



*Herramienta informática para la administración remota del
servicio de resolución de nombres de dominio DNS.*

Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas

Autor:

Julio César González Rodríguez

Tutores:

MSc. Yasiel Pérez Villazón (Prof.Inst)

Ing. Zuleimys García Rodríguez

La Habana, agosto 2020



“...No hay más que asomarse a las puertas de la tecnología y la ciencia contemporáneas para preguntarnos si es posible vivir y conocer ese mundo del futuro sin un enorme caudal de preparación y conocimientos.”

Fidel Castro Ruz.

Dedicatoria

Dedicatoria

El presente trabajo está dedicado a las dos personas más importantes de mi vida, si ellos no hubiese sido posible:

A mi querida madre que siempre me ayudo a enfrentar todos los retos y nunca dejó de ayudarme ni por la más mínimo. Me supo guiar hasta el final y le agradezco por todo el esfuerzo que dedicó en mí.

A mi grandísimo padre que siempre lo he tenido en lo más alto porque para mí es la persona mas inteligente que conozco y siempre lo será.

Declaración de Autoría

Agradecimientos

Este trabajo ha sido una gran carrera donde ha influido muchas personas:

Mis padres, ellos siempre han sido mi guía y gracias a ellos por llevarme siempre por el buen camino y de formarme como un profesional.

A mi hermana menor Mary Alejandra que siempre me ha apoyado desde lejos en los momentos difíciles durante estos cinco años.

A mi hermano mayor Diegui que me aconsejo de la vida y que me enfocara siempre en los estudio, que para otras cosas hay tiempo.

A mi familia por estar siempre cuando más los necesitaba.

A mi querida novia y mejor amiga Lily que es la persona que más quiero y amo de este mundo y le doy gracias por el apoyo que me dio aquí en la UCI y por ser la persona con quien pase mis mejores momentos en la universidad.

Le agradezco a mis tutores Yasiel Pérez Villazón y Zuleymis García Rodríguez por dedicarme tiempo y esfuerzo para poder cumplir esta meta

A mis amistades de la UCI, para los que se quedan junto conmigo y para los que se me van lejos, se que siempre estaremos en contacto y nos veremos de nuevo.

Un agradecimiento especial para mis amigos del aula que estuvimos juntos desde el principio y siempre nos ayudamos todos en las pruebas para salir bien.

A mis grandes amigos que los quiero como mis hermanos, Raidel Carbonell, Noel Luis Martínez, y Alvaro Castro.

Y un agradecimiento bien grande al claustro de profesores de la Facultad 1 gracias por brindarme sus conocimientos y experiencias profesionales.

Le agradezco a la UCI por abrimme sus puertas y formarme como ingeniero en ciencias informáticas.

Declaración de Autoría

Declaración de Autoría

Declaro por este medio que yo **Julio César González Rodríguez**, con carné de identidad **96120618846** soy el autor principal del trabajo titulado "**Herramienta informática para la administración remota del servicio de resolución de nombres de dominio DNS**" y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firman la presente a los ____ días del mes de agosto del año 2020.

Julio César González Rodríguez

Autor

Ing. Zuleimys García Rodríguez

Tutora

MSc. Yasiel Pérez Villazón

Tutor

Resumen

La presente investigación sostuvo como objetivo el desarrollo de una herramienta para la administración remota del servicio de resolución de nombres de dominio DNS, de forma que facilite el trabajo de los administradores de redes en el uso de este servicio. Se emplearon los métodos Analítico-Sintético, Modelación y Observación. Se analizaron las diversas herramientas que permiten la administración del servicio DNS y su uso en la conexión remota a otro servidor. La metodología de software que se utilizó para el desarrollo de la propuesta de solución fue el Proceso Unificado Ágil en su variación para la Universidad de Ciencias Informáticas y las principales tecnologías empleadas fueron: Visual Studio Code como editor de texto, JS como lenguaje de programación, Vue como framework de desarrollo web, Electron como framework de desarrollo de aplicaciones de escritorio y Visual Paradigm para el modelado de los procesos. La herramienta permite la gestión de las zonas y registro de un dominio de forma remota a través de una interfaz de escritorio, mediante la cual agiliza el proceso y evita errores humanos por parte de los administradores y disminuye la complejidad de hacer el proceso manualmente.

Palabras clave: gestión, administración remota, servicio DNS, zonas, registros.

Índice

INTRODUCCIÓN	4
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA SOBRE LA ADMINISTRACIÓN REMOTA DEL SERVICIO DE RESOLUCIÓN DE NOMBRES DE DOMINIO DNS.	9
1.1 CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.....	9
1.1.1 Consultas DNS.....	15
1.1.2 Funcionamiento del DNS.....	15
1.1.3 Tipos de servidores DNS.....	16
1.2 ANÁLISIS DE SERVIDORES DE NOMBRES QUE IMPLEMENTAN EL SERVICIO DNS.....	17
1.2.1 Dnsmasq.....	17
1.2.2 Unbound.....	18
1.2.4 PowerDNS.....	18
1.2.3 CoreDNS.....	19
1.2.5 Pihole.....	20
1.2.6 BIND9.....	20
1.2.7 Justificación del servidor de nombres seleccionado.....	21
1.3 ANÁLISIS DE HERRAMIENTAS DE ADMINISTRACIÓN DEL SERVICIO DE RESOLUCIÓN DE NOMBRES EN GNU/LINUX.	24
1.3.1 Webmin.....	24
1.3.2 Centos Web Panel.....	24
1.3.3 Virtualmin.....	24
1.4 METODOLOGÍAS DE DESARROLLO DE SOFTWARE.....	25
1.4.1 Variación AUP para la UCI.....	25
1.5 LENGUAJES Y HERRAMIENTAS PARA EL DESARROLLO DE LA HERRAMIENTA.....	26
1.5.1 Vue v4.1.2.....	26
1.5.2 Vuetify 1.14.....	28
1.5.3 JavaScript v1.8.5.....	28
1.5.4 Electron v7.1.2.....	28
1.5.5 Visual Studio Code v1.38.1.....	29

Índice

1.5.6 UML v2.5	29
1.5.7 Visual Paradigm v13.0	29
CONCLUSIONES PARCIALES	29
CAPÍTULO 2: ANÁLISIS Y DISEÑO DE LA HERRAMIENTA PARA LA ADMINISTRACIÓN REMOTA DEL SERVICIO DE RESOLUCIÓN DE NOMBRES DE DOMINIO DNS.	31
2.1 DESCRIPCIÓN GENERAL DE LA PROPUESTA DE SOLUCIÓN DE LA INVESTIGACIÓN	31
2.2 MODELO CONCEPTUAL	31
2.3 DESCRIPCIÓN DE REQUISITOS DEL SISTEMA	33
2.3.1 Requisitos funcionales	33
2.3.2 Requisitos no funcionales	35
2.4 HISTORIAS DE USUARIO	36
2.4.1 Historia de Usuario Adicionar Zona	36
2.4.2 Historia de Usuario Eliminar Registro	38
2.4.3 Historia de Usuario Iniciar Servicio Bind9	40
2.5 DISEÑO DE LA HERRAMIENTA PARA LA ADMINISTRACIÓN REMOTA DE SERVICIOS DNS EN GNU/LINUX/NOVA	41
2.5.1 Diseño Arquitectónico	41
2.5.2 Diagrama de Clases del Diseño	43
2.6 PATRONES DE DISEÑO	44
2.6.1 Patrones GRASP	44
2.6.2 Patrones GoF	45
2.6.3 Patrón DAO	46
CONCLUSIONES PARCIALES	46
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBAS DE LA HERRAMIENTA PARA LA ADMINISTRACIÓN REMOTA DEL SERVICIO DE RESOLUCIÓN DE NOMBRES DE DOMINIO DNS.	47
3.1 INTRODUCCIÓN	47
3.2 IMPLEMENTACIÓN	47
3.3 ESTÁNDAR DE CODIFICACIÓN	47
3.4 DIAGRAMA DE COMPONENTES	49
3.5 DIAGRAMA DE DESPLIEGUE	50
3.6 PRUEBAS	51

Índice

3.6.1 Pruebas unitarias	51
3.6.2 Pruebas de validación	52
3.6.3 Pruebas de rendimiento	55
3.6.4 Prueba de aceptación.....	58
CONCLUSIONES PARCIALES	60
CONCLUSIONES GENERALES	61
RECOMENDACIONES	62
REFERENCIAS BIBLIOGRÁFICAS	63
ANEXOS	66

Introducción

La aparición de Internet con el surgimiento de la *World Wide Web*¹, trajo consigo servicios como correo electrónico, transferencias de archivos, voz sobre IP (*Internet Protocol*, Protocolo de Internet), la Web, videoconferencias, consultas en línea y los servicios de mensajería instantánea (chat). En los últimos años el uso de estos servicios ha tenido un elevado impacto a escala global y han evolucionado vertiginosamente convirtiéndose en una herramienta esencial tanto en las actividades que desarrollan los sectores productivos y de servicios, como en la vida cotidiana de la sociedad (1).

En Internet, los paquetes de datos viajan mediante el protocolo de red IP, que tanto en su versión 4 como en 6, funciona mediante lo que se conoce como direcciones IP, que no es más que un conjunto de números que identifican de forma unívoca al ordenador al que deseamos enviar un paquete de datos. Sin embargo y de forma general, se utilizan nombres para identificar a los ordenadores (*www.google.es*, *www.debian.org*, *uci.cu*, etc.). Estos nombres sencillos de recordar deben ser convertidos en direcciones IP para poder conocer el ordenador destino. Para cumplir este proceso es que surge el servicio DNS (*Domain Name System*, Sistema de Nombres de Dominio) que permite convertir el nombre de cualquier ordenador en su dirección IP y viceversa.

Congruentemente, el DNS es un sistema global, escalable y jerárquico. Ofrece una base de datos dinámica que asocia direcciones IP de computadoras, servicios o cualquier recurso conectado a Internet o red privada con información de diversos tipos; la información se almacena en forma de registros de distintos tipos que contienen direcciones IP, dirección de correo, nombre de dominio, entre otros. Esta información se agrupa en zonas, que corresponden a un espacio de nombres de dominio y que son mantenidas por su servidor principal DNS (2).

La asignación de DNS se distribuye a través de Internet en una jerarquía de autoridad. Los proveedores de acceso y las empresas, así como instituciones, universidades y otras organizaciones, por lo general tienen sus propios rangos asignados de direcciones IP y nombres de dominio. Suelen ejecutar servidores DNS para administrar el mapeo de nombres a esas direcciones. Por consiguiente, el Sistema de Nombres

¹ World Wide Web o (red informática mundial): Sistema de distribución de documentos de hipertexto o hipermedia interconectados y accesibles a través de Internet.

Introducción

de Dominio proporciona un mecanismo que permite la gestión y el balanceo de carga para la traducción de los nombres de dispositivos (computadoras personales, servidores, routers) conectados a una red, hacia sus respectivas direcciones IP y viceversa.

En la actualidad internet cuenta con herramientas que permiten gestionar de manera centralizada el DNS y proporcionan servicios prácticos para administrar el sistema. Tal es el caso de Windows Server, un software privativo para sistema operativo Windows el cual posee un elevado costo de licencia. No obstante, en Cuba se buscan alternativas dirigidas a impulsar la migración del software privativo al software libre; y como parte fundamental sus servicios de red. Para ello dirige sus objetivos hacia la materialización de proyectos asociados al programa de informatización de la sociedad y fortalecer el desarrollo económico y sostenible del país. Dispone de diferentes entidades, entre las que se encuentra la Universidad de Ciencias Informáticas (UCI), que tiene como misión producir aplicaciones y servicios informáticos que sirvan de soporte a la industria cubana de la informática. En ese sentido, el Centro de Software Libre (CESOL) de la UCI brinda un conjunto de soluciones informáticas, entre ellas, la distribución cubana GNU/Linux Nova. Desde el año 2006 un conjunto de especialistas de CESOL mediante un proceso de migración hacia software libre realizado en diferentes Organismos de la Administración Central del Estado (OACE) han desplegado Nova como sistema operativo que contribuye a desarrollar el nivel de organización, eficiencia y eficacia de la industria cubana de software, de forma tal que satisfaga las necesidades de informatización de la sociedad, con un nivel de competitividad acorde a los estándares internacionales, logrando potenciar las exportaciones de software.

Estas entidades utilizan herramientas de administración del sistema que tienen implementado el servicio DNS. Sin embargo, aún es insuficiente la configuración y administración del servicio de resolución de nombres de dominio, el cual se realiza de forma manual a través de una interfaz de consola, lo que trae como resultado pérdida de tiempo en el despliegue del servicio y errores de configuración en los ficheros, tampoco se hace un uso efectivo de la administración remota, lo cual impide gestionar varios servidores a la vez.

A partir de lo anteriormente expuesto, se plantea como **problema científico**:

¿Cómo facilitar la administración remota del Servicio de Resolución de Nombres de dominio DNS en la distribución cubana GNU/Linux Nova?

Introducción

Para guiar la investigación se propone como **objeto de estudio**: El Servicio de resolución de nombres de dominio DNS, enmarcado en el **campo de acción**: administración remota del servicio de resolución de nombres de dominio DNS en Nova".

Para dar respuesta al problema científico se plantea como **objetivo general**:

Desarrollar una herramienta de escritorio para la administración remota del servicio de resolución de nombres de dominio DNS en la distribución cubana GNU/Linux Nova. A partir del objetivo general se definen los siguientes **objetivos específicos**:

- Elaborar el marco teórico referencial de la investigación teniendo en cuenta las tecnologías asociadas a la administración del servicio de resolución de nombres de dominio DNS.
- Describir el análisis y diseño de la solución para la administración del servicio de resolución de nombres de dominio DNS.
- Desarrollar la solución para la administración remota del servicio de resolución de nombres de dominio DNS.
- Validar la solución para la administración remota del servicio de resolución de nombres de dominio DNS.

Para guiar el cumplimiento del objetivo planteado, se formulan las siguientes **preguntas científicas**:

1. ¿Cuáles son los referentes teóricos relacionados con la administración remota del servicio de resolución de nombres de dominio DNS?
2. ¿Cuál es el estado actual de las herramientas informáticas para la administración remota del servicio de resolución de nombres de dominio DNS?
3. ¿Cuál es la estructura que debe poseer la herramienta informática para la administración del servicio de resolución de nombres de dominio DNS?
4. ¿Cómo validar la propuesta para la administración del servicio de resolución de nombres de dominio DNS en la distribución cubana GNU/Linux Nova?

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

Introducción

- Elaboración el diseño teórico metodológico sobre la administración remota del servicio de resolución de nombres de dominio DNS.
- Análisis de las tecnologías de desarrollo a utilizar en la herramienta para la administración remota del servicio de resolución de nombre de dominio DNS en la distribución cubana de GNU Linux/nova.
- Estudio diagnóstico de las herramientas para la administración remota del servicio de resolución de nombre de dominio DNS.
- Definición de la propuesta de solución y requisitos funcionales y no funcionales a implementar.
- Diseño de la herramienta para la administración remota del servicio de resolución de nombre de dominio DNS en la distribución cubana de GNU/Linux Nova.
- Implementación de la herramienta para la administración remota del servicio de resolución de nombre de dominio DNS en la distribución cubana de GNU/Linux Nova.
- Validación de la herramienta para la administración remota del servicio de resolución de nombre de dominio DNS en la distribución cubana de GNU/Linux Nova.

Durante el desarrollo de la investigación se utilizan los métodos teóricos y empíricos de la investigación científica que se exponen a continuación.

A nivel teórico:

- Analítico-Sintético: Se analizan las funcionalidades de las herramientas, las librerías que son útiles para desarrollar la aplicación, la descomposición del sistema de transferencias de zonas y la gestión de los registros del DNS, con el objetivo de determinar el funcionamiento del sistema y comprender cómo añadirle las funcionalidades necesarias sin perjudicar su funcionamiento.
- Modelación: Se utiliza para crear abstracciones, fundamentalmente del diseño, de la propuesta de solución.

A nivel empírico:

Introducción

- Observación: Utilizado para realizar pruebas con las herramientas existentes, observando sus fallos y funcionamiento correcto, lo cual permite obtener una visión general de las características de estas herramientas.

El presente trabajo está estructurado en introducción, tres capítulos, conclusiones generales y referencias bibliográficas.

El primer capítulo "Fundamentación teórica para una herramienta de administración remota de servicio de nombres de dominio DNS en GNU/Linux Nova" incluye el estado del arte de las aplicaciones que existen en la actualidad para la administración de servidores DNS y de servicios, además se definen técnicas, herramientas, lenguajes de programación y la metodología a utilizar.

En el segundo capítulo "Características de la herramienta de administración remota de servicios DNS en GNU/Linux/Nova" se define una propuesta del sistema y las funcionalidades a desarrollar. Se determinan además las historias de usuario, patrones de diseño y se define la arquitectura de la herramienta.

En el tercer capítulo "Implementación y pruebas" se documentan componentes asociados a la aplicación y se diseña y controlan los casos de prueba del sistema.

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

Capítulo 1: Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS.

En este capítulo se definen los conceptos asociados al servicio de resolución de nombres de dominio DNS. Además, se hace un estudio sobre el estado actual de las herramientas de administración de la configuración y se reflejan las tecnologías y metodologías propuestas para el desarrollo de la herramienta para la administración remota del servicio DNS.

1.1 Conceptos asociados al dominio del problema.

Según el autor del libro *Remote Administration Tools* de Mohammad Hajjar y otros autores donde plantean que: una herramienta de administración remota es un programa usado para conectar un ordenador a través de una red local y ejecutar las actividades deseadas en otro ordenador permitiéndose la configuración de servicios del sistema. Se basa en la tecnología servidor y cliente. La parte del servidor se ejecuta en un ordenador controlado y recibe indicaciones del cliente, el cual se instala en el host remoto (3). Algunas de sus principales características son:

- Instalaciones de software.
- Gestión de archivos y ejecución de programas.
- Configuración de servicios de red como DHCP y DNS.

Yoedusvany Hernández Mendoza en una revista científica plantea que, un servidor de nombres es un programa para servidores que contienen información sobre la estructura del árbol de dominios y la información. Además, puede almacenar en caché la estructura y la información sobre cualquier dominio del árbol, pero en general un servidor de nombres tiene información completa sobre un subconjunto del espacio de dominio. Un servidor de nombres es autoridad para los espacios de nombres. La información

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

autorizada es organizada en unidades de zonas y estas zonas son distribuidas a los servidores de nombres que propician servicio redundante para los datos de una zona.

La investigación se centra específicamente en los servidores DNS, por lo que, un **servidor DNS** es un servidor que traduce nombres de dominio a direcciones IP y viceversa. Para ello, el servidor DNS dispone de una base de datos en la cual se almacenan todas las direcciones IP y todos los nombres de los ordenadores pertenecientes a su dominio. No existe una base de datos única donde se almacenan todas las direcciones IP existentes en el mundo, sino que cada servidor almacena las direcciones IP correspondientes a su dominio (4).

Una **zona DNS** es la parte de un dominio que es administrada por una organización. Una zona y su dominio asociado no tienen por qué coincidir (5). Un dominio puede constar de una sola zona, pero los dominios extensos suelen estar divididos en varias. La información de la zona autorizada se organiza en ficheros llamados 'zones' y estas zonas se pueden distribuir a los servidores de nombres que proporcionan un servicio redundante para los datos de una zona (6).

Un **archivo de zona** es un archivo de texto que describe el contexto de una zona DNS. Dentro de ese contexto se define la estructura jerárquica del dominio, tal como relación entre nombre de dominio y direcciones IP. Un archivo de zona puede ser utilizado tanto para describir una zona como para listar el contenido de un caché de DNS. También puede contener varios registros RR (*Resource Record* ó Registro de Recursos) de distintos tipos los cuales pueden almacenar dirección IP u otro tipo de información que definen dónde están para el dominio en cuestión (7).

Un aspecto importante para la administración DNS son los **registros DNS**. Las zonas, con los espacios de nombres, se almacenan en una base de datos formada por uno o más registros de recursos y que pueden estar alojados en uno o más servidores DNS (8). Estos registros contienen cadenas de texto que definen algún valor dentro de la zona. Cada cadena puede ser de varios tipos:

- **Comentarios:** comienzan con un punto y coma (;) es incluyen notas aclaratorias sobre el fichero.

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

- **Directivas:** Comienzan con el signo de dólar (\$) al comienzo y una palabra clave. Las directivas más usadas son:

1. \$ORIGIN: define el nombre de dominio que será incluido al final de cualquier nombre que se defina en el RR y que no acabe en el (.). Esta directiva no es obligatoria y si se omite, el software deduce el origen a partir de la referencia del archivo de configuración del servidor `/etc/bind/named.conf`.
2. \$TTL (Time to Live): Es el valor del tiempo de vida en que un RR es válido. Si el cliente DNS no especifica el tiempo, se utilizará el valor por defecto definido en la cabecera del archivo.

- **Registros de Recursos:** Definen las características o las entidades de la zona. Hay varios tipos de registros de recursos cada uno identificado por un carácter o un acrónimo corto. Entre estos registros de recursos se encuentran:

1. PTR (Pointer To Register). Hace corresponder una dirección IPv4 o IPv6 con el nombre del ordenador. Estas entradas son usadas para la resolución inversa (de dirección IP a nombre).
2. A: Contienen una dirección IPv4 en su campo de datos. Gracias a esto, los usuarios pueden introducir un nombre de dominio en el navegador y hacer que el cliente envíe automáticamente una solicitud HTTP² a la dirección IP correspondiente. Estas entradas son utilizadas para la resolución directa (de nombre a dirección IP). Si un sistema tienes varias direcciones habrá un registro diferente para cada una de ellas. Este registro tiene como sintaxis:

```
<host> IN A <dirección IP>
```

² HTTP (Protocolo de Transferencia de hipertexto) del inglés Transfer Protocol: Es el protocolo de comunicación que permite la transferencia de información en la World Wide Web.

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

3. AAAA: Funcionan igual que los registros A salvo que, en vez de usar dirección IPv4 usan dirección IPv6. La denominación AAAA surge de la longitud del campo de datos, que es cuatro veces más larga que la de un registro A. Este registro define la siguiente sintaxis:

```
<host> IN AAAA <dirección IP>
```

4. SOA: Registro que define el comienzo de la zona. Debe estar colocado directamente detrás de las directivas y define información importante acerca de la autoridad de los RR para la zona. Además, contiene información sobre la zona que se organiza a través del archivo de zona y, por ello, son especialmente importante para la transferencia de zonas. Este registro tiene como sintaxis:

```
@ IN SOA <servidor de nombres primario> <email del administrador> (  
  <número de serie>  
  <tiempo de refresco>  
  <tiempo de reintento>  
  <tiempo de expiración>  
  TTL mínimo  
)
```

5. CNAME (Canonical Name): Contiene un nombre alternativo (alias) para un dominio registrado, y remite a otro registro A o AAAA ya existente. Este registro tiene la siguiente sintaxis:

```
<nombre alias> IN CNAME <nombre real>
```

6. NS: Hace corresponder el nombre de dominio con el nombre de un servidor de nombres del dominio. A diferencia del registro SOA, el registro NS no necesariamente tiene que ser único, ya que debe existir uno para cada DNS secundario autoritativo de la zona, más el correspondiente al DNS primario, que sí es obligatorio en cada zona. Si existen subzonas 'hijas' se necesitan entradas para los servidores de nombres de las subzonas de forma que el servidor de más alto nivel puede

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

facilitar el acceso a las zonas de bajo nivel. También se necesitan registros de direcciones para poder acceder a esos servidores hijos. Este registro tiene la siguiente sintaxis:

IN NS <nombre del servidor>

7. MX: Identifican a los sistemas que se encargan del correo de la organización. El comodín (*) permite reenviar correos dirigidos a hosts determinados que no aparecen registrados en el directorio del DNS. Los números que aparecen después de MX se llaman números de autoridad, y se utilizan para proveer servidores de respaldo que entren en funcionamiento en caso de fallo del servidor de correo principal. Este registro tiene como sintaxis:

IN MX <valor preferencia> <nombre servidor email>

8. SRV: Mediante un registro SRV se puede indicar que servicios ofrecer bajo un dominio determinado.

La **jerarquía de nombres** está organizada en forma de árbol, de manera que todos los nombres de dispositivos están estructurados por distintos niveles. Cada nivel es administrado por una entidad que regula el uso de los nombres (1).

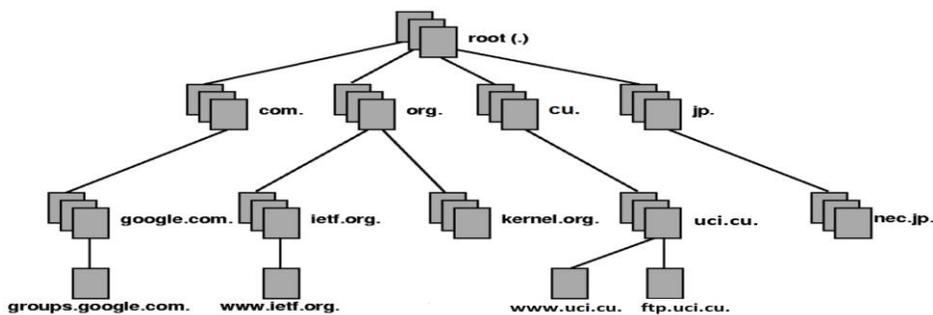


Fig 1. Arquitectura del Sistema de Nombres de Dominio.

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

Como se ilustra en la figura 1, se puede comprobar la estructura de árbol que organizaría los niveles jerárquicos en el sistema de nombres de dominio. Este estaría formado por:

- **Dominio raíz:** representado como un punto (.), es donde parten todos los nombres de dominio. La autoridad superior que se encarga de los nombres y las direcciones IP en internet es la IANA (Internet Assigned Numbers Authority). Esta es la responsable de coordinar y mantener todo el funcionamiento de los DNS a nivel mundial.
- **Dominios de Nivel Superior:** se conocen como TLD y son aquellos que corresponden a categorías generales como organizaciones, empresas o relacionados con códigos de países. Por ejemplo:
 1. COM: categoría destinada a entidades comerciales.
 2. ORG: categoría destinada a organizaciones gubernamentales.
 3. Código de país: categoría destinada a englobar todo lo relacionado con un país en concreto o una región. Por lo general, lo forman dos letras que indican el nombre del país (por ejemplo, ES- España, FR- Francia, CU- Cuba, MX- México, etc.).
 4. EDU: categoría destinada a entidades educativas, aunque estas también pueden utilizar el código de país.
- **Dominios de niveles inferiores:** Son todas aquellas categorías que se especifican dentro del dominio de primer nivel.

Por lo tanto, el DNS, organiza los nombres de dispositivos en una jerarquía de dominios separados por punto, se utiliza el carácter “.” como separador para delimitar el nombre de un dominio con sus subdominios inferiores. Por ejemplo, para acceder a un ordenador, la lectura del nombre completo se

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

realizaría componiendo los nombres de abajo hacia arriba, incluyendo todos los anteriores y separados por un punto. El nombre formado por toda la cadena se conoce como FQDN³.

1.1.1 Consultas DNS

Las consultas DNS se pueden enviar desde un cliente DNS a un servidor DNS o entre dos servidores DNS. Existen dos tipos de consultas:

- **Recursivas:** Cuando un cliente hace una consulta a un servidor DNS, el servidor comprueba los datos locales y ofrece la mejor respuesta. Si no la tiene, consulta al servidor raíz que menos tarde en ofrecer la respuesta, este a su vez, consultará al nivel intermedio, y así sucesivamente hasta llegar al autorizado. Una vez que el autorizado responde al nivel anterior, con acierto o con error, se van devolviendo los mensajes hasta llegar de nuevo al servidor local, y desde este al cliente.
- **Iterativas:** Las consultas iterativas es cuando el servidor DNS no tiene la información pedida, lanza una consulta al nivel superior hasta obtener una respuesta. Este le remite a un nivel intermedio, el servidor local le pregunta directamente al intermedio, este le contesta, y así hasta obtener la respuesta directamente del servidor autorizado (6).

1.1.2 Funcionamiento del DNS

Para la representación de cómo funciona el DNS, se utiliza como ejemplo la página web **www.debian.org**. El navegador busca la información de los servidores DNS del dominio **debian.org**. Internet esta ordenado de forma de árbol invertido, si no encuentra la información en el ordenador, irá a buscarla a los servidores que posee la configuración de red del ordenador, que son los que proporciona el ISP⁴, que son DNS

³ FQDN (Fully Qualified Domain Name, Nombre de Dominio Completo): Se refiere a la dirección completa y única necesaria para tener presencia en Internet.

⁴ ISP (*Internet Service Provider*, Proveedor de Servicios de Internet): Es la empresa que brinda conexión a Internet a los clientes.

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

primario y DNS secundario. De no estar, seguirá buscando a niveles superiores y, en último lugar, lo encontrará en el servidor de nombre raíz.

El Servidor de Nombres Raíz recibirá la petición del ordenador, preguntando a que dirección IP resuelve el dominio **debian.org**. El Servidor Raíz no sabe a qué dirección IP resuelve el dominio, pero si conoce los servidores DNS que resuelven el dominio **debian.org**, por lo que dirige la consulta a un servidor del dominio padre. Este servidor de nuevo dirigirá la consulta a los servidores autoritativos del dominio y por consiguiente el servidor autorizado dirigirá la consulta al ordenador con DNS1 y DNS2.

La información correspondiente al servidor DNS que gobierna el dominio **debian.org**, llega al ordenador: DNS1- 203.16.19.7 y DNS2- 128.81.2.10. Suelen ser dos porque las especificaciones de diseño de DNS recomiendan que como mínimo sean dos. El ordenador intentará conectar con el DNS1 y ante cualquier fallo de conexión intentará conectar con el DNS2. Estos son los servidores de nombres donde se encuentra la información donde se puede buscar la página web. Por consiguiente, el ordenador recibirá la dirección IP del servidor web donde se encuentra la página web. El ordenador se dirigirá luego al servidor web y buscará la página web en él. Por último, el servidor web devuelve la información pedida y el ordenador visualiza la página web en el navegador (6).

1.1.3 Tipos de servidores DNS

Dependiendo de la configuración y funcionamiento de los servidores estos pueden desempeñar distintos papeles:

- Servidor DNS primario: Es único y obligatorio para cada zona, ya que posee los ficheros sobre los que se introducen las modificaciones del dominio en el cual se podrá administrar, ya sea para añadir, modificar o eliminar los registros correspondientes. Será, el poseedor de la información original de la zona, ya que el resto de los servidores DNS, tendrán una copia de la misma.
- Servidor DNS secundario o esclavo: Son todos los servidores DNS que tengan una copia completa de la zona determinada. Su copia se actualizará, automáticamente, a partir de la información de un DNS maestro. Este proceso se conoce con el nombre de transferencia de zona.

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

No son obligatorios que existan, pero por medidas de seguridad son utilizados frente a un fallo del DNS primario, y para distribuir la carga del servicio DNS de la zona en cuestión. Un servidor puede ser secundario para una zona y primario para otra de forma simultánea.

- Servidor DNS maestro: Es un servidor DNS primario, secundario o esclavo, que permite que otros ordenadores accedan al mismo y obtengan los ficheros con la información del dominio. Este proceso es conocido como **transferencia del fichero de zona o dominio**. En todo dominio, el servidor primario debe ser un servidor maestro, aunque pueden existir servidores secundarios que actúen también como servidores maestros.
- Servidor DNS autoritativo: Todo aquel servidor DNS que tengan las condiciones para convertirse en DNS maestro de una zona y disponga de correspondiente registro NS definido en la zona en cuestión. Un servidor DNS, que solo disponga de la información de una zona, aunque sea completa, a través de la caché de su servicio DNS, no será un servidor autoritativo de la zona en cuestión.

A continuación, se expone un breve estudio de los principales servidores que implementan servicios DNS.

1.2 Análisis de servidores de nombres que implementan el servicio DNS.

Existen hoy en día un número grande de servidores para la implementación del servicio DNS, muchas de ellas son específicas para determinados DNS y otras que son utilizadas en varias plataformas.

1.2.1 Dnsmasq

Proporciona un sistema de infraestructura de red para redes pequeñas como DNS y DHCP. Es compatible en plataformas como Linux, Android, BSD y MAC OS X. Dnsmasq está incluido en la mayoría de las distribuciones de Linux y los sistemas de puertos FreeBSD, OpenBSD y NetBSD (9). Dnsmasq permite poner en marcha un servidor DNS de forma sencilla lo que permite utilizar un servidor DNS caché que resuelve los nombres de dominio que están configurados en el archivo `/etc/hosts` del servidor. Sin

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

embargo, permite también de forma limitada la definición de los registros DNS. En lo referente a este servicio sus desventajas más evidentes son:

- El tratamiento de varios dominios puede ser confuso y engorroso.
- Tiene limitado el soporte DNSSEC⁵: es capaz de validar registros externos o no hacerlo, y pasar a los clientes el registro de firma, pero no puede firmar automáticamente los registros (10).

1.2.2 Unbound

Es una resolución de DNS de validación que permite la recursividad y de almacenamiento en caché. Está diseñado para ser rápido e incorpora características modernas basadas en estándares abiertos. Admite DNS sobre TLS⁶, lo que permite a los clientes cifrar sus comunicaciones. Además, permite varios estándares modernos que limitan la cantidad de datos intercambiados con servidores autorizados. Estos estándares no solo mejoran la privacidad, sino que ayudan a hacer que el DNS sea más robusto. Lo más importante son la minimización del nombre de la consulta, el uso constante de la caché validada por DNSSEC y el soporte de zonas de autoridad, que se pueden usar para cargar una copia de la zona raíz. Unbound es un software gratuito y de código abierto bajo licencia BSD. Se puede ejecutar en plataformas BSD como FreeBSD, Open BSD y NetBSD, también en plataformas MacOS, Linux y Microsoft Windows (11).

1.2.4 PowerDNS

Esta herramienta ha sido diseñada para satisfacer las necesidades de pequeñas instalaciones al ser fácil de configurar, así como para atender grandes volúmenes de consultas en grandes cantidades de dominio. PowerDNS ofrece un rendimiento de resolución de dominio muy alto. Utiliza un servidor autorizado y uno recursivo, lo que permite combinarlos y ofrecerlos por separados. Como principal ventaja es que ofrece

⁵ DNSSEC (*Domain Name System Security Extensions*, Extensiones de Seguridad para el Sistema de Nombres de Dominio): es un conjunto de especificaciones para asegurar cierto tipo de información proporcionada por el DNS.

⁶ TLS (*Transport Layer Security*, Seguridad en capas de transporte): Es el protocolo criptográfico que garantiza las comunicaciones en Internet.

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

soporte DNSSEC ya que mediante el uso de lenguajes su código fuente es pequeño, lo que facilita la auditoría. Es un programa gratuito con una gran variedad de características de almacenamiento de datos (9). Posee soporte para registros NAPTR⁷ de manera nativa. Como pequeña desventaja se puede señalar que no trae soporte para el protocolo TSIG (*Transaction Signature*), muy empleado para proveer un método de autenticación a la hora de actualizar el contenido de una base de datos de DNS dinámica (12). PowerDNS brinda varios servicios en los que podemos encontrar:

- Soporte inmediato para tecnologías avanzadas como la función control parental.
- Soporta filtrado de malware.
- Mitigación de ataques automatizados.
- Comunicación de suscriptores.
- Búsqueda de consultas a largo plazo.

1.2.3 CoreDNS

CoreDNS es un servidor DNS que está escrito en Go que es un lenguaje de programación de código abierto. El objetivo principal de CoreDNS es apoyar el descubrimiento de servicios en entornos de contenedores, particularmente los administrados por kubernetes. Es diferente de otros servidores DNS, como Bind, PowerDNS y Unbound, porque es muy flexible y casi toda la funcionalidad se externaliza en complementos (13). Actualmente hay 30 complementos incluidos en la instalación predeterminada de CoreDNS, pero también hay una inmensa cantidad de complementos externos que puede compilar con CoreDNS para ampliar su funcionalidad CoreDNS utiliza complementos para proporcionar la funcionalidad de DNS por lo que existe un complemento para el almacenamiento en caché y un complemento para reenviar, un complemento para configurar un servidor DNS primario que lee los datos de zona de un archivo y un complemento para configurar un servidor DNS secundario. CoreDNS es rápido y seguro

⁷ NAPTR (Name Authority Pointer): Es un tipo de registro de recursos del Sistema de Nombres de Dominio de Internet.

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

debido a que no solo es sencillo de configurar cada complemento, sino que además si no necesita un complemento, no lo configura y su código no lo ejecuta (14).

1.2.5 Pihole

Pihole es un sumidero de DNS de código abierto que protege sus dispositivos del contenido no deseado, sin instalar ningún software del lado del cliente. Es receptivo ya que acelera sin problemas la sensación de la navegación diaria al almacenar en caché las consultas DNS. Es ligero porque funciona sin problemas con requisitos mínimos de hardware y software. Es robusto ya que posee una interfaz gráfica de líneas de comandos de calidad asegurada para la interoperabilidad. Es versátil porque puede funcionar opcionalmente como un servidor DHCP, asegurando que todos sus dispositivos estén protegidos. También es escalable ya que es capaz de manejar cientos de millones de consultas cuando se instala en hardware de nivel superior y es muy moderno porque bloquea los anuncios sobre IPv4 e IPv6 (15).

1.2.6 BIND9

Bind9 ha evolucionado para ser un sistema DNS muy flexible y con todas las funciones como utilizar nombres de dominio para referenciar las direcciones IP. Es un servicio de código abierto bajo licencia MPL 2.0. Se sometió a tres revisiones principales, cada una con arquitecturas significativamente diferentes: BIND4, BIND8 y BIND9. BIND4 y BIND8 ahora son técnicamente obsoletos, ya que BIND9 se reescribió desde cero basándose en las arquitecturas de BIND4 y BIND8. BIND9 es una reescritura básica de BIND con soporte completo de DNSSEC además de otras características y mejoras (16). En abril del año 2010 ISC (*Internet System Consortium*) comenzó el desarrollo de una nueva versión, BIND10, pero la participación de ISC concluyó con el lanzamiento de BIND10 versión 1.2 en abril de 2014 debido a la falta de recursos para continuar con el desarrollo de BIND10, y reafirmaron su compromiso con BIND9, por lo que BIND10 no será objetivo para considerar en este estudio (17).

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

1.2.7 Justificación del servidor de nombres seleccionado

Entre las ventajas que poseen estos servicios se encuentra la seguridad que estos brindan, la cantidad de recursos que se dedican o queda disponible para que el sistema utilice estos, ya que los servicios se instalan sin un entorno de escritorio, lo que posibilita un mejor aprovechamiento y disponibilidad de los recursos. Entre las desventajas: no cuentan con una interfaz gráfica para la configuración de los servicios, fallas de seguridad debido a la mala configuración de los ficheros que posee un determinado servicio, desconocimiento de las cláusulas de configuración y gran desaprovechamiento de las funcionalidades de este servicio.

Algunas características de DNS son relevantes solo para servidores recursivos o para servidores autorizados. Dado un estudio de los servidores de nombres que implementan el servicio DNS, se muestra un resultado en una matriz de características que representa la efectividad y madurez de una implementación determinada. Las características que se van analizar son las siguientes:

- **Servidor Autorizado:** Publican asignaciones DNS para su dominio bajo su control autorizado.
- **Servidor Recursivo:** Proporcionan resolución de nombres DNS, al transmitir las solicitudes de la aplicación cliente a la cadena de servidores de nombres autorizados para resolver completamente un nombre de red.
- **Control de Acceso Recursivo (ACL):** Los servidores proporcionan control sobre qué hosts tiene permitidas búsquedas recursivas de DNS. Esto es útil para el equilibrio de carga y la protección del servicio.
- **DNSSEC:** Los servidores con esta característica implementan alguna variante de los protocolos DNSSEC.
- **Almacenamiento en caché:** Los servidores proporcionan servicios recursivos para aplicaciones y almacenan en caché los resultados para que las futuras solicitudes del mismo nombre puedan ser respondidas rápidamente, sin una búsqueda completa de DNS.

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

- **IPv6:** Los servidores con esta función pueden manejar registros DNS que se refieren al protocolo de transporte IPv6.

Tabla 1. Comparación de servidores de nombres que implementan el servicio DNS.

Herramienta	Servidor Autorizado	Servidor Recursivo	Control de Acceso Recursivo	DNSSEC	Almacenamiento Caché	IPv6
Dnsmasq	Parcialmente	No	No	Limitado	Si	Si
Unbound	Parcialmente	Si	Si	Completo	Si	Si
PowerDNS	Si	Si	Si	Completo	Si	Si
CoreDNS	Si	No	No	Limitado	Si	Si
Pihole	Parcialmente	No	No	No	Si	Si
Bind9	Si	Si	Si	Completo	Si	Si

Durante el estudio realizado se ha llegado a la conclusión que los servidores de nombres CoreDNS, Unbound, PowerDNS y Bind9 son los servidores más potentes para el desarrollo de la aplicación, ya que poseen facilidad en su uso, seguridad, y permiten utilizar servicios como DNSSEC. Todos estos servidores tienen como desventajas que no poseen una interfaz gráfica. Unbound no se puede utilizar como servidor esclavo y el servidor autoritativo lo implementa parcialmente por lo que no se incluirá en esta comparación ya que esta son unas de las limitantes importantes para la elección del servidor de nombres. PowerDNS es un servidor de nombres que posee grandes utilidades para la administración, pero tiene una gran debilidad, pues no posee un servidor de horizonte dividido que no es más que una implementación del DNS para proporcionar diferentes conjuntos de información (12). Esta funcionalidad si

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

la poseen Bind9 y CoreDNS. A continuación, en la tabla 2 se muestra un resumen entre las diferencias claves de las funcionalidades de Bind9 y CoreDNS.

Tabla 2: Comparación de las características de los servidores

	CoreDNS	BIND9
Recursividad Completa	No	Sí
Actualizaciones dinámicas	No	Sí
Compatibilidad con extensiones de seguridad del DNS (DNSSEC)	Limitado	Completo
Soporte para DNS sobre Seguridad de la capa de transporte (DoT)	Sí	No

Por todas estas razones se utiliza el servidor de nombres Bind9 para la administración del servicio DNS debido a que a diferencia de CoreDNS, este posee un servidor recursivo, actualizaciones dinámicas y es completo en cuanto a la seguridad del DNS mediante DNSSEC, por otra parte, CoreDNS actualmente tiene algunas limitaciones importantes y no será adecuado para todos los servidores DNS, el principal de estos problemas es que la última versión no admite recursividad completa, en otras palabras, no puede procesar una consulta comenzando en la raíz de un espacio de nombres DNS, que consulta un servidor DNS raíz y sigue las referencias hasta que obtenga una respuesta en uno de los servidores autorizados (13). Es por este motivo que se reafirma la utilización del servidor de nombres Bind9 como servidor para la administración del servicio DNS, ya que la misma resuelve las desventajas de los servicios analizados.

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

1.3 Análisis de herramientas de administración del servicio de resolución de nombres en GNU/Linux.

En el presente epígrafe el autor expone un diagnóstico de las herramientas para la administración del servicio de resolución de nombre encontradas.

1.3.1 Webmin

Webmin es una herramienta para la configuración del sistema mediante la web accesible para OpenSolaris, GNU/Linux y otros sistemas Unix, además que es accesible desde la consola. Se puede utilizar también en Windows y MacOS. Permite configurar las características propias del sistema, como servicios, usuarios, cuotas de espacio y archivos. Tiene la posibilidad de modificar y controlar muchas aplicaciones de código abierto, como el servidor Web, PHP, MySQL, DNS, SSH, LDAP, DHCP y FTP. Sin embargo, esta herramienta carece de la posibilidad de administración remota, y por tanto de la posibilidad de administrar para de un servidor a la vez (18).

1.3.2 Centos Web Panel

Es un panel de alojamiento web gratuito diseñado para administración rápida de servidores sin el esfuerzo de tener que utilizar la consola ssh cada vez que se desee hacer algo. Ofrece una gran cantidad de opciones y características para la gestión del servidor en su panel de control. Permite administrar servidores web, DNS, correo y MySQL, Nginx, entre otros (19).

1.3.3 Virtualmin

Virtualmin es un módulo de Webmin para administrar varios hosts a través de una sola interfaz. Admite la creación y gestión de servidores virtuales como Apache, dominios DNS BIND, base de datos SQL. Virtualmin ha conseguido posicionarse como una alternativa libre y fiable a paneles de control de pago

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

como Cpanel⁸ o Plesk⁹ para la gestión de múltiples servidores web de apache. Virtualmin hace uso de los módulos existentes de Webmin para estos servidores, por lo que debería funcionar con cualquier configuración de sistema existente en lugar de necesitar su propio servidor de correo o servidor web (20).

A modo de conclusión ninguna de las herramientas estudiadas permite la administración remota del servicio de nombres de dominio DNS, sin embargo, el estudio de estas herramientas contribuyó a la utilización de algunas de sus funcionalidades, siendo necesario la implementación de una herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

1.4 Metodologías de desarrollo de software

Una metodología de desarrollo de software es el entorno que se utiliza para estructurar, planificar y controlar el proceso de desarrollo de software. Consiste principalmente en hacer uso de diversas herramientas, técnicas, métodos y modelos para el desarrollo.

1.4.1 Variación AUP para la UCI

AUP es una versión simplificada del Proceso Unificado Racional (RUP en sus siglas en inglés). Describe de manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio, al utilizar técnicas ágiles y conceptos que aún se mantienen en RUP. Como las metodologías de desarrollo se adaptan a las características de cada proyecto, se decide realizar una variación de la metodología AUP para la actividad productiva de la UCI. Esta metodología sufrió cambios con el fin de poder ser utilizadas en todos los proyectos de la UCI, en este momento cuenta con tres fases: inicio, ejecución y cierre (21).

Primera fase: Inicio, donde se llevan a cabo las actividades relacionadas con la planeación del proyecto.

Segunda fase: Ejecución, en esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto considerando los requisitos y la arquitectura.

⁸ Cpanel: Es un panel de control para administrar servidores de alojamiento web.

⁹ Plesk: Es una plataforma comercial de alojamiento web que permite configurar sitios web, cuentas de correo electrónico y entradas DNS a través de una interfaz basada en la web

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

Tercera fase: Cierre, se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto.

De la metodología seleccionada se escogió el escenario número cuatro, debido a que este es recomendable para proyectos que no modelen el negocio y solo puedan modelar el sistema con Historias de Usuarios.

Escenario No.4: Aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convertir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos, ya que una HU no debe poseer demasiada información. Todas las disciplinas antes definidas (desde Modelado de Negocio hasta Pruebas de Aceptación) se desarrollan en la Fase de Ejecución, de ahí que en la misma se realicen iteraciones y se obtengan resultados incrementales. En una iteración se repite el flujo de trabajo de las disciplinas, requisitos, análisis y diseño, implementación y pruebas internas. De esta forma se brinda un resultado más completo para un producto final de manera creciente. Para llegar a esto, cada requisito debe tener un completo desarrollo en una única iteración.

La presente investigación utilizará las fases Inicio, Ejecución y Cierre, y se decide utilizar el Escenario No.4 ya que es recomendable para proyectos pequeños y de corta duración en el que el equipo de desarrollo esté siempre en constante interacción con el cliente, para convenir los detalles de los requerimientos y así poder implementarlos, probarlos y validarlos.

1.5 Lenguajes y herramientas para el desarrollo de la herramienta.

1.5.1 Vue v4.1.2

Vue es un framework accesible, versátil y de buen rendimiento que te permite crear una base de código mucho más fácil de mantener y probar. Es un framework de JavaScript progresivo, lo que significa, que si utilizamos framework de aplicaciones web se puede incluir Vue a solo una parte de la aplicación que

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

necesite una experiencia más intuitiva e interactiva. Vue permite tomar una página web y dividirla en componentes reusables, cada uno con su propio HTML, CSS y JavaScript necesario para generar una parte de la página (22). A continuación, mostramos una comparación de Vue con otros framework mediante tablas de comparación en una escala de 1-5 donde se demuestra en cuanto supera Vue en algunas características a otros framework.

Comparación de Vue con otros framework

Framework	Rendimiento	Esfuerzos de Optimización	Renderizado con JSX	HTML y CSS	CSS con alcance a componentes	Enrutamiento	Renderización nativa
Vue	5	5	3	5	5	5	3
React	4	3	5	2	3	3	5

Fig 2: Comparación de Vue con React

Framework	Complejidad	Flexibilidad y Modularidad	Vinculación de datos	Directivas vs Componentes	Rendimiento
Vue	5	5	5	5	5
AngularJS	3	4	3	4	3

Fig 3: Comparación de Vue con AngularJS

Framework	TypeScript	Tamaño y Rendimiento	Flexibilidad	Curva de Aprendizaje
Vue	2	5	5	5
Angular	5	5	2	2

Fig 4: Comparación de Vue con Angular

Como se muestra en las figuras Vue resuelve mejor los problemas que los framework React, AngularJS y Angular, ya que posee un mejor rendimiento y es excepcionalmente rápido. Posee un menor tamaño lo que significa que es mucho más ligero.

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

1.5.2 Vuetify 1.14

Vuetify es la biblioteca de componente No.1 para Vue.js y ha estado en desarrollo activo desde 2016. Tiene como objetivo proporcionar a los usuarios todo lo que se necesita para crear aplicaciones web atractivas utilizando la especificación de Material Design. Posee un ciclo de actualizaciones constantes, soporte a largo plazo para versiones anteriores, participación de la comunidad receptiva, un vasto ecosistema de recursos y una dedicación a componentes de calidad (23).

1.5.3 JavaScript v1.8.5

JavaScript es un lenguaje de programación interpretado usado para múltiples propósitos, pero solo considerado como un complemento hasta ahora. Una de las innovaciones que ayudó a cambiar el modo en que vemos JavaScript fue el desarrollo de nuevos motores de interpretación, creados para acelerar el procesamiento de código. La clave de los motores más exitosos fue transformar el código de JavaScript en código máquina para lograr velocidades de ejecución similares a aquellas encontradas en aplicaciones de escritorio. Esta mejora permitió superar viejas limitaciones de rendimiento y conformar el lenguaje JavaScript como la mejor opción de la web.

Para aprovechar esta prometedora plataforma de trabajo ofrecida por los nuevos navegadores, JavaScript fue expandido en relación con portabilidad e integración. A la vez, interfaces de programación de aplicaciones (APIs) fueron incorporadas en cada navegador para asistir al lenguaje en funciones elementales (24).

1.5.4 Electron v7.1.2

Electron es un framework para JavaScript que permite el desarrollo de aplicaciones enriquecidas de escritorio mediante el uso de tecnologías web. Esta desarrollado por GitHub que es un sistema de gestión de proyectos y control de versiones de código, por lo que garantiza las versiones constantes de Electron. Electron funciona bajo un subconjunto de librerías de chromium. Además, proporciona acceso a APIs nativas enriquecidas mediante el motor de Chromium, el cual es controlado mediante JavaScript (25).

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

1.5.5 Visual Studio Code v1.38.1

Es un editor de código fuente, gratuito y de código abierto, es ligero y potente a la vez, que se ejecuta en el escritorio. Está disponible en plataformas Windows, Mac OS y Linux. Visual Studio Code se basa en Electron y viene con soporte incorporado para JavaScript, TypeScript y Node.js y posee una gran variedad de extensiones para otros lenguajes como C++, C#, Java, Python, PHP y Go, también posee extensiones de tiempos de ejecución como .Net y Unity (26).

1.5.6 UML v2.5

UML (Unified Modeling Language) es un lenguaje de modelado para sistemas de software orientados a objetos, permite representar en mayor o menor medida todas las fases de un proyecto informático: desde el análisis con los casos de uso, el diseño con los diagramas de clases, objetos, etc, hasta la implementación y configuración con los diagramas de despliegue. Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementarlo (27).

1.5.7 Visual Paradigm v13.0

Visual Paradigm es una herramienta CASE que soporta el ciclo de vida completo del proceso de desarrollo de un proyecto a través de la representación de todo tipo de diagramas, el código a partir de estos diagramas, así como la documentación. Permite el modelado visual UML y propicia la construcción de aplicaciones de calidad.

Esta herramienta ofrece: Soporte de aplicaciones web, disponibilidad en múltiples plataformas como Windows y Linux, una versión gratuita y otra comercial, permite la generación de código (modelo a código, diagrama a código) y permite exportar el diseño de la base de datos para varios gestores (28).

Conclusiones parciales

El estudio de los conceptos y las características de los servicios de nombres de dominio permitió comprender como funcionan y que estructura poseen estas herramientas en los servicios telemáticos. Luego de caracterizar los servicios que se encargan de la implementación del servicio DNS, se selecciona

Fundamentación teórica sobre la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova

el servicio Bind9 como servidor de nombres de dominio y el uso de sus utilidades para llevar a cabo el desarrollo de la herramienta propuesta. No se encontraron herramientas que permitan la administración remota del servicio DNS en GNU/Linux. Se define la metodología, tecnologías y herramientas a utilizar durante el diseño e implementación a partir de los elementos empleados para garantizar la integración de la herramienta.

Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS

Capítulo 2: Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

En el presente capítulo se realiza una descripción de los procesos del negocio pertenecientes a la administración del servicio de resolución de nombres de dominio DNS en sistemas operativos GNU/Linux, así como la definición de los requisitos funcionales y no funcionales. A continuación, se especifican las características y funcionalidades de la herramienta a desarrollar a través de las historias de usuario.

2.1 Descripción general de la propuesta de solución de la investigación

Se propone el desarrollo de una herramienta que realice la gestión del servicio de resolución de nombres de dominio de forma remota. El desarrollo de esta herramienta se propone para facilitar el manejo de las zonas y registros pertenecientes a un dominio mediante el uso del servidor de nombres bind9. La aplicación permite, gestionar servidores remotos y proporciona las funcionalidades de gestión del sistema mediante una interfaz gráfica. Las funcionalidades permiten gestionar zonas y registros de un dominio determinado y administrar el servicio bind9.

2.2 Modelo Conceptual

Un modelo conceptual explica los conceptos significativos en un dominio del problema; es el artefacto más importante a crear durante el análisis orientado a objetos (29).

Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio

DNS

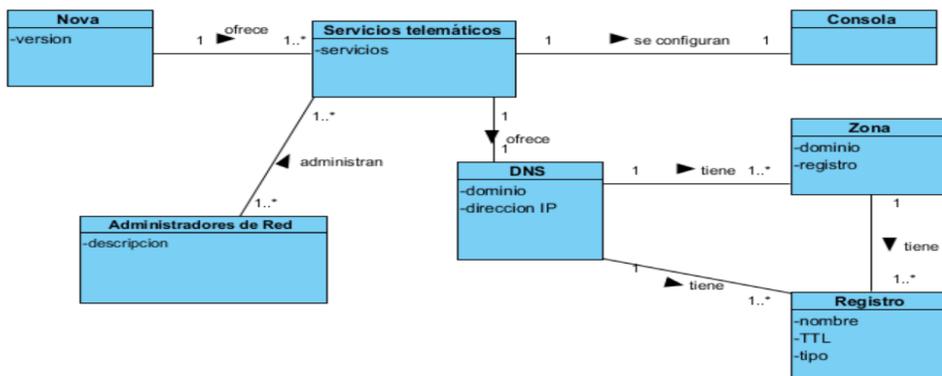


Fig 2: Modelo Conceptual

Nova: Distribución de GNU/Linux.

Servicios telemáticos: es el campo que engloba los conocimientos sobre los servicios de red y la comunicación para el desarrollo de aplicaciones que permiten transmitir datos.

Administradores de red: son los usuarios que se encargan de utilizar los servicios telemáticos en los centros de desarrollo.

Consola: es el dispositivo electrónico que permite al usuario realizar funcionalidades para acceder al sistema sin utilizar interfaz gráfica.

DNS: es el servicio que se encarga de traducir nombres de dispositivos a direcciones IP y viceversa.

Zona: es la forma en que los servidores de nombres distribuyen los datos de un dominio.

Registro: definen las características o las entidades de una zona.

Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio

DNS

2.3 Descripción de requisitos del sistema

Los requisitos de software son la base para el desarrollo de una solución al problema, detalla las funciones y restricciones del sistema o componente para satisfacer un contrato, estándar u otro documento impuesto formalmente. 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. Los requisitos se pueden clasificar en funcionales y no funcionales (30).

2.3.1 Requisitos funcionales

Los requisitos funcionales son aquellos directamente relacionados con las funciones o las reacciones que el sistema debe proporcionar. Generalmente, se asignan directamente a elementos o características de un sistema o software. A continuación, se muestra un listado donde se presentan 11 requisitos funcionales definidos para la implementación de la herramienta. Estos son clasificados a partir de su prioridad en alta, media y baja. Como el cliente no define una prioridad de ejecución específica para los requisitos, el analista se encarga de establecer un orden de ejecución para estos, en función de la necesidad y la dependencia entre ellos.

Tabla 3 : Especificación de requisitos funcionales.

No.	Nombre del requisito	Descripción	Prioridad	Complejidad
RF.1	Adicionar Zona	El sistema debe permitirle al administrador adicionar las zonas locales de resolución de nombres.	Alta	Alta
RF. 2	Editar Zona	El sistema debe permitirle al administrador modificar las zonas locales de resolución de nombres mediante el fichero named.conf.local.	Alta	Alta

Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio

DNS

RF.3	Eliminar Zona	El sistema debe permitirle al administrador eliminar la zona local.	Alta	Alta
RF.4	Mostrar Zona	El sistema debe permitir mostrar las zonas locales de resolución de nombres.	Media	Media
RF.5	Adicionar Registro	El sistema debe permitirle al administrador adicionar un registro para la configuración de zona.	Alta	Alta
RF.6	Eliminar Registro	El sistema debe permitirle al administrador eliminar un registro de configuración de zona.	Alta	Alta
RF.7	Editar Registro	El sistema debe permitirle al administrador modificar un registro de configuración de zona.	Alta	Alta
RF.8	Mostar Registro	El sistema debe permitir mostrar los registros de las zonas configuradas.	Media	Media
RF.9	Iniciar Servicio Bind9	El sistema debe permitir iniciar el servicio Bind9.	Media	Baja
RF.10	Reiniciar Servicio Bind9	El sistema debe permitir reiniciar el servicio Bind9.	Media	Baja
RF.11	Detener Servicio Bind9	El sistema debe permitir detener el servicio Bind9.	Media	Baja

Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio

DNS

2.3.2 Requisitos no funcionales

Los requisitos no funcionales representan características generales y restricciones que afectan a los servicios o funciones del sistema, tales como restricciones de tiempo, que afectan el proceso de desarrollo. Como requisitos no funcionales del sistema propuesto se encuentran:

Software

RNF1: la herramienta para administrar de forma remota el servicio DNS debe ser compatible con la plataforma GNU/Linux Nova 7.0.

Confiabilidad

RNF2: la herramienta debe ser capaz de recuperarse en caso de fallo de ejecución que puedan haber ocurrido en el software. Durante el fallo de alguna funcionalidad de la herramienta, el resto de esta debe continuar su ejecución.

Mantenibilidad

RNF3: la herramienta para administrar remotamente el servicio DNS en Nova 7.0 debe estar codificada en el estándar definido por el proyecto de desarrollo, con el fin de facilitar el análisis para futuras modificaciones, o simplemente para el mantenimiento del sistema.

Seguridad

RNF4: la herramienta permite contar con copias de seguridad y restauración de todos los archivos de la herramienta ya sean carpetas, permisos y configuraciones.

RNF5: la integridad de los datos no debe verse afectada por la restauración de la copia de seguridad.

Eficiencia

RNF6: el sistema debe permitir que los usuarios interactúen con él de manera simultánea.

Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio

DNS

RNF7: el tiempo de demora del sistema en cada transición debe ser menor de (5) segundos aproximadamente.

2.4 Historias de usuario

Las historias de usuarios se utilizan para captar las necesidades de los clientes en un proyecto de software, estas son escritas en primera persona y en lenguaje simple con el fin de mejorar el entendimiento entre el equipo de desarrollo y el cliente. Así en lugar de documentos técnicos y formales, las historias de usuarios hacen hincapié en la forma verbal de la comunicación. Las historias de usuarios que se describen a continuación son algunas que representan a varios requisitos de mayor prioridad en el desarrollo de la herramienta:

- Adicionar zona
- Eliminar Registro
- Iniciar servicio Bind9

2.4.1 Historia de Usuario *Adicionar Zona*

Tabla 4: HU-Adicionar Zona

Número: HU – 1	Nombre del requisito: Adicionar Zona
Programador: Julio González Rodríguez	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 3
Riesgo en Desarrollo: Medio	Tiempo Real: 1
Descripción: La funcionalidad comienza cuando el usuario selecciona la opción Administrar Zona , luego se despliega un menú con la opción Adicionar Zona . Esta despliega en una interfaz donde se muestra un listado con las zonas registradas, donde	

Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio

DNS

aparece un botón para adicionar una zona DNS. Una vez pulsado el botón, esta muestra una interfaz donde aparecen los campos donde se añaden los atributos con los que se adiciona una zona. En la especificación de los atributos para adicionar una zona se encuentran los siguientes parámetros:

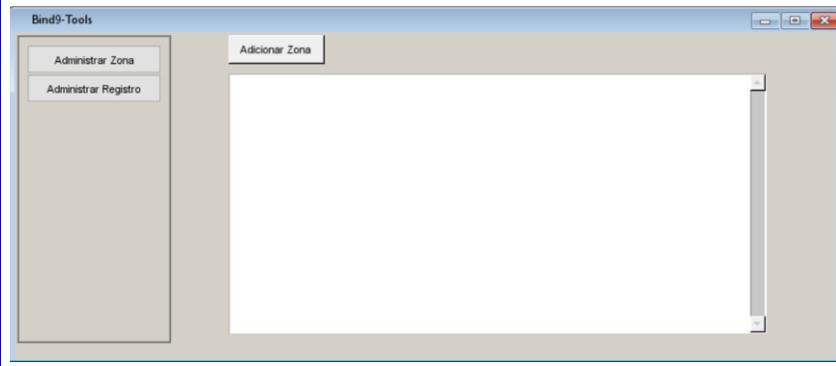
En la pestaña *Principal*:

Dominio (obligatorio): El nombre del dominio puede contener hasta 32 caracteres, no puede contener espacios en blanco o alguno de los caracteres: “,+*;;=\?'\<>”.

Adicionar Registro (obligatorio): Adiciona un registro a la zona.

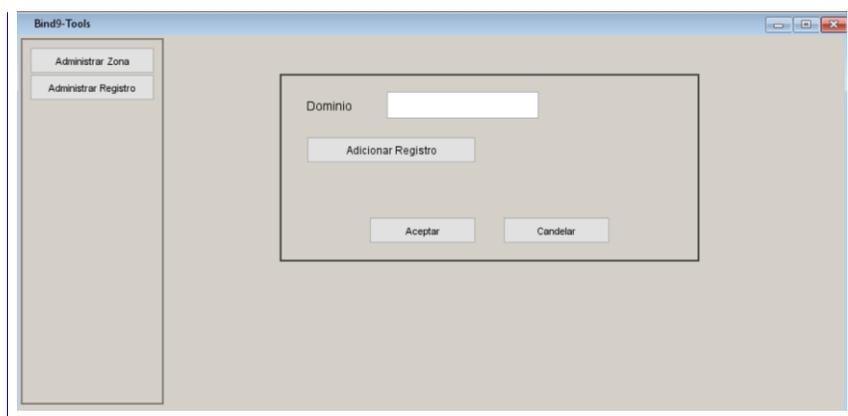
Observaciones: Una vez que se haya seleccionado la opción **Adicionar Zona** en **Administrar Zona**, se escribe el nombre del dominio y se adicionan los registros deseados. A continuación, haz clic en **Aceptar** o **Cancelar**.

Prototipo de interfaz gráfica de usuario:



Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio

DNS



2.4.2 Historia de Usuario *Eliminar Registro*

Tabla 5: HU- Eliminar Registro

Número: HU – 2	Nombre del requisito: Eliminar Registro
Programador: Julio González Rodríguez	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 3
Riesgo en Desarrollo: Medio	Tiempo Real: 1
Descripción: La funcionalidad comienza cuando el usuario selecciona la opción Administrar Registro . Esta despliega una interfaz donde se muestra un listado con todas las zonas con los registros adicionados. En la parte derecha del listado aparece un botón de Editar Registro . Luego se muestra un listado con los registros correspondientes a la zona seleccionada y se seleccionan los registros a eliminar	

Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio

DNS

mediante el botón **Eliminar registro**. En la especificación de los atributos para eliminar un registro se encuentran los siguientes parámetros.

En la pestaña *Principal*:

Dominio(obligatorio): Se selecciona el dominio de la zona a eliminar.

Listado de Registros: Se muestra un listado con los registros a eliminar

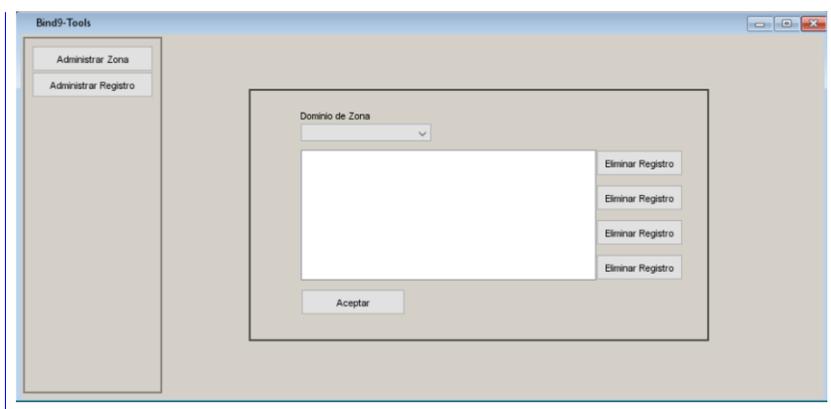
Observaciones: Una vez que se haya seleccionado la opción **Editar Registro** en **Administrar Registro**, se seleccionan los registros a eliminar mediante el botón **Eliminar Registro** en la zona especificada. A continuación, haz clic en **Aceptar**.

Prototipo de interfaz gráfica de usuario:



Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio

DNS



2.4.3 Historia de Usuario *Iniciar Servicio Bind9*

Tabla 6: HU- Iniciar Servicio Bind9

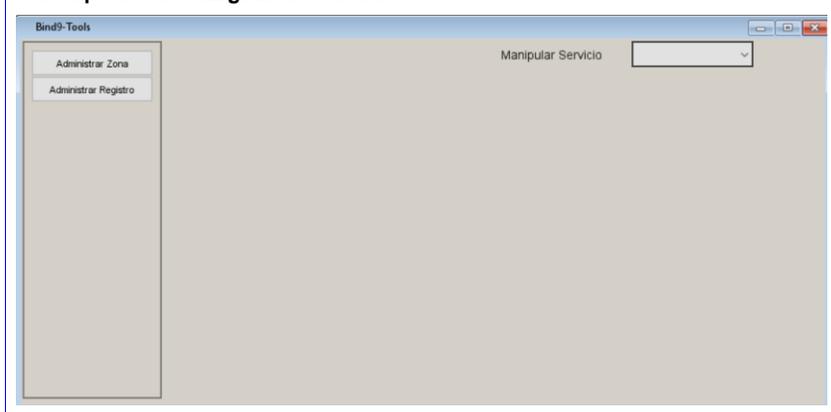
Número: HU – 3	Nombre del requisito: Eliminar Registro
Programador: Julio González Rodríguez	Iteración Asignada: 1
Prioridad: Media	Tiempo Estimado: 3
Riesgo en Desarrollo: Medio	Tiempo Real: 1
Descripción: La funcionalidad comienza cuando el usuario selecciona la opción Manipular Servicio que se encuentra a la derecha de la aplicación. Esta despliega un menú donde se muestra la opción Iniciar Servicio Bind9 .	
Observaciones: Una vez que se haya seleccionado la opción Manipular Servicio , se	

Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio

DNS

selecciona la opción **Iniciar Servicio Bind9**.

Prototipo de interfaz gráfica de usuario:



2.5 Diseño de la herramienta para la administración remota de servicios DNS en GNU/Linux/Nova

Según R. Pressman en el libro *Ingeniería de Software* plantea que el diseño de software agrupa el conjunto de principios, conceptos y prácticas que llevan al desarrollo de un sistema o producto de alta calidad. El objetivo del diseño es producir un modelo o representación que tenga resistencia y funcionalidad (31).

2.5.1 Diseño Arquitectónico

Se utilizará la arquitectura n-capas, específicamente el patrón 3-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 difícil puesto que se puedan crear nuevos componentes sobre sobre la

Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio

DNS

base para especializar las funcionalidades de un componente específico (31). Se utiliza el framework Vue en el cual se gestionan las conexiones y el tratamiento de las excepciones.

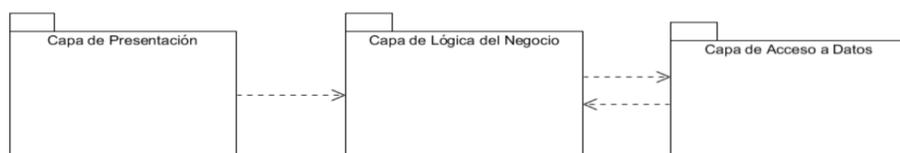


Fig 5 : Diseño Arquitectónico

En la capa Presentación se encuentran todas las interfaces (.vue) de la herramienta que son la CI_Principal, CI_AdministrarZona y la CI_AdministrarRegistro. En la capa de lógica del Negocio se encuentran las clases controladoras que son la clase ControllerZona y ControllerRegistro. La capa de acceso a datos se encuentran las clases entidades ZonaEntity y RegistroEntity, además del fichero de configuración del servidor de nombres Bind9. Mediante la conexión ssh se establece la comunicación remota que permite al administrador controlar y modificar servidores remotos, ya que la herramienta proporciona un mecanismo de autenticar un usuario remoto.

Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS

DNS

2.5.2 Diagrama de Clases del Diseño

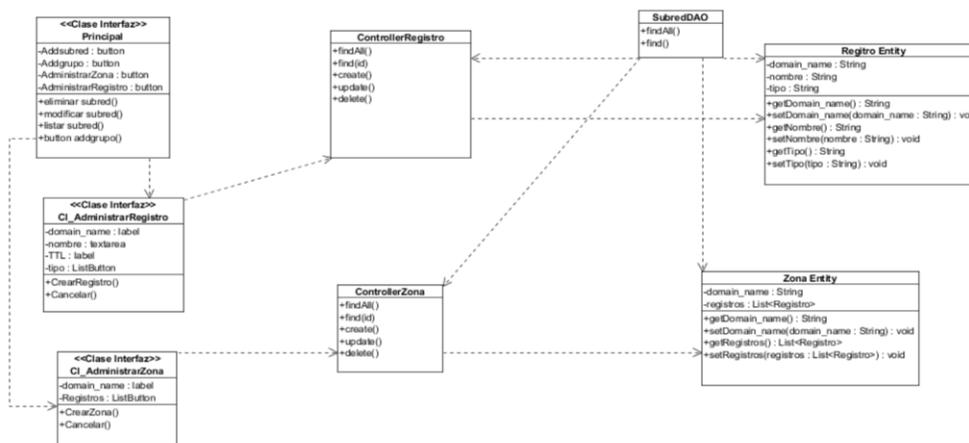


Fig 6 : Diagrama de Clases del Diseño

CI_Principal: representa los componentes relacionados con la administración del servicio y la conexión del servidor remoto mediante ssh.

CI_AdministrarRegistro: muestra los campos necesarios para gestionar un registro de una zona DNS.

CI_AdministrarZona: muestra los campos necesarios para registrar, modificar y eliminar una zona DNS.

ControllerRegistro: es la clase donde se ejecutan las operaciones de un registro de recursos.

ControllerZona: es la clase donde se ejecutan las operaciones de una zona DNS.

SubredDAO:

RegistroEntity: es la entidad que representa los datos de un registro de recursos.

ZonaEntity: representa los datos vinculados a una zona DNS.

Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio

DNS

2.6 Patrones de Diseño

Los patrones de diseño son una solución a un problema de diseño. Un patrón de diseño nomina, abstrae e identifica los aspectos clave de una estructura de diseño común, lo que los hace útiles para crear un diseño orientado a objetos reutilizable. Cada patrón de diseño se centra en un problema concreto, donde describe cuándo aplicarlo y si tiene sentido hacerlo teniéndose en cuenta otras restricciones de diseño, así como las consecuencias y las ventajas e inconveniencias de su uso (32). En el modelo del diseño de la herramienta se aplican patrones GRASP¹⁰ y patrones GoF¹¹ lo que favorecen la estructuración y entendimiento del diseño.

2.6.1 Patrones GRASP

Experto

El patrón Experto se usa más que cualquier otro para asignar responsabilidades referentes a la obtención de información. Es un principio básico que suele utilizarse en el diseño orientado a objetos. El patrón Experto guía hacia diseños donde el objeto de software realiza las operaciones que normalmente se aplican en el mundo real. Se conserva el encapsulamiento, ya que los objetos se valen de su propia información para hacer lo que se les pide, así como un bajo acoplamiento entre los objetos, lo que estimula las definiciones de clases más cohesivas. Este patrón es utilizado en las clases ZonaEntity y RegistroEntity

Creador

El patrón creador guía la asignación de responsabilidades relacionadas con la instanciación y creación de objetos. Una de las actividades más comunes en un sistema orientado a objetos es la creación de instancias, por lo que si se asignan correctamente las responsabilidades del diseño puede soportar un

¹⁰ GRASP (*General Responsibility Assignment Software Patterns*, Patrones Generales de Software para Asignar Responsabilidades).

¹¹ GoF (*Gang of Four*, Banda de los Cuatro): Nombre con el que se conoce a los autores del libro Design Patterns.

Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio

DNS

bajo acoplamiento y mayor cohesión. Este patrón es empleado en las clases ControllerZona y ControllerRegistro.

Bajo Acoplamiento

El patrón de bajo acoplamiento es un principio a tener en mente en todas las decisiones de diseño. Es un patrón evaluativo que el diseñador aplica al juzgar sus decisiones de diseño. A partir del framework Vue se obtienen inyecciones de dependencias, por lo que es necesario el uso de este patrón en el sistema. Con el uso de este patrón los componentes no se afectan por cambios de otros componentes, son fáciles de entender por separados y fáciles de reutilizar. Este patrón es utilizado en todas las clases.

Alta Cohesión

El patrón de alta cohesión una clase tiene responsabilidades moderadas en un área funcional y colabora con otra para realizar determinadas actividades. Mejora la calidad y la facilidad con que se entiende el diseño. Garantiza que las clases sean posibles de entender, reutilizar y de fácil mantenimiento. Este patrón se utiliza en todas las clases.

2.6.2 Patrones GoF

Los patrones GoF son patrones de asignación de responsabilidades y se pueden categorizar en tres grupos a según su propósito: patrones creacionales, patrones estructurales y patrones de comportamiento.

Singleton (Instancia Única)

Es un patrón creacional que garantiza la existencia de una única instancia para una clase y la creación de un mecanismo de acceso global a dicha instancia. En el caso de la herramienta el patrón se utiliza para establecer una conexión SSH a un servidor, debido a que con una sola instancia de la conexión se realizan las operaciones en el sistema.

Análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS

2.6.3 Patrón DAO

Este patrón se usa para abstraer y encapsular la lógica de acceso a datos en una clase. El objeto de acceso a datos abstrae a las clases del negocio de la implementación de acceso a datos y habilita el acceso transparente a la fuente de datos. Ejemplo clase DAO

Conclusiones parciales

A partir del estudio realizado en el Capítulo 2 sobre el análisis y diseño de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS, se definieron las funcionalidades y características de la herramienta propuesta, donde se obtienen 11 requisitos funcionales y 4 requisitos no funcionales. Se utilizó la arquitectura definida por el framework Vue, n-capas orientada al dominio, para mantener la homogeneidad de la herramienta. Además, el empleo de patrones del diseño GRASP y GoF permiten obtener un diseño robusto de la propuesta a implementar.

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

3.1 Introducción

En el presente capítulo se especifican los estándares de codificación utilizados durante el desarrollo de la solución para lograr una uniformidad respecto a las demás herramientas. Se definen y se llevan a cabo pruebas al sistema, de las que se documentan los resultados. Se realiza además el modelado del diagrama de despliegue, incluyendo la distribución de los componentes del sistema.

3.2 Implementación

La etapa de implementación del software es el proceso mediante el cual el sistema crea una versión ejecutable. Esta fase comprende la organización del código en términos de subsistemas estructurados en capas, por lo que permite codificar y estructurar clases y objetivos en términos de componentes ya sea el código fuente, ejecutables o base de datos (33).

Una vez desarrollado el software, el mismo debe ser sometido a una serie de pruebas donde se verifica las especificaciones para asegurar que sea lo que el cliente requiere. Según Somerville, esta etapa se conoce como validación del software y es el proceso de comprobar que el sistema se conforma a su especificación y que satisface las necesidades reales de los usuarios del sistema (33).

3.3 Estándar de codificación

A partir de las especificaciones definidas en la herramienta de administración remota del servicio DNS se definen los siguientes estándares empleados para el desarrollo de la herramienta.

Asignación de nombres: Emplear descriptores en inglés. Evitar nombres largos que difieran en una letra o mediante el uso de mayúsculas. En el caso de los nombres de variables y funciones se escriben con la primera letra en minúsculas.

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

Ficheros de código fuente: Cada fichero contiene una única clase o interfaz. Si hay una clase privada o una interfaz asociadas a una clase pública se puede poner en el mismo fichero. La clase pública debe ser la primera.

Indentación: La unidad de indentación de bloques de sentencias son 4 espacios.

Comentarios: Los comentarios deben añadir claridad al código, deben contar el por qué y no el cómo y ser concisos.

Declaraciones: Se debe declarar cada variable en una línea distinta, de esta forma cada variable se puede comentar por separado.

Inicialización: Inicializar cada variable en su declaración a menos que su valor inicial dependa de algún cálculo.

Líneas largas: Cuando una sentencia no quepa en una unidad de línea se debe fraccionar después de una coma u operador, y alinear a la nueva línea al mismo nivel con el comienzo de la expresión de la línea anterior.

Longitud de líneas: Limitar todas las líneas a un máximo de 120 caracteres.

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

3.4 Diagrama de Componentes

El diagrama de componentes ilustra cómo va a estar dividido los componentes y sus dependencias en la propuesta de solución. Los diagramas de componentes implementan un componente en tiempo de ejecución. Son bloques de construcción, por lo que un componente puede abarcar una gran parte de un sistema (31).

A continuación, se muestra en la Fig 7 el diagrama de componentes de la propuesta de solución.

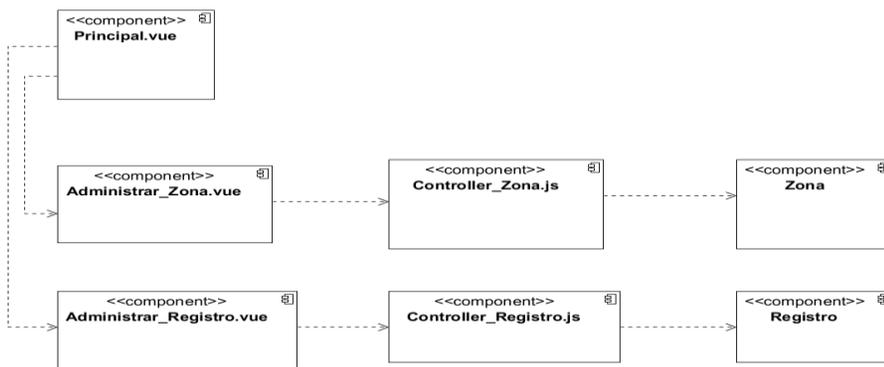


Fig 7: Diagrama de componentes de la propuesta de solución

Descripción de los componentes:

Principal.vue: muestra la interfaz principal de la herramienta.

Administrar_Zona.vue: contiene la información para gestionar una zona.

Administrar_Registro.vue: contiene la información para gestionar un registro de recursos de una zona.

Controller_Zona.js: permite administrar una zona DNS que se encuentra en el fichero de zona que se almacena en la carpeta /etc/bind9.

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

Controller_Registro.js: permite administrar un registro de recursos en la zona DNS que se encuentra en la carpeta /etc/bind9.

Zona: Contiene los datos de una zona DNS.

Registro: Contiene los datos de un registro de recursos.

3.5 Diagrama de despliegue

El diagrama de despliegue muestra como los componentes de software se despliegan físicamente en los procesadores, es decir, muestran como los componentes de software se distribuyen a través de plataformas de hardware (33).

A continuación, se muestra en la Fig 8 el diagrama de despliegue de la propuesta de solución.

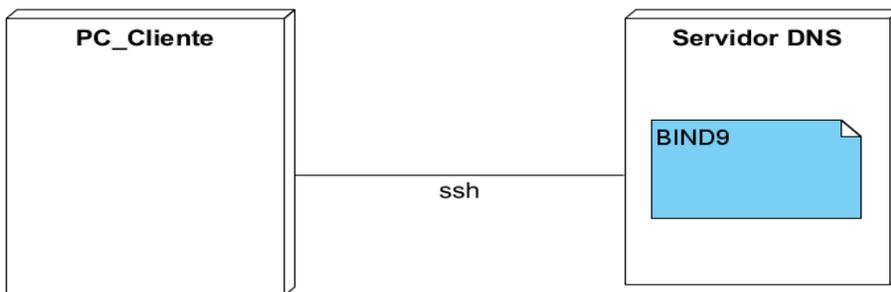


Fig 8: Diagrama de despliegue de la propuesta de solución.

Descripción de los nodos:

PC_Cliente: contiene la herramienta informática para la administración remota del servicio de resolución de nombres de dominio DNS.

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

Servidor DNS: es el servidor donde se realizan las configuraciones del servicio de nombres de dominio DNS.

3.6 Pruebas

Luego del desarrollo de la propuesta de solución, se hace necesario validar y verificar el sistema implementado a través de una estrategia de pruebas que permiten comprobar el cumplimiento de las especificaciones del diseño y de la codificación e identificar los posibles errores cometidos.

Las pruebas de software son un elemento crítico para la calidad del software y representan una revisión final de las especificaciones del diseño y de la codificación. En el desarrollo del software se plantean un conjunto de pruebas que permiten constatar que el software desarrollado satisface las especificaciones de esa fase.

3.6.1 Pruebas unitarias

Las pruebas de unidad consisten en comprobar que la herramienta funciona correctamente y que todos los caminos de control importantes se ejecutan de acuerdo con los especificado. La prueba de unidad enfoca los esfuerzos verificación en la unidad más pequeña del diseño de software (34). Las pruebas de unidad son de caja blanca y caja negra y tratan fundamentalmente de:

1. Analizar que la información entra y sale correctamente.
2. Las estructuras de los datos locales, ya que son una fuente potencial de errores.
3. Los valores límites establecidos para la herramienta.
4. El tratamiento de errores, comprobándose que se ejecutan y que son los adecuados.

Prueba de caja blanca: Se llevan a cabo sobre la interfaz del software y pretenden demostrar que el software funciona adecuadamente, es decir, que las entradas se aceptan de forma adecuada y que se produce una salida correcta. Estas pruebas no tienen en cuenta la estructura lógica interna del software.

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

Prueba de caja negra: Se basa en un minucioso examen de los detalles procedimentales. Se comprueban los caminos y lógicas de software en base a examinar trozos específicos del programa.

Por lo general se suelen combinar las pruebas de caja negra y caja blanca de forma que validen la interfaz del software y aseguren a la vez que el funcionamiento interno del software es correcto.

3.6.2 Pruebas de validación

Las pruebas de validación proporcionan un aseguramiento final donde se verifica que el sistema cumple con todos los requisitos funcionales y especificaciones. La validación de software se logra a través de una serie de pruebas que demuestran conformidad con los requerimientos. Una parte importante en el desarrollo de un sistema lo constituye la validación de los datos que interactúan con el usuario y el análisis de los errores que pueden generar. Tener esto en cuenta y solucionarlo contribuye en gran medida a la calidad del producto terminado (31).

Para realizar estas pruebas se aplicaron pruebas funcionales ya que se concentran en las acciones visibles para el usuario y en la salida del sistema que este puede reconocer. La validación se alcanza cuando el software funciona de tal forma que satisface las experiencias razonables del cliente.

A continuación, se muestran los casos de pruebas correspondientes a las historias de usuario "Adicionar Zona" y "Eliminar Registro".

Tabla 7: Caso de prueba de la historia de usuario No1 Adicionar Zona

Escenario	Descripción	Dominio	Respuesta del Sistema	Flujo Central
EC 1.1 Crear una zona DNS con datos los introducidos por el usuario.	El sistema permite crear una zona DNS en caso de que no exista un error.	Válido	Crea una zona DNS con datos entrados por el usuario.	1. Leer el dato del dominio entrado por el usuario 2. Registra
		uci.cu		

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

				una zona DNS en el archivo de zona.
EC 1.2 Crear una zona DNS insertando valores inválidos.	El sistema permite verificar si existen valores inválidos.	Inválido	No crea una zona DNS, muestra errores donde ocurren los valores inválidos.	<ol style="list-style-type: none"> 1. Leer el dato del dominio entrado por el usuario. 2. Verificar si existen campos vacíos. 3. Envía error en la interfaz en caso de que hayan valores inválidos para que el usuario arregle el error en el campo señalado.
		uci		
EC 1.3 Crear una zona DNS con campos vacíos.	El sistema permite verificar si existen campos vacíos.	Válido	No crea una zona DNS, muestra que existen campos vacíos en el sistema.	<ol style="list-style-type: none"> 1. Leer el dato del dominio entrado por el usuario. 2. Verificar si existen
		vacío		
		Válido		
		uci.cu		

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

				campos vacíos. 3. Envía error a la interfaz en caso de hayan campos vacíos para que el usuario arregle el error en el campo señalado.
--	--	--	--	--

Tabla 8: Caso de prueba de la historia de usuario No.2 Eliminar Registro

Escenario	Descripción	Dominio	Respuesta del Sistema	Flujo Central
EC 1.1 Eliminar registro seleccionando el campo por el usuario.	El sistema permite eliminar un registro en caso de que no exista un error	Válido	Elimina el registro del dominio seleccionado	1. Leer el dato del dominio entrado por el usuario 2. Elimina el registro del dominio.
		uci.cu		
EC 1.2 Eliminar un registro sin	El sistema permite verificar si existen	Inválido	No elimina el registro, muestra errores donde	1. Leer el dato del dominio

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

seleccionar el dominio.	valores inválidos.		ocurren valores inválidos.	valores	entrado por el usuario.
		null			2. Verificar si existen campos vacíos.
		vacío			3. Envía error en la interfaz en caso de que hayan valores inválidos para que el usuario arregle el error en el campo señalado.
		Válido			
		uci.cu			

3.6.3 Pruebas de rendimiento

Las pruebas de rendimiento se realizan para determinar el rendimiento del software en tiempo de ejecución, dentro del contexto de un sistema integrado. Pueden demostrar que el sistema cumpla con los criterios de rendimiento y comparar dos sistemas para encontrar cuál de ellos funciona mejor. La prueba de rendimiento ocurre a lo largo de todos los procesos de prueba. Incluso en el nivel de unidad, puede accederse al rendimiento de un módulo individual conforme se realizan las pruebas. Sin embargo, no es sino hasta que todos los elementos del sistema están plenamente integrados cuando puede determinarse el verdadero rendimiento del sistema (33).

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

La herramienta que se utilizó para realizar las pruebas de rendimiento fue TestStudio que es una herramienta que funciona como un todo y puede hacer pruebas de rendimiento, carga, gestión de pruebas y test exploratorio.

Hardware de prueba (PC servidor):

Sistema Operativo: Linux Mint 18

Microprocesador: Intel(R) Core(TM) i5-7300HQ CPU@ 2.50GHz 2.50 GHz.

Memoria RAM: 12.0GB

Disco Duro: 1024 GB

GPU: NVIDIA GeForce GTX 1050ti

Software Instalado en la PC (Electron JS)

Versión: 7.1.2

Sistema Operativo: GNU/Linux Nova 7.0

CPU: Intel(R) Core(TM) i3-4100U CPU@ 2.40GHz 2.40 GHz.

RAM: 4.00 GB

Los problemas detectados en el período de pruebas de validación y aceptación se clasificaron en: No Conformidades Significativas (NCS) y en No Conformidades No Significativas (NCNS). A continuación se describen los aspectos que se tuvieron en cuenta en cada clasificación.

NCS: son las no conformidades referentes a las funcionalidades de la aplicación: validaciones incorrectas o respuestas de la aplicación diferente a lo descrito previamente en las historias de usuario.

NCNS: son las no conformidades en cuanto al diseño de la propuesta de solución y errores ortográficos. Fueron realizadas 4 iteraciones de pruebas, ejecutándose al término de cada una de ellas pruebas de regresión, con el objetivo de asegurar que al resolverse las no conformidades detectadas, estas no introdujeran nuevos errores en la solución.

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

En la primera iteración se detectaron 4 NCS y 2 NCNS, las cuales fueron resueltas satisfactoriamente.

No Conformidades Significativas:

1. La aplicación se detuvo al registrar una zona donde el dominio era inválido.
2. La aplicación se detuvo al modificar un registro de una zona.
3. La aplicación no registro la zona de un dominio ya almacenado.
4. La aplicación se detuvo al crear un registro MX.

No Conformidades No Significativas:

1. La aplicación posee varias etiquetas con idiomas distintos.
2. La aplicación posee etiquetas sin traducir al idioma inglés.

En la segunda iteración fueron detectadas 3 NCS y 1 NCNS, estas siendo también solucionadas de igual manera que en la iteración anterior.

No Conformidades Significativas:

1. La aplicación no validó guardar los datos de los servidores conectados mediante la conexión .
2. La aplicación no validó que al presionar el botón Enter registrara una zona DNS.
3. La aplicación se detuvo editar el fichero de la zona.

No Conformidades No Significativas:

1. La aplicación muestra incorrectamente los textos presentes en la vista que se muestra al seleccionar la opción "Subred", que se encuentra en el panel desplegable.
2. Se detectaron en la lógica de la aplicación incorrectas llamadas a las pestaña principal del menú principal.

En la tercera iteración solo fue detectada una NCS, la cual fue solucionada satisfactoriamente.

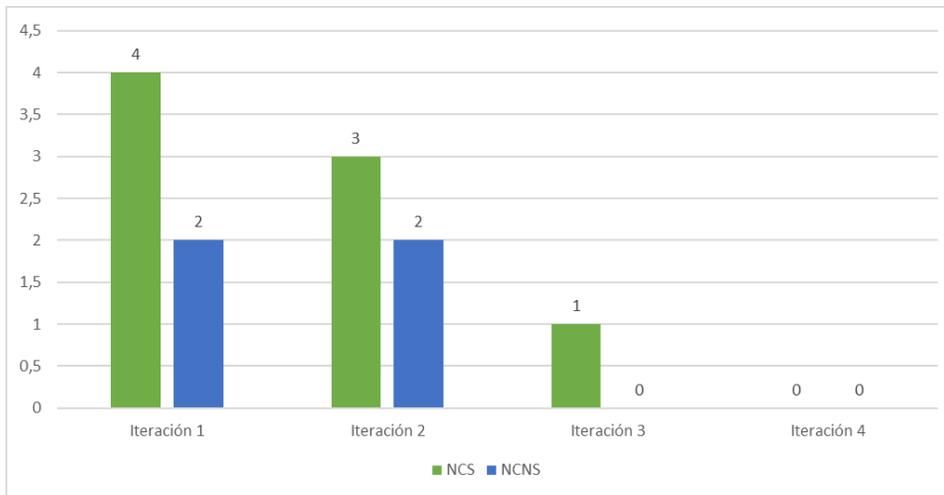
No Conformidades Significativas:

1. La aplicación no muestra el botón Aceptar al registrar un registro de tipo AAAA.

Por último en la cuarta iteración no fueron detectadas NCS ni NCNS, demostrándose entonces que la aplicación cumple con los requisitos funcionales establecidos y en consecuencia se considera concluida.

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

Se presenta a continuación un gráfico con un resumen de los resultados obtenidos en las cuatro iteraciones realizadas:



3.6.4 Prueba de aceptación

Las pruebas de aceptación pertenecen a la etapa final en el proceso de pruebas. Las pruebas son creadas en base a las historias de usuario, en cada ciclo de la iteración del desarrollo. El cliente debe especificar uno o diversos escenarios para comprobar que una historia de usuario ha sido correctamente implementada. Las pruebas de aceptación son consideradas como pruebas de caja negra. Los clientes son los **de verificar** que los resultados de estas pruebas son correctos. Asimismo, en caso de que fallen varias pruebas, debe indicar el orden de prioridad de resolución. Una historia de usuario no se puede considerar terminada hasta tanto pase correctamente todas las pruebas de aceptación (33). Durante la implementación de este sistema se diseñaron un conjunto de casos de prueba para comprobar su funcionamiento de acuerdo a los requerimientos funcionales descritos en las historias de usuarios, que

Comentado [m1]: Que verifican

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

fueron definidas en el capítulo anterior. A continuación, se exponen algunas de las pruebas de aceptación realizadas. En la tabla 9 se muestra una de las pruebas de aceptación realizadas al sistema.

Comentado [m2]: algunas

Caso de prueba aceptación		
Código de caso de prueba: RF_1	Nombre del requisito: Adicionar Zona	
Nombre de la persona que realiza la prueba: Yasiel Pérez Villazón		
Descripción de la prueba: Prueba la funcionalidad de adicionar una zona DNS		
Entrada/Pasos de ejecución: Luego de adicionar la zona DNS, se almacena en el archivo de zona. <ol style="list-style-type: none"> 1. Abrir remoteBind9 2. Seleccionar la opción de subred y conectarse al servidor. 3. Adicionar una zona DNS con datos introducidos por el usuario. 		
Escenarios:	Resultados Esperado	Evaluación de la prueba:
EC 2.1 Adicionar zona con datos introducidos.	Adiciona la zona con datos introducidos por el usuario.	Satisfactoria.
EC 2.2 Adicionar la zona con valores invalidos .	No adiciona la zona en caso de que existan valores incorrectos	Satisfactoria.
EC 2.3 Adicionar la zona con campos vacíos.	Muestra un mensaje de alerta de que existen campos vacíos.	Satisfactoria.

Comentado [m3]: falta pie de tabla

Comentado [m4]: inválidos

Resultados de las pruebas

Las pruebas de aceptación arrojaron como resultado que no existía ninguna no conformidad por parte del cliente. Los resultados en los casos de pruebas fueron satisfactorios por lo que se acepta el producto.

Capítulo 3: Implementación y pruebas de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS.

Conclusiones parciales

En el capítulo concluido se realizó la elaboración del diagrama de componentes, lo que permitió observar la integración de los componentes de software. Se establecieron los estándares de codificación a tener en cuenta para la implementación de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS permitiendo obtener en el sistema un código legible, estándar y fácil de comprender lo que asegura la calidad y facilita un futuro mantenimiento. Como mecanismo para asegurar la correcta ejecución de las funcionalidades del sistema se realizaron las pruebas de rendimiento, funcionalidad y aceptación. La consulta a expertos permitió identificar y corregir las no conformidades detectadas para obtener un producto de mayor calidad.

Conclusiones generales

Conclusiones generales

La realización de la presente investigación permitió desarrollar una herramienta que permite la administración remota del servicio de resolución de nombres de dominio DNS en GNU/Linux Nova, al dar respuesta a los planteamientos propuestos en las preguntas científicas descritas al inicio del trabajo. A partir de lo antes mencionado se concluye que:

- El estudio de las herramientas para la administración remota del servicio de resolución de nombres de dominio DNS en los sistemas operativos GNU/Linux permitió comprender su funcionamiento y estructura.
- A partir del análisis de las herramientas homólogas se logró identificar las funcionalidades a desarrollar en la solución propuesta.
- Mediante el diseño e implementación de la herramienta se logró un sistema informático que permite administrar las zonas y registros de un dominio determinado de forma remota.
- A partir de la ejecución de las pruebas se logró validar el correcto funcionamiento del sistema, además del cumplimiento de los requisitos definidos.

Recomendaciones

Recomendaciones

Una vez concluido el presente trabajo de diploma se recomienda, en función de ampliar las funcionalidades de la herramienta para la administración remota del servicio de resolución de nombres de dominio DNS:

- Permitir la administración remota del servicio DNS en otros sistemas operativos.

Referencias Bibliográficas

Referencias Bibliográficas

1. YOEDUSVANY HERNÁNDEZ MENDOZA, YORDANIS ARENCIBIA LÓPEZ and YANKIER CRESPO GONZÁLEZ. Sistema para la gestión de servicio de resolución de nombres de dominios. [online]. 2015. Available from: <https://dialnet.unirioja.es/servlet/articulo?codigo=4817566>
2. ANTONIO LÓPEZ PADILLA. *Guía de seguridad en servicios DNS* [online]. Available from: https://www.incibe.es/extfrontinteco/img/File/intecocert/ManualesGuias/guia_de_seguridad_en_servicios_dns.pdf
3. MOHAMMAD HAJJAR, ANIS ISMAIL and HAISSAM HAJJAR. *REMOTE ADMINISTRATION TOOLS* [online]. 2010. Available from: <https://books.google.com/cu/books?id=54z7MgEACAAJGoogle-Books-ID:54z7MgEACAAJ>
4. SIERRA, Ana María Sanz. *Servidor DNS, un enfoque práctico* [online]. 2010. Available from: <https://books.google.com/cu/books?id=54z7MgEACAAJGoogle-Books-ID:54z7MgEACAAJ>
5. BONET ESTEBAN, Enrique V. *Servicios avanzados I: Servidor de nombres (DNS)*. [no date].
6. OTROS, ROMERO Y. *Servicios en Red* [online]. Editorial Paraninfo, 2010. ISBN 978-84-9732-765-7. Available from: <https://books.google.com/cu/books?id=aOb3rTabO44CGoogle-Books-ID:aOb3rTabO44C>
7. *Instalación, configuración y administración de servidores DNS*. [no date].
8. *DNS. Servicios de nombres*.
9. JOSÉ LUIS COMESAÑA. Servicio de Nombres de Dominio. Aplicaciones Web. [online]. 3 January 2013. Available from: <http://www.sitiolibre.com/curso/pdf/DAW05.pdf>
10. NOITE.PL., DNSmasq mechanism. *Linux services*. [online]. 2016. Available from: https://books.google.com/cu/books?id=rPz2CwAAQBAJ&vq=dn+smas+q&dq=dnsmasq&hl=es&source=gb_s_navlinks_sGoogle-Books-ID:rPz2CwAAQBAJ
11. Unbound. About. NLnet Labs [.]. [online]. 28 November 2019. Available from: <https://nlnetlabs.nl/projects/unbound/about/>
12. Welcome to PowerDNS. [online]. [Accessed 1 December 2019]. Available from: <https://www.powerdns.com/documentation.html>

Referencias Bibliográficas

13. CRICKET LIU and JOHN BELAMARIC. Learning CoreDNS [Book]. [online]. 2019. [Accessed 29 November 2019]. Available from: <https://www.oreilly.com/library/view/learning-coredns/9781492047957/>
14. COREDNS. CoreDNS Manual. [online]. 16 February 2018. Available from: <https://coredns.io/manual/toc/Manual> covering all aspects of getting and running CoreDNS.
15. Overview of Pi-hole. [. [online]. 28 November 2019. Available from: <https://docs.pi-hole.net/>
16. Internet Systems Consortium. [online]. 25 November 2019. Available from: <https://www.isc.org/bind/>
17. AITCHISON, Ron. *Pro DNS and BIND 10* [online]. 2011. Available from: https://books.google.com/cu/books?id=siyW763RugsC&dq=bind9+vs+bind10&hl=es&source=gbs_navlinks_sGoogle-Books-ID: siyW763RugsC
18. Webmin. [online]. [Accessed 17 March 2020]. Available from: <http://www.webmin.com/>
19. CentOS Web Panel. [online]. [Accessed 3 August 2020]. Available from: Available from: <http://centos-webpanel.com/cwp-configuration>
20. Open Source Web Hosting and Cloud Control Panels | Virtualmin. [online]. [Accessed 18 March 2020]. Available from: Available from: <https://www.virtualmin.com/>
21. PEÑARROCHE DELGADO, JOSÉ ANGEL, SPECK GARCÍA, Nurianar. *Herramienta para la gestión de información, medias y servicios en la sede principal de la Central de Trabajadores de Cuba*. [online]. 2015. [Accessed 26 November 2019]. Available from: <http://repositorio.uci.cu/jspui/handle/123456789/7382>
22. Introducción — Vue.js. [online]. [Accessed 19 February 2020]. Available from: <https://es.vuejs.org/v2/guide/Vue.js> - The Progressive JavaScript Framework
23. Vue Material Design Component Framework — Vuetify.js. [online]. [Accessed 19 February 2020]. Available from: <https://vuetifyjs.com/es-MX>
24. GAUCHAT, Juan Diego. *El gran libro de HTML5, CSS3 y JavaScript* [online]. 2012. Available from: ISBN 978-84-267-1782-5. Google-Books-ID: szDMIRzwwUC
25. Acerca de Electron | Electron. [online]. [Accessed 26 November 2019]. Available from: <https://electronjs.org/docs/tutorial/about>
26. *Documentation for Visual Studio Code*. [online]. [no date]. Available from: <https://code.visualstudio.com/docs> Find out how to set-up and get the most from Visual Studio Code. Optimized for building and debugging modern web and cloud applications. Visual Studio Code is free and available on your favorite platform - Linux, macOS, and Windows.

Referencias Bibliográficas

27. What is UML | Unified Modeling Language. [online]. [Accessed 1 December 2019]. Available from: <https://www.uml.org/what-is-uml.htm>
28. About Visual Paradigm. [online]. [Accessed 1 December 2019]. Available from: <https://www.visual-paradigm.com/aboutus/>
29. CRAIG LARMAN. *UML y Patrones* [online]. [no date]. Available from: https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=2ahUKEwir1Y-7IdDnAhUE11kKHQO-DmoQFjAAegQIBRAB&url=http%3A%2F%2Froa.ult.edu.cu%2Fbitstream%2F123456789%2F2549%2F1%2FUML%2520y%2520Patrones%2520%2528Craig%2520Larman%2529.WWW.FREELIBROS.COM.pdf&usg=AOvVaw2Exb018_ms9gNIHziWcVRN
30. CARDOZZO, Daniel Ramos. *Desarrollo de Software: Requisitos, Estimaciones y Análisis*. [online]. 2 Edición. IT Campus Academy, 2016. [no date]. Available from: ISBN 978-1-5300-8861-4. Google-Books-ID: tBaYCwAAQBAJ
31. Roger S. Pressman. *Ingeniería de Software. Un enfoque Practico*. 7ma Edicion. [no date].
32. Erich Gamma. *Patrones de Diseño 1ra Edicion* [online]. [no date]. Available from: https://kupdf.net/download/patrones-de-diseo-1ra-edicion-erich-gammampdf_59b6e8bf08bbc5390ffdea5_pdf
33. Ian Sommerville. *Ingeniería de Software*. Novena Edicion. [no date].
34. KENDALL, Kenneth E. *Análisis y diseño de sistemas*. Pearson Educación, 2005. ISBN 978-970-26-0577-5.

Anexos