

Universidad de las ciencias informáticas

Facultad 1



**Herramienta informática para la administración
remota del servicio DHCP**

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

Autora

Yanisbel López Pérez

Tutores

MSc. PA Yurisbel Vega Ortiz

Ing. Enrique Muschett Cortina

La Habana, junio de 2020
Año 62 de la Revolución

DECLARACIÓN DE AUTORÍA

Declaro por este medio que yo **Yanisbel López Pérez**, con carné de identidad **97110100515** soy la autora principal del trabajo titulado **“Herramienta informática para la administración remota del servicio DHCP”** 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 firmamos la presente a los ____ días del mes ____ del_____

Yanisbel López Pérez
Autora

MSc. PA Yurisbel Vega Ortiz
Tutor

Ing. Enrique Muschett Cortina
Tutor

AGRADECIMIENTOS

DEDICATORIA

Este trabajo de diploma va dedicado a esas personas que creyeron en mí, que siempre pusieron un granito de arena, que aunque pequeño fuera, son grandes cosas para mí.

A mi mamá, la que siempre estará ahí, en lo malo y en lo bueno;

A mi papá, que junto a mi mamá me hicieron ser la persona que soy hoy;

A mi hermano, el mejor de todos los hermanos peleándonos pero a la vez amándonos;

A mis tutores, los más dedicados de todos los tutores, en especial Yurisbel;

A Osvaldo que ha sido mi principal apoyo en este proceso de tesis;

Y a mi familia que quiero tanto.

Resumen

La Universidad de la Ciencias Informáticas ha sido uno de los principales protagonistas en el proceso de migración a software libre. La misma ha contribuido al desarrollo de aplicaciones y estrategias para hacer más eficiente el despliegue de las tecnologías libres en las empresas cubanas. Entre los resultados más evidentes está el desarrollo de la distribución cubana de GNU/Linux Nova. Un sistema operativo que se despliega en cuatro variantes: Nova Escritorio, Nova Ligero, NovaDroid y Nova Servidor, este último destinado a la administración de servicios telemáticos. Las empresas normalmente tienen redes informáticas y por ende servidores que proveen servicios telemáticos como el de Asignación Dinámicas de Host (DHCP) que necesitan ser configurados adecuadamente por los administradores. En el presente trabajo se desarrolla una herramienta para la administración remota del servicio DHCP con el objetivo de configurar a distancia este servicio de host en las empresas. Se realiza un estudio de sistemas homólogos identificándose funcionalidades a tener en cuenta en la nueva herramienta. Para la implementación de la propuesta de solución se emplea la siguiente base tecnológica: el marco de trabajo Vue.js 2.6.12, NPM v3.5.2, Node.js v3.5.2, Electron framework en su versión 7.1.2 y el Visual Studio Code como editor de código. Como herramienta de modelado se utiliza el Visual Paradigm en su versión 8.0. Para validar la propuesta de solución se diseña una estrategia de prueba para asegurar que el sistema sea funcional y cumpla con los requisitos identificados con el cliente. Se realizaron pruebas funcionales, pruebas unitarias; caja blanca y la técnica de camino básico y las pruebas de regresión. Al culminar este trabajo se proporcionó una herramienta ajustadas a las exigencias del cliente.

Palabras clave: Administración remota, DHCP, GNU/Linux Nova, migración, servicio telemático, tecnologías libres.

Índice

RESUMEN	1
INTRODUCCIÓN	7
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.....	15
1.1 CONCEPTOS RELACIONADOS.....	15
1.2 ESTUDIO HOMÓLOGO.....	15
1.2.1 Webmin 1.930.....	15
1.2.2 Zentyal	16
1.2.3 YaST	18
1.3 METODOLOGÍA DE DESARROLLO DE SOFTWARE.....	21
1.3.1 Metodología AUP-Variación UCI	21
1.4 LENGUAJES Y HERRAMIENTAS PARA EL MODELADO DE LA SOLUCIÓN	22
1.4.1 Lenguaje Unificado de Modelado (UML) v8.0	22
1.4.2 Herramienta CASE: Visual Paradigm v5.0.....	22
1.5 TECNOLOGÍAS DE LA IMPLEMENTACIÓN	23
1.5.2 Framework de desarrollo.....	24
CAPÍTULO 2: CARACTERÍSTICAS Y DISEÑO DE LA HERRAMIENTA PARA LA ADMINISTRACIÓN REMOTA DEL SERVICIO DHCP.....	26
2.1 INTRODUCCIÓN	26
2.2 DESCRIPCIÓN DE LA PROPUESTA DE SOLUCIÓN.....	26
2.2.1 Modelo conceptual	27
2.2 REQUISITOS.....	28
2.2.1 Fuentes para la obtención de requisitos	28
2.2.2 Técnicas de identificación de requisitos	29
2.2.3 Especificación de requisitos de software	30
REQUISITOS NO FUNCIONALES.....	33
2.2.4 Descripción de los requisitos de software mediante historias de usuarios.....	34
2.2.5 Validación de requisitos de software	36
2.3.1 Diseño arquitectónico	37
2.3.2 Diagrama de clases por paquete	39
2.3 Diseño	40
2.3.1 Diseño de clases.....	40
CONCLUSIONES DEL CAPÍTULO	42
CAPÍTULO 3: IMPLEMENTACIÓN Y EVALUACIÓN DE LA HERRAMIENTA PARA LA ADMINISTRACIÓN REMOTA DEL SERVICIO DHCP.....	44
3.1 IMPLEMENTACIÓN.....	44
3.1.1 Estándares de codificación	44
3.2 DIAGRAMA DE DESPLIEGUE	45
3.3 PRUEBAS DE SOFTWARE.....	46
3.3.1 Pruebas internas	46
CONCLUSIONES GENERALES	49

Introducción

En los años 60 durante la guerra fría, las potencias involucradas en el conflicto tuvieron la necesidad de crear un sistema de comunicación distribuida que pudiera soportar cualquier catástrofe. Debido a que en toda guerra la información es vital, Estados Unidos crea una red exclusivamente militar, con el objetivo de que, en el hipotético caso de un ataque soviético, se pudiera tener acceso a la información desde cualquier punto del país. La red pasó de las agencias militares a las universidades y proyectos de la defensa de la nación cada vez con mayor fuerza, surgiendo así ARPANET. Luego con el auge de la comercialización de computadoras, el número de ordenadores conectados fue aumentando y a partir de los 80 aparecieron otras redes, lo que provocó que el año 1983 se considere el año del nacimiento del Internet(1).

El establecimiento de interconexiones entre ordenadores que proporcione una transmisión fiable de la comunicación y el intercambio de datos, se realiza sobre protocolos que en el día de hoy se han estandarizado en el mundo de la informática y las comunicaciones, tal es el caso del Protocolo de Control de Transmisión/Protocolo de Internet, Transmisión Control Protocol/Internet Protocol por sus siglas en inglés, (TCP/IP). En la relación entre los dispositivos que utilizan este protocolo, es preciso que se reconozcan entre ellos y para eso se necesita un identificador público para cada uno, al cual hoy en día se le denomina Protocolo de Internet o Internet Protocol, por sus siglas en inglés IP(2).

La asignación manual de estas direcciones públicas se convirtió en un inconveniente cuando era necesario trasladarse a cada lugar para darle un identificador a cada dispositivo, surge entonces el protocolo Bootstrap (*BootP*). BootP fue de los primeros métodos para asignar de forma dinámica direcciones IP a otros equipos (ordenadores, impresoras, etcétera.). Al ser las redes cada vez más grandes, BootP ya no era tan adecuado y surge entonces el Protocolo de Configuración Dinámica de Host, Dynamic Host Configuration Protocol por sus siglas en inglés (DHCP), el cual fue creado para cubrir las nuevas demandas. Este, es un protocolo de red de tipo cliente/ servidor, el cual asigna automáticamente información TCP/IP a equipos cliente. Cada cliente DHCP se conecta a un servidor DHCP centralizado que devuelve la configuración de red del cliente, incluyendo dirección IP, la puerta de enlace y los servidores de sistemas de nombres de dominio, DomainNameSystem por sus siglas en inglés, (DNS)(3).

A partir del mes de octubre del año 1996, Cuba quedó oficialmente conectada a la red internacional de computadoras INTERNET para dar acceso, a todos los profesionales de las diferentes instituciones del país, al gran cúmulo de información existente en el mundo en los más disímiles temas (4). Aunque la evolución del servicio desde su origen ha sido vertiginosa y siempre el ascenso, el bloqueo impuesto por Estados Unidos también llega en el área de la conectividad y las soluciones de red.

A partir de las experiencias vividas por el mundo y el particular asedio en el que se ha encontrado Cuba, incluido también el campo de la informática y las comunicaciones, en el año 2004 se publica por consenso del Consejo de Ministros de la República de Cuba el acuerdo 084, que orienta la migración paulatina de los Organismos de la Administración Central del Estado (OACE) hacia aplicaciones de código abierto(5).

La Universidad de las Ciencias Informáticas (UCI) forma parte del grupo técnico nacional y ha sido uno de los principales protagonistas en este proceso de migración. La UCI ha contribuido al desarrollo de aplicaciones y estrategias para hacer más eficiente el despliegue de las tecnologías libres en las empresas cubanas. Entre los resultados más evidentes está el desarrollo de la distribución cubana de GNU/Linux Nova, un sistema operativo que se despliega en cuatro variantes: Nova Escritorio, Nova Ligerero, NovaDroid y Nova Servidor, este último destinado a la administración de servicios telemáticos(6). Esta distribución está basada en cuatro principios, llamadas 4S:

- Soberanía tecnológica: Se basa en el principio de un desarrollo de forma autónoma, así como la capacidad decisonal sobre su uso y desarrollo.
- Socio-adaptabilidad: Un sistema operativo hecho por cubanos para cubanos, alineado a las políticas que orienta la informatización nacional y optimizado para las condiciones tecnológicas del país.
- Sostenibilidad: Se basa en el principio de mantener un proceso flexible y versátil en la construcción de la distribución. Promueve un desarrollo en constante innovación y consonancia con las nuevas tendencias tecnológicas. El equipo de desarrollo garantiza las actualizaciones, mejoras, valor agregado y el soporte del sistema luego de su despliegue a la sociedad.
- Seguridad: Este principio propone un modelo de desarrollo colaborativo que permita el acceso al código fuente y el exhaustivo proceso de revisión y auditoría del código para garantizar un sistema seguro(7).

La primera etapa del proceso de migración, según define la actualización de la “Guía Cubana para la Migración a Software Libre”, publicada bajo el título “Buenas prácticas para la migración a código abierto”, es realizar la Consultoría en Migración a Código Abierto. Esta actividad consiste en planificar todas las actividades que serán realizadas durante el diagnóstico a la entidad. Entre las actividades más importantes en el diagnóstico de la entidad están:

- Levantamiento de activos informáticos de la entidad: Consiste en registrar cuáles son las principales herramientas que utilizan los usuarios de la empresa. También se registran las propiedades de las estaciones de trabajo y de los dispositivos de hardware.
- Levantamiento de los servicios telemáticos: Consiste en identificar cuál es la distribución física de los servidores, cuáles son los servicios telemáticos que tiene la empresa y que tecnologías lo implementan.
- Entrevistas a usuarios y administradores de redes: Consiste en realizar entrevistas a los usuarios que interactúan con las estaciones de trabajo y servicios telemáticos(8).

Los resultados de la primera etapa del proceso de migración con los Organismos de la Administración Central del Estado (OACE) realizado por la UCI, permite identificar un grupo de debilidades y problemas a resolver por parte del Centro de Software Libre por sus siglas (CESOL) para llevar una migración exitosa. La adaptabilidad de las soluciones propuestas a las más disímiles formas de administración de red en la pequeña y mediana empresa cubana, deben ser lo suficientemente flexibles y escalables para apoyar el proceso de migración.

En la muestra obtenida a partir de los levantamientos realizados por la UCI se pudo identificar que la asignación dinámica de direcciones públicas en la pequeña y mediana empresa se hace a través del protocolo DHCP, y que la configuración del mismo se realiza de forma manual en los servidores que brindan el servicio. En una gran parte de la muestra analizada, existe más de un servidor DHCP en la empresa, lo que conlleva a que los administradores de red deban configurar cada uno de ellos de forma independiente. Debido a que los informes de diagnósticos realizados por la OACE para la migración a software libre son documentación confidencial. Se aplica una entrevista a varios miembros del centro que participaron en dicho proceso (ver anexo 1). Este comportamiento introduce variables que pueden afectar la disponibilidad e integridad del servicio que

pueden variar desde el error humano, hasta la duplicidad de rutinas casi idénticas por carecer de una administración remota y centralizada para la administración de los servicios de asignación dinámica de direcciones IP. Por otra parte, la distancia existente en ocasiones entre los varios servidores que alojan los servicios en una misma empresa, puede afectar la operatividad de los mismos y aumentar directamente el tiempo invertido por los administradores para la configuración del servicio. Además, la utilización de una herramienta específica para la administración del servicio DHCP alojada en el mismo servidor que presta el servicio, añade un consumo adicional de los recursos del mismo lo que puede provocar lentitud en el servicio.

Una vez identificada la situación descrita anteriormente, la presente investigación plantea como **problema científico**: ¿Cómo facilitar la administración de los servicios de asignación dinámica de direcciones IP (DHCP) en las pequeñas y medianas empresas?

Para darle solución al problema planteado la investigación se propone como **objeto de estudio**: “La administración de servicios de asignación dinámica de direcciones IP” enmarcado en el **campo de acción**: “Administración de servicios de asignación dinámica de direcciones IP utilizando DHCP3”.

Para dar respuesta al problema científico se plantea como **objetivo general**: Desarrollar una aplicación de escritorio para administrar de forma remota y centralizada el servicio DHCP de las pequeñas y medianas empresas cubanas.

Para dar cumplimiento al objetivo general planteado se definen los siguientes objetivos específicos:

1. Elaborar el marco teórico referencial de la investigación teniendo en cuenta las tecnologías y conceptos principales asociadas a la administración remota y centralizada del servicio DHCP para sentar las bases de la investigación.
2. Diseñar una aplicación de escritorio para la administración remota y centralizada del servicio DHCP que permita guiar la posterior implementación de la misma.
3. Implementar la aplicación de escritorio para la administración remota del servicio DHCP para dar solución a la situación problemática.
4. Validar la aplicación de escritorio para la administración remota del servicio DHCP para identificar y corregir los errores arrojados por la solución.

Hipótesis: El desarrollo de una aplicación de escritorio para la administración de servicios de asignación dinámica de direcciones IP, sustentada en tecnologías libres, contribuirá a la centralización y la accesibilidad de forma remota de la administración de servicios DHCP en las medianas y pequeñas empresas cubanas.

Variable independiente:

- Aplicación de escritorio para la administración de servicios de asignación dinámica de direcciones IP

Variable dependiente:

- Centralización de la administración de servicios DHCP
- Accesibilidad de forma remota a la administración de servicios DHCP

Definiciones conceptuales

- Aplicación de escritorio para la administración de servicios de asignación dinámica de direcciones IP: Constituye la parte de la solución a implementar y desplegar para dar respuesta al objetivo planteado.
- Centralización de la administración de servicios DHCP: Es la variable que permitirá medir el acceso de forma centralizada a toda la administración de los servicios de asignación dinámica de direcciones IP.
- Accesibilidad de forma remota a la administración de servicios DHCP: Es la variable que permitirá medir el acceso de forma remota a todos los servicios DHCP sin importar la ubicación de los servidores.

Definiciones operacionales

- Solución informática para la administración de servicios de asignación dinámica de direcciones IP: Se evaluará teniendo en cuenta la apariencia, rendimiento, aceptación y seguridad de la misma.
- Centralización de la administración de servicios DHCP: Se evaluará a partir de la clasificación del servicio en cuanto a la centralización del mismo.
- Accesibilidad de forma remota a la administración de servicios DHCP: Se evaluará a partir de la disponibilidad de forma remota de la plataforma.

Operacionalización de las variables:			
Variables	Dimensiones	Indicadores	Índice de los Indicadores
Solución informática para la administración de servicios de asignación dinámica de direcciones IP	Apariencia o interfaz externa	Utilización	Complejo
			Simple
	Rendimiento	Tiempo de Respuesta	Alto
			Medio
			Mínimo
	Social	Satisfacción	Alta
			Media
			Baja
	Seguridad	Integridad	Alta
			Media
			Baja
		Disponibilidad	Alta
Media			
Baja			
Centralización de la administración de servicios DHCP	Clasificación	Organización de la Información	Alto
			Medio
			Mínimo
Accesibilidad de forma remota a la administración de servicios DHCP	Disponibilidad	Accesibilidad al servicio	Alta
			Media
			Baja

Para el desarrollo de la presente investigación se utilizan los métodos de investigación científica. Los métodos teóricos y empíricos utilizados en el presente trabajo son:

Métodos teóricos:

Histórico-Lógico: se utiliza para llevar a cabo un análisis valorativo de la bibliografía existente, así como para conocer las tendencias actuales en cuanto a las configuraciones administrativas DHCP. También para el estudio de diferentes aplicaciones homólogas o similares para la administración del servicio DHCP

Análítico-Sintético: se utiliza para poder descomponer el problema, que en este caso es cómo facilitar la administración remota y centralizada del servicio DHCP. Este método permite analizar grandes volúmenes de información relacionados con la administración del servicio DHCP, así como un gran número de aplicaciones, terminologías y conceptos para finalmente identificar los elementos más importantes, resúmenes o síntesis de los argumentos fundamentales o esenciales.

Modelación: posibilita la creación de los diferentes diagramas y modelos, en cada una de las iteraciones según la metodología seleccionada, que ayudan a un mejor entendimiento de las funcionalidades del software y al estudio de las relaciones entre las mismas. Permitiendo así, realizar todos los diagramas para desarrollar el diseño de la aplicación de escritorio en el desarrollo del capítulo 2 del documento.

Métodos empíricos:

Entrevista: se utiliza para obtener información de algo determinado para con esa persona que está siendo entrevistada o saber de un tema específico. Se realizaron entrevistas a especialistas del Centro de Software Libre (CESOL) con experiencia en el empleo de las herramientas como son los administradores de los servicios de red, obteniendo de estos sus experiencias en la implementación y mantenimiento de servidores.

Consulta de fuentes de Información: se utiliza para determinar el estado del arte del objeto de investigación que es desarrollar el sistema de análisis de supervivencia, consultando una bibliografía completa y actualizada, referente a los métodos de análisis de supervivencia y la actualización de las bases de datos relacionadas con el tema.

El presente documento cuenta con 3 capítulos:

Capítulo 1: “Fundamentación teórica”, este contiene los principales conceptos que se utilizan, se definen las técnicas, tecnologías, metodologías y herramientas que se necesitan para el desarrollo del software a desarrollar.

Capítulo 2: “Características y diseño del sistema”, se describe la propuesta de solución, se especifican los requisitos del software, se describen las historias de usuario correspondiente a cada requisito funcional, la arquitectura y los patrones de diseños que se utilizarán durante de la implementación.

Capítulo 3: “Implementación y pruebas a la herramienta de administración remota y centralizada de DHCP”, se documentan los artefactos asociados a la implementación de la propuesta solución y se realizan los casos de prueba para validar la herramienta. Se

definen también los estándares de codificación que se tienen en cuenta para el desarrollo del software.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El presente capítulo contiene los principales conceptos relacionados con la investigación, además de un análisis de las soluciones actuales para las administraciones del servicio DHCP; sean libres o privativos. También se hace una caracterización de las tecnologías, metodologías, herramientas y lenguajes propuestos para el desarrollo de la aplicación de escritorio para la administración remota y centralizada del servicio DHCP.

1.1 Conceptos relacionados

A continuación, se describen los principales conceptos asociados al objeto de estudio de la investigación que es la administración de servicios de asignación dinámica de direcciones IP.

DHCP: Protocolo de configuración dinámica de host (en inglés Dynamic Host Configuration Protocol) DHCP, es un protocolo de red de tipo cliente/ servidor, el cual asigna automáticamente información TCP/IP a equipos cliente. Cada cliente DHCP se conecta a un servidor DHCP centralizado que devuelve la configuración de red del cliente, incluyendo dirección IP, la puerta de enlace y los servidores DNS(3).

Protocolo de internet: es un método de transmisión de datos por una red. Los datos que se envían se dividen en "paquetes" individuales y completamente independientes. Cada ordenador (o host) de Internet tiene como mínimo una dirección que lo identifica de forma exclusiva y lo distingue de todos los demás, y cada paquete de datos contiene la dirección del emisor y la del receptor(9).

1.2 Estudio Homólogo

Actualmente existen una gran variedad de aplicaciones que permiten la configuración de los servicios telemáticos. Para el proceso de selección de los requisitos funcionales que necesita la aplicación de escritorio a desarrollar y comprobar que una aplicación no resolviera el problema antes descrito, se realiza un estudio de las herramientas de escritorio webmin, YaST y Zentyal. A continuación, se describe estas aplicaciones:

1.2.1 Webmin 1.930

Webmin es una interfaz de escritorio para la administración y configuración de sistemas con distribuciones GNU/Linux y otros sistemas Unix de forma local y remota. Está debe ser instalada en cada servidor a administrar. La aplicación permite la configuración de la mayoría de los servicios telemáticos mediante módulos, entre ellos se encuentran: DNS,

DHCP, LDAP, de escritorio, Jabber, FTP, Correo, Proxy y SSH entre otros. Webmin se puede abrir por cualquier navegador que soporte tablas, formularios y Java para el módulo de administración de archivos, usando una conexión mediante capa de sockets seguros por sus siglas en inglés Secure Sockets Layer, (SSL); por el puerto 10000. Es una aplicación modular, escrita en el lenguaje Perl, lo que brinda la posibilidad de agregarle y quitarle funcionalidades. Sin embargo, no brinda todas las posibilidades de relación entre las declaraciones que ofrece el servidor y no permite exportar ni importar configuraciones de otro servidor DHCP y además decir que esta aplicación no es centralizada(10).

En la aplicación la ayuda que se ofrece a los usuarios no está completa, en el módulo de SSH se explica solamente los términos que son pocos usados, representando estos un bajo por ciento de las opciones para configurar que se muestran en la interfaz La aplicación brinda la posibilidad de configurar sus módulos, permitiendo cambiar el acceso a los ficheros con los con que trabaja, los cuales al ser modificados puede ocasionar que el módulo no funcione correctamente. Lo mismo sucede con los archivos de configuración de los servicios.

Se considera que el webmin no es adaptable a las empresas e instituciones de configuraciones que tiene en cada interfaz de usuario. Además, los módulos desarrollados hace varios años no han sido actualizados, por lo que puede causar problemas cuando se va a administrar un servicio que tiene una versión y en los repositorios tienen otra diferente. Esta herramienta tiene en cuenta cada detalle de configuración de los servicios de redes, por lo que para usar el mismo es necesario que se tengan conocimientos avanzados respecto a la administración de servidores. Tampoco está escrita la ayuda sobre el trabajo en todos los módulos.

1.2.2 Zentyal

Zentyal es desarrollado con el objetivo de acercar GNU/Linux a las PYMES (Pequeñas y medianas empresas) y permitirles aprovechar su potencial como servido. Es la alternativa en código abierto a Windows Small Business Server, basado en la distribución Ubuntu. Zentyal permite a todos los administradores de una institución configurar los servicios telemáticos necesarios en una infraestructura de servidores, tales como la seguridad de la red, la compartición de recursos, las comunicaciones, el acceso a Internet, entre otros, a través de una única plataforma y de forma sencilla.

En la aplicación, la ayuda que se ofrece a los usuarios es sencilla mostrándose en la parte superior izquierda de la mayoría de sus módulos, pero esta solo describe de forma general lo que puede hacer el módulo en el que se encuentra y no lo que puede hacer en cada opción de configuración necesaria. La administración de Zentyal se realiza por medio de un navegador de escritorio, lo que resulta accesible desde cualquier equipo que se disponga la red. Al ser una solución modular resulta bastante flexible para cubrir cualquier necesidad que tenga en la empresa, sea para crear una infraestructura más segura o un servidor de impresión dedicado.

Algunas de las funcionalidades principales de la aplicación son:

- ✓ Gestión de red: cortafuego, servidor DHCP, servidor NTP, servidor DNS, soporte para VPN, proxy TTP.
- ✓ Servidor de correo: POP3 e IMAP con SSL/TLS, filtro antispam y antivirus.
- ✓ Comunicaciones: FTP, servidor VoIP, Voicemail y servidor Jabber de mensajería instantánea.
- ✓ Compartición de recursos y trabajo en grupo: servidor de archivos, servidor de impresión y groupware (agenda, contactos).
- ✓ Gestión centralizada de usuarios mediante LDAP, sincronización con Directorio de Windows.
- ✓ Reportes y monitoreo del sistema, además del envío de notificaciones vía correo, RSS o jabber.
- ✓ Respaldos de configuraciones y datos de manera remota.
- ✓ Autoridad de certificación(11).

Esta aplicación se debe instalar en cada servidor a administrar, por defecto instala un servidor de escritorio (Apache) y uno de base de datos (Postgres), esto es innecesario cuando se quiere tener un servidor dedicado a otro servicio. Este sistema no permite gestionar correctamente un servidor DHCP, porque bloquea algunas versiones que en muchas ocasiones son necesarias. Por ejemplo, no se puede adicionar una nueva subred que no esté contenida en el servidor donde se instaló, así mismo permite solo las actualizaciones automáticas de DHCP con el servidor DNS que él instala. Las actualizaciones de Zentyal con garantía y calidad o unos de sus componentes se adquieren a través de una versión empresarial o profesional.

Esta herramienta no se puede adaptar a las necesidades de las empresas e instituciones cubanas, pues a diferencia del webmin que tenía muchas configuraciones, esta tiene muy pocas con respecto a las que se necesitan. También hay que destacar que este sistema no reconoce los cambios que se realicen directamente en los ficheros, por lo que todo lo que se necesite hacer tiene que ser por medio de su interfaz, debido a que la aplicación no es remota. En caso de escribir directamente en los ficheros, Zentyal lo borra cuando guarda sus configuraciones. Esto puede causar problemas ya que muchos administradores tienen ficheros de configuración guardados y cuando quieren restablecerlos no pueden, haciendo que los mismos sean completamente dependientes de este sistema para realizar cualquier cambio.

1.2.3 YaST

Es Otra Herramienta de Configuración Más, por sus siglas en inglés Yet another setup tool, (YaST). Esta es una herramienta de configuración e instalación OpenSUSE y SUSE Linux enterprise. Es popular por su facilidad de uso y atractiva interfaz gráfica, así como por la capacidad de configurar el sistema rápidamente durante la instalación y una vez terminada esta. Es un proyecto de código abierto bajo la licencia GPL patrocinado y desarrollado activamente por Novell. Siendo esta la herramienta con la que se interactúa durante la instalación de OpenSUSE. Es utilizado también para la configuración de sus dispositivos, así como entorno de red, servicios del sistema y la seguridad. Todas estas tareas pueden realizarse desde el centro de control YaSt(12).

YaST2 es una interfaz desarrollada en QT para la gestión de servicios telemáticos, la misma está conformada de forma modular, pero en este caso los módulos no se activan y desactivan como en otras herramientas similares, si no que esta requiere la instalación y desinstalación de cada uno de ellos según se necesiten. Al instalar un módulo no necesariamente significa que se instale el servicio pues esa versión es brindada al momento de configurar el mismo. Entre los módulos para la administración de servidores (según información del sitio oficial OpenSUSE) se encuentran las siguientes:

- ✓ DNS y Nombre del Host: Configurar el Nombre del Host (Hostname) y nombre de servidores.
- ✓ Servidor DHCP: Configurar un Servidor DHCP.
- ✓ Servidor HTTP: Configurar un Servidor de escritorio.
- ✓ Hostnames: Asignar nombres de host y alias para las direcciones IP.

- ✓ Cliente Kerberos: Configurar un cliente Kerberos.
- ✓ Cliente LDAP: Configurar el cliente LDAP.
- ✓ Agente de transferencia de correo: Configurar correo.
- ✓ Servidor NFS: Configurar un servidor NFS.
- ✓ Servidor NIS: Configurar un servidor NIS.
- ✓ Cliente NTP: Configurar un cliente NTP.
- ✓ Servicios de Red (xinetd): Configurar servicios de Red (xinetd).
- ✓ Proxy: Configurar un servidor Proxy

- ✓ Administración remota: Instalar la configuración remota.
- ✓ Encaminamiento o Routing: Configurar un Routing.
- ✓ Servidor Samba: Configurar el servidor Samba.
- ✓ Servidor TFTP: Configurar un servidor TFTP.

Al ejecutarse alguno de los anteriores módulos se verifica su instalación y se leen los archivos de configuración para así comprobar cambios realizados de su última ejecución. En caso de detectarse algún cambio informa al usuario el lugar de la modificación y advierte la posibilidad de que esto puede ser borrado. Por parte de la ayuda en estos módulos esta está conformada por pequeños diálogos, donde se describen las opciones que se muestran en la interfaz en donde se seleccionó la opción ayuda.

Esta herramienta no se puede adaptar a las características que tienen las empresas e instituciones cubanas ya que solo está desarrollada para las diferentes versiones de openSUSE, por lo tanto, habría que instalar este sistema operativo en los servidores y no la distribución cubana GNU/Linux Nova Server que es la propuesta para instalar en dichas empresas. Además, YaST fue creado principalmente para la administración completa del sistema operativo desde una sola aplicación, no para la administración de servidores. Esta no es remota ni centralizada.

Comparación de las herramientas

A continuación, se realiza una comparación entre las herramientas anteriormente descritas teniendo en cuenta su licencia, sistema operativo para el que está disponible, la facilidad de uso, la accesibilidad de forma remota, la seguridad, la detección de cambios manuales y la centralización de la solución.

Parámetros	Webmin	Zentyal	YaST
Licencia	BSD	GPL	GPL
¿Es una Herramienta privada?	No	No	No
Sistema Operativo	GNU/Linux	GNU/Linux	OpeSUSE
Facilidad de Uso	Baja	Media	Alta
¿Permite la Configuración remota?	No	No	No
¿Requiere estar instalada en el servidor que se desea administrar?	Sí	Sí	Sí
Seguridad	Media	Alta	Alta
¿Detecta cambios manuales en los archivos?	Sí	Sí	No
Permite la administración centralizada de varios servidores DHCP	No	No	No

Tabla 1: Comparación de las herramientas

En esta comparativa podemos apreciar que, a pesar de haber similitud entre algunas de estas herramientas, tienen sus diferencias. Todas estas herramientas son instalables en

sistemas UNIX y cuentan con licencias libres para variantes de cada una de ellas, aunque en el caso particular de Yast es solo utilizable en distribuciones OpenSUSE, lo que constituye un impedimento para esta investigación pues su objetivo es ser aplicable en la distribución cubana de GNU/Linux Nova. Por su parte, Zentyal tiene como inconveniente que no detecta la modificación manual de los archivos de configuración lo que atenta contra la integridad de la información. Todas estas herramientas tienen como desventaja la necesidad de estar instaladas en el servidor que se desea administrar afectando directamente la descentralización del servicio. Como una conclusión parcial podemos afirmar que ninguna de las herramientas analizadas realiza el proceso de configuración de DHCP de forma remota y centralizada.

1.3 Metodología de desarrollo de software

Una metodología es el conjunto de herramientas, técnicas, métodos y modelos que permiten conocer los elementos necesarios para definir un proyecto de software, es la base para la edificación de un producto de este tipo. Esencialmente, sirve para aumentar la "calidad" del software y controlar de manera transparente todo el proceso de desarrollo. Si se quiere que un proyecto sea escalable y flexible a los cambios, se recomienda tomar en cuenta una metodología de desarrollo de software(13).

1.3.1 Metodología AUP-Variación UCI

En el desarrollo de la aplicación se utiliza la metodología de desarrollo de software Variación de AUP (Proceso Ágil Unificado) para la UCI, una variante realizada por la Universidad de las Ciencias Informáticas a la metodología ágil AUP y está definida por la universidad como el documento rector de la actividad productiva.

Fases de Variación de AUP para la UCI:

La metodología Variación de AUP para la UCI está formada por tres fases, (Inicio, Ejecución y Cierre) para el ciclo de vida de los proyectos de la universidad(14). Las características de las fases de la metodología de la universidad son:

- Inicio: Durante el inicio del proyecto se llevan a cabo las actividades relacionadas con la planeación del proyecto. En esta fase se realiza un estudio inicial de la organización cliente que permite obtener información fundamental acerca del alcance del mismo.
- Ejecución: En esta fase se ejecutan las actividades requeridas para desarrollar el software, incluyendo el ajuste de los planes del proyecto teniendo en cuenta los requisitos

y la arquitectura. Durante el desarrollo, se obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto.

- Cierre: En el cierre se analizan tanto los resultados del proyecto como su ejecución y se realizan las actividades formales de cierre del proyecto(14).

El desarrollo de la investigación se centra en la fase de ejecución. A lo largo del proyecto se transita por las disciplinas: requisitos, análisis y diseño, implementación, pruebas interna y pruebas de aceptación, definidas de la metodología. Se propone esta metodología para el desarrollo de la solución informática debido a que es ágil y está diseñada para los procesos productivos de la UCI certificados en el nivel 2 de CMMI (certificación de calidad de la UCI en el proceso de desarrollo). Por otra parte, esta metodología se enfoca en técnicas ágiles que permiten un desarrollo guiado por pruebas, una gestión de cambios ágiles y un desarrollo fácil manejado por el modelo.

1.4 Lenguajes y herramientas para el modelado de la solución

En la presente investigación se hace uso del Lenguaje Unificado de Modelado (UML). Este es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software.

1.4.1 Lenguaje Unificado de Modelado (UML) v8.0

Un modelo UML describe lo que supuestamente hará un sistema, pero no dice cómo implementarlo. UML está compuesto por Elementos, Diagramas y Relaciones. Pretende proporcionar a los usuarios un lenguaje de modelado visual expresivo y utilizable para el desarrollo e intercambio de modelos significativos e integrar las mejores prácticas empleadas hasta el momento. Se utiliza con el objetivo de representar los artefactos generados por la metodología AUP en su variante UCI, como por ejemplo los diagramas de clases del diseño, despliegue y prototipo de interfaz de usuario entre otros(15).

1.4.2 Herramienta CASE: Visual Paradigm v5.0

Visual Paradigm for UML es una herramienta que soporta el modelado mediante UML y proporciona asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del Ciclo de Vida de desarrollo de un Software (16). Esta herramienta posibilita la obtención de parte de la documentación y la generación de artefactos propuestos por la metodología de desarrollo, ya que la misma se ajusta al estándar soportado por la herramienta. En el caso de la presente investigación se utiliza la versión 5.0 de la herramienta con la licencia utilizada por la universidad.

1.5 Tecnologías de la implementación

Para realizar la implementación de la propuesta de solución de este trabajo, y darles cumplimiento a los objetivos de esta investigación, teniendo en cuenta las características de la aplicación de escritorio que se desea obtener, es necesario utilizar las tecnologías asociadas al proceso de desarrollo del software, por lo que a continuación se describen:

1.5.1 Lenguajes de programación

En la informática se conoce como lenguaje de programación a un programa destinado a la construcción de otros programas informáticos. Es un lenguaje formal que está diseñado para organizar algoritmos y procesos lógicos que serán luego llevados a cabo por un ordenador o sistema informático, permitiendo controlar así su comportamiento físico, lógico y comunicación con el usuario. Compuesto por símbolos y reglas sintácticas y semánticas, expresadas en forma de instrucciones y relaciones lógicas, mediante las cuales se construye el código fuente de una pieza de software determinado(17). A continuación, se describen los lenguajes utilizados para desarrollar la propuesta de solución:

HTML5

Lenguaje de Formato de Documentos para Hipertexto, HyperTextMarkupLanguage por sus siglas en inglés (HTML) es un lenguaje que se utiliza para el desarrollo de sitios de escritorio. Este provee básicamente tres características fundamentales: estructura, estilo y funcionalidad. HTML5 es considerado el producto de la combinación de HTML, CSS y Javascript. HTML se encarga de la estructura, CSS de mostrar el contenido y la estructura y Javascript de lo demás para así tener un resultado deseado. Es un lenguaje muy simple y general que sirve para definir otros lenguajes que tienen que ver con el formato de los documentos (14).

CCS3

Las Hojas de Estilo en Cascada, Cascading Style Sheets por sus siglas en inglés, (CSS) es un lenguaje que trabaja junto con HTML para proveer estilos visuales a los elementos del documento, como tamaño, color, fondo, bordes, etcétera(18). Usado para definir la presentación de un documento estructurado escrito en HTML, en otras palabras. Esta versión de CSS permite hacer las aplicaciones de escritorio más dinámicas y más elaboradas, con mayor separación entre estilos y contenidos(19).

1.5.2 Framework de desarrollo

Framework, se pudiera traducir como marco de trabajo. Este se emplea en muchos ámbitos del desarrollo de sistemas software, no solo en el ámbito de aplicaciones de escritorio. Podemos encontrar frameworks para el desarrollo de aplicaciones médicas, de visión por computador, para el desarrollo de juegos, y para cualquier ámbito que pueda ocurrírse nos. En general, con el término framework, nos estamos refiriendo a una estructura software compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación. En otras palabras, un framework se puede considerar como una aplicación genérica incompleta y configurable a la que podemos añadirle las últimas piezas para construir una aplicación concreta. Un framework de escritorio, podemos definirlo como un conjunto de componentes (por ejemplo, clases en java y descriptores y archivos de configuración en XML) que componen un diseño reutilizable que facilita y agiliza el desarrollo de sistemas de escritorio(20).

Vue

Vue es un framework progresivo para construir interfaces de usuario. A diferencia de otros frameworks monolíticos, está diseñado desde cero para ser utilizado incrementalmente. La librería central está enfocada solo en la capa de visualización, y es fácil de utilizar e integrar con otras librerías o proyectos existentes. Por otro lado, Vue también es perfectamente capaz de impulsar sofisticadas Single-Page Applications cuando se utiliza en combinación con herramientas modernas y librerías de apoyo(21). En el caso que ocupa la presente investigación vue es utilizado para el desarrollo de la aplicación de escritorio que da solución al problema que se presenta.

Electron

Electron es una biblioteca de código abierto desarrollada por GitHub para construir aplicaciones de escritorio multiplataforma con HTML, CSS y JavaScript. Esta librería logra esto combinando Chromium y Node.js en un mismo ejecutable, así las aplicaciones pueden empaquetarse para Mac, Windows y Linux. Electron comenzó en 2013 como el entorno de trabajo sobre el que se construiría Atom, el editor de texto hackeable de GitHub. Ambos fueron liberados como proyectos de código abierto en la primavera de 2014. Desde entonces se ha convertido en una herramienta popular utilizada por desarrolladores de código abierto, nuevas empresas y compañías establecidas(22). La creación de prototipos de funcionamiento de la aplicación de escritorio de la presente investigación se realiza utilizando Electron.

1.6 Conclusiones del capítulo

El análisis del marco teórico sobre el proceso de investigación para desarrollar la aplicación remota y centralizada que configure el servicio DHCP para GNU/Linux NOVA y el estudio de los principales conceptos referentes al objeto de estudio, permite tener una idea más clara para alcanzar el objetivo e implementar la aplicación de escritorio propuesta.

A partir de la caracterización y comparación de varias aplicaciones para la administración del servicio DHCP, se demuestra la necesidad de desarrollar una aplicación de escritorio para GNU/Linux NOVA que aproveche las mejores funcionalidades de estas y complemente las funcionalidades necesarias que las mismas no presentan para resolver las insuficiencias en las pequeñas y medianas empresas cubanas.

En el desarrollo del capítulo se define la metodología de desarrollo de software que guiará el proceso de desarrollo, las herramientas, tecnologías y lenguajes de programación que se emplean para el desarrollo de la solución de manera que garantice la soberanía tecnológica y la sostenibilidad de la misma.

CAPÍTULO 2: Características y diseño de la herramienta para la administración remota del servicio DHCP.

2.1 Introducción

Este capítulo se centra en los aspectos técnicos de la solución referentes al diseño de la herramienta de administración remota del servicio DHCP. Se elabora un modelo conceptual para una mejor comprensión del contexto de la solución. Contiene la definición de los requisitos a través de la especificación de los mismos y se describen mediante las historias de usuario, a la vez que se realiza su validación. Se realizan los diagramas que corresponden a los artefactos propuestos por la metodología en la fase de análisis y diseño, como son el diagrama de paquetes y el de clases. Se muestra el diseño arquitectónico definido para la propuesta de solución.

A continuación, se expone la propuesta de la herramienta, donde se describen las principales funcionalidades.

2.2 Descripción de la propuesta de solución

Luego de analizar la situación problemática planteada en la investigación se propone como solución el desarrollo de una herramienta para la administración remota del servicio DHCP. El estudio de homólogos realizado en el capítulo 1 arrojó que, aunque existen herramientas para la administración de servicios de este tipo, ninguna de las estudiadas lo hace de forma remota. La herramienta que más se acerca a las necesidades de la problemática planteada en la presente investigación es la herramienta Webmin por lo que se toma la decisión de aprovechar algunas de las funcionalidades propuestas por esta herramienta e incorporar las funcionalidades que permitan:

- Hacer las configuraciones pertinentes para que el administrador pueda crear un servidor DHCP con las entradas que este inserte en la herramienta.
- Permitir la accesibilidad y configuración de forma remota a un servidor DHCP.
- Realizar otro tipo de configuraciones cómo la asignación de IP fijo a una máquina o a un grupo de máquinas, entre otras configuraciones.

Para una mejor descripción del contexto del negocio se realiza un modelo conceptual que permitió a la autora de la investigación comprender el funcionamiento del dominio del negocio informatizado.

2.2.1 Modelo conceptual

Un modelo conceptual contiene la descripción de cómo se relacionan los conceptos que intervienen en el negocio. Además, es usado para representar el problema de forma gráfica. Los conceptos son una representación de elementos del mundo real, no de componentes del software (23). A continuación, en la figura 1 se representa el modelo conceptual correspondiente a la descripción del contexto de negocio de la propuesta de solución:

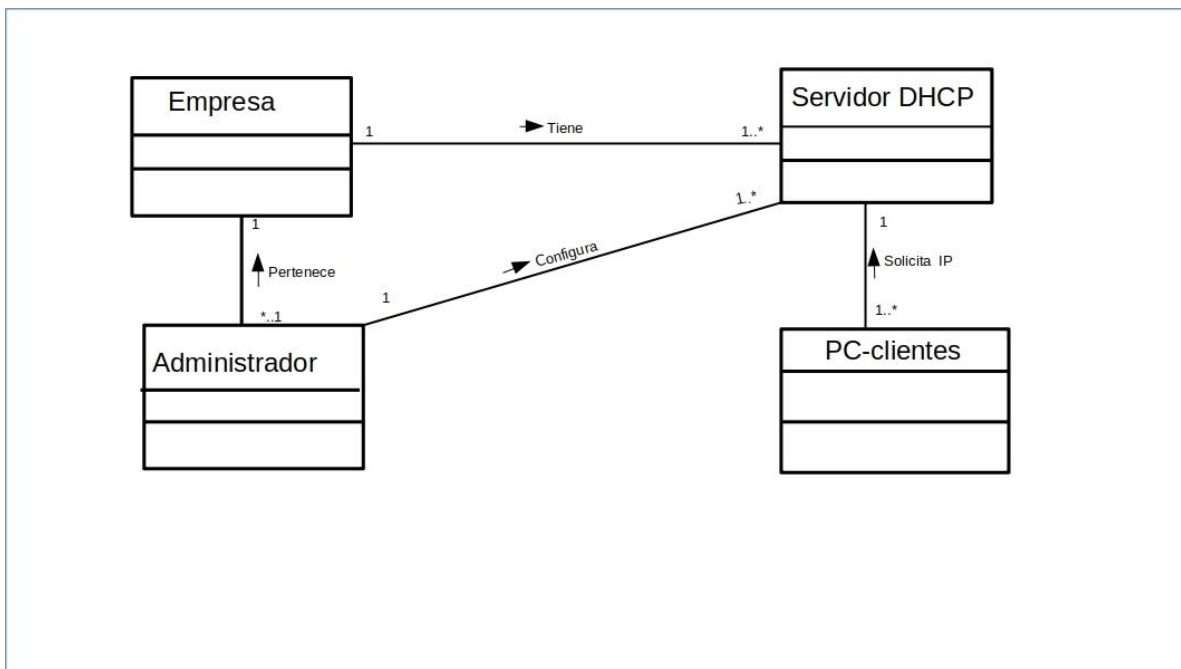


Figura 1. Modelo conceptual

Fuente: elaboración propia

Las definiciones que se muestran a continuación permiten comprender el significado que tienen los conceptos representados en el modelo conceptual del proceso que se desea informatizar.

Conceptos del contexto del negocio

Empresa: es la entidad que por sus funciones posee uno o más servidores DHCP que necesiten ser configurados.

Servidor DHCP: es el servidor DHCP que el administrador configura en la empresa.

Administrador: es el especialista de la empresa que se encarga de la configuración del servidor DHCP.

PC-clientes: son las máquinas que harán peticiones de direcciones IP, además, de otras configuraciones TCP/IP al servidor DHCP configurado por el administrador.

2.2 Requisitos

La disciplina de requisitos es una de las más importantes del proceso de desarrollo de un software; consiste en desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración de los requisitos funcionales y no funcionales del producto (14). A continuación, se presentan las actividades correspondientes en esta disciplina, así como los productos de trabajo elaborados en el desarrollo de la investigación.

2.2.1 Fuentes para la obtención de requisitos

Durante el proceso de desarrollo del software uno de los elementos más importantes del proceso es la obtención de los requisitos, sean funcionales o no funcionales. Este proceso permite conciliar conflictos de intereses entre los involucrados y determinar qué tipo de software se desea desarrollar. Por otra parte, se convierte en una vía efectiva para guiar al desarrollador con el fin de obtener lo deseado por el cliente.

En este proceso intervienen diferentes fuentes que permiten identificar los requisitos que forman parte de la herramienta a desarrollar. Las fuentes consultadas en esta parte de la investigación son:

- Modelo conceptual
- Aplicaciones informáticas para la administración del servicio DHCP
- Especialistas del centro CESOL

Una adecuada comprensión de los requisitos favorece en gran medida al desarrollo de los sistemas informáticos y que estos cumplan con las necesidades y expectativas del cliente. Para realizar este procedimiento existen diversas técnicas que guían al equipo de desarrollo en el proceso de comunicación con el cliente (13). En el siguiente epígrafe se definen las técnicas utilizadas en el levantamiento de requisitos de la propuesta de solución.

2.2.2 Técnicas de identificación de requisitos

En la obtención de requisitos para el desarrollo de la herramienta de administración remota del servicio DHCP, se emplean las técnicas de entrevista y desarrollo de prototipos. Su definición y forma de aplicación se describen a continuación.

Entrevista

Es una técnica muy utilizada para recoger información de otra persona a través de una comunicación interpersonal que se realiza por medio de una conversación estructurada. Es muy importante la forma en que se plantea la conversación y la relación que se establece en la entrevista, ya que deben quedar bien definidos cuales son los contenidos que se desean obtener (24). La técnica se aplicó a través de entrevistas a especialistas de CESOL (anexo 1) para conocer cuáles son las tecnologías compatibles con el entorno de desarrollo, los sistemas con que se integra, así como los requisitos y las características que debe tener una aplicación para ser instalada en la distribución cubana de GNU/Linux Nova.

Desarrollo de prototipos

Los prototipos suelen consistir en versiones reducidas, demos o conjuntos de pantallas (que no son totalmente operativos) de la aplicación pedida. En algunos casos también se utiliza como un medio para formalizar la aceptación previa del cliente de los requisitos del proyecto (24). Estos se desarrollan a partir de la especificación de requisitos. Constituye una representación previa del software. En los prototipos de interfaces de usuario se definen las vistas, los iconos, el estilo de las ventanas y menús, así como las acciones de la interfaz y los objetos, de manera que se muestra al usuario el cual podrá comunicar sus inconformidades(25).Se aplicó la técnica de desarrollo de prototipos para conocer cuáles son las características que debían tener las interfaces del módulo desarrollado. Los prototipos elaborados fueron revisados por miembros del equipo de desarrollo, quienes consideraron que se debía contar con estos prototipos para la representación visual de la propuesta de solución.

La aplicación de las técnicas descritas permite comprender con profundidad el problema de la investigación y facilita la identificación de las funcionalidades y características de la propuesta de solución, definidas en el siguiente epígrafe.

2.2.3 Especificación de requisitos de software

La Especificación de Requisitos de Software (ERS) es una descripción completa del comportamiento y las características del sistema que se desea desarrollar. Sus principales objetivos son auxiliar a los clientes a describir claramente lo que se desea obtener de un determinado software y ayudar a los desarrolladores a entender qué quiere exactamente el cliente. La ERS incluye un conjunto de requisitos funcionales y no funcionales que sirven para describir las necesidades de los clientes y de los usuarios, así como los requisitos que debe tener el sistema para su funcionamiento.

Requisitos funcionales

Los requisitos funcionales definen las acciones que debe realizar e sistema. Son capacidades o condiciones que el sistema debe de cumplir, como debe de comportarse en situaciones específicas(25). A continuación, se presenta el listado de los requisitos funcionales de la propuesta de solución:

Tabla 2. Listado de requisitos funcionales de la propuesta de solución

Fuente: elaboración propia

N.º	Nombre del requisito funcional	Descripción	Complejidad	Prioridad
RF 1	Agregar subred	Permite adicionar una nueva subred al servidor DHCP que se este administrando. Llenando todos los campos de forma obligatoria con los parámetros siguientes: Subred, Máscara, Servidor de dominio, Rango inicial, Rango final, Nombre del dominio, Máscara de subred, Dirección de enrutamiento, Dirección de difusión, Máximo tiempo de arrendamiento, Mínimo tiempo de arendamiento	Alta	Alta
RF 2	Modificar subred	Permite modificar una subred seleccionada en el servidor que se está	Media	Baja

		administrando, editando los parámetros deseados sin dejar ninguno en blanco.		
RF 3	Listar subredes	Permite mostrar un listado de las subredes que existen en el servidor DHCP que se está administrando.	Media	Media
RF 4	Eliminar subredes	Permite eliminar una o varias subredes de forma permanente del servidor DHCP.	Baja	Baja
RF 5	Adicionar una red compartida	Permite añadir una nueva red compartida, luego de seleccionar que declaraciones de subred, host y grupo formarán parte de esta. Teniendo en cuenta los parámetros necesarios para ello. Entre los parámetros más significativos se encuentran: Descripción de la red compartida, Nombre de red	Alta	Alta
RF 6	Modificar red compartida	Permite modificar una red compartida, luego de seleccionar que declaraciones de subred, host y grupo formarán parte de esta. Teniendo en cuenta los parámetros necesarios para ello. Entre los parámetros más significativos se encuentran: Descripción de la red compartida, Nombre de red.	Media	Baja
RF 7	Listar redes compartidas	Permite mostrar un listado con todas las redes compartidas existentes en el servidor DHCP.	Media	Media

RF 8	Eliminar redes compartidas	Permite eliminar una o varias redes compartidas de forma permanente en el servidor DHCP.	Baja	Baja
RF 9	Adicionar un host	Permite adicionar un host, pudiendo ser especial o no. Especificando de una subred determinada alguna dirección arrendada o a cuál dirección de nombre de dominio pertenece mientras este host le pida configuración por medio de DHCP. Llenando todos los campos de forma obligatoria con los parámetros siguientes: Dirección MAC, Dirección IP asignar, Nombre del host	Alta	Alta
RF 10	Modificar un host	Permite modificar un host de la subred seleccionada del servidor DHCP luego de modificar algunos de los siguientes campos: Dirección MAC, Dirección IP asignar, .Nombre del host.	Media	Baja
RF 11	Listar Host	Permite mostrar un listado de los host existentes en el servidor DHCP por subred.	Media	Medio
RF 12	Eliminar Host	Permite eliminar una o varios host previamente creados en las subredes del servidor DHCP.	Baja	Baja
RF 13	Acceder al servidor remoto	Permite acceder de forma remota al servidor DHCP.	Alta	Alta

Tabla 2: Requisitos funcionales

Fuente: elaboración propia

Requisitos no funcionales

Los requisitos no funcionales son cualidades y restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo sobre el proceso de desarrollo. Estas restricciones a menudo se aplican al sistema en su totalidad, normalmente apenas se aplican a las características o servicios individuales del sistema(26).

A continuación, se presenta el listado de los requisitos no funcionales para la herramienta de administración remota DHCP.

Para describir los requisitos funcionales se hace uso de las historias de usuarios. En el epígrafe 2.2.4 se presenta un ejemplo de este artefacto de la ingeniería de software.

Número	Requisito	Descripción
RnF 1	Requisito de interoperabilidad	La herramienta para la administración remota de servidores DHCP debe ser compatible con la distribución cubana de GNU/Linux Nova.
RnF 2	Requisito de seguridad	A la herramienta para la administración remota de servidores DHCP solo podrán acceder usuarios administradores.
RnF 3	Requisito de confiabilidad	La herramienta para la administración remota de servidores DHCP debe ser capaz de recuperarse después de haberse producido un fallo de software o fluido eléctrico.
RnF 4	Requisito de usabilidad	La herramienta para la administración remota de servidores DHCP debe mostrar sus interfaces en idioma español; mostrando en sus componentes una secuencia lógica de pasos a seguir para el usuario.
RnF 5	Requisito de mantenibilidad	La herramienta para la administración remota de servidores DHCP debe estar codificado en un estándar legible y de fácil entendimiento, con el fin de facilitar el análisis para futuras modificaciones, o

		simplemente para dar mantenimiento al sistema.
--	--	--

Fuente: elaboración propia

2.2.4 Descripción de los requisitos de software mediante historias de usuarios.

En la descripción de los requisitos se elaboraron 26 historias de usuarios. En la siguiente tabla se muestra la descripción de la historia de usuario correspondiente al RF1.1: Adicionar subred.

HISTORIA DE USUARIOS	
Número: RF1	Nombre del requisito: Adicionar subred.
Programador: Yanisbel López Pérez	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 4 días
Riesgo en Desarrollo: No aplica	Tiempo Real:
<p>Descripción:</p> <p>El usuario podrá adicionar una nueva subred al servidor seleccionando el botón del menú que se encuentra en la parte superior izquierda, en la cual sale del lateral un conjunto de opciones. Se selecciona la opción de "lista de subred" la cual mostrará un listado de subredes en el caso de que en el servidor existan. En la parte superior derecha de la vista del listado se encuentra un botón "agregar". Luego de hacer clic en el botón se muestra una ventana donde se especificarán los parámetros necesarios para que la subred sea creada. Los parámetros a especificar por el usuario son:</p> <ul style="list-style-type: none"> • Subred • Máscara de red • Servidor de dominio • Nombre del dominio • Dirección inicial para el rango IP 	

- Dirección final para el rango IP
- Máscara de la subred
- Dirección de enrutamiento
- Dirección de difusión
- Mínimo tiempo de arrendamiento
- Máximo tiempo de arrendamiento

Una vez que todos los campos estén llenos correctamente, el usuario presionará el botón “Guardar” para que la subred sea añadida al servidor DHCP. En caso de que existan campos incorrectos la interfaz mostrará un contorno rojo alrededor del campo, para que el usuario proceda a corregirlo. Si el usuario desea cancelar el proceso de creación de una subred, presionará el botón “Limpiar” y los datos antes llenados se eliminarán. En caso de que lo que necesite es regresar a la anterior ventana debido a que el usuario ya no desee adicionar una subred, debe hacer clic en el botón “Atrás” y este regresará a la ventana anterior donde se muestra el listado de las subredes.

Se necesita hacer cambios para crear la subred en el fichero *etc/dhcp/dhcpd.conf*

Observaciones:

Se necesita hacer cambios para crear la subred en el fichero *etc/dhcp/dhcpd.conf*

Prototipo elemental de interfaz gráfica de usuario:

Administración Remota | Sistema Conectado a 127.0.0.1

Agregar una nueva subred

Subred _____ 0/15	Máscara _____ 0/15	Servidor de dominio _____
Rango Inicial _____ 0/15	Rango Final _____ 0/15	Nombre de dominio _____
Máscara de subred _____ 0/15	Dirección de enrutamiento _____ 0/15	Dirección de difusión _____ 0/15
Mínimo rango de arrendamiento 600	Máximo tiempo de arrendamiento 7200	

Atrás Limpiar Guardar

Nova AST 2020

Figura 2. Prototipo de interfaz de usuario correspondiente al RF1.1 Adicionar subred

Fuente: elaboración propia

2.2.5 Validación de requisitos de software

El objetivo de la validación de requisitos es asegurar que todos los requisitos de la herramienta que se establecieron se correspondan con las necesidades del negocio y del cliente, que se han detectado las inconsistencias, omisiones y errores(27). Para validar los requisitos obtenidos se aplican las técnicas de prototipo de interfaz de usuario y el diseño de casos de prueba. Ambas técnicas son descritas a continuación.

Prototipo de interfaz de usuario: el prototipo de interfaz de usuario es una técnica de representación aproximada de la interfaz de usuario de un software que permite a los clientes, usuarios finales y equipo de desarrollo explorar un diseño de interfaz de usuario alcanzable y adecuado que cumpla los requisitos, ayudando a reducir las distancias entre lo que es necesario y lo que es factible. El objetivo principal de la creación de un prototipo de interfaz de usuario es poder probar el diseño de estas interfaces, incluyendo la capacidad de utilización antes de que empiece el desarrollo real. De este modo, se puede garantizar que se está construyendo el sistema correcto, antes de dedicar demasiado tiempo y recursos al desarrollo de la solución informática(28).

Los prototipos de interfaz de usuario realizados de acuerdo a los requisitos obtenidos, le permiten a la autora de la investigación conocer las deficiencias existentes y que se pudieran hacer las correcciones necesarias antes del diseño e implementación de las funcionalidades del sistema. Además, la presentación al cliente permite que el mismo tuviera una noción de la propuesta de solución y emitiera su valoración, las cuales son tomadas en cuenta en el diseño y desarrollo de la aplicación.

Diseño de caso de prueba (DCP): son una serie de acciones que se realizan para determinar una función o funcionalidad particular de su aplicación, su objetivo principal es crear casos de prueba para probar los posibles defectos que pudiera tener el sistema y demostrar que se satisfacen los requisitos obtenidos (29).

La generación de casos de prueba permite especificar los posibles datos de entrada, así como las salidas de los requisitos funcionales de la herramienta de administración remota del servicio DHCP. En total se generaron 11 diseños de casos de pruebas, uno para cada requisito funcional, los cuales fueron revisados por la analista del proyecto y se garantiza la corrección de las deficiencias encontradas y mejorar la calidad de las pruebas de software. A continuación, y debido a su extensión se presenta un fragmento del DCP que

corresponde al RF 1 Adicionar subred, con las variables principales que son modificadas en los escenarios correspondientes.

Caso de prueba del RF 1 Adicionar subred

Descripción general: La herramienta permite al usuario adicionar una subred al servidor DHCP.

Precondiciones: No aplica

Escenario	Descripción	Dirección de red	Máscara de red	Rango de direcciones	Red Compartida	Respuesta del sistema	Flujo central
EC 1.1 Adicionar una subred	El usuario selecciona la opción "Crear subred" en la pantalla principal, luego se activan los criterios de configuración de la subred	10.4.3.0/24	255.255.255.0	10.4.3.10 / 10.4.3.50	Ninguna	Se añade una subred al servidor DHCP	Seleccionar la opción "Crear subred" en la pantalla principal Presionar el botón "Crear subred" en la vista de creación de subred
EC 1.2 Cancelar la creación de una subred	El usuario se arrepiente de la creación de una subred en la opción "Atrás" en el menú superior	NA	NA	NA	NA	Se desactivan los criterios de configuración	Seleccionar la opción "Crear subred" en la pantalla principal Presionar el botón "Atrás"

Tabla 4: Caso de prueba de adicionar una subred

Fuente: elaboración propia

2.3.1 Diseño arquitectónico

El diseño arquitectónico es el proceso creativo que intenta establecer una organización del sistema informático que satisfaga los requisitos funcionales y no funcionales. Su objetivo es representar componentes que interactúen entre ellos y tener asignadas tareas específicas, ser flexibles, extensibles y representar las relaciones de un control entre las partes (30).

En la figura 2 se muestra la arquitectura propuesta para la herramienta de administración remota del servicio DHCP. Se emplea una arquitectura N-Capas, la cual tiene como objetivo estructurar de forma clara la complejidad de una aplicación empresarial basada

en diferentes capas de arquitectura siguiendo el patrón N-Capas. En la misma se definieron 3 capas gracias a la facilidad que ofrece esta arquitectura.

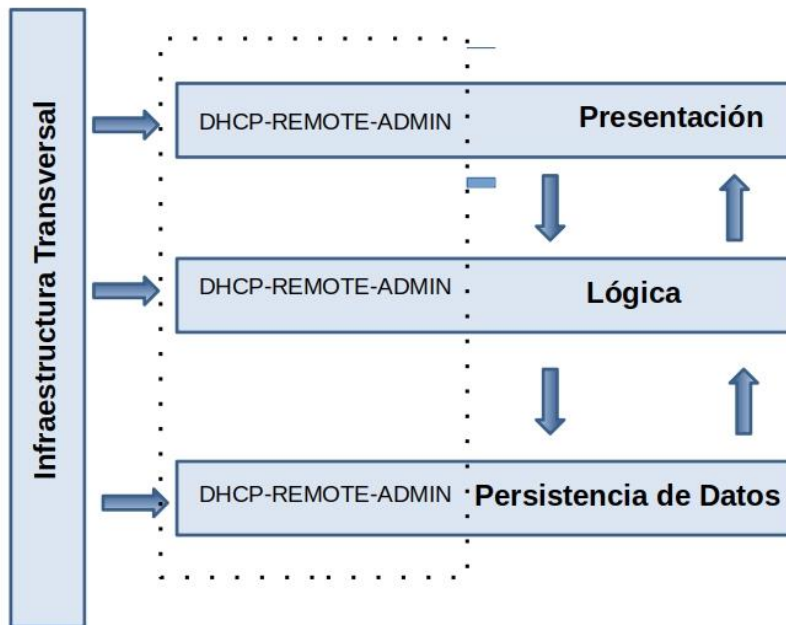


Figura 3.Arquitectura de la herramienta para la administración remota del servicio DHCP

Fuente: elaboración propia

La capa de presentación que es la responsable de la presentación visual de la herramienta, la misma captura los datos al usuario. La capa de lógica del negocio responde por el procesamiento que tiene lugar en la herramienta, interactúa con la anterior para recibir las solicitudes y presentar los resultados. La capa de datos es donde se define la conexión con el servidor y la manipulación de los ficheros de configuración que se modifican desde la aplicación.

2.3.2 Diagrama de clases por paquete

El diagrama de paquetes muestra las agrupaciones lógicas en que está dividido el sistema, así como las dependencias entre dichas agrupaciones. Para el diseño del diagrama de paquetes se toma como referencia la arquitectura anteriormente propuesta, de modo que existe un paquete por cada capa, y dentro de cada uno de estos paquetes otro, llamado DHCP-REMOTE-ADMIN que contiene todas las clases pertinentes a la herramienta.

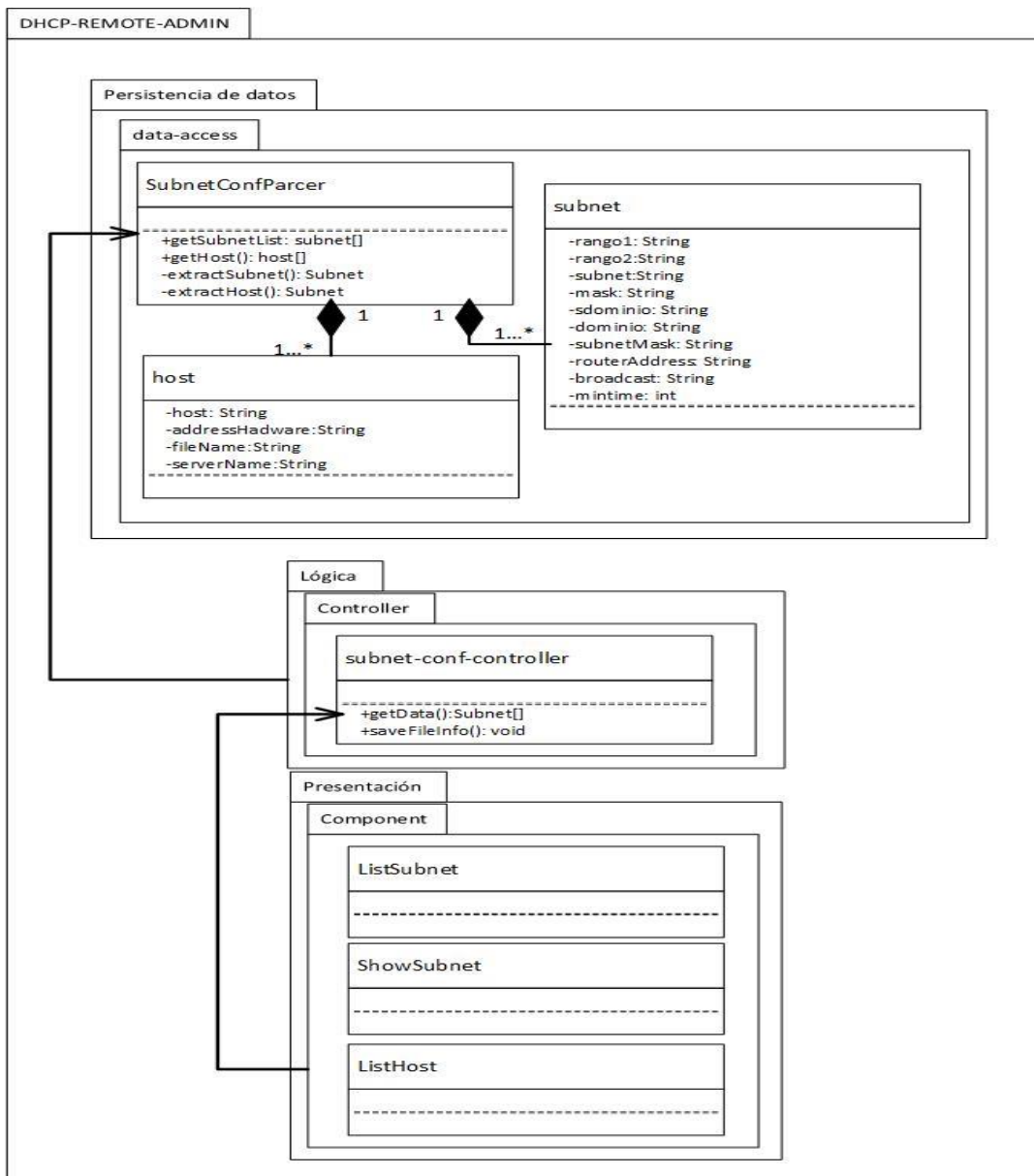


Figura 3. Diagrama de paquetes

Fuente: elaboración propia

2.3 Diseño

En esta disciplina los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa de estos, y una descripción que sea fácil de mantener y ayude a la estructuración del sistema (incluyendo su arquitectura). Se modela el sistema, su forma y su arquitectura para que soporte todos los requisitos, incluyendo los requisitos no funcionales (14). En los epígrafes siguientes se muestran las actividades desarrolladas en esta disciplina, así como los productos de trabajo elaborados en el proceso de desarrollo de la investigación.

2.3.1 Diseño de clases

En el diseño de clases de la herramienta de Administración Remota del Servicio DHCP, se tuvo en cuenta la aplicación de los patrones de diseño y patrones de asignación de responsabilidad. A continuación, se detallan los resultados obtenidos.

Diagrama de clases del diseño

El diagrama de clases del diseño describe gráficamente las especificaciones de las clases de software y de las interfaces: en él se muestran un conjunto de clases, interfaces, métodos, dependencias y relaciones (13). En la figura se presenta el diagrama de clases del diseño que corresponde al diseño de la herramienta de administración remota del servicio DHCP.

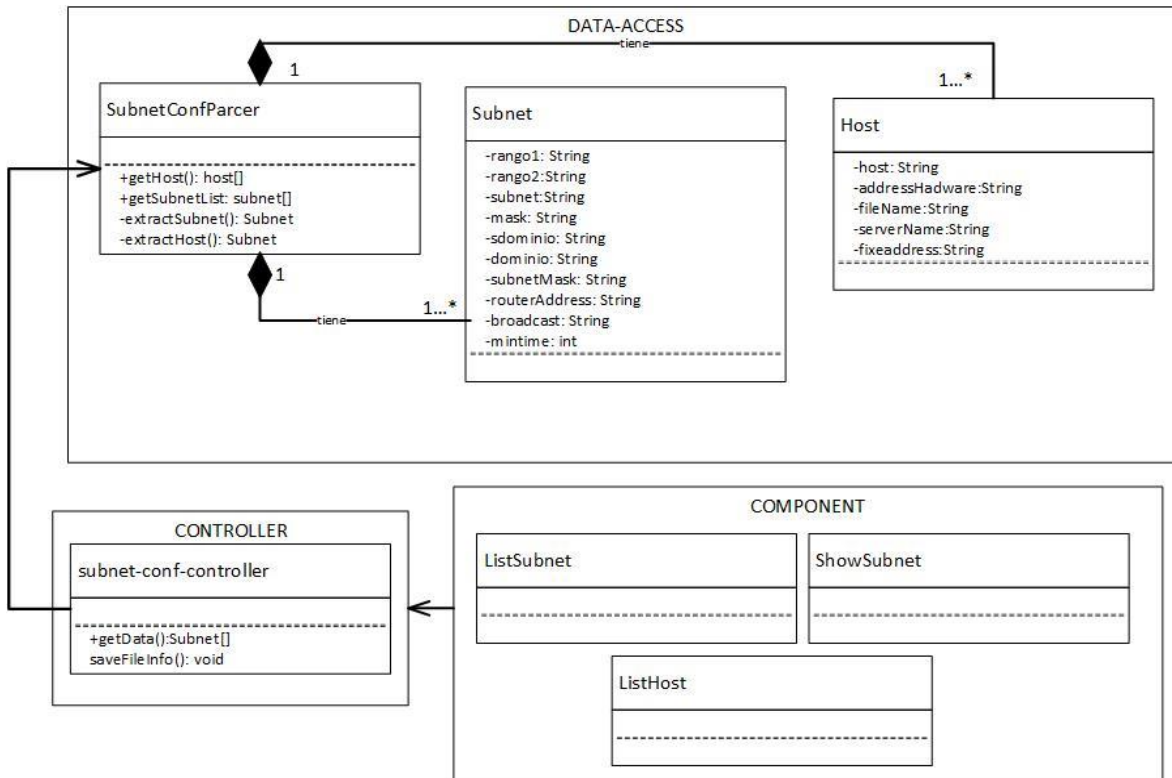


Figura 3. Diagrama de clases

Fuente: elaboración propia

El diagrama de clases anterior está formado por las clases

- Clases interfaces: definen todas las abstracciones necesarias para la interacción entre los usuarios y la aplicación. En este caso competen las clases *ListSubnet*, *ShowSubnet* y *ListHost*.
- Clase controladora: permite encapsular las funcionalidades necesarias para interactuar con las clases interfaces y las clases de acceso a datos. Este caso está presente en la clase *SubnetConfParcer*.
- Clases persistente: representa el almacenamiento de los datos que serán persistentes en el tiempo, aun cuando haya finalizado la ejecución de la aplicación. Está formada por las clases *SubnetConfParcer*.

Patrones de diseño de software

Los patrones de diseño son una técnica para resolver problemas comunes en el desarrollo del software. Los patrones son soluciones comunes a problemas similares

Patrones GRASP:

Los patrones GRASP describen responsabilidades que están relacionadas con las obligaciones de un objeto en cuanto a su comportamiento (33). Entre estos patrones se encuentran el experto, el creador, el controlador, bajo acoplamiento, alta cohesión. A continuación, se describen los patrones que fueron empleados en el diseño de la propuesta de solución.

Experto: es empleado para asignar una responsabilidad al experto en información, la clase que cuenta con la información necesaria para cumplir con una determinada responsabilidad. Con la aplicación del patrón experto se disminuye el acoplamiento y aumenta la cohesión del diseño. En la clase Subnet y Host se encuentran presentes.

Creador: ayuda a identificar qué clase debe ser responsable de la creación o instanciación de nuevos objetos o clases. Permite la visibilidad entre la clase creada y la clase creadora. Este patrón se evidencia en la clase subnet, en el método *created ()*.

Alta cohesión: a través de este patrón se asegura la necesidad de mantener el sistema con un bajo nivel de complejidad, lo que quiere decir que el sistema debe realizar solo las operaciones necesarias para cumplir una tarea dentro de un área funcional.

Bajo Acoplamiento: el acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, con que las conoce y con que recurre a ellas. Una clase con bajo o débil acoplamiento no depende de muchas otras. El bajo acoplamiento soporta el diseño de clases más independientes y reutilizables, lo cual reduce el impacto de los cambios y acrecienta la oportunidad de una mayor productividad. El uso de este patrón se demuestra mediante indicadores como el promedio de relaciones entre clases, que en este caso es de 1.7 relaciones por clases, lo que demuestra el uso del patrón.

Conclusiones del capítulo

En este capítulo se abordaron los aspectos técnicos del análisis y diseño de la aplicación que permite el desarrollo exitoso de la propuesta de solución. Se elaboró un modelo conceptual lo que permite una mejor comprensión del contexto de la solución.

Fueron definidos, especificados y validados los requisitos que permite el desarrollo de la herramienta de Administración Remota. Se realizaron los diagramas de paquetes y de clases que permite guiar el desarrollo de la aplicación. Fue definido el diseño arquitectónico que permite identificar las restricciones existentes a partir del diseño y se

identificaron los patrones de diseño que permite ajustar la solución a las buenas prácticas de desarrollo.

Una vez concluido este capítulo están sentadas las bases para el desarrollo y evaluación de la propuesta de solución.

CAPÍTULO 3: Implementación y evaluación de la herramienta para la Administración Remota del Servicio DHCP.

Introducción

3.1 Implementación

El principal objetivo de la disciplina de implementación es construir el sistema informático, a partir de los resultados del análisis y diseño (14). Una implementación de un software en desarrollo corresponde al proceso de convertir una especificación del sistema en un sistema ejecutable. Siempre incluyendo procesos de diseño y programación de software, aunque también puede involucrar la corrección en la especificación del software, si se utiliza un enfoque incremental de desarrollo. También son una manifestación en el mundo real de las funciones de procesamiento y estructuras de la información.

3.1.1 Estándares de codificación

El estándar de codificación es un conjunto de convenciones establecidas para la escritura de código. Estos estándares varían dependiendo del lenguaje de programación elegido y algunos son más extensos que otros. Permiten incrementar la calidad del código y que la ocurrencia de errores sea menor (38).

Estos reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores. Los estándares de codificación no descubren problemas existentes, sino evitan que los errores ocurran, lo que permite obtener un código de alta calidad. La aplicación de estándares de codificación posibilita que el software que se obtiene sea fácil de comprender y de mantener en el tiempo. Para el desarrollo de la herramienta de Administración Remota del Servicio DHCP se definió un estándar de codificación el cual a continuación se relacionan algunos aspectos:

Identación: La unidad de Identación de bloques de sentencias es de 4 espacios.

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

Declaraciones: Se debe declarar cada variable en una línea distinta.

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.

Se debe usar siempre una línea en blanco en las siguientes circunstancias:

- Entre métodos.
- Entre las variables locales de un método y su primera sentencia.
- Antes de un comentario de bloque o de un comentario de una línea.

Estándares de nomenclatura

- Se utiliza Upper Camel Case (primera letra de cada palabra en mayúscula) para los nombres de las clases.
- Se utiliza Lower Camel Case (letra inicial de la primera palabra en minúscula y el resto de las palabras comienzan con mayúscula) para los nombres de los métodos, las variables y las funciones.

3.2 Diagrama de despliegue

Los diagramas de despliegue muestran las relaciones físicas de los distintos nodos que componen un sistema y el reparto de los componentes sobre dichos nodos (37). Es el diagrama que despliega físicamente los componentes en los procesadores, muestra como los componentes de software se distribuyen a través de hardware.

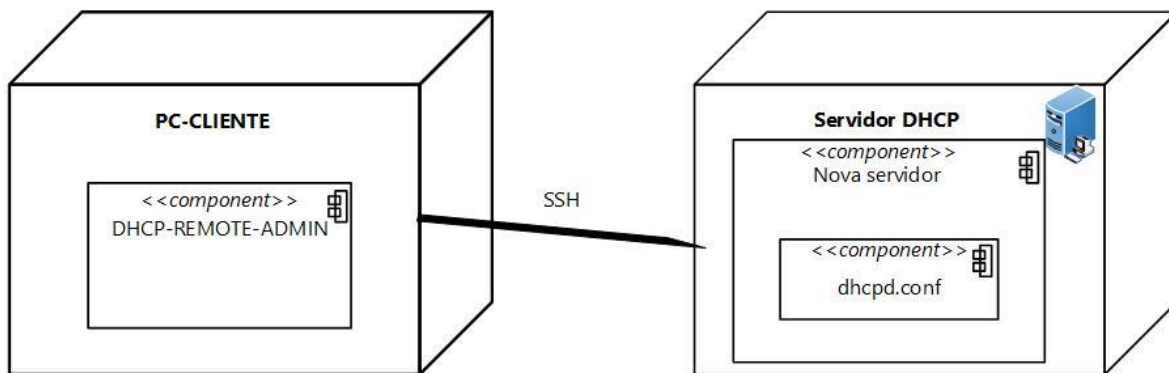


Figura 3. Diagrama de despliegue

Fuente: elaboración propia

Descripción de los nodos

PC-clientes: Contiene la herramienta informática para la Administración del Servicio DHCP (DHCP-REMOTE-ADMIN), para la comunicación con el servidor utiliza el protocolo SSH.

Servidor DHCP: Es el servidor donde se realiza las configuraciones pertinentes del servicio DHCP, en el fichero de configuración dhcpd.conf.

3.3 Pruebas de software

Las pruebas del Software pueden usarse para demostrar la presencia de errores, nunca su ausencia (34). Las pruebas de software son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto (13). Cuando se realiza el proceso de pruebas el programa es ejecutado con datos de ejemplo cumpliendo con los tipos de datos de cada campo. Es necesario verificar los resultados de la prueba que se ejecuta para buscar errores, anomalías o información de atributos no funcionales del programa. El proceso de prueba tiene dos metas distintas: Demostrar al desarrollador y al cliente que el software cumple con los requerimientos y encontrar situaciones donde el comportamiento del software sea incorrecto, indeseable o no esté de acuerdo con su especificación (26).

Luego del desarrollo de la propuesta de solución, se hace necesario validar y verificar el sistema implementado a través del diseño de la estrategia de prueba definida, que permite comprobar el cumplimiento de las especificaciones del diseño y de codificación e identificar los posibles errores cometidos. En el desarrollo del software se plantea un conjunto de pruebas que permiten constatar que el software desarrollado satisface las especificaciones del cliente.

3.3.1 Pruebas internas

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las internas como las intermedias, así como las versiones finales a ser liberadas (14). Durante esta disciplina se aplicaron pruebas unitarias, funcionales y de integración; descritas a continuación:

Pruebas Unitarias

Las pruebas unitarias se realizan a través del método de caja blanca y la técnica de camino básico. Para aplicar la prueba de camino básico, se debe realizar un análisis de la complejidad ciclomática de cada procedimiento que componen las clases del sistema (14).

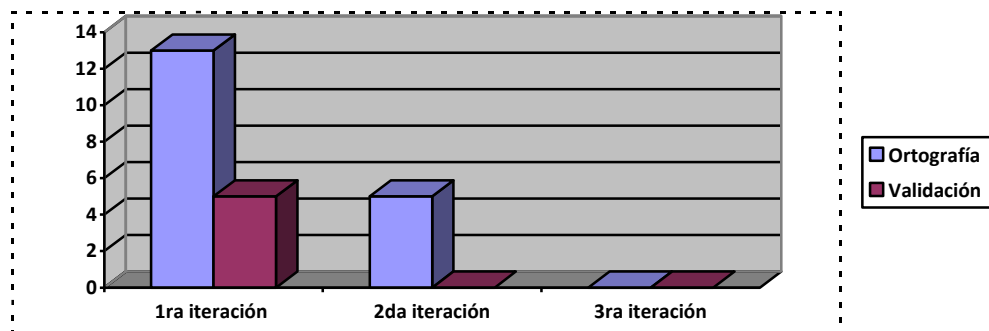
El método de caja blanca se centra en los detalles procedimentales del software, por lo que su diseño está ligado al código fuente (35). Garantiza que el ingeniero del software pueda obtener casos de pruebas que se ejerciten por lo menos una vez todos los caminos independientes de cada método y se ejecuten todos los bucles en sus límites

operacionales. Permite que disminuya el número de errores existentes en los sistemas y por ende una mayor calidad y confiabilidad.

La técnica empleada en este tipo de prueba fue el camino básico, el cual consiste en diseñar un caso de prueba por cada camino independiente del programa. Con esta técnica se garantiza que se prueben todos los caminos de ejecución del programa, al menos una vez. La técnica de prueba de caja blanca permite obtener una medida de la complejidad lógica de un diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución (26).

Pruebas funcionales

En las pruebas funcionales se tuvo en cuenta el método de caja negra y la técnica de partición equivalente. Para la aplicación de la técnica de partición equivalente se retomaron los casos de pruebas diseñados en el epígrafe 2.2.5 en el que se definen las clases de equivalencias válidas y no válidas para cada campo de entrada del sistema. En la figura 4 se presenta un gráfico con los resultados de las pruebas funcionales.



Las pruebas funcionales se realizaron en 3 iteraciones; en la primera se identificaron 18 no conformidades de ellas 13 fueron de errores de ortografía y 5 de validación. Durante la segunda iteración se detectaron 2 errores de ortografía y ninguna de validación y en la tercera iteración no se identificaron no conformidades.

Pruebas de regresión

Las pruebas de regresión se deben llevar a cabo cada vez que se hace un cambio en el sistema, ya sea para corregir un error como para realizar una mejora. No es suficiente probar solo los componentes modificados o añadidos, o las funciones que en ellos se realiza, sino que también es necesario controlar que las modificaciones no produzcan efectos negativos sobre el mismo u otros componentes.(26)

Este tipo de pruebas implica la repetición de las pruebas que ya se han realizado previamente, para asegurar que no se introduzcan errores que puedan comprometer el funcionamiento de otros componentes que no han sido modificados y confirmar que el sistema funciona correctamente una vez realizados los cambios.

Las pruebas de regresión pueden incluir (13):

- La repetición de las pruebas que se han realizado anteriormente y están directamente relacionadas con la parte del sistema modificada.
- La revisión de los procedimientos manuales preparados antes del cambio, para asegurar que permanecen correctamente.
- La obtención impresa del diccionario de datos de forma que se compruebe que los elementos de datos que han sufrido algún cambio son correctos.

Las pruebas de regresión se emplean en el desarrollo la herramienta de la Administración Remota del Servicio DHCP en cada una de las iteraciones realizadas como parte de la estrategia de pruebas de caja negra, verificando que quedaban resueltas las inconformidades detectadas y que su resolución no afectaba el resto de las funcionalidades, así como luego de realizadas las pruebas de caja blanca.

Conclusiones del capítulo

En el presente capítulo se logra la informatización de la herramienta de administración remota del servicio DHCP para el sistema operativo GNU/Linux NOVA, para ello se tiene en cuenta 13 requisitos funcionales y 6 no funcionales a partir de las necesidades del cliente, las restricciones del diseño y los estándares de codificación definidos por el desarrollador. La aplicación de las pruebas unitarias permite comprobar que el flujo del trabajo de las funcionalidades implementadas es correcta, pues cada sentencia se ejecuta al menos una vez. La ejecución de las pruebas funcionales posibilita las deficiencias de la propuesta de solución y garantizar el cumplimiento de los requisitos definidos. La prueba de regresión permite verificar que las no conformidades encontradas en las demás pruebas han sido corregidas satisfactoriamente. Al concluir el período de pruebas se obtuvo una aplicación que cumpla de forma correcta con la totalidad de las funcionalidades esperadas por el cliente.

Conclusiones Generales

El presente trabajo de diploma concluye con el desarrollo de la herramienta de administración remota del servicio DHCP para las pequeñas y medianas empresas cubanas. Una vez finalizado exitosamente el mismo se puede arribar a las siguientes conclusiones:

- El análisis del marco teórico sobre la administración remota del servicio DHCP, así como el estudio de aplicaciones informáticas para la administración de este servicio, demuestra la necesidad de desarrollar una aplicación que contribuya a la administración remota de la asignación dinámica de direcciones IP.
- A partir de los 13 requisitos funcionales y los 5 no funcionales, definidos según las necesidades del cliente, se diseña una herramienta para la administración remota del servicio DHCP. El uso de los patrones de diseño GRASP y el patrón arquitectónico 3-capas permitió brindar una mayor calidad al software desarrollado.
- Cumpliendo las exigencias del cliente se desarrolló una herramienta de administración remota para el servicio DHCP.
- La utilización del diseño de la estrategia de pruebas garantizó la identificación temprana de las deficiencias en la herramienta que se desarrolla; corrigiéndose las mismas logrando un producto más seguro y funcional.

Anexos

Anexos 1: Entrevista

Estimado Especialistas: Se necesita de su cooperación para una tesis de pregrado.

Para ello, será de gran ayuda que respondiera las siguientes preguntas:

- ¿Qué protocolo usualmente se usan en los OACE para la asignación dinámica de los host?
- ¿El proceso de asignación dinámica de host en las empresas se hace de forma manual o por alguna aplicación que corresponda a ello?
- ¿Es común encontrar en los OACE más de un servidor DHCP?

Referencias Bibliográficas

1. Historia de internet. [en ligne]. [Consulté le 20 février 2020]. Disponible à l'adresse : <https://www.fib.upc.edu/retro-informatica/historia/internet.html>
2. RedIRIS - A, B, C de Internet. [en ligne]. [Consulté le 18 février 2020]. Disponible à l'adresse : <https://www.rediris.es/difusion/publicaciones/boletin/28/enfoque1.html>
3. Red Hat Enterprise Linux 3. [en ligne]. [Consulté le 18 février 2020]. Disponible à l'adresse : <http://web.mit.edu/rhel-doc/3/rhel-sag-es-3/index.html>
4. MIRANDA, Regalado, MARÍA, Ester et REGALADO MIRANDA, Elsa. Internet: la red de redes en Cuba. Educación Médica Superior. juin 1997. Vol. 11, n° 1, pp. 39-46.
5. Repositorio Digital: Metodología para la migración a software libre de las Universidades del Ministerio de Educación Superior (MES). [en ligne]. [Consulté le 18 février 2020]. Disponible à l'adresse : https://repositorio.uci.cu/jspui/handle/ident/TD_1273_08
6. PÉREZ VILLAZÓN, Yasiel. Solución para la gestión del almacenamiento de datos en las instituciones cubanas. [en ligne]. 2018. [Consulté le 20 février 2020]. Disponible à l'adresse : <http://repositorio.uci.cu/jspui/handle/123456789/7912>
7. PIERRA FUENTES, Allan et RODRÍGUEZ FIGUEREDO, Héctor. Conceptualización y Reestructuración Estratégica de la Distribución Cubana de GNU/Linux « Nova ». [en ligne]. 30 octobre 2013. [Consulté le 20 février 2020]. Disponible à l'adresse : <http://repositorio.uci.cu/jspui/handle/ident/8004>
8. VITIER, Abel García, VILLAZON, Yoandy Pérez, ORAMAS, (Angel Goñi), HERNÁNDEZ, Amaury Viera, BLANCO, (Yurenia Hernández) et LLANES, (Eduardo Alejandro). Buenas prácticas para la migración a código abierto. 2014.
9. Que Es Un Protocolo de Internet | Protocolos de internet | Protocolo de Control de Transmisión. [en ligne]. [Consulté le 20 février 2020]. Disponible à l'adresse : <https://www.scribd.com/doc/22660807/Que-Es-Un-Protocolo-de-Internet>
10. Webmin. [en ligne]. [Consulté le 20 février 2020]. Disponible à l'adresse : <http://www.webmin.com/>
11. Características - Zentyal Server. [en ligne]. [Consulté le 20 février 2020]. Disponible à l'adresse : <https://zentyal.com/es/caracteristicas/>
12. openSUSE Wiki. [en ligne]. [Consulté le 20 février 2020]. Disponible à l'adresse : https://es.opensuse.org/Bienvenidos_a_openSUSE.org
13. PRESSMAN, Roger. Ingeniería de software. Un enfoque práctico. 2010.
14. RODRÍGUEZ, Tamara. Metodología de desarrollo para la actividad productiva de la UCI. 2015.
15. RUMBAUGH, J, JACOBSON, I et BOOCH, G. El lenguaje unificado de modelado. Manual de referencia. Segunda edición. 2007.
16. Guión Visual Paradigm for UML. studylib.es [en ligne]. [Consulté le 18 février 2020]. Disponible à l'adresse : <https://studylib.es/doc/7046646/guión-visual-paradigm-for-uml> Biblioteca en línea. Materiales de aprendizaje gratuitos.

17. Lenguaje de Programación: Concepto, Tipos y Ejemplos. [en ligne]. [Consulté le 20 février 2020]. Disponible à l'adresse : <https://concepto.de/lenguaje-de-programacion/>
18. El gran libro de HTML5, CSS3 y Javascript. primera. 2012.
19. ETNASSOFT. OpenLibra | Guía Completa de CSS3. OpenLibra [en ligne]. [Consulté le 18 février 2020]. Disponible à l'adresse : <https://openlibra.com/es/book/guia-completa-de-css3CSS> u hojas de estilo en cascada (en inglés Cascading Style Sheets) es un lenguaje usado para definir la presentación de un documento estructurado escrito…
20. GUTIÉRREZ, Javier J. ¿Qué es un framework web? [en ligne]. [Consulté le 20 février 2020]. Disponible à l'adresse : http://www.lsi.us.es/~javierj/investigacion_ficheros/Framework.pdf
21. Introducción — Vue.js. [en ligne]. [Consulté le 18 mars 2020]. Disponible à l'adresse : <https://es.vuejs.org/v2/guide/index.html>Vue.js - The Progressive JavaScript Framework
22. About Electron | Electron. [en ligne]. [Consulté le 20 février 2020]. Disponible à l'adresse : </docs/tutorial/about>
23. DALVIK, J. El conspirador. Obtenido de Qué es y para que sirve un modelo conceptual. [en ligne]. Diciembre del 2018. Disponible à l'adresse : <https://www.elconspirador.com/2013/12/21/que-es-y-para-que-sirve-un-modelo-conceptual/>
24. Obtención de Requerimientos. Técnicas y Estrategia. SG Buzz [en ligne]. [Consulté le 20 février 2020]. Disponible à l'adresse : <https://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategia>
25. PEÑA GROSS, Maylenis et MENO MASÍAS, Raúl Alex. Módulo de réportes para el subsistema Patrimonio de las Fuerzas Armadas Revolucionarias. La Habana : Universidad de las Ciencias Informáticas, 2018.
26. SOMERVILLE, I. Ingenieria de software. United Kingdom: Pearson Education Limited.[sans date].
27. MAREISZACK, Marcelo, CARDENAS, Marina, CASTRO, Claudia et RAMIRO, Pérez. Validación e Requerimientos a través de modelos conceptuales. 2012.
28. Artefacto: Prototipo de interfaz de usuario. [en ligne]. [Consulté le 20 février 2020]. Disponible à l'adresse : https://cgrw01.cgr.go.cr/rup/RUP.es/LargeProjects/core.base_rup/workproducts/rup_user_interface_prototype_7237E5AA.html
29. Casos de prueba, ejemplos y mejores prácticas. Best Read on ITpedia [en ligne]. 1 juin 2018. [Consulté le 20 février 2020]. Disponible à l'adresse : <https://es.itpedia.nl/2018/06/01/testcases-voorbeelden-en-best-practices/>
30. ALMEIRA, S.A et PÉREZ, V. Arquitectura de software.Estilos y patrones. . 2016.
31. Zorrilla Castro, César. Guía de arquitectura N-Capas orientada al Dominio con .NET 4.0. .2010.ISBN 978-84-936696-3-8.
32. Larman, Graig. UML y Patrones. Introducción al análisis y diseño orientados a objetos.

33. SANTIAGO Felipe, Gonzalo GONZÁLEZ VERA, Yosel Lázaro. Módulo para la administración y migración hacia samba4 del servicio directorio activo. Junio 2016
34. Lomprey,G y Hernández, S., 2008. La importancia de la calidad en el desarrollo de los productos de software. Recuperado el [en línea], disponible en: [http://fit.um.edu.mx/CI3/publicaciones/Technical Report COMP-018-2008.pdf](http://fit.um.edu.mx/CI3/publicaciones/Technical%20Report%20COMP-018-2008.pdf).
35. ALVARES, N.S y CLARA, N.L., 2019. Mutation test, control of variations in the source code. [en línea], disponible en: <https://pdfs.semanticscholar.org/0faa/ecdf09188b18d9101e8dc6ec30105778ce6c.pdf>.
36. LARROSA, D., FERNÁNDEZ, P. y DELGADO, M., 2018. GeCaP: Generador de casos de pruebas unitarias a partir del código fuente en lenguaje Java. Polibits, vol. 57, pp. 67-73.
37. [en línea], disponible en: <http://virtual.usalesiana.edu.bo/web/conte/practica/22012/2132.pdf>
38. DOMÍNGUEZ, E.D. DOMÍNGUEZ, V.B. RAMÍREZ, A. MORENO, J.A., 06 julio-diciembre-2015. Reglas de calidad para la codificación estandarizada en lenguaje C: una propuesta para la enseñanza a nivel superior.