

*Universidad Central “Marta Abreu” de las Villas
Facultad de Matemática, Física y Computación
Departamento de Ciencia de la Computación*



*Título: Herramienta para la administración del gestor
PostgreSQL*

*Tesis Presentada en Opción al Título Académico de Máster en
Ciencia de la Computación.*

Autor: Ing. Ismary Guzmán Fernández

Tutor: Dr. Rosendo Moreno Rodríguez

Santa Clara 2011

*“La inteligencia consiste no sólo en el conocimiento,
sino también en la destreza de aplicar los conocimientos
en la práctica.”*

Aristóteles

RESUMEN

Para nuestro país obtener la soberanía tecnológica es un reto grandioso, por lo que cada día es mayor la necesidad de contar con herramientas libres que faciliten a los usuarios la aceptación del software libre. Respecto a esto, el tema de las bases de datos posee amplia documentación y herramientas, aunque las mismas en su mayoría carecen de interfaces de trabajo amigables e intuitivas y de una amplia gama de servicios integrados a las mismas. Es por ello que en el presente trabajo se aborda el diseño e implementación de una herramienta libre, que funciona sobre la Web 2.0 y administra con eficiencia el gestor de base de datos PostgreSQL. Para su elaboración se partió del estudio y caracterización de las herramientas de software libre más utilizadas para la administración de bases de datos en PostgreSQL. Además se fundamentan las tecnologías y herramientas de software libre que fueron utilizadas durante el desarrollo de la aplicación. Se describen los Casos de Uso del sistema, diagramas de clase y de componentes más significativos y se detalla el diagrama de despliegue que documenta la implantación del sistema. Finalmente y con el objetivo de facilitar la interacción de los usuarios con la herramienta, se realiza una descripción de las funcionalidades que posee a través de un esbozo del Manual de Usuario.

ABSTRACT

For our country gaining technological sovereignty is a great challenge, so every day increases the need of free tools that can facilitate users the acceptance of free software. In this regard, the issue of databases has extensive documentation and tools, although they mostly lack of friendly and intuitive interfaces and a wide range of integrated services. That is why the aim of this research work is to design and implement a free tool that works on Web 2.0 and efficiently manages the database manager PostgreSQL. In order to start with development they were studied and characterized the most widely used free software tools for managing PostgreSQL databases. In addition, they were argued the underlying technologies and free software tools that were used during the development of the application. We describe the system use cases, class diagrams and most significant components and it was detailed the deployment diagram which documents the implementation of the system. Finally, with the aim of facilitate user interaction with the tool, it was performed a description of its functions through an outline of the User Manual.

TABLA DE CONTENIDO

INTRODUCCIÓN	- 1 -
<i>CAPÍTULO I: Herramientas de administración de base de datos. Aspectos generales.</i> ..	- 6 -
1.1. Generalidades de los SGBD	- 6 -
1.1.1 Objetivos de un SGBD	- 7 -
1.1.2 Características de un SGBD	- 9 -
1.1.3 Gestor de base datos libre PostgreSQL	- 10 -
Características de PostgreSQL	- 10 -
Ventajas de PostgreSQL	- 12 -
Conceptualización de los objetos gestionados en PostgreSQL	- 13 -
1.2. Herramientas para la administración de PostgreSQL	- 17 -
1.2.1 PgAdmin III	- 17 -
1.2.2 PhpPgAdmin	- 18 -
1.2.3 Navicat.....	- 19 -
1.2.4 PgAccess.....	- 20 -
1.2.5 EMS SQL Manager Lite.....	- 20 -
1.2.6 Comparación entre las herramientas de administración analizadas	- 21 -
1.3. Ambiente de desarrollo de la aplicación	- 23 -
1.3.1 Tecnologías.....	- 24 -
1.3.1.1 Lenguaje de programación del lado del servidor	- 24 -
□ Python.....	- 24 -
1.3.1.2 Lenguaje de programación del lado del cliente.....	- 26 -
□ HTML.....	- 26 -
□ JavaScript.....	- 27 -
□ CSS.....	- 28 -
1.3.1.3 Servidores.....	- 29 -
□ Apache.....	- 29 -
1.3.2 Frameworks de desarrollo	- 29 -
□ Django	- 30 -
1.3.3 Bibliotecas	- 30 -
□ ExtJS.....	- 31 -
1.3.4 Herramientas.....	- 32 -
□ Visual Paradigm.....	- 32 -
□ Eclipse	- 33 -
1.4. Conclusiones del capítulo	- 34 -

CAPÍTULO II: Modelado de la herramienta de administración para PostgreSQL.	35 -
2.1. Arquitectura del sistema	35 -
2.2. Casos de uso del sistema	39 -
2.2.1. Descripciones de casos de uso	40 -
2.3. Diagramas de clases	47 -
2.4. Diagramas de secuencia	50 -
2.5. Diseño de la base de datos	55 -
2.6. Conclusiones del capítulo	57 -
CAPÍTULO III: Implementación y descripción de uso.....	58 -
3.1. Implementación.....	58 -
3.1.1. Diagrama de componentes.....	58 -
3.1.2. Diagrama de despliegue	60 -
3.2. Esbozo de Manual de Usuario	61 -
3.2.1. Estructura de la interfaz principal de la herramienta	62 -
□ Componentes	62 -
□ Área de trabajo	65 -
3.2.2. Interfaces de usuario de los componentes	67 -
3.3. Conclusiones del capítulo	84 -
CONCLUSIONES	85 -
RECOMENDACIONES	86 -
REFERENCIAS	87 -
BIBLIOGRAFÍA	90 -

ÍNDICE DE FIGURAS

Figura 1: Funcionamiento de un SGBD-----	7 -
Figura 2: Diagrama de integración de tecnologías. -----	38 -
Figura 3: Diagrama de casos de uso del sistema -----	39 -
Figura 4: Diagrama de clases de la herramienta-----	48 -
Figura 5: Diagrama de secuencia del caso de uso Mostrar Datos Tabla-----	51 -
Figura 6: Diagrama de secuencia del caso de uso Gestionar Datos Tabla para la sección Insertar datos -----	52 -
Figura 7: Diagrama de secuencia del caso de uso Gestionar Datos Tabla para la sección Modificar datos -----	53 -
Figura 8: Diagrama de secuencia del caso de uso Gestionar Datos Tabla para la sección Eliminar datos -----	54 -
Figura 9: Modelo físico de datos de la aplicación-----	55 -
Figura 10: Diagrama de componentes para los CU Mostrar Datos Tabla y Gestionar Datos Tabla -----	59 -
Figura 11: Diagrama de despliegue de la herramienta-----	61 -
Figura 12: Interfaz principal de la herramienta-----	62 -
Figura 13: Explorador de objetos -----	63 -
Figura 14: Clic derecho para editar o eliminar un componente-----	64 -
Figura 15: clic derecho para crear componentes -----	65 -
Figura 16: Submenú servidores -----	66 -
Figura 17: Submenú base de datos-----	66 -
Figura 18: Submenú esquemas -----	67 -
Figura 19: Submenú tablas-----	67 -
Figura 20: Interfaz para registrar un servidor-----	68 -
Figura 21: Interfaz para conectarse a un servidor -----	69 -
Figura 22: Interfaz para crear un rol (Pestaña propiedades) -----	71 -
Figura 23: Interfaz para crear un rol (Pestaña membrecía)-----	71 -
Figura 24: Interfaz para crear base de datos -----	73 -

Figura 25: Interfaz para crear esquema-----	- 74 -
Figura 26: Interfaz para crear una tabla (Pestaña propiedades)-----	- 75 -
Figura 27: Interfaz para crear una tabla (Pestaña columnas)-----	- 76 -
Figura 28: Interfaz para visualizar datos de una tabla -----	- 77 -
Figura 29: Interfaz para insertar datos en una tabla -----	- 77 -
Figura 30: Interfaz para crear un nuevo índice -----	- 79 -
Figura 31: Interfaz para crear una llave foránea (Pestaña propiedades) -----	- 81 -
Figura 32: Interfaz para crear una llave foránea (Pestaña columnas) -----	- 81 -
Figura 33: Interfaz para crear una restricción único -----	- 82 -
Figura 34: Interfaz para crear un check -----	- 83 -

ÍNDICE DE TABLAS

Tabla 1: Variables de la hipótesis de la investigación	- 3 -
Tabla 2: Comparación ente herramientas de administración libres para PostgreSQL...	- 21 -
Tabla 3: Descripción del caso de uso Mostrar Datos Tabla	- 40 -
Tabla 4: Descripción del caso de uso Gestionar Datos Tabla	- 42 -
Tabla 5: Funciones de las tablas del catálogo de PostgreSQL	- 55 -

INTRODUCCIÓN

Las tecnologías en el mundo de hoy presentan un gran auge, la informatización de la sociedad ha crecido a nivel mundial y esto trae consigo el aumento de la generación y almacenamiento de la información. Para recopilar todo tipo de datos y atender las necesidades de la informatización de empresas surge la tecnología de bases de datos, considerada una de las más antiguas dentro de la ciencia de la informática, utilizada en la actualidad para manejar grandes cantidades de datos.

Como resultado de la evolución de las bases de datos hoy en día estas son mucho más seguras, flexibles y manejables en la mayor parte de los casos. Esto es posible gracias a la creación de los Sistemas Gestores de Bases de Datos (SGBD¹); software que permite de manera clara, sencilla y ordenada manejar un conjunto de datos que posteriormente se convertirán en información para la empresa.

Los SGBD existentes en el mundo se encuentran divididos en dos grandes conjuntos, los propietarios, aquellos por lo que hay que pagar una licencia para su uso, dentro de los más reconocidos se encuentran: Oracle (*Oracle Corporation 2010*), Fox Pro (*Microsoft Corporation 2010b*) y Microsoft SQL Server(*Microsoft Corporation 2010a*), y los libres que como su nombre lo indica pueden ser usados, modificado y distribuidos libremente sin restricciones, tales como, SQLite (*SQLite 2010*) y PostgreSQL(*PostgreSQL Global Development Group 1996*).

Las empresas cubanas al igual que cualquier otra organización, necesitan de SGBD que le permitan la gestión de la información que poseen. En el ámbito empresarial cubano, el pago por los derechos de uso del software propietario implica un gasto adicional, que en algunas ocasiones supone un importante esfuerzo económico difícil de asumir. Convirtiendo al software libre en una posibilidad cada vez más atractiva para el país.

¹ En inglés DBMS (*Data Base Management System*)

El Software Libre ha evolucionado y se ha consolidado en muchas partes del mundo teniendo un gran respaldo por parte de académicos, organizaciones educativas, grandes corporaciones, empresas, desarrolladores y usuarios de software. El tema ha trascendido del aspecto técnico para llegar a ser un tema estratégico en muchas organizaciones y un tema político en algunos países. El Software Libre no es una moda, sino es, además de lo indicado, un modelo de negocio para una nueva industria de software basada en servicios, más que en productos (*Ing. Ramón Paumier Samón. et al. 2010*). Por ello, el país se ha trazado una estrategia para migrar sus soluciones informáticas al mundo del software libre, proponiéndose que las empresas cubanas empleen en sus soluciones, siempre que sea posible, tecnologías de bases de datos libres, de manera que se avance en el proceso de adquisición de la soberanía tecnológica. Un paso decisivo en aras de alcanzar este objetivo es contar con un potente SGBD libre, que pueda ser personalizado según las necesidades de la empresa que lo utilice.

PostgreSQL es un magnífico gestor de bases de datos libre. Tiene prácticamente todo lo que tienen los gestores comerciales, haciendo de él una muy buena alternativa para la administración de bases de datos. Se caracteriza por ser un gestor de base de datos estable, de alto rendimiento y gran flexibilidad, que soporta todas las características de una base de datos profesional y es altamente adaptable a las necesidades del cliente; lo que lo convierte en una atractiva opción a la hora de decidirse por el uso de un gestor de bases de datos libre.

Sin embargo los administradores de bases de datos que usan el gestor PostgreSQL muestran problemas al administrarlo, esto se debe fundamentalmente a que las herramientas libres existentes para su administración no poseen las características necesarias para un trabajo interactivo y rápido sobre el gestor. Sus interfaces de trabajo son poco amigables, carecen de servicios de migración, monitoreo y control, representación gráfica de datos espaciales y diseño de diagramas entidad - relación. Obviamente esto trae como consecuencia que el personal informático que interactúa con el gestor no se sienta interesado en él, provocando que su usabilidad disminuya, la calidad se vea reducida con

respecto a las herramientas privativas y se obstaculice la migración del país a software libre.

Con el fin de solucionar las dificultades antes planteadas surge el siguiente **problema**: La carencia en nuestro medio de herramientas libres e interactivas para la administración del gestor PostgreSQL, afectan su uso y obstaculizan la migración de las aplicaciones cubanas a software libre.

Como **objetivo general** de la presente investigación se plantea: elaborar una herramienta con interfaces gráficas interactivas, sobre la Web 2.0, que facilite a los usuarios la administración del gestor PostgreSQL.

Trazándose para su cumplimiento los siguientes **objetivos específicos**:

1. Definir tecnologías de software libre apropiadas para el desarrollo de una herramienta de administración para PostgreSQL.
2. Caracterizar múltiples herramientas libres de administración para PostgreSQL.
3. Desarrollar el análisis y diseño de la herramienta con la ayuda del UML.
4. Implementar una herramienta de administración para PostgreSQL sobre la Web 2.0.

A raíz de la especificación del problema, se plantea la siguiente **hipótesis**: si se implementara una herramienta Web libre para la administración de PostgreSQL, se logrará una mayor aceptación del gestor por parte de los usuarios y se mejorará la usabilidad del producto.

Tabla 1: Variables de la hipótesis de la investigación

<i>Independientes</i>	<i>Dependientes</i>
Herramienta Web libre para la administración de PostgreSQL	Aceptación de PostgreSQL
	Usabilidad de PostgreSQL

El **tipo de investigación** utilizada es la descriptiva y los **métodos** empleados en el desarrollo de la misma son:

- El análisis histórico lógico y el analítico sintético de los “*teóricos*”
- La observación y la entrevista de los “*empíricos*”

Justificación de la investigación

Hoy en día las tecnologías libres han tomado fuerza y se han convertido en una buena opción, por lo que el país ha decidido su total migración a este mundo. Para lograr este paso tan significativo en las empresas cubanas es necesario contar con herramientas interactivas que les faciliten a los usuarios el trabajo y les permitan una mayor aceptación al software libre.

El gestor de base datos PostgreSQL es una de las tantas tecnologías libres que necesitan la confección de herramientas nuevas que proporcionen a los usuarios interfaces agradables, sencillas y completas, posibilitando con ello una mayor aceptación y usabilidad del mismo.

Una vez que PostgreSQL cuente con una herramienta Web para su administración, que contenga interfaces de usuarios sugerentes, intuitivos y fáciles de usar; los usuarios se sentirán más identificados con el gestor y no necesitarán tener grandes conocimientos de bases de datos para su manejo.

Otro aspecto a tener en cuenta es el considerable ahorro que se obtiene con la creación de la herramienta, pues todas las tecnologías a utilizar en su desarrollo son libres, por lo que no hay que invertir dinero en pago de licencias de software.

Aportes de la investigación

El ***valor práctico*** consiste en la confección de una herramienta libre, sobre la Web 2.0, que posibilite a los usuarios administrar de forma sencilla e interactiva el gestor PostgreSQL, permitiendo una mayor aceptación del mismo e incrementando su uso en las aplicaciones de nuestro país.

El ***valor metodológico*** consiste en la integración de varias tecnologías libres como la biblioteca de componentes gráficos ExtJS y el framework de desarrollo Web Django, a

través de la confección de una plantilla HTML donde se incluyen los ficheros JavaScript que contienen la implementación de las interfaces de usuario, siendo la misma publicada a través de una vista en Django. Además, para cargar los datos de las interfaces se crean vistas en Django, hacia donde serán enviadas las consultas AJAX de la biblioteca ExtJS en formato JSON.

Otros aportes fueron la creación en ExtJS de componentes dinámicos como el gridPanel donde las columnas del grid son variables dependiendo del número de columnas que contengan las tablas a mostrar del gestor, y la posibilidad de simular desde la biblioteca ExtJS aplicaciones de escritorio sobre la Web con paneles ,árboles, ventanas, formularios y grid.

El documento está estructurado en tres capítulos que se describen a continuación:

En el **Capítulo I “Herramientas de administración de base de datos. Aspectos generales”** se detallan las principales herramientas utilizadas para la administración del gestor de base de datos PostgreSQL, especificando en cada una, ventajas y desventajas y dejando clara la posición del autor. Además se fundamenta la elección de las tecnologías escogidas para la implementación de la herramienta de administración.

En el **Capítulo II “Modelado de la herramienta de administración para PostgreSQL”** se especifica la arquitectura de la herramienta y el modelado del sistema, apoyándose para ello en los diagramas de casos de uso del sistema, clases, secuencia y modelo físico de la base de datos.

En el **Capítulo III “Implementación y Manual de Usuario”** se representa a través de los diagramas de componentes y despliegue la implementación del sistema y se especifican las principales funcionalidades de la herramienta a través un esbozo de Manual de Usuario, que describe el uso de esta herramienta.

Seguidamente se presentan las conclusiones y las recomendaciones del trabajo, así como las referencias y bibliografía utilizadas.

CAPÍTULO I: Herramientas de administración de base de datos. Aspectos generales.

Las bases de datos, hoy en día, ocupan un lugar determinante en cualquier área del quehacer humano, comercial, y tecnológico. Su administración es esencial para que una empresa o institución mantenga actualizada la información recopilada durante sus procesos laborales. Es por ello, que las herramientas gráficas de administración de bases de datos juegan un rol muy importante en el ámbito empresarial, pues permiten a los usuarios gestionar las bases de datos, manteniéndolas actualizadas en todo momento.

En este Capítulo se exponen los conceptos fundamentales de los SGBD, así como, las características y ventajas del gestor PostgreSQL. Se describen múltiples herramientas libres de administración de base de datos caracterizando cada una de ellas y comparando su deficiencia y ventajas en la administración de base de datos, dejando clara la posición del autor. Además se describe y explica la selección de las tecnologías libres escogidas para la implementación de la herramienta de administración a desarrollar.

1.1. Generalidades de los SGBD

Es el software que permite la utilización y/o la actualización de los datos almacenados en una (o varias) base(s) de datos por uno o varios usuarios desde diferentes puntos de vista y a la vez, se denomina sistema de gestión de bases de datos (SGBD). (*Indira Martínez Bravo 1999*)

Un SGBD no es más que: un conjunto coordinado de programas, procedimientos, lenguajes, etc. que suministra, tanto a usuarios no informáticos como a los analistas, programadores o al administrador, los medios necesarios para describir, recuperar y manipular los datos almacenados en la base, manteniendo su integridad, confidencialidad y seguridad. (*José Miguel Rubio L. 2011*)

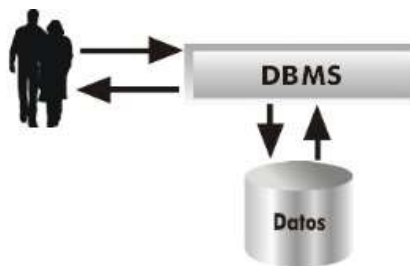
Herramientas de administración de base de datos

Figura 1: Funcionamiento de un SGBD

1.1.1 Objetivos de un SGBD

- Definir la Base de Datos mediante el Lenguaje de Definición de Datos, el cual permite especificar la estructura, tipo de datos y las restricciones sobre los datos, almacenándolo todo en la base de datos.
- Separar la descripción y manipulación de la data, permitiendo un mayor entendimiento de los objetos, además de flexibilidad de consulta y actualización de los datos.
- Permitir la inserción, eliminación, actualización, consulta de los datos mediante el Lenguaje de Manejo de Datos. Existen dos tipos de programas de Manejo de Datos, los cuales se diferencian por la forma en que acceden a los datos.
 - ✓ *Lenguajes procedurales:* manipulan la base de datos registro a registro y se deben especificar las operaciones a realizar para obtener los datos resultados.
 - ✓ *Lenguajes no procedurales:* manipulan la base de datos en conjuntos de registros y se especifican qué datos deben obtenerse como resultado sin plantear la forma de hacerlo. El lenguaje no procedural más utilizado es SQL² que se ha convertido en un estándar y el lenguaje por defecto de los SGBD relacionales.

² En inglés SQL (*Structure Query Language*)

Herramientas de administración de base de datos

- Proporcionar acceso controlado a la base de datos.
 - ✓ *Seguridad:* los usuarios no autorizados no pueden acceder a la base de datos.
 - ✓ *Integridad:* mantiene la integridad y consistencia de la base de datos.
 - ✓ *Control de Recurrencia:* permite el acceso compartido a la base de datos.
 - ✓ *Control de Recuperación:* restablece la base de datos después de producirse un fallo de software o hardware.
 - ✓ *Diccionario de datos o Catálogo:* contiene la descripción de los datos de la base de datos y es accesible por el usuario.
- Gestionar la estructura física de los datos y su almacenamiento, proporcionando eficiencia en las operaciones de la base de datos y el acceso al medio de almacenamiento.
- Proporcionar un mecanismo de vistas, que permita a cada usuario tener su propia vista o visión de la base de datos. El lenguaje de definición nos permite definir las vistas como subconjuntos de la base de datos, permitiendo:
 - ✓ Proporcionar un nivel de seguridad excluyendo datos para que no sean vistos por determinados usuarios.
 - ✓ Permiten que los usuarios vean los datos en el formato deseado.
 - ✓ Una vista representa una imagen consistente y permanente de la base de datos, aún cuando a la base de datos se le hagan cambios en sus estructuras.
- Eliminar la redundancia de datos, establecer una mínima duplicidad en los datos y minimizar el espacio en disco utilizado.

Herramientas de administración de base de datos

- Proveer interfaces procedimentales y no procedimentales, permitiendo la manipulación por usuarios interactivos y programadores.
- Independizar la estructura de la organización lógica de los datos (Independencia física).
- Independizar la descripción lógica de la Base de datos y las descripciones particulares de los diferentes puntos de vistas de los usuarios.
- Permitir una fácil administración de los datos. (*Fidel Gil et al. 2005*)

1.1.2 Características de un SGBD

- *Abstracción de la información:* Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. Este hecho se hace transparente al usuario. Así, se definen varios niveles de abstracción.
- *Independencia:* La independencia de los datos consiste en la capacidad de modificar el esquema (físico o lógico) de una base de datos sin tener que realizar cambios en las aplicaciones que se sirven de ella.
- *Redundancia mínima:* Un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. De entrada, lo ideal es lograr una redundancia nula; no obstante, en algunos casos la complejidad de los cálculos hace necesaria la aparición de redundancias.
- *Consistencia:* En aquellos casos en los que no se ha logrado esta redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente, es decir, que todos los datos repetidos se actualicen de forma simultánea.
- *Seguridad:* La información almacenada en una base de datos puede llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentra segura frente a usuarios malintencionados, que intenten leer

Herramientas de administración de base de datos

información privilegiada; frente a ataques que deseen manipular o destruir la información; o simplemente ante las torpezas de algún usuario autorizado pero despistado. Normalmente, los SGBD disponen de un complejo sistema de permisos a usuarios y grupos de usuarios, que permiten otorgar diversas categorías.

- *Integridad:* Se trata de adoptar las medidas necesarias para garantizar la validez de los datos almacenados. Es decir, se trata de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados, o cualquier otra circunstancia capaz de corromper la información almacenada.
- *Respaldo y recuperación:* Los SGBD deben proporcionar una forma eficiente de realizar copias de respaldo de la información almacenada en ellos, y de restaurar a partir de estas copias los datos que se hayan podido perder.
- *Control de la concurrencia:* En la mayoría de entornos, lo más habitual es que sean muchas las personas que acceden a una base de datos, bien para recuperar información, bien para almacenarla. Y es también frecuente que dichos accesos se realicen de forma simultánea. Un SGBD debe controlar el acceso concurrente a la información, que podría derivar en inconsistencias.

1.1.3 Gestor de base datos libre PostgreSQL

PostgreSQL es un Sistema de Gestión de Bases de Datos objeto-relacional, distribuido bajo licencia BSD y con su código fuente disponible libremente. Es el sistema de gestión de bases de datos de código abierto más potente del mercado y en sus últimas versiones no tiene nada que envidiarle a otras bases de datos comerciales. (*Rafael Martínez 2009*)

Características de PostgreSQL

PostgreSQL es un gestor de base de datos excepcional, sus características técnicas la hacen una de las bases de datos más potentes y robustas del mercado. Su desarrollo comenzó hace más de 15 años, y durante este tiempo, estabilidad, potencia, robustez, facilidad de administración e implementación de estándares han sido las características que más se han

Herramientas de administración de base de datos

tenido en cuenta durante su desarrollo. PostgreSQL funciona muy bien con grandes cantidades de datos y una alta concurrencia de usuarios accediendo a la vez al sistema. (Rafael Martínez 2009)

A continuación se describen las principales características del gestor PostgreSQL:

- *Soporte SQL92/SQL99:* implementa un subconjunto extendido de los estándares SQL92 y SQL99
- *Transacciones:* Permiten el paso entre dos estados consistentes manteniendo la integridad de los datos.
- *Integridad referencial:* soporta integridad referencial, la cual es utilizada para garantizar la validez de los datos de la base de datos. Llaves primarias *PK* y llaves foráneas o extranjeras *FK*.
- *Bloqueos de tabla y filas:* ofrece varios modos de bloqueo para controlar el acceso concurrente a los datos en tablas.

Algunos de estos modos de bloqueo los adquiere PostgreSQL automáticamente antes de la ejecución de una declaración, mientras que otros son proporcionados para ser usados por las aplicaciones.

- *Constraints y triggers:* tienen la función de mantener la integridad y consistencia en la BD. Consisten en la ejecución de acciones antes o después de un evento de BD.
- *Múltiples tipos de datos predefinidos:* como todos los manejadores de bases de datos, PostgreSQL implementa los tipos de datos definidos para el estándar SQL3 y aumenta algunos otros.
- *Conectividad TCP/IP, JDBC y ODBC.*

Herramientas de administración de base de datos

- *Interfaz con diversos lenguajes:* C, C++, Java, Delphi, Python, Perl, PHP, Bash, etc.

Además de todas las características antes mencionadas PostgreSQL en sus últimas versiones presenta:

- *Savepoints:* permite hacer un *roll back* sin tener que repetir la transacción entera.
- *Point in Time Recovery:* permite salvar el estado de la BD en momentos concretos, para su posterior recuperación.
- *Tablespaces:* permite destinar discos físicos a un índice o a una tabla concreta.
- *Improved Memory and I/O:* optimizaciones en la velocidad de ejecución y en el consumo de memoria de la aplicación.
- *Añadidos o ampliados los lenguajes de procedimientos almacenado:* PL/Java, PL/J, PL/PHP y PL/Perl.
- *Instalación:* sobre Windows (2000,2003 y XP).

Ventajas de PostgreSQL

- *Instalación ilimitada:* nadie puede demandarlo por violar acuerdos de licencia, puesto que no hay costo asociado a la licencia del *software*.
- *Soporte:* cuenta con una importante comunidad de profesionales y entusiastas de los que las compañías pueden obtener beneficios y contribuir.
- *Ahorros considerables en costos de operación:* ha sido diseñado y creado para tener un mantenimiento y ajuste mucho menor que otros productos, conservando todas las características, estabilidad y rendimiento.

Herramientas de administración de base de datos

- *Estabilidad y confiabilidad legendarias:* es extremadamente común que compañías reporten que PostgreSQL nunca ha presentado caídas en varios años de operación de alta actividad.
- *Extensible:* el código fuente está disponible para todos sin costo. Si su equipo necesita extender o personalizar PostgreSQL de alguna manera, pueden hacerlo con un mínimo esfuerzo, sin costos adicionales.

Esto es complementado por la comunidad de profesionales y entusiastas de PostgreSQL alrededor del mundo que también extienden PostgreSQL todos los días.

- *Multiplataforma:* está disponible en casi cualquier Unix y para Windows.
- *Diseñado para ambientes de alto volumen:* usa una estrategia de almacenamiento de filas llamada MVCC para conseguir una mejor respuesta en ambientes de grandes volúmenes.

Los principales proveedores de sistemas de bases de datos comerciales usan también esta tecnología, por las mismas razones.

- *Herramientas gráficas de administración de BD:* cuenta con varias herramientas gráficas de alta calidad para la administración de bases de datos.

Conceptualización de los objetos gestionados en PostgreSQL

Las bases de datos en PostgreSQL no son archivos que se puedan subir a su sitio Web como los de Access, sino que residen en un servidor de datos separado. Por ello se debe utilizar algún programa que le permita al gestor administrar sus bases de datos satisfactoriamente.

Herramientas de administración de base de datos

La administración de bases de datos se realiza con un sistema de administración llamado DBMS³. Un DBMS no es más que 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).

PostgreSQL al igual que otros gestores para su administración está compuesto por múltiples objetos de la base de datos que a continuación se detallan:

- *Servidores*: instancia del servidor PostgreSQL que se desea administrar. Este objeto contiene el registro del servidor a gestionar.
- *Bases de datos*: conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.
- *Esquemas*: representa el conjunto de objetos que posee una base de datos. Es la descripción lógica de la base de datos, proporciona los nombres de las tablas y sus atributos especificando las relaciones que existen entre ellos.
- *Tablas*: conjunto de registros con características similares donde se guardan los datos recogidos por un programa.

Las tablas se componen de dos estructuras:

- ✓ Registro: es cada una de las filas en que se divide la tabla. Cada registro contiene datos de los mismos tipos que los demás registros.

³ Sistema de administración de base de datos, en inglés DBMS (Database management system)

Herramientas de administración de base de datos

- ✓ Campo o columnas: es cada una de las columnas que forman la tabla. Contienen datos de tipo diferente a los de otros campos.

A los campos se les pueden asignar, además, propiedades especiales que afectan a los registros insertados. Un campo puede ser definido como índice o auto incrementable, lo cual permite que los datos de ese campo cambien solos o sea el principal indicador a la hora de ordenar los datos contenidos.

- *Restricciones*: son las encargadas de asegurar la integridad referencial en una base de datos.

Tipos de restricciones:

- ✓ Check: son usados para asegurar reglas simples de negocio sobre el contenido de los datos en las tablas.
 - ✓ Único: protege a una o más columnas de una tabla, asegurando que no existan dos filas que contengan información duplicada en las columnas aseguradas.
 - ✓ Llave única o primaria: conjunto de una o más columnas de una tabla, que tomadas colectivamente nos permiten identificar un registro como único, es decir, en una tabla podemos saber cual es un registro en específico sólo con conocer la llave primaria.
 - ✓ Llave foránea o extranjera: protege una o más columnas de una tabla, asegurando que cada valor de la fila es, o bien nulo en su conjunto, o bien apunta a un valor de una llave única o primaria.
- *Dominio*: es el conjunto de valores que pueden tomar los atributos en una relación.

Herramientas de administración de base de datos

- *Índices:* estructura de datos que mejora la velocidad de las operaciones, permitiendo un rápido acceso a los registros de una tabla en una base de datos.
- *Reglas:* son estructuras que permiten que una acción alternativa sea realizadas al actualizar, insertar o eliminar datos en una tabla. También son utilizadas para implementar vistas de tablas.
- *Secuencias:* son estructuras que se emplea para generar valores enteros secuenciales únicos y asignárselos a campos numéricos; se utilizan generalmente para las claves primarias de las tablas garantizando que sus valores no se repitan.
- *Disparadores:* procedimiento que se ejecuta cuando se cumple una condición establecida al realizar una operación. Dependiendo de la base de datos, los triggers pueden ser de inserción (*INSERT*), actualización (*UPDATE*) o borrado (*DELETE*). Algunas bases de datos pueden ejecutar triggers al crear, borrar o editar usuarios, tablas, bases de datos u otros objetos.
- *Usuarios:* conjunto de usuarios creados en el servidor de base de datos, los cuales en dependencia de sus privilegios pueden realizar múltiples tareas sobre el servidor.
- *Vistas:* Es una relación virtual, que se construye a partir de tablas o incluso otras vistas, formada por atributos de estas otras tablas de forma directa o como resultado de una consulta.
- *Funciones o procedimientos:* se puede definir como un programa, procedimiento ó función, el cual está almacenado en la base de datos y listo para ser usado.

Herramientas de administración de base de datos

- *Operadores*: son símbolos que permiten realizar distintos tipos de operaciones dentro de la base de datos. PostgreSQL contiene múltiples operadores (relacionales o de comparación, lógicos, aritméticos y de concatenación), aunque también da la posibilidad de crear operadores nuevos en caso de ser necesario.

1.2. Herramientas para la administración de PostgreSQL

La administración de una base de datos opera dentro de un marco proporcionado por la administración de datos, facilitándose de esta manera el desarrollo y el uso de una base de datos y sus aplicaciones.

PostgreSQL posee múltiples herramientas gráficas para la administración de bases de datos, aunque estas en su mayoría presentan deficiencia, provocando cierto rechazo al gestor por parte de los usuarios. Como consecuencia de ello, se decidió realizar un análisis detallado de algunas de estas herramientas, con el fin de seleccionar sus características y desventajas más significativas y tenerlas en cuenta a la hora de elaborar la aplicación propuesta.

1.2.1 PgAdmin III

PgAdmin III (*pgAdmin group 2011*) es una aplicación gráfica para gestionar el gestor de bases de datos PostgreSQL, siendo la más completa y popular con licencia *Open Source*. Está escrita en C++ usando la librería gráfica multiplataforma *wxWidgets*, lo que permite que se pueda usar en Linux, FreeBSD, Solaris, Mac OS X y Windows. Es capaz de gestionar versiones a partir de la PostgreSQL 7.3 ejecutándose en cualquier plataforma, así como versiones comerciales de PostgreSQL como Pervasive Postgres, EnterpriseDB, Mammoth Replicator y SRA PowerGres. (*Comunidad de Desarrollo Ubuntu 2008*)

PgAdmin III está diseñado para responder a las necesidades de todos los usuarios, desde escribir consultas SQL simples hasta desarrollar bases de datos complejas. El interfaz gráfico soporta todas las características de PostgreSQL y facilita enormemente la administración. La aplicación también incluye un editor SQL con resaltado de sintaxis, un

Herramientas de administración de base de datos

editor de código de la parte del servidor, un agente para lanzar scripts programados, soporte para el motor de replicación Slony-I y mucho más. La conexión al servidor puede hacerse mediante conexión TCP/IP o Unix Domain Sockets (en plataformas *nix), y puede encriptarse mediante SSL para mayor seguridad. (*Comunidad de Desarrollo Ubuntu 2008*)[*Comunidad de Desarrollo Ubuntu 2008*]

PgAdmin III es una herramienta de código abierto para la administración de bases de datos PostgreSQL, que incluye además:

- Interfaz administrativa gráfica.
- Herramienta de consulta SQL (con un EXPLAIN gráfico).
- Editor de código procedural.
- Agente de planificación SQL/shell/batch .
- Administración de Slony-I.
- Está disponible para varios sistemas operativos, incluyendo Microsoft Windows, Linux, FreeBSD, Mac OSX y Solaris.

1.2.2 PhpPgAdmin

PhpPgAdmin (*phpPgAdmin group 2011*) es una aplicación realizada en PHP que permite administrar desde la Web los servidores de base de datos PostgreSQL y provee una manera conveniente a los usuarios para crear bases de datos, tablas, alterarlas y consultar sus datos usando el lenguaje estándar SQL.

PhpPgAdmin estuvo basado en phpMyAdmin, pero hoy día ya no comparte código con él; incluso provee las mismas funcionalidades y más, a los usuarios del servidor de base de datos PostgreSQL.

Herramientas de administración de base de datos

Características de PhpPgAdmin

- Posibilidad de administrar varios servidores.
- Soporte a múltiples versiones de PostgreSQL.
- Administración de todos los objetos del gestor PostgreSQL (usuarios, grupos, bases de datos, esquemas, etc.).
- Se pueden lanzar sentencias SQL contra la base de datos.
- Proporciona el acceso de múltiples usuarios a la vez, debido a su condición de aplicación Web.

1.2.3 Navicat

Navicat (*Navicat for PostgreSQL group 1999*) es un excelente software para gestionar bases de datos MySQL, Oracle, PostgreSQL y SQLite. Diseñado para satisfacer las necesidades de los administradores de bases de datos, desarrolladores y personas con pequeñas y medianas empresas. Navicat integra una interfaz gráfica de usuario bastante intuitiva utilizada para crear, organizar, acceder y compartir información de una manera segura y sencilla. Permite además, realizar copias de seguridad/restaurar, importar/exportar datos, sincronizar bases de datos y conectar a servidores MySQL, SQLite, SQL Server, Oracle y PostgreSQL, etc.

Características de Navicat

- Soporte a servidores PostgreSQL 7.3 o superior.
- Gestiona todos los objetos PostgreSQL (tablas, vistas, funciones, etc.).
- Soporta todos los subobjetos (campos, índices, llaves foráneas, únicas, reglas etc.).
- Incluye un Constructor Visual de Consultas.

Herramientas de administración de base de datos

- Opciones de Exportación/Importación.
- Constructor de Reportes.
- Opciones de Copia de Seguridad / Restauración.

1.2.4 PgAccess

PgAccess (*pgaccess group 2011*) es una herramienta de interface gráfica para el gestor de bases de datos PostgreSQL escrito por *Constantin Teodorescu* en el lenguaje Tcl/Tk. Permite a los usuarios interactuar con PostgreSQL de una manera similar a muchas aplicaciones de bases de datos para PC, con menús de opciones y diversas herramientas gráficas. PgAccess es una herramienta *open source*⁴ que no cambia el modo de actuar de PostgreSQL, sólo hace más fácil su uso para aquellos que estén habituados a interfaces gráficas.

PgAccess permite:

- Conectarse a cualquier base de datos en un determinado host, especificando host, puerto, nombre de usuario y contraseña.
- Gestionar a través de una interfaz gráfica: tablas, secuencias, vistas, funciones, etc.
- Generar informes simples desde una tabla

1.2.5 EMS SQL Manager Lite

EMS SQL Manager Lite para PostgreSQL (*EMS Database Management Solutions 1999*) es una herramienta gráfica sencilla, liviana y gratuita para la administración de PostgreSQL. PostgreSQL Manager funciona con cualquier versión de PostgreSQL, hasta la 8.1, y soporta todas las nuevas características de PostgreSQL incluyendo espacios de tablas (tablespaces), argumentos nombrados (*named arguments*) en funciones y otras más. Cuenta

⁴ Código abierto: El código está disponible para el usuario, y puede ser modificado.

Herramientas de administración de base de datos

con un conjunto mínimo de instrumentos para aquellos usuarios principiantes de PostgreSQL y necesitan usar únicamente funcionalidades básicas.

Características de EMS SQL Manager Lite

- Ofrece la oportunidad de administrar completamente los servidores basados en lenguaje PostgreSQL.
- Posee una interfaz gráfica bastante similar al clásico explorador de Windows.
- Soporta servidores PostgreSQL 7.3 o superior.

1.2.6 Comparación entre las herramientas de administración analizadas

Las herramientas analizadas presentan numerosas características para la administración del gestor PostgreSQL, quedando representadas a continuación, mediante una tabla comparativa.

Tabla 2: Comparación ente herramientas de administración libres para PostgreSQL

<i>Herramientas de administración</i>	<i>Herramienta Web</i>	<i>Multiplataforma</i>	<i>Replicación de datos integrada</i>	<i>Gestión de objetos</i>
PgAdmin III		X	X	X
PhpPgAdmin	X	X	X	X
Navicat		X		X
PgAccess		X	X	X
EMS SQL Manager Lite				X

Herramientas de administración de base de datos

Después del análisis realizado, se determinó que las herramientas expuestas presentan importantes ventajas para la administración de objetos en PostgreSQL, posibilitando a los usuarios gestionar bases de datos y dar mantenimiento a las mismas.

Es válido destacar la diferencia que existe entre las herramientas comerciales que brindan a los usuarios comodidad y confianza en su trabajo, y las libres que a pesar de ser distribuidas y usadas sin restricciones, carecen de interfaces agradables e intuitivas para la administración del gestor. Obstaculizando con ello, la migración del país a software libre y provocando la reducción de su calidad con respecto a las herramientas comerciales.

En el caso de la herramienta PhpPgAdmin, es importante resaltar que a pesar de ser Web, brindar a los usuarios facilidades de uso desde cualquier navegador las 24 horas del día y permitir la gestión de todos los objetos del gestor PostgreSQL, presenta la dificultad de no contar con las facilidades que ofrece la Web 2.0, lo que implica, entre otras cosas, recargar constantemente las páginas cada vez que se realice una acción sobre la misma.

El resto de las herramientas analizadas, incluida PhpPgAdmin, contienen interfaces poco sugerentes e intuitivas, obligando a los usuarios a poseer conocimientos previos de administración de base de datos para interactuar con la herramienta sin contradicciones. Además, carecen de funcionalidades para la administración y visualización de datos espaciales y el monitoreo del rendimiento del gestor.

Una vez finalizado el análisis de cada una de las herramientas, se llega a la conclusión de elaborar una herramienta Web libre, que contenga:

- Las facilidades que ofrecen la Web 2.0, como el uso de AJAX⁵ para impedir la recarga constante de las páginas y la separación de contenido del diseño con uso de hojas de estilo.

⁵ AJAX: Acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas.

Herramientas de administración de base de datos

- Interfaces sencillas, intuitivas y fáciles de usar por usuarios inexpertos en la administración de base de datos.
- Eficiencia en la administración de los objetos del gestor PostgreSQL.
- Permita la total integración con otras herramientas para la administración y representación de objetos espaciales, la réplica entre base de datos, el monitoreo y control del servidor, la representación gráfica de diagramas y consultas y la migración de otros gestores de bases de datos como MySQL y Oracle hacia PostgreSQL.

1.3. Ambiente de desarrollo de la aplicación

En la actualidad, gracias al avance de Internet, han surgido diversidad de tecnologías y lenguajes libres para la creación de aplicaciones Web, las cuales ofrecen grandes facilidades para establecer comunicación con el usuario mediante las páginas Web, por su capacidad para ser visualizadas desde cualquier parte del mundo haciendo uso de un navegador.

Para el desarrollo de la aplicación se optó por el uso de tecnologías, framework, bibliotecas y herramientas libres, esto se debe fundamentalmente a la necesidad que tiene el país de migrar todas sus soluciones informáticas al mundo de software libre. Para ello se determinó el uso de la biblioteca Ext JS en la elaboración de las interfaces de usuario, auxiliándose de CSS, JavaScript y HTML, el framework de desarrollo Web Django y el lenguaje de programación Python para el lado del servidor, la herramienta Visual Paradigm para el modelado UML, el IDE Eclipse para facilitar la programación de la herramienta y el servidor Web Apache para la publicación de la aplicación Web.

A continuación se describen cada una de las tecnologías, framework, bibliotecas y herramientas libres mencionadas:

Herramientas de administración de base de datos

1.3.1 Tecnologías

1.3.1.1 Lenguaje de programación del lado del servidor

Los lenguajes de programación del lado del servidor son aquellos que son reconocidos, ejecutados e interpretados por el propio servidor y que se envían al cliente en un formato comprensible para él.

❖ Python

Python (*Python Software Foundation 1990*) es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparado para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas Web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo. En los últimos años el lenguaje se ha hecho muy popular, gracias a varias razones como:

- La gran cantidad de librerías que contiene, tipos de datos y funciones incorporadas en el propio lenguaje, que ayudan a realizar muchas tareas habituales sin necesidad de tener que programarlas desde cero.
- La sencillez y velocidad con la que se crean los programas. Un programa en Python puede tener de 3 a 5 líneas de código menos que su equivalente en Java o C.
- La cantidad de plataformas en las que podemos desarrollar, como Unix, Windows, OS/2, Mac, Amiga y otros.
- Además, Python es gratuito, incluso para propósitos empresariales.” (*Miguel Ángel Álvarez 2003*)

Características de Python

- *Propósito general:* Con Python se pueden crear todo tipo de programas.

Herramientas de administración de base de datos

- *Multiplataforma:* Hay versiones disponibles de Python en múltiples sistemas operativos. Originalmente se desarrolló para Unix, aunque cualquier sistema es compatible con el lenguaje siempre y cuando exista un intérprete programado para él.
- *Interpretado:* Quiere decir que no es necesario compilar el código antes de su ejecución. Aunque en realidad sí que se realiza una compilación, pero esta es transparente para el programador.
- *Interactivo:* Python dispone de un intérprete por línea de comandos en el que se pueden introducir sentencias. Cada sentencia se ejecuta y produce un resultado visible, ayudando a entender mejor el lenguaje y probar rápidamente los resultados con la ejecución de porciones de código.
- *Orientado a Objetos:* La programación orientada a objetos está soportada en Python y ofrece en muchos casos una manera sencilla de crear programas con componentes reutilizables.
- *Funciones y librerías:* Dispone de muchas funciones incorporadas en el propio lenguaje, para el tratamiento de cadenas (*strings*), números, archivos, etc. Además, existen muchas librerías que pueden ser incorporadas a los programas para tratar temas específicos como la programación de ventanas o sistemas en red o cosas tan interesantes como crear archivos comprimidos en .zip.
- *Sintaxis clara:* Python tiene una sintaxis muy visual, gracias a una notación indentada (con márgenes) de obligado cumplimiento. En muchos lenguajes, para separar porciones de código, se utilizan elementos como las llaves o las palabras clave *begin* y *end*. Para separar las porciones de código en Python se debe tabular hacia dentro, colocando un margen al código que iría dentro de una función o un bucle. Esto ayuda a que todos los programadores

Herramientas de administración de base de datos

adopten unas mismas notaciones y que los programas de cualquier persona tengan un aspecto muy similar.

Al igual que Java, Python cuenta con una amplia librería estándar que acompaña al lenguaje. Ésta incluye módulos para manejar expresiones regulares, crear interfaces gráficas, conectarse a bases de datos entre muchos otros.

- **Psycopg** (*Daniele Varrazzo 2010*) es un adaptador de PostgreSQL pequeño, práctico y muy popular para el lenguaje de programación Python.

En su núcleo implementa plenamente las especificaciones de Python DB API 2.0. Varias extensiones permiten el acceso a muchas de las características de PostgreSQL.

Características de Psycopg:

- ✓ Admite las versiones de Python de 2.4 a 3.2 y versiones de PostgreSQL de 7.4 a 9.1.
- ✓ Es totalmente compatible con la implementación de la especificación de API de Python DB para los adaptadores de base de datos.

1.3.1.2 Lenguaje de programación del lado del cliente

Los lenguajes de lado del cliente son aquellos que pueden ser directamente comprendidos por el navegador y no necesitan un pre tratamiento.

❖ HTML

Definiéndolo de forma sencilla, HTML⁶ es lo que se utiliza para crear todas las páginas Web de Internet. Más concretamente, HTML es el lenguaje con el que se escriben la mayoría de páginas Web. (*Javier Eguíluz Pérez 2008*)

⁶ HTML como indican sus siglas significa (*HyperText Mark-Up Language*), que viene a ser, en nuestro idioma, “Lenguaje para el Formato de Documentos de Hipertexto”.

Herramientas de administración de base de datos

HTML es un lenguaje de marcas hipertextuales, diseñado para estructurar textos para generar páginas Web. Gracias a Internet y a los navegadores Web, HTML se ha convertido en el formato más fácil para la creación de páginas Web debido a su sencillez.

La mayoría de las etiquetas del lenguaje HTML son semánticas y su interpretación es realizada por el navegador Web. HTML se caracteriza por ser un lenguaje extensible, que se le pueden añadir características, etiquetas y funciones adicionales para el diseño de páginas Web, generando un producto vistoso, rápido y sencillo.

❖ *JavaScript*

JavaScript es un lenguaje de programación utilizado para crear pequeños programas encargados de realizar acciones dentro del ámbito de una página Web. Con JavaScript se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso, y tal vez el único, con que cuenta este lenguaje es el propio navegador. (*Miguel Ángel Álvarez 2002*)

Java Script es un lenguaje basado en objetos y guiado por eventos, diseñado específicamente para el desarrollo de aplicaciones cliente-servidor dentro del ámbito de Internet. Los programas escritos con este lenguaje se pueden probar directamente en cualquier navegador sin necesidad de procesos intermedios, convirtiéndolo en un lenguaje interpretado.

Ventajas de JavaScript

- Los programas escritos en este lenguaje no requieren de mucha memoria ni tiempo adicional de transmisión, por ser pequeños y compactos.
- JavaScript no requiere un tiempo de compilación; ya que los scripts se pueden desarrollar en un período de tiempo relativamente corto.

Herramientas de administración de base de datos

- Es independiente de la plataforma, hardware o sistema operativo, y funciona correctamente siempre y cuando exista un navegador con soporte JavaScript.

JavaScript es el siguiente paso, después del HTML, que puede dar un programador de la Web que decida mejorar sus páginas y la potencia de sus proyectos. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica. (*Miguel Ángel Álvarez 2002*)

❖ CSS

Es un lenguaje de hojas de estilos (*Cascading Style Sheets*) creado para controlar la presentación de documentos estructurados y escritos en XHTML, aspectos como: el color, el tamaño, el tipo de letra, la separación entre párrafos y la tabulación con la que se muestran los elementos de una lista. El propósito del desarrollo de CSS es separar la estructura y el contenido de la presentación estética en un documento, esto permite un control mayor del documento y sus atributos, convirtiendo al XHTML en un documento muy versátil y liviano.

Separar la definición de los contenidos y la definición de su aspecto presenta numerosas ventajas, ya que obliga a crear documentos HTML/XHTML bien definidos y con significado completo (también llamados “documentos semánticos”). Además, mejora la accesibilidad del documento, reduce la complejidad de su mantenimiento y permite visualizar el mismo documento en infinidad de dispositivos diferentes. (*Javier Eguíluz Pérez 2009*)

Ventajas de CSS

- Control de la presentación de muchos documentos desde una única hoja de estilo.
- Control más preciso de la presentación.

Herramientas de administración de base de datos

- Aplicación de diferentes presentaciones a diferentes tipos de medios (pantalla, impresión, etc.)

1.3.1.3 Servidores

❖ Apache

Apache (*Apache Software Foundation 2011*) es un servidor Web potente, flexible y disponible para distintas plataformas y entornos. Es altamente configurable de diseño modular, posibilitando que los administradores de sitios Web puedan elegir los módulos que serán incluidos y ejecutados en el servidor.

Características de Apache

- Es una tecnología gratuita y de código abierto, lo que proporciona transparencia en todo el proceso de instalación.
- Es prácticamente universal, por su disponibilidad en multitud de sistemas operativos.
- Posee una alta configurabilidad en la creación y gestión de *logs*, de este modo es posible tener un mayor control sobre lo que sucede en el servidor.

Este servidor Web Apache tiene una fácil integración con varios lenguajes de programación como: Java, Perl, PHP y especialmente Python. Dicha relación está dada por Mod_python que es un módulo de Apache que integra el intérprete de Python dentro del servidor.

Con mod_python se pueden desarrollar aplicaciones Web basadas en Python que se ejecutan mucho más rápido que el CGI tradicional y tienen acceso a características avanzadas como la capacidad de mantener conexiones con la base de datos.

1.3.2 Frameworks de desarrollo

Un framework en el desarrollo de software, es una estructura de soporte definida en la cual varios proyectos de software pueden ser organizados y desarrollados.

Herramientas de administración de base de datos

❖ *Django*

Django (*Django Software Foundation 2005*) es un framework Web de código abierto escrito en Python que permite construir aplicaciones Web más rápido y con menos código.

Django fue inicialmente desarrollado para gestionar aplicaciones Web de páginas orientadas a noticias de World Online, más tarde se liberó bajo licencia BSD. Django se centra en automatizar todo lo posible y se adhiere al principio *DRY (Don't Repeat Yourself)*. (*Django Software Foundation 2010*)

Características de Django

- Mapeador objeto-relacional.
- Aplicaciones integrables que pueden instalarse en cualquier página gestionada con Django.
- API de base de datos robusta.
- Un sistema extensible de plantillas basado en etiquetas, con herencia de plantillas.
- Un despachador de URLs basado en expresiones regulares.
- Un sistema middleware para desarrollar características adicionales.
- Documentación incorporada accesible a través de la aplicación administrativa (incluyendo documentación generada automáticamente de los modelos y las bibliotecas de plantillas añadidas por las aplicaciones).

1.3.3 Bibliotecas

Las bibliotecas en el desarrollo de software son de gran importancia, ya que enriquecen los lenguajes de programación incorporando funcionalidades nuevas y facilitando el trabajo con los mismos.

Herramientas de administración de base de datos

❖ *ExtJS*

ExtJS (*Sencha 2011*) es una biblioteca Java Script de componentes gráficos, que facilita las herramientas necesarias para la creación de aplicaciones Web. Posee una considerable colección de componentes para el diseño de interfaces, ventanas, cuadros de diálogo, menús, tablas editables, layouts, paneles, pestañas y todo lo necesario para construir atractivos desarrollos al estilo de la Web 2.0.

ExtJS brinda soporte para:

- Construir interfaces gráficas complejas y dinámicas.
- Comunicar datos de forma asíncrona con el servidor.
- Diversos navegadores como: Internet Explorer, Firefox, Safari y Opera.

La biblioteca ExtJS incluye componentes UI⁷ del alto performance y personalizables, modelo de componentes extensibles, un API fácil de usar y presenta dos tipos de licencia *open source* y comercial.

Beneficios de ExtJS

- *Existe un balance entre Cliente – Servidor:* La carga de procesamiento se distribuye, permitiendo que el servidor, al tener menor carga, pueda manejar más clientes al mismo tiempo.
- *Comunicación asíncrona:* En este tipo de aplicación el motor de *render* puede comunicarse con el servidor sin necesidad de estar sujeta a un clic o una acción del usuario, dándole la libertad de cargar información sin que el cliente se dé cuenta.

⁷ UI (User Interface): La interfaz de usuario es el sistema a través del cual las personas (usuarios) pueden interactuar con las máquinas.

Herramientas de administración de base de datos

- *Eficiencia de la red:* El tráfico de red puede disminuir al permitir que la aplicación elija que información desea transmitir al servidor y viceversa, sin embargo la aplicación que haga uso de la pre-carga de datos puede que revierta este beneficio por el incremento del tráfico.

1.3.4 Herramientas

Las herramientas de desarrollo son aquellos programas o aplicaciones que tienen gran importancia en el desarrollo de un programa, apoyando el proceso activo de construcción del mismo.

❖ Visual Paradigm

Visual Paradigm para UML⁸ (VP-UML) (*Visual Paradigm Group 2011*) es una poderosa herramienta CASE⁹ multiplataforma (Windows/Linux/Mac OS X) que soporta el ciclo de vida completo del desarrollo de software. Está diseñado para un amplio rango de usuarios, incluyendo ingenieros de software, analistas de sistema, analistas de negocio, arquitectos de sistema y quienes estén interesados en la construcción de sistemas de software confiables mediante el uso de la Orientación a Objetos.

Este software facilita una rápida construcción de aplicaciones de calidad y a un menor coste. Visual Paradigm para UML soporta un conjunto de lenguajes (Java, C ++, PHP, Ada y Python), tanto en generación de código como ingeniería inversa.

Entre sus características principales se pueden citar:

- Soporta UML en su versión 2.1.
- Facilita la comunicación de todo el equipo de desarrollo mediante el uso de un lenguaje estándar común.
- Posibilita el desarrollo de la ingeniería directa e inversa.

⁸ UML (Unified Modeling Language): Lenguaje Unificado de Modelado.

⁹ CASE: Computer Aided Software Engineering

Herramientas de administración de base de datos

- Durante todo el ciclo de desarrollo el modelo y el código permanecen sincronizados, permitiendo la generación de código a partir de diagramas y viceversa.
- Se encuentra disponible en múltiples versiones y plataformas.
- Permite la generación de bases de datos a partir de la transformación de diagramas de Entidad-Relación en tablas de base de datos y viceversa.

❖ *Eclipse*

Eclipse (*Eclipse Foundation 2011*) es un entorno de desarrollo integrado (IDE¹⁰) de código abierto y multiplataforma, que facilita enormemente las tareas de edición, compilación y ejecución de programas durante su fase de desarrollo.

Características de Eclipse

- Dispone de un Editor de texto con resaltado de sintaxis
- La compilación es en tiempo real.
- Tiene asistentes (*wizards*) para creación de proyectos, clases, pruebas, etc.
- Mediante el uso de *plugins*¹¹ libremente disponibles es posible añadir control de versiones con *Subversion*.

La principal ventaja que posee Eclipse es la cantidad de *plugins* con que se cuenta, los cuales posibilita desarrollar en cualquier lenguaje de programación así como utilizar herramientas que facilitan el trabajo a los desarrolladores en la elaboración de aplicaciones.

¹⁰ En inglés IDE (*Integrated Development Environment*)

¹¹ *Plugins: Aplicación que se relaciona con otra para aportarle una función nueva y generalmente muy específica.*

Herramientas de administración de base de datos

A continuación se mencionan los *plugins* utilizados para el desarrollo de la aplicación

- *Pydev* para el trabajo con Python.
- *Subversive* para el control de versiones del código fuente de la aplicación.

1.4. Conclusiones del capítulo

Después del análisis de múltiples herramientas para la administración del gestor PostgreSQL, se llegó a la conclusión de elaborar una herramienta Web libre para la administración del gestor PostgreSQL, que contenga las principales características de las herramientas analizadas durante el capítulo e integre nuevas funcionalidades como la administración y visualización de datos espaciales, la réplica entre bases de datos, la migración de datos hacia diferentes gestores y el monitoreo y control del funcionamiento del servidor. Utilizándose para su elaboración herramientas libres que permitan desarrollar una aplicación consistente y robusta.

Modelado de la herramienta de administración para PostgreSQL

CAPÍTULO II: Modelado de la herramienta de administración para PostgreSQL.

Durante la elaboración de un sistema, es de vital importancia mantener una adecuada planificación y documentación del mismo. Esto ayuda a comprender el problema, permitiendo comunicar ideas acerca de su solución y además sirve de guía para su implementación. Todo este proceso posibilita un trabajo más organizado y reduce errores durante el desarrollo de la aplicación.

En este capítulo se realiza una detallada descripción de la arquitectura de la herramienta y se representa el modelado del sistema a través de los diagramas de casos de uso del sistema, clases, secuencia y diseño de la base de datos.

2.1. Arquitectura del sistema

Las necesidades actuales que tiene toda organización para el logro de sus objetivos, demandan la construcción de grandes y complejos sistemas de software que requieren de la combinación de diferentes tecnologías y plataformas de hardware y software para alcanzar un funcionamiento acorde con dichas necesidades. Lo anterior, exige de los profesionales dedicados al desarrollo de software poner especial atención y cuidado al diseño de la arquitectura, bajo la cual estará soportado el funcionamiento de sus sistemas.

La herramienta de administración propuesta a desarrollar, permite la gestión de todos los objetos del gestor PostgreSQL y además incluye un conjunto de funcionalidades, a continuación detalladas, que permitan a los usuarios sentirse cómodos al administrar servidores PostgreSQL:

- Administración de un sistema de réplica empleando el mecanismo SymmetricDS
- Herramientas de migración de datos desde otros gestores de BD hacia PostgreSQL.

Modelado de la herramienta de administración para PostgreSQL

- Administración de objetos espaciales.
- Herramienta para diseñar diagramas de relación entre tablas.
- Herramienta para diseñar y visualizar consultas SQL, incluyendo la visualización de objetos espaciales.
- Monitoreo y control de los servidores de BD registrados.

El desarrollo de la herramienta para la administración de PostgreSQL, persigue la creación de módulos independientes que tributen a una interfaz común diseñada sobre la Web 2.0 con el objetivo de proporcionar a los usuarios un trabajo con el gestor PostgreSQL de forma compacta y con mayor interactividad. A continuación se describen cada uno de estos módulos:

- *Módulo Común:* contiene las funcionalidades comunes de los 5 módulos restantes y permite la integración de estos en una interfaz común.
- *Módulo Gestión de Objetos:* se centra en la administración de todos los objetos de una BD PostgreSQL, dígame con ello tablas, columnas, vistas, chequeos, dominios, funciones, disparadores, etc.
- *Módulo Réplica:* consiste en la administración del mecanismo de replicación SymmetricDS, que tiene como característica principal el Multy Master (capacidad de replicar para ambos sentidos).
- *Módulo Migración:* se enmarca en los mecanismos de migración de estructuras y datos desde varios gestores como SQL Server 2000, MySql, Oracle y FoxPro hacia PostgreSQL.
- *Módulo Postgis:* se basa en el desarrollo de herramientas para administración de bases de datos espaciales sobre PostgreSQL empleando Postgis. Además este módulo contiene:

Modelado de la herramienta de administración para PostgreSQL

- ✓ Herramienta para la migración de datos hacia formato *postgis* (.shp).
- ✓ Herramienta para visualización de consultas espaciales sobre la Web.
- *Módulo Monitoreo y Control*: supervisa todos los servicios que ofrece el servidor de BD PostgreSQL y proporciona una serie estadísticas relacionadas con su funcionamiento.

Para una mayor comprensión en cuanto a la estructura de implementación de la herramienta, su arquitectura se dividió en dos partes: cliente y servidor *ver figura 2*.

- La parte cliente está compuesta por las bibliotecas JavaScript a utilizar durante el desarrollo de los prototipos de usuario. A continuación se detallan cada una de ellas:
 - ✓ Biblioteca ExtJS 3.0 usada para el diseño de interfaces de usuario, utilizando AJAX para las peticiones al servidor Web. ExtJS es una biblioteca de componentes gráficos que facilita las herramientas necesarias para la creación de aplicaciones Web.
 - ✓ Biblioteca Open Jacob Draw 2D usada para el diseño de las interfaces del módulo replica y en los diseñadores de diagramas y consultas, facilitando con ello el trabajo de la representación y arrastrado de los objetos. Open-Jacob Draw2D es una biblioteca JavaScript que permite crear gráficos y diagramas.
 - ✓ Biblioteca Open Layers utilizada para la visualización y renderización de datos espaciales en el módulo *postgis*. Open Layers permite, de manera sencilla, mostrar un mapa dinámico y marcadores cargados desde cualquier fuente de datos.
- La parte servidor está desarrollada sobre el lenguaje de programación Python, auxiliándose del marco de trabajo de desarrollo Django para el

Modelado de la herramienta de administración para PostgreSQL

trabajo sobre la Web. El proyecto Django está vinculado con la parte cliente a través de la biblioteca ExtJS para la creación de sus vistas, haciendo uso de plantillas para su integración con la biblioteca.

- ✓ La herramienta en su totalidad está formada por un proyecto en Django denominado *Int_admin*, que usa para la conexión a las bases de datos el driver Psycopg2 y además está compuesto por 6 aplicaciones: *monitoreo*, *migración*, *réplica*, *gestión objeto*, *común* y *postgis*, que contienen las funcionalidades específicas de cada uno de los módulos identificados dentro de la herramienta.

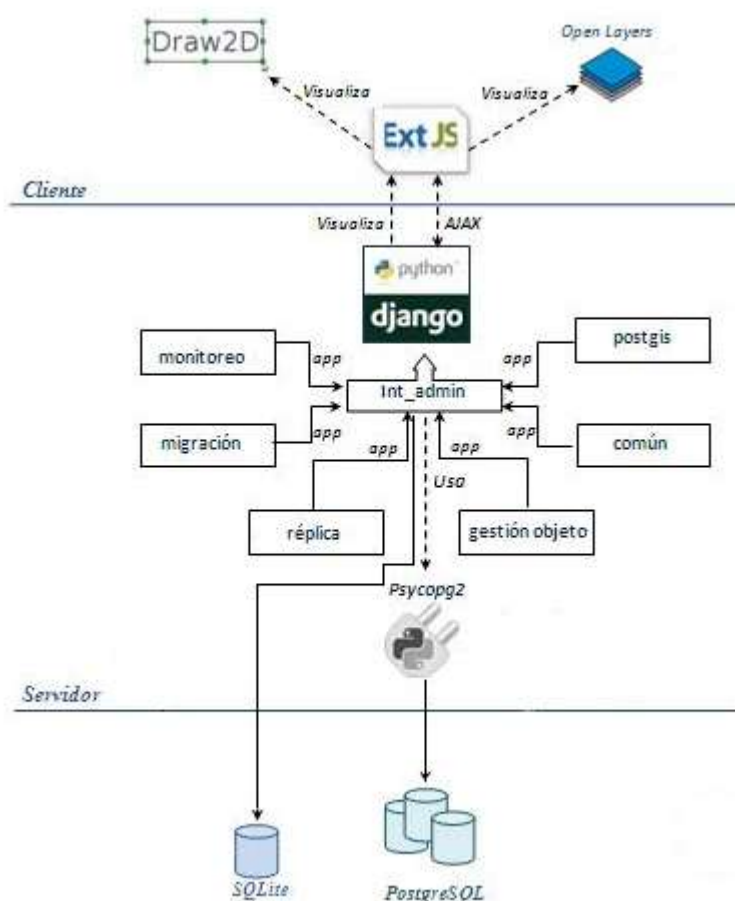


Figura 2: Diagrama de integración de tecnologías.

Modelado de la herramienta de administración para PostgreSQL

2.2. Casos de uso del sistema

Los diagramas de casos de uso sirven para facilitar la comunicación con los futuros usuarios del sistema y clientes, resultando especialmente útiles para determinar las características necesarias que tendrá el sistema. En otras palabras, los diagramas de casos de uso describen qué es lo que debe hacer el sistema, pero no cómo.

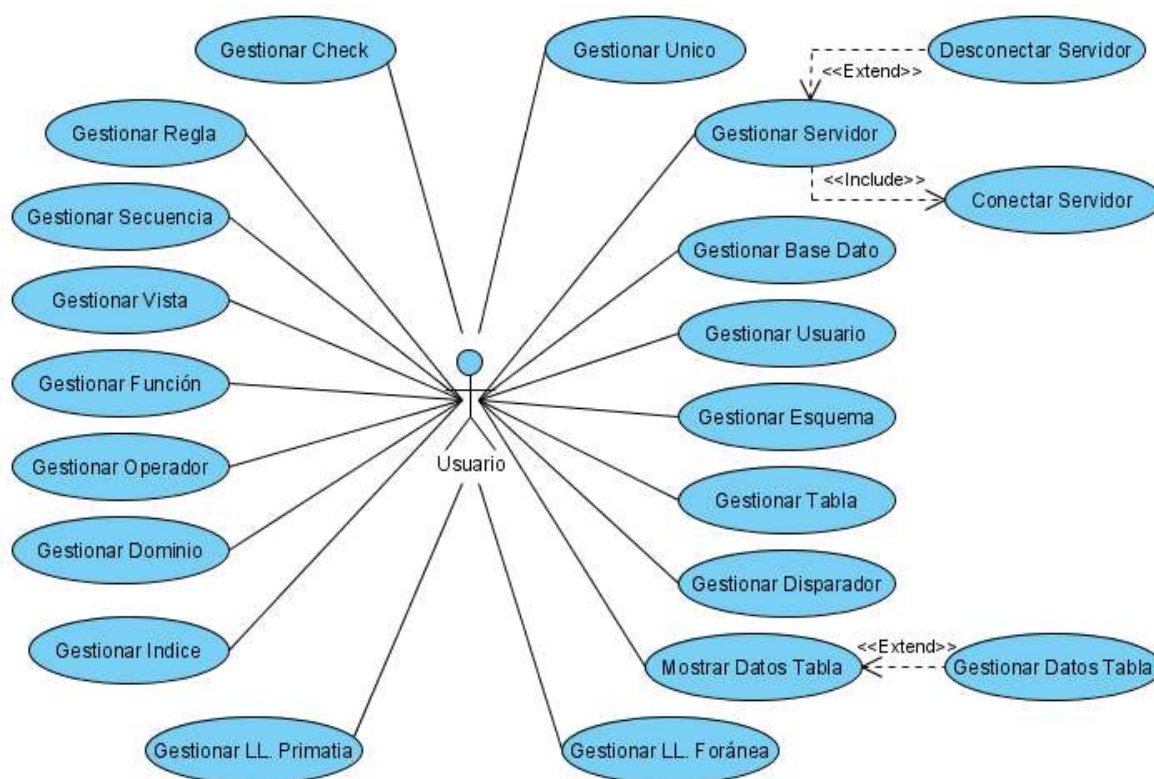


Figura 3: Diagrama de casos de uso del sistema

El diagrama de casos de uso (CU) del sistema representado en la figura 3 consta de 21 CU, de los cuales dos son extendidos el *Desconectar Servidor* que extiende del CU *Gestionar Servidor* y el *Gestionar Datos Tabla* del CU *Mostrar Datos Tabla*. Además, dentro de este diagrama también existe el CU *Conectar Servidor* que es incluido al CU *Gestionar Servidor*.

Cada uno de los CU gestionar que aparecen representados en el diagrama anterior constan de las funcionalidades crear, modificar y eliminar, excepto el caso de uso *Gestionar Datos Tabla* que incluye las funcionalidades siguientes:

Modelado de la herramienta de administración para PostgreSQL

- Insertar datos en tabla
- Modificar datos en tabla
- Eliminar datos de tabla

Los Actores del Sistema definen el comportamiento y responsabilidades (rol) de un individuo, grupo de individuos, sistema automatizado o máquina, que interactúan con el sistema. Ellos realizan las actividades y son propietarios de elementos.

En el diagrama de CU representado en la figura 3 el actor que inicializa los CU un usuario, que puede ser un administrador de base de datos, dado que debe tener dominio sobre el tema de la administración de base de datos.

2.2.1. Descripciones de casos de uso

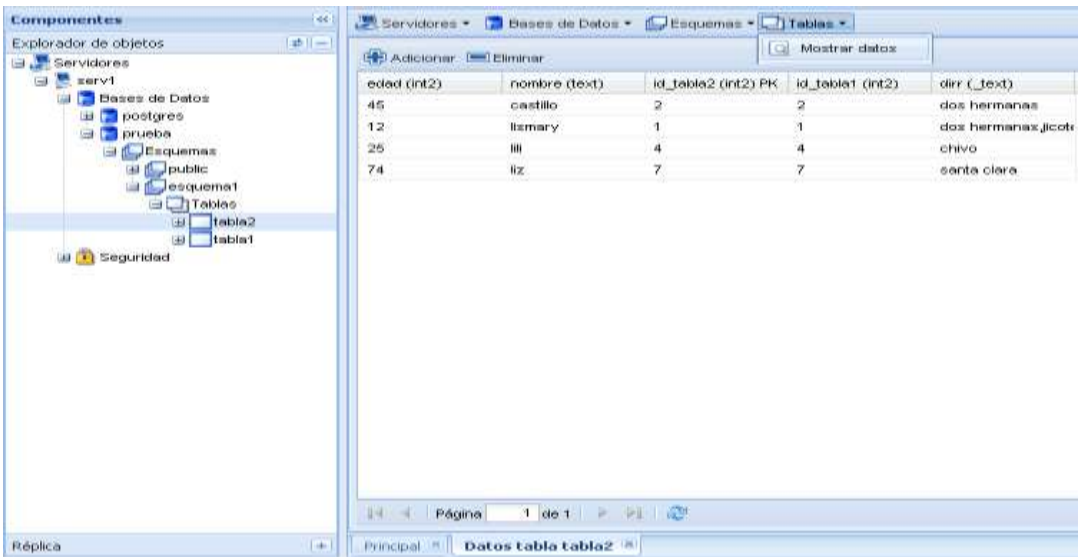
Las descripciones de CU son reseñas textuales del CU en sí mismo. Normalmente tienen el formato de una nota o un documento relacionado de alguna manera con el CU, y explica los procesos o actividades que tienen lugar en el CU.

A continuación se describen los CU Mostrar Datos Tabla y Gestionar Datos Tabla considerados unos de los más importantes en la aplicación.

Tabla 3: Descripción del caso de uso Mostrar Datos Tabla

<i>Caso de Uso:</i>	Mostrar Datos Tabla
<i>Actores:</i>	Usuario
<i>Resumen:</i>	El caso de uso se inicia cuando el usuario selecciona en el árbol de componentes uno de las tablas creadas. Una vez seleccionada esta, elige la opción Mostrar datos. Seguidamente el sistema muestra una tabla con todos los datos que contiene dicha tabla, finalizando así el CU.
<i>Precondiciones:</i>	Debe existir una tabla creada.

Modelado de la herramienta de administración para PostgreSQL


Referencias	
Prioridad	
Flujo Normal de Eventos	
1. El usuario selecciona en el árbol de componentes una de las tablas creadas.	2. Una vez seleccionada la tabla el sistema nos permite seleccionar la opción Mostrar datos.
3. El usuario selecciona la opción Mostrar datos.	4. Muestra la interfaz Mostrar datos, que a través de una tabla permite visualizar todos los datos que contiene la tabla seleccionada. Además permite: <ul style="list-style-type: none"> • Insertar nuevos datos • Modificar datos • Eliminar datos
5. Si el usuario desea insertar, modificar o eliminar datos de la tabla. ver caso de uso extendido “Gestionar Datos Tabla”	
Prototipo de Interfaz	
	
Poscondiciones	Datos visualizados

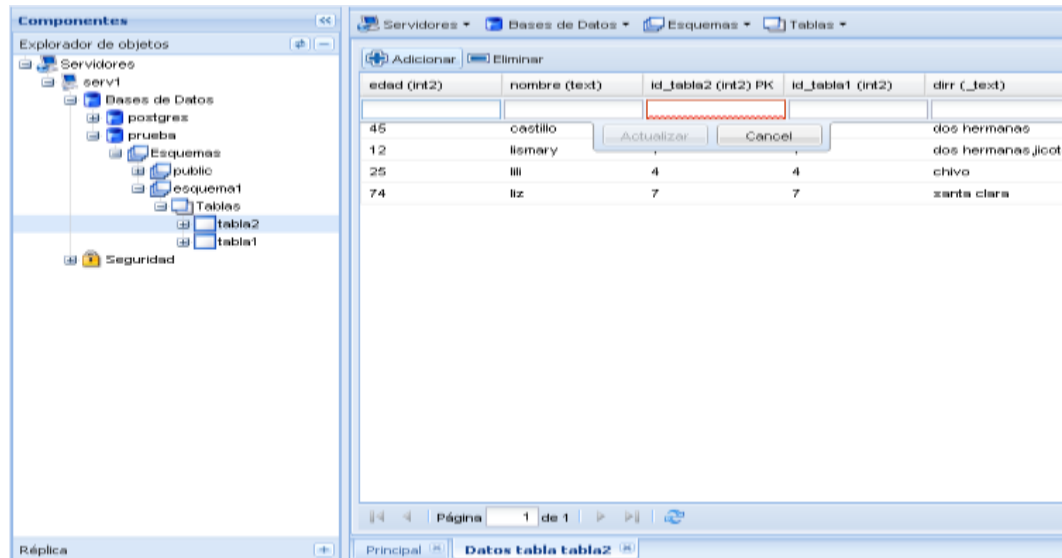
Modelado de la herramienta de administración para PostgreSQL


Tabla 4: Descripción del caso de uso Gestionar Datos Tabla

Caso de Uso:	Gestionar Datos Tabla
Actores:	Usuario
Resumen:	El caso de uso se inicia cuando el usuario necesita gestionar los datos existentes en una tabla, ya sea para insertar datos nuevos, modificar datos ya existentes o eliminarlos. El sistema proporciona las interfaces necesaria para realizar estas operaciones, finalizando así el CU.
Precondiciones:	Debe existir una tabla creada y debe haberse ejecutado el CU “Mostrar Datos Tabla”.
Referencias	
Prioridad	
Flujo Normal de Eventos	
Acción del Actor	Respuesta del Sistema
1. El usuario desea insertar, modificar o eliminar datos en una tabla.	2. El sistema ejecuta alguna de las siguientes acciones: a. Si decide insertar datos en la tabla ir a la Sección “Insertar datos”. b. Si decide modificar datos en la tabla ir a la Sección “Modificar datos”. c. Si decide eliminar datos de la tabla ir a la Sección “Eliminar datos”.
Sección “Insertar datos”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona el botón “Adicionar”.	2. El sistema agrega una nueva fila a la tabla para introducir los nuevos datos.

Modelado de la herramienta de administración para PostgreSQL

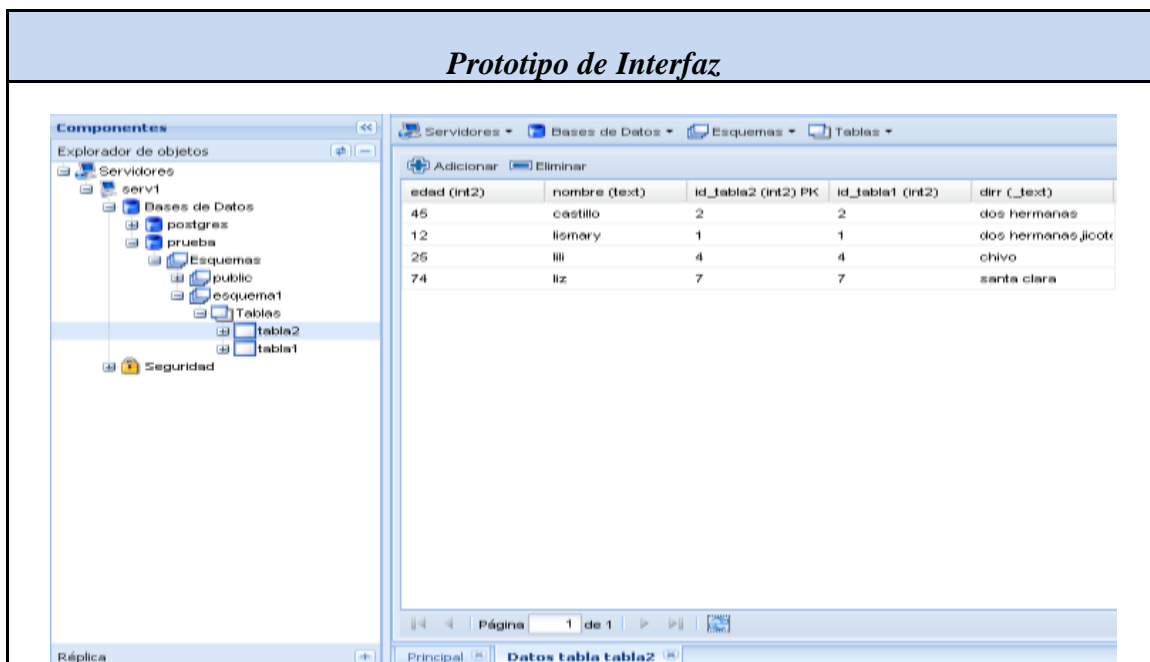
3. El usuario introduce en la fila los nuevos datos a insertar en la tabla.	
4. El usuario selecciona el botón “Actualizar”.	5. Comprueba que los nuevos datos introducidos sean válidos.
	6. Inserta los nuevos datos en la tabla.
7. El usuario selecciona el ícono  “Actualizar” para comprobar que se insertaron satisfactoriamente los datos.	8. Muestra la tabla con todos los datos existentes en la misma, incluidos los nuevos datos insertados

Prototipo de Interfaz**Flujos Alternos****Flujo Alterno 4a Selecciona el botón “Cancelar”.**

<i>Acción del Actor</i>	<i>Respuesta del Sistema</i>
4a.1 El usuario selecciona el ícono  “Actualizar” para verificar que se haya cancelado la acción.	4a.2 Muestra la tabla con todos los datos existentes en la misma y elimina la nueva fila creada para la inserción.

Modelado de la herramienta de administración para PostgreSQL

Prototipo de Interfaz

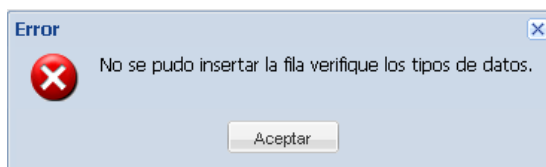


Flujos Alternos


Flujo Alterno 5a Datos erróneos.

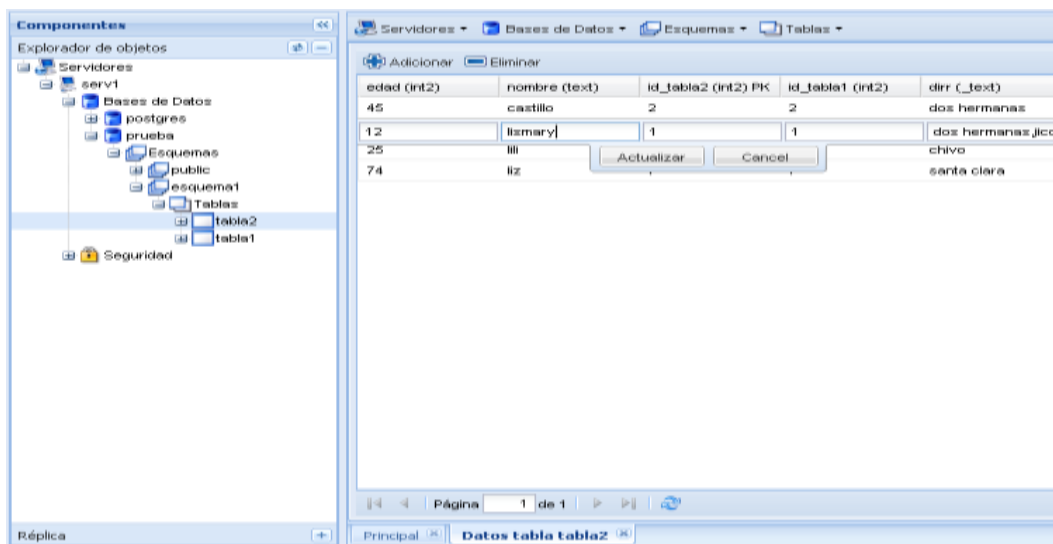
Acción del Actor	Respuesta del Sistema
	7a.1 Comprueba que existen datos erróneos.
	7a.2 Muestra el mensaje de error “No se pudo insertar la fila verifique los tipos de datos”.
7a.3 El usuario selecciona la opción “Aceptar”.	7a.5 Muestra nuevamente la interfaz para reiniciar el proceso de inserción de datos.


Prototipo de Interfaz



Modelado de la herramienta de administración para PostgreSQL

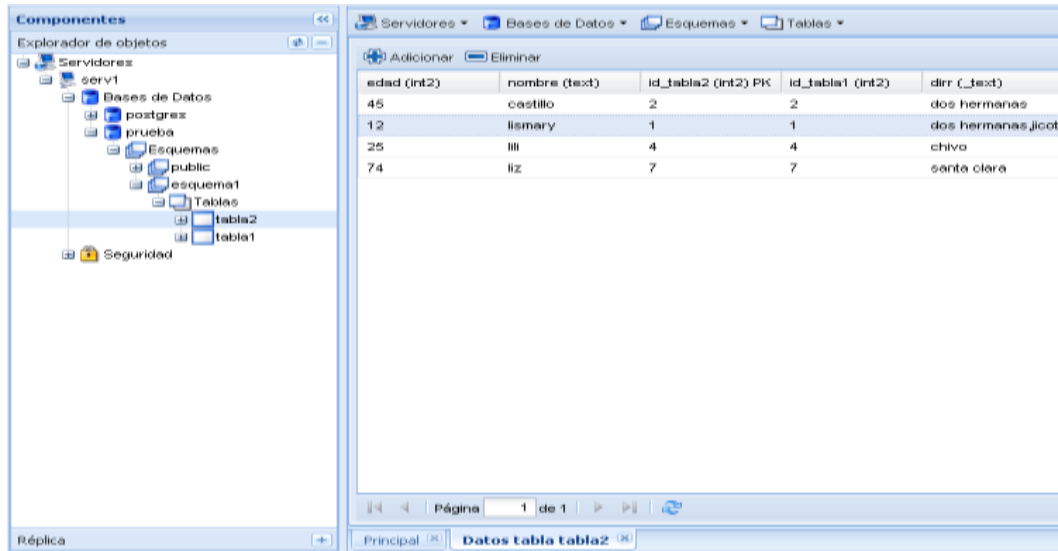
Sección “Modificar datos”	
Acción del Actor	Respuesta del Sistema
1. El usuario selecciona en la tabla la fila que desea modificar.	2. El sistema activa la fila seleccionada para que sea modificada por el usuario.
3. El usuario selecciona el botón “Actualizar”.	4. Comprueba que los datos introducidos sean válidos.
	5. Actualiza los nuevos datos en la tabla.
6. El usuario selecciona el ícono  “Actualizar” para comprobar que se actualizaron satisfactoriamente los datos.	7. Muestra la tabla con todos los datos existentes en la misma, incluidos los datos actualizados.

Prototipo de Interfaz**Flujos Alternos****Flujo Alterno 3a Selecciona el botón “Cancelar”.**

Acción del Actor	Respuesta del Sistema
3a.1 El usuario selecciona el ícono  “Actualizar” para verificar que se haya cancelado la acción.	3a.2 Muestra la tabla con todos los datos existentes en la misma sin las modificaciones.

Modelado de la herramienta de administración para PostgreSQL

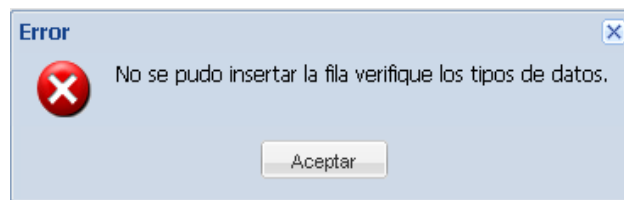
Prototipo de Interfaz




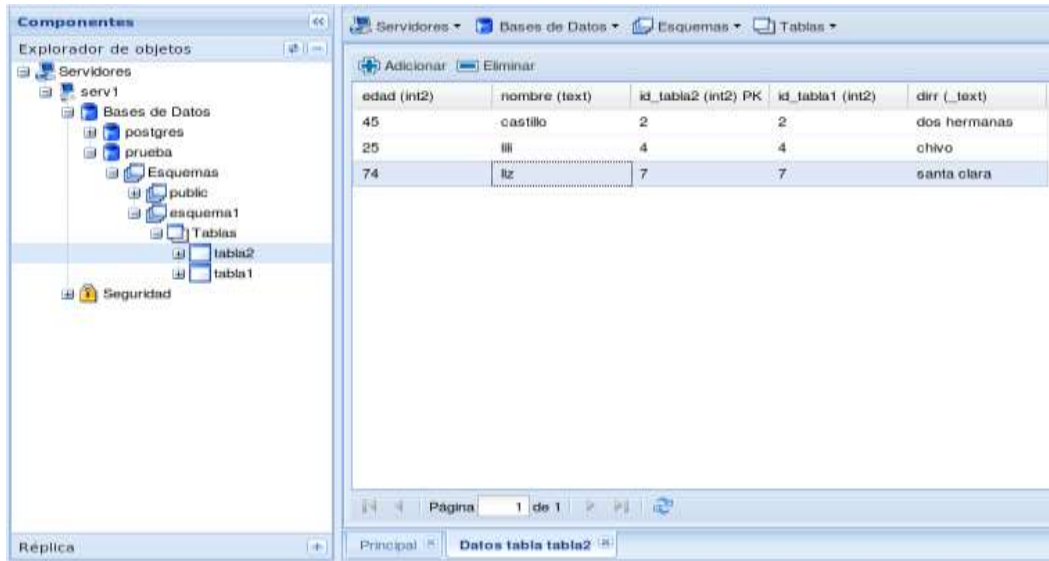
Flujo Alterno 4a Datos erróneos.

Acción del Actor	Respuesta del Sistema
	4a.1 Comprueba que existen datos erróneos.
	4a.2 Muestra el mensaje de error “No se pudo insertar la fila verifique los tipos de datos”.
4a.3 El usuario selecciona la opción “Aceptar”.	4a.5 Muestra nuevamente la interfaz para reiniciar el proceso de actualización de datos.

Prototipo de Interfaz



Modelado de la herramienta de administración para PostgreSQL

<i>Sección “Eliminar datos”</i>	
<i>Acción del Actor</i>	<i>Respuesta del Sistema</i>
1. El usuario marca la fila que desea eliminar y selecciona el botón “Eliminar”.	2. El sistema elimina de la tabla la fila seleccionada.
3. El usuario selecciona el ícono  “Actualizar” para verificar que se haya eliminado la fila.	4. Muestra los datos actuales que contiene la tabla.
<i>Prototipo de Interfaz</i>	
	
Poscondiciones	Nodo creado.

2.3. Diagramas de clases

En la elaboración de software, el modelado y definición de las clases conforman un elemento de vital importancia para el desarrollo de aplicaciones informáticas, por tanto, este proceso debe realizarse con sumo cuidado, para dejar representado en los diagramas de clases la solución propuesta al problema en cuestión.

Los diagramas de clases muestran las diferentes clases que componen un sistema y cómo se relacionan unas con otras. Se dice que los diagramas de clases son diagramas «estáticos»

Modelado de la herramienta de administración para PostgreSQL

porque muestran las clases, junto con sus métodos y atributos, así como las relaciones estáticas entre ellas: qué clases «conocen» a qué otras clases o qué clases «son parte» de otras clases, pero no muestran los métodos mediante los que se invocan entre ellas.

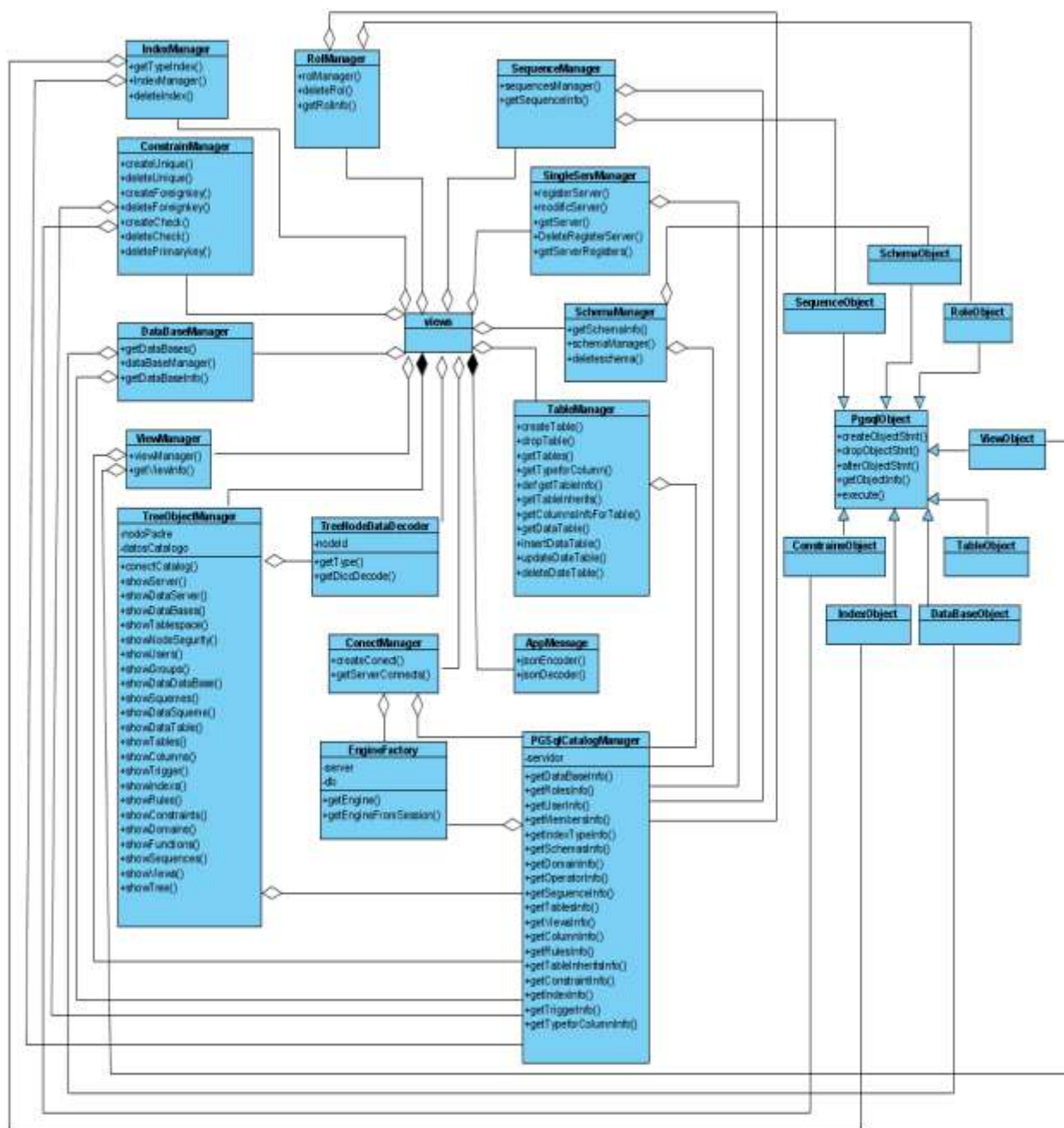


Figura 4: Diagrama de clases de la herramienta

Modelado de la herramienta de administración para PostgreSQL

A continuación se realiza una breve descripción de las clases representadas en el diagrama de la *figura 4*:

- Las clases *SingleServManager*, *RolManager*, *DataBaseManager*, *SchemaManager*, *TableManager*, *ViewManager*, *SequenceManager*, *IndexManager* y *ConstrainManager* tienen como función principal la gestión de los objetos del gestor PostgreSQL, brindando las funcionalidades crear, modificar y eliminar objetos. Estas clases hacen uso de las clases del modelo definida en el modulo `models.py` de las aplicaciones del proyecto Django.
- La clase *AppMessage* es creada específicamente para las vistas del proyecto Django y su función es codificar y decodificar a formato JSON los datos en la aplicación. En el caso de codificar se transforma las listas y los diccionarios de Python a Objetos JavaScript, y cuando se decodifica los arreglos u objetos JavaScript son convertidos en listas o diccionarios Python respectivamente. Además, esta clase hace uso del módulo `simpleJson` de Django.
- La clase *TreeObjectManager* tiene como función proporcionarle al árbol explorador de la interfaz los nodos a cargar en cada nivel del árbol. Además se auxilia de la clase *TreeNodeDataDecoder* para descomponer el id conformado en cada uno de los nodos del árbol explorador y devolver el tipo de objeto que es dicho nodo.
- La clase *ConectManager* contiene implementadas las funcionalidades para la conexión y desconexión de un servidor de base de datos en la herramienta. Para ello se auxilia de la clase *EngineFactory* que tiene la función de crear una conexión al servidor.
- La clase *PGSqlCatalogManager* tiene como función principal brindar información de todos los objetos existentes en el servidor PostgreSQL registrado en la herramienta. Todo este proceso se realiza a través de las tablas del catálogo del gestor.

Modelado de la herramienta de administración para PostgreSQL

- La clase *PgsqObject* es la clase padre de *SequenceObject*, *SchemaObject*, *RoleObject*, *ViewObject*, *TableObject*, *DataBaseObject*, *IndexObject* y *ConstraintsObject* que se encargan de conformar las sentencias SQL para crear, modificar, eliminar e insertar los objetos gestionados en el servidor de base de datos registrado.
- La clase *Views* son un módulo del proyecto Django empleado para visualizar las vistas de la herramienta.

2.4. Diagramas de secuencia

Un diagrama de secuencia es una forma de diagrama de interacción que muestra los objetos como líneas de vida a lo largo de la página y con sus interacciones en el tiempo representadas como mensajes dibujados como flechas desde la línea de vida origen hasta la línea de vida destino. Los diagramas de secuencia son buenos para mostrar qué objetos se comunican con qué otros objetos y qué mensajes disparan esas comunicaciones.

La herramienta en Django crea las interfaces de usuarios a través de las vistas, estas a su vez se comunican con las clases controladoras de cada uno de los objetos gestionados en la herramienta para realizar las operaciones indicadas por el usuario. Por otra parte, se encuentran las clases del modelo de Django que son utilizadas para el acceso a la base de datos de la aplicación donde se almacenan los datos de registro de los servidores en una base de datos SQLite. También se utiliza la clase *PGSqlCatalogManager* que permite el acceso a las tablas del catálogo de PostgreSQL para obtener información de los objetos existentes en el servidor registrado dentro de la herramienta.

La mejor forma para representar la comunicación entre las clases de la herramienta es a través de diagramas de secuencias. A continuación se representa los diagramas de secuencias para los casos de uso *Mostrar Datos Tabla* y *Gestionar Datos Tabla*, dividiendo este último en tres escenarios *Insertar Datos*, *Modificar Datos* y *Eliminar Datos* para una mejor comprensión del mismo.

Modelado de la herramienta de administración para PostgreSQL

Diagrama de secuencia para el CU Mostrar Datos Tabla

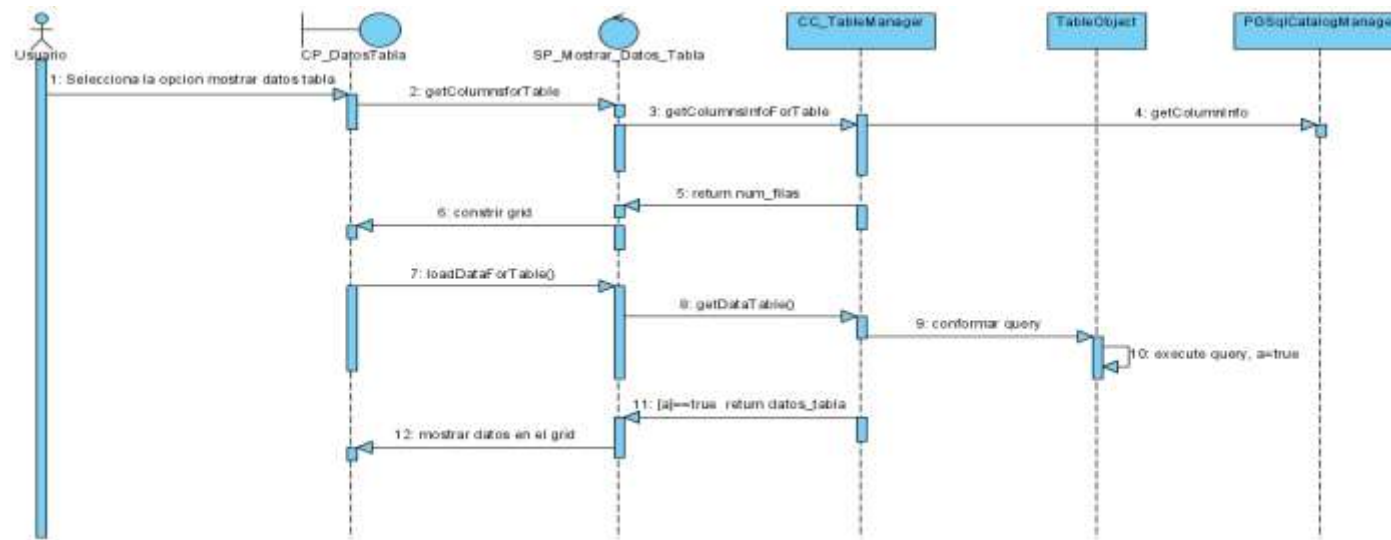


Figura 5: Diagrama de secuencia del caso de uso Mostrar Datos Tabla

En la figura anterior aparece reflejado el diagrama de secuencias del caso de uso Mostrar Datos Tabla, donde se representa al usuario que desea visualizar los datos que contiene una determinada tabla. Para ello se envía una solicitud a la vista de Django con el objetivo de cargar la cantidad de columnas que contiene la tabla. La vista a su vez, pide a la clase controladora los datos solicitados, quien para obtener esta información se vale de la clase *PGSqlCatalogManager* que realiza una consulta a las tablas del catálogo del gestor PostgreSQL. Una vez obtenida la información, es enviada a las vistas para conformar el grid donde se van a visualizar los datos de la tabla. Luego se procede a cargar los datos de la tabla, realizándose para ello una petición desde la vista hacia la clase controladora, quien solicita a la clase *TableObject* conformar la consulta SQL que proporcionara los datos a las vistas para luego ser mostrados al usuario a través del grid conformado.

Modelado de la herramienta de administración para PostgreSQL

Diagrama de secuencia para el escenario Insertar Datos

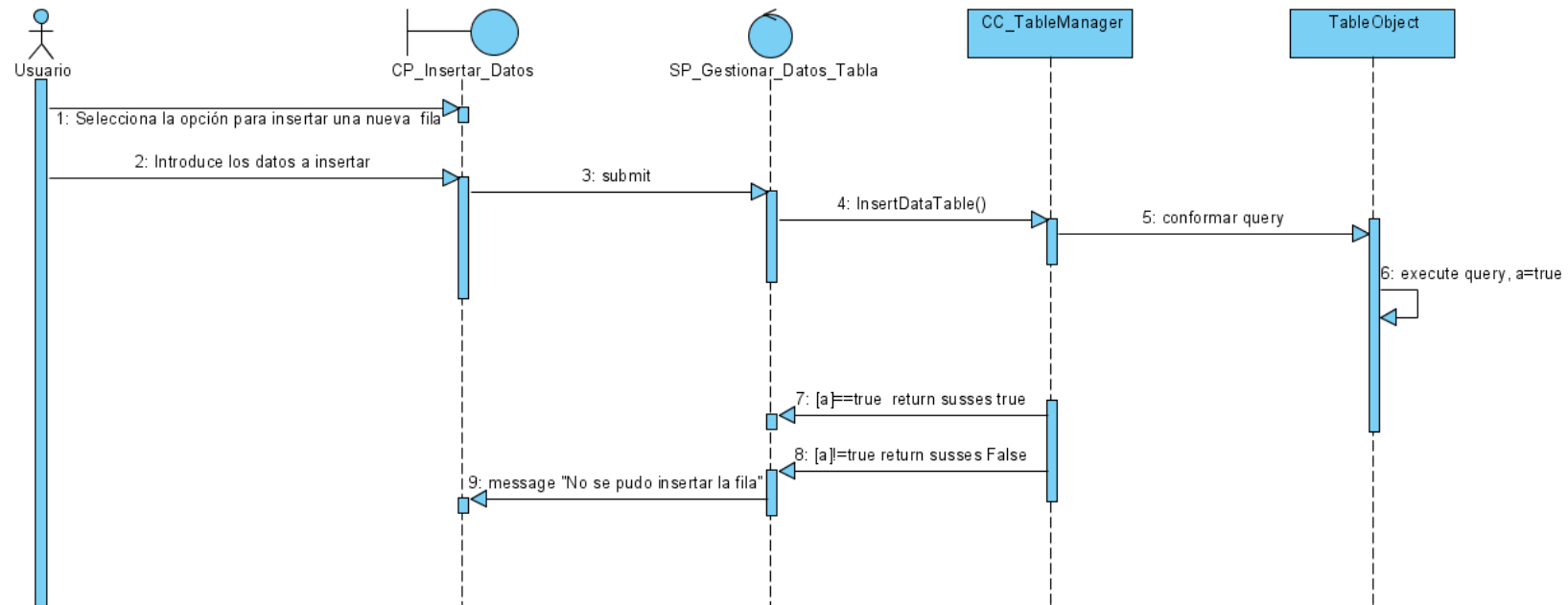


Figura 6: Diagrama de secuencia del caso de uso Gestionar Datos Tabla para la sección Insertar datos

En la figura anterior aparece reflejado el diagrama de secuencias del caso de uso Gestionar Datos Tabla, sección Insertar Datos, donde se representa al usuario que desea insertar datos en una tabla. Para ello se envía una solicitud a la vista de Django con los datos a insertar. La vista a su vez, envía a la clase controladora los datos obtenidos, quien solicita a la clase *TableObject* conformar la consulta SQL que permita insertar los nuevos datos en la tabla. En caso de que exista algún dato erróneo y la consulta no pueda ser ejecutada con éxito, la clase controladora envía a la vista un mensaje de error que será mostrado al usuario para que reinicie el proceso de inserción, en caso contrario los datos son insertados correctamente en la tabla.

Modelado de la herramienta de administración para PostgreSQL

Diagrama de secuencia para el escenario Modificar Datos

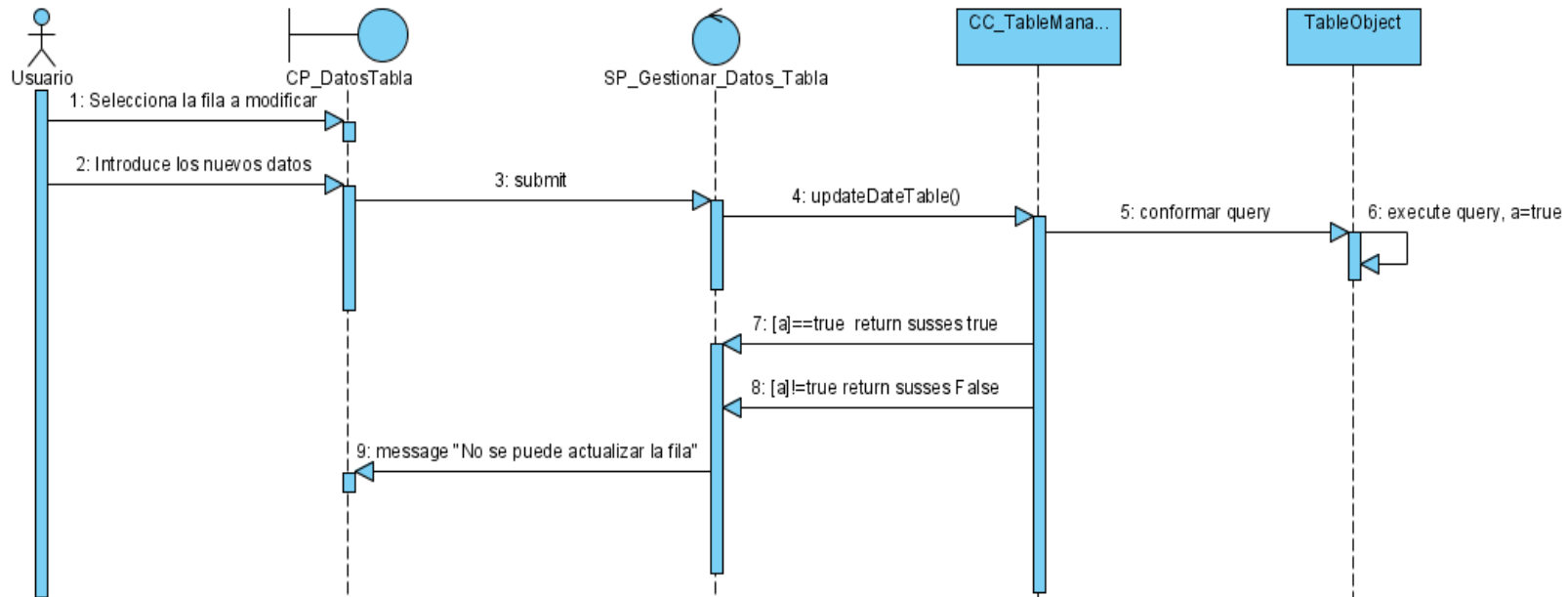


Figura 7: Diagrama de secuencia del caso de uso Gestionar Datos Tabla para la sección Modificar datos

En la figura anterior aparece reflejado el diagrama de secuencias del caso de uso Gestionar Datos Tabla, sección Modificar Datos, donde se representa al usuario que desea modificar datos en una tabla. Para ello se envía una solicitud a la vista de Django con los datos a modificar. La vista a su vez, envía a la clase controladora los datos obtenidos, quien solicita a la clase *TableObject* conformar la consulta SQL que permita la actualización de los nuevos datos en la tabla. En caso de que exista algún dato erróneo y la consulta no pueda ser ejecutada con éxito, la clase controladora envía a la vista un mensaje de error que será mostrado al usuario para que reinicie el proceso, en caso contrario los datos son actualizados correctamente en la tabla.

Modelado de la herramienta de administración para PostgreSQL

Diagrama de secuencia para el escenario Eliminar Datos

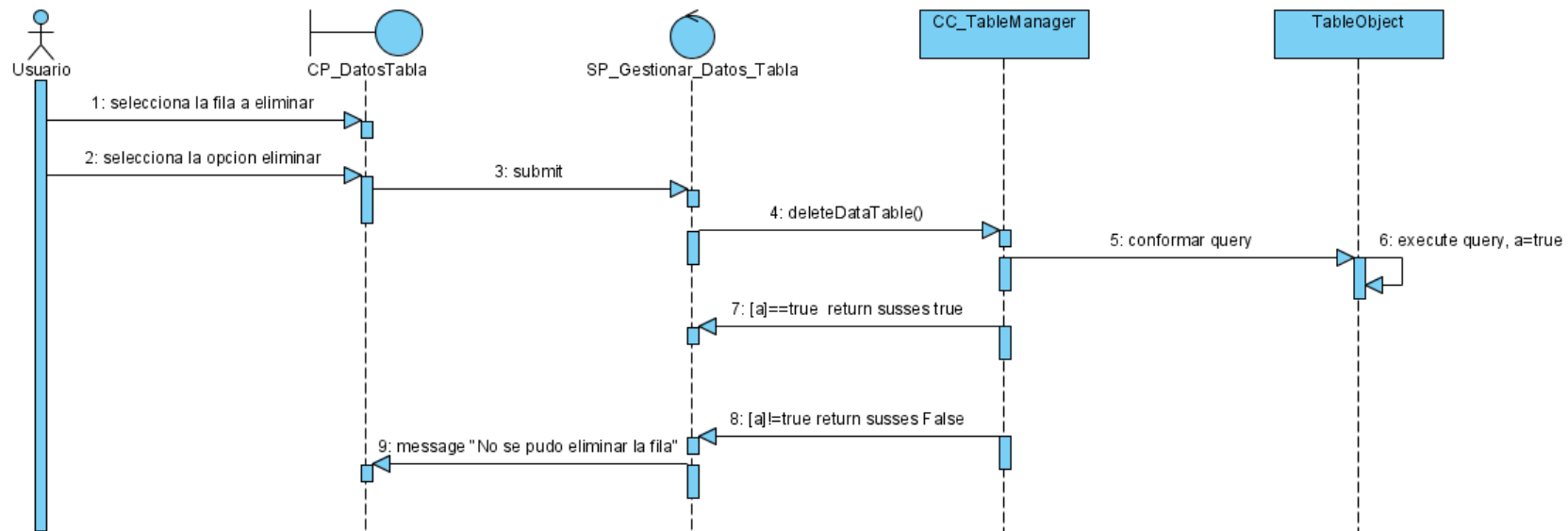


Figura 8: Diagrama de secuencia del caso de uso Gestionar Datos Tabla para la sección Eliminar datos

En la figura anterior aparece reflejado el diagrama de secuencias del caso de uso Gestionar Datos Tabla, sección Eliminar Datos, donde se representa al usuario que desea eliminar una fila de una tabla. Para ello se envía una solicitud a la vista de Django con los datos de la fila a eliminar. La vista a su vez, los envía a la clase controladora, quien solicita a la clase **TableObject** conformar la consulta SQL que permita eliminar una fila de una tabla. En caso de que exista alguna complicación y la consulta no pueda ser ejecutada con éxito, la clase controladora envía a la vista un mensaje de error que será mostrado al usuario para que reinicie el proceso, en caso contrario los datos son eliminados correctamente de la tabla.

Modelado de la herramienta de administración para PostgreSQL

2.5. Diseño de la base de datos

El diagrama de diseño de la base de datos es muy importante para el desarrollo de sistemas informáticos, pues muestra las tablas que van a contener la base de datos y las relaciones existentes entre ellas.

La herramienta de administración para PostgreSQL a desarrollar cuenta con una BD en SQLite, que contiene una sola tabla y su función es almacenar datos referentes a los servidores registrados en la aplicación *ver figura 9*. Se decidió emplear SQLite por su sencillez y rapidez en que almacena los datos.

SimpleServer		
+server_name	char(50)	Nullable = false
dns	char(50)	Nullable = true
port	integer(10)	Nullable = true
user	char(25)	Nullable = true
version	float(10)	Nullable = true

Figura 9: Modelo físico de datos de la aplicación

La herramienta de administración a parte de la base de datos SQLite, utiliza para obtener información de los objetos a gestionar, tablas del catálogo del gestor PostgreSQL. A continuación se muestran las tablas que conforman el catálogo de PostgreSQL y su función dentro del mismo.

Tabla 5: Funciones de las tablas del catálogo de PostgreSQL

Tablas del catálogo	Función
pg_aggregate	Almacena funciones de agregación. Son funciones que operan sobre un conjunto de valores. Por ejemplo <i>sum</i> , <i>count</i> y <i>max</i> .
pg_attrdef	Almacena los valores por defecto de las columnas
pg_attribute	Almacena las columnas de las tablas

Modelado de la herramienta de administración para PostgreSQL

pg_authid	Almacena los roles
pg_auth_members	Almacena las relaciones entre roles
pg_cast	Almacena los tipos de datos de conversiones
pg_class	Almacena las relaciones entre tablas, índices, secuencias y vistas.
pg_constraint	Almacena las restricciones check, único, llave primaria y llave foránea.
pg_database	Almacena las bases de datos
pg_depend	Almacena las dependencias entre los objetos de la base de datos
pg_index	Almacena información de los índices
pg_inherits	Almacena la herencia entre tablas
pg_language	Almacena los lenguajes para el desarrollo de funciones
pg_namespace	Almacena los esquemas
pg_operator	Almacena los operadores
pg_proc	Almacena funciones y procedimientos
pg_tablespace	Almacena los tablespaces
pg_trigger	Almacena los triggers
pg_type	Almacena los tipos de datos
pg_rules	Almacena las reglas

Las tablas del catálogo de PostgreSQL contienen una descripción completa de la estructura de la BD y sus restricciones. La información almacenada en el catálogo se denomina metadatos y su principal funcionalidad es la de conocer que datos existen sin acceder a ellos.

Modelado de la herramienta de administración para PostgreSQL

2.6. Conclusiones del capítulo

En este capítulo se realizó una completa descripción de la arquitectura de la herramienta de administración para PostgreSQL desarrollada, quedando representado a través de un diagrama la interacción entre las tecnologías empleadas en su desarrollo. Además se representó el diagrama de clases del sistema y se describieron dos casos de uso del mismo. También se representaron a través de un diagrama, las clases que intervienen en la herramienta, quedando descritas cada una de ellas.

Otra parte importante de este capítulo es donde se representa el modelo de la base de datos en SQLite, y el diagrama de tablas del catálogo de PostgreSQL empleado durante el desarrollo de la herramienta para obtener información de cada uno de los objetos gestionados.

CAPÍTULO III: Implementación y descripción de uso

En el desarrollo de un sistema informático, la etapa de implementación es de vital importancia, aquí es donde se concreta lo planificado durante toda la etapa de diseño y se obtiene como resultado un producto listo para ser probado y puesto en práctica.

En este capítulo se aborda el tema referente a la implementación del sistema, para ello se exponen los diagramas de componente, representando la implementación de las clases del diseño en términos de componentes y el modelo de despliegue, informando los nodos en que estará distribuida la aplicación. También se describen las funcionalidades que posee la herramienta a través de un manual de usuario, que brinda las facilidades necesarias para la familiarización de los usuarios con la aplicación.

3.1. Implementación

La implementación de un sistema informático es un proceso complejo en el que interactúan diversos factores tanto humanos como materiales, con el fin de alcanzar el objetivo trazado con éxito.

3.1.1. Diagrama de componentes

Un diagrama de componentes representa la separación de un sistema de software en componentes físicos (*por ejemplo archivos, cabeceras, módulos, paquetes, etc.*) y muestra las dependencias entre estos componentes.

A continuación se representan el diagramas de componentes para los CU *Mostrar Datos Tabla* y *Gestionar Datos Tabla*.

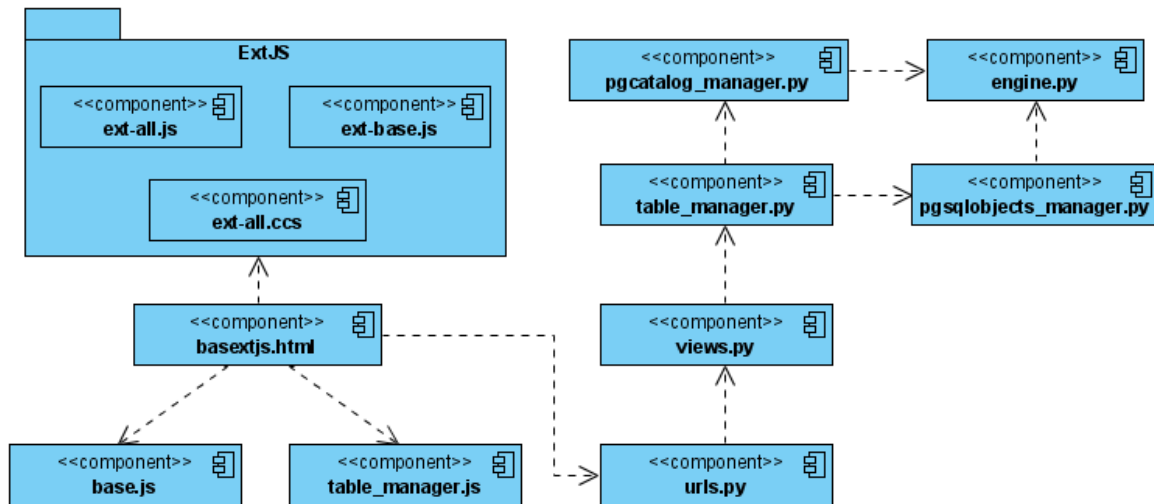


Figura 10: Diagrama de componentes para los CU Mostrar Datos Tabla y Gestionar Datos Tabla

En el diagrama representado en la **figura 10** aparecen encarnadas mediante el diagrama de componentes, las relaciones entre los ficheros que involucran los CU Mostrar Datos Tabla y Gestionar Datos Tabla.

A continuación se detallan cada uno de los componentes representados en el diagrama anterior:

- *Paquete ExtJS*: este paquete contiene la biblioteca JavaScript ExtJS que es utilizada durante el desarrollo de la herramienta para el diseño de las interfaces de usuario.
- *Componente baseextjs.html*: plantilla en *HTML* que contiene incluidos todos los ficheros *js* relacionados con las interfaces de usuario.
- *Componente base.js*: fichero *js* donde se encuentra implementada la interfaz principal de la de la herramienta, tomándose como plantilla para la creación de las vistas.
- *Componente table_manager.js*: este fichero contiene implementadas las funciones JavaScript necesarias para visualizar las interfaces de usuario de los casos de uso Mostrar datos Tabla y Gestionar Datos Tabla.

Implementación y descripción de uso

- *Componente urls.py*: fichero generado por el marco de trabajo Django, que cumple el rol de controlador, definiéndose en él todas las *urls* disponibles en el proyecto.
- *Componente views.py*: fichero generado por el marco de trabajo de desarrollo Web Django. Es válido aclarar que por cada aplicación creada dentro de un proyecto Django existe un fichero de este tipo. Su propósito es contener todas las vistas de la aplicación a la que pertenece, mostrándole al usuario las interfaces del proyecto.
- *Componente table_manager.py*: fichero en Python que contiene una clase controladora para gestionar todas las acciones relacionadas con las tablas dentro de la herramienta.
- *Componente engine.py*: fichero encargado de la conexión al gestor PostgreSQL a administrar en la herramienta.
- *Componente pgcatalog_manager.py*: fichero encargado de acceder a las tablas del catálogo del gestor PostgreSQL, para obtener información de los objetos.
- *Componente pgsqlobjects_manager.py*: fichero encargado de conformar las sentencias SQL para gestionar los objetos del gestor PostgreSQL, dentro de la herramienta.

3.1.2. Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas entre los distintos nodos que componen un sistema y la distribución de los componentes sobre dichos nodos. La vista de despliegue representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación.

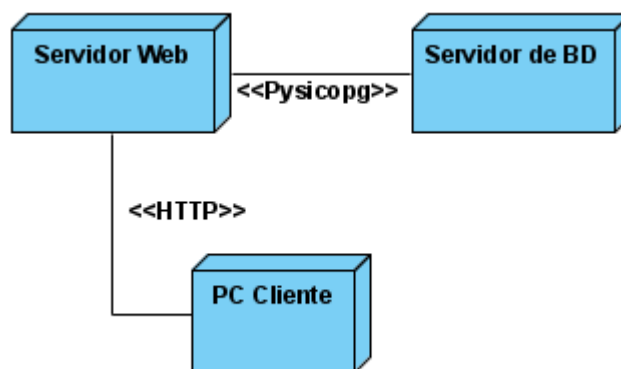
Implementación y descripción de uso

Figura 11: Diagrama de despliegue de la herramienta

A continuación se realiza una breve descripción del diagrama de despliegue representado en la **figura 11**:

Servidor Web: este nodo representa el lugar donde estará montada la aplicación Web sobre el servidor Apache. La cual se comunica con el Servidor de BD a través del driver <<Pysicpg>>.

Servidor de BD: En este nodo se encuentra el gestor de base dato PostgreSQL a administrar.

PC Cliente: Desde este nodo los usuarios podrán acceder a la aplicación a través de un navegador ya sea Internet Explorer o Firefox, haciendo uso del protocolo <<HTTP>>.

3.2. Esbozo de Manual de Usuario

La interfaz principal de la herramienta como puede observarse en la **figura 12** se encuentra estructurada de la siguiente forma:

- *Cabecera*: muestra el nombre de la aplicación y el botón Ayuda en la parte derecha.
- *Componentes*: muestra el explorador para acceder al árbol de componentes y un vínculo a la herramienta Symm Réplica desarrollada en el módulo réplica.
- *Área de trabajo*: contiene el menú principal de la herramienta.

Implementación y descripción de uso

- *Pie de página:* se muestran los temas Acerca de y Créditos.

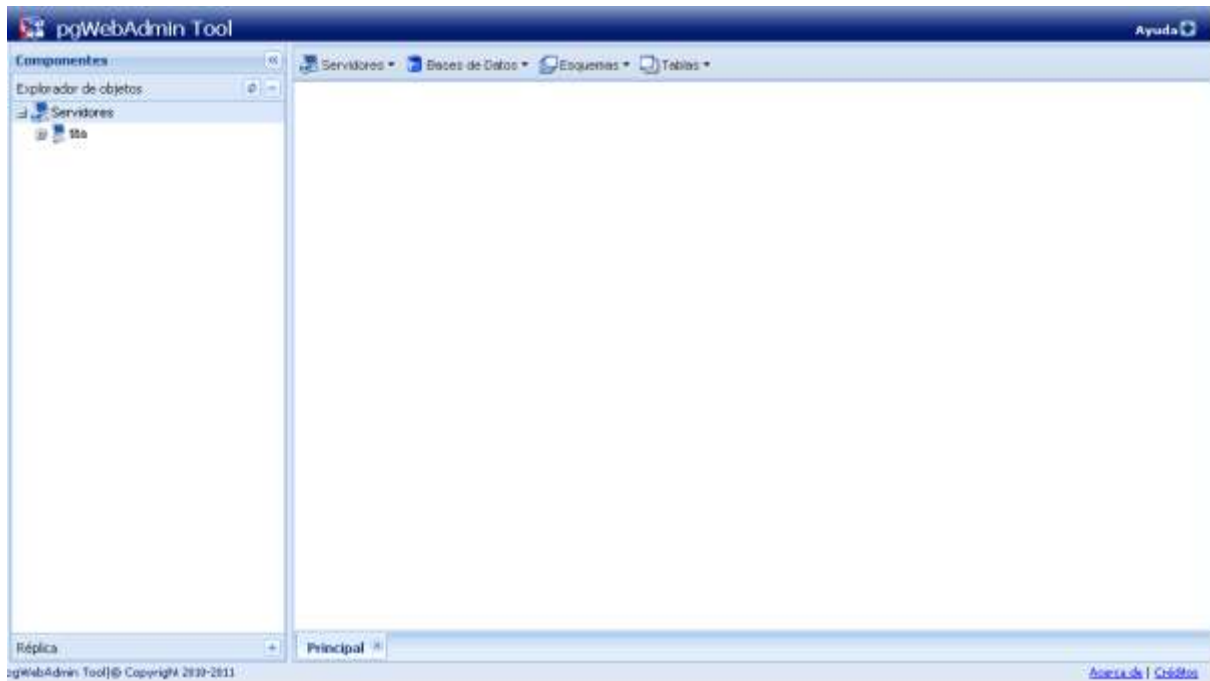


Figura 12: Interfaz principal de la herramienta

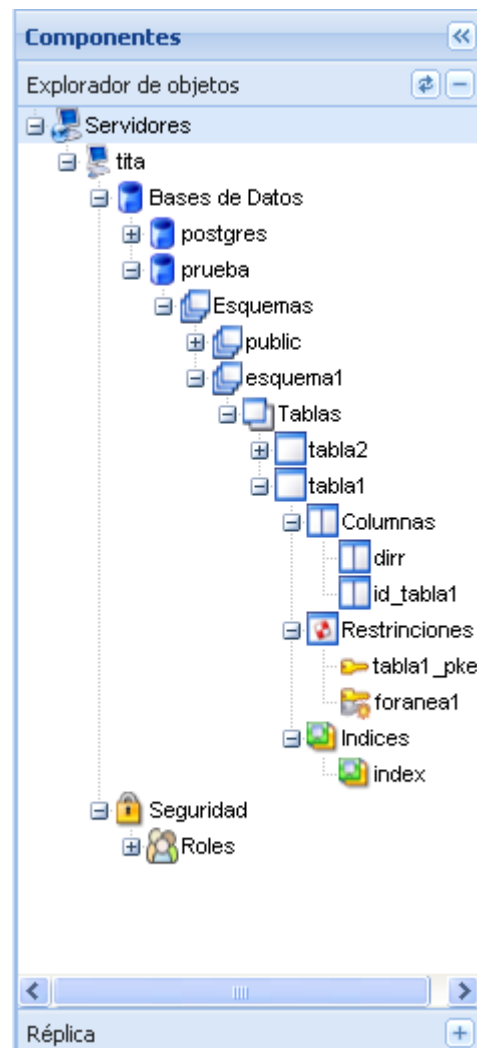
3.2.1. Estructura de la interfaz principal de la herramienta

A continuación se realiza una descripción detallada de las partes principales de la herramienta.

❖ Componentes

En esta parte se encuentra el explorador que contiene el árbol de los objetos a gestionar en la herramienta y un vínculo (*Réplica*) a la herramienta Symm Réplica desarrollada en el módulo réplica (*Msc. Reinier Castillo Figueroa 2011*).

En la **figura 13** aparece representado el árbol de componentes de la aplicación, donde se muestran los objetos a gestionar con la misma.

Implementación y descripción de uso**Figura 13: Explorador de objetos**

Cada uno de los componentes a gestionar tiene la opción de ser creados, editados y eliminados. Para acceder a las dos últimas opciones se debe dar clic derecho sobre el componente en particular que se desee gestionar, como se muestra en la *figura 14*.

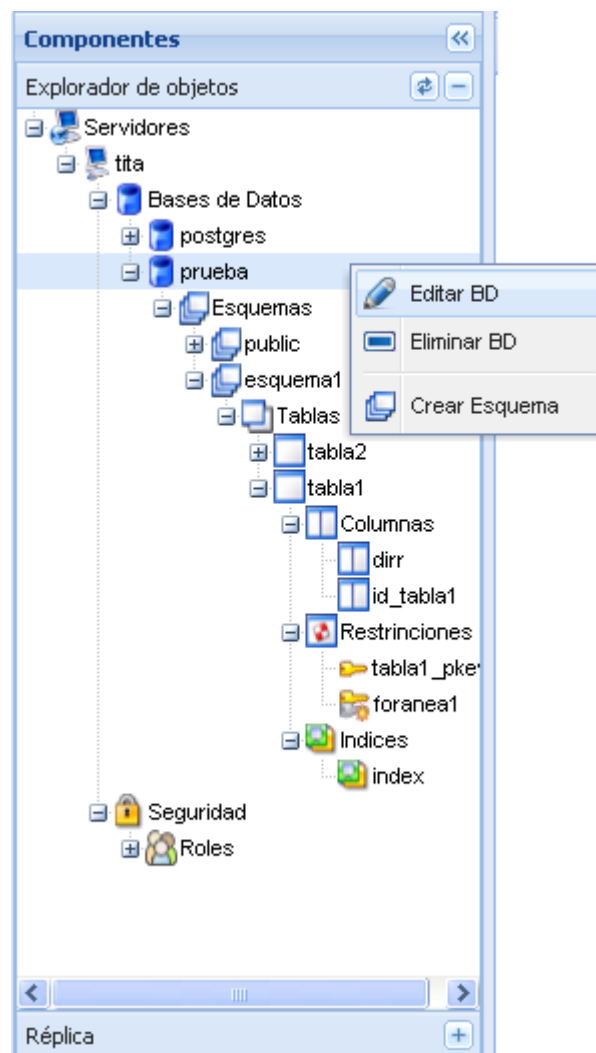


Figura 14: Clic derecho para editar o eliminar un componente

De igual forma para crear un nuevo componente se debe dar clic derecho sobre el componente padre, como se muestra en la *figura 15* para el caso de los componentes hijos de una tabla.

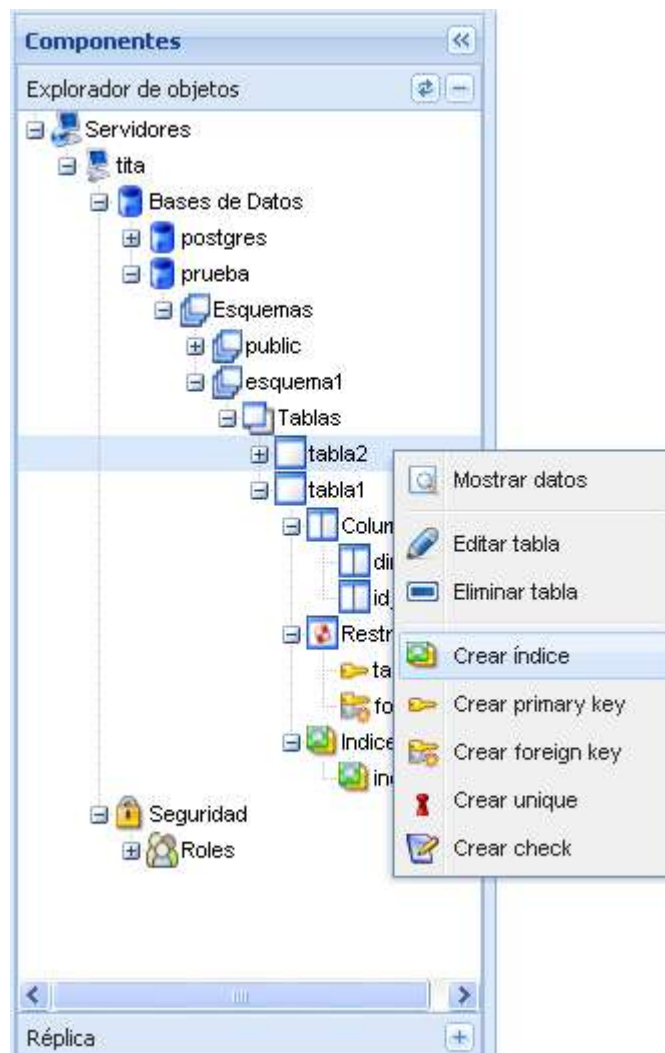


Figura 15: clic derecho para crear componentes

❖ Área de trabajo

En la parte superior del área de trabajo está ubicada la barra de menú de la aplicación que contiene el menú principal el cual está compuesto por:

- **Servidores:** como muestra la *figura 16*, este menú contiene el submenú para *Registrar*, *Modificar*, *Eliminar*, *Conectar* y *Desconectar* un servidor, siempre que para estas cuatro últimas opciones esté previamente seleccionado en el árbol de componentes un servidor en particular y de forma separada aparece un submenú que permite de igual forma *Crear*, *Modificar* y *Eliminar* roles dentro de la

Implementación y descripción de uso

herramienta. Con un servidor seleccionado en el árbol, se puede modificar, eliminar, conectar y desconectar el servidor y además se pueden crear roles y bases de datos.

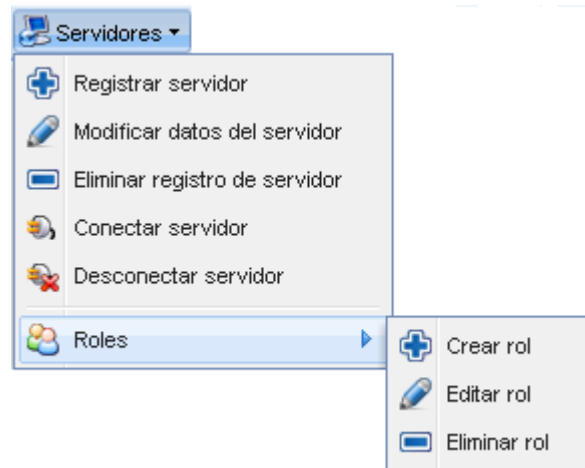


Figura 16: Submenú servidores

- **Bases de Datos:** como muestra la *figura 17*, este menú contiene el submenú para *Crear*, *Modificar* y *Eliminar* una base de datos, siempre que para estas dos últimas opciones esté previamente seleccionado en el árbol de componentes una base de datos en particular. Además, con una base de datos seleccionada en el árbol, se puede modificar y/o eliminar la misma.



Figura 17: Submenú base de datos

- **Esquemas:** como muestra la *figura 18*, este menú contiene el submenú para *Crear*, *Modificar* y *Eliminar* un esquema, siempre que para estas dos últimas opciones esté previamente seleccionado en el árbol de componentes un esquema en particular. Además, con un esquema seleccionado en el árbol, se puede modificar y/o eliminar el mismo.

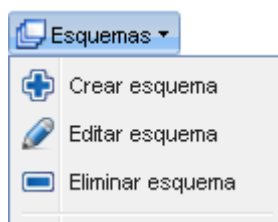
Implementación y descripción de uso

Figura 18: Submenú esquemas

- **Tablas:** como muestra la *figura 19*, este menú contiene el submenú para *Crear*, *Mostrar Datos*, *Modificar* y *Eliminar* una tabla, siempre que para estas tres últimas opciones esté previamente seleccionado en el árbol de componentes una tabla en particular y de forma separada aparece un submenú que permite de igual forma *Crear*, *Modificar* y *Eliminar* índices y restricciones (Llave primaria, Llave foránea, Único y Check) dentro de la herramienta. Con una tabla seleccionada en el árbol, se puede modificar, eliminar y mostrar datos de la misma y además se pueden crear índices y restricciones






Figura 19: Submenú tablas

3.2.2. Interfaces de usuario de los componentes

Con el objetivo de permitirle al usuario un trabajo fácil e interactivo con la aplicación se han diseñado las interfaces con varios componentes en común, como se describen a continuación:

Implementación y descripción de uso

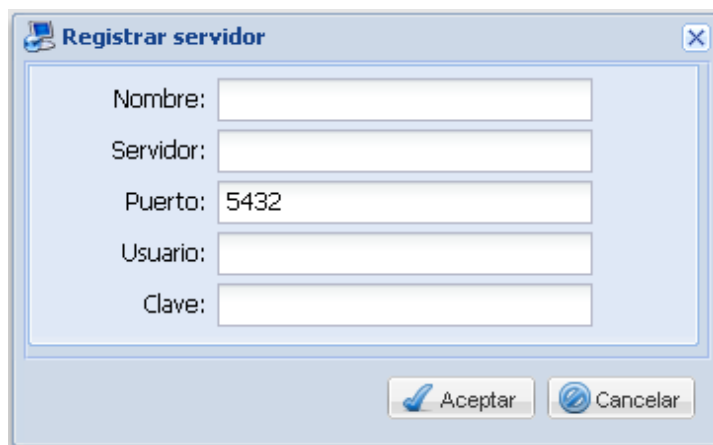
- Botón *Cerrar* (): permite cerrar cualquier ventana, se hayan o no introducido datos.
- Botón *Aceptar* (): permite guardar los datos introducidos por el usuario y mostrar cualquier validación referente a estos datos.
- Botón *Cancelar* (): permite cancelar los datos introducidos por el usuario y cierra la ventana.

Interfaces de usuario para el componente servidor

- Para **Registrar un nuevo servidor**, se debe hacer clic en la opción *Registrar servidor* del submenú *Servidores* o haciendo clic derecho sobre el componente padre *Servidores* en el árbol del *Explorador*. **Ver figura 20.**

Los datos necesarios para registrar un nuevo servidor son:

- ✓ *Nombre*: que debe ser escrito de forma obligatoria.
- ✓ *Servidor*: que debe ser escrito de forma obligatoria.
- ✓ *Puerto*: que viene por defecto el 5432 y debe ser escrito de forma obligatoria.
- ✓ *Usuario*: debe ser escrito de forma obligatoria.
- ✓ *Clave*: debe ser escrita de forma obligatoria.



The image shows a Windows-style dialog box titled "Registrar servidor". It has a standard title bar with a minimize, maximize, and close button. The main area contains five labeled text input fields stacked vertically: "Nombre:", "Servidor:", "Puerto:", "Usuario:", and "Clave:". The "Puerto:" field has the number "5432" entered. At the bottom right of the dialog, there are two buttons: "Aceptar" (with a checkmark icon) and "Cancelar" (with a circular arrow icon).

Figura 20: Interfaz para registrar un servidor

Implementación y descripción de uso

Una vez llenados los campos, se hace clic en el botón *Aceptar*, para completar el registro del nuevo servidor.

- Para **Modificar datos del servidor** se debe hacer clic en la opción *Modificar datos del servidor* del submenú *Servidores*, habiendo seleccionado previamente en el árbol de componentes un servidor en particular, o haciendo clic derecho sobre el servidor deseado en el árbol del *Explorador*. La interfaz que se muestra es igual a la interfaz de la **figura 20** pero con los datos que posee el servidor seleccionado.
- Para **Eliminar registro de servidor**, se debe hacer clic en la opción *Eliminar* del submenú *Servidores* habiendo seleccionado previamente en el árbol de componentes un servidor en particular o haciendo clic derecho sobre el servidor deseado en el árbol del *Explorador*. Para completar esta acción aparece un mensaje de aviso donde se debe seleccionar la opción *Si* o *No*.
- Para **Conectarse a un servidor**, se debe hacer clic en la opción *Conectar servidor* del submenú *Servidores* habiendo seleccionado previamente en el árbol de componentes un servidor en particular o haciendo clic derecho sobre el servidor deseado en el árbol del *Explorador*. **Ver figura 21.**

Los datos necesarios para la conexión al servidor son:

- ✓ *Usuario*: debe ser escrito de forma obligatoria.
- ✓ *Clave*: debe ser escrita de forma obligatoria.



Figura 21: Interfaz para conectarse a un servidor

Implementación y descripción de uso

Una vez llenados los campos, se hace clic en el botón *Aceptar*, para conectarse al servidor deseado.

- Para **desconectar un servidor**, se debe hacer clic en la opción *Desconectar servidor* del submenú *Servidores* habiendo seleccionado previamente en el árbol de componentes un servidor en particular o haciendo clic derecho sobre el servidor deseado en el árbol del *Explorador*. Para completar esta acción aparece un mensaje de aviso donde se debe seleccionar la opción *Si* o *No*.

Interfaces de usuario para el componente roles

- Para **Crear un nuevo rol**, se debe hacer clic en la opción *Crear rol* del submenú *Servidores* o haciendo clic derecho sobre un servidor en el árbol del *Explorador*. La interfaz *Crear rol* consta de dos pestañas *Propiedades* y *Membrecía* para más información **Ver figuras 22 y 23**.

Los datos necesarios para la pestaña propiedades son los siguientes:

- ✓ *Nombre*: que debe ser escrito de forma obligatoria.
- ✓ *Contraseña*: que debe ser escrito de forma obligatoria.
- ✓ *Confirmar*: que es para confirmar la contraseña y que debe ser escrito de forma obligatoria
- ✓ *Privilegios del rol*: que permiten ser marcados en dependencia de los permisos que tenga el mismo.
- ✓ *Límite de conexiones*: que es opcional.
- ✓ *Comentario*: que es opcional.
- ✓ *Opción puede conectarse*: que viene marcado por defecto.
- ✓ *Opción caducidad de la cuenta*: que permiten ser marcada si caduca la cuenta.

The screenshot shows the 'Crear rol' (Create role) dialog box with the 'Propiedades' (Properties) tab selected. The dialog has two tabs: 'Propiedades' and 'Membrecía'. Under 'Propiedades', there are three text input fields for 'Nombre:', 'Contraseña:', and 'Confirmar:'. To the right of these fields are two checkboxes: 'Puede conectarse.' (checked) and 'La cuenta caduca.' (unchecked). Below these is a section titled 'Privilegios del rol' (Role privileges) containing five checkboxes: 'Hereda permisos de los roles padres.' (unchecked), 'Superusuario.' (unchecked), 'Puede crear objetos de base de datos.' (unchecked), 'Puede modificar el catálogo.' (unchecked), and 'Puede crear roles.' (unchecked). At the bottom of the 'Propiedades' section are two more input fields: 'Límite conex.: ' and 'Comentario: '. At the very bottom of the dialog are 'Aceptar' (Accept) and 'Cancelar' (Cancel) buttons.

Figura 22: Interfaz para crear un rol (Pestaña propiedades)

En la pestaña membrecía es donde se especifica si un rol es miembro de otro rol o de un grupo específico y además permite marcar si el rol tiene opciones de administración o no. *Ver figura 23.*

The screenshot shows the 'Crear rol' dialog box with the 'Membrecía' (Membership) tab selected. The dialog has two tabs: 'Propiedades' and 'Membrecía'. It features two list boxes. The left list box is titled 'No es miembro de:' and contains three entries: 'lis', 'prueba', and 'Titi'. The right list box is titled 'Es miembro de:' and contains one entry: 'postgres'. Between the two list boxes are two arrow buttons: a right-pointing arrow (>) and a left-pointing arrow (<). Below the list boxes is a checkbox labeled 'Con opciones de administración' (checked). At the bottom of the dialog are 'Aceptar' (Accept) and 'Cancelar' (Cancel) buttons.

Figura 23: Interfaz para crear un rol (Pestaña membrecía)

Implementación y descripción de uso

Una vez llenados los campos, se hace clic en el botón *Aceptar*, para crear el nuevo rol.

- Para **Modificar un rol** se debe hacer clic en la opción *Modificar rol* del submenú *Servidores* habiendo seleccionado previamente en el árbol de componentes un rol en particular o haciendo clic derecho sobre el rol deseado en el árbol del *Explorador*. La interfaz que se muestra es igual a la interfaz de las **figuras 23 y 24** pero con los datos que posee el rol seleccionado.
- Para **Eliminar un rol**, se debe hacer clic en la opción *Eliminar* del submenú *Servidores* habiendo seleccionado previamente en el árbol de componentes un rol en particular o haciendo clic derecho sobre el rol deseado en el árbol del *Explorador*. Para completar esta acción aparece un mensaje de aviso donde se debe seleccionar la opción *Si* o *No*.

Interfaces de usuario para el componente base de datos

- Para **Crear una nueva base de datos**, se debe hacer clic en la opción *Crear base de datos* del submenú *Base de datos* o haciendo clic derecho sobre un servidor en el árbol del *Explorador*. **Ver figuras 24.**

Los datos necesarios para crear una nueva base de datos son:

- ✓ *Nombre*: que debe ser escrito de forma obligatoria.
- ✓ *Tablespace*: que no es obligatoria su selección pues en caso de no seleccionarse toma por defecto *pg_default*.
- ✓ *Propietario*: que no es obligatoria su selección pues en caso de no seleccionarse toma por defecto *postgres*.
- ✓ *Plantilla*: que no es obligatoria su selección pues en caso de no seleccionarse toma por defecto *default*.
- ✓ *Comentario*: que no es obligatorio.

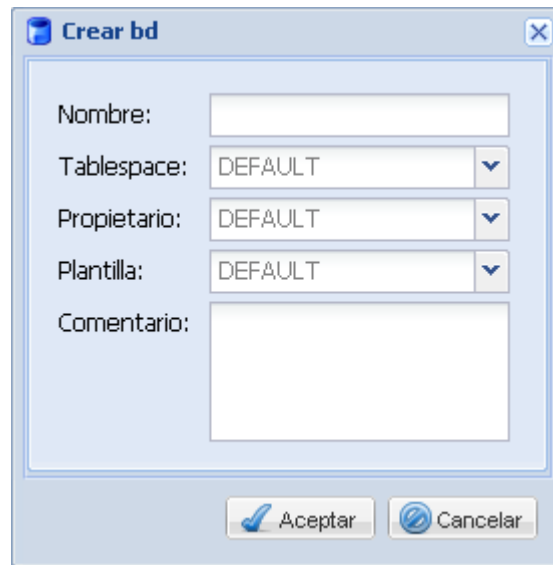
Implementación y descripción de uso

Figura 24: Interfaz para crear base de datos

Una vez llenados los campos, se hace clic en el botón *Aceptar*, para crear la nueva base de datos.

- Para ***Modificar una base de datos*** se debe hacer clic en la opción *Modificar base de datos* del submenú *Base de datos* habiendo seleccionado previamente en el árbol de componentes una base de datos en particular o haciendo clic derecho sobre la base de datos deseada en el árbol del Explorador. La interfaz que se muestra es igual a la interfaz de la **figura 24** pero con los datos que posee la base de datos seleccionada.
- Para ***Eliminar una base de datos***, se debe hacer clic en la opción *Eliminar base de datos* del submenú *Base de datos* habiendo seleccionado previamente en el árbol de componentes una base de datos en particular o haciendo clic derecho sobre la base de datos deseada en el árbol del Explorador. Para completar esta acción aparece un mensaje de aviso donde se debe seleccionar la opción *Si* o *No*.

Interfaces de usuario para el componente esquema

- Para ***Crear un nuevo esquema***, se debe hacer clic en la opción *Crear esquema* del submenú *Esquemas* o haciendo clic derecho sobre una base de datos en el árbol del Explorador. *Ver figuras 25*

Implementación y descripción de uso

Los datos necesarios para crear un nuevo esquema son:

- ✓ *Nombre*: que debe ser escrito de forma obligatoria.
- ✓ *Propietario*: que no es obligatoria su selección pues en caso de no seleccionarse toma por defecto postgres.
- ✓ *Comentario*: que no es obligatorio.

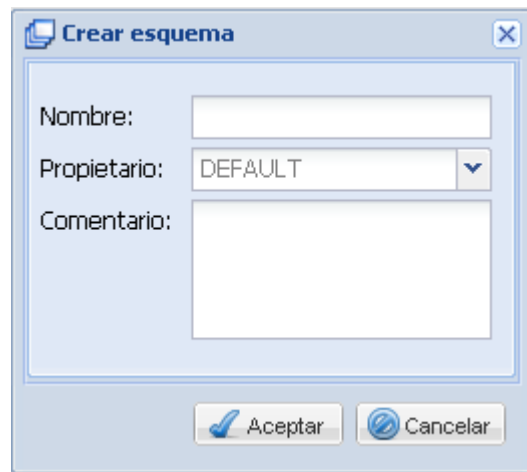


Figura 25: Interfaz para crear esquema

Una vez llenados los campos, se hace clic en el botón *Aceptar*, para crear el nuevo esquema.

- Para ***Modificar esquema*** se debe hacer clic en la opción *Modificar esquema* del submenú *Esquemas* habiendo seleccionado previamente en el árbol de componentes un esquema en particular o haciendo clic derecho sobre el esquema deseado en el árbol del Explorador. La interfaz que se muestra es igual a la interfaz de la **figura 25** pero con los datos que posee el esquema seleccionado.
- Para ***Eliminar un esquema***, se debe hacer clic en la opción *Eliminar esquema* del submenú *Esquemas* habiendo seleccionado previamente en el árbol de componentes un esquema en particular o haciendo clic derecho sobre el esquema deseado en el árbol del Explorador. Para completar esta acción aparece un mensaje de aviso donde se debe seleccionar la opción *Si* o *No*.

*Implementación y descripción de uso**Interfaces de usuario para el componente tablas*

- Para **Crear una nueva tabla**, se debe hacer clic en la opción *Crear tabla* del submenú *Tablas* o haciendo clic derecho sobre un esquema en el árbol del *Explorador*. La interfaz *Crear tabla* consta de dos pestañas *Propiedades* y *Columnas* para más información ver *figuras 26 y 27*

Los datos necesarios para la pestaña propiedades son los siguientes:

- ✓ *Nombre*: que debe ser escrito de forma obligatoria.
- ✓ *Propietario*: que no es obligatoria su selección pues en caso de no seleccionarse toma por defecto postgres.
- ✓ *Tablespace*: que no es obligatoria su selección pues en caso de no seleccionarse toma por defecto *pg_default*.
- ✓ *Comentario*: que es opcional.
- ✓ *Hereda de*: que agregar las tablas de las que hereda, además es opcional.

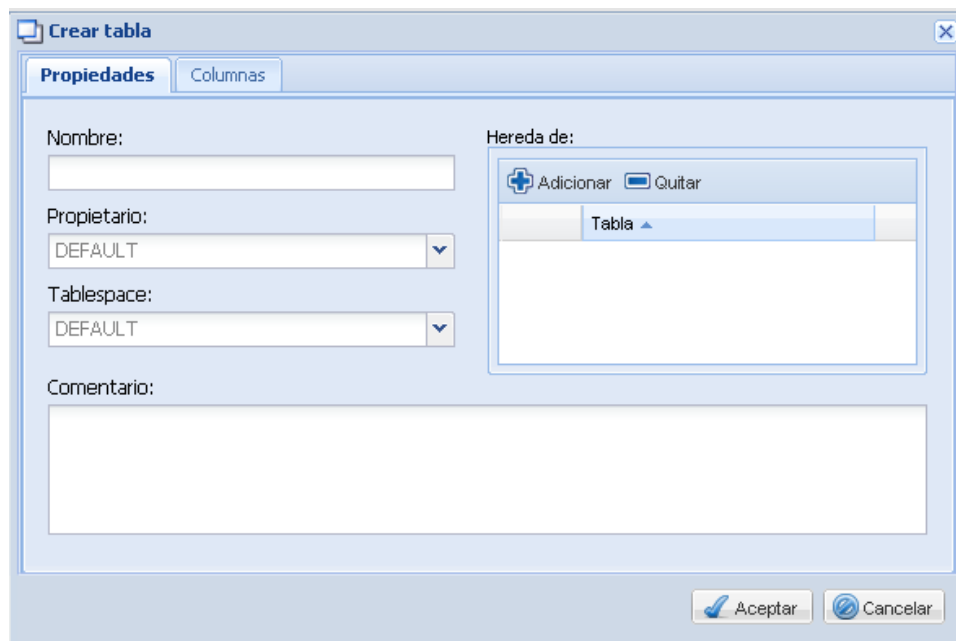


Figura 26: Interfaz para crear una tabla (Pestaña propiedades)

Implementación y descripción de uso

En la pestaña columnas es donde se crean o eliminan las columnas de una tabla, además se especifican el nombre, tipo de datos, si es no nulo, la longitud, el valor por defecto que debe tener en caso de necesitarse y si es llave primaria o no. *Ver figura 27.*

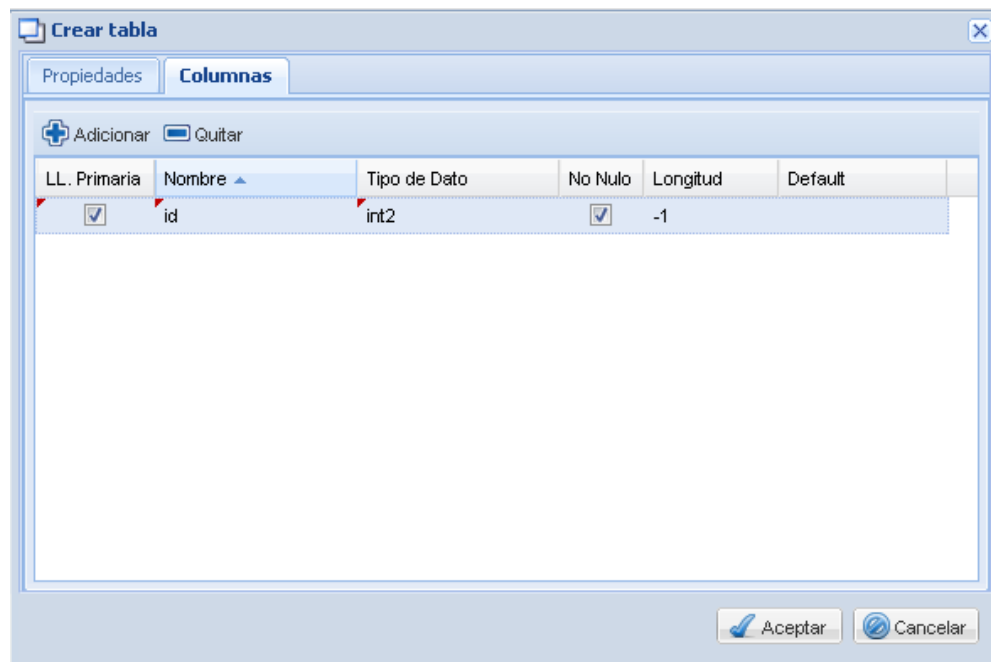


Figura 27: Interfaz para crear una tabla (Pestaña columnas)

Una vez llenados los campos, se hace clic en el botón *Aceptar*, para crear la nueva tabla.

- Para **Modificar una tabla** se debe hacer clic en la opción *Modificar tabla* del submenú *Tablas* habiendo seleccionado previamente en el árbol de componentes una tabla en particular o haciendo clic derecho sobre la tabla deseada en el árbol del Explorador. La interfaz que se muestra es igual a la interfaz de las *figuras 26 y 27* pero con los datos que posee la tabla seleccionada.
- Para **Eliminar una tabla**, se debe hacer clic en la opción *Eliminar tabla* del submenú *Tablas* habiendo seleccionado previamente en el árbol de componentes una tabla en particular o haciendo clic derecho sobre la tabla deseada en el árbol del

Implementación y descripción de uso

Explorador. Para completar esta acción aparece un mensaje de aviso donde se debe seleccionar la opción *Si* o *No*.

- Para **Visualizar los datos de una tabla** se debe hacer clic en la opción *Mostrar datos* del submenú *Tablas* habiendo seleccionado previamente en el árbol de componentes una tabla en particular o haciendo clic derecho sobre la tabla deseada en el árbol del Explorador. *Ver figura 28*

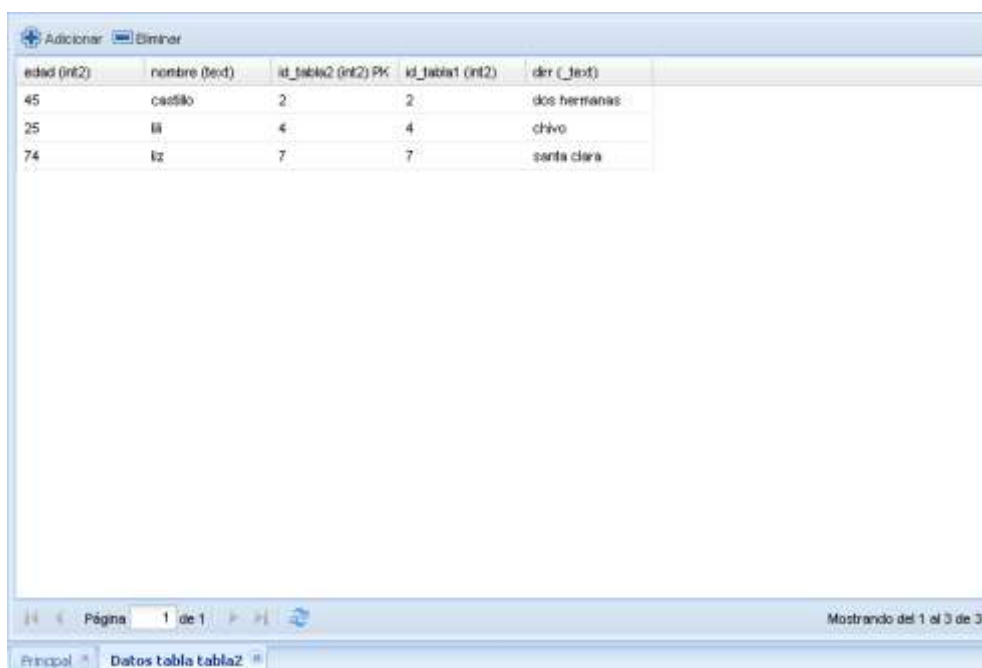



Figura 28: Interfaz para visualizar datos de una tabla

- Para **Insertar datos en una tabla** se debe hacer clic en la opción ( Adicionar) de la interfaz Visualizar datos de una tabla. *Ver figura 29*

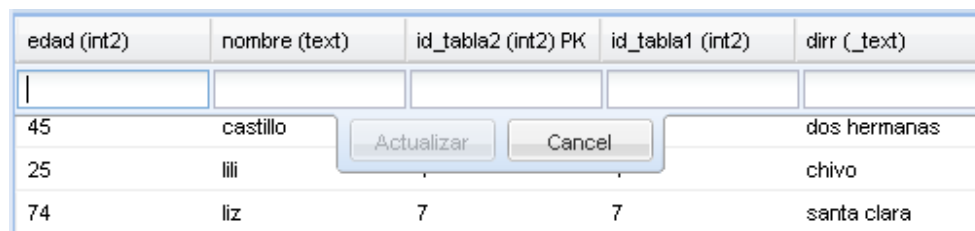






Figura 29: Interfaz para insertar datos en una tabla

Implementación y descripción de uso

Se introducen los datos necesarios para crear la nueva fila de la tabla y se hace clic en el botón Aceptar, para crear la nueva fila. Para ver los cambios realizados en la tabla se da clic en el ícono actualizar ().

- Para **Modificar datos en una tabla** se debe hacer doble clic sobre la fila que se desea modificar. La interfaz que se muestra es igual a la interfaz de la **figuras 29** pero con los datos que posee la fila seleccionada y de igual forma para ver los cambios realizados en la tabla se da clic en el ícono actualizar ().
- Para **Eliminar una fila de una tabla** se debe hacer clic en la fila deseada y luego en la opción () de la interfaz Visualizar datos de una tabla. Para ver los cambios realizados en la tabla se da clic en el ícono actualizar ().

Interfaces de usuario para el componente índice

- Para **Crear un nuevo índice**, se debe hacer clic en la opción *Crear índice* del submenú *Tablas* o haciendo clic derecho sobre una tabla en el árbol del *Explorador*.
Ver figuras 30

Los datos necesarios para crear un nuevo índice son:

- ✓ *Nombre*: que debe ser escrito de forma obligatoria.
- ✓ *Tablespace*: que no es obligatoria su selección pues en caso de no seleccionarse toma por defecto *pg_default*.
- ✓ *Tipo*: que no es obligatoria su selección pues en caso de no seleccionarse toma por defecto *btree*.
- ✓ *Comentario*: que no es obligatorio.
- ✓ *Condición*: que no es obligatoria.
- ✓ *Las columnas* donde se va a aplicar el índice creado.

Implementación y descripción de uso

Una vez llenados los campos, se hace clic en el botón *Aceptar*, para crear el nuevo índice.

Crear índice

Propiedades

Nombre:

Tablespace:

Tipo:

☐ Concurr. ☐ Único

Comentario:

Condición:

+ Adicionar - Quitar

Columnas	Ascendente	Primero Nulo

Aceptar Cancelar

Figura 30: Interfaz para crear un nuevo índice

- Para ***Eliminar un índice***, se debe hacer clic en la opción *Eliminar índice* del submenú *Tablas* habiendo seleccionado previamente en el árbol de componentes un índice en particular o haciendo clic derecho sobre el índice deseado en el árbol del *Explorador*. Para completar esta acción aparece un mensaje de aviso donde se debe seleccionar la opción *Si* o *No*.

*Implementación y descripción de uso**Interfaces de usuario para el componente restricciones***Llave primaria**

- Para **Crear una llave primaria**, se debe hacer clic en la opción *Crear llave primaria* del submenú *Tablas* o haciendo clic derecho sobre una tabla en el árbol del *Explorador*. La interfaz que se muestra es igual a la interfaz de la **figura 27** pero con los datos que posee la tabla seleccionada y brindando la opción de marcar el campo que desea hacer llave primaria.
- Para **Eliminar una llave primaria**, se debe hacer clic en la opción *Eliminar llave primaria* del submenú *Tablas* habiendo seleccionado previamente en el árbol de componentes una llave primaria en particular o haciendo clic derecho sobre la llave primaria deseada en el árbol del *Explorador*. Para completar esta acción aparece un mensaje de aviso donde se debe seleccionar la opción *Si* o *No*.

Llave foránea

- Para **Crear una llave foránea**, se debe hacer clic en la opción *Crear llave foránea* del submenú *Tablas* o haciendo clic derecho sobre una tabla en el árbol del *Explorador*. La interfaz *Crear llave foránea* consta de dos pestañas *Propiedades* y *Columnas* para más información ver **figuras 31 y 32**

Los datos necesarios para la pestaña propiedades son los siguientes:

- ✓ *Nombre*: que debe ser escrito de forma obligatoria.
- ✓ *Referencia*: que es obligatoria su selección.
- ✓ *Update*: que es obligatoria su selección.
- ✓ Las opciones *posponible* y *pospuesto*
- ✓ *Comentario*: que no es obligatorio.

Implementación y descripción de uso

The screenshot shows the 'Crear llave foránea' dialog box with the 'Llave foránea' tab selected. The dialog has a title bar with a close button. Below the title bar are two tabs: 'Llave foránea' (selected) and 'Columnas'. The 'Llave foránea' tab contains the following fields and options:

- Nombre: A text input field.
- Referencia: A dropdown menu with 'DEFAULT' selected.
- UPDATE: A dropdown menu with 'Seleccione' selected.
- DELETE: A dropdown menu with 'Seleccione' selected.
- Options: Three checkboxes labeled 'Posponible', 'Pospuesto', and 'Coínc.comp'.
- Comentario: A large text area.
- Buttons: 'Aceptar' and 'Cancelar' at the bottom.

Figura 31: Interfaz para crear una llave foránea (Pestaña propiedades)

En la pestaña columnas es donde se seleccionan las columnas de las tablas donde será creada la llave foránea. *Ver figura 32.*

The screenshot shows the 'Crear llave foránea' dialog box with the 'Columnas' tab selected. The dialog has a title bar with a close button. Below the title bar are two tabs: 'Llave foránea' and 'Columnas' (selected). The 'Columnas' tab contains the following elements:

- Buttons: '+ Adicionar' and '- Quitar' at the top.
- Table: A table with two columns, 'Col.Local' and 'Col.Referencia'. The first row contains 'id_tabla1' in both columns.
- Buttons: 'Aceptar' and 'Cancelar' at the bottom.

Col.Local	Col.Referencia
id_tabla1	id_tabla1

Figura 32: Interfaz para crear una llave foránea (Pestaña columnas)

Implementación y descripción de uso

Una vez llenados los campos, se hace clic en el botón *Aceptar*, para crear la nueva llave foránea.

- Para ***Eliminar una llave foránea***, se debe hacer clic en la opción *Eliminar llave foránea* del submenú *Tablas* habiendo seleccionado previamente en el árbol de componentes una llave foránea en particular o haciendo clic derecho sobre la llave foránea deseada en el árbol del *Explorador*. Para completar esta acción aparece un mensaje de aviso donde se debe seleccionar la opción *Si* o *No*.

Único

- Para ***Crear una restricción único***, se debe hacer clic en la opción *Crear único* del submenú *Tablas* o haciendo clic derecho sobre una tabla en el árbol del *Explorador*.

Ver figuras 33

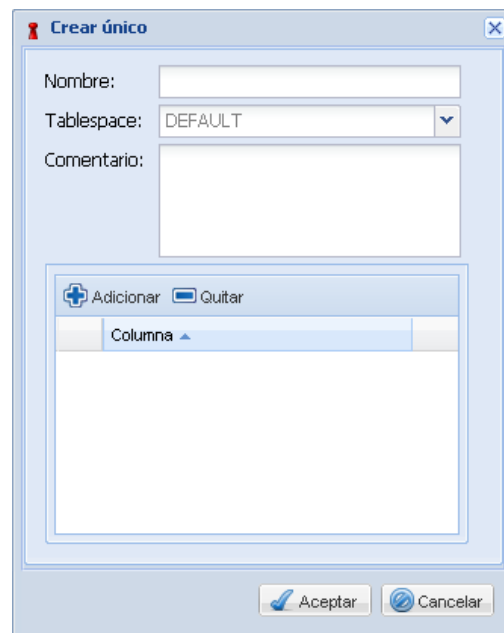


Figura 33: Interfaz para crear una restricción único

Los datos necesarios para crear una restricción única son los siguientes:

- ✓ *Nombre*: que debe ser escrito de forma obligatoria.
- ✓ *Tablespace*: que no es obligatoria su selección pues en caso de no

Implementación y descripción de uso

seleccionarse toma por defecto *pg_default*.

- ✓ *Comentario*: que no es obligatorio.
- ✓ *Columnas* donde se aplica esta restricción, además es obligatoria su selección.

Una vez llenados los campos, se hace clic en el botón Aceptar, para crear la nueva restricción único.

- Para ***Eliminar restricción único***, se debe hacer clic en la opción *Eliminar único* del submenú *Tablas* habiendo seleccionado previamente en el árbol de componentes una restricción único en particular o haciendo clic derecho sobre la restricción único deseada en el árbol del *Explorador*. Para completar esta acción aparece un mensaje de aviso donde se debe seleccionar la opción *Si* o *No*.

Check

- Para ***Crear un check***, se debe hacer clic en la opción *Crear check* del submenú *Tablas* o haciendo clic derecho sobre una tabla en el árbol del *Explorador*. ***Ver figuras 34***

Figura 34: Interfaz para crear un check

Implementación y descripción de uso

Los datos necesarios para crear un check son los siguientes:

- ✓ *Nombre:* que debe ser escrito de forma obligatoria.
 - ✓ *Condición:* que debe ser escrito de forma obligatoria.
 - ✓ *Comentario:* que no es obligatorio.
- Para ***Eliminar un check***, se debe hacer clic en la opción *Eliminar check* del submenú *Tablas* habiendo seleccionado previamente en el árbol de componentes un check en particular o haciendo clic derecho sobre el check deseado en el árbol del Explorador. Para completar esta acción aparece un mensaje de aviso donde se debe seleccionar la opción *Si* o *No*.

3.3. Conclusiones del capítulo

En este capítulo quedó representada la implementación del sistema a través del diagrama de componentes para los CU Mostrar Datos Tabla y Gestionar Datos Tabla, quedando representada la interacción entre los ficheros físicos de la aplicación. También se detalla el diagrama de despliegue que servirá de apoyo para la implantación y se describen las funcionalidades de la herramienta a través de un esbozo de manual de usuario que proveerá una mayor comprensión de la aplicación por parte de los usuarios.

CONCLUSIONES

Como resultado de la investigación se obtuvo una versión 1.0 de una herramienta Web libre para la administración del gestor de base de datos PostgreSQL.

Además se obtuvieron los siguientes resultados:

- Se definieron las tecnologías de software libre apropiadas para el desarrollo de la herramienta de administración para PostgreSQL que son: la biblioteca ExtJS para elaborar las interfaces de usuario, auxiliándose de CSS, JavaScript y HTML, el framework Django y el lenguaje de programación Python para el lado del servidor, la herramienta Visual Paradigm para el modelado UML, el IDE Eclipse para facilitar la programación de la herramienta, el servidor Web Apache para la publicación de la aplicación Web y el gestor de base de datos SQLite para almacenar el registro de los servidores de base de datos PostgreSQL a gestionar en la herramienta.
- Se caracterizaron múltiples herramientas libres de administración para PostgreSQL, expresándose sus características y deficiencias. Decidiéndose realizar una nueva herramienta sobre Web 2.0 que contenga interfaces de usuario interactivas y de fácil uso, que reúna los principales requisitos de las herramientas analizadas y además permita la administración y visualización de datos espaciales, la migración y replica de datos y el monitoreo del gestor, todo incluido en una misma herramienta.
- Se desarrollaron y representaron los diagramas de análisis y diseño de la herramienta con la ayuda del UML.
- Se implementó la herramienta de administración para PostgreSQL sobre la Web 2.0, y se incluyó un esbozo de manual de usuario que explica sus funcionalidades.

RECOMENDACIONES

Una vez concluido este trabajo, se recomienda:

- Realizar pruebas a la herramienta durante un período, para comprobar que cumple con las funcionalidades propuestas.
- Integrar a la herramienta los módulos *Postgis*, *Migración* y *Monitoreo* e incorporarle un módulo de diseño gráfico de consultas y diagramas.
- Continuar la investigación con el propósito de aumentar las funcionalidades de la herramienta, obteniendo mejoras en futuras versiones de la misma.

REFERENCIAS

1. Apache Software Foundation, 2011. Welcome to the Apache Software Foundation! Disponible en: <http://www.apache.org/> [Accedido Junio 28, 2011].
2. Comunidad de Desarrollo Ubuntu, 2008. PgAdmin III - Guía Ubuntu. *PgAdmin III*. Disponible en: http://www.guia-ubuntu.org/index.php?title=PgAdmin_III [Accedido Noviembre 16, 2010].
3. Daniele Varrazzo, 2010. PostgreSQL + Python | Psycopg. Disponible en: <http://www.initd.org/psycopg/> [Accedido Junio 28, 2011].
4. Django Software Foundation, 2005. Django | The Web framework for perfectionists with deadlines. Disponible en: <https://www.djangoproject.com/> [Accedido Junio 28, 2011].
5. Django Software Foundation, 2010. Django Web framework | Django en Español, django-es. *¡Descubre Django!* Disponible en: <http://django.es/> [Accedido Noviembre 11, 2010].
6. Eclipse Foundation, 2011. Eclipse - The Eclipse Foundation open source community Website. Disponible en: <http://www.eclipse.org/> [Accedido Junio 29, 2011].
7. EMS Database Management Solutions, 1999. EMS SQL Manager - EMS SQL Manager – EMS SQL Manager for PostgreSQL: PostgreSQL Tools, PL/pgSQL Query Builder, PostgreSQL Database Management and Administration, Export and Import Tools for PostgreSQL. *EMS SQL Manager for PostgreSQL*. Disponible en: <http://www.sqlmanager.net/products/postgresql/manager> [Accedido Junio 27, 2011].
8. Fidel Gil, Javier Albrigo & Javier Do Rosario, 2005. SISTEMAS DE GESTIÓN DE BASE DE DATOS SGBD / DBMS. Disponible en: http://www.google.com.cu/url?sa=t&source=Web&cd=8&ved=0CFAQFjAH&url=http%3A%2F%2Fmachtiani.icyt.df.gob.mx%2Ffile.php%2F1%2Fmoddata%2Fforum%2F1%2F9240%2FSGBD_2.pdf&rct=j&q=%20SGBD&ei=zQncTlaeLMWnnQehtuUW&usg=AFQjCNGFcVAs91PHuc4TzzAE3BjpKNrBiQ&cad=rja [Accedido Noviembre 11, 2010].

9. Indira Martínez Bravo, 1999. Informática. *¿Qué es un sistema de gestión de base de datos (SGBD)?* Disponible en: <http://indira-informatica.blogspot.com/2007/09/qu-es-un-sistema-de-gestin-de-base-de.html> [Accedido Junio 24, 2011].
10. Ing. Ramón Paumier Samón., Ing. Yoandy Pérez Villazón & Ing. Abel Meneses Abad., 2010. Guía Cubana (Migración a Software Libre). *GUÍA CUBANA DE MIGRACIÓN A SOFTWARE LIBRE*. Disponible en: <http://es.scribd.com/doc/17779155/Guia-Cubana-Migracion-a-Software-Libre> [Accedido Mayo 18, 2011].
11. Javier Eguíluz Pérez, 2009. Introducción a CSS. Disponible en: http://www.librosWeb.es/css/pdf/introduccion_css.pdf.
12. Javier Eguíluz Pérez, 2008. Introducción a XHTML. Disponible en: http://www.librosWeb.es/xhtml/pdf/introduccion_xhtml.pdf.
13. José Miguel Rubio L., 2011. Base de Datos. Disponible en: http://informaticapucv.cl/civil/attachments/article/52/Capitulo_II.pdf.
14. Microsoft Corporation, 2010a. Microsoft SQL Server 2008 - Introducción. *Introducción a SQL Server 2008*. Disponible en: <http://www.microsoft.com/spain/sql/2008/overview.aspx> [Accedido Noviembre 11, 2010].
15. Microsoft Corporation, 2010b. Visual FoxPro Home. *Visual FoxPro Developer Center*. Disponible en: [http://msdn.microsoft.com/es-ar/vfoxpro/default\(en-us\).aspx](http://msdn.microsoft.com/es-ar/vfoxpro/default(en-us).aspx) [Accedido Noviembre 11, 2010].
16. Miguel Ángel Álvarez, 2003. Qué es Python. *Lenguaje de programación de propósito general, orientado a objetos, que también puede utilizarse para el desarrollo Web*. Disponible en: <http://www.desarrolloWeb.com/articulos/1325.php> [Accedido Noviembre 11, 2010].
17. Miguel Ángel Álvarez, 2002. Introducción a JavaScript. *Introducción a JavaScript*. Disponible en: <http://www.desarrolloWeb.com/articulos/490.php> [Accedido Junio 27, 2011].
18. Msc. Reinier Castillo Figueroa, 2011. *Herramienta administrativa para mecanismo de replicación SymmetricDS*. Tesis Presentada en Opción al Título Académico de Máster en Ciencia de la Computación. Santa Clara: Universidad Central “Marta Abreu” de las Villas.

19. Navicat for PostgreSQL group, 1999. Navicat - PostgreSQL Overview - The best free MySQL Client Management tool, SQLite GUI, Oracle admin tool, SQL Server frontend Frontend, SQL Azure & PostgreSQL tool for Windows, Mac OS X & Linux - Download Now! *Navicat for PostgreSQL*. Disponible en: http://www.navicat.com/en/products/navicat_pgsql/pgsql_overview.html [Accedido Junio 24, 2011].
20. Oracle Corporation, 2010. Product Category - Oracle Database. *Oracle Database*. Disponible en: <https://shop.oracle.com/pls/ostore/product?p1=oracledatabase> [Accedido Noviembre 11, 2010].
21. pgaccess group, 2011. PgAccess | Download PgAccess software for free at SourceForge.net. *PgAccess*. Disponible en: <http://sourceforge.net/projects/pgaccess/> [Accedido Junio 24, 2011].
22. pgAdmin group, 2011. pgAdmin: PostgreSQL administration and management tools. Disponible en: <http://www.pgadmin.org/> [Accedido Junio 24, 2011].
23. phpPgAdmin group, 2011. phpPgAdmin - start. *What is phpPgAdmin?* Disponible en: <http://phppgadmin.sourceforge.net/doku.php?id=start> [Accedido Junio 24, 2011].
24. PostgreSQL Global Development Group, 1996. PostgreSQL: About. *About*. Disponible en: <http://www.postgresql.org/about/> [Accedido Noviembre 11, 2010].
25. Python Software Foundation, 1990. Python Programming Language – Official Website. Disponible en: <http://www.python.org/> [Accedido Junio 28, 2011].
26. Rafael Martínez, 2009. Sobre PostgreSQL | www.postgresql.org.es. Disponible en: http://www.postgresql.org.es/sobre_postgresql [Accedido Mayo 26, 2011].
27. Sencha, 2011. Desktop and Mobile Web App Frameworks Using JavaScript and HTML5 | Ext JS, Ext GWT and Sencha Touch | Sencha. Disponible en: <http://www.sencha.com/> [Accedido Junio 29, 2011].
28. SQLite, 2010. About SQLite. *About SQLite*. Disponible en: <http://www.sqlite.org/about.html> [Accedido Noviembre 11, 2010].
29. Visual Paradigm Group, 2011. UML CASE tool for software development. Disponible en: <http://www.visual-paradigm.com/product/vpuml/> [Accedido Junio 29, 2011].

BIBLIOGRAFÍA

1. Adrian Holovaty, Jacob Kaplan-Moss, 2006. The Django Book: Version 1.0 (English). Disponible en: <http://www.djangobook.com/en/1.0/> [Accedido Junio 29, 2011].
2. Andrés Marzal, Isabel Gracia, 2003. Introducción a la programación con Python. Disponible en: <http://marmota.act.uji.es/mtp/pdf/python.pdf>.
3. Comunidad de Desarrollo PgAdmin, 2010. pgAdmin: PostgreSQL administration and management tools. Introduction. Disponible en: <http://www.pgadmin.org/> [Accedido Noviembre 16, 2010].
4. Comunidad de Desarrollo phpPgAdmin, 2010. phpPgAdmin :: Web Based PostgreSQL Administration Tool. What is phpPgAdmin? Disponible en: <http://phppgadmin.sourceforge.net/> [Accedido Noviembre 16, 2010].
5. David Asorey Álvarez, 2007. Tutorial de Django. Disponible en: <http://davidasorey.net/static/django-tutorial/> [Accedido Junio 29, 2011].
6. Django Software Foundation, 2005. Django | The Web framework for perfectionists with deadlines. Meet Django. Disponible en: <http://www.djangoproject.com/> [Accedido Noviembre 11, 2010].
7. Editorial McGraw-Hill, 2008. Curso sgbd. sistemas gestores de bases de datos (primera parte). Disponible en: <http://www.mailxmail.com/curso-sistemas-bases-datos> [Accedido Junio 29, 2011].
8. Eduardo León, 2011. Tutorial Visual Paradigm. Disponible en: <http://es.scribd.com/doc/36636137/Tutorial-Visual-Paradigm> [Accedido Junio 29, 2011].
9. Fidel Gil, Javier Albrigo & Javier Do Rosario, 2005. SISTEMAS DE GESTIÓN DE BASE DE DATOS SGBD / DBMS. Disponible en: http://www.google.com/cu/url?sa=t&source=Web&cd=8&ved=0CFAQFjAH&url=http%3A%2F%2Fmachtiani.icyt.df.gob.mx%2Ffile.php%2F1%2Fmoddata%2Fform%2F1%2F9240%2FSGBD_2.pdf&rct=j&q=%20SGBD&ei=zQncTiaeLMWnnQehtuUW&usg=AFQjCNGFcVAs91PHuc4TzzAE3BjpKNrBiQ&cad=rja [Accedido Noviembre 11, 2010].

10. Jairo Chapela Martínez, 2007. Introducción al entorno de desarrollo Eclipse. Disponible en: http://www-gris.det.uvigo.es/wiki/pub/Main/MiscResources/Manual_Eclipse.pdf.
11. Javier Eguíluz Pérez, 2009a. Introducción a CSS. Disponible en: http://www.librosWeb.es/css/pdf/introduccion_css.pdf.
12. Javier Eguíluz Pérez, 2009b. Introducción a JavaScript. Disponible en: http://www.librosWeb.es/javascript/pdf/introduccion_javascript.pdf.
13. Javier Eguíluz Pérez, 2008. Introducción a XHTML. Disponible en: http://www.librosWeb.es/xhtml/pdf/introduccion_xhtml.pdf.
14. Jesús Manuel Montero Garrido, 2010. Plataforma Eclipse Introducción Técnica. Disponible en: http://150.244.56.228/descargas_Web/cursos_verano/20040801/Jesus_Montero/documentacion_eclipse.pdf.
15. José Miguel Rubio L., 2011. Base de Datos. Disponible en: http://informaticapucv.cl/civil/attachments/article/52/Capitulo_II.pdf.
16. Marcos Dacosta Balboa, 2009. Navicat – Cliente MySQL, PostgreSQL y Oracle. Navicat – Cliente MySQL, PostgreSQL y Oracle. Disponible en: <http://www.dacostabalboa.com/es/navicat-cliente-mysql-postgresql-y-oracle/4722> [Accedido Junio 28, 2011].
17. María Sierra, 2011. Trabajando con Visual Paradigm for UML. Disponible en: <http://personales.unican.es/ruizfr/is1/doc/lab/01/is1-p01-trans.pdf>.
18. Pablo Perales Díaz, 2008. APUNTES SGBD. Disponible en: <http://www.utomde.com/asigna/sghd/MisResumenes.pdf>.
19. PostgreSQL Global Development Group, 1996. PostgreSQL: About. About. Disponible en: <http://www.postgresql.org/about/> [Accedido Noviembre 11, 2010].
20. PremiumSoft, 1999a. Navicat - Download Center - Download the World's Best Oracle Manager, MySQL Front End, SQLite GUI Frontend, SQL Azure & PostgreSQL GUI for Windows, Mac OS X & Linux - Download Now! Support Access to MySQL, Excel to MySQL, MySQL editor, MySQL administrator, MySQL frontend, MySQL manager, phpmyadmin, MySQL management, MySQL administration, MySQL front, MySQL for Toad, MySQL query builder, MySQL

- download, MySQL migration, MySQL control centre, MySQL client and MySQL report builder. Disponible en: <http://www.navicat.com/download/download.html> [Accedido Junio 28, 2011].
21. PremiumSoft, 1999b. Navicat - PostgreSQL. Navicat for PostgreSQL. Disponible en: http://www.navicat.com/en/products/navicat_pgsql/pgsql_overview.html [Accedido Noviembre 17, 2010].
22. Rafael Martínez, 2009. Sobre PostgreSQL | www.postgresql.org.es. Disponible en: http://www.postgresql.org.es/sobre_postgresql [Accedido Mayo 26, 2011].
23. Raúl Eduardo Chavarría, 2006. Ide Eclipse, Breve Guía. Disponible en: <http://www.slideshare.net/Benedeti/ide-eclipse-breve-gua-201399> [Accedido Junio 29, 2011].
24. Raúl González Duque, 2011. Python para todos. Disponible en: <http://launchpadlibrarian.net/18980633/Python%20para%20todos.pdf>.
25. Ricardo J. Barrios Díaz, 2007. Tutorial: Introduction to Ext (Spanish) - Sencha - Learn. Disponible en: [http://www.sencha.com/learn/Tutorial:Introduction_to_Ext_\(Spanish\)](http://www.sencha.com/learn/Tutorial:Introduction_to_Ext_(Spanish)) [Accedido Junio 29, 2011].
26. Sencha Inc., 2006. Sencha - Ext JS - Client-side JavaScript Framework. Ext JS Cross-Browser Rich Internet Application Framework. Disponible en: <http://www.sencha.com/products/js/> [Accedido Noviembre 11, 2010].
27. The Apache Software Foundation, 2011. Welcome to The Apache Software Foundation! The Apache Software Foundation. Disponible en: <http://www.apache.org/> [Accedido Mayo 23, 2011].
28. Tomás Javier Robles Prado & Raúl Turienzo Fernández, Introducción a PostgreSQL. Disponible en: <http://users.servicios.retecal.es/tjavier/intdb.pdf>.
29. Valentín Starck, 2011. ExtJS « Aijoona. Disponible en: <http://blog.aijoona.com/tag/extjs/> [Accedido Junio 29, 2011].
30. Visual Paradigm, 2010. UML CASE tool for software development. Visual Paradigm for UML - UML tool for software application development. Disponible en: <http://www.visual-paradigm.com/product/vpum/> [Accedido Noviembre 11, 2010].