

UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

FACULTAD 1



**Herramienta informática para la administración remota de la
configuración de red en Nova Servidores**

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

Autora:

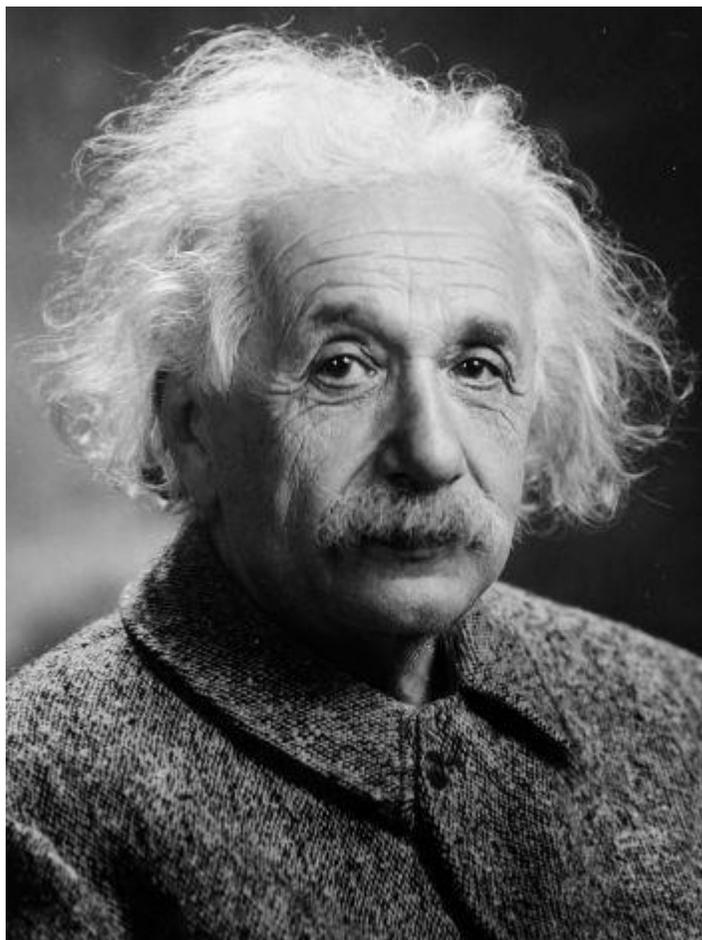
Betsy Alfonso Colás

Tutores:

Ing. Ivaniel Díaz Romeu

Ing. Lexys Manuel Díaz Alonso

LA HABANA, DICIEMBRE 2021



**“Hay una fuerza motriz más poderosa que el vapor, la electricidad y la energía automática:
la voluntad”**

Albert Einstein

AGRADECIMIENTOS

Juaniet Díaz Romeu: Lo distes todo, no tengo una sola queja, me enseñaste que se puede dar más, simplemente te quiero.

Odalys Colas Pérez.

Alfredo Ernesto Díaz Flaquet.

Bertha Pérez Mesa.

Miguel Ángel Colas Bejerano.

Gricerio Alfonso Abreu.

Lexys Manuel Díaz Alonso: Gracias por la ayuda brindada.

Yailen Esmeralda Salgado Reyes alias la Flacas gracias por la tecnología.

DEDICATORIA

A memoria de Lázaro Mazorra, donde quiera que estés sé que estarás orgulloso de tu "LOCA".

Odalys Colas Pérez el amor de mi vida mi madre, mi amiga, mi universo la persona que cree en mi ni cuando yo no lo hago. Alfredo Ernesto Díaz Flaquet el que me hace reír dando chuchos innecesarios, mi amigo, mi padre en estos años, el que me iba a buscar a la escuela cuando me dejaban castigada por mala disciplina, el que me dice (pa lante) y me llama todo el tiempo. Bertha Pérez Mesa mi abuela, mi amiga, la mujer más importante de mi vida porque nunca ha querido cambiarme simplemente me ama y sin condición. Miguel Ángel Colas Bejerano mi viejito bello mi otro padre que se me ponía a enseñarme matemáticas todos los días con paciencia y amor. A mi Padre Glicerio Alfonso Abreu (papucho) por su apoyo. Por supuesto a mi familia en general tíos, primos que formaron parte de mi formación. Gracias por todo.

Y por último en la lista, pero no menos importante me agradezco a mí por todas las noches sin dormir, por querer ser una mejor versión cada día, por tener presente quien soy y a donde voy, por el tiempo que estuve lejos de personas queridas y seguir adelante, por no caer en tentaciones que corromperían mi alma, por tener los ovarios de no decepcionar a mi tutor: Ivanié Díaz Romeu, y a Dios por hacerme tal cual soy. MRPV.

DECLARACIÓN DE AUTORÍA

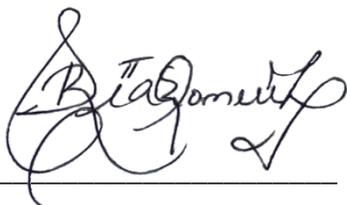
Declaro por este medio que yo **Betsy Alfonso Colás**, con carné de identidad 95052009057, soy el autor principal del trabajo titulado **“Herramienta informática para la administración remota de la configuración de red en Nova Servidores”** y autorizo a la Universidad de las Ciencias Informáticas a hacer uso de la misma en su beneficio, así como los derechos patrimoniales con carácter exclusivo.

Para que así conste firman la presente a los 7 días del mes diciembre del 2021.



Betsy Alfonso Colás

Autora



Ing. Ivaniét Díaz Romeu Prof. Instructor

Tutor



Ing. Lexys Manuel Días Alonso

Tutor

RESUMEN

Una red informática, es un conjunto de equipos conectados por medio de cables o señales electromagnéticas para compartir información o recursos. Su configuración permite que sus componentes activos puedan comunicarse entre sí. Uno de estos componentes son los sistemas operativos existentes en las computadoras que hacen función de servidor atendiendo las peticiones de los dispositivos clientes. En diferentes Organismos de la Administración Central del Estado (OACE) de Cuba, se emplea la distribución GNU/Linux Nova Servidores (desarrollada por el Centro de Software Libre (CESOL) de la Universidad de las Ciencias Informáticas), en su infraestructura de redes de comunicaciones, garantizando la soberanía tecnológica en el país. En una encuesta aplicada a 2 especialistas de CESOL, relacionados con la migración a software libre en diferentes OACEs, se evidenció que en el proceso de configuración de la red en Nova Servidores se presentaron las siguientes deficiencias: pérdida de tiempo como consecuencia de que esta configuración se realiza de forma manual directamente en la consola del sistema operativo, e incurrancia en errores humanos debido a que los administradores de red necesitan dominar los comandos necesarios para realizar las configuraciones, y los ficheros que se modifican. La investigación tiene como objetivo “Desarrollar una herramienta informática que permita la administración remota de la configuración de red en Nova Servidores”. En su diseño se empleó la metodología Variación de AUP - UCI y en la implementación se utilizaron tecnologías de software libre. La herramienta obtenida permite mejorar el proceso de administración de la configuración de red en Nova Servidores.

Palabras clave: administración remota, configuración de red, red informática, Nova Servidores

ÍNDICE

RESUMEN	V
ÍNDICE	VI
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	IX
INTRODUCCIÓN	1
CAPÍTULO 1. Fundamentación teórica sobre el proceso de configuración de red en sistemas operativos GNU/Linux	6
1.1 Definición de conceptos	6
1.2 Descripción del proceso de configuración de red en sistemas operativos GNU/Linux	7
1.2.1 Modos de configuración de red	7
1.2.2 Parámetros de configuración de la red.....	9
1.2.3 Administración de la configuración de red en distribuciones GNU/Linux	10
1.3 Herramientas informáticas para la administración de la configuración de red sistemas operativos GNU/Linux	14
1.3.1 Análisis comparativo de herramientas informáticas para la administración de la configuración de red	15
1.4 Caracterización de Nova Servidores.....	16
1.5 Metodología de desarrollo de software	17
1.5.1 Variación de AUP para la UCI	17
1.6 Lenguajes y herramientas para el modelado de la solución	18
1.6.1 Herramienta para el modelado de la solución	18
1.7 Tecnologías para la implementación.....	19
1.7.1 Lenguajes de programación	19
1.7.2 Entorno de desarrollo integrado	20
1.8 Conclusiones del capítulo	21
CAPÍTULO 2. Análisis y diseño de una herramienta informática para la administración remota de la configuración de red en Nova Servidores	22
2.1 Descripción del contexto de la propuesta de solución desarrollada	22
2.1.1 Modelo conceptual.....	22
2.1.2 Diccionario de datos	23
2.2 Requisitos.....	24
2.2.1 Fuentes para la obtención de requisitos.....	24
2.2.2 Técnicas de identificación de requisitos	25
2.2.3 Especificación de requisitos de software.....	26

2.2.4 Descripción de requisitos de software mediante Historias de Usuario: 28

2.2.5 Validación de requisitos de software 31

2.3 Análisis y diseño..... 34

 2.3.1 Diseño de clases 34

 2.3.2 Diseño arquitectónico 36

 2.3.3 Modelado de datos 37

2.4 Conclusiones parciales 37

CAPITULO 3: Implementación y evaluación de la herramienta informática para la administración remota de la configuración de red en Nova Servidores 39

3.1 Implementación 39

 3.1.1 Estándares de codificación 39

 3.1.2 Diagrama de componentes..... 40

 3.1.3 Diagrama de despliegue..... 41

 3.1.4 Interfaz gráfica de usuario 42

3.2 Pruebas de software 43

 3.2.2 Métodos de prueba..... 43

 3.2.3 Técnicas de prueba 44

3.3 Aplicación de las pruebas de software..... 44

 3.3.1 Pruebas Internas 44

3.4 Conclusiones parciales 48

CONCLUSIONES GENERALES..... 49

RECOMENDACIONES 50

REFERENCIAS BIBLIOGRÁFICAS..... 51

ANEXOS 54

ÍNDICE DE FIGURAS

Figura 1. Descripción del contexto del negocio: Modelo conceptual22

Figura 2. Seleccionar la opción: Configuración de la interfaz de red 30

Figura 3. Configuración de la red estática..... 31

Figura 4. Configuración de red dinámica31

Figura 5. Diagrama de clases del diseño de la propuesta de solución 34

Figura 6. Diseño arquitectónico de la propuesta de solución basado en capas36

Figura 7. Ejemplo de la aplicación de los estándares de codificación 40

Figura 8. Diagrama de componente de la propuesta de solución..... 40

Figura 9. Diagrama de despliegue de la configuración de red..... 41

Figura 10. Interfaz gráfica de usuario para la administración de red 42

Figura 11. Método redConfig() correspondiente al RF2. Configurar red..... 45

Figura 12. Grafo de flujo..... 45

Figura 13. Resultados de las pruebas funcionales 47

ÍNDICE DE TABLAS

Tabla 1. Tabla comparativa de herramientas informáticas para la configuración de la red 16

Tabla 2. Diccionario de datos..... 23

Tabla 3. Listado de requisitos funcionales de la propuesta de solución 27

Tabla 4. Listado de requisitos no funcionales de la propuesta de solución 28

Tabla 5. Descripción de la historia de usuario de los RF2. Configurar red 29

Tabla 6. Diseño de caso de prueba correspondiente al RF2 Configurar red 32

Tabla 6. Estructura del fichero de configuración de los parámetros de la red en Nova Servidores. 37

Tabla 8. Caso de prueba para el camino 2 del método redConfig() 47

INTRODUCCIÓN

En el mundo computarizado de hoy, las redes informáticas están presentes en casi todos los ámbitos de la sociedad. Las redes informáticas, de comunicaciones de datos o de computadoras son un número de sistemas informáticos conectados entre sí mediante una serie de dispositivos que permiten compartir información en paquetes de datos, transmitidos mediante impulsos eléctricos, ondas electromagnéticas o cualquier otro medio físico. Las redes informáticas no son distintas en su lógica de intercambio de los demás procesos de comunicación conocidos: cuentan con un emisor, un receptor y un mensaje, así como un medio a través del cual transmitirlo y una serie de códigos o protocolos para garantizar su comprensión mediante sistemas computacionales automatizados (Gómez, 2016).

Cuando se dispone de computadores en red, es posible crear una comunicación interna, compartir un punto de acceso a Internet o la administración de periféricos como impresoras o escáneres así como el envío de datos y archivos sin necesidad de dispositivos de almacenamiento secundario. Esto se logra a través de una serie de estándares de comunicación, que “traducen” a un mismo idioma los procesos de las diversas computadoras (Gómez, 2016). En la estructura de una red existen elementos de hardware (las piezas y equipos que permiten el establecimiento físico de la red) y elementos de software (los programas requeridos para administrar los componentes de la red y permitir la transmisión de datos) (Donostia-San Sebastián, 2021).

Existen equipos que desempeñan las funciones de servidor y otros son clientes permitiendo ofrecer y consumir información o recursos. Los servidores son los que procesan el flujo de los datos y centralizan el control de la red. Los clientes se refieren a las computadoras o dispositivos móviles que tienen la función de consumir información, formando parte de la red y permitiendo a los usuarios el acceso a esta. Los medios de transmisión es la vía por la que viaja los datos y puede ser cableada o inalámbrica. Tanto en los servidores como en los clientes existen sistemas operativos capaces de la interconexión de los diferentes componentes de hardware y software de las nuevas tecnologías de las comunicaciones (Donostia-San Sebastián, 2021).

El desarrollo de las Tecnologías de la Información y las Comunicaciones (TICs) es un proceso global y constante del que Cuba no está exenta. Aunque existen progresos en estas esferas, concurren limitaciones del ámbito legal que impiden acceder a plenitud a todos los avances presentes en el contexto mundial; hechos que han obligado a la sociedad cubana a buscar soluciones propias para enfrentar estas deficiencias. Dentro de las soluciones creadas se

encuentra el sistema operativo cubano (Nova), una distribución de GNU/Linux¹, con el fin de garantizar soberanía tecnológica; y una de las propuestas de sistemas operativos a emplear durante el proceso de migración hacia plataformas de software libre y código abierto en Cuba (Carballeira, 2021).

Nova, fue desarrollado por especialistas de la Universidad de Ciencias Informáticas. Para Cuba es de gran importancia, puesto que la necesidad de desarrollo de las TICs en la isla, permite que cada día se busquen nuevas formas de progreso y en este caso el código abierto brinda mejorar de manera segura, la estrategia nacional para la informatización de la sociedad cubana. Teniendo en cuenta la visión del país de su desarrollo económico, político y social (Carballeira, 2021). Nova, dispone de un ecosistema de aplicaciones y versiones que responden de manera más precisa a las necesidades de los usuarios en el país. Se despliega en cuatro variantes: Nova Escritorio, Nova Ligero, NovaDroid y Nova Servidores, este último destinado a la administración de servicios telemáticos (Domínguez, 2018).

En la actualidad, en diferentes organizaciones cubanas que disponen de servicios de red para la gestión de sus procesos se ha empleado, la distribución cubana de sistemas operativos GNU/Linux, Nova Servidores. Debido a que esta tecnología está basada en software libre y como parte de la estrategia nacional para garantizar la independencia y soberanía tecnológica en el país. Especialistas del Centro de Software Libre (CESOL), perteneciente a la Universidad de las Ciencias Informáticas (UCI), y líder en la migración hacia tecnologías libres y de código abierto, ha desplegado esta distribución en diferentes Organismos de la Administración Central del Estado (Carballeira, 2021).

En una encuesta aplicada a 2 especialistas (Ver anexo 1) de CESOL, relacionados con estos procesos de migración en los diferentes OACEs se evidenció que en el proceso de configuración de la red en Nova Servidores se presentaron las siguientes deficiencias:

- Pérdida de tiempo como consecuencia de que esta configuración se realiza de forma manual directamente en la consola del sistema operativo.
- Incurrencia en errores humanos debido a que los administradores de red necesitan dominar los comandos necesarios para realizar las configuraciones, así como, los ficheros que son necesarios modificar y su ubicación.

Dada la situación anterior se define como **problema científico**:

¿Cómo mejorar el proceso de configuración de red en Nova Servidores?

¹ GNU/Linux es la denominación técnica y generalizada que reciben una serie de sistemas operativos de tipo Unix, que también suelen ser de código abierto, multiplataforma, multiusuario y multitarea

Se determina como **objeto de estudio** el proceso de configuración de red en sistemas operativos GNU/Linux; delimitando como **campo de acción** herramientas informáticas para la administración de la configuración de red en sistemas operativos GNU/Linux.

Para dar solución al problema científico identificado se establece como **objetivo general** de la investigación: Desarrollar una herramienta informática que permita la administración remota de la configuración de red en Nova Servidores.

Se determinó como **objetivos específicos**:

1. Elaborar el marco teórico de la investigación sobre el proceso de configuración de red en sistemas operativos GNU/Linux.
2. Diseñar una herramienta informática para la administración remota de la configuración de red en Nova Servidores.
3. Implementar una herramienta informática para la administración remota de la configuración de red en Nova Servidores.
4. Evaluar la herramienta informática para la administración remota de la configuración de red en Nova Servidores.

Se definen como **preguntas científicas**:

1. ¿Cuáles son los presupuestos teóricos que fundamentan el proceso de configuración de red en sistemas operativos GNU/Linux?
2. ¿Qué elementos se deben tener en el diseño de una herramienta informática para la administración remota de la configuración de red en Nova Servidores?
3. ¿Qué componentes de software son necesarios implementar con el objetivo de obtener una herramienta informática para la administración remota de la configuración de red en Nova Servidores?
4. ¿Qué pruebas de software se deben aplicar en la evaluación de la herramienta informática para la administración remota de la configuración de red en Nova Servidores?

En el desarrollo de la investigación se utilizaron los siguientes **métodos de la investigación científica**:

Métodos teóricos

Analítico-Sintético: se utiliza para el análisis de las fuentes bibliográficas existentes referente al proceso de configuración de red en diferentes distribuciones GNU/Linux y en Nova Servidores. De igual forma, se identifican y sintetizan los elementos más importantes y necesarios para dar solución al problema planteado.

Inductivo-deductivo: a través de este método se pudo arribar a conclusiones generales sobre la configuración de la red. Permitted definir la estructura de configuración de la red, a partir de las necesidades de los administradores de redes y de los sistemas informáticos estudiados.

Métodos empíricos

Entrevista: La entrevista (Anexo 1) se realizó para conocer las características de la propuesta de solución, la obtención de funcionalidades de software y comprender el contexto de Nova Servidores durante el proceso de administración de la configuración de la red. Se pone de manifiesto en los contactos realizados con 2 especialistas del proyecto Nova Servidores.

Observación: La observación (Anexo 2) se empleó para registrar el comportamiento del proceso de configuración de la red, así como el análisis de diferentes herramientas informáticas que permiten la administración de redes informáticas.

Modelación: Se utilizó para elaborar los diferentes diagramas y modelos de la propuesta de solución. En la representación de las características y las relaciones entre los objetos presentes en el contexto del negocio informatizado, las funcionalidades de la solución y sus restricciones de diseño.

El documento consta de 3 capítulos estructurados de la siguiente forma:

Capítulo 1. Fundamentación teórica sobre el proceso de configuración de red en sistemas operativos GNU/Linux: se realiza el estudio del arte del proceso de configuración de la red de ellas se analiza la metodología de desarrollo a emplear, y se investigan los sistemas operativos existentes que den solución a problemas similares, así como las tecnologías y herramientas a utilizar en el desarrollo de configuración de la red.

Capítulo 2. Análisis y diseño de una herramienta informática para la administración remota de la configuración de red en Nova Servidores: se describe la solución propuesta y se ofrecen detalles de los principales aspectos relacionados con su diseño. Se muestra el levantamiento de los requisitos funcionales y no funcionales de la solución propuesta. Se explica el funcionamiento de la aplicación a través del modelo de casos de uso y las historias de usuario, así como las descripciones correspondientes de cada uno de ellos.

Capítulo 3. Implementación y evaluación de la herramienta informática para la administración remota de la configuración de red en Nova Servidores: en este capítulo se describen los elementos relacionados con la implementación, reflejando las prácticas de programación y los estándares de codificación utilizados para el cumplimiento de los requisitos funcionales especificados previamente. Además, se detallan las estrategias trazadas para el

desarrollo de las pruebas realizadas a la aplicación, con el objetivo de validar su correcto funcionamiento y su correspondencia con las funcionalidades implementadas, así como los resultados obtenidos en la aplicación de las pruebas.

CAPÍTULO 1. Fundamentación teórica sobre el proceso de configuración de red en sistemas operativos GNU/Linux

El presente capítulo aborda aspectos sobre la fundamentación teórica de la investigación. Contiene un estudio sobre el proceso de configuración de red en sistemas operativos GNU/Linux, con la finalidad de conocer sus características. Presenta un análisis sobre diferentes herramientas informáticas para la administración de la configuración de red en estos sistemas operativos. Se define la metodología de desarrollo de software utilizada para la elaboración de la propuesta de solución, así como los lenguajes y herramientas empleados en el modelado y la implementación de la solución.

1.1 Definición de conceptos

Red informática

Sistema de comunicaciones que permite comunicarse con otros usuarios y compartir archivos y periféricos. Es decir, es un sistema de comunicaciones que conecta a varias unidades y que les permite intercambiar información como compartir recursos, alta fiabilidad, ahorro económico, medio de comunicación. Las redes de computadoras tienen los cuatro componentes básicos que identifican dicho proceso nodo transmisor o fuente, medio de comunicación por el que fluye la información, nodo receptor o destino, el mensaje o la información (Moleros, 2018).

Configuración de red

La configuración de red es el diseño general de algún tipo de red de comunicaciones. Como parte de esta configuración, tanto el software como el hardware están dispuestos de manera que permitan una eficiencia óptima en el proceso de transmisión de datos entre dos o más nodos² físicos de la red. Una configuración de red puede ser local y algo contenida, o involucrar una colección de ubicaciones o nodos que se encuentran en todo el país o incluso en todo el mundo (McCarthy, 2021).

Administración remota

Es la posibilidad de acceder y controlar, total o parcialmente, computadoras desde cualquier parte del mundo. Lo hace a través de Internet o una red local para ejecutar diferentes actividades deseadas entre un usuario y otro (Velásquez, 2018).

² Nodo (host) a una máquina que se conecta a la red en un sentido amplio un nodo puede ser un ordenador, una impresora, una torre de CD, es decir, un elemento activo y diferenciable en la red que reclama o presta algún servicio o comparte información.

Administración remota de la configuración de red

La administración remota de la configuración de red consiste en la gestión los parámetros necesarios para garantizar la interconexión de los equipos informáticos asociado a una red informática. Brinda servicios de asistencia técnica remota, aumenta la productividad laboral y ahorra tiempo y costos. Permite supervisar de forma segura las configuraciones de red en servidores de difícil acceso (Cillero, 2020).

1.2 Descripción del proceso de configuración de red en sistemas operativos GNU/Linux

La configuración de red en Linux se guarda en diferentes archivos dependiendo de la distribución de Linux utilizada, existiendo diversas ubicaciones, esto implica crear o editar los archivos de configuración de red. El modo en que la información de configuración está disponible para un sistema puede variar (Linux-Ubunto, 2016).

1.2.1 Modos de configuración de red

La configuración de red en una computadora con sistema operativo GNU/Linux puede ser de dos modos: manual o estática y automática o dinámica.

Configuración manual o estática

En la configuración manual de la red, se definen parámetros (epígrafe 1.2.2) estáticos para cada computadora en el archivo de configuración “*interfaces*” ubicado en la dirección “*/etc/network/*”. Los usuarios que no pertenecen al grupo “*network*” deben ejecutar este archivo por medio de “*sudo*” si quieren hacer modificaciones. Si se ejecuta Linux en una máquina conectada mediante una red cableada, el nombre de la interfaz de red es “*eth0*”, si la conexión es inalámbrica la identificación de la interfaz es “*wifi0* o *wlan0*” (Linux-Ubunto, 2016).

Si se ejecuta “*ifconfig*” pasando esta “*interface*” como parámetro se obtiene la configuración de red. A continuación, se muestra un ejemplo de la aplicación de este comando:

ifconfig eth0

```
eth0 Link encap:Ethernet HWaddr 00:11:22:33:44:55
```

```
inet addr:192.168.42.10 Bcast:192.168.42.255 Mask:255.255.255.0
```

```
inet6 addr: fe80::211:2fff: fe91:2437/64 Scope:Link
```

```
UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
```

```
RX packets:60918 errors:0 dropped:0 overruns:0 frame:0
```

```
TX packets:56872 errors:0 dropped:0 overruns:0 carrier:0
```

```
collisions:0 txqueuelen:1000
```

```
RX bytes:63734800 (60.7 Mb) TX bytes:5711244 (5.4 Mb)
```

Una explicación de las líneas de comandos anteriores:

- “Link encap” determina qué tipo de “interface” se trabaja, en este caso, es una conexión a una red Ethernet.
- “HWaddr” es la dirección MAC (*Media Access Control*, Control de Acceso al Medio) de la “interface” de red en cuestión.
- “inet addr” aa.bb.cc.dd Bcast: zz.yy.xx.ww corresponde a las direcciones de IP y broadcast respectivamente.
- “netmask” aa.bb.cc.dd corresponde a la expresión decimal de la máscara de red.
- “RX packets” representa el número de paquetes recibidos.
- “TX packets” representa el número de paquetes enviados.

Cuando el servidor tiene a su disposición una tabla que empareja direcciones MAC (*Media Access Control*, Control de Acceso al Medio) con direcciones IP, creada manualmente por el administrador de la red. Sólo clientes con una dirección MAC válida recibirán una dirección IP del servidor (Glinux, 2020).

Para establecer manualmente los parámetros de red: “IP”, “netmask”, “Gateway” debe ser de la manera descrita a continuación.

```
auto eth0
```

```
iface eth0 inet static
```

```
address 192.168.82.209
```

```
netmask 255.255.255.0
```

```
gateway 192.168.82.1
```

```
dns-nameservers 8.8.8.8 8.8.4.4
```

Posteriormente se reinicia el servicio de red y se tiene la configuración de red fija.

Configuración dinámica o automática

En la configuración dinámica o automática de la red, se definen parámetros (epígrafe 1.2.2) de forma automática, mediante el servicio DHCP (*Dynamic Host Configuration Protocol*, Protocolo de Configuración Dinámica de Host). El administrador de la red determina un rango de direcciones IP y cada computadora conectada a la red está configurada para solicitar su dirección IP al servidor

DHCP cuando la tarjeta de interfaz de red se inicializa. Esto facilita la reutilización dinámica de las direcciones IP (Glinux, 2020).

1.2.2 Parámetros de configuración de la red

En el proceso de configuración de la red se definen los siguientes parámetros:

Host name

Cada nodo debe tener además un único nombre en la red. Ellos pueden ser sólo nombres o bien utilizar un esquema de nombres jerárquico basado en dominios. Los nombres de los nodos deben ser únicos, lo cual resulta fácil en pequeñas redes, más dificultoso en redes extensas e imposible en Internet si no se realiza algún control (Ferrer, 2020).

Dirección de red Ethernet (Ethernet address o MAC address)

Un número de 48 bits que se encuentra en el dispositivo físico (hardware) del controlador de red Ethernet y es grabado por el fabricante del mismo; este número debe ser único en el mundo, por lo que cada fabricante de “network” “interface” “card” (tarjeta de interfaz de red) NIC tiene un rango preasignado (Ferrer, 2020).

Dirección IP

Es un número que identifica de manera lógica y jerárquica a una interfaz de un dispositivo (habitualmente una computadora) dentro de una red que utilice el protocolo “IP,” que corresponde al nivel de red o nivel 3 del modelo de referencia OSI (Ferrer, 2020).

Dirección de Internet (IP address)

Está compuesto por cuatro números en el rango 0-255 separados por puntos (por ejemplo 192.168.0.1) y es utilizado universalmente para identificar los ordenadores sobre una red o Internet. La traslación de nombres en direcciones “IP” es realizada por un servidor “DNS” (Domain Name System) que transforma los nombres de nodo (legibles por humanos) en direcciones IP (Ferrer, 2020).

Puerto (port)

Identificador numérico “del buzón” en un nodo que permite que un mensaje (TCP, UDP) pueda ser leído por una aplicación concreta dentro de este nodo (por ejemplo, dos máquinas que se comuniquen por telnet lo harán por el puerto 23, pero las dos mismas máquinas pueden tener una comunicación ftp por el puerto 21). Se pueden tener diferentes aplicaciones comunicándose entre dos nodos a través de diferentes puertos simultáneamente (Ferrer, 2020).

Gateway

Es un nodo que realiza encaminamientos. Un router, según sus características, podrá transferir información entre dos redes de protocolos similares o diferentes y puede ser además selectivo (Ferrer, 2020).

Domain Name System (DNS, Sistema de Nombres del Dominio)

Permite asegurar un único nombre y facilitar la administración de las bases de datos que realizan la traslación entre nombre y dirección de Internet, y se estructuran en forma de árbol. Para ello, se especifican dominios separados por puntos, de los que el más alto describe una categoría, institución o país. El segundo nivel representa la organización, el tercero y restantes departamentos, secciones o divisiones dentro de una organización (Ferrer, 2020).

DHCP (Protocolo de Configuración Dinámica de Host), bootp

DHCP y bootp son protocolos que permiten a un nodo cliente obtener información de la red (tal como la dirección IP del nodo). Muchas organizaciones, con gran cantidad de máquinas, utilizan este mecanismo para facilitar la administración en grandes redes o donde existe una gran cantidad de usuarios móviles (Ferrer, 2020).

1.2.3 Administración de la configuración de red en distribuciones GNU/Linux

En el epígrafe se presentan ejemplos de cómo realiza la configuración de red en las distribuciones de GNU/Linux Debian y Redhat.

Debian

En Debian la configuración de red se guarda en el archivo *“interfaces”* disponible en la dirección *“/etc/network”*. En el archivo se guarda, para cada interfaz de red los siguientes parámetros (Glinux, 2020):

- Su estado de inicio (automático, manual o asistido)
- La estructura de configuración (IP estática, DHCP, promisc, asistida por booteo).
- La dirección de red y la máscara de red asociadas, a menos que la interface sea DHCP.
- Los nombres de los scripts (secuencia de comandos o guión) que se usa para designar a un programa relativamente simple) responsables del inicio (ifconfig interface up) y parada (ifconfig interface down) de cada interface.

El formato del archivo de configuración es como sigue en el ejemplo:

```
# this file describes the network interfaces available on your system
```

```

# and how to activate them. For more information, see interfaces ()
# Interface loopback, un dispositivo que el kernel utiliza
# para comunicación de redes interna ("consigo mismo")
# la primera interface disponible recibe la IP 127.0.0.1 auto lo iface lo inet loopback
# Interface ethernet servida por el modelo de gestión de red "static"
# (configuración de red estática)
auto eth0

iface eth0 inet static

address 200.22.22.71

netmask 255.255.255.240

gateway 200.22.10.65

dns-nameservers 8.8.8.8

post-up echo "Interface eth0 levantada con exito"
post-down echo "Interface eth0 bajada con exito"
# Interface ethernet servida por el modelo "dhcp"
# (configuración IP se recibe de un servidor DHCP)
auto eth1

iface eth1 inet dhcp

leasehours 86400

client dhcpd

```

Redhat

En Redhat (Fedora, CentOS, Red Hat Enterprise Linux), la configuración de red se guarda en entradas bajo el directorio `“etc/sysconfig/network”`, siendo un archivo para cada interface de red. Cada archivo contiene una serie de variables describiendo distintas características de la interface. El directorio asociado `“scripts/”` guarda los scripts utilizados para la inicialización (`ifconfig interface up`) y parada (`ifconfig interface down`) de estas interfaces (Debian, 2021). A continuación, se presenta un ejemplo de este archivo:

ifcfg-eth0 que guarda la configuración de la tarjeta de red:

```

# Network Card brand, model

DEVICE=eth0

HWADDR=00:11:22: 33:44:55

ONBOOT=yes

IPADDR=192.168.20.2

BOOTPROTO=none

NETMASK=255.255.0.0

TYPE=Ethernet

GATEWAY=192.168.20.1

USERCTL=no

PREFIX=16

NAME="Ethernet-0 Cableado"

UUID=5fb06bd0-0bb0-7ffb-45f1-d6edd65f3e03

NM_CONTROLLED=yes

DEFROUTE=yes

```

Ubuntu

Hay varias maneras de establecer una dirección IP en Ubuntu. Puede configurar la interfaz de red para que utilice IP dinámica utilizando el servidor DHCP o puede configurar manualmente una dirección IP estática. En Ubuntu, puede establecer la dirección IP a través de los comandos de la terminal. Inicialmente se debe escribir el siguiente comando *“netstat -I”* para encontrar el nombre de la interfaz (Debian, 2015). Luego se debe escribir el siguiente comando:

```
sudo ifconfig eth0 192.168.72.6 netmask 255.255.255.0
```

Posteriormente, para agregar una pasarela predeterminada, agregue el siguiente comando:

```
sudo route add default gw 192.168.72.1 eth0
```

En la ficha IPv4, puede seleccionar el botón de opción Automático (DHCP) para que el sistema pueda obtener la dirección IP de forma dinámica. Aquí hay otro método que puede utilizar para configurar la dirección IP. Para establecer la dirección IP de forma dinámica, debe editar

/etc/network/ se edita el fichero “*interfaces*”. En /etc/network/ se almacena la configuración básica de las interfaces.

Para la edición o configuración a realizar: /etc/network/interfaces introduciendo el siguiente comando en el terminal.

```
sudo nano /etc/network/interfaces
```

Luego se debe agregar las siguientes líneas:

```
auto eth1
```

```
iface eth1 inet dhcp
```

Se prosigue a guardar el archivo y reinicie los servicios de red utilizando el siguiente comando:

```
sudo systemctl restart networking
```

Para establecer una dirección IP estática, debe editar:

```
etc/network/interfaces
```

```
sudo nano /etc/network/interfaces
```

Se escriben las siguientes líneas “a /etc/network/interfaces” para que muestres:

```
auto eth1
```

```
iface eth1 inet static
```

```
address 192.168.72.8
```

```
netmask 255.255.255.0
```

```
gateway 192.168.72.1
```

```
dns-nameservers 8.8.8.8 4.4.2.2
```

Se guarda el archivo y proseguimos a reiniciar los servicios de red:

```
sudo systemctl restart networking
```

Distribución cubana GNU/Linux Nova

En la observación realizada al fichero de configuración de nova se analizó que la estructura de la configuración de la red se realiza en el siguiente fichero: “*interfaces*”.

```
# nano /etc/network/ interfaces.
```

Primeramente, es necesario conocer las interfaces de red, se hace a través del comando:

ifconfig -a

ifconfig es el encargado de mostrar toda la información asociada a las tarjetas de red del servidor.

Ejemplo de configurar la interfaz de red

eth0

iface eth0 inet static

address 192.168.10.15

netmask 255.255.255.0

network 192.168.10.0

broadcast 192.168.10.255

gateway 192.168.10.1

address: dirección única usada para identificar el equipo en la red.

netmask: indica al equipo el tamaño de la red a la que está conectada. Usa el mismo formato que la dirección IP, por ejemplo, 255.255.255.0.

gateway: es la dirección IP del dispositivo que busca el equipo para acceder a Internet. Normalmente será la dirección IP del router.

1.3 Herramientas informáticas para la administración de la configuración de red sistemas operativos GNU/Linux

En el siguiente epígrafe se efectuará un análisis comparativo de algunas herramientas informáticas para la configuración de red. Mediante este análisis se quiere identificar si el software cumple con los requisitos técnicos y funcionales, mediante la comparación. En la actualidad existen diversos sistemas, para la configuración y administración de la red dedicadas a modificar y controlar este tipo de servicio de ahí que se arribó al siguiente estudio:

NetworkManager: es un servicio de red del sistema que gestiona los dispositivos y conexiones de red e intenta mantener la conectividad de la red activa cuando está disponible. Gestiona dispositivos Ethernet, WiFi, de banda ancha móvil, a la vez que proporciona integración VPN con una variedad de diferentes servicios VPN. Gestiona los dispositivos de red en el sistema mediante la creación de un archivo de configuración netplan en el que se establece como el renderizando de red por defecto (Domínguez, 2018).

Ethtool: se puede utilizar para consultar y cambiar ajustes como la velocidad, la negociación automática y la descarga de suma de comprobación en muchos dispositivos de red, especialmente los dispositivos Ethernet. Se puede utilizar para controlar y ver la configuración de los dispositivos Ethernet cableados, también permite obtener la información de identificación y diagnóstico de la red, obtener estadísticas ampliadas de los dispositivos de red, controlar la velocidad, el dúplex, la autonegociación y el control de flujo de los dispositivos Ethernet, controlar las características de descarga de hardware, controlar la moderación de las interrupciones, controlar la selección de la cola de recepción para dispositivos de cola múltiple y actualizar el firmware en la memoria flash (Domínguez, 2018).

IP Changer: Las conexiones de la red están configuradas bajo un esquema de ajustes y preferencias que son seleccionadas por el computador de manera automática, cada vez que intente conectarse a ella. Sin embargo, estos ajustes muchas veces pueden estar erróneos, ya sea que hayas cambiado de red o porque tu proveedor haya realizado algunas modificaciones. Todo tiene solución si utilizas IP Changer, un increíble software que te permitirá cambiar todo desde el principio. Aunque también serás capaz de hacer ajustes más pequeños, por si solo necesitas modificar un pequeño detalle. En general, la dirección IP almacena este tipo de información mediante algo llamado TCP/IP y podrás configurar los servidores DNS, puertos de enlaces, los entornos de redes locales, entre otros detalles increíbles. La aplicación es compatible con dispositivos Windows y es bastante efectiva para solucionar problemas comunes, pero todo tendrás que hacerlo manualmente (Domínguez, 2018).

NetSetMan: es un programa cuyo origen se remonta al año 2008. Se trata de una buena herramienta que puede servir de ayuda a los usuarios y administradores de Windows en lo que se refiere a la gestión de redes. Hay que señalar que este programa es muy útil en equipos que se conectan a diferentes tipos de redes a lo largo de una jornada. La última versión NetSetMan permite gestionar diferentes perfiles de red y hay que señalar también que además es gratuita. Con este programa, un administrador de redes puede configurar perfiles con información de red específica con direcciones IP y máscara de subred, puertos de enlace, servidores DNS (Domínguez, 2018).

1.3.1 Análisis comparativo de herramientas informáticas para la administración de la configuración de red

En la tabla 1 se presenta una comparación de las herramientas estudiadas basada en los criterios de análisis definidos en correspondencia a las deficiencias relacionadas en la situación problemática y las características del contexto de negocio y la propuesta de solución.

Criterios para el análisis: Sí (cumple con el criterio), No (incumple con el criterio). Para ser tomado como propuesta de solución debe tener (Sí) en todos los campos de comparación.

Tabla 1. Tabla comparativa de herramientas informáticas para la configuración de la red

Fuente: Elaboración propia

Parámetros	Herramientas			
	NetworkManager	Ethtool	IP Changer	NetSetMan Pro
Interfaz gráfica de usuario	Sí	Sí	Sí	Sí
Se integra a Nova Servidores	No	No	No	No
Permite la configuración remota de la red	No	No	Sí	Sí
Garantiza la integridad de los datos	Sí	Sí	Sí	Sí

Lo que lleva a concluir que para la configuración remota de Nova Servidores estas herramientas cumplen con algunas características, pero no todas, llevando a plantear la propuesta de solución de una nueva herramienta que no solo permita la configuración de red, sino que sea capaz de hacerlo de manera remota y de integrarse a Nova Servidores.

1.4 Caracterización de Nova Servidores

En una entrevista realizada al programador principal de Nova Servidores se llega a concluir que Nova Servidores es la distribución cubana de GNU/Linux Nova en su variante para servidores. Provee a la universidad y el país un producto libre de licencias privativas. Responde a las características de la infraestructura tecnológica de los diferentes Organismos de la Administración Central del Estado. Es un producto completo que desde su base posibilita la instalación y configuración de diversas tecnologías para la prestación de servicios. Entre las tecnologías escogidas por su carácter estratégico se encuentran los siguientes servicios:

- Apache, un servidor web que procesa aplicaciones del lado del servidor, realizando conexiones bidireccionales o unidireccionales y síncronas o asíncronas con el cliente y generando o cediendo una respuesta en el lado del cliente.
- Correo, implementa las tecnologías para soportar la infraestructura del servicio de correo.

- Fail2ban, es un servicio para la prevención de intrusos en un sistema informático, actúa penalizando o bloqueando las conexiones remotas que intentan accesos por fuerza bruta.
- NFS, *Network File System*, Sistema de archivos de red: Es utilizado para sistemas de archivos distribuido en un entorno de red de computadoras de área local. Posibilita que distintos sistemas conectados a una misma red accedan a ficheros remotos como si se tratara de locales.
- Nginx, un servidor web.
- ProFTPD, un servidor FTP, *File Transfer Protocol*, Protocolo de transferencia de archivos.
- Proxy, un servidor que hace de intermediario en las peticiones de recursos que realiza una computadora cliente a un servidor.
- Samba4, como controlador de dominio. Permite la autenticación, que es el proceso de garantizar o denegar a un usuario el acceso a recursos compartidos o a otra máquina de la red y el control de recursos.
- Báculo, un servidor que implementa herramientas para el respaldo de archivos.
- LTSP, *Linux Terminal Server Project*, Servidor de Clientes Ligeros, servidor que implementa un conjunto de aplicaciones que proporcionan la capacidad de ejecutar Linux en computadores de pocas prestaciones.
- Antivirus, servidor que contiene herramientas para detectar y eliminar programas malignos.
- Cortafuegos, servidor para bloquear el acceso no autorizado.
- Zabbix, herramientas para el monitoreo en tiempo real y para el análisis de trazas de redes.
- IDS, *Intrusion Detection System*, Sistema de detección de intrusiones, un programa de detección de accesos no autorizados a un computador o a una red.

1.5 Metodología de desarrollo de software

En el desarrollo de software la necesidad de organizar o estructurar de forma correcta y disciplinada, es uno de los factores más importantes para evitar pérdidas de tiempo y recursos. Con el fin de prevenir tales errores es preciso definir una estrategia que ordene las posibles tareas a desarrollar, así como llevar a cabo una guía de cómo efectuar las actividades, procedimientos y pasos que se deben de seguir para el desarrollo de un software (Rodríguez, 2015). En el epígrafe 1.5.1 se define la metodología de desarrollo de software empleada en la elaboración de la propuesta de solución.

1.5.1 Variación de AUP para la UCI

Para el desarrollo de este proyecto se decidió utilizar una variante de la metodología AUP para la UCI (Proceso Unificado Ágil para la Universidad de Ciencias Informáticas) que permite

estandarizar los procesos para la implementación y desarrollo del software; este se denominada AUP-UCI, la cual es idónea para la entrega de productos en cortos períodos de tiempo; además de satisfacer las necesidades del proceso convirtiéndose en la metodología rectora de su desarrollo productivo (Rodríguez, 2015).

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 proyecto, realizar estimaciones de tiempo, esfuerzo y costo y decidir si se ejecuta o no el proyecto. En una manera resumida que se necesita para la ejecución del proyecto. En esta se tendrá en cuenta la tecnología a utilizar, la duración, las herramientas.

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. Durante el desarrollo se modela el negocio, obtienen los requisitos, se elaboran la arquitectura y el diseño, se implementa y se libera el producto. Esta fase no necesariamente se realiza al finalizar la fase inicial debido a que en ella se puede estar ejecutando mientras la fase inicial aún está en curso. Se implementará la aplicación.

Cierre: En esta fase se analizan tanto los resultados del proyecto como la ejecución y se realizan las actividades formales de cierre del proyecto. Esta fase solo se ejecutará si las anteriores descritas están culminadas.

De igual forma se definen por parte de la metodología 4 escenarios para la disciplina de Requisitos. Para el caso de la presente investigación se selecciona el escenario 4. El mismo se aplica a los proyectos que hayan evaluado el negocio a informatizar y como resultado obtengan un negocio muy bien definido. El cliente estará siempre acompañando al equipo de desarrollo para convenir los detalles de los requisitos y así poder implementarlos, probarlos y validarlos. Se recomienda en proyectos no muy extensos.

1.6 Lenguajes y herramientas para el modelado de la solución

En el modelado de la propuesta de solución se utilizó el Lenguaje Unificado de Modelado y las herramientas descritas a continuación.

1.6.1 Herramienta para el modelado de la solución

Las herramientas para el modelado de software ayudan al proceso de desarrollo del software en tareas como: realizar el diseño, implementación de parte del código con el diseño dado, compilación automática, documentación o detección de errores. Aumentan la productividad en la

elaboración de soluciones informáticas reduciendo el coste de las mismas en términos de tiempo y de dinero (Mendez, 2018). A continuación, se presenta una breve descripción de la herramienta empleada en el diseño de la propuesta de solución.

Visual Paradigm 8.0

Constituye una herramienta CASE (*Computer Aided Software Engineering*, Ingeniería de Software Asistida por Computadoras) que utiliza UML (*Unified Modeling Language*, Lenguaje Unificado de Modelado). Soporta el ciclo de vida del desarrollo de software, además de tener la ventaja de ser multiplataforma. Propicia un conjunto de funcionalidades para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación (Visual Paradigm Internationals, 2018).

Lenguaje de modelado UML 2.0

El UML fue creado para forjar un lenguaje de modelado visual común, semántica y sintácticamente para elaborar el diseño de sistemas de software, tanto en estructura como en comportamiento. Consiste en diferentes tipos de diagramas que describen los límites, la estructura y el comportamiento del sistema y los objetos que contiene. El UML se perfecciona continuamente de forma tal que la facilidad de uso, la implementación y la adaptación se pueden realizar de manera sencilla. (Lucid, 2018).

1.7 Tecnologías para la implementación

Para efectuar la implementación de la propuesta de solución se utilizó las siguientes tecnologías: Python (como lenguaje de programación), Augeas (administración de archivos de configuración del sistema), Python dialog (librería de Python para la interfaz de escritorio), ParaMiko (para la conexión remota desde la computadora cliente al servidor) librería de Python. En los epígrafes siguientes se brinda una descripción de estas tecnologías.

1.7.1 Lenguajes de programación

Un lenguaje de programación es un lenguaje formal que mediante una serie de instrucciones, le permite a un programador escribir un conjunto de órdenes, acciones consecutivas, datos y algoritmos para crear el programa que controle el comportamiento físico y lógico de una máquina (Domínguez, 2018).

Python 3.10

Es un lenguaje de programación potente y fácil de aprender. Tiene estructuras de datos de alto nivel eficientes y un simple pero efectivo sistema de programación orientado a objetos. La sintaxis de Python y su tipado dinámico, junto a su naturaleza interpretada lo convierten en un lenguaje

ideal para scripting (secuencia de comandos o guion) y desarrollo rápido de aplicaciones en muchas áreas, para la mayoría de plataformas (Python, 2021).

1.7.2 Entorno de desarrollo integrado

Un Entorno de Desarrollo Integrado (IDE, *Integrated Development Environment*) es una aplicación informática que proporciona servicios integrales para facilitarle al desarrollador o programador la implementación del software. Consta de un editor de código fuente, herramientas de construcción automáticas y un depurador. La mayoría de estos tienen autocompletado inteligente de código. Algunos IDE contienen un compilador, un intérprete, o ambos (Perez, 2020). Para el desarrollo de la propuesta de solución se empleó Pycharm, descrito a continuación.

Pycharm 3.2

PyCharm proporciona una finalización del código inteligente, inspecciones del código, indicación de errores sobre la marcha y arreglos rápidos, así como refactorización de código automática y completas funcionalidades de navegación, ofrece una gran compatibilidad con marcos de trabajo de desarrollo web moderno como Django, Flask, Google App Engine, Pyramid y web2py, además se integra con IPython Notebook, cuenta con una consola de Python interactiva y es compatible con Anaconda y varios paquetes científicos como matplotlib y NumPy (Python, 2021).

Augeas 3.2

Augeas es una librería integrada a Python que permite la administración de ficheros de la configuración del sistema (Python, 2021).

Dialogs 3.2

Dialogs es una ventana emergente de cuadros de diálogo, sirven para mostrar o pedir información rápida al usuario. Reciben ese nombre porque no forma parte de la ventana principal. Para ello se utilizó Tkinter es un módulo de Dialogs el cual contiene clases y funciones para crear diálogos modales que obtienen un valor de entrada de usuario permitiendo así la implementación de las interfaces visuales (Python, 2021).

Paramiko 3.2

Paramiko es un módulo escrito en lenguaje Python, siguiendo el protocolo SSH³, que soporta la conexión de servidores remotos mediante encriptación y autenticación. Es compatible con Linux, Solaris, BSD, MacOS X, Windows y otras plataformas para conectarse de una plataforma a otra a

³ SSH (*Secure Shell*, intérprete de órdenes o intérprete de comandos seguro) es el nombre de un protocolo y del programa que lo implementa cuya función principal es el acceso remoto a un servidor por medio de un canal seguro en el que toda la información está cifrada.

través de SSH. Con este módulo, la conexión ssh y el protocolo sftp se pueden utilizar para la transferencia de archivos sftp cómodamente. Permitirá la conexión del ssh garantizando la integridad de datos e el sistema de configuración (Python, 2021). Permite hacer la configuración remota.

1.8 Conclusiones del capítulo

La elaboración y análisis del marco teórico sobre el proceso de configuración de red en sistemas operativos GNU/Linux y el estudio de los principales conceptos asociados al problema planteado, permitió establecer las bases para el desarrollo de la investigación y conocer las características del objeto de estudio. La caracterización y comparación realizada a varias herramientas para la administración de la configuración de red demostró la necesidad de desarrollar una herramienta que mejore el proceso de configuración de red en Nova Servidores. En el desarrollo de la propuesta de solución se empleó la metodología de desarrollo de software, Variación de AUP para la UCI, que guió la ejecución de las tareas y la elaboración de los productos de trabajo necesarios. Se definió el empleo de tecnologías libres para garantizar la soberanía tecnológica de la solución.

CAPÍTULO 2. Análisis y diseño de una herramienta informática para la administración remota de la configuración de red en Nova Servidores

2.1 Descripción del contexto de la propuesta de solución desarrollada

Este capítulo está enfocado al diseño de la solución del problema propuesto. En el mismo se describe el negocio y los requisitos funcionales y no funcionales. Se confecciona las Historias de Usuario (HU) necesarias para la solución de la problemática y la arquitectura de software a utilizar. A partir de los requisitos identificados se modela el diagrama de clase del diseño el cual permite obtener una visión más clara de la implementación del sistema y el modelo de base de datos del módulo propuesto. Además, se muestran ejemplos de estándares de codificación presentes en la implementación y los patrones utilizados.

2.1.1 Modelo conceptual

El modelo conceptual muestra los conceptos, sus atributos y la relación o asociación entre ellos. Un concepto es una idea, cosa u objeto. Para describirlos se debe analizar los sustantivos en las descripciones textuales del dominio del problema (Rand, 2015). En la Figura 1 se presenta el modelo conceptual que describe el contexto del negocio informatizado.

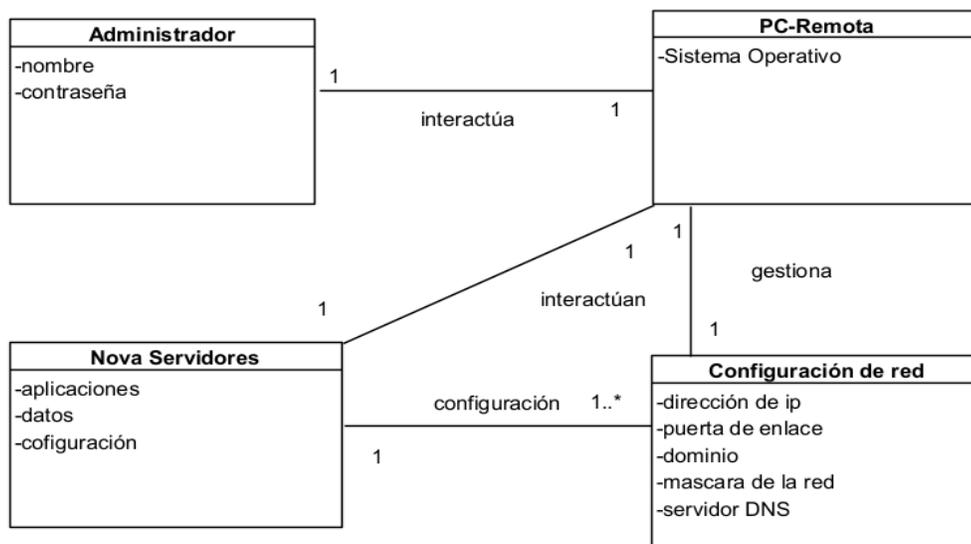


Figura 1. Descripción del contexto del negocio: Modelo conceptual

Fuente: Elaboración propia

En el modelo conceptual del negocio se observan todos los objetos que interactúan en él y a continuación se da una breve descripción de ellos:

- **Administrador:** es la persona designada en una organización cuya responsabilidad incluye el mantenimiento de las infraestructuras informáticas con énfasis en las redes.
- **PC-Remota:** Computadora cliente empleada por el administrador para acceder la información de la configuración de la organización o electromecánico que se utiliza para interactuar con un computador. También recibe el nombre de nodo o estación de trabajo.
- **Nova Servidores:** Es una Distribución de GNU/Linux desarrollada por estudiantes y profesores de la Universidad de las Ciencias Informáticas, con la participación de miembros de otras instituciones, para apoyar la migración a tecnologías de Software Libre y Código Abierto en Cuba. Esta versión en particular está orientada a servidores.
- **Configuración de la red:** Variables y parámetros que son almacenados en diferentes archivos de configuración, ya sea de manera centralizada o distribuida y son empleados por los distintos programas y servicios encargados de hacer funcionar la red tanto de los clientes y los servidores.

Para describir las características de los atributos se empleó el diccionario de datos. En el epígrafe 2.1.2 se evidencia un ejemplo de su utilización.

2.1.2 Diccionario de datos

En este epígrafe se describen los atributos de los conceptos del negocio mediante el diccionario de datos. Un diccionario de datos, es un repositorio centralizado de información sobre datos tales como significado, relación con otros datos, origen, uso y formato. En un diccionario de datos se encuentra la lista de todos los elementos que forman parte del flujo de datos en todo el sistema. Los elementos más importantes son flujos de datos, almacenes de datos y procesos. El diccionario guarda los detalles y descripciones de todos estos elementos (Álvares, 2017). En la tabla siguiente se presenta el diccionario de datos del concepto configuración de red.

Tabla 2. Diccionario de datos

Fuente: Elaboración propia

Descripción	Esta entidad describe toda la información referente a la configuración de red Nova Servidores.					
Atributos						
Nombre	Descripción	Tipo	¿Puede ser nulo?	¿Es único?	Restricciones	
					Clases válidas	Clases no válidas

Dirección de IP	Es un número que identifica de manera lógica y jerárquica a una interfaz	Alfanumérico	No	Sí	Cadena de caracteres numéricos. Y puntos.	Otros Símbolos, letras
Puerta de enlace	En este campo se coloca la dirección del Gateway que da acceso a otras redes	Alfanumérico	No	Sí	Caracteres numéricos. Y puntos	Otros Símbolos, letras
Dominio	Es una red de identificación asociada a un grupo de dispositivos o equipos conectados a una red	Alfanumérico	Sí	Sí	Cadena de caracteres numéricos. Y puntos.	Otros Símbolos, letras
Máscara	Indica al equipo el tamaño de la red a la que está conectada. Usa el mismo formato que la dirección IP	Alfanumérico	No	Sí	Cadena de caracteres numéricos. Y puntos.	Otros Símbolos, letras
DNS	Permite asegurar un único nombre y administración de las bases de datos, realizan la traslación entre nombre y dirección de Internet.	Alfanumérico	No	Sí	Cadena de caracteres numéricos. Y puntos.	Otros Símbolos, letras

2.2 Requisitos

La disciplina de requisitos es una de las más importantes en el proceso de desarrollo de software; consiste en desarrollar un modelo del sistema que se va a construir. Esta disciplina comprende la administración y gestión de los requisitos funcionales y no funcionales del producto (Rodríguez, 2015). A continuación, se presenta las actividades desarrolladas en esta disciplina, así como los productos de trabajo obtenidos en su ejecución.

2.2.1 Fuentes para la obtención de requisitos

Para identificar las necesidades del cliente se utilizan técnicas que permiten determinar y documentar los requisitos para el desarrollo de la solución. Esta actividad es continua durante el ciclo de desarrollo y combina, en diferentes puntos, diversas técnicas de identificación para obtener la visión más completa de las necesidades del usuario final (Pressman, 2010). Contribuye

a ajustar aprietos de beneficios entre los implicados, y establecer qué tipo de herramienta se desea desarrollar.

- Modelo conceptual elaborado
- Herramientas informáticas estudiadas
- Especialistas del centro CESOL

2.2.2 Técnicas de identificación de requisitos

Existen diferentes técnicas para obtener requisitos. A continuación, se describe las utilizadas en la obtención de requisitos de la propuesta de solución.

Entrevista

La entrevista es de gran utilidad para obtener información cualitativa como opiniones, o descripciones subjetivas de actividades. Es una técnica muy utilizada, y requiere una mayor preparación y experiencia por parte del analista. La entrevista se puede definir como un “intento sistemático de recoger información de otra persona” a través de una comunicación interpersonal que se lleva a cabo por medio de una conversación estructurada. Debe quedar claro que no basta con hacer preguntas para obtener toda la información necesaria. Es muy importante la forma en que se plantea la conversación y la relación que se establece en la entrevista (Guerra, 2015).

La técnica se aplicó a través de una entrevista realizada a 2 miembros del proyecto Nova Servidores (ver Anexo 1) para conocer cuáles son las tecnologías compatibles con el entorno de desarrollo, las particularidades de Nova Servidores, así como requerimientos y características que debe tener la propuesta de solución.

Desarrollo de prototipos

Los prototipos suelen consistir en versiones reducidas, demos o conjuntos de pantallas (que no son totalmente operativos) de la aplicación pedida. Esta técnica es particularmente útil cuando; el área de la aplicación no está bien definida (posiblemente por ser algo muy novedoso). Es necesario evaluar previamente el impacto del sistema en los usuarios y en la organización. Los prototipos de sistema permiten a los usuarios experimentar para ver cómo éste ayuda a su trabajo. Fomentan el desarrollo de ideas que desembocan en requerimientos. Además de permitir a los usuarios mejorar las especificaciones de requisitos, el desarrollo de un prototipo tiene otras ventajas; al demostrar las funciones del sistema se identifican las discrepancias entre los desarrolladores y los usuarios (Domínguez, 2018).

Durante el desarrollo del prototipo, el personal del desarrollo de software puede darse cuenta de que los requerimientos son inconsistentes y/o están incompletos. Aunque limitado, se dispone

rápidamente de un sistema que funciona y demuestra la factibilidad y usabilidad de la aplicación a administrar. El prototipo se utiliza como base para escribir la especificación de los requisitos (Domínguez, 2018). Se aplicó esta técnica para conocer cuáles son las características que debían tener las interfaces de la herramienta. Los prototipos elaborados fueron revisados por 2 miembros del equipo de desarrollo de Nova Servidores y emitieron sus valoraciones correspondientes.

La aplicación de las técnicas descritas permitió comprender con profundidad el problema de la investigación y facilitó 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

Los requisitos constituyen un punto clave en el desarrollo de aplicaciones informáticas. Un gran número de proyectos de software fracasan debido a una mala definición, especificación o administración de requisitos. Factores tales como requisitos incompletos o mal manejo de los cambios de los requisitos, llevan a proyectos completos al fracaso total. Los requisitos se enfocan en las especificaciones de lo que se desea desarrollar y tienen dos clasificaciones: requisitos funcionales y no funcionales. Los requisitos funcionales son declaraciones de los servicios que debe proporcionar el sistema, de la manera en que éste debe reaccionar a entradas particularidades y de cómo se debe comportar en distintas situaciones y los requisitos no funcionales son restricciones de los servicios o funciones ofrecidas por el sistema (Sommerville, 2011).

Se definieron 9 requisitos funcionales de los cuales tres de ellos son de complejidad alta, uno de complejidad media y cuatro de complejidad baja. También se definen cuatro requisitos no funcionales, para la conformación de la implementación.

Requisitos funcionales

Los requisitos funcionales (RF) de un sistema, son aquellos que describen cualquier actividad que este deba realizar, en otras palabras, el comportamiento o función particular de un sistema cuando se cumplen ciertas condiciones. Por lo general, estos deben incluir funciones desempeñadas por pantallas específicas, descripciones de los flujos de trabajo a ser desempeñados por el (Pressman, 2010). En la tabla 3 se prosigue con el listado de los requisitos funcionales.

Tabla 3. Listado de requisitos funcionales de la propuesta de solución

Fuente: Elaboración propia

Identificador	Requisito	Descripción	Prioridad
RF1	Listar configuración de red	Mostrar las interfaces de red disponibles y sus detalles (dirección Mac, dirección ipv4, ipv6, si está conectada o no, si está habilitada o no)	alta
RF2	Configurar red	Permite modificar las interfaces de red de manera estática o dinámica con los parámetros: dirección ip, la máscara de subred, la puerta de enlace y el servidor DNS.	alta
RF3	Configurar el DNS	Utilizar los programas que trae el servidor para establecer la configuración del DNS	alta
RF4	Mostrar la configuración de la red	Muestra la dirección Mac, dirección ip, de los clientes conectados al servidor.	media
RF5	Iniciar servicio de red	Inicia el servicio de red	baja
RF6	Detener servicio de red	Detiene el servicio de red	baja
RF7	Mostrar servidor de red	Mostrar el estado del servicio de red	baja
RF8	Mostrar la interfaz de red	Mostrar detalles del servicio de red del servidor	baja
RF9	Mostrar estado del servicio de red	Mostrar estado del servicio de red del servidor.	baja

Requisitos no funcionales

Los requisitos no funcionales representan características generales y restricciones de la aplicación o sistema que se esté desarrollando. Suelen presentar dificultades en su definición dado que su conformidad o no conformidad podría ser sujeto de libre interpretación, por lo cual es recomendable acompañar su definición con criterios de aceptación que se puedan medir

(Pressman, 2010). Los requisitos no funcionales de la propuesta de solución se listan en la Tabla 4, fueron identificados siguiendo la norma ISO 2500⁴.

Tabla 4. Listado de requisitos no funcionales de la propuesta de solución

Fuente: elaboración propia

Identificador	Clasificación	Requisito
RnF1	Usabilidad	La herramienta debe poseer interfaces gráficas bien formadas, en idioma español.
RnF2	Interoperabilidad	La herramienta para la administración remota de la configuración de red debe ser compatible con Nova Servidores.
RnF3	Integridad	La herramienta debe garantizar la integridad de los datos de la configuración de red mediante el protocolo SSH.
RnF4	Seguridad	La herramienta solo puede ser utilizada por usuarios administradores de red.
RnF5	Mantenibilidad	La herramienta debe estar codificada en el estándar definido, con el fin de facilitar el análisis para futuras modificaciones o para su mantenimiento.

2.2.4 Descripción de requisitos de software mediante Historias de Usuario:

Las Historias de Usuario (HU) son una forma rápida de administrar los requisitos de los usuarios sin tener que elaborar gran cantidad de documentos formales y sin requerir de mucho tiempo para administrarlos. Las historias de usuario permiten responder rápidamente a los requisitos cambiantes, por lo que una historia de usuario puede tener varios cambios a lo largo de un desarrollo sin afectarse el tiempo (Rouse, 2019). En la siguiente tabla se muestran la historia de usuario del RF2. Configurar red.

⁴ ISO (*International Organization for Standardization*, Organización Internacional de Normalización)

Tabla 5. Descripción de la historia de usuario de los RF2. Configurar red

Fuente: elaboración propia

Historia de Usuario	
Número: HU_2	Nombre del requisito: RF2. Configurar red
Programador: Betsy Alfonso Colás	Iteración Asignada: 1
Prioridad: Alta	Tiempo Estimado: 10 horas
Riesgo en Desarrollo: No aplica	Tiempo Real: 10 horas
<p>Descripción:</p> <p>La herramienta permite al usuario la configuración de la red tanto estática como dinámica y la selección de qué tipo de red va a utilizar cableada o inalámbrica. Para ello el administrador debe seleccionar la cual de ellas desea configurar. Se visualizarán la interfaz de red disponible en la herramienta.</p> <p>Dirección IP*: Es un campo de texto obligatorio. Los valores permitidos en este campo de texto son cuatro números entre 0 y 255 separados por puntos.</p> <p>Máscara de red: Es un campo de texto obligatorio. Los valores permitidos en este campo de texto son cuatro números entre 0 y 255 separados por puntos.</p> <p>Puerta de enlace*: Es un campo de texto obligatorio. Los valores permitidos en este campo de texto son cuatro números entre 0 y 255 separados por puntos.</p> <p>Servidores DNS*: Es un campo de texto obligatorio. Los valores permitidos en este campo de texto son cuatro números entre 0 y 255 separados por puntos. Se puede introducir más de una dirección de ip, en este caso, se deben separar las direcciones por espacios.</p> <p>Dominio: Es un campo de texto opcional que permite la entrada de un nombre de dominio, solo se permite la entrada de letras.</p>	
<p>Observaciones:</p> <p>Se utilizará el comando ip para configurar la red, ejemplo:</p> <pre>ip addr add 192.168.1.2/24 dev eth0</pre>	

ip route add default via 192.168.1.1

Si se dejan campos obligatorios vacíos se mostrará un mensaje de error indicando los campos que se dejaron vacíos.

Si se introduce algún parámetro en algún campo que no sea el permitido se mostrará un mensaje de error indicando los valores permitidos.

Al modificar la configuración de la red el sistema mostrará el mensaje de información: “Los cambios han sido aplicados satisfactoriamente.”.

Prototipo elemental de interfaz gráfica de usuario:

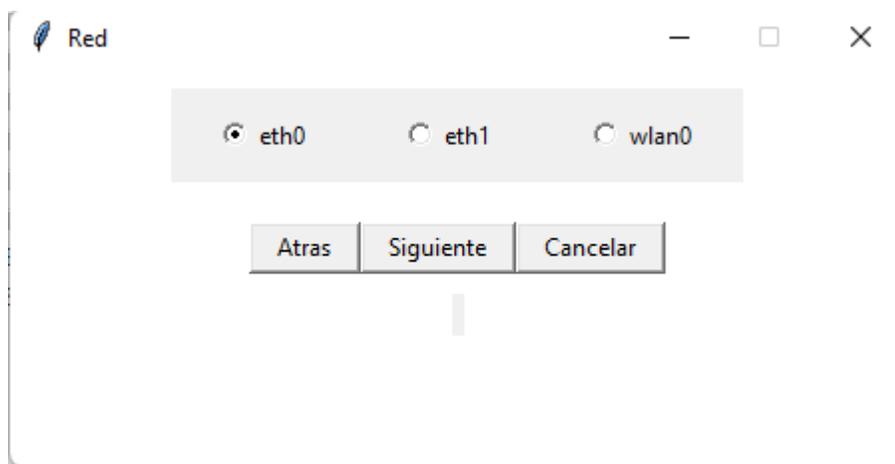


Figura 2. Seleccionar la opción: Configuración de la interfaz de red

Fuente: Elaboración propia

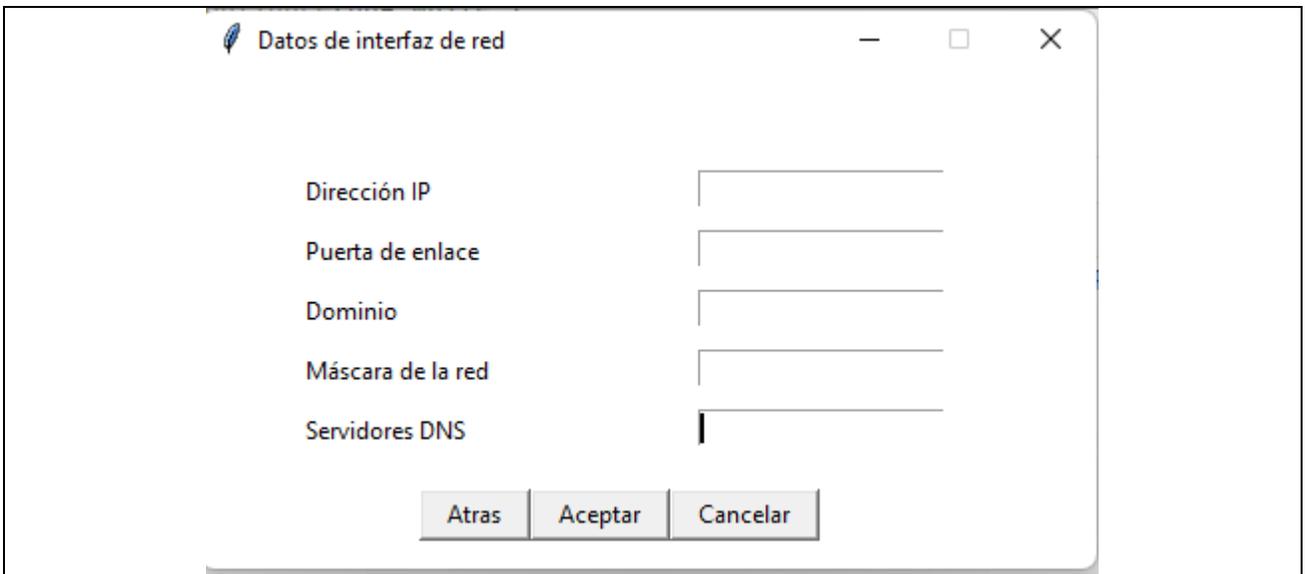


Figura 3. Configuración de la red estática.

Fuente: Elaboración propia

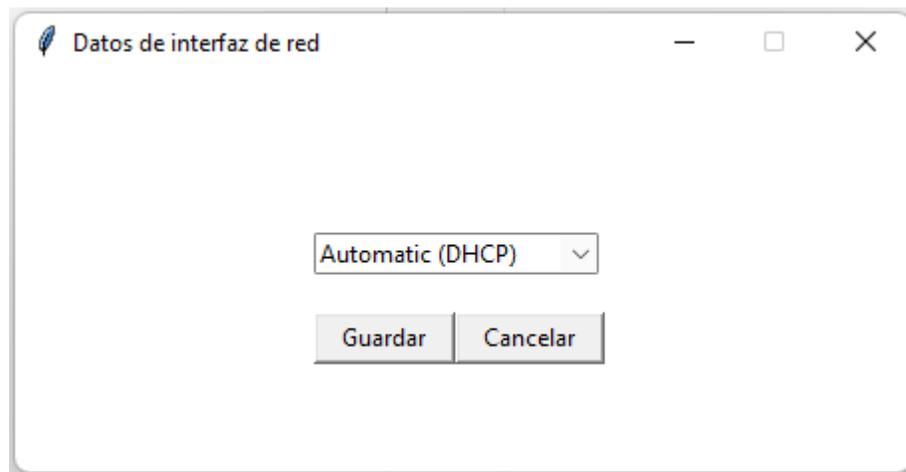


Figura 4. Configuración de red dinámica

Fuente: Elaboración propia

2.2.5 Validación de requisitos de software

Es muy importante asegurar la validez de los requisitos previamente a comenzar un desarrollo de software. Para ello debe de hacerse una comprobación de la correspondencia entre las descripciones iniciales y si el modelo es capaz de responder al planteamiento inicial. Para llevar a cabo esto, se suele realizar comprobando que el modelo obtenido responde de la misma forma

deseada que la que el cliente pide, por un lado, y por otro a la inversa si otras respuestas del modelo convencen al cliente. En algunos casos será necesario construir prototipos con una funcionalidad similar muy reducida para que el cliente se haga una idea aproximada del resultado (2019). La validación en el proyecto se determinará por notificación de errores, confirmación de edición de campo, validación de campos y tratamientos de errores de ejecución en el servidor.

Prototipado de interfaz de usuario

Los prototipos de interfaz de usuario pueden ser prototipos formales o informales, ejecutables o no ejecutables, de baja fidelidad o de alta fidelidad. Un prototipo de interfaz de usuario puede variar desde una serie de imágenes representando capturas de pantalla realizadas. El objetivo del prototipo de interfaz de usuario no es más que crear habilidades necesarias para la construcción del prototipo. Un prototipo de interfaz de usuario requiere habilidades de diseño de interfaz de usuario (Domínguez, 2018).

En el prototipado de interfaz de usuario se observan la configuración de red en las variantes que se pueden realizar, las selecciones de la red y la inserción de datos de la interfaz de red. Para ello se utilizó capturas de pantallas de la implementación de la herramienta.

Diseño de caso de prueba (DCP)

Los diseños de prueba se crean utilizando información de una variedad de productos de trabajo, incluidos los productos de trabajo de diseño como las realizaciones de guiones de uso, los modelos de diseño o las interfaces clasificadoras. Las pruebas se ejecutan después de crear los componentes. Es típico crear los diseños de prueba justo antes de que se vayan a ejecutar las pruebas, bastante después de que se hayan creado los productos de trabajo de diseño de software (Cillero, 2020).

La generación de casos de prueba permitió especificar los posibles datos de entrada, así como las salidas de los requisitos funcionales de la herramienta para la configuración de red. A continuación, se presenta el DCP que corresponde al RF 2. Configurar red.

Caso de prueba correspondiente al RF 2. Configurar red

Descripción general: La herramienta debe ser capaz de configurar los parámetros de la red.

Precondiciones: Se debe seleccionar la interfaz de red a configurar.

Tabla 6. Diseño de caso de prueba correspondiente al RF2 Configurar red

Fuente: Elaboración propia

Escenario	Variables	Descripción	Respuesta del sistema	Flujo central

<p>EC1. Configurar la red con todos los parámetros correctos</p>	<p>Dirección IP V: 10.53.3.109 Máscara V: 255.255.255.0 Puerta de enlace V: 10.53.3.254 Dominio V: 10.53.3.112 DNS V: 10.53.3.11</p>	<p>Permite configurar la red con todos los parámetros correctos</p>	<p>La herramienta configura la interfaz de red</p>	<p>Insertar los datos y realizar clic en Aceptar</p>
<p>EC1.2 Configurar la red con parámetros incorrectos</p>	<p>Dirección IP I: v.53.3.109 Máscara V: 255.255.255.0 Puerta de enlace I: 10.*.3.254 Dominio V: 10.53.3.112 DNS V: 10.53.3.11</p>	<p>No permite configurar la red con parámetros incorrectos</p>	<p>La herramienta muestra un mensaje de error</p>	<p>Insertar los datos y realizar clic en Aceptar</p>
<p>EC1.3 Configurar la red con parámetros vacíos</p>	<p>Dirección IP V: 10.53.3.109 Máscara I: . . .0 Puerta de enlace V: 10.53.3.254 Dominio V: 10.53.3.112 DNS V: 10.53.3.11</p>	<p>No permite configurar la red con parámetros vacíos</p>	<p>La herramienta muestra un mensaje de error</p>	<p>Insertar los datos y realizar clic en Aceptar</p>
<p>EC1.4 Cancelar</p>		<p>Cancela la operación</p>	<p>Cierra la interfaz gráfica de usuario</p>	<p>Realizar clic en Cancelar</p>

2.3 Análisis y diseño

En esta disciplina los requisitos pueden ser refinados y estructurados para conseguir una comprensión más precisa y una descripción que sea fácil de mantener y ayude a la estructuración del sistema incluyendo la arquitectura. En esta disciplina se modela la herramienta y su forma para que soporte todos los requisitos, incluyendo los requisitos no funcionales (Mestras, 2007).

2.3.1 Diseño de clases

El diseño consiste en ser fácil de entender, mantenible, reutilizable, hay principios de diseño que ayudan en el diseño de las clases y la estructura del programa, diseño dirigido por responsabilidades, acoplamiento, cohesión y refactorización (Mestras, 2007).

Diagrama de clases del diseño

El diagrama de clases recoge las clases de objetos y sus asociaciones. En este diagrama se representa la estructura de cada uno de los objetos del sistema y sus relaciones con los demás objetos. Representa claramente los detalles de las clases de software y de las interfaces; en él se exponen una colección de, interfaces, dependencias, relaciones, clases, y métodos. (Cillero, 2020). El siguiente diagrama es la representación de las clases que intervienen en el proceso de negocio.

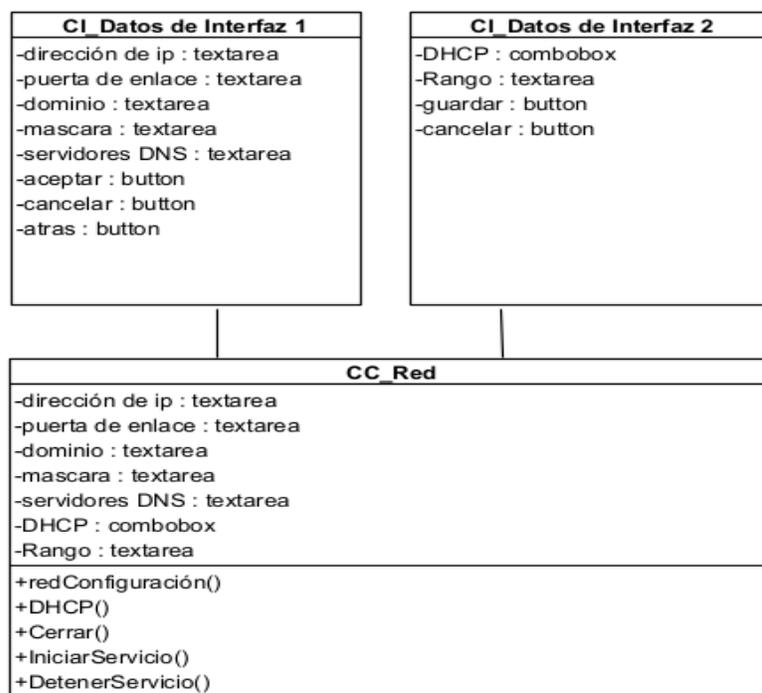


Figura 5. Diagrama de clases del diseño de la propuesta de solución

Fuente: Elaboración propia

Patrones de diseño de software

Los patrones de diseño son soluciones a problemas recurrentes de diseño y que se están aplicando a diario en la industria del software (Ferrer, 2020).

Patrones GRASP

Los Patrones Generales de Software para Asignar Responsabilidades (*General Responsibility Assignment Software Patterns*) se emplean en el diseño de las clases de un programa informático. La importancia de captar estos principios permite diseñar eficazmente el software orientado a objetos. Constituyen un apoyo para la asignación de responsabilidades; permiten entender el diseño de objetos y aplica el razonamiento entre estos patrones se encuentran el experto, el creador, el controlador, bajo acoplamiento, alta cohesión (Rojas, 2004).

Experto: asigna 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 (Rcasalla, 2021). En la clase *CC_Red()* se emplea a través de los métodos *redConfiguracion()* y *DHCP()*, que contiene la información necesaria para configurar la red, enviada mediante el protocolo SSH a Nova Servidores.

Creador: ayuda a identificar qué clase debe ser responsable de la creación o instanciación de nuevos objetos o clases (Rcasalla, 2021). Este patrón se emplea en la clase *CC_Red()*, responsable de la creación de las instancias de la configuración de red.

Controlador: asigna la responsabilidad de gestionar mensajes en el sistema a una clase que represente al sistema global, a un subsistema, manejador de sesión o de eventos de un escenario de caso de uso (Rcasalla, 2021). Este patrón se evidencia en la clase *CC_Red()*, responsable del flujo de parámetros de configuración de la red.

Alta cohesión: asegura la necesidad de mantener el sistema con un bajo nivel de complejidad, asegura que se realicen solo las operaciones necesarias para cumplir una tarea dentro de un área funcional. Este patrón queda evidenciado en cada una de las clases del diseño, ya que cada una responde con las funcionalidades que debe cumplir para realizar los trabajos correspondientes a cada clase (Rcasalla, 2021). El uso de este patrón se manifiesta en las clases *CI_Datos_de_Interfaz_1()*, *CI_Datos_de_Interfaz_2()* y *CC_Red()*.

Bajo Acoplamiento: es la medida de cuánto una clase está conectada con otras clases. Un alto acoplamiento en un sistema es problemático porque, debido a la cantidad de relaciones entre las clases dificulta el mantenimiento en sistema muy acoplado, ya que los cambios realizados en una

clase afecta a otras (Rcasalla, 2021). El uso de este patrón se manifiesta en las clases *CI_Datos_de_Interfaz_1()*, *CI_Datos_de_Interfaz_2()* y *CC_Red()*.

2.3.2 Diseño arquitectónico

El diseño arquitectónico garantiza cómo debe organizarse un sistema y cómo tiene que diseñarse la estructura global del mismo. En el modelo del proceso de desarrollo de software, el diseño arquitectónico es la primera etapa en el proceso de diseño del software. Es el enlace crucial entre el diseño y la ingeniería de requerimientos, ya que identifica los principales componentes estructurales en un sistema y la relación entre ellos (Sommerville, 2011). En el diseño arquitectónico de la propuesta de solución se empleó el estilo en capas.

Arquitectura en capas

La arquitectura en capas consta en dividir la aplicación en capas, con la intención de que cada capa tenga un rol muy definido, como podría ser, una capa de presentación, una capa de reglas de negocio y una capa de acceso a datos, sin embargo, este estilo arquitectónico no define cuantas capas debe de tener la aplicación, sino que se centra en la separación de la aplicación en capas; aplicando el principio Separación de preocupaciones (SoC). En la práctica, la mayoría de las veces este estilo arquitectónico es implementado en 4 capas, presentación, negocio, persistencia y base de datos, sin embargo, es habitual ver que la capa de negocio y persistencia se combinan en una solo capa, sobre todo cuando la lógica de persistencia está incrustada dentro de la capa de negocio (Blancarte, 2020). En la figura 6 se muestra el diseño arquitectónico elaborado para la propuesta de solución.

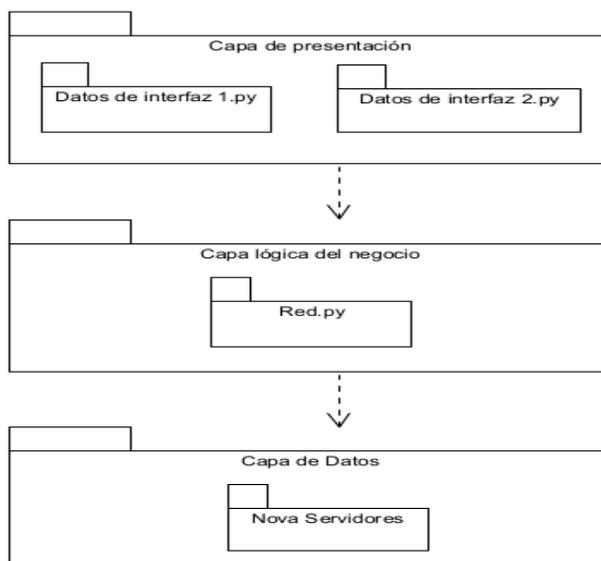


Figura 6. Diseño arquitectónico de la propuesta de solución basado en capas

Fuente: Elaboración propia

- En la **Capa de presentación** están las interfaces gráficas de usuario: *Datos de interfaz 1.py* y *Datos de interfaz 2.py*, estas permiten que el administrador de red pueda administrar los parámetros necesarios para configurar la red en Nova Servidores.
- En la **Capa lógica del negocio** está la clase controladora *Red.py*, que contiene las funcionalidades necesarias para controlar el flujo de parámetros de configuración de red entre las vistas y Nova Servidores.
- La **Capa de Datos** está compuesta por *Nova Servidores* que es donde se realiza la persistencia de los parámetros de configuración de la red.

2.3.3 Modelado de datos

El modelo de datos se utiliza para describir la estructura lógica y física de la información persistente gestionada por el sistema. Se utiliza para definir la correlación entre las clases de diseño persistentes y las estructuras de datos persistentes, y para definir las estructuras de datos persistentes (Tecnologías-Información, 2018). En la propuesta de solución la persistencia de los datos se realiza en el fichero de configuración “*interfaces*”, ubicado en la dirección “*etc/network*” del sistema de archivos de Nova Servidores. En la tabla 6 se presenta la estructura de este fichero de configuración.

Tabla 7. Estructura del fichero de configuración de los parámetros de la red en Nova Servidores

Fuente: Elaboración propia

Parámetros	Estructura de persistencia
Interfaz de red	<i>auto eth1</i> <i>iface eth1 inet static</i>
Dirección IP	<i>address</i>
Máscara	<i>netmask</i>
Puerta de enlace	<i>gateway</i>
DNS	<i>dns-nameservers</i>

2.4 Conclusiones parciales

El análisis y diseño de la herramienta para la administración remota de la configuración de red en Nova Servidores permitió identificar 9 requisitos funcionales y 5 no funcionales, de acuerdo a las necesidades del cliente y las características de la propuesta de solución. El empleo del estilo arquitectónico en capas posibilitó definir la organización y la comunicación entre los diferentes

componentes de la propuesta de solución, garantizando la integridad de los datos. El diseño de clases propició modelar las diferentes clases, sus atributos y relaciones en el contexto del negocio informatizado y en correspondencia con las restricciones del diseño.

CAPITULO 3: Implementación y evaluación de la herramienta informática para la administración remota de la configuración de red en Nova Servidores

Introducción

El capítulo contiene los productos de trabajos elaborados en las disciplinas de Implementación y Pruebas. En él se presenta los estándares de codificación utilizados durante la implementación de la solución; así como los diagramas de componentes y despliegue, y un ejemplo de la interfaz gráfica de usuario implementada. Contiene las pruebas de software ejecutados y los resultados obtenidos.

3.1 Implementación

La implementación constituye la realización de determinados procesos y estructuras en un sistema. Representa así la capa más baja en el proceso de paso de una capa abstracta a una capa más concreta (Hernández, 2016). Durante la implementación se utilizaron los estándares de codificación descritos a continuación que permitirá que el programador mejore tanto en legibilidad y testabilidad del código de la herramienta implementada.

3.1.1 Estándares de codificación

Se define como la forma de escribir código es propia de cada programador y completamente diferente a la forma de cualquier otro. De la forma usada depende la facilidad para entender el código y retomar ciertas partes realizadas por otros integrantes, así como la depuración de las mismas (Marin, 2003).

Indentación: Se utilizan 4 espacios sin caracteres de tabulación o espacios en blanco.

Tabuladores o Espacios: Las estructuras de control deben tener un espacio entre la palabra reservada de la estructura y el signo de apertura de paréntesis para distinguir entre las llamadas de las funciones. Las funciones son llamadas sin espacio entre el nombre de la función, el signo del paréntesis y el primer parámetro; espacios entre cada coma por parámetro y sin espacios entre el último paréntesis, el signo de paréntesis cerrado y el signo de punto y coma (;).

Estándares de nomenclatura: se utilizó *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 de las clases, las variables y funciones.

En la figura 7 se presenta un fragmento del código de la clase controladora, donde se evidencia el empleo de los estándares de codificación:

```

buttonsFrame = Frame(raiz)
buttonsFrame.place(x=80, y=120)

```

Lower Camel Case

```

def dhcp()
    if str(combo.get()).find('Automatic') >= 0:
        comp_process = subprocess.run('ifconfig -a', stdout=subprocess.PIPE,
                                     stderr=subprocess.PIPE)

```

Indentación

Figura 7. Ejemplo de la aplicación de los estándares de codificación

Fuente: Elaboración propia

3.1.2 Diagrama de componentes

El diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software, sean éstos de código fuente, binarios o ejecutables. Muestra las clases del diseño del software y las relaciones existentes entre ellos en términos de componentes o unidades físicas de implementación con interfaces bien definidas y pensada para ser utilizada como parte reemplazable de un sistema informático (UNAD, 2021). En la Figura 8 se presenta el diagrama de componentes de la propuesta de solución. Contiene una representación de las diferentes clases de la herramienta para la administración de la configuración de red vistas como componentes.

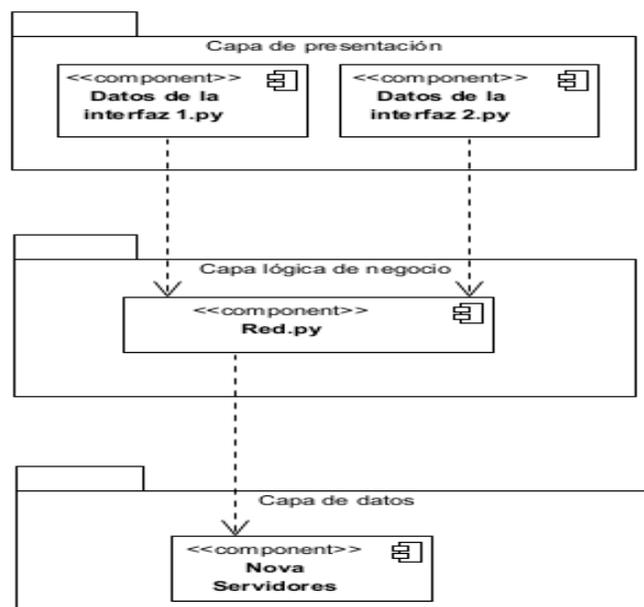


Figura 8. Diagrama de componente de la propuesta de solución

Fuente: Elaboración propia

Los componentes representados en el diagrama anterior son:

- Los componentes *Datos de interfaz 1.py* y *Datos de interfaz 2.py* contenidos en el paquete de Capa de presentación representan las interfaces gráficas de usuario que permiten la comunicación de los administradores de red con la herramienta desarrollada.
- El componente *Red.py* incluido en el paquete Capa lógica del negocio representa la clase controladora que permite el flujo seguro de los parámetros de la configuración de la red (introducidos por los administradores de red) hasta Nova Servidores que es donde se realiza la configuración.
- El componente *Nova Servidores* implícito en paquete en la Capa de datos representa la variante de la distribución cubana GNU/Linux Nova para servidores que es donde se efectúa la configuración de la red.

3.1.3 Diagrama de despliegue

El diagrama de despliegue se utiliza para modelar los nodos físicos (componentes de hardware) y sus relaciones, así como los componentes (software) que contenido en los nodos. Permite modelar aspectos de hardware de un sistema informático con el nivel de detalle adecuado para que se pueda especificar la plataforma sobre la que se ejecutará el software (Serna, 2011). El diagrama de despliegue presentado en la Figura 9, muestra los nodos (recursos de hardware necesarios para la ejecución de la herramienta para la administración de la configuración de la red en Nova Servidores), así como los componentes asociados a cada uno de ellos y las relaciones de comunicación existentes entre los nodos a través de protocolos.

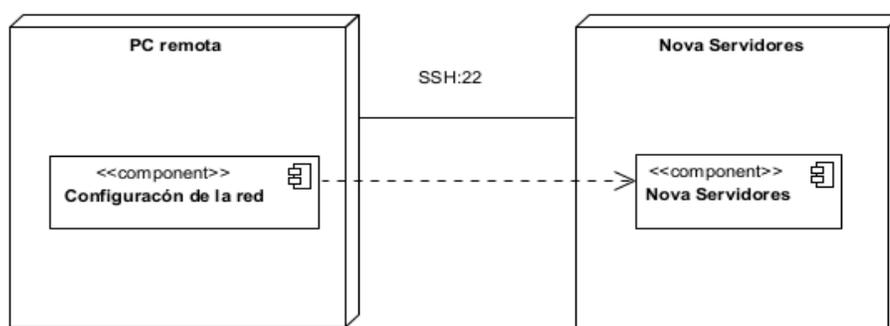


Figura 9. Diagrama de despliegue de la configuración de red.

Fuente: Elaboración propia

Los nodos que integran el diagrama de despliegue son:

- **PC remota:** contiene el sistema operativo GNU/Linux mediante el cual los administradores de red pueden configurar la red en Nova Servidores

- **Nova Servidores:** contiene la distribución cubana GNU/Linux en su variante para servidores (Nova Servidores) es donde se ejecuta la configuración de la red.

El protocolo que permite la relación entre los nodos es:

- **SSH:** permite la conexión segura desde la PC remota a Nova Servidores mediante el puerto 22, garantizando la integridad de los parámetros de la configuración de red.

3.1.4 Interfaz gráfica de usuario

Una interfaz gráfica de usuario (GUI) es una interfaz que se puede utilizar para controlar una computadora, tabletas y otros dispositivos. Utilizan elementos gráficos como iconos, menús e imágenes para facilitar la interacción del usuario. Tanto los sistemas operativos como las aplicaciones utilizan una interfaz gráfica de usuario. De hecho, casi todos los programas para usuarios finales de hoy en día vienen con esta interfaz. En el diseño y la implementación de las interfaces gráficas de la herramienta para la administración remota de la configuración de red en Nova Servidores se emplearon las Reglas de oro propuestas por (Pressman, 2010). A continuación, se relacionan estas reglas.

Reglas de oro:

1. Dar control al usuario
2. Reducir la carga de memoria del usuario
3. Construcción de una interfaz consistente

En la Figura 10 se muestra la interfaz gráfica de usuario para la administración de red en la herramienta desarrollada.

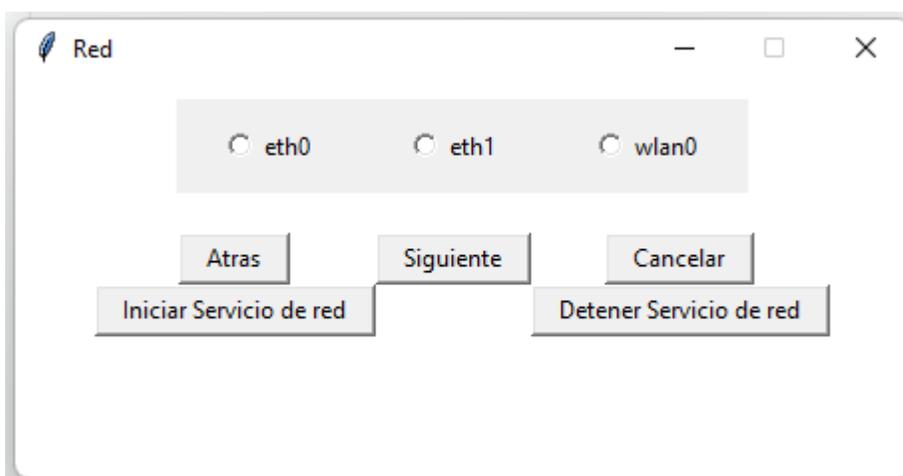


Figura 10. Interfaz gráfica de usuario para la administración de red

Fuente: Elaboración propia

El desarrollo de la disciplina Implementación permitió la codificación del diseño elaborado para solucionar el problema de la investigación. Posteriormente se ejecutaron las pruebas de software para verificar el cumplimiento de las funcionalidades y las características de la propuesta de solución. En el epígrafe 3.2 se evidencian las pruebas de software aplicadas.

3.2 Pruebas de software

Las pruebas de software son un conjunto de acciones que pueden ser planificadas con antelación y ejecutarse sistemáticamente durante la implementación o al finalizar el desarrollo del software. Estas comprenden un conjunto de actividades que se realizan para identificar posibles fallos de funcionamiento, configuración o usabilidad de un programa o aplicación. Las pruebas de software se ejecutan a partir de la aplicación de métodos y técnicas (Pressman, 2010). Para la evaluación de la herramienta propuesta se definieron las pruebas a partir de las disciplinas establecidas por la metodología AUP-UCI, descritas a continuación.

3.2.1 Tipos de pruebas de software

Pruebas Unitarias

Las pruebas a nivel de unidad se concentran en el esfuerzo de verificación de la unidad más pequeña del diseño, el componente o módulo de software. Tomando como guía la descripción del diseño a nivel de componente, se prueban importantes caminos de control para describir errores dentro de los límites de la herramienta. El restringido que se ha determinado para las pruebas de unidad limita la relativa complejidad de las pruebas y los errores que éstas revelan (Pressman, 2010). En el caso de la presente investigación en este nivel de prueba se decide aplicar los métodos de caja blanca y caja negra.

Pruebas Funcionales

Las pruebas funcionales se centran en comprobar que el sistema funcione acorde a los requisitos funcionales y no funcionales, detectando posibles defectos derivados de errores en la fase de programación; dichas pruebas se llevan a cabo a través de la interfaz gráfica de la aplicación. Mediante estas pruebas se demuestra que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce una salida correcta, además de mantener íntegra la información externa del sistema (Pressman, 2010).

3.2.2 Métodos de prueba

Método de caja blanca

El método de caja blanca posibilita el desarrollo de casos de prueba que garanticen la ejecución, al

menos una vez, de los caminos independientes (Pressman, 2010). En el caso del presente trabajo de investigación el método fue ejecutado aplicando la técnica de camino básico.

Método de caja negra

El método de caja negra se aplica sobre la interfaz del software, obviando el comportamiento interno y la estructura del programa. A través de estas pruebas se pretende demostrar que las funcionalidades del software son operativas; las entradas son aceptadas de manera correcta (Ojeda, 2014). En la investigación se empleó este método aplicando la técnica partición de equivalencias.

3.2.3 Técnicas de prueba

Camino básico

La técnica del camino básico tiene como objetivo comprobar que cada camino se ejecute de manera independiente de un componente o programa, obteniéndose una medida de la complejidad lógica del diseño. Esta técnica debe ser utilizada para evaluar la efectividad de los métodos asociados a una clase, al tratar de confirmar que cada camino independiente sea ejecutado al menos una vez en el sistema (Sommerville, 2011).

Partición de equivalencia

La técnica de partición de equivalencia consiste en probar cada una de las funcionalidades del sistema, mediante la definición de casos de prueba que sean capaces de encontrar diferentes tipos de errores a nivel de interfaz. Es un conjunto de estados válidos y no válidos para las condiciones de entrada. Por lo general, una condición de entrada es un valor numérico específico, un rango de valores, un conjunto de valores relacionados o una condición booleana (Pressman, 2010).

3.3 Aplicación de las pruebas de software

En el epígrafe se presentaron los resultados alcanzados de las disímiles pruebas de software anteriormente determinadas y aplicadas utilizando los métodos y técnicas descritas durante la realización de las disciplinas Pruebas internas y Pruebas de aceptación propuestas por la metodología de desarrollo de software variación AUP para la UCI.

3.3.1 Pruebas Internas

En esta disciplina se verifica el resultado de la implementación probando cada construcción, incluyendo tanto las construcciones internas como intermedias, así como las versiones finales a

ser liberadas (Rodríguez, 2015). Durante esta disciplina se aplicaron pruebas unitarias, funcionales y de integración, descritas a continuación.

Pruebas unitarias

Las pruebas unitarias se realizaron a través del método de prueba caja blanca y la técnica del 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. En la figura 11 se presenta el método `redConfig()` correspondiente al RF2 Configurar red.

```

def redConfig():
1  if red == 1:      2
3  param = 'eth0'
   elif red == 2:   4
5  param = 'eth1'
   else:
6  param = 'wlan0'
   comp_process = subprocess.run(f'sudo ifconfig {param} {str(entry.get())}', stdout=subprocess.PIPE,
7  with open('result.txt', 'w+') as result:
   result.write(str(comp_process.stdout))
8  cerrar2()

```

Figura 11. Método `redConfig()` correspondiente al RF2. Configurar red

Fuente: Elaboración propia

1. Dibujar el grafo de flujo de la funcionalidad

En la Figura 12 se utilizó la notación de grafo de flujo para representar el código de la funcionalidad probada:

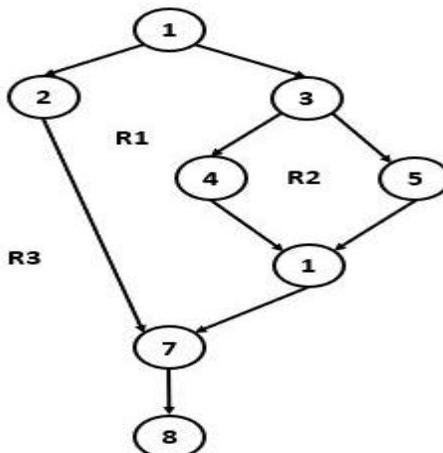


Figura 12. Grafo de flujo

Fuente: Elaboración propia

2. Determinar la complejidad ciclomática

La complejidad ciclomática de un grafo $V(G)$ se puede calcular de tres formas diferentes:

$V(G) = A - N + 2$ Donde A es el número de aristas del grafo de flujo y N es la cantidad de

$V(G) = 9 - 8 + 2$ nodos del grafo

$V(G) = 3$

$V(G) = P + 1$ Donde P es el número de nodos predicados (nodos con más de una

$V(G) = 2 + 1$ arista de salida) contenidos en el grafo

$V(G) = 3$

$V(G) = R$ Donde R es el número de regiones (áreas delimitadas por nodos y aristas

$V(G) = 3$ en el grafo)

3. Determinar los caminos linealmente dependientes

Camino básico 1: 1, 2, 7, 8

Camino básico 2: 1, 3, 4, 6, 7, 8

Camino básico 3: 1, 3, 5, 6, 7, 8

4. Definir los casos de prueba para comprobar la ejecución

En el diseño de los casos de prueba se debe especificar los siguientes elementos:

- Descripción: contiene una descripción sobre las restricciones de los datos de entrada que debe tener el caso de prueba.
- Condición de ejecución: se especifican los parámetros que debe poseer el caso de prueba para que se cumpla una condición deseada como respuesta del funcionamiento del método.
- Entrada: se muestran los parámetros de entradas del método.
- Resultados esperados: se explica el resultado esperado de la ejecución del método.

En la Tabla 8 se presenta el diseño de caso de prueba del camino 2 del conjunto de caminos básicos linealmente independientes correspondientes a la funcionalidad *redConfig()*:

Tabla 8. Caso de prueba para el camino 2 del método redConfig()

Fuente: Elaboración propia

Diseño de caso de prueba para el camino 2	
Descripción	Método para seleccionar el tipo de interfaz de red a modificar
Condiciones	El usuario selecciona la interfaz de red a modificar
Entradas	param = 'eth1'
Resultados	Permite seleccionar que tipo de interfaz red se va configurar y muestra el resultado de la selección

La ejecución de las pruebas unitarias permitió conocer que el flujo de trabajo de las funcionalidades de la herramienta para la configuración de la red es correcto; pues se comprobó que las sentencias de código fuente se ejecuta al menos una vez.

Pruebas funcionales

En las pruebas funcionales se tuvieron 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 prueba diseñados en el epígrafe 2.2.5 “Validación de requisitos de software”. En la Figura 13 se presenta un gráfico con los resultados de las pruebas funcionales realizadas.

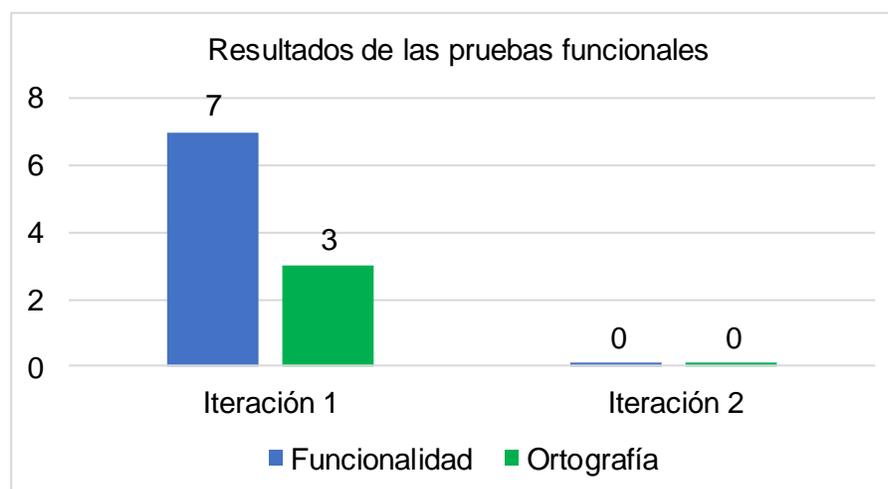


Figura 13. Resultados de las pruebas funcionales

Fuente: Elaboración propia

La ejecución de las pruebas funcionales permitió identificar y corregir las deficiencias presentes en la herramienta para la administración remota de la configuración de red en Nova Servidores. Garantizó la obtención de un producto que responde al problema de investigación con la calidad requerida.

3.4 Conclusiones parciales

Se logró la informatización de la administración remota de la configuración de red en Nova Servidores, mediante la herramienta desarrollada, cumpliendo con los estándares de codificación definidos. La aplicación de las pruebas unitarias permitió comprobar que el flujo de las funcionalidades implementadas es el correcto, pues cada sentencia de código se ejecuta al menos una vez. La ejecución de las pruebas funcionales posibilitó identificar las deficiencias de la propuesta de solución y garantizar la calidad y el cumplimiento de los requisitos definidos.

CONCLUSIONES GENERALES

La presente investigación evidenció el proceso de desarrollo de software ejecutado para la obtención de la herramienta informática que permite la administración remota de la configuración de red en Nova Servidores. Una vez finalizada la misma se puede arribar a las siguientes conclusiones:

- La elaboración del marco teórico sobre el proceso de configuración de red en sistemas operativos GNU/Linux, así como el estudio de herramientas informáticas para la administración remota de la configuración de la red, demostraron la necesidad de desarrollar una solución que mejore la administración de la configuración de la red en Nova Servidores.
- Los 9 requisitos funcionales y los 5 no funcionales definidos permitieron cumplir con las necesidades del cliente. El diseño propició el modelado de una herramienta para la administración remota de la configuración de la red en Nova Servidores. El uso de los patrones de diseño GRASP y el estilo arquitectónico brindó una mayor calidad al software desarrollado.
- Se implementó una herramienta para la administración remota de la configuración de la red en Nova Servidores permitiendo cumplir con las exigencias del cliente y las restricciones del diseño elaborado.
- La aplicación de las pruebas de software posibilitó la evaluación de la solución desarrollada y garantizar el correcto funcionamiento la herramienta implementada. Permitted comprobar que se satisfacen las deficiencias planteadas en la situación problemática y se mejora el proceso de configuración de red en Nova Servidores.

RECOMENDACIONES

- Integrar la herramienta desarrollada a NovaRSAT (plataforma para la administración remota de Nova Servidores).

REFERENCIAS BIBLIOGRÁFICAS

Álvares, Sara. 2017. Qué es un diccionario de datos. *Desarrolloweb.com*. [En línea] 2017. <https://desarrolloweb.com/faq/452.php>.

Araneda, Patricio. 2021. *Base de Datos el camino de los datos a la información*. Chile : s.n., 2021.

Blancarte, Oscar. 2020. *Introducción a la arquitectura de software. Un enfoque práctico*. 2020.

Carballeira, Félix García. 2021. *Sistemas Operativos: Una Visión Aplicada. 3 Ed.: Volumen II, Volumen2*. s.l. : Independently Published, 2021, 2021. 9798599338956.

Cillero, Manuel. 2021. Diagrama de Clases. *manuel.cillero.es*. [En línea] 2021. <https://manuel.cillero.es/doc/metodologia/metrica-3/tecnicas/diagrama-de-clases/>.

Debian . 2021. Gestión de la configuración de red. *Computerweekly.es*. [En línea] 2021. <https://www.computerweekly.com/es/definicion/Gestion-de-la-configuracion-de-red-NCM>.

Debian. 2021. Capítulo 5. Configuración de red. *Debian*. [En línea] 9 de enero de 2021. <https://www.debian.org/doc/manuals/debian-reference/ch05.es.html>.

Domínguez, Dorado M. 2018. *Todo Programación. Nº 13. Págs. 32-34*. s.l. : Editorial Iberprensa (Madrid), 2018.

Donostia-San Sebastián. 2021. implika.es. *Implika*. [En línea] Donostia-San Sebastián, 14 de abril de 2021. <https://www.implika.es/blog/que-son-redes-informaticas>. 900 264 650.

Fernandez, Juio Cesar. 2019. Applesfera. *Así es Electron: el último intento del desarrollo híbrido en una carrera que ya han ganado las apps nativas*. [En línea] 23 de Enero de 2019. <https://www.applesfera.com/analisis/asi-electron-ultimo-intento-desarrollo-hibrido-carrera-que-han-ganado-apps-nativas>.

Ferrer, Raúl. 2020. raulferrergarcia. *raulferrergarcia*. [En línea] 1 de 2 de 2020. [Citado el: 11 de 11 de 2021.] <https://www.raulferrergarcia.com/patrones-de-diseno-en-programacion/>.

Gómez, Daniel Desiderio Borrego. 2016. *TIC y herramientas digitales*. Mexico : Palibrio, 2016, 2016. 1506518028, 9781506518022.

Guerra, César Arturo. 2015. sg. sg. [En línea] 2015. <https://sg.com.mx/revista/17/obtencion-requerimientos-tecnicas-y-estrategia>.

Hernández, Walfredo González. 2016. redalyc. *redalyc*. [En línea] 10 de Mayo de 2016. [Citado el: 11 de 11 de 2021.] <https://www.redalyc.org/journal/4768/476852098003/html/>.

Lainez, Juan Carlos Rodríguez. 2016. *Implementación de un Sistema de Administración Web para la Indexación de la Revista Ciencias Pedagógicas e Innovación de la UPSE. Tesis de grado previo a la obtención del título Ingeniero en Sistemas.* Ecuador : Universidad Estatal Península Santa Elena, 2016.

Linux. 2017. cambiatealinux. *cambiatealinux.* [En línea] linux, 29 de enero de 2017. <https://cambiatealinux.com/configurar-una-red-manualmente-desde-consola>.

Linux-Ubunto. 2016. ayudalinux. *ayudalinux.* [En línea] Copyright 2016 - 2019,, 2 de noviembre, de 2016. <https://ayudalinux.com/author/migue/>.

Marin, David. 2003. aspl. *aspl.* [En línea] 05 de 05 de 2003. [Citado el: 11 de 11 de 2021.] <http://www.aspl.es/fact/files/aspl-fact/estandares-node2.html>.

McCarthy, Eamon. 2021. Gestión de la configuración de red (NCM). *ComputerWeekly.es.* [En línea] 2021. <https://www.computerweekly.com/es/definicion/Gestion-de-la-configuracion-de-red-NCM>.

Mendez. 2018. uniciencista.gfrodriguez.online. *uniciencista.gfrodriguez.online.* [En línea] 10 de febrero de 2018. <http://uniciencista.gfrodriguez.online/2018/02/herramientas-case-principales-usos.html>.

Mestras, Juan Pavón. 2007. fdi.ucm. *fdi.ucm.* [En línea] 08 de 2007. [Citado el: 11 de 11 de 2021.] <https://www.fdi.ucm.es/profesor/jpavon/poo/1.7.Diseno%20de%20clases.pdf>.

Ojeda, Oliver. 2014. Técnicas de pruebas. [En línea] 2014. [Citado el: 23 de marzo de 2019.] <https://es.slideshare.net/catalinocordero/unidad-10-tecnicas-de-pruebas>.

Oliveros, Alejandro. 2004. *Fuentes y técnicas para elicitación de requerimientos.* . Mexico : Departamento de Ciencia y Tecnología, UNTREF , 2004.

Perez, Juan Carlos Moreno. 2020. *Entornos de desarrollo.* España : Síntesis,S.A., 2020.

PRESSMAN, ROGER. 2010. *Ingeniería de software. Un enfoque práctico.* séptima edición. s.l. : McGraw-Hill, 2010. 978-0-07-337597-7.

Pressman, Roger S. 2010. *Ingeniería de Software. Un enfoque práctico.* Septima . México DF : McGraw-Hill, 2010. 978-0-07-337597-7.

Python. 2021. Python. [En línea] 3 de mayo de 2021. <https://docs.python.org/es/3/library/dialog.html#module-tkinter.simpledialog>.

- Rand, Paul. 2015.** *EL DISEÑO ES SIMPLE POR ESO ES TAN COMPLICADO.* MEXICO : Universidad Autónoma del Estado de Hidalgo, 2015. http://cidecame.uaeh.edu.mx/lcc/mapa/PROYECTO/libro10/222_modelo_conceptual_de_uml.html.
- Rcasalla. 2021.** Libro Desarrollo de software. Patrones básicos de Asignación de responsabilidades. [En línea] 2021. https://rcasalla.gitbooks.io/libro-desarrollo-de-software/content/libro/temas/t_disenio/di_grasp.html.
- Rodríguez, Tamara. 2015.** *Metodología de desarrollo para la actividad productiva de la UCI.* La Habana : s.n., 2015.
- Rojas, Oscar Andrés Sandoval Carlos y Wilson Andrés Castillo. 2004.** *researchgate.* 2004.
- Rouse, Margaret. 2019.** TechTarget. *User Story.* [En línea] 2019. <https://searchsoftwarequality.techtarget.com/definition/user-story>.
- Ruiz, P. 2020.** somebooks. *somebooks.* [En línea] 14 de 12 de 2020. <http://somebooks.es/copias-de-seguridad-integradas-en-ubuntu-20-04-lts-parte-i/>.
- Sanchez, Jose Manuel. 2015.** *prueba de software.Fundametos y Tecnica.* madrid : s.n., 2015.
- Serna, S. 2011.** *Especificación formal de requisitos temporales no funcionales, Tesis para optar el título de Magister en Ingeniería de Sistemas.* s.l. : Universidad Nacional de Colombia, 2011.
- Sommerville. 2011.** *Ingeniería de Software, 9na Edición.* s.l. : Person Education Inc, 2011.
- ubuntu. 2020.** adaweb.es. *adaweb.* [En línea] ubuntu, 4 de enero de 2020. <https://www.adaweb.es/servidor-ssh-en-ubuntu/>.
- UNAD. 2021.** *Lección 29. Diagramas de componentes.* s.l. : Universidad Nacional Abierta y a Distancia, 2021.
- Vargas, David Hernan Ballen. 2015.** *Trabajo de Grado para optar al título de Especialista en auditoría de sistemas de información.* Bogotá : UNIVERSIDAD CATÓLICA DE COLOMBIA, 2015.
- Velásquez, Luis. 2018.** Integraciones tecnológicas . *cinusual.com.* [En línea] Integraciones tecnológicas , 2 de Agosto de 2018. <https://cinusual.com/que-es-un-software-de-administracion-remota.20602824889>.
- Visual Paradigm Internationals. 2018.** Visual Paradigm. [En línea] 2018. <http://www.visual-paradigm.com/>.
- Vue.js. 2021.** Vue.js. *¿Qué es Vue.js?* [En línea] 2021. <https://es.vuejs.org/v2/guide/>.

ANEXOS

Anexo 1: Entrevista realizada a 2 especialistas de CESOL relacionados con el proceso de migración hacia tecnologías libres y de código abierto en diferentes OACE

Objetivo: conocer las características del proceso de configuración de red en Nova Servidores, deficiencias que se presentan en la ejecución de este proceso y posibles soluciones a dichas deficiencias.

1. ¿Qué características tiene el proceso de configuración de red en Nova Servidores?
2. ¿Cuáles son los comandos que deben dominar los administradores de red para realizar la configuración de red en Nova Servidores?
3. ¿Qué ficheros son necesarios modificar el proceso de configuración de red en Nova Servidores? ¿Cuál es su ubicación?
4. ¿Existen deficiencias durante el proceso de configuración de red en Nova Servidores?
5. ¿Cuáles son estas deficiencias?
6. ¿Cuáles son las más comunes?
7. ¿Cómo usted las solucionaría?
8. ¿Cree usted que es importante el uso de una interfaz gráfica de usuario para agilizar el proceso de configuración de red en Nova Servidores? ¿Por qué?
9. Considera que sería factible administrar de forma remota la configuración de red en Nova Servidores. ¿Por qué?

Anexo 2: Guía de observación para verificar la gestión de copias de seguridad en aplicaciones informáticas

Observadora: Betsy Alfonso Colás

Lugar: Laboratorio 103 del Centro de Software Libre

Objetivo: Identificar los elementos fundamentales para la administración de la configuración de red en herramientas informáticas.

1. Datos de las herramientas informáticas que permiten la administración de la configuración de red:
 - Nombre de las aplicaciones
 - Tipo de licencia
 - Cuenta con interfaz gráfica de usuario
 - Compatibilidad con diferentes sistemas operativos
2. Características de las herramientas informáticas que permiten administración de la configuración de red:
 - ¿Cómo son las herramientas informáticas que permiten la administración de la configuración de red?
 - ¿Qué herramientas necesitan para ser instaladas?
 - ¿Cuáles son los pasos a seguir para instalar estas herramientas?
3. Impacto social de las herramientas informáticas que permiten administración de la configuración de red.