UNIVERSIDAD DE ANTIOQUIA

AUTOMATIZACIÓN DE LA CONFIGURACIÓN DE LOS SERVICIOS CARRIER ETHERNET E IP NEXT GENERATION (CE/IPNG) EN LA TOPOLOGÍA DE RED MPLS HUAWEI DE INTERNEXA COLOMBIA

Autor
Luis Felipe Erazo Solarte

Universidad de Antioquia
Facultad de Ingeniería, Departamento de Ingeniería Electrónica y Telecomunicaciones
Medellín, Colombia

2020

AUTOMATIZACIÓN DE LA CONFIGURACIÓN DE LOS SERVICIOS CARRIER ETHERNET E IP NEXT GENERATION (CE/IPNG) EN LA TOPOLOGÍA DE RED MPLS HUAWEI DE INTERNEXA COLOMBIA

Luis Felipe Erazo Solarte

Informe de práctica empresarial
como requisito para optar al título de:
Ingeniero Electrónico

Asesores

Erwin Alexander Leal Piedrahita – Ingeniero Electrónico
Julián Esteban Montoya Gallego – Ingeniero de Telecomunicaciones

Universidad de Antioquia
Facultad de Ingeniería, Departamento de Ingeniería
Electrónica y Telecomunicaciones
Medellín, Colombia
2020

Resumen

Carrier Ethernet (CE) e IP Next Generation (IPNG) son algunos de los servicios ofertados por InterNexa Colombia sobre su red MPLS (Multiprotocol Label Switching). Desde hace 20 años, la configuración de estos servicios se ha realizado de forma manual mediante el uso de una interfaz de línea de comandos (CLI, Command Line Interface). En este contexto, el ingeniero de configuración debe conocer muy bien la topología de la red que va a configurar, así como cuáles serán los comandos necesarios para proceder a estructurar un servicio o, en su defecto, consultar constantemente los manuales del fabricante de los dispositivos de red. Este esquema de operación hace que la configuración de un servicio pueda tardar aproximadamente 45 minutos en promedio, sin mencionar los errores que se pueden generar al ingresar de manera errónea un parámetro de configuración.

El NOC Latam (Network Operations Center - Centro de operaciones de red de Latinoamérica) de INTERNEXA no poseía una herramienta de gestión automatizada de redes que permitiera tener control sobre los equipos de la red MPLS de InterNexa en Colombia. Es por esto que haciendo uso del lenguaje de programación de Python se desarrolló e implementó un software de automatización de configuración de los servicios CE e IPNG.

Este informe describe el proceso realizado para la concepción y construcción de esta herramienta, la cual permitirá al ingeniero de configuración reducir los tiempos de puesta en operación de un servicio.

Tabla de Contenido

Índice de Figuras y Tablas

Lista de anexos

1. INTRODUCCIÓN

INTERNEXA es una empresa colombiana, filial del grupo ISA, la cual cuenta con una experiencia de 20 años en el aprovisionamiento de soluciones tecnológicas de negocio, confiables, seguras e innovadoras para operadores de telecomunicaciones, instituciones del gobierno y empresas. La compañía cuenta con un sistema de redes internacionales, nacionales y metropolitanas, las cuales dan respaldo a sus servicios y están equipadas con tecnología de punta que permite conectar a diferentes países de América Latina (Centroamérica, Colombia, Ecuador, Brasil, Perú, Chile, Argentina), y a éstos con los Estados Unidos. Esta red cuenta con más de 50.000 kilómetros de fibra óptica, lo que permite difundir la oferta de servicios en más de 257 puntos de presencia en todo el continente americano, beneficiando a más de 868 clientes. En Colombia, la red MPLS (Multiprotocol Label Switching) de Huawei está en funcionamiento desde el año 2011, cuenta con 7 anillos de transporte conformados por 37 nodos ubicados en las principales ciudades y brinda una capacidad de comunicación de 100GB. Carrier Ethernet (CE) e IP Next Generation (IPNG) son algunos de los servicios ofertados sobre la red MPLS de InterNexa en Colombia. De manera general se puede decir que un servicio CE es un canal donde fluye un tráfico de datos entre empresas, servicios de telefonía y cloud. Por otra parte, IPNG es un servicio que provee un canal con tráfico de internet que cuenta con acceso a contenidos locales, CDN (Content Distribution Network), que son contenidos almacenados temporalmente dentro de la red de Internexa.

En el NOC Latam de INTERNEXA existen diversos profesionales encargados de la supervisión y monitoreo de los servicios prestados por la compañía, entre ellos destacan los ingenieros de configuración, de gestión y de red. Los ingenieros de configuración trabajan entre las capas 1, 2, 3 y 4 del modelo OSI (Open System Interconnection – Modelo de Interconexión de Sistemas Abiertos), y entre sus funciones se destaca la de programar y establecer los servicios de CE/IPNG cuando una empresa solicita.

Hasta el desarrollo de este proyecto, la configuración de estos servicios se realizaba de forma manual haciendo uso de una interfaz de línea de comandos. En consecuencia, el ingeniero de configuración debía conocer muy bien la topología de red que sería configurada y además el ingeniero debía tener claro los comandos necesarios para proceder a estructurar un servicio o, en su defecto, consultar constantemente los manuales del

fabricante de los dispositivos de red. Este esquema de operación hace que la configuración de un servicio pueda tardar aproximadamente 45 minutos en promedio, sin tener en cuenta los errores que se pueden producir al ingresar de manera errónea un parámetro dentro de un comando ejecutado.

Inicialmente, para dar solución a la problemática planteada, se analizaron tres herramientas disponibles en el mercado que posibilitan la gestión automatizada de redes de telecomunicaciones: NetConf, Ansible y Python. Cada una de estas herramientas provee funciones altamente eficientes, facilitando configuraciones e interacciones con los dispositivos de la red.
Luego de estudiar la topología de red MPLS Huawei de Internexa en Colombia, sus respectivos equipos y las necesidades de los ingenieros de configuración, se procedió a escoger la herramienta que mejor se adaptara a los requerimientos de operación del sistema. En este caso se eligió el lenguaje de programación Python. Los argumentos que justifican esta selección se mencionan en la sección 4.2.
De esta manera, utilizando Python como lenguaje de programación se desarrolló una herramienta de gestión capaz de automatizar el proceso de configuración de los dispositivos que soportarán la implementación de servicios CE e IPNG en la red MPLS Huawei de Internexa en Colombia.
La implementación de un software de automatización aporta seguridad en los datos procesados, eficacia en la ejecución de procesos y reducción en errores por digitación. Así mismo, este tipo de desarrollo puede ser llevado a otra clase de topologías de red, para realizar diferentes tareas y optimizar procesos.

En este proyecto, haciendo uso del software de automatización de la configuración de los servicios CE e IPNG se logró tener una optimización de tiempos de operación del 80%. Esta reducción en tiempos se verá reflejada más adelante en términos monetarios. La hora labor de un ingeniero de configuración en InterNexa tiene un costo aproximado de $31.250 (pesos colombinos) y en una hora el ingeniero puede realizar en promedio una sola configuración de un servicio CE e IPNG. Por otra parte, en una hora el software de configuración es capaz de realizar en promedio 9 configuraciones de servicios CE e IPNG.
A diario en promedio se presentan 13 solicitudes de configuración de servicios, por lo tanto, al usar el software desarrollado se lograría atenderlas en menos de dos horas (en comparación a las 13 horas requeridas en modo manual).

2. OBJETIVOS

OBJETIVO GENERAL

Implementar una plataforma de gestión que permita la automatización del proceso de configuración de servicios CE/IPNG en la red MPLS Huawei Colombia de InterNexa, haciendo uso del protocolo SSH (Secure Shell) y herramientas de gestión automatizada de redes.

OBJETIVOS ESPECÍFICOS

- Establecer los requisitos que debe tener la plataforma de gestión, a partir del estudio del actual esquema de operación del proceso de configuración de los servicios CE e IPNG en la topología de la red MPLS Huawei de InterNexa en Colombia.

- Identificar las herramientas disponibles en el mercado para la automatización de redes y seleccionar la que mejor se ajuste con el modelo de configuración de los equipos ya existentes en la red de InterNexa.

- Desarrollar la plataforma de gestión en un ambiente controlado de pruebas de acuerdo con la herramienta seleccionada y los requerimientos de operación del NOC.

- Validar el correcto funcionamiento de la plataforma a través de la validación de los archivos de configuración registrados en el dispositivo, así como el tiempo utilizando para implementar un servicio con referencia al tiempo requerido por un ingeniero en forma manual.

3. MARCO TEÓRICO

3.1. TECNOLOGÍA MPLS

La conmutación de etiquetas de protocolo múltiple, MPLS (Multiprotocol Label Switching), es un mecanismo de transporte de datos estándar. Fue creado por la IETF (Internet Engineering Task Force), una organización dedicada a mejorar el flujo de trabajo de Internet, con el propósito de unificar el servicio de transporte de datos para las redes basadas en circuitos y en paquetes. MPLS puede ser utilizado para transportar diferentes tipos de tráfico, incluyendo tráfico de voz y de paquetes IP. Esta tecnología opera entre la capa de enlace de datos (capa 2) y la capa de red (capa 3) del modelo OSI. Por esta razón, a menudo se describe informalmente como una tecnología que funciona en la capa 2.5 [1].

MPLS funciona anexando un encabezado a cada paquete. Dicho encabezado contiene una o más "etiquetas", y al conjunto de etiquetas se le llama pila o "stack". La principal característica de este estándar es el uso de etiquetas. Una etiqueta es un identificador de cuatro bytes (*32 bits*) que transmite la ruta de reenvío predeterminada del paquete en una red MPLS. Las etiquetas también pueden contener información relacionada con la calidad de servicio (QoS - Quality of Service), que indica el nivel de prioridad de un paquete. Las etiquetas MPLS constan de cuatro partes (Ver Figura 1) [1], [2]:

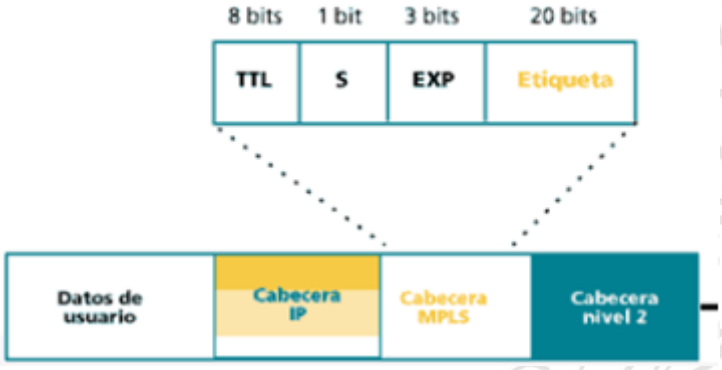


Figura 1: Cabecera MPLS.
Tomado de [2].

- *Valor de Etiqueta*: Campo de 20 bits utilizado para la identificación de la etiqueta.
- *Experimental*: Campo de 3 bits para uso experimental. Aparece como calidad de servicio en otros textos, y afecta al encolado y descarte de paquetes.
- *Final de la pila*: Campo de un bit que sirve para el apilado jerárquico de las etiquetas. Cuando vale 0 indica que hay más etiquetas añadidas al paquete. Cuando vale 1 estamos en el fondo de la jerarquía.
- *Tiempo de vida* (TTL- Time To Live): Campo de 8 bits que se decrementa en cada enrutador y al llegar a cero, el paquete es descartado.

Este esquema de operación solventa el problema presente en el enrutamiento IP, estableciendo rutas predeterminadas que permiten incrementar la velocidad y configurar flujos de tráfico en una red de una manera más eficiente. MPLS brinda alto rendimiento para aplicaciones en tiempo real, como voz y video. Además de esto, ofrece un mecanismo sencillo para crear VPNs (Virtual Private Networks), ya que permite la creación de circuitos o túneles virtuales dentro de la red IP, y esto a su vez, garantiza poder aislar el tráfico y el acceso al mismo [1].

Las rutas, que se denominan rutas conmutadas por etiquetas (LSPs – Label Switched Paths), permiten a los proveedores de servicios decidir con anticipación la mejor manera para que ciertos tipos de tráfico fluyan dentro de una red pública o privada [1].
En una red MPLS, cada paquete se etiqueta al entrar en la red del proveedor de servicios por el enrutador de entrada, también conocido como enrutador de borde de etiqueta (LER - Label Edge Router). Este enrutador también decide el LSP que tomará el paquete para alcanzar su dirección de destino [1].
Todos los enrutadores de conmutación de etiquetas (LSRs – Label Switching Routers) posteriores realizan el reenvío de paquetes basado únicamente en esas etiquetas MPLS, (Ver Figura 2).

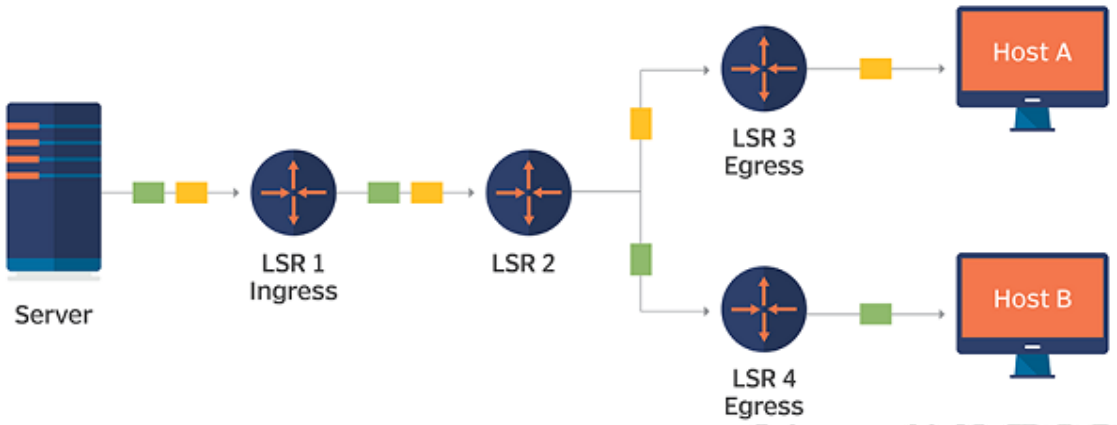


Figura 2: Red MPLS básica.
Tomado de [1].

Finalmente, el enrutador de salida elimina las etiquetas MPLS y reenvía el paquete IP original hacia su destino final, (Ver Figura 3) [1].

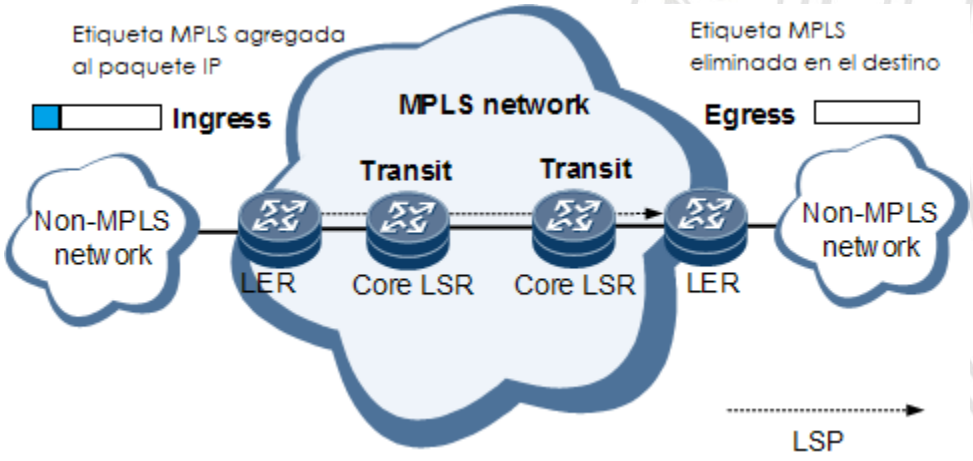


Figura 3: Etiquetas en la red MPLS.
Editada y tomado de www.support.huawei.com.

3.2. RED MPLS HUAWEI DE INTERNEXA EN COLOMBIA

En Colombia, la red MPLS de Huawei está en funcionamiento desde el año 2011. Cuenta con 7 anillos de transporte conformados por 37 nodos ubicados en las principales ciudades brindando una capacidad de comunicación de 100GB. Cada nodo de la red MPLS de Huawei es nombrado con base en el código de aeropuertos IATA (Asociación

Internacional de Transporte Aéreo), de acuerdo a la ciudad donde el nodo MPLS esté ubicado [4].

La disposición en anillo (Ver Figura 4) proporciona redundancia y tolerancia a fallos, ya que, ante la caída de uno de los nodos principales, el flujo de datos tendría continuidad a través del otro lado del anillo. Esta característica también provee ventajas en la escalabilidad. Por ejemplo, si se necesitara agregar un nuevo equipo al Core del anillo MPLS, no existiría caída de servicio para los clientes actuales debido a que habría flujo de datos a través de los otros equipos de la red que conforman la disposición en anillo [3].

Figura 4: Topología de red en anillo de InterNexa en Colombia. Tomado de www.internexa.com/Maps.

Los nodos que integran la red MPLS de Internexa hacen referencia a switches de enrutamiento S9300 de Huawei [4].

3.2.1. SWITCHES DE ENRUTAMIENTO S9300 HUAWEI

Los switches de enrutamiento terabit de la serie Huawei S9300 son switches diseñados para redes multiservicio. El S9300 utiliza el concepto de conmutación inteligente multicapa de Huawei para proporcionar servicios de conmutación Capa 2 / Capa 3 de alto rendimiento. Suministra aplicaciones de red enriquecidas como video de alta definición (HD - High Definition), computación en la nube elástica, hardware IPv6 (Internet Protocol version 6), seguridad unificada y H-QoS (Hierarchical Quality of Service – Jerarquía de Calidad de Servicio). Los switches S9300 se caracterizan por proporcionar funciones de enrutamiento y conmutación convergentes de extremo a extremo. Se utilizan en redes de área amplia (WAN – Wide Area Network), redes de área metropolitana (MAN – Metropolitan Area Network) y centros de datos para ayudar a los operadores a construir redes centradas en aplicaciones [5].

El S9300 está disponible en tres modelos: S9303, S9306 y S9312. (Ver Figura 5) Todos los modelos S9300 usan módulos y componentes intercambiables para una capacidad de conmutación económica y expansión de puertos. Además, los modelos S9300 utilizan tecnologías innovadoras de ahorro de energía que reducen considerablemente el consumo de energía y el ruido sin comprometer el rendimiento o la estabilidad [5].

Figura 5: Modelos switches Huawei S9300.
Tomado de [5].

3.2.2. SERVICIO CARRIER ETHERNET

Carrier Ethernet (CE) es el despliegue de tecnología y servicios que aprovecha todas las ventajas de Ethernet mediante el desarrollo de un marco flexible, con formato universal de tramas y diseño sencillo. Además, CE provee capacidades de Operaciones, Administración y Mantenimiento (OAM) mejoradas para las redes de alto rendimiento, que son ideales para empresas, instituciones académicas, compañías operadoras y las redes de área local del gobierno. Al igual que ocurrió con Ethernet, la mayor adopción de Carrier Ethernet está cambiando la forma de crear redes [6].

En este servicio se pueden establecer conexiones punto a punto y conexiones punto a multipunto (entre un mismo cliente o entre clientes diferentes), trabajando solamente hasta la capa 2 (Switching) del modelo OSI (Open System Interconnection – Modelo de Interconexión de Sistemas Abiertos). EL tráfico existente a través de estos canales puede ser por ejemplo servicios de voz, telefonía, datos entre empresas y cloud. A diferencia del servicio IPNG, el servicio CE no cuenta con salida a internet (Ver Figura 6) [4].

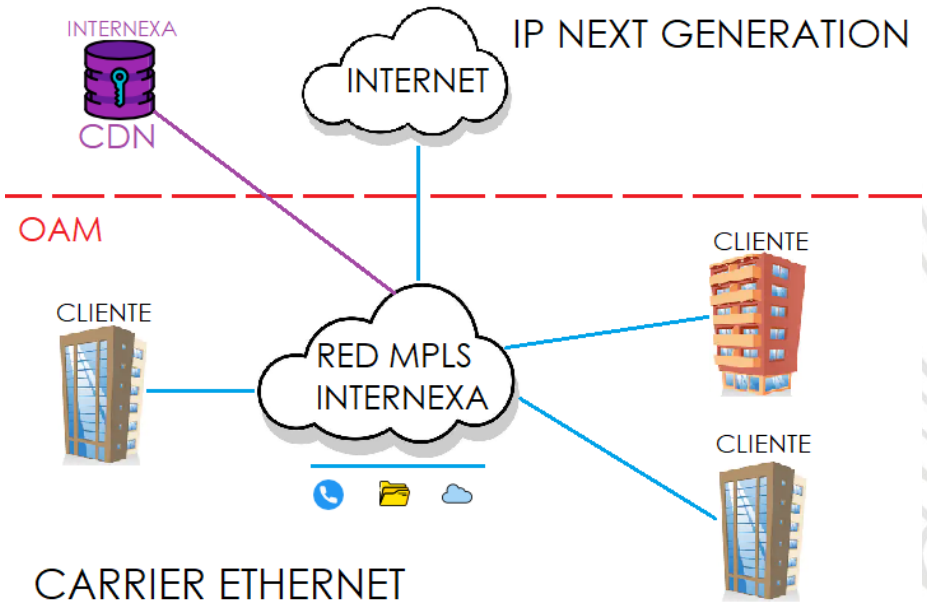


Figura 6: Esquema de servicios CE e IPNG.
Elaboración propia basado en [6].

3.2.3. SERVICIO IPNG

Una Red de Siguiente Generación es una red basada en la transmisión de paquetes capaz de proveer servicios integrados, incluyendo los tradicionales como acceso a Internet, llamadas de voz, mensajería, etc. Es capaz de explotar al máximo el ancho de banda del canal haciendo uso de las Tecnologías de Calidad del Servicio (QoS) de modo que el transporte sea totalmente independiente de la infraestructura de red utilizada. Además, ofrece acceso libre para usuarios de diferentes compañías y apoya la movilidad que permite acceso multipunto a los usuarios [7].

IPNG "next generation", es un servicio que provee un canal con tráfico de internet (Ver Figura 6) que cuenta con acceso a contenidos locales y CDN (Content Distribution Network), que son contenidos almacenados dentro de la red de Internexa. Se pueden establecer conexiones punto a punto y conexiones punto a multipunto donde se trabaja entre las capas dos y tres del modelo OSI. Se puede decir que el servicio CE está implícito en los servicios IPNG [4].

En la red MPLS de InterNexa en Colombia el servicio de internet llega por medio de cables submarinos. El cable submarino *PCCS* (Pacific Caribbean Cable System) llega a la ciudad de Cartagena y el cable submarino *Sam-1* (South America-1) llega a la ciudad de Barranquilla (Ver Figura 7). Desde estas dos ciudades se provee de internet a ciudades como Cali, Medellín y Bogotá, las cuales se encargan finalmente de llevar el internet al resto del país. [4]

Figura 7: Cables submarinos Internexa en Colombia.
Tomado de www.submarinecablemap.com

3.3. GESTIÓN DE REDES DE TELECOMUNICACIONES

En la actualidad las redes de telecomunicación se caracterizan por un constante incremento del número, complejidad y heterogeneidad de los recursos que los componen.

Los principales problemas relacionados con la expansión de las redes son la gestión de su correcto funcionamiento día a día y la planificación estratégica de su crecimiento. De hecho, se estima que más del 70% del coste de una red corporativa se atribuye a su gestión y operación [8].

De esta manera se define la gestión como un conjunto de capacidades que permiten el intercambio y procesamiento de información. Su propósito es ayudar a cualquier organización que opera o utiliza una red de comunicaciones, a realizar sus actividades de planificación, instalación, operación y administración con eficacia [8], [9].

Los tres recursos principales con los que cuenta un centro de gestión de red son los siguientes [8]:

- *Métodos de gestión:* Definen los criterios de comportamiento de los elementos del centro de gestión de red ante determinadas circunstancias.

- *Recursos humanos:* Personal encargado del correcto funcionamiento del centro de gestión de red.

- *Herramientas de apoyo:* Herramientas que facilitan las tareas de gestión a los operadores humanos y posibilitan minimizar errores de operación.

La mayoría de las herramientas de apoyo de gestión de red se basan en el paradigma gestor-agente, cuyo esquema se muestra en la Figura 8 [8].

Los elementos del sistema de gestión de red, bajo el paradigma gestor-agente, se clasifican en dos grandes grupos [8], [9]:

- Los *gestores* son los elementos del sistema de gestión que interaccionan con los operadores humanos y desencadenan acciones necesarias para llevar a cabo las tareas por ellos invocadas.

- Los *agentes*, por otra parte, son los componentes del sistema de gestión invocados por el gestor o gestores de la red.

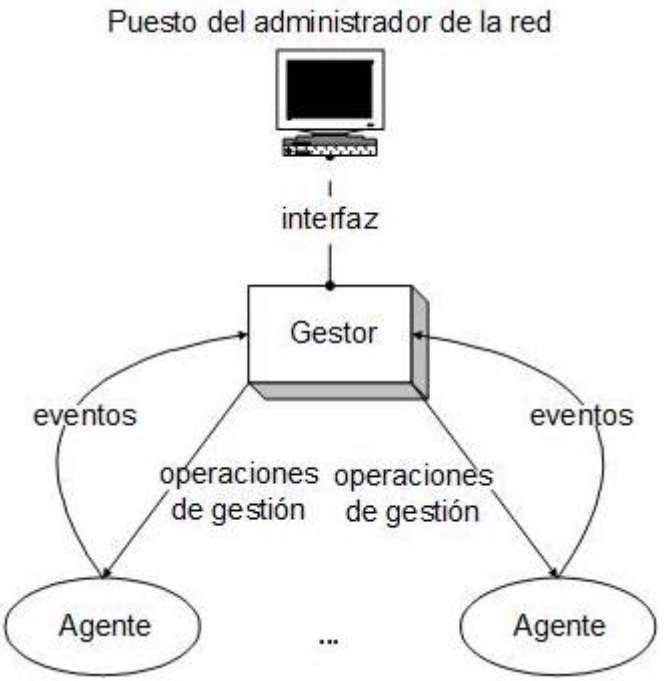


Figura 8: Paradigma gestor-agente.
Tomado de [8].

3.4. HERRAMIENTAS PARA LA GESTIÓN AUTOMATIZADA DE REDES

En el mercado se encuentran diferentes herramientas que ayudan a la gestión y automatización de redes, facilitando configuraciones e interacciones con los dispositivos de la red. Entre estas podemos destacar: NetConf, Ansible y Python.

3.4.1. NetConf

NetConf (Network Configuration – Configuración de red) es un protocolo de configuración de red que proporciona mecanismos para instalar, manipular y eliminar la configuración de dispositivos de red. Este protocolo utiliza archivos XML (eXtensible Markup Language) como método de configuración y codificación de datos, y RPC (Remote Procedure Call) como mecanismo para la operación y el control entre

un administrador y un dispositivo de la red. Se utilizan sesiones para realizar el intercambio de datos de configuraciones de los dispositivos de red. Una sesión Netconf, es la conexión lógica entre un administrador de red o una aplicación de configuración de red y un dispositivo de red [10].

3.4.2. Ansible

Ansible es una herramienta de software que sirve para configurar y administrar sistemas. Permite desplegar configuraciones de servidores y servicios por lotes y administrar configuraciones. Además de esto, Ansible es categorizado como una herramienta de orquestación. Esta herramienta gestiona nodos a través del protocolo SSH (Secure Shell) y no requiere ningún software remoto adicional (excepto Python 2.4 o posterior para instalarlo). Dispone de módulos que trabajan sobre JSON (JavaScript Object Notation) y la salida estándar puede ser escrita en cualquier lenguaje. Originalmente utiliza el formato YAML (Yet Another Markup Language) para describir configuraciones reusables de los sistemas [11].

3.4.3. Python

Por otro lado, Python es un lenguaje de programación que está desarrollado bajo una licencia de código abierto, permitiendo trabajar más rápidamente e integrar sus sistemas de manera más efectiva. Python cuenta con una amplia gama de librerías que permite realizar programas para automatizar procesos de configuración de dispositivos de red. Entre estos se destaca *paramiko* (modulo para implementar SSH) [12], [13].

3.5. PROTOCOLO DE CONEXIÓN REMOTA SEGURA - SECURE SHELL (SSH)

SSH es un protocolo que facilita las comunicaciones seguras entre dos sistemas usando una arquitectura *cliente/servidor* y que permite a los usuarios conectarse a un host remotamente. Una vez conectado al host, este protocolo permite al cliente enviar comandos hacia el servidor y recibir la respuesta correspondiente a cada comando. Además de esto, SSH posibilita la copia de archivos entre diferentes hosts y la realización de túneles IP cifrados. A diferencia de otros protocolos de comunicación remota tales como FTP (File Transfer Protocol) o Telnet, SSH cifra la sesión de conexión y cifra el tráfico existente haciendo imposible que alguien pueda obtener información no cifrada (Ver Figura 9).

SSH está diseñado para reemplazar los métodos más viejos y menos seguros para registrarse remotamente en otro sistema a través de la shell de comando, tales como telnet o rsh (remote shell), ya que estas aplicaciones antiguas no cifran la información que viaja por la conexión.
El uso de métodos seguros para registrarse remotamente a otros sistemas reduce los riesgos de seguridad tanto para el sistema cliente como para el sistema remoto. [14]

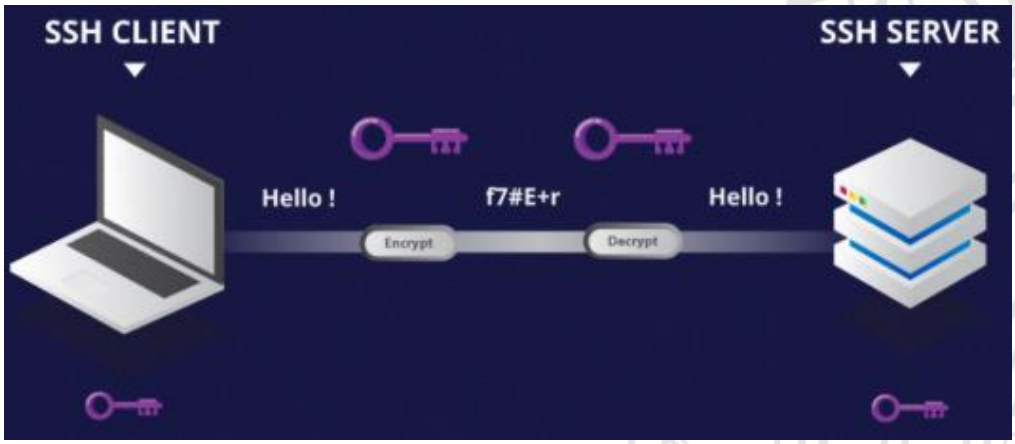


Figura 9: Representación simple de Secure Shell (SSH).
Tomado de www.hostinger.es/tutoriales/que-es-ssh

3.6. DESCRIPCIÓN DE DESARROLLOS DE SOFTWARE

Durante el ciclo de vida de un sistema software, un gran número y variedad de partes interesadas intervienen de diversas maneras. Por esta razón se ve la necesidad de realizar una descripción del desarrollo de software clara y concisa con el fin de que los involucrados comprendan su funcionamiento. Existen muchas formas de representar los desarrollos de software de manera completa, la norma ISO/IEC/IEEE 42010 es una opción [20].

3.6.1. Norma ISO/IEC/IEEE 42010

El mundo de la ingeniería se ha visto en la necesidad de elaborar estándares de descripción software, de tal manera que cuando exista un desarrollo, sus autores puedan representar las relaciones entre los diferentes elementos que integran, los procesos que se gestan al interior del sistema, la manera en que el usuario interactúa con el sistema, cómo debe instalarse el sistema, etc. Un estándar como el 42010 permite que cualquier persona comprenda y tenga noción sobre el

desarrollo planteado por alguien más, tal como se explica a continuación:

> *"La complejidad de los sistemas hechos por el hombre ha crecido a un nivel sin precedentes. Esto no solo ha creado nuevas oportunidades, sino que también ha traído mayores desafíos para las organizaciones que crean y utilizan sistemas. Los conceptos, principios y procedimientos de una arquitectura se aplican cada vez más para ayudar a administrar la complejidad con que se enfrentan las partes interesadas de los sistemas. La conceptualización de la arquitectura de un sistema, tal como se expresa en su descripción arquitectónica, facilita la comprensión de la esencia del sistema y de las propiedades claves relacionadas con su comportamiento, composición y evolución, que a su vez afecta aspectos tales como viabilidad, utilidad y mantenimiento del sistema. Las descripciones arquitectónicas son utilizadas por las partes que crean, utilizan y administran sistemas modernos para mejorar la comunicación y la cooperación, lo que les permite trabajar de manera integrada y coherente.*
>
> *Marcos de referencia arquitectónicos (architecture frameworks) y lenguajes de descripción de arquitecturas (ADLs, por sus siglas en inglés) se están creando como activos para codificar las convenciones y prácticas comunes del proceso arquitectónico y la descripción de arquitecturas dentro de diferentes comunidades y dominios de aplicación. Esta norma internacional aborda la creación, el análisis y el mantenimiento de arquitecturas de sistemas mediante la utilización de descripciones arquitectónicas. Esta norma internacional proporciona una ontología central para la descripción de arquitecturas. Las disposiciones de esta norma internacional sirven para imponer las propiedades deseadas en una descripción arquitectónica. Esta norma internacional también especifica disposiciones para imponer las propiedades deseadas de los marcos de referencia arquitectónicos y lenguajes de descripción de arquitecturas (ADLs), con el fin de apoyar de forma útil el desarrollo y utilización de descripciones arquitectónicas. Esta norma internacional proporciona una base sobre la cual comparar e integrar los marcos de referencia arquitectónicos y ADLs proporcionando una ontología común para especificar sus contenidos. Esta norma internacional se puede utilizar para establecer una práctica coherente para el desarrollo de descripciones arquitectónicas, marcos de referencia arquitectónica y lenguajes de descripción arquitectónica en el contexto de un ciclo de vida y sus procesos (no definidos por esta norma internacional). Esta norma internacional se puede usar adicionalmente para evaluar la conformidad de una descripción arquitectónica, de un marco de referencia arquitectónico, de un lenguaje de descripción de arquitectura, o desde un punto de vista arquitectónico (architecture viewpoints) según se disponga" [20].*

A continuación, en la Figura 10, se presenta un extracto de los principales elementos que integra la norma ISO/IEC/IEEE 42010.

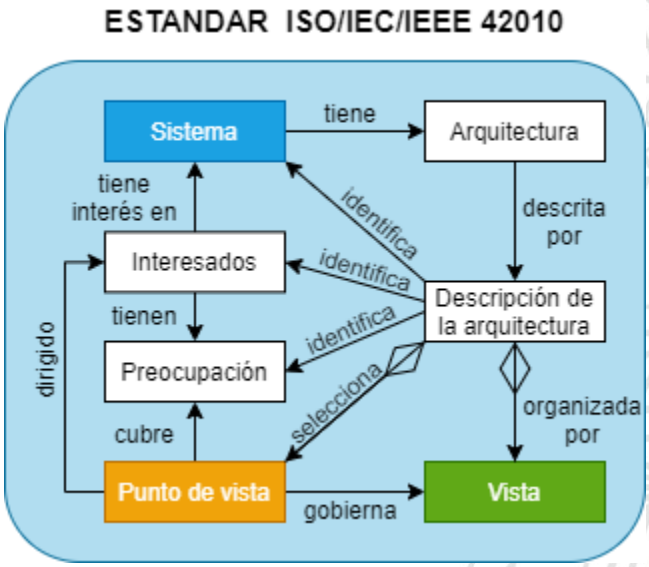


Figura 10: Representación simple del estándar ISO/IEC/IEEE 42010. Elaboración propia, basado en [20].

De acuerdo con la Figura 10, para un *Sistema* determinado van a existir varios *Interesados*, y cada uno de los interesados va a tener una *Preocupación* con respecto al sistema. Así como cada interesado tiene una preocupación en específico, también tendrá un punto de vista de su preocupación con respecto al sistema [20]. Un punto de vista no es más que un conjunto de reglas para la construcción de vistas o representaciones de diferentes aspectos del software.

Además de lo anterior, cada sistema tendrá una respectiva *Arquitectura* y por consiguiente una *Descripción* de la arquitectura dada ,que no es más que la recopilación de toda la documentación asociada a: el sistema, los interesados, las preocupaciones, los puntos de vista y las vistas como tal [20].

Para la representación del punto de vista se puede hacer uso de varios modelos, entre ellos destaca el "Modelo de Kruchten 4+1", el cual proporciona una serie de vistas que permiten describir completamente la arquitectura de un desarrollo de software [21].

3.6.2. MODELO DE KRUCHTEN 4 + 1

El modelo de Kruchten 4+1, es un modelo de vistas diseñado por el profesor Philippe Kruchten y que se acopla con el estándar "*ISO/IEC/IEEE 42010*" que se utiliza para describir la arquitectura de un sistema software haciendo uso de múltiples puntos de vista [21].

Kruchten plantea un sistema de documentación de software de cinco vistas. Estas cinco vistas las denominó Kruchten como: vista lógica, vista de procesos, vista de despliegue, vista física y la vista "+1". Esta última vista tiene la función de relacionar las 4 primeras vistas citadas, y la denominó vista de escenario (Ver Figura 11) [21].

Cada una de estas vistas describe toda arquitectura del software que se esté documentando, pero cada una de ellas se documenta de forma diferente y muestra aspectos diferentes del sistema. A continuación, se explica que información debe contener la documentación de cada una de estas vistas [21].

Figura 11:Vistas del modelo de Kruchten.
Elaboración propia basada en [21].

3.6.2.1. Vista Lógica

Esta vista representa la distribución del sistema y lo que éste debe hacer. Aquí se presentan las funciones y/o clases implementadas en el desarrollo de software.

Para completar la documentación de esta vista se pueden incluir los diagramas de clases, de comunicación o de secuencia de UML (Unified Modeling Language - Lenguaje Unificado de Modelado) [21].

3.6.2.2. Vista de Despliegue

En esta vista se presenta el sistema desde la perspectiva de un programador o desarrollador. En otras palabras, se va a mostrar cómo está dividido el sistema software en componentes o módulos y las dependencias que hay entre éstos.

Para completar la documentación de esta vista se pueden incluir los diagramas de componentes y de paquetes de UML [21].

3.6.2.3. Vista de Procesos

En esta vista se muestran detalladamente los procesos que hay en el sistema y la forma en la que se comunican entre ellos. Se representa desde la perspectiva de un integrador de sistemas, el flujo de trabajo paso a paso y las operaciones de los componentes que conforman el sistema. Para completar la documentación de esta vista se puede incluir el diagrama de actividad de UML o un diagrama de flujo [21].

3.6.2.4. Vista Física

En esta vista se muestra, desde la perspectiva de un ingeniero de redes, todos los componentes físicos del sistema, así como las conexiones físicas y lógicas entre los componentes que conforman la solución (incluyendo los servicios). Se deben detallar también los protocolos de conexión de las componentes físicas.

Para completar la documentación de esta vista se puede incluir el diagrama de despliegue de UML [21].

3.6.2.5. "+1" Vista de Escenarios

Esta vista representa la funcionalidad que el sistema software proporcionará a los usuarios finales. Para completar la documentación de esta vista se pueden incluir el diagrama de casos de uso de UML [21].

4. METODOLOGÍA

De manera general, el flujo del proyecto consistió en una primera etapa de estudio e identificación de conceptos bases y datos fundamentales dentro de la tecnología MPLS y los servicios CE e IPNG de InterNexa en Colombia. Una segunda etapa de análisis de las herramientas disponibles para la gestión y automatización de redes de telecomunicaciones y posterior selección de la que mejor se adaptaba a la necesidad de la empresa y del proyecto. Y finalmente, una tercera etapa donde se diseñó, desarrolló e implementó el software de automatización teniendo en cuenta los requerimientos de la empresa y de los ingenieros de configuración del NOC.

Con el propósito de asegurar el cumplimiento de los objetivos específicos de manera directa, eficiente y consistente, se presentan cuatro etapas. En éstas se describe una a una las actividades que fueron ejecutadas, así como las técnicas empleadas para el desarrollo del proyecto.

4.1. ESTUDIO DE CONCEPTOS BASES E IDENTIFICACIÓN DE DATOS

Haciendo uso de motores de búsqueda se comenzó con el estudio de la *tecnología MPLS*. Donde fue de suma importancia entender previamente la dinámica de este estándar, su uso de etiquetas, sus capacidades y sus velocidades. Además de esto fue necesario entender las ventajas de este estándar sobre el direccionamiento IP tradicional.

Mediante la ayuda de los ingenieros del NOC fue posible realizar el estudio de la topología de la red MPLS Huawei de InterNexa en Colombia. Aquí quedaron claros los datos que conforman cada nodo de la red, tales como la IP de gestión, la IP loopback, el nombre específico de cada nodo, los túneles MPLS y los puertos de entrega de servicios.

InterNexa tiene una amplia oferta de servicios, tales como: Servicios administrados, cloud y data center, seguridad de datos y conectividad. Este proyecto se basó principalmente en el servicio de conectividad y dentro de éste, en los servicios CE e IPNG. Después de estudiar cada uno de estos servicios, quedaron claros aspectos fundamentales a la hora del diseño del software de automatización. El servicio CE (conectividad entre sitios) se crea haciendo uso únicamente de los equipos (S9300) Huawei de la red MPLS de InterNexa. Por otro lado, el servicio IPNG (conectividad a internet) se crea haciendo uso de los equipos (S9300) Huawei y adicionalmente haciendo uso de los equipos (7750 y 7410) Nokia.

Posteriormente se estudió la forma de configuración manual de los servicios CE e IPNG, identificando el protocolo necesario para establecer una comunicación entre el ingeniero y el equipo a configurar, los comandos de configuración, así como los parámetros de cada servicio.

4.1.1. CONFIGURACIÓN MANUAL

Como se mencionó anteriormente, los ingenieros de configuración son quienes se encargan de estructurar cada servicio cuando un cliente lo solicita. Para estructurar un servicio se necesita configurar todos los equipos que se involucran dentro de éste. La configuración de cada equipo se ha venido realizando de forma manual, haciendo uso de una interfaz de línea de comandos, donde las más comunes son las ofertadas por *SecureCRT* y *MobaXterm.* Mediante la interfaz de línea de comandos, que esta alojada dentro del computador del ingeniero, se establece una comunicación haciendo uso del protocolo SSH (Secure Shell) y en el puerto 22 del dispositivo, tal como se muestra en la Figura ***12***.

Una vez que se ha establecido conexión entre el ingeniero y el equipo de la red mediante el protocolo SSH, el ingeniero procede a ejecutar de forma manual uno a uno los comandos necesarios para la configuración del servicio. En este proyecto se trabajó principalmente con dispositivos de red Huawei y en menor proporción con dispositivos de red Nokia. Cada fabricante de equipos de red define cuáles serán los respectivos comandos que ejecutarán ciertas acciones en el equipo. Por esta razón el ingeniero debe conocer de antemano dichos comandos o en su defecto, consultar constantemente los manuales del fabricante de los dispositivos de red.

Después de haber realizado la configuración de los equipos presentes en el servicio, el ingeniero también debe realizar una prueba donde se observe el correcto funcionamiento del mismo. En esta prueba lo que se hace es encender los puertos de entrega del servicio en cada equipo, y generar una conexión temporal donde se va a enviar y recibir tráfico. De tal manera que, si se hay una correcta recepción de tráfico en cada equipo, se concluye que el servicio quedó establecido.

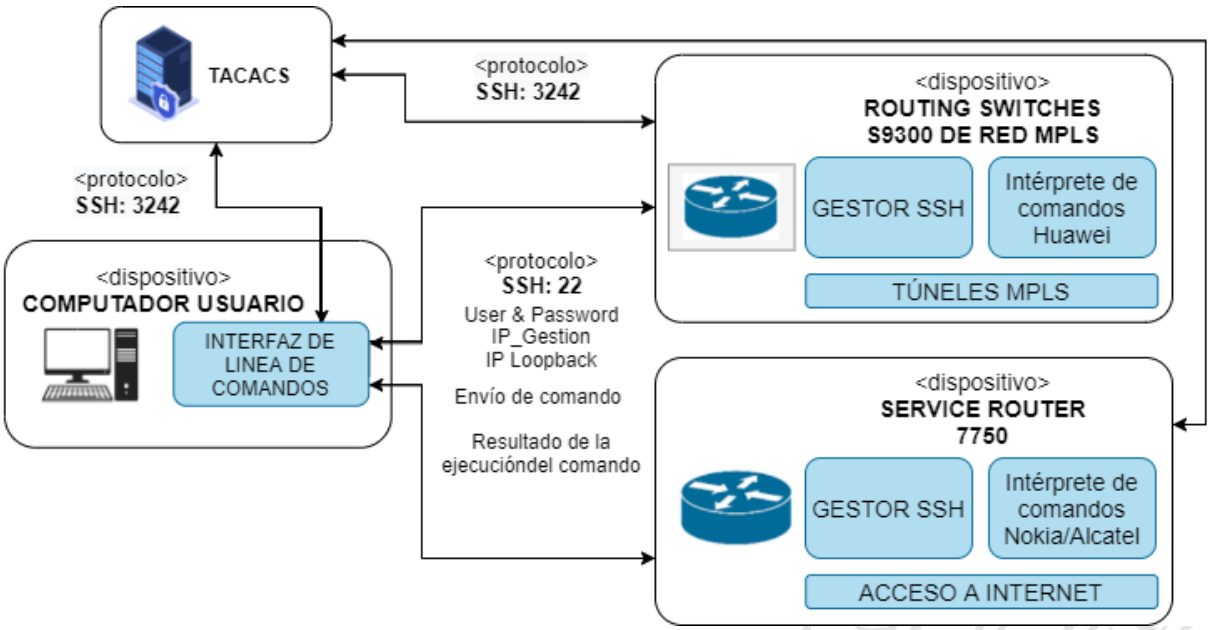


Figura 12: Comunicación con el equipo de la red MPLS desde la CLI de usuario. Elaboración propia.

En InterNexa, para realizar conexiones con los equipos a configurar en la forma manual, el protocolo SSH utiliza el puerto 22 de TCP (Transmission Control Protocol). Mientras que para conectarse al servidor TACACS (Terminal Access Controller Access Control System) utiliza el puerto 3242. Cabe resaltar que en la empresa aún existen conexiones que hacen uso del protocolo Telnet (conexiones de servicios antiguos), pero estas están siendo cambiadas al protocolo SSH [4].

TACACS permite a un servidor de acceso remoto comunicarse con un servidor de autenticación para determinar si el usuario tiene acceso de lectura y/o escritura en los equipos de la red. Además de esto, el servidor TACACS aloja información histórica de los accesos a su sistema, almacenando datos como los siguientes:

- Usuario que se conectó.
- Horario en que estuvo conectado.
- Comandos ejecutados.

Estos datos serán de gran ayuda en temas relacionados con auditoria.

4.1.2. DATOS NECESARIOS EN CADA SERVICIO

A la hora de estructurar un servicio hay una serie de datos fundamentales que el ingeniero requiere para iniciar la configuración de los equipos. Cada uno de esos datos va a causar que el servicio tenga variaciones en su proceso de configuración. Cuando un cliente solicita un servicio, esta solicitud primero pasa a través de los ingenieros comerciales, quienes se encargan de organizar los requerimientos y los datos a procesar. Con esta información, el ingeniero de configuración ya tiene los datos necesaria para proceder a estructurar el servicio en cuestión.

En la Figura 13 se observa todos los datos necesarios (excepto los comandos) para iniciar a configurar un servicio.

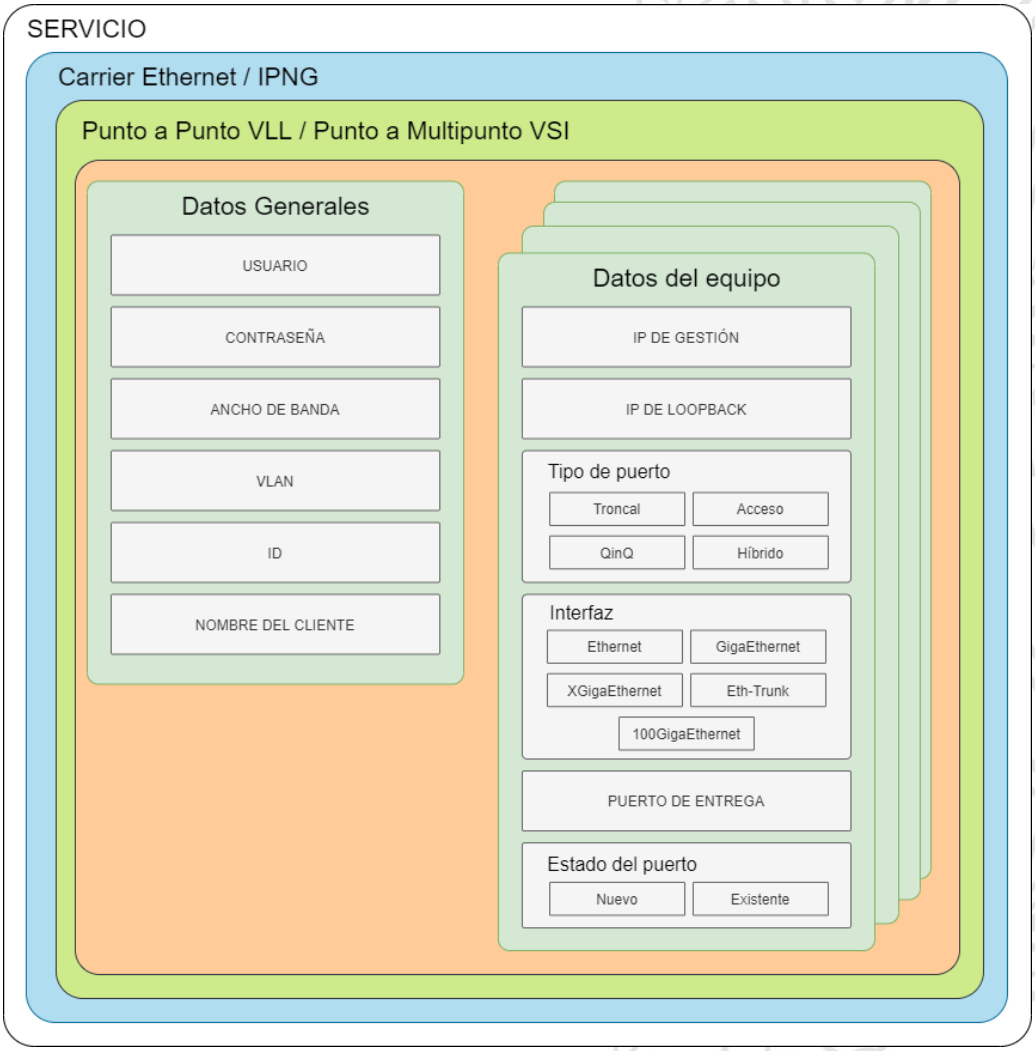


Figura 13: Datos necesarios para configurar un servicio. Elaboración propia.

Inicialmente se debe tener claro si se va a configurar un servicio ***CE*** o un servicio ***IPNG***. Ya que, si el servicio es CE, el ingeniero solamente va a trabajar con los dispositivos de red de Huawei en la capa dos del modelo OSI. Y si el servicio es IPNG, además de trabajar con los dispositivos de red de Huawei en la capa dos del modelo OSI, también tendrá que trabajar con los dispositivos de red de Nokia en la capa tres del modelo OSI.
Si en la configuración solamente actúan dos equipos, se tiene un servicio ***punto a punto VLL*** (Virtual Leased Lines), pero si se va a configurar más de dos equipos se tiene un servicio ***punto a multipunto VSI*** (Virtual Switch Instance).

Posteriormente se tiene dos grupos de datos, los datos generales y los datos de cada equipo, los cuales se describen a continuación:

4.1.2.1. Datos generales

Dentro de los datos generales se encuentra el nombre de ***usuario*** y su ***contraseña***. Cada ingeniero tiene un usuario y una contraseña definidos con sus respectivos permisos dentro del servidor de autenticación TACACS. De tal manera que personas externas no podrán acceder a la información de los equipos, y menos aún podrán configurarlos. El ***ancho de banda*** ofertado por InterNexa está en un rango que va desde los 2 Mbps (Megabits por segundo) hasta los 100 Gbps (Gigabits por segundo). El número de la ***VLAN*** (Virtual Local Area Network) en InterNexa está en un rango que va desde la VLAN número 2 hasta la VLAN número 4095. El ***ID*** (Identification) corresponde al número que va a identificar el servicio, este número es único, exclusivo e irrepetible, solamente se podrá volver a utilizar siempre y cuando el servicio que tenía asociado haya sido cancelado. El último dato de esta sección corresponde al ***nombre del cliente*** (empresa o compañía) que adquiere el servicio. Este nombre va a ser utilizado más adelante, durante la ejecución de comandos de configuración para describir la VLAN, la interfaz VLAN y los puertos de entrega del servicio.

4.1.2.2. Datos del equipo

Dentro de los datos del equipo se encuentra la ***IP de gestión***, la cual permite realizar la conexión SSH con el equipo, para realizar su posterior configuración o monitoreo. La ***IP loopback*** permite localizar los túneles de comunicación entre este equipo y el resto de los equipos de la red MPLS. Actualmente existen 4 tipos de configuración de puerto, la configuración en ***Troncal*** hace referencia a un puerto por donde van a pasar varias VLAN. La configuración en ***Acceso***

hace referencia a un puerto por donde va a pasar únicamente una sola VLAN. La configuración **QinQ** hace referencia a un puerto por donde van a pasar demasiadas VLAN, muchas más que en la configuración en Troncal. Y finalmente, la configuración **Hibrido** hace referencia a la utilización de configuraciones en Acceso, Troncales y QinQ de manera conjunta.

InterNexa tiene a la disposición de sus clientes 5 tipos de interfaces de enlaces físicos, la utilización de cada una de estas viene definida por el ancho de banda del servicio. Se utilizará la interfaz **Ethernet** para aquellos servicios que cuenten con un ancho de banda en un rango que va desde 2 Mbps hasta 10 Mbps. La interfaz **GigaEthernet** se utilizará para aquellos servicios que cuenten con un ancho de banda en un rango que va desde 10 Mbps hasta 900 Mbps. La interfaz **XGigaEthernet** se utilizará para aquellos servicios que cuenten con un ancho de banda en un rango que va desde 1 Gbps hasta 9 Gbps. La interfaz **100GigaEthernet** se utilizará para aquellos servicios que cuenten con un ancho de banda de 100 Gbps. Por otra parte, la interfaz **Eth-Trunk** es un enlace lógico constituido a partir de los enlaces físicos vistos anteriormente. El ancho de banda que pasa a través de este enlace será la sumatoria del ancho de banda existente en los enlaces físicos asociados al servicio.

Finalmente, a un puerto se le pueden asociar dos estados. **Nuevo**, este estado implica que cuando se vaya a entregar un servicio en dicho puerto se requiere la ejecución de comandos adicionales para su creación y configuración. **Existente**, estado que no requiere la ejecución de órdenes para la creación de un puerto.

4.2. SELECCIÓN DE LA HERRAMIENTA DE TRABAJO

Como se mencionó anteriormente, la única herramienta con la que cuentan los ingenieros de configuración del NOC de InterNexa para trabajar con su red MPLS en Colombia es una interfaz de línea de comandos (CLI, Command Line Interface). Es una herramienta bastante sencilla que hasta el momento ha sido de gran ayuda para la gestión de red en la empresa, pero no cuenta con opciones que permitan automatizar los procesos. Por esta razón, al inicio de este proyecto fue necesario investigar sobre herramientas de gestión automatizada de redes. En el mercado de las telecomunicaciones existen varias herramientas que ayudan a la gestión y automatización de redes, facilitando configuraciones e interacciones con los dispositivos de la red.

Después de consultar y estudiar las diferentes opciones, en una primera instancia se eligieron tres posibles herramientas: *NetConf*, *Ansible* y *Python*.
Y posteriormente para la selección final de la herramienta empleada en este proyecto se tuvo en cuenta tres criterios.

4.2.1. CRITERIOS DE SELECCIÓN

Los tres criterios de selección presentados a continuación fueron concertados con los directivos y con los ingenieros de configuración de InterNexa.

- Curva de aprendizaje: Este criterio hace referencia al tiempo que tardaría una persona en adquirir cierta cantidad de aprendizaje con respecto a una herramienta. Depende mucho de los conocimientos previos que tenga la persona y de la dificultad del trabajo con dicha herramienta.

- Documentación de la herramienta: Es de suma importancia también la documentación de la herramienta, ya que con la carencia de información sería muy complicado o quizá imposible el desarrollo del proyecto.

- Herramienta no invasiva: Es tal vez el criterio más importante debido a que se buscaba hacer uso de una herramienta que no requiriera instalación de software adicional en los dispositivos de la red. Ya que esto no estaba permitido por los directivos de InterNexa para el desarrollo de este proyecto.

4.2.2. ANÁLISIS DE LAS HERRAMIENTAS

Después de realizar una *comparación objetiva* fueron evidentes las diferencias entre cada una de estas herramientas y el alcance que cada una de ellas permite tener en el ámbito de la automatización de procesos en redes de telecomunicaciones. Así como cada herramienta cuenta con virtudes que la hacen única, también cuenta con limitaciones ya sea de costos o funcionamiento.

Aunque NetConf es una herramienta ampliamente utilizada en la gestión de redes de telecomunicaciones, ésta demanda la instalación y configuración de agentes en cada uno de los dispositivos de red. Estos agentes son los encargados de establecer comunicaciones entre un administrador y un dispositivo de la red y son necesarios para su correcta operación. Cabe resaltar que la instalación de estos agentes

en cada uno de los equipos de la red conlleva un gasto de tiempo y de costos elevado. Además de esto, la IETF recomienda la utilización de YANG (Yet Another Next Generation, Otra Nueva Generación) como lenguaje de modelado de datos para el protocolo de configuración de red NETCONF, lo cual implicaría comenzar desde cero el aprendizaje de un nuevo leguaje, demandando tiempo adicional.

Por otro lado, Ansible es una herramienta de software que sirve para configurar y administrar sistemas. Permite desplegar configuraciones de servidores, servicios por lotes y administrar configuraciones, siendo categorizado como una herramienta de orquestación. Es ampliamente utilizada en el mundo de las redes de telecomunicaciones, pero a la hora de trabajar con dispositivos de red de la marca Huawei requiere la instalación de módulos adicionales, entre ellos el módulo CloudEngine.
Cabe resaltar que para trabajar con esta herramienta se debía comenzar desde cero, debido a que en InterNexa ningún ingeniero había trabajado con ella y de igual forma el estudiante en práctica tampoco había tenido conocimiento de esta plataforma.

Finalmente, aunque Python no requiere la instalación y configuración de software adicional en los dispositivos de red, exige el uso de buenas prácticas de programación para garantizar que el producto final opere satisfactoriamente en producción. Cabe resaltar que es un lenguaje de programación que está desarrollado bajo una licencia de código abierto, por tal motivo no requiere el pago de una licencia. También cuenta con una amplia documentación y gama de módulos que permiten realizar programas para automatizar procesos de configuración de dispositivos de red haciendo uso del protocolo SSH. Además, el estudiante en práctica ya había trabajado con este lenguaje de programación, lo cual implica que cuenta con conocimientos básicos, lo cual a su vez facilita el desarrollo del proyecto.

4.2.3. RESULTADO DE LA SELECCIÓN

Teniendo en cuenta los tres criterios de selección, las características y los alcances de cada herramienta se procedió a elegir la que mejor se acoplara a todos los requerimientos de la empresa y del proyecto.
Junto con los ingenieros de configuración del NOC InterNexa se decidió que el proyecto se desarrollaría haciendo uso del lenguaje de programación de ***Python***. Se eligió esta herramienta debido a que su uso no requiere el pago de una licencia, adicionalmente cuenta con buena documentación y no requiere la instalación de software

adicional en cada equipo de la red MPLS Huawei de InterNexa. Además, el estudiante en práctica contaba con buenas bases en el lenguaje de programación de Python lo cual fue de suma importancia a la hora de la selección, debido a que se ahorró tiempo de aprendizaje de un nuevo lenguaje en el desarrollo del proyecto.
Cabe resaltar que a pesar de no ser seleccionadas como herramientas de trabajo NetConf y Ansible servirán como referencia en el desarrollo de este proyecto.

Después de realizada esta elección se debía tener en cuenta los siguientes aspectos:

- Documentación del desarrollo: Se dejó claro que el desarrollo de software además de automatizar las tareas presentes en los servicios CE e IPNG, debería estar bien estructurado y con sus respectivos comentarios que documentaran el código. Así mismo, el desarrollador debía proporcionar a los usuarios el manual de configuración y el manual de uso.

- Seguridad: También se definió que el software desarrollado debería tener buenas prácticas de seguridad informática, debido a que desde éste se tiene acceso a información sensible de la empresa.

4.3. DESARROLLO DEL SOFTWARE

Una vez definida la herramienta a utilizar, los ingenieros de configuración procedieron a especificar los requisitos que debía tener el software de automatización de configuración de los servicios CE e IPNG en la red MPLS de InterNexa en Colombia. Posteriormente, después de tener claro los requisitos, se procedió a diseñar, desarrollar e implementar el software de automatización.

4.3.1. REQUISITOS PARA LA GESTIÓN DE LOS SERVICIOS

El fundamento de este proyecto es la automatización de la configuración de los servicios CE e IPNG en la red MPLS Huawei de InterNexa en Colombia. Para esto los ingenieros de configuración del NOC definieron los siguientes requisitos:

- El desarrollo de software debía tener una interfaz gráfica de usuario (GUI) donde el ingeniero pudiera ingresar y leer los datos de forma cómoda.

- El software implementado debía generar conexiones seguras entre el gestor (equipo de gestión) y el agente (equipo a configurar), garantizado que no existieran fallos de seguridad, ni perdida de información.

- El software debía generar los comandos necesarios para cada configuración a partir de la información suministrada por el ingeniero.

- El software debía estar en la capacidad de enviar comandos a los equipos a configurar y esperar las respuestas a estos comandos.

- El software desarrollado debía ser interactivo, de tal manera que en cada configuración comunicara al usuario su estado y le preguntara si todo es correcto para así poder continuar.

- La herramienta debía ser escalable, de tal manera que, si se añaden nuevos equipos a la red MPLS Huawei de Internexa en Colombia, el software siga haciendo su trabajo con todos los equipos de la red.

- La herramienta debía ser eficiente, esto significa que el tiempo que tarde en realizar una configuración sea menor al tiempo que tarda el ingeniero en realizar la configuración de forma manual.

4.3.2. DISEÑO DE LA INTERFAZ GRAFICA DE USUARIO (GUI)

Teniendo en cuenta cada uno de los requisitos presentados por los ingenieros del NOC, se inició con el diseño del software. Se comenzó realizando un boceto referenciando cual iba a ser la interfaz gráfica de usuario (GUI, Graphical User Interface) con la que iba a interactuar el ingeniero. Dentro de la GUI debían existir tres elementos principalmente: un área para el ingreso de información (ENTRADAS), otra área para la visualización de resultados de operación (SALIDAS) y un área donde se

controlará el proceso de automatización (CONTROLES). Tal como se muestra en la Figura 14.

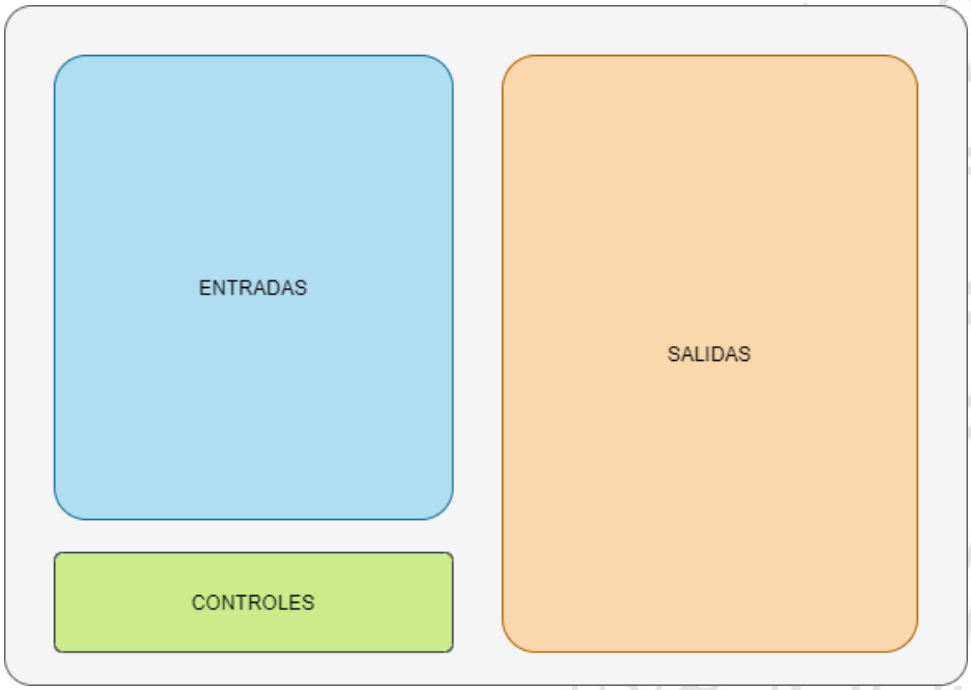


Figura 14: Boceto de la GUI de la herramienta. Elaboración propia.

Las entradas hacen referencia a todos los datos necesarios que el ingeniero va a ingresar para iniciar con la configuración de un servicio. Estos datos indispensables fueron presentados en la sección 4.1.3 de este informe.
Las salidas hacen referencia a toda la información que los equipos a ser configurados entregan al usuario. Después de la ejecución de cada comando, el equipo que está siendo configurado emite una respuesta, esta respuesta va a ser plasmada en el campo de salidas.
Los controles son los botones que permiten iniciar o pausar acciones dentro del software de automatización.

Después de tener claro como seria la estructura de la GUI del software se procedió a implementarla en Python. Existe una amplia gama de librerías que implementan interfaces gráficas de usuario en Python, entre ellas, destacan las siguientes:

- Tkinter
- WxPython

- PyQT

4.3.2.1. Librería Tkinter

Tkinter es fácil de usar, es multiplataforma y, además, viene incluido con Python en su versión para Windows, para Mac y para la mayoría de las distribuciones GNU/Linux. Se le considera el estándar de facto en la programación GUI con Python [15].

Tkinter es un binding (adaptación de una biblioteca para ser usada en un lenguaje de programación distinto de aquel en el que ha sido escrita) de la biblioteca Tcl/Tk que está también disponible para otros lenguajes como Perl y Rubí.

Una de las ventajas de elegir Tkinter es que, dado que viene por defecto, existe una gran cantidad de documentación y de recursos, tanto códigos como libros de referencia. Además, cuenta con una comunidad antigua y activa, donde hay muchos usuarios que pueden ayudar en caso de dudas [15].

4.3.2.2. Librería WxPython

WxPython se implementa como un módulo de extensión de Python. Es un contenedor de código abierto para la biblioteca GUI multiplataforma WxWidgets (anteriormente conocida como WxWindows). Con WxPython, usted como desarrollador puede crear aplicaciones nativas para Windows, Mac OS y Unix [16].

4.3.2.3. Librería PyQT

PyQT implementa la biblioteca Qt para el marco de desarrollo de GUI. Actualmente, PyQT está disponible para Unix / Linux, Windows, Mac OS X y Sharp Zaurus. Combina Python y Qt y depende del programador decidir si crear un programa codificando o usando Qt Designer para crear los elementos de la GUI.

Está disponible tanto en licencia comercial como en licencia GPL (Licencia Publica General). Aunque algunas características pueden no estar disponibles en la versión gratuita, si la aplicación es de código abierto, puede usarse bajo la licencia gratuita [16].

En este caso se eligió la librería ***Tkinter*** para el desarrollo de la GUI del proyecto, debido a que esta librería es multiplataforma y, además, viene incluida con Python.

Para desarrollar una GUI haciendo uso de esta librería se debe crear inicialmente una ventana que va a ser la raíz del proyecto. Dentro de esa ventana se podrán insertar más elementos (widgets), tales como:

- Botones
- Campos de entrada
- Campos de salida
- Botones de control
- Caja de listas
- Frames
- Menú de opciones
- Botones de menú
- Ventanas emergentes
- Botón de radio
- Barra de desplazamiento
- Spinbox

Cada uno de los anteriores elementos se puede añadir mediante código en Python. Y para personalizar cada elemento se deben cambiar sus respectivos parámetros, también mediante código.

El software fue organizado por pestañas, donde cuenta con cinco secciones, cada una con su respectiva funcionalidad. Las pestañas fueron nombradas: Principal, Configuración, Ayuda, Historial y Troncales Protegidas.

- Ventana principal: Dentro de la pestaña "Principal" va a estar toda la interfaz necesaria para proceder a configurar los equipos involucrados en un servicio, con sus respectivos campos de entrada, salida y sus controles (Ver Figura 15).

- Ventana de configuración: Dentro de la pestaña "Configuración" se encuentran opciones para modificar (Añadir o eliminar) la información almacenada en la base de datos que va asociada al software de automatización (Ver Figura 16).

- Ventana de ayuda: En la pestaña "Ayuda" se aloja información de contacto y enlaces para acceder a los manuales de la herramienta (Ver Figura 17).

- Venta de historial: En la pestaña "Historial" se aloja información de cada configuración realizada haciendo uso del software de automatización. Por cada configuración, en esta pestaña se

guardaran los siguientes datos: Fecha y hora, usuario, tiempo de ejecución y una breve descripción de la configuración (Ver Figura 18).

- Ventana de troncales protegidas: En la pestaña "Troncales Protegidas" se encuentran opciones para modificar (Añadir o eliminar) la información almacenada en la base de datos sobre equipos que poseen información importante y no podrán ser modificados desde el software, sino que deberán ser modificados de forma manual. Además, en esta pestaña se puede visualizar la información almacenada sobre los equipos protegidos en la base de datos (Ver Figura 19).

Cabe resaltar que la ventana "Principal" (Figura 15) cuenta con una serie de ventanas emergentes, que podrán ser informativas o de decisión. Las informativas solamente muestran algunos datos y esperan a que el usuario las cierre (Ver Figura 20 a). Por otro lado, las ventanas de decisión se dividen en dos. Están las ventanas de decisión que únicamente le permiten al usuario aceptar o denegar una acción (Ver Figura 20 b). Y están las ventanas de decisión que solicitan que el usuario ingrese algún valor para así continuar o abortar una acción (Ver Figura 20 c).

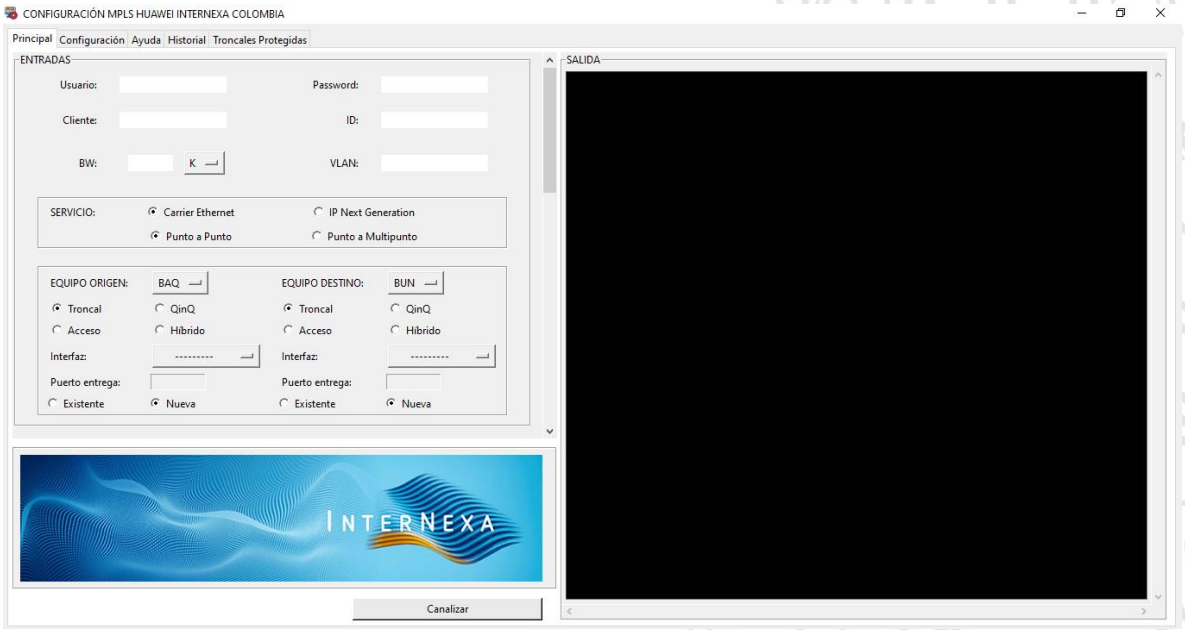


Figura 15: Pestaña Principal de la GUI Final.

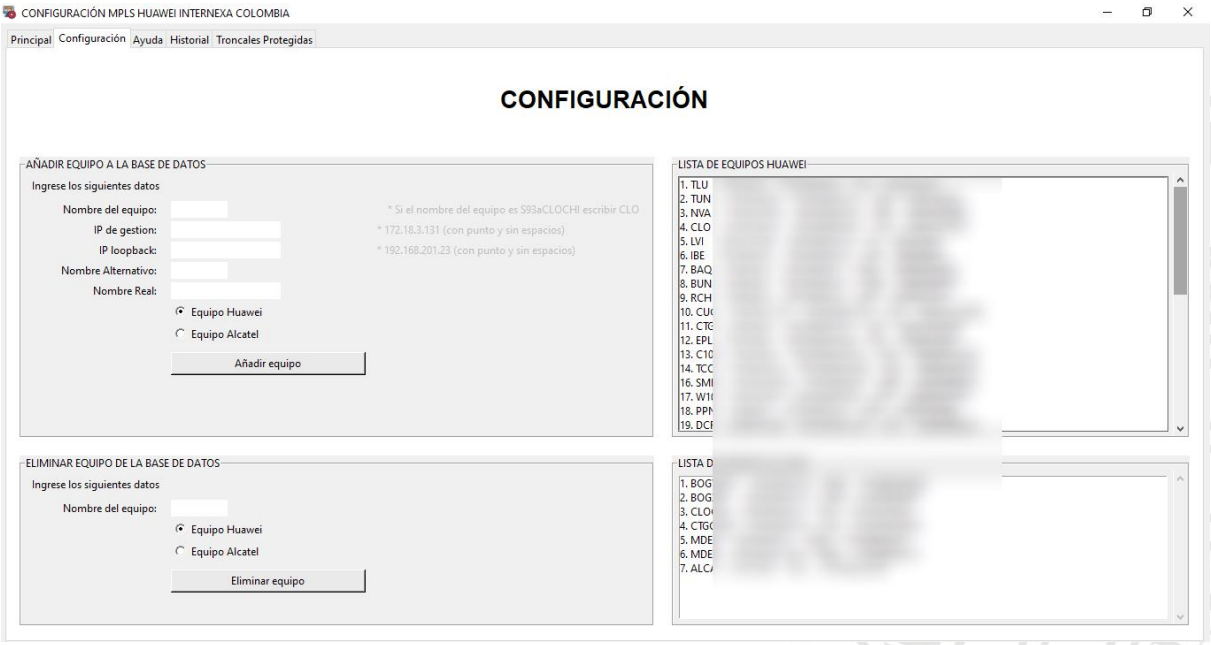


Figura 16: Pestaña de Configuración de la GUI Final.

Figura 17: Pestaña de Ayuda de la GUI Final.

Figura 18: Pestaña de Historial de Ejeción de la GUI final.

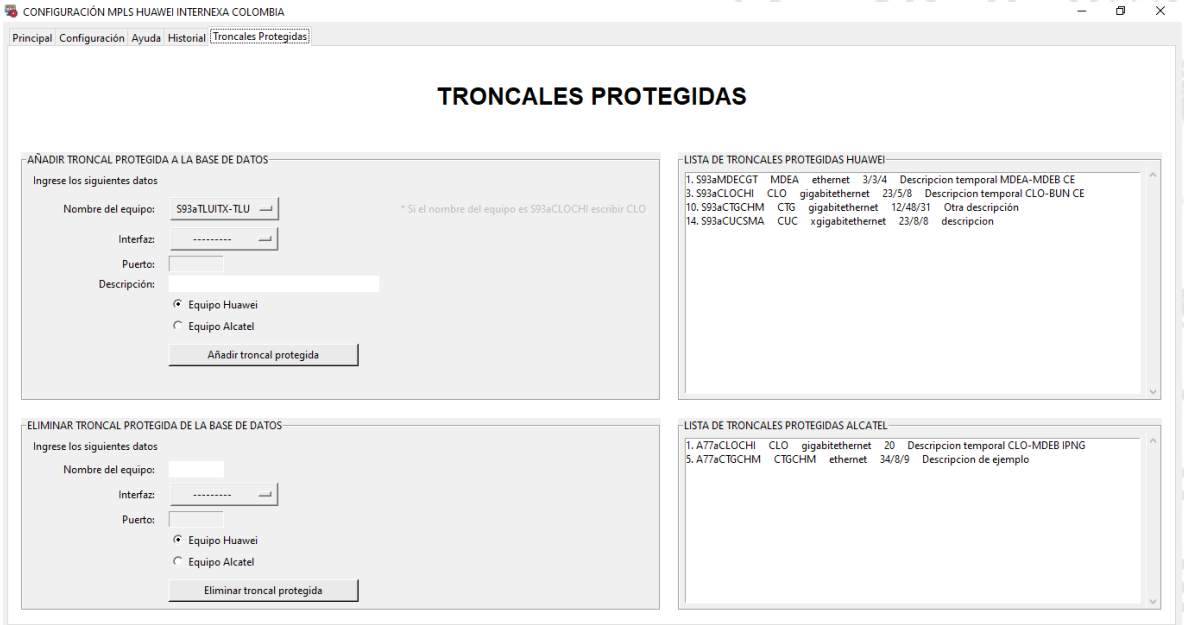


Figura 19: Pestaña de Troncales Protegidas de la GUI Final.

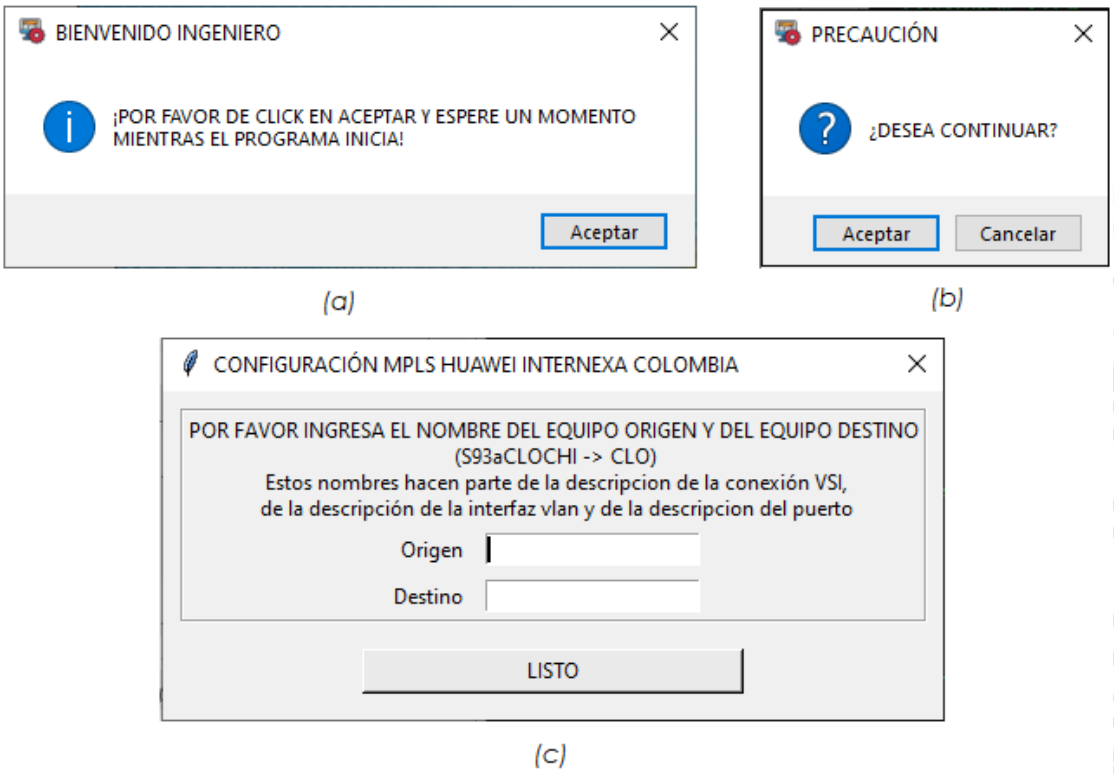


Figura 20: Ventanas emergentes.
(a)Ventana informativa. (b) Ventana de decisión. (c) Ventana de decisión con entradas.

4.3.3. CONEXIÓN REMOTA

El fundamento de este software de gestión automatizada es la conexión remota entre el computador del ingeniero de configuración y el equipo de la red MPLS que se va a configurar.
Existen diversos protocolos que ayudan a crear conexiones remotas, tales como: SSH, Telnet y FTP.
Como se dijo en la sección 4.1.2, actualmente en el proceso de configuración manual se establecen conexiones seguras haciendo uso del protocolo SSH, por tal motivo el software desarrollado también hizo uso de este protocolo de conexión, ya que brinda seguridad en los datos procesados.
En Python se pueden generar conexiones SSH haciendo uso del módulo paramiko, propio de este lenguaje de programación.

La librería Paramiko de Python proporciona capacidades para automatizar tareas que se quieren ejecutar en un host remoto, como reiniciar servicios, realizar actualizaciones, capturar archivos de registro o realizar configuraciones. Es una de las principales herramientas a la

hora de trabajar con redes de telecomunicaciones, debido a que permite establecer conexiones SSH. Este módulo permite enviar y recibir información desde un host remoto, proporcionando funcionalidad de cliente y servidor, donde el cliente va a tener total control del servidor al que acceda.
Si se hace buen uso de esta herramienta y se potencializa su capacidad mediante Python, es posible el desarrollo de un software de gestión que permita la automatización de tareas en una red de telecomunicaciones. Cabe resaltar que para hacer uso de esta librería se debe instalar dentro del entorno de Python, ya que ésta no viene incluida por defecto [17].

Inicialmente se procedió con la instalación del módulo Paramiko en el computador donde se realizó el desarrollo. Para esto fue necesario hacer uso del comando *"pip install paramiko" dentro de la terminal de Windows, tal como se muestra en la* Figura 21.
El comando pip (Python Install Package) es una utilidad que facilita la descarga e instalación de paquetes en Python.

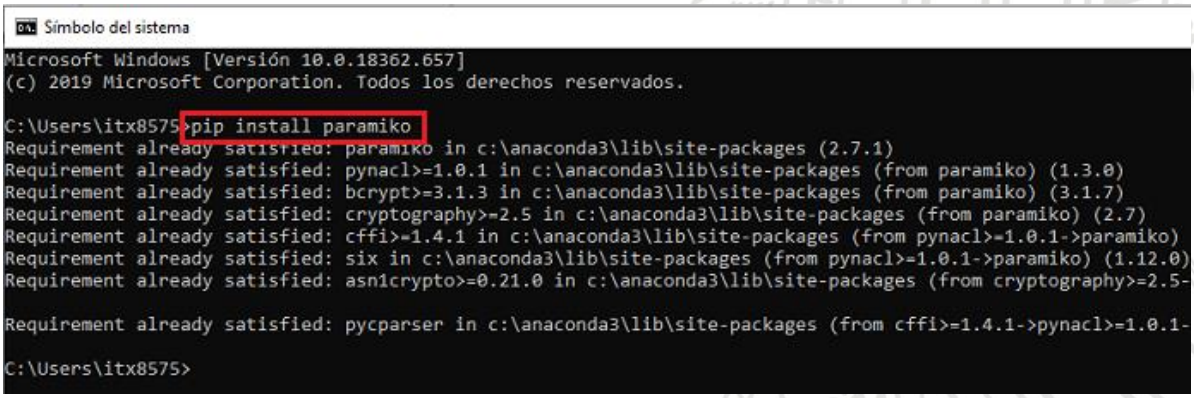


Figura 21: Comando para instalar la librería Paramiko de Python.

4.3.3.1. Establecimiento de conexión SSH

Una vez instalada la librería Paramiko ya se tenía a la disposición sus métodos y todas sus funcionalidades.
Paramiko ofrece varias formas de conexión SSH con servidores remotos. En el desarrollo de este proyecto, se establecieron conexiones cliente/servidor, donde se proporcionaron canales específicos de comunicación entre ambas partes. A continuación, se

describe la forma en que Paramiko pone en marcha la comunicación entre un cliente y un servidor:

- Para establecer conexión con un servidor de manera remota, Paramiko define el tipo de conexión haciendo uso de la instrucción "*paramiko.SSHClient()*", donde inicia un cliente SSH.

- Una vez definido el tipo de conexión e iniciado el cliente SSH, Paramiko fija una política por defecto para localizar la llave del host localmente haciendo uso de la instrucción "*conexion.set_missing_host_key_policy(paramiko.AutoAddPolicy())*". Después de esto se procede a establecer la conexión con el servidor remoto haciendo uso de la instrucción "*conexion.connect(ip_gestion, 22, ssh_usuario, ssh_clave)*". Donde sus parámetros están conformados por la IP de gestión del equipo al cual se va a conectar, el puerto 22 por donde se establece la conexión y finalmente los datos de usuario y contraseña para validar los permisos que el usuario tiene sobre esa conexión.

- Una vez establecida la conexión, Paramiko procede a crear un canal de comunicación entre el cliente y el servidor, haciendo uso de la instrucción "*conexion.invoke_shell()*".

El proceso de configuración explicado es representado en la Figura 22.

```
#Inicia un cliente SSH
conexion = paramiko.SSHClient()

# Establecer política por defecto para localizar la llave del host localmente
conexion.set_missing_host_key_policy(paramiko.AutoAddPolicy())

#Establecimiento de la conexión SSH
conexion.connect(ip_gestion, 22, ssh_usuario, ssh_clave)

#Generación de un Canal de comunicación
channel = conexion.invoke_shell()
```

Figura 22: Configuración de conexión remota SSH en Python con Paramiko.

4.3.3.2. Mantenimiento de conexión SSH

En esta etapa se procederá a generar y enviar los comandos generados desde el software hasta el host remoto. Cada comando ejecutado será sometido a verificaciones, de tal manera que se tenga certeza de que el comando fue enviado y recibido con éxito en el equipo que está siendo configurado. También, se procederá a verificar la salida o resultado obtenido después de la ejecución de cada comando, de tal manera que se garantice que se recibirán todos los datos correspondientes.

4.3.3.2.1. Generación de comandos

Cuando Paramiko ya ha creado el canal de comunicación, el usuario está en la disposición de poder enviar comandos hacia el servidor remoto. Pero antes, se deben definir qué tipo de comandos se van a enviar. Los comandos se generan a partir de la información que el usuario ingresa en los campos de entrada de la interfaz gráfica de usuario. En el caso de los servicios CE e IPNG ofertados por InterNexa existen variantes en los comandos utilizados a la hora de poner en marcha alguno de los servicios mencionados.

Inicialmente para configurar un equipo, el usuario debe ejecutar el comando "system-view", con este comando el sistema ya entra en modo configuración.
Una vez el sistema esté en modo configuración se procede a ejecutar el comando para desplegar el listado de túneles de comunicación entre dos equipos: "display tunnel-policy".

Después de esto se procederá a enviar el comando "mpls l2vc ip_loopback id_servicio mtu 9000 tunnel-policy Tunel raw" si se trata de un servicio VLL, o los comandos "vsi cliente Id static", "pwsignal ldp", "vsi-id Id" y ""peer lista_ip_loopback tnl-policy Tuneles" si se trata de un servicio VSI.

Luego de haber configurado la conexión VLL o VSI, se procede a configurar el respectivo puerto de cada equipo, haciendo uso de los comandos presentados a continuación:

- Troncal:
  - vlan *num_vlan*
  - description *cliente* [*id_servicio*] *origen_ destino_tipo_servicio*

- interface vlan *num_vlan*
- description *cliente – origen destino tipo_servicio [id_servicio]*
- interface *interfaz puerto*
- port trunk allow-pass vlan *num_vlan*

- Acceso:
  - display current interface *interfaz puerto*
  - display interface *interfaz puerto*
  - interface *interfaz puerto*
  - clear configuration this
  - undo portswitch
  - description *cliente – origen destino bw tipo_servicio [id_servicio]*

- QinQ:
  - interface *interfaz puerto*
  - description *cliente – origen destino bw tipo_servicio [id_servicio]*
  - set flow-stat interval 10
  - port link-type dot1q-tunnel
  - port default vlan *num_vlan*
  - stp disable

- Hibrido:
  - interface *interfaz puerto*
  - description *cliente – origen destino bw tipo_servicio [id_servicio]*
  - port link-type hybrid
  - set flow-stat interval 10
  - port hybrid tagged vlan *num_vlan*
  - stp disable
  - broadcast-suppression 20"

Una vez que todos los equipos involucrados en un servicio han sido configurados se procede a realizar una prueba de establecimiento para la cual se hace uso de los siguientes comandos:

- Puerto nuevo:
  - loopback internal
  - undo shutdown
  - dis this interface

- dis mpls l2vc id_servicio
- undo loopback
- shutdown
- quit
- save

- <u>Puerto existente:</u>
  - dis mpls l2vc id_servicio
  - quit
  - save

En la Figura 23 se observa de manera resumida el procedimiento para la generación y envío de comandos que permiten configurar los servicios mencionados.

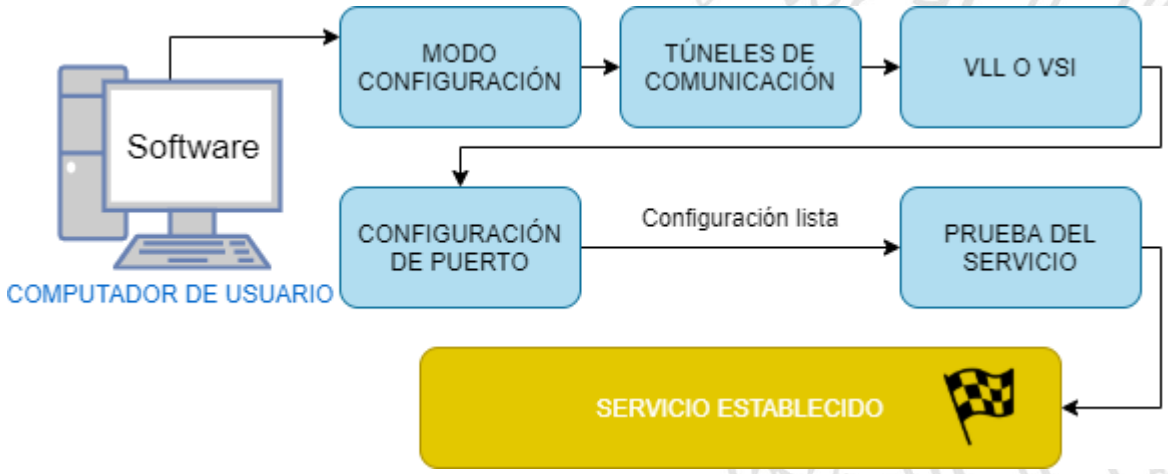


Figura 23: Procedimiento para la configuración de servicios VLL/VSI.

Por ejemplo, si se quisiera configurar un servicio CE en VLL con ID 8888 y vlan 3333, entre los equipos de Cartagena (CTG) y Neiva (NVA), donde el equipo de CTG está en troncal con su puerto nuevo ethernet 2/0/32 y el equipo de NVA está en acceso con su puerto eth-trunk 456, los comandos generados serían los siguientes:

- Equipo CTG:
  - <S93CTGITX> ***system-view***
  - [S93CTGITX] display tunnel-policy

- [S93CTGITX] mpls l2vc 192.168.201.4 8888 mtu 9000 tunnel-policy ctg-nva raw
- [S93CTGITX] vlan *3333*
- [S93CTGITX] description *Cliente* [*8888*] *ctg_ nva_CE*
- [S93CTGITX] interface vlan *3333*
- [S93CTGITX] description *Cliente – ctg nva CE [8888]*
- [S93CTGITX] interface *ethernet 2/0/32*
- [S93CTGITX] port trunk allow-pass vlan *3333*

- Equipo NVA:
  - <S93NVAITX> ***system-view***
  - [S93NVAITX] display tunnel-policy
  - [S93NVAITX]mpls l2vc 192.168.201.6 8888 mtu 9000 tunnel-policy nva-ctg
  - [S93NVAITX] display current interface *eth-trunk 456*
  - [S93NVAITX] display interface *eth-trunk 456*
  - [S93NVAITX] interface *eth-trunk 456*
  - [S93NVAITX] clear configuration this
  - [S93NVAITX] undo portswitch
  - [S93NVAITX] description *Cliente – nva ctg BW CE [8888]*

Prueba del servicio:

- Equipo CTG:
  - [S93CTGITX] loopback internal
  - [S93CTGITX] undo shutdown
  - [S93CTGITX] dis this interface

- Equipo NVA:
  - [S93NVAITX] loopback internal
  - [S93NVAITX] undo shutdown
  - [S93NVAITX] dis this interface
  - [S93NVAITX] dis mpls l2vc id_servicio
  - [S93NVAITX] undo loopback
  - [S93NVAITX] shutdown
  - [S93NVAITX] quit
  - <S93NVAITX> ***save***

- Equipo CTG:
  - [S93CTGITX] undo loopback
  - [S93CTGITX] shutdown

- [S93CTGITX] quit
- <S93CTGITX> ***save***

4.3.3.2.2. Envío de comandos

Paramiko, proporciona una serie de métodos que son de gran ayuda al momento de enviar comandos a un servidor remoto y de recibir la respectiva respuesta al comando ejecutado.

Los principales modos de operación son la conexión basada en canales (Channel-Based) y la conexión basada en una única capa (Shell-Based). En este proyecto se hizo uso de la conexión basada en una única capa, de tal manera que cuando el software estableciera la conexión con el servidor remoto, únicamente se creara un canal de comunicación para enviar todos los comandos y recibir todas las respectivas respuestas (Ver Figura 24).

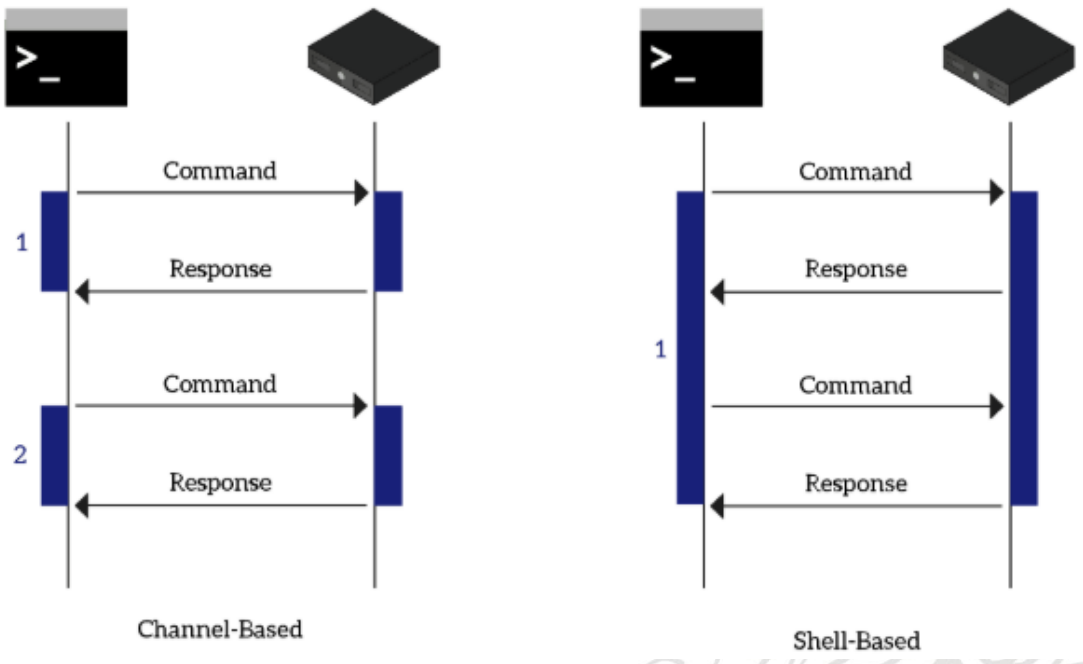


Figura 24: Paramiko en modo canal y en modo shell.

Se considera buena práctica utilizar la conexión basada en canales, pero para este proyecto si se hubiese creado un canal para cada

comando y su respectiva respuesta, los tiempos de ejecución hubiesen aumentado.

La instrucción "*channel.send(comando + '\n')*" hace uso del canal creado dentro de la conexión SSH para enviar mediante el método 'send()' el comando específico. Una vez enviado el comando el usuario debe esperar su respectiva respuesta. La instrucción "*channel.recv_ready()*" hace uso del método 'recv_ready()' y devuelve 'True' si los datos están almacenados en el búfer y listos para leerse desde este canal o devuelve 'False' si se requiere esperar antes de que lleguen más datos.
Una vez que la instrucción "*channel.recv_ready()*" devuelva 'True' ya será posible leer los datos recibidos en el canal. Para leer los datos recibidos se hace uso de la instrucción "channel.recv(10192)", donde el número que se le pasa como parámetro hace referencia al número máximo de bytes que se pretende leer. Para este proyecto se sabe que los datos resultantes al ejecutar un comando no van a sobrepasar los 10024 bytes, por tal razón se coloca éste número como parámetro. Todo lo anterior puede ser visto en la Figura 25.

```
#Envia el comando por el canal de la conexion SSH
channel.send(comando)
while not channel.send_ready():
    #El comando no ha llegado completamente hasta el servidor remoto
    #Espera un segundo
    time.sleep(1)

output=channel.recv(10192)
while not channel.recv_ready():
    #Los datos aun no están almacenados en el bufer
    #Espera un segundo
    time.sleep(1)
    output=channel.recv(10192)
```

Figura 25: Envío de comandos a través del canal de la conexión SSH.

4.3.3.2.3. Verificación de envío y recepción de comandos

Una vez que el comando ha sido enviado de manera remota haciendo uso del método send(), se hace uso del método send_ready(), el cual garantiza que el comando llegue correctamente al servidor remoto.
Después de ejecutar el comando, el servidor con el que se está comunicando el usuario comenzará a retornar bytes que contienen información del resultado correspondiente a la ejecución del

comando. Cabe resaltar que esa información no se retorna instantáneamente, por lo que es necesario aplicar el método recv_ready(). Este método asociado al canal de comunicación garantiza que el servidor remoto podrá entregarle al software toda su información sin que haya perdidas.

En la Figura 26 se observa el proceso por el cual debe pasar cada comando para garantizar su correcto envío, adecuada recepción y correspondiente verificación del resultado obtenido.

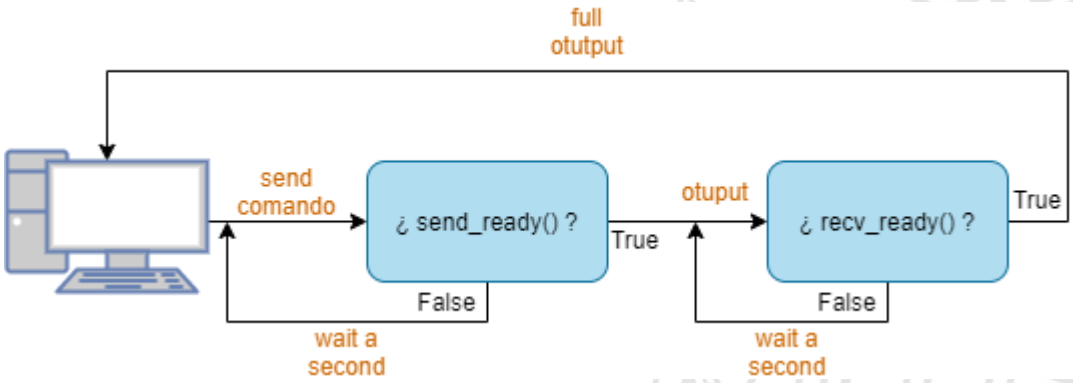


Figura 26: Verificación de cada comando.

4.3.3.3. Finalización de conexión SSH

Una vez que ya se haya configurado todos los equipos involucrados en un servicio, se procede a hacer la respectiva prueba de establecimiento, que simplemente consiste en enviar otra serie de comandos en forma ordenada. Cuando ya se tenga la prueba lista es posible cerrar las conexiones que se crearon con los equipos que fueron configurados. Para realizar la finalización de la conexión SSH en paramiko se hace uso del método close() acompañado de la conexión que se desea cerrar. Por ejemplo, si se configuraron tres equipos, es claro que se crearon tres conexiones SSH con cada equipo respectivamente, conexión_1, conexión_2 y conexión_3. Para finalizar conexión con esos equipos simplemente se deben ejecutar las siguientes líneas: conexión_1.close(), conexión_2.close(), conexión_2.close().

4.3.4. MÓDULOS ADICIONALES

Para que el software de automatización tuviese un funcionamiento óptimo además de hacer uso de las librerías Tkinter y Paramiko, se hizo uso de módulos adicionales presentados a continuación:

4.3.4.1. Librería Threading

Con el uso de la librería threading, el software puede emular que se están ejecutando varias tareas o procesos al mismo tiempo. Cada hilo (se define hilo como un proceso o subproceso) tiene su propio flujo de control. Mientras que un hilo podría estar leyendo datos de un archivo, otro hilo puede mantener actualizada la pantalla. Para evitar que dos hilos accedan a la misma estructura de datos interna al mismo tiempo, Python usa un bloqueo global de intérprete. Solo un hilo puede ejecutar código Python al mismo tiempo; Python cambia automáticamente al siguiente subproceso después de un corto período de tiempo, o cuando un subproceso hace algo que puede llevar un tiempo (como esperar que llegue el siguiente byte a través de un socket de red o leer datos de un archivo) [18].
En el software de automatización fue muy útil esta librería debido a que mientras un hilo se encargaba de mantener actualizada la GUI, otro hilo se encargaba de la ejecución de las tareas dentro del flujo del programa.

Cuando el software inicia, carga completamente su GUI, y en el momento que el usuario presione el botón "Canalizar", éste realizará un llamado a la función donde se activará el hilo secundario que se va a encargar de la ejecución de los procesos necesarios dentro del software (Ver Figura 27).
Mientras que el hilo secundario se encarga de la ejecución del código necesario para la configuración de un conjunto de servidores remotos, el hilo principal permite que la GUI no quede bloqueada y funcione correctamente.

```
def fcn_thread():
    Hilo_Secundario = Thread(target=Ejecucion_Configuracion_Servicio)
    Hilo_Secundario.start()
```

Figura 27: Hilo secundario.

El hilo secundario se encargará de la ejecución de la función "Ejecucion_Configuracion_Servicio()" la cual contiene los procesos necesarios para configurar un servicio (Ver Anexo 1). Una vez que esta función termine de ejecutarse dentro del hilo secundario, éste terminará su trabajo y quedará inactivo hasta que se vuelva hacer un llamado a la función "fcn_thread()".

En la Figura 28 se puede observar la distribución tanto del hilo principal como del hilo secundario dentro del flujo del programa.

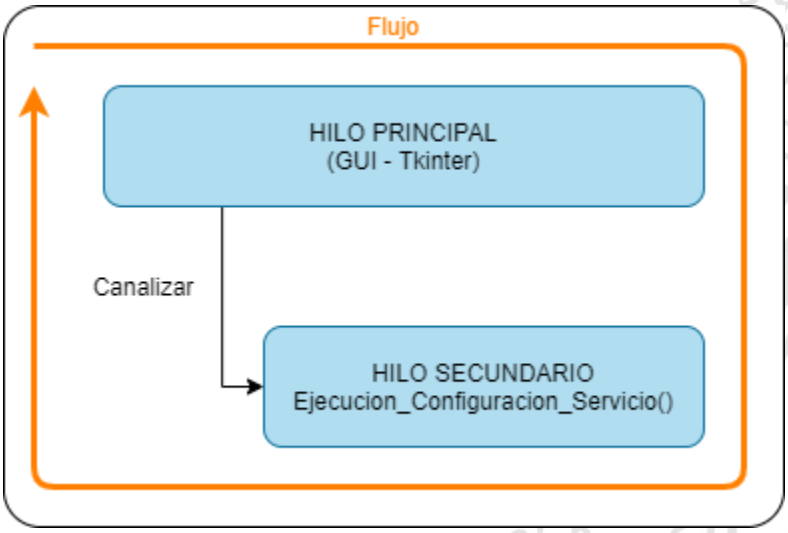


Figura 28: Hilos dentro del software.

4.3.4.2. MySQL

El estándar MySQL constituye la base de datos (BD) de software libre más popular del mercado. Es un sistema de gestión de base de datos que ofrece los mecanismos para añadir, acceder y procesar información almacenada en una base de datos.

MySQL es un software libre, lo cual significa que cualquiera puede hacer uso del código fuente que lo constituye para usarlo libremente o incluso modificarlo sin restricciones. Esta herramienta ofrece un modelo de cliente/servidor consistente en un servidor SQL multihilo que es capaz de soportar diferentes clientes, librerías, herramientas administrativas y APIs (Application Programming Interface - Interfaz de programación de aplicaciones) [19].

El software de automatización hizo uso de una base de datos en MySQL, donde se registró la información asociada a cada equipo de la red MPLS Huawei y Alcatel de InterNexa en Colombia.

La base de datos va a estar alojada en el sistema de almacenamiento en la nube de InterNexa y dentro de la base de datos van a existir cinco tablas inicialmente:

- La primera con información asociada a los equipos de la marca Huawei (Ver Figura 29).
- La segunda con información asociada a los equipos de la marca Alcatel (Ver Figura 30).
- La tercera con información de las troncales protegidas asociadas a los equipos Huawei (Ver Figura 31***Figura 31: Tabla de Troncales Protegidas Huawei.***)
- La cuarta con información de las troncales protegidas asociadas a los equipos Alcatel (Ver Figura 32)
- La quinta con información del historial de las configuraciones realizadas haciendo uso del software (Ver Figura 33)

Indice	Nombre	IP_Gestion	IP_Loopback	Nombre_Alternativo	Nombre_Real
1	TLU			TLU	S93aTLUITX
2	TUN			TUN	S93aTUNITX
3	NVA			NVA	S93aNVAEBO
4	CLO			CLO	S93aCLOCHI
5	LVI			LVI	S93aLVIISA
6	IBE			IBE	S93aIBEISA
7	BAQ			BAQ	S93aBAQNOG
8	BUN			BUN	S93aBUNPTC
9	RCH			RCH	S93aRCHITX
10	CUC			CUC	S93aCUCSMA
11	CTG			CTG	S93aCTGCHM
12	EPL			EPL	S93aBOGEPL
13	C100			C100	S93aBOGC100
14	TCO			TCO	S93aBOGTCO
16	SMR			SMR	S93aSMRBCH
17	W10			W10	S93aBOGW10
18	PPN			PPN	S93aPPNSBN

Figura 29: Tabla de equipos Huawei en la BD.

Indice	Nombre	IP Gestion	Nombre_Alternativo	Nombre_Real
1	BOGWSO		BOG	A77aBOGWSO
2	BOGZFT		BOG	A77aBOGZFT
3	CLOCHI		CLO	A77aCLOCHI
4	CTGCHM		CTG	A77aCTGCHM
5	MDEA		MDE	A77aMDECGT
6	MDEB		MDE	A77bMDECGT

Figura 30: Tabla de equipos Alcatel en la BD.

Indice	Nombre_Real	Nombre	Interfaz	Lag_Puerto	Descripcion
1	S93aMDECGT	MDEA	ethernet	3/3/4	Descripcion temporal MDEA-MDEB CE
3	S93aCLOCHI	CLO	gigabitethernet	23/5/8	Descripcion temporal CLO-BUN CE
10	S93aCTGCHM	CTG	gigabitethernet	12/48/31	Otra descripción
14	S93aCUCSMA	CUC	xgigabitethernet	23/8/8	descripcion

Figura 31: Tabla de Troncales Protegidas Huawei.

Indice	Nombre_Real	Nombre	Interfaz	Lag_Puerto	Descripcion
1	A77aCLOCHI	CLO	gigabitethernet	20	Descripcion temporal CLO-MDEB IPNG
5	A77aCTGCHM	CTGCHM	ethernet	34/8/9	Descripcion de ejemplo

Figura 32: Tabla de Troncales Protegidas Alcatel.

Indice	Fecha	Usuario	Tiempo	Descripcion
7	2020-06-01 20:49:39	ferazo	21.879706859588623	PRUEBA_VLL_LUNES [406] PEI_CTG_CE
8	2020-06-01 20:51:22	ferazo	46.23112750053406	PRUEBA_VSI_LUNES [1789] MTR_PEI_CE
9	2020-06-02 15:45:05	ferazo	24.348188161849976	PRUEBA_MARTES_VLL [3245] CUC_SMR_CE
10	2020-06-02 15:47:28	ferazo	55.48730754852295	PRUEBA_MARTES_VLL [3245] NVA_PPN_CE
11	2020-06-03 09:33:17	ferazo	355.55935525894165	PRUEBA_MIERCOLES_VLL [678] BAQ_BUN_CE
12	2020-06-03 10:00:29	ferazo	1471.4691798686981	PRUEBA_MIERCOLES_VLL [948] LVI_MDEA_CE
13	2020-06-03 16:11:37	ferazo	61.14621186256409	PRUEBA_IPNG [12345] LVI_PPN_IPNG

Figura 33: Tabla de Historial de configuraciones.

Con el fin de proteger información sensible se censuraron los campos correspondientes a la IP de gestión y la IP loopback de cada equipo. Cabe resaltar que cada tabla es independiente, y realizar modificaciones en una de ellas no tendrá repercusiones en la otra.

4.3.5. RESUMEN DEL FUNCIONAMIENTO DEL PROGRAMA

A continuación, se ilustra el procedimiento a realizar cada vez que se inicie la configuración de un servicio CE o IPNG haciendo uso del software de automatización desarrollado en este proyecto:

- Primero se validan los campos de entrada de la interfaz gráfica de usuario. Inicialmente todos los campos deben estar llenos para poder continuar.

- Se valida que en los campos de entrada numéricos (BW, VLAN, ID, IP) no aparezcan caracteres alfanuméricos que puedan intervenir en el correcto funcionamiento del software.

- Después de esto se verifica que datos como la VLAN, ID, puertos y ancho de banda del servicio estén dentro de los rangos establecidos por la compañía.

- Posteriormente el software ingresa a la base de datos MySQL donde va a tomar la IP de gestión y la IP loopback correspondiente al equipo que está siendo configurado.

- Haciendo uso de la IP de gestión, el nombre de usuario, su contraseña y el módulo Paramiko, el programa procede a realizar la conexión remota SSH con el equipo referenciado por la IP de gestión.

- Si la conexión se estableció correctamente se procede a crear un canal de comunicación entre el software y el equipo que está siendo configurado. De lo contrario, si la conexión no se estableció, significa que el usuario o contraseña son incorrectos o que el software no tiene acceso a internet. En esta situación el usuario será notificado a través de una ventana emergente.

- Una vez creado el canal de comunicación entre el software y el equipo a ser configurado, se procede a generar uno a uno los

comandos necesarios para configurar el equipo, dependiendo de los requerimientos solicitados por el cliente.

- Cuando los comandos se han generado correctamente se procede a enviar uno a uno cada comando. Teniendo en cuenta que después de la ejecución de cada comando se va a recibir una respuesta, la cual va a ser plasmada en la GUI.

- Una vez que los equipos involucrados en el servicio han sido configurados, se procede a realizar una prueba de verificación, para garantizar que el servicio se haya establecido correctamente. Y en este punto finaliza la configuración de un equipo de red presente en los servicios CE e IPNG.

4.4. DESCRIPCIÓN DE LA PLATAFORMA DESARROLLADA

Realizar la descripción del desarrollo de un software suele ser complicado, sobre todo si se trata de explicar las líneas del código elaborado. Para esto, organizaciones como ISO (International Organization for Standardization), IEC (International Electrotechnical Commission) y la IEEE (Insitute of Electrical and Electronics Engineers) han establecido un estándar para la descripción de la arquitectura de sistemas y productos software.

Para realizar una descripción completa, detallada y legible del software desarrollado se hizo uso de la norma *ISO/IEC/IEEE 42010* (Ver Figura 34).

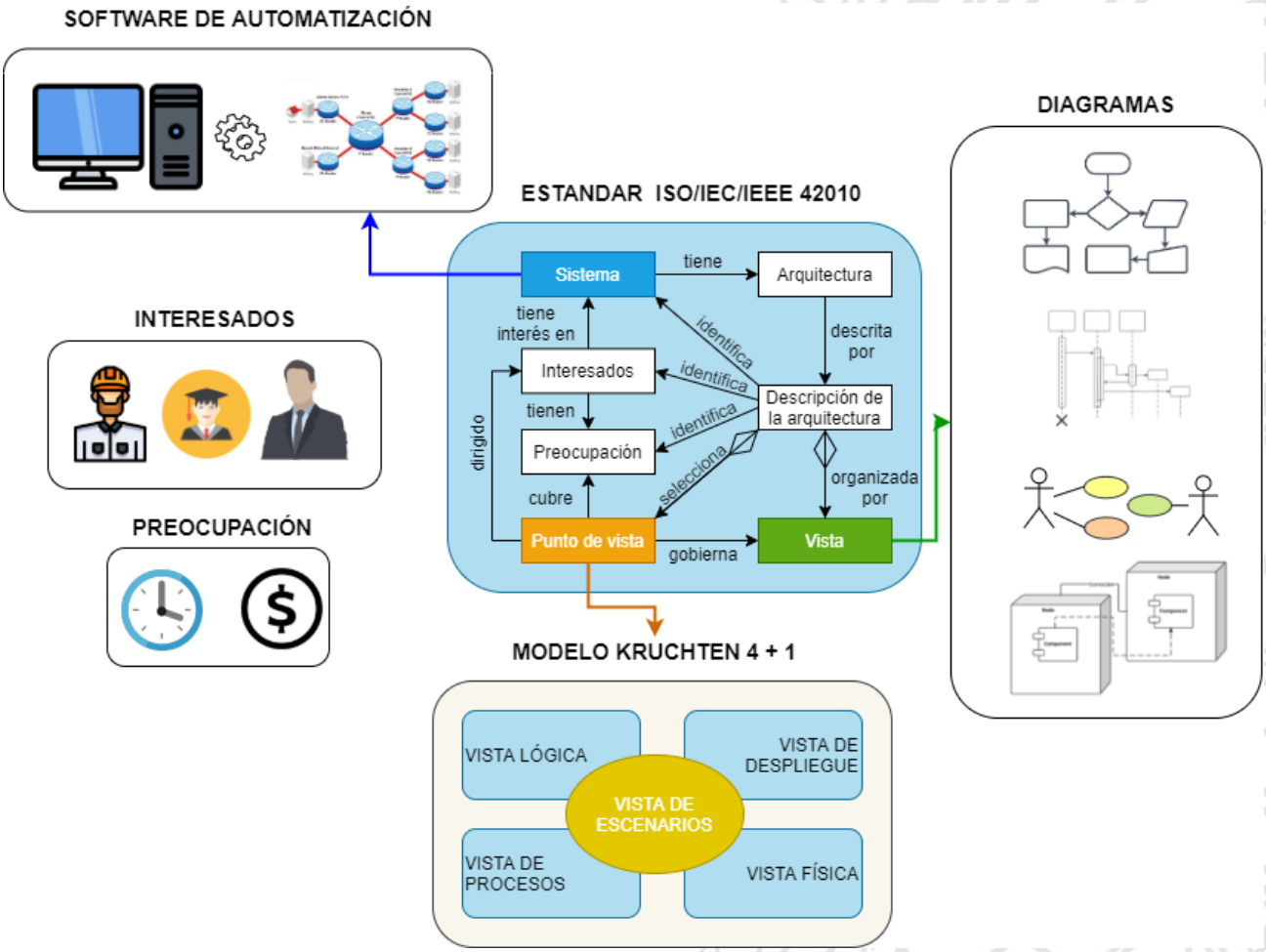


Figura 34: Descripción general del software desarrollado de acuerdo a la norma ISO/IEC/IEEE 42010.
Elaboración propia, basado en [20] y [21].

Para el caso de este proyecto:

- El **Sistema** en el que se trabajó fue el software de automatización de la configuración de los servicios CE e IPNG en la red MPLS Huawei de InterNexa en Colombia.

- Este sistema cuenta con varios **Interesados**, como por ejemplo los ingenieros del NOC, los directivos de la compañía y el estudiante en práctica que desarrolló este software de automatización.

- Cada uno de los interesados del sistema va a tener sus **Preocupaciones**, puede ser que las preocupaciones de los directivos de la compañía sea la parte económica y comercial del proyecto. Por otra parte, la preocupación de los ingenieros del NOC puede ser el tiempo que tarda la plataforma en realizar su trabajo. Y para el estudiante en práctica la preocupación se origina en el funcionamiento óptimo del software que automatiza la configuración de los servicios CE e IPNG.

- En el sistema a su vez se identifica una **Arquitectura** que está integrada por los elementos que conforman el sistema, junto con sus relaciones.

- Finalmente, la **Descripción de la Arquitectura** hace referencia a la documentación que permite identificar el sistema, las partes interesadas con sus respectivas preocupaciones, haciendo uso del **Punto de Vista** seleccionado y de la **Vista** correspondiente.

Para la representación del punto de vista se hizo uso del "*Modelo de Kruchten 4+1*", el cual proporciona una serie de vistas que permiten describir completamente la arquitectura de un desarrollo de software.

4.4.1. DESCRIPCIÓN DE LA ARQUITECTURA

Como se mencionó anteriormente, para realizar una descripción clara de la arquitectura del software desarrollado se hizo uso del estándar *ISO/IEC/IEEE 42010* y dentro de éste, se utilizó como referencia para la representación de las vistas el modelo Kruchten 4+1 (punto de vista que dará los lineamientos de construcción de las vistas).

4.4.1.1. Vista Lógica

Esta vista fue representada por un diagrama global de funcionamiento. En este diagrama (Ver Figura 35), se observa que el software de automatización desarrollado está en la capacidad de realizar conexiones SSH con equipos como Switches de enrutamiento S9300 de Huawei y Enrutadores A7750 de Nokia. Además de esto, brinda la posibilidad de enviar de manera automática comandos y recibir sus respectivas respuestas, facilitando realizar configuraciones en cada equipo.

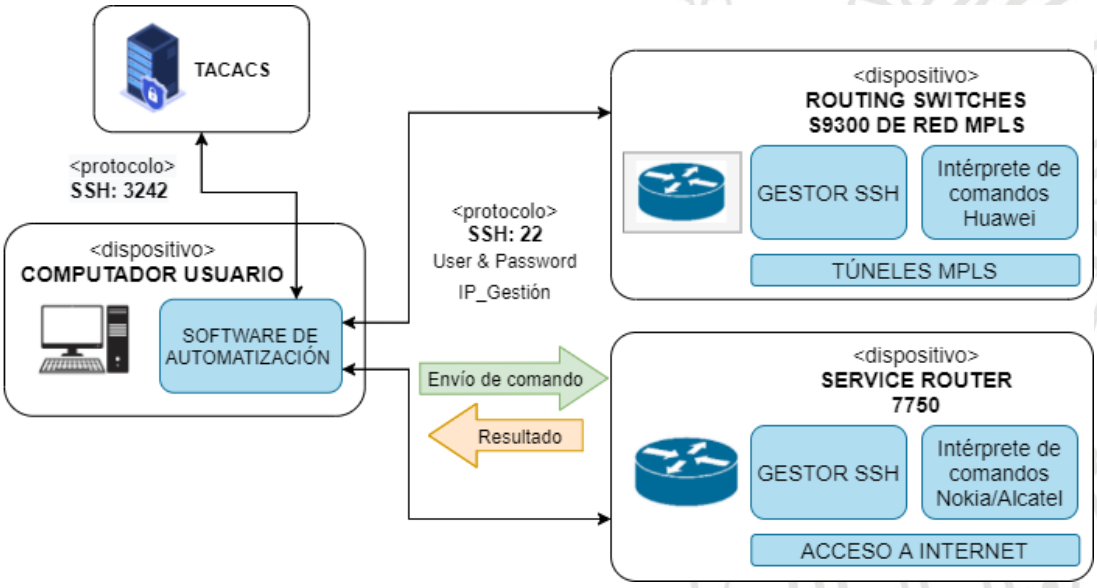


Figura 35: Vista Lógica, diagrama global.
Elaboración propia basado en [21].

4.4.1.2. Vista de Procesos

En esta vista se ve reflejada cada acción realizada por el software, donde se muestra de manera global el flujo y los procesos de trabajo del sistema. En la Figura 36 se puede observar el diagrama, donde se muestra la secuencia de procesos resumida del trabajo de cada una de las funciones que hacen parte del software.

Aquí es posible observar la distribución que tiene el sistema software. Inicialmente se cuenta con una sección de **validación** de parámetros ingresados, que se va a encargar de realizar un primer procesado de los datos y va a garantizar que estos cuenten con el formato establecido. Después de esto se tiene una sección de

conexión SSH. Es aquí donde se va a alojar la dinámica principal del software, la generación y envío de comandos de manera automática y la recepción de su respectiva respuesta por parte del equipo que está siendo configurado.

El software está constituido por una serie de funciones que van a ir ejecutando tareas. Cada función es de suma importancia y se presenta a continuación:

- **Ejecucion_Configuracion_Servicio():** Esta es la función principal, la cual va a ser llamada una vez que el usuario haya ingresado todos los datos en la GUI y presione el botón "Canalizar". En esta función inicialmente se validan los campos de la GUI y posteriormente se definen las condiciones de configuración de cada servicio, teniendo en cuenta los datos ingresados por el usuario. Cabe resaltar que dentro de esta función se va a llamar al resto de las funciones.

- **Base_Datos(equipo):** Esta función accede a la base de datos haciendo uso de la librería MySQL de Python. Tiene como parámetro de entrada el nombre del equipo y retorna la IP de gestión y la IP loopback respectiva.

- **Conexion_Servidor(ip_gestion, ssh_usuario, ssh_clave):** Esta función hace uso de la librería Paramiko estableciendo la conexión SSH con el servidor y generando el canal de comunicación. Tiene como parámetros de entrada la IP de gestión del equipo al cual se va a conectar, el nombre del usuario y su respectiva contraseña. Finalmente retorna la conexión y el canal de comunicación.

- **Generar_Comandos(datos):** Esta función se encarga de estructurar cada comando de acuerdo con los datos ingresados por el usuario. El parámetro 'datos' es un arreglo que contiene los siguientes valores:

    - Tipo de servicio (CE o IPNG)
    - Tipo de conexión (VLL o VSI)
    - Ancho de banda
    - VLAN
    - ID
    - Cliente

- Nombre del equipo
- Tipo de puerto (Troncal, Acceso, QinQ o Hibrido)
- Interfaz de enlace (Ethernet, Gigabitethernet, Xgigabitethernet, 100gigabitethernet o eth-truunk)
- Puerto de entrega
- Estado del puerto de entrega (Nuevo o existente)

- **Enviar_Comando(channel, comando):** Esta función hace uso de la librería Paramiko y procede a enviar uno a uno los comandos estructurados en el servicio a través del canal de comunicación. Tiene como parámetros el canal de comunicación y el comando que se quiere enviar al servidor remoto. Retorna el resultado de la ejecución del comando enviado.

- **Buscador_Salida(valor, resultado):** Esta función busca valores específicos en el resultado después de ejecutar un comando. Tiene como parámetros de entrada el valor a buscar y el lugar donde buscarlo. Retorna la línea donde se encuentra el valor buscado.

- **Prueba_Servicio(canal, id_servicio):** Esta función se encarga de realizar la prueba con la que se garantiza que el servicio quedó establecido. Tiene como parámetros de entrada un arreglo con los canales de los equipos que fueron configurados y el ID correspondiente al servicio.

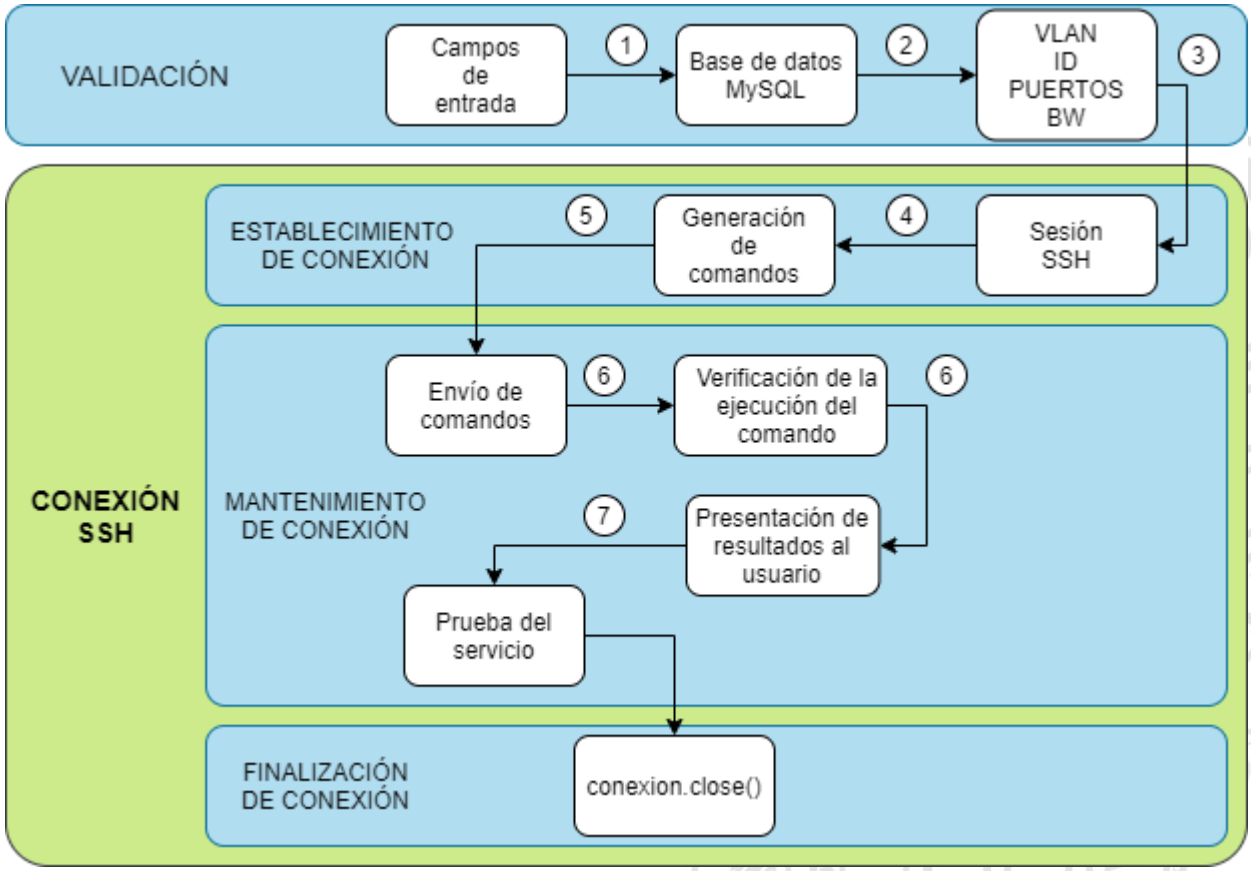


Figura 36: Vista de Procesos.
Elaboración propia basado en [21].

Donde los números hacen referencia a datos que son el resultado de la ejecución de cada bloque:

1. Parámetros de configuración, parámetros de autenticación y equipo a gestionar.
2. IP de gestión e IP loopback.
3. Confirmación de que la VLAN, el ID, el BW y los puertos ingresados están dentro del rango establecido.
4. Conexión y canal de comunicación.
5. Comandos necesarios para la configuración.
6. Resultado después de ejecutar cada comando.
7. Confirmación de que la configuración quedó lista.

4.4.1.3. Vista de Desarrollo

En esta vista se observa cómo está dividido el sistema software y es posible detallar los diferentes componentes o módulos y las dependencias que hay entre éstos (Ver Figura 37).

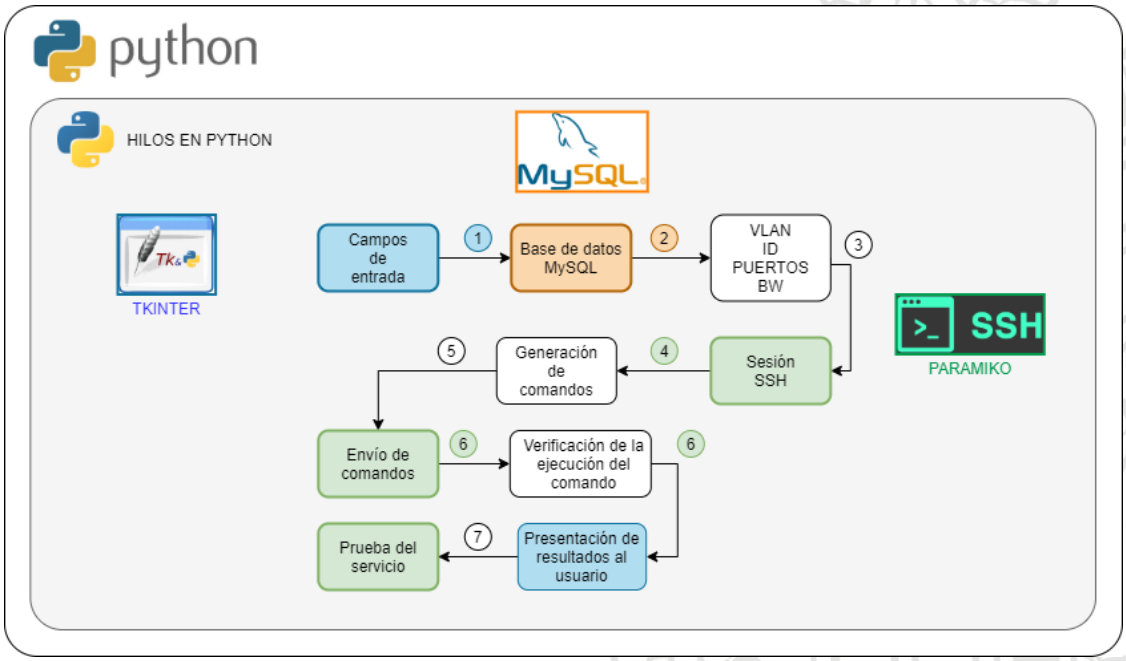


Figura 37: Vista de Desarrollo.
Elaboración propia basado en [21].

En la Figura 38 se observa el diagrama de paquetes, el cual va a complementar la vista de desarrollo, ya que proporciona información de los módulos o librerías presentes en el software y la relación entre ellos.

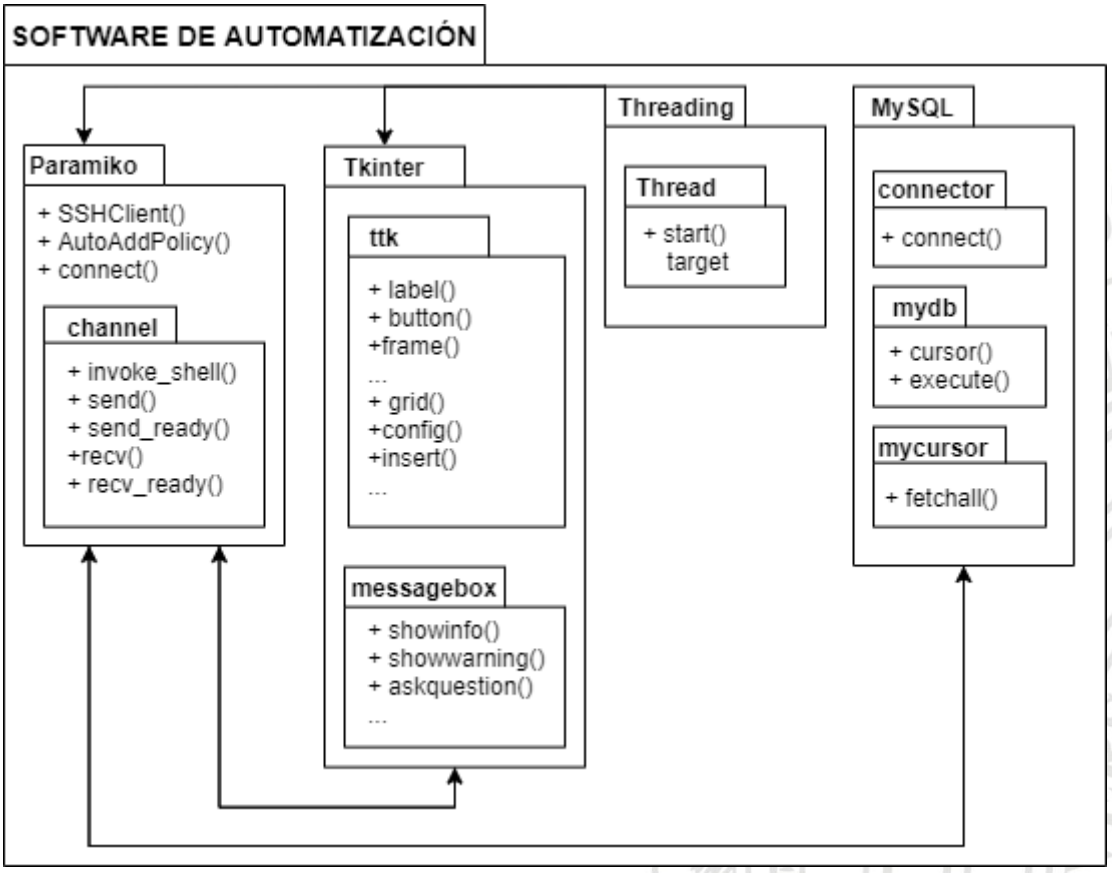


Figura 38: Diagrama de paquetes.

4.4.1.4. Vista Física

En esta vista se presentan todos los componentes físicos y las conexiones del sistema. Este diagrama (Ver Figura 39) permite tener una noción completa ya que proporciona una vista global, donde se muestra la relación entre el software desarrollado y las demás partes involucradas. Además, se detallan los protocolos de conexión de las componentes físicas.

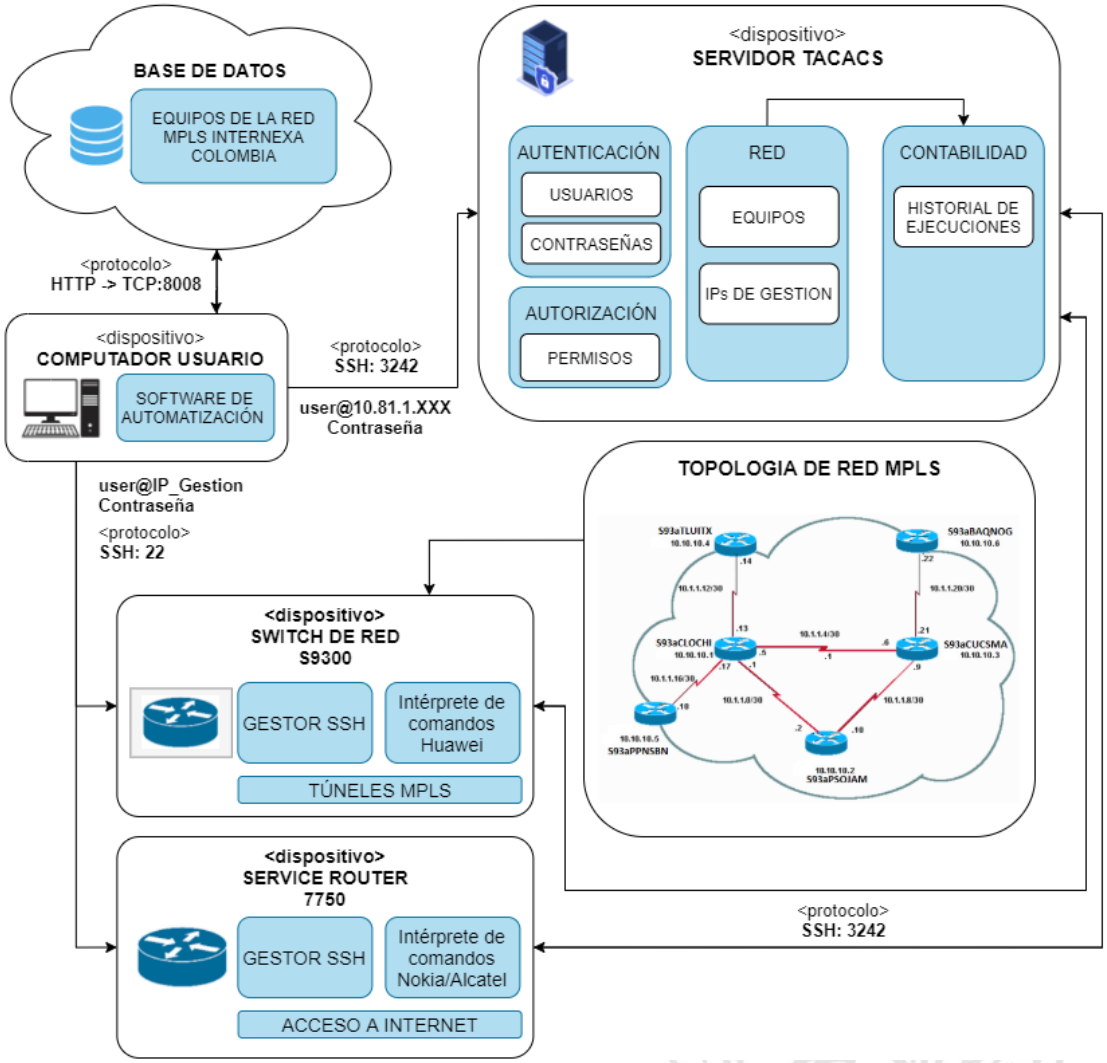


Figura 39: Vista Física.
Elaboración propia basado en [21].

4.4.1.5. Vista de Escenarios

Finalmente, esta vista representa la funcionalidad que el sistema proporcionará a los ingenieros del NOC. Haciendo uso de un diagrama de casos (Ver Figura 40) fue posible mirar como esta vista relaciona las 4 vistas anteriores.

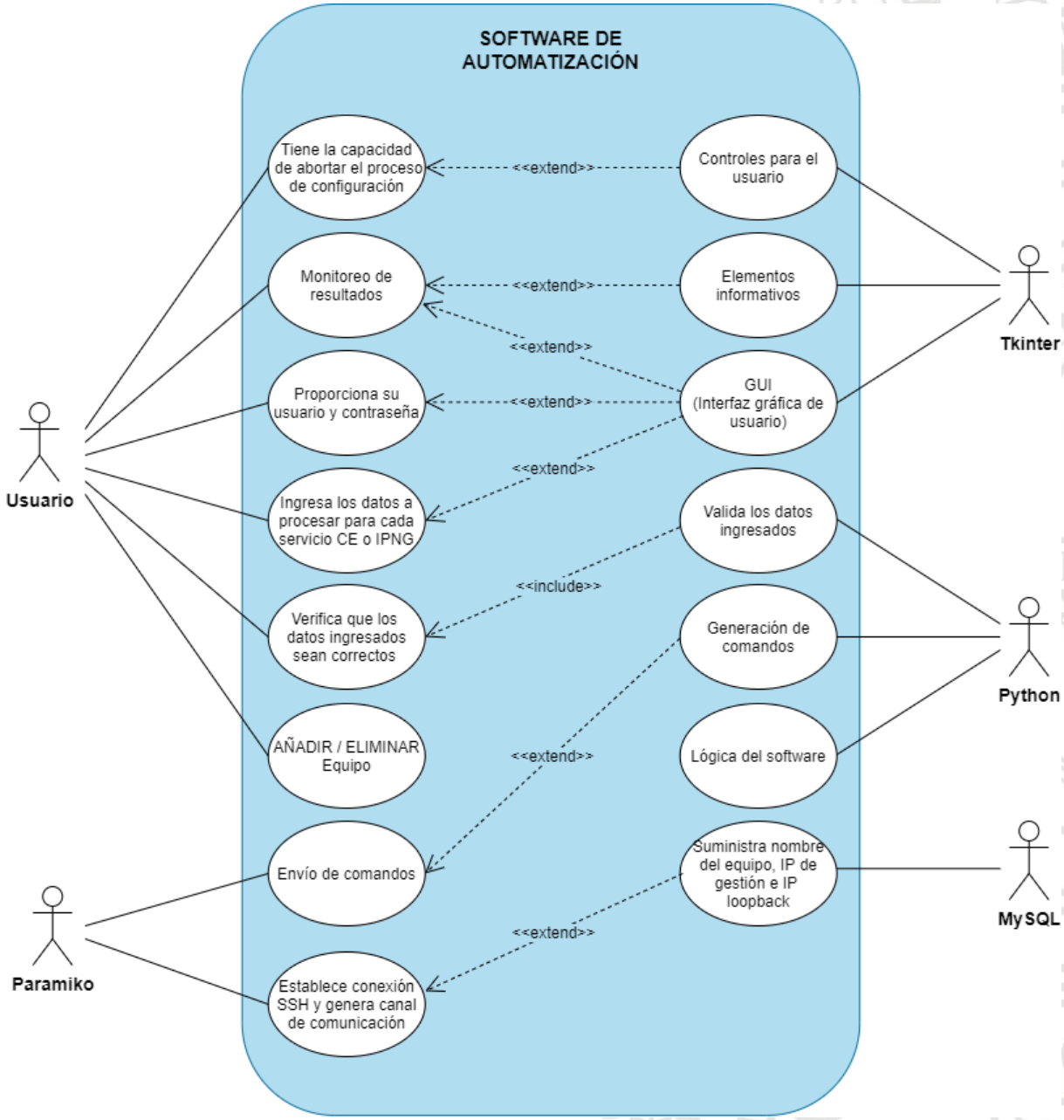


Figura 40: Vista de Escenarios.
Elaboración propia basado en [21].

5. RESULTADOS Y ANÁLISIS

El resultado de este proyecto fue la creación de un software de automatización de la configuración de los servicios CE e IPNG en la red MPLS Huawei de InterNexa en Colombia (Ver Figura 41). En principio, el funcionamiento del sistema radica en la capacidad de enviar comandos remotamente de manera automática haciendo uso del protocolo SSH, la librería Paramiko y otros módulos de Python. Esta característica, la automatización en el envío de comandos, genera confianza debido a que suprime los posibles errores por digitación existentes en configuraciones manuales.

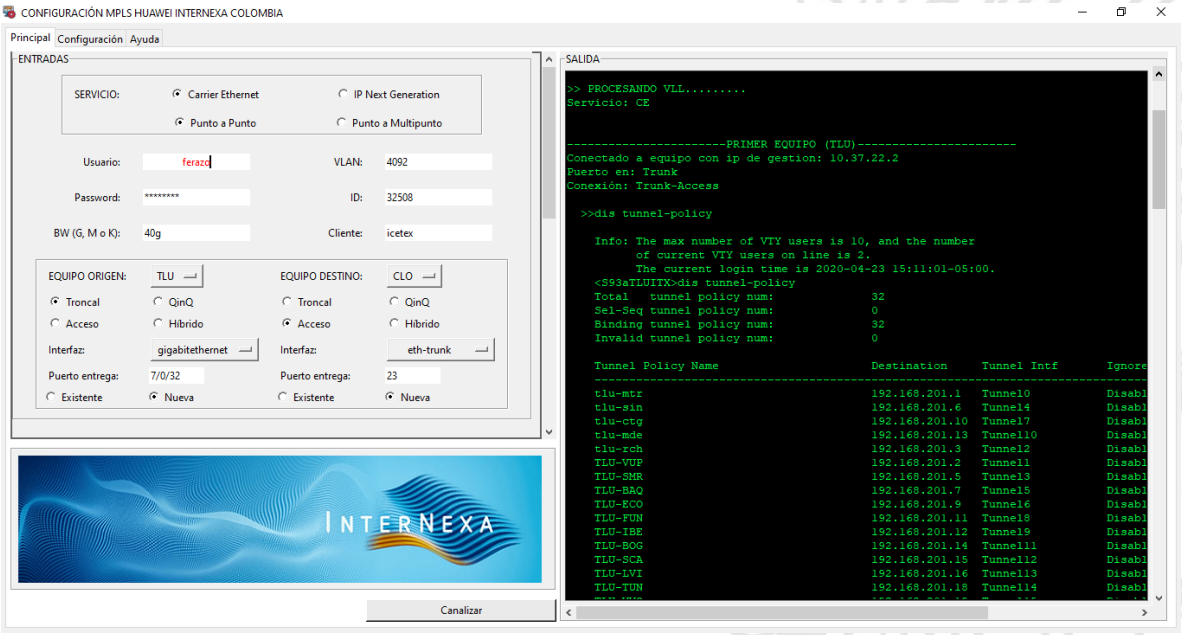


Figura 41: Pestaña principal del software de automatización.

Cabe resaltar que, para garantizar la validez del producto final, éste tuvo que pasar por un protocolo de pruebas.

5.1. PROTOCOLO DE PRUEBAS

Una vez desarrollado el software era claro que inicialmente éste no podía ser implementado en producción, debido a la existencia de posibles errores y al peligro que estos podrían generar al Backbone de la red MPLS de InterNexa en Colombia. En este contexto, el estudiante en práctica junto con los ingenieros de configuración del NOC desarrollaron un

protocolo de tres pasos. Este procedimiento garantiza que después de la ejecución del protocolo, el software estará listo para implementarse en un ambiente de producción.

- **Primer paso:** Este paso consistió en la simulación de la configuración de los equipos involucrados en el establecimiento de los servicios CE e IPNG en la red MPLS de Internexa en Colombia.
  Inicialmente el estudiante en práctica contaba con un usuario dentro del servidor TACACS con permisos únicamente de lectura, por esta razón no se podían ejecutar comandos de configuración. Debido a que no se podían ejecutar comandos de configuración, se procedió a simular el envío de cada comando, esto se hizo plasmando todos los comandos en la sección de salida de la GUI del software. Una vez simulada la configuración de un servicio, los ingenieros revisaban que los comandos quedaran bien estructurados y en el orden correspondiente. Cuando ya se tuvo completa seguridad de que el software estaba estructurando adecuadamente cada comando y ejecutándolo en el orden correspondiente, se continuó con el segundo paso.

- **Segundo paso:** Este paso consistió en la configuración de equipos apartados del Backbone de la red MPLS de InterNexa, simulando que estos equipos prestarían servicios CE e IPNG.
  Después de tener listos los comandos necesarios para configurar cada equipo y el orden adecuado ya se puede realizar pruebas en equipos reales. Pero es conveniente que estos equipos no pertenezcan al Backbone de la red principal, sino que sean dispositivos aislados. Al trabajar con dispositivos aislados se evita causar problemas a la red principal en caso de la existencia de errores en el proceso de configuración de cada equipo. Si las pruebas de configuración en estos dispositivos aislados de la red principal resultan exitosas se procede con el tercer paso.

- **Tercer paso:** Finalmente, este paso consistió en la configuración de equipos pertenecientes a la red MPLS de InterNexa Colombia por donde circula poco tráfico. En este paso se establecieron servicios CE e IPNG en un ambiente cuasi de producción, debido a que se configuraron equipos pertenecientes al Backbone de la red, pero con la condición de que fueran equipos con mínima circulación de tráfico.
  En esta instancia se configuraron equipos con poco tráfico para evitar caída en servicios en funcionamiento de la red y fue posible

observar que además de configurar correctamente cada equipo, el software no generaba daño al tráfico existente en esos equipos.

Una vez realizado el protocolo de pruebas ya se tiene plena seguridad de que el software funciona adecuadamente en su campo de acción.

Además del correcto funcionamiento, en este proyecto es de suma importancia el tiempo que toma el software de automatización en realizar la ejecución de una configuración.

5.2. TIEMPO DE CONFIGURACIÓN

El software de automatización de la configuración de servicios CE e IPNG se considerará optimo siempre y cuando el tiempo que requiera para establecer un servicio sea menor que el tiempo que un ingeniero de configuración tarda en poner en marcha un servicio de forma manual.
Para tener claridad en la comparación de los tiempos empleados por el software y los tiempos de configuración manual se configuraron varios servicios, con diferentes variantes. Inicialmente se comparó los tiempos de establecimiento de servicios Carrier Ethernet, en donde se configuraron los equipos en distribución punto a punto (VLL) y punto a multipunto (VSI). Después se comparó los tiempos de establecimiento de servicios IP Next Generation, en donde se configuraron los equipos en distribución punto a punto (VLL) y punto a multipunto (VSI).

El tiempo de configuración de servicios CE en general va a ser más corto que el tiempo de configuración de servicios IPNG. Esto se debe a que para realizar la configuración de un servicio CE el ingeniero únicamente debe ingresar a los equipos de la red MPLS Huawei, mientras que para realizar una configuración de un servicio IPNG el ingeniero además de ingresar a los equipos de la red MPLS Huawei, también debe ingresar a los equipos MPLS Nokia.
El tiempo requerido para realizar una configuración con distribución VLL va a ser más corto que el tiempo requerido para realizar una configuración con distribución VSI. Esto se debe a que en la distribución VLL solamente van a intervenir dos equipos, mientras que en la distribución VSI pueden intervenir desde dos equipos hasta seis equipos.

5.2.1. TIEMPO DE ESTABLECIMIENTO SERVICIO CARRIER ETHERNET

Para este servicio, se compararon los tiempos tanto en la distribución punto a punto, como en la distribución punto a multipunto.

- **Punto a punto (VLL):** En la Tabla 1 se observan los tiempos requeridos en cada una de las diez pruebas realizadas, tanto en la configuración manual como en la configuración automática (configuración del software de automatización). En la configuración manual se tuvieron en cuenta los tiempos empleados por los dos ingenieros de configuración del NOC.
  Cabe resaltar que en cada prueba variaron los equipos configurados, así como el tipo de puerto, la interfaz de conexión y el estado de cada puerto.

SERVICIO CE - DISTRIBUCIÓN VLL			
Config / Servicio	CONFIGURACIÓN MANUAL		CONFIGURACIÓN AUTOMÁTICA (min)
	Tiempo Ingeniero A (min)	Tiempo Ingeniero B (min)	
Servicio 1	23	23	2.849
Servicio 2	24	20	2.353
Servicio 3	22	22	2.389
Servicio 4	20	20	2.500
Servicio 5	21	23	2.878
Servicio 6	20	22	2.510
Servicio 7	24	20	2.186
Servicio 8	22	22	3.292
Servicio 9	22	22	2.705
Servicio 10	23	21	3.052
PROMEDIO	**22.1**	**21.5**	**2.671**

Tabla 1: Comparación tiempos de configuración servicio CE - VLL.

Es claro que el software emplea menos tiempo en realizar la configuración de los equipos correspondientes al servicio. Haciendo uso de la Ecuación 1 y de la Ecuación 2 es posible mirar de manera porcentual la equivalencia de los tiempos. Donde el valor "*TpromAuto*" corresponde al tiempo promedio de configuración automática, "*TpromA*" corresponde al tiempo promedio de

configuración del ingeniero A, "*TpromB*" corresponde al tiempo promedio de configuración del ingeniero B y "*TpromManual*" corresponde al tiempo promedio de configuración manual.

$$TpromManual = \frac{TpromA + TpromB}{2}$$

Ecuación 1: Tiempo promedio de la configuración manual.

$$ValorPorcentual = \frac{TpromAuto}{TpromManual} * 100$$

Ecuación 2: Valor porcentual del tiempo de ejecución.

Para este caso, los tiempos se presentan a continuación:

TpromAuto = 2.671 min
TpromA = 22.1 min
TpromB = 21.5 min
TpromManual = 21.8 min
ValorPorcentual = 12.252%

De esta manera se concluye que el software de automatización en una configuración de un servicio CE con distribución punto a punto (VLL) utiliza **12.252%** del tiempo que necesitaría un ingeniero para configurar dicho servicio.

- **Punto a multipunto (VSI):** En esta distribución (VSI) las pruebas fueron hechas inicialmente con dos equipos, después con tres equipos y así sucesivamente hasta llegar a los seis equipos. Cada prueba fue ejecutada dos veces realizando variaciones en el servicio. Los tiempos requeridos para la configuración manual y para la configuración automática se pueden ver en la Tabla 2.

SERVICIO CE - DISTRIBUCIÓN VSI				
Servicio \ Config		CONFIGURACIÓN MANUAL		CONFIGURACIÓN AUTOMÁTICA (min)
		Tiempo Ingeniero A (min)	Tiempo Ingeniero B (min)	
Servicio 1	2 Equipos	20	21	2.836
Servicio 2	3 Equipos	27	28	3.561
Servicio 3	4 Equipos	34	33	4.268
Servicio 4	5 Equipos	41	43	5.113
Servicio 5	6 Equipos	50	48	5.814
PROMEDIO		**34.4**	**34.6**	**4.318**
Servicio 6	2 Equipos	23	22	2.699
Servicio 7	3 Equipos	28	26	3.490
Servicio 8	4 Equipos	31	35	4.266
Servicio 9	5 Equipos	43	42	4.990
Servicio 10	6 Equipos	52	49	5.783
PROMEDIO		**35.4**	**34.8**	**4.246**

Tabla 2: Comparación tiempos de configuración servicio CE – VSI.

En este caso también es posible observar que el software emplea menos tiempo en realizar la configuración de los equipos correspondientes al servicio. Haciendo uso de la Ecuación 1 y de la Ecuación 2 es posible mirar de manera porcentual la equivalencia de los tiempos.

Para este caso se va a tener dos valores porcentuales, los cuales van a ser promediados para tener un valor porcentual final:

TpromAuto = 4.318 min
TpromA = 34.4 min
TpromB = 34.6 min
TpromManual = 34.5 min
ValorPorcentual = 12.516%

TpromAuto = 4.246 min
TpromA = 35.4 min
TpromB = 34.8 min
TpromManual = 35.1 min
ValorPorcentual = 12.097%

Valor Porcentual Final = 12.307%

De esta manera se concluye que el software de automatización en una configuración de un servicio CE con distribución punto a multipunto (VSI) utiliza **12.307%** del tiempo que necesitaría un ingeniero para configurar dicho servicio.

5.2.2. TIEMPO DE ESTABLECIMIENTO SERVICIO IP NEXT GENERATION

Este servicio será establecido únicamente en distribución punto a punto (VLL), por tal razón solamente se van a comparar los tiempos correspondientes a esta distribución.

- **Punto a punto (VLL):** En la Tabla 3 se observan los tiempos requeridos en cada una de las diez pruebas realizadas, tanto en la configuración manual como en la configuración automática (configuración del software de automatización). En la configuración manual se tuvieron en cuenta los tiempos empleados por los dos ingenieros de configuración del NOC. Cabe resaltar que en cada prueba variaron los equipos configurados, así como el tipo de puerto, la interfaz de conexión y el estado de cada puerto.

SERVICIO IPNG - DISTRIBUCIÓN VLL			
Config / Servicio	CONFIGURACIÓN MANUAL		CONFIGURACIÓN AUTOMÁTICA
	Tiempo Ingeniero A (min)	Tiempo Ingeniero B (min)	
Servicio 1	30	29	4.012
Servicio 2	28	28	3.399
Servicio 3	25	26	3.847
Servicio 4	26	28	2.740
Servicio 5	31	27	3.797
Servicio 6	29	27	2.915
Servicio 7	31	25	4.196
Servicio 8	31	28	3.116
Servicio 9	29	30	3.504
Servicio 10	29	30	2.979
PROMEDIO	**28.9**	**27.8**	**3.450**

Tabla 3: Comparación tiempos de configuración servicio IPNG – VLL.

Dentro del servicio IPNG también es posible observar que el software emplea menos tiempo en realizar la configuración de los equipos correspondientes al servicio.

Haciendo uso nuevamente de la Ecuación 1 y la Ecuación 2 es posible mirar de manera porcentual la equivalencia entre los tiempos de configuración manual y automática.

Para este caso, los tiempos se presentan a continuación:

TpromAuto = 3.450 min
TpromA = 28.9 min
TpromB = 27.8 min
TpromManual = 28.35 min
ValorPorcentual = 12.169%

De esta manera se concluye que el software de automatización en una configuración de un servicio IPNG con distribución punto a punto (VLL) utiliza ***12.169%*** del tiempo que necesitaría un ingeniero para configurar dicho servicio de forma manual.

Finalmente, después de analizar las anteriores tablas, se presenta un resultado global en la Tabla 4, donde es posible observar los tiempos de configuración automática de forma porcentual con respecto a los tiempos de configuración manual.

PORCENTAJES (%)		
SERVICIO CE	VLL	12.252
	VSI	12.307
SERVICIO IPNG	VLL	12.169
PROMEDIO		**12.243**

Tabla 4: Porcentajes de los tiempos de ejecución.

De los datos presentados en la Tabla 4 es posible concluir que en general, el software de automatización de la configuración de servicios CE e IPNG de la red MPLS Huawei de InterNexa en Colombia emplea aproximadamente un octavo (1/8) del tiempo necesario para realizar

configuraciones de manera manual. Por esta razón el software de automatización fue considerado óptimo y ya está listo para utilizarse en un ambiente de producción.

NOTA: Es importante tener en cuenta el tiempo que un usuario tarda en ingresar los datos al software de automatización, el tiempo de ingreso de datos puede variar entre 1 minuto y 3 minutos. Teniendo en cuenta lo anterior y tomando el tiempo de ingreso de datos igual a 3 minutos se tienen los siguientes datos:

En la Tabla 5 se presentan los mismos resultados presentados en la Tabla 1, a diferencia que al tiempo de configuración se le sumo el tiempo de ingreso de datos al software de automatización.

SERVICIO CE - DISTRIBUCIÓN VLL			
Config / Servicio	Tiempo Ingeniero A (min)	Tiempo Ingeniero B (min)	CONFIGURACIÓN AUTOMÁTICA + TIEMPO DE INGRESO DE DATOS (min)
PROMEDIO	**22.1**	**21.5**	**5.671**

Tabla 5: Comparación de tiempos CE-VLL más tiempo de ingreso de datos.

En la Tabla 6 se presentan los mismos resultados presentados en la Tabla 2, a diferencia que al tiempo de configuración se le sumo el tiempo de ingreso de datos al software de automatización.

SERVICIO CE - DISTRIBUCIÓN VSI			
Config / Servicio	Tiempo Ingeniero A (min)	Tiempo Ingeniero B (min)	CONFIGURACIÓN AUTOMÁTICA + TIEMPO DE INGRESO DE DATOS (min)
PROMEDIO A	**34.4**	**34.6**	**7.318**
PROMEDIO B	**35.4**	**34.8**	**7.246**

Tabla 6: Comparación de tiempos CE-VSI más tiempo de ingreso de datos.

En la Tabla 7 se presentan los mismos resultados presentados en la Tabla ***3***, a diferencia que al tiempo de configuración se le sumo el tiempo de ingreso de datos al software de automatización.

SERVICIO IPNG - DISTRIBUCIÓN VLL			
Config / Servicio	Tiempo Ingeniero A (min)	Tiempo Ingeniero B (min)	CONFIGURACIÓN AUTOMÁTICA + TIEMPO DE INGRESO DE DATOS (min)
PROMEDIO	**28.9**	**27.8**	**6.45**

Tabla 7: Comparación de tiempos IPNG-VLL más tiempo de ingreso de datos.

Finalmente, después de analizar las anteriores tablas, se presenta un resultado global en la Tabla 8, donde es posible observar los tiempos de configuración automática de forma porcentual con respecto a los tiempos de configuración manual teniendo en cuenta el tiempo de ingreso de datos al software de automatización.

PORCENTAJES (%)		
SERVICIO CE	VLL	26.013
	VSI	20.927
SERVICIO IPNG	VLL	22.751
PROMEDIO		**23.230**

Tabla 8: Porcentajes de los tiempos de ejecución teniendo en cuenta el tiempo de ingreso de datos.

De los datos presentados en la Tabla 8 es posible concluir que en general, el software de automatización de la configuración de servicios CE e IPNG de la red MPLS Huawei de InterNexa en Colombia emplea aproximadamente un octavo (1/4) del tiempo necesario para realizar configuraciones de manera manual. Por esta razón el software de automatización fue considerado óptimo y ya está listo para utilizarse en un ambiente de producción.

Con los datos anteriores es posible afirmar que el software tiene una eficacia del **76.77%.**

6. CONCLUSIONES

- El desarrollo de este proyecto me permitió conocer el mundo de las telecomunicaciones a nivel industrial. Se conoció la manera en la cual un país se interconecta a través de servicios CE e IPNG, haciendo uso de la tecnología MPLS. Además, fue posible aprender el funcionamiento de los equipos presentes en la topología de red de InterNexa.

- Al realizar un trabajo en equipo y tener comunicación asertiva con los ingenieros de configuración y la parte comercial de la compañía, fue posible identificar la problemática presente y de esta manera se logró dar una solución eficaz a la misma.

- En la implementación en Python se logró emplear buenas prácticas de programación haciendo uso de normas y estándares definidos por entidades como la IEEE, IETF, ISO, entre otras. Lo cual brinda soporte a la hora de realizar ajustes a la plataforma.

- Se logró desarrollar una plataforma de automatización capaz de realizar tareas de configuración de dispositivos involucrados en los servicios CE e IPNG en la topología de red MPLS de InterNexa en Colombia.

- El software de automatización de la configuración de servicios CE e IPNG de la red MPLS Huawei de InterNexa en Colombia emplea aproximadamente un octavo (1/4) del tiempo necesario para realizar configuraciones de manera manual. Por esta razón el software fue considerado óptimo.

- La reducción en los tiempos empleados en la configuración de los dispositivos presentes en un servicio permitirá que la compañía tenga capacidad para atender aún más servicios en periodos cortos de tiempo, lo cual se verá reflejado en mayores ingresos.

REFERENCIAS BIBLIOGRÁFICAS

- [1] M. Rouse. (2019, Diciembre). Multiprotocol Label Switching. [Online]. Disponible: https://searchnetworking.techtarget.com/definition/Multiprotocol-Label-Switching-MPLS

- [2] J. Barberá. (2007, Noviembre). MPLS: Una Arquitectura de Backbone para la Internet del Siglo XXI. [Online]. Disponible: https://www.rediris.es/difusion/publicaciones/boletin/53/enfoque1.html

- [3] J. López. (2019, Junio 09). Implementación de una red MPLS. [Online]. Disponible: http://openaccess.uoc.edu/webapps/o2/bitstream/10609/95887/7/pceballosbTFG0619memoria.pdf

- [4] J. Montoya, A. Amaya. (comunicación personal), 2020.

- [5] Huawei Technologies. (2015, Agosto 03). Huawei S9300 Switch Product Brochures. [Online]. Disponible: https://www.huawei.com/ucmf/groups/public/documents/attachments/hw_093969.pdf

- [6] J. Hawkins. (2017, Junio 22). Carrier Ethernet 101: velocidades, estándares y servicios. [Online]. Disponible: https://www.ciena.com.mx/insights/articles/Carrier-Ethernet-Speeds-Standards-and-Services_es_LA.html

- [7] F. Romero. (2013, Enero). Redes de próxima generación. [Online]. Disponible: https://www.researchgate.net/publication/259312180_Redes_de_proxima_generacion

- [8] L. Pérez. (2013). Estudio comparativo de los sistemas de gestión y monitoreo basados en los requerimientos generales de la red de un campus universitario. [Online]. Disponible: http://repositorio.puce.edu.ec/bitstream/handle/22000/6031/T-PUCE-6285.pdf?sequence=1&isAllowed=y

- [9] R. Belloso. (2010). Planificación y Gestión de Red. [Online]. Disponible: https://www.urbe.edu/info-consultas/web-profesor/12697883/archivos/planificacion-gestion-red/Unidad-I.pdf

- [10] A. Amarillo. (2013). Prototipo del protocolo NETCONF para la Gestión Integrada de elementos de red. [Online]. Disponible: https://repository.javeriana.edu.co/bitstream/handle/10554/15397/AmarilloRojasAdrianaFernanda2013.pdf?sequence=1&isAllowed=y

- [11] Red Hat. (2019, Junio 21). Ansible Tower User Guide. [Online]. Disponible: https://docs.ansible.com/ansible-tower/3.4.2/pdf/AnsibleTowerUserGuide.pdf

- [12] G. Rossum. (2009, Septiembre). El tutorial de Python. [Online]. Disponible: http://docs.python.org.ar/tutorial/pdfs/TutorialPython2.pdf

- [13] E. Bahit. (2013). Conexiones SSH y SFTP desde Python con paramiko. [Online]. Disponible: http://46.101.4.154/Art%C3%ADculos%20t%C3%A9cnicos/Python/Paramiko%20-%20Conexiones%20SSH%20y%20SFTP.pdf

- [14] Red Hat, Inc. (2005). Manual de referencia. Protocolo SSH. [Online]. Disponible: https://web.mit.edu/rhel-doc/4/RH-DOCS/rhel-rg-es-4/ch-ssh.html

- [15] A. Suárez. (2015, Diciembre 25). Tkinter: interfaces graficas en Python. [Online]. Disponible: https://python-para-impacientes.blogspot.com/2015/12/tkinter-interfaces-graficas-en-python-i.html

- [16] H. Fátima. (2017, Noviembre 13). Los 6 mejores marcos de Python GUI para desarrolladores. [Online]. Disponible: https://blog.resellerclub.com/the-6-best-python-gui-frameworks-for-developers/

- [17] T. Birchard. (2020, Enero 03). SSH y SCP en Python con Paramiko. [Online]. Disponible: https://hackersandslackers.com/automate-ssh-scp-python-paramiko/

- [18] F. Lundh. (2003). Python Standard Library: Threads and Processes. [Online]. Disponible: https://effbot.org/media/downloads/librarybook-threads-and-processes.pdf

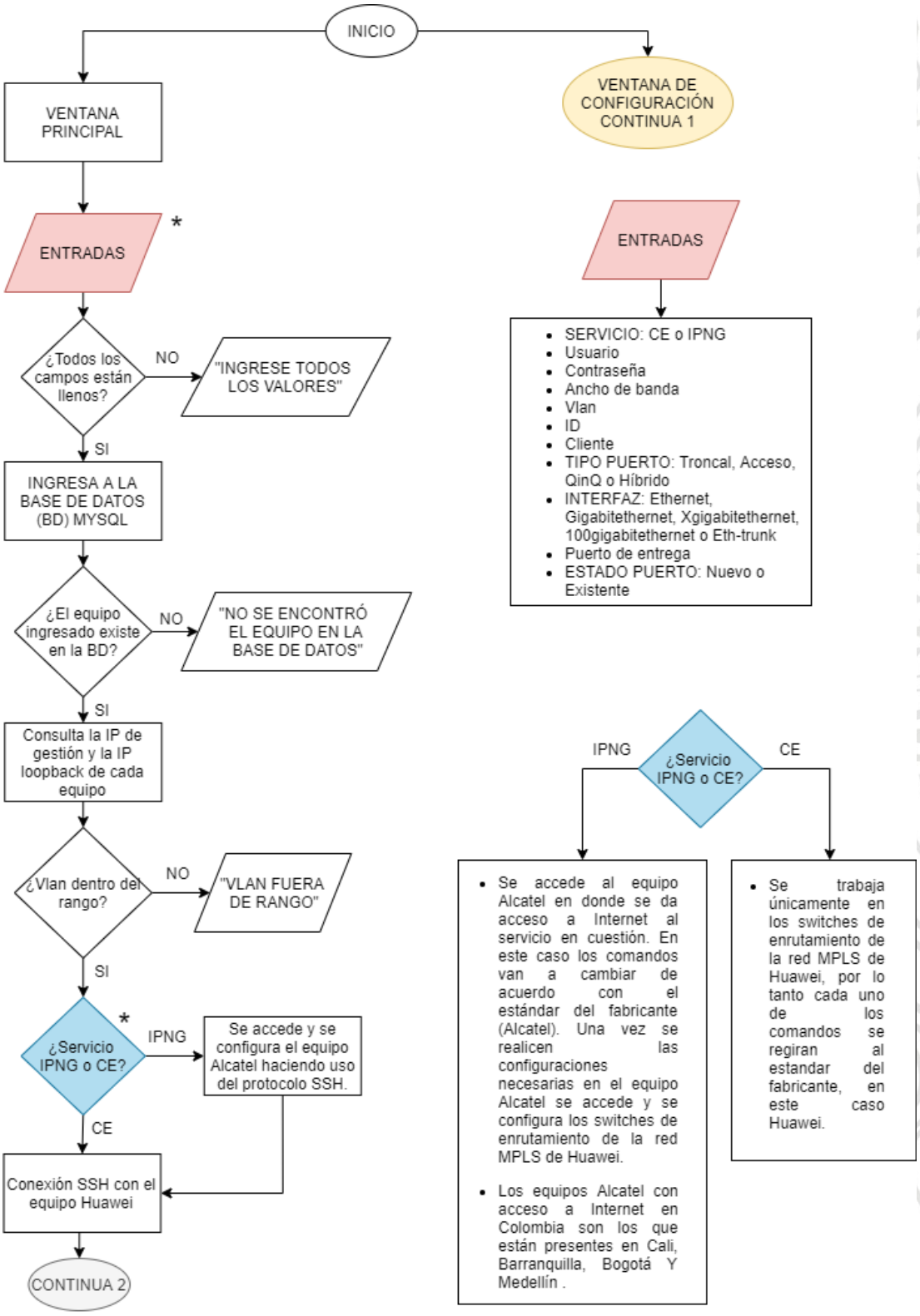
- [19] A. Barroso. (2005, Septiembre). Instalación de una plataforma para prueba de Servicios Web XML sobre la J2ME. [Online]. Disponible: http://bibing.us.es/proyectos/abreproy/11096/fichero/Memoria%252F04+Cap%C3%ADtulo+4+Base+de+Datos+mySQL.pdf+

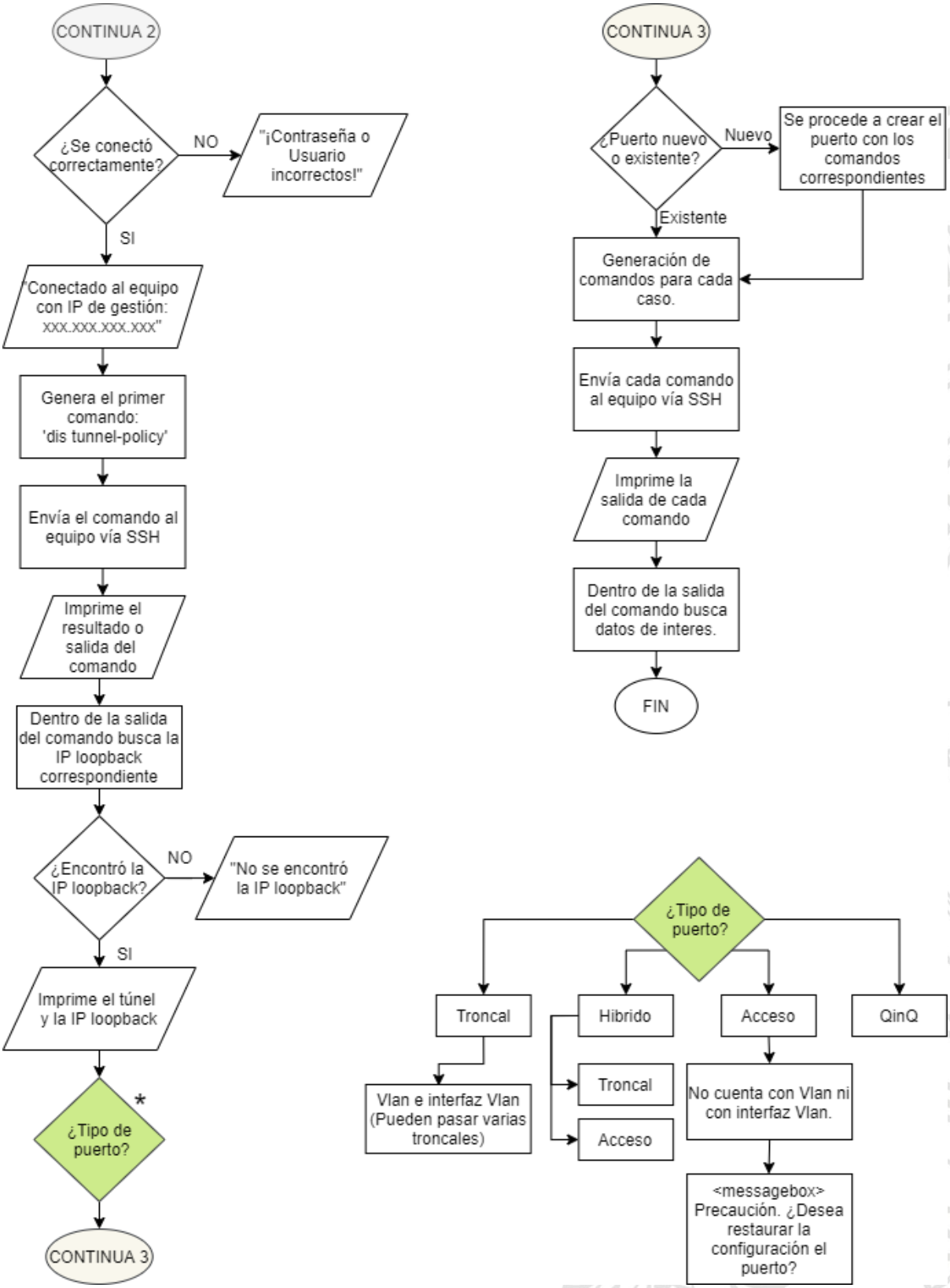
- [20] M. Medina. (2018, Junio). Diseño de una Arquitectura de Sistemas de Información para la Administración del Alineamiento a Estándares Académicos. [Online]. Disponible: https://repositoriotec.tec.ac.cr/handle/2238/9875

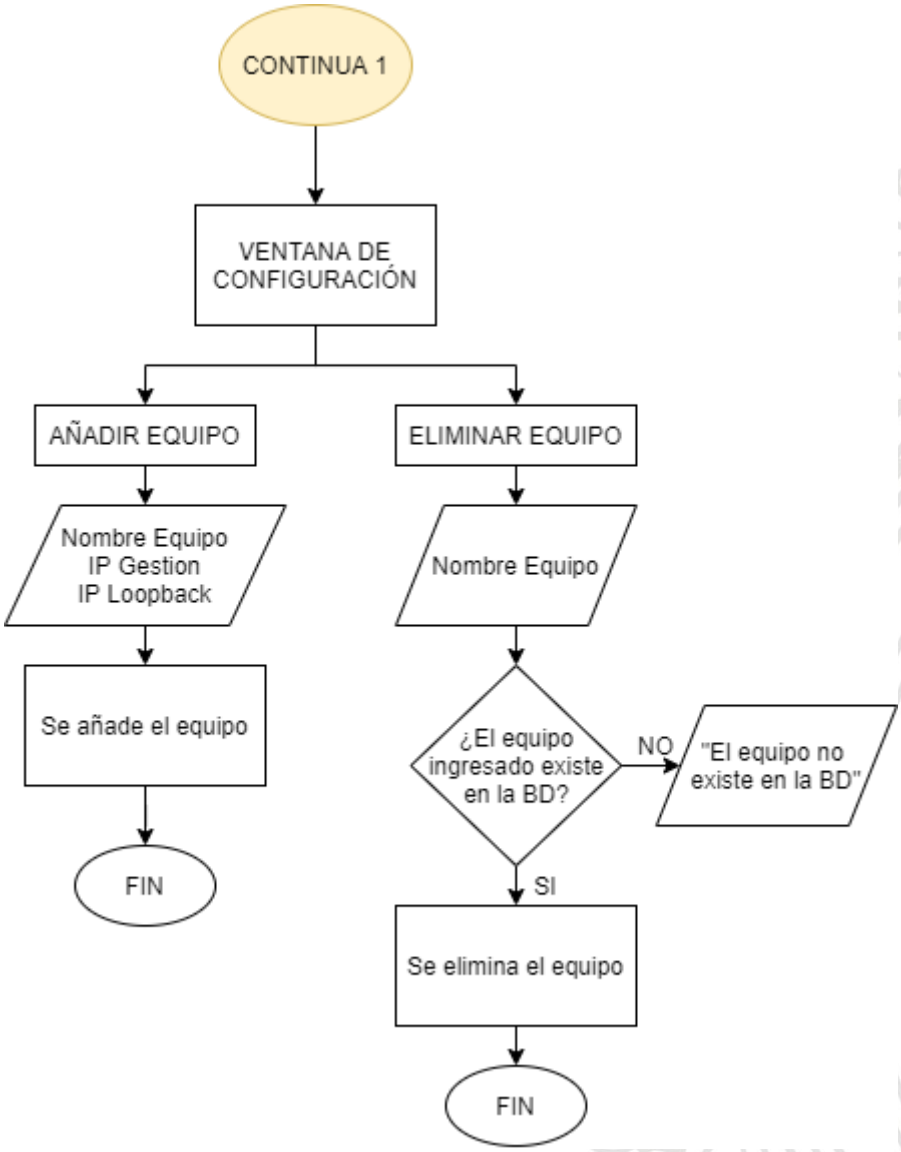
- [21] R. Puerta. (2015, Junio). Análisis, diseño e implementación de una aplicación web mediante java para la gestión de favoritos de Spotify. [Online]. Disponible: http://oa.upm.es/44381/2/TFM_RAUL_PUERTA_SANCHEZ_MANUEL_GERARDO_ACEVEDO_COELHO.pdf

ANEXOS

Anexo 1. DIAGRAMA DE FLUJO DEL SOFTWARE




CONTINUA 2
¿Se conectó correctamente?
NO
"¡Contraseña o Usuario incorrectos!"
SI
"Conectado al equipo con IP de gestión: xxx.xxx.xxx.xxx"
Genera el primer comando: 'dis tunnel-policy'
Envía el comando al equipo vía SSH
Imprime el resultado o salida del comando
Dentro de la salida del comando busca la IP loopback correspondiente
¿Encontró la IP loopback?
NO
"No se encontró la IP loopback"
SI
Imprime el túnel y la IP loopback
¿Tipo de puerto?
*
CONTINUA 3
CONTINUA 3
¿Puerto nuevo o existente?
Nuevo
Se procede a crear el puerto con los comandos correspondientes
Existente
Generación de comandos para cada caso.
Envía cada comando al equipo vía SSH
Imprime la salida de cada comando
Dentro de la salida del comando busca datos de interes.
FIN
¿Tipo de puerto?
Troncal
Hibrido
Acceso
QinQ
Vlan e interfaz Vlan (Pueden pasar varias troncales)
Troncal
Acceso
No cuenta con Vlan ni con interfaz Vlan.
<messagebox> Precaución. ¿Desea restaurar la configuración el puerto?


CONTINUA 1
VENTANA DE CONFIGURACIÓN
AÑADIR EQUIPO
ELIMINAR EQUIPO
Nombre Equipo
IP Gestion
IP Loopback
Nombre Equipo
Se añade el equipo
¿El equipo ingresado existe en la BD?
NO
"El equipo no existe en la BD"
FIN
SI
Se elimina el equipo
FIN