Módulo de administración para el servicio PostgreSQL en la Herramienta para la Migración y Administración de Servicios Telemáticos

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autora

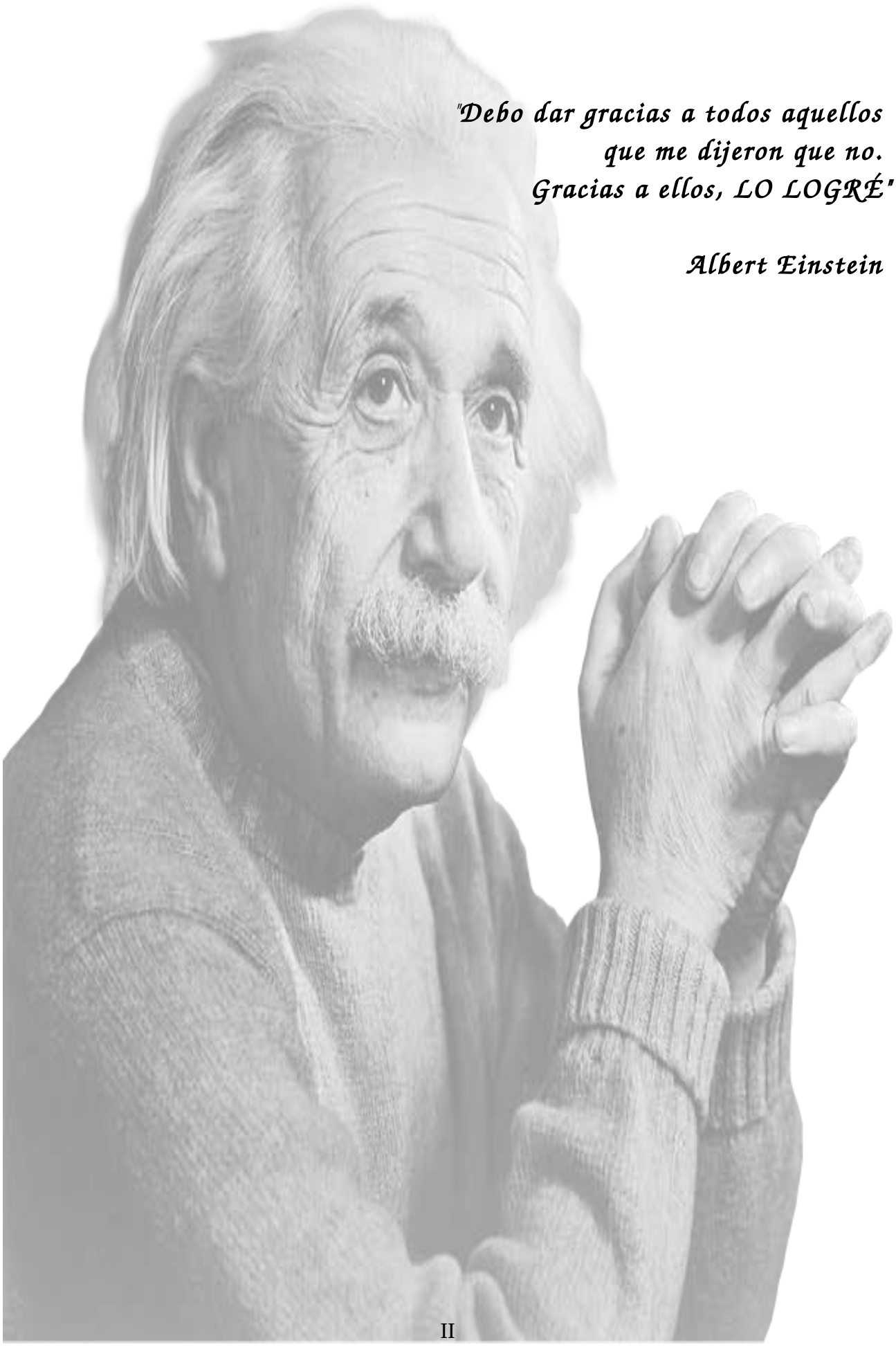
Llilian Fernández Casasayas

Tutores

Msc. Yariel Ramos Negrín
Ing. Yurenia Hernández Blanco
Ing. Pablo Soria Acosta

Junio del 2014

La Habana, Cuba



"Debo dar gracias a todos aquellos
que me dijeron que no.
Gracias a ellos, LO LOGRÉ"

Albert Einstein

Declaración de autoría

Declaro ser la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ___días del mes de_____del año 2014.

_____
Llilian Fernández Casasayas

_____
Msc. Yariel Ramos Negrín

_____ _____
Ing. Pablo Soria Acosta Ing. Yurenia Hernández Blanco

Agradecimientos

A mi mamá linda por andar siempre conmigo por los caminos de la vida. Por quererme tanto, por reír conmigo, por llorar conmigo y por tener paciencia en mis momentos de desespero. Gracias.

A mi papá por apoyarme siempre y brindarme su ejemplo.

A mis tíos y tías por enseñarme que en la unión familiar está la fuerza, y que esa fuerza no hay quien la destruya.

A mis primos que han sido mis hermanos por parte de madre, incluso a pesar de la distancia, gracias infinitas por su ayuda y preocupación.

A mis hermanos por todo su cariño, por su amistad y sobre todo por brindarme la seguridad de saber que siempre podré contar con ustedes.

A mis mejores y viejos amigos Yani y Rey, por ser los mejores amigos del mundo.

A Reidiel, Yasiel y Susana las eternas gracias, sin ustedes esto no hubiera sido posible.

A mis amigas: Soyma, Yuni, Ani, Day, Sury, Eva. Su amistad fue una de los cosas más linda que me ha regalado esta universidad. Espero que nunca nos olvidemos.

A mis varones preferidos por soportarme todas mis malcriadeces Dany, Gustavo, Nestor y Jesus.

A los mejores compañeros de aula, por acogerme estos dos cursos y hacerme sentir parte

de ustedes.

A mis tutores Yurenia, Pablo y Yariel por enseñarme el significado de esfuerzo.

A todos los integrantes del Dpto SIMAYS por la ayuda brindada, en especial al Flaqui, Yadiel y Amaury.

Dedicatoria

A mi mamá y papá por todo su amor y sacrificio para cumplir este sueño. Los AMO.

A toda mi familia por estar en los momentos buenos y malos incondicionalmente.

A todos aquellos que aportaron un granito de amor durante todos estos años para que yo cumpliera mi sueño.

Para todos va esto.

Resumen

El departamento de Servicios Integrales en Migración, Asesoría y Soporte perteneciente al Centro de Soluciones Libres de la Universidad de las Ciencias Informáticas, está desarrollando una Herramienta para la Migración y Administración de Servicios Telemáticos (HMAST) sobre GNU/Linux. La misma se encuentra en su primera versión y cuenta con módulos para administrar DHCP y Proxy pero carece de funcionalidades que le permitan configurar los servidores de base de datos PostgreSQL. Por tanto, el objetivo del presente trabajo de investigación es desarrollar un módulo que garantice incrementar la facilidad de administración del servicio PostgreSQL desde dicha herramienta. Para el cumplimiento de este objetivo se realiza un estudio de las herramientas que permiten administrar el servicio PostgreSQL, con el fin de contar con aquellas características que puedan ser útiles para el desarrollo del módulo en cuestión. Además, se documentan las tecnologías, herramientas y lenguajes de programación utilizados, así como los artefactos requeridos por la metodología ágil de desarrollo SXP. Como resultado se obtuvo un módulo robusto y con una interfaz intuitiva que cuenta con funcionalidades que permiten instalar y desinstalar el servicio, gestionar los ficheros de configuración y el control de acceso. También posibilita garantizar la integridad de los datos, así como administrar el uso de los recursos que se le asignan al servidor, los registros del sistema y las estadísticas de ejecución. Además, realiza ajustes de consultas y revisiones periódicas a las tablas con modificaciones considerables.

Palabras claves: administración, migración, postgreSQL, servicios, telemáticos.

Índice de contenido

Índice de Figura

Índice de Tabla

Introducción

Las aplicaciones informáticas como parte de las Tecnologías de la Información y las Comunicaciones (TICs) posibilitan variadas opciones. En la actualidad no se concibe el funcionar de una empresa, entidad o institución sin estas aplicaciones que tanto facilitan la gestión de los procesos, recursos, productos y la toma de decisiones. Aplicaciones que tienen como soporte generalmente base de datos, las cuales proporcionan numerosas facilidades como rapidez y actualidad en la información y para que estas ofrezcan soluciones de una manera rentable, íntegra y segura, requieren de servidores potentes para un mejor control.

Los servidores de base de datos, según Menéndez Barzanallana del Departamento de Informática y Sistemas de la Universidad de Murcia, surgen con motivo de la necesidad de las empresas de manejar grandes y complejos volúmenes de datos, y que además requieren compartir la información con un conjunto de clientes de una manera segura [1].

Dentro de los servidores de base de datos más usados actualmente por su robustez, simplicidad y seguridad se encuentra PostgreSQL. Hoy en día muchos sistemas en los que la consistencia y persistencia de las base de datos es fundamental, utilizan como mejor opción los servidores de base de datos PostgreSQL, delegando en los mismos la responsabilidad de la gestión y almacenamiento de la información. PostgreSQL es un proyecto de código abierto que está dirigido por una comunidad de desarrolladores que trabajan de forma desinteresada, siendo esta una de las ventajas que hace que se ubique entre los sistemas gestores de bases de datos.

Para los países subdesarrollados o de bajos ingresos económicos, el Software Libre (SWL) o de Código Abierto (CA) es la alternativa ideal para potenciar el desarrollo de la informática y la práctica social de las TICs. Esto representa la no utilización de productos informáticos que demanden la autorización de sus propietarios para su explotación. Ofrece beneficios como: la libertad para adquirirlo sin la limitante económica e igualdad en las personas para obtenerlo. Es desarrollado de forma colectiva y cooperativa, con el objetivo de beneficiar a toda la comunidad, permitiendo el conocimiento y la obtención de las actualizaciones y las nuevas versiones realizadas [2].

Actualmente el país se encuentra inmerso en un proceso paulatino de migración hacia SWL y aplicaciones de CA, debido a que esto constituye una necesidad por el impacto que representa para la sociedad cubana en los aspectos políticos, económicos, sociales y tecnológicos. En este proceso de migración el gobierno cubano ha involucrado algunas de sus entidades y Organismos de la Administración Central del Estado (OACE), ya que tienen instalado en la mayoría de sus

servidores sistemas operativos privativos, como Windows Server en alguna de sus variantes.

Para llevar a cabo la tarea de migración en el país se creó una estructura compuesta por cuatro grupos de trabajo (legal, capacitación, divulgación y técnico) y uno de dirección denominado Grupo Ejecutivo. El Grupo Técnico Nacional posee su núcleo fundamental y principal fuerza de trabajo en el Centro de Software Libres (CESOL) de la Universidad de las Ciencias Informáticas (UCI); conformado por el Departamento de Sistemas Operativos y el Departamento de Servicios Integrales en Migración, Asesoría y Soporte (SIMAYS). CESOL tiene como objetivos fundamentales: elaborar el sistema operativo para la migración en Cuba (GNU/Linux Nova) y definir las directrices, lineamientos y soluciones que guiarán la migración nacional. SIMAYS por su parte se encuentra altamente capacitado y especializado en brindar servicios integrales relacionados con la migración a código abierto: asesoría, consultaría, capacitación y soporte técnico. Además, desarrolla aplicaciones para apoyar el proceso de migración a código abierto [3]. Actualmente SIMAYS está inmerso en el desarrollo de la Herramienta para la Migración y Administración de Servicios Telemáticos (HMAST). El desarrollo de HMAST tiene el propósito de brindar a los OACE, que se encuentren en proceso de migración, una aplicación que permita migrar a plataformas libres y mantener en ejecución los servicios telemáticos de las entidades.

Hoy en día en los OACE se hace necesario la administración del servicio de base de datos PostgreSQL por parte de los administradores de redes. Esta labor requiere de un conjunto de tareas que resultan complejas debido a que no se cuenta con una herramienta gráfica que permita gestionar los archivos de configuración, así como la administración de forma remota y centralizada. Para llevar a cabo esta labor los administradores deben conectarse al servicio a través de la consola de comandos (terminal), lo que origina la necesidad de conocer una gran cantidad de comandos, ficheros, parámetros y estructuras de los directorios disponibles en el servicio; tarea que la mayoría de las veces no se logra realizar de manera eficiente.

A pesar de que HMAST permite la administración de varios servicios telemáticos tales como: DHCP y Proxy, no ofrece funcionalidades que garanticen la realización del proceso anteriormente mencionado.

Una vez planteada la problemática descrita anteriormente se presenta como **problema a resolver**: ¿Cómo incrementar la facilidad de administración del servicio PostgreSQL desde HMAST?

El objeto de estudio del presente trabajo de investigación se centra en la administración de servidores de base de datos en entornos libres.

Por lo que se propone como **objetivo general** desarrollar un módulo que garantice incrementar la facilidad de administración del servicio PostgreSQL desde HMAST.

El **campo de acción** está enmarcado en la administración del servidor de base de datos PostgreSQL.

En el marco de esta investigación se ha desglosado el objetivo general en los siguientes **objetivos específicos**:

- Elaborar la fundamentación teórica de la investigación.
- Analizar y diseñar el módulo de administración del servicio PostgreSQL.
- Implementar y probar el módulo de administración del servicio PostgreSQL.

Para dar cumplimiento a los objetivos específicos se planifican las siguientes **tareas de investigación**:

- Conceptualización de los elementos necesarios para comprender el ámbito del problema a resolver.
- Caracterización de las aplicaciones que permiten administrar el servicio PostgreSQL.
- Definición de los requisitos funcionales y no funcionales del módulo a desarrollar.
- Implementación de las funcionalidades definidas para administrar el servicio PostgreSQL.
- Diseño y ejecución de pruebas para el módulo implementado.

En el presente trabajo se define la siguiente **idea a defender:** El desarrollo de un módulo para la administración del servicio PostgreSQL incrementará la facilidad de administración de este servicio desde HMAST.

En el desarrollo de la investigación se utilizan los siguientes **métodos teóricos**:

Analítico-Sintético: Este método se utiliza en el análisis de documentos, teorías y otros materiales sobre la administración de servidores de base de datos PostgreSQL, de manera que se procese la información y se elaboren conclusiones relacionadas con el objeto de estudio.

La investigación se ha estructurado, para una mejor comprensión del contenido, en introducción, 3

capítulos, conclusiones generales y bibliografía utilizada, además de un glosario de términos, donde se explican en detalles los términos técnicos que han sido utilizados en la elaboración del documento. También cuenta con los anexos que complementan el trabajo realizado. Los capítulos se estructuran de la siguiente forma:

Capítulo 1: Fundamentación teórica: Se definen los conceptos asociados y se analizan las principales herramientas que permiten administrar el servicio de base de datos PostgreSQL. Una vez realizado este análisis se describen brevemente las principales características de HMAST, sistema base del módulo a desarrollar. Por último se fundamentan las tecnologías, lenguajes de programación, metodología y herramientas utilizadas en el desarrollo del módulo.

Capítulo 2: Análisis y diseño de la solución propuesta: Se presenta la propuesta de solución al problema planteado, quedando definidos los requisitos funcionales y no funcionales del módulo. También se describen las Historias de Usuario (HU) que quedaron definidas. Por último, se detalla la arquitectura del módulo y elementos del diseño, como son los patrones utilizados y la descripción del Diagrama de paquetes.

Capítulo 3: Implementación y pruebas del módulo: Se elabora el Plan de *release*, donde se define cuáles funcionalidades serán desarrolladas en cada etapa del proceso de implementación. Además, se detallan las tareas de ingeniería más importantes, las cuales contribuyeron al desarrollo del módulo. Por último, se aborda sobre las pruebas realizadas al *software*, con el fin de comprobar el correcto funcionamiento del mismo y su correspondencia con los requisitos planteados.

Capítulo 1. Fundamentación teórica

La configuración de servidores de base de datos en entornos de los OACE juega un papel fundamental en el proceso de gestión de la información. Los servidores de base de datos facilitan la manipulación de grandes volúmenes de información de forma fiable y coherente. En el presente capítulo se realiza un estudio sobre los conceptos fundamentales de la administración del servicio PostgreSQL. Se muestran las herramientas y tecnologías utilizadas en la investigación, así como la metodología de desarrollo de *software*.

Servidor de Base de Datos

Para el desarrollo de este trabajo se debe comprender el término servidor de base de datos como concepto. Se plantea que un servidor de base de datos es un sistema bajo arquitectura cliente/servidor que proporciona servicios de gestión, administración y protección de la información (datos) a través de conexiones de red, gobernadas por unos protocolos definidos y a los que acceden los usuarios, de modo concurrente, a través de aplicaciones clientes (bien sean herramientas del propio sistema como aplicaciones de terceros) [4].

La información en estos servidores será almacenada en tablas que disponen de un conjunto de campos que describen su contenido, denominados columnas, y se relacionan entre sí a través de un conjunto definido de claves. El sistema de base de datos, será el responsable de mostrar la información requerida a través de conjuntos de datos planos o cursores, independizando las relaciones establecidas y la arquitectura de las base de datos de la necesidad de información del usuario. Mediante mecanismos de control de transacciones basados en reglas (disparadores) de definición del tipo de entrada de datos y reglas de validación de las entradas de datos, el sistema es capaz de proteger la información. A través de complejos sistemas de indexación, estos sistemas serán capaces de ordenar y acelerar las consultas a la información requerida. Cuanto mejor se indexen los datos, más rápidas se realizarán las consultas. Por último, y como un factor muy importante de cara al diseño de base de datos, los sistemas deben proporcionar la posibilidad de automatizar operaciones de acceso, filtrado y control de los datos, a través de los procedimientos almacenados.

Sistema gestor de Base de Datos

Un Sistema Gestor de Base de Datos o *Data Base Management System (SGBD)* es el *software* que permite a los usuarios procesar, describir, administrar y recuperar los datos almacenados en una base de datos. El éxito de un SGBD reside en mantener la seguridad e integridad de los datos.

Un SGBD debe permitir [5]:

- Definir base de datos: especificar tipos, estructuras y restricciones de datos.
- Construir las base de datos: guardar los datos en algún medio controlado por el mismo SGBD.
- Manipular base de datos: realizar consultas, actualizarlas, generar informes.

Las características de un SGBD son [6]:

- Abstracción de la información: Los SGBD no muestran a los usuarios detalles acerca del almacenamiento físico de los datos. No importa la cantidad de archivos que ocupen las base de datos, este hecho se hace transparente al usuario y de esta forma 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: Debido al gran volumen de información almacenada en las base de datos se corre el riesgo de contener redundancia de los datos, un buen diseño de una base de datos logrará evitar la aparición de información repetida o redundante. 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 encuentre segura frente a usuarios malintencionados, que intenten leer 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 de permisos.

- 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. Los SGBD proveen herramientas para la creación de copias de seguridad.

- Control de la concurrencia: En la mayoría de los entornos (excepto quizás el doméstico), 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. Así pues, un SGBD debe controlar este acceso concurrente a la información, que podría derivar en inconsistencias.

PostgreSQL

PostgreSQL es un sistema gestor de base de datos objeto-relacional basado en el proyecto POSTGRE de la Universidad de Berkeley. El director de este proyecto fue el profesor Michael Stonebraker, y estuvo patrocinado por la Agencia de Proyectos de Investigación Avanzados de Defensa (DARPA), la Oficina de Investigación del Ejército (ARO) y la Fundación Nacional para la Ciencia (NSF).

PostgreSQL utiliza un modelo cliente/servidor y usa multiprocesos en vez de multihilos para garantizar la estabilidad del sistema. Un fallo en uno de los procesos no afectará el resto y el sistema continuará funcionando. Se ejecuta en casi todos los principales sistemas operativos: GNU/Linux, Unix, BSDs, Mac OS, Beos, Windows. Contiene una documentación muy bien organizada, pública y libre, con comentarios de los propios usuarios. Además, cuenta con comunidades muy activas, varias de estas en castellano. Tiene un soporte nativo para los lenguajes más populares del medio: PHP, C, C++, Perl, Python, entre otros. También soporta todas las características de una base de datos profesional (*triggers*, *store procedures,* funciones, secuencias, relaciones, reglas, tipos de datos definidos por usuarios, vistas, vistas materializadas). Altamente adaptable a las necesidades del cliente [7].

Entre las principales características de este gestor de base de datos se puede mencionar que soporta distintos tipos de datos como fecha, monetarios, elementos gráficos, datos sobre redes y

cadenas de bits. Permite la creación de tipos propios, la declaración de funciones propias y la definición de disparadores. Incluye herencia entre tablas, lo que lo sitúa entre los gestores objeto-relacionales. Admite la gestión de usuarios, los permisos asignados a cada uno de ellos. Permite la duplicación de base de datos maestras en múltiples sitios de réplica. Cuenta con funciones de compatibilidad para ayudar en la transición desde otros sistemas menos compatibles con SQL [8].

Actualidad de PostgreSQL

Hoy en día el grupo central (*core team*) de desarrolladores está formado por 7 personas, existen 24 desarrolladores principales y más de 18 desarrolladores habituales. En total alrededor de 50 personas activas, contribuyendo con el desarrollo de PostgreSQL.

Existe una gran comunidad de usuarios, programadores y administradores que colaboran activamente en numerosos aspectos y actividades relacionadas con el proyecto. A través de informes, soluciones de problemas, comprobación del funcionamiento, contribuciones de nuevas ideas, discusiones sobre problemas y documentación de PostgreSQL.

Existen varias empresas que colaboran con dinero y/o con tiempo/personas en mejorar PostgreSQL. Muchos desarrolladores y nuevas características están patrocinadas por empresas privadas. Esto se debe a que PostgreSQL es una excelente alternativa al *software* privativo, resultando un gestor de base de datos realmente económico, propiciando su extensión comercial por todo el mundo.

En el mundo muchas agencias y empresas utilizan este poderoso sistema gestor de base de datos. En Norteamérica lo usa la fuerza armada y algunos proyectos de la Biblioteca del Congreso de los Estados Unidos. También son dignos de resaltar algunas iniciativas del Estado de California, de la Universidad de *Oxford* y del Laboratorio Nacional de Sandia (encargado de dar soluciones tecnológicas para resolver las amenazas nacionales y mundiales para la paz y la libertad). En Latinoamérica son conocidos los casos de Loma Negra y Quilmas en Argentina, los casos de Entel (Empresa Nacional de Telecomunicaciones) y la Superintendencia de AFPs (Asociación de Administradoras de Fondos de Pensiones) en Chile y los casos de varias empresas de telecomunicaciones brasileñas. Pero el caso más paradigmático lo constituye el uso del PostgreSQL y otros productos de CA por una de las corporaciones financieras más grandes del mundo: Deutsche Bank (Banco de inversión internacional) [9]. Además se encuentran proyectos como la Armada Nacional de la República de Colombia donde todo su sistema informático se encuentra funcionando con Linux, Apache, PHP y PostgreSQL. Otro caso donde se evidencia algo similar es en el Servicio Autónomo de la Propiedad Intelectual (SAPI) de la

República Bolivariana de Venezuela en la cual se trabaja con Linux, Apache, Squid, Sendmail y PostgreSQL [10].

En Cuba existen más de 150 proyectos que utilizan PostgreSQL, algunos de ellos se encuentran en la Universidad de Holguín. También cabe destacar a la Comunidad Técnica Cubana de PostgreSQL que tiene como objetivos contribuir al desarrollo de tecnologías de base de datos tomando como base el gestor PostgreSQL, proveer soluciones integrales y consultorías relacionadas con la migración y la explotación de PostgreSQL y contribuir a la formación de especialistas de alto nivel apoyando el desarrollo tecnológico cubano [11].

En la UCI el uso de PostgreSQL es muy extendido, siendo esta la primera en usar este gestor en el país. En la universidad existen varios proyectos productivos que trabajan con PostgreSQL, entre los que se encuentra el Centro de Tecnologías de Gestión de Datos (DATEC) centro que tiene como premisa desarrollar base de datos utilizando PostgreSQL [12].

Componentes de PostgreSQL.

PostgreSQL funciona con una arquitectura Cliente/Servidor, un proceso servidor (*postmaster*) y una serie de aplicaciones clientes que realizan solicitudes de acciones contra las base de datos a su proceso servidor. Por cada una de estas aplicaciones el *postmaster* crea un proceso *postgres*. La comunicación entre el cliente y el servidor se realiza mediante el protocolo TCP/IP, normalmente por el puerto 5432, creando un *socket*[1]. Para un mejor entendimiento de los componentes que interactúan en su funcionamiento ver la Figura 1: Funcionamiento de PostgreSQL.

Componentes de la Figura 1 [13]:

- **Aplicación cliente**: aplicación que utiliza PostgreSQL como gestor de base de datos; la conexión puede ocurrir vía TCP/IP o *sockets* locales.
- **Procesos hijos *postgres***: son los encargados de autenticar a los clientes, de gestionar las consultas y mandar los resultados a las aplicaciones clientes.
- **PostgreSQL *shared buffer cache***: memoria compartida usada por PostgreSQL para almacenar datos en la caché.
- ***Write-Ahead Log* (WAL)**: componente del sistema encargado de asegurar la integridad de

[1] Socket: concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiar cualquier flujo de datos, generalmente de manera fiable y ordenada.

los datos, al registrar los cambios en un registro antes de ser escritos.

- ***Kernel disk buffer cache***: caché de disco del sistema operativo.

- **Disco**: disco físico donde se almacenan los datos y toda la información necesaria para que PostgreSQL funcione.

- **Demonio *postmaster***: proceso principal de PostgreSQL que se encarga de escuchar por un puerto las conexiones entrantes de clientes y de crear los procesos hijos que se encargaran de autentificar estas peticiones, gestionar las consultas y enviar los resultados a las aplicaciones clientes.

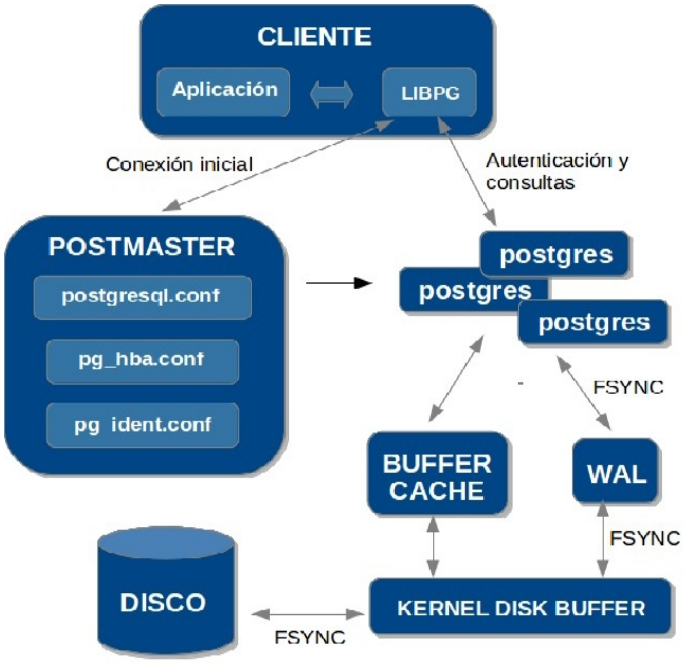


Figura 1: Funcionamiento de PostgreSQL.

En este último componente es donde mayor incide el presente trabajo ya que contiene los principales ficheros de configuración como son [13]:

- pg_ident.conf: Este fichero se utiliza para definir la información necesaria en el caso de que se utilice un acceso del tipo *ident* en pg_hba.conf.

- pg_hba.conf: Este fichero se utiliza para definir los diferentes tipos de accesos que un usuario tiene en el *clúster*. En el mismo se define cómo, dónde y desde qué sitio un usuario puede utilizar el *clúster* PostgreSQL. Todas las líneas que empiecen con el

símbolo numeral (#) se interpretan como comentarios. El resto debe de tener el siguiente formato:

[Tipo de conexión] [*database*] [usuario] [IP] [*Netmask*] [Tipo de autenticación] [Opciones]

Dependiendo del tipo de conexión y del tipo de autenticación, los parámetros [IP], [*Netmask*] y [Opciones] pueden ser opcionales. El tipo de conexión puede tener los siguientes valores: *local, host, hostssl y hostnossl*. El tipo de método puede tener los siguientes valores: *trust, reject, md5, crypt, password, krb5, ident, pam o ldap.* Son muchas las posibilidades que brinda este fichero de configuración. Por supuesto que las base de datos y usuarios usados en este fichero tienen que existir en el *clúster* para que todo funcione.

- postgresql.conf: En este fichero se pueden cambiar todos los parámetros de configuración que afectan el funcionamiento y el comportamiento de PostgreSQL en la máquina. Los cambios que se realicen en este fichero afectan a todas las base de datos definidas en el *clúster* PostgreSQL. La mayoría de los cambios se pueden poner en ejecución con un *reload*, aunque otros cambios necesitan que se arranque de nuevo el *clúster*.

  Existen muchos parámetros en este fichero que se pueden y con el tiempo se deberán ajustar, los cuales se deben cambiar antes de empezar a utilizar PostgreSQL. Ejemplos de estos son:

  - *listen_addresses**:*** especifica la dirección TCP/IP en el que el servidor está a la escucha para las conexiones desde las aplicaciones clientes.
  - *max_connections*: número máximo de clientes conectados a la vez a las base de datos.
  - *authentication_timeout*: el tiempo máximo para completar la autenticación del cliente.
  - *shared_buffers*: este parámetro es de gran importancia ya que define el tamaño del *buffer* de memoria utilizado por PostgreSQL.
  - *checkpoint_segments*: este parámetro es muy importante en base de datos con

numerosas operaciones de escritura.

Principales herramientas existentes para la administración de PostgreSQL.

Actualmente existe una gran variedad de aplicaciones que permiten la administración de servicios telemáticos. Siguiendo el hilo de la investigación, es necesario centrar el estudio en cómo estas herramientas administran el servicio PostgreSQL, con el fin de identificar aquellas características que puedan ser útiles para el desarrollo del módulo en cuestión y desechar los aspectos negativos, dos factores que dan paso al logro de un producto con la calidad requerida.

Para la concepción de HMAST se realizó un análisis de las principales herramientas de este tipo como son: PgAdmin III y Webmin. Las cuales se describen a continuación, haciendo énfasis principalmente en el proceso de administración del servicio PostgreSQL.

Webmin.

Webmin es una herramienta de administración de servicios telemáticos accesible vía web. Fue escrita por Jaime Cameron en lenguaje Perl. Está desarrollada para sistemas UNIX, aunque las versiones más recientes pueden ejecutarse también en entornos Windows. Con esta herramienta se pueden gestionar servicios y aplicaciones libres como el servidor web Apache, PHP, PostgreSQL, MySQL, DNS, Samba, DHCP, entre otros. Es importante aclarar que Webmin no modifica opciones desconocidas para él en los archivos de configuración y que no es necesario tener un entorno gráfico, ya que se puede utilizar con navegadores modo texto [14].

Esta herramienta permite a través de una interfaz web administrar el servicio PostgreSQL, pero para esto es necesario tener instalado el Webmin en la misma máquina donde se encuentra el servidor de base de datos. Este módulo cuenta con múltiples opciones, dentro de ellas se destacan: gestionar base de datos, los usuarios y la conexión al servidor; sin embargo está diseñado con una interfaz poco intuitiva por los numerosos campos de configuración que presenta en una misma página. Además, no posibilita configurar de forma remota en el servidor de base de datos, parámetros como el puerto de escucha. Ni permite realizar configuraciones en el uso de recursos del servidor, ni en la integridad de los datos.

PgAdmin III

Es una aplicación gráfica para gestionar el gestor de base de datos PostgreSQL. Está escrita en C++ usando la librería gráfica multiplataforma wxWigdets, lo que permite que se pueda ejecutar en cualquier plataforma [15].

PgAdmin III a pesar de ser una herramienta para administrar base de datos, posibilita configurar

los parámetros del servidor de base de datos; pero para esto es necesario que el servidor se encuentre instalado en la misma máquina donde se ejecuta la herramienta. Esta posee algunas deficiencias, ejemplo de esto es que no localiza automáticamente los archivos de configuración, además requiere que se establezcan permisos especiales a los mismos y otra de sus desventajas es que no gestiona los errores al introducir una configuración incorrecta.

A pesar de las deficiencias encontradas se identificaron aspectos importantes a tener en cuenta en el desarrollo de la solución como son: la forma de administrar el acceso al servidor que posee Webmin y la manera de mostrar los parámetros de configuración en PgAdmin III mediante una interfaz intuitiva.

Descripción de HMAST

HMAST es el sistema al cual se integrará la solución a desarrollar. Esta herramienta permite administrar los servidores de forma remota, contemplando las funcionalidades necesarias para administrar los usuarios, las tareas programadas y los servicios. A continuación se describen aspectos relevantes de la misma, tales como la arquitectura que utiliza, las principales funcionalidades que ofrece, y por último, se exponen las consideraciones a tener en cuenta al implementar un módulo para dicha herramienta.

Arquitectura de HMAST

La arquitectura presentada por la herramienta HMAST propone el diseño de un estilo arquitectónico N-Capas orientado al dominio, distribuida en cinco capas las cuales son descritas a continuación:

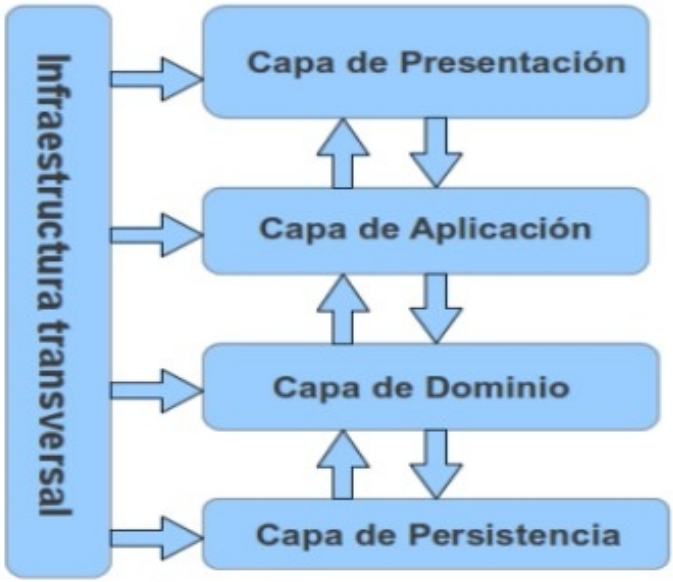


Figura 2: Arquitectura de HMAST

La capa **Presentación** se encarga de presentar al usuario los conceptos de negocio mediante una interfaz de usuario (IU), facilitar la explotación de dichos procesos, informar sobre la situación de los procesos de negocio e implementación de las reglas de validación de dicha interfaz.

La capa **Aplicación** es responsable de transmitir los datos que llegan desde la capa Presentación, realizando las funcionalidades para que la capa Dominio reciba la información que le permita satisfacer las peticiones. En esta capa también se ubican operaciones de trazas, seguridad, envío de correos electrónicos, cuando no forman parte estricta del negocio.

La capa **Dominio** conforma el hilo conductor de la aplicación, sus componentes solo dependen de la Capa de Infraestructura Transversal. Implementa la lógica de dominio (reglas de negocio), es responsable de las validaciones. Define las interfaces de persistencia a datos (contratos de repositorio) pero no los implementa. Sus componentes no están ligados a tecnologías específicas.

La capa **Persistencia** tiene asignada la responsabilidad de contener el código necesario para persistir los datos. Los principales componentes que contendrá la capa son los repositorios, que son clases que implementan los contratos de repositorios definidos en la capa de Dominio.

Las responsabilidades de la capa **Infraestructura Transversal** se basan en promover la reutilización de código, contiene las operaciones de seguridad, *logging*, monitoreo del sistema, mecanismos de persistencia reutilizables, validadores genéricos y todas aquellas operaciones que se puedan llamar desde otras capas [16]**.**

Funcionalidades que ofrece

- **Gestión de servidores lógicos**: permite la adición, edición y eliminación de los datos de un servidor lógico, además posibilita la conexión remota y desconexión a un servidor seleccionado.

- **Gestión de servicios telemáticos asociados a un servidor lógico:** permite la adición, edición y eliminación de los datos de un módulo, así como activación y desactivación de los mismos.

- **Gestión de las variables de configuración asociadas a un servidor lógico:** permite cargar y salvar las variables de configuración de los servicios telemáticos en un servidor lógico determinado (ficheros de configuración, nombre de módulos).

Consideraciones a tener en cuenta para implementar un módulo para HMAST

Para la implementación de un módulo que se desee integrar en HMAST se debe tener en cuenta que:

- La lógica de Aplicación no deberá incluir ninguna lógica del Dominio, solo tareas de coordinación relativas a requerimientos técnicos de la aplicación, como conversiones de formatos de datos de entrada a entidades del Dominio, llamadas a componentes de Infraestructura para que realicen tareas complementarias.

- Se garantizará que no se envíen hacia y desde la capa de Presentación objetos de Dominio, en su lugar deben viajar Objetos de Transferencia de Datos (*DTO, Data Object Transfer*).

- Las clases de servicios deben ser las únicas responsables (vías de acceso) de acceder a los repositorios, no se puede implementar código de persistencia a datos en la capa de Dominio.

- Solo se puede acceder a la información almacenada en los servidores haciendo uso de los repositorios.

- Es necesario que todo el código reutilizable por más de un repositorio se ponga a disposición de todos en la capa de Infraestructura Transversal.

Metodología, lenguaje y herramientas de desarrollo a utilizar

Para la creación de un *software* se hacen necesarios disímiles elementos como metodologías, herramientas de desarrollo y tecnologías, los cuales ayudan en gran medida, a obtener un resultado final de calidad. Una selección correcta de las mismas permite que estas se adapten a las necesidades y características del desarrollador. Con el objetivo de lograr una mayor compatibilidad con el sistema a desarrollar se decide utilizar las mismas herramientas y metodología que se usaron en el desarrollo de HMAST. Las que se describen a continuación.

Metodología de desarrollo

El Departamento SIMAYS, teniendo en cuenta las ventajas que proporciona la metodología ágil SXP, orienta el uso de la misma en los trabajos de desarrollo de aplicaciones que pertenezcan a él.

SXP

SXP es un híbrido de metodologías ágiles que recopila las mejores prácticas de las metodologías SCRUM y XP, además de regirse por los lineamientos de calidad de la UCI. Con la aplicación de esta metodología se logra la gestión de un equipo de forma que se tengan siempre medidos los progresos y que el trabajo se realice de forma eficiente. Se caracteriza por una programación rápida o extrema, teniendo como parte del equipo al usuario final, siendo uno de los requisitos para llegar el éxito del proyecto.

Consta de 4 fases principales: Planificación-Definición, Desarrollo, Entrega y Mantenimiento. Se caracteriza por ser una metodología iterativa e incremental, basada en Historias de Usuario (HU), con pequeñas mejoras unas tras otras, está atenta al cambio y permite que el equipo de programación se mantenga en una interacción frecuente con el cliente o usuario. Está indicada especialmente para proyectos de pequeños equipos de trabajo, con requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Permite que el trabajo se realice de forma unida, en la misma dirección, con un objetivo claro, siguiendo el avance de las tareas a realizar [17].

Lenguajes de programación.

Un lenguaje de programación es aquel elemento dentro de la informática que permite crear programas mediante un conjunto de instrucciones, operadores y reglas de sintaxis; que pone a disposición del programador para que este pueda comunicarse con los dispositivos de *hardware* y *software* existentes [18].

Java

Java fue diseñado como un lenguaje orientado a objetos, los cuales agrupan en estructuras encapsuladas tanto sus datos como las funcionalidades. Proporciona una colección de clases para su uso en aplicaciones de red, que permiten abrir *sockets*, establecer y aceptar conexiones con servidores o clientes remotos, facilitando así la creación de aplicaciones distribuidas. Es interpretado y compilado a la vez. Proporciona numerosas comprobaciones en compilación y en tiempo de ejecución. Sus características de memoria liberan a los programadores de una familia entera de errores. Soporta sincronización de múltiples hilos de ejecución a nivel de lenguaje, especialmente útiles en la creación de aplicaciones de red distribuidas [19].

Características del lenguaje

Dado que es un lenguaje orientado a objetos, implica que su concepción es muy próxima a la forma de pensar humana. También posee otras características como son [20]:

- Compilado: genera ficheros de clases compiladas, pero estas clases compiladas son en realidad interpretadas por la máquina virtual de Java, siendo esta la que mantiene el control sobre las clases que se estén ejecutando.

- Multiplataforma: el mismo código java que funciona en un sistema operativo, funcionará en cualquier otro sistema operativo que tenga instalada la máquina virtual de Java.

- Seguro: la máquina virtual, al ejecutar el código Java, realiza comprobaciones de seguridad, además el propio lenguaje carece de características inseguras, como por ejemplo los punteros.

Spring Framework

Spring Framework, también conocido simplemente como Spring, es un *framework* de código abierto para el desarrollo de aplicaciones en la plataforma Java. Su diseño le ofrece mucha libertad a los desarrolladores en Java, así como soluciones bien documentadas y fáciles de usar para las prácticas comunes en la industria.

Características [21]:

- Inyección de dependencias flexible con estilos de configuración XML y anotaciones basadas en soporte avanzado para la programación orientada a aspectos con variantes basado en *proxy* y *AspectJ* base. Apoya a las transacciones declarativas, almacenamiento

en caché declarativo y validaciones declarativas.

- Abstracciones de gran alcance para trabajar con especificaciones de Java EE comunes tales como JDBC, JPA, JMS JTA y soporte de primera clase para los marcos comunes de código abierto, como Hibernate y Quartz. Un marco flexible para la creación de Web RESTful, aplicaciones MVC y los puntos finales de servicio.

- Posee instalaciones de pruebas unitarias, así como de pruebas de integración.

Augeas

Augeas es una herramienta de edición de ficheros de configuración del sistema GNU/Linux. Es un editor que analiza los archivos en sus formatos nativos y los transforma en un tipo abstracto de datos llamado árbol y viceversa. La manipulación de la configuración la realiza mediante este árbol y los cambios que sean aplicados a él se guardan de forma nativa en dichos archivos de configuración. Un nodo del árbol tiene asignada una etiqueta y un valor, además le corresponden varios hijos que son una lista de nodos que pueden tener la misma etiqueta. Augeas es una herramienta modular, pero a diferencia de otras de su tipo, esta depende totalmente de sus módulos, debido a que son los encargados de permitir la transformación de un fichero en árbol y viceversa. A estos módulos se le llaman *Lenses* y son los encargados de brindar soporte a cada fichero de configuración, pues cada uno describe un archivo [22].

Herramientas utilizadas

Las herramientas son los programas, aplicaciones o sencillamente instrucciones usadas para realizar otras tareas de modo más fácil [23]. A continuación se describen las herramientas a utilizar en el desarrollo de la presente investigación.

Netbeans IDE

Netbeans es un Entorno de Desarrollo Integrado (IDE) libre y gratuito, hecho principalmente para el lenguaje de programación Java, pero puede servir para cualquier otro lenguaje de programación.

Dispone de soporte para crear interfaces gráficas de forma visual, desarrollo de aplicaciones web, control de versiones, colaboración entre varias personas, creación de aplicaciones compatibles con teléfonos móviles, resaltado de sintaxis, entre otros. Es una herramienta para que los programadores puedan escribir, compilar, depurar y ejecutar programas.

Para el desarrollo de la solución se utilizará Netbeans en su versión 7.4, debido a que ofrece un

rendimiento significativamente mejorado con respecto a versiones anteriores y brinda una amplia documentación. Incluye características como el detector de errores y posibles soluciones para los mismos, autocompletado de código y ordenamiento de código, legible y fácil de entender. Así como mejoras en Java EE, Maven, C/C + + y PHP. Además, se integra con la herramienta de modelado Visual Paradigm, brindando de esta forma una gran ventaja al programador [24].

Visual Paradigm for UML

Para el modelado se utilizará la herramienta CASE Visual Paradigm en su versión 8.0 ya que soporta el ciclo de vida completo del desarrollo de *software*: análisis y diseño orientado a objetos, construcción, pruebas y despliegue. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Esta herramienta CASE también proporciona abundantes tutoriales, demostraciones interactivas y proyectos UML [25].

RapidSVN

RapidSVN en su versión 0.12, es un cliente de interfaz gráfica para la comunicación con servidores Subversion. Está escrito en C++ y distribuido bajo licencia GPL. Facilita el versionado de ficheros, desde una interfaz sencilla e intuitiva y se encuentra disponible para plataformas Windows, GNU/Linux, MAC OS X y Solaris. Es una herramienta rápida y eficiente [26].

Pencil

Pencil en su versión 2.0.5 es una herramienta gratuita y de código abierto para diseñar gráficos y animarlos en dos dimensiones. La interfaz gráfica de Pencil dispone de los elementos básicos para crear dibujos y personalizarlos al estilo deseado. Se caracteriza por poseer soporte para dibujo de diagramas y exportar los dibujos a diferentes formatos de salida. Permite el vínculo entre páginas, debido a que los elementos de un dibujo se pueden vincular a una página específica en el mismo documento [27].

Conclusiones Parciales.

En este capítulo se analizaron los conceptos claves en la administración del servidor de base de datos PostgreSQL, análisis que permitió comprender mejor su funcionamiento y de esta forma seleccionar los parámetros con mayor relevancia para la administración del servicio. Además, se realizó un estudio de las principales herramientas que administran el servicio permitiendo obtener los aspectos positivos que contribuyeron a la concepción del módulo a desarrollar. Se definieron

las tecnologías a emplear en el desarrollo del módulo, el cual estará guiado por la metodología ágil de desarrollo de *software* SXP.

Capítulo 2: Análisis y diseño de la solución propuesta

En el presente capítulo se definirá el sistema propuesto donde se reflejarán los requisitos funcionales y no funcionales. Se recogen las funcionalidades de la solución propuesta reflejadas en las Historias de Usuario con sus prototipos de interfaces, además se define la arquitectura y los patrones de diseño a emplear.

Propuesta del módulo.

Con el propósito de darle solución al problema planteado, se propone desarrollar las funcionalidades necesarias para administrar el servicio PostgreSQL en HMAST. Dicho módulo permitirá administrar, a través de una interfaz de usuario, los parámetros de configuración del servidor de base de datos. Esto será posible debido a que la aplicación deberá conectarse vía SSH al servidor y acceder a los ficheros de configuración y los modificará de acuerdo a las necesidades del gestor de configuración. Entre las principales funcionalidades que contendrá la aplicación se encuentran instalar y desinstalar el servicio, gestionar los ficheros de configuración y el control de acceso. También posibilita garantizar la integridad de los datos, así como administrar el uso de los recursos que se le asignan al servidor, los registros del sistema y las estadísticas de ejecución. Además, realizar ajustes de consultas y revisiones periódicas a las tablas con modificaciones considerables.

Artefactos generados para el desarrollo

En el desarrollo del módulo guiado por la metodología SXP, se generaron artefactos como son: la Lista de Reserva del Producto, las Historias de Usuario y las Tareas de Ingeniería. A continuación se presenta como quedaron los mismos para la presente investigación.

Lista de Reserva del Producto (LRP)

La LRP contiene los requisitos agrupados y ordenados según su prioridad, definiendo así la planificación del trabajo a realizar. Durante el proceso de desarrollo de *software* (entre iteraciones) esta puede ser modificada y mejorada con nuevos requisitos, garantizando una mejor calidad del producto final. Para consultar la LRP ver Anexo 1: Lista de reserva del producto.

Los requisitos funcionales son las características o funcionalidades con las que cuenta el sistema. A continuación se muestran los 135 requisitos funcionales definidos por el cliente para la implementación del módulo. De estos 51 con prioridad alta y 84 tienen prioridad media.

Requisitos de prioridad alta

RF 1: Instalar el servicio PostgreSQL.

RF 2: Desinstalar el servicio PostgreSQL.

RF 3: Iniciar servicio PostgreSQL.

RF 4: Reiniciar servicio PostgreSQL.

RF 5: Detener servicio PostgreSQL.

RF 6: Mostrar el estado del servicio PostgreSQL.

RF 7: Mostrar ruta del directorio de almacenamiento de datos.

RF 8: Modificar ruta del directorio de almacenamiento de datos.

RF 9: Mostrar ruta del archivo de configuración para la autenticación.

RF 10: Modificar ruta del archivo de configuración para la autenticación.

RF 11: Mostrar ruta del archivo de configuración del usuario mapeo.

RF 12: Modificar ruta del archivo de configuración del usuario mapeo.

RF 13: Mostrar direcciones de escucha.

RF 14: Modificar direcciones de escuchas.

RF 15: Eliminar direcciones de escuchas.

RF 16: Mostrar puerto.

RF 17: Modificar puerto.

RF 18: Mostrar máximo de conexiones permitidas al PostgreSQL.

RF 19: Modificar máximo de conexiones permitidas al PostgreSQL.

RF 20: Mostrar número de conexiones reservadas para los superusuarios.

RF 21: Modificar número de conexiones reservadas para los superusuarios.

RF 22: Mostrar tiempo de espera para completar la autenticación del cliente.

RF 23: Modificar tiempo de espera para completar la autenticación del cliente.

RF 24: Habilitar la opción de conexiones SSL.

RF 25: Deshabilitar la opción de conexiones SSL.

RF 26: Habilitar el cifrado de contraseña.

RF 27: Deshabilitar el cifrado de contraseña.

RF 28: Habilitar la opción de ponerle el nombre del usuario por cada base de datos.

RF 29: Deshabilitar la opción de ponerle el nombre del usuario por cada base de datos.

RF 30: Mostrar cantidad de memoria que el servidor de base de datos utiliza para *buffers*.

RF 31: Modificar cantidad de memoria que el servidor de base de datos utiliza para *buffers*.

RF 32: Mostrar número de búferes temporales utilizados por cada sesión de base de datos.

RF 33: Modificar número de búferes temporales utilizados por cada sesión de base de datos.

RF 34: Mostrar número de transacciones que pueden estar en el estado preparado.

RF 35: Modificar número de transacciones que pueden estar en el estado preparado.

RF 36: Mostrar la cantidad de memoria que puede utilizar el servicio antes de crear archivos temporales para el procesamiento de los resultados intermedios.

RF 37: Modificar la cantidad de memoria que puede utilizar el servicio antes de crear archivos temporales para el procesamiento de los resultados intermedios.

RF 38: Mostrar la cantidad de memoria que se utilizará por las operaciones de mantenimiento.

RF 39: Modificar la cantidad de memoria que se utilizará por las operaciones de mantenimiento.

RF 40: Mostrar la cantidad de tiempo en que el proceso debe hacer una pausa.

RF 41: Modificar la cantidad de tiempo en que el proceso debe hacer una pausa.

RF 42: Mostrar el límite en que el proceso debe hacer una pausa.

RF 43: Modificar el límite en que el proceso debe hacer una pausa.

RF 44: Mostrar tiempo para activar la escritura en segundo plano.

RF 45: Modificar tiempo para activar la escritura en segundo plano.

RF 46: Mostrar la cantidad de búferes para la escritura en segundo plano.

RF 47: Modificar la cantidad de búferes para la escritura en segundo plano.

RF 48: Listar reglas de control de acceso.

RF 49: Crear reglas de control de acceso al servidor PostgreSQL.

RF 50: Modificar reglas de control de acceso al servidor PostgreSQL.

RF 51: Eliminar reglas de control de acceso al servidor PostgreSQL.

Requisitos de prioridad media

RF 52: Mostrar métodos para almacenar la cantidad de información que se escribe en el WAL.

RF 53: Modificar métodos para almacenar la cantidad de información que se escribe en el WAL.

RF 54: Habilitar que se escriban físicamente las actualizaciones en el disco.

RF 55: Deshabilitar que se escriban físicamente las actualizaciones en el disco.

RF 56: Habilitar la opción de que las transacciones se comprometen a esperar a los registros WAL que se escriben en el disco.

RF 57: Deshabilitar la opción de que las transacciones se comprometen a esperar a los registros WAL que se escriben en el disco.

RF 58: Mostrar método utilizado para forzar cambios WAL al disco.

RF 59: Modificar método utilizado para forzar cambios WAL al disco.

RF 60: Mostrar la cantidad de memoria compartida utilizada como búferes para los datos del WAL.

RF 61: Modificar la cantidad de memoria compartida utilizada como búferes para los datos del WAL.

RF 62: Mostrar la cantidad de segmentos que se chequearán.

RF 63: Modificar la cantidad de segmentos que se chequearán.

RF 64: Mostrar el tiempo en que se chequearán los segmentos.

RF 65: Modificar el tiempo en que se chequearán los segmentos.

RF 66: Mostrar el destino de finalización del punto de chequeo.

RF 67: Modificar el destino de finalización del punto de chequeo.

RF 68: Mostrar el costo de leer una única página de base de datos en disco de forma secuencial.

RF 69: Modificar el costo de leer una única página de base de datos en disco de forma secuencial.

RF 70: Mostrar el valor por el cual el planificador de consulta considera los accesos no secuenciales a disco.

RF 71: Modificar el valor por el cual el planificador de consulta considera los accesos no secuenciales a disco.

RF 72: Mostrar el tamaño efectivo de la caché de disco que está disponible para una sola consulta.

RF 73: Modificar el tamaño efectivo de la caché de disco que está disponible para una sola consulta.

RF 74: Habilitar la opción de optimización de la consulta genética.

RF 75: Deshabilitar la opción de optimización de la consulta genética.

RF 76: Mostrar el valor para que el optimizador de consulta planifique las consultas.

RF 77: Modificar el valor para que el optimizador de consulta planifique las consultas.

RF 78: Mostrar el valor que el planeador de consultas utilizará en las restricciones de tablas.

RF 79: Modificar el valor que el planeador de consultas utilizará en las restricciones de tablas.

RF 80: Mostrar los métodos que indican la salida a la que irán dirigidos los mensajes de registros.

RF 81: Modificar los métodos que indican la salida a la que irán dirigidos los mensajes de registros.

RF 82: Habilitar la opción del colector de registro.

RF 83: Deshabilitar la opción del colector de registro.

RF 84: Mostrar el directorio en el que se crearán los archivos de registros.

RF 85: Modificar el directorio en el que se crearán los archivos de registros.

RF 86: Mostrar formato de los archivos de registros creados.

RF 87: Modificar formato de los archivos de registros creados.

RF 88: Mostrar el tiempo para actualizar los registros.

RF 89: Modificar el tiempo para actualizar los registros.

RF 90: Mostrar el tamaño de un archivo de registro.

RF 91: Modificar el tamaño de un archivo de registro.

RF 92: Habilitar la opción de sobrescribir un archivo de registro.

RF 93: Deshabilitar la opción de sobrescribir un archivo de registro.

RF 94: Mostrar niveles de mensajes que se envían al cliente.

RF 95: Modificar niveles de mensajes que se envían al cliente.

RF 96: Mostrar niveles de mensajes que se escriben en el servidor.

RF 97: Modificar niveles de mensajes que se escriben en el servidor.

RF 98: Mostrar los errores que se registren en el servidor.

RF 99: Modificar los errores que se registren en el servidor.

RF 100: Mostrar la opción de registrar las consultas que hayan durado más tiempo.

RF 101: Modificar la opción de registrar las consultas que hayan durado más tiempo.

RF 102: Mostrar el nivel de detalle que se incluye en cada mensaje añadido al servidor.

RF 103: Modificar el nivel de detalle que se incluye en cada mensaje añadido al servidor.

RF 104: Mostrar qué sentencias SQL se registrarán.

RF 105: Modificar qué sentencias SQL se registrarán.

RF 106: Habilitar la opción de recopilar la información sobre el comando que se está ejecutando.

RF 107: Deshabilitar la opción de recopilar la información sobre el comando que se está ejecutando.

RF 108: Mostrar el número de bytes reservados para mostrar la consulta que se está ejecutando.

RF 109: Modificar el número de bytes reservados para mostrar la consulta que se está ejecutando.

RF 110: Habilitar la recopilación de estadísticas sobre la actividad de las base de datos.

RF 111: Deshabilitar la recopilación de estadísticas sobre la actividad de las base de datos.

RF 112: Mostrar seguimientos de las estadísticas.

RF 113: Modificar seguimientos de las estadísticas.

RF 114: Mostrar directorio para guardar los datos de las estadísticas.

RF 115: Modificar directorio para guardar los datos de las estadísticas.

RF 116: Habilitar la opción de ejecutar el demonio lanzador autovacuum.

RF 117: Deshabilitar la opción de ejecutar el demonio lanzador autovacuum.

RF 118: Mostrar el número máximo de procesos autovacuum.

RF 119: Modificar el número máximo de procesos autovacuum.

RF 120: Mostrar el tiempo que transcurre entre cada autovacuum.

RF 121: Modificar el tiempo que transcurre entre cada autovacuum.

RF 122: Mostrar la cantidad de filas para ejecutar VACUUM.

RF 123: Modificar la cantidad de filas para ejecutar VACUUM.

RF 124: Mostrar la cantidad de filas para ejecutar ANALYZE.

RF 125: Modificar la cantidad de filas para ejecutar ANALYZE.

RF 126: Mostrar el por ciento del tamaño de tabla que lanzará el VACUUM.

RF 127: Modificar el por ciento del tamaño de tabla que lanzará el VACUUM.

RF 128: Mostrar la cantidad de transacciones para que se ejecute un VACUUM aunque se haya desactivado el autovacuum.

RF 129: Modificar la cantidad de transacciones para que se ejecute un VACUUM aunque se haya desactivado el autovacuum.

RF 130: Mostrar el por ciento del tamaño de tabla que lanzará el ANALYZE.

RF 131: Modificar el por ciento del tamaño de tabla que lanzará el ANALYZE.

RF 132: Mostrar el tiempo que el proceso de VACUUM automático debe dormir entre ejecuciones.

RF 133: Modificar el tiempo que el proceso de VACUUM automático debe dormir entre ejecuciones.

RF 134: Mostrar el número de operaciones que pueden llevar a cabo los procesos del autovacuum.

RF 135: Modificar el número de operaciones que pueden llevar a cabo los procesos del autovacuum.

Los requisitos no funcionales se refieren a cualidades que imponen restricciones en el diseño y la implementación. A continuación se muestran los 6 requisitos no funcionales definidos para la implementación del módulo.

De diseño:

- Usar como lenguaje de programación Java y el *framework* Spring.
- Utilizar Netbeans como entorno de desarrollo integrado.
- Utilizar HTML5 para la interfaz de la aplicación.

De *software*:

- Disponer para el SO GNU/Linux de los siguientes paquetes: *java, nmap, openssh, gksu2*.

- La aplicación se ejecutará en sistemas operativos libres.

De seguridad:

- Las conexiones con los servidores deben ser mediante SSH2 y las contraseñas no serán guardadas.

Historias de Usuario (HU)

En la metodología SXP, las HU constituyen la técnica utilizada con el propósito de especificar los requisitos del *software*. Sirven de guía en la construcción de las pruebas de aceptación y son utilizadas para estimar tiempos de desarrollo. En este sentido, solo proveen detalles suficientes para hacer una estimación razonable del tiempo que llevará implementarlas [28]. Son escritas en lenguaje natural, sin un formato predeterminado, no excediendo su tamaño de unas pocas líneas de texto, guían la construcción de las pruebas de aceptación y son utilizadas para estimar tiempos de desarrollo.

A continuación se describen las HU correspondientes para el *software* a implementar, teniendo en cuenta los requisitos funcionales asociados a cada una de ellas. Para un mejor entendimiento ver Anexo 2: Historias de usuario.

Tabla 1: Listado de Historias de Usuarios

Número	**Nombre de la HU**	**Requisitos asociados**	**Descripción**
HMAST_**PostgreSQL**_1	Instalar y desinstalar el servicio PostgreSQL.	1 y 2.	El *software* permitirá instalar y desinstalar el servicio PostgreSQL.
HMAST_**PostgreSQL**_2	Administrar el servicio PostgreSQL.	3, 4, 5 y 6.	El sistema brindará la posibilidad de Iniciar, Detener, Reiniciar y Recargar el servicio.
HMAST_**PostgreSQL**_3	Configuración de la localización de los directorios de instalación del servicio PostgreSQL.	7, 8, 9, 10, 11 y 12.	El sistema brindará la posibilidad de mostrar y modificar la ruta de los directorios de almacenamiento de datos, el de configuración para la autenticación y el de

			configuración del usuario que viene por defecto, se guardarán en el directorio de datos de la agrupación de base de datos.
HMAST_**PostgreSQL**_4	Configuración de la conexión y autenticación.	13, 14, 15, 16, 17, 18,19, 20, 21, 22, 23,24,25,26,27,28, 29.	El software será capaz de gestionar las direcciones de escucha por las que el servidor escuchará las peticiones de los clientes. Permitirá mostrar y modificar el puerto por el cual el servicio aceptará conexiones, así como la cantidad de conexiones reservadas para los superusuarios. También brindará la posibilidad de configurar el tiempo máximo para que un cliente complete la autenticación y permitirá habilitar y deshabilitar las conexiones SSL y la encriptación de las contraseñas.
HMAST_**PostgreSQL**_5	Configuración de uso de recursos.	30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46 y 47.	El sistema brindará la posibilidad de realizar una serie de configuraciones en el uso de recursos del servicio como son: mostrar y modificar la cantidad de memoria que el servidor de base de datos utiliza para bufers, el número de búferes temporales utilizados por cada sesión de base de datos, el número de transacciones que pueden estar en el estado preparado, la cantidad de memoria que puede utilizar el servicio antes de crear archivos temporales para el procesamiento de los resultados intermedios.
HMAST_**PostgreSQL**_6	Gestionar acceso al servidor.	48, 49,50 y 51	El sistema permitirá gestionar las reglas para la conexión al

			servidor.
HMAST_**PostgreSQL**_7	Configuración para la integridad de los datos.	52 ,53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66 y 67.	El sistema brindará la posibilidad de realizar una serie de configuraciones para guardar toda la información sobre las transacciones y cambios realizados en las base de datos, así como para garantizar la integridad de los mismos como son: mostrar y modificar los métodos para guardar la cantidad de información que se escribe en el WAL, si la transacción se compromete a esperar a los registros WAL que se escriben en el disco. También permitirá habilitar y deshabilitar que se escriban las actualizaciones en el disco, todo esto se realiza para evitar que si el servidor sufre un accidente se pierdan los datos del mismo.
HMAST_**PostgreSQL**_8	Configuración del ajuste de consulta.	68, 69, 70, 71, 72, 73, 74,75,76,77,78 y 79.	El sistema brindará la posibilidad de realizar una serie de configuraciones del planificador de consultas, esto permitirá que se modifique el comportamiento por defecto del planificador, impidiendo que realice ciertas operaciones como: búsqueda por índices, ordenamiento, entre otras. Algunas de estas configuraciones son: mostrar y modificar el valor del costo de leer una única página de bases de dato en disco de forma secuencia, el valor por el cual el planificador de consulta considera los accesos no secuenciales a disco. También

			permite mostrar y modificar el tamaño efectivo de la caché de disco que está disponible para una sola consulta, el valor para que el optimizador de consulta planifique las consultas y el valor que el planeador de consultas utilizará en las restricciones de tablas. Además, permitirá activar y desactivar la optimización de la consulta genética.
HMAST_**PostgreSQL**_9	Configurar los registros del sistema.	80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90,91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104 y 105.	El *software* permite activar o desactivar la generación de registros en el servidor. En caso de estar activado permite establecer, mostrar y modificar la ubicación de los registros en el servidor.
HMAST_**PostgreSQL**_10	Configurar las estadísticas de ejecución.	106, 107, 108, 109, 110, 111, 112, 113, 114 y 115.	El sistema brindará la posibilidad de realizar una serie de configuraciones que permitirán saber qué estadísticas son recolectadas de forma constante por el servicio y así tener una idea más detallada sobre lo que está sucediendo en el servidor. La aplicación permitirá habilitar y deshabilitar si se recopila la información sobre el comando que se está ejecutando, así como la recopilación de estadísticas sobre la actividad de las base de datos. Además, muestra y modifica el número de *bytes* reservados para mostrar la consulta que se está ejecutando y el seguimiento de las funciones y el tiempo utilizado por el servicio

			para recopilar las estadísticas.
HMAST_**PostgreSQL**_11	Configurar parámetros para revisar periódicamente las tablas con modificaciones considerables.	116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134 y 135.	El sistema brinda la posibilidad de realizar una serie de configuraciones para revisar periódicamente las tablas con modificaciones considerables. Esta información es suministrada al recolector de estadísticas, para llevar a cabo las tareas de mantenimiento y análisis.

Arquitectura del módulo

Con el objetivo de mantener una homogeneidad entre los componentes del sistema, se define la utilización de la arquitectura establecida para HMAST (ver Figura 3). En la misma se incluye la distribución del módulo para la administración del servicio PostgreSQL entre las diferentes capas según las responsabilidades que estas establezcan. Además, es representado el flujo de información entre las diferentes capas del módulo, indicado por las flechas representadas. A continuación se muestra una representación de la arquitectura del módulo:

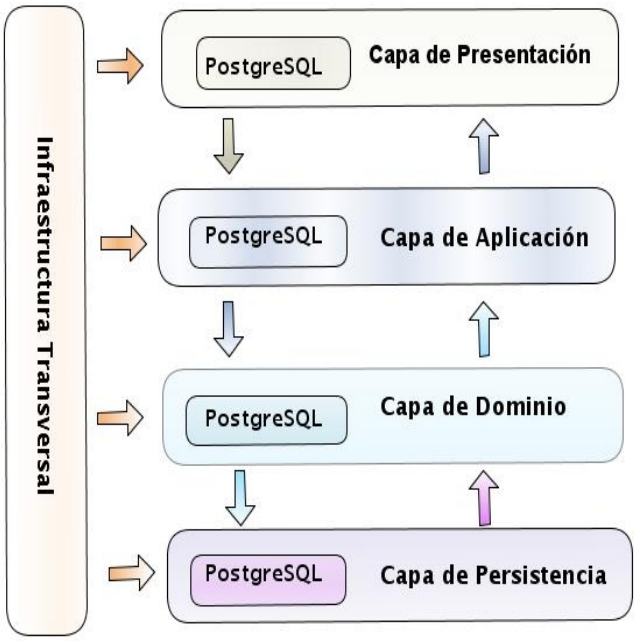


Figura 3: Arquitectura del módulo PostgreSQL.

Una de las ventajas asociadas al uso de una Arquitectura N-Capas orientada al Dominio, es que facilita el mantenimiento de la aplicación, debido a que las responsabilidades están perfectamente

distribuidas en las diferentes capas de la misma. Además, es recomendable para aquellos proyectos cuya lógica del negocio esté sujeta a constantes cambios y mantenimientos. La arquitectura propuesta permite la interoperabilidad en entornos distribuidos con un nivel de abstracción superior aumentando el bajo acoplamiento y la alta cohesión.

Diagrama de paquetes

El Diagrama de paquetes muestra la forma en que un sistema está dividido en agrupaciones lógicas mostrando las dependencias entre las mismas. Dado que normalmente un paquete está pensado como un directorio, el Diagrama de paquetes suministra una descomposición de la jerarquía lógica de un sistema.

Se diseñó el Diagrama de paquetes del módulo de PostgreSQL como se muestra en la Figura 4, tomando como marco de referencia la arquitectura propuesta. Este cuenta con un paquete representando cada capa y dentro de este uno llamado postgresql que contiene todo lo referente al módulo.

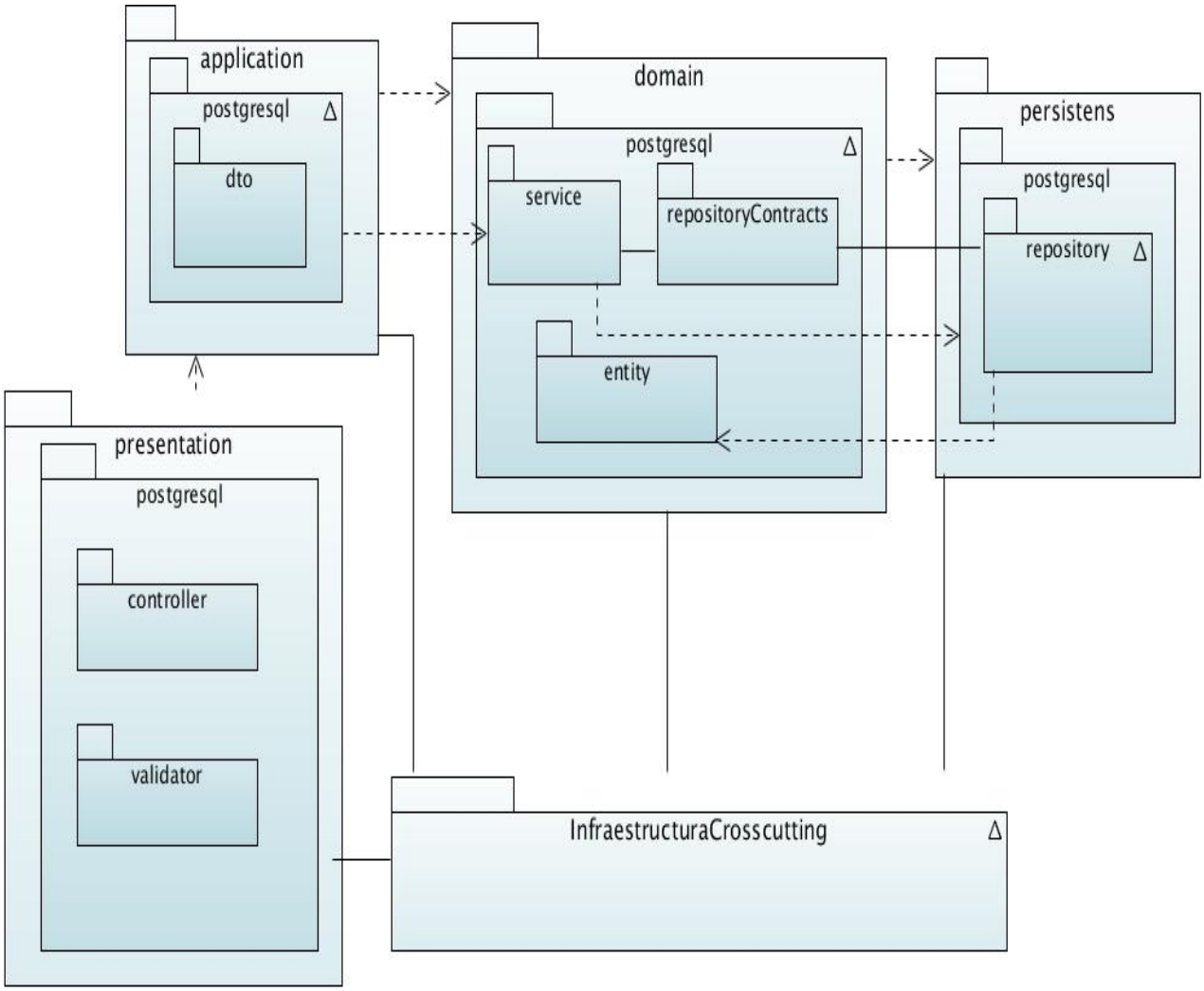


Figura 4: Diagrama de Paquetes.

En las siguientes figuras (5, 6, 7, 8, 9) se representa la distribución de las clases para la implementación de la HU Gestionar acceso al servidor.

La capa Presentación (*presentation*) (Ver Figura 5) es la responsable de presentar al usuario los conceptos de negocio mediante una interfaz de usuario (IU), facilitar la explotación de dichos procesos, informar sobre la situación de los procesos de negocio e implementar las reglas de validación.

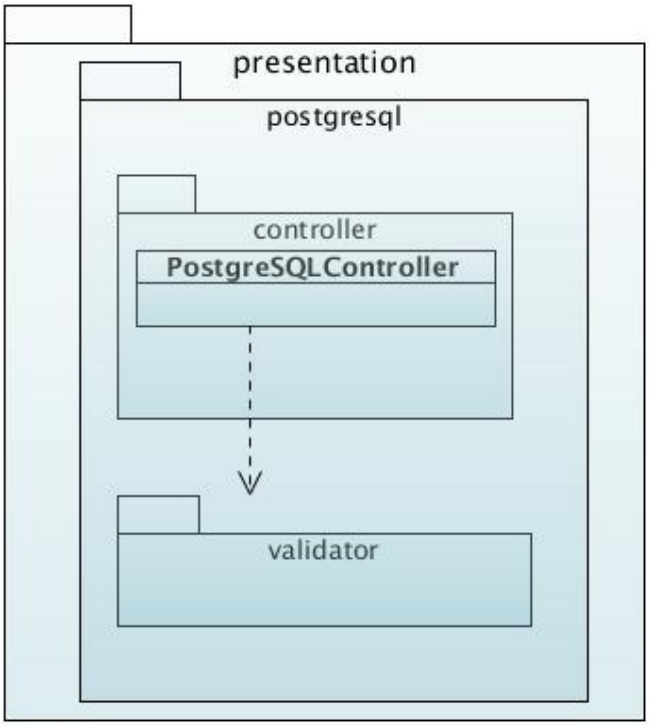


Figura 5: Diagrama de paquetes correspondiente a la capa Presentación

En la capa Aplicación (*application*) (Ver Figura 6) se encuentra la clase interfaz IRuleAppService que contiene métodos que permiten el acceso desde la capa Presentación, los cuales son implementados por la clase RuleAppService. Esta última es la responsable de adaptar la información que le llega desde la interfaz de usuario, a los requerimientos de los servicios del dominio, estos servicios son accedidos a través del atributo IRuleService contenido en esta clase. Esta capa contiene además, el paquete DTO *(Data Object Transfer)*, el mismo almacena clases que representan la información que llega desde la capa Presentación.

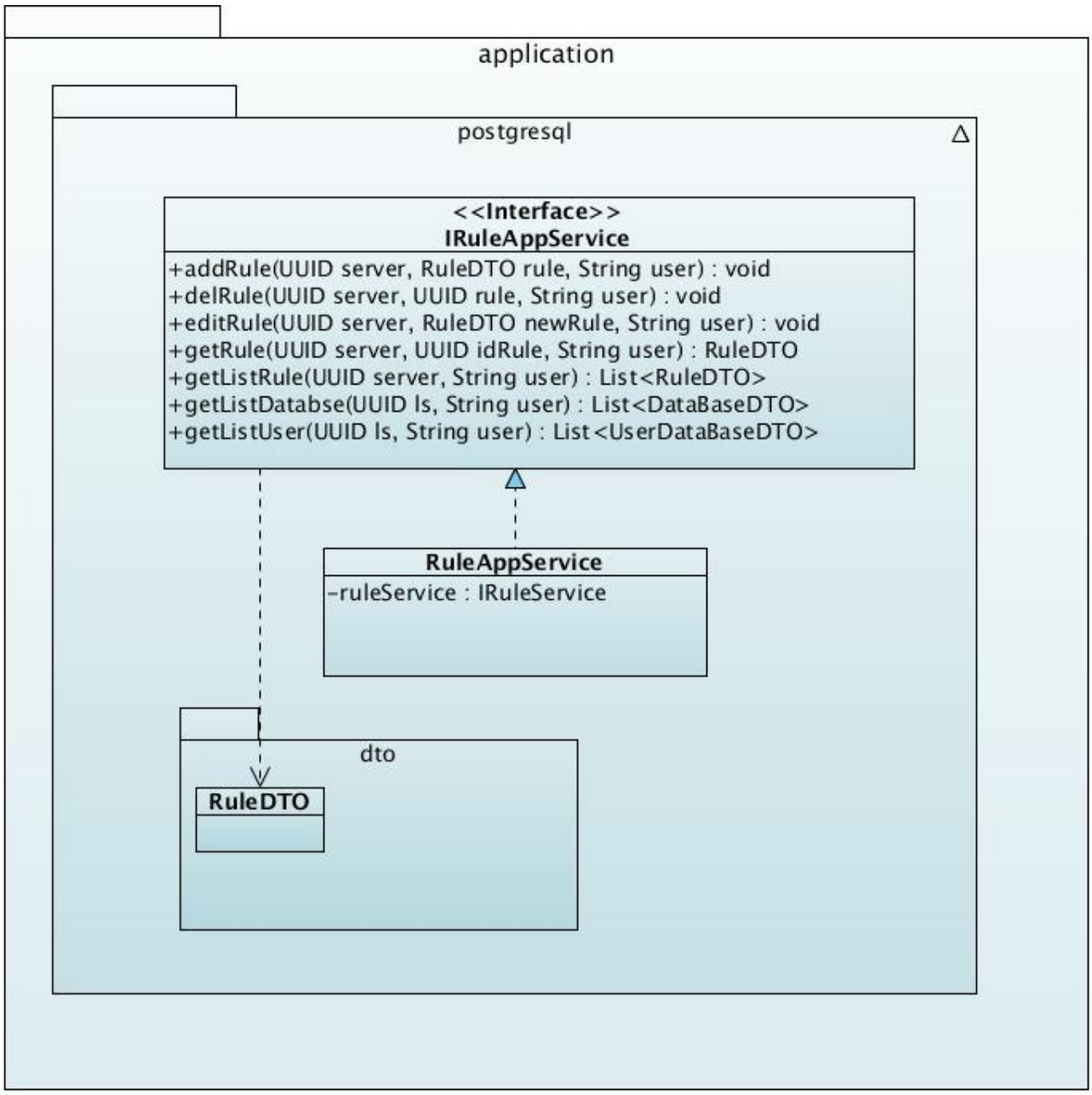


Figura 6: Diagrama de paquetes correspondiente a la capa Aplicación

La capa Dominio *(domain)* (Ver Figura 7) contiene los subpaquetes Entidades *(entitys)*, Servicio *(service)* y Contratos de repositorios *(repositoryscontrats)*. En Entidades se encuentra la clase Rule, quien representa las reglas de control de acceso definidas en el servidor PostgreSQL. El subpaquete Servicio contiene la clase *IRuleService* que define métodos que son accedidos desde la capa Aplicación, implementados por la clase *RuleService***,** quien es la encargada de realizar las validaciones de los datos antes de realizar las operaciones en el repositorio. El acceso al

repositorio se realiza a través del atributo *IRuleRepository* encontrado en esta clase. En el subpaquete Contratos de repositorios se encuentra la clase *IRruleRepository*, la cual define los métodos que son accedidos desde el subpaquete Servicio y son implementados por la clase *PostgreSQLRepository* encontrada en la capa de persistencia de los datos (*persistens*) (Ver Figura 8). Esta clase es responsable de cargar y salvar la información en los repositorios.

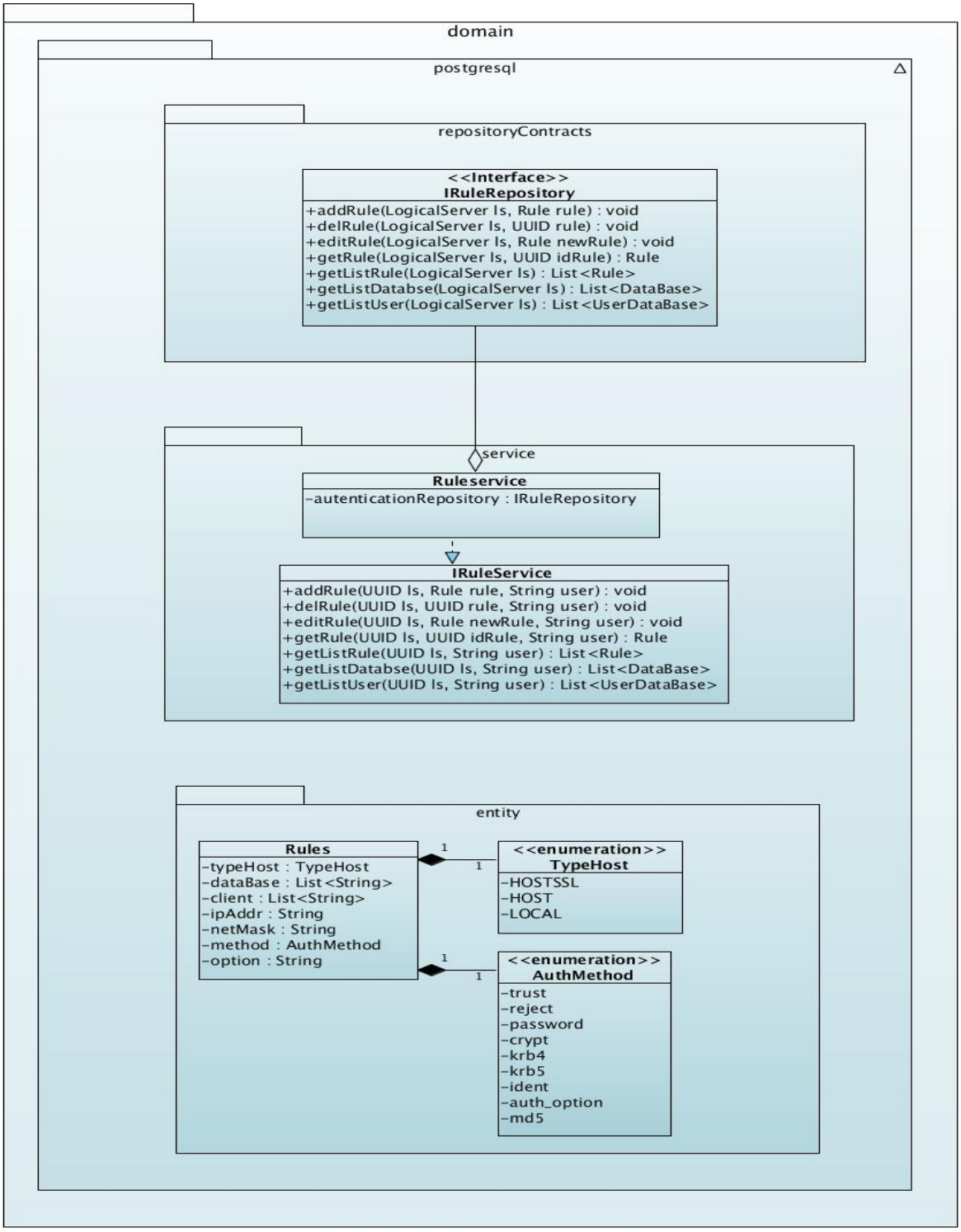


Figura 7: Diagrama de paquetes correspondiente a la capa Dominio

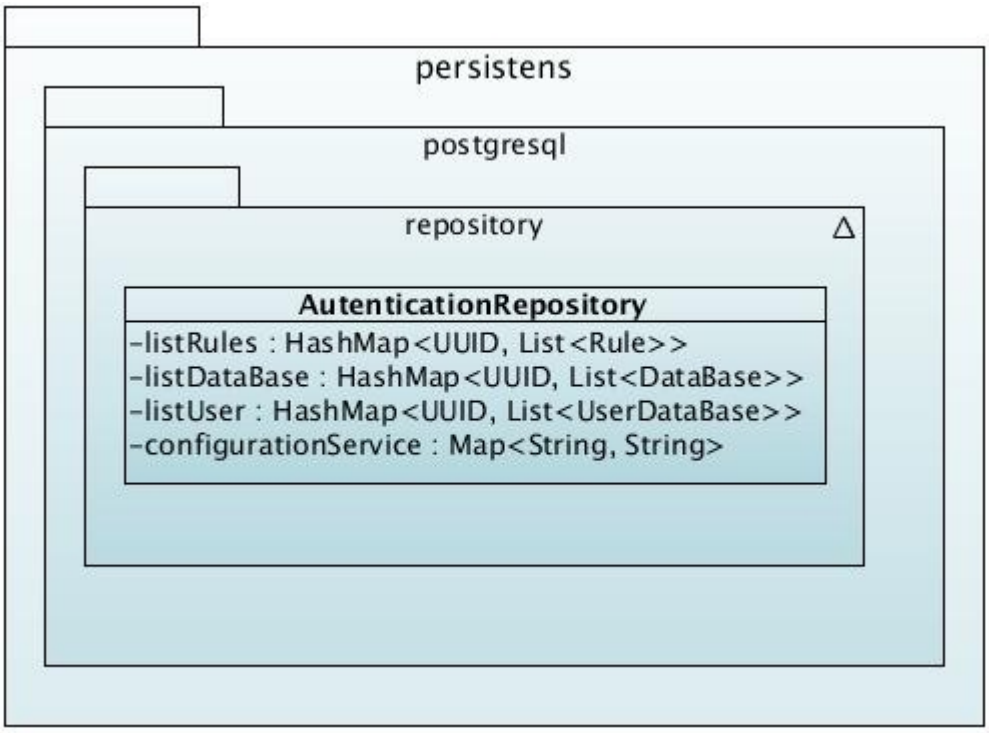


Figura 8: Diagrama de paquetes correspondiente a la capa Persistencia

La Figura 9 representa la parte del diagrama de paquetes correspondiente a la capa de Infraestructura Transversal, que contiene la entidad *Entity* la cual es padre de todas las entidades definidas en Dominio. En esta, en caso necesario, se incluyen además clases que sean reutilizables por otras capas.

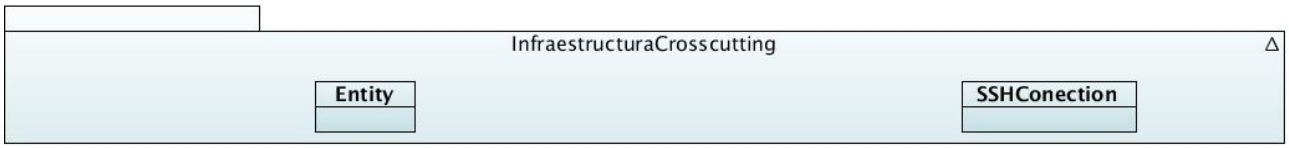


Figura 9: Diagrama de paquetes correspondiente a la capa Infraestructura Transversal

Las demás clases que reflejan el resto de las HU, independientemente de las funcionalidades que estas contengan, presentan el mismo flujo de comunicación dentro de cada uno de los paquetes como quedó descrito anteriormente.

Patrones de diseño

Los patrones de diseño son principios generales de soluciones que aplican ciertos estilos que ayudan a la creación de *software*. Brindan una solución ya probada y documentada a problemas de desarrollo de *software* que están sujetos a contextos similares [29].

Patrones GRASP

GRASP es un acrónimo que significa *General Responsibility Asignment Software Patterns,* del español Patrones Generales de *Software* para Asignar Responsabilidades. Los patrones GRASP describen los principios fundamentales para la asignación de responsabilidades a objetos, expresado en forma de patrones [30]. A continuación, son descritos los de este tipo que son utilizados para el diseño de la solución propuesta.

- Experto: Es un patrón que se usa más que cualquier otro al asignar responsabilidades. Es un principio básico que suele ser útil en el diseño orientado a objetos. Con el uso de este patrón se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide. Esto soporta un bajo acoplamiento, lo que favorece al hecho de tener sistemas más robustos y de fácil mantenimiento. Con la utilización de este patrón, se hizo posible definir dónde colocar en cada clase del módulo las funcionalidades que necesitan de esa información, dicha clase sería el experto en información. En la aplicación se necesita devolver y editar un objeto de tipo *Authentication* que define las opciones de acceso al servidor. En este caso la clase PostgreSQL que contiene un objeto de tipo *Authentication* es la experta en información y por tanto, la encargada de realizar las operaciones anteriormente mencionadas sobre el objeto.

- Creador: Es un patrón que tiene responsabilidades relacionadas con la creación de objetos. El propósito fundamental del mismo, es encontrar un creador que se debe conectar con el objeto producido en cualquier evento. En la aplicación el uso de este patrón se evidencia cuando las clases presentes en el dominio crean instancias de las entidades. Un ejemplo es cuando la clase *RuleService* adiciona un objeto de tipo Rule.

- Bajo acoplamiento: El acoplamiento es una medida de la fuerza con que una clase está conectada a otras. Acoplamiento bajo significa que una clase no depende de muchas otras. El uso de este patrón permite que las clases no se afecten por cambios de otros componentes, haciendo posible que sean fáciles de entender y de reutilizar.

- Alta cohesión: La cohesión es una medida de cuán relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Se hizo necesario la utilización de este patrón en el *software* en cuestión con el fin de controlar la complejidad de cada clase utilizada para mantener un buen comportamiento de las mismas, por esta razón, las que se identificaron con una amplia cantidad de funcionalidades, se dividieron en otras clases siguiendo el propósito de distribuir de forma equitativa el peso de la complejidad, manteniendo además, la coherencia entre ellas. Los patrones de alta cohesión y bajo acoplamiento se evidencian en el uso de la inyección de dependencias, la cual consiste en hacer que las clases del *software* sean independientes comunicándose únicamente a través de interfaces.

Patrones GOF

GoF, acrónimo de *Gang o Four*, Pandilla de los Cuatro, en su traducción al español. Los patrones de diseño, conocidos como GoF se clasifican en tres grandes categorías basadas en su propósito: creacionales, estructurales y de comportamiento [31]. A continuación se detalla el utilizado en el diseño de la solución.

- Patrón Solitario (*Singleton*): Garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. Es usado debido a la necesidad de trabajar con el mismo objeto en distintos momentos. Este patrón se evidencia en la conexión a un servidor ya que con una única instancia del objeto SSHConection, presente en el paquete infrastructureCrosscutting se realizan todas las operaciones sobre el servidor.

Conclusiones Parciales

El módulo a implementar tendrá un total de 135 funcionalidades, las cuales se agruparon en un total de 11 Historias de Usuario y 6 requisitos no funcionales. La descripción de las Historias de Usuario permitió un mejor entendimiento entre el programador y el cliente (Departamento SIMAYS), llegando a un acuerdo sobre las capacidades y cualidades con las que el *software* debe cumplir. Además, la correcta definición de los patrones, tanto de arquitectura como de diseño, permitió que se obtuviera una línea base del sistema capaz de soportar posteriores cambios en los requerimientos.

Capítulo 3: Implementación y pruebas del módulo

En el presente capítulo se refleja el flujo de actividades que conforman la implementación y pruebas de la solución propuesta. Describiéndose el estándar de código, las Tareas de Ingeniería, y las pruebas realizadas.

Planificación de la implementación

El Plan de Liberación (Plan de *Realase*) contiene las iteraciones que se van a realizar y las características correspondientes a cada una. Se definen las Historias de Usuario que se van a desarrollar en cada iteración y la estimación del tiempo en semanas. Teniendo en cuenta lo expuesto anteriormente se decide implementar el sistema en dos iteraciones, como se muestra seguidamente. Para mayor comprensión ver Anexo 3: Planificación de la implementación.

- **Iteración 1**: tiene como objetivo dar cumplimiento a las HU de la 1 a la 6. Estas han sido calificadas de prioridad alta, pues son de gran relevancia para la aplicación, ya que ellas constituyen el eje central de la estructura básica del sistema. Además, del correcto funcionamiento de ellas dependen el resto de las funcionalidades. En esta primera etapa se realizará la instalación y configuración del módulo.

- **Iteración 2**: tiene como objetivo dar cumplimiento a las Historias de Usuario de la 7 a la 11. Con la implementación de estas se culmina con las funcionalidades secundarias del sistema propuesto, permitiendo la administración del servicio PostgreSQL. Una vez terminada esta iteración se tendrá la aplicación lista para su entrega.

Tareas de Ingeniería

Las Tareas de Ingeniería se generan como artefactos en la metodología SXP, proporcionando un mejor entendimiento en el proceso de implementación ya que organizan el mismo con la definición de actividades asociadas a las Historias de Usuario. A continuación se mencionan las Tareas de Ingeniería correspondientes a cada una de las Historias de Usuario especificadas en el capítulo anterior. Para mayor comprensión ver Anexo 4: Tareas de ingeniería.

Tabla 2: Listado de Tareas de Ingeniería

HU asociadas	Tareas de Ingeniería
HMAST_PostgreSQL_1	Comprobar comando para realizar la instalación del servidor.
	Creación de un método para realizar la instalación del servidor.
	Diseño de interfaz para realizar la instalación y administración del servicio PostgreSQL.

HMAST_PostgreSQL_2	Comprobar el comando para administrar el servicio PostgreSQL.
HMAST_PostgreSQL_3	Comprobar el formato del archivo principal de configuración donde se guardan las configuraciones realizadas en el servidor.
HMAST_PostgreSQL_4	Comprobar las directivas que permiten la configuración de la conexión del servidor.
	Diseño de una interfaz que permita la configuración de la conexión y autenticación en el servidor.
HMAST_PostgreSQL_5	Diseño de una interfaz que permita la configuración de los parámetros de uso de recursos.
HMAST_PostgreSQL_6	Comprobar el formato de las reglas de conexión en el fichero de configuración pg_hba.conf del servidor.
	Comprobar las reglas de conexión de tipo de Host hostssl.
	Comprobar las reglas de conexión de tipo de Host host.
	Comprobar las reglas de conexión de tipo de Host local.
	Implementación de un método que permita crear o modificar una regla de conexión así como sus valores.
	Diseño de una interfaz que permita gestionar las reglas de conexión al servidor.
	Diseño de una interfaz que permita adicionar o modificar una regla de conexión.
HMAST_PostgreSQL_7	Diseño de una interfaz que permita la configuración de los parámetros para la integridad de los datos.
HMAST_PostgreSQL_8	Diseño de una interfaz que permita la configuración del planificador de consultas.
HMAST_PostgreSQL_9	Comprobar las directivas relacionadas con las configuraciones de formato de los registros del sistema del servidor PostgreSQL.
HMAST_PostgreSQL_10	Diseño de una interfaz que permita la configuración de las estadísticas de ejecución.
HMAST_PostgreSQL_11	Comprobar las directivas que permiten configurar los parámetros para revisar periódicamente las tablas con modificaciones considerables.

Estándares de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código. Si bien los programadores deben implementar un estándar de forma prudente, este debe tender siempre a lo práctico. Un código fuente completo debe reflejar un estilo armonioso, como si un único programador hubiera escrito todo el código de una sola vez [32].

Estándar para nombrar las clases

Los nombres de las clases deben comenzar con la primera letra en mayúscula y el resto se pondrá con minúscula, cuando sea un nombre compuesto se utilizará la notación Pascal.

Ejemplo: *QueryTuning*

Estándar para nombrar las funciones

El nombre a emplear para las funciones se escribe con la primera palabra en minúscula, en caso de que sea un nombre compuesto se utiliza la notación CamelCase, y con sólo leerlo se reconoce el propósito de la misma.

Ejemplo: *saveFileLocation*

Estándar para nombrar las variables

El nombre de estas variables se escribe la primera palabra con minúscula, si es un nombre compuesto se utilizará notación CamelCase.

Ejemplo: *seqPageCost*

Estándar para nombrar los componentes

Todos los paquetes comienzan con cu.uci.hmast.xxx.yyy.zzz.kkk

xxx →*nombre del módulo (squid3, dhcp3, postgresql).*

yyy →*presentation, application, domain, persistence.*

zzz →elementos que pueden contener los componentes verticales (*entitys, repositorys*).

kkk → subpaquetes.

Ejemplo: cu.uci.hmast.PostgreSQL.domain.postgreSQL.entitys

Pruebas de Software

El instrumento adecuado para determinar el estado de la calidad de un producto de *software* es el proceso de pruebas. En el mismo se ejecutan pruebas dirigidas a componentes del sistema de *software* en su totalidad, con el objetivo de medir el grado en que la aplicación cumple con los requerimientos [33]. Las pruebas propuestas por la metodología SXP son conocidas como pruebas de aceptación destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida diseñada por el cliente final. Además, se realizan pruebas unitarias las

cuales se encargan de verificar el código y son diseñadas por los programadores, sin embargo la metodología usada no requiere que sean documentadas.

Pruebas de Aceptación

Las pruebas de aceptación también conocidas como pruebas del cliente, son utilizadas para probar que las HU han sido implementadas de forma correcta al final de cada iteración y por lo tanto comprueban que las funcionalidades del sistema se encuentran en relación con las HU [34]. A continuación se muestra un listado con las 237 pruebas de aceptación realizadas a la aplicación. Para una mejor comprensión de las mismas remitirse al artefacto Plantilla de Pruebas de Aceptación en el expediente de proyecto.

Tabla 3: Listado de las Pruebas de Aceptación para la primera iteración.

Historia de Usuario	Pruebas	**Evaluación de la Prueba**
HU 1: Instalar y desinstalar el servicio PostgreSQL.	Verificar que se instala el servicio PostgreSQL.	Satisfactoria
	Verificar que se desinstala el servicio PostgreSQL.	Satisfactoria
	Verificar que lanza excepción cuando el servicio no se puede instalar.	Satisfactoria
	Verificar que lanza excepción cuando el servicio no se puede desinstalar.	Satisfactoria
	Verificar que se abre la interfaz de administración cuando se instala el servicio.	Satisfactoria
	Verificar que la interfaz del módulo no se abre cuando el servicio no está instalado.	Satisfactoria
HU 2: Administrar el servicio PostgreSQL.	Verificar que se inicia el servicio PostgreSQL.	Satisfactoria
	Verificar que se lanza excepción cuando no se puede iniciar el servicio.	Satisfactoria
	Verificar que se reinicia el servicio PostgreSQL.	Satisfactoria
	Verificar que se lanza excepción cuando el servicio no se puede reiniciar.	Satisfactoria
	Verificar que se detiene el servicio PostgreSQL.	Satisfactoria
	Verificar que se lanza excepción cuando el servicio no se detiene.	Satisfactoria
	Verificar que se puede recargar la configuración del servicio.	Satisfactoria
	Verificar que se lanza excepción cuando el servicio no se puede recargar.	Satisfactoria

	Verificar que se muestra el estado del servicio.	Satisfactoria
HU 3: Configuración de la locación de los directorios de instalación del servicio PostgreSQL.	Verificar que se muestra la ruta del directorio de almacenamiento de los datos.	Satisfactoria
	Verificar que se modifica la ruta del directorio de almacenamiento de los datos.	Satisfactoria
	Verificar que se lanza excepción cuando la ruta de almacenamiento de datos es incorrecta.	Satisfactoria
	Verificar que se muestra la ruta del archivo de configuración para la autenticación basada en *host*.	Satisfactoria
	Verificar que se modifica la ruta del archivo de configuración para la autenticación basada en *host*.	Satisfactoria
	Verificar que se lanza excepción cuando la ruta del archivo de configuración para la autenticación basada en *host* es incorrecta.	Satisfactoria
	Verificar que se muestra la ruta del archivo de configuración del usuario mapeo.	Satisfactoria
	Verificar que se modifica la ruta del archivo de configuración del usuario mapeo.	Satisfactoria
	Verificar que se lanza excepción cuando la ruta del archivo de configuración del usuario mapeo es inválida.	Satisfactoria
	Verificar que se guardan los cambios realizados.	Satisfactoria
	Verificar que se pueden descartar los cambios que se realizaron en la interfaz.	Satisfactoria
HU 4: Configuración de la conexión y autenticación.	Verificar que se muestran las direcciones de escucha.	Satisfactoria
	Verificar que se modifican las direcciones de escucha.	Satisfactoria
	Verificar que se puede poner un IP v4 en las direcciones de escucha.	Satisfactoria
	Verificar que se puede pasar asterisco en las direcciones de escucha.	Satisfactoria
	Verificar que se puede pasar más de un ip en las direcciones de escucha.	Satisfactoria
	Verificar que se lanza excepción cuando se deja	Satisfactoria

	en blanco las direcciones de escucha.	
	Verificar que se lanza excepción cuando se elimina una dirección de escucha y no hay ninguna seleccionada.	Satisfactoria
	Verificar que se muestra el puerto de escucha.	Satisfactoria
	Verificar que se modifica el puerto de escucha.	Satisfactoria
	Verificar que se lanza excepción cuando el puerto de escucha introducido está siendo usado por otra aplicación.	Satisfactoria
	Verificar que se lanza excepción cuando el puerto de escucha se deja en blanco.	Satisfactoria
	Verificar que se lanza excepción cuando el puerto de escucha introducido es inválido.	Satisfactoria
	Verificar que se muestra el máximo de conexiones.	Satisfactoria
	Verificar que se modifica el máximo de conexiones.	Satisfactoria
	Verificar que se lanza excepción cuando el máximo de conexiones introducido es inválido.	Satisfactoria
	Verificar que se muestra el número de conexiones reservadas para los superusuarios al servidor PostgreSQL.	Satisfactoria
	Verificar que se modifica el número de conexiones reservadas para los superusuarios al servidor PostgreSQL.	Satisfactoria
	Verificar que se lanza excepción cuando el número de conexiones reservadas para los superusuarios al servidor PostgreSQL no es un número entero.	Satisfactoria
	Verificar que se lanza excepción cuando el número introducido en las conexiones reservadas es mayor que el de las conexiones permitidas.	Satisfactoria
	Verificar que se muestra el tiempo de espera para completar la autenticación del cliente.	Satisfactoria
	Verificar que se modifica el tiempo de espera para completar la autenticación del cliente.	Satisfactoria
	Verificar que se lanza excepción cuando el	Satisfactoria

	tiempo de espera para completar la autenticación del cliente es menor que cero o mayor que 600.	
	Verificar que se muestra el valor que tenga la directiva de las conexiones seguras al servidor.	Satisfactoria
	Verificar que se habilita las conexiones seguras al servidor.	Satisfactoria
	Verificar que se deshabilita las conexiones seguras al servidor.	Satisfactoria
	Verificar que se muestra el valor que tenga el cifrado de contraseña.	Satisfactoria
	Verificar que se habilita el cifrado de contraseña.	Satisfactoria
	Verificar que se deshabilita el cifrado de contraseña.	Satisfactoria
	Verificar que se muestra el valor de la directiva de poner un nombre de usuario por cada base de datos.	Satisfactoria
	Verificar que se habilita la opción de poner un nombre de usuario por cada base de datos.	Satisfactoria
	Verificar que se deshabilita la opción de poner un nombre de usuario por cada base de datos.	Satisfactoria
	Verificar que se pueden salvar todas las configuraciones en la interfaz.	Satisfactoria
	Verificar que se pueden descartar todas las modificaciones realizadas.	Satisfactoria
HU 5: Configuración de uso de recursos.	Verificar que se muestra el tamaño de memoria para *buffer*.	Satisfactoria
	Verificar que se modifica el tamaño de memoria para *buffer*.	Satisfactoria
	Verificar que se lanza excepción cuando la cantidad de memoria para el *buffer* no es válida.	Satisfactoria
	Verificar que se lanza excepción cuando la cantidad de memoria introducida es mayor que el 25 % de la RAM del servidor.	Satisfactoria
	Verificar que se muestra el número de *búferes* temporales utilizados por cada sección de base de datos.	Satisfactoria

	Verificar que se modifica el número de *búferes* temporales utilizados por cada sección de base de datos.	Satisfactoria
	Verificar que se lanza una excepción cuando el número de *búferes* es menor que 800 o mayor que 10240.	Satisfactoria
	Verificar que se muestra el número máximo de transacciones preparadas.	Satisfactoria
	Verificar que se modifica el número máximo de transacciones preparadas.	Satisfactoria
	Verificar que se lanza una excepción cuando el número máximo de transacciones preparadas introducido es menor que 0 o mayor que 700.	Satisfactoria
	Verificar que se muestra el valor de la cantidad de memoria que puede utilizar PostgreSQL antes de crear archivos temporales para el procesamiento de los resultados intermedios.	Satisfactoria
	Verificar que se modifica el valor de la cantidad de memoria que puede utilizar PostgreSQL antes de crear archivos temporales para el procesamiento de los resultados intermedios.	Satisfactoria
	Verificar que se lanza una excepción cuando la cantidad de memoria que puede utilizar PostgreSQL antes de crear archivos temporales para el procesamiento de los resultados intermedios introducido es menor que 64 o mayor que 1572864.	Satisfactoria
	Verificar que se muestra la cantidad máxima de memoria que se utilizará por las operaciones de mantenimiento.	Satisfactoria
	Verificar que se modifica la cantidad máxima de memoria que se utilizará por las operaciones de mantenimiento.	Satisfactoria
	Verificar que se lanza una excepción cuando el valor introducido en la cantidad máxima de memoria que se utilizará por las operaciones de mantenimiento es menor que 1 o mayor que 1536.	Satisfactoria

	Verificar que se muestra la cantidad de tiempo en que el proceso debe hacer una pausa.	Satisfactoria
	Verificar que se modifica la cantidad de tiempo en que el proceso debe hacer una pausa.	Satisfactoria
	Verificar que se lanza una excepción cuando la cantidad de tiempo en que el proceso debe hacer una pausa es menor que 0 ó mayor que 1000.	Satisfactoria
	Verificar que se muestra el límite para detener el proceso de vacío.	Satisfactoria
	Verificar que se modifica el límite para detener el proceso de vacío.	Satisfactoria
	Verificar que lanza una excepción cuando el límite para detener el proceso de vacío es menor que 1 o mayor que 10000.	Satisfactoria
	Verificar que se muestra el tiempo en el que se lanzará el proceso que se encarga de actualizar los cambios llevados a cabo.	Satisfactoria
	Verificar que se modifica el tiempo en el que se lanzará el proceso que se encarga de actualizar los cambios llevados a cabo.	Satisfactoria
	Verificar que se lanza una excepción cuando el tiempo en el que se lanzará el proceso que se encarga de actualizar los cambios llevados a cabo es menor que 10 ó mayor que 10000.	Satisfactoria
	Verificar que se muestra la cantidad de *búferes* que serán actualizados cuando se lance el proceso que se encarga de actualizar los cambios llevados a cabo.	Satisfactoria
	Verificar que se modifica la cantidad de *búferes* que serán actualizados cuando se lance el proceso que se encarga de actualizar los cambios llevados a cabo.	Satisfactoria
	Verificar que se lanza una excepción cuando la cantidad de *búferes* que serán actualizados cuando se lance el proceso que se encarga de actualizar los cambios llevados a cabo es menor que 0 o mayor que 1000.	Satisfactoria

	Verificar que se pueden salvar todos los cambios realizados en la interfaz.	Satisfactoria
	Verificar que se pueden descartar todos los cambios realizados en la interfaz.	Satisfactoria
HU 6: Gestionar acceso al servidor.	Verificar que se pueden crear reglas de control de acceso.	Satisfactoria
	Verificar que el tipo de conexión puede ser solo local, *host* y *hotssl*.	Satisfactoria
	Verificar que *database* solo puede ser all, *sameuser*, *name*.	Satisfactoria
	Verificar que en el campo de IP se puede pasar un ip o una subred.	Satisfactoria
	Verificar que se lanza una excepción cuando en el campo de IP no se entra una subred o un IP v4.	Satisfactoria
	Verificar que en la máscara se pueda pasar un ip v4.	Satisfactoria
	Verificar que se lanza una excepción cuando la máscara introducida no es ip v4.	Satisfactoria
	Verificar que el método de autenticación solo puede ser t*rust*, *reject, password, crypt, ident y auth_option*.	Satisfactoria
	Verificar que cuando el método de autenticación es *password*, se debe introducir una contraseña que se almacene en *pg_shadow.*	Satisfactoria
	Verificar que se pueden modificar las reglas de control de acceso.	Satisfactoria
	Verificar que se lanza una excepción cuando se intenta modificar una regla de control de acceso y no existe ninguna seleccionada.	Satisfactoria
	Verificar que cuando se edita una regla, los datos modificados aparecerán en la interfaz cuando se vuelva a acceder.	Satisfactoria
	Verificar que se puede eliminar una regla de control de acceso.	Satisfactoria
	Verificar que se lanza excepción cuando se intenta eliminar una regla y no está seleccionada.	Satisfactoria

Tabla 4: Listado de las Pruebas de Aceptación para la segunda iteración.

Historia de Usuario	Pruebas	**Evaluación de la Prueba**
HU 7: Configuración para la integridad de los datos.	Verificar que se muestran los métodos para guardar la información que se escribe en el WAL.	Satisfactoria
	Verificar que se modifican los métodos para guardar la información que se escribe en el WAL.	Satisfactoria
	Verificar que los métodos para guardar la información que se escriben en el WAL solo pueden ser *minimal, archive y host_stanby.*	Satisfactoria
	Verificar que muestre el valor que tenga la escritura en disco de las actualizaciones.	Satisfactoria
	Verificar que se habilita la escritura en el disco de las actualizaciones.	Satisfactoria
	Verificar que se deshabilita la escritura en disco de las actualizaciones.	Satisfactoria
	Verificar que se muestre el valor si después de que una transacción realice un *commit*, espera a guardar los registros WAL en disco antes de dar por válidas las operaciones.	Satisfactoria
	Verificar que se puede habilitar la opción referente a que después de que una transacción realice un commit, espera a guardar los registros WAL en disco antes de dar por válidas las operaciones.	Satisfactoria
	Verificar que se puede deshabilitar la opción de que después de que una transacción realice un commit, espera a guardar los registros WAL en disco antes de dar por válidas las operaciones.	Satisfactoria
	Verificar que se muestran los métodos para forzar los cambios WAL al disco.	Satisfactoria
	Verificar que se pueden modificar los métodos para forzar los cambios WAL al disco.	Satisfactoria
	Verificar que solo se pueden pasar los métodos: *open_datasync, fdatasync, fsync,*	Satisfactoria

	fsync_writethrough, open_sync.	
	Verificar que se muestra la cantidad de la memoria compartida utilizada como *búferes* para los datos del WAL.	Satisfactoria
	Verificar que se modifica la cantidad de la memoria compartida utilizada como *búferes* para los datos del WAL.	Satisfactoria
	Verificar que se lanza excepción cuando la cantidad de la memoria compartida utilizada como *búferes* para los datos del WAL que se introduce es menor que cero o mayor que 1 000 000 000.	Satisfactoria
	Verificar que se muestra el tiempo en que se realizarán los puntos de comprobación automáticos WAL.	Satisfactoria
	Verificar que se modifica el tiempo en que se realizarán los puntos de comprobación automáticos WAL.	Satisfactoria
	Verificar que se muestra la opción de seleccionar qué valor se quiere introducir, si es minuto, hora o segundo.	Satisfactoria
	Verificar que se lanza una excepción cuando el tiempo en que se realizarán los puntos de comprobación automáticos WAL es menor que 30 segundos o mayor que 60 segundos.	Satisfactoria
	Verificar que se muestra el valor en que se ejecutará uniformemente el punto de chequeo actual durante el período de espera del siguiente.	Satisfactoria
	Verificar que se modifica el valor en que se ejecutará uniformemente el punto de chequeo actual durante el período de espera del siguiente.	Satisfactoria
	Verificar que se lanza una excepción cuando el valor introducido no está entre 0.0 y 1.0.	Satisfactoria
	Verificar que se pueden salvar todas las configuraciones en el servidor.	Satisfactoria
HU 8: Configuración del	Verificar que se muestra el costo de leer una	Satisfactoria

planificador de consultas.	única página de base de datos en disco de forma secuencial cuando la información se encuentra contigua en él.	
	Verificar que se modifica el costo de leer una única página de base de datos en disco de forma secuencial cuando la información se encuentra contigua en él.	Satisfactoria
	Verificar que se lanza excepción cuando el valor del costo no está entre 0.0 y 10000000000.	Satisfactoria
	Verificar que se muestra el valor por el cual el planificador de consulta considera los accesos no secuenciales a disco.	Satisfactoria
	Verificar que se modifica el valor por el cual el planificador de consulta considera los accesos no secuenciales a disco.	Satisfactoria
	Verificar que se lanza excepción cuando el valor por el cual el planificador de consulta considera los accesos no secuenciales a disco es menor que 0 o mayor que 1 000 000 000.	Satisfactoria
	Verificar que se muestra el tamaño efectivo de la caché de disco que está disponible para una sola consulta.	Satisfactoria
	Verificar que se modifica el tamaño efectivo de la caché de disco que está disponible para una sola consulta.	Satisfactoria
	Verificar que se lanza excepción cuando el tamaño efectivo de la caché de disco que está disponible para una sola consulta es menor que 0.0 o mayor que 2048.	Satisfactoria
	Verificar que se muestra la optimización de la consulta genética.	Satisfactoria
	Verificar que se activa la optimización de la consulta genética.	Satisfactoria
	Verificar que se desactiva la optimización de la consulta genética.	Satisfactoria
	Verificar que se muestra el valor para que el optimizador de consulta planifique las consultas.	Satisfactoria
	Verificar que se modifica el valor para que el	Satisfactoria

	optimizador de consulta planifique las consultas.	
	Verificar que se lanza excepción cuando el valor para que el optimizador de consulta planifique las consultas es menor que 0 o mayor que 10000000000.	Satisfactoria
	Verificar que se muestra el valor que el planeador de consultas utilizará en las restricciones de tablas.	Satisfactoria
	Verificar que se muestra el valor que el planeador de consultas utilizará en las restricciones de tablas.	Satisfactoria
	Verificar que se desactiva el valor que el planeador de consultas utilizará en las restricciones de tablas.	Satisfactoria
	Verificar que se salvan desde la interfaz, los parámetros en el servidor remoto.	Satisfactoria
	Verificar que se descartan todas las configuraciones realizadas en la interfaz.	Satisfactoria
HU 9: Configurar los registros del sistema.	Verificar que se muestran los métodos que indican la salida a la que irán dirigidos los mensajes de *log*.	Satisfactoria
	Verificar que se modifican los métodos que indican la salida a la que irán dirigidos los mensajes de *log*.	Satisfactoria
	Verificar que solo se puedan pasar los métodos que indican la salida a la que irán dirigidos los mensajes de *log*.	Satisfactoria
	Verificar que se muestra el valor del colector de registros.	Satisfactoria
	Verificar que se habilita el valor del colector de registros.	Satisfactoria
	Verificar que se deshabilita el valor del colector de registros.	Satisfactoria
	Verificar que se muestra la ruta del directorio por defecto donde se guardarán los *logs*.	Satisfactoria
	Verificar que se modifica la ruta del directorio por defecto donde se guardarán los *logs*.	Satisfactoria
	Verificar que se lanza una excepción cuando es	Satisfactoria

	inválida la ruta del directorio por defecto donde se guardarán los *logs*.	
	Verificar que se muestra cómo están definidos los nombres que tienen los archivos de los *logs*.	Satisfactoria
	Verificar que se puede cambiar cómo están definidos los nombres que tienen los archivos de los *logs*.	Satisfactoria
	Verificar que se lanza una excepción cuando el nombre de los ficheros introducidos no es válido.	Satisfactoria
	Verificar que se muestra el tiempo de vida de los registros.	Satisfactoria
	Verificar que se modifica el tiempo de vida de los registros.	Satisfactoria
	Verificar que se lanza una excepción cuando el tiempo de vida de los *logs* es menor que cero o mayor que 100000.	Satisfactoria
	Verificar que se muestra el tamaño de los archivos de *logs*.	Satisfactoria
	Verificar que se modifica el tamaño de los archivos de *logs*.	Satisfactoria
	Verificar que se lanza una excepción cuando el tamaño de los archivos de *logs* es menor que 0 o mayor que 1GB.	Satisfactoria
	Verificar que se muestra el valor que tenga al sobrescribir un archivo de registro.	Satisfactoria
	Verificar que se habilita la opción de sobrescribir un archivo de registro.	Satisfactoria
	Verificar que se deshabilita la opción de sobrescribir un archivo de registro.	Satisfactoria
	Verificar que se muestran los niveles de los mensajes que se mostrarán a los clientes.	Satisfactoria
	Verificar que se modifican los niveles de los mensajes que se mostrarán a los clientes.	Satisfactoria
	Verificar que los niveles solo pueden ser: *DEBUG, INFO, NOTICE, WARNING, ERROR, LOG, FATAL, PANIC*.	Satisfactoria
	Verificar que se muestran los niveles de los	Satisfactoria

	mensajes que se escriben en el servidor.	
	Verificar que se modifican los niveles de los mensajes que se escriben en el servidor.	Satisfactoria
	Verificar que los niveles de los mensajes que se escriben en el servidor solo pueden ser: DEBUG5, DEBUG4, DEBUG3, DEBUG2, DEBUG1, INFO, AVISO, WARNING, ERROR LOG, FATAL, PANIC.	Satisfactoria
	Verificar que se muestra el nivel de los errores que se registran en el servidor.	Satisfactoria
	Verificar que se modifica el nivel de los errores que se registran en el servidor.	Satisfactoria
	Verificar que el nivel de los errores que se registran en el servidor solo pueden ser: *DEBUG5, DEBUG4, DEBUG3, DEBUG2, DEBUG1, INFO, AVISO, WARNING, ERROR LOG, FATAL, PANIC.*	Satisfactoria
	Verificar que se muestra el valor de si se registran las instrucciones que hayan durado más tiempo que el especificado.	Satisfactoria
	Verificar que se habilita el registro de las instrucciones que hayan durado más tiempo que el especificado.	Satisfactoria
	Verificar que se deshabilita el registro de las instrucciones que hayan durado más tiempo que el especificado.	Satisfactoria
	Verificar que se muestra el nivel de detalle que se incluye en cada mensaje añadido al *log* del servidor.	Satisfactoria
	Verificar que se modifica el nivel de detalle que se incluye en cada mensaje añadido al *log* del servidor.	Satisfactoria
	Verificar que el nivel de detalle que se incluye en cada mensaje añadido al *log* del servidor solo pueda ser *terse*, *default* y *verbose.*	Satisfactoria
	Verificar que se muestran las sentencias SQL que se registrarán en los *logs*.	Satisfactoria
	Verificar que se modifican las sentencias SQL	Satisfactoria

	que se registrarán en los *logs*.	
	Verificar que las sentencias SQL que se registrarán en los *logs* son *none*, *ddl*, *mod* y *all*.	Satisfactoria
	Verificar que todos los cambios se realizan en el servidor remoto.	Satisfactoria
	Verificar que se pueden descartar todos los cambios realizados en la interfaz.	Satisfactoria
HU 10: Configurar las estadísticas de ejecución.	Verificar que se muestra si está activada la recopilación de información sobre el comando que se está ejecutando de cada sesión.	Satisfactoria
	Verificar que se habilita la recopilación de información sobre el comando que se está ejecutando de cada sesión.	Satisfactoria
	Verificar que se deshabilita la recopilación de información sobre el comando que se está ejecutando de cada sesión.	Satisfactoria
	Verificar que se muestra si está activado la recopilación de estadísticas sobre la actividad de la base de datos.	Satisfactoria
	Verificar que se activa la recopilación de estadísticas sobre la actividad de la base de datos.	Satisfactoria
	Verificar que se desactiva la recopilación de estadísticas sobre la actividad de la base de datos.	Satisfactoria
	Verificar que se muestran las funciones y el tiempo utilizado.	Satisfactoria
	Verificar que se modifican las funciones y el tiempo utilizado.	Satisfactoria
	Verificar que las funciones y el tiempo utilizado solo pueden tomar los valores *pl, all y none*.	Satisfactoria
	Verificar que se muestra el número de *bytes* reservados para mostrar la consulta que se está ejecutando.	Satisfactoria
	Verificar que se modifica el número de *bytes* reservados para mostrar la consulta que se está ejecutando.	Satisfactoria
	Verificar que se lanza una excepción cuando el	Satisfactoria

	número de *bytes* reservados para mostrar la consulta que se está ejecutando es menor que cero o mayor que 2048.	
	Verificar que se muestra el tamaño reservado para rastrear el comando que se está utilizando.	Satisfactoria
	Verificar que se modifica el tamaño reservado para rastrear el comando que se está utilizando.	Satisfactoria
	Verificar que se lanza una excepción cuando el tamaño reservado para rastrear el comando que se está utilizando es menor que cero o mayor que 2048.	Satisfactoria
	Verificar que se muestra la ruta del directorio definido para almacenar las estadísticas.	Satisfactoria
	Verificar que se modifica la ruta del directorio definido para almacenar las estadísticas.	Satisfactoria
	Verificar que se lanza una excepción cuando la ruta del directorio definido para almacenar las estadísticas es inválida.	Satisfactoria
	Verificar que se pueden salvar todas las configuraciones de la interfaz en el servidor remoto.	Satisfactoria
	Verificar que se puede descartar todos los cambios realizados en la interfaz.	Satisfactoria
HU 11: Configurar parámetros para revisar periódicamente las tablas con modificaciones considerables.	Verificar que se muestra si se deberá ejecutar el demonio *autovacum*.	Satisfactoria
	Verificar que se activa que el servidor ejecute el demonio *autovacum*.	Satisfactoria
	Verificar que se desactiva la opción de que el servidor ejecute el demonio del lanzador *autovacum*.	Satisfactoria
	Verificar que se muestra el número máximo de procesos *autovacuum*.	Satisfactoria
	Verificar que se modifica el número máximo de procesos *autovacuum*.	Satisfactoria
	Verificar que se lanza una excepción cuando el número máximo de procesos *autovacuum* no está entre 0 y 100.	Satisfactoria
	Verificar que se muestra el tiempo mínimo que	Satisfactoria

	transcurre entre cada *autovacuum*.	
	Verificar que se modifica el tiempo mínimo que transcurre entre cada *autovacuum*.	Satisfactoria
	Verificar que se lanza excepción cuando el tiempo mínimo que transcurre entre cada *autovacuum* es menor que cero o mayor que 30 000.	Satisfactoria
	Verificar que se muestra el número mínimo de tuplas que serán actualizadas o eliminadas.	Satisfactoria
	Verificar que se modifica el número mínimo de tuplas que serán actualizadas o eliminadas.	Satisfactoria
	Verificar que se lanza una excepción cuando el número de tuplas introducido es menor que 0 o mayor que 1 000 000 000.	Satisfactoria
	Verificar que se muestra el número mínimo de tuplas insertadas, actualizadas o eliminadas.	Satisfactoria
	Verificar que se modifica el número mínimo de tuplas insertadas, actualizadas o eliminadas.	Satisfactoria
	Verificar que se lanza una excepción cuando el número mínimo de tuplas insertadas, actualizadas o eliminadas introducido es menor que 0 o mayor que 1 000 000 000.	Satisfactoria
	Verificar que se muestra la cantidad de transacciones para que se ejecute un *vacuum* aunque se haya desactivado el *autovacuum*.	Satisfactoria
	Verificar que se modifica la cantidad de transacciones para que se ejecute un *vacuum* aunque se haya desactivado el *autovacuum*.	Satisfactoria
	Verificar que se lanza una excepción cuando la cantidad de transacciones para que se ejecute un *vacuum* aunque se haya desactivado el *autovacuum* es menor que cero o mayor 200 000 000.	Satisfactoria
	Verificar que se muestra el tiempo que el proceso de VACUUM automático debe dormir entre ejecuciones.	Satisfactoria
	Verificar que se modifica el tiempo que el proceso de VACUUM automático debe dormir	Satisfactoria

	entre ejecuciones.	
	Verificar que se lanza una excepción cuando el tiempo que el proceso de VACUUM automático debe dormir entre ejecuciones es menor que -1 o mayor que 100.	Satisfactoria
	Verificar que se muestra el número de operaciones que pueden llevar a cabo los procesos del *autovacuum*.	Satisfactoria
	Verificar que se modifica el número de operaciones que pueden llevar a cabo los procesos del *autovacuum*.	Satisfactoria
	Verificar que se lanza una excepción cuando el número de operaciones que pueden llevar a cabo los procesos del *autovacuum* es menor que cero o mayor que 10 000.	Satisfactoria
	Verificar que se pueden salvar todas las configuraciones en un servidor remoto.	Satisfactoria
	Verificar que se pueden cancelar todas las configuraciones realizadas en la interfaz.	Satisfactoria

Al terminar cada iteración de desarrollo se le aplicaron las pruebas correspondientes a las HU implementadas con el objetivo de detectar los posibles errores y corregirlos en el menor tiempo posible. En la primera iteración se ejecutaron 104 Pruebas de Aceptación (PA) de las cuales 12 no fueron satisfactorias. Para la segunda iteración se realizaron 132 PA y las pruebas pendientes de la primera iteración, detectándose 3 no conformidades las cuales fueron corregidas y revisadas posteriormente con el cliente. También se realizaron pruebas por parte del Jefe de Proyecto y del Jefe de Departamento con el objetivo de verificar el correcto funcionamiento del módulo, resultando todas las pruebas satisfactorias y quedando como constancia el acta de aceptación emitida por el Jefe de Departamento la cual se puede consultar en el Anexo 5: Carta de Aceptación.

Luego de la culminación del desarrollo del módulo de administración para el servicio PostgreSQL y las pruebas realizadas al mismo, se puede asegurar que dicho módulo incrementa la facilidad de la administración de este servicio desde HMAST. Se puede afirmar lo anterior debido a que los administradores de red ya no requieren conectarse al servicio a través de la consola, ni conocer la totalidad de los parámetros que se encuentran en los ficheros de configuración. El módulo

desarrollado cuenta con una interfaz gráfica, permite instalar y desinstalar el servicio, gestionar los ficheros de configuración y el control de acceso, también posibilita garantizar la integridad de los datos, así como administrar el uso de los recursos que se le asignan al servidor, los registros del sistema y las estadísticas de ejecución. Además, realiza ajustes de consultas y revisiones periódicas a las tablas con modificaciones considerables.

Conclusiones Parciales

La ejecución de las Tareas de Ingeniería permitió la realización de las funcionalidades a implementar. Durante la etapa de implementación el uso de los estándares de codificación definidos permitió desarrollar un código reutilizable. A partir de las pruebas realizadas al *software* se logró verificar el correcto funcionamiento del mismo. Los resultados obtenidos a partir de la ejecución de las PA demuestran que el producto final cuenta con la calidad deseada y esperada por el cliente.

Conclusiones generales

Con la realización del presente trabajo se dio cumplimiento a cada uno de los objetivos trazados destacándose de manera general las conclusiones siguientes:

- Se realizó un estudio de las principales herramientas para la administración del servicio de base de datos PostgreSQL lo que contribuyó a la definición de las funcionalidades necesarias que debía incorporar el módulo a implementar.

- Se desarrolló un módulo para la administración del servicio PostgreSQL, que posibilita la administración del acceso al servidor y el uso de recursos que se le asignan. También garantiza entre otras opciones, la integridad de los datos, los ajustes de consultas y los registros del sistema. Todo esto a través de una interfaz gráfica que facilita la administración del servicio.

- Se comprobó el correcto funcionamiento del módulo de PostgreSQL desarrollado a través de la ejecución de pruebas de aceptación, determinándose que las funcionalidades implementadas se encuentran en correspondencia con las necesidades requeridas por el cliente.

Recomendaciones

La presente investigación recomienda que se desarrollen funcionalidades que permitan:

- La gestión de las base de datos.
- La gestión de los usuarios que tienen permisos asignados en las base de datos.

Referencias bibliográficas

1. BARZANALLANA RAFAEL. Servidores de bases de datos. [En línea] 2014 [Consultado el: 14 de Febrero del 2014]. Disponible en: http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/sgbd.html.

2. SCRIBID. Guía Cubana (Migración a Software Libre). [En línea] 2014. [Consultado el: 14 de Febrero del 2014]. Disponible en: http://es.scribd.com/doc/17779155/Guía-Cubana-Migracion-a-Software-Libre.

3. simays - Revisión 23091: /investigaciones/publicaciones. [En línea] 2014. [Consultado el: 14 de Febrero del 2014]. Disponible en: https://repositorio.geitel.prod.uci.cu/svn/simays/investigaciones/publicaciones/.

4. BARZANALLANA RAFAEL. Servidores de bases de datos. [En línea] 2014 [Consultado el: 14 de Febrero del 2014]. Disponible en: http://www.um.es/docencia/barzana/DIVULGACION/INFORMATICA/sgbd.html.

5. BASEDATOSOFIMATICAS – 3. Características de los Sistemas Gestores de Bases de Datos. [En línea] 2014. [Consultado el: 14 de Febrero del 2014]. Disponible en: http://basedatosofimaticas.wikispaces.com/3++Caracter%C3%ADsticas+de+los+Sistemas+Gestores+de+Bases+de+Datos.

6. CAVSI. ¿Qué es un Sistema Gestor de Bases de Datos o SGBD? [En línea] 2014. [Consultado el: 14 de Febrero del 2014]. Disponible en: http://www.cavsi.com/preguntasrespuestas/que-es-un-sistema-gestor-de-bases-de-datos-o-sgbd/.

7. COMUNIDAD TÉCNICA DE DESARROLLO DE POSTGRESQL. Acerca de PostgreSQL. [En línea] 2014. [Consultado el: 14 de Febrero del 2014]. Disponible en: http://postgresql.uci.cu/?page_id=30.

8. GEEKWARE. PostGreSQL vs. MySQL [En línea] 2014. [Consultado el: 14 de Febrero del 2014]. Disponible en: http://danielpecos.com/documents/postgresql-vs-mysql/#AEN17.

9. HTTP-PERU. PROVEEDOR DE TECNOLOGÍAS DE INFORMACIÓN. Sistema Gestor de Base de Datos PostgreSQL. [En línea] 2007. [Consultado el: 13 de Febrero del 2014]. Disponible en: http://www.http-peru.com/postgresql.php.

10. LINKED GLOBAL SERVICES. *PresentacionES_PSQL.pdf* [En línea] 2013 [Consultado el 13 de Febrero del 2014]. Disponible en: http://www.lgs.com.ve/pres/PresentacionES_PSQL.pdf.

11. REYES NIURISLEIDY, CAMBAR RAÚL. *Indicadores que determinan la seguridad de las bases de datos realizadas en PostgreSQL en cuanto a la configuración, diseño, arquitectura y codificación*. Universidad de las Ciencias Informáticas. La Habana, 2010.

12. REYES NIURISLEIDY, CAMBAR RAÚL. *Indicadores que determinan la seguridad de las bases de datos realizadas en PostgreSQL en cuanto a la configuración, diseño, arquitectura y codificación*. Universidad de las Ciencias Informáticas. La Habana, 2010.

13. POSTGRESQL-ES. Sobre postgresql. [En línea] 2013. [Consultado el 14 de Febrero del 2014]. Disponible en: http://www.postgresql.org.es/sobre_postgresql.

14. WEBMIN. Documentación Webmin [En línea] 2014. [Consultado el: 14 de Febrero del 2014]. Disponible en: http://www.webmin.com/.

15. PGADMIN. PostgreSQL administration and management tools. [En línea] 2014. [Consultado el: 14 de Febrero del 2014]. Disponible en: http://www.pgadmin.org/.

16. ORTIZ, HENRY. *Guía de Arquitectura en N Capas*. [En línea] 2010. [Consultado el: 24 de Mayo 2014]. Disponible en: http://guiaarquitecturancapas.blogspot.com/.

17. PEÑALVER ROMERO, GLADYS MARSI. *Metodología ágil para proyectos de software libre.* Universidad de las Ciencias Informáticas, La Habana, 2008.

18. DEFINICIÓN. ORG. Definición de lenguaje de programación. [En línea] 2014. [Consultado el: 14 de Febrero del 2014]. Disponible en: http://www.definicion.org/lenguaje-de-programacion.

19. ZUKOWSKI, John. *Programación Java 2 J2SE 1.4 Volumen 1*. Ediciones Anaya. 2006.

20. MAILXMAIL. Características del lenguaje java - Java. [En línea] 2004. [Consultado el: 14 de Febrero del 2014]. ISSN: 1699-4914. Disponible en: http://www.mailxmail.com/curso-java/caracteristicas-lenguaje-java-1.

21. SPRING. Spring Framework. [En línea] 2014. [Consultado el 14 de Febrero del 2014]. Disponible en: http://projects.spring.io/spring-framework/.

22. TAIN DOMÍNGUEZ, Yuliette, VALDIVIA GENÓ Román Miguel. *Augeas, propuesta tecnológica para la gestión de archivos de configuración de sistemas GNU/Linux.* 2009.

23. MASTERMAGAZINE. Definición de Herramienta. [En línea] 2007. [Consultado el 14 de Febrero del 2014]. Disponible en: http://www.mastermagazine.info/termino/5234.php.

24. ALEGSA.com.ar. Definición de IDE [En línea] 2014. [Consultado el 4 de Junio del 2014]. Disponible en: http://www.alegsa.com.ar/Dic/ide.php.

25. SOFTWARE DESIGN TOOLS FOR AGILE TEAMS, WITH UML, BPMN AND MORE. [En línea] 2014. [Consultado el 2 de Junio del 2014]. Disponible en: http://www.visual-paradigm.com/.

26. RAPIDSVN. [En línea] 2012. [Consultado el 14 de Febrero del 2014]. Disponible en: http://www.rapidsvn.org/.

27. LA CIUDAD MÁS GRANDE DE ANDALUCÍA. Herramienta Pencil. [En línea] 2011 [Consultado el: 10 de Febrero del 2014]. Disponible en: http://www.edukanda.es/mediatecaweb/data/zip/1333/page_19.htm.

28. PEÑALVER ROMERO, GLADYS MARSI. *Propuesta de un expediente, para los proyectos productivos del Polo de Software Libre, de la Facultad 10.* Universidad de las Ciencias Informáticas, 2008.

29. OLIVARES ROJAS, Juan Carlos. *Patrones de diseño.* [2003].

30. LARMAN, CRAIG. *UML y Patrones, Introducción al análisis y diseño orientado a objetos.* [1999]. ISBN 970-17-0261-1.

31. DEVELOPER NETKWORK. Revisiones de código y estándares de codificación. [En línea] 2014. [Consultado el: 9 de Mayo del 2014]. Disponible en: http://msdn.microsoft.com/es-es/library/aa291591%28v=vs.71%29.aspx.

32. R.S., PRESSMAN. *Ingeniería del software un enfoque práctico*. Quinta edición. Madrid., 2002.

33. PRUEBAS DE SOFTWARE. Gestión de la calidad y Pruebas de software. [En línea] 2005. [Consultado el: 12 de Mayo del 2014]. Disponible en: http://www.pruebasdesoftware.com/pruebadeaceptacion.htm.

Glosario de términos

Analyze: colecta estadísticas sobre los contenidos de las tablas de las base de datos y almacena los resultados en la tabla del sistema pg_statistic. Posteriormente, el planificador de consultas utiliza estas estadísticas para determinar los planes de ejecución más eficientes para consultas.

Autovacuum: encargado de revisar periódicamente la tablas con modificaciones considerables, información esta que suministra el recolector de estadísticas, que lo ayudan a llevar a cabo las tareas: vacuum y analyze.

Framework: es una estructura de soporte definida en la cual otro proyecto de *software* puede ser organizado y desarrollado. Un *framework* puede incluir soporte de programas, bibliotecas y un lenguaje interpretado.

Historias de Usuario (HU): secuencias de acciones que el sistema puede llevar a cabo interactuando con sus actores, incluyendo alternativas dentro de las secuencias.

Host: computadoras conectadas a una red, que proveen y utilizan servicios de ella.

IDE: Entorno de desarrollo integrado o en inglés *Integrated Development Environment*. Es un programa compuesto por un conjunto de herramientas para un programador desde el que se pueden editar programas, compilarlos y depurarlos.

Logs: registro oficial de eventos durante un rango de tiempo en particular.

Máscara de red: es una combinación de bits que sirve para delimitar el ámbito de una red de computadoras. Su función es indicar a los dispositivos qué parte de la dirección IP es el número de la red, incluyendo la subred, y qué parte es la correspondiente al host.

Software Privativo: el *software* no libre se refiere a cualquier programa informático en el que los usuarios tienen limitadas las posibilidades de usarlo, modificarlo o redistribuirlo, cuyo código fuente no está disponible.

RAM: se utiliza como memoria de trabajo para el sistema operativo, los programas y la mayoría del *software*. Es allí donde se cargan todas las instrucciones que ejecutan el procesador y otras unidades de cómputo.

Servidor: computadora central de un sistema de red que provee servicios y recursos

(programas, comunicaciones, archivos, entre otros) a otras computadoras (clientes) conectadas a ella.

SSH: es el nombre de un protocolo y del programa que lo implementa, y se utiliza para acceder a máquinas remotas a través de una red. Permite manejar por completo la computadora mediante un intérprete de comandos, y también puede redirigir el tráfico de X para poder ejecutar programas gráficos si se tiene un servidor X (en sistemas Unix y Windows) corriendo.

SSL: *Secure Sockets Layer*. Protocolo de Capa de Conexión Segura. Protocolos criptográficos que proporcionan comunicaciones seguras por una red.

TCP/IP: *Transfer Control Protocol / Internet Protocol.* Protocolos que se utilizan en Internet para transmitir datos. El TCP está orientado a la conexión que establece una línea de diálogo entre el emisor y el receptor antes de que se transfieran los datos.

WAL: ficheros para guardar toda la información sobre las transacciones y cambios realizados en las base de datos, así como para garantizar la integridad de los mismos. También emplea estos archivos para reparar automáticamente posibles inconsistencias en las base de datos después de una caída súbita del servidor.

Anexos

Anexo 1: Lista de Reserva del Producto (LRP).

Asigando a	Ítem *	Descripción	Estimación (Horas)	Estimado por
Prioridad		**Alta**		
Programador	1.	Instalar el servicio PostgreSQL.	4	Llilian Fernández Casasayas
Programador	2.	Desinstalar el servicio PostgreSQL.	4	Llilian Fernández Casasayas
Programador	3.	Iniciar servicio PostgreSQL.		Llilian Fernández Casasayas
Programador	4.	Reiniciar servicio PostgreSQL.	4	Llilian Fernández Casasayas
Programador	5.	Detener servicio PostgreSQL.	4	Llilian Fernández Casasayas
Programador	6.	Mostrar el estado del servicio PostgreSQL.	4	Llilian Fernández Casasayas
Programador	7.	Mostrar ruta del directorio de almacenamiento de datos.	4	Llilian Fernández Casasayas
Programador	8.	Modificar ruta del directorio de almacenamiento de datos.	4	Llilian Fernández Casasayas
Programador	9.	Mostrar ruta del archivo de configuración para la autenticación.	4	Llilian Fernández Casasayas
Programador	10.	Modificar ruta del archivo de configuración para la autenticación.	4	Llilian Fernández Casasayas
Programador	11.	Mostrar ruta del archivo de configuración del usuario mapeo.	4	Llilian Fernández Casasayas
Programador	12.	Modificar ruta del archivo de configuración del usuario mapeo.	4	Llilian Fernández Casasayas
Programador	13.	Mostrar direcciones de escucha.	4	Llilian Fernández Casasayas
Programador	14.	Modificar direcciones de escucha.	4	Llilian Fernández Casasayas
Programador	15.	Eliminar direcciones de escucha.	4	Llilian Fernández Casasayas
Programador	16.	Mostrar puerto.	4	Llilian Fernández Casasayas
Programador	17.	Modificar puerto.	4	Llilian Fernández Casasayas
Programador	18.	Mostrar máximo de conexiones permitidas al PostgreSQL.	4	Llilian Fernández Casasayas

Programador	19.	Modificar máximo de conexiones permitidas al PostgreSQL.	4	Llilian Fernández Casasayas
Programador	20.	Mostrar número de conexiones reservadas para los superusuarios.	4	Llilian Fernández Casasayas
Programador	21.	Modificar número de conexiones reservadas para los superusuarios.	4	Llilian Fernández Casasayas
Programador	22.	Mostrar tiempo de espera para completar la autenticación del cliente.	4	Llilian Fernández Casasayas
Programador	23.	Modificar tiempo de espera para completar la autenticación del cliente.	8	Llilian Fernández Casasayas
Programador	24.	Habilitar la opción de conexiones SSL.	4	Llilian Fernández Casasayas
Programador	25.	Deshabilitar la opción de conexiones SSL.	4	Llilian Fernández Casasayas
Programador	26.	Habilitar el cifrado de contraseña.	4	Llilian Fernández Casasayas
Programador	27.	Deshabilitar el cifrado de contraseña.	4	Llilian Fernández Casasayas
Programador	28.	Habilitar la opción de ponerle el nombre del usuario por cada base de datos.	4	Llilian Fernández Casasayas
Programador	29.	Deshabilitar la opción de ponerle el nombre del usuario por cada base de datos.	4	Llilian Fernández Casasayas
Programador	30.	Mostrar cantidad de memoria que el servidor de base de datos utiliza para buffers.	4	Llilian Fernández Casasayas
Programador	31.	Modificar cantidad de memoria que el servidor de base de datos utiliza para buffers.	4	Llilian Fernández Casasayas
Programador	32.	Mostrar número de búferes temporales utilizados por cada sesión de base de datos.	4	Llilian Fernández Casasayas
Programador	33.	Modificar número de búferes temporales utilizados por cada sesión de base de datos.	4	Llilian Fernández Casasayas
Programador	34.	Mostrar número de transacciones que pueden estar en el estado preparado.	4	Llilian Fernández Casasayas
Programador	35.	Modificar número de transacciones que pueden estar en el estado preparado.	4	Llilian Fernández Casasayas
Programador	36.	Mostrar la cantidad de memoria que puede utilizar el servicio antes	4	Llilian Fernández Casasayas

		de crear archivos temporales para el procesamiento de los resultados intermedios.		
Programador	37.	Modificar la cantidad de memoria que puede utilizar el servicio antes de crear archivos temporales para el procesamiento de los resultados intermedios.	4	Llilian Fernández Casasayas
Programador	38.	Mostrar la cantidad de memoria que se utilizará por las operaciones de mantenimiento.	4	Llilian Fernández Casasayas
Programador	39.	Modificar la cantidad de memoria que se utilizará por las operaciones de mantenimiento.	4	Llilian Fernández Casasayas
Programador	40.	Mostrar la cantidad de tiempo en que el proceso debe hacer una pausa.	4	Llilian Fernández Casasayas
Programador	41.	Modificar la cantidad de tiempo en que el proceso debe hacer una pausa.	4	Llilian Fernández Casasayas
Programador	42.	Mostrar el límite en que el proceso debe hacer una pausa.	4	Llilian Fernández Casasayas
Programador	43.	Modificar el límite en que el proceso debe hacer una pausa.	4	Llilian Fernández Casasayas
Programador	44.	Mostrar tiempo para activar la escritura de segundo plano.	4	Llilian Fernández Casasayas
Programador	45.	Modificar tiempo para activar la escritura de segundo plano.	4	Llilian Fernández Casasayas
Programador	46.	Mostrar la cantidad de buffers para la escritura en segundo plano.	4	Llilian Fernández Casasayas
Programador	47.	Modificar la cantidad de buffers para la escritura en segundo plano.	4	Llilian Fernández Casasayas
Programador	48.	Listar reglas de control de acceso.	4	Llilian Fernández Casasayas
Programador	49.	Crear reglas de control de acceso al servidor PostgreSQL.	4	Llilian Fernández Casasayas
Programador	50.	Modificar reglas de control de acceso al servidor PostgreSQL.	4	Llilian Fernández Casasayas
Programador	51.	Eliminar reglas de control de acceso al servidor PostgreSQL.	4	Llilian Fernández Casasayas
		Media		
Programador	52.	Mostrar métodos para almacenar la cantidad de información que se escribe en el WAL.	2	Llilian Fernández Casasayas
Programador	53.	Modificar métodos para almacenar la	2	Llilian Fernández

		cantidad de información que se escribe en el WAL.		Casasayas
Programador	54.	Habilitar que se escriban físicamente las actualizaciones en el disco.	2	Llilian Fernández Casasayas
Programador	55.	Deshabilitar que se escriban físicamente las actualizaciones en el disco.	2	Llilian Fernández Casasayas
Programador	56.	Habilitar la opción de que las transacciones se comprometen a esperar a los registros WAL que se escriben en el disco.	2	Llilian Fernández Casasayas
Programador	57.	Deshabilitar la opción de que las transacciones se comprometen a esperar a los registros WAL que se escriben en el disco.	2	Llilian Fernández Casasayas
Programador	58.	Mostrar método utilizado para forzar cambios WAL al disco.	2	Llilian Fernández Casasayas
Programador	59.	Modificar método utilizado para forzar cambios WAL al disco.	2	Llilian Fernández Casasayas
Programador	60.	Mostrar la cantidad de memoria compartida utilizada como buffer para los datos del WAL.	2	Llilian Fernández Casasayas
Programador	61.	Modificar la cantidad de memoria compartida utilizada como buffer para los datos del WAL.	2	Llilian Fernández Casasayas
Programador	62.	Mostrar la cantidad de segmentos que se chequearán.	2	Llilian Fernández Casasayas
Programador	63.	Modificar la cantidad de segmentos que se chequearán.	2	Llilian Fernández Casasayas
Programador	64.	Mostrar el tiempo en que se chequearán los segmentos.	2	Llilian Fernández Casasayas
Programador	65.	Modificar el tiempo en que se chequearán los segmentos.	2	Llilian Fernández Casasayas
Programador	66.	Mostrar el destino de finalización del punto de chequeo.	2	Llilian Fernández Casasayas
Programador	67.	Modificar el destino de finalización del punto de chequeo.	2	Llilian Fernández Casasayas
Programador	68.	Mostrar el costo de leer una única página de base de datos en disco de forma secuencial.	2	Llilian Fernández Casasayas
Programador	69.	Modificar el costo de leer una única página de base de datos en disco de	2	Llilian Fernández Casasayas

		forma secuencial.		
Programador	70.	Mostrar el valor por el cual el planificador de consulta considera los accesos no secuenciales a disco.	2	Llilian Fernández Casasayas
Programador	71.	Modificar el valor por el cual el planificador de consulta considera los accesos no secuenciales a disco.	2	Llilian Fernández Casasayas
Programador	72.	Mostrar el tamaño efectivo de la caché de disco que está disponible para una sola consulta.	2	Llilian Fernández Casasayas
Programador	73.	Modificar el tamaño efectivo de la caché de disco que está disponible para una sola consulta.	2	Llilian Fernández Casasayas
Programador	74.	Habilitar la opción de optimización de la consulta genética.	2	Llilian Fernández Casasayas
Programador	75.	Deshabilitar la opción de optimización de la consulta genética.	2	Llilian Fernández Casasayas
Programador	76.	Mostrar el valor para que el optimizador de consulta planifique las consultas.	2	Llilian Fernández Casasayas
Programador	77.	Modificar el valor para que el optimizador de consulta planifique las consultas.	2	Llilian Fernández Casasayas
Programador	78.	Mostrar el valor que el planeador de consultas utilizará en las restricciones de tablas.	2	Llilian Fernández Casasayas
Programador	79.	Modificar el valor que el planeador de consultas utilizará en las restricciones de tablas.	2	Llilian Fernández Casasayas
Programador	80.	Mostrar los métodos que indican la salida a la que irán dirigidos los mensajes de registro.	2	Llilian Fernández Casasayas
Programador	81.	Modificar los métodos que indican la salida a la que irán dirigidos los mensajes de registro.	2	Llilian Fernández Casasayas
Programador	82.	Habilitar la opción del colector de registro.	2	Llilian Fernández Casasayas
Programador	83.	Deshabilitar la opción del colector de registro.	2	Llilian Fernández Casasayas
Programador	84.	Mostrar el directorio en el que se crearán los archivos de registro.	2	Llilian Fernández Casasayas

Programador	85.	Modificar el directorio en el que se crearán los archivos de registro.	2	Llilian Fernández Casasayas
Programador	86.	Mostrar formato de los archivos de registros creados.	2	Llilian Fernández Casasayas
Programador	87.	Modificar formato de los archivos de registros creados.	2	Llilian Fernández Casasayas
Programador	88.	Mostrar el tiempo para actualizar los registros.	2	Llilian Fernández Casasayas
Programador	89.	Modificar el tiempo para actualizar los registros.	2	Llilian Fernández Casasayas
Programador	90.	Mostrar el tamaño de un archivo de registro.	2	Llilian Fernández Casasayas
Programador	91.	Modificar el tamaño de un archivo de registro.	2	Llilian Fernández Casasayas
Programador	92.	Habilitar la opción de sobrescribir un archivo de registro.	2	Llilian Fernández Casasayas
Programador	93.	Deshabilitar la opción de sobrescribir un archivo de registro.	2	Llilian Fernández Casasayas
Programador	94.	Mostrar niveles de mensajes que se envían al cliente.	2	Llilian Fernández Casasayas
Programador	95.	Modificar niveles de mensajes que se envían al cliente.	2	Llilian Fernández Casasayas
Programador	96.	Mostrar niveles de mensajes que se escriben en el servidor.	2	Llilian Fernández Casasayas
Programador	97.	Modificar niveles de mensajes que se escriben en el servidor.	2	Llilian Fernández Casasayas
Programador	98.	Mostrar los errores que se registren en el servidor.	2	Llilian Fernández Casasayas
Programador	99.	Modificar los errores que se registren en el servidor.	2	Llilian Fernández Casasayas
Programador	100.	Mostrar la opción de registrar las consultas que hayan durado más tiempo.	2	Llilian Fernández Casasayas
Programador	101.	Modificar la opción de registrar las consultas que hayan durado más tiempo.	2	Llilian Fernández Casasayas
Programador	102.	Mostrar el nivel de detalle que se incluye en cada mensaje añadido al servidor.	2	Llilian Fernández Casasayas
Programador	103.	Modificar el nivel de detalle que se incluye en cada mensaje añadido al servidor.	2	Llilian Fernández Casasayas
Programador	104.	Mostrar qué sentencias SQL se registrarán.	2	Llilian Fernández Casasayas

Programador	105.	Modificar qué sentencias SQL se registrarán.	2	Llilian Fernández Casasayas
Programador	106.	Habilitar la opción de recopilar la información sobre el comando que se está ejecutando.	2	Llilian Fernández Casasayas
Programador	107.	Deshabilitar la opción de recopilar la información sobre el comando que se está ejecutando.	2	Llilian Fernández Casasayas
Programador	108.	Mostrar el número de bytes reservados para mostrar la consulta que se está ejecutando.	2	Llilian Fernández Casasayas
Programador	109.	Modificar el número de bytes reservados para mostrar la consulta que se está ejecutando.	2	Llilian Fernández Casasayas
Programador	110.	Habilitar la recopilación de las estadísticas sobre la actividad de las base de datos.	2	Llilian Fernández Casasayas
Programador	111.	Deshabilitar la recopilación de las estadísticas sobre la actividad de las base de datos.	2	Llilian Fernández Casasayas
Programador	112.	Mostrar seguimiento de las estadísticas.	2	Llilian Fernández Casasayas
Programador	113.	Modificar seguimiento de las estadísticas.	2	Llilian Fernández Casasayas
Programador	114.	Mostrar directorio para guardar datos de las estadísticas.	2	Llilian Fernández Casasayas
Programador	115.	Modificar directorio para guardar datos de las estadísticas.	2	Llilian Fernández Casasayas
Programador	116.	Habilitar la opción de ejecutar el demonio lanzador autovacuum.	2	Llilian Fernández Casasayas
Programador	117.	Deshabilitar la opción de ejecutar el demonio lanzador autovacuum.	2	Llilian Fernández Casasayas
Programador	118.	Mostrar el número máximo de procesos autovacuum.	2	Llilian Fernández Casasayas
Programador	119.	Modificar el número máximo de procesos autovacuum.	2	Llilian Fernández Casasayas
Programador	120.	Mostrar el tiempo que transcurre entre cada autovacuum.	2	Llilian Fernández Casasayas
Programador	121.	Modificar el tiempo que transcurre entre cada autovacuum.	2	Llilian Fernández Casasayas
Programador	122.	Mostrar la cantidad de filas para ejecutar VACUUM.	2	Llilian Fernández Casasayas
Programador	123.	Modificar la cantidad de filas para	2	Llilian Fernández

		ejecutar VACUUM.		Casasayas
Programador	124.	Mostrar la cantidad de filas para ejecutar ANALYZE.	2	Llilian Fernández Casasayas
Programador	125.	Modificar la cantidad de filas para ejecutar ANALYZE.	2	Llilian Fernández Casasayas
Programador	126.	Mostrar el por ciento del tamaño de tabla que lanzará el VACUUM.	2	Llilian Fernández Casasayas
Programador	127.	Modificar el por ciento del tamaño de tabla que lanzará el VACUUM.	2	Llilian Fernández Casasayas
Programador	128.	Mostrar la cantidad de transacciones para que se ejecute un vacuum aunque se haya desactivado el autovacuum.	2	Llilian Fernández Casasayas
Programador	129.	Modificar la cantidad de transacciones para que se ejecute un vacuum aunque se haya desactivado el autovacuum.	2	Llilian Fernández Casasayas
Programador	130.	Mostrar el por ciento del tamaño de tabla que lanzará el ANALYZE.	2	Llilian Fernández Casasayas
Programador	131.	Modificar el por ciento del tamaño de tabla que lanzará el ANALYZE.	2	Llilian Fernández Casasayas
Programador	132.	Mostrar el tiempo que el proceso de VACUUM automático debe dormir entre ejecuciones.	2	Llilian Fernández Casasayas
Programador	133.	Modificar el tiempo que el proceso de VACUUM automático debe dormir entre ejecuciones.	2	Llilian Fernández Casasayas
Programador	134.	Mostrar el número de operaciones que pueden llevar a cabo los procesos del autovacuum.	2	Llilian Fernández Casasayas
Programador	135.	Modificar el número de operaciones que pueden llevar a cabo los procesos del autovacuum.	2	Llilian Fernández Casasayas
RNF (Requisitos No Funcionales)				
Programador	1	Usar como lenguaje de programación Java y el framework Spring.		
Programador	2	Disponer para el SO GNU/Linux de los siguientes paquetes: • java- augeas • nmap • openssh • gksu2		
Programador	3	Utilizar Netbeans como entorno de desarrollo integrado.		

Programador	4	Utilizar HTML5 para la interfaz de la aplicación.		
Programador	5	Las conexiones con los servidores deben ser mediante SSH2 y las contraseñas no serán guardadas.		
Programador	6	La aplicación se ejecutará en sistemas operativos libres.		

Tabla 5: Lista de Reserva del Producto (LRP).

Anexo 2: Historias de Usuario.

Historia de Usuario	
Número:HMAST_**PostgreSQL**_1	**Nombre Historia de Usuario:** Instalar y desinstalar el servicio PostgreSQL.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Llilian Fernández Casasayas	**Iteración Asignada:** 1
Prioridad en Negocio: Alta	**Puntos Estimados:** 2 semanas
Riesgo en Desarrollo: Muy Alto	**Puntos Reales:** 2 semanas
Descripción: EL *software* permite instalar y desinstalar el servicio PostgreSQL.	
Observaciones: La aplicación le permitirá al usuario instalar el servicio PostgreSQL en caso de que no se encuentre instalado. A partir de ese instante una imagen animada indica el progreso de la instalación. Para la instalación del servicio se utiliza el comando **apt-get install postgresql-server,** el cual instalará la versión que se encuentre en el repositorio. Para la desinstalación del servicio se utilizará el comando **apt-get purge postgresql-server**. Una vez realizadas todas las configuraciones la aplicación brinda la posibilidad de guardar los cambios o cancelarlos.	
Prototipo de interfaz:	

Figura 10: Prototipo_Instalar

Historia de Usuario	
Número:HMAST_**PostgreSQL**_2	**Nombre Historia de Usuario:** Administrar el servicio PostgreSQL.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Llilian Fernández Casasayas	**Iteración Asignada:** 1
Prioridad en Negocio: Alto	**Puntos Estimados:** 2 semanas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 2 semanas
Descripción: El sistema brinda la posibilidad de Iniciar, Detener, Reiniciar y Recargar el servicio.	

Observaciones:

Se brinda la opción de realizar las acciones de Iniciar, Reiniciar y Detener el servicio cada vez que el usuario lo desee, ya que la interfaz cuenta con tres botones en correspondencia con cada una de las operaciones y son ubicados de forma permanente, independientemente de la configuración que esté realizando. En el caso de Recargar además de poder hacerse en el momento deseado, debe realizarse de forma automática después que se acepta cada configuración.

- **Iniciar el servicio:** El *software* permite realizar el proceso de iniciar el servicio cada vez que el usuario lo desee. Al seleccionar dicha opción el sistema ejecuta el comando service postgresql start

- **Reiniciar el servicio:** El *software* brinda la posibilidad de reiniciar el servicio cada vez que el usuario lo desee. Al seleccionar dicha opción el sistema ejecuta el comando service postgresql restart.
- **Detener el servicio:** El *software* permite realizar el proceso de detener el servicio cada vez que el usuario lo desee. Al seleccionar dicha opción el sistema ejecuta el comando service postgresql stop.
- **Mostrar el estado del servicio:** El *software* permite realizar el proceso de mostrar el estado del servicio cada vez que el usuario lo desee. Al seleccionar dicha opción el sistema ejecuta el comando netstat -tlnup | grep postgres

Prototipo de interfaz:

Figura 11: Prototipo_Gestionar Servicio

Historia de Usuario	
Número:HMAST_**PostgreSQL**_3	**Nombre Historia de Usuario:** Configuración de la localización de los directorios de instalación del servicio PostgreSQL.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Llilian Fernández Casasayas	**Iteración Asignada:** 1
Prioridad en Negocio: Alto	**Puntos Estimados:** 2 semanas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 2 semanas
Descripción: El *software* permitirá mostrar y modificar la locación de los directorios de instalación del servicio PostgreSQL.	
Observaciones: Se brinda la opción de mostrar las locaciones de los archivos de instalación del servicio, además de poder modificarlos, ya que la interfaz cuenta con los botones para realizar estas acciones. Todos los cambios que se apliquen en esta interfaz se realizarán mediante la edición del fichero **/etc/postgresql/9.1/main/postgresql.conf.**	

Mostrar ruta del directorio de almacenamiento de datos: Esta opción permitirá mostrar la ruta que tiene definida por defecto el directorio que fue seleccionado en el proceso de instalación donde se almacenarán todos los datos del servicio. La ruta predeterminada es /var/lib/postgresql/9.1/main.

Modificar ruta del directorio de almacenamiento de datos. Una vez mostrada la ruta el usuario puede modificarla, teniendo en cuenta que la nueva ruta tiene que ser válida, en caso contrario se muestra un mensaje de error. Esto se podrá modificar a través del parámetro **data_directory**.

Mostrar ruta del archivo de configuración para la autenticación: Se muestra al usuario la ruta del archivo de configuración para la autenticación basada en host, habitualmente llamado pg_hba.conf. El parámetro que permitirá mostrar esta directiva es **hba_file.**

Modificar ruta del archivo de configuración para la autenticación: Una vez mostrada la ruta del archivo de configuración para la autenticación, el usuario puede modificarla. La ruta predeterminada es /etc/postgresql/9.1/main/pg_hba.conf y se debe tener en cuenta que la nueva ruta sea válida, en caso contrario se muestra un mensaje de error.

Mostrar ruta del archivo de configuración del usuario mapeo: Se muestra al usuario la ruta del archivo de configuración del usuario mapeo habitualmente llamada pg_ident.conf. El parámetro que permitirá mostrar esta directiva es **ident_file.**

Modificar ruta del archivo de configuración del usuario mapeo: Una vez mostrada la ruta del archivo de configuración del usuario mapeo, el usuario puede modificarla. La localización predeterminada es /etc/postgresql/9.1/main/pg_ident.conf y se debe tener en cuenta que la nueva ruta sea válida, en caso contrario se muestra un mensaje de error.

Una vez realizadas todas las configuraciones la aplicación brinda la posibilidad de guardar los cambios o cancelarlos.

Prototipo de interfaz:

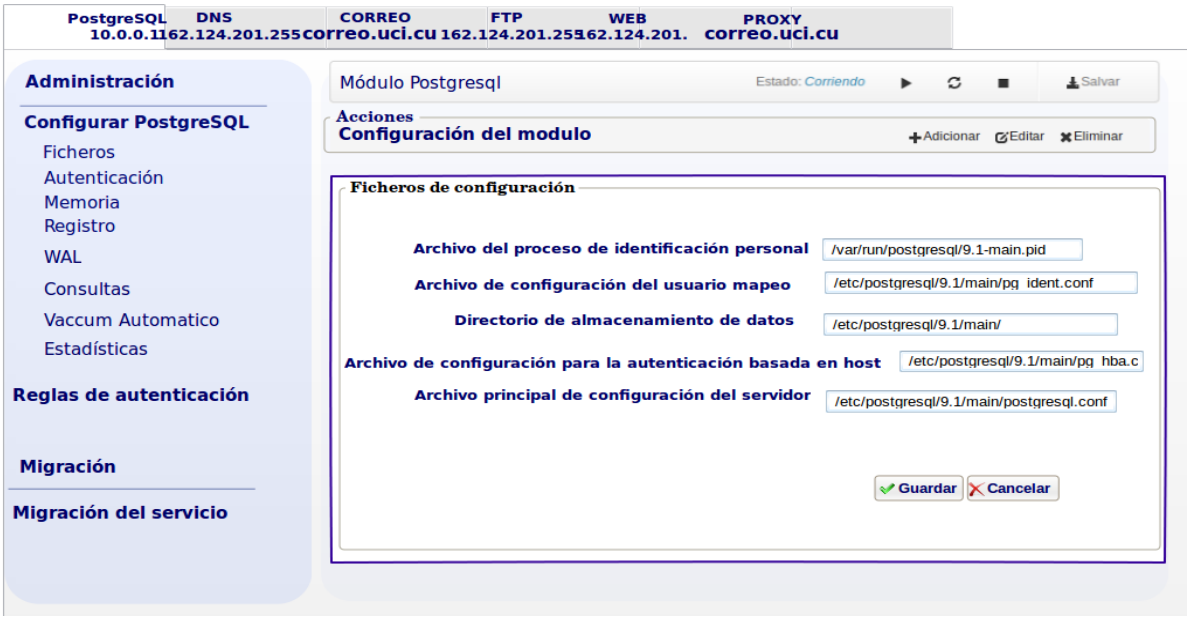


Figura 12: Prototipo_Ficheros de Configuración

Historia de Usuario	
Número: HMAST_**PostgreSQL**_4	**Nombre Historia de Usuario:** Configuración de la conexión y autenticación.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Llilian Fernández Casasayas	**Iteración Asignada:** 1
Prioridad en Negocio: Alta	**Puntos Estimados:** 2 semanas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 2 semanas
Descripción: El módulo mostrará los parámetros de configuración de la conexión al servidor. También permitirá adicionar, modificar y eliminar los mismos.	
Observaciones: Todos los cambios que se apliquen en esta interfaz se realizarán mediante la edición del fichero /etc/postgresql9.1/main/postgresql.conf. Estos no serán guardados hasta que se seleccione el botón Aceptar. Cuando se realiza esta acción, se recargan las configuraciones del servicio. **Mostrar direcciones de escucha:** La aplicación permitirá que se establezcan las direcciones desde las que se escucharán conexiones a PostgreSQL. Por defecto es *, lo que implica que permitirá peticiones desde cualquier IP, aunque se pueden indicar direcciones específicas y deben ser un IPv4 (Ej: 10.8.32.15). Estas se mostrarán a través del parámetro **ListenAddress.**	

Modificar direcciones de escucha: La aplicación permitirá modificar la dirección desde la que se escucharán conexiones a PostgreSQL. Si el usuario desea conectarse desde la propia máquina se dejará **ListenAddress = 'localhost'**, como está predefinido. En caso de querer un IP específico se escribirá **ListenAddress =10.8.121.63** y en caso de que desee conectar todos los IP y subred se escribirá **ListenAddress = ' * '.** Si se necesita adicionar más de un IP o una subred se adicionarán al ListenAddress separados por coma.

Eliminar direcciones de escucha: La aplicación permitirá que se eliminen las direcciones por las cuales el servidor PostgreSQL escuchará las peticiones de conexión. Se debe de verificar que al menos exista una ListenAddress.

Mostrar puerto: La aplicación mostrará el puerto por el cual el servidor PostgreSQL escuchará las peticiones de conexión. Por defecto está establecido el puerto 5432, este estará definido por el parámetro **port**.

Modificar puerto: El usuario podrá modificar a través de un campo de texto, el puerto por el cual el servidor PostgreSQL escuchará las peticiones de conexión, y verificará que el puerto no esté en uso.

Mostrar máximo de conexiones permitidas al PostgreSQL: Se muestra al usuario el máximo de conexiones al servidor PostgreSQL, el valor por defecto es 100. El máximo de conexiones se mostrará a través del parámetro **max_connections.**

Modificar máximo de conexiones permitidas al PostgreSQL: Una vez mostrada el máximo de conexiones al servidor PostgreSQL, se le brinda la opción al usuario de modificar ese parámetro a través de un campo de texto, y el valor tendrá que ser un número entero de 0 a 100 en caso de que no sea válido, se muestra un mensaje de error.

Mostrar número de conexiones reservadas para los superusuarios: Se muestra al usuario el número de conexiones reservadas para los superusuarios al servidor PostgreSQL. Esto se hará a través del parámetro **superuser_reserved_connections**.

Modificar número de conexiones reservadas para los superusuarios: Una vez mostrado el número de conexiones reservadas para los superusuarios al servidor PostgreSQL, se le brinda la posibilidad al usuario de modificar este parámetro a través de un campo de texto, el valor será un

número entero y debe ser menor que el máximo de conexiones y que 10. Tomará el valor predeterminado de 3. En caso de que no sea válido y no se encuentre en ese rango se mostrará un mensaje de error.

Mostrar tiempo de espera para completar la autenticación del cliente: La aplicación mostrará al usuario el tiempo de espera para completar la autenticación del cliente. El parámetro encargado de mostrar esto es **authentication_timeout.**

Modificar tiempo de espera para completar la autenticación del cliente: Una vez mostrado el tiempo de espera para la autenticación, se le brinda la posibilidad al usuario de modificar este parámetro. Puede ser modificado en un rango desde 1 hasta 600 segundos, pero un valor adecuado podría ser entre 15 a 30 segundos en caso de que no se encuentre en ese rango, se mostrará un mensaje de error.

Habilitar la opción de conexiones SSL: El *software* permite habilitar las conexiones seguras al servidor, a través de un checkbox. Su valor por defecto es true y esto se realizará por el parámetro **ssl.**

Deshabilitar la opción de conexiones SSL: El *software* no permitirá conexiones seguras al servidor por lo cual el valor será *false*.

Habilitar cifrado de contraseña: La aplicación le brinda al usuario la posibilidad de especificar si la contraseña se debe cifrar. Esta opción será mostrada a través de un *chekbox*. El valor del parámetro **password_encryption,** toma valor *on*, que es el predefinido**.**

Deshabilitar cifrado de contraseña: Se brinda la posibilidad de que las contraseñas no sean cifradas, lo cual hace que el parámetro tome un valor *off*.

Habilitar la opción de ponerle el nombre del usuario por cada base de datos: La aplicación le brinda al usuario la posibilidad de ponerle a cada base de datos el nombre de los usuarios, a través de un *checkbox* y toma el valor *on.* El parámetro que permitirá esto será **db_user_namespace.**

Deshabilitar la opción de ponerle el nombre del usuario por cada base de datos: Se brinda la posibilidad de que no se le ponga a cada base de datos el nombre de los usuarios, lo cual hace que el parámetro tome un valor *off*, que es el predeterminado.

Una vez realizadas todas las configuraciones la aplicación brinda la posibilidad de guardar los cambios o cancelarlos.

Prototipo de interfaz:

Figura 13: Prototipo_Direcciones de escucha

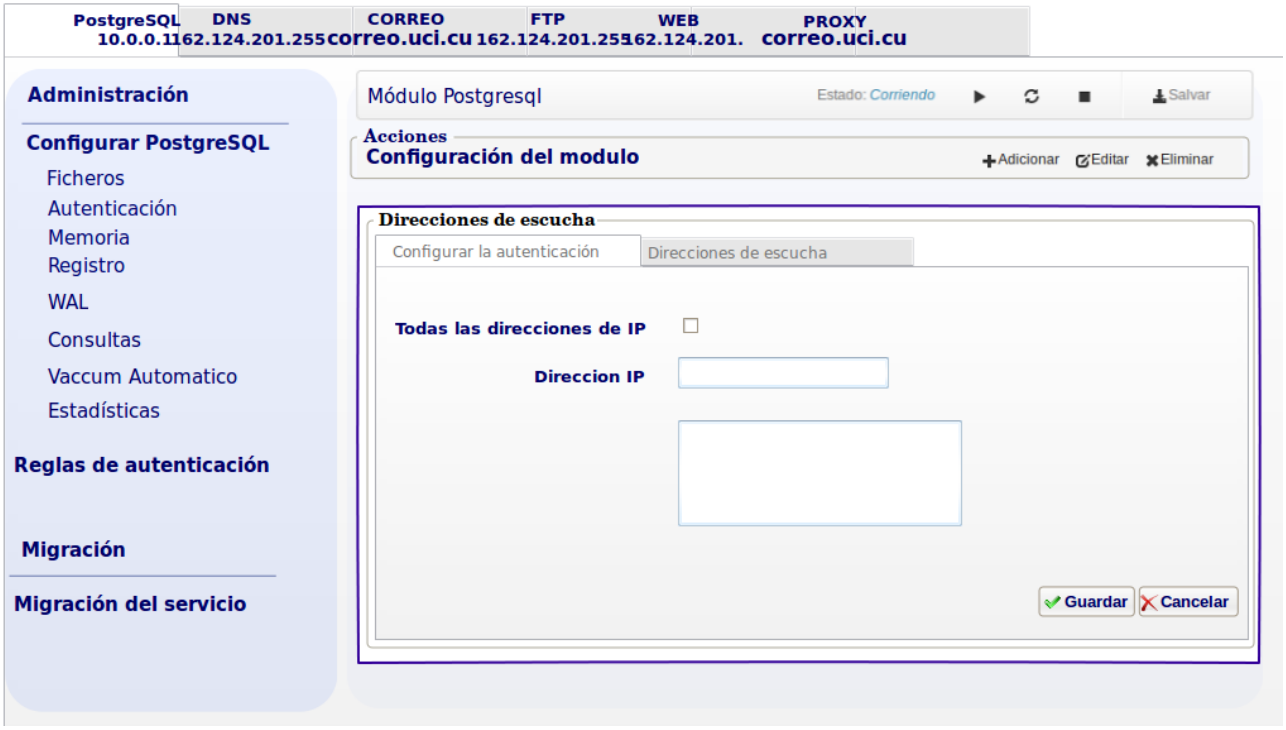


Figura 14: Prototipo_Conexión y autenticación

Historia de Usuario	
Número: HMAST_**PostgreSQL**_5	**Nombre Historia de Usuario:** Configuración de uso de recursos
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Llilian Fernández Casasayas	**Iteración Asignada:** 1
Prioridad en Negocio: Alto	**Puntos Estimados:** 2 semanas
Riesgo en Desarrollo: Ato	**Puntos Reales:** 2 semanas
Descripción: La aplicación mostrará los parámetros de configuración para el uso de recursos y los podrá modificar.	
Observaciones: Todos los cambios que se apliquen en esta interfaz se realizarán mediante la edición del fichero **/etc/postgresql/9.1/main/postgresql.conf.** Estos no serán guardados hasta que se seleccione el botón Aceptar. Cuando se realiza esta acción, se recargan las configuraciones del servicio **Mostrar la cantidad de memoria que el servidor de base de datos utiliza para buffers**: La aplicación mostrará el tamaño del buffer de memoria. El parámetro encargado de mostrar esto es **shared_buffers.** **Modificar la cantidad de memoria que el servidor de base de datos utiliza para buffers:** Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto y el valor será un número entero, luego contendrá un campo de selección con valores de medidas Gigabyte, Kilobyte, Megabyte, el predeterminado es 32 Mb, El parámetro puede tomar valores desde 128 kb hasta el 25 % de la memoria RAM, en caso de que no se encuentre en dicho rango o no sea válido se mostrará un mensaje de error. **Mostrar el número de búferes temporales utilizados por cada sesión de base de datos:** La aplicación permitirá mostrar la cantidad de memoria utilizada por cada sesión de base de datos para acceder a tablas temporales. El parámetro encargado de mostrar esto es **temp_buffers**. **Modificar el número de búferes temporales utilizados por cada sesión de base de datos:** Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo a	

través de un campo de texto, el valor será un número entero. Luego contendrá un campo de selección con valores de medidas Gigabyte, Kilobyte, Megabyte, el predeterminado es ocho megabytes (8 Mb). Este parámetro puede tomar valores desde 800 kb hasta el 10240 Kb en caso de que no se encuentre en dicho rango y no sea válido se mostrará un mensaje de error.

Mostrar el número de transacciones que pueden estar en el estado preparado: La aplicación permitirá mostrar el número máximo de transacciones preparadas. Esto se realizará a través del parámetro **max_prepared_transactions**.

Modificar el número de transacciones que pueden estar en el estado preparado: Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto, el valor tendrá que ser un número entero. Al establecer este parámetro a 0 (que es el valor predeterminado) desactiva la función de transacciones preparadas. El parámetro puede tomar valores de 0 hasta 700. En caso de que no se encuentre en dicho rango se mostrará un mensaje de error.

Mostrar la cantidad de memoria que puede utilizar el servicio antes de crear archivos temporales para el procesamiento de los resultados intermedios: La aplicación permitirá mostrar la cantidad de memoria que puede utilizar PostgreSQL antes de crear archivos temporales para el procesamiento de los resultados intermedios. Esto se hará a través del parámetro **work_mem.**

Modificar la cantidad de memoria que puede utilizar el servicio antes de crear archivos temporales para el procesamiento de los resultados intermedios: El usuario podrá modificar la cantidad de memoria que puede utilizar PostgreSQL antes de crear archivos temporales para el procesamiento de los resultados intermedios, a través de un campo de texto y el valor tendrá que ser un número entero. Luego contendrá un campo de selección con valores de medidas Gigabyte, Kilobyte, Megabyte, en caso de que no sea válido, se mostrará un mensaje de error. El valor predeterminado es un megabyte (1 MB). El valor tiene un rango desde 64 kb hasta 1572864 kb, en caso de que el valor no se encuentre en ese rango, se mostrará un mensaje de error.

Mostrar la cantidad de memoria que se utilizará por las operaciones de mantenimiento: La aplicación permitirá mostrar la cantidad máxima de memoria que se utilizará por las operaciones de mantenimiento. Esto se podrá realizar a través del parámetro **maintenance_work_mem.**

Modificar la cantidad de memoria que se utilizará por las operaciones de mantenimiento: El usuario podrá modificar la cantidad máxima de memoria que se utilizará por las operaciones de mantenimiento a través de un campo de texto y el valor tendrá que ser un número entero. Luego contendrá un campo de selección con valores de medidas Gigabyte, Kilobyte, Megabyte, en caso de que no sea válido, se mostrará un mensaje de error. El valor predeterminado es 16 megabytes (16MB). Este parámetro puede tomar valores desde 1 MB hasta 1536 MB y si el valor no está en ese rango, se mostrará un mensaje de error.

Mostrar la cantidad de tiempo en que el proceso debe hacer una pausa: La aplicación permitirá mostrar la cantidad de tiempo en que el proceso debe hacer una pausa. Esto se mostrará por el parámetro **vacuum_cost_delay.**

Modificar la cantidad de tiempo en que el proceso debe hacer una pausa: El usuario podrá modificar la cantidad de tiempo que el proceso va a dormir a través de un campo de texto y el valor tendrá que ser un número entero, en caso de que no sea válido, se mostrará un mensaje de error. El valor predeterminado es de 0 milisegundos, y los valores deben ser múltiplos de 10. El valor 0 desactiva la función de ajuste de la carga de vacío sin pausa. El parámetro puede tomar valores desde 0 hasta 1000 milisegundos y si el valor no está en ese rango, se mostrará un mensaje de error.

Mostrar el límite en que el proceso debe hacer una pausa: La aplicación mostrará el límite para detener el proceso de vacío. Esto se podrá realizar a través del parámetro **vacuum_cost_limit.**

Modificar el límite en que el proceso debe hacer una pausa : El usuario podrá modificar el límite para detener el proceso de vacío a través de un campo de texto y el valor será un número entero, en caso de que no sea válido, se mostrará un mensaje de error. El valor por defecto es 200 milisegundos**.** Este parámetro puede tomar valores desde 1 a 10000 milisegundos y si el valor no está en ese rango, se mostrará un mensaje de error.

Mostrar tiempo para activar la escritura en segundo plano: La aplicación mostrará el tiempo en el que se lanzará el proceso que se encarga de actualizar los cambios llevados a cabo. Esto se podrá realizar a través del parámetro **bgwriter_delay.**

Modificar tiempo para activar la escritura en segundo plano: El usuario podrá modificar el tiempo en el que se lanzará el proceso que se encarga de actualizar los cambios llevados a cabo, a

través de un campo de texto y el valor tendrá que ser un número entero. El valor por defecto es 200 milisegundos, en caso de que no sea válido, se mostrará un mensaje de error. Este parámetro puede tomar valores desde 10 a 10000 milisegundos y si el valor no está en ese rango se mostrará un mensaje de error.

Mostrar la cantidad de buffers para la escritura en segundo plano: La aplicación mostrará la cantidad de buffers que serán actualizados cuando se lance el proceso que se encarga de actualizar los cambios llevados a cabo. Esto se podrá realizar a través del parámetro **bgwriter_lru_maxpages.**

Modificar la cantidad de buffers para la escritura en segundo plano: El usuario podrá modificar la cantidad de buffers que serán actualizados cuando se lance el proceso que se encarga de actualizar los cambios llevados a cabo, a través de un campo de texto y el valor tendrá que ser un número entero, en caso de que no sea válido se mostrará un mensaje de error. El valor por defecto es de 100 y poniendo 0 deshabilitaríamos el proceso que se encarga de actualizar los cambios llevados a cabo. El parámetro puede tomar valores desde 0 a 1000 y si el valor no está en ese rango, se mostrará un mensaje de error.

Una vez realizadas todas las configuraciones la aplicación brinda la posibilidad de guardar los cambios.

Prototipo de interfaz:

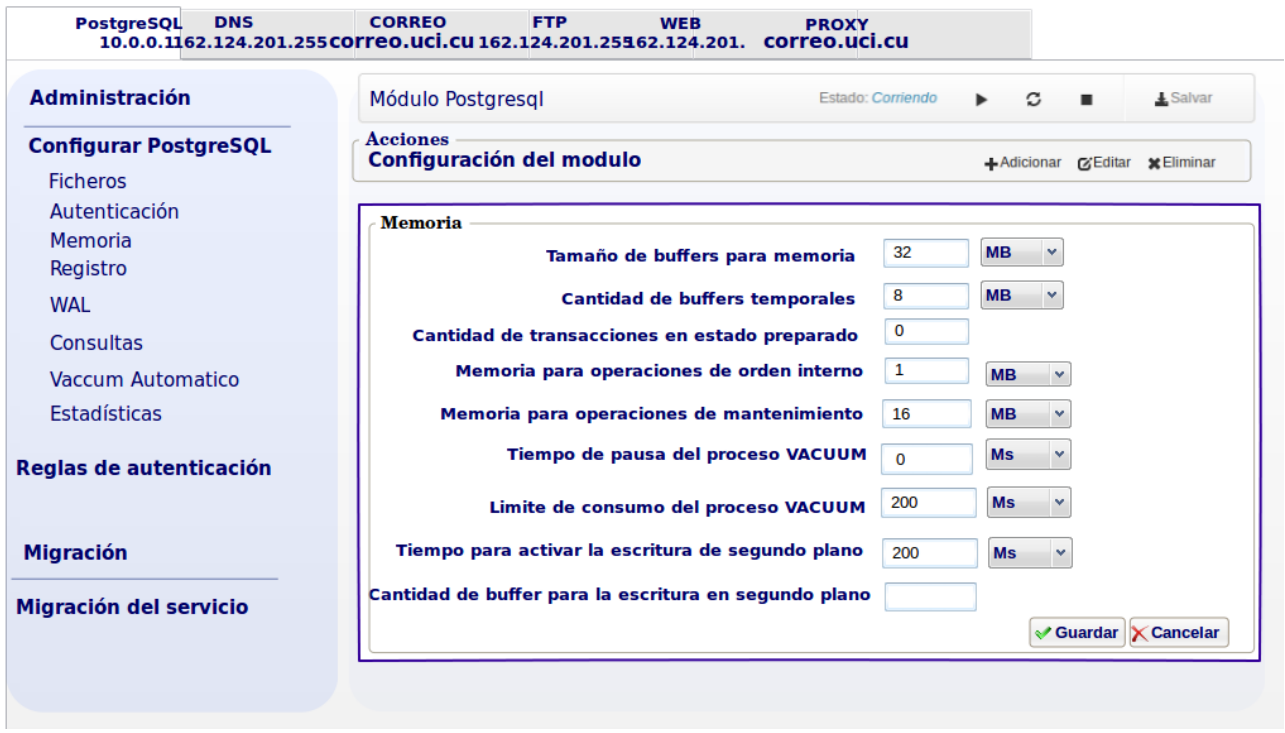


Figura 15: Prototipo_Memoria

Historia de Usuario	
Número:HMAST_**PostgreSQL**_5	**Nombre Historia de Usuario:** Gestionar acceso al servidor.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Llilian Fernández Casasayas.	**Iteración Asignada:** 1
Prioridad en Negocio: Alto	**Puntos Estimados:** 2 semanas
Riesgo en Desarrollo: Alto	**Puntos Reales:** 2 semanas
Descripción: La aplicación permitirá configurar los métodos de acceso al servidor, así como modificarlos	
Observaciones: Todos los cambios que se apliquen en esta interfaz se realizarán mediante la edición del fichero /etc/PostgreSQLql/9.1/main/pg_hba.conf. Estos no serán guardados hasta que se seleccione el botón Aceptar. Cuando se realiza esta acción, se recargan las configuraciones del servicio. **Listar reglas de control de acceso**: La aplicación permitirá listar las reglas de control de acceso, a través de una tabla, donde se mostrarán todos los parámetros que contienen una regla de acceso. **Crear reglas de control de acceso al servidor PostgreSQL:** La aplicación permitirá crear las	

reglas de control de acceso. Las mismas deben de cumplir con esta característica**:**

[Tipo de conexión] [database] [usuario] [IP] [Netmask] [Tipo de autentificación] [opciones]

Dependiendo del tipo de conexión y del tipo de autenticación [IP], [Netmask] y [opciones] pueden ser opcionales.

El **Tipo de conexión** puede tomar los siguientes valores:

- **host**: Una entrada *host* es usada para especificar máquinas remotas que están autorizadas para conectar al servidor PostgreSQL. El *postmaster* de PostgreSQL debe ser ejecutado con la opción -i (TCP/IP) para que una entrada de máquina funcione correctamente. Un ejemplo de *host* puede ser 10.53.3.174.
- **local:** Una entrada local es semánticamente lo mismo que una entrada host. Sin embargo, no necesita especificar una máquina a la que le esté permitido conectar. La entrada local es usada para conexiones de clientes que son iniciadas desde la misma máquina en la que está operando el servidor PostgreSQL, ejemplo 127.0.0.1 o localhost.
- **Hostssl:** Una entrada *hostssl* es usada para especificar máquinas (remotas o locales) que están autorizadas para conectar al servidor PostgreSQL usando SSL. El uso de SSL le garantiza que todas las comunicaciones entre el cliente y el servidor están encriptadas. Para que esto funcione, tanto el cliente como el servidor deben soportar SSL. El proceso postmaster debe ser ejecutado con las opciones -l (SSL) y -i (TCP/IP).

Este es el nombre de la base de datos a la que la máquina especificada está autorizada a conectar. La palabra clave ***database*** tiene tres posibles valores:

- **all:** La palabra clave *all* especifica que el cliente puede conectar a cualquier base de datos alojada en el servidor PostgreSQL.
- **sameuser:** Especifica que el cliente sólo puede conectar a una base de datos en la que coinciden los nombres de usuarios autenticados de los clientes.
- **name:** Un nombre determinado puede ser especificado, de forma que el cliente sólo puede conectar a la base de datos especificada por ese nombre.

Los campos **ip_addr** y **netmask** especifican una dirección IP, o rango de direcciones que están autorizadas a conectar al servidor PostgreSQL. El rango puede ser especificado describiendo una red IP con su máscara de red asociada. De lo contrario, para una única dirección IP, el campo *netmask* debería ser establecido a 255.255.255.255. Ejemplo de esto puede ser:

- 10.0.0.0 255.255.255.0
- 10.0.0.0/24

El **método de autenticación** especifica el tipo de autenticación que el servidor debería usar para que un usuario que intenta conectar a PostgreSQL. Los valores que puede tomar son:

- **trust**: El método *trust* permite a cualquier usuario del host (máquina) definido conectar a una base de datos PostgreSQL sin el uso de una contraseña. Este es el método predefinido
- **reject**: El método *reject* automáticamente deniega el acceso a PostgreSQL para esa máquina o usuario. Esta puede ser una configuración prudente para sitios en los que nadie está autorizado a conectar a su servidor de base de datos.
- **password:** El método *password* especifica que debe existir una contraseña para una conexión de usuario. El uso de éste método requerirá que el usuario que se conecta proporcione una contraseña que coincida con la contraseña encontrada en la tabla global del sistema *pg_shadow* para ese nombre de usuario, la contraseña será enviada en texto plano.
- **Crypt:** El método *crypt* es similar al método *password*. Cuando se usa *crypt*, la contraseña no es enviada en texto plano, sino a través de un formato simple de encriptación. El uso de éste método no es muy seguro, pero es mejor que usar el método de contraseñas planas.
- **Ident:** El método *ident* especifica que un mapa de identidad debería ser usado cuando una máquina está solicitando conexiones desde una IP válida listada en el archivo *pg_hba.conf*. Este método requiere una opción. La opción requerida puede ser tanto el término especial *sameuser*, o un mapa nombrado que es definido dentro del archivo *pg_ident.conf*.
- **auth_option**: El campo *auth_option* puede o no ser requerido, en función del tipo de método de autenticación que sea usado; sólo el método *ident* requiere una opción.

Modificar reglas de control de acceso al servidor PostgreSQL: Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificar las reglas de control de acceso al servidor. Se realizarán las mismas validaciones que en la funcionalidad Crear reglas.

Eliminar reglas de control de acceso al servidor PostgreSQL: Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de eliminar las reglas.

Prototipo de interfaz:

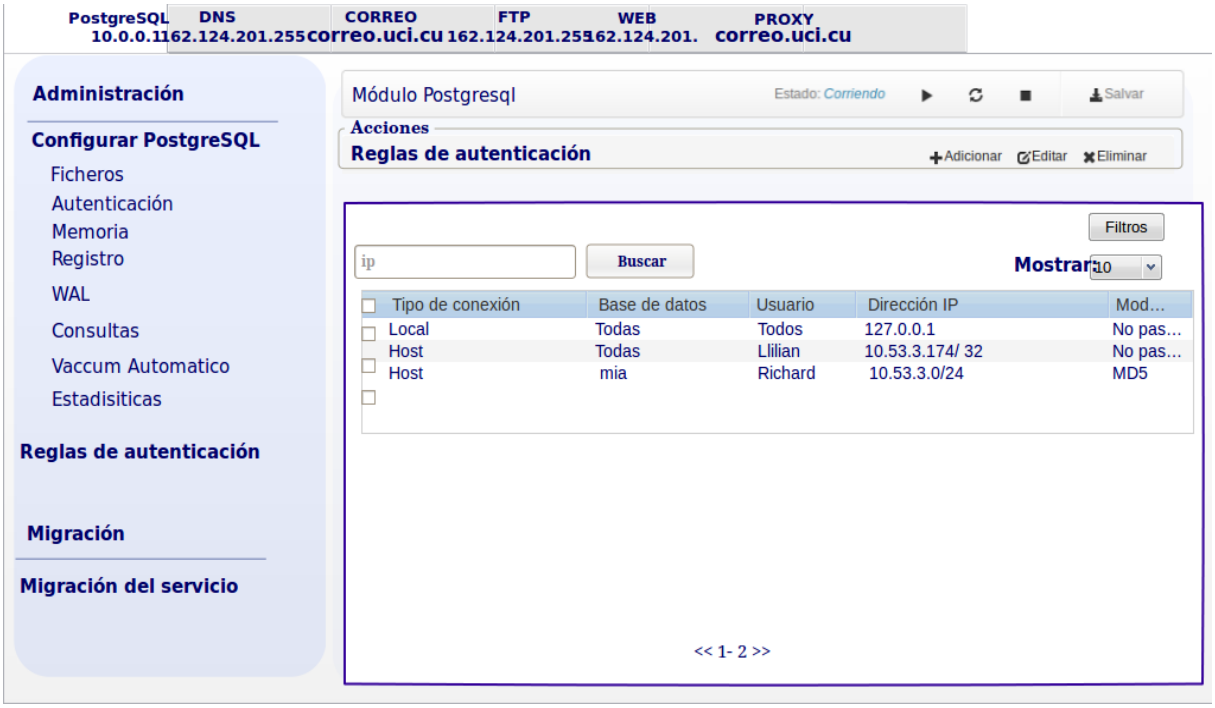


Figura 16: Prototipo_Control de acceso

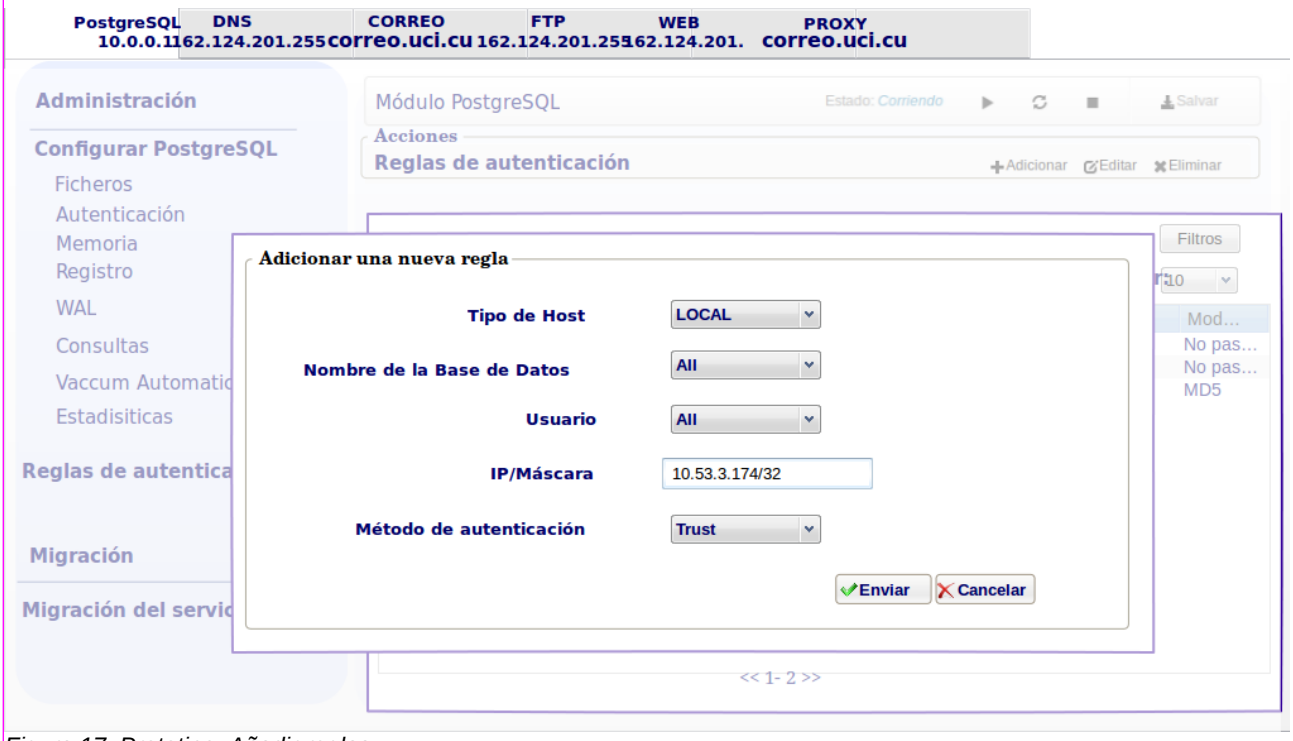


Figura 17: Prototipo_Añadir reglas

Historia de Usuario	
Número: HMAST_**PostgreSQL**_7	**Nombre Historia de Usuario:** Configuración para la integridad de los datos
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Llilian Fernández Casasayas	**Iteración Asignada:** 1
Prioridad en Negocio: Media	**Puntos Estimados:** 1 semana
Riesgo en Desarrollo: Media	**Puntos Reales:** 1 semana
Descripción: La aplicación mostrará los parámetros de configuración para guardar toda la información sobre las transacciones, cambios realizados en las base de datos, así como para garantizar la integridad de los mismos y los podrá modificar.	
Observaciones: Todos los cambios que se apliquen en esta interfaz se realizarán mediante la edición del fichero **/etc/postgresql/9.1/main/postgresql.conf.** Estos no serán guardados hasta que se seleccione el botón Aceptar. Cuando se realiza esta acción, se recargan las configuraciones del servicio. **Mostrar métodos para almacenar la cantidad de información que se escribe en el WAL:** La aplicación permitirá mostrar los métodos para guardar la información que se escribe en el WAL. Este parámetro se mostrará por el parámetro **wal_level.** **Modificar métodos para almacenar la cantidad de información que se escribe en el WAL:** El usuario podrá modificar el método para guardar cuánta información se escribe en el WAL, a través de un campo de selección que tiene tres posibles valores, estos son: • **minimal***:* Este escribe sólo la información necesaria para recuperarse de un accidente o parada inmediata. • **archive:** Añade almacenamiento requeridos para el archivado WAL • **hot_standby:** Añade información adicional necesaria para ejecutar consultas de sólo lectura en un servidor en espera siempre y cuando, el valor sea válido. **Habilitar que se escriban físicamente las actualizaciones en el disco:** Se brinda la posibilidad de que el servidor se asegure de que las actualizaciones se escriban físicamente en el disco y así garantizar que el clúster de base de datos se pueda recuperar a un estado coherente, después de un fallo en el sistema operativo o un accidente de hardware. Esto se realizará a través del	

parámetro **fsync** y el valor predeterminado es *on.*

Deshabilitar que se escriban físicamente las actualizaciones en el disco: El usuario puede deshabilitar que las actualizaciones se escriban físicamente en el disco, el parámetro tomará valor *off*.

Habilitar la opción de que las transacciones se comprometen a esperar a los registros WAL que se escriben en el disco: La aplicación permitirá habilitar si después de que una transacción realice un COMMIT, espera a guardar los registros WAL en disco antes de dar por válidas las operaciones. Este parámetro se mostrará por la directiva **synchronous_commit.**

Deshabilitar la opción de que las transacciones se comprometen a esperar a los registros WAL que se escriben en el disco: El usuario podrá decidir si la transacción se compromete a esperar a los registros WAL que se escriben en el disco a través de un campo de selección que tiene tres posibles valores: *on, off y local*. El valor por defecto y seguro, es *on*.

Mostrar método utilizado para forzar cambios WAL al disco: Si la opción fsync está activada, la aplicación mostrará el método empleado para actualizar las operaciones pendientes a los ficheros de segmentos WAL. Esta funcionalidad se mostrará por el parámetro **wal_sync_method.**

Modificar método utilizado para forzar cambios WAL al disco: El usuario podrá modificar los métodos utilizados para forzar cambios WAL al disco, a través de un campo de selección que tiene estos posibles valores:

- *open_datasync* escribe en los archivos WAL con *open* y la opción O_DSYNC.
- *fdatasync* invoca a *fdatasync* tras cada COMMIT.
- *fsync* invoca a *fsync* ttras cada COMMIT.
- f*sync_writethrough* invoca a *fsync* tras cada COMMIT, asegurando que la función realmente escribirá sus datos en disco.
- *open_sync* escribe en los archivos WAL con *open* y la opción O_SYNC.

Mostrar la cantidad de memoria compartida utilizada como buffer para los datos del WAL: La aplicación mostrará la cantidad de memoria compartida utilizada como *buffer* para los datos del WAL. Esto se podrá realizar a través del parámetro **wal_buffers.**

Modificar la cantidad de memoria compartida utilizada como buffer para los datos del WAL: El usuario podrá modificar la cantidad de memoria compartida utilizada como buffer para los datos del WAL, a través de un campo de texto y el valor tendrá que ser un número entero. Luego contendrá un campo de selección con valores de medidas Gigabyte, Kilobyte, Megabyte, en caso de que no sea válido, se mostrará un mensaje de error. El valor por defecto es de 64 KB y el ajuste de -1 seleccionará un tamaño aproximadamente a un 3 % de shared_buffers.

Mostrar la cantidad de segmentos que se chequearán: La aplicación permitirá mostrar el número de segmentos de archivos de registro entre puntos de comprobación automáticos WAL. Este parámetro se mostrará por el parámetro **checkpoint_segments.**

Modificar la cantidad de segmentos que se chequearán: El usuario podrá modificar la cantidad de segmentos de archivos de registro entre puntos de comprobación automáticos WAL, a través de un campo de texto y el valor tendrá que ser un número entero. El valor predeterminado es de 3 segmentos. Este parámetro tiene un rango de 1 a 1 000 000 000 de segmentos, en caso de que el valor no sea válido, se mostrará un mensaje de error.

Mostrar el tiempo en que se chequearán los segmentos: La aplicación mostrará el tiempo en que se realizarán los puntos de comprobación automáticos WAL. El valor predeterminado es de cinco minutos y se mostrará por el parámetro **checkpoint_timeout.**

Modificar el tiempo en que se chequearán los segmentos: El usuario tendrá la opción de modificar el tiempo máximo entre puntos de comprobación automáticos WAL, a través de un campo de texto y el valor tendrá que ser un número entero. Luego contendrá un campo de selección con los valores de medidas Hora, Minutos y Segundos. Los valores válidos están de 30 segundos a 1 hora en caso de que el valor no sea válido, se mostrará un mensaje de error.

Mostrar el destino de finalización del punto de chequeo: La aplicación permitirá mostrar el valor en que se ejecutará uniformemente el punto de chequeo actual durante el período de espera del siguiente. Este parámetro se mostrará por el parámetro **checkpoint_completion_target.**

Modificar el destino de finalización del punto de chequeo: El usuario podrá modificar el valor en que se ejecutará uniformemente el punto de chequeo actual durante el período de espera del siguiente, a través de un campo de texto y el valor tendrá que ser un número decimal. Su valor por

defecto es del 0.5. Siempre que el valor sea válido puede tomar valores de 0,0 a 1,0, si no toma valores válidos y está en ese rango mostrará un mensaje de error.

Una vez realizadas todas las configuraciones la aplicación brinda la posibilidad de guardar los cambios o cancelarlos.

Prototipo de interfaz:

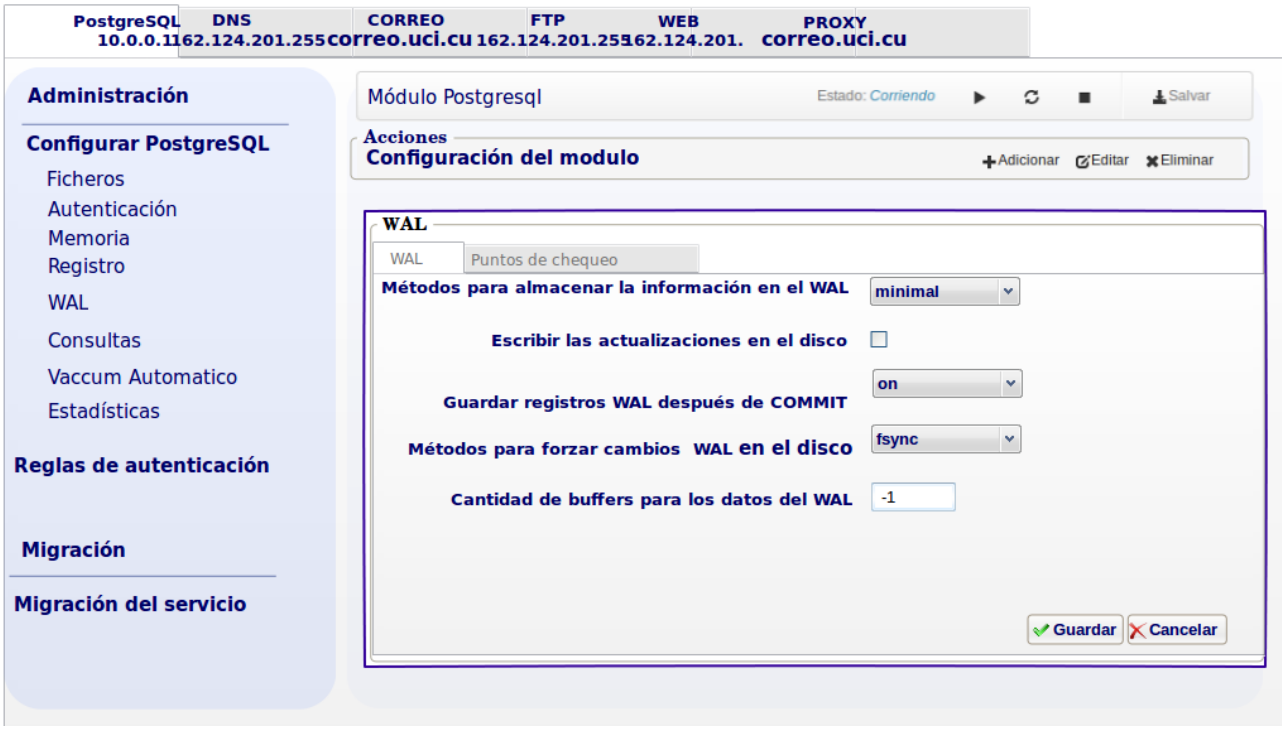


Figura 18: Prototipo_WAL

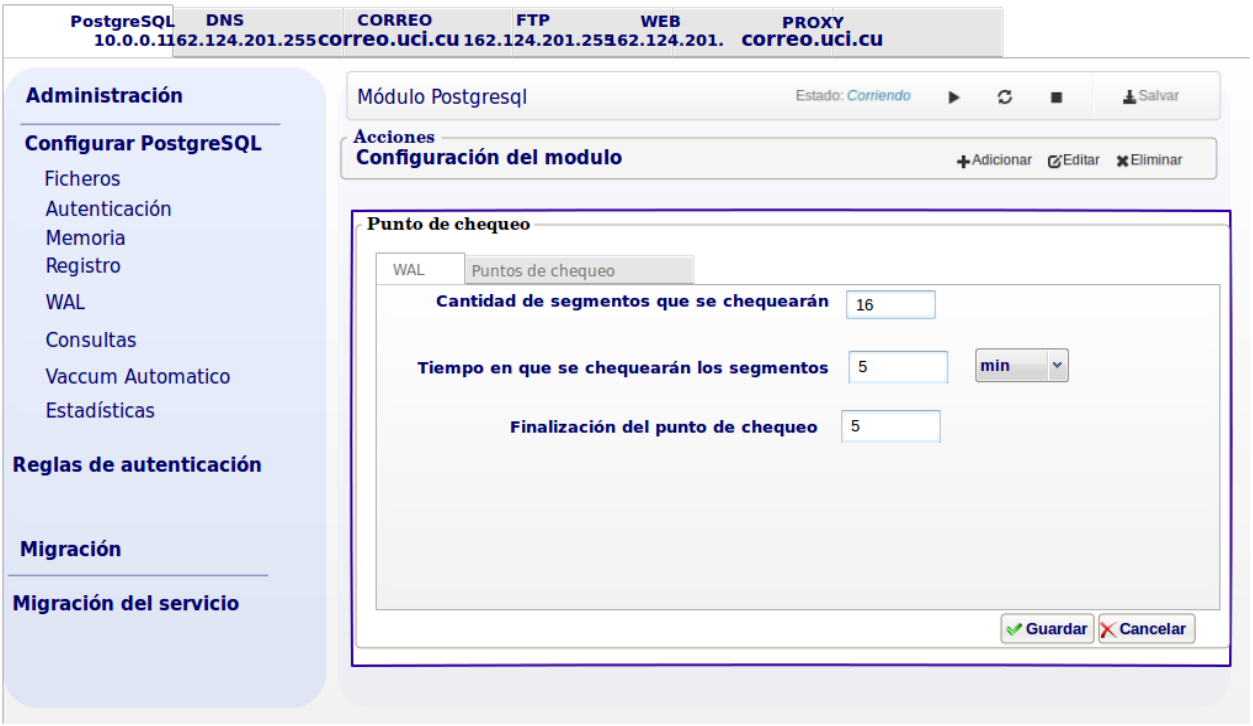


Figura 19: Prototipo_Punto de Chequeo

Historia de Usuario	
Número: HMAST_**PostgreSQL**_8	**Nombre Historia de Usuario:** Configuración de ajustes de consultas
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Llilian Fernández Casasayas	**Iteración Asignada:** 1
Prioridad en Negocio: Media	**Puntos Estimados:** 1 semana
Riesgo en Desarrollo: Media	**Puntos Reales:** 1 semana
Descripción: La aplicación mostrará los parámetros de configuración de ajustes de consultas y se podrán modificar.	
Observaciones: Todos los cambios que se apliquen en esta interfaz se realizarán mediante la edición del fichero **/etc/postgresql/9.1/main/postgresql.conf.** Estos no serán guardados hasta que se seleccione el botón Aceptar. Cuando se realiza esta acción, se recargan las configuraciones del servicio. **Mostrar el costo de leer una única página de base de datos en disco de forma secuencial:** La aplicación permitirá mostrar el costo de leer una única página de base de datos en disco de forma secuencial cuando la información se encuentra contigua en él. Este parámetro se mostrará por el parámetro **seq_page_cost.** **Modificar el costo de leer una única página de base de datos en disco de forma secuencial:** Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto y el valor tendrá que ser un número decimal. El valor predeterminado es 1,0. Este parámetro tiene un rango de 0,0 a 1 000 000 000.0, en caso de que el valor no se encuentre en ese rango, se mostrará un mensaje de error. **Mostrar el valor por el cual el planificador de consulta considera los accesos no secuenciales a disco:** La aplicación permitirá mostrar el valor por el cual el planificador de consulta considera los accesos no secuenciales a disco. Este parámetro se mostrará por el parámetro **random_page_cost.** **Modificar el valor por el cual el planificador de consulta considera los accesos no secuenciales a disco:** Una vez mostrado el valor de este parámetro se le brinda la posibilidad al	

usuario de modificarlo a través de un campo de texto y el valor tendrá que ser un número decimal, el valor predeterminado es 4,0. Este parámetro tiene un rango de 0,0 a 1 000 000 000.0, en caso de que el valor no se encuentre en ese rango, se mostrará un mensaje de error.

Mostrar el tamaño efectivo de la caché de disco que está disponible para una sola consulta: La aplicación permitirá mostrar el tamaño efectivo de la caché de disco que está disponible para una sola consulta. Este parámetro se mostrará por el parámetro **effective_cache_size.**

Modificar el tamaño efectivo de la caché de disco que está disponible para una sola consulta: Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto y el valor tendrá que ser un número entero. El valor predeterminado es 128 Mb. Los valores válidos están de 0 a 2048, en caso de que el valor no se encuentre en ese rango, se mostrará un mensaje de error.

Habilitar la opción de optimización de la consulta genética: La aplicación permitirá activar la opción de la optimización de la consulta genética. Esto está en *on* de forma predeterminada. Este parámetro se mostrará por el parámetro **geqo.**

Deshabilitar la opción de optimización de la consulta genética: En caso de que esté activada la opción de la optimización de la consulta genética, el usuario puede desactivarla y toma valor de *off.*

Mostrar el valor para que el optimizador de consulta planifique las consultas: Se mostrará el valor para que el optimizador de consulta planifique las consultas. Este parámetro se mostrará por el parámetro **geqo_threshold.**

Modificar el valor para que el optimizador de consulta planifique las consultas: Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto y el valor tendrá que ser un número entero. El valor predeterminado es 12. Los valores válidos están de 0 a 1 000 000 000, en caso de que el valor no se encuentre en ese rango, se mostrará un mensaje de error.

Mostrar el valor que el planeador de consultas utilizará en las restricciones de tablas: Se mostrará el valor que el planeador de consultas utilizará en las restricciones de tablas. Este parámetro se mostrará a través de **constraint_exclusion.**

Modificar el valor que el planeador de consultas utilizará en las restricciones de tablas: Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo a través de un campo de selección. Los valores permitidos son:

- ***on*** : examinar las restricciones de todas las tablas.
- ***off*** : nunca examinar las restricciones.
- ***partición:*** examinar las restricciones sólo para las tablas de la herencia.

La configuración predeterminada es partición.

Una vez realizadas todas las configuraciones la aplicación brinda la posibilidad de guardar los cambios o cancelarlos.

Prototipo de interfaz:

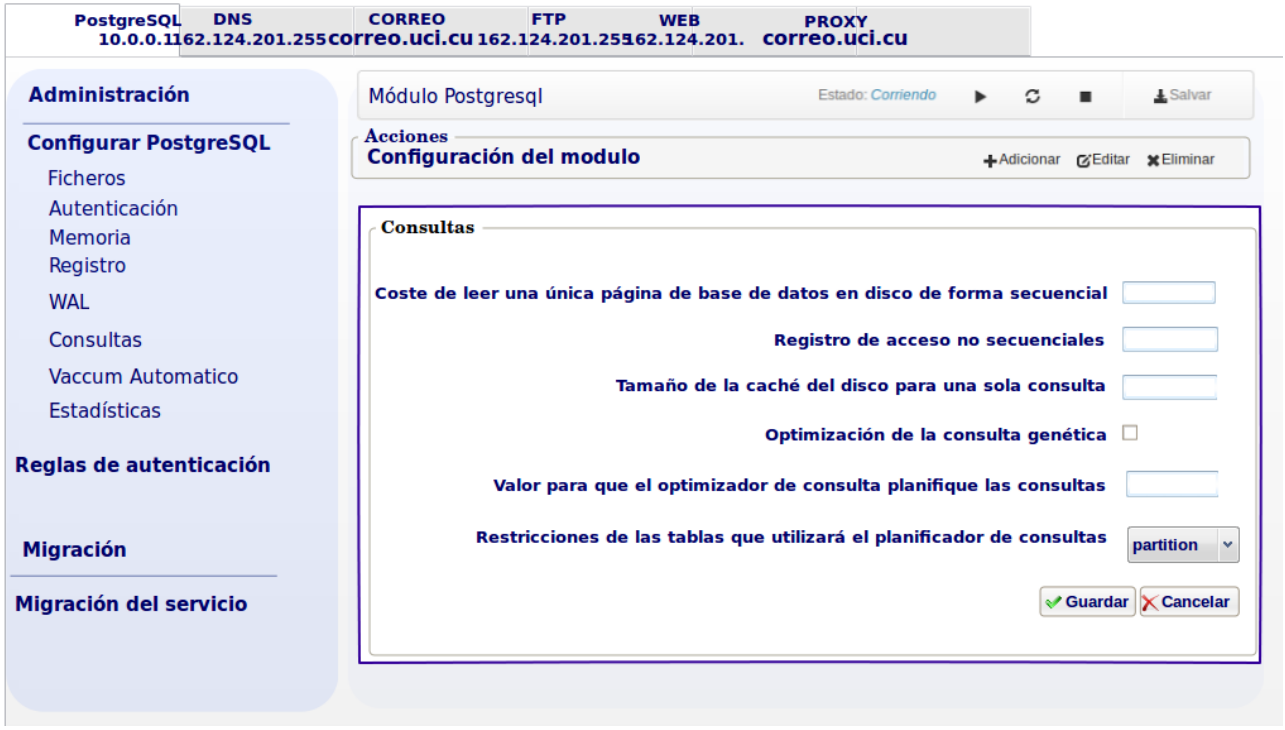


Figura 20: Prototipo_Consulta

Historia de Usuario	
Número: HMAST_**PostgreSQL**_9	**Nombre Historia de Usuario:** Configurar los registros del sistema.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Llilian Fernández Casasayas	**Iteración Asignada:** 1

Prioridad en Negocio: Media	**Puntos Estimados:** 1 semana
Riesgo en Desarrollo: Media	**Puntos Reales:** 1 semana
Descripción: La aplicación mostrará los parámetros de configuración para los registros del sistema y permitirá modificarlos.	
Observaciones: Todos los cambios que se apliquen en esta interfaz se realizarán mediante la edición del fichero **/etc/postgresql/9.1/main/postgresql.conf.** Estos no serán guardados hasta que se seleccione el botón Aceptar. Cuando se realiza esta acción, se recargan las configuraciones del servicio. **Mostrar los métodos que indican la salida a la que irán dirigidos los mensajes de registros:** La aplicación permitirá mostrar los métodos que indican la salida a la que irán dirigidos los mensajes de registros, y se mostrará por el parámetro **log_destination.** **Modificar los métodos que indican la salida a la que irán dirigidos los mensajes de registros:** Una vez mostrado el valor de este parámetro se le brinda la posibilidad al usuario de modificarlo a través de un campo de selección con los siguientes valores *stderr*, *csvlog*, *syslog* y eventlog. Este parámetro establecerá en una lista de los mensajes de registros, separados por coma. El valor predeterminado es *stderr*. **Habilitar la opción del colector de registro:** La aplicación permitirá activar la captura de los mensajes enviados por el servidor y redirigirlos, por ejemplo, a un archivo de registros. Se habilitará por el parámetro **logging_collector** y toma valor *on***.** **Deshabilitar la opción de colector de registro:** En caso de que esté activada la generación de registros, el usuario puede desactivarla y toma valor *off*. **Mostrar el directorio en el que se crearán los archivos de registro:** Si está activada la generación de registros**.** Esta opción permitirá mostrar la ruta del directorio por defecto donde se guardarán los registros y se encontrarán en el archivo pg_log. Se mostrará por el parámetro **log_directory** a través de un campo de texto. **Modificar el directorio en el que se crearán los archivos de registro:** Una vez que se muestre la ubicación del directorio donde se crearán los archivos de registro, el usuario podrá modificarla si lo desea, al seleccionar un botón que se encuentra ubicado al lado derecho del campo de texto, siempre que la ruta sea válida.	

Mostrar formato de los archivos de registros creados: Cuando el colector de registro está habilitado la aplicación mostrará cómo definir los nombres de los archivos de registros creados. El valor, se mostrará por el parámetro **log_filename.**

Modificar formato de los archivos de registros creados: Una vez que se muestre cómo definir los nombres de los archivos de registros creados**,** el usuario podrá modificarlo a través de un campo de texto, por lo que las variables de tiempo deberán escaparse empleando % (%Y = año, %m = mes, %d = día, entre otros), en caso de que no sea válido, se mostrará un mensaje de error.

Mostrar el tiempo para actualizar los registros: Si esta activada la opción del colector de registro la aplicación mostrará el tiempo de vida máximo de un archivo de registro y se mostrará por el parámetro **log_rotation_age.**

Modificar el tiempo para actualizar los registros: Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto y el valor tendrá que ser un número entero. El valor por defecto es 1 día. Una vez transcurrido este tiempo, se creará un nuevo archivo de registro. Si se ajusta a cero se desactivará la creación basada en el tiempo de los archivos de registros nuevos. Sus valores estarán en un rango de 0 a 10000 minutos, si el valor no se encuentra en ese rango se mostrará un mensaje de error.

Mostrar el tamaño de un archivo de registro: Si está activado el colector de registro la aplicación permitirá mostrar el tamaño máximo de un archivo de registro individual, y se mostrará por el parámetro **log_rotation_size.**

Modificar el tamaño de un archivo de registro: Una vez mostrado el valor de este parámetro se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto, y el valor tendrá que ser un número entero. Cuando alcance el tamaño deseado se creará un nuevo archivo de registro. El valor por defecto es 10 Mb. Si se ajusta a cero se desactivará la creación basada en el tamaño de los nuevos archivos de registro. Sus valores estarán en un rango de 0 a 1024 Mb, si el valor no se encuentra en ese rango se mostrará un mensaje de error.

Habilitar la opción de sobrescribir un archivo de registro: Cuando la opción colector de registro está habilitada, la aplicación permitirá sobrescribir, en lugar de anexar cualquier archivo de registro existente con el mismo nombre. Esto se producirá sólo cuando un nuevo archivo se crea debido a

la rotación basada en el tiempo. Se mostrará por el parámetro **log_truncate_on_rotation** y el valor será *on***.**

Deshabilitar la opción de sobrescribir un archivo de registro: En caso de que este activado sobrescribir un archivo de registro, el usuario puede desactivarlo y tomará valor *off.*

Mostrar niveles de mensajes que se envían al cliente: La aplicación mostrará los niveles de los mensajes que se envían al cliente y se mostrarán en orden decreciente por el parámetro **client_min_messages.**

Modificar niveles de mensajes que se envían al cliente: Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo a través de un campo de selección con los siguientes valores:

- **DEBUG [5...1]***:* información detallada para los desarrolladores.
- **INFO:** proporciona información implícitamente solicitada por el usuario, por ejemplo, al ejecutar VACUUM VERBOSE.
- **NOTICE:** información que puede ayudar a los usuarios, por ejemplo, cuando se trunca el nombre de un identificador por ser demasiado largo.
- **WARNING:** avisos al usuario, por ejemplo, en caso de hacer un COMMIT fuera de una transacción.
- **ERROR:** informa de un error que ha causado que aborte un comando.
- **LOG:** información interesante para los administradores, por ejemplo, la actividad de los puntos de chequeo.
- **FATAL:** errores que han producido que aborte la sesión.
- **PANIC:** errores que han producido que aborten todas las sesiones del servidor.

En caso de no ser válido, se mostrará un mensaje de error. El valor predeterminado es *notice.* Cada nivel incluye todos los niveles que le siguen.

Mostrar niveles de mensajes que se escriben en el servidor: La aplicación mostrará los niveles de los mensajes que se escriben en el servidor y se mostrarán en orden decreciente por el parámetro **log_min_messages.**

Modificar niveles de mensajes que se escriben en el servidor: Una vez mostrado el valor de

este parámetro, se le brinda la posibilidad al usuario de modificarlo a través de un campo de selección con los siguientes valores: DEBUG5, DEBUG4, DEBUG3, DEBUG2, DEBUG1, INFO, AVISO, WARNING, ERROR LOG, FATAL, PANIC que tienen las mismas propiedades que en client_min_messages. El valor predeterminado es *warning*.

Mostrar los errores que se registren en el servidor: La aplicación controlará las sentencias SQL que causan una condición de error y se registran en el registro del servidor. Estos errores se mostrarán en orden decreciente y por el parámetro **log_min_error_statement.**

Modificar los errores que se registren en el servidor: Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo a través de un campo de selección con los siguientes valores: DEBUG5, DEBUG4, DEBUG3, DEBUG2, DEBUG1, INFO, AVISO, WARNING, ERROR LOG, FATAL y PANIC, que tienen las mismas propiedades que en client_min_messages. El valor predeterminado es ERROR.

Mostrar la opción de registrar las consultas que hayan durado más tiempo: La aplicación mostrará si se registran las instrucciones que hayan durado más tiempo que el especificado. Esto se mostrará por el parámetro l**og_min_duration_statement.**

Modificar la opción de registrar las consultas que hayan durado más tiempo: Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto y el valor tendrá que ser un número entero. El valor por defecto será -1, lo que conlleva que nada sea registrado, y si se cambia a 0, todo se registrará. Puede llegar a ser útil para identificar aquellas sentencias SQL que consumen recursos del sistema durante demasiado tiempo.

Mostrar el nivel de detalle que se incluye en cada mensaje añadido al servidor: La aplicación mostrará el nivel de detalle que se incluye en cada mensaje añadido al log del servidor y se mostrarán en orden decreciente por el parámetro **log_error_verbosity.**

Modificar el nivel de detalle que se incluye en cada mensaje añadido al servidor: Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo a través de un campo de selección con los siguientes valores: *terse*, *default* y *verbose*. El valor predeterminado es *default*.

Mostrar qué sentencias SQL se registrarán: La aplicación controlará qué sentencias SQL se registrarán y se mostrará por el parámetro **log_statement.**

Modificar qué sentencias SQL se registrarán: Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo a través de un campo de selección con los siguientes valores: *none*, *ddl*, *mod* y *all.*

- **ddl:** registra todas las sentencias de definición de datos como: CREATE, ALTER y DROP
- **mod:** registra las incluidas en *ddl* y además las modificaciones de datos como: INSERT, UPDATE, DELETE, TRUNCATE y COPY FROM. También PREPARE, EXECUTE y EXPLAIN ANALYZE
- **all:** las registra todas.

El valor por defecto es *none*.

Una vez realizadas todas las configuraciones la aplicación brinda la posibilidad de guardar los cambios o cancelarlos.

Prototipo de interfaz:

Figura 21: Prototipo_Registro

Historia de Usuario	
Número: HMAST_PostgreSQL_10	**Nombre Historia de Usuario:** Configurar las estadísticas de ejecución.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Llilian Fernández Casasayas	**Iteración Asignada:** 1
Prioridad en Negocio: Media	**Puntos Estimados:** 1 semana
Riesgo en Desarrollo: Media	**Puntos Reales:** 1 semana
Descripción: La aplicación mostrará los parámetros de configuración de las estadísticas de ejecución y se podrán modificar.	
Observaciones: Todos los cambios que se apliquen en esta interfaz se realizarán mediante la edición del fichero **/etc/postgresql/9.1/main/postgresql.conf.** Estos no serán guardados hasta que se seleccione el botón Aceptar. Cuando se realiza esta acción, se recargan las configuraciones del servicio. **Habilitar la opción de recopilar la información sobre el comando que se está ejecutando:** La aplicación permitirá activar la opción de recopilar la información sobre el comando que se está ejecutando de cada sesión, así como el momento en que se inició la ejecución de comandos. Este parámetro se encuentra en *on* de forma predeterminada y se mostrará por el parámetro **track_activities.** **Deshabilitar la opción de recopilar la información sobre el comando que se está ejecutando:** En caso de que esté activada la recopilación de información sobre el comando que se está ejecutando, el usuario puede desactivarla y el parámetro tomará valor *off*. **Habilitar la recopilación de estadísticas sobre la actividad de las base de datos:** La aplicación permitirá activar la opción de recopilar las estadísticas sobre la actividad de la base de datos. Este parámetro está activado por defecto y se mostrará por el parámetro **track_counts.** **Deshabilitar la recopilación de estadísticas sobre la actividad de las base de datos:** En caso de que esté activada la opción de recopilar las estadísticas sobre la actividad de la base de datos el usuario puede desactivarla, y el parámetro tomara valor *off.* **Mostrar seguimientos de las estadísticas:** La aplicación permitirá mostrar el seguimiento de las funciones y el tiempo utilizado. Se mostrará por el parámetro **track_functions.**	

Modificar seguimientos de las estadísticas : Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo a través de un campo de selección. Los valores que puede tomar dicho parámetro son: *pl* para rastrear sólo las funciones del lenguaje procesal, *all* para realizar un seguimiento de las funciones de SQL y el lenguaje C y *none* para desactivar la función de seguimiento de estadísticas. El valor predeterminado es none.

Mostrar el número de bytes reservados para mostrar la consulta que se está ejecutando: La aplicación permitirá mostrar el número de bytes reservados para mostrar la consulta que se está ejecutando. Este parámetro se mostrará por el parámetro **track_activity_query_size.**

Modificar el número de bytes reservados para mostrar la consulta que se está ejecutando: Una vez mostrado el valor de este parámetro se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto, el valor tendrá que ser un número entero. El valor por defecto es 1024**.** Los valores válidos son de 0 a 2048 bytes, y si el valor no se encuentra en dicho rango, se mostrará un mensaje de error.

Mostrar directorio para guardar los datos de las estadísticas: La aplicación permitirá mostrar el directorio para guardar datos de las estadísticas temporales. Esto puede ser una ruta relativa al directorio de datos o una ruta absoluta. El valor predeterminado es pg_stat_tmp y se mostrará por el parámetro **stats_temp_directory.**

Modificar directorio para guardar los datos de las estadísticas: Una vez mostrado el valor de este parámetro, se le brinda la posibilidad al usuario de modificarlo.

Una vez realizadas todas las configuraciones la aplicación brinda la posibilidad de guardar los cambios o cancelarlos.

Prototipo de interfaz:

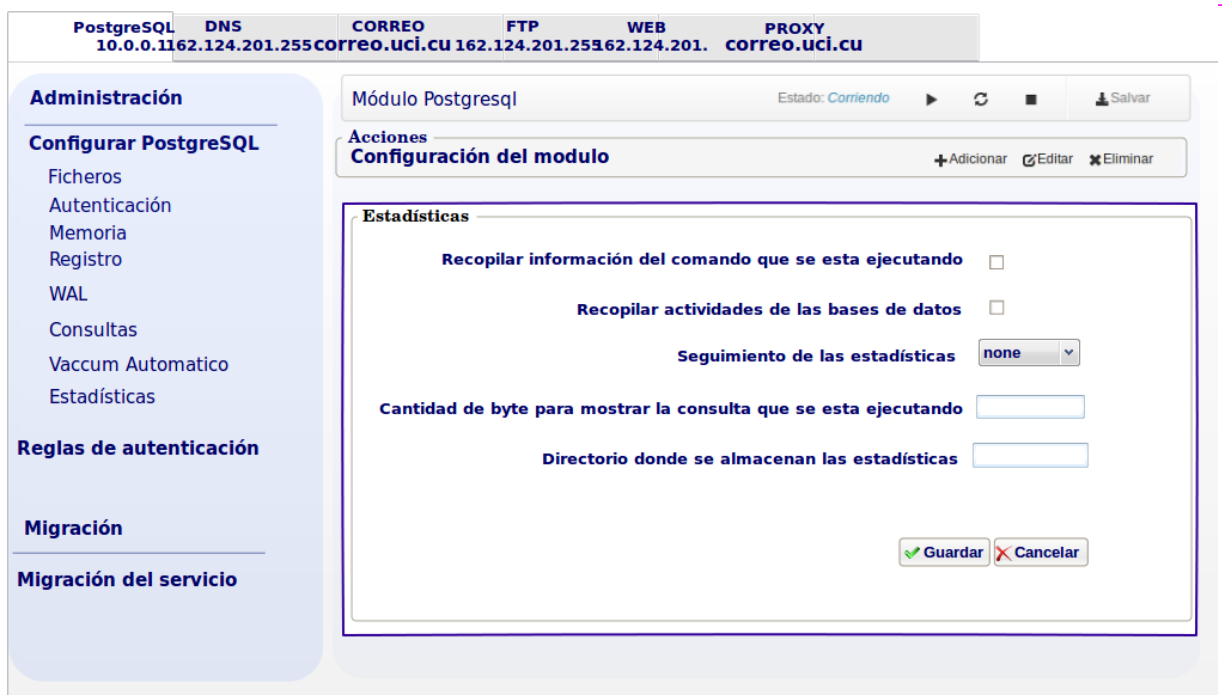


Figura 22: Prototipo_Estadísticas

Historia de Usuario	
Número: HMAST_PostgreSQL_11	**Nombre Historia de Usuario:** Configurar parámetros para revisar periódicamente las tablas con modificaciones considerables.
Modificación de Historia de Usuario Número: Ninguna	
Usuario: Llilian Fernández Casasayas	**Iteración Asignada:** 1
Prioridad en Negocio: Media	**Puntos Estimados:** 1 semana
Riesgo en Desarrollo: Media	**Puntos Reales:** 1 semana
Descripción: La aplicación mostrará los parámetros de configuración para revisar periódicamente las tablas con modificaciones considerables y se podrán modificar.	
Observaciones: Todos los cambios que se apliquen en esta interfaz se realizarán mediante la edición del fichero **/etc/postgresql/9.1/main/postgresql.conf.** Estos no serán guardados hasta que se seleccione el botón Aceptar. Cuando se realiza esta acción, se recargan las configuraciones del servicio. **Habilitarla opción de ejecutar el demonio lanzador autovacuum:** La aplicación permitirá habilitar la opción de ejecutar el demonio lanzador autovacuum. El valor por defecto es *on*, aunque para un funcionamiento correcto debe estar habilitada la opción **track_counts** y se mostrará por el	

parámetro **autovacuum.**

Deshabilitar la opción de ejecutar el demonio lanzador autovacuum: En caso de que esté activada la opción de ejecutar el demonio lanzador autovacuum, el usuario puede desactivarla y toma valor *off.*

Mostrar el número máximo de procesos autovacuum: La aplicación permitirá que se muestre el número máximo de procesos autovacuum que se esté ejecutando en un momento dado. Este parámetro se mostrará por el parámetro **autovacuum_max_workers.**

Modificar el número máximo de procesos autovacuum: Una vez mostrado el valor de este parámetro se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto, el valor tendrá que ser un número entero. El valor predeterminado es 3. Los valores válidos son de 0 a 100 y si el valor no se encuentra en dicho rango, se lanzará un mensaje de error.

Mostrar el tiempo que transcurre entre cada autovacuum: La aplicación permitirá que se muestre el tiempo mínimo que transcurre entre cada autovacuum, para una determinada base de datos, y se mostrará por el parámetro **autovacuum_naptime.**

Modificar el tiempo que transcurre entre cada autovacuum: Una vez mostrado el valor de este parámetro se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto, el valor tendrá que ser un número entero, luego tendrá un campo selector por el cual se seleccionará si se quiere mostrar el tiempo en horas o minutos. El retardo se mide en segundos, y el valor predeterminado es de 1 minuto. Los valores válidos son de 0 a 30000 minutos, si el valor no se encuentra en dicho rango, se lanzará un mensaje de error.

Mostrar la cantidad de filas para ejecutar VACUUM: La aplicación permitirá mostrar el número mínimo de tuplas actualizadas o eliminadas, y se mostrará por el parámetro **autovacuum_vacuum_threshold.**

Modificar la cantidad de filas para ejecutar VACUUM: Una vez mostrado el valor de este parámetro se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto, el valor tendrá que ser un número entero. El valor predeterminado es de 50 tuplas. Los valores válidos son de 0 a 1 000 000 000, si el valor no se encuentra en dicho rango se lanzará un mensaje de error.

Mostrar la cantidad de filas para ejecutar ANALYZE: La aplicación mostrará el número mínimo de tuplas insertadas, actualizadas o eliminadas y se mostrará por el parámetro **autovacuum_analyze_threshold.**

Modificar la cantidad de filas para ejecutar ANALYZE: Una vez mostrado el valor de este parámetro se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto, el valor tendrá que ser un número entero. El valor predeterminado es 50 tuplas. Los valores válidos son de 0 a 1 000 000 000, si el valor no se encuentra en dicho rango se lanzará un mensaje de error.

Mostrar el por ciento del tamaño de tabla que lanzará el VACUUM: La aplicación mostrará una fracción del tamaño de la tabla para agregar el número mínimo de tuplas actualizadas o eliminadas (autovacuum_vacuum_threshold) a la hora de decidir si se inicia un vacío, y se mostrará por el parámetro **autovacuum_vacuum_scale_factor.**

Modificar el por ciento del tamaño de tabla que lanzará el VACUUM: Una vez mostrado el valor de este parámetro se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto, el valor tendrá que ser un número decimal. El valor predeterminado es 0,2 (20% del tamaño de la tabla). Los valores válidos es de 0,0 a 100.0, si el valor no se encuentra en dicho rango, se lanzará un mensaje de error.

Mostrar la cantidad de transacciones para que se ejecute un vacuum aunque se haya desactivado el autovacuum: La aplicación mostrará la cantidad de transacciones para que se ejecute un vacuum aunque se haya desactivado el autovacuum y se mostrará por el parámetro **autovacuum_freeze_max_age.**

Modificar la cantidad de transacciones para que se ejecute un vacuum aunque se haya desactivado el autovacuum: Una vez mostrado el valor de este parámetro se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto, el valor tendrá que ser un número entero. El valor por defecto es de 200 000 000. Los valores válidos son de 0 a 200 000 000 de transacciones, si el valor no se encuentra dentro de ese rango se mostrará un mensaje de error.

Mostrar el por ciento del tamaño de tabla que lanzará el ANALYZE: La aplicación mostrará una fracción del tamaño de la tabla para agregar el número mínimo de tuplas insertadas, actualizadas o

borradas (autovacuum_analyze_threshold) si se dispara un ANALIZAR y se mostrará por el parámetro **autovacuum_analyze_scale_factor.**

Modificar el por ciento del tamaño de tabla que lanzará el ANALYZE: Una vez mostrado el valor de este parámetro se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto, el valor tendrá que ser un número decimal. El valor predeterminado es 0,1. Los valores válidos son de 0,0 a 100.0, si el valor no se encuentra dentro de ese rango se mostrará un mensaje de error.

Mostrar el tiempo que el proceso de VACUUM automático debe dormir entre ejecuciones: La aplicación mostrará el tiempo que el proceso de VACUUM automático debe dormir entre ejecuciones y se mostrará por el parámetro **autovacuum_vacuum_cost_delay.**

Modificar el tiempo que el proceso de VACUUM automático debe dormir entre ejecuciones: Una vez mostrado el valor de este parámetro se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto, el valor tendrá que ser un número entero. Si se especifica -1, se utilizará el valor establecido por el parámetro **vacuum_cost_delay**. El valor predeterminado es de 20 milisegundos. Los valores válidos van desde -1 a 100 milisegundos, si el valor no se encuentra dentro de ese rango, se mostrará un mensaje de error.

Mostrar el número de operaciones que pueden llevar a cabo los procesos del autovacuum: La aplicación mostrará el número de operaciones que pueden llevar a cabo los procesos del autovacuum. Se mostrará por el parámetro **autovacuum_vacuum_cost_limit.**

Modificar el número de operaciones que pueden llevar a cabo los procesos del autovacuum: Una vez mostrado el valor de este parámetro se le brinda la posibilidad al usuario de modificarlo a través de un campo de texto, el valor tendrá que ser un número entero. EL valor predeterminado es -1 lo que implica que se empleará **vacuum_cost_limit** para establecer este parámetro. Este valor se distribuye proporcionalmente entre los autovacuum que se ejecutan. Los valores válidos van desde -1 a 10000, si el valor no se encuentra dentro de ese rango se mostrará un mensaje de error.

Una vez realizadas todas las configuraciones la aplicación brinda la posibilidad de guardar los cambios o cancelarlos.

Prototipo de Interfaz:

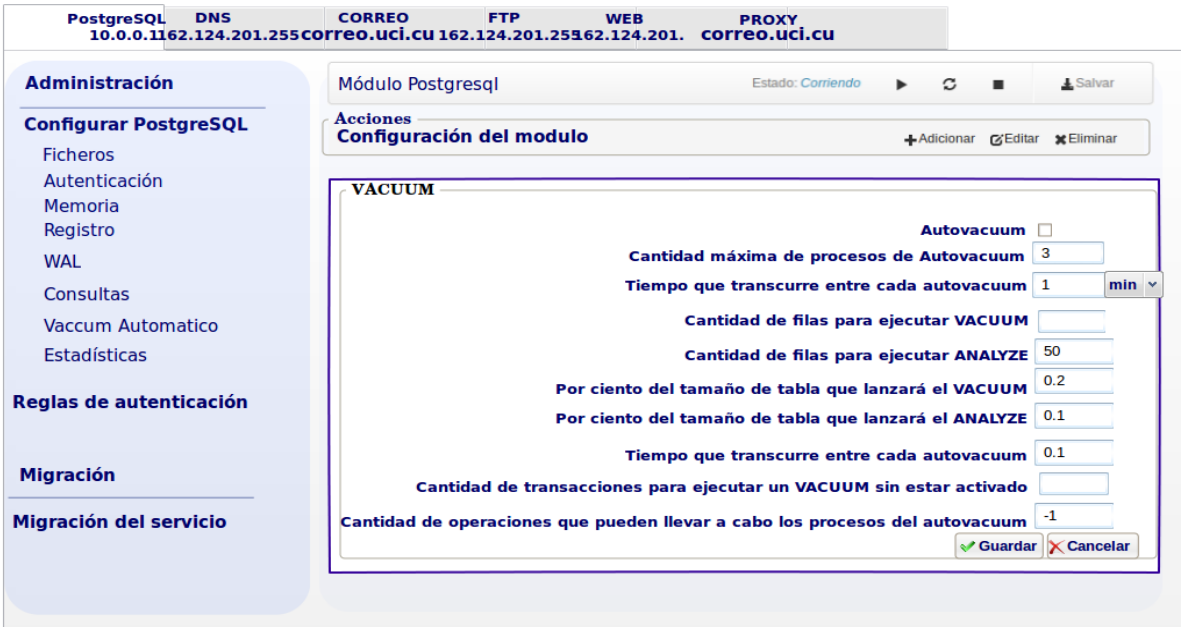


Figura 23: Prototipo_Autovacuum

Anexo 3: Planificación de la implementación.

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
1	Dar cumplimiento a las Historias de Usuario que describen los requisitos funcionales de prioridad Alta.	HU_1, HU_2, HU_3, HU_4, HU_5, HU_6	3 semanas.
2	Dar cumplimiento a las Historias de Usuario que describen los requisitos funcionales de prioridad Media.	HU_7, HU_8, HU_9, HU_10 y HU_11.	7 semanas.

Anexo 4: Tareas de Ingeniería

	Tarea de Ingeniería
Número Tarea: 1	**Número Historia de Usuario:** HMAST_PostgreSQL_1
Nombre Tarea: Comprobar comando para realizar la instalación del servidor.	

Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 17/02/14	**Fecha Fin: 19/02/14**
Programador Responsable: Llilian Fernández Casasayas	
Descripción: Se estudia el comando **apt-get install postgresql**, el cual tiene como finalidad la instalación del servidor postgresql.	

	Tarea de Ingeniería
Número Tarea: 2	**Número Historia de Usuario:** HMAST_PostgreSQL_1
Nombre Tarea: Creación de un método para realizar la instalación del servidor.	
Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.2
Fecha Inicio: 18/02/14	**Fecha Fin: 21/02/14**
Programador Responsable: Llilian Fernández Casasayas	
Descripción: Se crea un método que sea capaz de ejecutar el comando **apt-get install postgresql**	

	Tarea de Ingeniería
Número Tarea: 3	**Número Historia de Usuario:** HMAST_PostgreSQL_1
Nombre Tarea: Diseño de interfaz para realizar la instalación y administración del servicio.	
Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 19/02/14	**Fecha Fin: 23/02/14**
Programador Responsable: Llilian Fernández Casasayas	
Descripción: Se diseña una interfaz amigable que brinde la posibilidad de instalar el servidor, o en caso de estar instalado, se muestran las opciones de administrar el servicio. Además, se muestra una imagen animada la cual indica que se está realizando el proceso de instalación. En caso de que se desee cancelar la acción, la interfaz cuenta con un botón que tiene como finalidad realizar dicha operación.	

	Tarea de Ingeniería
Número Tarea: 4	**Número Historia de Usuario:** HMAST_PostgreSQL_2
Nombre Tarea: Comprobar comando para administrar el servicio.	
Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 24/02/14	**Fecha Fin: 26/02/14**
Programador Responsable: Llilian Fernández Casasayas	
Descripción: Se ejecutan los comandos para la administración del servicio: **service postgresql start** (Iniciar) **service postgresql restart** (Reiniciar) **service postgresql stop** (Detener) Posteriormente se comprueba su correcto funcionamiento.	

	Tarea de Ingeniería
Número Tarea: 5	**Número Historia de Usuario:** HMAST_PostgreSQL_3
Nombre Tarea: Comprobar formato del archivo principal de configuración donde se guardan las configuraciones realizadas en el servidor.	
Tipo de Tarea: Investigación	**Puntos Estimados:** 0.2
Fecha Inicio: 3/03/14	**Fecha Fin: 9/03/14**
Programador Responsable: Llilian Fernández Casasayas	
Descripción: Estudiar la estructura del fichero donde se guardan las asignaciones que realiza el servidor, con el objetivo de ver los parámetros más importantes que se pueden mostrar y administrar mediante la herramienta.	

	Tarea de Ingeniería
Número Tarea: 6	**Número Historia de Usuario:** HMAST_PostgreSQL_4
Nombre Tarea: Comprobar las directivas que permiten la configuración de la conexión del servidor.	

Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 10/03/14	**Fecha Fin: 12/03/14**
Programador Responsable: Llilian Fernández Casasayas	
Descripción: El fichero de configuración del servidor postgresql, ubicado en **etc/postgresql/9.1/main/postgresql.conf** cuenta con la directiva port, la cual establece el o los puertos por los cuales el servidor escuchará las peticiones de conexión, esté trae por defecto el puerto 5432. Por su parte la directiva ListenAddress, indica **l**a dirección desde las que se escucharán conexiones a PostgreSQL y la directiva **max_connections** establece la cantidad de conexiones que se realizarán al servidor PostgreSQL.	

	Tarea de Ingeniería
Número Tarea: 7	**Número Historia de Usuario:** HMAST_PostgreSQL_4
Nombre Tarea: Diseño de una interfaz que permita la configuración de la conexión y autenticación en el servidor.	
Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 10/03/14	**Fecha Fin: 20/03/14**
Programador Responsable: Llilian Fernández Casasayas	
Descripción: Se diseña una interfaz que incluye varios campos de texto, uno para modificar el puerto por el cual el servidor escuchará las peticiones de conexión, al lado tendrá un botón por si se desea establecer otro puerto. También contendrá un campo para establecer las direcciones de escucha. Así como un campo con el máximo de conexiones al servidor PostgreSQL, algunos parámetros de configuración de la seguridad, como el de encriptar contraseña y establecer una conexión SSL	

	Tarea de Ingeniería
Número Tarea: 8	**Número Historia de Usuario:** HMAST_PostgreSQL_5
Nombre Tarea: Diseño de una interfaz que permita la configuración de los parámetros de uso de recursos.	
Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.1

Fecha Inicio: 21/03/14	**Fecha Fin: 30/03/14**
Programador Responsable: Llilian Fernández Casasayas	
Descripción: Se diseña una interfaz que muestra los campos con los parámetros de configuración de uso de recursos. La misma contiene un campo de texto para modificar el tamaño de buferes para memoria y otro campo de texto al cual se le podrá establecer la cantidad de buferes temporales. Además se podrá establecer la cantidad de transacciones que estarán en el estado preparado, así como la cantidad de memoria utilizada para operaciones de mantenimiento en campos de textos. También, en campos de textos se podrán configurar los parámetros del VACUUM, como son el límite y el tiempo en que este se ejecutará.	

	Tarea de Ingeniería
Número Tarea: 9	**Número Historia de Usuario:** HMAST_PostgreSQL_6
Nombre Tarea: Comprobar el formato de las reglas de conexión en el fichero de configuración pg_hba.conf del servidor.	
Tipo de Tarea : Investigación	**Puntos Estimados:** 0.2
Fecha Inicio: 7/12/13	**Fecha Fin: 10/12/13**
Programador Responsable: Llilian Fernández Casasayas	
Descripción: Estudiar la estructura de las reglas de conexión en el fichero de configuración pg_hba.conf con el objetivo de buscar todos los parámetros que se pueden poner en la creación de las mismas. Comprobar además el funcionamiento de las reglas de conexión cuando se cambian los valores que tiene definido.	

	Tarea de Ingeniería
Número Tarea: 10	**Número Historia de Usuario:** HMAST_PostgreSQL_6
Nombre Tarea: Comprobar las reglas de conexión de tipo de Host hostssl	
Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.2
Fecha Inicio: 1/04/14	**Fecha Fin: 13/04/14**
Programador Responsable: Llilian Fernández Casasayas	
Descripción: Colocar varias reglas en el pg_hba.conf con diferentes formatos para comprobar su funcionamiento y los parámetros que lleva.	

	Tarea de Ingeniería
Número Tarea: 11	**Número Historia de Usuario:** HMAST_PostgreSQL_6
Nombre Tarea: Comprobar las reglas de conexión de tipo de Host host	
Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.2
Fecha Inicio: 2/04/14	**Fecha Fin: 3/04/14**
Programador Responsable: Llilian Férnandez Casasayas	
Descripción: Colocar varias reglas en el pg_hba.conf con diferentes formatos para comprobar su funcionamiento y los parámetros que lleva.	

	Tarea de Ingeniería
Número Tarea: 12	**Número Historia de Usuario:** HMAST_PostgreSQL_6
Nombre Tarea: Comprobar las reglas de conexión de tipo de Host local	
Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.2
Fecha Inicio: 4/04/14	**Fecha Fin: 5/04/14**
Programador Responsable: Llilian Fernández Casasayas	
Descripción: Colocar varias reglas en el pg_hba.conf con diferentes formatos para comprobar su funcionamiento y los parámetros que lleva.	

	Tarea de Ingeniería
Número Tarea: 13	**Número Historia de Usuario:** HMAST_PostgreSQL_6
Nombre Tarea: Implementación de un método que permita crear o modificar una regla de conexión así como sus valores.	
Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 6/04/14	**Fecha Fin: 9/04/14**
Programador Responsable: Llilian Fernández Casasayas	
Descripción: Se implementa un método que sea capaz de crear o modificar una regla de conexión, así como establecer o modificar los valores correspondientes a la regla de conexión.	

	Tarea de Ingeniería
Número Tarea: 14	**Número Historia de Usuario:** HMAST_PostgreSQL_6
Nombre Tarea: Diseño de una interfaz que permita gestionar las reglas de conexión al servidor.	
Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 10/04/14	**Fecha Fin: 13/03/14**
Programador Responsable: Llilian Fernández Casasayas	
Descripción: Se diseña una interfaz que muestra un listado con todas las reglas de conexión creadas en el servidor. Mediante la selección del nombre de una regla de conexión se puede acceder a otra interfaz para modificar los valores ya establecidos. Además, cuenta con un botón el cual permite acceder a una nueva interfaz para la creación de una nueva regla de conexión.	

	Tarea de Ingeniería
Número Tarea: 15	**Número Historia de Usuario:** HMAST_PostgreSQL_6
Nombre Tarea: Diseño de una interfaz que permita adicionar o modificar una regla de conexión.	
Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 17/03/14	**Fecha Fin: 21/03/14**
Programador Responsable: Llilian Fernández Casasayas	
Descripción: Se diseña una interfaz con varios campos que contendrán los valores de la nueva regla de conexión. En los campos *select* se podrán escoger el tipo de host y la máscara. Además las opciones del método y el ip y serán especificados en campos de textos.	

	Tarea de Ingeniería
Número Tarea: 16	**Número Historia de Usuario:** HMAST_PostgreSQL_7
Nombre Tarea: Diseño de una interfaz que permita la configuración de los parámetros para la integridad de los datos	
Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.1

Fecha Inicio: 21/04/14	**Fecha Fin: 30/04/14**
Programador Responsable: Llilian Fernández Casasayas.	
Descripción: Se diseña una interfaz que muestra los campos con los parámetros de configuración para la integridad de los datos. Contiene un campo de selección para escoger los métodos para almacenar la información en el WAL, se podrá definir si se escriben las actualizaciones en el disco Además se podrá establecer la cantidad de bufferes para los datos del WAL, así como seleccionar guardar registros WAL después del commit y los métodos para forzar cambios WAL en el disco.	

	Tarea de Ingeniería
Número Tarea: 17	**Número Historia de Usuario:** HMAST_PostgreSQL_9
Nombre Tarea: Comprobar las directivas relacionadas con las configuraciones de formato de los registros del sistema del servidor PostgreSQL.	
Tipo de Tarea : Desarrollo	**Puntos Estimados:** 0.1
Fecha Inicio: 30/04/14	**Fecha Fin: 10/05/14**
Programador Responsable: Llilian Fernandez Casasayas	
Descripción: En el fichero de configuración postgresql.conf, la directiva log_directory establece donde se registran los formatos del sistema y el formato especifico definido por la directiva log_filename.	

Anexo 5: Carta de Aceptación

El Jefe del Departamento de Servicios Integrales de Migración Asesoría y Soporte (SIMAYS) y el Jefe del proyecto Herramienta para la Migración y Administración de Servicios Telemáticos (HMAST), certifican por este medio que el Trabajo de Diploma, titulado **"Módulo de administración para el servicio PostgreSQL en la Herramienta para la Migración y Administración de Servicios Telemáticos(HMAST)"**, realizado en el **Departamento SIMAYS** cumple con los objetivos trazados inicialmente. Ha pasado satisfactoriamente por las pruebas y está listo para ser utilizado.

Y para que así conste, se firma la presente a los 29 días del mes de mayo de 2014.

Ing. Amaury Viera Hernández
Jefe del departamento SIMAYS

Ing. Pablo Soria Acosta
Jefe del proyecto HMAST