



UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS
FACULTAD 2 “TELECOMUNICACIONES Y SEGURIDAD INFORMÁTICA”

TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS

Título: Plataforma de Gestión de Servicios Telemáticos en GNU/Linux.
Módulo DNS v2.0.

Autores:

Alexei Pérez Hurtado

Álvaro Luis Padilla Moya

Tutor:

Ing. Yaily Rodríguez Rodríguez

Ing. Ramón Alexander Anglada Martínez

Ciudad de La Habana, Junio del 2010.

“Año del 50 Aniversario del Triunfo de la Revolución Cubana”

Declaración de Autoría

Declaramos que _____ y _____ somos los únicos autores de este trabajo y autorizamos a la Universidad de las Ciencias Informáticas (UCI) y a la Facultad (2) para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmamos la presente a los ____ días del mes de julio del 2009.

Firma del Autor
<Autor 1>

Firma del Autor
<Autor 2>

Firma del Tutor
<Tutor>



AGRADECIMIENTOS

Alexei Pérez Hurtado

A mi familia, por estar todo el tiempo pendientes de mi, y sobre todo a mis dos grandes tesoros, mi mamá y mi abuela, por apoyarme todo el tiempo, por sacrificarse todos los días y por entregarme todo para verme hoy aquí, por hacerme lo que soy hoy y por el amor que solo ustedes saben darme, porque son lo más grande que tengo, por ser mi motivación y mis guías.

A Pipe por ser mi padre y estar todo el tiempo ahí.

A Daya, mi guapín, por el apoyo diario sin importar la distancia, por las revisiones constantes del documento y por ser tan exigente para que todo saliera bien, eres parte de mi inspiración.

A Yaily por apoyarnos tanto con el documento y tener la paciencia de virarlo tantas veces para atrás lleno de correcciones y por no dejarnos cometer errores para que todo estuviera bien.

A Anglada por más que el tutor, por convertirse en nuestro guía, amigo y líder, por no descansar junto a nosotros y por coger las luchas que muchas veces nos faltaron, gracias por haber sido tan exigente, y por querer siempre que nos graduáramos como ingenieros preparados.

A Harry por hacerme polista, hombre y amigo, creo que también eres responsable de que hoy haya llegado aquí y de que tenga el espíritu de siempre seguir adelante y proponerme metas altas.

A los muchachos de mi grupo, a los que comenzamos juntos, los que ya no están y a los que se han incorporados, a todos muchas gracias por haberlos conocidos y pasar junto a ustedes 5 años de vida universitaria.

A Andrés, Alejandro, Pancho, Daniel, Raymon por haber sabido compartir la casa uci y siempre entendernos, por la ayuda cuando hizo falta.

A Yanetsy Y Osmany, ustedes son muy importantes para mí, gracias por los consejos, las enseñanzas y el apoyo para seguir adelante, espero no haberlos defraudado nunca.

A Javier, el loco, por venir juntos desde hace mucho tiempo y por tenernos el uno al otro siempre, sin importar desde donde para apoyarnos.

A los trigueños Yamil, Rainer, Hussey, Jorge, a Veitia, Erik, la gente del Condado, Jorge Félix, sin dudas lo mejor de la facultad.

Al piquete de la FEU, Ricardo, Baby, Daima, Nilda, Yasiel, Sachie, Marlen, entre ustedes me hice mejor persona, más humano, aprendí a hacer mucho con poco y a saber que "el no se puede" no existe.

A los muchachos de Servicios Telemáticos, por ser una gran familia.

A Alvaro, mi gran compañero de tesis, por llegar a ser más que eso, por confiar en mí, creo que hoy me hago ingeniero y gano un amigo para toda la vida al mismo tiempo.

Alvaro L Padilla Moya

Quisiera comenzar mis agradecimientos primeramente con mi mama, a la cual le debo todo esto y mucho mas, pues siempre fue la que me dio el impulso y me llevo por el camino correcto para llegar hasta aquí. Agradecer también mucho a mi papa que también siempre me ha apoyado en todo, decir que los dos dieron mucho pero mucho de ellos para que yo pudiera lograr este sueño. Otra de las personas que no se puede pasar por alto es mi abuela que tanto batallo conmigo y que para ella todavía yo soy el niño lindo. Otra personita que no puede falta es mi novia Yilian que siempre estaba en los momentos más difíciles de la tesis. Por otra parte agradecerle a toda mi familia no hago referencia de cada uno pues serian interminables estos agradecimientos. Saliendo de la familia pudiera empezar por mi tutor Ramón Alexander Anglada Martínez, que gracias a todo el trabajo que realizo hemos llegado a graduarnos, decir que es muy jodedor e inconforme, pero estas cualidades nos ayudaron para terminar en tiempo, aunque en ocasiones nos poníamos bravos por los puntos de vista de cada cual pero todo marchó bien. A mi tutora Yaili Rodríguez Rodríguez que por poco me vuelve loco con el documento, nunca está conforme siempre quiere más. A mi compañero de tesis Alexei Pérez Hurtado por hacer llegado a ser más que un compañero de tesis. Agradecer también a otras muchas personas como Denys Buedo Hidalgo, Yoannis Álvarez Chacón, Deivis Ricardo Álvarez Mendoza, Pedro A Miralles Lorenzo, Alberto Carlos Carbonell Marcé, Daymel Bonne Solís. Lázaro Gálvez, por favor los que me falten por mencionar aquí les pido encarecidamente que no se me pongan bravos que soy una persona y se me pueden olvidar algunas personas no porque sean menos importantes. Y por ultimo y no menos importante a toda le gente de mi grupo no voy a mencionar nombres porque si no me meto en problemas si se me que alguien.

DEDICATORIA

Alexei Pérez Hurtado

A mi familia, especialmente a mi mamá y mi abuela, pues son las verdaderas responsables que hoy me encuentre donde estoy y que tenga los resultados logrados, por estar en las buenas y las malas.

A todas las personas que nunca han dudado que podía llegar hasta aquí.

Alvaro L Padilla Moya

Le dedico este trabajo de diploma a toda mi familia en particular a mi mamá, a mi papá y a mi abuela, a ellos pues son los que siempre han estado presentes en todas las etapas de mi vida y han hecho posible que llegue hasta aquí. A mis tutores que supieron guiarme en todo el proceso de desarrollo del trabajo y a todos los que en algún momento me ayudaron para formarme como un profesional.

RESUMEN

El software privativo y las ganancias que aportan a sus grandes empresas han marcado el hito desde el surgimiento de la industria del software, situación que ha venido cambiando con el incremento del software libre y las ventajas que trae consigo el conocimiento de una serie de recursos que brindan a la hora de desarrollar y trabajar.

Cuba está totalmente comprometida y llena de restricciones en el área del software, ya que la mayoría del software que se emplea en las empresas, instituciones, organismos y centros, es propietario. Por ello, surge la idea de buscar alternativas a estas dependencias, tomando como comienzo la migración de los servicios telemáticos.

A la hora de iniciar este complicado proceso nos encontramos el punto de que el conocimiento sobre el tema y el personal encargado sabe muy poco o nada de trabajar sobre un sistema operativo GNU/Linux.

El objetivo fundamental del presente trabajo es darle solución a uno de los problemas planteados anteriormente, mediante el desarrollo de una aplicación de escritorio que permita gestionar el Servidor de Nombre de Dominios(DNS) en la red de una forma intuitiva y agradable sobre un sistema operativo GNU/Linux.

La aplicación será capaz de facilitar el proceso de configurar y administrar el Servidor de Nombre de Dominios(DNS) para el administrador mediante una interfaz cómoda y fácil de utilizar, escalable y con un nivel de detalle adecuado.

Índice

Índice.....	8
Capítulo 1 Fundamentación Teórica	13
1.1 Introducción	13
1.2 Soluciones existentes en el mundo	13
1.3 Metodologías de desarrollo	18
1.4 Lenguajes	20
1.5 Herramientas	22
1.6 Conclusiones	23
Capítulo 2 Características del sistema.....	24
2.1 Introducción	24
2.2 Modelo de negocio.....	24
2.2.1 Representación de los procesos con IDEF0.....	24
2.2.2 Descripción detallada de los procesos de negocio.....	27
2.3 Especificación de los requisitos de software	28
2.3.1 Requerimientos funcionales	28
2.3.2 Requerimientos funcionales del módulo DNS	28
2.3.3 Requerimientos no funcionales.....	30
2.4 Modelo de Casos de Uso del Sistema.....	33
2.4.1 Definición de los actores del sistema a automatizar.....	33
2.4.2 Diagrama de Paquetes.....	33
2.4.3 Diagrama de Casos de Uso del Sistema	34
2.4.4 Casos de Uso del Sistema (CUS).....	35
2.5 Conclusiones.	41
Capítulo 3 Diseño del Sistema.....	43
3.1 Introducción.	43
3.2 Diseño.....	43
3.2.1 Estructura detallada del Sistema	45
3.3 Diagrama de clases del diseño.	46

3.4	Conclusiones	50
Capítulo 4	Implementación	51
4.1	Introducción.	51
4.2	Diagrama de despliegue	51
4.3	Diagrama de componentes	51
4.4	Conclusiones	58
Conclusiones	59
Recomendaciones	60
Bibliografía	61
Anexos	62
Anexo 1	Diagramas de Clase del Diseño	62
Glosario de Términos	71

Introducción

Hoy en día la administración de las redes ha venido a ocupar un punto importante en cualquier tipo de organización, ya que los avances tecnológicos actuales han permitido a las empresas que cuentan con grandes redes de datos, tengan procedimientos establecidos para la administración efectiva de este tipo de redes. El administrador de red como pieza clave en el proceso de administración tiene como tareas fundamentales servir a los usuarios, garantizar el correcto funcionamiento de todos los servicios que funcionan en su sistema, el acceso y la seguridad de las prestaciones que brinda.

La administración de redes como conjunto de técnicas y habilidades debe estar enfocada en asegurar que el administrador cuente con los recursos y la experiencia necesaria para poder diseñar, instalar y brindar soporte a las soluciones de red, teniendo en cuenta que estos sistemas se han vuelto cada vez más complejos y críticos con la incorporación de novedosas tecnologías. La finalidad de una red y de cualquier administrador, es brindar a los usuarios servicios de correo, mensajería instantánea, obtener información e imprimir documentos, entre otros; mediante un conjunto de servidores básicos que garantizan el funcionamiento de cualquier red. Algunos de ellos son:

- Servidor DHCP (del inglés, Dynamic Host Configuration Protocol) para la asignación dinámica de direcciones IP a las máquinas.
- Servidor LDAP (del inglés, Lightweight Directory Access Protocol) para proveer un acceso rápido a la información.
- Servidor FTP (del inglés, File Transfer Protocol), para la transferencia de archivos.

No es posible dejar de mencionar otro servicio fundamental: el Sistema de Nombres de Dominio (en lo adelante DNS), que permite a los usuarios de una red utilizar nombres descriptivos para localizar fácilmente ordenadores y otros recursos en la red, evitando de esta manera tener que recordar la dirección IP de cada ordenador al que se desea acceder.

A diferencia de hace algunos años la administración de las redes se llevaba en forma manual, ahora por el contrario, los administradores de redes cuentan con diferentes alternativas de software, por lo que la diferencia y complejidad del trabajo del administrador puede estar influida por diversos factores como: el sistema operativo que usa y las herramientas seleccionadas, sobre todo si la naturaleza de las mismas es propietaria o libre. En ese sentido, muchas corporaciones en el mundo han guiado sus esfuerzos al desarrollo de un sistema que permita administrar los servicios básicos de una red. Microsoft es una de las empresas que presenta mayores avances en el tema y como resultado ha incorporado al mercado varios Sistemas Operativos Windows Server, que incluyen un apreciable conjunto de tecnologías, características y servicios diseñados para

facilitar en gran medida el trabajo de los administradores de red. Sin embargo, muchos países del tercer mundo como Cuba no tienen posibilidades de adquirir estos productos debido a los elevados costos de las licencias y restricciones de uso.

En Cuba diversas entidades entre ellas: El Ministerio de la Informática y las Comunicaciones (MIC) en conjunto con la Universidad de las Ciencias Informáticas (en lo adelante, UCI) tienen la tarea de impulsar la migración del software privativo al software libre; y como parte fundamental, sus servicios de red. Por esta razón, la UCI partiendo de un trabajo de diploma realizado en el curso 2008 – 2009, comenzó a desarrollar una aplicación de escritorio que permite gestionar algunas funcionalidades de un DNS, pero este software carece de las siguientes operaciones:

- Importar y exportar las configuraciones de un servidor DNS.
- Integración del servidor DNS con el servidor DHCP.
- Integración del servidor DNS con una base de datos externa donde residan las configuraciones del servidor.
- Gestión de las llaves del RNDG con el objetivo de proveer seguridad al DNS.
- Integración con servidores máster y esclavos.

En ese sentido, partiendo de las necesidades y las características propias de los administradores de red cubanos se desea incorporar estas funcionalidades en un software cubano, con sello propio, que pueda ser comercializado en un futuro como un producto de calidad al nivel de cualquier software en el mundo.

Con el propósito de resolver la problemática existente se plantea el siguiente **problema científico**: ¿Cómo facilitar la gestión del servicio de DNS en sistemas operativos GNU/Linux? En vista a resolver el problema planteado, el **objeto de estudio** de la siguiente investigación son los procesos que garantizan la gestión de servicios telemáticos en redes, enfocado en **campo de acción**: procesos que garantizan la administración del servicio de DNS en GNU/Linux.

Se define como **objetivo general**: Implementar las nuevas funcionalidades al Módulo DNS v1.0 para la Plataforma de Gestión de Servicios Telemáticos en GNU/Linux.

Para dar cumplimiento al objetivo general, se llevaron a cabo las siguientes tareas de la investigación:

- Realización de un estudio de la Plataforma de Gestión de Servicios Telemáticos del Módulo DNS en su primera versión para identificar mejoras en la aplicación.
- Estudio de las metodologías y herramientas informáticas que existen para utilizar la mejor propuesta en la construcción del software.

- Análisis de información sobre el tema DNS, para identificar las nuevas funcionalidades y características que va a tener la aplicación.
- Integración de un servidor DNS con un servidor DHCP en GNU/Linux para comprobar el funcionamiento y sus principales características.
- Descripción de las nuevas funcionalidades que va a tener el producto final.
- Desarrollo de las nuevas funcionalidades para el Módulo DNS v1.0.

Para una mejor organización y un mejor entendimiento del documento se encuentra estructurado el contenido con una breve explicación en 5 capítulos:

Capítulo 1, “Fundamentación teórica”, incluye un estado del arte de las soluciones que existen en el mundo de la problemática abordada, además se definen técnicas, metodologías y herramientas a utilizar para un mejor desarrollo del software.

Capítulo 2, “Características del Sistema”, se define una propuesta del sistema, especificándose los requisitos de la herramienta, los cuales ayudan a desarrollar los casos de uso basándose en la metodología

Capítulo 3, “Diseño del Sistema”, se define cómo continúa el desarrollo del sistema llevándose a cabo el diseño del mismo.

Capítulo 4, “Implementación”, se llevan a cabo el desarrollo del diagrama de despliegue y los diagramas de componentes.

Capítulo 1 Fundamentación Teórica

1.1 Introducción

El presente capítulo recoge un análisis crítico referente a las tendencias y técnicas actuales para el desarrollo del tema y la solución tratados, tanto en Cuba como en el resto del mundo. Se hace especial énfasis en lograr aprender las buenas prácticas que caracterizan a las soluciones líderes del mercado actual. Además se reflejarán las tecnologías y metodologías propuestas el desarrollo de la aplicación de gestión de servicios telemáticos en GNU/Linux. Módulo DNS v 2.0.

1.2 Soluciones existentes en el mundo

En esta sección se abordarán las soluciones informáticas que existen en el mundo para realizar la gestión de los servidores de red y que se utilizan en las redes del país. El desarrollo de estos sistemas ha estado asociado principalmente a sistemas operativos, aunque existen soluciones independientes muy competitivas que se abordarán también con el objetivo de estudiar sus funcionalidades y características para el mejoramiento del propio producto.

Microsoft Windows NT Server

Microsoft Windows NT Server es un sistema operativo diseñado para su uso en servidores de red de área local (LAN). Ofrece la potencia, la manejabilidad y la capacidad de ampliación de Windows NT en una plataforma de servidor e incluye características, entre las cuales se pueden destacar la administración centralizada de la seguridad y la tolerancia a fallos más avanzada, las cuales hacen de él un sistema operativo idóneo para servidores de red, además de la facilidad de su uso y la posibilidad de administración de sus funciones e incorporación de procesadores adicionales cuando se desee aumentar el rendimiento.

Entre sus prestaciones incluye los siguientes servicios:

- Capacidad de controlador de dominio.
- Perfiles de usuarios centralizados.
- Replicación de directorios.
- Servicio de Acceso Remoto (RAS).
- Servicios para Macintosh.
- Dynamic Host Configuration Protocol (DHCP) y Windows Internet Named Services (WINS).

En grandes entornos de redes, Windows NT Server da soporte a los siguientes entornos de computación:

- Servidor de bases de datos.

- Servidor de mensajería.
- Servidor de archivos y de impresión.
- Servidor de comunicaciones.
- Servidor WEB.
- Soporte a múltiples plataformas (Intel 80386, 80486, Pentium y procesadores futuros, PowerPC, MIPS, DEC Alpha AXP, Computadoras con simple o múltiple procesador (SMP))¹

Microsoft Windows Server 2003²

Windows Server 2003 es un sistema operativo para servidores que fue presentado al público por primera vez en el año 2003. Está basada en tecnología NT y brinda la capacidad de manejar la gran gama de funciones de servidor que posee, en base a sus necesidades, tanto de manera centralizada como distribuida. Entre las principales funcionalidades del servidor se encuentran:

- Servidor de archivos e impresión.
- Servidor Web y aplicaciones Web.
- Servidor de correo.
- Terminal Server.
- Servidor de acceso remoto/Red privada virtual (VPN).
- Servicio de directorio, Sistema de dominio (DNS), y servidor DHCP.
- Servidor de transmisión de multimedia en tiempo real (Streaming).
- Servidor de infraestructura para aplicaciones de negocios en línea (tales como planificación de recursos de una empresa y software de administración de relaciones con el cliente).

Microsoft Windows Server 2003 ofrece más rapidez, fiabilidad, escalabilidad y disponibilidad que Microsoft Windows NT Server.

- Permite a los clientes ser más productivos.
- Está construido sobre la robustez y fiabilidad de Microsoft Windows 2000 Server.
- Es un Sistema Operativo Windows muy rápido y fiable.

Ventajas con versiones anteriores³

¹ **Gualdarrama Reyes, Jorge Luis.** *Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo DNS.* Ciudad La Habana: s.n., 2009. p. 5.

² **Gualdarrama Reyes, Jorge Luis.** *Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo DNS.* Ciudad La Habana: s.n., 2009. p. 5.

³ **Microsoft.com,** *Presentación de la familia Windows Server 2003,* 3 Marzo 2010. <http://www.microsoft.com/latam/windowsserver2003/evaluation/overview/family.mspx>

- Como servidor de ficheros es de un 100% a un 139% más rápido que Windows 2000 Server y un 200% más que Windows NT Server 4.0.
- Como servidor de impresión, es un 135% más eficiente que Windows NT Server 4.0. Como servidor web es de un 100% a un 165% más rápido que Windows 2000 Server.
- Las características mejoradas del Directorio Activo permiten realizar tareas más fácilmente, entre las que destacan la habilidad de renombrar dominios, la posibilidad de redefinir el esquema y una replicación más eficiente.
- Ofrece la mejor conectividad, facilitando al máximo la configuración de enlaces entre delegaciones, acceso inalámbrico seguro y acceso remoto a aplicaciones a través de los Terminal Services, así como en su integración mejorada con dispositivos y aplicaciones.

Aunque estas plataformas son idóneas para el trabajo con redes por la amplia gama de servicios que ofrece y la comodidad con que se administran, son muy costosas y se necesita de una gran inversión de capital para el mantenimiento de sus actualizaciones, lo cual es una opción desechable para países como Cuba, que poseen una economía de escasos ingresos. Además poseen la desventaja de imponer muchas restricciones al comprador, entre ellas están la de no proporcionar su código fuente para posibles mejoras y correcciones del software.

SuSE Linux Enterprise Server.

SuSE Linux Enterprise Server es uno de los sistemas operativos para servidores líderes en el campo de la implementación profesional para entornos informáticos heterogéneos de cualquier tamaño y estructura. Está disponible para las principales plataformas de hardware: desde procesadores AMD e Intel de 32 y 64 bits hasta mainframes, pasando por toda la gama eServer de IBM. Un único sistema operativo para servidor con una base de código Linux uniforme.

SuSE Linux Enterprise Server incorpora todos los servicios de servidor y redes relevantes para la empresa: desde servicios de ficheros, impresión, web y seguridad, hasta soluciones de aplicaciones y middleware como son bases de datos, gestión empresarial integrada (ERP) y almacenamiento. El producto integra también soporte para soluciones informáticas de alta disponibilidad y alto rendimiento.

Ventajas

- **Reducción masiva de costes:** SuSE Linux Enterprise Server elimina los costes derivados de las licencias de acceso de clientes o de acceso a Internet. Asimismo, gracias a la consolidación de servidores evitará recursos y cursos de formación innecesarios.
- **Seguridad de inversión gracias al mantenimiento del sistema y al código fuente abierto:** Los servicios de soporte, actualizaciones y mantenimiento del software se han unificado en el SuSE Linux Maintenance Programm. Esto le permite beneficiarse de un

servicio completo procedente de una sola fuente y, por ejemplo, acordar el mantenimiento "perpetuo" de sus sistemas. También proporcionan el código fuente del sistema operativo.

- **Escalabilidad y estabilidad:** Para que implemente SuSE Linux Enterprise Server donde y como quiera: el producto incluye equilibrio de red, alta disponibilidad y posibilidad de clustering, y soporta 32 CPUs, 64 GB RAM y 600 discos duros en hardware basado en x86.
- **Administración eficaz:** Mantenimiento y administración remotos de forma segura, independientemente de donde se encuentren los ordenadores. SuSE Linux Enterprise Server permite automatizar los procesos de administración e incorpora para ello todas las herramientas necesarias.

A pesar de las ventajas que ofrece SuSE Linux Enterprise Server no es una buena opción para su implantación en Cuba, por estar desarrollado por una compañía como Novell que a consecuencia del bloqueo económico le imposibilitaría al país recibir asistencia técnica. También esta es una solución no genérica debido a que las funcionalidades que brindan están estrechamente ligadas al uso de este tipo de distribución Linux.⁴

Red Hat Enterprise Linux 5 Server.

Red Hat Enterprise Linux está disponible en dos versiones para servidores. La versión básica Red Hat Enterprise Linux Server está diseñada para implantaciones en redes de pequeño tamaño, mientras que Red Hat Enterprise Linux Advanced Platform está diseñada para escenarios más grandes, proporcionándoles el entorno más rentable, flexible y escalable. Ambas versiones comparten la misma base tecnológica e incorporan un amplio conjunto de aplicaciones de servidor de código abierto y prestaciones de virtualización.

Red Hat Enterprise Linux es compatible con las siguientes arquitecturas de sistemas:

- Intel y AMD x86/x86-64
- Intel Itanium2
- IBM POWER
- IBM z-Series y S/390

Proporciona un conjunto de aplicaciones para servicios de red, como: DHCP, DNS, cortafuegos, así como servicios de servidor de archivos y de impresión, con integración de Directorio Activo. En cuanto a seguridad la incorporación de SELinux Troubleshooter, un analizador dotado de una interfaz gráfica de usuario que ayuda a los administradores en la toma de decisiones en caso de alertas de seguridad, facilita considerablemente el uso.

⁴ **Gualdarrama Reyes, Jorge Luis.** *Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo DNS.* Ciudad La Habana: s.n., 2009. p. 8-9.

Este es uno de los sistemas operativos de red más competitivos del mercado mundial, incluso restándole seguidores a los productos desarrollados por la Microsoft, pero debido a que es comercial en sus versiones posteriores a Red Hat Linux 9.0 no debe ser una opción a tener en cuenta por países en vías de desarrollo.⁵

Webmin

Webmin representa una herramienta para la configuración mediante la Web desde GNU/Linux y otros sistemas Unix de las características propias de sistemas operativos, como servicios, usuarios, cuotas de espacio, archivos, entre otros; así como la instalación y control de muchas aplicaciones entre las que destacan: servidores DNS, DHCP, LDAP, Web, Jabber, FTP, Correo, Proxy y SSH. Permite el control de todos los servidores no solo de forma local, sino también de forma remota. Esta solución tiene una interfaz gráfica muy intuitiva y bastante clara, fue creada para la Web, aunque no es necesario un monitor con interfaz gráfica para su funcionamiento, ya que puede ejecutarse en un navegador de consola. Es una aplicación modular, escrita en el lenguaje Perl, lo que brinda la posibilidad de agregarle y quitarle funcionalidades. Su licencia inicial, la BSD o Berkeley Software Distribution (en español, Distribución de Software Berkeley) no es exactamente GPL, aunque Webmin en su versión “free” es de libre distribución, siempre y cuando no se modifique, pues, en ese caso habrá que enviar las modificaciones realizadas a su autor.

Plataforma de Gestión de Servicios Telemáticos. Módulo DNS.

Este producto fue desarrollado en el 2009 en la Universidad de las Ciencias Informáticas con el objetivo de darle solución a varios problemas que tiene Cuba como la adquisición de los productos que salen al mercado y la poca posibilidad de pagar los altos precios de sus licencias, además de la necesidad de desarrollar un software propio con las mismas características que los grandes productores de estos software en el mundo.

La aplicación desarrollada fue programada en Python utilizando todas las ventajas que proporciona este lenguaje, posibilita gestionar todo lo relacionado con las zonas del servidor de nombres de dominio y constituye un gran paso de avance, sin embargo el producto no permite las siguientes funcionalidades:

- Integrar un servidor DNS con un servidor DHCP.
- Integrar con máster y esclavos.
- Integrar el servidor DNS con una base de datos externa donde residan las configuraciones.
- Importar y exportar las configuraciones del servidor DNS.

⁵ **Gualdarrama Reyes, Jorge Luis.** *Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo DNS.* Ciudad La Habana: s.n., 2009. p. 9.

- Gestion de las llaves del RNDC con el objetivo de proveer seguridad al DNS.

1.3 Metodologías de desarrollo

El desarrollo de la producción de software ha traído consigo que los usuarios esperen software mejor elaborados, adaptados a sus necesidades y donde se optimice el tiempo de realización. Sin embargo alcanzar este resultado es un poco difícil y la mayoría de los desarrolladores afrontan la dificultad de coordinar sus esfuerzos e integrar múltiples fases del desarrollo, además de la necesidad de una guía para ordenar las actividades de un equipo, dirigir las tareas de cada desarrollador por separado y del equipo como un todo, especificar los artefactos que deben desarrollarse y ofrecer criterios para el control y la medición de los productos y actividades del proyecto. Para la solución de estos problemas aparecen las metodologías de desarrollo, las cuales brindan la posibilidad de darle seguimiento al desarrollo del software, estableciendo una guía para el control de las actividades desarrolladas durante la realización del producto. Una de estas metodologías es RUP.

RUP

RUP es el resultado de varios años de desarrollo y uso práctico en el que se han unificado técnicas de desarrollo, a través del UML, y trabajo de muchas metodologías utilizadas por los clientes.

Como RUP es un proceso, en su modelación define como sus principales elementos:

- Trabajadores (“quién”)
- Actividades (“cómo”)
- Artefactos (“qué”)
- Flujo de actividades (“Cuándo”)

RUP tiene tres características principales.⁶

- **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo).
- **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo

⁶ James Rumbaugh, Ivar Jacobson, Grady Booch. *El Proceso Unificado del Desarrollo de Software*, 2000.

[Disponible en: <http://biblioteca.uci.cu/titdigitales.htm#igs>]

que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente.

- **Iterativo e Incremental:** Es práctico dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. Cada iteración se realiza de forma planificada es por eso que se dice que son miniproyectos.

IDEF0

El programa ICAM identificaba las necesidades de una mejora en las técnicas y en análisis de la comunicación para personal involucrado en la producción. Entre dichas técnicas se encuentra IDEF0, utilizado para la representación de actividades o procesos.

IDEF0 es una técnica de modelación para representar de manera estructurada y jerárquica las actividades que conforman un sistema o empresa y los objetos o datos que soportan la interacción de esas actividades.

IDEF0 tiene las siguientes características:

- Es comprensivo y expresivo, capaz gráficamente, de representar una variedad amplia de negocio, de fabricación y de otros tipos de operaciones de la empresa a cualquier nivel del detalle.
- Es un lenguaje coherente y simple, que prevé la expresión rigurosa y exacta, y promueve la consistencia del uso y de la interpretación.
- Puede ser generado por una gran variedad de herramientas gráficas en computadores. Muchos productos comerciales apoyan específicamente el desarrollo y el análisis de los diagramas IDEF0 y de sus modelos.

Además, permite analizar, documentar y mejorar los procesos del negocio; empleando para ello las notaciones AS-IS y TO-BE. La forma AS-IS es la empleada por el negocio para representar los procesos a la manera actual, antes de la automatización.

La forma TO-BE es utilizada para transformar el proceso a la manera deseada para los objetivos de la organización, pudiendo identificarse mejoras en la disposición de los procesos. Los elementos gráficos utilizados para la construcción de los diagramas IDEF0 son unos rectángulos o cajas y una serie de flechas. Es importante destacar que esta técnica se utiliza como homóloga del flujo de trabajo Modelamiento del Negocio que propone RUP.

1.4 Lenguajes

UML

Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema. Permite la modelación de sistemas con tecnología orientada a objetos, combinar diferentes elementos gráficos y crear diagramas. Un modelo UML describe lo que supuestamente hará un sistema pero no dice cómo implementarlo. Ayuda a los usuarios a entender la realidad de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades en proyectos que no estén seguros en su desarrollo, reduciendo el coste y el tiempo empleado en la construcción de los productos.

UML está compuesto por Elementos, Diagramas y Relaciones, los principales diagramas de UML están clasificados en:

- Diagramas de estructura estática.
- Diagrama de comportamiento.
- Diagramas de implementación.

Pretende proporcionar a los usuarios un lenguaje de modelado visual expresivo y utilizable para el desarrollo e intercambio de modelos significativos e integrar las mejores prácticas empleadas hasta el momento. No posee propietario y está basado en el común acuerdo de gran parte de la comunidad informática.⁷

UML posee una gran cantidad de propiedades que han sido las que en realidad han contribuido a hacer del mismo, el estándar de facto de la industria que es en realidad, entre las cuales se pueden mencionar:

- Concurrencia, es un lenguaje distribuido y adecuado a las necesidades de conectividad

actuales y futuras.

- Ampliamente utilizado por la industria desde su adopción por OMG.
- Reemplaza a decenas de notaciones empleadas con otros lenguajes.
- Modela estructuras complejas.
- Las estructuras más importantes que soportan tienen su fundamento en las tecnologías

orientadas a objetos, tales como objetos, clase, componentes y nodos.

⁷ **James Rumbaugh, Ivar Jacobson, Grady Booch.** *El Lenguaje Unificado de Modelado. Manual de Referencia*, 1998. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg03050.pdf>]

- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.
- Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboraciones, que sirven para evaluar el estado de las máquinas.

Python

Python es un lenguaje de programación dinámico muy poderoso que hoy en día se utiliza en un amplio dominio de aplicaciones, creado por Guido Van Rossum en 1991. Es extremadamente sencillo para iniciarse en la programación ya que ofrece una sintaxis extraordinariamente simple.

Algunas de las principales características que tiene son:

- Clara sintaxis de lectura.
- Fuerte capacidad de introspección.
- Orientación a objetos intuitiva.
- Expresión natural del código de procedimiento.
- Modularidad completa, apoyado en paquetes jerárquicos.
- Manejo de errores basado en Excepciones.
- Tipos de datos de alto nivel.
- Extensa biblioteca estándar.
- Extensiones y módulos fácilmente escritos en C, C++ (o Java para Jython, o .NET para IronPython).
- Incrustable dentro de las aplicaciones como una interfaz de secuencias de comandos.

Su lenguaje de alto nivel permite la posibilidad de despreocuparse por detalles de bajo nivel, como manejar la memoria empleada por el programa. El ser interpretado, hace que se ahorre un tiempo considerable en el desarrollo de los programas, pues no es necesario compilarlos. Esto garantiza su portabilidad, ya que el código del programa Python es capaz de reutilizarse en vez de crearse un nuevo código cuando el software pasa de una plataforma a otra, característica que es muy importante para el desarrollo de reducción costos.

Python está disponible para los principales sistemas operativos: Windows, Linux / Unix, OS / 2, Mac, Amiga, entre otros, posee una documentación muy amplia y completa, tanto en la web como en los manuales integrados que posee el propio lenguaje, además tutoriales en línea que se dirigen tanto al programador experimentado como para los novatos, todos bien diseñados para lograr alcanzar dominio de sus ventajas con mucha rapidez.

La aplicación de Python está bajo una licencia de código abierto, que lo hace libremente utilizable y distribuible, incluso para uso comercial. La licencia de Python es administrada por la Python Software Foundation.⁸

1.5 Herramientas

Visual Paradigm

Es una herramienta profesional que soporta el ciclo completo del desarrollo del software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue, así como representar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar abundante documentación.

Es una herramienta que está diseñada para ser utilizada por cualquier usuario que tenga un rol definido en la creación de grandes sistemas de software de manera confiable a través de la Programación Orientada a Objetos, incluyendo Ingenieros de Software, Analistas de Sistemas, Analistas de Negocios y Arquitectos de Sistema.

Entre sus principales características se encuentra:

- Ingeniería inversa y directa.
- Generación de código.
- Disponibilidad de integrarse en los principales IDE (Integrated Development Environment).
- Disponibilidad en múltiples plataformas.
- Generación de código fuente en varios lenguajes de programación.
- Generador de informes para generación de documentación.
- Editor de Detalles de Casos de Uso - Entorno todo-en-uno para la especificación de los detalles de los casos de uso, incluyendo la especificación del modelo general y de las descripciones de los casos de uso.

Visual Paradigm gestiona una rápida construcción de aplicaciones con calidad a un menor costo. Además, soporta UML en su versión 2.1 y se encuentra disponible en múltiples versiones y plataformas.

Easy Eclipse

⁸ Alfonso de la Guarda Reyes. *A Byte of Python*, 2004. [Disponible en: http://dev.laptop.org/~edsiper/byteofpython_spanish/index.html]

Existen IDEs realmente completos como: “EasyEclipse” con capacidad de completamiento de código, ambiente flexible, rápido, entre otras. Constituye un proyecto de software de código abierto el cual provee un IDE precompilado con plug-ins de propósitos especiales para la programación en lenguajes como: Java, PHP, Ruby, Python, C y C++. Precisamente, PyDev es un plug-in que permite programar en Python usando este IDE y es distribuido bajo la Eclipse Public License. EasyEclipse integra editor de código, depurador e intérprete. Al estar escrito en Java es multiplataforma, por lo que no existen problemas con el sistema operativo en el que se instale.⁹

Qt Designer

Se decidió emplear Qt Designer como herramienta necesaria para el desarrollo de formularios y presentaciones gráficas de las aplicaciones. Qt Designer permite acelerar el desarrollo de interfaces de alto rendimiento de forma fácil, generando el código fuente para las mismas, lo que permite al desarrollador ajustarlo a sus necesidades. Inicialmente fue creado por la empresa TROLLTECH para trabajar en varias distribuciones Linux, sin embargo actualmente puede instalarse en otras plataformas como Windows y Mac OsX. Esta herramienta provee características muy poderosas como la previsualización de la interfaz, soporte para widgets y un editor de propiedades con gran variedad de opciones.¹⁰

1.6 Conclusiones

En el capítulo se realizó un estudio de las principales herramientas que existen en Cuba y en el mundo a la vanguardia de la gestión de servidores entre ellos servidores DNS, para aprovechar sus características y sus mayores ventajas en la construcción de la aplicación a desarrollar, buscando que el software tenga lo mejor de cada. También se realizó una justificación de la metodología de desarrollo, los lenguajes de modelado y de programación, así como las herramientas de desarrollo de interfaz gráfica y el IDE, los cuales se emplearán en el desarrollo del sistema.

⁹ **Noa Cabrera, Sandy.** *Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo Directorio.* Ciudad La Habana: s.n., 2009.

¹⁰ **Noa Cabrera, Sandy.** *Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo Directorio.* Ciudad La Habana: s.n., 2009.

Capítulo 2 Características del sistema

2.1 Introducción

En el presente capítulo se hace una descripción de los procesos del negocio asociados a la administración del Servicio de Nombre de Dominio (DNS) en sistemas operativos GNU/Linux. Se utiliza la notación IDEF0 para una mayor comprensión de todos los procesos y la manera en que se desarrollan en la actualidad. Se exponen los requisitos funcionales del sistema, el diagrama de casos de uso del sistema, la descripción de casos de uso del sistema y los prototipos de interfaz de usuario para cada caso de uso crítico.

2.2 Modelo de negocio

2.2.1 Representación de los procesos con IDEF0

Se ha definido un proceso general denominado “Administrar Servidor DNS”, que a su vez está dividido en otros cinco procesos que detallan su funcionamiento:

- Configurar básico servidor.
- Importar configuración.
- Integrar Servicios.
- Exportar datos de configuración.
- Iniciar servicio.

El proceso “Integrar Servicios”, está dividido a su vez en otros tres procesos que detallan su funcionamiento:

- Integrar DNS y DHCP.
- Integrar servidores máster y esclavos.
- Integrar BD.

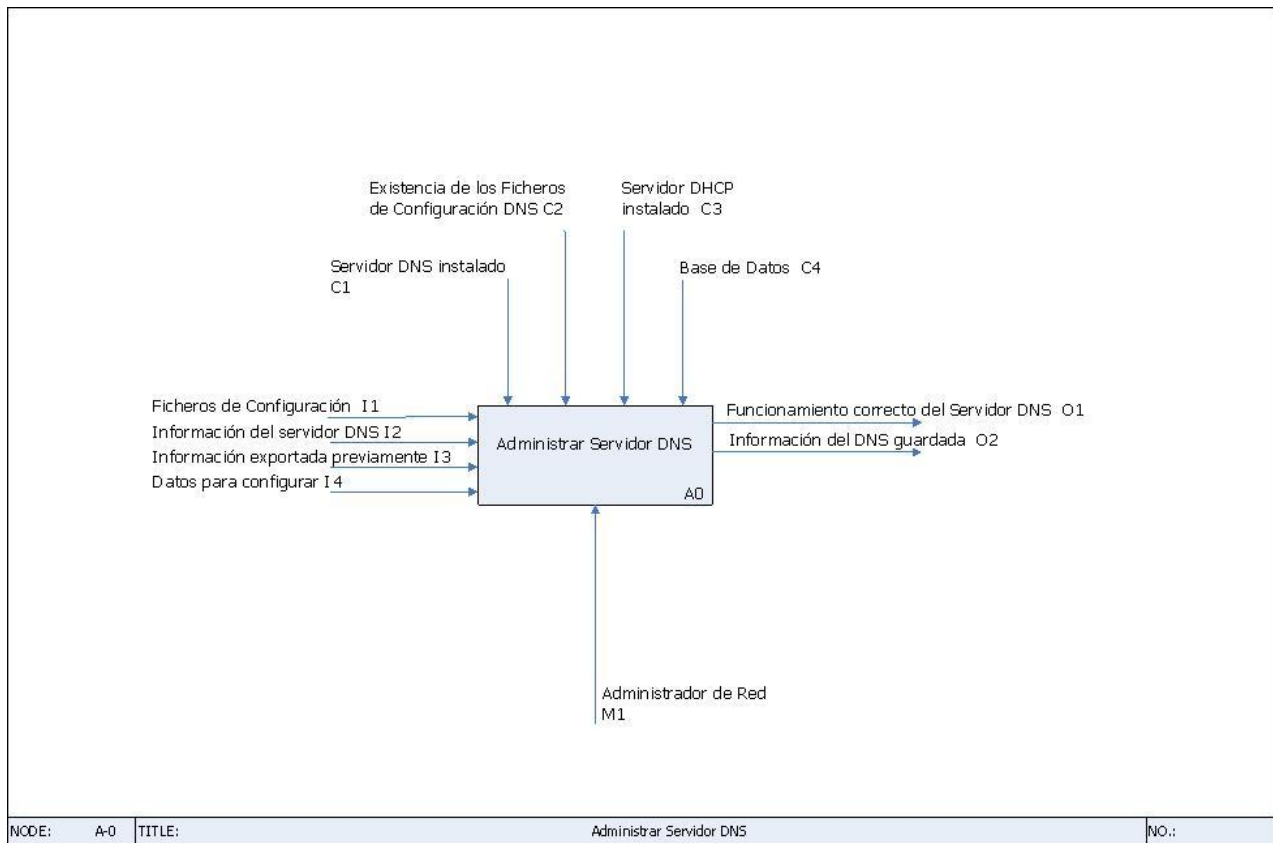


Figura 1 Proceso de Negocio Administrar Servidor DNS.

- Proceso Administrar Servidor DNS: Constituye el proceso base que es realizado por el administrador de red, el cual necesita los ficheros de configuración del servidor DNS ya instalado en los cuales se harán cambios, la información del servidor, en caso de que existan información exportada de un servidor DNS, la cual puede servir para cargar una configuración previamente creada y los datos necesarios para la nueva configuración. Para realizar una correcta administración del servidor DNS es necesario tener instalado un servidor de este tipo, un servidor DHCP y una Base de Datos que permitirán en algún momento integrar estos servicios y de esta forma poder brindar en correcto funcionamiento y poder guardar esta configuración de ser necesario.

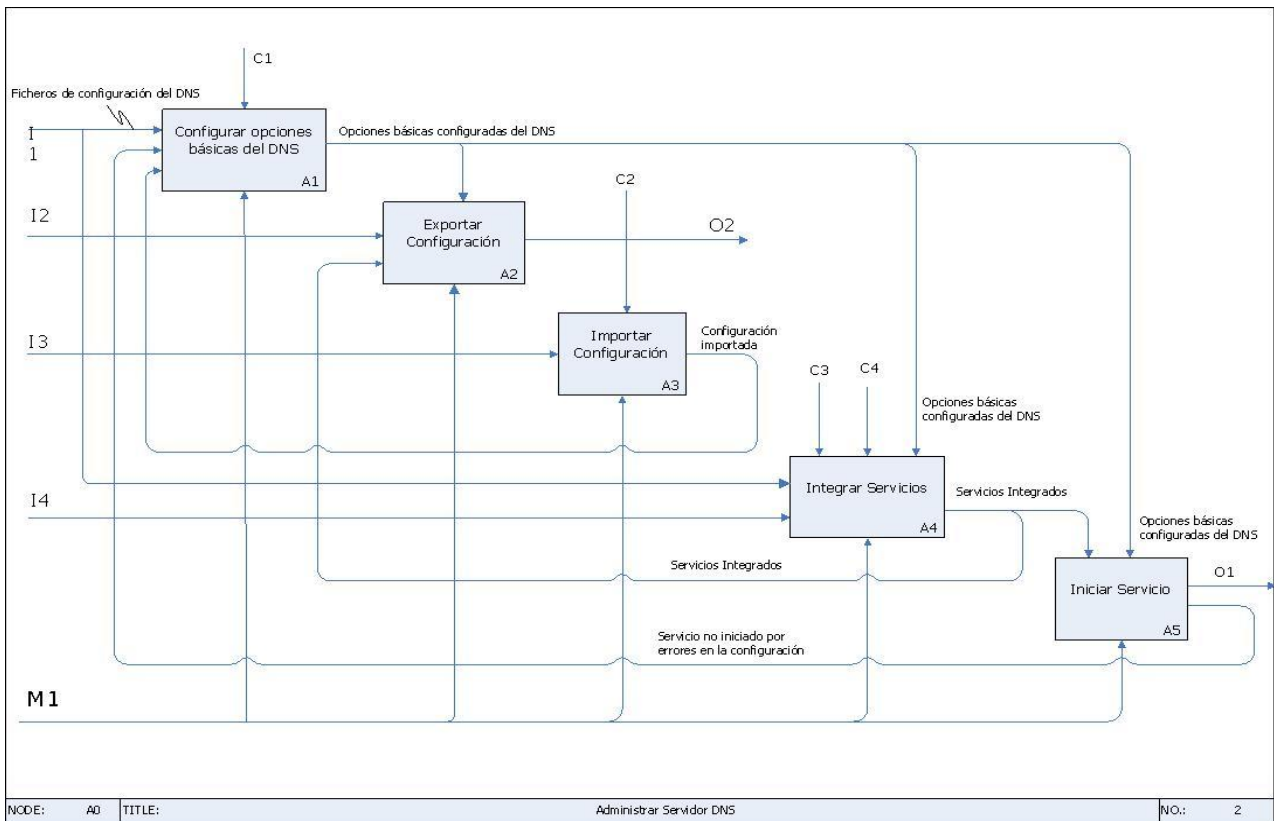


Figura 2 Descripción del Proceso de Negocio Administrar Servidor DNS.

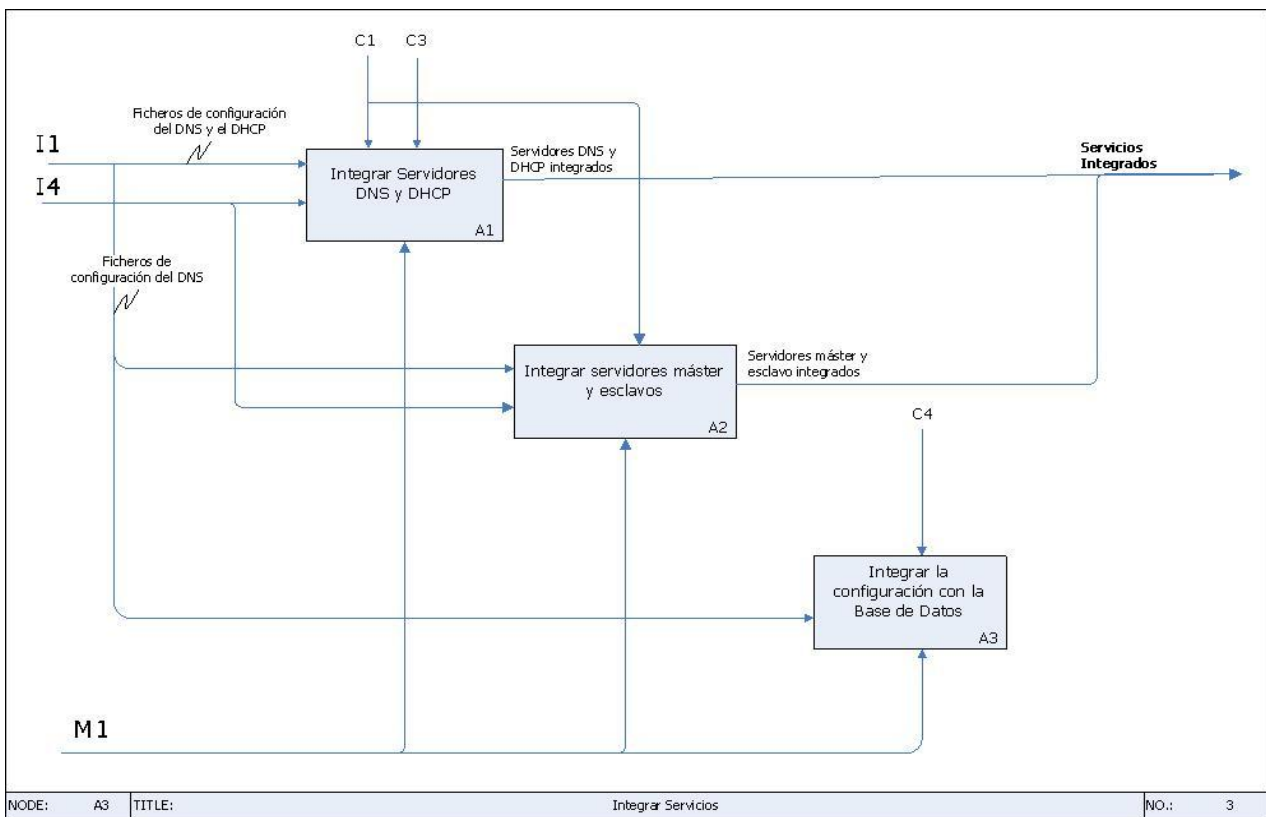


Figura 3 Descripción del Proceso de Negocio Integrar Servicios.

2.2.2 Descripción detallada de los procesos de negocio

- Configurar básico servidor: No es suficiente con tener un DNS instalado, es necesario configurarlo y ajustarlo a las características propias del entorno que se encuentra, esta configuración hace referencia a los ficheros de configuración del servidor como named.conf, named.conf.options, los ficheros de configuración de zonas, entre otros, donde es posible crear zonas, eliminarlas, agregar o eliminar reenviadores y configurar los valores de las zonas; todo esto es posible hacerlo en cualquier momento solo basta con modificar cualquiera de los valores necesarios y reiniciar el servicio.
- Importar configuración: En ocasiones no es necesario comenzar una configuración desde cero ya que previamente se pudo hacer una salva de otra que funcionaba correctamente y esto evita perder ese tiempo, para esto es necesario que existan los ficheros de configuración del servidor DNS y que se conozca la ruta en donde fue guardada por el administrador de red que es la persona de hacer esta tarea.
- Integrar Servicios: Muchas veces no es suficiente con tener instalado y bien configurado el servidor, ya que necesita estar integrado con otros servicios para un correcto funcionamiento de las prestaciones que se brindan, en este caso es necesaria la integración con un servidor DHCP y una base de datos y que exista una configuración de los servicios previamente. El proceso es realizado por el administrador de la red.
- Exportar datos de configuración: La configuración del servidor después que se haya realizado correctamente, conteniendo toda la información de la misma, partiendo de una configuración básica del DNS puede ser exportada, para que pueda ser reutilizada en otra ocasión que ocurra un fallo o simplemente la persona responsable de configurar el servicio no desee comenzar de cero, para esto el administrador de red debe conocer como se realiza el procedimiento y la ruta donde se hará la salva.
- Iniciar servicio: Tras haber ejecutado los procesos anteriores de configuración del servidor, importación y/o exportación de las configuraciones y/o la integración de los servicios debemos iniciarlo. Iniciar un servidor significa tenerlo listo para comenzar a brindar su servicio en la red, basado en las configuraciones realizadas previamente. En caso que el servicio no funcione correctamente, es necesario volver a configurar.
- Integrar DNS y DHCP: Para integrar el servidor DNS con el servidor DHCP es necesario configurar el fichero de configuración del DHCP para que el BIND9 se actualice con la información que brinda a los diferentes host que son usuarios de sus red y a la vez configurar el servidor DNS para actualice su información del DCHP.
- Integrar servidores máster y esclavos: La integración de los servidores másteres y esclavos consiste en modificarle a los diferentes servidores su atributo **type**.

- Integrar BD: Para integrar el servidor BIND9 en este caso con una base de datos PostgreSQL es necesario compilar el código fuente para incluir el driver DLZ y de esta habilitar la integración. En sistemas GNU/Linux el proceso consta de tres pasos: Configuración del proceso de compilación, Compilación y generación de binarios y la Instalación de los ficheros binarios, así como que existan los datos de la configuración, los ficheros de configuración del DNS, que exista la base de datos ya instalada y que el administrador de la red tenga conocimientos de las siguientes variables de configuración – enable-libbind –with-libtool-with-openssl --enable-threads.

2.3 Especificación de los requisitos de software

El modelamiento del negocio brinda una visión general de los procesos que existen en un determinado negocio, permitiendo comprender a qué se dedica el mismo, así como establecer una comunicación entre el cliente y el equipo de desarrollo. Una vez en este punto, se hace necesario comenzar a definir que es lo que debe hacer el sistema, siendo necesario ir a la captura de los requerimientos que este debe cumplir. Debido a ellos todas las ideas que los clientes, usuarios y miembros del equipo de proyecto tengan acerca de lo que debe hacer el sistema, deben ser analizadas como candidatas a requisitos.

Lograr una comunicación efectiva entre los usuarios y el equipo de proyecto con el objetivo de llegar a un entendimiento de lo que hay que hacer, es la clave del éxito en la producción de un software con calidad. Durante años muchas aplicaciones han fallado debido a incongruencias entre lo que el usuario deseaba, lo que realmente necesitaba, lo que interpretaba cada miembro del equipo de proyecto y lo que realmente se obtiene.

2.3.1 Requerimientos funcionales

Los requerimientos funcionales constituyen capacidades o condiciones que el sistema debe cumplir, los mismos se mantienen invariables sin importar con qué propiedad o cualidad se relacionen. A continuación se tratan los requerimientos funcionales asociados al módulo:

2.3.2 Requerimientos funcionales del módulo DNS

A continuación se muestran los requerimientos funcionales del módulo DNS, donde se resaltan en **Negrita** los casos de uso de la v2.0.

R1. Gestionar Zona.

R1.1. Adicionar Zona.

R1.2. Modificar Zona.

R1.3. Eliminar Zona.

R1.4. Mostrar Zona.

R2. Gestionar Reenviador

R2.1. Adicionar Reenviador.

R2.2. Modificar Reenviador.

R2.3. Eliminar Reenviador.

R2.4. Mostrar Reenviador.

R3. Gestionar Vista.

R3.1. Adicionar Vista.

R3.2. Modificar Vista.

R3.3. Eliminar Vista.

R3.4. Mostrar Vista.

R4. Gestionar ACL (Lista de Control de Acceso, del inglés, Access Control List).

R4.1. Adicionar ACL.

R4.2. Modificar ACL.

R4.3. Eliminar ACL.

R4.4. Mostrar ACL.

R5. Gestionar Servidor Externo.

R5.1. Adicionar Servidor Externo.

R5.2. Modificar Servidor Externo.

R5.3. Eliminar Servidor Externo.

R5.4. Mostrar Servidor Externo.

R6. Gestionar Llave.

R6.1. Adicionar Llave.

R6.2. Modificar Llave.

R6.3. Eliminar Llave.

R6.4. Mostrar Llave.

R7. Gestionar Registro.

R7.1. Adicionar Registro.

R7.2. Modificar Registro.

R7.3. Eliminar Registro.

R7.4. Mostrar Registro.

R8. Integrar DNS con DHCP.

R9. Integrar DNS con BD externa.

R10. Importar y exportar configuraciones DNS.

R11. Buscar Zona.

R12. Buscar Reenviador.

R13. Buscar Vista.

R14. Buscar ACL.

R15. Buscar Servidor Externo.

R16. Buscar Llave.

R17. Buscar Registro.

R18. Autenticar

2.3.3 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Representan características que hacen al producto atractivo, rápido, confiable, entre otras. En muchos casos los requerimientos no funcionales son fundamentales en el éxito del producto. Normalmente están vinculados a requerimientos funcionales, es decir una vez se conozca lo que el sistema debe hacer, se puede determinar cómo ha de comportarse y las cualidades que debe tener.

Apariencia o interfaz externa:

La interfaz debe tener un diseño sencillo y agradable, permitiendo la utilización del sistema por cualquier persona sin mucha experiencia. No se debe sobrecargar la aplicación con muchas imágenes y gráficos para acelerar la velocidad de respuesta del sistema, por lo que tendrá iconos sencillos que representen claramente la operación a realizar.

Usabilidad:

Garantizar un acceso fácil y rápido a los usuarios. El sistema podrá ser usado por cualquier persona que posea conocimientos básicos en la administración de servicios telemáticos, se le pedirán los requerimientos mínimos necesarios para gestionar los servidores. La aplicación contará

con un manual para su uso, donde se explicará detalladamente las funcionalidades del software para garantizar la eficiente gestión de los servicios telemáticos.

Rendimiento:

Garantizar tiempos de respuestas generalmente rápidos al igual que la velocidad de procesamiento de la información. La aplicación debe ser capaz de realizar consultas a las base de datos de los servicios utilizados en el menor tiempo posible, para así evitar la saturación de la red.

Soporte:

Se requiere que el sistema cuente con la documentación apropiada para agilizar la instalación, configuración y administración de los servicios telemáticos eficientemente. Se brindarán cursos de capacitación al personal que utilizará el software en caso de ser necesario.

Portabilidad:

Se realizará una versión del software que solo permita instalarse sobre una distribución GNU/Linux, pero que se pueda administrar sobre plataformas Windows y GNU/Linux, garantizando con ello la portabilidad del producto.

Seguridad:

El sistema garantiza la seguridad de la información utilizando el protocolo de conexión SSH, con el cual todas las transferencias se realizan de forma encriptada, usando llaves públicas y privadas.

Políticos-culturales:

Contar con logotipos e imágenes que se encuentren en correspondencia con el carácter científico y profesional del tema.

Legales:

La aplicación se realizará utilizando herramientas de software libre, pero será un producto propio de la UCI que podrá comercializarse en un futuro.

Confiabilidad:

Validar la captación de datos en el caso de los administradores de red al inscribirse para evitar entradas inadecuadas.

Software:

Los ordenadores donde se vaya a ejecutar la aplicación necesitan tener instalado una distribución de GNU/Linux. Además debe poseer los paquetes, librerías y dependencias mínimas necesarias que van a hacer usadas por la aplicación.

Hardware:

Los requisitos de hardware están asociados fundamentalmente al tamaño de la red y a la cantidad de usuarios que recibirán el servicio. Es importante acotar que los servidores debido a su funcionalidad específica de atender las solicitudes de un grupo de clientes necesitan de una estructura de hardware superior que les permita dar cumplimiento a su labor. Si se tratase de una pequeña red, con servicio para algunas decenas de usuarios, se podría emplear:

- Microprocesador Pentium IV.
- 512 MB de memoria RAM.
- 2 GB de espacio libre en el disco duro.

Incluso pudiera operar correctamente en condiciones inferiores; sin embargo, si el tamaño de la red aumentara y se tratase ya de algunos miles de usuarios, el cambio más significativo estaría en aumentar la capacidad de almacenamiento empleando discos duros en RAID como una de las mejores alternativas para las grandes empresas, además debe emplearse un procesador con capacidad de multiprocesamiento. Por otra parte se hace necesario aumentar la capacidad de memoria RAM hasta un rango de 2 a 8 GB.

Restricciones del diseño y la implementación:

Este tipo de requerimiento especifica o restringe la codificación o construcción de nuestro sistema, constituyen restricciones que deben ser cumplidas estrictamente.

- Será empleado Python 2.5 como lenguaje de programación para la construcción del sistema.
- La codificación estará regida por la guía de estilo propuesta por Guido van Rossum creador del lenguaje de programación Python.
- Las herramientas utilizadas están divididas fundamentalmente en dos grupos: herramientas de diseño y herramientas de codificación; en el primer caso se ha seleccionado *Qt Designer* en su versión 4.2.1 para elaborar las interfaces de usuario, en el segundo caso aunque Python permite extender su codificación a los más simples editores de texto, hemos decidido emplear *Eclipse con Pydev*. Pydev es un *plug-in* o complemento para el entorno de desarrollo Eclipse que permite la programación en Python; estos complementos constituyen una forma de expandir programas de forma modular, de manera que se puedan añadir funcionalidades sin afectar a las ya existentes.
- Como arquitectura general del sistema se ha concluido la utilización de una arquitectura en capas debido a las ventajas que ello proporciona en términos de claridad, facilidad en la corrección de errores producidos durante el desarrollo, rápido desarrollo al permitir el trabajo paralelo en cada una de las capas definidas, entre otros factores.

- Sería realmente ineficiente si en cada ocasión que los desarrolladores de sistemas informáticos se proponen una nueva meta deban comenzar la elaboración de su producto desde la base, sobretodo en casos donde se pueden reutilizar muchas funcionalidades. Con el objetivo de resolver la problemática anterior han sido creadas bibliotecas de clases o frameworks que faciliten el desarrollo y acorten los tiempos de producción, permitiendo a los desarrolladores concentrarse en aspectos de mayor nivel. El sistema propuesto se ha apoyado en un framework desarrollado para resolver problemas puntuales asociados a la gestión de servidores.

2.4 Modelo de Casos de Uso del Sistema

2.4.1 Definición de los actores del sistema a automatizar

En el módulo de DNS interviene un actor: el administrador del servidor.

Tabla 1 Descripción de los actores del sistema.

Actores del sistema	Justificación
Administrador de Red	Es el encargado de gestionar el servicio DNS a través de la aplicación. Tiene permisos para instalar el servidor, realizarle modificaciones a las configuraciones y modificar su estado.

2.4.2 Diagrama de Paquetes

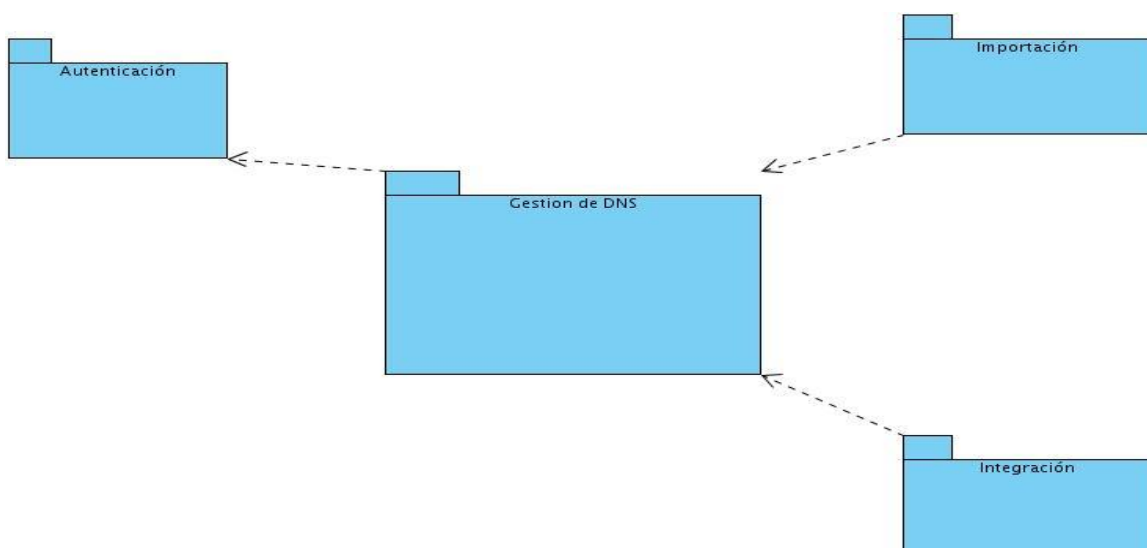


Figura 4 Diagrama de Paquetes del Sistema.

2.4.3 Diagrama de Casos de Uso del Sistema

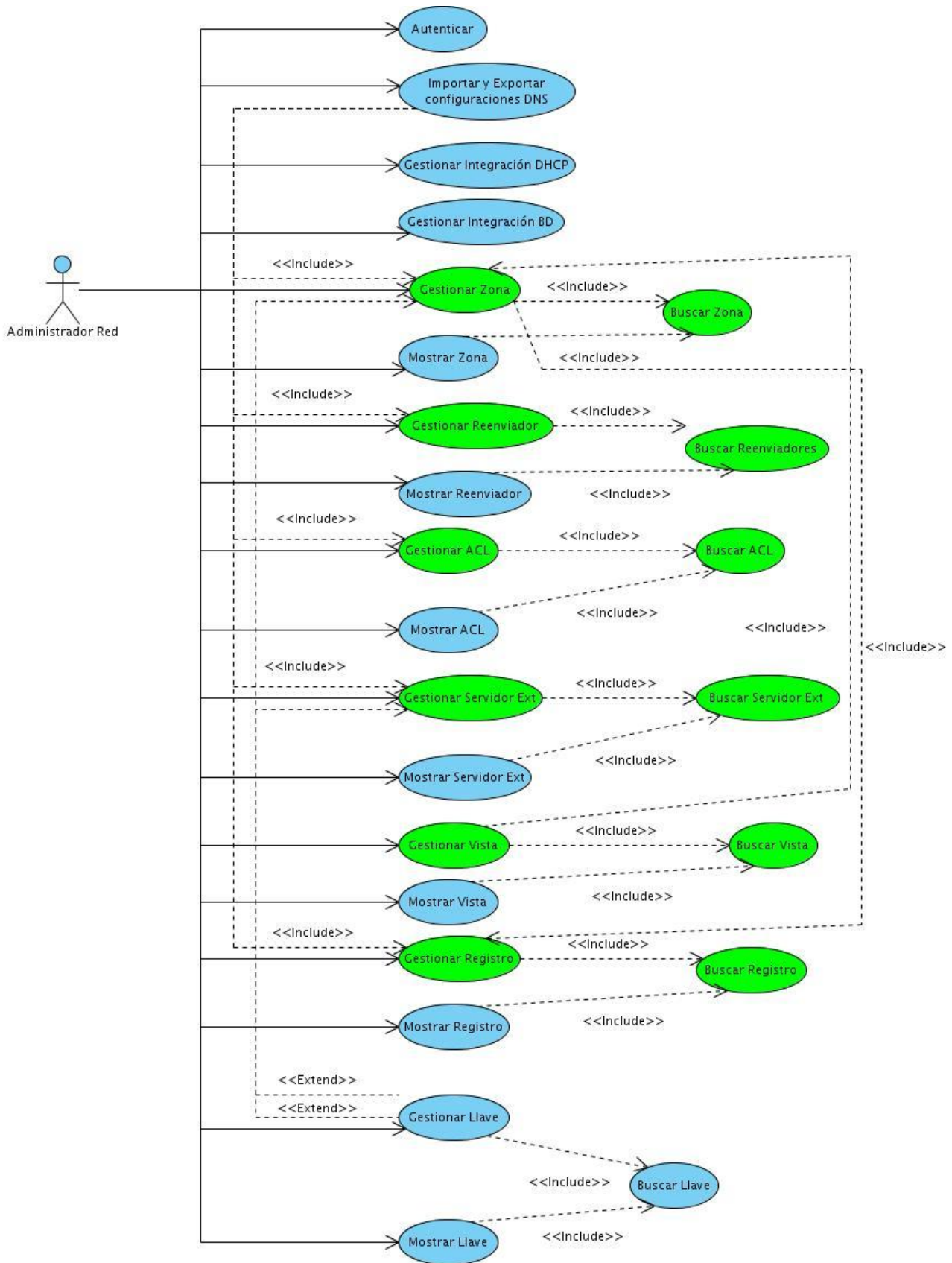


Figura 5 Diagrama de Casos de Uso del Sistema.

2.4.4 Casos de Uso del Sistema (CUS)

Los casos de uso son artefactos narrativos que describen, bajo la forma de acciones y reacciones, el comportamiento del sistema desde el punto de vista del usuario. Por lo tanto, establece un acuerdo entre clientes y desarrolladores sobre las condiciones y posibilidades (requisitos) que debe cumplir el sistema.

2.4.4.1 CUS del módulo DNS

A continuación se muestran los CUS del módulo DNS, donde se resaltan en Negrita los casos de uso de la v2.0.

1. Gestionar Zona
 - 1.1. Mostrar Zona**
2. Gestionar Reenviador
 - 2.1. Mostrar Reenviador**
3. Gestionar Vista
 - 3.1. Mostrar Vista**
4. Gestionar ACL
 - 4.1. Mostrar ACL**
5. **Gestionar Servidor Externo**
6. **Gestionar Llave**
7. Gestionar Registro
 - 7.1. Mostrar Registro**
8. **Integrar DNS con DHCP**
9. **Integrar DNS con BD externa**
10. **Importar y exportar configuraciones DNS**
11. **Buscar Zona**
12. **Buscar Reenviador**
13. **Buscar Vista**
14. **Buscar ACL**
15. **Buscar Servidor Externo**
16. **Buscar Llave**
17. **Buscar Registro**
18. Autenticar

2.4.4.2 Descripción de Casos de Uso del Sistema

Tabla 2 Descripción de Casos de Uso del Sistema. CU Gestionar Servidores Externos

CU 5	Gestionar Servidores Externos
Actor	Administrador
Descripción	Este caso de uso permite al administrador agregar nuevos servidores externos al servidor. Podrá además modificar o eliminar dichos servidores externos en caso que lo necesite. El sistema ofrece la posibilidad de listar todos los servidores externos y buscarlos en caso que sea necesario.
Referencia	R15

Tabla 3 Descripción de Casos de Uso del Sistema. CU Gestionar Llave

CU 6	Gestionar Llave
Actor	Administrador
Descripción	Este caso de uso permite al administrador agregar nuevas llaves al servidor. Podrá además modificar o eliminar dichas llaves en caso que lo necesite. El sistema ofrece la posibilidad de listar todas las llaves y buscarlas en caso que sea necesario.
Referencia	R16

Tabla 4 Descripción de Casos de Uso del Sistema. CU Integrar DNS con DHCP

CU 8	Integrar DNS con DHCP
Actor	Administrador
Descripción	El caso de uso permite al administrador integrar el servicio de nombres de dominio con el servicio de configuración dinámica de computadores para que actualice la información de las zonas en el servicio de nombres de dominio.
Referencia	R11, R1

Tabla 5 Descripción de Casos de Uso del Sistema. CU Integrar DNS con BD externa

CU 9	Integrar DNS con BD externa
Actor	Administrador
Descripción	El caso de uso permite al administrador integrar el servicio de nombres de dominio con una base de datos externa y guardar todas las configuraciones de las zonas, registros, servidores externos, vistas, reenviadores, llaves en registros de una base de datos postgresQL.
Referencia	R11, R12, R13, R14, R15, R16, R17

Tabla 6 Descripción de Casos de Uso del Sistema. CU Importar y exportar configuraciones DNS

CU 10	Importar y exportar configuraciones DNS
Actor	Administrador
Descripción	El caso de uso permite al administrador importar y exportar todas las configuraciones que residan en un Windows Server 2003 y almacenar todas estas configuraciones en una base de datos postgresQL.
Referencia	R1, R2, R3, R4, R5, R6, R7

Tabla 2 Descripción de Casos de Uso del Sistema. CU Buscar Zona

CU 11	Buscar Zona
Actor	Administrador
Descripción	Este caso de uso permite al administrador la posibilidad de buscar zonas del servidor y apoyar la gestión de las mismas.
Referencia	

Tabla 8 Descripción de Casos de Uso del Sistema. CU Buscar Reenviador

CU 12	Buscar Reenviador
Actor	Administrador
Descripción	Este caso de uso permite al administrador la posibilidad de buscar reenviadores del servidor y apoyar la gestión de los mismos.
Referencia	

Tabla 9 Descripción de Casos de Uso del Sistema. CU Buscar Vista

CU 13	Buscar Vista
Actor	Administrador
Descripción	Este caso de uso permite al administrador la posibilidad de buscar vistas del servidor y apoyar la gestión de las mismas.
Referencia	

Tabla 10 Descripción de Casos de Uso del Sistema. CU Buscar ACL

CU 14	Buscar ACL
Actor	Administrador
Descripción	Este caso de uso permite al administrador la posibilidad de buscar ACL del servidor y apoyar la gestión de las mismas.
Referencia	

Tabla 11 Descripción de Casos de Uso del Sistema. CU Buscar Servidor Externo

CU 15	Buscar Servidor Externo
Actor	Administrador
Descripción	Este caso de uso permite al administrador la posibilidad de buscar servidores externos y apoyar la gestión de los mismos.
Referencia	

Tabla 12 Descripción de Casos de Uso del Sistema. CU Buscar Llave

CU 16	Buscar Llave
Actor	Administrador
Descripción	Este caso de uso permite al administrador la posibilidad de buscar llaves y apoyar la gestión de las mismas.
Referencia	

Tabla 13 Descripción de Casos de Uso del Sistema. CU Buscar Registro

CU 17	Buscar Registro
Actor	Administrador
Descripción	Este caso de uso permite al administrador la posibilidad de buscar registros y apoyar la gestión de los mismos.
Referencia	

Tabla 14 Descripción de Casos de Uso del Sistema. CU Mostrar Zona

CU 1.1	Mostrar Zona
Actor	Administrador
Descripción	Este caso de uso permite al administrador mostrar las zonas que tiene el servidor DNS, funcionalidad que posibilita conocer la zona y ver sus detalles.
Referencia	

Tabla 15 Descripción de Casos de Uso del Sistema. CU Mostrar Reenviador

CU 2.1	Mostrar Reenviador
Actor	Administrador
Descripción	Este caso de uso permite al administrador mostrar los reenviadores que tiene el servidor DNS, funcionalidad que posibilita conocer los reenviadores y ver sus detalles.
Referencia	

Tabla 16 Descripción de Casos de Uso del Sistema. CU Mostrar Vista

CU 3.1	Mostrar Vista
Actor	Administrador
Descripción	Este caso de uso permite al administrador mostrar las vistas que tiene el servidor DNS, funcionalidad que posibilita conocer las vistas y ver sus detalles.
Referencia	

Tabla 17 Descripción de Casos de Uso del Sistema. CU Mostrar ACL

CU 4.1	Mostrar ACL
Actor	Administrador
Descripción	Este caso de uso permite al administrador mostrar las ACL que tiene el servidor DNS, funcionalidad que posibilita conocer las ACL y ver sus detalles.
Referencia	

Tabla 18 Descripción de Casos de Uso del Sistema. CU Mostrar Registro

CU 5.1	Mostrar Registro
Actor	Administrador
Descripción	Este caso de uso permite al administrador mostrar los registros que tiene el servidor DNS, funcionalidad que posibilita conocer los registros y ver sus detalles.
Referencia	

2.5 Conclusiones.

En el presente capítulo se realizó un estudio detallado de los procesos fundamentales del negocio que sirvieron de guía para la correcta identificación de los requisitos del sistema. Posteriormente fueron identificados los casos de uso del sistema, los cuales constituyen una entrada esencial para

el flujo de trabajo de “Análisis y Diseño”. Además, el objeto de estudio, centrado básicamente en el análisis de la situación problemática y el objeto de automatización complementan una visión general del funcionamiento de la aplicación.

Capítulo 3 Diseño del Sistema

3.1 Introducción.

Como uno de los resultados principales del flujo de trabajo de requerimientos se obtiene una vista externa del sistema, reflejando lo que se espera del mismo a través del Diagrama de Casos de Uso. A partir de este punto se hace imprescindible profundizar en los Casos de Uso, detallándolos; de manera que se logre obtener una vista de la estructura interna del sistema, descrita en el lenguaje de los desarrolladores de software. Dicha vista refleja las clases necesarias para dar cumplimiento a las funcionalidades descritas por los Casos de Uso. De acuerdo con la metodología RUP este proceso debe efectuarse fundamentalmente durante la fase de Elaboración asociada al ciclo de vida del producto, correspondiente al flujo de Análisis y Diseño. El presente capítulo tiene como objetivo la realización del Diagrama de Clases del Diseño de cada caso de uso.

3.2 Diseño

El diseño de un sistema está estrechamente ligado a la arquitectura escogida para su desarrollo. Una arquitectura con clara definición de capas que permitan la descentralización del trabajo, con componentes que cumplan funciones específicas asociadas a cada una de ellas; es sin lugar a dudas una alternativa de lujo para elaborar un sistema informático. La arquitectura en capas permite practicar el trabajo en paralelo de los desarrolladores, mejora la detección y corrección de errores, además de ofrecer una mayor transparencia y entendimiento del sistema que se construya. Los siguientes diagramas muestran la estructura general del sistema en términos de capas y componentes:

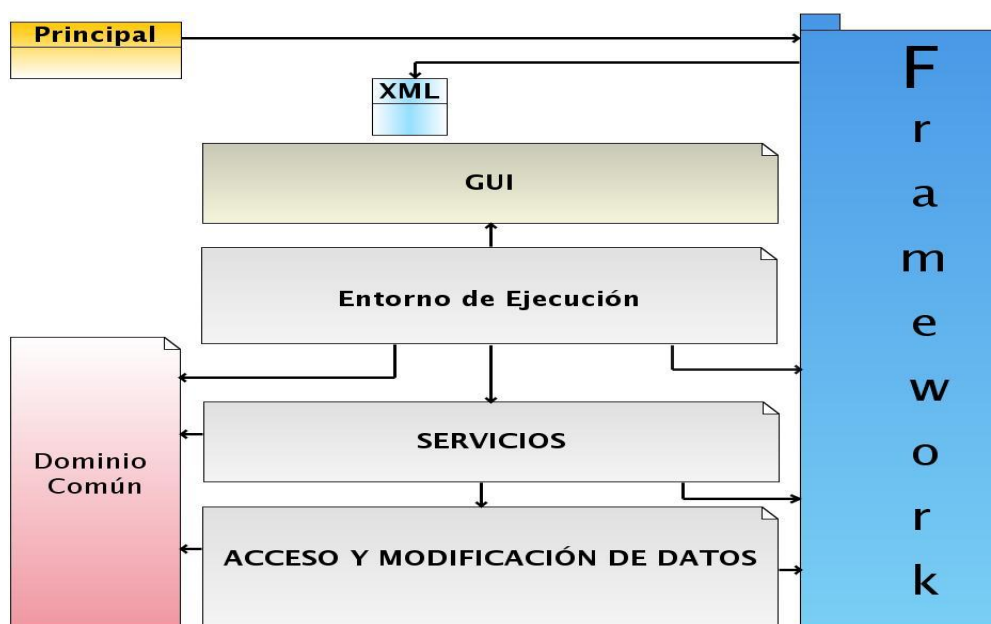


Figura 6 Estructura general del sistema

El patrón de arquitectura empleado es el patrón n-capas, además de las capas de Interfaz, Servicios y Acceso y Modificación de Datos; existen algunos elementos importantes. En primer lugar se ha decidido construir un framework que se encargue de resolver problemas puntuales relacionados con la iniciación del sistema, la instalación desde fuentes, la ejecución de funciones en paralelo, las conexiones remotas, entre otros aspectos; con el objetivo de poder reutilizar estas funcionalidades en futuros desarrollos de software y proporcionarle al desarrollador la posibilidad de concentrarse en aspectos puramente de negocio. Se ha generado una nueva capa denominada Entorno_Ejecución cuya función principal es tratar los eventos externos al sistema, así como el trabajo con funcionalidades del framework. La posibilidad que brinda Python como lenguaje para modelar las interfaces de manera separada a su codificación, haciendo que solo la Interfaz contenga la definición de los componentes empleados ofrece aún mayor claridad a la solución de software. La configuración de la estructura general del sistema estará reflejada en un archivo XML que será utilizado por el framework para conocer que clases serán empleadas y donde residirá su implementación particular. Otro aspecto importante es comprender el funcionamiento del componente Dominio Común, el mismo contiene entidades o clases que serán reutilizados por más de una clase en el sistema.

3.2.1 Estructura detallada del Sistema

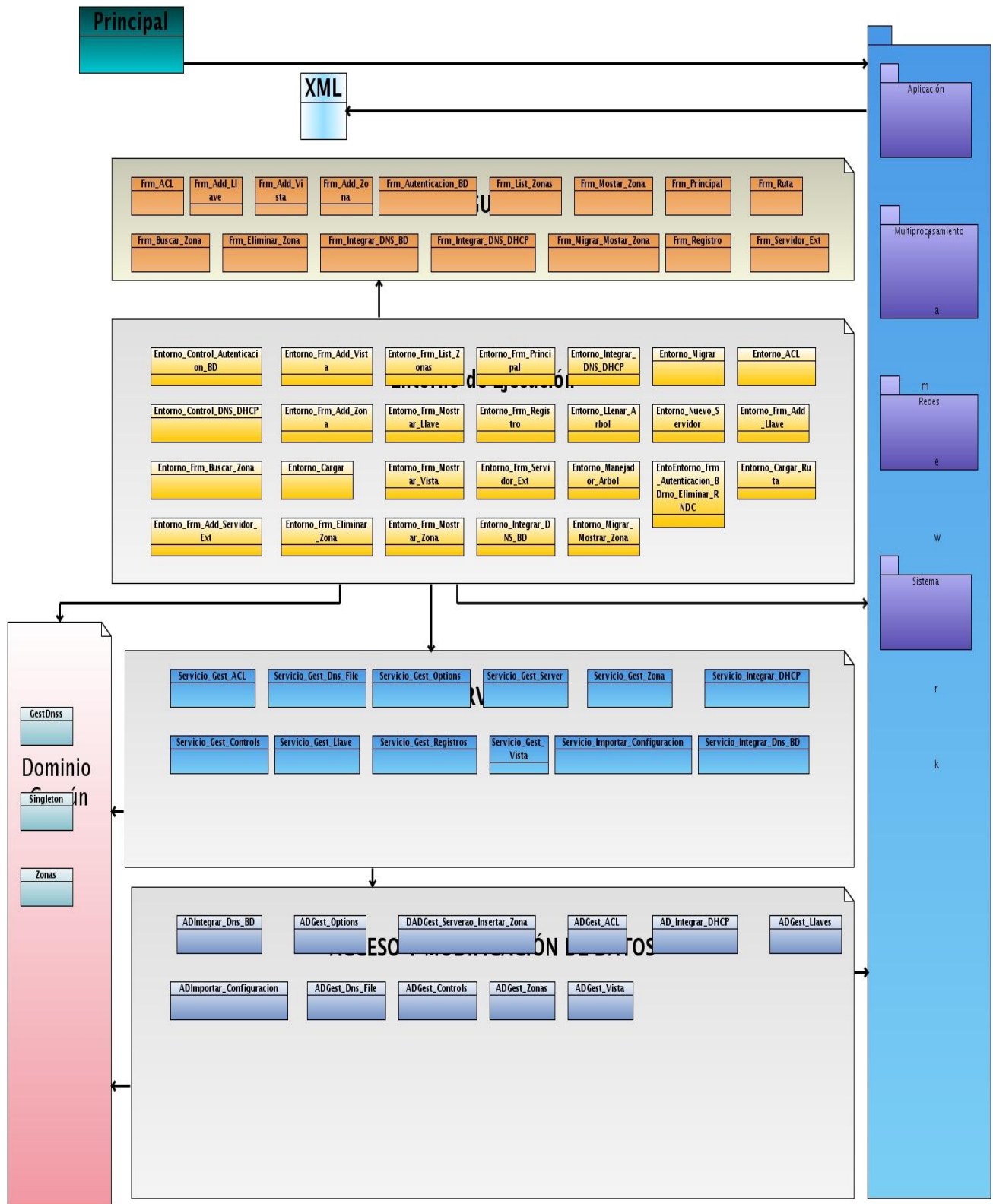


Figura 7 Estructura detallada del Sistema

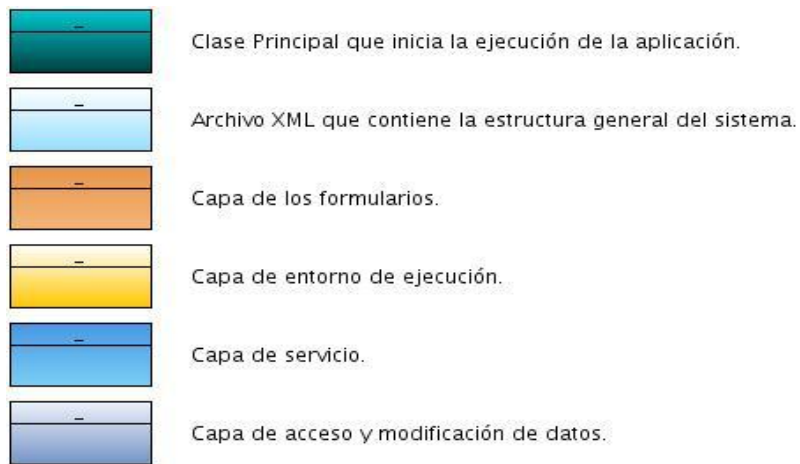


Figura 8 Definición de los términos fundamentales

3.3 Diagrama de clases del diseño.

Ver en el **Anexo 1** los diagramas de clases del diseño especificados con todos los atributos de las clases.

Tabla 3 Diagrama de Clases del diseño. CU Gestionar Zona

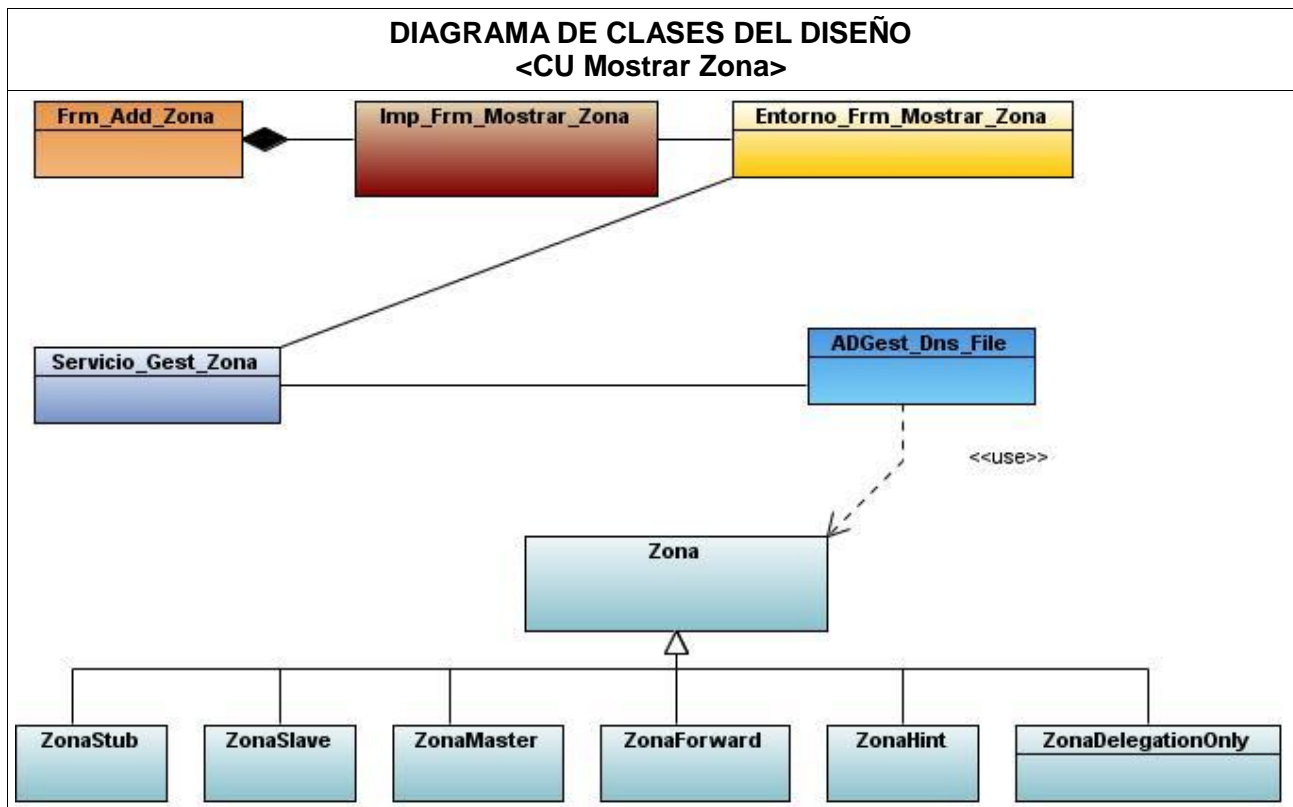


Tabla 4 Diagrama de Clases del diseño. CU Gestionar ACL

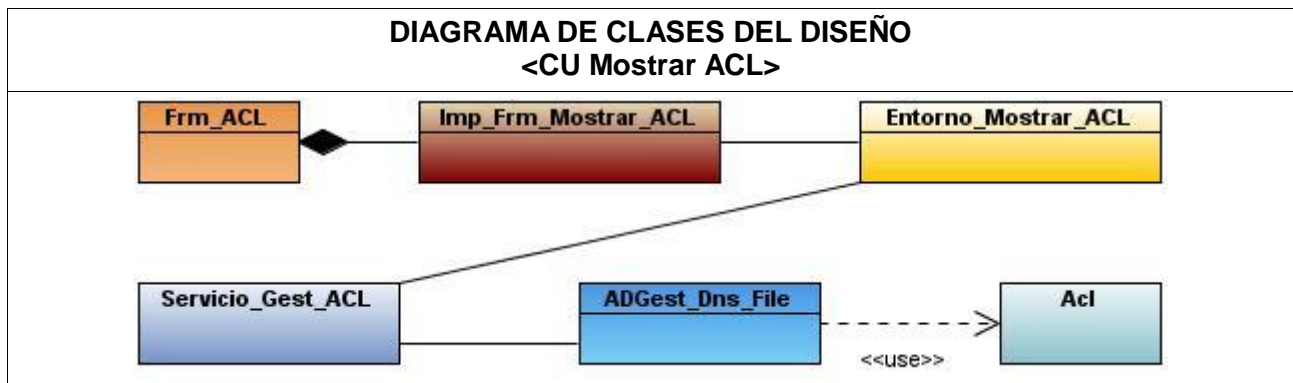


Tabla 5 Diagrama de Clases del diseño. CU Gestionar Registro

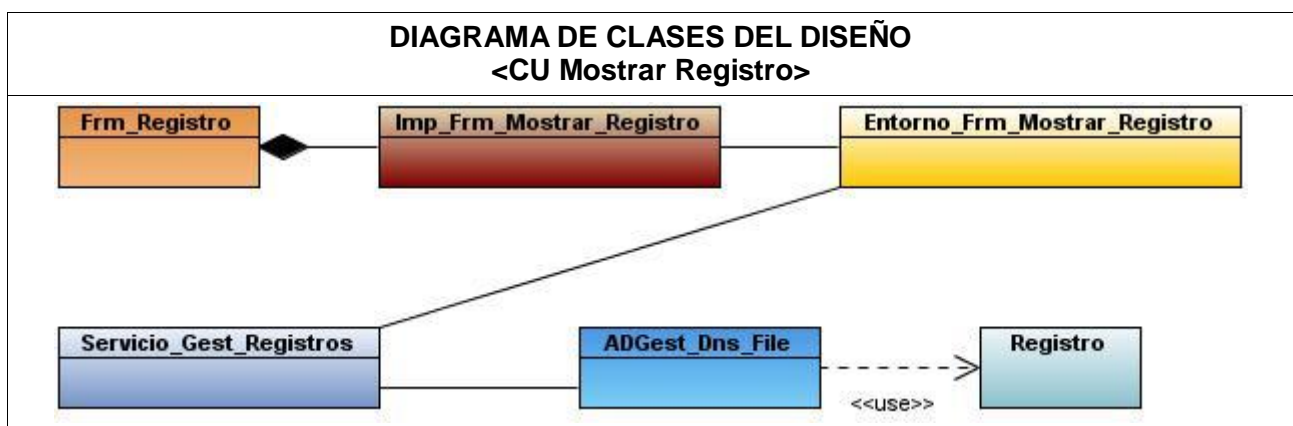


Tabla 6 Diagrama de Clases del diseño. CU Gestionar Servidor Externo

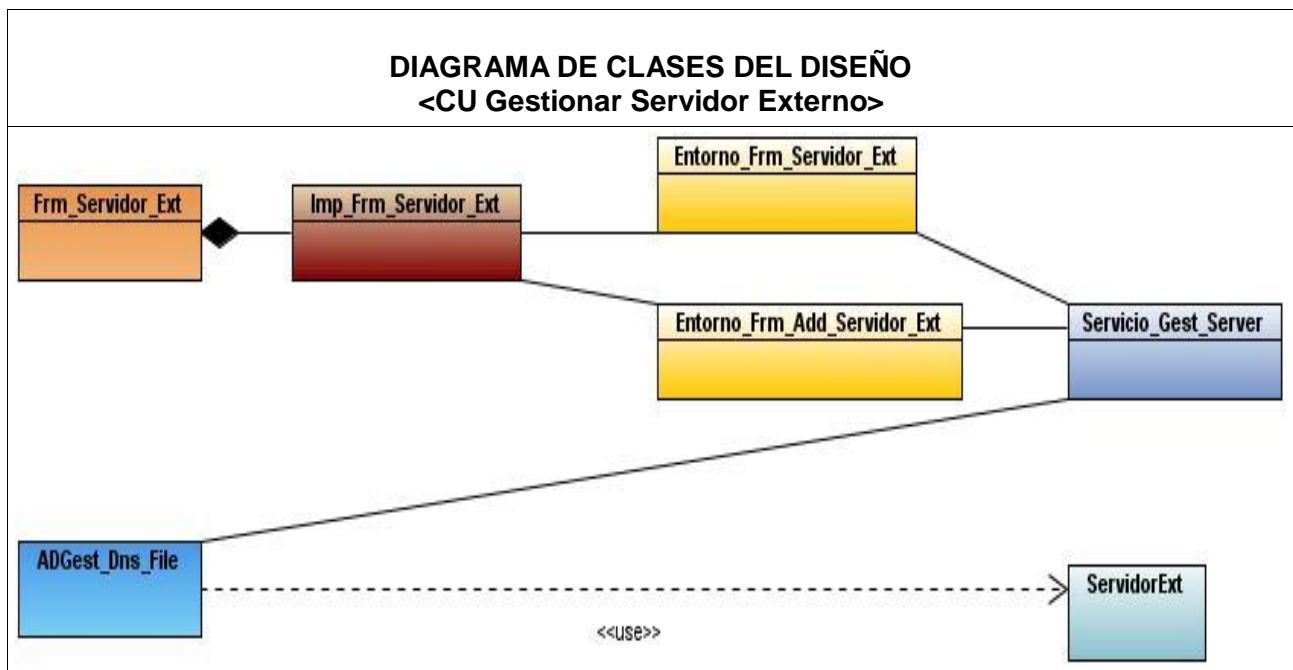


Tabla 7 Diagrama de Clases del diseño. CU Gestionar Vista

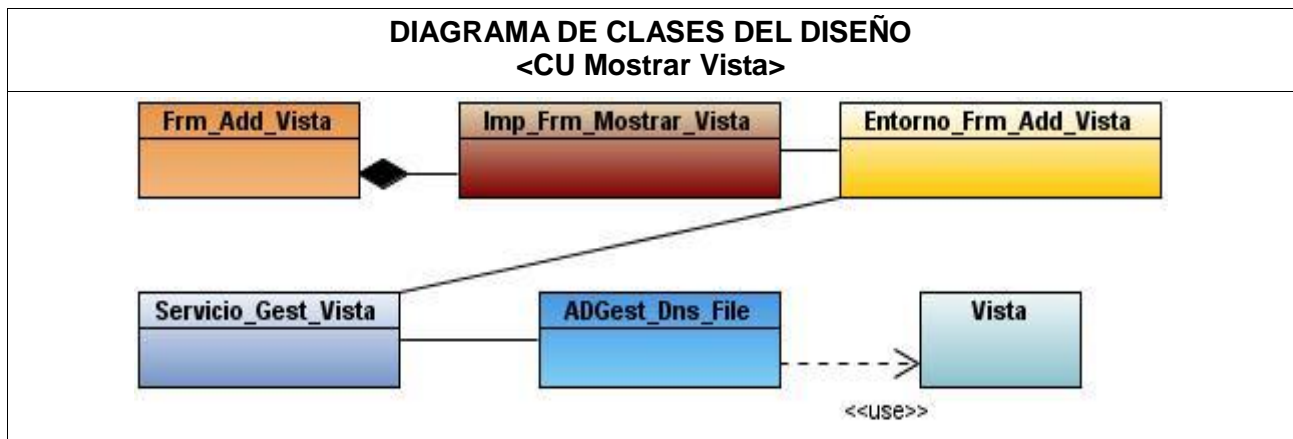


Tabla 8 Diagrama de Clases del diseño. CU Integrar DNS con BD

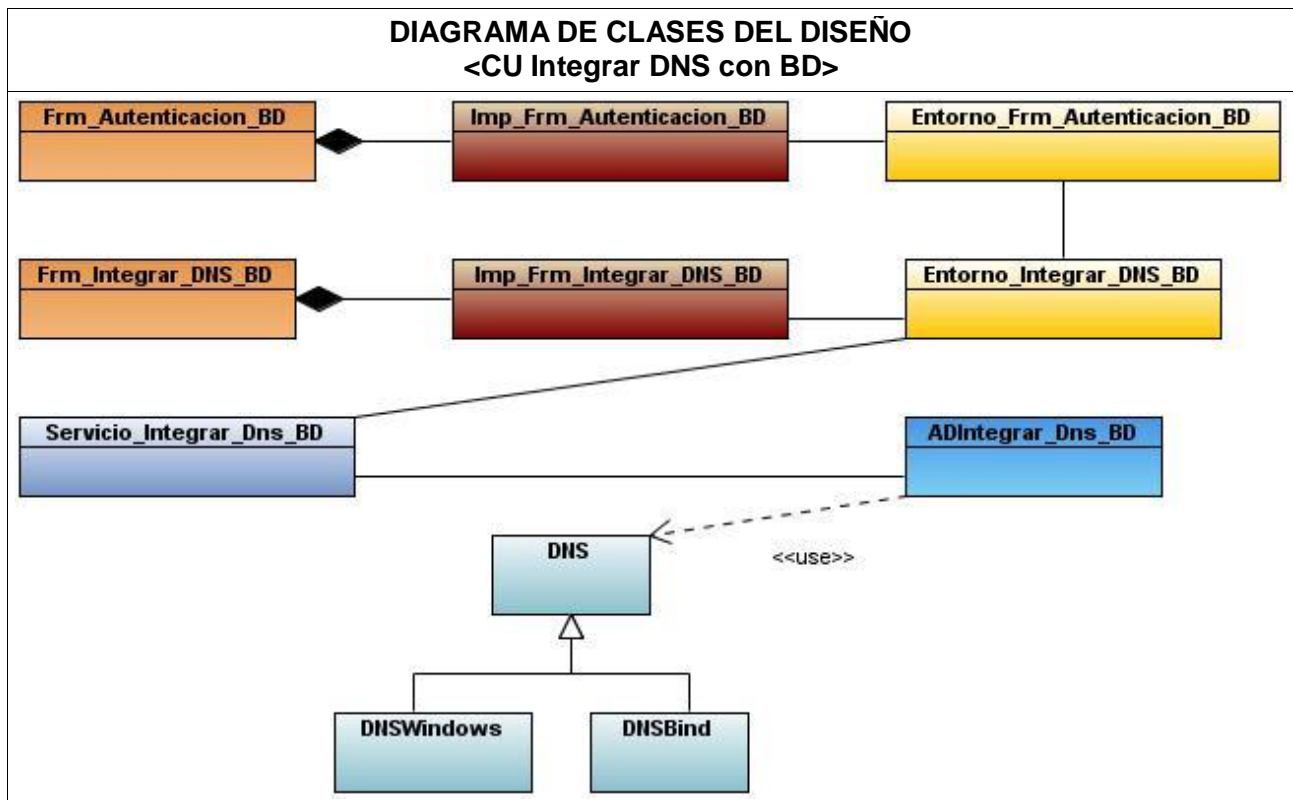


Tabla 9 Diagrama de Clases del diseño. CU Integrar DNS con DHCP

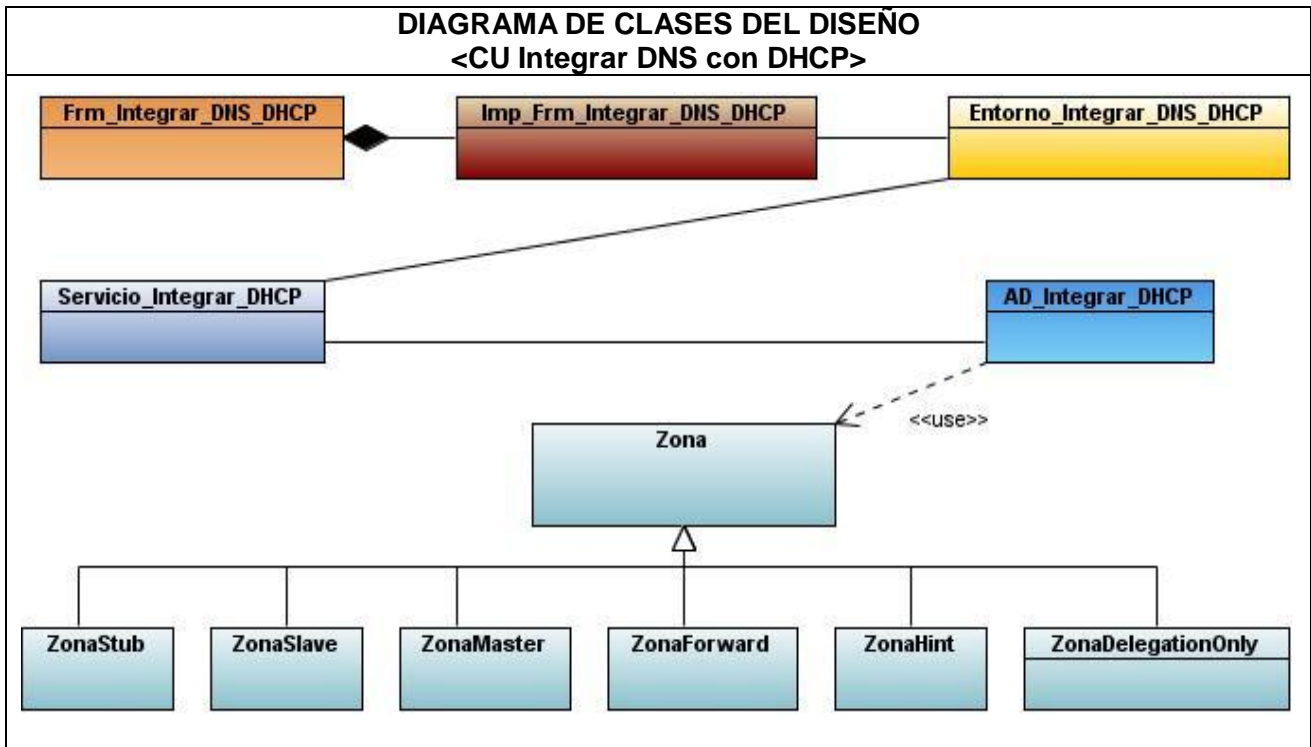


Tabla 10 Diagrama de Clases del diseño. CU Importar y exportar configuraciones DNS

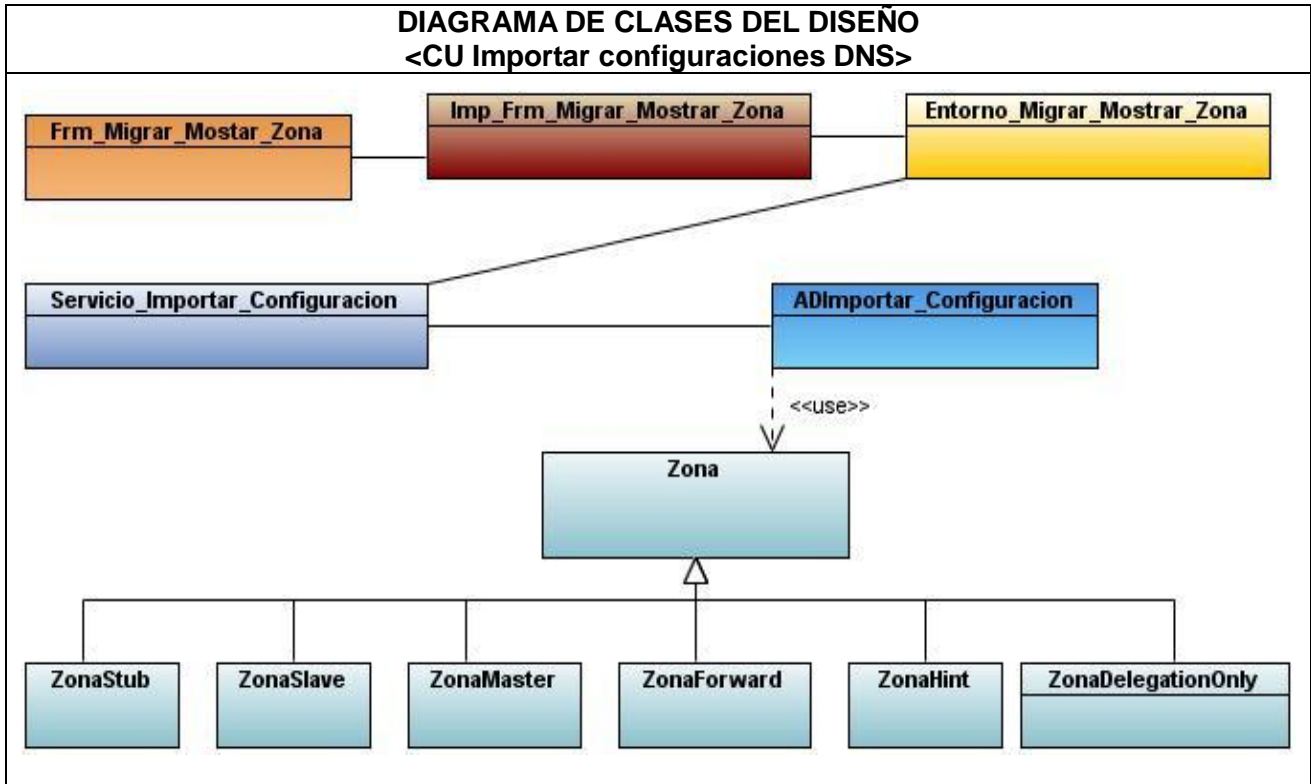
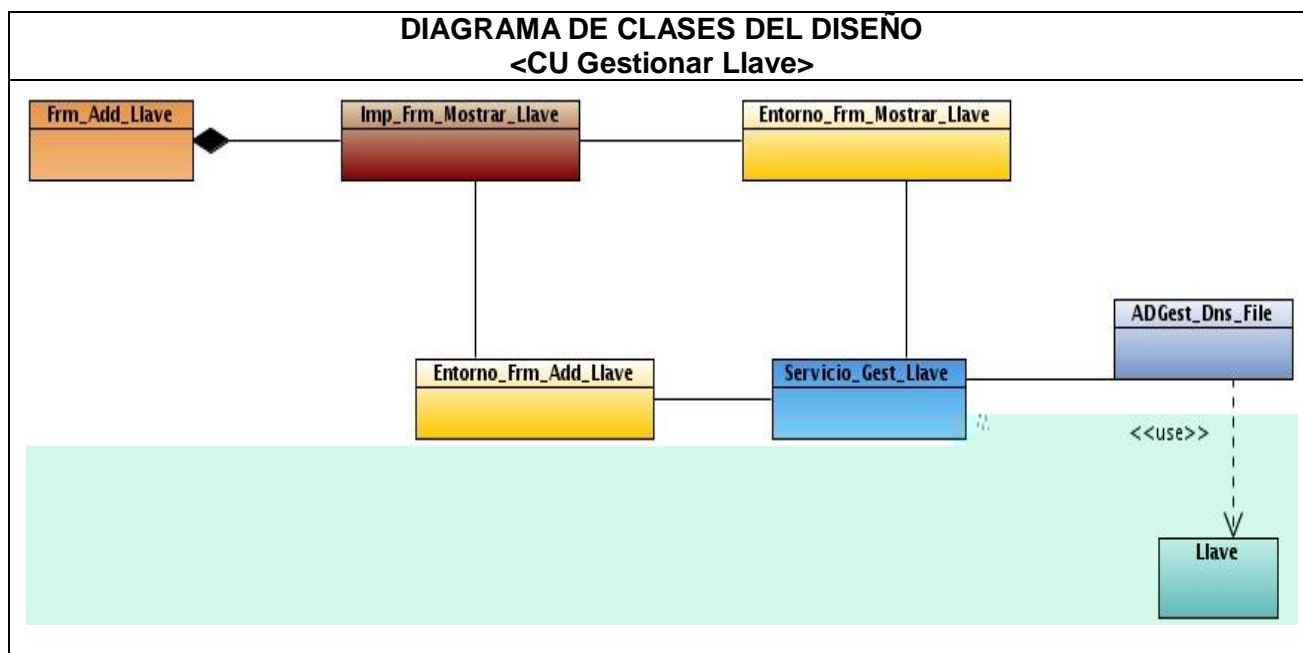


Tabla 11 Diagrama de Clases del diseño. CU Gestionar Llave



3.4 Conclusiones

En el capítulo fueron presentados los artefactos fundamentales correspondientes al Diseño del sistema, entre ellos los modelos diseño, los cuales constituyen la entrada principal para la futura implementación del sistema, y también se presentó la arquitectura empleada, donde se hizo una explicación detallada de la misma, así como se explicó la nueva capa que fue creada, buscando que la aplicación sea lo más escalable y extensible posible..

Capítulo 4 Implementación

4.1 Introducción.

En el presente capítulo se desarrolla el flujo de trabajo correspondiente a la implementación. Para el desarrollo del mismo se tuvieron en cuenta algunos artefactos que fueron generados en el diseño del sistema, a partir de los cuales se desarrolló la implementación del sistema en términos de componentes, es decir, ficheros de código fuente, scripts, ficheros de código binario, ejecutables y similares. Además se representa el diagrama de despliegue que contiene la topología hardware sobre la que se ejecuta el sistema.

4.2 Diagrama de despliegue

El diagrama de despliegue representa la arquitectura de tiempo de ejecución de los procesadores, dispositivos y los componentes de software que se ejecutan en esa arquitectura. Es la última descripción física de la topología del sistema y describe la estructura de las unidades de hardware. Además, representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un procesador, un dispositivo o memoria. En los procesadores es donde se encuentran alojados los componentes.

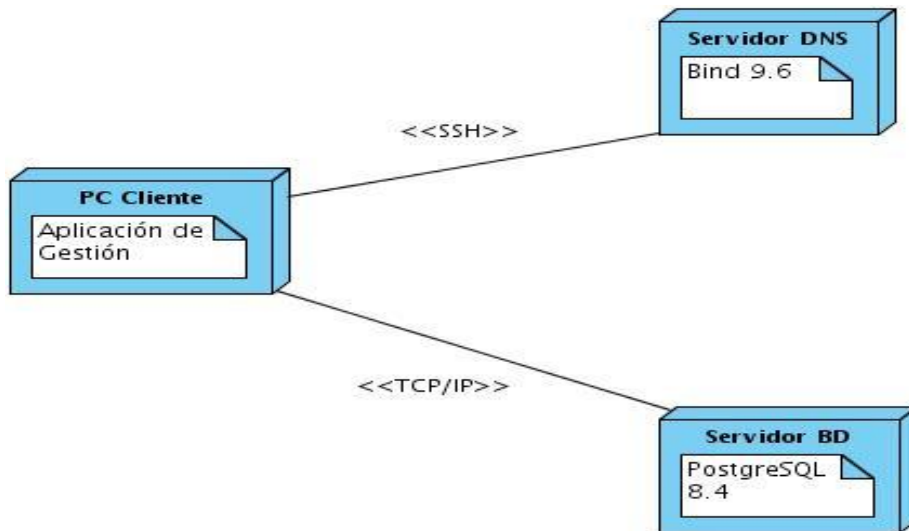


Figura 9 Diagrama de Despliegue

4.3 Diagrama de componentes

Para lograr una mejor comprensión de los componentes que forman el sistema, se presentarán los diagramas de componentes. Un componente es el empaquetamiento físico de los elementos de un modelo, como son las clases en el modelo de diseño.

Tabla 12 Diagrama de Componentes General

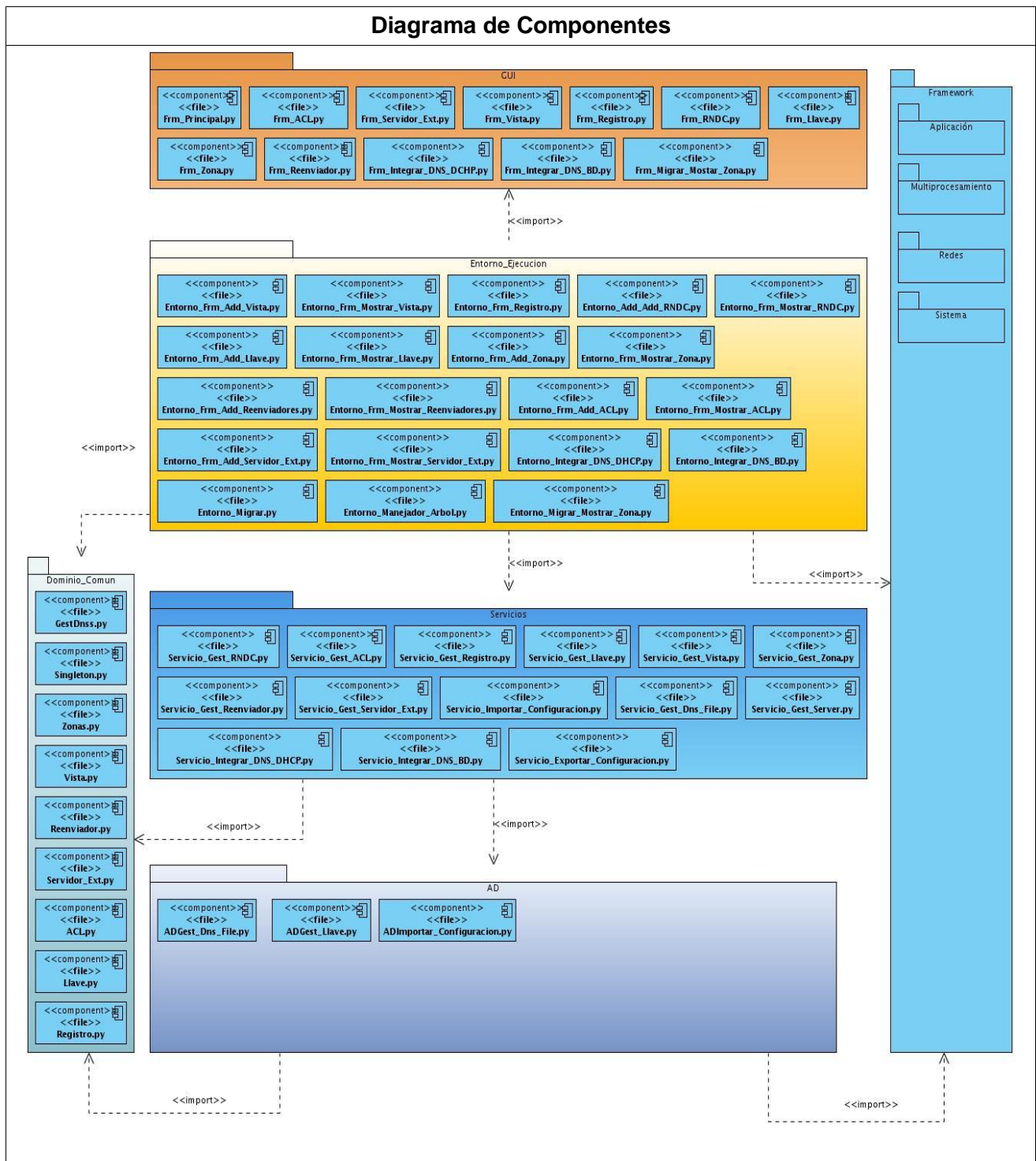


Tabla 13 Diagrama de Componentes. CU Gestionar Vista

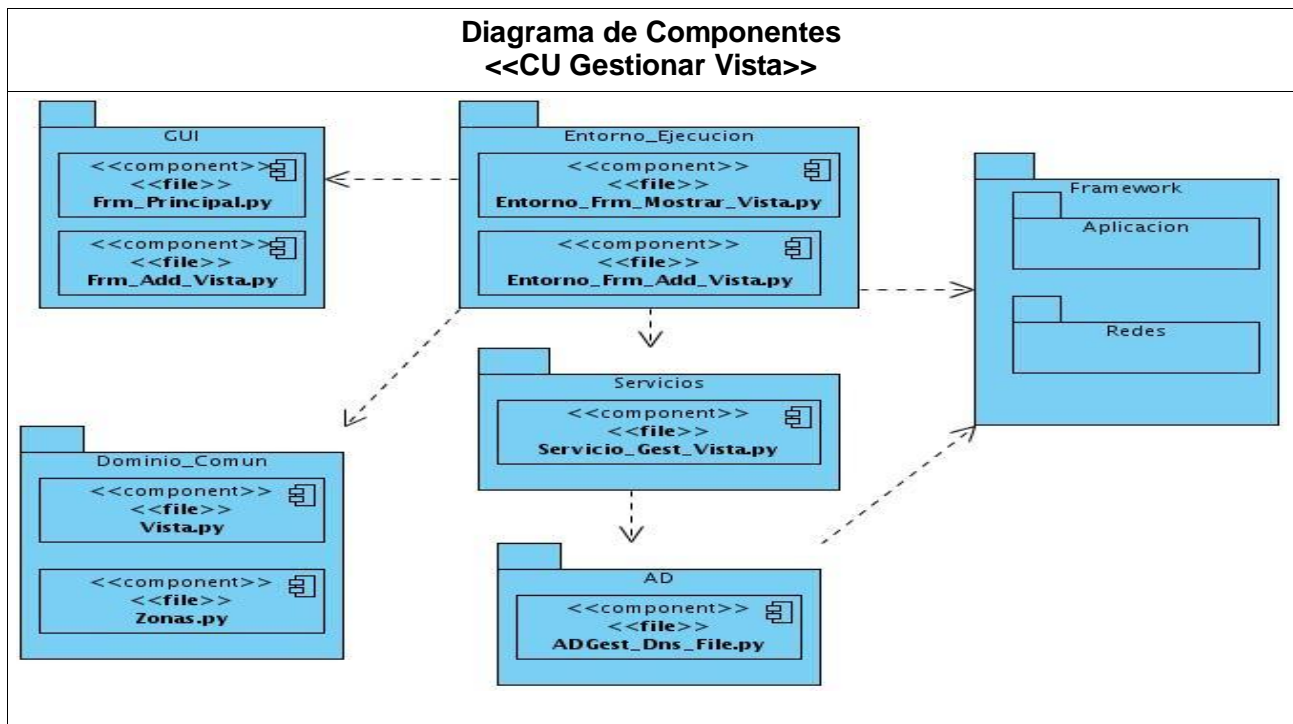


Tabla 14 Diagrama de Componentes. CU Gestionar Zona

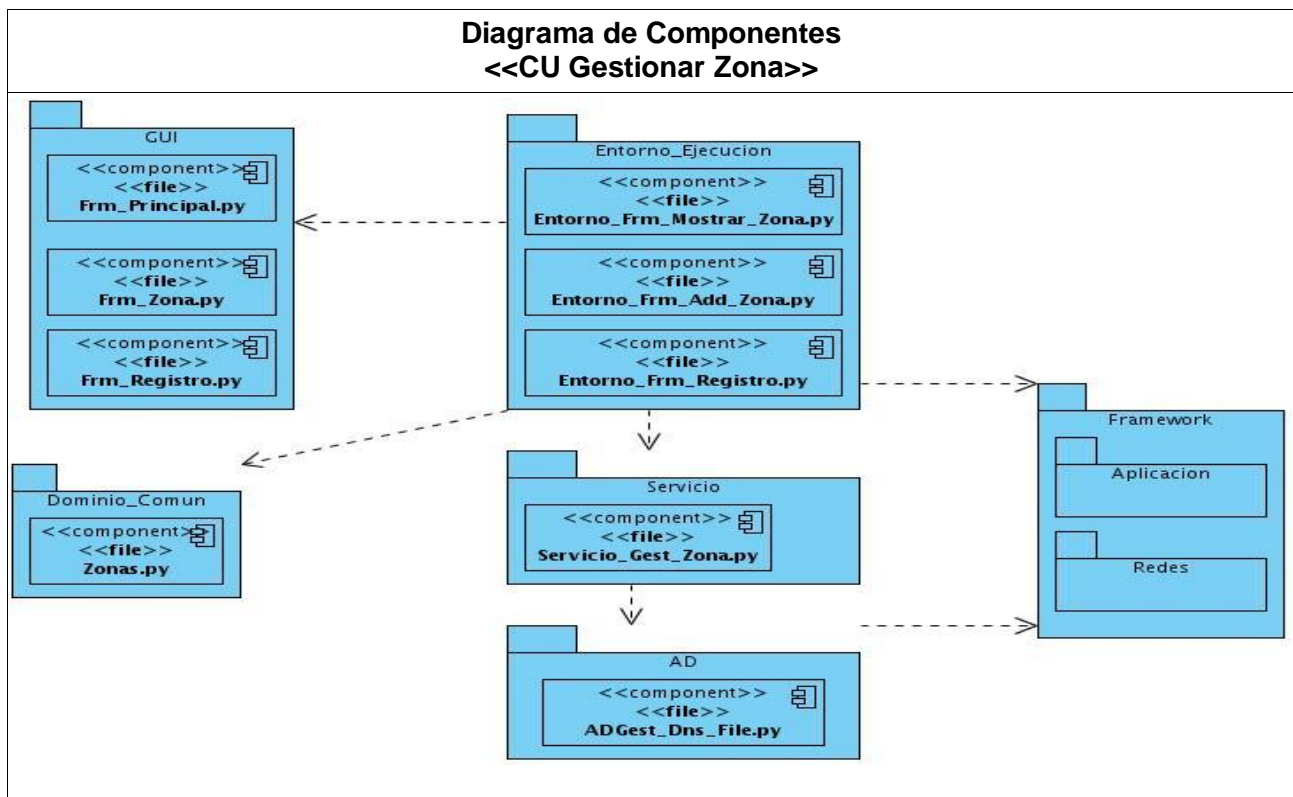


Tabla 15 Diagrama de Componentes. CU Gestionar Llave

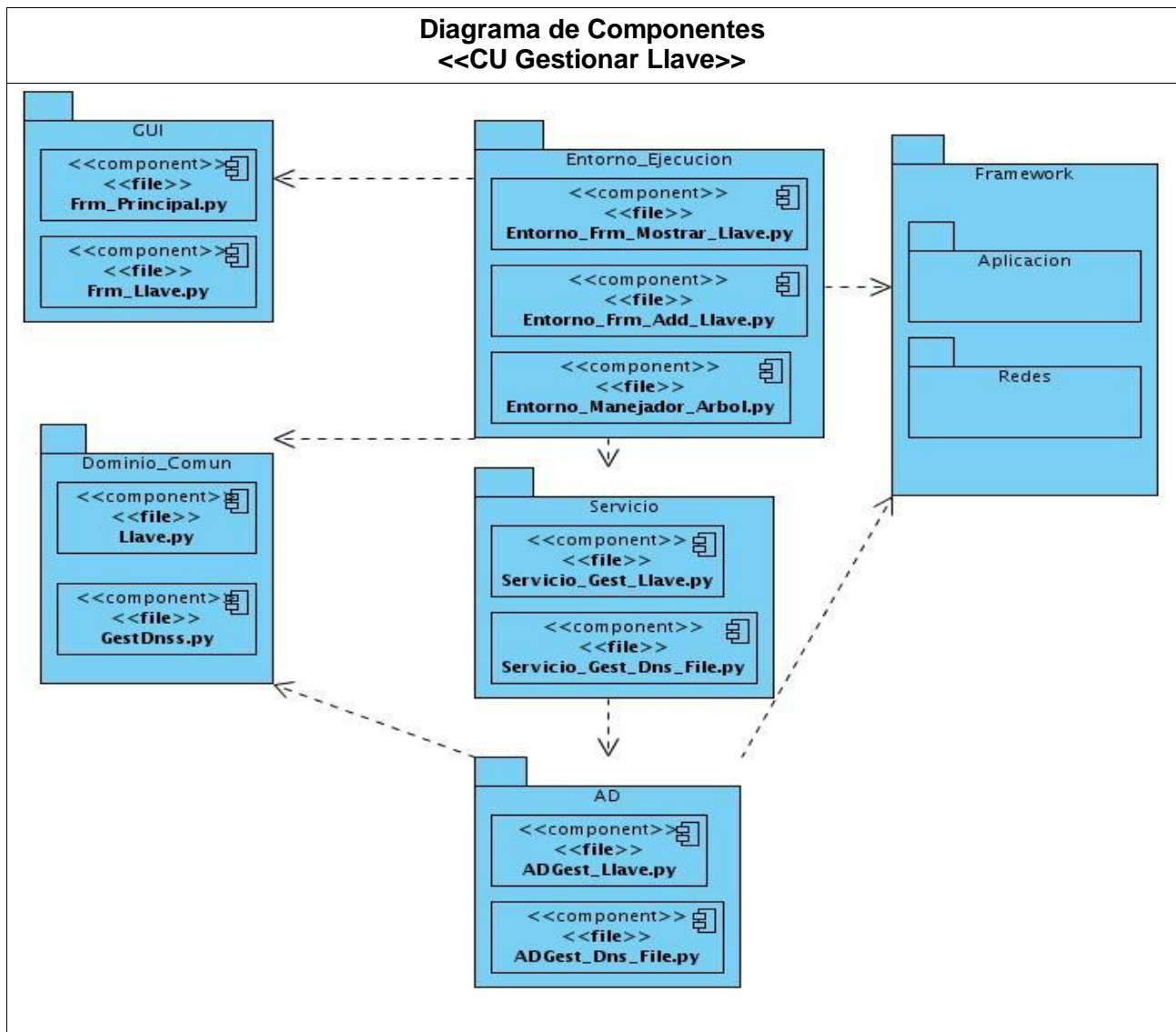


Tabla 16 Diagrama de Componentes. CU Gestionar ACL

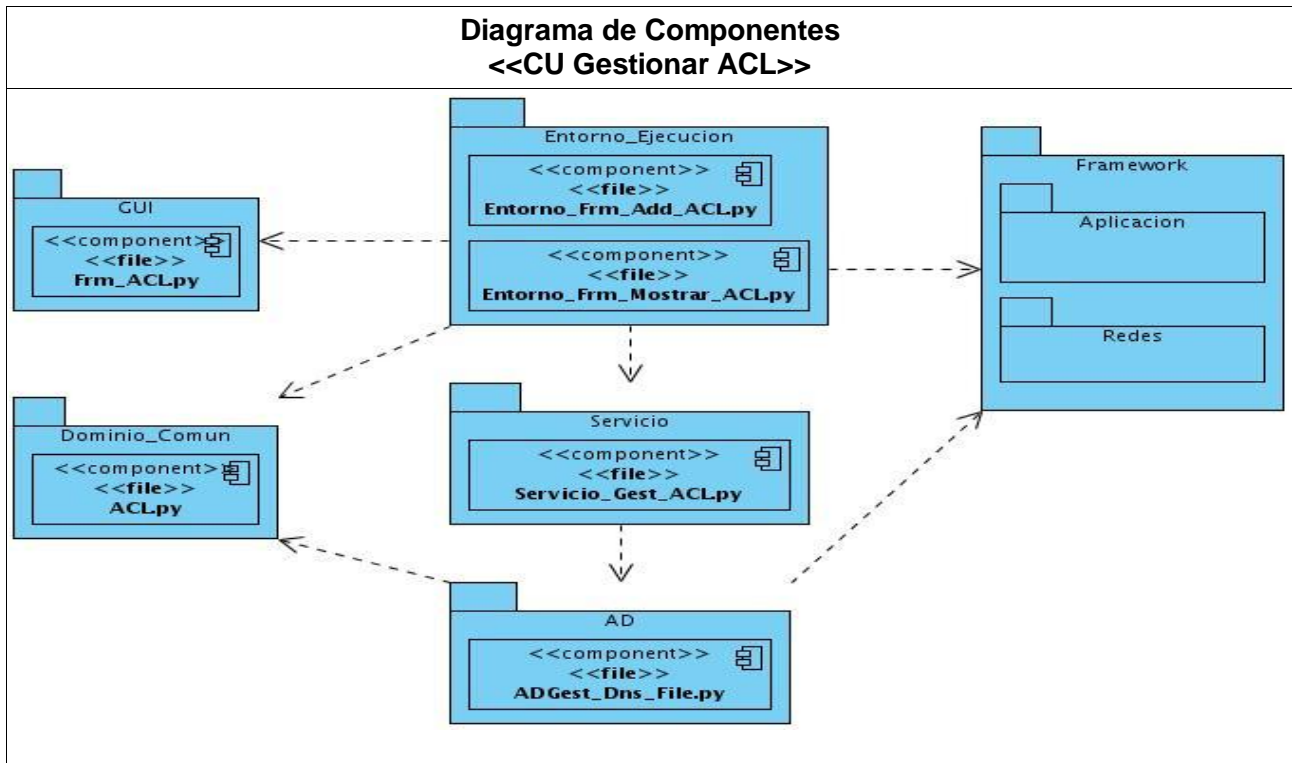


Tabla 17 Diagrama de Componentes. CU Gestionar Reenviadores

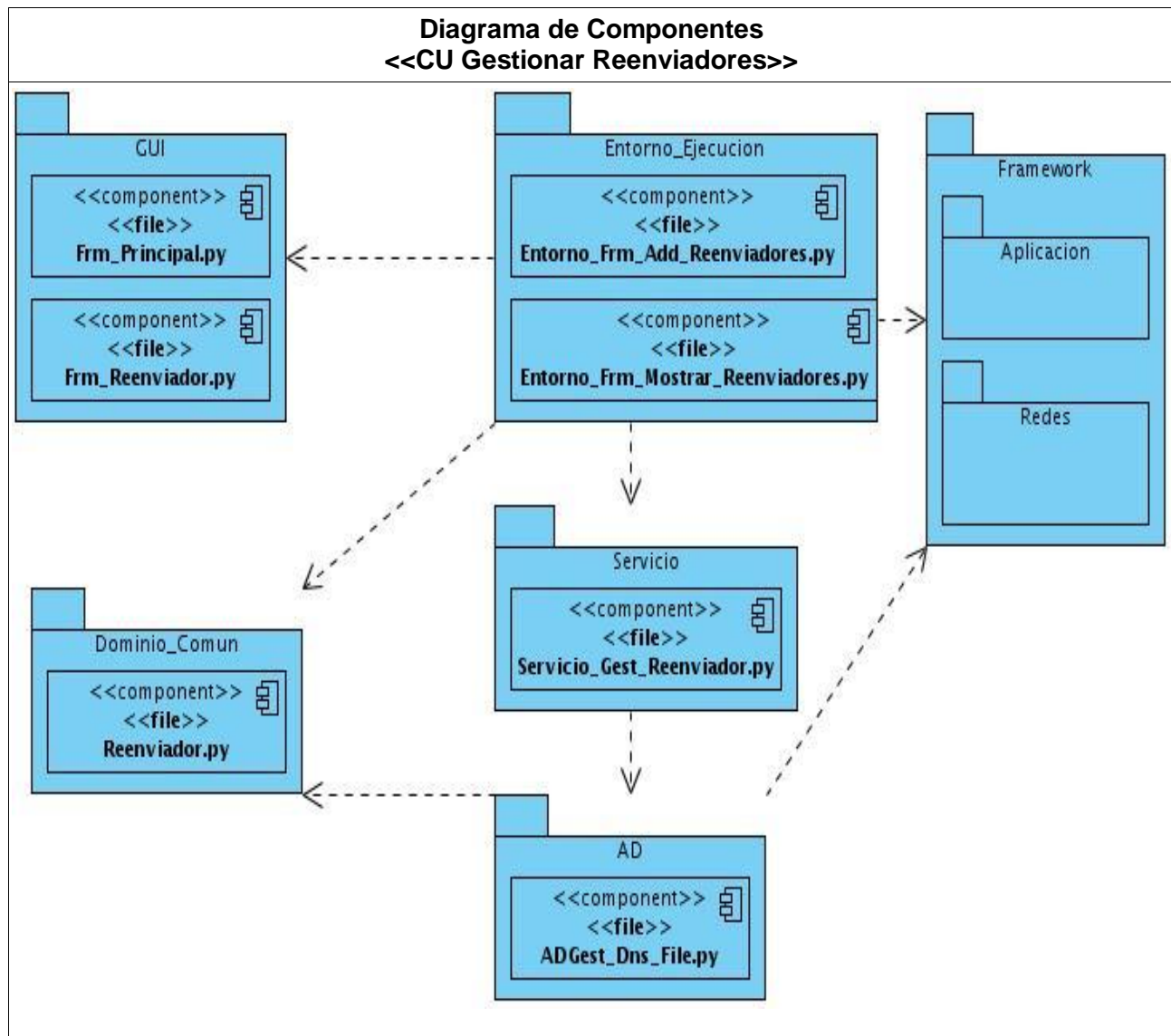


Tabla 18 Diagrama de Componentes. CU Gestionar Servidor Externo

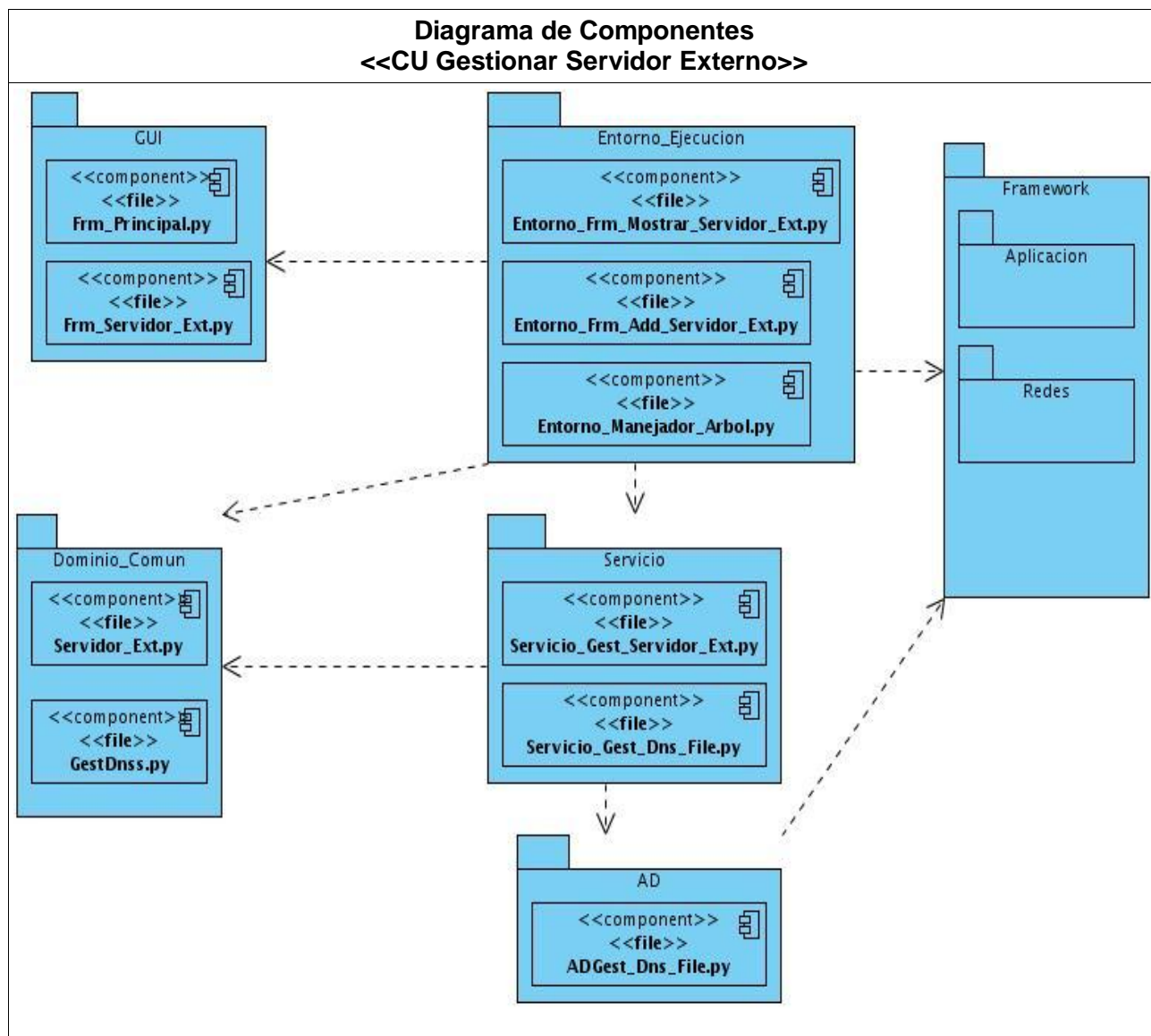
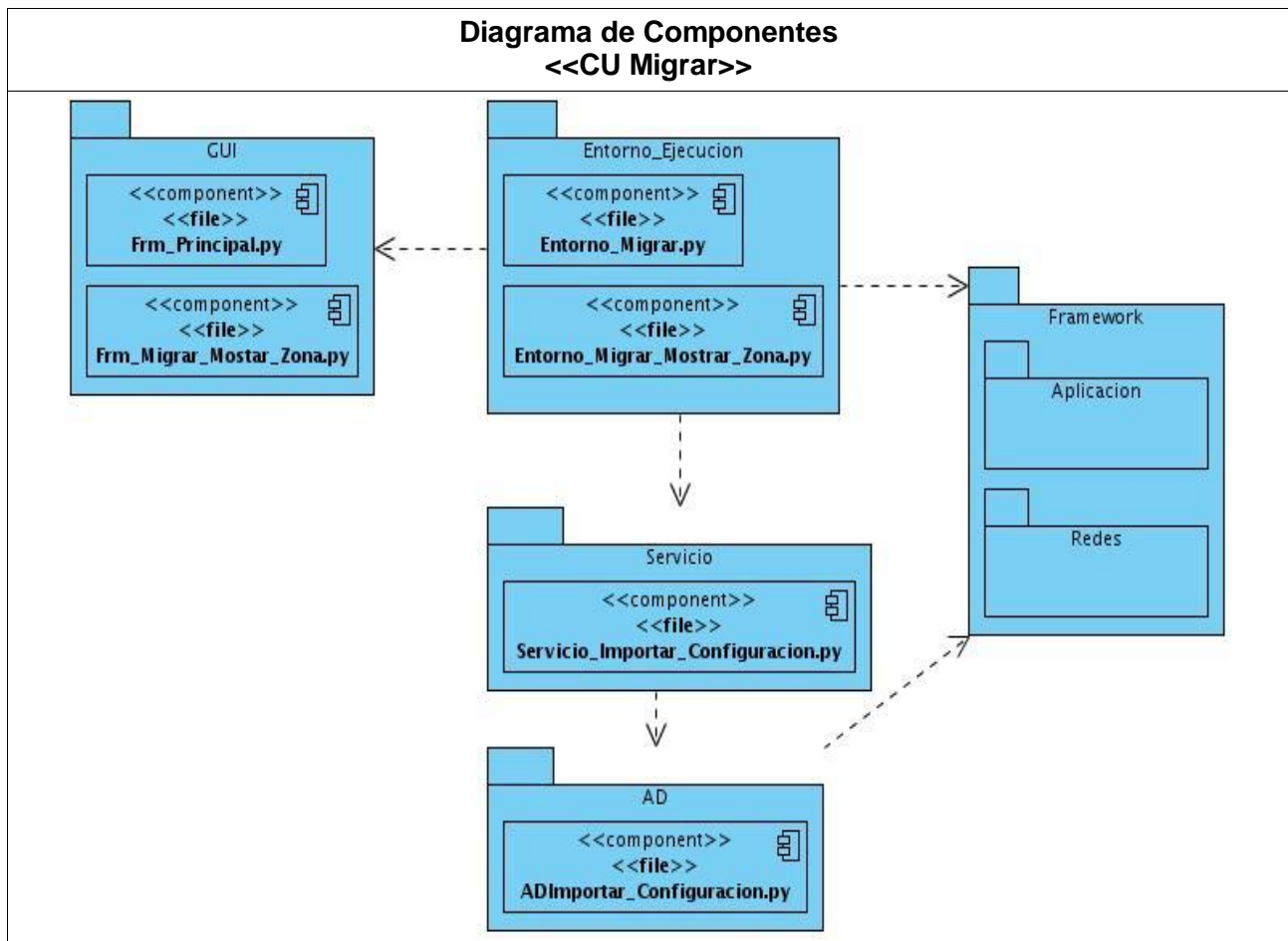


Tabla 19 Diagrama de Componentes. CU Migrar



4.4 Conclusiones

En el capítulo fueron presentados los artefactos fundamentales correspondientes a la Implementación del Sistema, entre los que se destacan el modelo de despliegue que deja clara la arquitectura en términos de nodos o computadoras y servidores necesarios para la gestión del DNS, el diagrama general de componentes del sistema y los diagramas de componentes de cada Caso de Uso, donde se puede identificar todas las clases empleadas para realizar las funcionalidades planteadas en la aplicación.

Conclusiones

Se llevó a cabo una investigación muy detallada sobre el servidor de nombres de dominio donde se identificaron nuevas funcionalidades y características para ser incorporadas al Módulo DNS de la Plataforma de Gestión de Servicios Telemáticos en GNU/Linux, se estudiaron herramientas que permiten la gestión del DNS y de cada una se tomaron las mejores prácticas para lograr una aplicación con un alto nivel y con la posibilidad de competir en el mercado, se hizo un refinamiento de las funcionalidades que se desarrollaron en la versión 1.0 del software, se desarrolló un asistente que permite compilar un DNS BIND9 y se logró una aplicación de escritorio que brinda la posibilidad de gestionar el servidor de nombres de dominio tanto en sistema operativo Linux como Windows, con nuevas funcionalidades como la integración del DNS con DHCP, con Base de Datos y la gestión de las llaves y los servidores externos del DNS, así como importar las configuraciones que residan en un servidor Windows Server 2003 para un servidor BIND9 y que el mismo pueda seguir funcionando en perfectas condiciones sin cambios en sus configuraciones, además la aplicación desarrollada se adecua a las necesidades de los administradores de red cubanos, permite trabajar evitando el uso de las consolas, de una forma intuitiva y fácil de usar.

Recomendaciones

Se recomienda seguir con la investigación que se ha venido desarrollando para incorporar nuevas funcionalidades a la aplicación, optimizar al máximo todas las funciones que realiza, utilizar bases de datos distribuidas para la gestión de toda la información almacenada del servidor DNS al realizar la migración de las configuraciones, además que se comience una investigación para integrar el Servidor de Nombres de Dominio (DNS) con el Directorio Activo.

Bibliografía

- [1] **Gualdarrama Reyes, Jorge Luis.** *Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo DNS.* Ciudad La Habana: s.n., 2009. p. 5.
- [3] **Microsoft.com,** *Presentación de la familia Windows Server 2003,* 3 Marzo 2010. <http://www.microsoft.com/latam/windowsserver2003/evaluation/overview/family.msp>
- [4] **Gualdarrama Reyes, Jorge Luis.** *Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo DNS.* Ciudad La Habana: s.n., 2009. p. 8-9.
- [5] **Gualdarrama Reyes, Jorge Luis.** *Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo DNS.* Ciudad La Habana: s.n., 2009. p. 9.
- [6] **James Rumbaugh, Ivar Jacobson, Grady Booch.** *El Proceso Unificado del Desarrollo de Software,* 2000. [Disponible en: <http://biblioteca.uci.cu/titdigitales.htm#igs>]
- [7] **James Rumbaugh, Ivar Jacobson, Grady Booch.** *El Lenguaje Unificado de Modelado. Manual de Referencia,* 1998. [Disponible en: <http://bibliodoc.uci.cu/pdf/reg03050.pdf>]
- [8] **Alfonso de la Guarda Reyes.** *A Byte of Python,* 2004. [Disponible en: http://dev.laptop.org/~edsiper/byteofpython_spanish/index.html]
- [9] **Noa Cabrera, Sandy.** *Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo Directorio.* Ciudad La Habana: s.n., 2009.
- [10] **REED, J. C.** *BIND 9 Administrator Reference Manual.* 2007. 211 p. ISBN 0-9790342-1-3.
- [11] **JAMES RUMBAUGH, I. J., GRADY BOOCH.** *El lenguaje unificado de modelado. Manual de referencia.*
- [12] **RODERA RODA, C.** *Microsoft Windows NT Server Guía de Redes.* 1997. 817 p. ISBN 1-57231-344-7.
- [13] **AITCHISON, R. G. F.** *Pro DNS and BIND.* 2005. 571 p.
- [14] **DUQUE, R. G.** *Python para Todos.* 160 p.
- [15] **MAXWELL, S.** *Red hat linux. Herramientas para la administración de redes.* 516 p.
- [16] **LUGO, R. H.** *Tele Identificador Personal. Configuración de los servidores BIND de DNS y almacenamiento de la información de los subscriptores.* Universidad Ciencias Informáticas, 2009.
- [17] **LARMAN, C.** *Uml y patrones: Introducción al análisis y programación orientada a objetos.* 1999. 536 p. ISBN 970-17-2061-1.
- [18] **What is BIND and what does it do?** [Consultado el: 15-2-2010 de 2010]. Disponible en: <http://www.isc.org/software/bind/whatis>.
- [19] **BOSWELL, W.** *Windows Server 2003.* 2004. 1088 p. ISBN 84-415-1615-4.

Anexo 1 Diagramas de Clase del Diseño

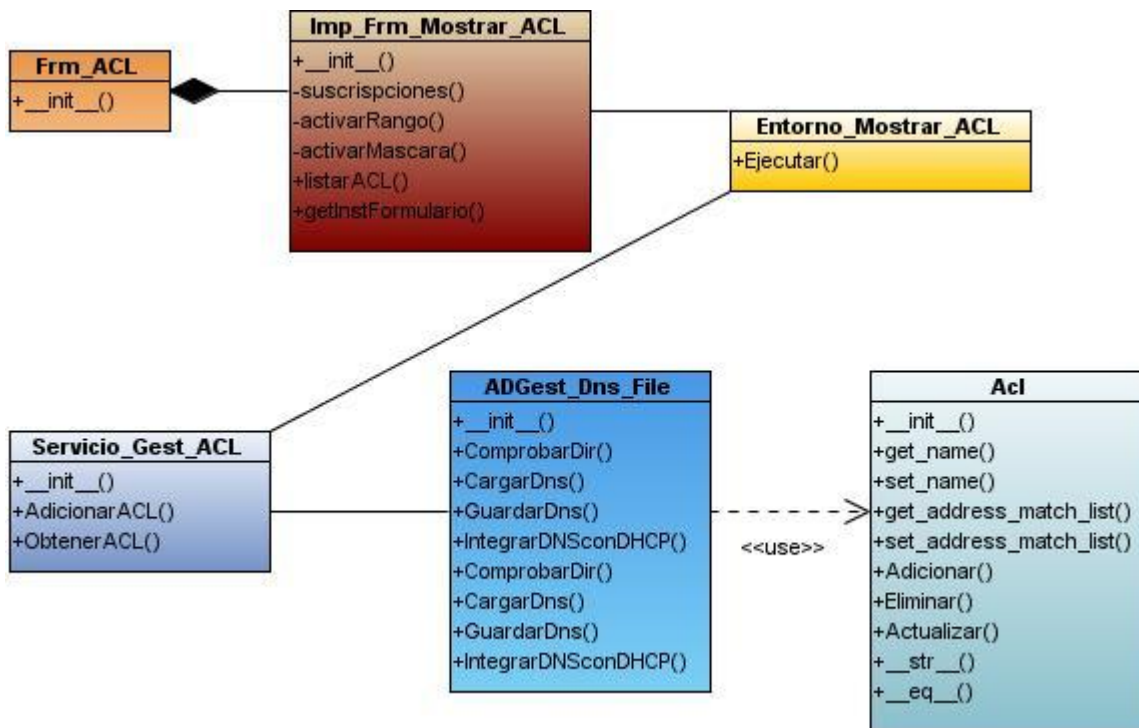


Imagen 1 Diagrama de Clase del Diseño. CU Mostrar ACL

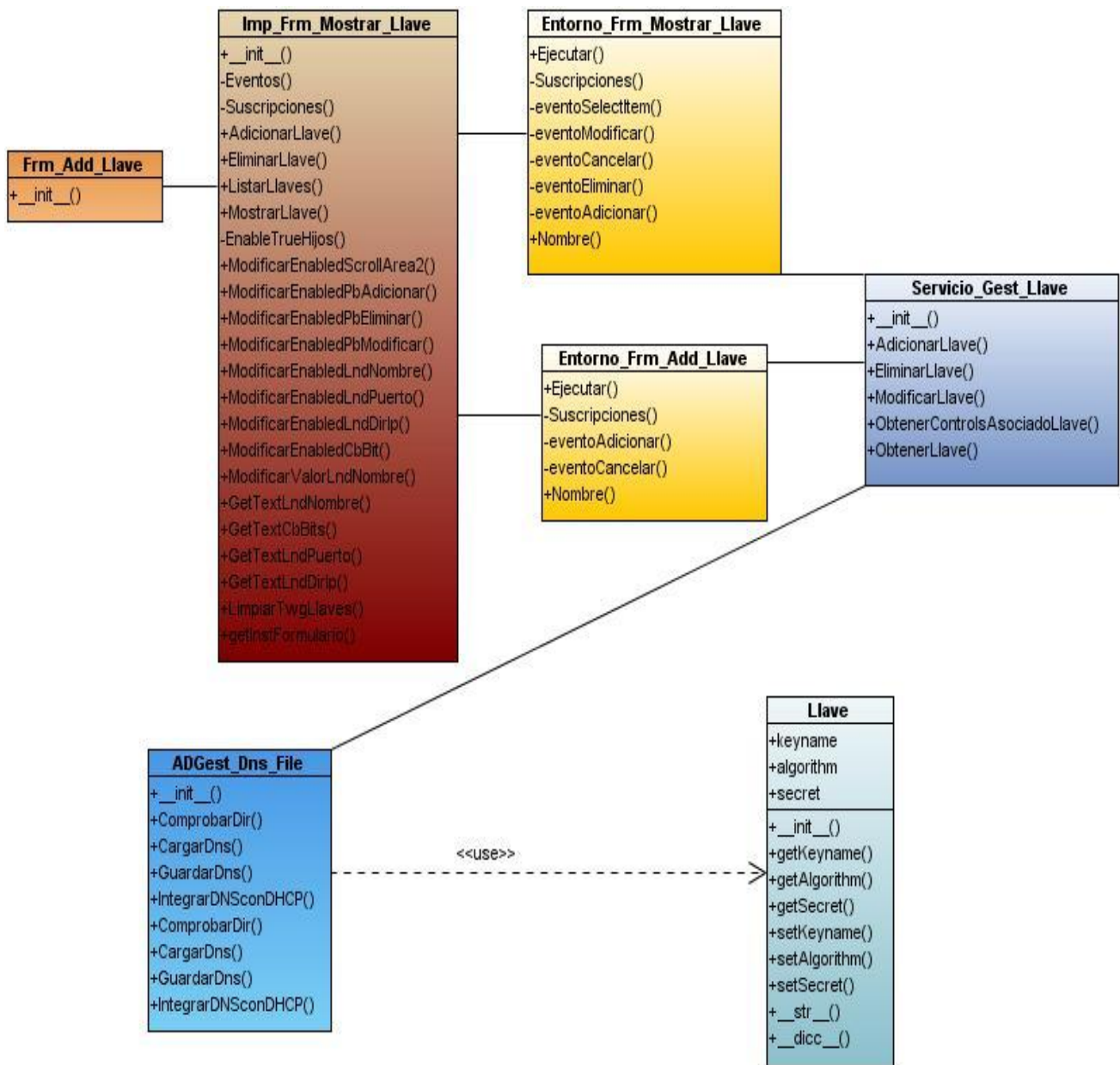


Imagen 2 Diagrama de Clase del Diseño. CU Gestionar Llave

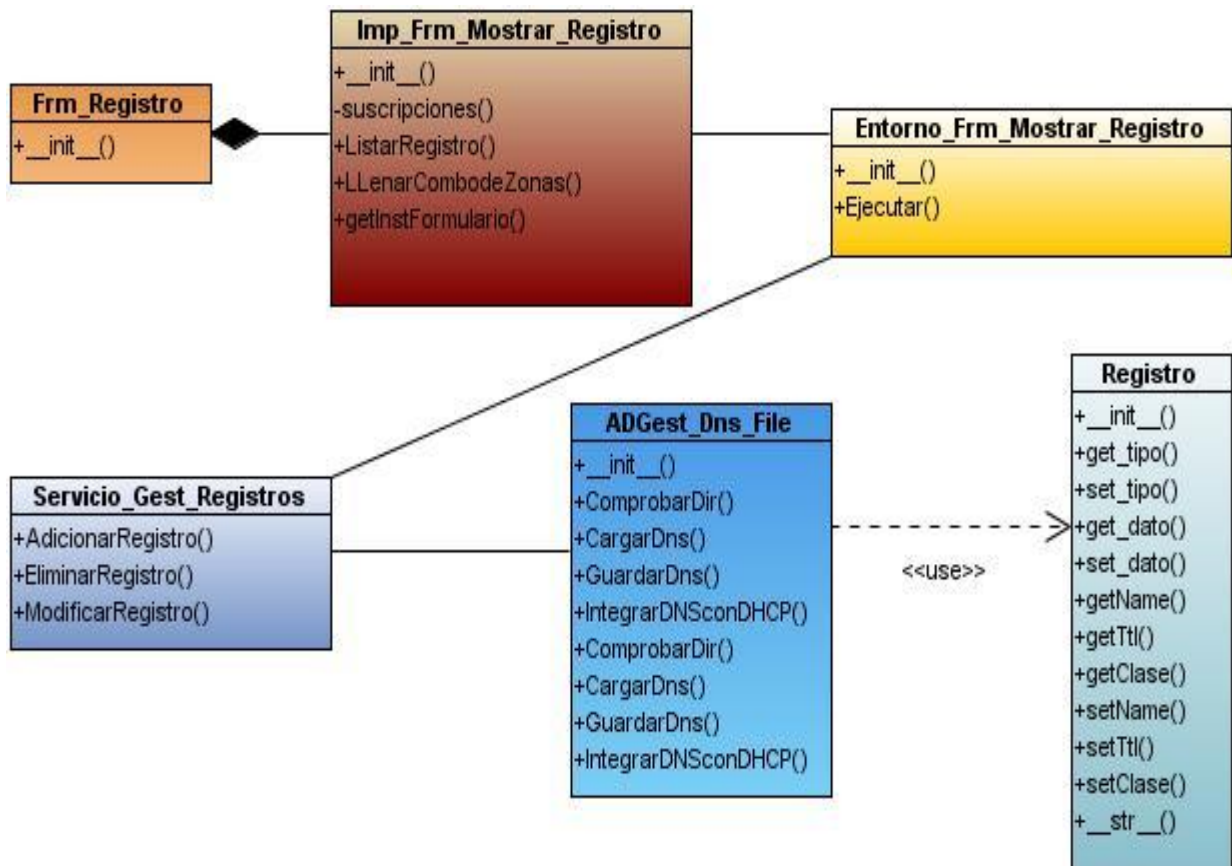


Imagen 3 Diagrama de Clase del Diseño. CU Mostrar Registro

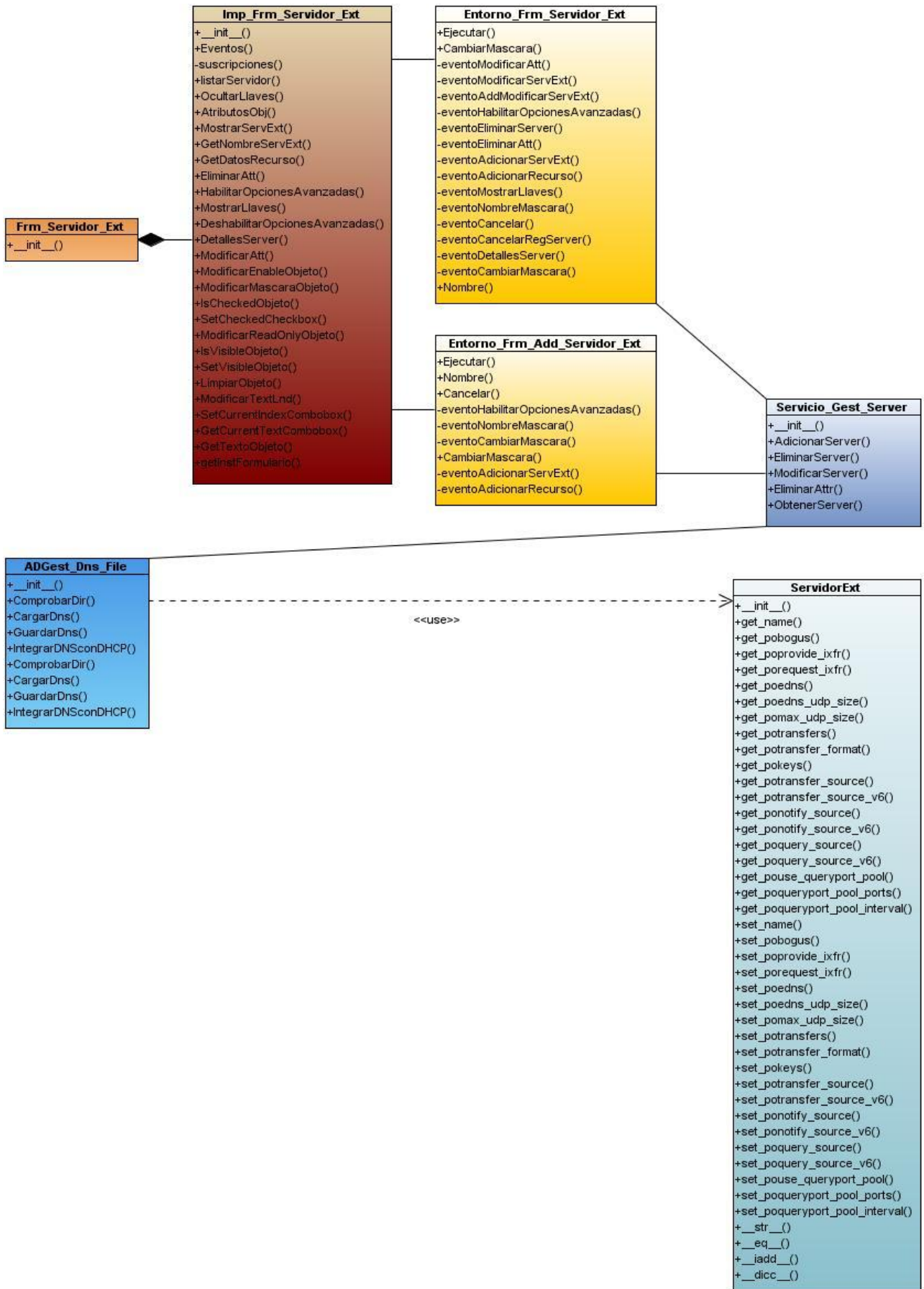


Imagen 4 Diagrama de Clase del Diseño. CU Gestionar Servidor Externo

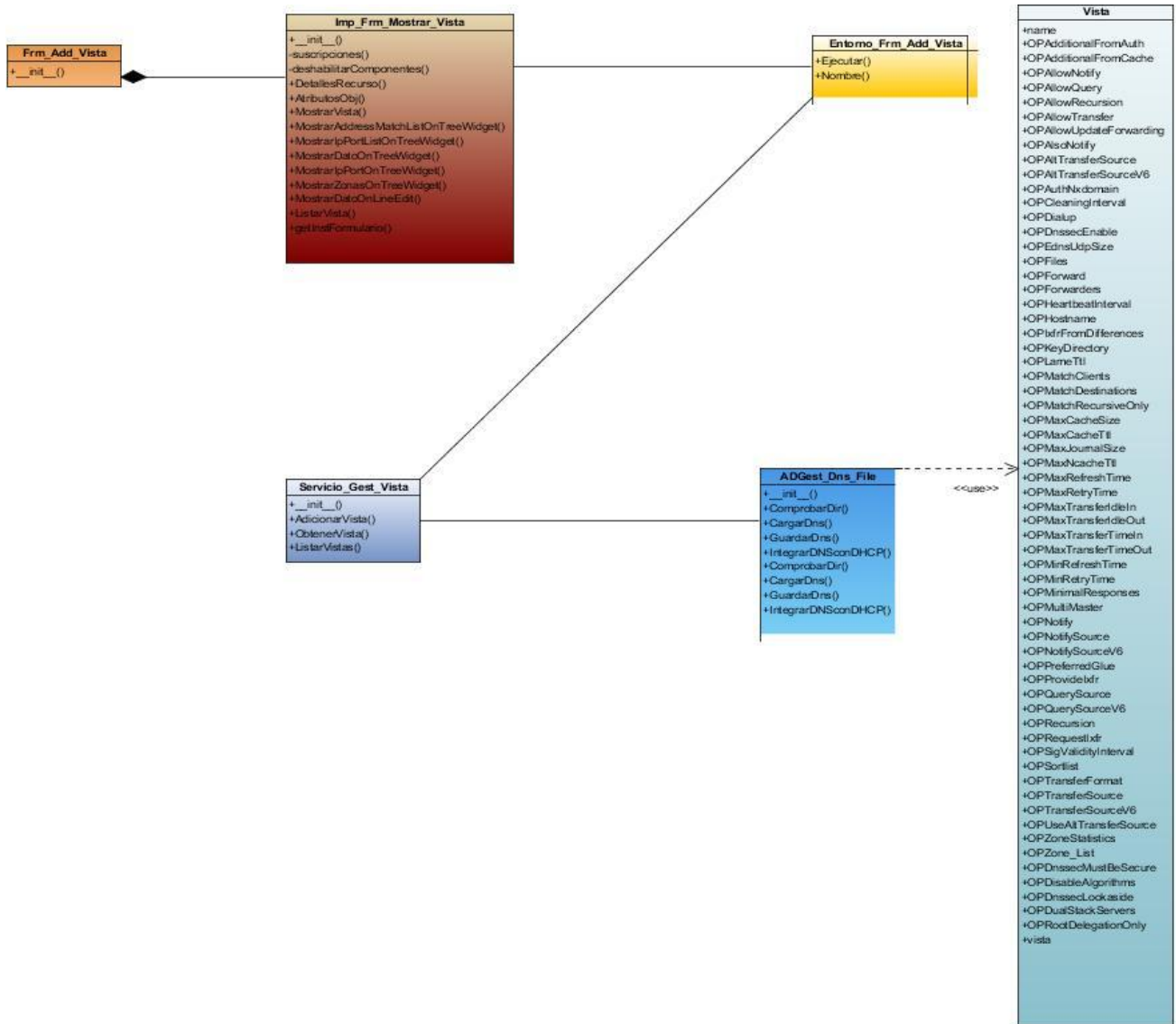


Imagen 5 Diagrama de Clase del Diseño. CU Gestionar Vista

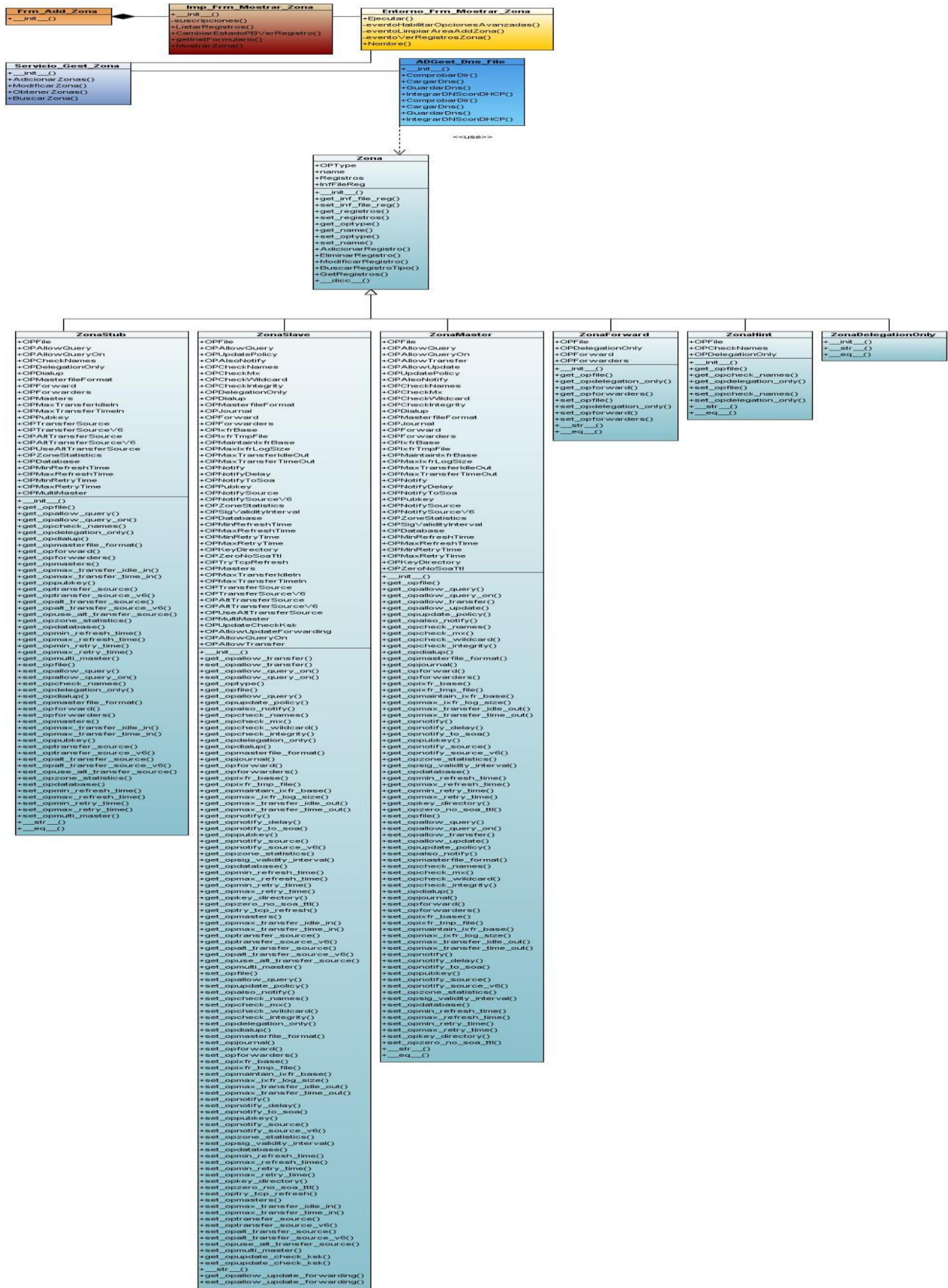


Imagen 6 Diagrama de Clase del Diseño. CU Mostrar Zona

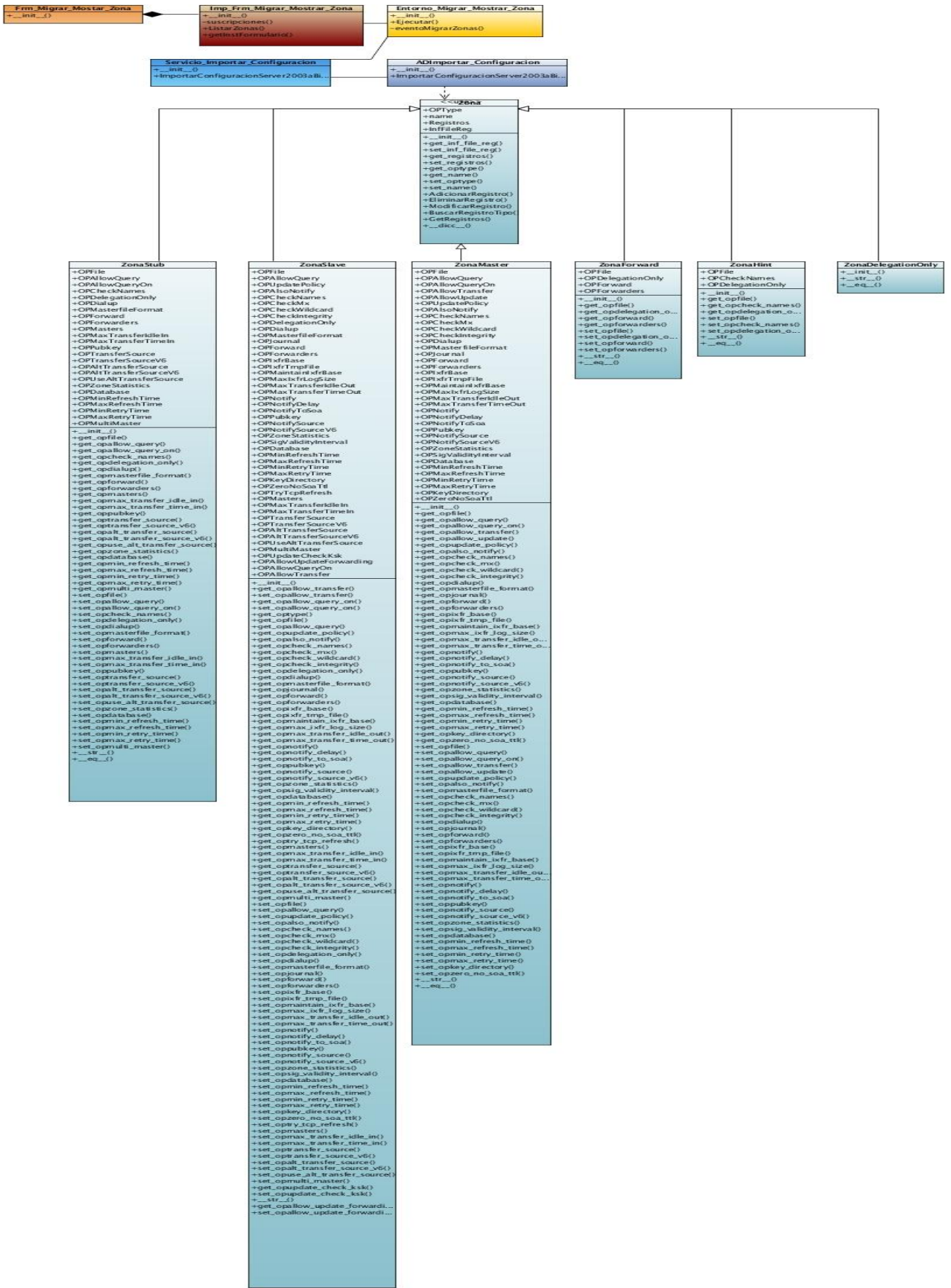


Imagen 7 Diagrama de Clase del Diseño. CU Importar configuraciones DNS

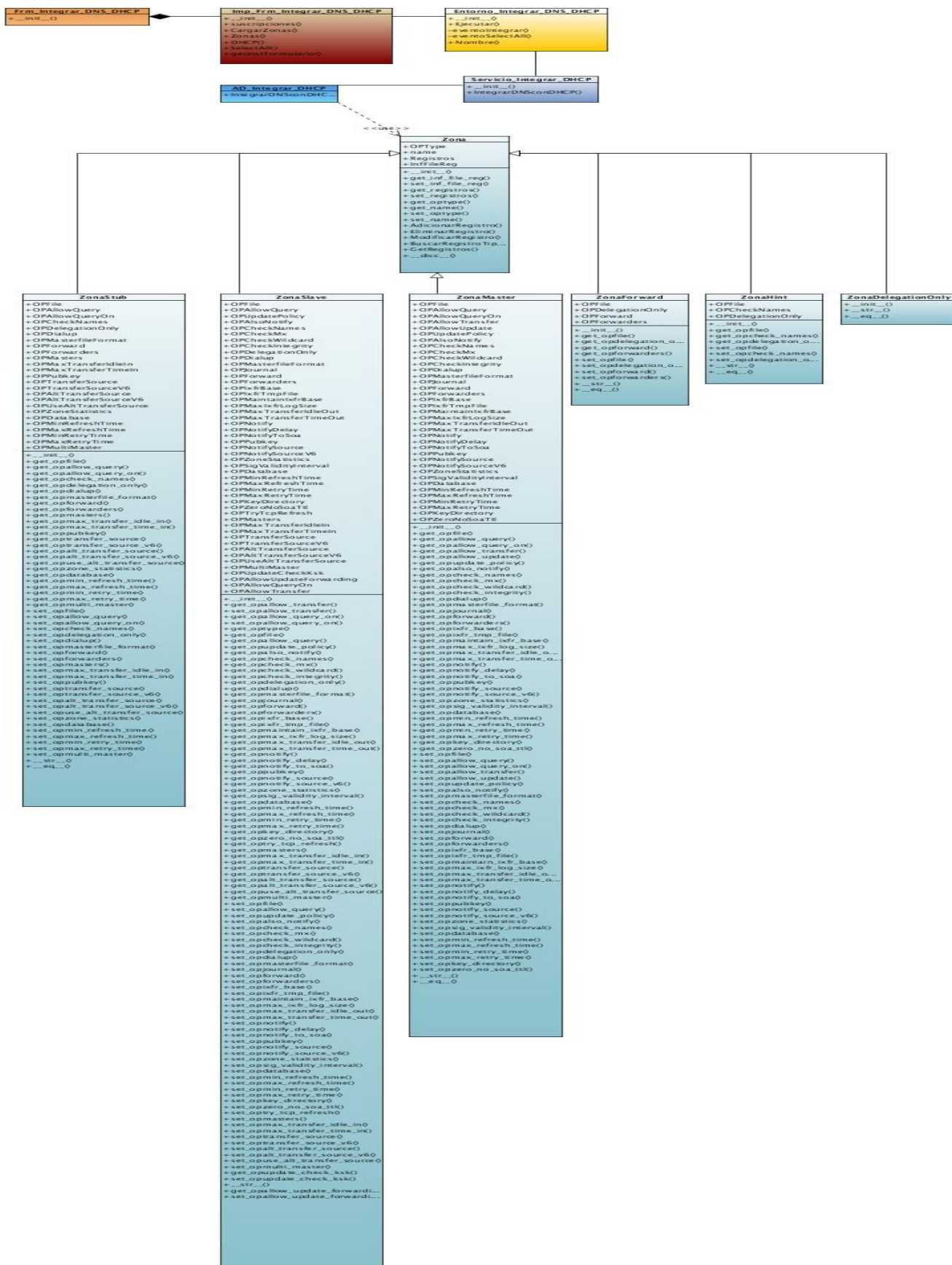


Imagen 8 Diagrama de Clase del Diseño. CU Integrar DNS con DHCP

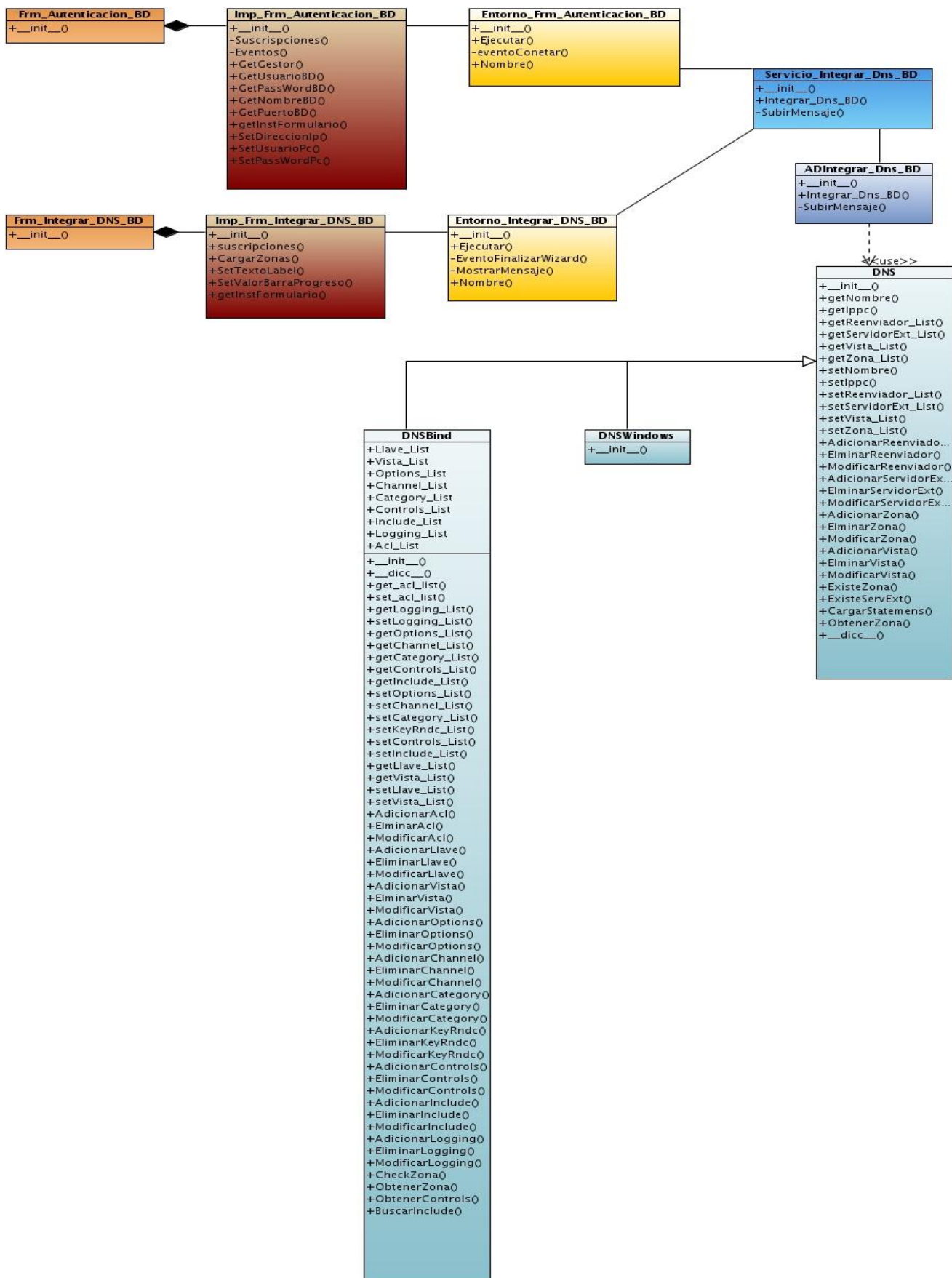


Imagen 9 Diagrama de Clase del Diseño. CU Integrar DNS con BD externa

Glosario de Términos

DNS: El Domain Name System (DNS) es una base de datos distribuida y jerárquica que almacena información asociada a nombres de dominio en redes como Internet.

GNU: GNU es un acrónimo recursivo que significa GNU No es Unix (GNU is Not Unix).

GPL: Es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software.

DHCP: DHCP (sigla en inglés de Dynamic Host Configuration Protocol - Protocolo Configuración Dinámica de Anfitrión) es un protocolo de red que permite a los nodos de una red IP obtener sus parámetros de configuración automáticamente.

SSH (Secure SHell, en español: intérprete de órdenes segura) es el nombre de un protocolo y del programa que lo implementa, y sirve para acceder a máquinas remotas a través de una red.

FTP (sigla en inglés de File Transfer Protocol - Protocolo de Transferencia de Archivos) en informática, es un protocolo de red para la transferencia de archivos entre sistemas conectados a una red TCP (Transmission Control Protocol), basado en la arquitectura cliente-servidor.

IDE o entorno de desarrollo integrado (acrónimo en inglés de integrated development environment), es un programa informático compuesto por un conjunto de herramientas de programación. Puede dedicarse en exclusiva a un sólo lenguaje de programación o bien, poder utilizarse para varios.