



Universidad de las Ciencias Informáticas
Facultad 2

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

Plataforma de Gestión de Servicios Telemáticos en GNU/Linux.
Módulo de DNS v3.0.

Autor:

Jany Ramos Mosqueda

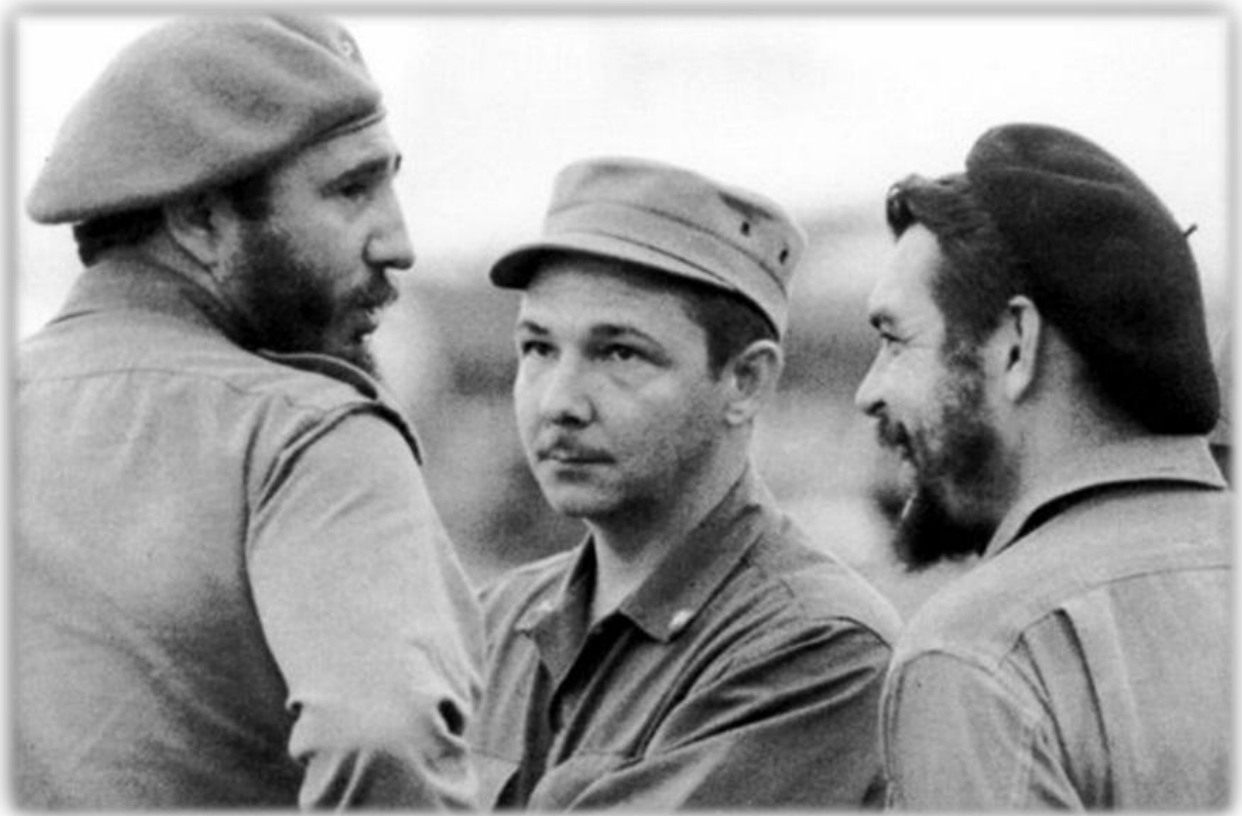
Tutor:

Ing. Ramón Alexander Anglada Martínez

Consultor:

Ing. Álvaro Luis Padilla

Ciudad de La Habana, 2011.
"Año 53 de la Revolución"



Dime y lo olvido, enséñame y lo recuerdo, involúcrame y lo aprendo.

Benjamin Franklin

Declaración de Autoría

Declaro que Jany Ramos Mosqueda es la única autora de este trabajo y autorizo a la Universidad de las Ciencias Informáticas (UCI) y a la Facultad 2 para que hagan el uso que estimen pertinente con dicho trabajo.

Para que así conste firmamos la presente a los ____ días del mes de junio de 2010

Firma del Autor
Jany Ramos Mosqueda

Firma del Tutor
Ing. Ramón Alexander Anglada Martínez

Agradecimientos

A mis padres, por ser mi mayor inspiración y ejemplo de sacrificio, por alentarme a entrar a la universidad y confiar siempre en que lo lograría, por apoyarme en cada una de mis decisiones aún cuando eran equivocadas, gracias por existir. A mi Deysi, por ser mi segunda madre, por estar atenta a todo lo que pasaba en mi vida y por considerarme su niña pequeña, por haber estado en mi vida. A mis Ide y Yoly, por vivir para mí, por ser mis madres, amigas, hermanas, profesoras y todo lo que yo necesité; sin ellas no habría podido llegar hasta aquí. A mis tíos y tías, a Iraida, por correr cada vez que pedí algo y darme su apoyo constante. A Anita y Sandra, mis hermanas incondicionales, por transmitirme su confianza en el éxito incluso cuando yo perdí la fe. A Yona y Niu, por cuidar de mí, por convivir con mis gustos y llevarse algo de ellos, por enseñarme la parte divertida de la UCI y brindarme su amistad incondicional aún cuando yo no estaba aquí para recibirla. A Jesse, por no permitirme volverme loca con el encierro en esos primeros años, por ser amigo incondicional incluso si yo nunca devuelvo las llamadas. A mis amigos de siempre, Yisso, Ráiner, Javier, Denia, Yei, Yane, Alfonso, Alberto, Lisbetti, Yasiel, por la compañía maravillosa, por escuchar mis disparates sin considerarme loca, nada será igual sin ellos. A Álvaro y Anglada, mis tutores, por crear ese mundo tan maravilloso que es Servicios Telemáticos, y mantenerlo a costa de mil esfuerzos; por todo el apoyo, las revisiones, las horas extras, las no conformidades, por ser el ejemplo que todos queremos seguir, gracias. A mis compañeros de proyecto, a Yari, Julio, Osmani, Wilfredo, Juan, Yandy, por contestar pacientemente mis preguntas y por ser unos maravillosos amigos en estos tres años que llevamos juntos. Al tribunal y a mi oponente, por las correcciones que ayudaron a hacer de este un trabajo mejor. A todos mis profesores, de cada uno aprendí algo nuevo. Gracias

Dedicatoria

A Yoly, a Deysi.

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 brinda a la hora de desarrollar y trabajar.

Cuba está totalmente comprometida y llena de restricciones en el área de la informática, 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 con la dificultad de que el conocimiento sobre el tema es escaso 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.

| | |
|---|-----------|
| INTRODUCCIÓN..... | 8 |
| CAPÍTULO 1: “FUNDAMENTACIÓN TEÓRICA” | 16 |
| 1.1 INTRODUCCIÓN..... | 16 |
| 1.2 SISTEMAS PARA LA GESTIÓN DE LOS SERVIDORES DE RED EXISTENTES EN EL MUNDO | 16 |
| 1.3 LENGUAJE DE MODELADO..... | 23 |
| 1.4 METODOLOGÍA DE DESARROLLO | 25 |
| 1.5 LENGUAJE DE PROGRAMACIÓN Y HERRAMIENTAS | 26 |
| 1.6 CONCLUSIONES..... | 29 |
| CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA | 30 |
| 2.1 INTRODUCCIÓN..... | 30 |
| 2.2 PROBLEMA Y SITUACIÓN PROBLEMÁTICA | 30 |
| 2.3 PROPUESTA DE SISTEMA..... | 31 |
| 2.4 MODELO DE NEGOCIO | 32 |
| 2.5 REQUERIMIENTOS..... | 35 |
| 2.6 MODELO DE CASOS DE USO DEL SISTEMA..... | 39 |
| 2.7 CONCLUSIONES..... | 47 |
| CAPÍTULO 3: ANÁLISIS Y DISEÑO..... | 48 |
| 3.1 INTRODUCCIÓN | 48 |
| 3.2 DISEÑO | 48 |
| 3.4 PATRONES DE DISEÑO..... | 61 |
| 3.5 CONCLUSIONES..... | 61 |
| CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBA | 62 |
| 4.1 INTRODUCCIÓN | 62 |
| 4.2 DIAGRAMAS DE COMPONENTES | 62 |
| 4.3 DIAGRAMA DE DESPLIEGUE | 70 |
| 4.4 ESTRATEGIA DE PRUEBA..... | 70 |
| 4.5 CONCLUSIONES..... | 73 |
| CONCLUSIONES..... | 74 |
| RECOMENDACIONES..... | 75 |
| BIBLIOGRAFÍA | 76 |
| GLOSARIO DE TÉRMINOS | 78 |

Índice de Figuras

| | |
|---|-----------|
| <i>Figura 1 Proceso Administrar Servidor DNS.....</i> | <i>32</i> |
| <i>Figura 2 Descripción del proceso Administrar Servidor DNS</i> | <i>33</i> |
| <i>Figura 3 Diagrama de paquetes.....</i> | <i>39</i> |
| <i>Figura 4 Modelo de Casos de Uso del Sistema</i> | <i>40</i> |
| <i>Figura 5 Prototipo de Interfaz del Caso de Uso Gestionar DNSSEC.....</i> | <i>46</i> |
| <i>Figura 6 Estructura general del Sistema</i> | <i>49</i> |
| <i>Figura 7 Estructura detallada del sistema</i> | <i>50</i> |
| <i>Figura 8 Leyenda.....</i> | <i>51</i> |
| <i>Figura 9 Diagrama de Clases del Diseño del CU Integrar con LDAP</i> | <i>52</i> |
| <i>Figura 10 Diagrama de Clases del Diseño del CU Activar DNSSEC.....</i> | <i>53</i> |
| <i>Figura 11 Diagrama de Clases del Diseño del CU Actualizar en tiempo real</i> | <i>54</i> |
| <i>Figura 12 Diagrama de Clases del Diseño del CU Gestionar ACL</i> | <i>55</i> |
| <i>Figura 13 Diagrama de Clases del Diseño del CU Gestionar Servidor Externo.....</i> | <i>56</i> |
| <i>Figura 14 Diagrama de Clases del Diseño del CU Gestionar Llaves.....</i> | <i>57</i> |
| <i>Figura 15 Diagrama de Clases del Diseño del CU Gestionar Reenviador.....</i> | <i>58</i> |
| <i>Figura 16 Diagrama de Clases del Diseño del CU Gestionar Vista</i> | <i>59</i> |
| <i>Figura 17 Diagrama de Clases del Diseño del CU Gestionar Zona.....</i> | <i>60</i> |
| <i>Figura 18 Diagrama General de Componentes.....</i> | <i>63</i> |
| <i>Figura 19 Diagrama de Componentes del CU Activar DNSSEC</i> | <i>64</i> |
| <i>Figura 20 Diagrama de Componentes del CU Integrar con LDAP.....</i> | <i>64</i> |
| <i>Figura 21 Diagrama de Componentes del CU Actualizar en Tiempo Real.</i> | <i>65</i> |
| <i>Figura 22 Diagrama de Componentes del CU Gestionar ACL.....</i> | <i>65</i> |
| <i>Figura 23 Diagrama de Componentes del CU Gestionar Llave.....</i> | <i>66</i> |
| <i>Figura 24 Diagrama de Componentes del CU Gestionar Reenviador.</i> | <i>67</i> |
| <i>Figura 25 Diagrama de Componentes del CU Gestionar Servidor Externo.</i> | <i>68</i> |
| <i>Figura 26 Diagrama de Componentes del CU Gestionar Zona.</i> | <i>69</i> |
| <i>Figura 27 Diagrama de Despliegue.....</i> | <i>70</i> |

Índice de Tablas

| | |
|--|----|
| TABLA 1. PERSONAS QUE INTERVIENEN EN LOS PROCESOS DEL NEGOCIO..... | 33 |
| TABLA 2. LISTADO DE CASOS DE USO..... | 42 |
| TABLA 3. ACTORES DEL SISTEMA..... | 44 |
| TABLA 4. DATOS DEL CASO DE USO GESTIONAR DNSSEC..... | 45 |
| TABLA 5. DESCRIPCIÓN DEL CASO DE USO GESTIONAR DNSSEC..... | 45 |
| TABLA 6. DATOS DEL CASO DE USO INTEGRAR CON LDAP..... | 48 |
| TABLA 7. DESCRIPCIÓN DEL CASO DE USO INTEGRAR CON LDAP..... | 48 |
| TABLA 8. ESTRATEGIA DE PRUEBAS..... | 77 |
| TABLA 9. PRUEBAS AL CASO DE USO ACTIVAR DNSSEC..... | 77 |
| TABLA 10. NO CONFORMIDADES..... | 79 |

Introducción

El incremento del uso de las tecnologías se ha convertido en una realidad para el mundo. La aparición de Arpanet en 1990 y luego su expansión como Internet con el surgimiento de la World Wide Web, trajo consigo servicios como Mensajería Instantánea, Correo Electrónico, Publicación de Sitios Web, Recursos Compartidos, Videoconferencias, Sistema Global de Navegación por Satélite, Consultas Online, que hoy constituyen opciones igual de cotidianas como la comunicación vía telefónica, a pesar de que hace 40 años las computadoras eran sólo asunto de estudiantes universitarios y expertos. Debido a esta fusión entre la informática y las comunicaciones nace "(...) la Telemática como concepto para explicar el trabajo conjunto y organizado de especialistas informáticos, electrónicos y de telecomunicaciones para desarrollar y mantener plataformas de adquisición, distribución, almacenamiento y procesamiento de datos, y generar herramientas de gestión y telecontrol de procesos, basados en la estadística y procesamiento de la información"¹. El uso de los servicios telemáticos, por tanto, garantiza la disponibilidad, autenticidad y confidencialidad de dicha información; los mismos incluyen: consultas de datos, transmisión de imágenes, datos, voz, comunicación instantánea, entre otros.

Para brindar estos servicios el administrador de red, debe diseñar, instalar, configurar, y dar soporte al servicio, cuidando con cada cambio a realizar, que la seguridad del sistema no se vea comprometida, conociendo además que muchas de las aplicaciones diseñadas para gestionar los servicios poseen vulnerabilidades.

Con el crecimiento de los servicios se hace necesario instalar más servidores, como ejemplo podemos tomar un servidor FTP (Protocolo de Transferencia de Ficheros, del inglés File Transfer Protocol), que permite manipular de manera remota un sistema de archivos, útil para compartir grandes volúmenes de información y un servidor Proxy HTTP, servidor intermediario entre dos sistemas, mayormente se utiliza para controlar y monitorizar el uso de Internet por parte de los usuarios de una red.

¹ *Wamtech (Wireless and Mobile Technologies), 2010*

Sin embargo, hay otro grupo de servicios que es necesario instalar para que funcione correctamente la red; tal es el caso de DNS (Sistema de Nombres de Dominio, del inglés Domain Name System), encargado de relacionar los nombres de dominio con su dirección IP de manera jerárquica.

Un DNS no es más que un sistema que utiliza una base de datos distribuida y jerárquica, cuya función más conocida es asociar las direcciones IP con que funciona Internet, con un nombre de dominio. Se basa en un árbol que define la jerarquía entre dominios y subdominios, cada uno de los cuales ha de tener un servidor primario aunque es recomendable tener además, un servidor secundario como mínimo. Por lo general el programa para servidor de nombres que se utiliza para los sistemas GNU/Linux es Bind (Dominio de Nombres de Internet de Berkeley, del inglés Berkeley Internet Name Domain). Su funcionamiento se basa en tres componentes fundamentales: un cliente, que no es más que la aplicación que realiza la consulta desde un host de no tener la información guardada en la memoria temporal o caché; un servidor, que se encargará de solicitar la información al tercer componente: las zonas. De no encontrar la respuesta en sus zonas autorizadas, el servidor busca la información en su caché, realiza el proceso de resolución, el cual implica otros servidores DNS. Este proceso puede llevarse a cabo de dos formas: recursiva o iterativa. La forma recursiva realiza la búsqueda preguntando a otros servidores autorizados hasta obtener la dirección y devuelve la misma. Una búsqueda iterativa, de no tener en sus manos la respuesta consulta a los servidores autorizados y en lugar de devolver la respuesta devuelve un mensaje con la dirección del servidor que posee la respuesta, con una lista de recursos o una respuesta de no encontrado. La opción establecida por defecto es la resolución recursiva; sin embargo, la resolución iterativa permite realizar consultas adicionales. Internet es una red en constante crecimiento, donde comienzan a agotarse las direcciones IP disponibles. El establecimiento de la próxima generación de direcciones, con un formato de ocho grupos de cuatro dígitos hexadecimales, reafirma la necesidad de los servidores DNS, pues recordar para cada sitio que visitemos, su dirección, sería casi imposible, además de incómodo y poco práctico.

Mientras mayor sea la cantidad de servicios que desee brindar una entidad, mayor será la dificultad de su administrador de red para mantenerlos, puesto que cada uno necesita de una configuración y la seguridad

de cada uno se puede tratar de manera distinta, comenzando por la autenticación -un usuario y clave distinto para cada servidor-.

Afortunadamente se han diseñado herramientas que gestionan de manera centralizada, la administración de los servicios. Éstas se basan mayormente en la integración de cada uno de los servicios y la inclusión de un módulo que gestione la seguridad para el sistema en su totalidad. Actualmente la más difundida de estas herramientas en Cuba es Windows Server, la cual proporciona una amplia gama de servicios fáciles de implementar y administrar. No obstante, la producción de software se incrementa y el bloqueo no permite comercializar productos de la Isla si para desarrollarlos se utiliza software cuyos derechos pertenezcan a una empresa privada.

“Dentro de la estrategia de migración por la cual Cuba ha optado, hay algunas instituciones que impulsan este desarrollo tecnológico, entre ellas están el Ministerio de la Informática y las Comunicaciones (MIC), dentro de esta la Oficina para la Seguridad de la Redes Informáticas (OSRI) y la Universidad de las Ciencias Informáticas (UCI)”².

El software libre ha derivado como una alternativa confiable y económica, que cada vez gana en seguidores y cuya documentación es abundante y fácil de hallar, incluso es posible obtener soporte técnico. No obstante, no existe una integración total entre las herramientas que gestionan estos servicios para GNU/Linux para un funcionamiento del sistema como un todo; un ejemplo de esto es la falta de integración del servicio DNS con el servicio de LDAP.

LDAP, o Protocolo Ligero de Acceso a Directorios por sus siglas en inglés, es un protocolo de acceso unificado a un servicio de directorio, en el cual se encuentran objetos de la red organizados de manera jerárquica. Dichos objetos pueden representar personas, impresoras, documentos, permisos o cualquier otro elemento que represente una entrada o múltiples entradas en el árbol.

Dada la carencia de una plataforma profesional y accesible que integre los servicios esenciales para una red en sistemas operativos GNU/Linux surge en el 2008 en la Facultad 2, el proyecto Servicios Telemáticos, con el objetivo de crear una plataforma de servicios integrados que ofreciera la posibilidad de

² Noa Cabrera, Sandy; “Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo de Directorio”; p. 2.

administrar los servicios LDAP, DNS y DHCP de manera eficiente, centralizada, intuitiva y segura; colaborando de este modo con la migración hacia el software libre en nuestro país.

DHCP, o Protocolo de Configuración Dinámica de Servidor, es un protocolo de red que permite a los administradores gestionar de forma dinámica y automatizada, la dirección IP de los nodos de una red. Este protocolo envía automáticamente los parámetros de configuración de red a cada ordenador que envía un mensaje de solicitud, de modo que se conoce en todo momento la dirección IP de cada nodo conectado mediante este servicio facilitando así su supervisión.

En pasados cursos 2008-2009 y 2009-2010 se avanzó en el cumplimiento de esta misión, desarrollando como temas de tesis³, los módulos DNS y LDAP en sus versiones uno y dos, las mismas cuentan con las funcionalidades básicas; sin embargo, aún carecen de importantes funcionalidades para su seguridad y para lograr su integración. Específicamente el módulo DNS carece de soporte, a nivel de aplicación, para el protocolo IPv6, el cual se encuentra actualmente en expansión y aunque no tan estable con el IPv4, posee numerosas ventajas y existen mecanismos que le permiten interactuar con el protocolo IPv4; soporte DNSSEC (DNS Security, Seguridad del DNS) para la transferencia segura de objetos y la sincronización de servidores; no realiza la actualización en tiempo real de los datos de la aplicación, la cual posibilita visualizar desde el servidor los cambios realizados en un lugar ajeno al mismo; no se integra con el Servidor de Directorio, la cual facilitaría el proceso de administración en la medida de que los registros del DNS se puedan guardar en la base de datos del directorio y las reglas del directorio se pueden aplicar a los servicios del DNS.

Teniendo en cuenta lo anteriormente expuesto, se continúa trabajando en el desarrollo de dichas aplicaciones, para lo cual se incorporarán las funcionalidades necesarias para crear la plataforma de servicios con las características de una red empresarial cubana capaz de funcionar en un sistema GNU/Linux, dado lo cual se plantea como **problema científico**:

³ Noa Cabrera, Sandy; "Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo de Directorio". Gualdarrama Reyes, Jorge Luis; "Plataforma de Gestión de los Servicios Telemáticos en GNU/Linux. Módulo DNS". De la Rosa Pasteur, Jenny; Zulueta Sarria, Edgardo; "Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo de Directorio v2.0". Pérez Hurtado, Alexei; Padilla Moya, Álvaro Luis; "Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo DNS v2.0"

¿Cómo incrementar las funcionalidades del módulo DNS de la Plataforma de Gestión de Servicios Telemáticos para GNU/Linux?

Por tanto el **objetivo general** del presente trabajo es añadir nuevas funcionalidades al módulo DNSv2.0 de la Plataforma de Gestión de Servicios Telemáticos. En la búsqueda de un resultado constituyen **objeto de estudio** los procesos relacionados con la administración de servicios telemáticos, tomando como **campo de acción** los procesos que garanticen la administración de un servidor DNS Bind9.

Para ello los **objetivos específicos** son

1. Realizar una investigación sobre posibles soluciones existentes, estudiar las características de cada cual, así como de las herramientas y métodos que puedan ser útiles en la búsqueda de un resultado.
2. Elaborar una propuesta de sistema.
3. Determinar requerimientos funcionales y no funcionales.
4. Realizar diagramas de Caso de uso.
5. Definir patrones arquitectónicos y de diseño.
6. Implementar la integración con LDAP.
7. Implementar la actualización en tiempo real.
8. Implementar la gestión de los registros pertinentes para la transferencia de objetos del DNS.
9. Implementar la sincronización de servidores DNS para la transferencia de objetos del DNS.
10. Implementar el caso de uso Ipv6
11. Realizar pruebas a la aplicación

Para lograr el objetivo general se definen las siguientes **tareas de la investigación** por objetivo específico:

Objetivo 1:

- a. Realización de una búsqueda y selección de las herramientas a utilizar a través del estudio de las mismas.

- b. Selección de la metodología a seguir en el curso de la investigación.
- c. Realización de un estudio de las características de Bind9 como servidor DNS.

Objetivo 3:

- a. Realización de un estudio de la versión anterior del módulo.

Objetivo 5:

- a. Elaboración de diagramas de clases del diseño.

Objetivo 6:

- a. Prueba de los métodos de integración con el servidor LDAP.
- b. Implementación de la integración con LDAP.

Objetivo 7:

- a. Pruebas de las vías de notificación del sistema.
- b. Implementación de la actualización en tiempo real

Objetivo 8:

- a. Investigación del sistema de transferencia de objetos del DNS, con llave pública y privada.

Objetivo 9:

- a. Creación de una cadena de confianza mediante DNSSEC en un servidor con BIND9.

Objetivo 10:

- a. Investigación de las librerías de Python que gestionan el protocolo IPv6.

Objetivo 11:

- a. Definición de una estrategia de prueba
- b. Definición de no conformidades
- c. Solución de no conformidades

Constituye la **idea a defender**, que con la incorporación de una nueva versión del módulo DNS a la Plataforma de Gestión de Servicios Telemáticos, que permita la integración con el servidor de Directorio y añada otras funcionalidades como la gestión de IPv6, transferencia segura de objetos y la actualización en

tiempo real de los datos del DNS se podrá mejorar sustancialmente la Plataforma de Gestión de Servicios Telemáticos en GNU/Linux que permita a los administradores de red cubanos brindar un servicio profesional, seguro y de calidad.

Para llevar a cabo las tareas propuestas, se utilizan los siguientes métodos de la investigación:

- ✿ Analítico-Sintético: al analizar las funcionalidades del Bind9 que no se encuentran disponibles en la aplicación y que se desean añadir, las librerías de Python que pueden resultar útiles para el desarrollo del módulo, al descomponer el sistema de transferencia de datos del DNS y la gestión de los registros del DNS; con el objetivo de determinar el funcionamiento del sistema y comprender cómo añadirle las funcionalidades necesarias de manera óptima y sin perjudicar su funcionamiento actual.
- ✿ Histórico-Lógico: al realizar un estudio de las características actuales del protocolo IPv6, de DNSSEC, del estado de la integración DNS-LDAP en otras herramientas que pudieran comprender una solución o brindar una guía para definir los requisitos funcionales.
- ✿ Observación: utilizado al realizar pruebas con las versiones existentes, observando sus fallos y funcionamiento correcto, lo cual permite obtener una visión general de las características de estas herramientas.

Para documentar dicho proceso se realiza este documento cuya estructura se detalla a continuación:

- ✿ **Capítulo 1.** Fundamentación teórica: incluye los aspectos teóricos que soportan este proyecto y se analizan las aplicaciones que existen en la actualidad para la administración de servidores DNS y de servicios; además de las herramientas y lenguajes de programación más utilizadas en el mundo para el desarrollo de las aplicaciones, para definir a partir de la información obtenida, las que se usarán para realizar la aplicación. Se plantea la metodología a seguir en la elaboración de la misma.
- ✿ **Capítulo 2.** Características del sistema: en este capítulo se definen las reglas del negocio y los

conceptos fundamentales del Módulo de DNS. Se determinan además los requerimientos funcionales y no funcionales del mismo, agrupándolos en casos de usos y se da una idea general del funcionamiento de esta aplicación.

- ✿ **Capítulo 3.** Análisis y diseño del sistema: capítulo donde se determinan las clases que se utilizarán en la implementación del sistema y la relación entre ellas. Además se muestra todo el proceso de obtención de la Base de Datos y se definen los principios de diseño de interfaz, codificación y tratamientos de errores.
- ✿ **Capítulo 4.** Implementación y Pruebas: finalmente se define el hardware sobre el que se sustenta la aplicación y sus componentes, se verifica su integridad y ajuste a los requerimientos planteados.

Capítulo 1: “Fundamentación teórica”

1.1 Introducción

Hoy día el acelerado desarrollo de las tecnologías de la informática y las telecomunicaciones trae como consecuencia que las redes de cómputo se vuelvan cada vez más complejas. Éstas soportan numerosas aplicaciones y servicios estratégicos para las organizaciones en las que se implementan, y la exigencia de su correcto funcionamiento es sumamente solicitada por sus usuarios.

En este capítulo se definen los principales conceptos asociados al tema de la administración de servicios mediante una plataforma y específicamente de un servidor de nombres de dominio. Se realiza un estudio sobre el estado actual de las herramientas que manejan una posible solución y sobre las herramientas y metodologías propuestas para desarrollar la versión 3.0 del módulo DNS.

1.2 Sistemas para la gestión de los servidores de red existentes en el mundo

Windows Server 2003:⁴

Quizás el más popular de los Windows Servers, toma lo mejor de la tecnología de Windows Server 2000, proporcionando una amplia plataforma fácil de implementar y administrar, contando además con una amplia compatibilidad de hardware. Entre las funcionalidades que posee, se encuentran:

*Servidor de Seguridad de conexión a Internet.

*IIS 6.0

*Active Directory

*Servidor de Acceso Remoto

*Terminal server

⁴ <http://www.microsoft.com/spain/windowsserver2003/technologies/networking/nap/default.aspx>

*Servidor de aplicaciones web, que integra el framework .net y Servidor web.

*Servidor de archivos e impresión.

*Sistema de dominio (DNS), y servidor DHCP.

*Servidor de transmisión de multimedia en tiempo real (Streaming)

Windows Server 2003 cuenta con características que lo hacen la plataforma más utilizada en Cuba.

* Disponibilidad.

* Escalabilidad.

* Fiabilidad.

Su popularidad se debe esencialmente a que:

- ❖ Proporciona fiabilidad, disponibilidad, y escalabilidad para que usted pueda ofrecer la infraestructura de red que los usuarios solicitan.
- ❖ Ayuda a administrar su red proactivamente al reforzar las políticas, tareas automatizadas y simplificación de actualizaciones.
- ❖ Proporciona herramientas flexibles que ayuden a ajustar su diseño e implementación a sus necesidades organizativas y de red.
- ❖ Proporciona un servidor Web integrado y un servidor de transmisión de multimedia en tiempo real para ayudarle a crear más rápido, fácil y seguro una Intranet dinámica y sitios de Internet.
- ❖ Proporciona un servidor de aplicaciones integrado que le ayude a desarrollar, implementar y administrar servicios Web en XML más fácilmente.

No obstante, es uno de los Windows Servers con más bugs o fallos de seguridad conocidos. Además, no implementa el protocolo IPv6, ni las extensiones de seguridad del DNS.

Windows Server 2008 R2:⁵

A pesar de estar basado en la tecnología de Windows NT 6.x y de tomar algunas características de

⁵ <http://www.microsoft.com/windowsserver2008/es/xl/default.aspx>

Windows Server 2003, no se trata de una continuación del mismo, sino de toda una nueva plataforma, que incluye mejoras en el soporte DNS como:

- ✿ Carga de zonas en background: los servidores DNS que alojan zonas de DNS muy grandes y que se guardan dentro de Servicios de Dominio del Directorio Activo (ADDS), pueden responder a las peticiones de los clientes de forma más rápida porque la información de la zona se efectúa en un proceso de carga en segundo plano;
- ✿ Soporte para IPv6 (IP versión 6): El servicio de servidor DNS ahora dispone de soporte completo para las direcciones IP más largas de la especificación IPv6; Soporte para Controladores de Dominio de Solo-Lectura (RODC): el rol de Servidor DNS dispone de zonas primarias en modo de solo-lectura en servidores RODC;
- ✿ Nombres globales únicos: la zona GlobalNames permite la resolución de nombres con identificación exclusiva en grandes redes corporativas donde no se utiliza el servicio WINS (Windows Internet Name Service). La zona GlobalNames es muy útil cuando no resulta viable o práctico el uso de sufijos de nombre DNS para la resolución de nombres de máquina exclusivos en la red.

No obstante a constituir un gran paso de avance respecto a la versión 2003, continúa la imposibilidad de gestionar DNSSEC, además, no permite su administración remota, lo cual va en detrimento de las opciones de movilidad que se desea en las aplicaciones actuales.

“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.”⁶

⁶ Padilla Moya, Álvaro Luis; Pérez Hurtado, Alexei; *Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo*

Red Hat Enterprise Linux (RHEL)⁷

Es una distribución para servidores Linux, dirigida hacia el mercado, por tanto su copia y distribución no están permitidas, aunque existen versiones con cambios insignificantes como CentOS. Con un largo historial que habla de eficiencia, incluye entre sus servicios:

- ✿ Almacenamiento y virtualización de servidores extendidos.
- ✿ Herramientas de desarrollo y rendimiento.
- ✿ Conjunto completo de aplicaciones de código abierto del servidor, incluyendo: Apache, Samba, NFS, FTP, Tomcat, MySQL, PostgreSQL y servidores de red.
- ✿ Mejoras a IPSec para una mayor seguridad y rendimiento.
- ✿ Soporte para IPv6 y mejoras de conformidad.
- ✿ RHEL está disponible en las arquitecturas:
 - ✓ i386
 - ✓ AMD64/Intel64
 - ✓ System z
 - ✓ IBM Power (64-bit)

Entre sus mejoras se encuentran:

- ✿ Apoyo al desarrollo y tiempo de ejecución: mejoras al SystemTap, ABRT es un nuevo marco para la simple recopilación y reportes de errores, y se realizan las mejoras a GCC (versión 4.4.3), glibc (versión 2.11.1), y GDB (versión 7.0.1).
- ✿ Incluye además el Directory Server, que es un servidor basado en LDAP que centraliza configuración de aplicaciones, perfiles de usuarios, información de grupos, políticas así como información de control de acceso dentro de un sistema operativo independiente de la plataforma.

DNS v2.; 2009; p. 14.

⁷ <http://www.redhat.com/rhel/>

- ✿ Forma un repositorio central para la infraestructura de manejo de identidad, Red Hat Directory Server simplifica el manejo de usuarios, eliminando la redundancia de datos y automatizando su mantenimiento.

“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.”⁸

Posee una versión gratuita, sin embargo, no implementa todas sus funcionalidades. Además, carece incluso en la versión comercial, de las mismas funcionalidades de Windows Server 2008 R2, ya que no brinda la posibilidad de utilizar DNSSEC, ni la opción de administrar desde otra locación el servidor, y por tanto, se debe administrar sólo un servidor DNS a la vez.

SUSE Linux Enterprise Server⁹

Es una distribución de Linux suministrado por Novell y orientada al mercado empresarial. Está dirigida a servidores, mainframes, y estaciones de trabajo, pero se puede instalar además en computadoras de escritorio para realizar pruebas. La versión actual es SLES 11 Service Pack 1, lanzado 02 de junio 2010, que se desarrolla a partir de una base de código común con SUSE Linux Enterprise Desktop y otros productos SUSE Enterprise Linux. Cuenta con asistencia técnica de Novell y Microsoft, y entre sus servicios se encuentran:

- ✿ Servicios integrados de redes y protocolos, incluyendo CUPS, DNS, DHCP, IMAP, NTP, SLP, Postfix, PXE, Proxy, Samba, SNMP, SMTP, entre otros.
- ✿ Servicios de aplicaciones y bases de datos como Apache, Tomcat, MySQL y PostgreSQL y

⁸ Gualdarrama Reyes, Jorge Luis; *Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo DNS; 2009; p. 10.*

⁹ http://es.opensuse.org/Bienvenidos_a_openSUSE.org

soluciones populares de entre cientos de software independientes.

- ✿ Incluye herramientas de administración que permiten manejar servicios como DNS y DHCP de manera automática e incluye plug-ins para terceras aplicaciones.
- ✿ Soporte de múltiples kernels.
- ✿ SLES 11 permite ejecutar aplicaciones asp.net en Linux, y ofrece más aplicaciones certificadas que ninguna otra distribución Linux.

A su favor cuenta con las siguientes características:

- ✿ Es la única distribución de Linux recomendada por Microsoft y SAP
- ✿ Tecnología de virtualización incorporada
- ✿ Potentes herramientas de administración mediante la integración con el conjunto de herramientas Novell ZENworksOrchestration
- ✿ Asistencia técnica de Novell y Microsoft
 - ✓ Más de 900 técnicos de Linux certificados y entrenados
 - ✓ Acuerdos de nivel de servicio cumplidos en más del 99% de los incidentes
 - ✓ Respuesta inferior a 5 minutos en el 65% de los incidentes
- ✿ Seguridad por niveles con la aplicación Novell AppArmor®
- ✿ Hasta 10 TB de memoria física y 1024 CPU
- ✿ Soluciones de clúster de alta disponibilidad incorporadas

“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.”¹⁰

Al ser un software de licencia GPL, las modificaciones que se realicen sobre el código de la versión

¹⁰ Galdarrama Reyes, Jorge Luis; *Plataforma de Gestión de Servicios Telemáticos en GNU/Linux. Módulo DNS; 2009; p. 9.*

adquirida para ajustarlo al sistema de cómputos de la isla, deben ser notificadas a la empresa propietaria del software, lo cual dada la situación política de Cuba y Estados Unidos, no es una opción a tener en cuenta. Posee la desventaja de no brindar soporte para el protocolo IPv6, ni posibilitar activar las extensiones de seguridad para el DNS. Por tanto, queda aún por debajo de las opciones anteriores.

Webmin¹¹

Es una herramienta de administración del sistema, accesible mediante interfaz web para OpenSolaris, GNU/Linux y otros sistemas Unix que además es accesible desde la consola. Puede ser instalada en Windows y MacOS. Posee versiones de distribución libre pero su licencia pertenece a BSD (Distribución de Software Berkeley, del inglés Berkeley Software Distribution). El hecho de ser una aplicación modular, la convierte en una opción flexible, práctica y con numerosas funcionalidades, entre las que se encuentran:

- ✿ Posibilidad de modificar y controlar muchas aplicaciones de código abierto, como el Servidor Web, PHP, MySQL, DNS, SSH, LDAP, DHCP, FTP, entre otros.
- ✿ Ampliaciones Modulares: Además de las distintas funciones que integra, es posible sumar numerosos complementos opcionales orientados a configurar y administrar todo tipo de programas y servicios
- ✿ Monitoreo del tráfico de red.

Sin embargo carece de la posibilidad de administración remota, y por tanto de la posibilidad de administrar más de un servidor a la vez; además, su integración con el servidor LDAP no tuvo pruebas exitosas, a pesar de ser anunciada no es una característica estable y funcional, siendo esta su mayor desventaja puesto que ello imposibilitaría su integración a la plataforma de gestión deseada.

¹¹ <http://www.webmin.com>

1.3 Lenguaje de modelado

Un lenguaje de modelado es un lenguaje creado para visualizar, especificar, construir y documentar los elementos de un sistema de software.

1.3.1 UML

UML (Unified Modeling Language) es un lenguaje de modelado para sistemas de software orientados a objetos. Está compuesto por diversos elementos gráficos que mediante relaciones, se combinan para conformar diagramas que esencialmente se clasifican en Diagramas de Comportamiento, Diagramas de Implementación y Diagramas de Estructura Estática. Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementar dicho sistema, de modo que no constituye una guía para realizar el análisis del sistema. UML es el primer método en publicar un modelo en su propia notación, incluyendo la notación para la mayoría de la información de requisitos, análisis y diseño, ya que es una consolidación de muchas de las notaciones y conceptos más utilizados en la orientación a objetos: Grady Booch, Ivar Jacobson y Jim Rumbaugh. Fue seleccionado puesto que puede ser utilizado para modelar casi cualquier tipo de sistema: sistemas de hardware, sistemas de software y organizaciones del mundo real, incluyendo todos los artefactos definidos por RUP. Además, su independencia del ciclo de desarrollo, posibilita su uso tanto en un tradicional ciclo en cascada, como en un ciclo en espiral. En resumen, el Lenguaje de Modelado Unificado, permite:

- ✿ Representar visualmente las reglas de creación, estructura y comportamiento de un grupo relacionado de objetos y procesos.
- ✿ Visualizar de forma eficiente la complejidad de un sistema u organización en un reducido número de diagramas.
- ✿ Mantener más ágilmente las especificaciones ante los cambios y nuevas actualizaciones de arquitectura.
- ✿ Matricular proyectos

- ✓ Definir misión, objetivos y alcance
- ✓ Elaborar modelos de referencia
- ✓ Estimar los recursos
- ✓ Analizar los riesgos
- ✓ Establecer un plan director
- ✿ Planificar proyectos
 - ✓ Plan de producción de software basado en casos de uso
 - ✓ Plan de pruebas de testing basado en casos de test
 - ✓ Plan de certificación de entregables en un ciclo de iteraciones

1.3.2 IDEF0

IDEF0 (Integrated Definition Methods) es una técnica de modelado que permite representar de forma estructurada las actividades que conforman un sistema o empresa, y los objetos o datos que soportan la interacción de dichas actividades. El IDEF0 es muy utilizado para describir procesos de negocio (atendiendo a los objetivos centrales) y existen numerosas aplicaciones de software que apoyan su desarrollo. Un modelo IDEF0 se compone de una serie de diagramas que muestra los procesos fundamentales que sustentan la organización, en sus vistas superiores, y organizados de manera jerárquica, permiten describir detalladamente las funciones especificadas en el nivel superior, a medida que se desglosan. Los elementos gráficos utilizados son cuadros y flechas. Fue seleccionado como técnica para modelar el negocio por su facilidad para mostrar una actividad; su sencillez y organización dejan claro el resultado de cada actividad, así como los medios necesarios para llevarla a cabo. Ello es evidente en la descripción de cada proceso pues se toman en cuenta cinco elementos básicos:

- ✿ Procesos o actividades
- ✿ Entradas(insumos)
- ✿ Controles

- ✿ Mecanismos para la realización de tareas
- ✿ Salidas o resultados conseguidos en el proceso

Ventajas de IDEF0 para modelar procesos de negocio

- ✿ Permite representar todo el proceso cronológicamente. Describe el flujo orientado al cliente final pasando por cada una de las actividades de la organización que permiten dar cumplimiento a la solicitud.
- ✿ Utiliza una notación muy fácil de comprender de modo que cualquier actor del negocio pueda utilizarla para describir su función en el mismo.
- ✿ Permite incorporar en el flujo los datos que entran y salen de las actividades, así como las reglas del negocio y los actores, todo en la misma vista. Permite además representar el flujo de datos que sale de una actividad como entrante de otra actividad y descomponer una actividad como un proceso a su vez.
- ✿ Permite descubrir problemas de organización en el negocio que deben ser arreglados; al crear una estructura perfectamente jerarquizada posibilita encontrar fallos en la estructura del negocio, y organizarlo para luego informatizarlo.

1.4 Metodología de desarrollo

En un proyecto de desarrollo de software la metodología define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. No existe una metodología de software universal. Las características de cada proyecto exigen que el proceso sea configurable.

1.4.1 RUP

“El Proceso Unificado de Rational (del inglés Rational Unified Process, RUP) es un proceso de desarrollo de software que junto al Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a

objetos. Basa sus actividades en cuatro fases interceptadas por nueve flujos, de los cuales de los cuales los 6 primeros son conocidos como flujos de ingeniería y los tres últimos como de apoyo. Sus principales características lo definen como definido por casos de uso, lo cual permite visualizar todo el tiempo lo que el usuario quiere y en base a ello definir los esfuerzos; centrado en la arquitectura, lo cual constituye un eslabón fundamental para la construcción de un sistema robusto y para la toma de decisiones de cómo será construido el mismo; e iterativo e incremental, para el cual se divide el proyecto en pequeñas partes o metas, en las cuales se puede ir evaluando el proceso y permite volver atrás con mayor facilidad en caso de fallo. RUP permite definir roles, los cuales fijan a una persona según las funciones que realiza o el papel que desempeña en un momento determinado. Además incluye artefactos del proceso, resultados de los análisis que constituyen a la vez guía y documentación, entre ellos: los modelos mencionados a continuación”.

- Modelo de Casos de Uso del Negocio
- Modelo de Objetos del Negocio
- Modelo de Casos de Uso
- Modelo de Análisis
- Modelo de Diseño
- Modelo de Despliegue
- Modelo de Datos
- Modelo de Implementación
- Modelo de Pruebas

Fue seleccionada como metodología ya que su característica de ser iterativo e incremental, facilita la corrección de errores en tiempo, lo cual es una ventaja para un proyecto con corto tiempo de desarrollo. Además, la documentación que genera sobre el sistema es necesaria para futuras mejoras al software.

1.5 Lenguaje de programación y herramientas

1.5.1 Python

Python es un lenguaje de programación creado por Guido van Rossum 1991 cuyo nombre está inspirado en el grupo de cómicos ingleses “MontyPython”.

Fue seleccionado dada su característica de ser orientado a objetos, su alto grado de legibilidad y su tipado dinámico.

Este lenguaje posee un desarrollo abierto, por lo que está en un proceso de continuo desarrollo por una gran comunidad de desarrolladores. Aproximadamente cada seis meses se hace pública una nueva versión de Python y posee una variada documentación comenzando por la expuesta en su sitio web, así como sus librerías, todos disponibles en la web de manera gratuita.

El intérprete de Python tiene versiones para prácticamente cualquier sistema operativo (incluyendo GNU/Linux, Windows, VMS, MAC, AmigaOS...) y para los entornos más variados (incluyendo .NET de Microsoft y Java, de Sun). Es extensible con módulos de C, C++, Java (Jython), entre otros. Entre sus características están:

- ✓ Puede utilizarse como lenguaje imperativo procedimental o como lenguaje orientado a objetos.
- ✓ Su entorno de ejecución detecta muchos de los errores de programación que escapan al control de los compiladores y proporciona información muy rica para detectarlos y corregirlos.
- ✓ Soporta objetos y estructuras de datos de alto nivel: strings, listas, diccionarios, etc.
- ✓ Múltiples niveles de organizar código: funciones, clases, módulos, y paquetes.
- ✓ No requiere paso de compilación/linkage.
- ✓ Open Source.
- ✓ Utiliza la tabulación para mostrar estructura de bloques.
- ✓ Al ser un lenguaje de alto nivel, no posee un control directo sobre memoria y otras tareas de bajo nivel.
- ✓ Alto grado de legibilidad

“Python ha sido parte importante de Google desde el principio, y lo sigue siendo a medida que el sistema crece y evoluciona. Hoy día, docenas de ingenieros de Google usan Python y seguimos buscando gente diestra en este lenguaje. “ Peter Norvig, director de calidad de búsquedas de Google Inc. ¹²

1.5.2 Visual Paradigm

Como herramienta de modelado fue seleccionado Visual Paradigm puesto que permite modelar todos los artefactos del RUP en lenguaje UML para las cuatro fases del desarrollo de un proyecto de software, inicio, elaboración, construcción y transición. Además, permite generar otros artefactos u otro diagrama a partir de un diagrama, minimizando así el tiempo de desarrollo del proyecto y mejorando la calidad de la documentación del mismo salvaguardándolo de errores humanos. Es una herramienta multiplataforma, que posee una versión libre distribuida de forma gratuita para Linux y permite la integración con el IDE Eclipse. Su documentación es extensa, y la misma puede ser accedida tanto desde la web como desde la herramienta UML CASE.

1.5.3 EasyEclipse

EasyEclipse es un IDE (Entorno de Desarrollo Integrado, del inglés Integrated Development Environment), que proporciona paquetes precompilados que integran todo lo necesario para utilizar Eclipse en dependencia de las necesidades de la plataforma que se utilice, Windows, Linux, MacOS; del lenguaje de programación, Java, Python, Perl, C++; o del tipo de proyecto que se esté desarrollando, web, gui, etc. Es software de código abierto, de instalación sencilla, sin dependencias, configurable de acuerdo a las necesidades del programador. Existen configuraciones para desarrollo de servidores, de aplicaciones de escritorio en Java, para dispositivos móviles, y por supuesto, para Python, existe el plug-in llamado PyDev, dado lo cual, además de las características antes mencionadas, lo hacen una opción totalmente viable para el desarrollo del software.

¹² <http://www.python.org>

1.5.4 QT Designer

QT-Designer es una herramienta, perteneciente a la librería QT perteneciente a la empresa Trolltech AS. Está escrita en código C++ y es orientada a objetos. Es utilizada para el desarrollo de la estructura primaria de aplicaciones de interfaz gráfica y se basa en los conceptos de widgets (objetos), señales y eventos. Se encuentra disponible para un gran número de plataformas, como Linux, Windows, Unix, MacOS y lenguajes como C++, Python o Perl. Sencilla y fácil de utilizar, ofrece funcionalidades como el uso de librerías básicas para entrada/salida de datos, interface con bases de datos, generación de código fuente, etc; que permiten agilizar el desarrollo de software sin poner en peligro la calidad del mismo, por lo cual fue seleccionada para la realización de las interfaces de usuario.

1.6 Conclusiones

En este capítulo se realizó un estudio de las principales soluciones que existen en el mundo para la gestión de los servicios telemáticos en aras de conocer sus principales ventajas; y de las metodologías y herramientas a utilizar en el desarrollo de la aplicación. Fue seleccionado UML como lenguaje de modelado, IDEF0 como técnica de modelado del negocio, RUP como metodología de desarrollo, Python como lenguaje de programación, EasyEclipse como entorno de desarrollo, QT Designer como herramienta para el diseño de interfaces gráficas y Visual Paradigm como herramienta para el modelado.

Capítulo 2: Características del sistema

2.1 Introducción

En el presente capítulo se realiza una descripción de los procesos del negocio pertenecientes al servicio de DNS, así como la definición de los requisitos funcionales y no funcionales. Posteriormente se muestra el diagrama de casos de uso, describiendo aquellos que, por su importancia para el correcto funcionamiento de la aplicación se consideran críticos, junto al prototipo interfaz de usuario correspondiente.

2.2 Problema y situación problemática

En el módulo DNS v2.0 una vez el usuario abre la aplicación, se le presenta un árbol mostrando el servidor que inició, el cual muestra Servidores Externos, Zonas, ACLs, Vistas, Controles y Reenviadores. El usuario tiene la opción de seleccionar el elemento del árbol que desee y en el mismo puede añadir un nuevo objeto, y modificar o eliminar uno ya existente.

Actualmente estas opciones carecen de soporte a nivel de aplicación para su ejecución en redes o nodos con la versión IPv6 del protocolo IP; esto constituye una limitante puesto que en nuestros días se está efectuando un gradual pero progresivo empleo de este protocolo debido a la escasez de direcciones Ipv4 y a las numerosas ventajas que posee este protocolo en cuanto a seguridad y calidad de la transmisión. Independientemente a esto, actualmente el servidor no cuenta con una forma legítima de garantizar la seguridad de las transferencias ni la autenticidad de la información que envía, dado lo cual es necesario añadir la opción de activar DNSSEC, la cual asegura mediante una encriptación asimétrica, la integridad y confidencialidad de las informaciones que transitan entre dos servidores. Actualmente el sistema brinda la opción de administrar un servidor de forma remota; sin embargo, en el caso de haber dos personas modificando una misma información se pierden datos al sobrescribir la información de uno u otro de manera exclusiva. Para evitar esto, se implementará la actualización en tiempo real de los datos. El proyecto a desarrollar se vincula con el módulo LDAP, del cual existe igualmente una versión 2.0. Este

servidor actualmente posibilita la gestión de esquemas y atributos binarios, generar un certificado y mantener una comunicación segura con el mismo, sin embargo aún no se encuentra configurada para recibir y procesar los registros del servidor DNS, debido a ello se desarrolla actualmente su versión 3.0.

2.2.3 Información que se maneja

Para realizar los disímiles procesos de un servidor DNS, se emplea el fichero de configuración, el cual contiene la información sobre los servidores externos, activación o no de IPv6 y de DNSSEC, y del resto de las opciones que el administrador decida incorporar a su servidor; o sea, contiene todos los indicadores sobre cómo debe proceder el servidor. Además se trabaja con los archivos .key en la gestión de la seguridad y con el fichero zones.rfc1918, el cual contiene la información sobre las zonas definidas por el administrador. Es por ello que solo una persona con los conocimientos adecuados debe modificar estos ficheros, ya que una elaboración incorrecta del mismo podría traer como consecuencia desde un mal funcionamiento o un comportamiento inesperado del servidor hasta la interrupción del servicio.

2.3 Propuesta de sistema

Para optimizar el módulo DNS de la Plataforma de Gestión de Servicios Telemáticos en GNU/Linux se desea añadirle funcionalidades como son la gestión de IPv6 a nivel de aplicación y de DNSSEC, la integración con el servidor LDAP y la actualización en tiempo real de los datos del servidor. De este modo el sistema será capaz de añadir, modificar, y/o eliminar una Vista, Zona, Llave, ACL, Servidor Externo o Reenviador, de servidores tanto de subredes Ipv6 como Ipv4. El usuario podrá realizar transferencias de datos cifradas y anónimas entre dos servidores DNS sincronizados y guardar los registros concernientes en el Servidor de Directorio. Además la información de configuración se actualizará automáticamente una vez sujeto a modificaciones eliminando el riesgo de una posible pérdida de datos. La propuesta es la incorporación al módulo DNSv2.0, de funcionalidades que permitan al administrador gestionar y aprovechar al máximo las capacidades de Bind9.

2.4 Modelo de negocio

Para el modelado del negocio se utilizó IDEF0, ya que permite representar de forma estructurada las actividades que conforman el sistema facilitando la comprensión de los procesos y manteniendo una visión de qué se necesita para llevarlos a cabo, quién los realiza y cuál es el resultado de dicho proceso. A continuación se muestra el modelo del negocio para los procesos correspondientes a la administración de un servidor de nombres de dominio.

2.4.1 Personas que intervienen en los procesos del Negocio

| Nombre | Tarea que realiza |
|---------------|--|
| Administrador | Persona que manejará la aplicación y ejecuta todos los procesos que se relacionan con un Servidor DNS. |

Tabla 1 Personas que intervienen en los procesos del negocio

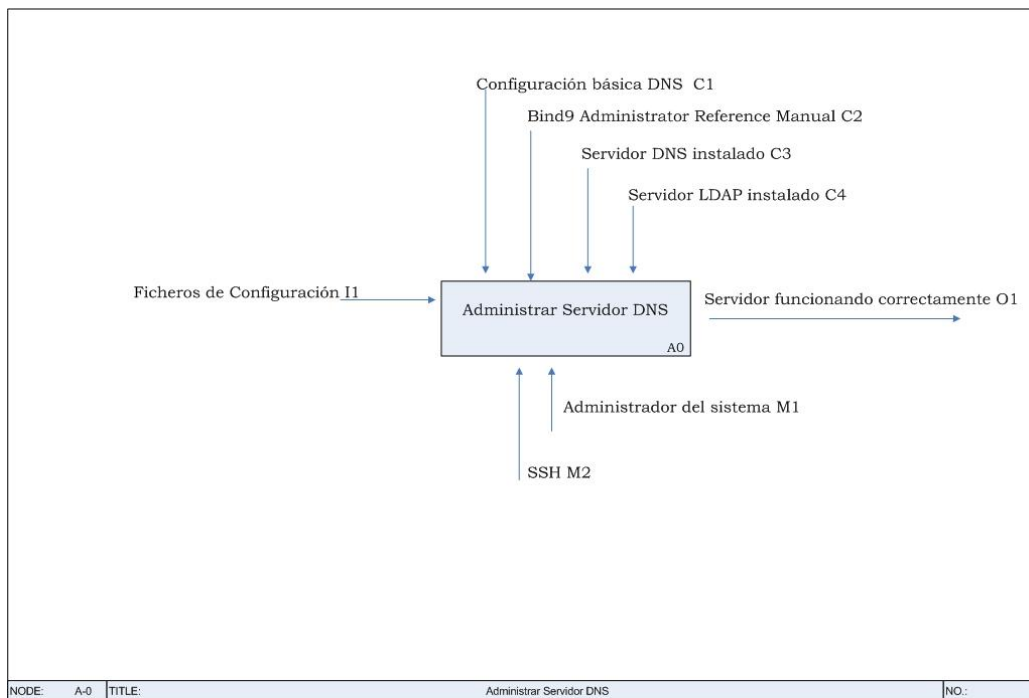


Figura 1 Proceso Administrar Servidor DNS.

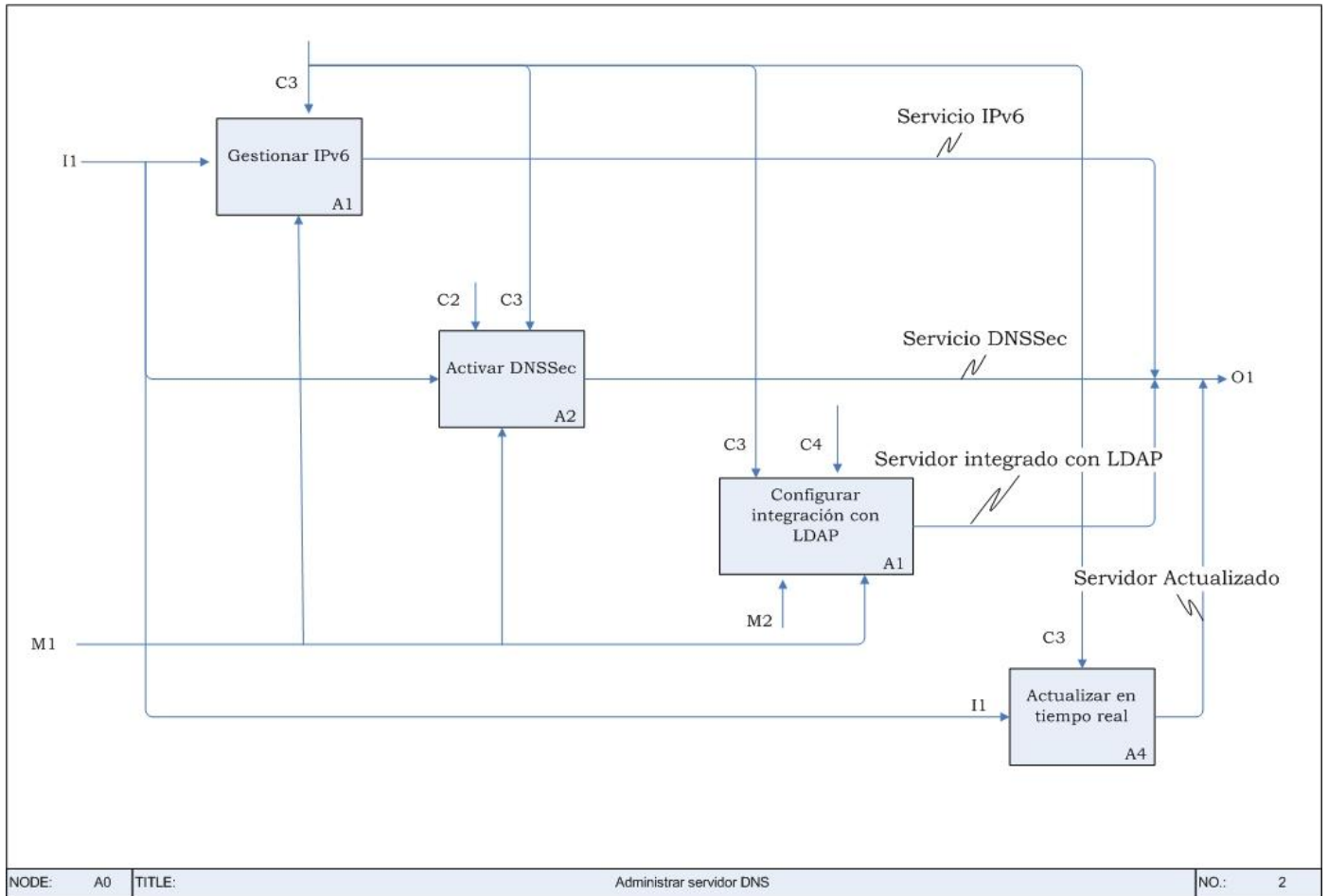


Figura 2 Descripción del proceso Administrar Servidor DNS

2.4.2 Descripción de los Procesos del Negocio

Los procesos identificados para el módulo DNS en su versión 3.0 son:

- ⤴ A0: Administrar Servidor DNS: Proceso que engloba todos los elementos que intervienen en la administración de un servidor de sistema de nombres de dominio. Es realizado por el administrador de red, el cual requiere de los ficheros de configuración para realizarle los cambios. Dicho fichero constituye la base sobre la que siempre se trabajará. Para administrar correctamente el servidor es una condición necesaria contar con un Servidor DNS instalado.
- ⤴ A1: Gestionar IPv6: Permite gestionar las funciones del Servidor DNS para hosts que utilicen el protocolo IPv6. Este proceso es realizado por el administrador de red, el cual toma como entrada el fichero de configuración y necesita de un servidor DNS instalado para su funcionamiento.
- ⤴ A2: Activar DNSSEC: Proceso que posibilita establecer la transferencia de datos garantizando la autenticidad de los mismos, entre dos servidores sincronizados tomando como controles el Manual de Referencia de Bind9 y la condición de un servidor DNS instalado.
- ⤴ A3: Configurar Integración con LDAP: El administrador de red, con ayuda del módulo LDAP, selecciona si desea integrar el servidor con el Servidor de Directorio. Para ello se necesita de ambos servidores instalados.
- ⤴ A4: Actualizar en tiempo real: Permite actualizar dinámicamente los datos del servidor que han sido modificados de forma remota. Es una tarea que se realiza automáticamente y necesita del fichero de configuración y de un servidor DNS instalado.

2.5 Requerimientos

Los requisitos de software suelen ser definidos como el conjunto de funcionalidades y/o características que debe tener un producto de software para cumplir con un acuerdo o contrato. Los mismos constituyen un eslabón fundamental para tener una idea clara sobre qué se espera y se desea de la aplicación que se va a construir. En el caso del módulo DNS se trata de cumplir con un producto que satisfaga la necesidad de un servidor DNS con las características deseadas. Las funcionalidades se tienen en cuenta en los requisitos funcionales; mientras las características se citan en los requisitos no funcionales.

2.5.1 Listado de los requerimientos funcionales

RF1. Gestionar ACLs con direcciones IPv6

- RF 1.1 Adicionar ACL con dirección IPv6
- RF1.2 Modificar ACL con dirección IPv6
- RF 1.3 Eliminar ACL con dirección IPv6
- RF 1.4 Mostrar ACL con dirección IPv6

RF2. Gestionar Llaves con direcciones IPv6

- RF 2.1 Adicionar Llave con dirección IPv6
- RF 2.2 Modificar Llave con dirección IPv6
- RF 2.3 Eliminar Llave con dirección IPv6
- RF 2.4 Mostrar Llave con dirección IPv6

RF3. Gestionar Servidores Externos con dirección IPv6

- RF 3.1 Adicionar Servidor Externo con dirección IPv6
- RF 3.2 Modificar Servidor Externo con dirección IPv6
- RF 3.3 Eliminar Servidor Externo con dirección IPv6
- RF 3.4 Mostrar Servidor Externo con dirección IPv6

RF4. Gestionar Reenviadores con dirección IPv6

- RF 4.1 Adicionar Reenviador con dirección IPv6

RF 4.2 Modificar Reenviador con dirección IPv6

RF 4.3 Eliminar Reenviador con dirección IPv6

RF 4.4 Mostrar Reenviador con dirección IPv6

RF5. Gestionar Zonas con direcciones IPv6

RF 5.1 Adicionar Zona con dirección IPv6

RF 5.2 Modificar Zona con dirección IPv6

RF 5.3 Eliminar Zona con dirección IPv6

RF 5.4 Mostrar Zona con dirección IPv6

RF 6. Gestionar Vistas con direcciones IPv6

RF 6.1 Adicionar Vista con dirección IPv6

RF 6.2 Modificar Vista con dirección IPv6

RF 6.3 Eliminar Vista con dirección IPv6

RF 6.4 Mostrar Vista con dirección IPv6

RF7. Activar el sistema de transferencia de objetos con DNSSEC

RF 7.1 Generar Llaves

RF 7.2 Firmar las zonas

RF 7.3 Configurar servidores DNS

RF8. Integrar el servidor DNS con el servidor LDAP

RF9. Actualizar en tiempo real los datos de configuración del servidor

2.5.2 Definición de los requerimientos no funcionales

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.

Fiabilidad: Garantizar tiempos de respuestas generalmente rápidos al igual que la velocidad de procesamiento de la información. La aplicación debe ser capaz de realizar consultas en el menor tiempo posible, para así evitar la saturación de la red.

Eficiencia: 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.

GB de espacio libre en el disco duro.

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 (sistema de almacenamiento que permite al sistema tratar a un conjunto de discos duros como si se tratara de una sola unidad lógica) 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.

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

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.

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.

Restricciones de diseño

- ✿ Será empleado Python 2.6 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, sobre

todo 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.

2.6 Modelo de Casos de Uso del Sistema

2.6.1. Paquetes y relación entre ellos

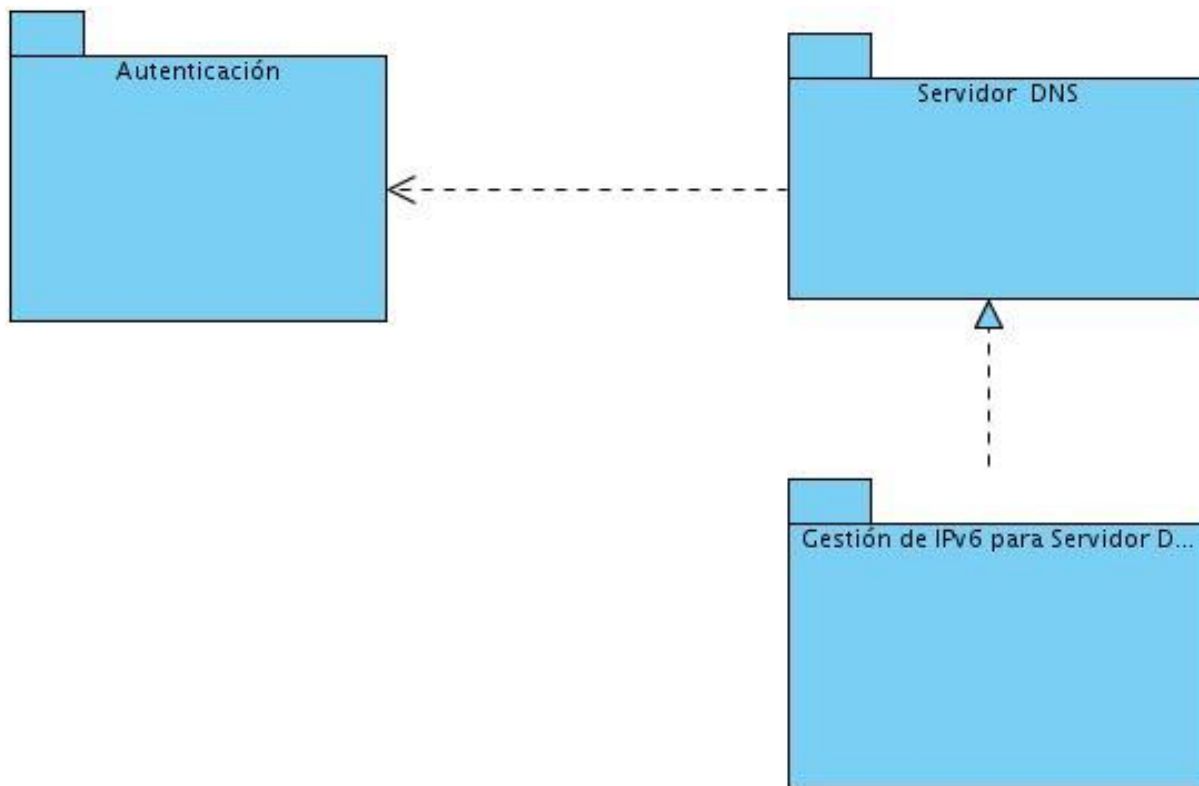


Figura 3 Diagrama de paquetes

DIAGRAMA DE CASOS DE USO

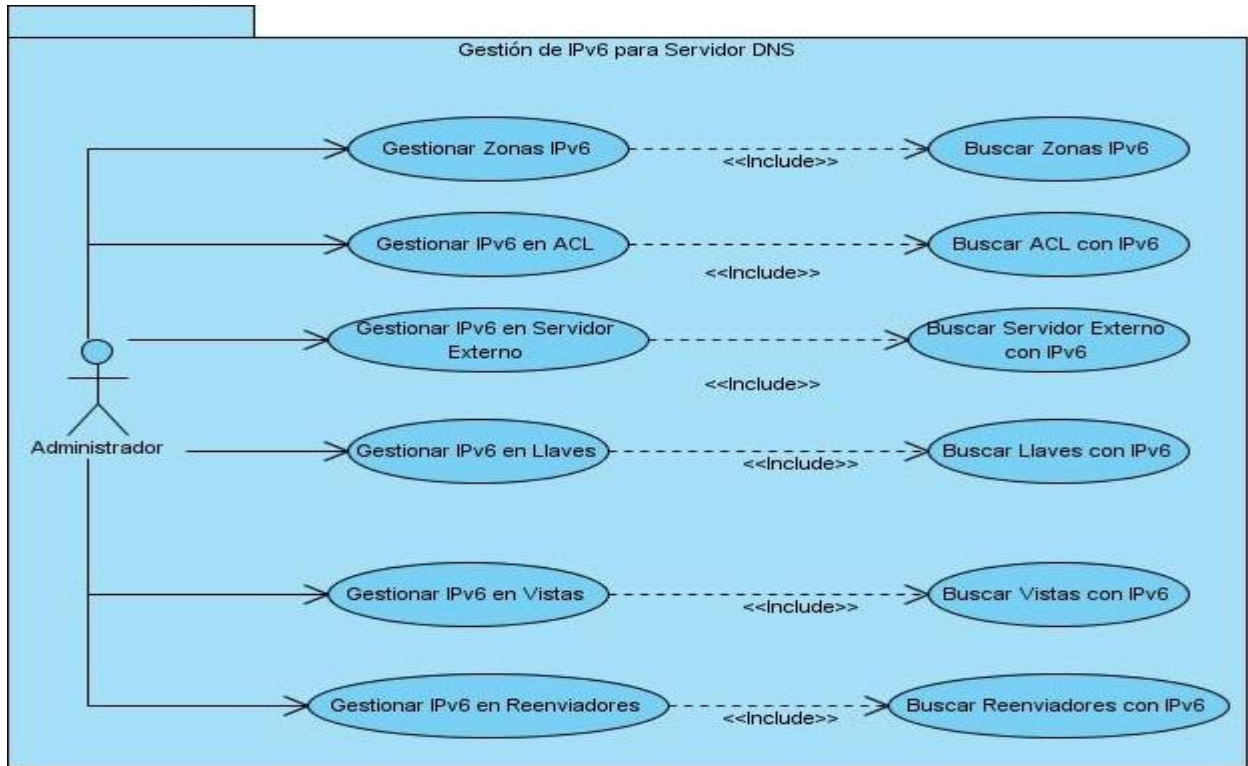
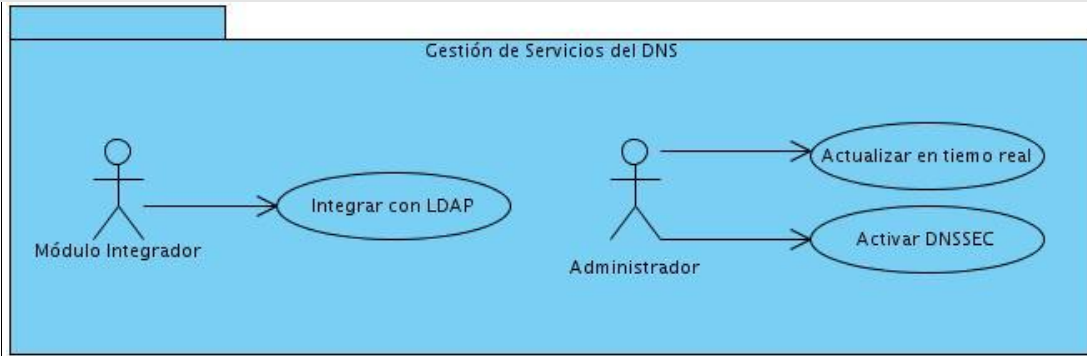


Figura 4 Modelo de Casos de Uso del Sistema

2.6.2 Listado de Casos de Uso

| | |
|--------------------|--|
| CU 1 | Actualizar en tiempo real |
| Actor | Administrador |
| Descripción | Caso de uso que posibilita actualizar los datos del servidor una vez realizados cambios al mismo local o remotamente. |
| Referencia | RF9 |
| CU 2 | Integrar con LDAP |
| Actor | Administrador |
| Descripción | Caso de uso que permite integrar las funcionalidades del servidor DNS con las del servidor LDAP, posibilitando gestionar una o varias zonas desde este último. |
| Referencia | RF8 |
| CU 3 | Gestionar DNSSEC |
| Actor | Administrador |
| Descripción | Caso de uso que gestiona la seguridad de las transferencias de datos entre los servidores DNS. |
| Referencia | RF7 |
| CU 4 | Gestionar Zonas IPv6 |
| Actor | Administrador |
| Descripción | Caso de uso que permite agregar nuevas zonas con dirección IPv6, además de modificarlas o eliminarlas. |
| Referencia | RF5 |
| CU 5 | Buscar Zonas IPv6 |
| 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 | RF5 |
| CU 6 | Gestionar IPv6 en ACL |
| Actor | Administrador |
| Descripción | Caso de uso que permite agregar nuevas reglas que incluyan nodos con dirección IPv6, o modificar o eliminar una ACL existente. |
| Referencia | RF1 |
| CU 7 | Buscar ACL con IPv6 |
| 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 | RF1 |
| CU 8 | Gestionar IPv6 en Servidor Externo |
| Actor | Administrador |
| Descripción | Caso de uso que permite agregar nuevos servidores externos cuya dirección sea IPv6, además de modificar o eliminar dichos servidores. |
| Referencia | RF3 |
| CU 9 | Buscar Servidor Externo con IPv6 |
| Actor | Administrador |
| Descripción | Posibilita listar todos los servidores externos y buscarlos cuando se requiera. |
| Referencia | RF3 |
| CU 10 | Gestionar IPv6 en Llaves |
| Actor | Administrador |
| Descripción | Este caso de uso permite agregar nuevas llaves, modificar, o eliminar llaves ya existentes. |
| Referencia | RF2 |
| CU 11 | Buscar Llaves con IPv6 |
| Actor | Administrador |
| Descripción | El sistema ofrece la posibilidad de listar todas las llaves y buscarlas cuando sea necesario. |
| Referencia | RF2 |
| CU 12 | Gestionar IPv6 en Vistas |
| Actor | Administrador |
| Descripción | El caso de uso posibilita la adición de nuevas vistas, además de permitir modificar o eliminar una vista ya existente |
| Referencia | RF6 |
| CU 13 | Buscar Vistas con IPv6 |
| 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 | RF6 |
| CU 14 | Gestionar IPv6 en Reenviadores |
| Actor | Administrador |

| | |
|--------------------|---|
| Descripción | Este caso de uso brinda la posibilidad de adicionar un nuevo reenviador cuya dirección sea IPv6, además de posibilitar la modificación o eliminación de uno ya existente. |
| Referencia | RF4 |
| CU 15 | Buscar Reenviadores con IPv6. |
| 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 | RF4 |

Tabla 2 Listado de Casos de Uso.

2.6.3. Definición de actores

| Actor | Descripción |
|--------------------------|--|
| Administrador | Usuario que acciona los casos de uso del sistema relacionados con su administración. |
| Módulo Integrador | Sistema externo que acciona los casos de uso relacionados con la integración. |

Tabla 3 Actores del sistema

2.6.4 Descripción textual de los Casos de Uso Críticos.

Se consideran críticos los casos de uso que por su importancia definen la utilidad del software a implementar. En el caso del módulo DNSv3.0, el caso de uso Integrar con LDAP, a pesar de no tener una interfaz de usuario definida, es esencial para el mismo ya que define su integración con el resto de la plataforma que se desea obtener. El caso de uso DNSSEC es necesario dado que fuera de la autenticación, la versión anterior del módulo carece de implementaciones que gestionen la seguridad del servidor, tornando la plataforma un software poco fiable para ser utilizado en entornos reales.

| | |
|------------------------|--|
| Caso de Uso: | Gestionar DNSSEC |
| Actores: | Administrador |
| Resumen: | Caso de uso que gestiona la seguridad de las transferencias de datos entre los servidores DNS. |
| Precondiciones: | Existencia de un Servidor DNS Maestro y uno o más Servidores DNS Esclavos. |

Tabla 4 Datos del Caso de Uso Gestionar DNSSEC

| Acción del Actor | Respuesta del Sistema |
|--|--|
| 1- Inicia seleccionando en el menú Operaciones la opción Habilitar DNSSEC. | 2- Muestra la interfaz Gestionar DNSSEC. La cual contiene las opciones Generar Claves, Firmar Zonas y Configurar Servidores. |

Sección “Generar claves”

| Acción del Actor | Respuesta del Sistema |
|---|---|
| 3- El usuario selecciona el algoritmo, tamaño de la clave y tipo de clave y da clic en el botón Generar Claves. | 4- Genera dos archivos .key; los cuales contienen las claves pública y privada respectivamente. |

Flujos Alternos

| Acción del Actor | Respuesta del Sistema |
|------------------|-----------------------|
|------------------|-----------------------|

3- El usuario selecciona el algoritmo, tamaño de la clave, tipo de clave y activa la casilla para generar una clave KSK , posteriormente da clic en el botón Generar Claves.

4- Genera dos archivos .key de tipo KSK; los cuales contienen las claves pública y privada respectivamente.

Poscondiciones Las claves deben ser insertadas en el archivo de las zonas.

Sección “Firmar Zona”

| Acción del Actor | Respuesta del Sistema |
|--|---|
| 3- Selecciona los nombres de las claves KSK y FSK, y el dominio superior. Luego presiona el botón Firmar Zona. | 4- Produce los registros NSEC y RRSIG, y genera una versión firmada de la zona. |

Flujos Alternos

| Acción del Actor | Respuesta del Sistema |
|--|---|
| 3- Selecciona los nombres de las claves KSK y FSK y presiona el botón Firmar Zona. | 4- Produce los registros NSEC y RSSIC y una versión firmada de la Zona. |

Poscondiciones Referenciar la zona firmada en la declaración de la zona en el archivo named.conf.

Sección “Configurar Servidores”

| Acción del Actor | Respuesta del Sistema |
|--|--|
| 1-El usuario escribe el nombre del servidor esclavo y del servidor maestro correspondientes y selecciona el botón Activar. | 2- Verifica que los datos sean correctos, realiza las modificaciones pertinentes y muestra un mensaje indicando la necesidad de reiniciar el Servidor. |

Tabla 5 Descripción del Caso de Uso Gestionar DNSSEC

Prototipo de Interfaz

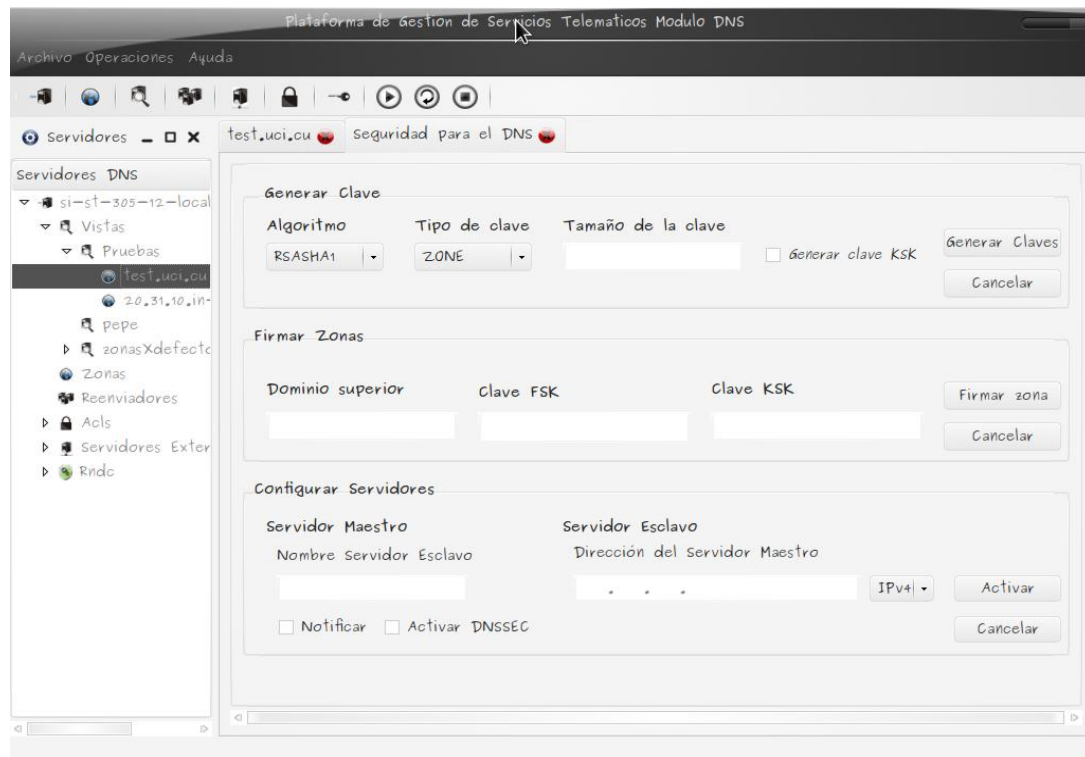


Figura 5 Prototipo de Interfaz del Caso de Uso Gestionar DNSSEC

| | |
|------------------------|--|
| Caso de Uso: | Integrar con LDAP |
| Actores: | Administrador, Módulo integrador |
| Resumen: | Caso de uso que integra el servidor DNS con el servidor LDAP permitiendo almacenar los registros del DNS en el servidor de Directorio. |
| Precondiciones: | Existencia de un Servidor DNS y un Servidor LDAP. |

Tabla 6 Datos del Caso de Uso Integrar con LDAP

| Acción del Actor | Respuesta del Sistema |
|--|---|
| 1- Inicia cuando en el Módulo Integrador, se selecciona la opción Integrar DNS con LDAP. | 2- El sistema escribe en la configuración de la zona, la dirección del servidor LDAP donde se almacenarán sus datos y crea un fichero con la información de la misma en formato Idif. |

| Flujos Alternos | |
|------------------|---|
| Acción del Actor | Respuesta del Sistema |
| | 2- De no ser correctos los datos entrados por el usuario, el sistema envía un mensaje de error. |
| Poscondiciones | - |

Tabla 7 Descripción del Caso de Uso Integrar con LDAP

2.7 Conclusiones

En el capítulo que concluye se ha realizado un minucioso estudio de los procesos del negocio correspondientes a los requisitos funcionales del sistema que se desea implementar. Tomando como punto de partida la problemática a resolver, se determinó una propuesta de solución la cual. Profundizando en ello, se identificaron y posteriormente se describieron los casos de uso del sistema, los cuales son esenciales para una correcta construcción de los diagramas del flujo de trabajo Análisis y Diseño, el cual constituye el siguiente capítulo.

Capítulo 3: Análisis y Diseño

3.1 Introducción

En el presente capítulo se presenta la estructura general del sistema, así mismo se dan a conocer la estructura detallada del sistema para una mejor comprensión del mismo, y los diagramas de clases. Además se define la arquitectura a utilizar y los patrones de diseño que se aplicarán en la solución del problema planteado.

3.2 Diseño

Una arquitectura común de los sistemas de información que abarca una interfaz para el usuario, la lógica que rige el proceso y el almacenamiento persistente de datos se conoce con el nombre de arquitectura de tres capas. Sin embargo, en un diseño orientado a objetos, la capa de la lógica de aplicaciones se divide en otras menos densas. Esto permite el aislamiento de la lógica de aplicaciones en componentes independientes que pueden ser reutilizados posteriormente en otros sistemas. Tal es el caso de la arquitectura multicapas, utilizada para desarrollar la aplicación que ocupa este trabajo de diploma, siendo el objetivo principal de este diagrama, ofrecer una visión simplificada del sistema. Como se observa, se pueden identificar fácilmente los componentes que intervienen en el mismo, representadas por paquetes se encuentran las diferentes capas que integran el sistema y cómo interactúan entre sí.

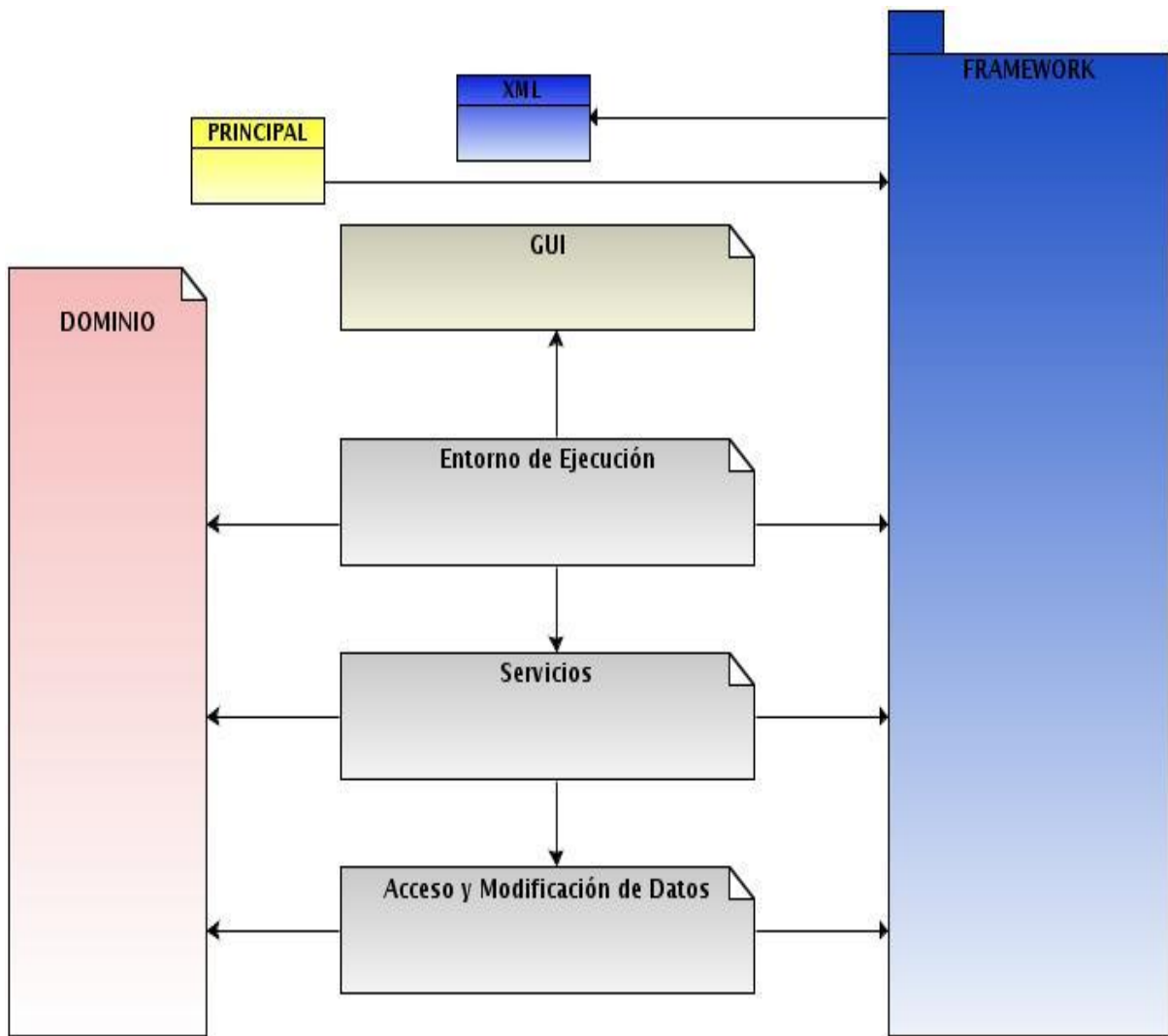


Figura 6 Estructura general del Sistema

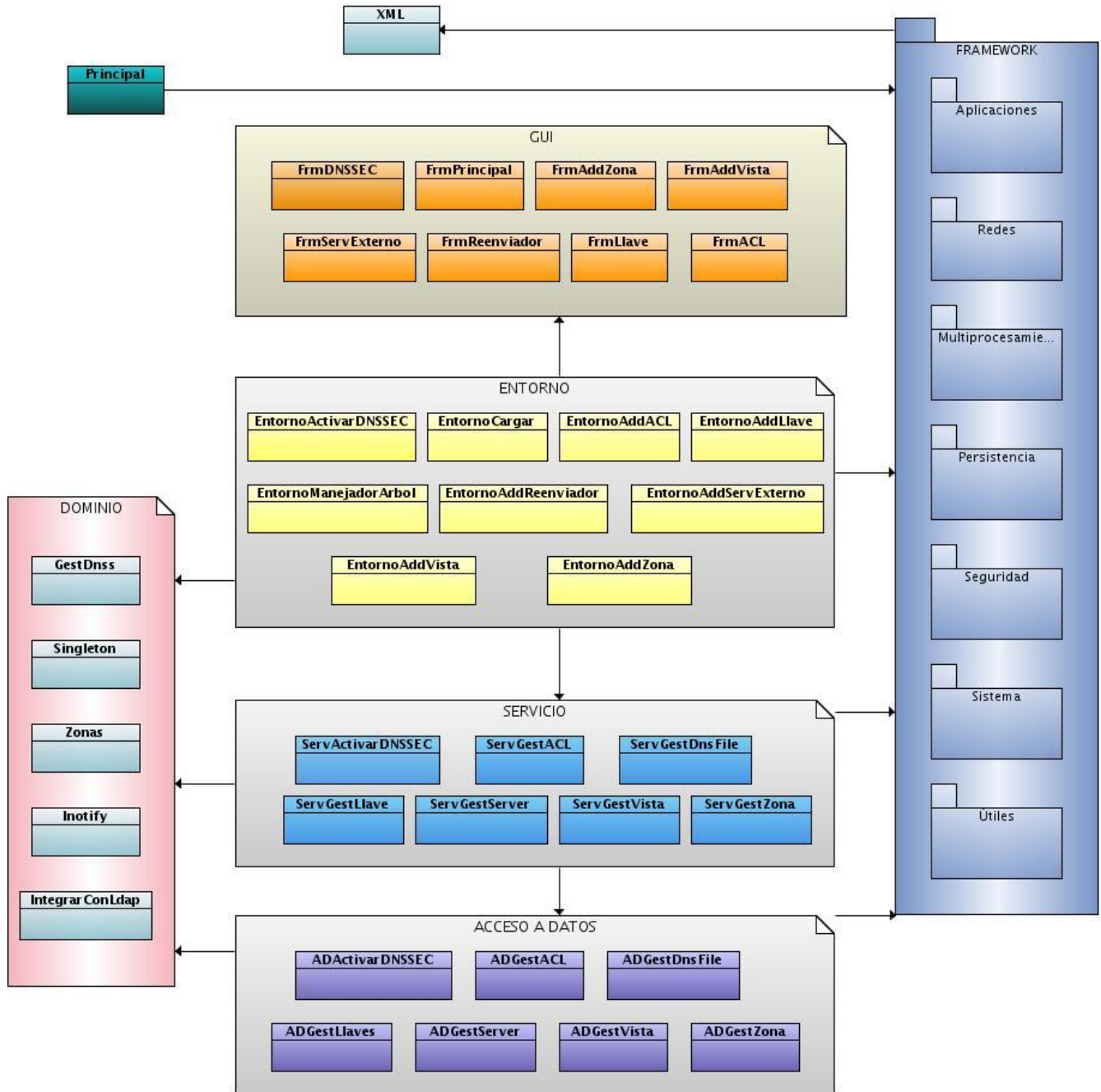


Figura 7 Estructura detallada del sistema

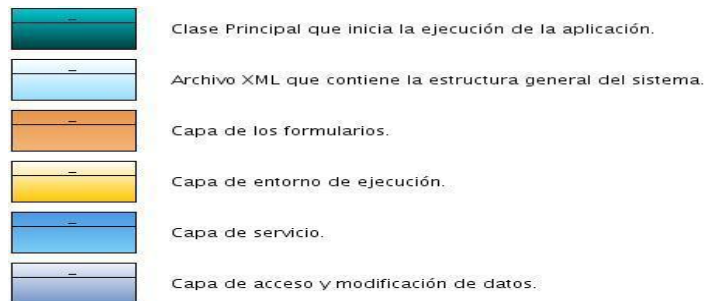


Figura 8 Leyenda

3.2.1 Patrones arquitectónicos

Se utilizará la arquitectura en capas, específicamente el patrón n-capas, el mismo permite que los componentes sean reutilizados por diversos tipos de aplicaciones y facilita la tarea de corregir errores, lo cual reduce además el tiempo de mantenimiento, por otro lado su escalabilidad posibilita que el añadir funcionalidades no sea una tarea escabrosa puesto que se pueden crear nuevos componentes sobre la base para especializar las funcionalidades de un componente específico. En el caso del módulo DNS se añaden, además de las capas de Interfaz (GUI), Servicios, y Acceso a Datos, la capa Entorno, la cual se encarga de mediar entre la capa GUI y la de servicios, independizando a la segunda del tipo de interfaz gráfica a utilizar y facilitando la programación puesto que la gestión de cambios es menos trabajosa. Se utiliza además un Framework del proyecto, en el cual, se gestionan la autenticación e inicialización del sistema, la instalación del mismo, gestión de las conexiones, el multiprocesamiento o ejecución en hilos, tratamiento de excepciones, entre otras funciones que posibilitan definir en el módulo sólo lo que refiere estrictamente a sus funcionalidades y la reutilización de este código por el resto de los módulos, adquiriendo una uniformidad para todas las aplicaciones. Se hace uso de una capa adicional llamada Dominio, la cual contiene clases que serán empleadas por más de una capa en el sistema. De manera independiente se gestionan las configuraciones, organizadas de manera más general en el archivo XML.

3.3 Diagramas de clases del diseño.

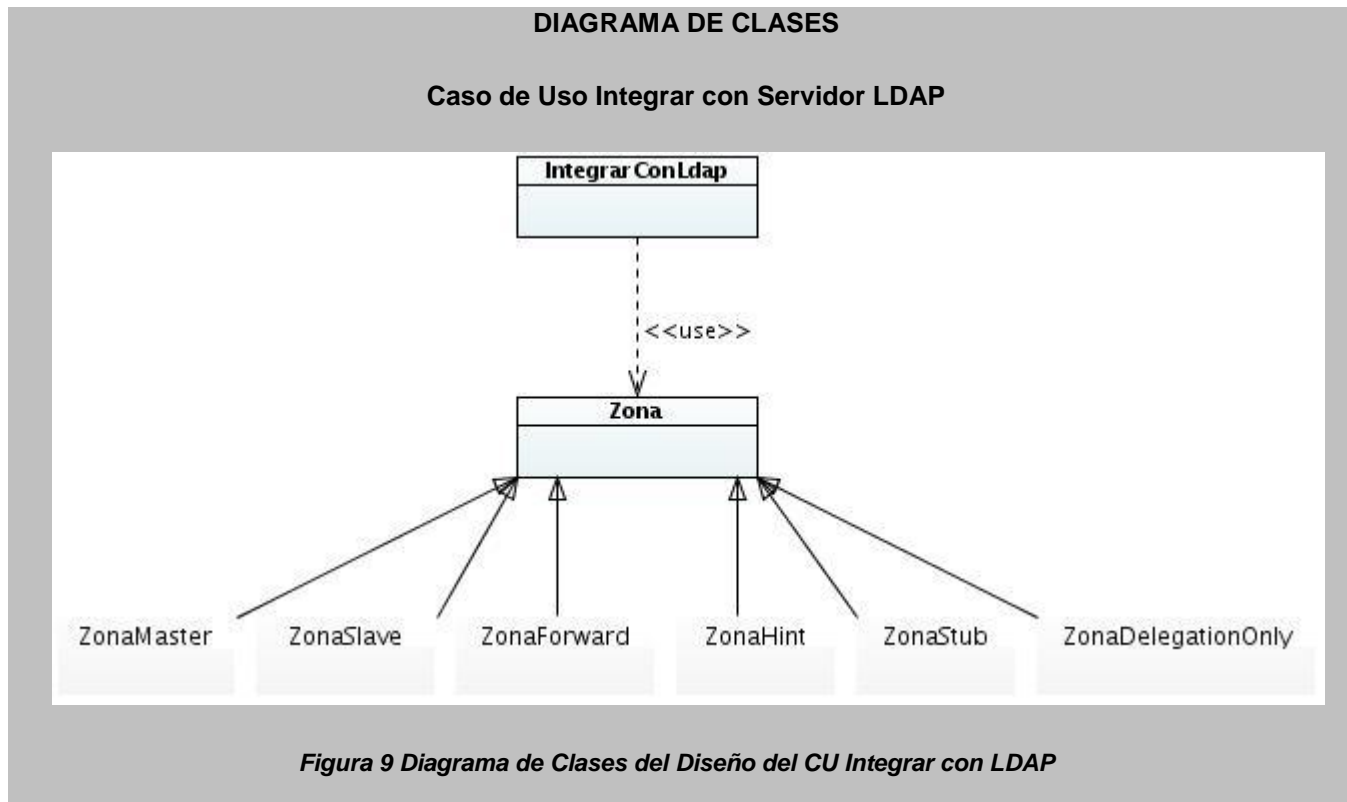


DIAGRAMA DE CLASES

Caso de Uso Activar DNSSEC

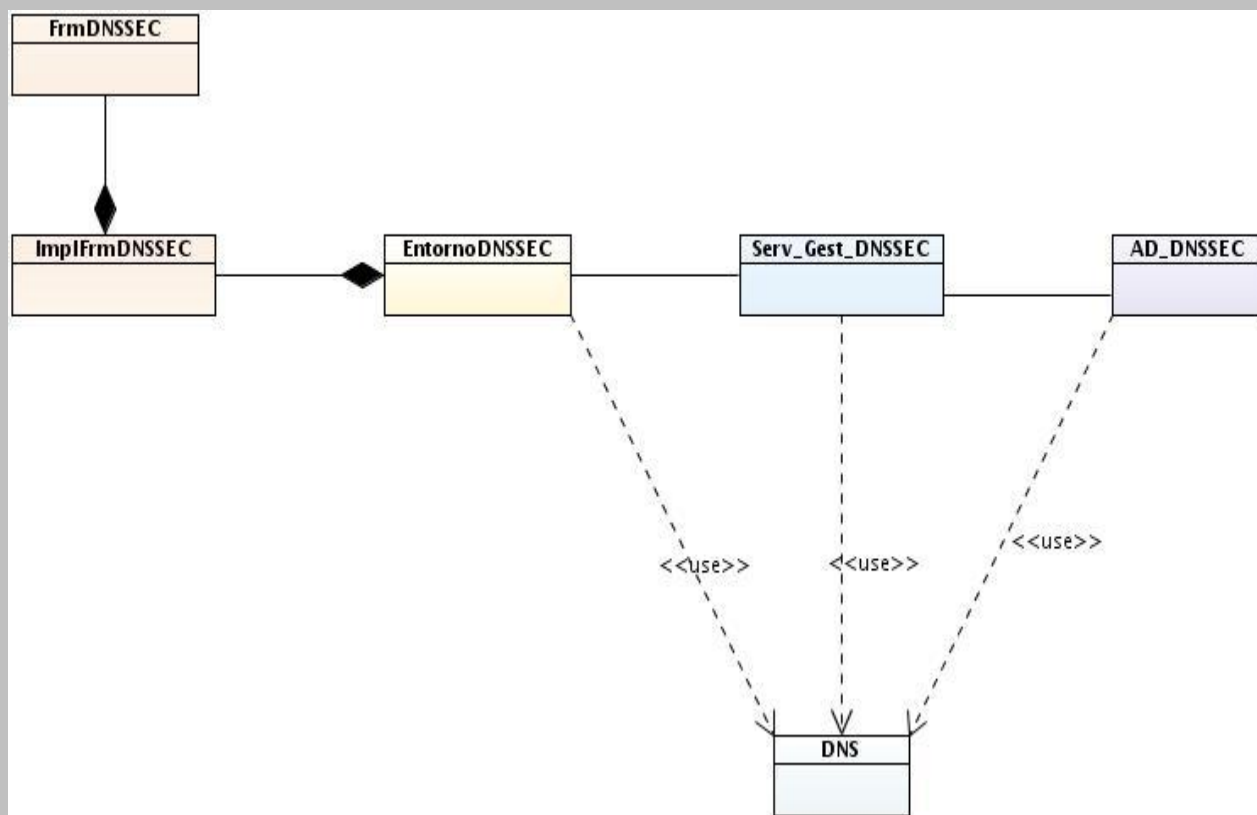


Figura 10 Diagrama de Clases del Diseño del CU Activar DNSSEC

DIAGRAMA DE CLASES

Caso de Uso Actualizar en tiempo real

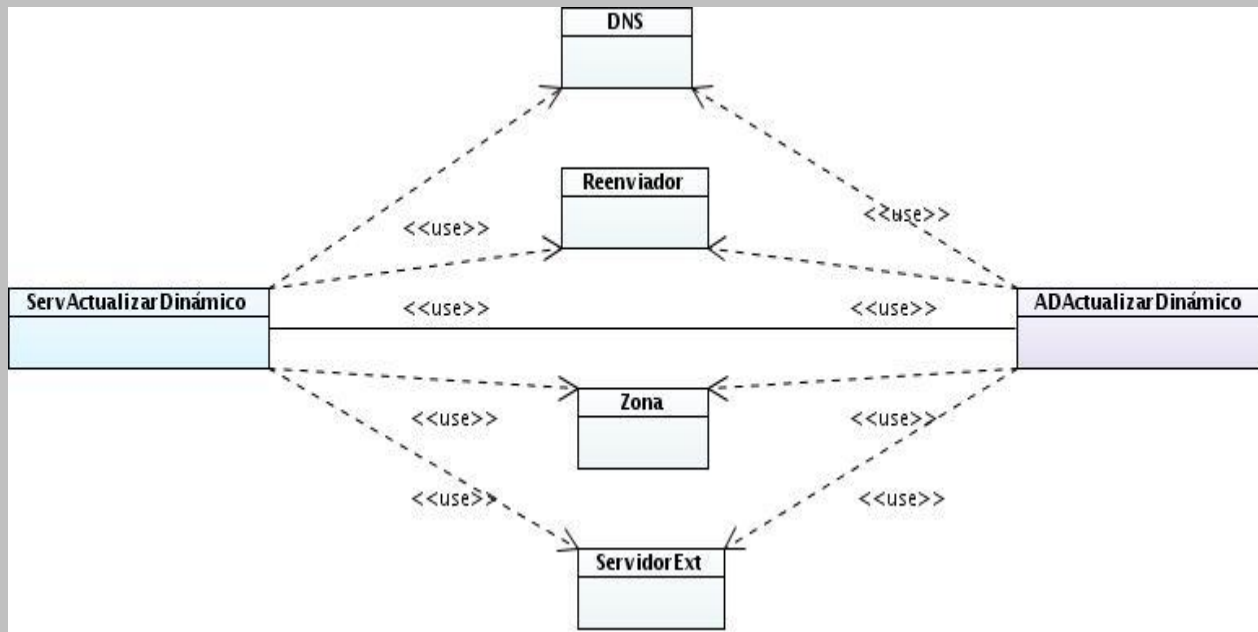


Figura 11 Diagrama de Clases del Diseño del CU Actualizar en tiempo real

DIAGRAMA DE CLASES

Caso de Uso Gestionar ACL

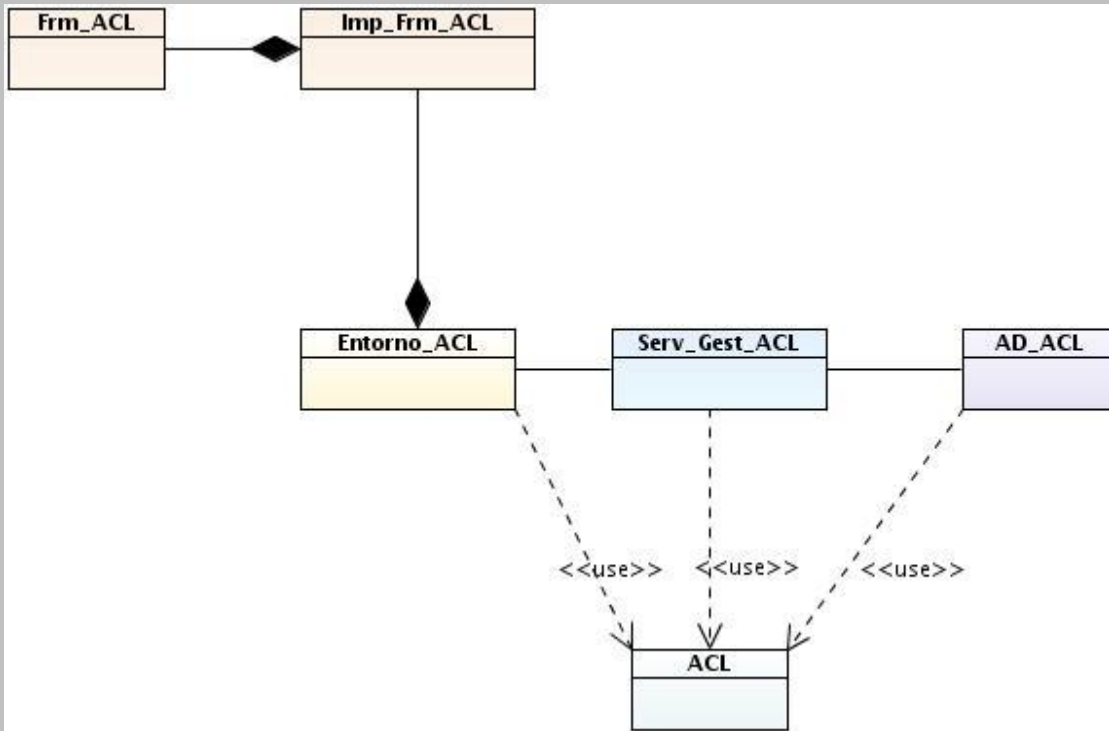


Figura 12 Diagrama de Clases del Diseño del CU Gestionar ACL

DIAGRAMA DE CLASES

Caso de Uso Gestionar Servidor Externo

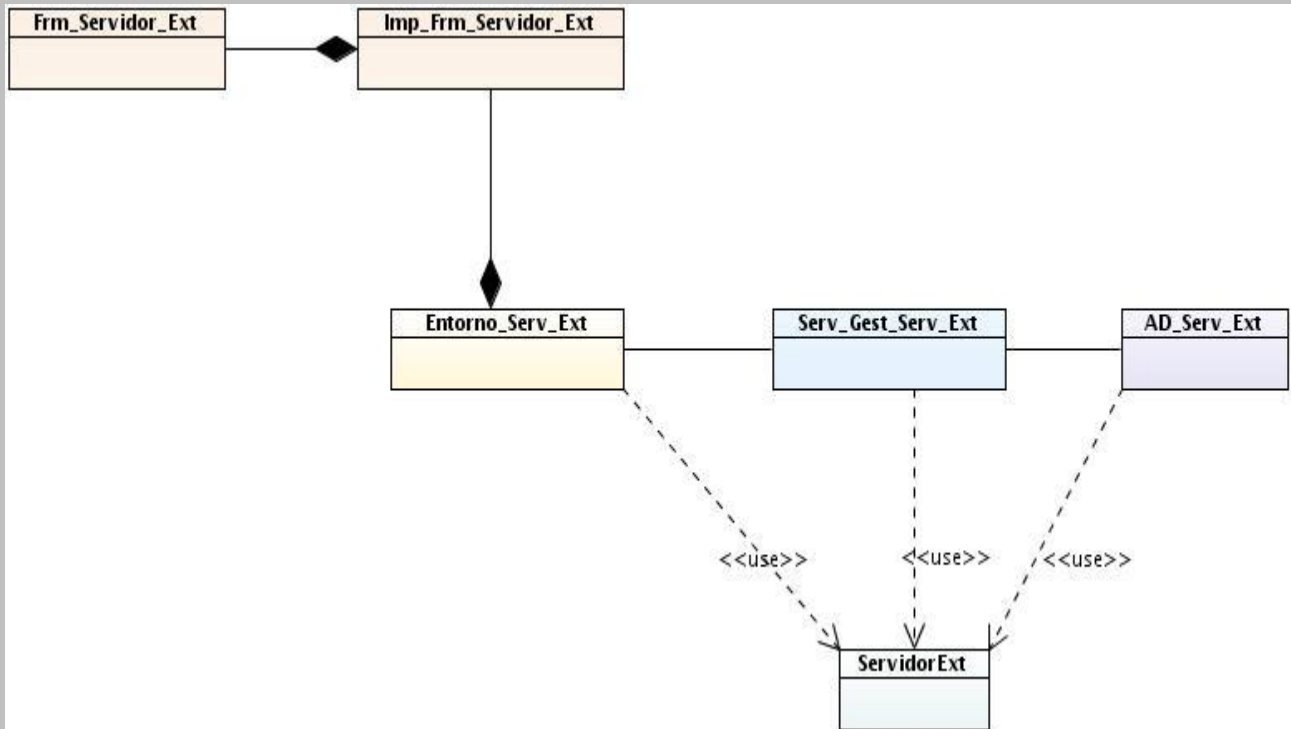


Figura 13 Diagrama de Clases del Diseño del CU Gestionar Servidor Externo

DIAGRAMA DE CLASES

Caso de Uso Gestionar Llaves

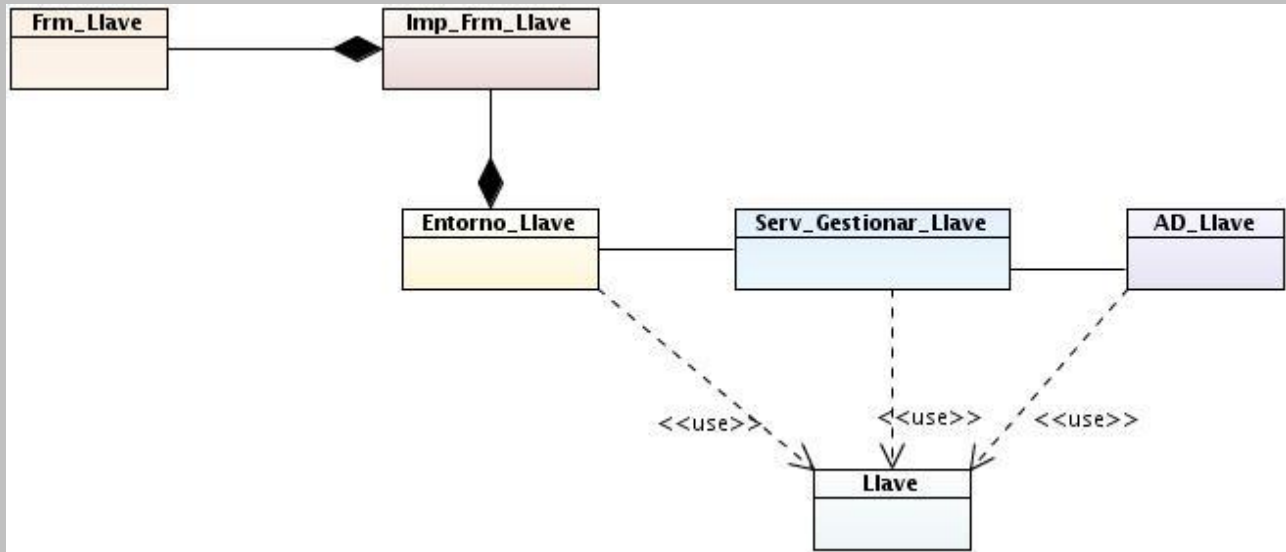


Figura 14 Diagrama de Clases del Diseño del CU Gestionar Llaves

DIAGRAMA DE CLASES

Caso de Uso Gestionar Reenviador

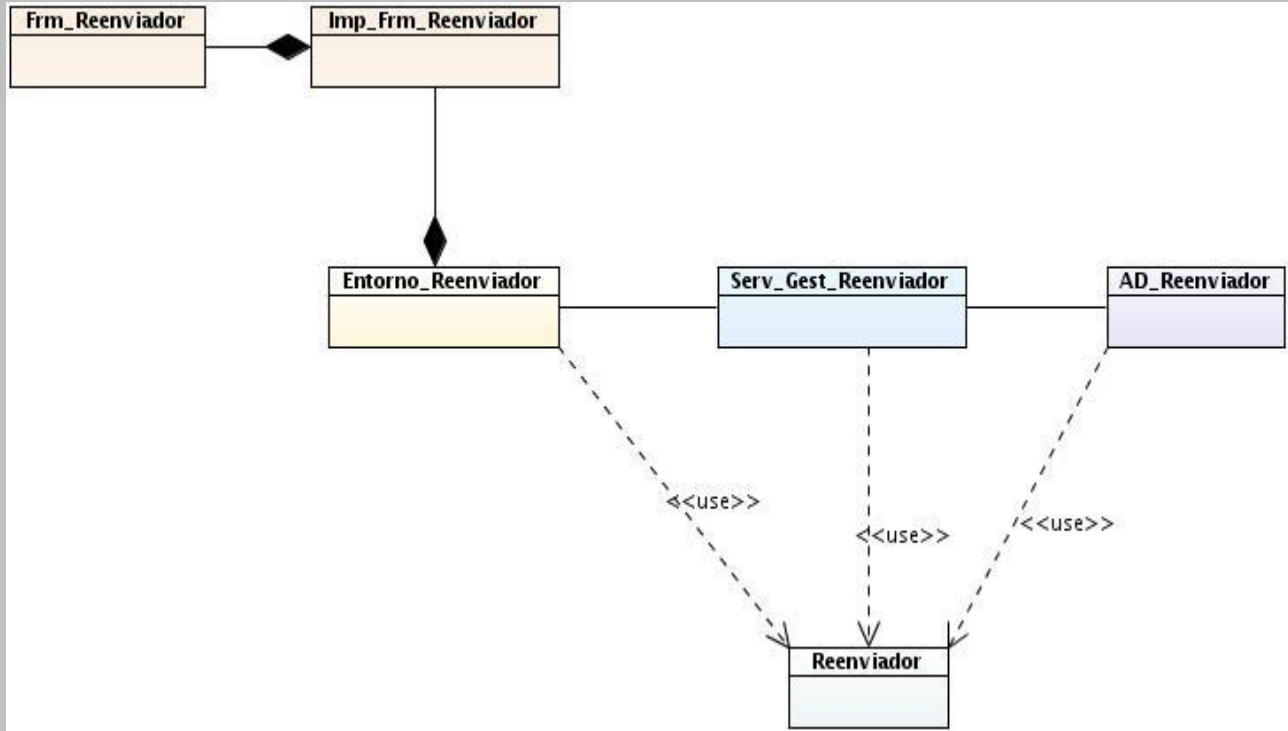


Figura 15 Diagrama de Clases del Diseño del CU Gestionar Reenviador

DIAGRAMA DE CLASES

Caso de Uso Gestionar Vista

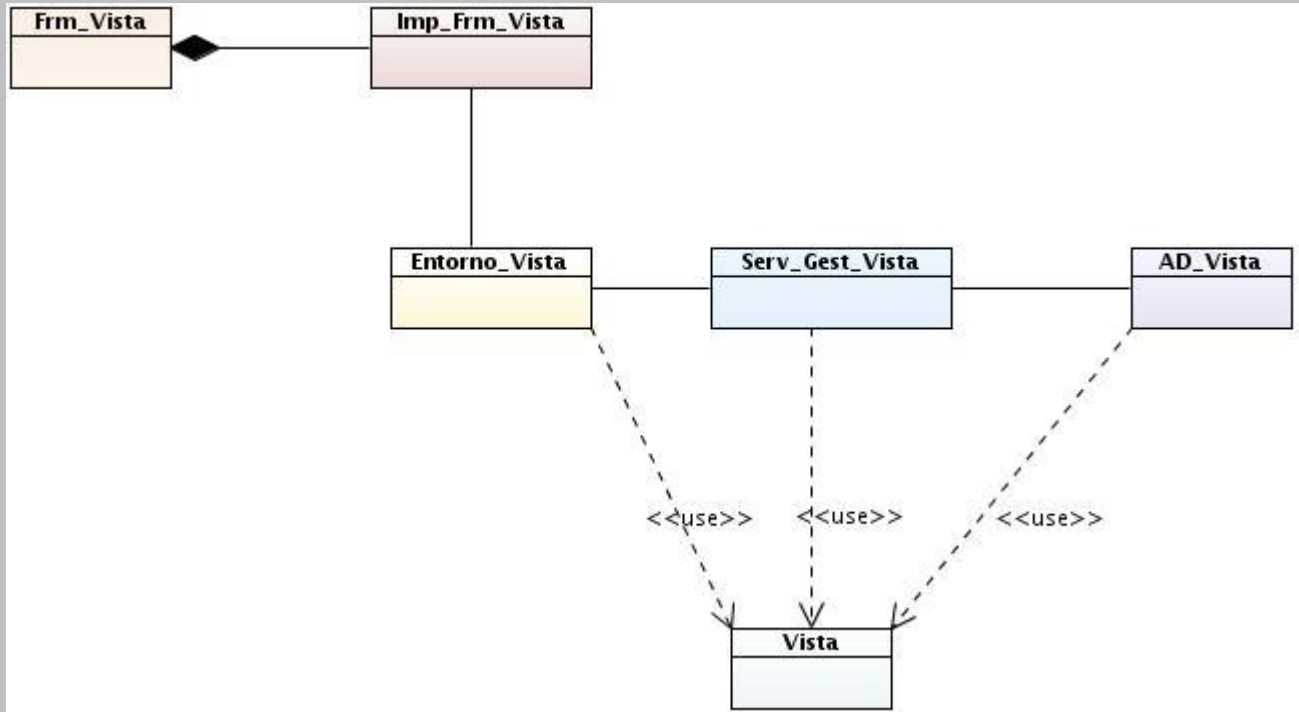


Figura 16 Diagrama de Clases del Diseño del CU Gestionar Vista

DIAGRAMA DE CLASES

Caso de Uso Gestionar Zona

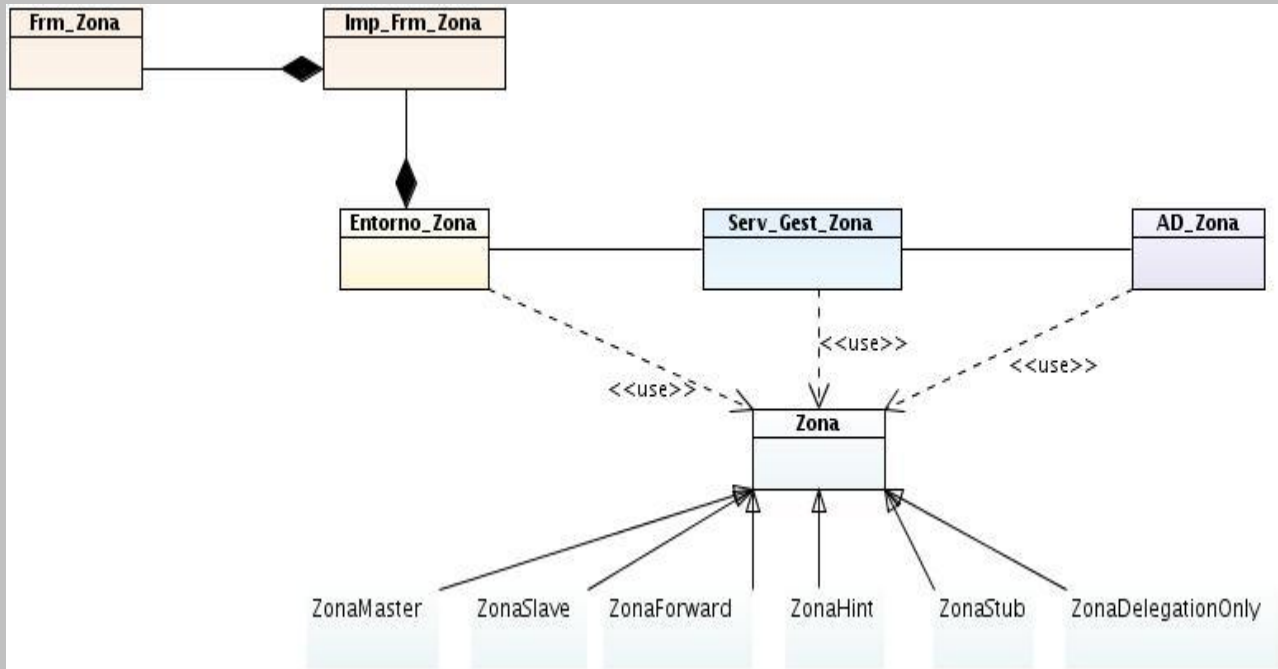


Figura 17 Diagrama de Clases del Diseño del CU Gestionar Zona

3.4 Patrones de diseño

A menudo nos encontramos con dificultades en la programación que son comunes para el problema que nos proponemos resolver. Con el objetivo de proponer una solución a estas dificultades, elegante y basada en la experiencia, surgen los patrones de diseño, los cuales constituyen esquemas de solución para un tipo específico de problema que puede ser adaptada a cualquier entorno. En el módulo DNS, se utilizan los patrones de creación Abstract Factory, que proporciona una interfaz para crear familias de objetos sin necesidad de especificar sus clases concretas; y Singleton, el cual garantiza que una clase tenga sólo una instancia. De los patrones estructurales se utilizan Decorator, el cual consiste en asignar dinámicamente nuevas responsabilidades a un objeto; y Adapter, el cual modifica la interfaz de una clase para lograr cooperación entre dos clases que de otro modo sería imposible por incompatibilidad de interfaces. De los patrones de comportamiento se utiliza el Observer, que define una dependencia de uno a muchos entre objetos, lo cual permite actualizar automáticamente los objetos una vez que uno ha cambiado de estado.

3.5 Conclusiones

En el presente capítulo se presentaron los diagramas y artefactos que conforman el Diseño del sistema, entre ellos los modelos y clases del diseño, los cuales constituyen la guía principal para la futura implementación del sistema. Además se definió la arquitectura empleada, explicándola detalladamente y presentando aquellas capas que, a efectos de lograr las funcionalidades deseadas, se han añadido.

Capítulo 4: Implementación y Prueba

4.1 Introducción

El presente capítulo presenta los artefactos correspondientes al flujo de trabajo de implementación y pruebas. En el desarrollo del mismo se presenta la descripción del modelo de componentes, el cual no es más que la implementación del modelo de diseño en términos de scripts, ficheros de código fuente, y ejecutables. Se representa además en el modelo de despliegue, la topología de hardware sobre la cual funcionará la aplicación. Ambos diagramas, de despliegue y de componentes, conforman lo que se conoce como Modelo de Implementación al describir los componentes de la aplicación a construir, así como su organización y dependencia entre las estructuras físicas en las que funcionará la misma. Se presenta además, la estrategia de prueba definida para validar que los casos de uso implementados funcionan adecuadamente.

4.2 Diagramas de Componentes

Para lograr una mejor comprensión del sistema se muestra el Diagrama de Componentes, el cual expone una vista estática del sistema, que representa la organización y dependencia que existe entre los componentes físicos que se necesitan para ejecutar la aplicación, sean éstos componentes de código fuente, librerías, binarios o ejecutables.

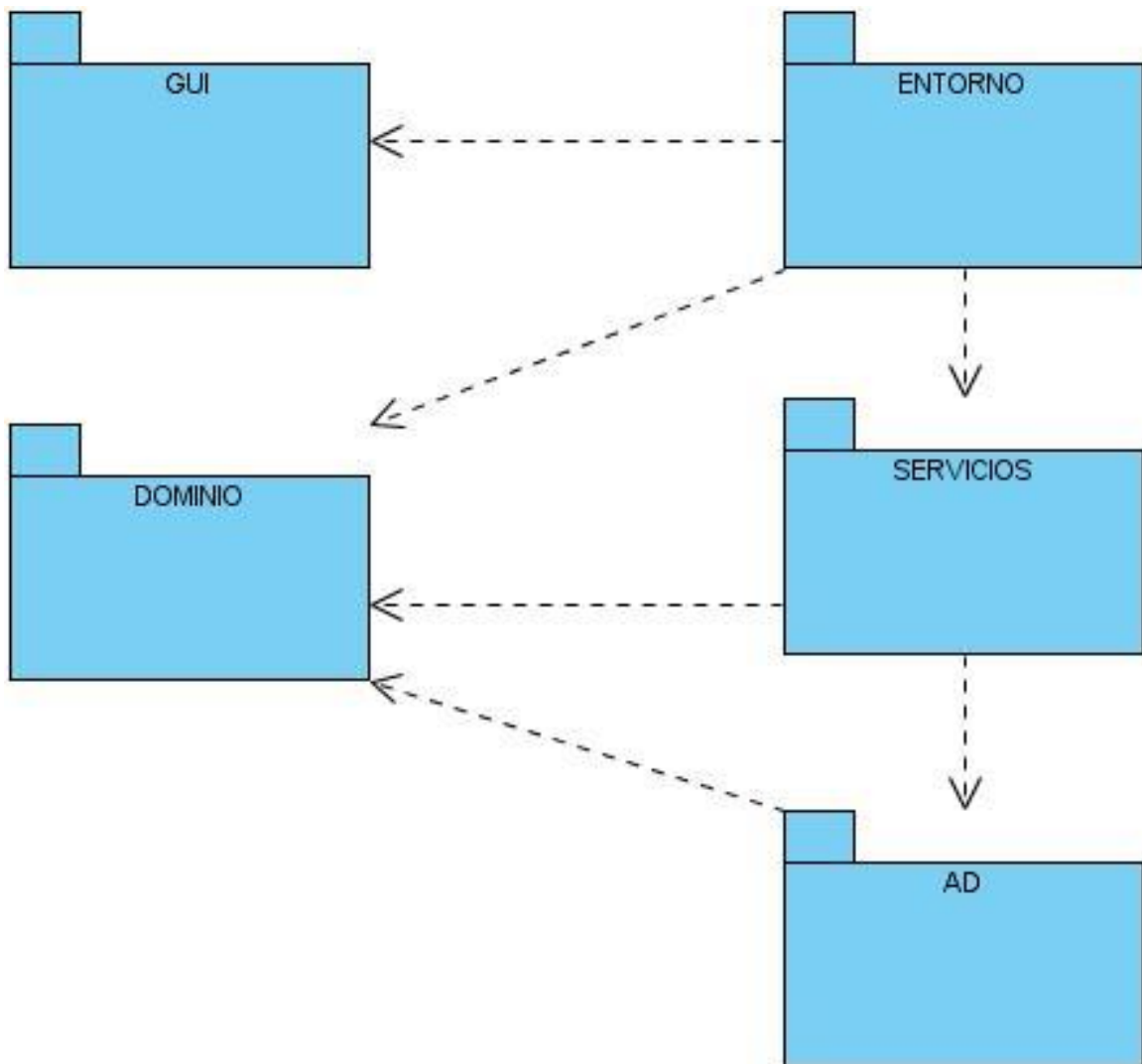


Figura 18 Diagrama General de Componentes

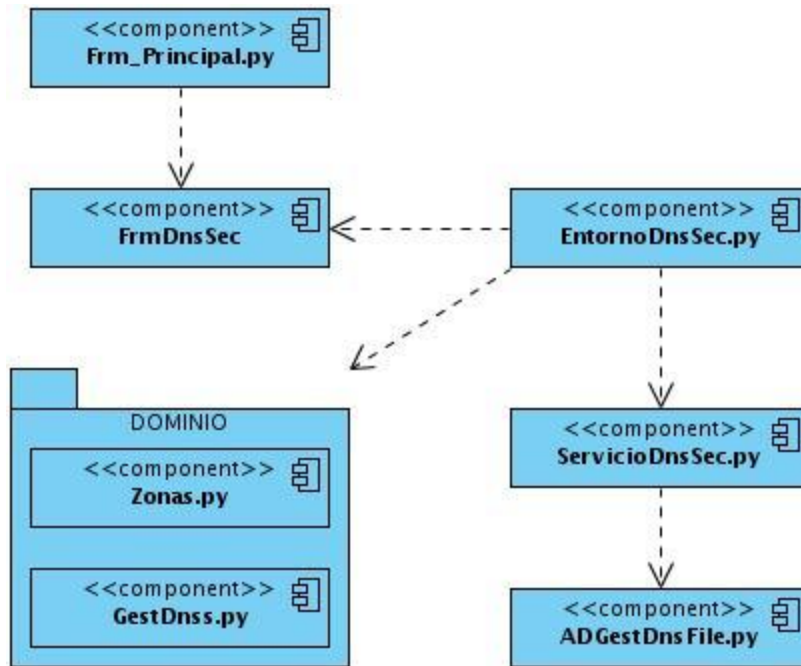


Figura 19 Diagrama de Componentes del CU Activar DNSSEC

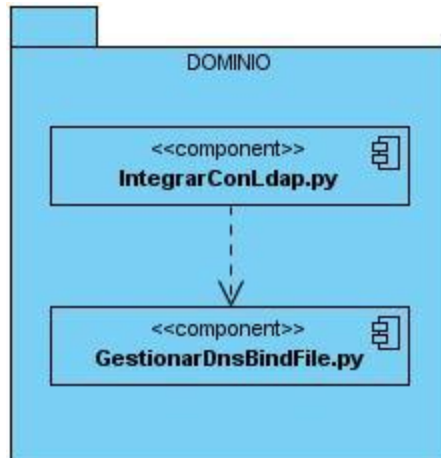


Figura 20 Diagrama de Componentes del CU Integrar con LDAP

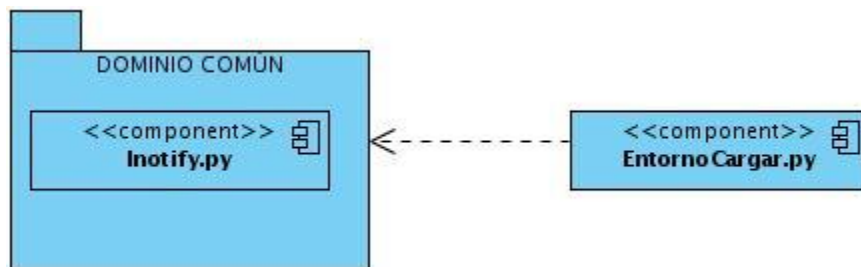


Figura 21 Diagrama de Componentes del CU Actualizar en Tiempo Real.

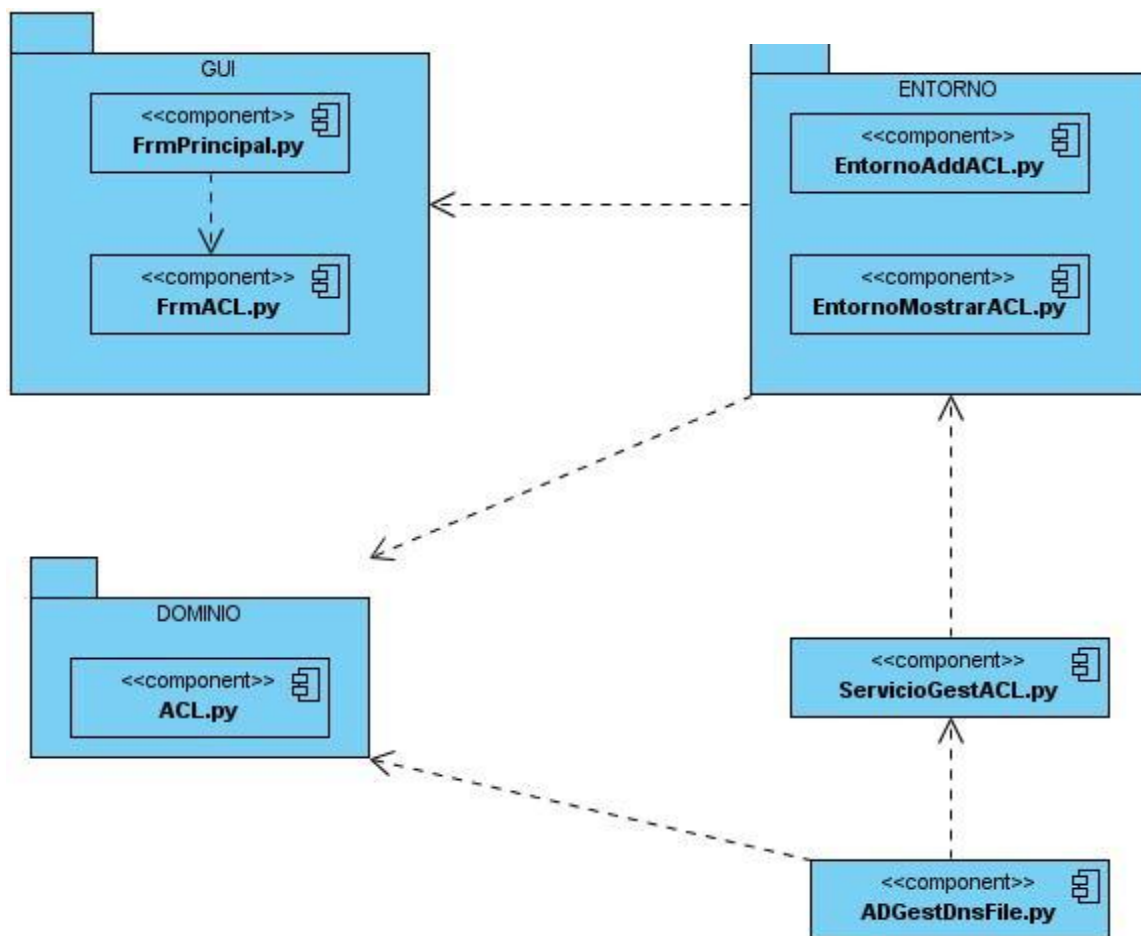


Figura 22 Diagrama de Componentes del CU Gestionar ACL.

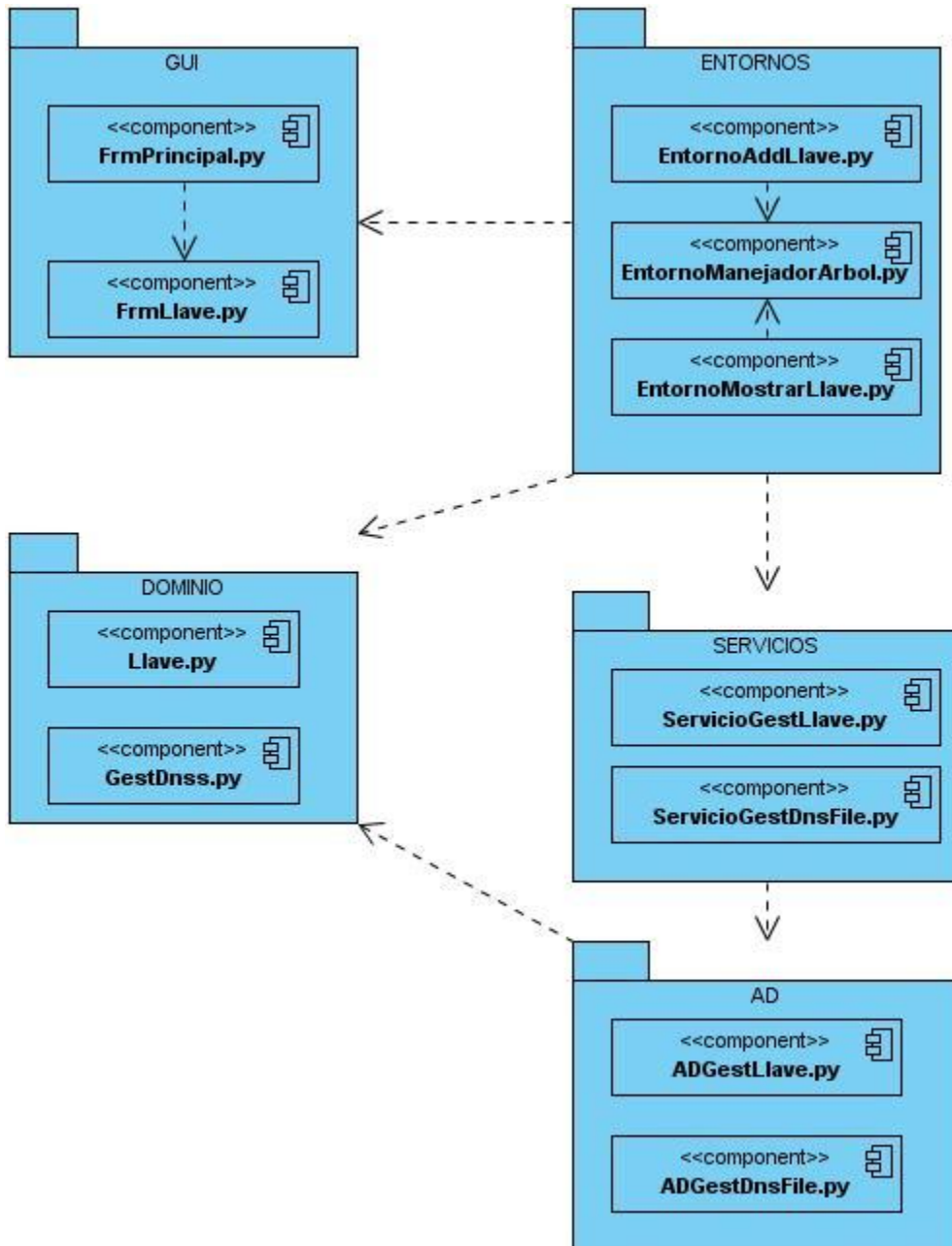


Figura 23 Diagrama de Componentes del CU Gestionar Llave

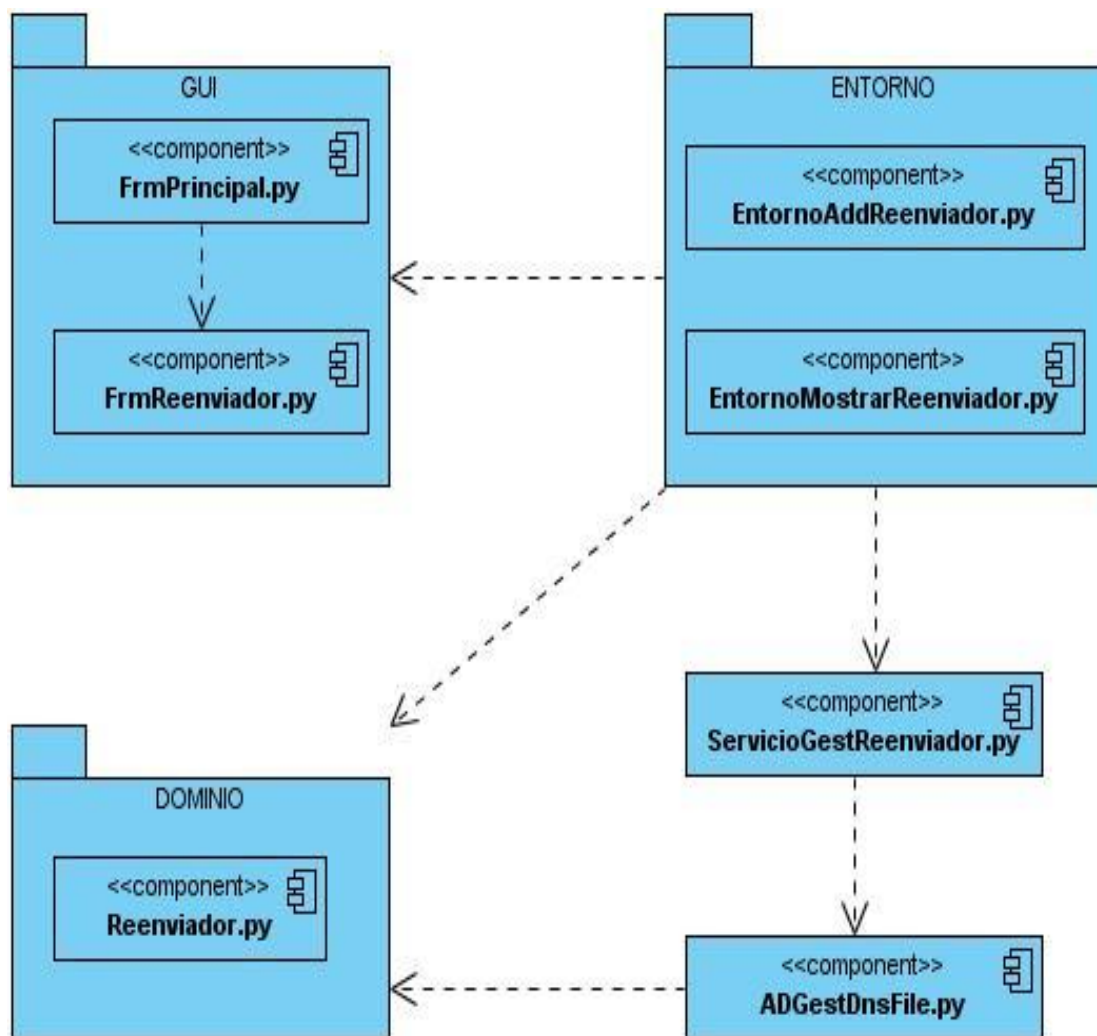


Figura 24 Diagrama de Componentes del CU Gestionar Reenviador.

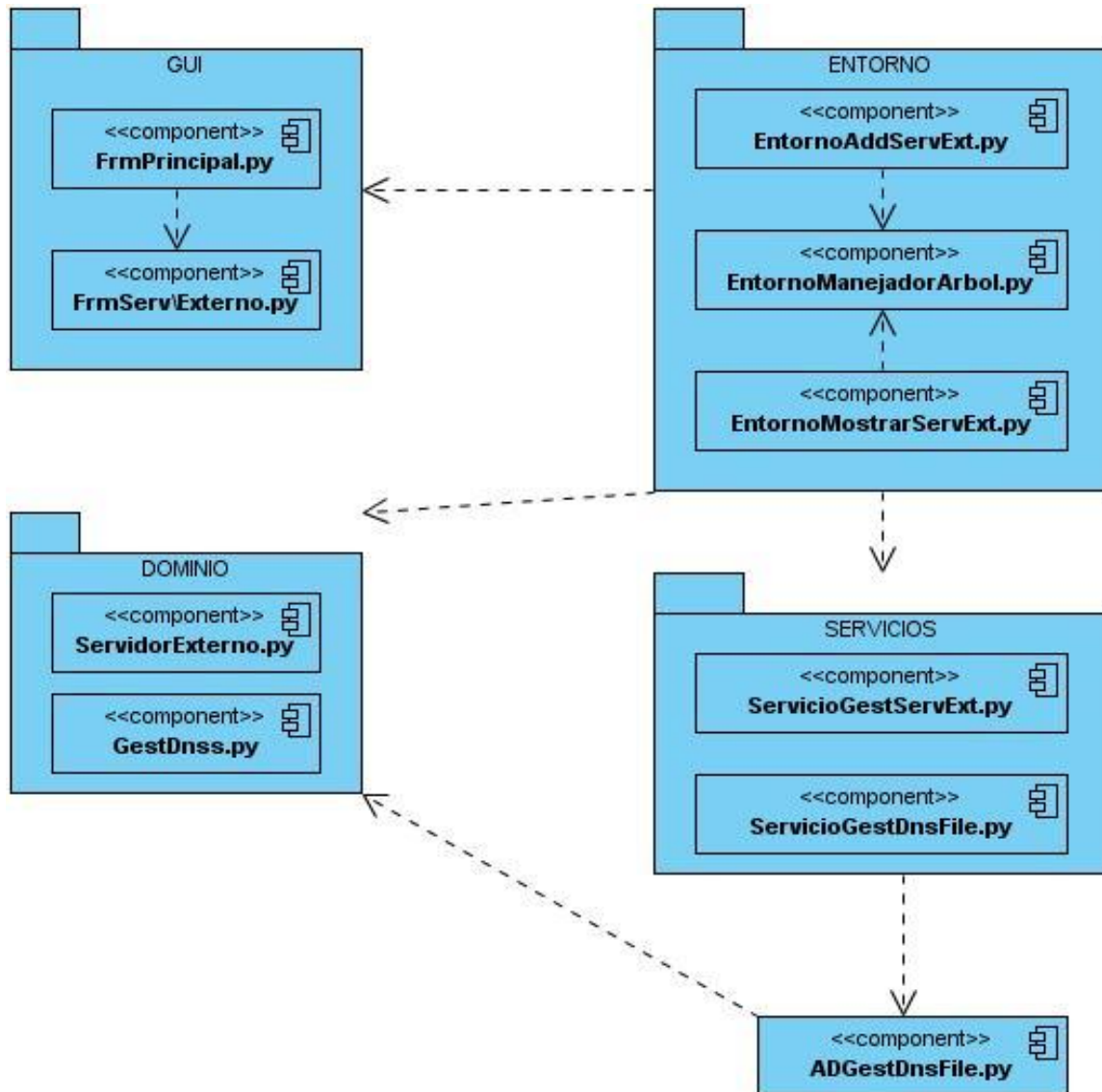


Figura 25 Diagrama de Componentes del CU Gestionar Servidor Externo.

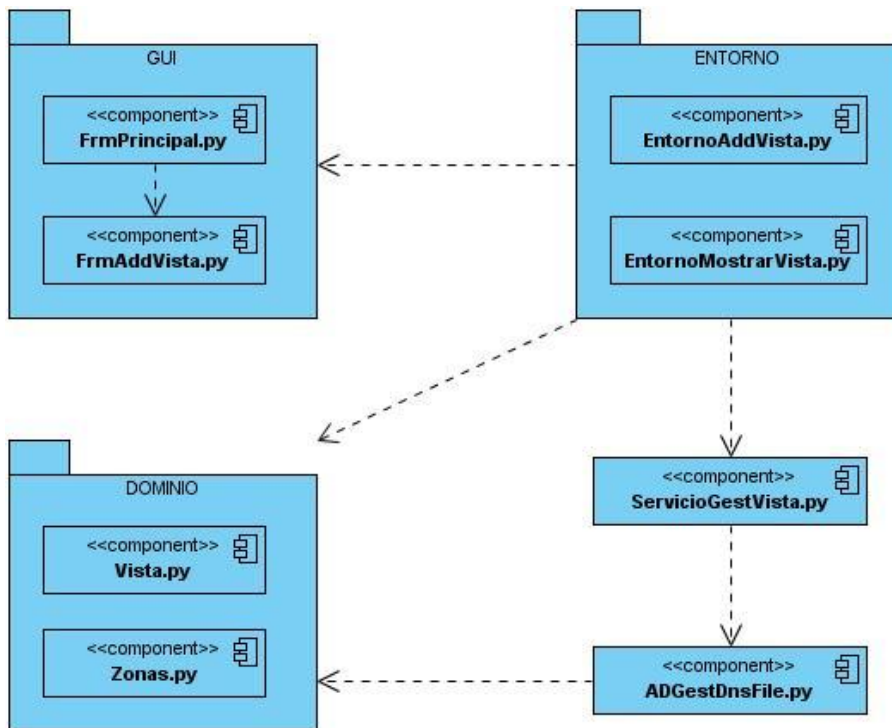


Figura 26 Diagrama de Componentes del CU Gestionar Vista.

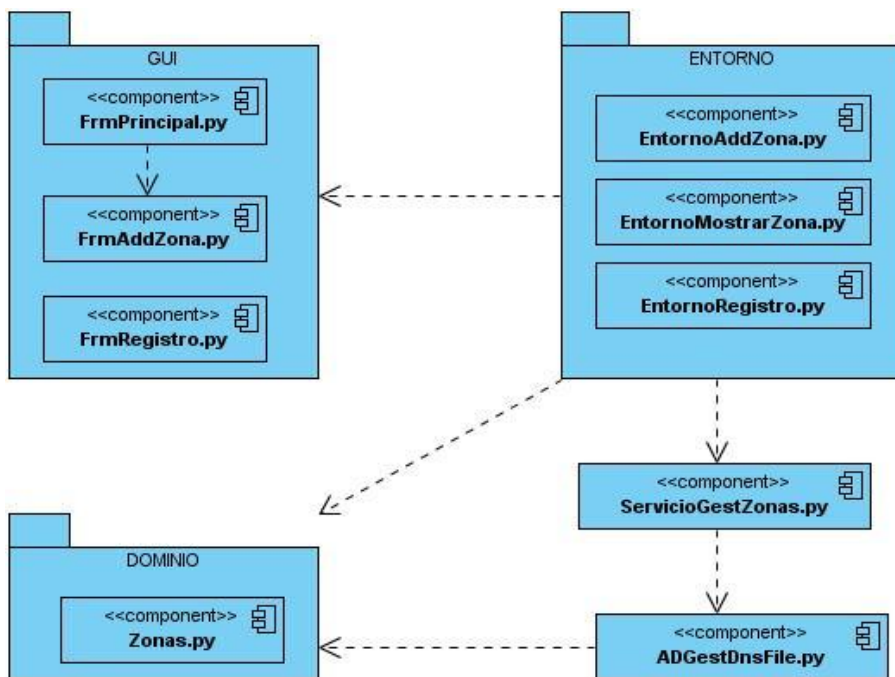


Figura 26 Diagrama de Componentes del CU Gestionar Zona.

4.3 Diagrama de Despliegue

En el presente diagrama se muestran las relaciones entre el hardware y el software en el sistema final mediante una representación de la arquitectura en tiempo de ejecución de los procesadores, dispositivos y los componentes de software que se ejecutan en la arquitectura propuesta. Representa además, la disposición en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un procesador, dispositivo o memoria, donde se ejecutan los componentes.

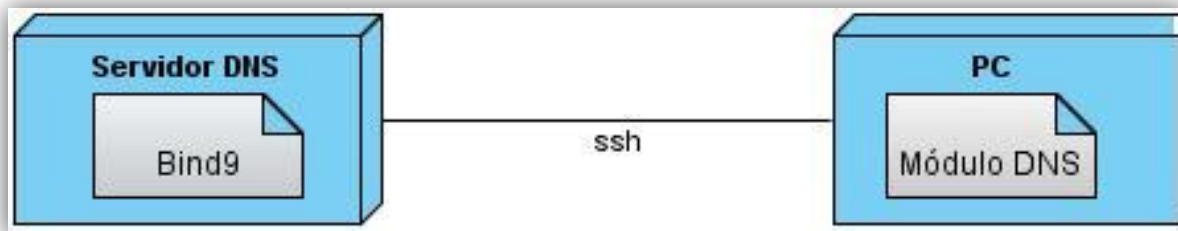


Figura 27 Diagrama de Despliegue

4.4 Estrategia de Prueba

Evaluar la calidad del software desarrollado constituye el principal objetivo del flujo de pruebas. Para ello se vale de la búsqueda y documentación de errores, así como la validación del desempeño y el cumplimiento de los requerimientos, brindando como respuesta una indicación de la calidad del producto. En el módulo DNSv3.0, se aplicarán pruebas de Sistema, las cuales están dirigidas a evaluar el software final. El objetivo es descubrir errores respecto al tratamiento de datos de la configuración del servidor, así como en el manejo de los datos de salida que produce el mismo. Así mismo con los errores respectivos a la escritura en el fichero de configuración ocurrido un acceso externo simultáneo. Para ello se definió la siguiente estrategia:

| Niveles de Pruebas | Tipos de pruebas | Métodos | Técnicas | Módulos |
|--------------------|-------------------------|------------|---------------------------|---------|
| Sistema | Funcionalidad-Función | Caja Negra | Partición de equivalencia | Todos |
| | Funcionalidad-Seguridad | | Partición de equivalencia | Todos |

Tabla 8 Estrategia de pruebas

Una vez definida la estrategia de pruebas se considerará exitosa si se encuentran errores no identificados hasta ese entonces. Las mismas culminarán una vez se hayan comprobado los resultados obtenidos para todos los tipos de entradas posibles (idóneas o no). Es necesario aclarar que para probar la funcionalidad de integrar con LDAP, se necesita de la colaboración del módulo Integrador, ya que las pruebas de caja negra se llevan a cabo sobre la interfaz de usuario, y la función de integración se lleva a cabo sobre una interfaz del módulo antes mencionado.

Pruebas al Caso de Uso Activar DNSSEC

| Escenario | Descripción | Variable 1 | Variable 2 | Variable 3 | Respuesta del | |
|----------------------------------|---|------------|------------|------------|---|--|
| | | | | | sistema | Flujo central |
| EC 1.1 Entrada ideal de datos | Microprocesador Core 2 Duo. 1 GB de memoria RAM. Sistema Operativo Ubuntu | V | V | V | El sistema muestra un cartel indicando que el proceso puede tardar unos minutos, mostrando posteriormente | El usuario selecciona los datos indicados. Para ello cuenta con dos opciones seleccionables y una en la que debe escribir el |
| | | RSASHA1 | ZONE | 768 | | |
| | | V | V | V | | |
| | | DSA | ZONE | 1024 | | |

| | | | | | | |
|---------------------------------------|---|---|---|---|---|--|
| | Maverick. | | | | otro que informa que el proceso se realizó con éxito. Las claves han sido generadas. | dato. Luego pulsa el botón Generar Claves. |
| EC 1.2 Entrada incorrecta de datos | Microprocesador Core 2 Duo. 1 GB de memoria RAM. Sistema Operativo Ubuntu Maverick. | I DH V RSASHA1 V RSASHA1 | V ZONE I HOST V ZONE | V 768 V 768 I 1000 | En el caso del algoritmo y el tipo de clave, se crea la misma, sin embargo no todos los tipos de algoritmos son válidos para firmar una zona, ni todos lo son los tipos de clave. En el caso del tamaño de clave, de entrar un dato incorrecto, el sistema da un error que el usuario no ve, sin embargo informa que la clave ha sido generada. | El usuario selecciona los datos indicados. Para ello cuenta con dos opciones seleccionables y una en la que debe escribir el dato. Luego pulsa el botón Generar Claves |

Tabla 9 Pruebas al Caso de Uso Activar DNSSEC

Para el caso de uso Activar DNSSEC la prueba se considera exitosa, puesto que se encuentran varias irregularidades en la respuesta del sistema, que quedan inscritas como No Conformidades. Para la sección Generar Claves, las No Conformidades quedan expuestas a continuación:

| Elemento | No | No conformidad | Etapas de detección | Clasificación | Estado NC |
|------------|----|--|---------------------|---------------|----------------------|
| Módulo DNS | 2 | En el caso de que las llaves no sean RSASHA1 o DS, el sistema no avisa al usuario que las llaves creadas no podrán firmar las zonas. | Prueba | R | 30-05-2011 RA |
| | 3 | En el caso de la creación de la clave con un tamaño de clave incorrecto el sistema no informa del error al usuario. | Prueba | R | 30-05-2011 RA |

Tabla 10 No Conformidades

4.5 Conclusiones

En el capítulo se presentaron los elementos correspondientes al modelo de implementación, destacando el diagrama general de componentes del sistema y los Diagramas de Componentes para cada Caso de Uso; se realizó un estudio de la configuración física sobre la que debe funcionar la aplicación, definiendo la relación del sistema con dispositivos de hardware en el Diagrama de Despliegue. Se definió además la estrategia de prueba a utilizar realizando las mismas al caso de uso Activar DNSSEC.

Conclusiones

En la realización del presente trabajo de diploma, se realizó un estudio detallado de las características del servidor de Nombres de Dominio, en el cual se identificaron funcionalidades que brinda el Bind9 y que podían ser incorporadas al Módulo DNSv2.0 de la Plataforma de Servicios Telemáticos para GNU/Linux. Para ello, se estudiaron las distintas vías de implementar dichas funcionalidades, tomando las más efectivas para implementar, logrando una gestión de los servicios cómoda y eficaz y se realizó un refinamiento de las funcionalidades de la versión anterior del módulo. Se desarrolló una herramienta de gestión de servidores de nombres de dominio multiplataforma, que además de los servicios básicos, permite la gestión de todas sus funcionalidades para protocolos tanto IPv6 como IPv4, la actualización en tiempo real de los datos de la aplicación una vez realizado un cambio ajeno a la misma, así como la incorporación del sistema de transferencia de objetos del DNS o DNSSEC y la integración con el servidor LDAP, todo ello mediante una intuitiva interfaz gráfica amigable, fácil de utilizar y adecuada a las necesidades de los administradores de red cubanos. Por tanto, se consideran cumplidos los objetivos trazados al inicio de este trabajo de diploma. Se espera sea de ayuda a quienes deseen continuar apoyando el proceso de migración en Cuba y la calidad del trabajo de los administradores de red.

Recomendaciones

A lo largo de la investigación realizada se encontraron otras funcionalidades de Bind9 que pudieran ser de ayuda para garantizar la seguridad y un mejor funcionamiento del Servidor de Nombres de Dominio. Una de ellas es TSIG, la cual garantiza la transmisión encriptada de los datos entre servidor maestro y esclavo. Es por ello que la recomendación es añadir al módulo la funcionalidad de activar esta funcionalidad, para que junto a DNSSEC, garanticen la integridad, y el origen de los datos recibidos en cada consulta. Se recomienda además, añadir la posibilidad de descargar y adicionar opciones actualizadas de dominios firmados por otras organizaciones para maximizar la cobertura DNSSEC una vez creada la cadena de confianza.

Bibliografía

[1] Universidad de San Carlos de Guatemala; Facultad de Ingeniería. UML.

[Disponible en: www.scribd.com/doc/2080534/UML.pdf]

[2] González Duque, Raúl. Python para todos.

[3] Python.org [Online]

[Disponible en: www.python.org]

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

[Disponible en: biblioteca.uci.cu/titdigitales.htm#iqs]

[5] James Rumbaugh, Ivar Jacobson, Grady Booch. *El Lenguaje Unificado de Modelado. Manual de Referencia*, 1998.

[Disponible en: bibliodoc.uci.cu/pdf/reg03050.pdf]

[6] Hernández Orallo, Enrique. *El Lenguaje Unificado de Modelado*.

[Disponible en: www.disca.upv.es/enheror/pdf/ActaUML.PDF]

[7] Redhat.com [Online]

[Disponible en: www.es.redhat.com]

[8] Tornés Medina, Yordanis; IDEF. *Una alternativa para modelamiento de negocio con RUP*; UCI.

[9] Colectivo de autores UCI. *Introducción a la disciplina de análisis y diseño*. 2011

[10] Sommerville, Ian. *Ingeniería del software*.

[11] Jacobson, Ivar; Rumbaugh, James; Booch, Grady, "El proceso unificado de desarrollo de software". 2000.

- [12] Rumbaugh, James, Jacobson, Ivar; Booch, Grady, "El lenguaje unificado de modelado. Manual de referencia".2000.
- [13] Pressman, Roger; "Ingeniería de software. Un enfoque práctico." 2002
- [14] Brueck, Dave; Tanner, Stephen, "Python 2.1 Bible". 2001. Hungry Minds.
- [15] Maldonado, Daniel M. "PyQt. Desarrollando aplicaciones de escritorio". 2008.
- [16] Boddie, David. "Creating GUI Applications with PyQt and Qt Designer". 2007. Birmingham.
- [17] Ward, Greg. "Distribución de módulos Python". 2000
- [18] Matloff, Norman; Hsu, Francis. "Tutorial on Threads Programming with Python". 2007.
- [19] Núñez Zuleta, José Vicente. "DNS (Domain Name System)". 2007.
[Disponible en: www.ing.ula.ve]
- [20] Thompson, Kerry. "DNS Security II: DNSSEC"
[Disponible en: www.crypt.gen.nz/papers/dns_security_2.html]
- [21] DNSSEC.net
[Disponible en: www.dnssec.net]
- [22] Lins, Emanuela; Agarwal, Saurabh. " A study of IPv6 Implementation in Linux". 2003
- [23] Larman, C. "UML y patrones". Tomo I Capítulo 18.
- [24] Visconti, Marcello; Astudillo, Hernán. "Fundamentos de Ingeniería de Software". Universidad Técnica Federico Santa María
[Disponible en www.inf.utfsm.cl]
- [25] Osorio Rojas, Alan D. "Qt 4 Manual introductorio". 2008
- [26] Documentación de Python v2.6.2. 2009.
- [27] van Rossum, Guido. "Guía de estilo del código Python". 2007
- [28] Billy Reynoso, Carlos. " Introducción a la Arquitectura de Software". 2004

Glosario de términos

A

Apache: Es un servidor web HTTP de código abierto para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP.

B

Bind: Berkeley Internet Name Domain, o Nombres de Dominio de Internet de Berkeley. Es uno de los servidores DNS más usados en Internet, especialmente en sistemas Unix.

D

DHCP: Protocolo de configuración de host dinámico es un protocolo que permite configurar dinámicamente los parámetros de red necesarios para que un sistema pueda comunicarse.

DNS: Domain Name System es una base de datos distribuida, con información que se usa para traducir los nombres de dominio, fáciles de recordar y usar por las personas, en números de IP que es la forma en la que las máquinas pueden encontrarse en Internet.

G

GPL: GNU General Public License, licencia creada por la Free Software Foundation a mediados de los años 80, orientada principalmente a proteger la libre distribución, modificación y uso del software. Su propósito es declarar que el software cubierto por esta licencia es Software Libre.

M

MySQL: Sistema de gestión de base de datos relacional, multiplataforma y bajo licencia GNU GPL.

O

Open Source: Término con el que se conoce al software distribuido y desarrollado libremente. Sin embargo, mucho de este software no da a sus usuarios la libertad de distribuir sus modificaciones, restringe el uso comercial, o en general restringe los derechos de los usuarios.

P

PHP: PHP Hypertext Pre-processor es un lenguaje de programación interpretado generalmente usado para la creación de aplicaciones para servidores, o creación de contenido dinámico para sitios web.

Q

Qt: Biblioteca multiplataforma para desarrollar interfaces gráficas de usuario desarrollada por la compañía noruega Trolltech.

S

Samba: Es una implementación libre del protocolo de archivos compartidos de Microsoft Windows para sistemas de tipo UNIX. También permite validar usuarios haciendo de Controlador Principal de Dominio (PDC).

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

SSL: Secure Sockets Layer o Protocolo de Capa de Conexión Segura, proporciona autenticación y privacidad de la información mediante comunicaciones seguras por una red, comúnmente Internet con el uso de criptografía.

W

Widgets: es un componente gráfico, o control, con el cual el usuario interactúa, como por ejemplo, una ventana, una barra de tareas o una caja de texto.