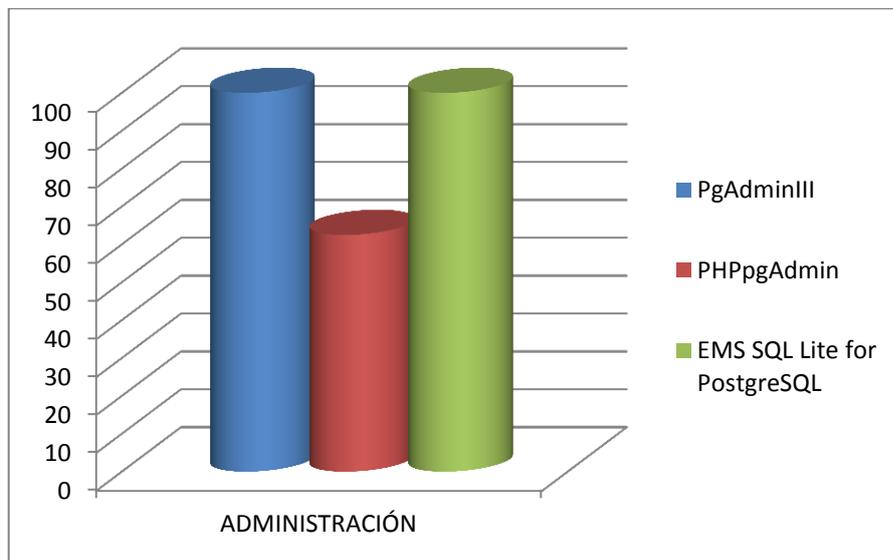


Figura III. 34 Resultado Final del Indicador 3: Administración

3.6.3.4. Interpretación

Tabla XXXIV Representación del Indicador 3: Administración

	ADMINISTRACIÓN		
	PgAdminIII	PHPpgAdmin	EMS SQL Lite for PostgreSQL
Conexión local	♣♣♣♣	♣♣♣♣	♣♣♣♣
Conexión remota	♣♣♣♣	X	♣♣♣♣
Facilidad diseño de Base de Datos.	♣♣♣♣	♣♣♣	♣♣♣♣
Facilidad diseño de Tablas	♣♣♣♣	♣♣♣	♣♣♣♣
Facilidad diseño de Funciones	♣♣♣♣	♣♣♣	♣♣♣♣
Diseño y Consultas de Querys	♣♣♣♣	♣♣	♣♣♣♣

3.6.3.5. Descripción Resultados

Los resultados expuestos en la tabla XXXII “representación del Indicador 3: Administración” serán explicados por índices.

Conexión local: Cuando se instala un servidor de base de datos la manera de acceder a ella es por medio de la misma máquina por tal motivo se puede decir de acuerdo a la tabla que representa al indicador de Administración que tanto PgAdminIII, EMS SQL Lite para Postgresql y PhppgAdmin siempre se accederán a las bases de datos creadas de manera local por eso la calificación otorgada es igual para todas las herramientas.

Conexión remota: Guiándose en los resultados de la tabla de referencia se indica que PgAdminIII y EMS SQL Lite para Postgresql tienen modos similares de conectarse a una base de datos desde un equipo diferente o de modo remoto para acceder a sus datos mientras que PhpPgAdmin no posee esta característica que es muy importante ya que se puede acceder desde cualquier parte siempre y cuando posea una cuenta y contraseña respectiva para su uso.

Facilidad diseño de Base de Datos: Como se puede apreciar en los resultados los mejores puntajes obtienen PgAdminIII y EMS SQL Lite para Postgresql con 4 puntos cada una debido a las maneras en la que se puede crear y diseñar las bases de datos ya sea por medio de sentencias sql, o de manera gráfica que permite crear bases de datos, PhpPGAdmin queda rezagada debido a que solo se puede crear bases de datos de manera gráfica.

Facilidad diseño de Tablas: Destacando nuevamente a PgAdminIII y EMS SQL Lite para Postgresql con 4 puntos cada una debido a que se pueden crear tablas dentro de las bases de datos de manera gráfica o por sentencias sql, seguido muy de cerca también por PhpPgAdmin con 3 puntos la cual también permite crear tablas dentro de una base de datos.

Facilidad diseño de Funciones: Guiándose en los resultados de la tabla de referencia podemos afirmar nuevamente que PgadminIII y EMS SQL Lite para Postgresql permiten la creación de funciones de manera ágil a varias bases de datos mientras que PHPpgAdmin permite crear funciones a una sola base de datos creada ya anteriormente

Diseño y Consultas de Querys: Los mejores puntajes obtienen PgAdminIII y EMS SQL Lite para Postgresql con 4 puntos ya que el diseño y consultas sql se las puede realizar de manera rápida por medio de la interfaz o editores sql que poseen y crean mientras que PhpPGAdmin con 2 puntos debido a que solo se pueden realizar consultas querys de una sola base de datos.

3.6.4. Indicador 4: Calidad

3.6.4.1. Determinación del Indicador

La sección de Calidad posee 3 índices que serán evaluados tomando sus valores mínimos, máximos para cada índice, así realizar una escala de valorización y determinar los valores no satisfactorio, poco satisfactorio, satisfactorio y muy satisfactorio respectivamente.

Tabla XXXV Valorización para el Indicador 4: Calidad

Cuantitativa	0	1	2	3	4
Cualitativa	NO				SI
	Mal o	Regular	Bueno	Medianament e Bueno	Muy Bueno
	Muy Difí cil	Difícil	Medianame nte Fácil	Fácil	Muy Fácil
Representati vo	X	♣	♣♣	♣♣♣	♣♣♣♣

- **Índice 4.1: Uso de Recursos Hardware:** En este índice se valorizará para cada herramienta el hardware requerido y usado en la administración

Tabla XXXVI Valorización del Índice 4.1: Uso de Recursos Hardware

Porcentajes de Recursos Mínimos	Valor Cualitativo
0%-20%	Muy Bueno
21%-40%	Medianamente Bueno
41%-60%	Bueno
61%-80%	Regular
81%-100%	Malo

- **Índice 4.2: Facilidad de Uso:** En este índice se valorizará para cada una de las configuraciones necesarias en la facilidad de uso de la Herramienta.

Tabla XXXVII Valorización del Índice 4.2: Facilidad de Uso

Valor Cuantitativo	Valor Cualitativo
0	Muy Difícil
1	Difícil
2	Medianamente Fácil
3	Fácil
4	Muy Fácil

Tabla XXXVIII Preguntas y Consulta del Índice 4.2: Facilidad de Uso

Pregunta	PgAdminIII (Respuestas)	PHPpgAdmin (Respuestas)	EMS SQL Lite forPostgreSQL (Respuestas)
¿Considera usted que al abrir el software por primera vez, la función que cumple el mismo fue?			
Incomprensible ()	0	0	0
Medianamente Incomprensible ()	0	0	0
Comprensible()	4	5	10
Medianamente Comprensible()	12	11	3
Muy Comprensible()	0	0	0
¿Según usted los servicios de ayuda y realimentación en línea son?			
Malo ()	0	0	0
Regular ()	0	0	
Bueno ()	2	8	2
Muy Bueno ()	15	0	15
Excelente ()	0	0	0
¿Luego de utilizar la herramienta su nivel de satisfacción es?			

Malo ()	0	0	0
Regular ()	0	0	0
Bueno ()	10	8	5
Muy Bueno ()	3	6	8
Excelente ()	0	0	0
¿Para usted la interface presentada por las herramientas es?			
Desagradable ()	0	0	0
Medianamente desagradable()	0	0	0
Agradable ()	10	8	5
Medianamente agradable ()	6	8	3
Muy agradable ()	0	0	0
¿Los íconos y colores de las herramientas que conforman la interfaz indican las acciones que realizan?			
No ()	0	0	0
Pocas herramientas ()	0	0	0
Algunas herramientas ()	8	7	10
La mayoría de herramientas ()	6	8	4
Todas las herramientas ()	0	0	0
¿Considera usted que el manejo del software es?			

Muy difícil ()	0	0	0
Difícil ()	0	0	0
Fácil ()	18	15	12
Muy Fácil ()	0	0	0
Resultado	Medianamente Fácil	Medianamente Fácil	Medianamente Fácil

- **Índice 4.3: Compatibilidad Software:** En este índice se valorizará para cada una de las Herramientas en la posibilidad que ofrece el software de abrir y/o editar un diseño en otro programa con similares funcionalidades en base a la siguiente valoración.

Tabla XXXIX Valorización del Índice 4.3Compatibilidad Software.

Compatibilidad de Software de Similares Características	Valor Cualitativo
Es Compatible	SI
No es compatible	NO

3.6.4.2. Valoraciones

Tabla XL Resultados del Indicador 4: Calidad, después de la obtención de la ejecución de la prueba Ver Anexo 2.

	CALIDAD								
	PgAdminIII			PHPpgAdmin			EMS SQL Lite for PostgreSQL		
	Procesador	Memoria RAM	Promedio	Procesador	Memoria RAM	Promedio	Procesador	Memoria RAM	Promedio
Uso de Recursos Hardware	8 %	37%	22,5 %	7 %	40%	23,5%	50%	38%	44%
Facilidad de Uso	Medianamente Fácil			Medianamente Fácil			Medianamente Fácil		
Compatibilidad Software	SI			SI			SI		

3.6.4.3. Calificación

Tabla XLICalificación del Indicador 4: Calidad, se basarán en la calificación de las Tablas (6-10)

	CALIDAD					
	PgAdminIII		PHPpgAdmin		EMS SQL Lite for PostgreSQL	
	Cuantitativo	Cualitativo	Cuantitativo	Cualitativo	Cuantitativo	Cualitativo
Uso de Recursos Hardware	3	Mediana menteBue no	3	Mediana menteBue no	2	Bueno
Facilidad de Uso	2	Mediana menteFáci 1	2	Mediana menteFáci 1	2	Medianam enteFácil
Compatibilidad Software	4	SI	4	SI	4	SI

Para sacar los porcentajes de cada índice se utilizará las siguientes formulas.

Dónde:

X = Equivale a PgAdminIII.

Y = Equivale a PHPpgAdmin

Z = Equivale a EMS SQL Lite forPostgreSQL.

C1(X)= Puntaje acumulativo de Uso de Recursos Hardware de PgAdminIII

C2(X)= Puntaje acumulativo de Facilidad de Uso de PgAdminIII

C3(X)= Puntaje acumulativo Compatibilidad Software de PgAdminIII

C1 (Y)= Puntaje acumulativo de Uso de Recursos Hardware de PHPpgAdmin

C2 (Y)= Puntaje acumulativo de Facilidad de Uso de PHPpgAdmin

C3 (Y)= Puntaje acumulativo Compatibilidad Software de PHPpgAdmin

C1 (Z)= Puntaje acumulativo de Uso de Recursos Hardware de EMS SQL Lite forPostgreSQL.

C2 (Z)= Puntaje acumulativo de Facilidad de Uso de EMS SQL Lite forPostgreSQL.

C3 (Z)= Puntaje acumulativo Compatibilidad Software de EMS SQL Lite forPostgreSQL.

Vi= Resultado de las pruebas.

$$\left[\begin{array}{l} \mathbf{C1(X)} = \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C2(X)} = \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C3(X)} = \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C4(X)} = \\ \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C5(X)} = \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C6(X)} = \sum_{i=1}^{n=1} \mathbf{vi} \end{array} \right] \quad [4.1]$$

$$\left[\begin{array}{l} \mathbf{C1(Y)} = \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C2(Y)} = \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C3(Y)} = \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C4(Y)} = \\ \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C5(Y)} = \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C6(Y)} = \sum_{i=1}^{n=1} \mathbf{vi} \end{array} \right] \quad [4.2]$$

$$\left[\begin{array}{l} \mathbf{C1(Z)} = \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C2(Z)} = \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C3(Z)} = \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C4(Z)} = \\ \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C5(Z)} = \sum_{i=1}^{n=1} \mathbf{vi} ; \mathbf{C6(Z)} = \sum_{i=1}^{n=1} \mathbf{vi} \end{array} \right] \quad [4.3]$$

Pa(X)= Valor acumulativo de PgAdminIII

Pa (Y)= Valor acumulativo de PHPpgAdmin

Pa (Z)= Valor acumulativo de EMS SQL Lite forPostgreSQL.

$$\left(\begin{array}{l} Pa(X) = \sum C1(X) + C2(X) + C3(X) \\ Pa(Y) = \sum C1(Y) + C2(Y) + C3(Y) \\ Pa(Z) = \sum C1(Z) + C2(Z) + C3(Z) \end{array} \right) \quad [4.4]$$

PpT (X)= Porcentaje Parcial Total de PgAdminIII

PpT (Y)= Porcentaje Parcial Total de PHPpgAdmin

PpT (Z)= Porcentaje Parcial Total de EMS SQL Lite forPostgreSQL

Vp= Valor máximos de índices es de 4

$$\left(\begin{array}{l} PpT(X) = \left(\frac{Pa(X)}{Vp} \right) * 100 \\ PpT(Y) = \left(\frac{Pa(Y)}{Vp} \right) * 100 \\ PpT(Z) = \left(\frac{Pa(Z)}{Vp} \right) * 100 \end{array} \right) \quad [4.5]$$

Tabla XLII Valores y Porcentajes del Indicador 4: Calidad con sus respectivos Índices

	CALIDAD					
	PgAdminIII		PHPpgAdmin		EMS SQL Lite for PostgreSQL	
	Valor (X)	% PpT	Valor (Y)	% PpT	Valor (Z)	% PpT
Uso de Recursos Hardware C1	3	75	3	75	2	50
Facilidad de Uso C2	2	50	2	50	2	50
Compatibilidad Software C3	4	100	4	100	4	100

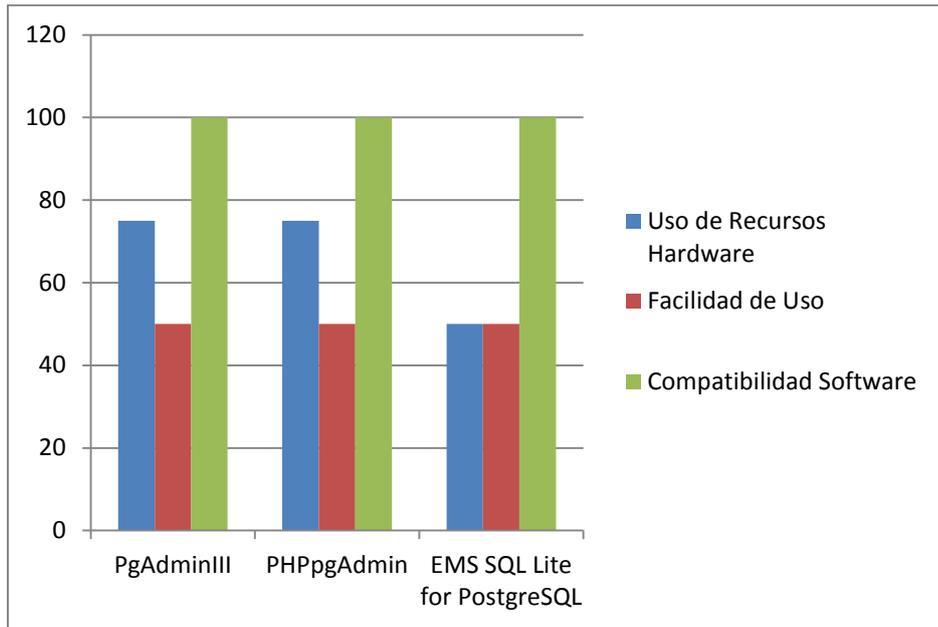


Figura III.35 Resultado por índice del Indicador 4: Calidad

Para sacar los porcentajes de cada índice se utilizará las siguientes formulas.

Dónde:

PT (X)= Porcentaje Total de PgAdminIII

PT (Y)= Porcentaje Total de PHPPgAdmin

PT (Z)= Porcentaje Total de EMS SQL Lite forPostgreSQL

VA= Valor máximos de índices es de 12

$$\left(\begin{array}{l} PT(X) = \left(\frac{Pa(X)}{VA} \right) * 100 \\ PT(Y) = \left(\frac{Pa(Y)}{VA} \right) * 100 \\ PT(Z) = \left(\frac{Pa(Z)}{VA} \right) * 100 \end{array} \right) \quad [4.6]$$

Tabla XLIII Valores y Porcentajes Finales del Indicador 4: Calidad.

	Calidad	
	Valor (Pa)	% (PT)
PgAdminIII: (X)	9	75
PHPpgAdmin: (Y)	9	75
EMS SQL Lite for PostgreSQL: (Z)	8	66,66

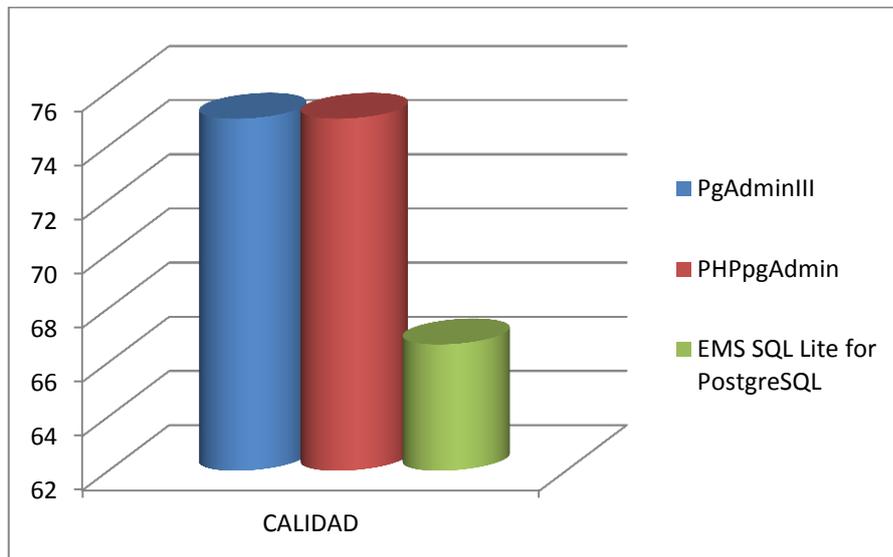


Figura III.36 Resultado Final del Indicador 4: Calidad

3.6.4.4. Interpretación

Tabla XLIV Representación del Indicador 4: Calidad

	CALIDAD		
	PgAdminIII	PHPpgAdmin	EMS SQL Lite for PostgreSQL
Uso de Recursos Hardware	♣♣♣	♣♣♣	♣♣
Facilidad de Uso	♣♣	♣♣	♣♣
Compatibilidad Software	♣♣♣♣	♣♣♣♣	♣♣♣♣

3.6.4.5. Descripción Resultados

Los resultados expuestos en la tabla XLII “representación del Indicador 4: Calidad” serán explicados por índices.

Uso de Recursos Hardware: Cuando se utilizan cada una de las herramientas los recursos hardware requeridos por el sistema son los que se describen en la tabla de representación del indicador Calidad indicando que se PgadminIII y PhppgAdmin con una calificación de 3 puntos su uso de recursos es medianamente bueno en comparación con lo que necesita EMS SQL Lite para Postgresql con 2 puntos indicando que ocupa un poco más de recursos, en comparación de los dos anteriormente mencionadas.

Facilidad de Uso: Guiándose en los resultados de la tabla de referencia se indica que todas las herramientas consiguieron el mismo puntaje debido a que es medianamente

fácil utilizarlas por presentan una interfaz intuitiva cómoda para la comunicación con el usuario se requiere además de conocimientos previos de administración para sacar el mejor provecho al momento de administrar bases de datos con ellas

Compatibilidad Software: Como se puede apreciar en los resultados los mejores puntajes los obtuvieron todas las herramientas indicando que es posible utilizar el mismo lenguaje sql y esté disponible para utilizar en cualquier sistema multiplataforma y con software de características similares.

3.7. Puntajes Alcanzados

Tabla XXXIX Cuadro Resultados de Indicador e Índice

INDICADORES	INDICES	HERRAMIENTAS GUI POSTGRESQL			
		PROMEDIO	PgAdminIII	PHPpgAdmin	EMS SQL Lite for PostgreSQL
Instalación	Tiempo en Completar Instalación	4	4	3	4
	Pasos de Configuración	4	3	2	3
	Nivel de Control de Acceso	4	4	4	4
	Nivel de ayuda en los procesos de instalación.	4	4	3	4

Seguridad	Tipo de autenticación usada	4	4	2	4
	Tipos de algoritmos de seguridad	4	4	3	3
	Nivel de seguridad para establecer sesión	4	4	2	4
	Tipos de soporte de respaldo y restauración	4	4	2	3
Administración	Conexión local	4	4	4	4
	Conexión remota	4	4	0	4
	Facilidad diseño de Base de Datos.	4	4	3	4
	Facilidad diseño de Tablas	4	4	3	4
	Facilidad diseño de Funciones	4	4	3	4
	Diseño y Consultas de Querys	4	4	2	4
Calidad	Uso de recursos hardware	4	3	2	1
	Facilidad de Uso	4	4	0	4
	Compatibilidad Software	4	4	3	4
TOTAL		68	66	41	62

3.8. Interpretación

Las tabulaciones de los resultados obtenidos en las pruebas dan a conocer el comportamiento global de cada herramienta en los diferentes escenarios expuestos en el desarrollo de este análisis comparativo.

La instalación fue la encargada de probar la utilización de los recursos de hardware al momento de realizar todas las operaciones de los escenarios planteados anteriormente,

donde se destaca a la herramienta PgAdminIII ya que su principal función es un desarrollo rápido y seguro de bases de datos.

La seguridad es el indicador encomendado a dar un punto base de partida que es lo primordial al momento de proteger información, es decir el tipo de autenticación usada, los algoritmos de seguridad necesarios para establecer sesión y el tipo de soporte para respaldar dicha información, donde se destaca PgAdminIII seguido muy de cerca de EMS SQL Lite forPostgreSQL, ya que ambos poseen maneras de utilizar certificados de seguridad necesarios en la protección de la información.

La Administración es el punto principal para poder diseñar bases de datos con los permisos y recursos necesarios dentro de ellas donde PgAdminIII y EMS SQL Lite forPostgreSQL son las mejores en el manejo de base de datos postgresql porque poseen varias maneras de gestionarlas.

La Calidad también juega un papel muy importante ya que cada una de las herramientas al momento de utilizarlas debe ser rápidas y ligeras sin ocupar muchos recursos de nuestro sistema donde cabe destacar que en esta ocasión PgAdminIII en los resultados fue ligeramente mejor que PhpPgAdmin ya que posee como una de sus características la seguridad y sobre todo compatibilidad necesaria con otras bases de datos.

Los resultados son claros que cada herramienta fue desarrollada para gestionar bases de datos postgresql, ya que sus componentes y características están enfocados a ciertos tipos de desarrolladores, PgAdminIII y EMS SQL Lite forPostgreSQL están altamente recomendados para el desarrollo de bases de datos ya sean estas grandes o pequeñas en

cambio que PhpPgAdmin está orientada a la web en una menor escala ya que no posee una cantidad extensa de los elementos que las otras herramientas poseen.

3.9. Análisis de Resultados

Luego de la interpretación de los resultados realizado anteriormente destaca la herramienta PgAdminIII como la ideal para realizar el desarrollo del Sistema Web de Convenios Epoch con los objetivos planteados al principio de la tesis, ya que PgAdminIII tiene una completa gama de facilidades al momento de realizar y habilitar la conexión de base de datos postgresql, las mismas que permitan realizar la conexión de formularios y validación de datos, manejo de sesiones y perfiles que ayudará optimizar y comprobar el desempeño de nuestra aplicación web.

En conclusión por tener un dominio amplio sobre las demás herramientas de administración para postgresql y contando con la experiencia como usuarios desarrolladores y de haber utilizado las 3 Herramientas para el desarrollo de aplicaciones web, se concluye que la herramienta de administración gráfica para postgresqlPgAdminIII es el más adecuado para el desarrollo de la aplicación web.

3.10. Comprobación de Hipótesis

La hipótesis planteada es:

H1: La utilización de Herramientas de Administración de Base de Datos adecuada para el sistema Web permitirá mejorar la Administración de Bases de Datos en Postgresql.

3.10.1. Análisis Comparativo en porcentajes de las herramientas de administración para postgresql, PgAdminIII, PhpPgAdmin y EMS SQL Lite forPostgreSQL.

Tabla XLV Cuadro Resultados de Indicador e Índice

INDICADORES	INDICES	CON HERRAMIENTA S GUI POSTGRESQL			SIN HERRAMIENTA
		PgAdminIII	PHPpgAdmin	EMS SQL Lite for PostgreSQL	MODO CONSOLA
Instalación	Tiempo en Completar Instalación	4	3	4	4
	Pasos de Configuración	3	2	3	0
	Nivel de Control de Acceso	4	4	4	3
	Nivel de ayuda en los procesos	4	3	4	1

	de instalación.				
Seguridad	Tipo de autenticación usada	4	2	4	1
	Tipos de algoritmos de seguridad	4	3	3	3
	Nivel de seguridad para establecer sesión	4	2	4	1
	Tipos de soporte de respaldo y restauración	4	2	3	2
Administración	Conexión local	4	4	4	4
	Conexión remota	4	0	4	4
	Facilidad diseño de Base de Datos.	4	3	4	0
	Facilidad diseño de Tablas	4	3	4	0
	Facilidad diseño de Funciones	4	3	4	0
	Diseño y Consultas de Querys	4	2	4	0
Calidad	Uso de recursos hardware	3	2	1	3
	Facilidad de Uso	4	0	4	4
	Compatibilidad Software	4	3	4	3
TOTAL		66	41	62	33

Resultados del Análisis Comparativo de las herramientas de administración para postgresql, PgAdminIII, PhpPgAdmin y EMS SQL Lite forPostgreSQL vs Modo Consola de postgresql expresado en porcentajes.

Tabla LVI Valores y Porcentajes Finales

	TOTAL	
	Valor	Porcentaje (%)
PgAdminIII	66	97,05
PhpPgAdmin	41	60,29
EMS SQL Lite for PostgreSQL	62	91,17
ConsolaPostgresql	33	48,52

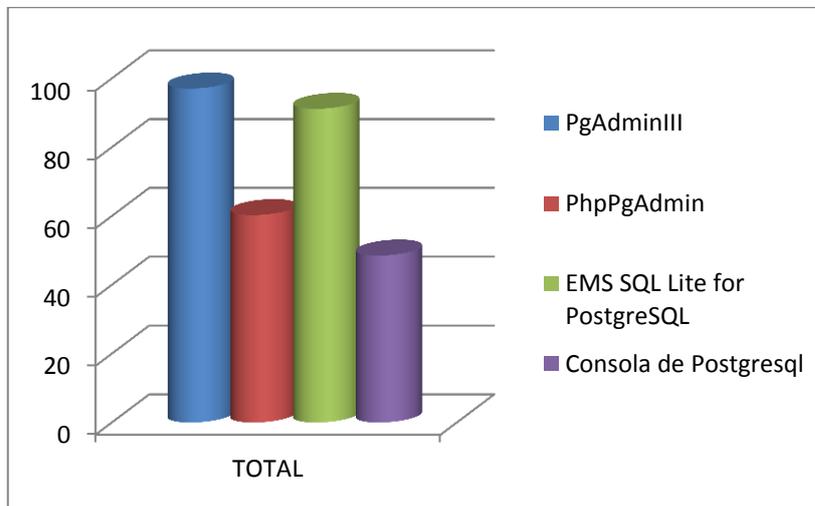


Figura III.37 Resultado Herramientas y no herramientas para postgresql

Haciendo referencia a la tabla “LVI” se puede observar que utilizando herramientas gráficas de administración en postgresql y comparándola con el modo de administración en consola que tiene postgresql, se obtiene que

PgAdminIII con un 97,05% vs Modo Consola con un 48,52% se tiene una diferencia 48,53% esto quiere decir que PgAdminIII es mucho mejor que el modo consola de postgresql.

PHPPgAdmin con un 60,29% vs Modo Consola 48,52% con una diferencia 11,77% esto indica nuevamente que PhpPgAdmin ha superado a la consola de postgresql EMS SQL Lite for PostgreSQL con un 91,17% vs Modo Consola 48,52% con una diferencia de 42,65% mostrando nuevamente que estas herramientas son buenas para la administración de bases de datos postgresql.

Tabla LVII Porcentajes Finales con y sin Herramientas administración

PROMEDIO DE PORCENTAJES (%)	
CON HERRAMIENTAS GUI POSTGRESQL	82,82
SIN HERRAMIENTAS GUI POSTGRESQL	48,52

3.10.2. Conclusión de la Comprobación de la Hipótesis

Haciendo referencia a la tabla "LVII" y por observación directa se concluye que en la hipótesis planteada: Al utilizar las herramientas de administración para postgresqlPgAdminIII, EMS SQL Lite for PostgreSQL, PhpPgAdmin, se concluye que las herramientas son más idóneas para la administración de base de datos en postgresql ya que tienen las mejores características de acuerdo a los resultados obtenidos por los indicadores planteados con un 82,82%, debido a que ofrecen una muy buena administración y los datos de las base de datos pueden estar mucho más seguros.

CAPÍTULO IV

DESARROLLO DEL PORTAL

Al momento del desarrollo del sistema web para el departamento de vinculación de la ESPOCH, se va a proceder a diseñar completamente el sistema web, siguiendo los objetivos y alcances planteados en nuestra justificación práctica.

La aplicación contempla el diseño de la apariencia web con el uso de menús amigables, gestión de usuarios como área de registro y de login, para realizar el ingreso de convenios y seguimiento de ellos se tendrá la sección de personal responsable de los convenios, fecha que inicia y termina un convenio, renovaciones de convenios según el caso y se podrá mantener actualizados a los datos de los convenios.

El sistema archiva todos los convenios registrados que se podrán revisar instantáneamente similar a un registro cronológico hasta la fecha, la sección de registro de usuarios se crea para mantener niveles de acceso a la información de los convenios muy importante para tener un control que garantice la privacidad y seguridad del sistema.

4.1. Ingeniería de la información

4.1.1. Definición del ámbito

El departamento de Vinculación de la Escuela Superior Politécnica de Chimborazo en la unidad Convenios se maneja el control de los convenios que se generan en la institución. Este control lo realiza para llevar un ordenamiento cronológico de los convenios que se almacenan en dicha unidad con el fin de obtener la información necesaria cuando sea solicitada.

Este proceso se lo realiza actualmente mediante matrices que se almacena en Word y Excel donde detallan los ingresos de los convenios previamente aprobados por las dependencias responsables provenientes de las coordinaciones, unidades y facultades que conforman la ESPOCH, persona responsable del convenio, beneficiarios de los convenios.

La información almacenada es extensa debido a ciertas redundancias que existe además del consumo excesivo de recursos materiales, talento humano y tiempo.

4.1.2. Identificar Requerimientos

Para solucionar los inconvenientes que se presentan el manejo del sistema de archivo se ha establecido los siguientes requerimientos:

REQ1:El sistema deberá permitir la autenticación de los usuarios.

REQ2: El sistema permitirá el ingreso de un nuevo usuario.

REQ3: El sistema permitirá la modificación de un usuario.

REQ4: El sistema permitirá eliminar un usuario.

REQ5:El sistema permitirá el registro de los convenios.

REQ6:El sistema permitirá mantener un registro de los coordinadores de los convenios

REQ7:El sistema permitirá llevar un registro de las actividades que se desarrollan en los convenios.

REQ8:El sistema permitirá registrar información de los beneficiarios de los convenios.

REQ9:El sistema permitirá mantener un registro de los beneficios que se logran mediante los convenios.

REQ10:El sistema permitirá gestionar los datos de las entidades con las cuales se realizan los convenios.

REQ11:El sistema permitirá mantener un registro de los reportes emitidos

REQ12:El sistema permitirá mantener un registro de los informes de los convenios.

REQ13:El sistema permitirá mantener un registro de los convenios que presenten problemas en su cumplimiento.

REQ14:El sistema permitirá llevar un control de evaluación de convenios.

4.1.3. Estudio de Factibilidad

4.1.3.1. Factibilidad Económica

El departamento de Comisión de Vinculación con la Colectividad posee todos los equipos hardware y software necesarios por lo que no se tendrá que realizar ninguna inversión para conseguir recursos.

4.1.3.2. Factibilidad Técnica

- **Recursos humanos**

Tabla XLVI Recursos Humanos

Recurso Humano	Cargo
Ing. Danilo Pastor	Director
Ing. Víctor Viñan	Jefe de Comisión de Vinculación con la Colectividad
Edgar Cuenca	Desarrollador

- **Recursos Software**

Tabla XLVII Recursos Software

Recurso Software	Característica
Sistema Operativo	Microsoft Windows SevenUltimate
Servidor de Base de Datos	Postgresql 9.0
Herramientas de desarrollo	Netbeans 7
Tecnología	Java (Spring MVC)

4.1.3.3. Factibilidad Operativa

Para desarrollar este sistema web se cuenta con el apoyo del jefe de la Unidad Comisión de Vinculación con la Colectividad, de la persona encargada de la Unidad de Vinculación y del recurso humano ya mencionado.

4.1.3.4. Factibilidad Legal

Se tiene el permiso de las autoridades respectivas por lo que no existe ningún tipo de impedimento legal para el desarrollo de la aplicación.

4.1.4 Planificación Temporal

Ver Anexo 4

4.2. Análisis del sistema

4.2.1. Casos de Uso del Sistema

Una forma para ver el comportamiento de un sistema desde el punto de vista del usuario es con los casos de usos. Los cuales ayudan a determinar los requerimientos representándolos de manera clara, sencilla y comprensiva las funciones que un sistema puede ejecutar.

Los objetivos de los casos de usos del sistema son:

- Determinar el comportamiento que el usuario necesita del sistema.
- Relacionar cada actor con las acciones que quiere conseguir.
- Aplicar las relaciones que existen entre los casos de uso en el sistema propuesto.

Actores que intervienen y sus funciones

Tabla XLVIII Actores y sus funciones

Actor	Función
Usuario Convenio (administrador del sistema general)	Verifica la información de los convenios que llegan de los coordinadores. Crea general de convenios. Ingresa, modifica y elimina información de los convenios. Consulta de los convenios referente de los distintos coordinadores.
Usuario Coordinador (secretaria)	Ingresa, modifica y elimina información de los convenios que se generan. Consulta de convenios referente de los distintos coordinadores.

Para la solución del problema se ha identificado los siguientes casos de usos para el sistema de convenios ESPOCH. (CONVENIOS-ESPOCH).

- Autenticación
- Gestión de información de los convenios
- Cancelación convenios
- Consultas

4.2.2. Detalle de los casos de uso identificados

4.2.2.1. Funcionalidad de los casos de uso

Tabla XLIX Caso de uso de Autenticación

Identificación:	Autenticación.	
Actores:	Usuario (Usuario Convenio), Sesión.	
Propósito:	Acceder al sistema.	
Visión General:	El usuario debe estar registrado en el sistema para que pueda ingresar, modificar o eliminar la información de los convenios.	
Tipo: Primario Esencial		
Curso Típico de Eventos:		
Actores		Sistema
1. El usuario ingresa al sistema.		2. Solicita autenticación.
3. Ingresa su nombre y contraseña		4. Verifica los datos ingresados.
Curso Alternativo:		
4. Si los datos ingresados no son correctos solicita que se ingrese nuevamente la información, caso contrario ingresa a su respectiva página de inicio.		

Tabla LCaso uso de gestión de los Convenios

Identificación:	Gestión de los convenios	
Actores:	Usuario Coordinador, Usuario Convenio	
Propósito:	Ingreso de información de documentos y manipulación de los mismos.	
Visión General:	El usuario Coordinador debe recopilar la información de los convenios que se producen e ingresarlo en forma detallada los cuales también puede ser modificado o eliminado, el usuario Convenio verifica la información en caso de ser correcta crea el convenio.	
Tipo: Primario Esencial		
Curso Típico de Eventos:		
	Actores	Sistema
	1. El usuario Coordinador se autentica.	2. Presenta las opciones para ingresar la información de los convenios en forma detallada.
	3. Decide que acción realizar.	4. Se almacena en la base de datos.
	5. El usuario Convenio se autentica.	6. Presenta las opciones de verificar o ingresar la información de los convenios en forma detallada.

7. Si verifica la información que le llega del usuario convenio.	8. Presenta una lista de convenios.
9. Revisa la información si la acepta.	10. Actualiza la base de datos. Y presenta opciones de ingreso.
11. Se ingresa la información aceptada.	12. Se almacena en la base de datos.
Curso Alternativo:	
<p>3. Si los parámetros a ingresar o modificar no son correctos el sistema envía un mensaje de error.</p> <p>9. Si no acepta la información que le envía un mensaje de rechazo al usuario Coordinador.</p> <p>11. Se ingresa la información al formulario registro convenio.</p>	

Tabla LI Caso de uso de cancela convenio

Identificación:	Cancelación de Convenio
Actores:	Usuario Convenio
Propósito:	Ingreso de información cancelar convenios existentes en la unidad de Vinculación.
Visión General:	El usuario convenio ingresa información del motivo de cancelación convenios y en caso de mediación igual se registra.
Tipo: Primario Esencial	
Curso Típico de Eventos:	
Actores	Sistema
1. El usuario convenio se autentica.	2. Presenta las opciones de ingresar

	información.
3. Selecciona la opción cancela convenio.	4. Presenta opciones necesarias para cancelar convenio.
5. Ingresa información de la razón porque se cancela convenio.	6. Actualiza la base de datos. Y presenta opciones de ingreso.
7. Ingresa la información del convenio cancelado.	8. Se almacena en la base de datos.
Curso Alternativo:	
1. Si el usuario no está registrado no tendrá acceso	

Tabla LII Caso de uso de Consultas

Identificación:	Consultas	
Actores:	Usuario (Administrador o usuario convenio).	
Propósito:	Obtener información de los convenios.	
Visión General:	Los usuarios del sistema pueden acceder a obtener información de los convenios.	
Tipo: Primario Esencial		
Curso Típico de Eventos:		
	Actores	Sistema
	1. El usuario se autentica.	2. Presenta las opciones de la consulta.
	3. Selecciona lo requerido.	4. Presenta reporte de la consulta

	solicitada.
Curso Alternativo:	
1. Si el usuario no está registrado no tendrá acceso a las consultas.	

4.2.2.2. Diagrama de casos de uso

De manera general se presentan los casos de uso que intervienen en funcionamiento del sistema.

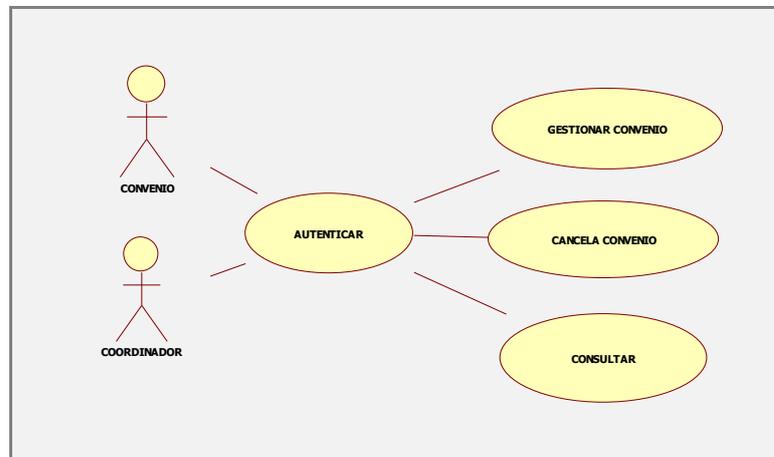


Figura IV.38 Diagrama de casos de uso general

4.2.3. Diagramas de secuencia

El diagrama de secuencias es el núcleo del modelo dinámico, y muestra todos los cursos alternos que pueden tomar todos nuestros casos de uso. Los diagramas de secuencias se componen de 4 elementos que son: el curso de acción, los objetos, los mensajes y los métodos (operaciones).

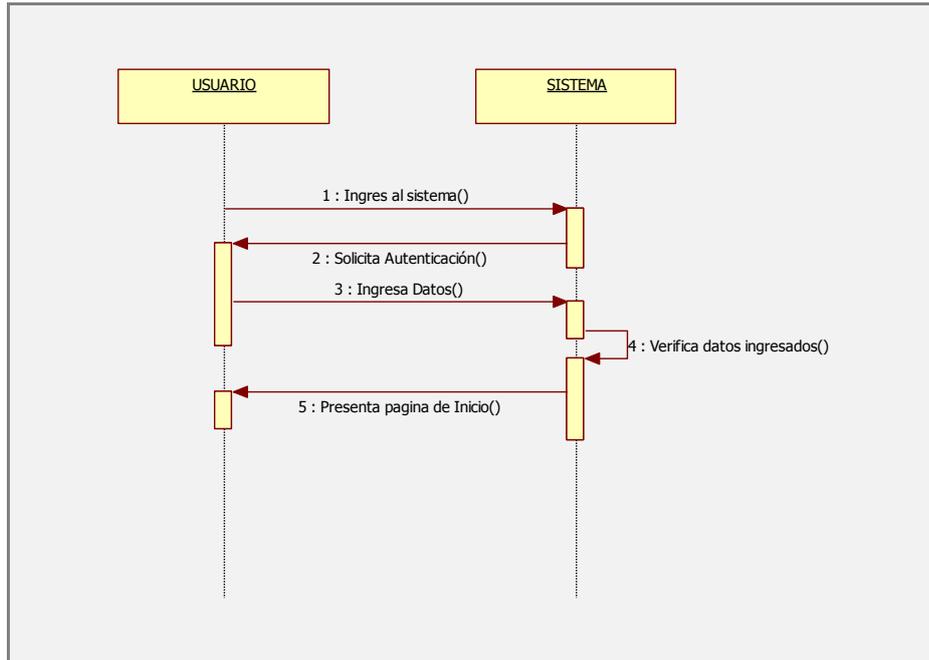


Figura IV.39 Diagrama de secuencia de autenticación

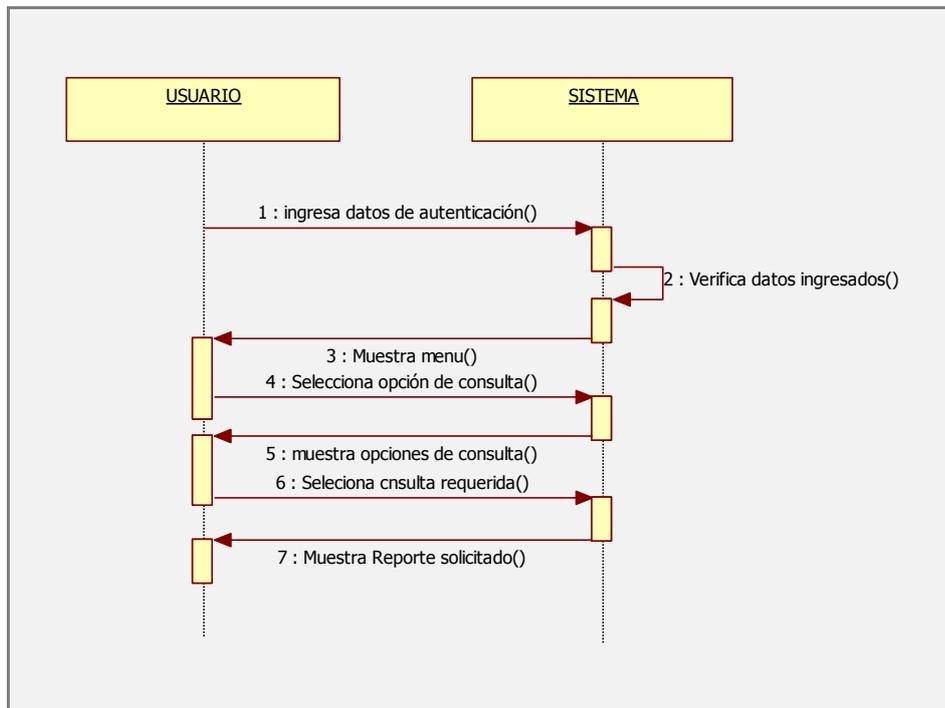


Figura IV.40 Diagrama de secuencia de consultas

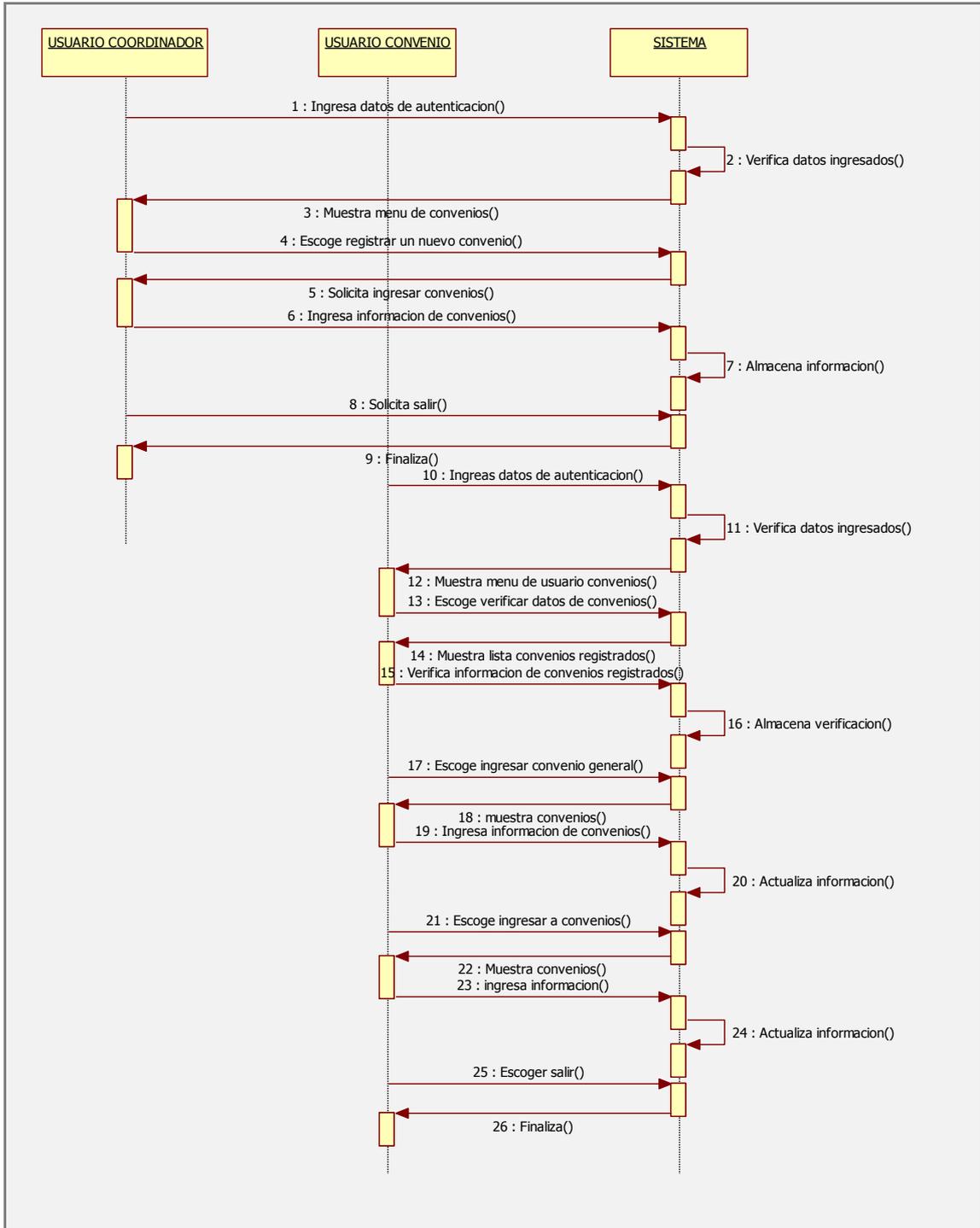


Figura IV.41 Diagrama de secuencia de gestión de convenios

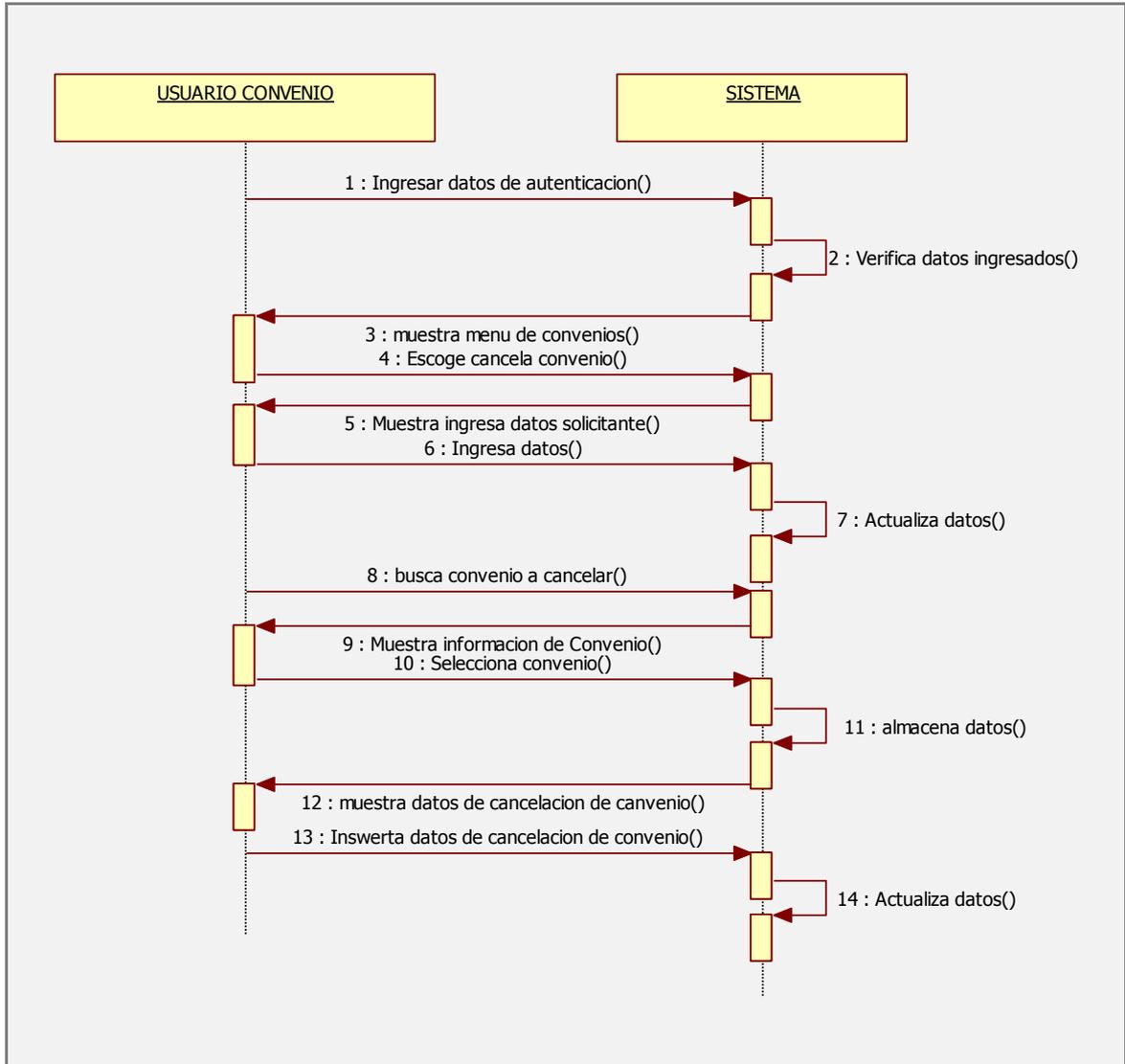


Figura IV.42 Diagrama de secuencia de cancelación de convenios

4.3. Diseño

4.3.1. Casos de uso reales

Tabla LIII Caso de uso de Autenticación

Actores	Sistema
1. El usuario ingresa al sistema.	2. Solicita autenticación.
3. Ingresa su nombre y contraseña.	4. Verifica los datos ingresados.

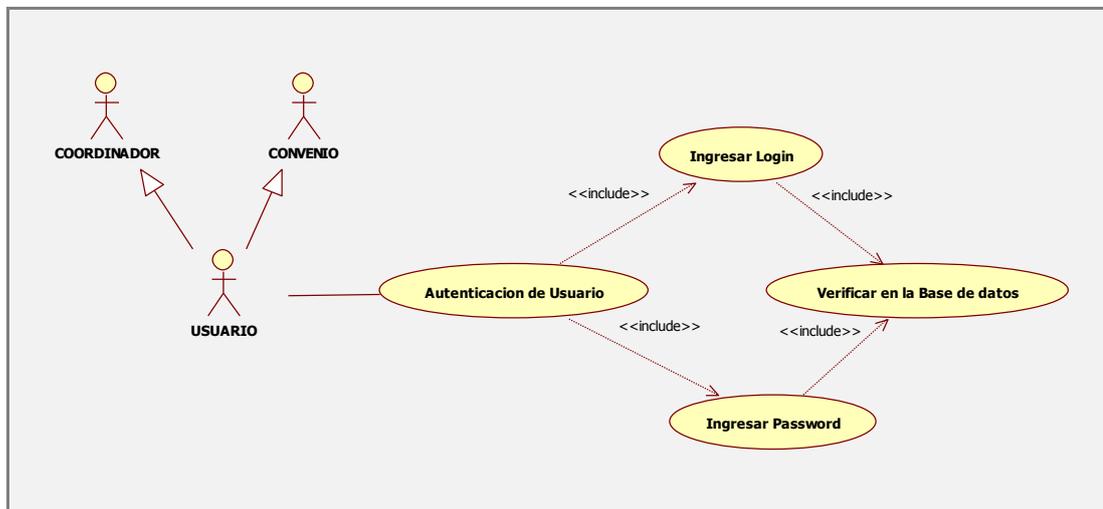


Figura IV.43 Diagrama de casos de uso de autenticación

Tabla LIV Caso de uso de Gestión de información de los Convenios

Actores	Sistema
1. El usuario Coordinador se autentica.	2. Presenta las opciones para ingresar la información de los convenios en forma detallada.
3. Decide que acción realizar.	4. Se almacena en la base de datos.
5. El usuario convenio se autentica.	6. Presenta las opciones de verificar o

	ingresar la información de los convenios en forma detallada.
7. Si verifica la información que le llega del usuario convenio.	8. Presenta una lista de convenios.
9. Revisa la información si la acepta.	10. Actualiza la base de datos. Y presenta opciones de ingreso.
11. Se ingresa la información aceptada.	12. Se almacena en la base de datos.

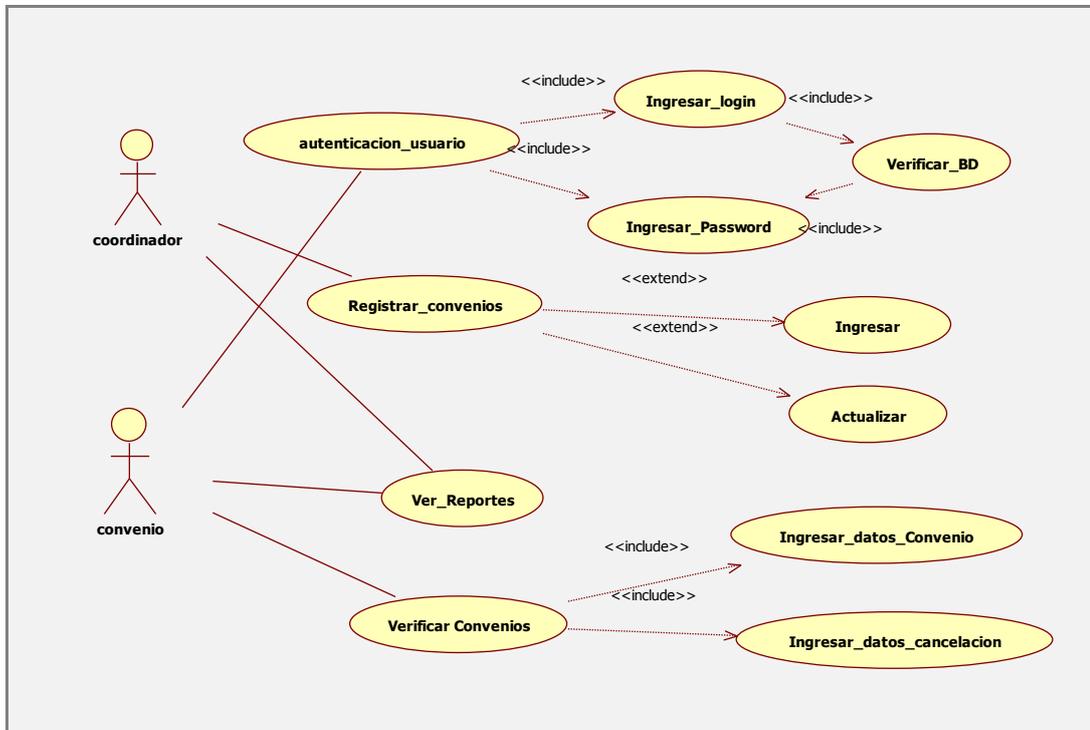


Figura IV.44 Diagrama de casos de uso de gestión de convenios

Tabla LV Caso de uso de Cancelación de Convenios

Actores	Sistema
1. El usuario convenio se autentica.	2. Presenta las opciones de ingreso de convenio.

3. Selecciona la opción cancelación de convenio.	4. Presenta opciones necesarias para cancelación de convenios
5. Ingresar información del solicitante.	6. Actualiza la base de datos. Y presenta opciones de ingreso.
7. Ingresar la información del convenio a crear y en caso de cancelación.	8. Se almacena en la base de datos.

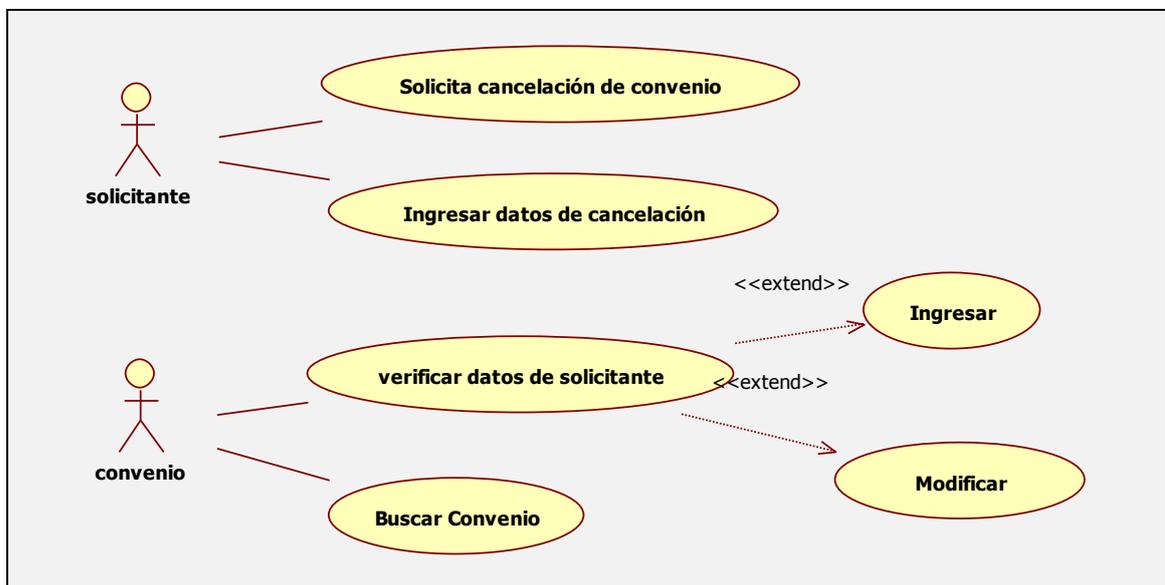


Figura IV.45 Diagrama de casos de uso de cancelación de convenios

Tabla LVI Caso de uso de Consultas

Actores	Sistema
1. El usuario se autentica.	2. Presenta las opciones de la consulta.
3. Selecciona lo requerido.	4. Presenta reporte de la consulta solicitada.

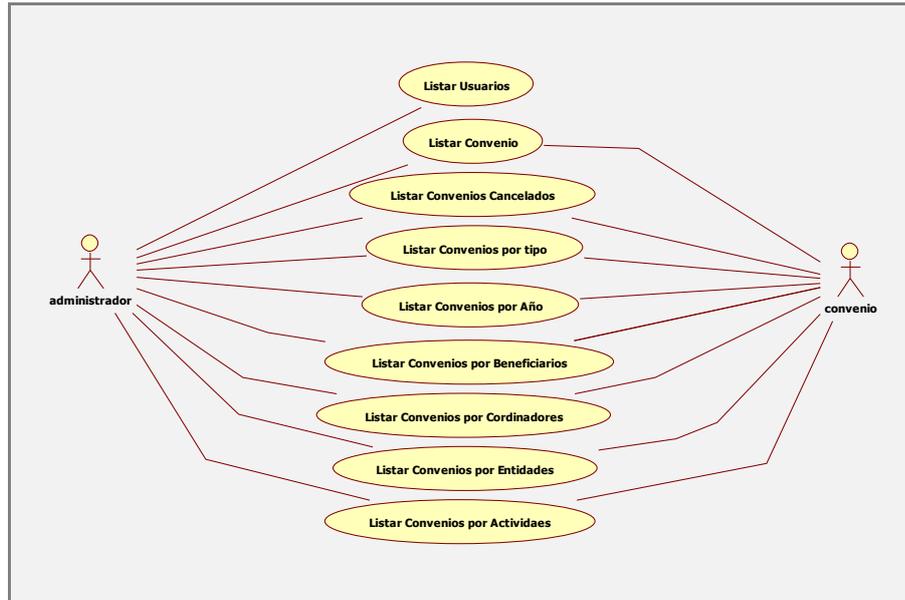


Figura IV.46 Diagrama de casos de uso de consultas

4.3.2. Definición de informes e interfaces de usuario

4.3.2.1. Definición de la información de la interfaz de usuario

Para la interfaz de usuario se ha partido de los casos de usos reales, que ejercen como modelos para definir la interfaz del usuario tanto para las pantallas como para los documentos.

4.3.2.2. Lenguaje de Comunicación

- **Comunicación con el Usuario:** Es de fácil entendimiento para el usuario debido a que se maneja como una aplicación web.

4.3.3. Diagramas de interacción

4.3.3.1. Diagramas de secuencia

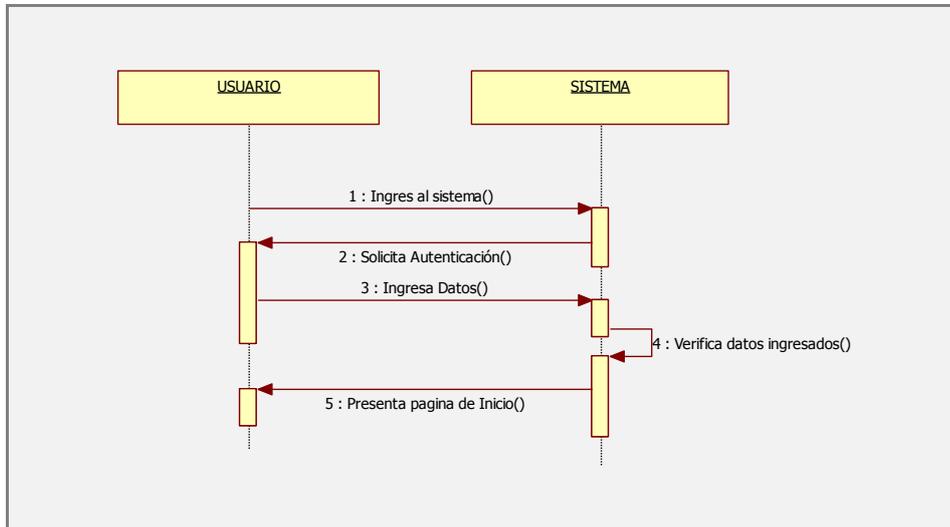


Figura IV.47 Diagrama de secuencia de autenticación

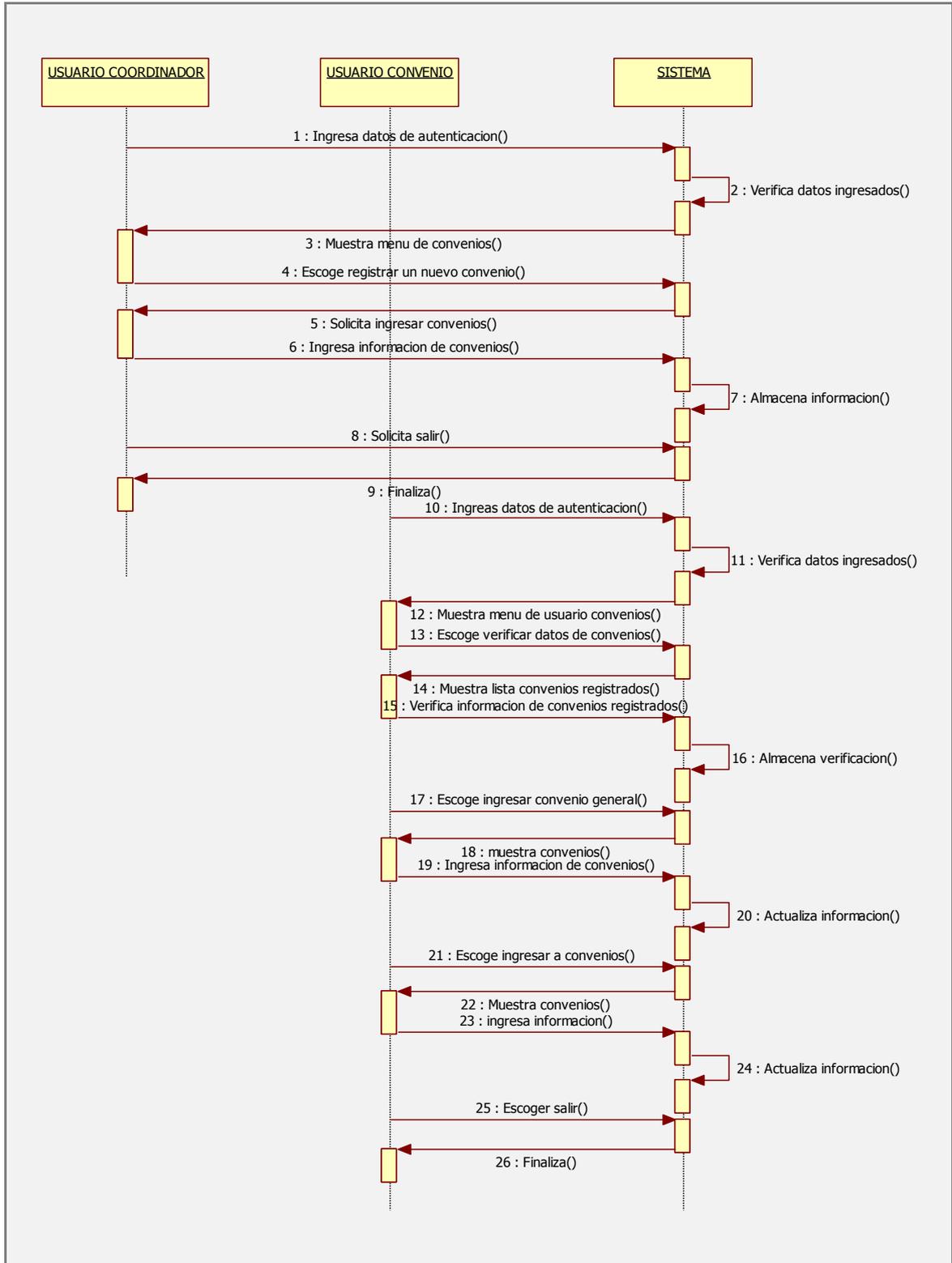


Figura IV.48 Diagrama de secuencia de gestión de convenios

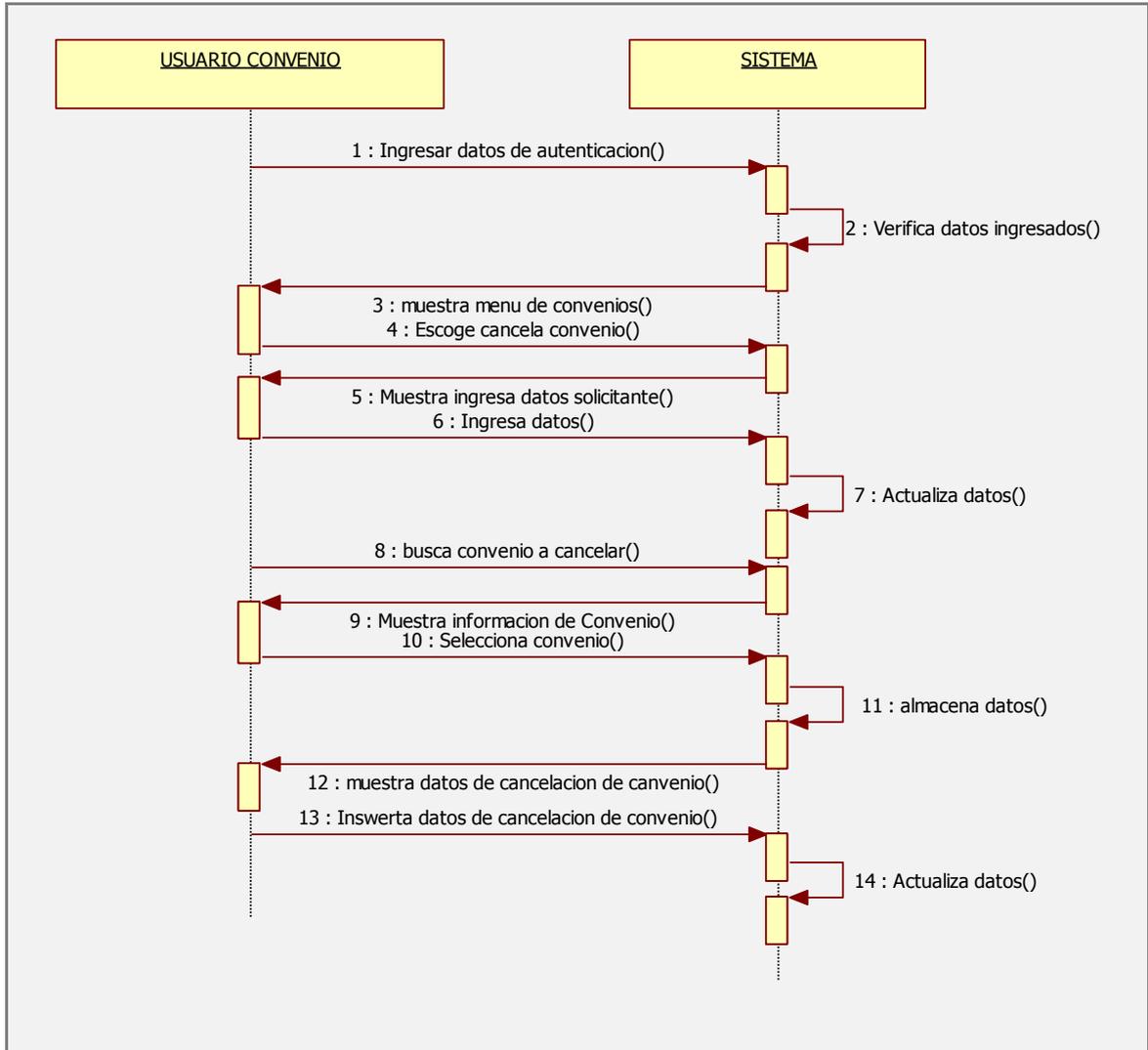


Figura IV.49 Diagrama de secuencia de cancelación de convenio

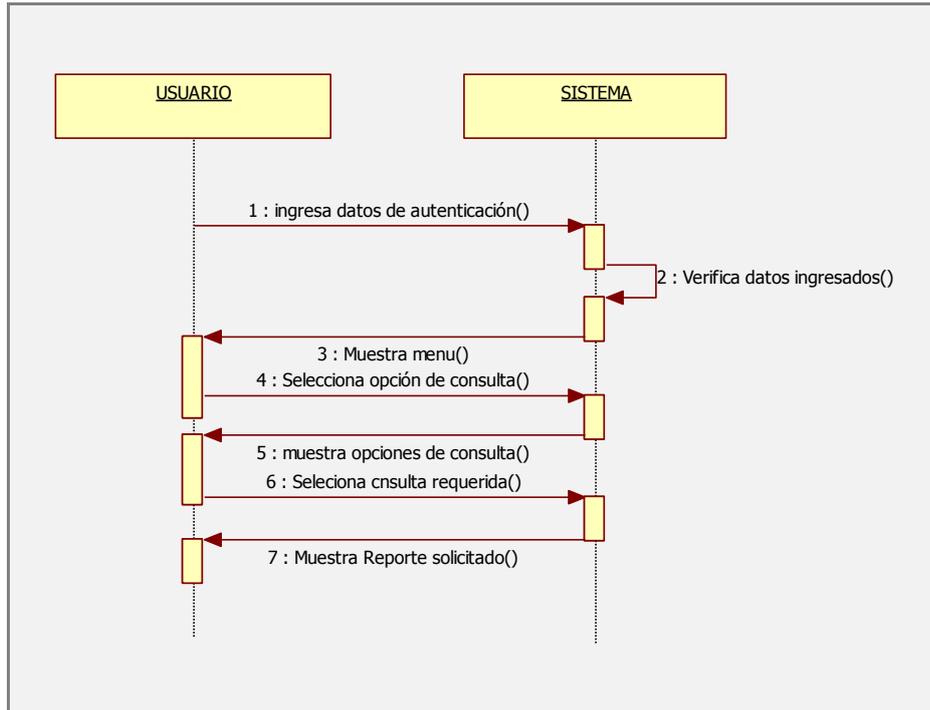


Figura IV.50 Diagrama de secuencia de consultas

4.3.3.2. Diagrama de calles

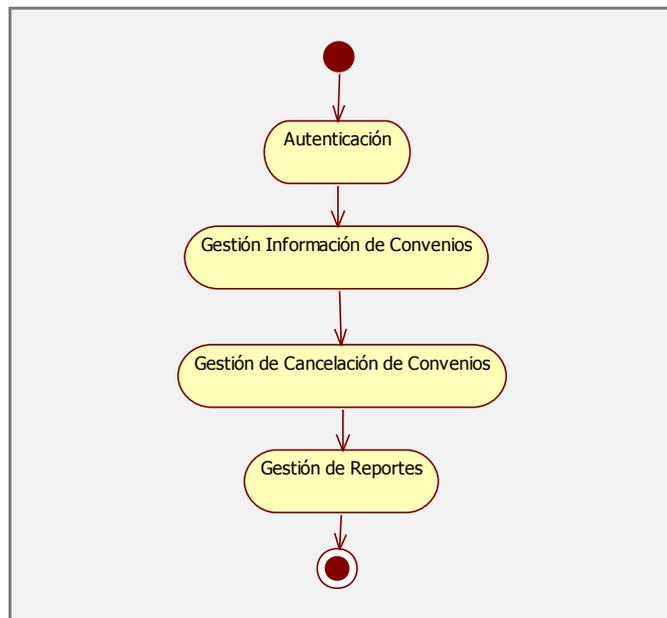


Figura IV.51 Diagrama de calles

4.3.4. Diagrama de clases

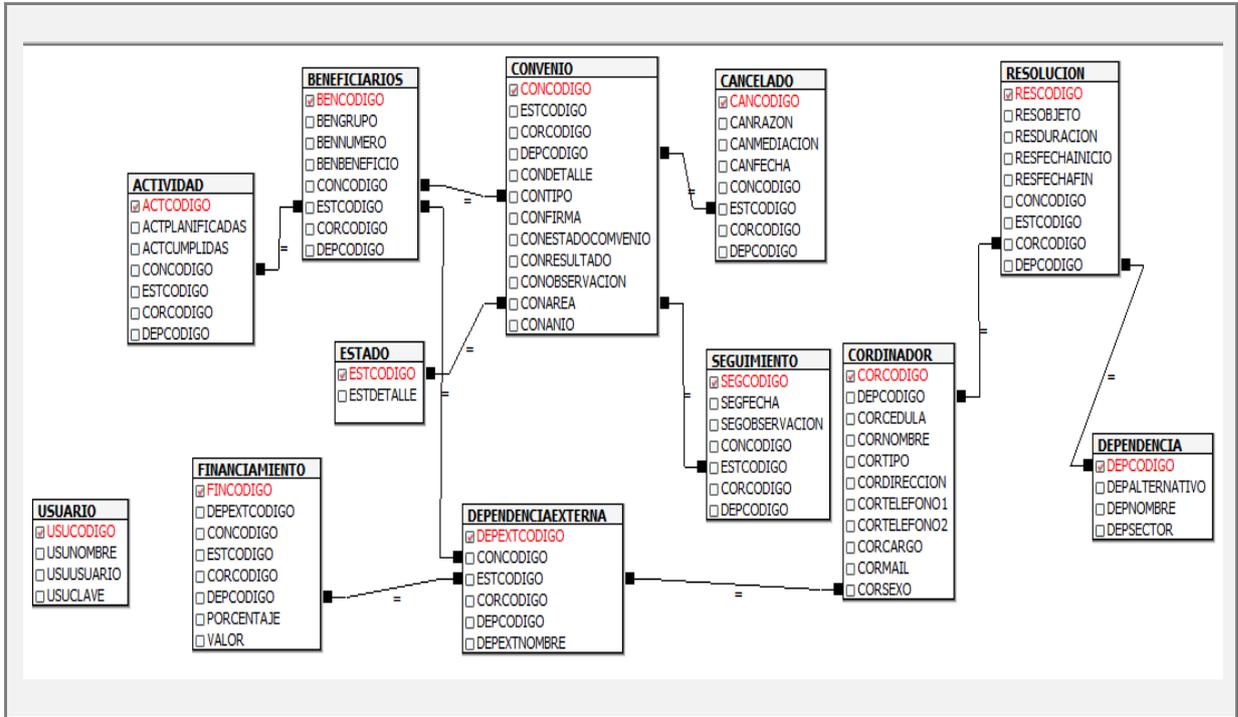


Figura IV.52 Diagrama de clases

4.3.5. Diagrama de Despliegue

4.3.5.1. Diagrama de componentes

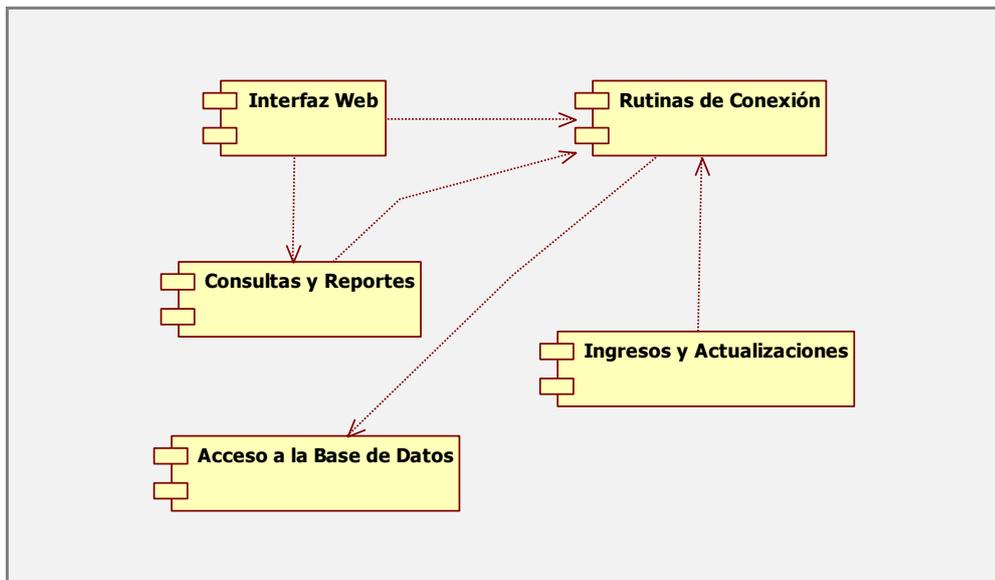


Figura IV.53 Diagrama de componentes

4.3.5.2. Diagrama de nodos

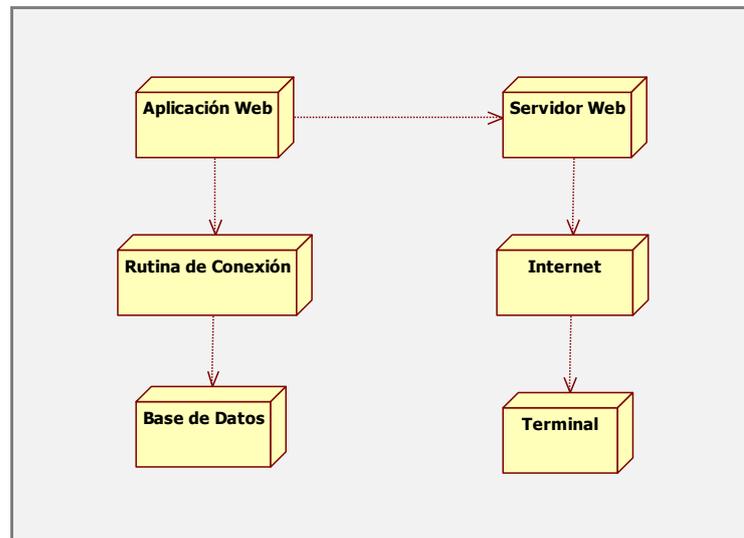


Figura IV.54 Diagrama de nodos

4.4. Implementación y Pruebas

4.4.1. Definición de estándares de programación

Para la realización de la codificación del sistema se ha definido los siguientes estándares:

- El nombre de las tablas comienzan con letra minúscula, seguido de () en caso de que tenga más de una palabra.
- El nombre de los campos empieza con letra minúscula, seguido de una letra mayúscula en caso de que tenga más de una palabra.
- El nombre de los atributos de la clase empieza con letra minúscula, seguido de una letra mayúscula en caso de que tenga más de una palabra.
- Los métodos que interactúan con la base de datos reciben como parámetro de entrada:
 - ✓ Connection: Representa la cadena de conexión hacia la base de datos.
 - ✓ Lista de parámetros en el caso de ser necesaria.

- Los métodos que interactúan con la base de datos en el caso de funciones de inserción, deben devolver como resultado el código del campo que ha sido insertado.

4.4.2. Pruebas Unitarias

Para asegurar el correcto funcionamiento del sistema se han probado sus clases y sus métodos de forma independiente, enviando datos de entrada desde el código para luego obtenerlos a través de los diferentes métodos para realizar consultas.

4.4.3. Pruebas de modulo y del sistema

Las pruebas finales consistieron en verificar que la información ingresada se vea inmediatamente reflejada en las consultas del sistema, esto sirve para comprobar que la información se está registrando correctamente en la base de datos.

Se provocaron errores intencionales para verificar el correcto funcionamiento del sistema, así como de las funciones de validación de datos, como por ejemplo:

- Realizar consultas a tablas sin valores
- Ingresar campos vacíos
- Buscar información que no existe

CONCLUSIONES

1. Mediante un adecuado estudio de las tres Herramientas Gráficas de Administración para PostgreSQL seleccionadas, se puede tener una mejor visión en el marco de la administración.
2. El estudio permitió determinar que la Herramienta Gráfica de Administración para PostgreSQL más adecuada para la administración, desarrollo y creación de base de datos es PgAdminIII.
3. En el Análisis de Instalación, PgAdminIII posee un 93,75/100% al igual que EMS SQL Lite for PostgreSQL de tiempo en el proceso de instalación del cliente de postgresql, ya que el tiempo es primordial cuando se trata de realizar tareas administrativas, y superando con un 18,75% a la herramienta PhpPgAdmin. Demostrando que PgAdminIII como una de sus mejores características es el proceso de configuración y el nivel de ayuda que presta para el acceso a las base de datos.
4. El indicador Seguridad es el punto promedio para determinar el nivel de partida y funcionamiento óptimo de la aplicación, PgAdminIII supera en gran medida a PhpPgAdmin y EMS SQL Lite for PostgreSQL al utilizar la máxima seguridad posible a la base de datos con un 100/100%, superando en un 12,5% a EMS SQL Lite for PostgreSQL y en un 43,75% a PhpPgAdmin, se ha demostrado nuevamente que PgAdminIII es una de las mejores herramientas de administración al momento de proteger y asegurar información.
5. El manejo y Administración de bases de datos en una aplicación es fundamental ya que se maneje miles de transacciones en tiempos cortos, PgAdminIII y EMS SQL Lite for PostgreSQL superan en un 37,5% a PhpPgAdmin en el manejo de transacciones en el menor tiempo posible, ya que poseen características similares y mejores maneras de administrar base de datos.

6. En la Calidad el uso de herramientas para postgresql requiere de utilización de recursos de hardware y compatibilidad de software al momento de utilizar las herramientas de administración, demostrando que PgAdminIII posee una de las mejores características para el manejo de bases de datos con un 75/100%, al igual que PhpPgAdmin con 75%, superando a EMS SQL Lite for PostgreSQL con un 8,34%, en el manejo y comprensión de las herramientas.
7. La utilización de Herramientas de administración de base de datos para el desarrollo de aplicaciones nos facilita a la estandarización y la programación efectiva del desarrollo.
8. La investigación realizada sobre el análisis comparativo de herramientas para postgresql facilitará a los desarrolladores a la optimización de recursos en la implementación de sus aplicaciones.
9. El manejo de recursos de hardware y software es complejo por lo tanto al momento de elegir la herramienta de administración para postgresql adecuada para el desarrollo de la aplicación web se debe realizar un previo análisis obligatorio de diferentes herramientas para elegir cual se adapta a las necesidades de la aplicación para obtener el mejor sistema web.

RECOMENDACIONES

1. Plantear correctamente con el usuario final todos los requisitos que desea para no perder tiempo en el desarrollo de la aplicación.
2. Se recomienda el uso de PgAdminIII como herramienta de administración de base de datos con postgresql.
3. Las Herramientas Gráficas de Administración para Postgresql son las más adecuadas para la administración, desarrollo y creación de base de datos.
4. Investigar características adicionales y funciones de la herramienta para postgresql, para el desarrollo de sistemas, ya que aquello ayudará al desarrollo rápido de aplicaciones en tiempos cortos.
5. Mediante PgAdminIII para postgresql se puede optimizar el diseño de bases de datos físicas del cliente del motor de base de datos.

RESUMEN

Se realizó el “ANÁLISIS COMPARATIVO DE HERRAMIENTAS PARA POSTGRESQL, CASO PRÁCTICO: SISTEMA WEB DE GESTIÓN DE CONVENIOS ESPOCH” que ha permitido seleccionar la herramienta de administración de base de datos postgresql, que ofrece acelerar el proceso de desarrollo de base de datos en las aplicaciones web, teniendo en cuenta la seguridad y promover buenas prácticas de desarrollo como el uso de patrón Modelo-Vista-Controlador.

El análisis comparativo se basó en el Método Científico, con técnicas de observación directa, de laboratorio y experimento, para alcanzar los objetivos se utilizó las siguientes herramientas: Servidor Web Apache 2.2.17, Lap Master, PostgreSQL9.0, Php 5.3.5 (VC6 X86 32bit) lenguaje de programación Java y Netbeans7.0 como IDE de desarrollo.

Las Herramientas de administración de postgresql motivo del análisis investigativo fueron integrados al IDE Netbeans7, para el desarrollo de sistemas de pruebas que permitieron determinar cuál posee la mejor administración, el resultado cuantitativo obtenido mediante la comparación de Indicadores: Instalación, Seguridad, Administración, Calidad, para evaluar el rendimiento en el desarrollo fue: 97,05%(Exelente) para PgAdminIII; 60.29%(bueno) para PhpPgAdmin; 91,17%(Muy Bueno) para EMS SQL Lite forPostgreSQL.

Los resultados denotados anteriormente permiten concluir que la herramienta de administración de datos postgresql ideal para el desarrollo de aplicaciones web óptimas es PgAdminIII, a fin de desarrollar portal web de gestión de convenios para el departamento de Vinculación de la ESPOCH.

En conclusión la herramienta de administración de datos para postgresql que ofrece la mejor administración para el desarrollo de aplicaciones web óptimas con un puntaje de 97,05%/100 es PgAdminIII que supera a sus competidores.

Se recomienda a PgAdminIII como herramienta de administración de base de datos en postgresql por responder a las necesidades de la mayoría de los usuarios, desde escribir simples consultas SQL hasta desarrollar bases de datos complejas, la interfaz gráfica soporta todas las características de PostgreSQL y hace simple la administración, donde está disponible en más de una docena de lenguajes y para varios sistemas operativos, incluyendo Microsoft Windows, Linux, FreeBSD, Mac OSX y Solaris.

ABSTRACT

It was made the “COMPARATIVE ANALYSIS OF TOOLS FOR POSTGRESQL, PRACTICAL CASE: MANAGEMENT WEB SYSTEMS OF ESPOCH CONVENTIONS” which has permitted to select the data base management tool PostgreSQL, that offer accelerating the process of data base development in web applications, having into account the security and promoting good practice of developments like the Vista-Controlador Pattern.

The comparative analysis was based on the Scientific Method, which observation techniques: direct, from lab and experimental, to reach the objectives were used the following tools: Web Apache Server 2.2.17, Lap Master, PostgreSQL9.0, Php 5.3.5 (VC6 X86 32 bit), programming language Java and Netbeans7.0 as development IDE.

The Management Tools of PostgreSQL objective of the investigative analysis were integrated to IDE Netbeans 7, for the system development of proofs which allowed to determine what possess the best management, the quantitative outcome obtained through the comparative indicators: Installing, Security, Management, and Quality to evaluate the performance in the development were: 97,05% (Excellent) for PgAdminIII; 60,29%(Well) for PhpPgAdmin; 91,17% (Very Well) for EMS SQL Lite for PostgreSQL.

The results previously denoted to make concluding that the management data tool PostgreSQL ideal for the optimal development of web applications is PgAdminIII, in order to develop Web Management Portals for conventions in the Entailment Department of the ESPOCH.

In conclusion the database management tool for PostgreSQL that offers the best management for optimal web application development with a score of 97,05% is PgAdminIII which comes over its competitors.

It is recommended PgAdminIII as a database management tool in PostgreSQL to answers towards the needs of the most of the users, since writing simple SQL, queries up to develop complex databases, the graphical interphase hold all the characteristics of PostgreSQL and make a simple administration, where is available in more than a dozen of languages and for several Operative Systems included Microsoft Windows, Linux, FreeBSD, Mac OSX and Solaris.

BIBLIOGRAFÍA

1. ADMINISTRADOR LITE

<http://www.sqlmanager.net/products/postgresql/manager>

2012/07/10

2. ADMINISTRACIÓN EMS LITE POSTGRESQL

<http://www.softpedia.com/get/Internet/Servers/Database-Utils/EMS-PostgreSQL-Manager-Lite.shtml>

2012/06/05

3. ADMINISTRACIÓN DE BASE DE DATOS

<http://www.questsoftware.es/database-management/>

2011/05/10

4. ARQUITECTURA SISTEMAS DE BASES DE DATOS

<http://www.emagister.com/curso-sistemas-bases-datos/sghd-arquitectura-sistemas-bases-datos-primera-parte>

2011/01/03

5. ARQUITECTURA POSTGRESQL

<http://www.dataprix.com/72-arquitectura-postgresql>

2011/04/10

6. BASE DE DATOS

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

2010/10/06

7. BASE DE DATOS ORIENTADA A OBJETOS

http://es.wikipedia.org/wiki/Base_de_datos_orientada_a_objetos

2011/06/02

8. COMPONENTES DE BASE DE DATOS

<https://sites.google.com/site/is11801/contenido/componentes>

2011/06/02

9. CARACTERÍSTICAS POSTGRESQL

<http://www.postgresql.org/about/press/presskit90/es/>

2012/03/06

10. COMO ES POSTGRESQL

[http://ibiblio.org/pub/linux/docs/LuCaS/Postgresqles/web/navegable/Ho
wto/PostgreSQL-COMO-11.html](http://ibiblio.org/pub/linux/docs/LuCaS/Postgresqles/web/navegable/Ho
wto/PostgreSQL-COMO-11.html)

2012/01/13

11. CARACTERÍSTICAS DE PGADMINIII

<http://www.pgadmin.org>

2012/05/30

12. CARACTERÍSTICAS PHPPGADMIN

<http://es.wikipedia.org/wiki/PhpPgAdmin>

2012/07/31

**13. FUNCIONES DE LOS SISTEMAS GESTORES DE BASES DE
DATOS**

<http://cnx.org/content/m17543/latest/>

2011/04/10

14. HERRAMIENTAS GUI POSTGRESQL

[http://wiki.postgresql.org/wiki/Guía de la Comunidad para las herramientas GUI de PostgreSQL](http://wiki.postgresql.org/wiki/Guía_de_la_Comunidad_para_las_herramientas_GUI_de_PostgreSQL)

2012/01/04

15. POSTGRESQL

<http://es.wikipedia.org/wiki/PostgreSQL>

2011/06/04

16. PGADMINIII

<http://www.josegutierrez.com.co/index.php/manual/86/articulos/manual/capitulo-6/115-pgadmin-3>

2012/06/11

17. QUE ES PHPPGADMIN

<http://phppgadmin.sourceforge.net/doku.php>

2012/06/05

18. SISTEMA DE GESTIÓN DE BASES DE DATOS

[http://es.wikipedia.org/wiki/Sistema de gestión de bases dedatos](http://es.wikipedia.org/wiki/Sistema_de_gestión_de_bases_dedatos)

2011/02/11

19. SISTEMAS DE BASES DE DATOS

<http://usuarios.multimania.es/cursosgbd/UD2.htm>

2011/05/06

20. SISTEMA GESTIÓN DE BASE DE DATOS

<http://www.slideshare.net/hugofern/sistema-gestin-de-bases-de-datos>

2011/01/04

21. SISTEMA DE GESTIÓN DE BASES DE DATOS

[http://es.wikipedia.org/wiki/Sistema de gesti3n de basesdedatos](http://es.wikipedia.org/wiki/Sistema_de_gesti3n_de_basesdedatos)

2011/01/06

22. SOBRE POSTGRESQL

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

2011/07/02

ANEXOS

ANEXO 1

Tablas del Motor de Base de datos postgresql en la herramienta de administración PgadminIII

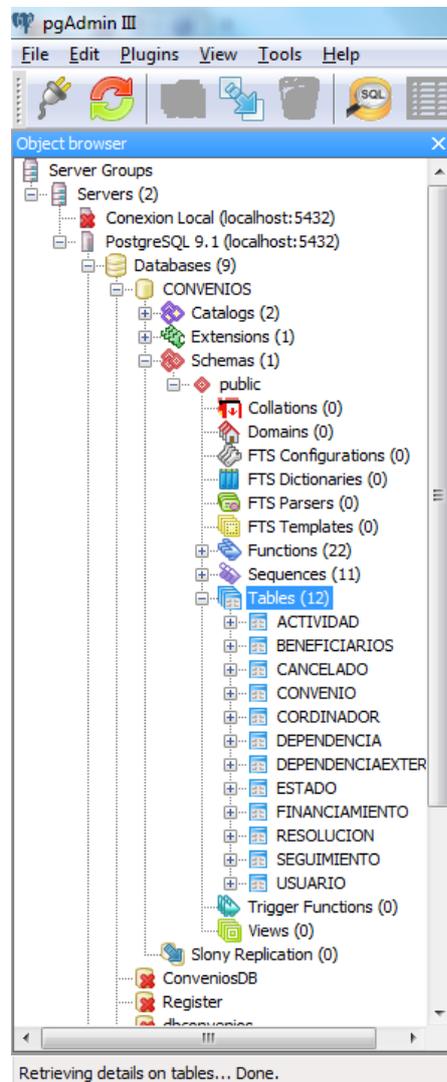


Figura 1: Mapa del administrador PgAdminIII

ESCRIPS PARA CREAR LA BASE DE DATOS CONVENIOS

- Database: "CONVENIOS"

-- DROP DATABASE "CONVENIOS";

CREATE DATABASE "CONVENIOS"

WITH OWNER = postgres

ENCODING = 'UTF8'

TABLESPACE = pg_default

```
LC_COLLATE = 'Spanish_Spain.1252'  
LC_CTYPE = 'Spanish_Spain.1252'  
CONNECTION LIMIT = -1;
```

TABLAS

```
-- Table: "ACTIVIDAD"
```

```
-- DROP TABLE "ACTIVIDAD";
```

```
CREATE TABLE "ACTIVIDAD"
```

```
(  
  "ACTCODIGO" bigserial NOT NULL,  
  "ACTPLANIFICADAS" character varying(200),  
  "ACTCUMPLIDAS" character varying(200),  
  "CONCODIGO" character varying(20) NOT NULL,  
  "ESTCODIGO" bigint NOT NULL,  
  "CORCODIGO" bigint NOT NULL,  
  "DEPCODIGO" bigint NOT NULL,  
  CONSTRAINT "ACTIVIDAD_pkey" PRIMARY KEY ("ACTCODIGO" ,  
  "CONCODIGO" , "ESTCODIGO" , "CORCODIGO" , "DEPCODIGO" ),  
  CONSTRAINT "ACTIVIDAD_CONCODIGO_fkey" FOREIGN KEY  
  ("CONCODIGO", "ESTCODIGO", "CORCODIGO", "DEPCODIGO")  
  REFERENCES "CONVENIO" ("CONCODIGO", "ESTCODIGO",  
  "CORCODIGO", "DEPCODIGO") MATCH SIMPLE  
  ON UPDATE RESTRICT ON DELETE RESTRICT  
)  
WITH (  
  OIDS=FALSE  
);  
ALTER TABLE "ACTIVIDAD"  
  OWNER TO postgres;
```

```
-- Index: "IX_6"
```

```
-- DROP INDEX "IX_6";
```

```
CREATE INDEX "IX_6"
```

```
ON "ACTIVIDAD"
```

```
USING btree
```

```
("CONCODIGO" COLLATE pg_catalog."default" , "ESTCODIGO" ,  
"CORCODIGO" , "DEPCODIGO" );
```

```
-- Table: "BENEFICIARIOS"
```

```
-- DROP TABLE "BENEFICIARIOS";
```

```

CREATE TABLE "BENEFICIARIOS"
(
  "BENCODIGO" bigserial NOT NULL,
  "BENGRUPO" character varying(1000),
  "BENNUMERO" character varying(1000),
  "BENBENEFICIO" character varying(1000),
  "CONCODIGO" character varying(20) NOT NULL,
  "ESTCODIGO" bigint NOT NULL,
  "CORCODIGO" bigint NOT NULL,
  "DEPCODIGO" bigint NOT NULL,
  CONSTRAINT "BENEFICIARIOS_pkey" PRIMARY KEY ("BENCODIGO" ,
"CONCODIGO" , "ESTCODIGO" , "CORCODIGO" , "DEPCODIGO" ),
  CONSTRAINT "BENEFICIARIOS_CONCODIGO_fkey" FOREIGN KEY
("CONCODIGO", "ESTCODIGO", "CORCODIGO", "DEPCODIGO")
  REFERENCES "CONVENIO" ("CONCODIGO", "ESTCODIGO",
"CORCODIGO", "DEPCODIGO") MATCH SIMPLE
ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITH (
  OIDS=FALSE
);
ALTER TABLE "BENEFICIARIOS"
OWNER TO postgres;

-- Index: "IX_5"

-- DROP INDEX "IX_5";
REATE INDEX "IX_5"
ON "BENEFICIARIOS"

```

```

USING btree

("CONCODIGO" COLLATE pg_catalog."default" , "ESTCODIGO" , "CORCODIGO"
, "DEPCODIGO" );

-- Table: "BENEFICIARIOS"

-- DROP TABLE "BENEFICIARIOS";

CREATE TABLE "BENEFICIARIOS"
(
    "BENCODIGO" bigserial NOT NULL,
    "BENGRUPO" character varying(1000),
    "BENNUMERO" character varying(1000),
    "BENBENEFICIO" character varying(1000),
    "CONCODIGO" character varying(20) NOT NULL,
    "ESTCODIGO" bigint NOT NULL,
    "CORCODIGO" bigint NOT NULL,
    "DEPCODIGO" bigint NOT NULL,
    CONSTRAINT "BENEFICIARIOS_pkey" PRIMARY KEY ("BENCODIGO" ,
"CONCODIGO" , "ESTCODIGO" , "CORCODIGO" , "DEPCODIGO" ),
    CONSTRAINT "BENEFICIARIOS_CONCODIGO_fkey" FOREIGN KEY
("CONCODIGO", "ESTCODIGO", "CORCODIGO", "DEPCODIGO")
    REFERENCES "CONVENIO" ("CONCODIGO", "ESTCODIGO",
"CORCODIGO", "DEPCODIGO") MATCH SIMPLE
    ON UPDATE RESTRICT ON DELETE RESTRICT
)
WITH (
    OIDS=FALSE
);
ALTER TABLE "BENEFICIARIOS"
    OWNER TO postgres;

```

```
-- Index: "IX_5"
```

```
-- DROP INDEX "IX_5";
```

```
CREATE INDEX "IX_5"
```

```
ON "BENEFICIARIOS"
```

```
USING btree
```

```
("CONCODIGO" COLLATE pg_catalog."default" , "ESTCODIGO" , "CORCODIGO"  
 , "DEPCODIGO" );
```

Todas la líneas de códigos que fueron generadas para crear base de datos con cada Herramienta en las diferentes Herramientas Graficas de base de datos para Postgresql la cuales son:PgAdminIII, PhpPgAdmin, EMS SQL Lite forPostgreSQL.

1. PgAdminIII

Creando visualmente la base de datos

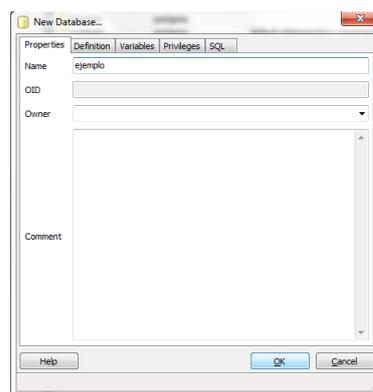


Figura 2: Creando base de datos pgadminIII

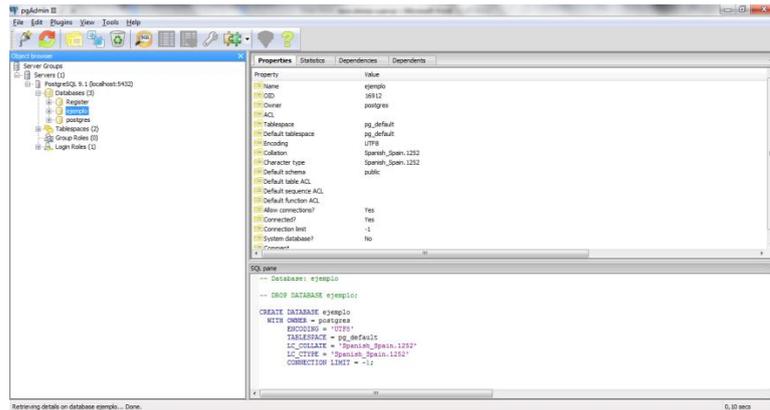


Figura 3: Muestra la base de datos creada

Código generado al crear la base de datos ejemplo

-- Database: ejemplo

-- DROP DATABASE ejemplo;

CREATE DATABASE ejemplo

WITH OWNER = postgres

ENCODING = 'UTF8'

TABLESPACE = pg_default

LC_COLLATE = 'Spanish_Spain.1252'

LC_CTYPE = 'Spanish_Spain.1252'

CONNECTION LIMIT = -1;

2. PHPPgAdmin

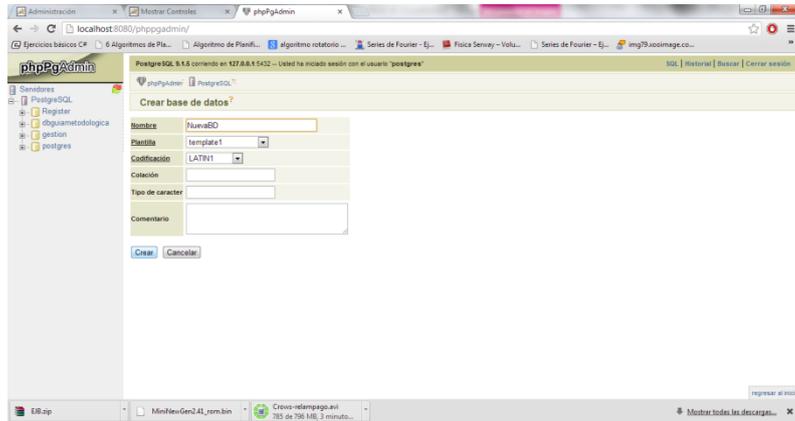


Figura 4: Creación de base de datos con phppgadmin

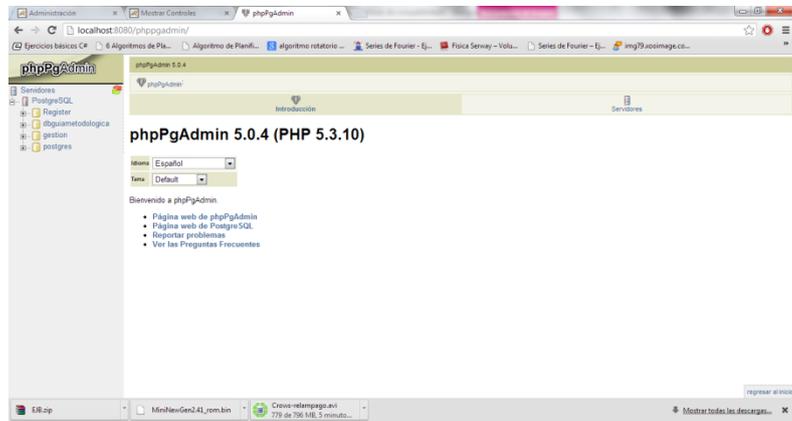


Figura 5: Base de datos creada con phppgadmin

3. EMS SQL Manager for PostgreSQL.



Figura 6: creando base de datos

Se muestra la lista de bases de datos creadas anteriormente

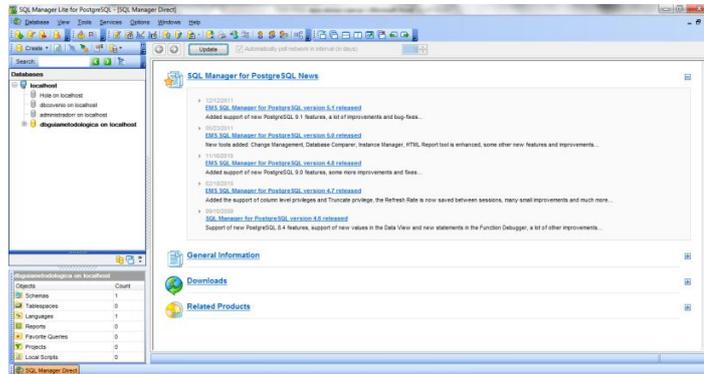


Figura 7: Lista de bases de datos en EMS SQL Manager Lite for PostgreSQL

ANEXO 2

Se presentaran las pantallas de los indicadores sacados de cada uno de los escenarios realizados a cada uno de las Herramientas para PostgreSQLPgAdminIII, PHPpgAdmin, EMS SQL Managerfor PostgreSQL, con sus respectivas pruebas.

A) PgAdminIII

Indicador 1. Tiempo en Completar Instalación

Con la aplicación Lap Master Beta, se ha procedido a cargar la aplicación para su posterior instalación respectiva



Figura 8: Tiempo en marcha de instalación PgAdminIII

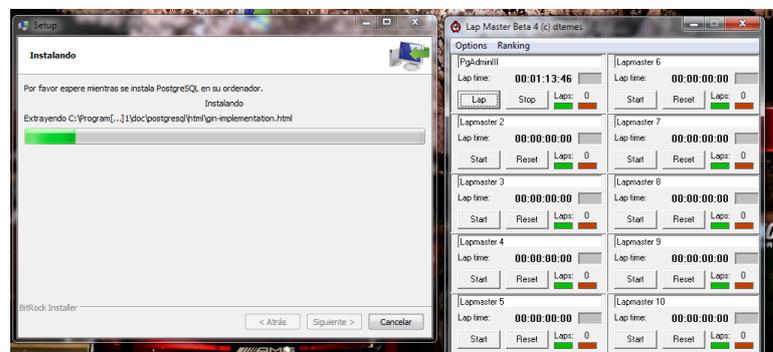


Figura 9: Proceso de Instalación PgAdminIII

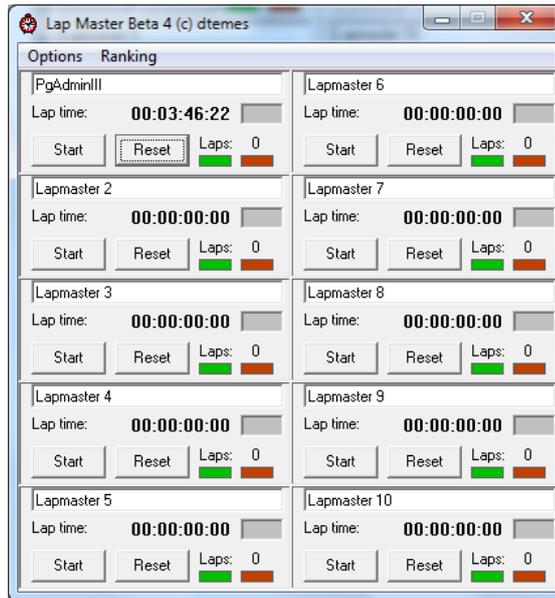


Figura 10: Tiempo en instalar PgAdminIII

B) PhpPgAdmin

Proceso de instalación de PhpPgAdmin

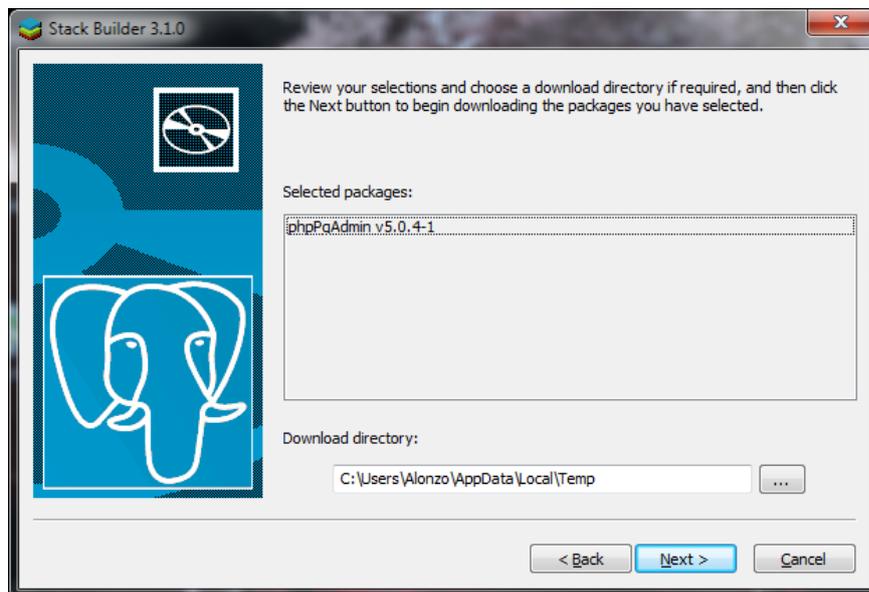


Figura 11: Empezando el proceso de instalación

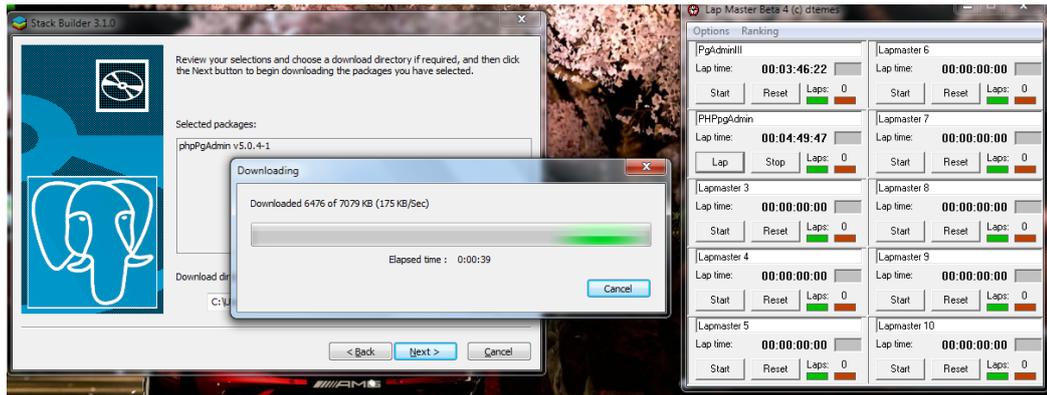


Figura 12: cargando e instalando

Finalizando el proceso de instalación de PhpPgAdmin

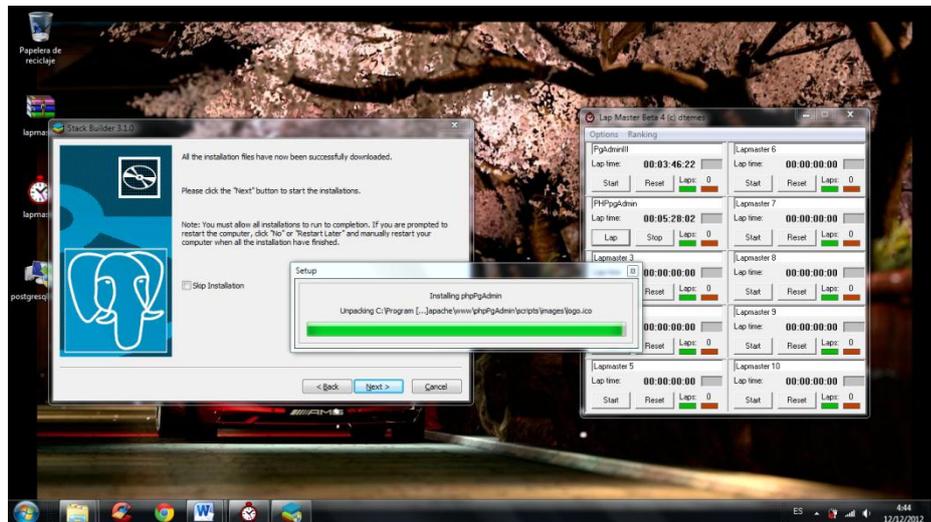


Figura 13: Finalizando instalación

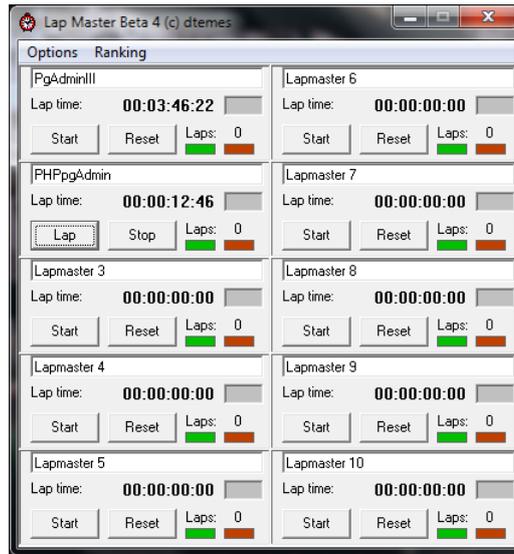


Figura 14: Empezando Tiempo Instalación de PhpPgAdmin

Proceso de Instalación de PhpPgAdmin finalizando y capturando el tiempo que emplea en completar la instalación respectiva.

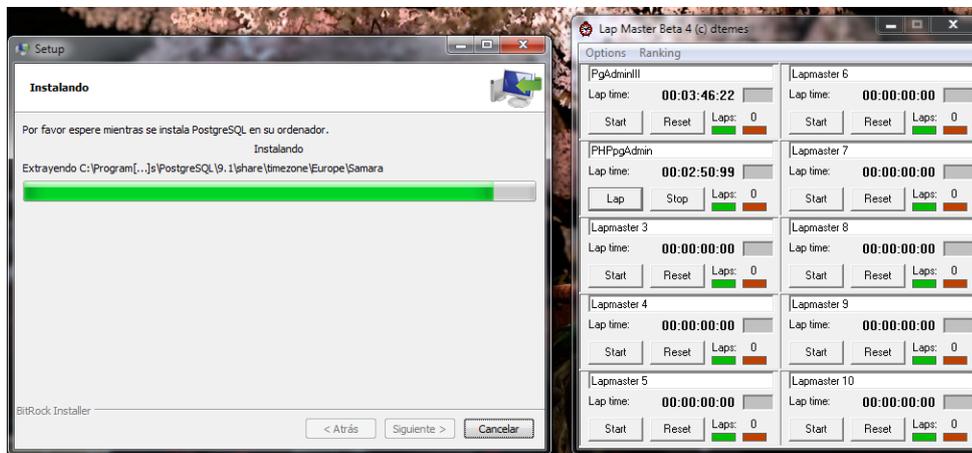


Figura 15: Proceso de instalación de PhpPgAdmin

Tiempo empleado en instalar PhpPgAdmin

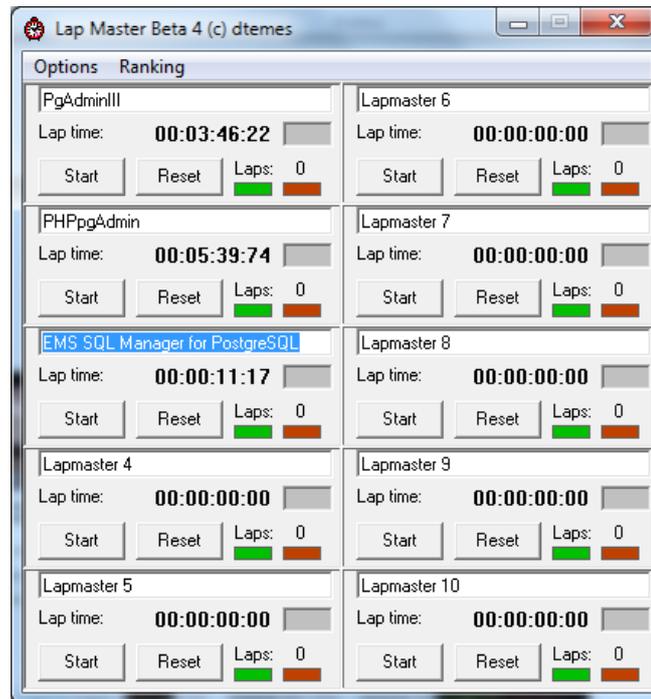


Figura 16:PhpPgAdmin instalado y finalizado

C) EMS SQL Manager forPostgreSQL

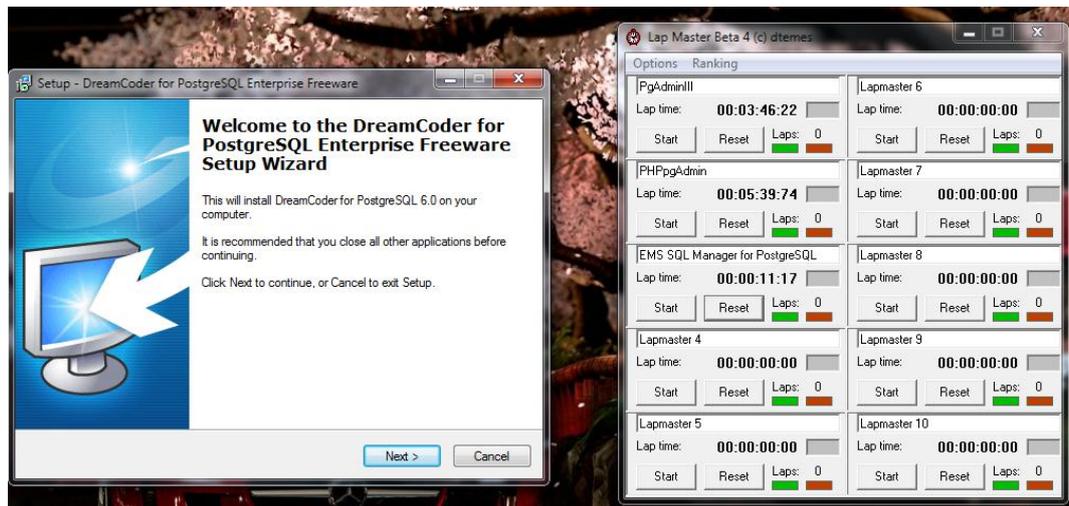


Figura 17:Iniciando la instalación

Iniciando el proceso de contabilización del tiempo que tomará instalar la herramienta.

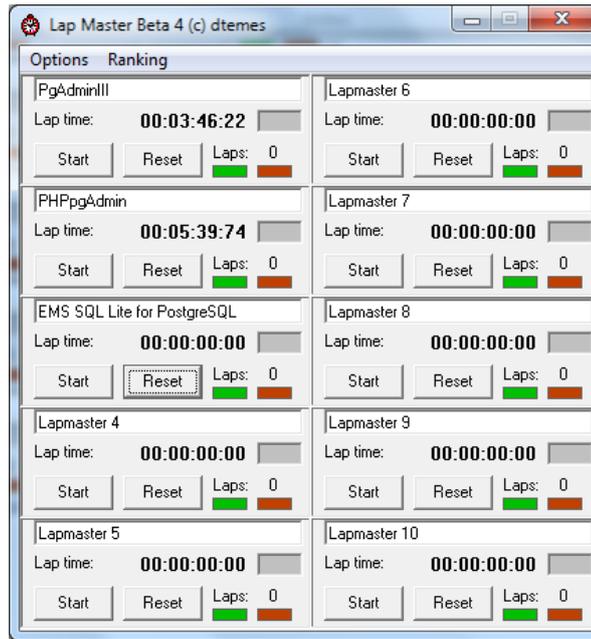


Figura 18: Cargando el software Lap Master

Empezando el Wizard de instalación de la herramienta

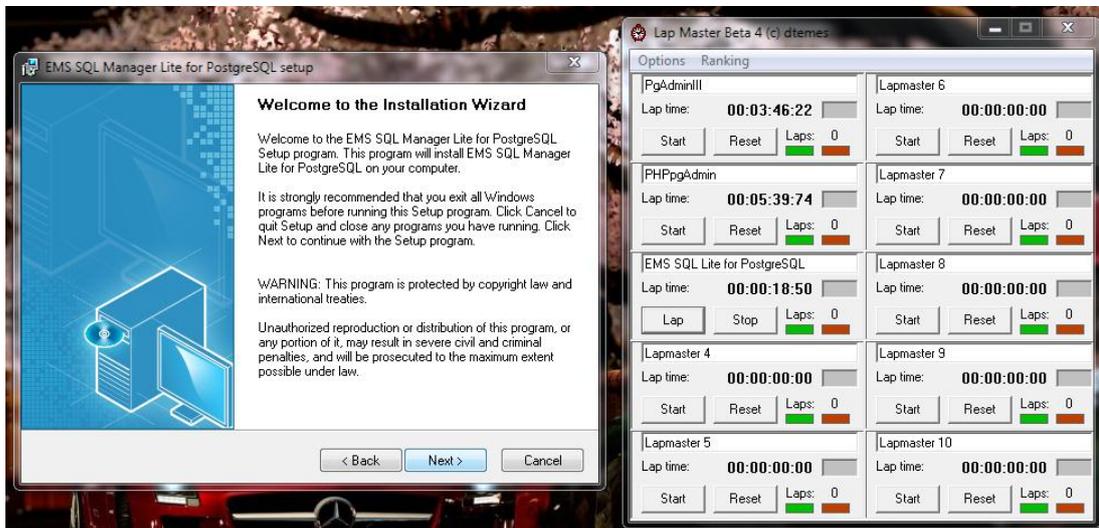


Figura 19: Iniciando herramienta

Tiempo empleado en instalar EMS SQL Lite of Postgresql

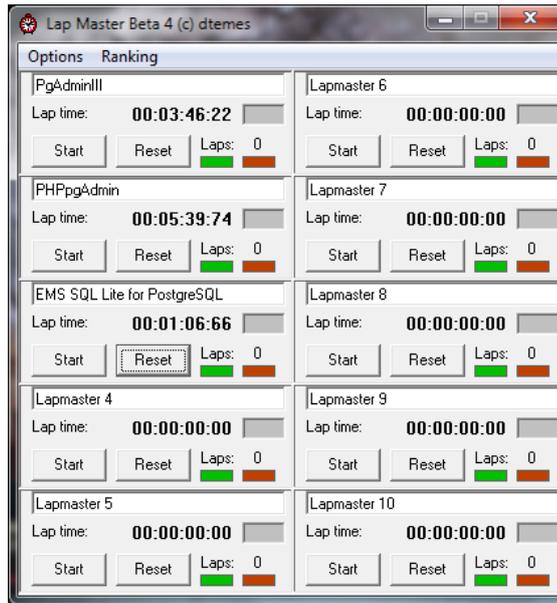


Figura 20: Tiempo empleado en instalar cada herramienta

TABLA I. Tiempos de Instalación

Tiempo de Instalación de Herramientas	PgAdminIII	PhpPgAdmin	EMS SQL Lite of Postgresql
	03:46:22	05:39:74	01:06:66

Calculo tiempo en Segundos

$$(3*60)+46+(22*10)= 446\text{segundos}$$

$$(05*60)+39+(74*10) = 1079 \text{ segundos}$$

$$(01*60)+06+(66*10) = 726\text{segundos}$$

Total= 2251 segundos.

$$2251= 100\%$$

$$446 \quad X= 19,81\% \text{ PgAdminIII}$$

$$2251 \quad 100\%$$

$$1079 \quad X = 47,93\% \text{ PhpPgAdmin}$$

$$2251 \quad 100\%$$

$$726 \quad X= 32,25\% \text{ EMS SQL Lite of Postgresql}$$

INDICADOR 2

PASOS DE CONFIGURACION

1. PGADMINIII

Iniciando el Wizard de instalación de Posatgresql



Figura 21:Instalando postgresql

Definiendo path de dirección de instalación de Postgresql. Y haciendo click en siguiente.



Figura 22:Path de instalación

Mensaje de ayuda al detectar donde puede instalar postgresql en la dirección mencionada

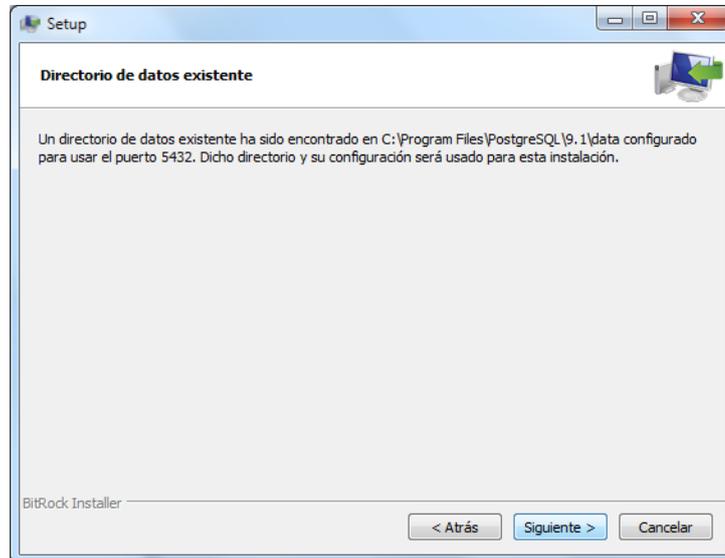


Figura 23: Instalando el puerto de acceso a postgresql

Setup de configuración ingresando contraseña para el servidor postgresql

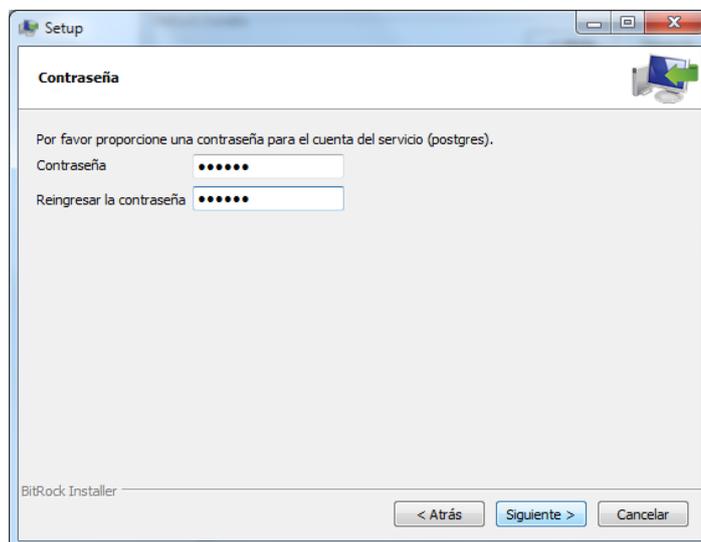


Figura 24: Contraseña de seguridad

Setup listo para inicializar el proceso de instalación

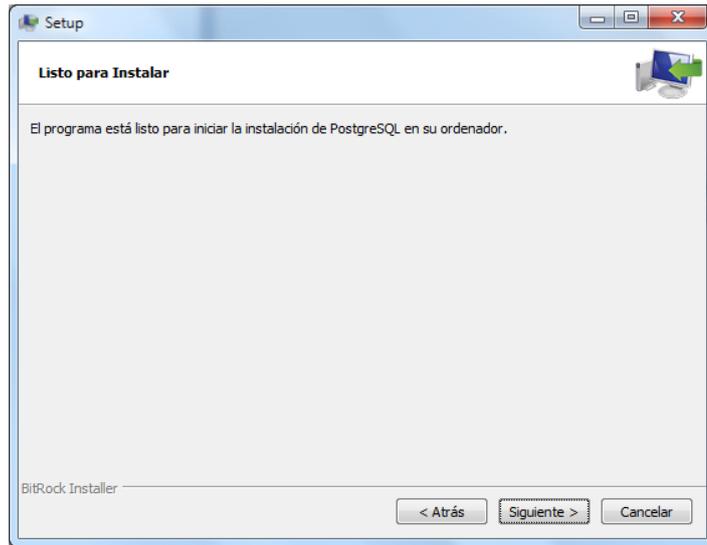


Figura 25: configuración completada.

Instalando postgresql 9.1.1

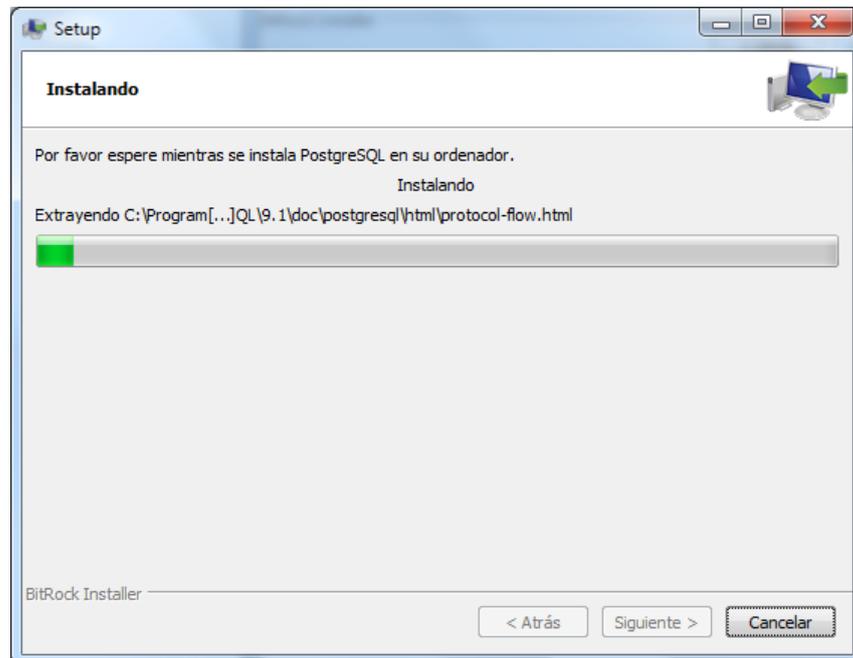


Figura 26: Instalación de paquetes

Terminal de haber completado la instalación



Figura 27: Instalación terminada de postgresql

2. PHPPGADMIN

Mismo proceso de instalación de la herramienta



Figura 28: Instalando postgresql

Mensaje de advertencia ya que existe una instalación de postgresql.

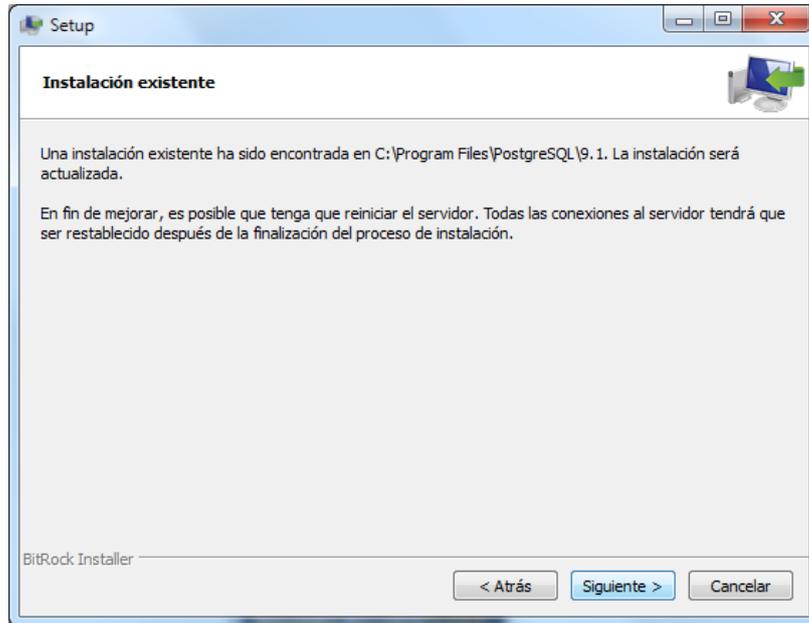


Figura 29: Advertencia de instalación existente

Asignado contraseña para el servidor

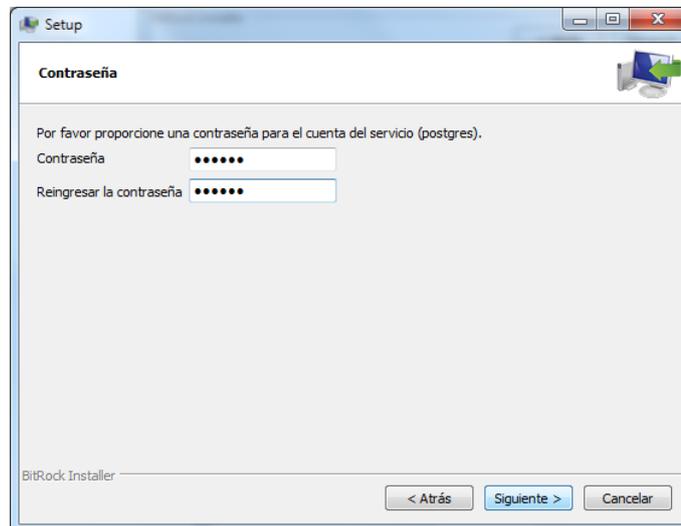


Figura 30: ingresando contraseña segura

Esperando para realizar el proceso para la instalación de la herramienta

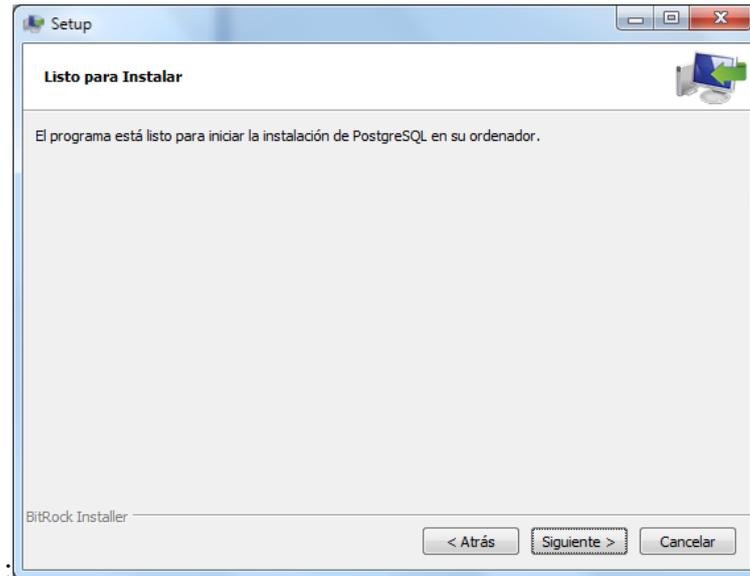


Figura 31: Inicio de instalación

Proceso de instalación y cargando los paquetes necesarios para su utilización.

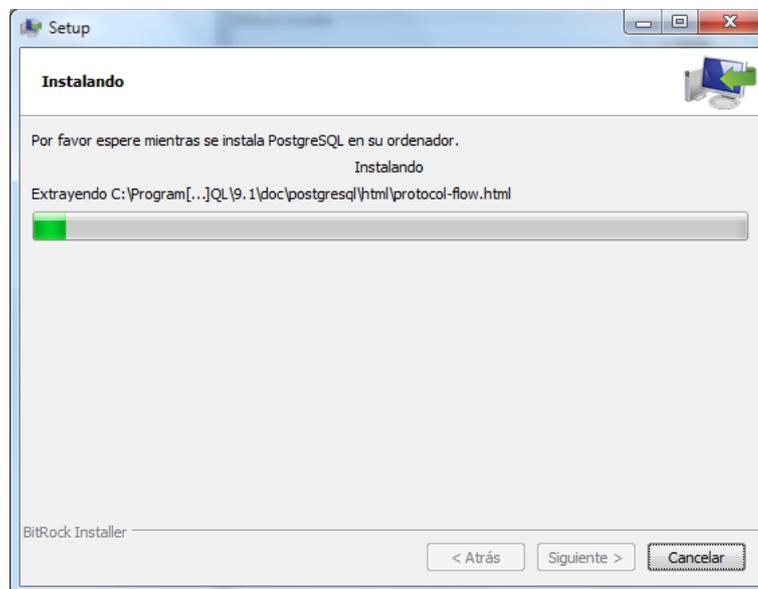


Figura 32: instalando phppgadmin

Finalizando la primera parte de phppgadmin



Figura 33: instalación postgresql

Instalando el Stack Builder de postgresql se necesita seleccionar en postgresql con el puerto 5432

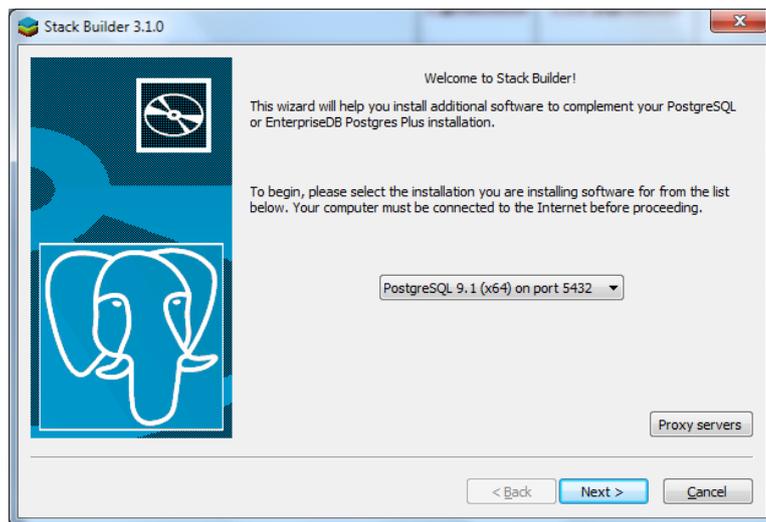


Figura 34: Stack Builder

Dirigirse hasta la opción de Desarrollo web y seleccionar PhpPgAdmin y dar click a siguiente.

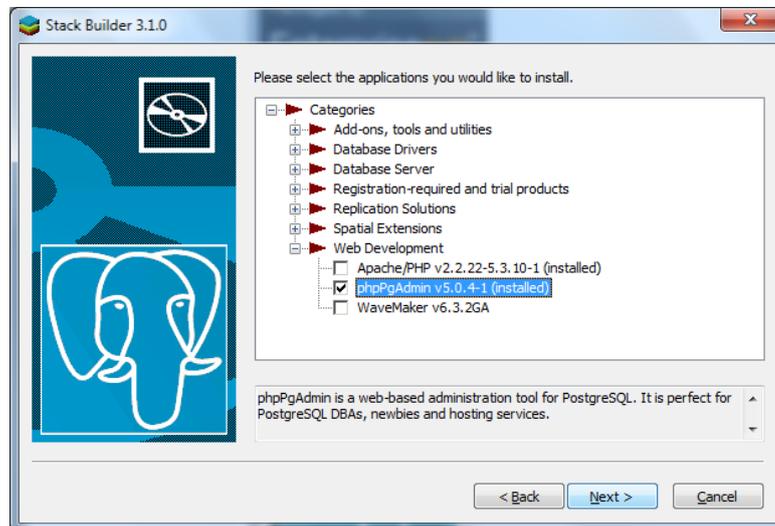


Figura 35: Seleccionar el phppgadmin ha instalar.

Una vez seleccionado damos click a continuación en next

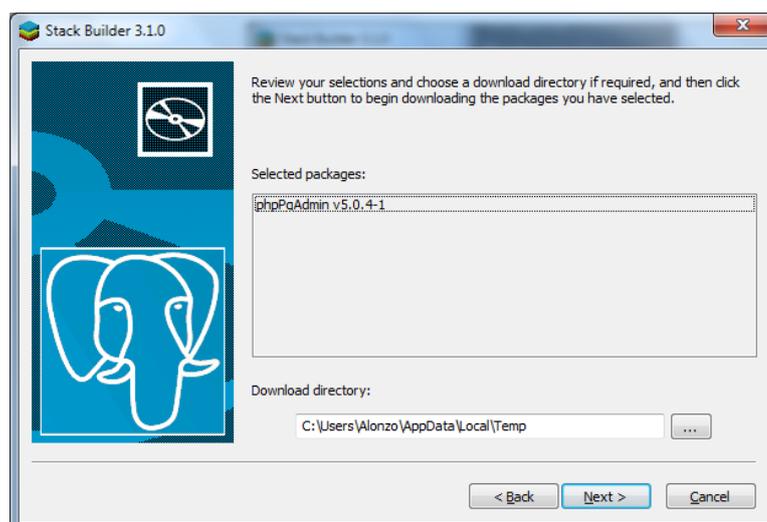


Figura 36: Selección PhpPgAdmin

Últimos pasos antes de comenzar con la instalación.

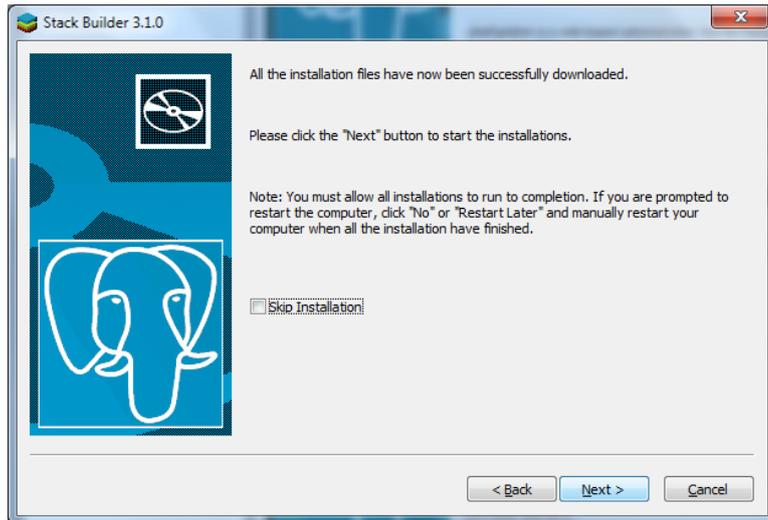


Figura 37: Iniciando la instalación

Progreso de la instalación de la herramienta

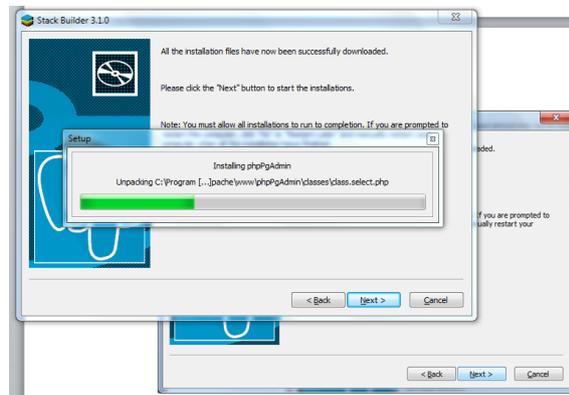


Figura 38: instalando componentes necesarios.

Información sobre cómo se ha completado la información

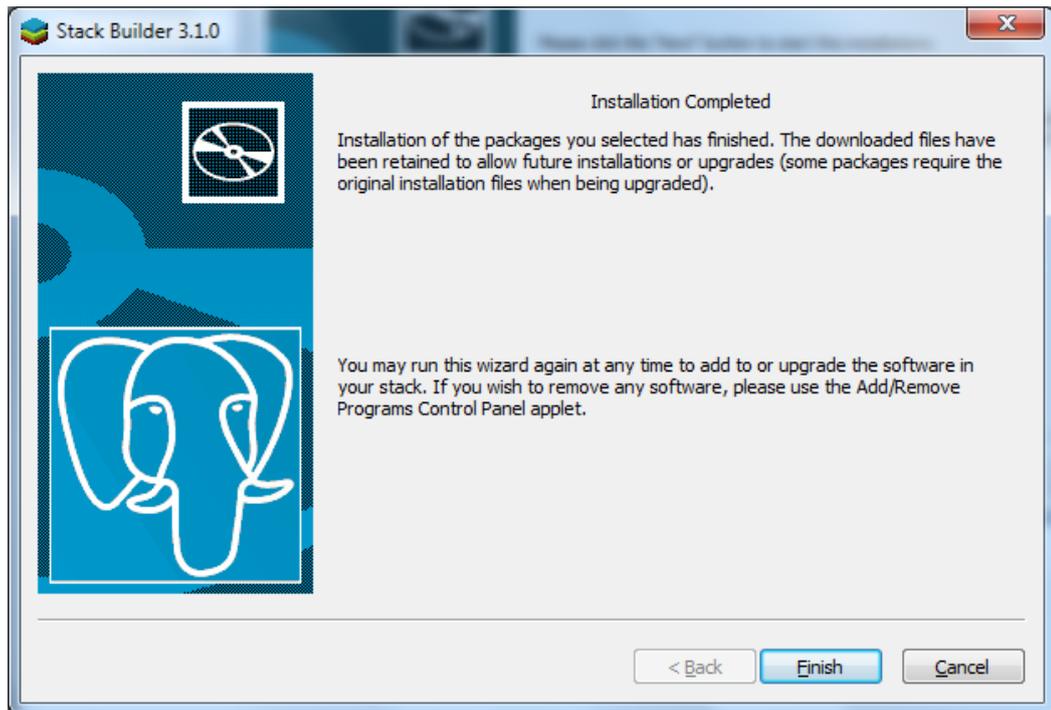


Figura 39: Instalación completa

3. EMS SQL LITE FOR POSTGRESQL

Idiomas disponibles para la instalación inicial

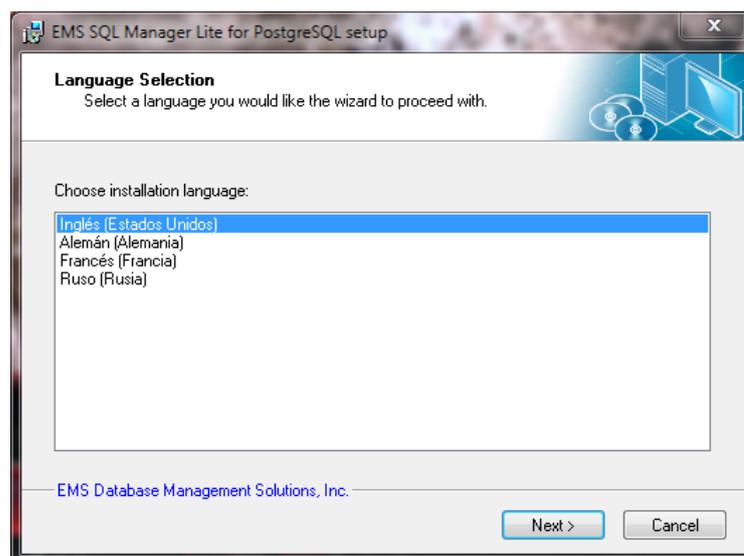


Figura 40: Proceso de instalación.

Wizard de ayuda en el proceso de instalación.

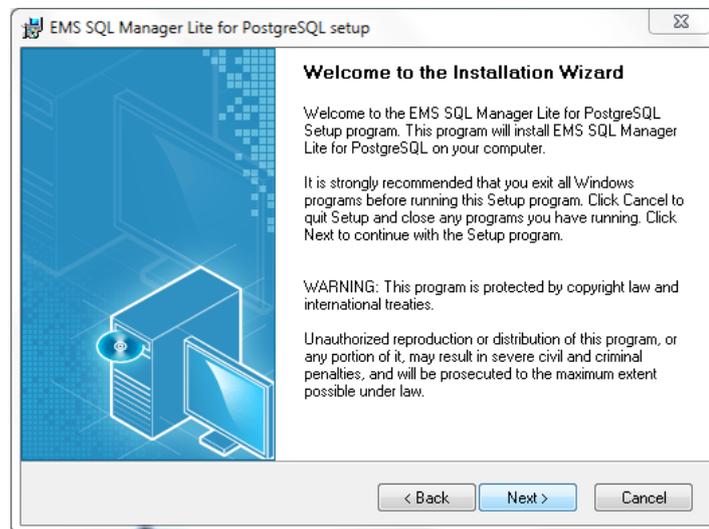


Figura 41: Bienvenida al proceso de Instalación.

Acuerdo de licencia para proseguir con la instalación y configuración de EMS SQL.

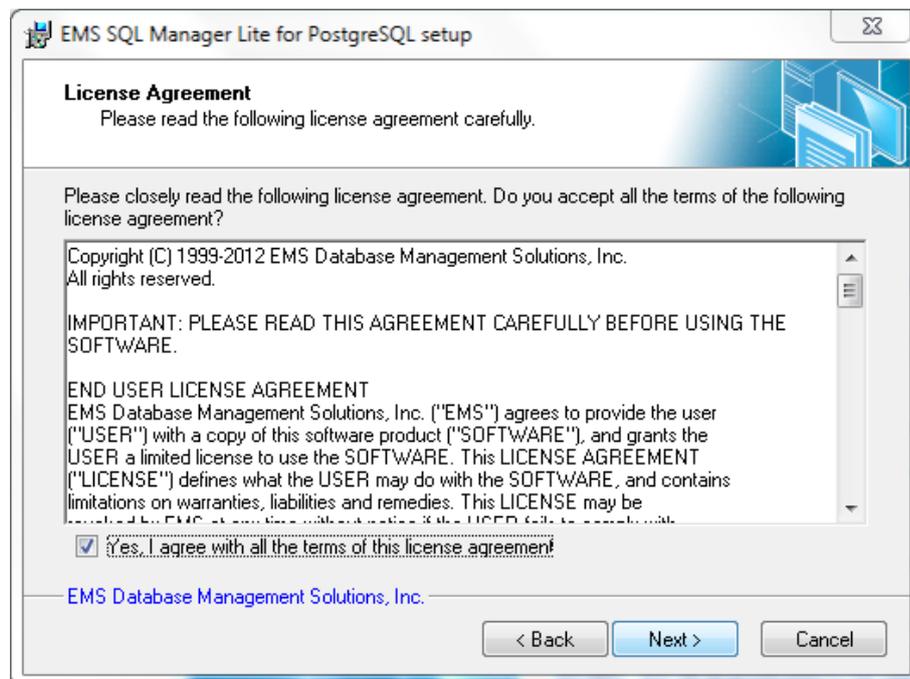


Figura 42: Acuerdo de Licencia.

Directorio donde se instalará EMS SQL

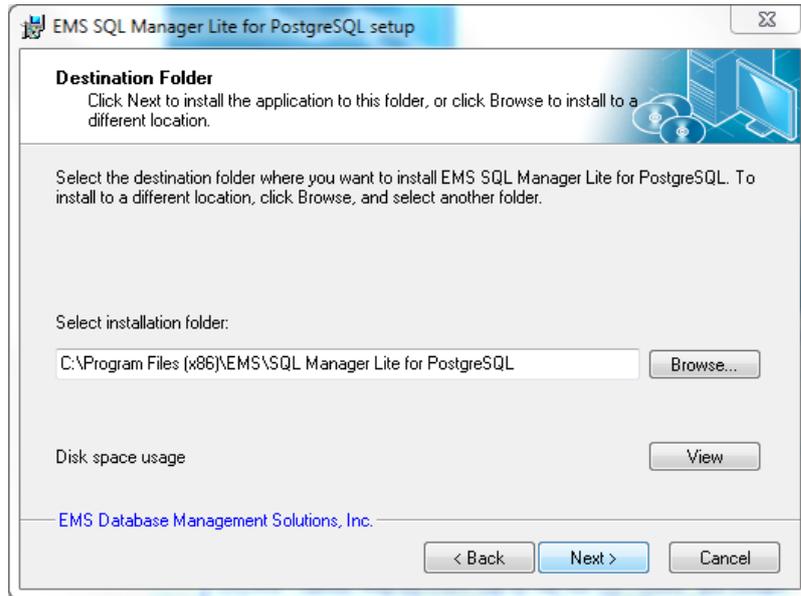


Figura 43: Path Instalación.

Instalar como un programa en grupo

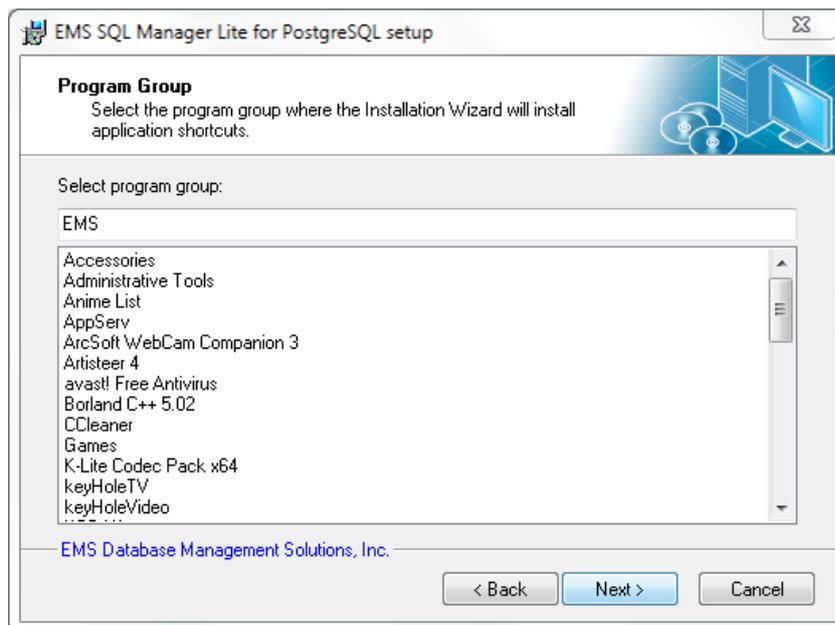


Figura 44: Programa Inicio

Mensaje de seguridad que da inicio a la instalación luego de haber disparado

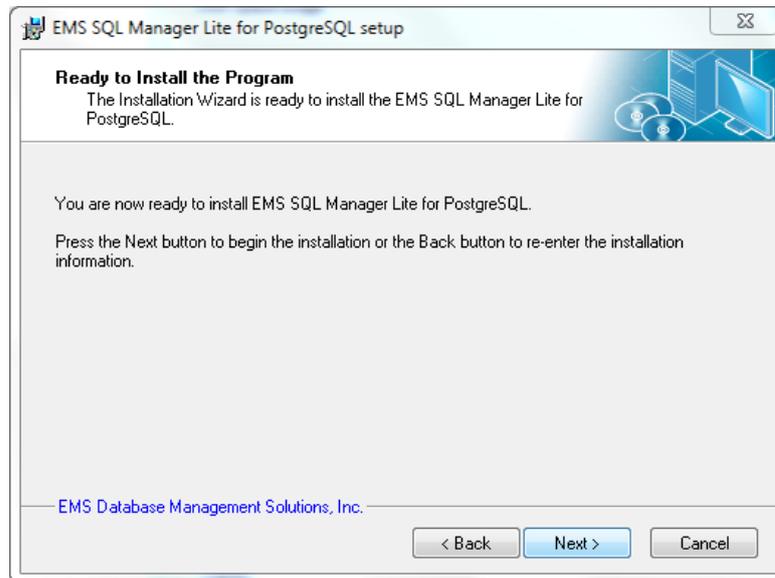


Figura 45: todos los controles

Status de Navegación en la herramien

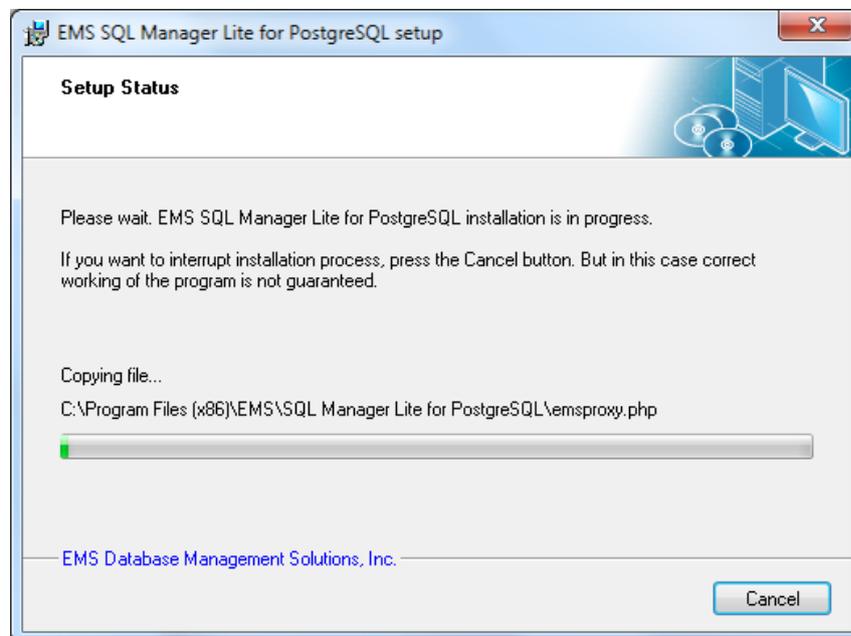


Figura 46: Progreso de la Instalación

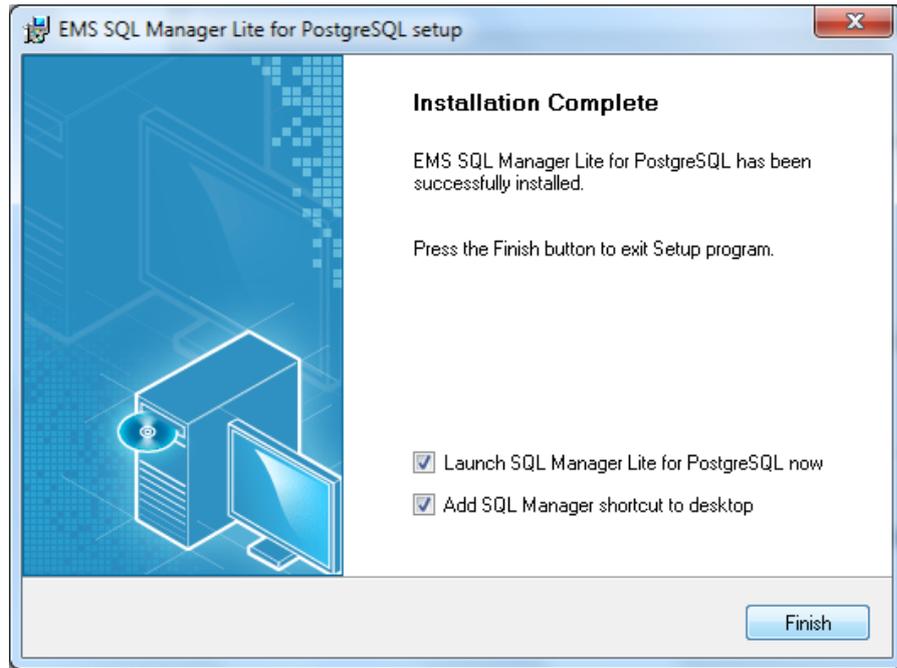


Figura 47: Instalación completa

NIVELES DE CONTROL DE ACCESO

1. PGADMINIII

Modo de controlar el acceso al sistema por medio de una contraseña

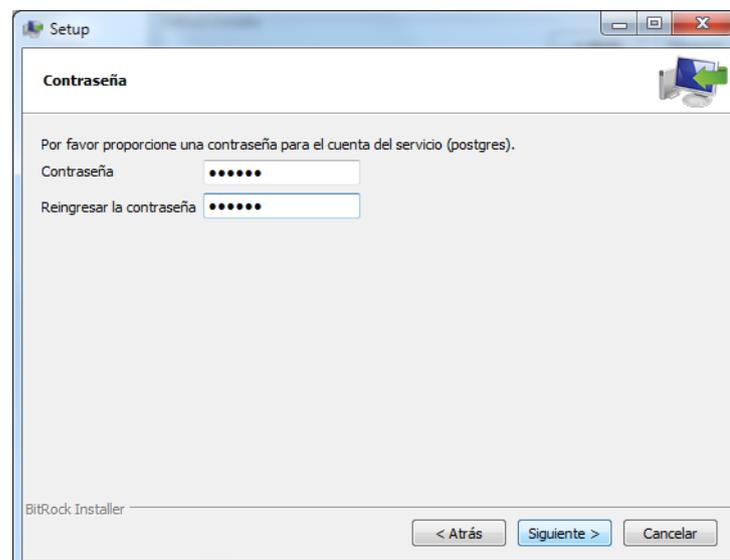


Figura 48: Acceso por contraseña

Control de acceso a

1. PHPPGADMIN

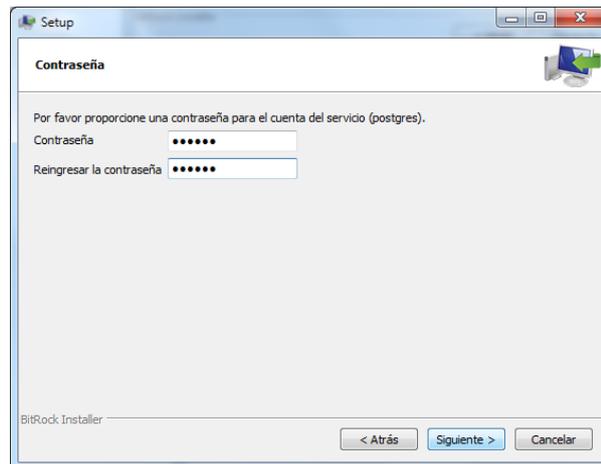


Figura 49: Acceso por contraseña phppgadmin

Pasos necesarios de configuraci

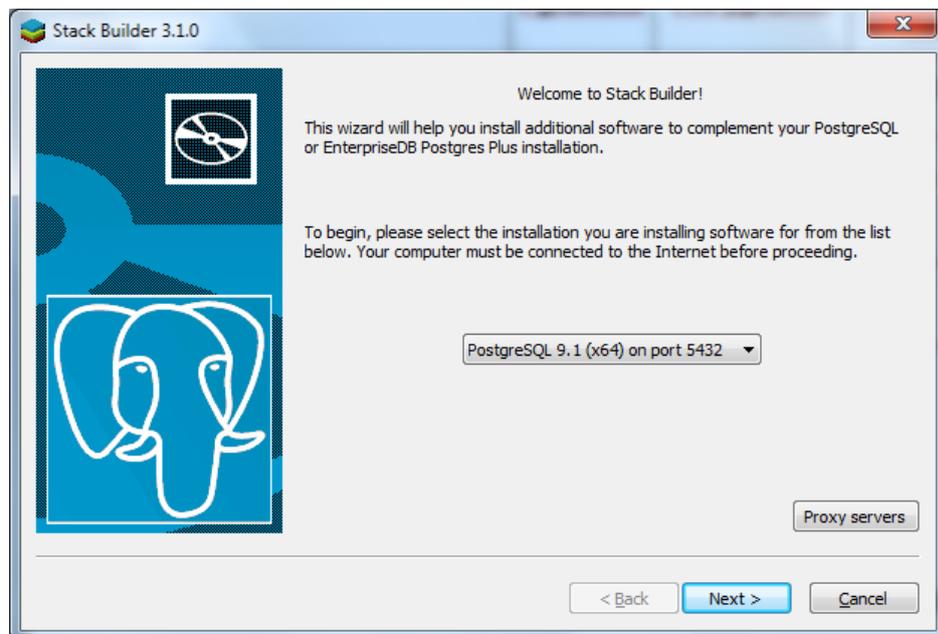


Figura 50: Stack Builder.

INDICADOR 2: SEGURIDAD

PGADMINIII

Conexión Local

Formas de conectarse con postgresql propiedades

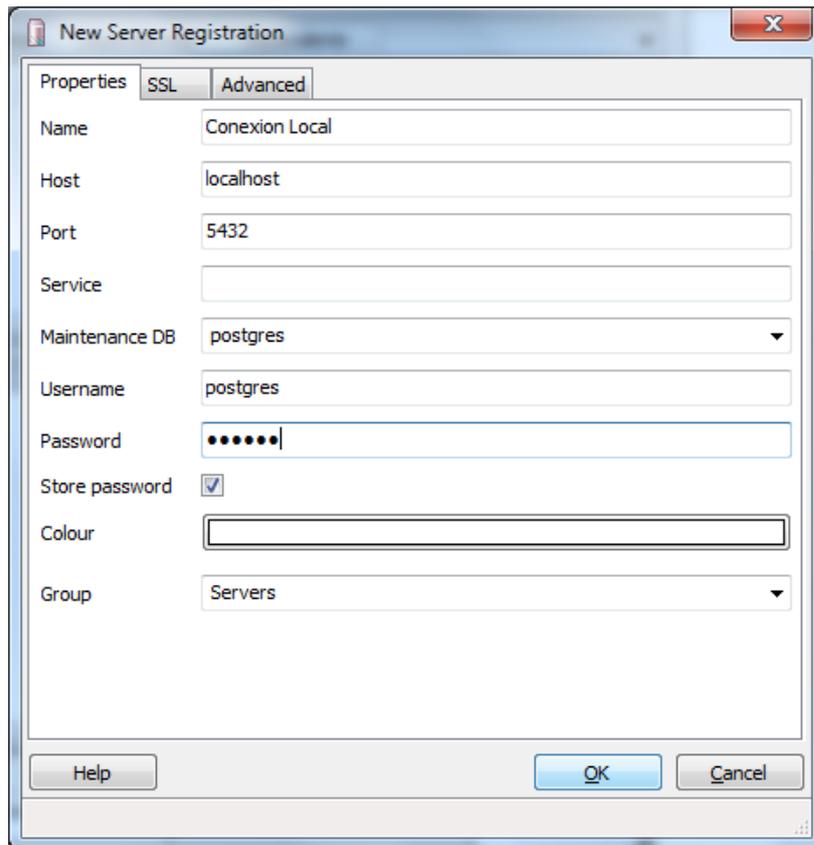


Figura 51: Conexión Local

Modos de conectarse localmente a bases de datos con PgadminIII

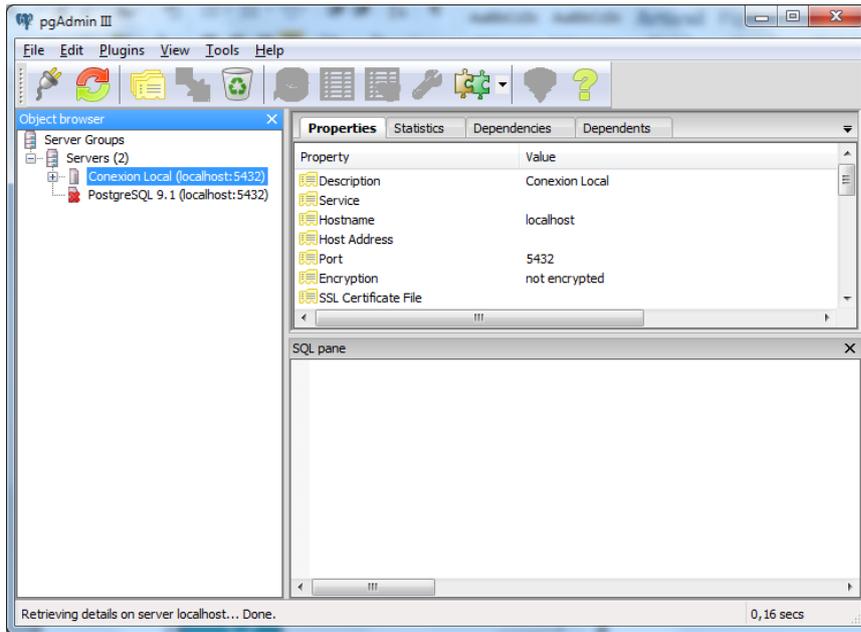


Figura 52: Objetos de PgadminIII

Características generales y propiedades

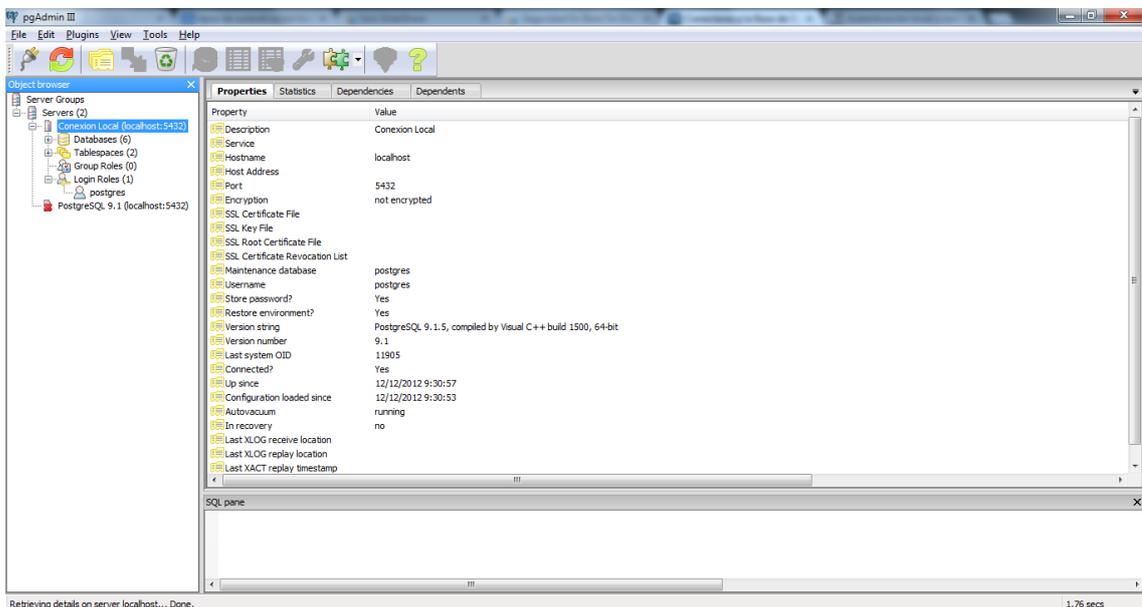


Figura 53: Conexión de Servidores

Conexión Remota de PgAdminIII

Se debe conocer host, usuario y permisos necesarios para acceder a la base de datos.

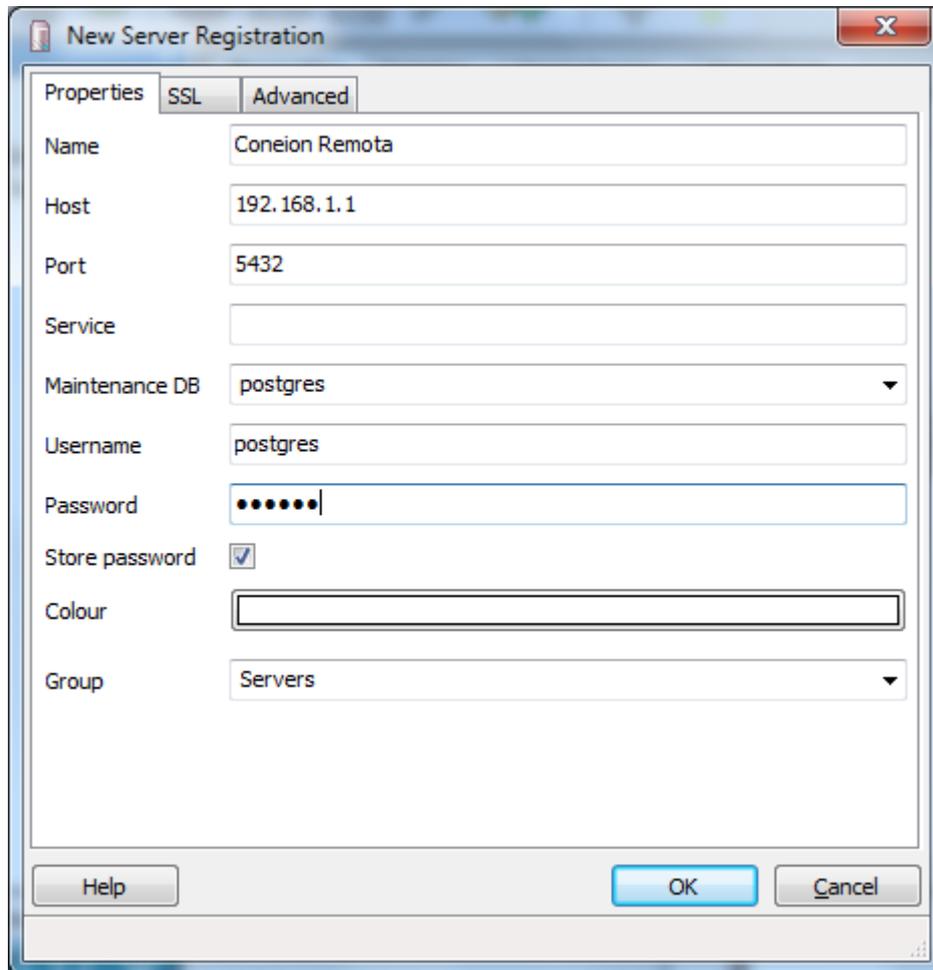


Figura 54: Conexión Remota

2. PhpPgAdmin

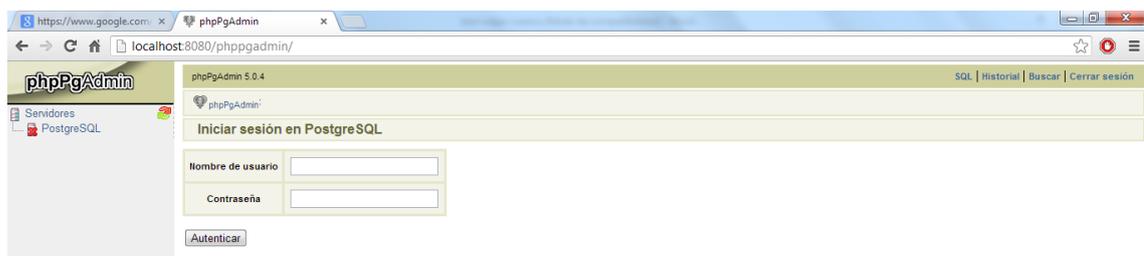


Figura 55: Conexiones de Clientes de Postgresql

3. EMS SQL LITE PARA POSTGRESQL

Conexión Local

Se requiere de información relevante sobre la base de datos, puertos utilizados, para acceder a la base de datos.

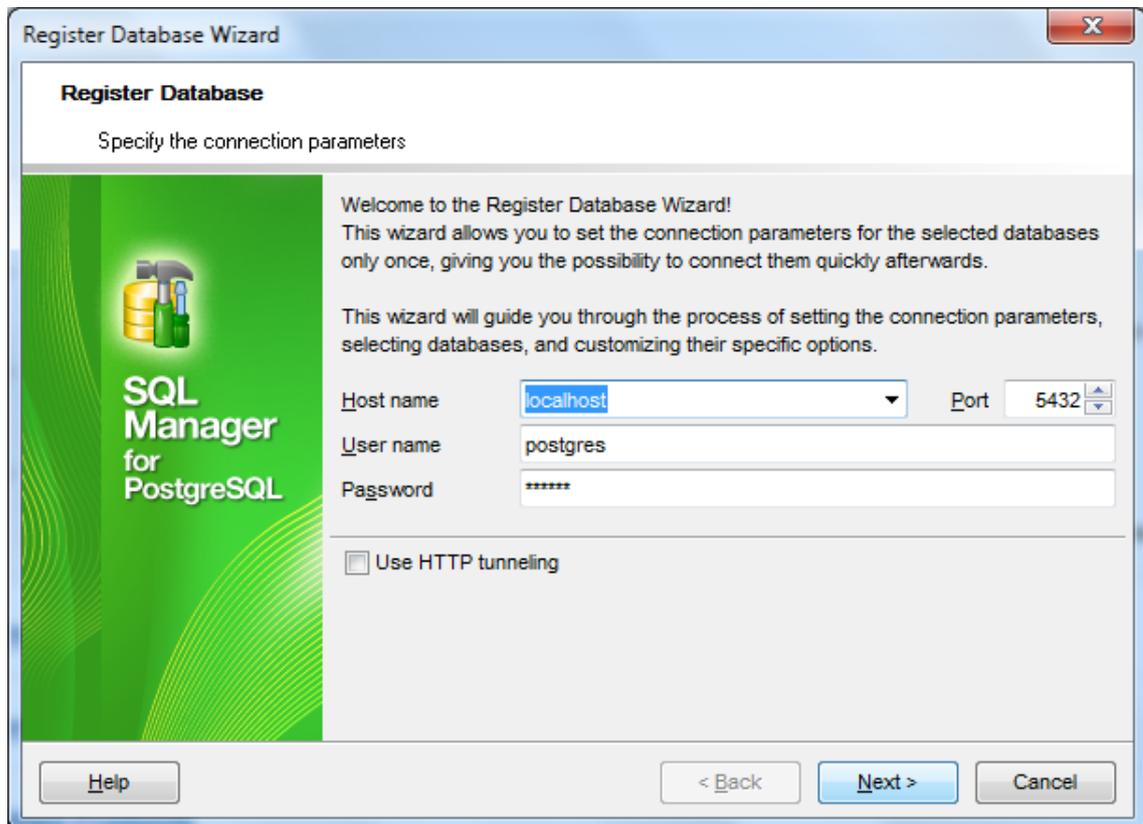


Figura 56: Conexión Local de bases de datos

Conexión Remota

Se requiere de credenciales para su acceso respectivo

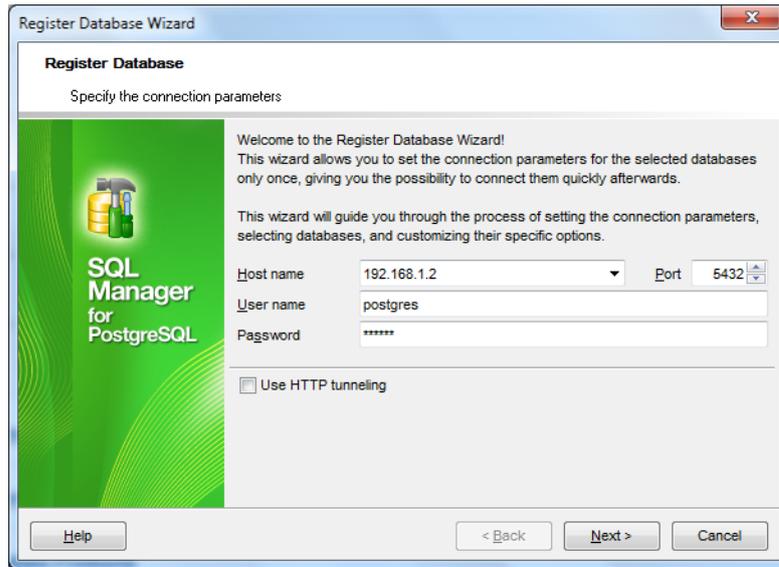
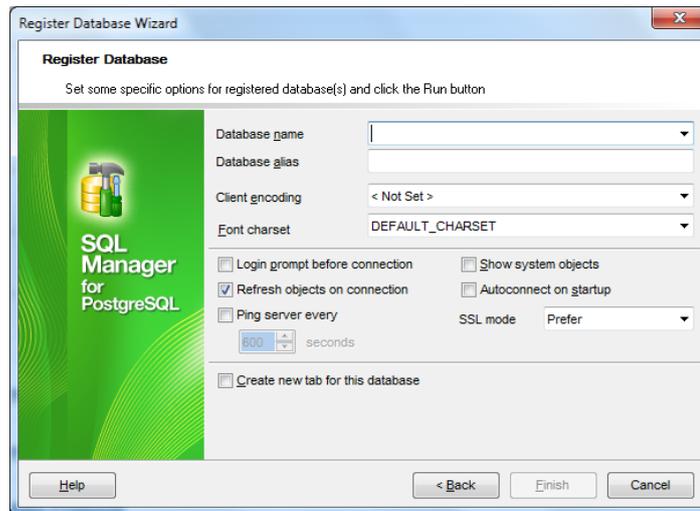


Figura 57: Conexión remota a EMS SQL Lite

Información requerida para registrar una nueva base de datos.



3.11. PgAdminIII

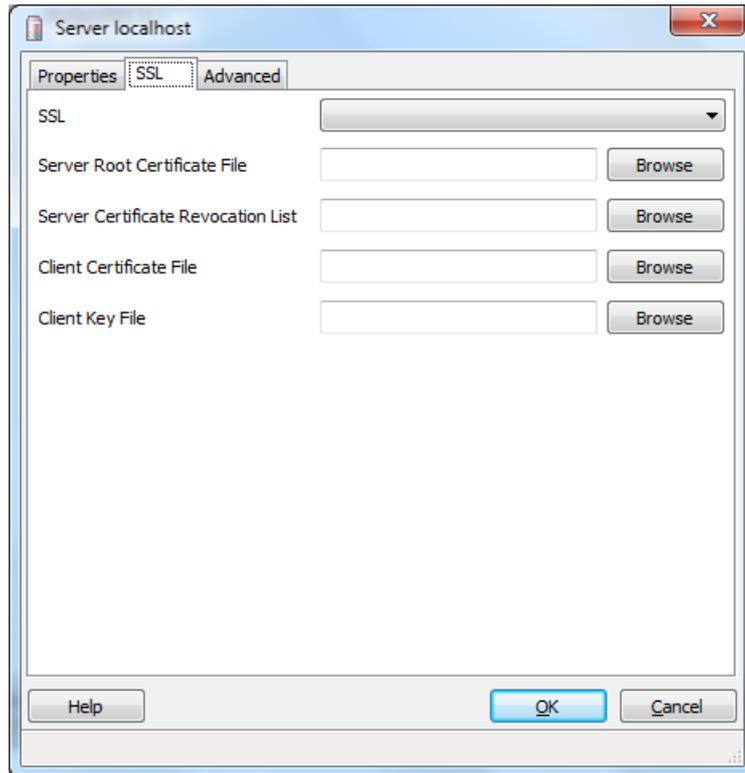


Figura 58: Certificados de Seguridad

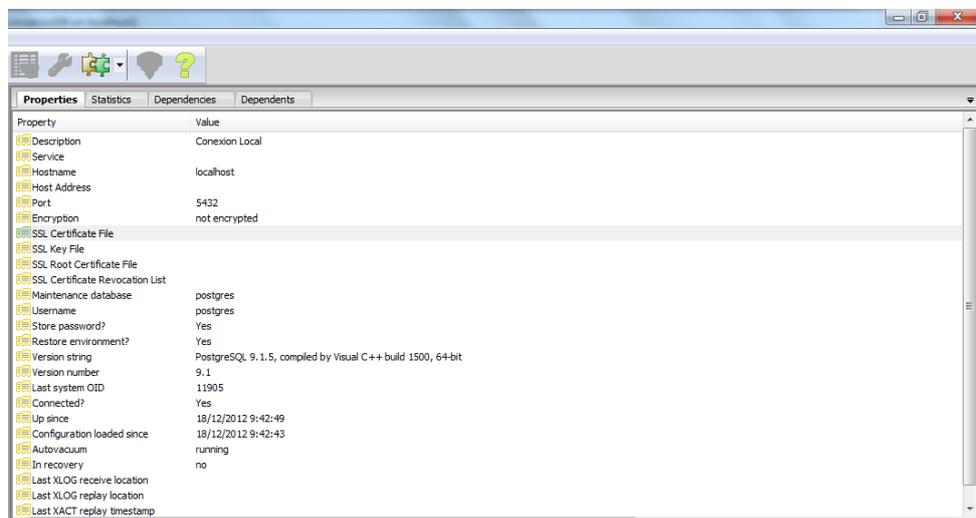


Figura: Descripción general de una base de datos en PgadminIII

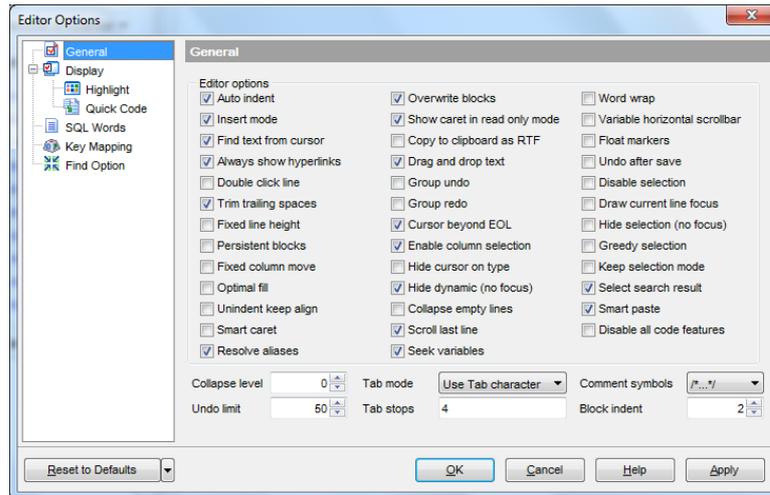


Figura 59: Opciones generales EMS SQL Lite for postgresql

Indice 4: Calidad

1. PgAdminIII

Ejecución del cliente postgresql pgadminIII, como se aprecia al ejecutar

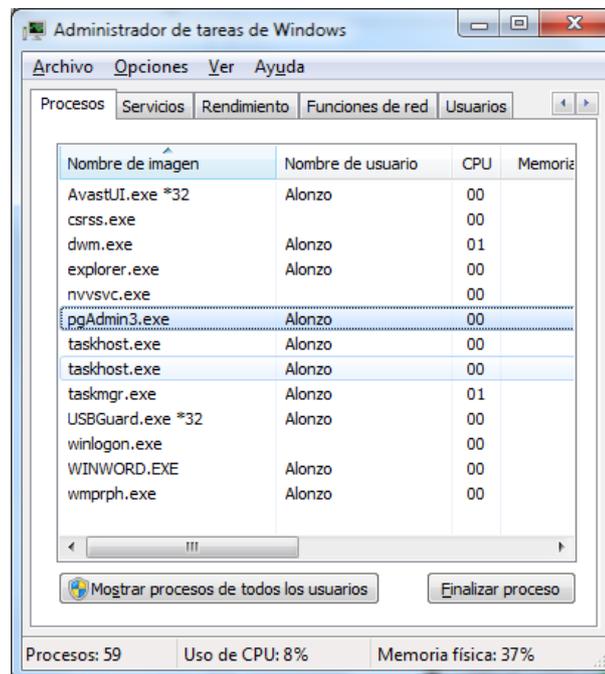


Figura 60: Recursos necesarios al ejecutar cliente

Medición de recursos finales ocupados

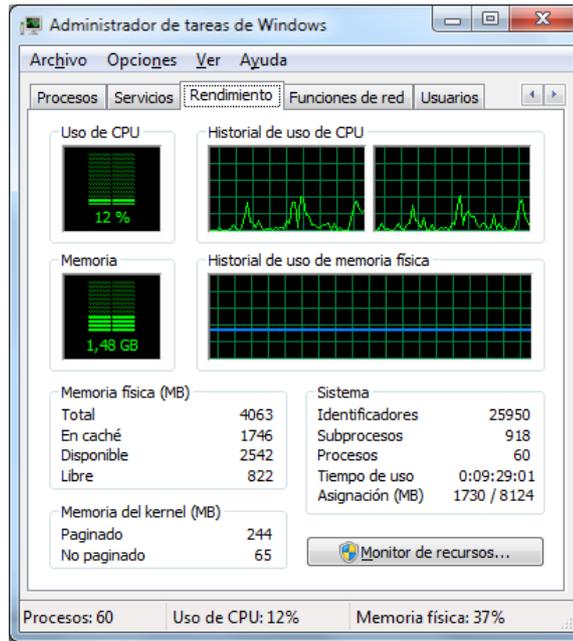


Figura 61: Resultados Generales

2. PhpPgAdmin

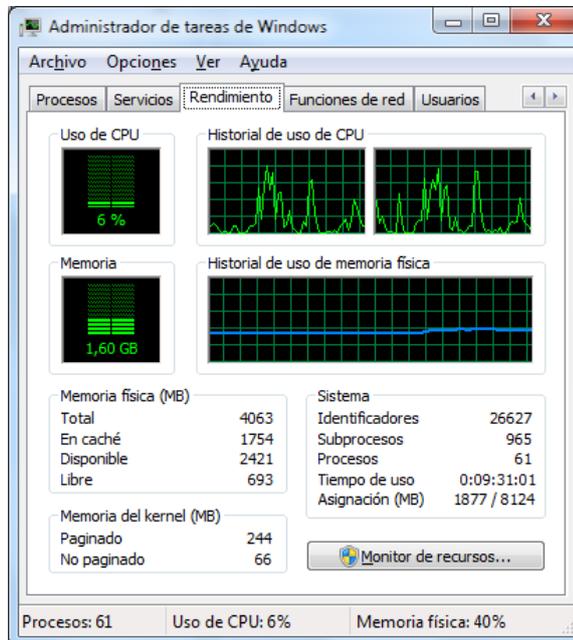


Figura 62: resultados obtenidos de la búsqueda

Cliente ejecutando aplicaciones de bases de datos.

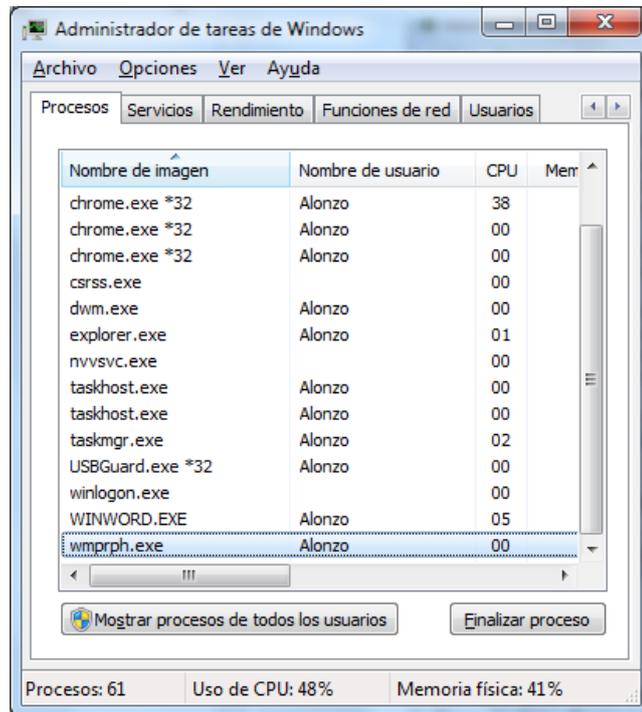


Figura 63: Ejecución proceso phppgadmin

EMS SQL Lite for Postgresql

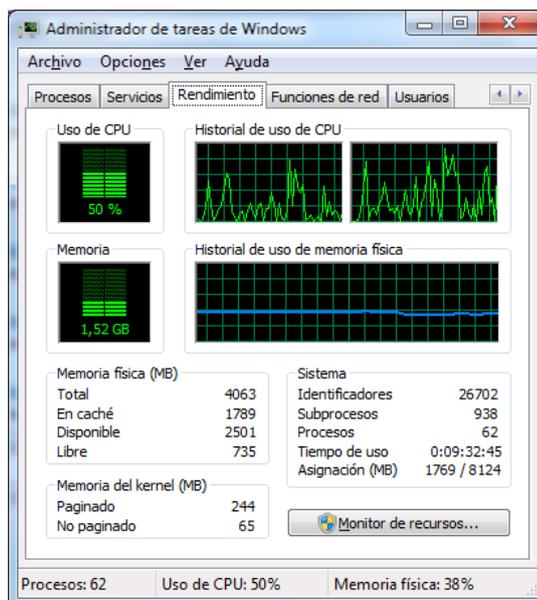


Figura 64: Resultados parciales obtenidos

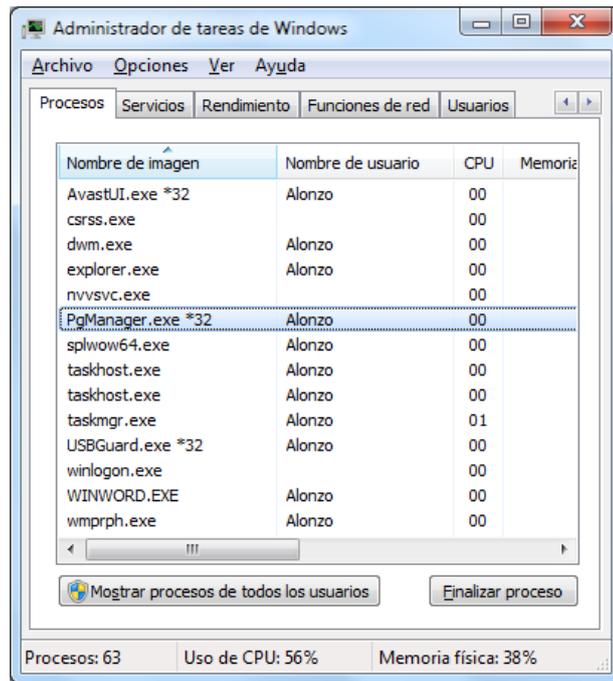
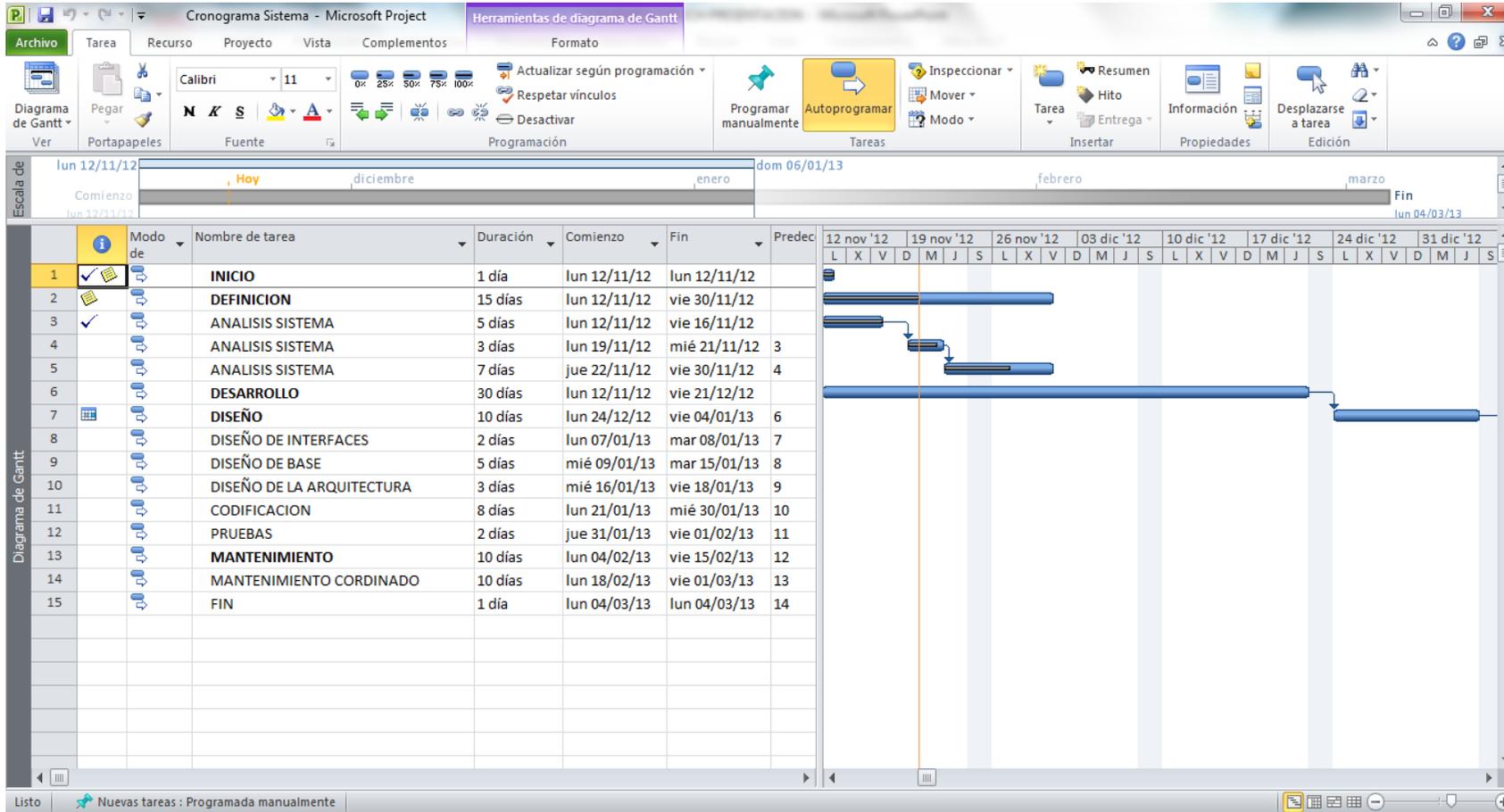


Figura 65: Resultados Obtenidos

ANEXO 4

Diagrama de Actividades Planificadas



ANEXO 5

Análisis Comparativo de Herramientas de Administración GUI Postgresql vs Modo Consola de Postgresql

Preguntas de la Encuesta del modo consola de postgresql

¿Es rápido el proceso en instalación de postgresql?

Si () No ()

¿Hay que configurar la consola de postgresql para acceder a la información?

Si () No ()

¿El acceso al modo consola de postgresql tiene algún control?

Si () No ()

¿Existe ayuda para manipular en modo consola de postgresql?

Si () No ()

¿Para acceder a la consola de postgresql hay que autenticarse?

Si () No ()

¿Tiene algoritmos de seguridad la consola de postgresql?

Si () No ()

¿Es segura la sesión del modo consola de postgresql?

Si() No()

¿Se pueden realizar respaldos por medio del modo consola de postgresql?

Si() No()

¿Se puede acceder al modo consola de postgresql de forma local?

Si() No()

¿Se puede acceder al modo consola de postgresql de forma remota?

Si() No()

¿Se pueden crear bases de datos por medio de la consola de postgresql?

Si() No()

¿Se pueden crear tablas dentro de una base de datos por medio de la consola de postgresql?

Si() No()

¿Se pueden crear funciones dentro de una base de datos por medio de la consola de postgresql?

Si() No()

¿Se pueden realizar consultas query por medio de la consola de postgresql?

Si() No()

¿Es ligera la consola de postgresql?

Si() No()

¿Es fácil de usar la consola de postgresql?

Si() No()

¿Es compatible con otros programas la consola de postgresql?

Si() No()

Resultados de Valores obtenidos

en: <https://docs.google.com/spreadsheets/ccc?key=0AusLyFwUzndWdGI5YTdwYldZRFdoaVNoVmlGZFozLUE#gid=0>

Tabla II. Valorización modo consola postgresql

Cuantitativa	0	1	2	3	4
Cualitativa	NO				SI
	Malo	Regular	Bueno	Medianamente Bueno	Muy Bueno
	Muy Difícil	Difícil	Medianamente Fácil	Fácil	Muy Fácil

Tabla III. Cuadro Resultados de Comparación

INDICADORES	INDICES	CON HERRAMIENTAS GUI POSTGRESQL			SIN HERRAMIENTAS GUI POSTGRESQL
		PgAdminIII	PHPpgAdmin	EMS SQL Lite for PostgreSQL	MODO CONSOLA
Instalación	Tiempo en Completar Instalación	4	3	4	4
	Pasos de Configuración	3	2	3	0

	Nivel de Control de Acceso	4	4	4	3
	Nivel de ayuda en los procesos de instalación.	4	3	4	1
Seguridad	Tipo de autenticación usada	4	2	4	1
	Tipos de algoritmos de seguridad	4	3	3	3
	Nivel de seguridad para establecer sesión	4	2	4	1
	Tipos de soporte de respaldo y restauración	4	2	3	2
Administración	Conexión local	4	4	4	4
	Conexión remota	4	0	4	4
	Facilidad diseño de Base de Datos.	4	3	4	0
	Facilidad diseño de Tablas	4	3	4	0
	Facilidad diseño de Funciones	4	3	4	0
	Diseño y Consultas de Querys	4	2	4	0
Calidad	Uso de recursos hardware	3	2	1	3
	Facilidad de Uso	4	0	4	4
	Compatibilidad Software	4	3	4	3
TOTAL		66	41	62	33

Herramientas = 169

Consola = 33

Total = 202----100%

Al utilizar herramientas de administración 83,86%

Al utilizar modo consola postgresql 16,34%

Modo Consola postgresql

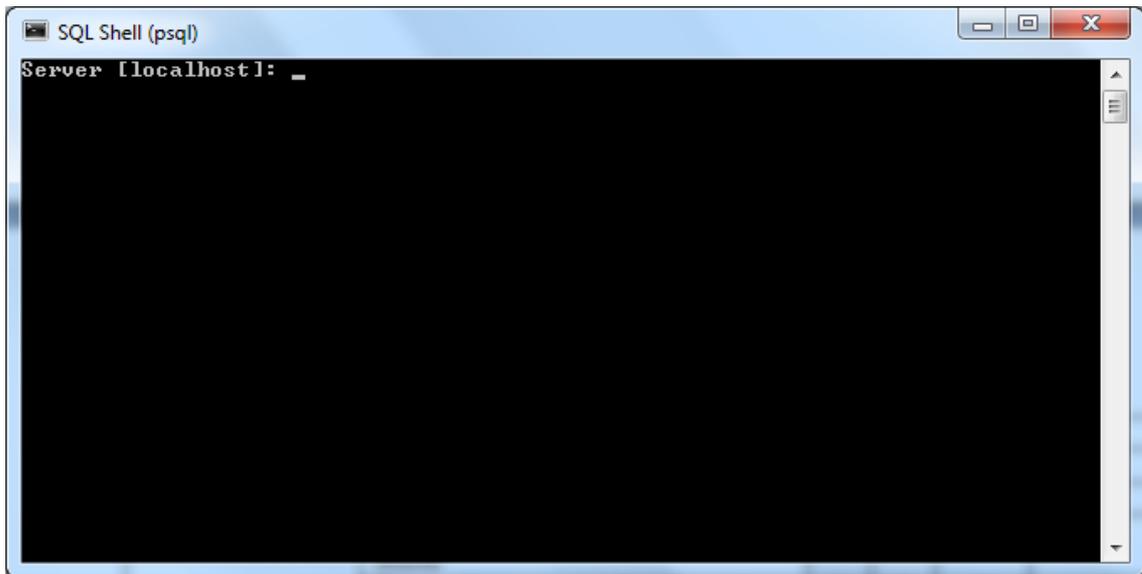


Figura Acceso modo consola postgresql

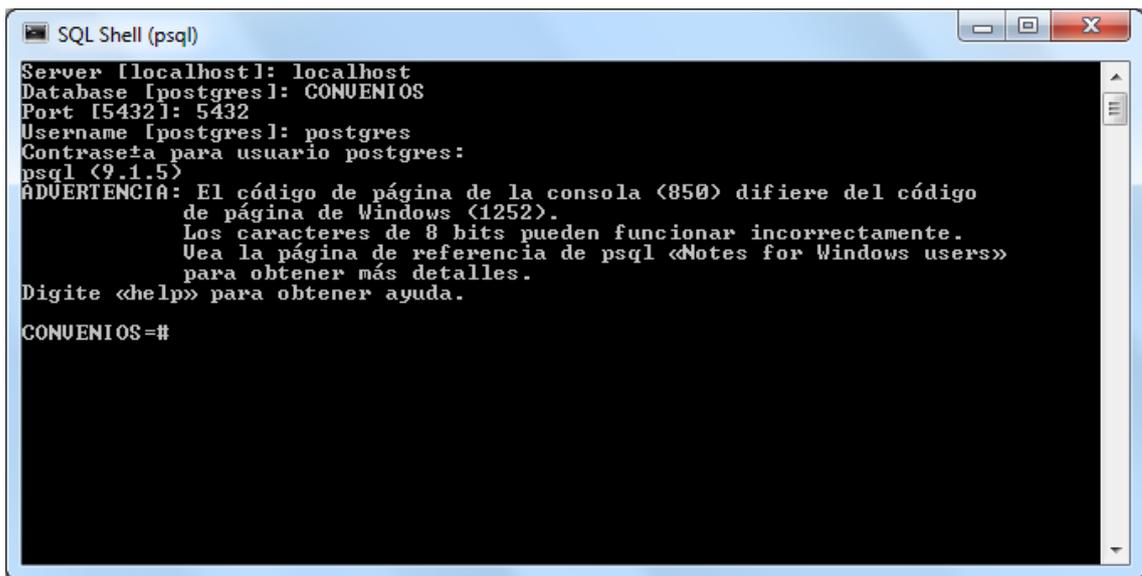


Figura Modo consola de postgresql con una base de datos

ANEXO 6

MANUAL DE USUARIO

SISTEMA CONVENIOS-ESPOCH

A. Introducción

El Manual de usuario del portal web de de la Comisión de Vinculación con la Colectividad denominado CONVENIOS-ESPOCH, se mostrará toda la información necesaria para que pueda operar con mucha facilidad. Su estructura le permitirá identificar sencillamente cuales son las operaciones que puede realizar de forma tal que aproveche su máxima capacidad y funcionalidad de acuerdo con todo lo que ofrece el Sistema.

B. Requisitos mínimos para la Instalación

Los requerimientos mínimos que necesitamos para su correcto funcionamiento del portal web son muy importantes, por lo que sugerimos el tipo de plataforma, librerías que deben estar instaladas previamente tanto como al servidor y el cliente.

➤ Requerimientos de Infraestructura

Hablaremos del Hardware que necesitamos como mínimo

Servidor:

- Procesador PIV o Superior
- Memoria: 512MB o Superior
- CD-ROM: 52X
- Espacio Disco Duro: 2GB o Superior
- Tarjeta de Red: 10/100/1000

Cliente:

- Procesador: PII o Superior
- Memoria: 256 MB o Superior
- CD.ROM:52X
- Espacio Disco Duro: 2 GB o Superior
- Tarjeta de Red: 10/100/1000

➤ **Requerimientos de Plataforma**

Hablaremos del Software que necesitamos:

Servidor:

Apache 2.2.14 o Superior

PHP 5.2 o Superior

PgAdminIII 9.0

Netbeans 7.0

Conexión a Internet.

Cliente:

Navegador IExplorer, Firefox,Chrome, Opera u otro

Conexión a Internet

Se recomienda para una óptimavisualización, resolución de pantalla de 1024 x 768 píxeles o superior.

C. Operatoria general del Sistema

Se detallará de manera correcta el funcionamiento del portal como la administración por parte de los usuarios.

Ingreso al Sistema

USO DEL SISTEMA

1.1. Autenticación

- Para utilizar el sistema el usuario debe autenticarse mediante el ingreso de su usuario y su contraseña, caso contrario si no ingresa dichos datos muestra un mensaje de aviso.



The screenshot shows a web browser window with the address bar displaying 'localhost:8084/Convenio/Login.jsp'. The page has a dark red background. At the top, there is a header with a logo on the left consisting of two interlocking gears, the text 'CONVENIOS ESPOCH' in the center, and a circular seal on the right. Below the header, the main content area features the title 'Acceso Sistema' in white. To the left of the login fields is an illustration of a padlock with a keyhole, and two stylized human figures (one blue, one yellow) standing in front of it. To the right of the illustration are two white input fields: the first is labeled 'LOGIN:' and the second is labeled 'PASSWORD:'. Below these fields are two buttons: 'Aceptar' and 'Cancelar', both with a red gradient and white text.

1.2. Pantalla Principal

Seguido de la autenticación el sistema muestra el menú principal en el cual el usuario puede seleccionar la opción que desee.



La pantalla principal tiene una barra de menú que consta de lo siguiente:

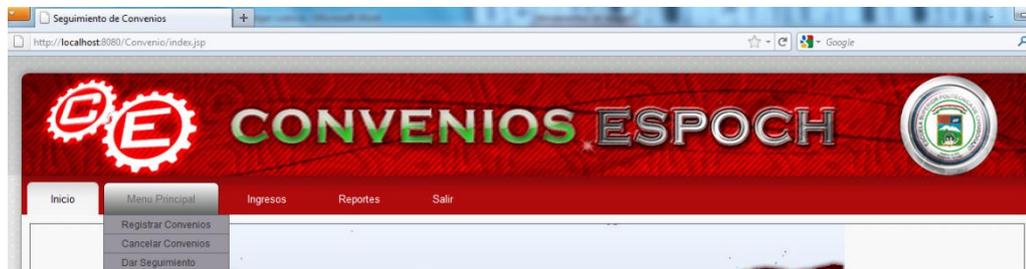
- Inicio
- Menú Principal
- Ingresos
- Reportes

2. MENÚ INGRESO

Al seleccionar la opción **Ingreso** se despliega un módulo de componentes el cual intervienen en un convenio; dichos componentes son el propio convenio, la entidad con la realiza el convenio, coordinador designado, beneficio, beneficiarios del convenio, actividades que se realiza, área en el que se desarrolla, una rendición de cuentas y la emisión y recepción de informes de cada convenio, en donde se puede acceder a cada uno de ellos.

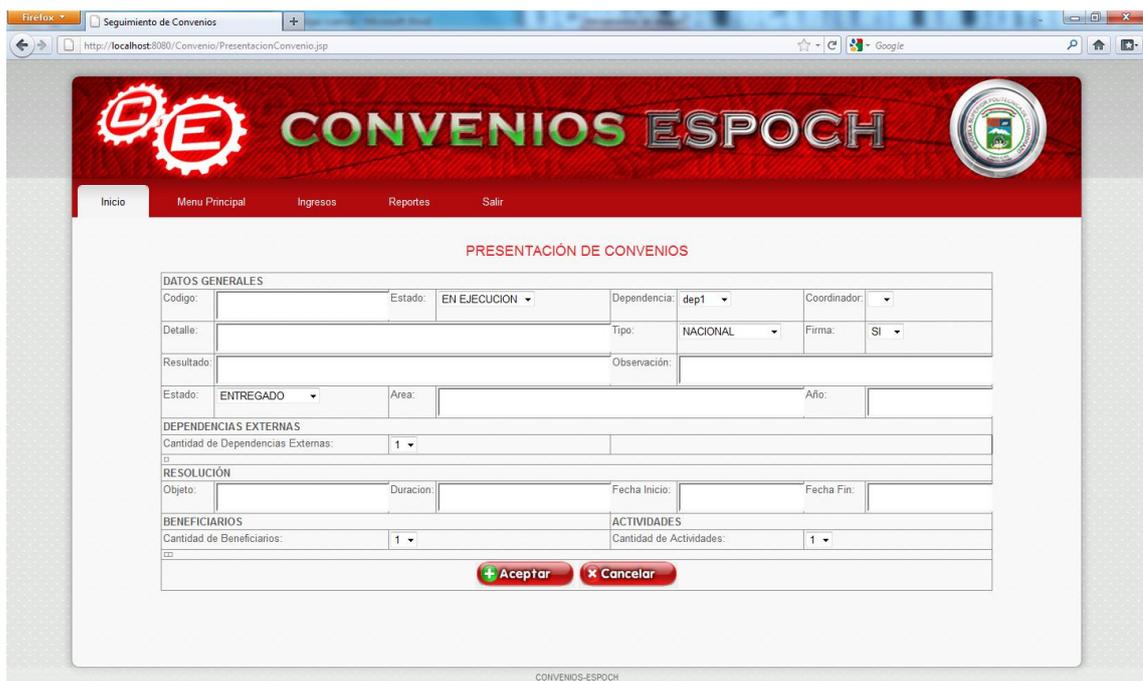


2.1 Convenio



En el **Menú Principal** tiene la particularidad de elegir varias opciones como seleccionar Registrar Nuevo convenio, cancelar convenio, dar seguimiento de un convenio

En primer lugar, podrá seleccionar la opción **Registrar Convenios** y luego proceder a llenar los campos de los datos solicitados en la pantalla.



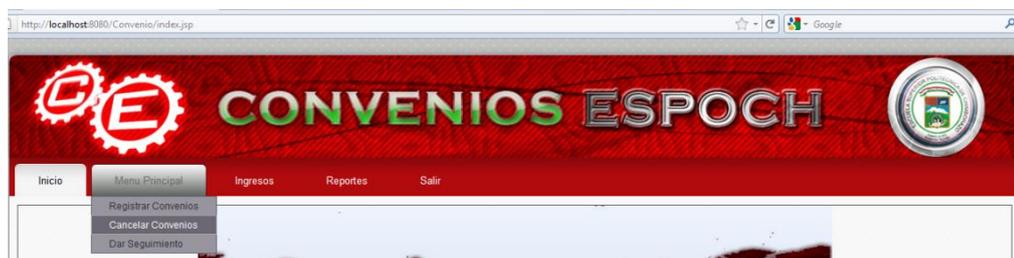
DATOS GENERALES					
Codigo:	Estado:	EN EJECUCION	Dependencia:	dep1	Coordinador:
Detalle:			Tipo:	NACIONAL	Firma:
Resultado:			Observación:		
Estado:	ENTREGADO	Area:			Año:
DEPENDENCIAS EXTERNAS					
Cantidad de Dependencias Externas: 1					
RESOLUCIÓN					
Objeto:	Duracion:	Fecha Inicio:	Fecha Fin:		
BENEFICIARIOS			ACTIVIDADES		
Cantidad de Beneficiarios: 1			Cantidad de Actividades: 1		

El código del convenio se debe ingresar de acuerdo al año en curso y numero respectivo de convenio, Ejemplo: Año 2012, convenio #1 = 12.01

PRESENTACIÓN DE CONVENIOS

DATOS GENERALES				
Código:	Estado: EN EJECUCION	Dependencia: dep1	Coordinador:	
Detalle:		Tipo: NACIONAL	Firma: SI	
Resultado:	Observación:			
Estado: ENTREGADO	Área:		Año:	
DEPENDENCIAS EXTERNAS				
Cantidad de Dependencias Externas:	1			
RESOLUCIÓN				
Objeto:	Duración:	Fecha Inicio:	Fecha Fin:	
BENEFICIARIOS			ACTIVIDADES	
Cantidad de Beneficiarios:	1	Cantidad de Actividades:	1	
<input type="button" value="+ Aceptar"/> <input type="button" value="x Cancelar"/>				

El título y objetivos son campos de ingreso textual y los demás campos de selección; dar clic sobre el campo que desee agregar información y seleccionar de la lista que muestra, caso contrario si no existe datos para seleccionar se muestra un mensaje de advertencia para que ingrese las entidades, coordinadores antes de ingresar un convenio, para lo cual se detalla posteriormente.



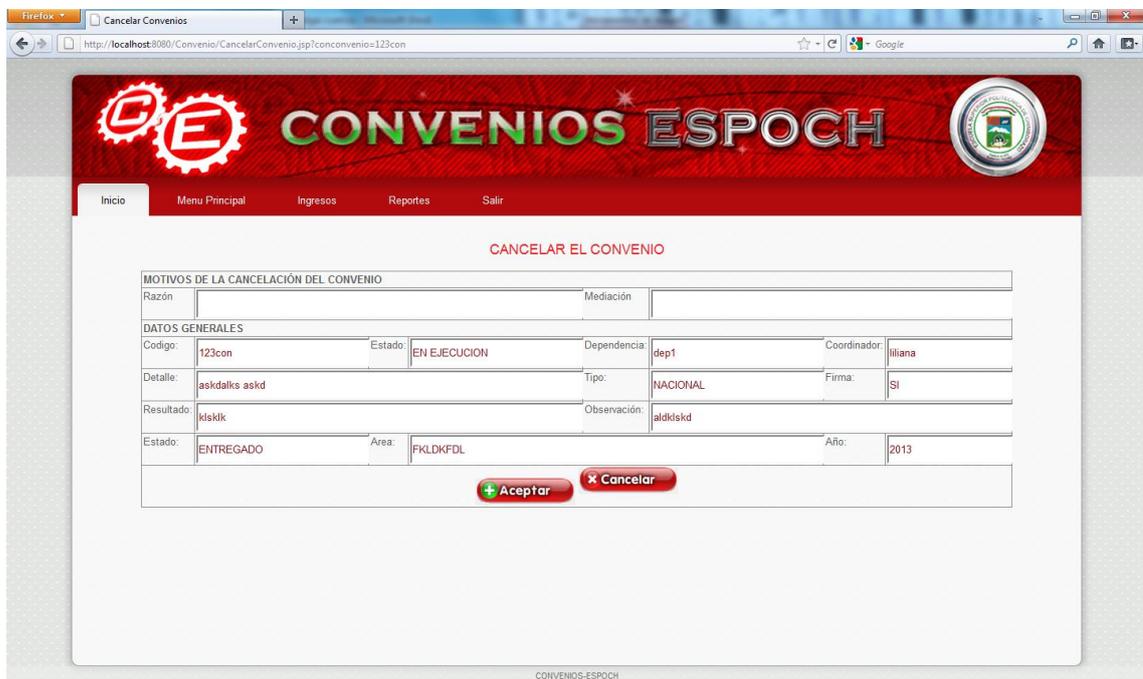
Para la cancelación de un determinado convenio nos dirigimos al menú principal opción **Cancelar Convenios**, nos muestra lo siguiente.



Se puede realizar la búsqueda de un determinado convenio por código o detalle, al seleccionar por código se muestra



Se ha realizado la búsqueda respectiva de un convenio por su código mostrándonos la descripción, observación y área a la que pertenece el convenio; pudiendo seleccionarlo para poder cancelar el convenio si el caso lo requiere.



A continuación procedemos a dar seguimiento



de un convenio determinado

Para lo cual se procede a buscar por código o detalle el convenio



Una vez buscado procedemos a seleccionar el convenio a dar seguimiento.

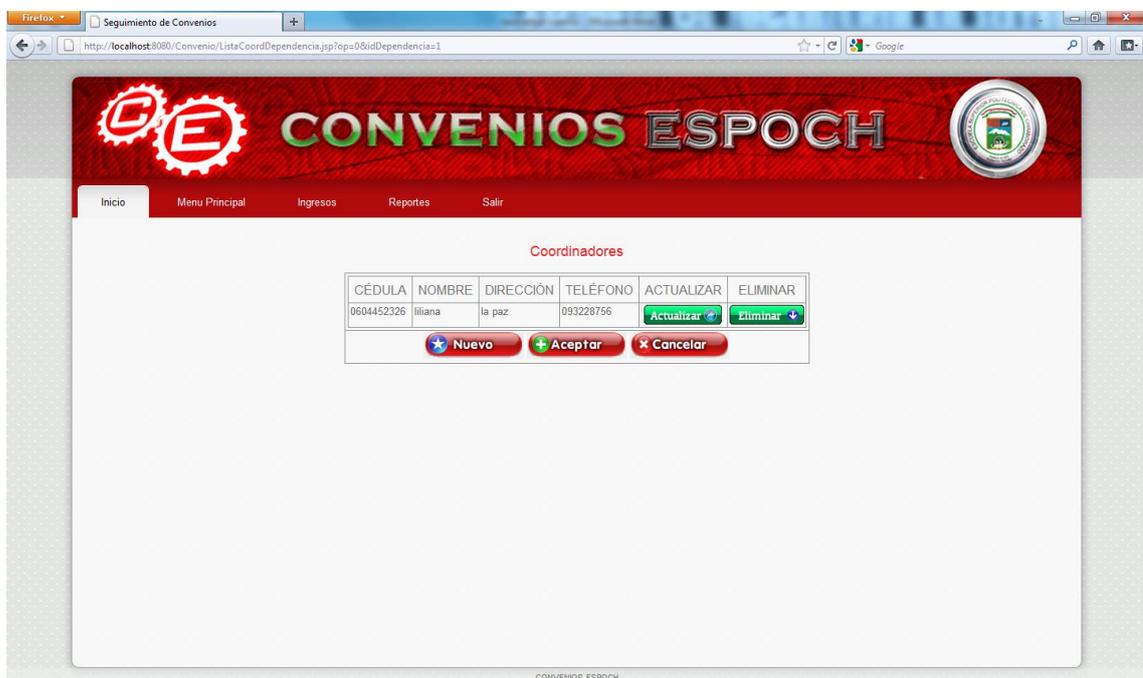


El mismo caso se para las demás opciones como en la lista de actividades, beneficios y beneficiarios del convenio y el área en la que se encuentra. Sin embargo, sino existe una lista de datos al momento de cargar el usuario debe ingresar los datos desde el menú como se detalla más adelante.



El siguiente campo muestra una pantalla para el ingreso de las obligaciones de las entidades que participan, el usuario debe seleccionar la entidad y escribir su responsabilidad en el espacio en blanco.

Finalmente cuando el convenio este con sus datos completos el usuario procede a guardar.



Al seleccionar convenio renovado se muestra la siguiente pantalla, en el cual debe llenar solo campo requerido previo a una búsqueda de convenios finalizados, se debe registrar los nuevos coordinadores o mantener los mismos al nuevo convenio

Convenio Cancelado

Finalmente al elegir Convenio Cancelado indica la pantalla en el cual debe selecciona el año de convenios y se carga todos los convenios existentes en donde se puede elegir dicho convenio a cancelar y llenar los datos solicitados. Si uno de los campos no contiene se muestra un mensaje de advertencia.

The screenshot shows a web browser window with the URL `localhost:8084/Convenio/CancelarConvenio.jsp?convenio=123con`. The page header features the 'CONVENIOS ESPOCH' logo and a navigation menu with 'Inicio', 'Menu Principal', 'Ingresos', and 'Reportes'. The main content area is titled 'CANCELAR EL CONVENIO' and contains a form with the following sections:

- MOTIVOS DE LA CANCELACIÓN DEL CONVENIO:** Includes a 'Razón' field and a 'Mediación' field.
- DATOS GENERALES:** A grid of fields including 'Codigo' (123con), 'Estado' (EN EJECUCION), 'Dependencia' (dep1), 'Coordinador' (Miana), 'Detalle' (askdalks askd), 'Tipo' (NACIONAL), 'Firma' (SI), 'Resultado' (askd), and 'Observación' (askd).
- Estado:** Includes 'Estado' (ENTREGADO), 'Area' (FKLDKFDL), and 'Año' (2013).

The footer of the page reads 'CONVENIOS-ESPOCH'.

2.3.2 Entidad

En esta opción el sistema permite guardarlos datos de una nuevaentidad mediante el ingreso de los datos solicitados en la pantalla y luego guardar.

The screenshot shows a web browser window with the URL `http://localhost:8080/Convenio/NuevaDependencia.jsp`. The page header features the 'CONVENIOS ESPOCH' logo and a navigation menu with 'Inicio', 'Menu Principal', 'Ingresos', 'Reportes', and 'Salir'. The main content area is titled 'DATOS DE NUEVA DEPENDENCIA' and contains a form with the following fields:

- Nombre:** A text input field.
- Alternativo:** A text input field.
- Sector:** A text input field.

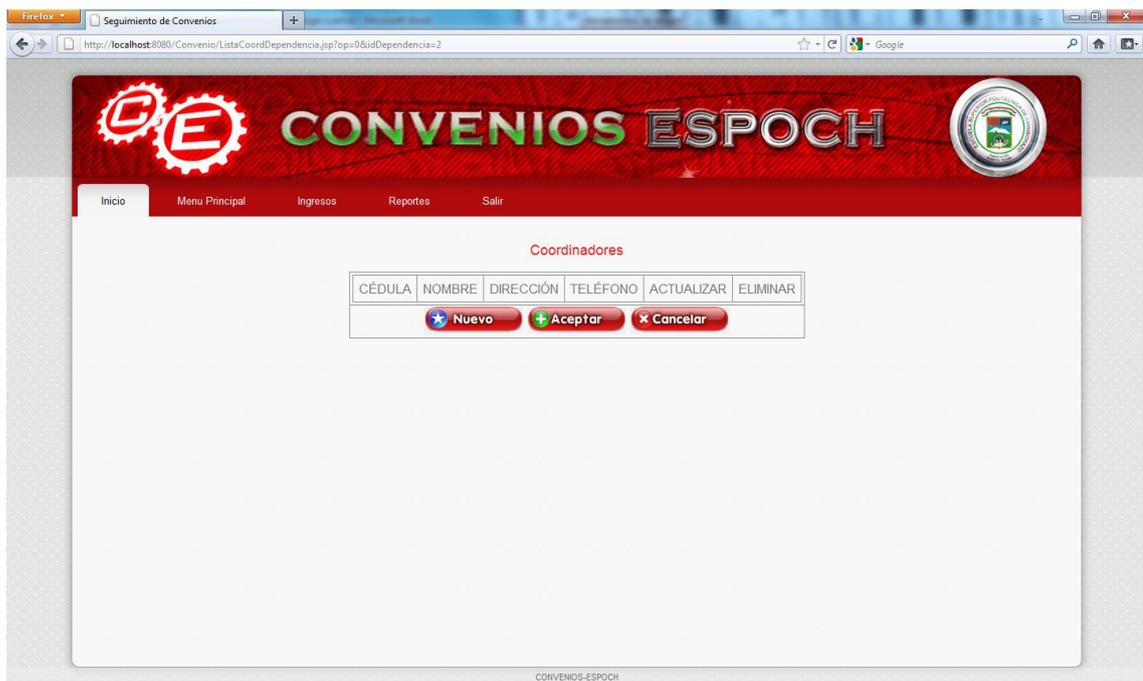
At the bottom of the form are two buttons: 'Aceptar' (with a green plus icon) and 'Cancelar' (with a red X icon). The footer of the page reads 'CONVENIOS-ESPOCH'.

Mensajes de Control

Pregunta si el usuario quiere guardar datos si selecciona SI=muestra Datos Guardados, si selecciona NO=sale, si ya existe una entidad con el mismo nombre muestra mensaje de datos duplicados.

2.3.3 Coordinador

En esta opción el sistema permite ingresar los datos del coordinador que fue designado para un convenio determinado,debiendo llenar los datos personales más importantes.



2.3.4 Beneficio

En la opción de Beneficio muestra una pantalla de datos descriptivos que deben ingresar de cada beneficio que se obtiene con el convenio.

2.3.5 Beneficiario

La siguiente pantalla muestra los campos que debe ingresar de los datos de uno o varios beneficiarios y luego guardar.

DEPENDENCIAS EXTERNAS					
Cantidad de Dependencias Externas: 5					
Nombre:	Porcentaje:	Valor:	Nombre:	Porcentaje:	Valor:
...
Nombre:	Porcentaje:	Valor:	Nombre:	Porcentaje:	Valor:
...
Nombre:	Porcentaje:	Valor:			
...			

2.3.6 Actividad

Este componente consta de un campo descriptivo que detalla la actividad de un convenio, logrando guardar una a una para formar una lista que luego se hace uso de la misma en el convenio.

BENEFICIARIOS		ACTIVIDADES	
Cantidad de Beneficiarios:	1	Cantidad de Actividades:	4
Planificadas:	Cumplidas:		
...	...		
Planificadas:	Cumplidas:		
...	...		
Planificadas:	Cumplidas:		
...	...		
Planificadas:	Cumplidas:		
...	...		

2.3.7 Rendición de Cuentas

En la opción de rendición de cuentas se detalla el seguimiento del convenio por medio de evidencias recopiladas durante el proceso del convenio, en primer lugar seleccionar cargar convenios para optimizar tiempo y espacio, al seleccionar el año del convenio inmediatamente se despliega la lista de convenios de ese año; luego se procede a la descripción general de la rendición de dicho convenio.

localhost:8084/Convenio/PresentacionConvenio.jsp

RESOLUCIÓN			
Objeto:	Duracion:	Fecha Inicio:	Fecha Fin:
BENEFICIARIOS		ACTIVIDADES	
Cantidad de Beneficiarios:	4	Cantidad de Actividades:	1
Grupo:	Numero:		
Beneficio			
Grupo:	Numero:		
Beneficio			
Grupo:	Numero:		
Beneficio			
Grupo:	Numero:		
Beneficio			

2.3.8 Informe Recibido

En la opción de Informe se despliega dos sub opciones que indican el primero Informe Recibido, el cual por cada convenio seleccionado por año respectivamente se va a ingresar los datos más relevantes del convenio como son: título del informe, el tipo puede ser de avance o final, fecha y en caso de no ser evaluado seleccionar el botón para su evaluación y luego guardar.

Pantalla de Evaluación solicita el ingreso descriptivo de cómo se desarrolla el convenio con una respectiva calificación.

2.3.9 Reporte Emitido

La tercera opción de informe muestra la pantalla de una lista de reportes necesarios de los convenios que se han registrado dentro del sistema reporte emitido en forma de informe para dicho solicitante debe ingresar que información es entregada de un determinado convenio.

Como se describe antes primero cargar los convenios, seleccionar el año y el convenio requerido, luego ingresar nombre del solicitante y una descripción de lo informado.

2.3.10 Usuario

En la siguiente pantalla muestra los campos a llenar del usuario que hará uso del sistema, debe ingresar sus datos personales para mantener un registro en la base de datos, con el objetivo de que dicho usuario quiera autenticarse.

2.4 MENÚ MODIFICACIÓN

En el menú modificación muestran los componentes que el usuario puede modificarlo iniciando por el convenio; el usuario selecciona la búsqueda, por convenios vigentes o todos, el año y el título de dicho convenio a modificar; luego actualiza los campos de escritura y para campos de selección debe habilitar la casilla de Editar Datos.

NOTA: Todos los componentes del menú modificación llevan el mismo proceso a la hora de actualizar los datos requeridos, iniciando por una búsqueda y posteriormente guarda los datos.

2.5 MENÚ REPORTE

Al seleccionar el menú reporte se despliega todos los posibles reportes que se pueden generar del sistema CONVENIOS-ESPOCH, en cual el usuario puede seleccionar uno de ellos y generar el reporte solicitado.



2.5.1 Reporte de Convenios por Año

Primero debe marcar la casilla de Cargar Convenios, luego seleccionar el año y el presionar reporte, finalmente después de unos segundos el documento será mostrado en pantalla.

REPORTE CONVENIOS-ESPOCH

Lista General Convenios

marzo 22 enero 2013

Código	Convenio	Fecha	Duración	Coordinador
9000	9000	2009-12-12	3	Rosa Saeteros
90001	9000	2009-12-12	3	Rosa Saeteros
2012009	MI B R & R 1	2009-12-12	3	Rosa Saeteros
0001	al.faldu al.fal	2009-12-12	3	Rosa Saeteros
0002	al.faldu al.fal	2009-12-12	3	Rosa Saeteros
0002104	al.faldu al.fal	2009-12-12	3	Rosa Saeteros
121000	al.faldu al.fal	2009-12-12	3	Rosa Saeteros
1210002	al.faldu al.fal	2009-12-12	3	Rosa Saeteros
121000310	al.faldu al.fal	2009-12-12	3	Rosa Saeteros
99510	jh jh jh	2009-12-12	3	Rosa Saeteros
99516	jh jh jh	2009-12-12	3	Rosa Saeteros
99516	jh jh jh	2009-12-12	3	Rosa Saeteros

marzo 22 enero 2013 Página 1 de 416

2.5 MENÚ ADMINISTRAR

En este menú se muestran opciones que solo el administrador puede realizar como por ejemplo buscar un usuario, realizar una copia de seguridad y obtener una ayuda, reporte dependencia

REPORTE CONVENIOS-ESPOCH

Cordinadores por Año

marzo 22 enero 2013

Año	Nombre	Cedula
2012	Rosa Saeteros	0603256556
	Pablo Luzariaga	0602326623
	liliana	0604452326
	Carlos Rodriguez	0603186056
	Rosa Saeteros	0603256556
	Pablo Luzariaga	0602326623
	liliana	0604452326
	Carlos Rodriguez	0603186056
	Rosa Saeteros	0603256556
	Pablo Luzariaga	0602326623
	liliana	0604452326
	Carlos Rodriguez	0603186056
2013	Rosa Saeteros	0603256556
	Pablo Luzariaga	0602326623
	liliana	0604452326
	Carlos Rodriguez	0603186056
	Rosa Saeteros	0603256556
	Pablo Luzariaga	0602326623
	liliana	0604452326
	Carlos Rodriguez	0603186056
	Rosa Saeteros	0603256556
	Pablo Luzariaga	0602326623
	liliana	0604452326
	Carlos Rodriguez	0603186056

marzo 22 enero 2013 Página 1 de 6

INDICACIONES GENERALES:

El sistema posee un método de control de errores que se ejecuta en cada una de las pantallas al momento de ingreso o modificación de los datos, en caso de presentarse dicho error o inconveniente el sistema muestra un mensaje de aviso de lo ocurrido.

AYUDA

En esta opción el botón Help el sistema nos indica una descripción sobre el uso del mismo y su navegabilidad, solo está habilitado para el usuario administrador.



Ayuda en Línea del uso del sistema con información necesaria del mismo, descripción desde el ingreso hasta la presentación de reportes.

Ayuda Online CONVENIOS-ESPOCH

ACCESO SISTEMA

Menu

- Acceso Sistema
- Principal
- Menu
- Seguimiento
- Entidad
- Reportes

Acceso Sistema



1.1. Autenticación

Para utilizar el sistema el usuario debe autenticarse mediante el ingreso de su usuario y su contraseña, caso contrario si no ingresa dichos datos muestra un mensaje de de aviso.