
# UNIVERSIDAD TÉCNICA DEL NORTE

## FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

## CARRERA DE TELECOMUNICACIONES

### TRABAJO DE INTEGRACIÓN CURRICULAR

### TEMA:

"AUTOMATIZACIÓN DE UNA RED SD-WAN MEDIANTE EL USO DE APIs PARA UNA ADMINISTRACIÓN EFICIENTE DE LA RED"

**Trabajo de titulación previo a la obtención del título de Ingeniero en Telecomunicaciones**

**Línea de investigación:** Desarrollo, aplicación de software y cyber security (seguridad cibernética).

**AUTOR:**

BRAYAN DANIEL BENAVIDES JACOME

**DIRECTOR:**

MSc. CARLOS ALBERTO VÁSQUEZ AYALA

**Ibarra – Ecuador 2026**

**UNIVERSIDAD TÉCNICA DEL NORTE**
**BIBLIOTECA UNIVERSITARIA**

**AUTORIZACIÓN DE USO Y PUBLICACIÓN**
**A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE**

## 1. IDENTIFICACIÓN DE LA OBRA

En cumplimiento del Art. 144 de la Ley de Educación Superior, hago la entrega del presente trabajo a la Universidad Técnica del Norte para que sea publicado en el Repositorio Digital Institucional, para lo cual pongo a disposición la siguiente información:

| **DATOS DE CONTACTO** | | | |
|---|---|---|---|
| **CÉDULA DE IDENTIDAD:** | 0401226816 | | |
| **APELLIDOS Y NOMBRES:** | BENAVIDES JACOME BRAYAN DANIEL | | |
| **DIRECCIÓN:** | Tulcán, Calle Bolivar y General Plaza | | |
| **EMAIL:** | bdbenavidesj@utn.edu.ec / brayanb28@gmail.com | | |
| **TELÉFONO FIJO:** | NO REGISTRA | **TELÉFONO MÓVIL:** | 0990259798 |

| **DATOS DE LA OBRA** | |
|---|---|
| **TÍTULO:** | AUTOMATIZACIÓN DE UNA RED SD-WAN MEDIANTE EL USO DE APIs PARA UNA ADMINISTRACIÓN EFICIENTE DE LA RED |
| **AUTOR (ES):** | BENAVIDES JACOME BRAYAN DANIEL |
| **FECHA: DD/MM/AAAA** | 05/02/2026 |
| SOLO PARA TRABAJOS DE GRADO | |
| **PROGRAMA:** | ■ **PREGRADO** ☐ **POSGRADO** |
| **TITULO POR EL QUE OPTA:** | INGENIERO EN TELECOMUNICACIONES |
| **DIRECTOR:** | MSC. CARLOS ALBERTO VÁSQUEZ AYALA |
| **ASESOR:** | MSC. FABIÁN GEOVANNY CUZME RODRÍGUEZ |

## 2. CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 05 días del mes de febrero de 2026

**EL AUTOR:**


BENAVIDES JACOME BRAYAN DANIEL

# CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Ibarra, 05 de febrero de 2026

ING. CARLOS ALBERTO VÁSQUEZ AYALA, MSC

DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

Firmado electrónicamente por:
CARLOS ALBERTO VASQUEZ AYALA
Validar únicamente con FirmaEC

*(f)* ……

*MSC. CARLOS ALBERTO VÁSQUEZ AYALA*

*C.C.: 1002424982*

## DEDICATORIA

*Con mucho cariño y gratitud dedico este trabajo:*

*A mi familia, quienes siempre han sido mi guía, mi mayor inspiración y el ese pilar inquebrantable, quienes, con su amor incondicional, paciencia, sacrificio y apoyo constante han sido mi motor en los momentos más desafiantes.*

*A mi abuelita Anita María, quien con su cuidado y amor forjo lo que hoy en dia estoy cosechando, siendo ese guía espiritual y de vida que ha forjado el carácter necesario para afrontar cada uno de los retos que se han presentado en mi vida.*

*A mi madre, por enseñarme el significado del esfuerzo y la constancia, valores característicos en ella, cuyas palabras de apoyo y llenas de amor me permitieron forjar y superar este camino.*

*A mis hermanas, por estar a mi lado, animándome y motivándome a alcanzar mis metas, cuyo ejemplo de perseverancia y constancia fue valioso en momentos que pensé desfallecer.*

*A mis amigos Raúl, Adonis, Roberth, "esiosotros" entre otros, cómplices de risas y desvelos, que me recordaron que el camino, aunque arduo, se recorre mejor acompañado. Aquellas personas que han pasado por mi vida dejándome valiosos momentos y lecciones, y todos aquellos que creyeron en mí, incluso cuando yo dudaba, dedico este trabajo con gratitud y orgullo, porque cada paso dado es también suyo.*

*BENAVIDES JACOME BRAYAN DANIEL*

## AGRADECIMIENTO

*Llegar a este momento no ha sido un camino solitario, y por ello quiero expresar mi más profundo agradecimiento a quienes han sido parte de este viaje. A mis profesores, por su guía, paciencia y por compartir su conocimiento con pasión, despertando en mí la curiosidad y el deseo de aprender. A mi director de tesis Msc. Carlos Vásquez, por su orientación incansable y por ayudarme a transformar ideas en realidades. A mi asesor Msc. Fabián Cuzme por condicionarnos a ser mejores cada dia. A los amigos y compañeros de la carrera con los cuales compartí momentos de alegría, trasnochadas y aprendizaje. Finalmente, a mi familia y amigos, por su apoyo incondicional, por cada palabra de aliento en los momentos de duda y por celebrar conmigo cada pequeño logro. Este trabajo es el resultado de un esfuerzo personal y colectivo, a todos ustedes infinitas gracias.*

## RESUMEN EJECUTIVO

El presente trabajo de integración curricular aborda la automatización de una red SD-WAN para una administración y gestión eficiente mediante el uso de interfaces de programación de aplicaciones (APIs), específicamente la API REST del controlador Cisco vManage. El objetivo principal del trabajo es demostrar que la gestión programable de la red permite mejorar la eficiencia operativa, reducir la dependencia de procesos manuales y facilitar la supervisión del estado de los dispositivos y enlaces de comunicación. Para el desarrollo de la solución, se implementaron scripts en Python que permiten establecer una sesión autenticada con el controlador, consumir distintos endpoints para la obtención de información relevante y procesar los datos obtenidos de forma automatizada. Como parte del sistema propuesto, se incorporaron scripts en lenguaje Python de monitoreo, auditoría e inventario automatizado, así como un mecanismo de notificaciones que informa al administrador de la red sobre eventos relevantes de manera oportuna.

Las pruebas realizadas en un entorno de laboratorio dentro de GNS3 permitieron validar el correcto funcionamiento de la solución, evidenciando que el sistema es capaz de autenticar, consultar información del estado de la red, generar reportes con información específica y generar alertas sin intervención manual. Los resultados obtenidos demuestran que el uso de APIs REST constituye una alternativa viable para la automatización de redes SD-WAN, proporcionando una base sólida para futuras ampliaciones orientadas a entornos de producción y escenarios de mayor complejidad debido al sin fin de posibilidades que ofrece este tipo de soluciones.



**Palabras clave**: SD-WAN, automatización de redes, APIs REST, vManage, monitoreo de red.

## ABSTRACT

This curricular integration project addresses the automation of an SD-WAN network for efficient administration and management using Application Programming Interfaces (APIs), specifically the Cisco vManage controller's REST API. The main objective of this work is to demonstrate that programmable network management improves operational efficiency, reduces reliance on manual processes, and facilitates the monitoring of device status and communication links. For the development of the solution, Python scripts were implemented to establish authenticated sessions with the controller, consume various endpoints to retrieve relevant information, and process the data collected automatically. As part of the proposed system, Python scripts for automated monitoring, auditing, and inventory were incorporated, along with a notification mechanism that informs the network administrator of relevant events in a timely manner.

Tests conducted in a GNS3 laboratory environment validated the proper functioning of the solution, showing that the system can authenticate, querying network status information, generating reports with specific data, and triggering alerts without manual intervention. The results demonstrate that the use of REST APIs constitutes a viable alternative for SD-WAN network automation, providing a solid foundation for future expansions into production environments and more complex scenarios, given the endless possibilities offered by these types of solutions.



**Keywords:** SD-WAN, network automation, REST APIs, vManage, network monitoring.

# ÍNDICE DE CONTENIDO

## ÍNDICE DE FIGURAS

# ÍNDICE DE TABLAS

## LISTA DE SIGLAS

**API:** Application Programming Interface

**CLI:** Command Line Interface

**CSRF:** Cross-Site Request Forgery

**GET:** Método HTTP para la obtención de información

**HTTP:** Hypertext Transfer Protocol

**JSON:** JavaScript Object Notation

**PDF:** Portable Document Format

**POST:** Método HTTP para el envío de información

**REST:** Representational State Transfer

**SD-WAN:** Software-Defined Wide Area Network

**vEdge:** Virtual Edge Device

**vManage:** Virtual Network Management Controller

**vSmart:** Virtual Smart Controller

**vBond:** Virtual Bond Orchestrator

# CAPÍTULO I: Antecedentes

## 1.1. Tema

AUTOMATIZACIÓN DE UNA RED SD-WAN MEDIANTE EL USO DE APIs PARA UNA ADMINISTRACIÓN EFICIENTE DE LA RED.

## 1.2. Problema

El aumento de carga administrativa para mantener múltiples enlaces de comunicación, personal especializado para su administración, configuración, monitoreo, despliegue y elevados costos en equipos tradicionales que se necesitan para comunicar una matriz con un sucursal o sucursales, ha obligado con el tiempo a que las empresas busquen opciones a fin de mejorar la comunicación entre ellas (Sarmiento, 2023).

La virtualización de las redes y la necesidad de procesar grandes cantidades de datos están provocando una convergencia con el sector de las tecnologías de la información lo que está fomentando la entrada de nuevos actores (ECLAC, 2022). Por ello, la red SD-WAN se ve hoy como una alternativa para mejorar la visibilidad, control, disponibilidad, rendimiento, escalabilidad y seguridad dentro de las redes tradicionales que se disponían hasta hoy.

El aumento de tráfico dentro de las redes WAN se traduce en una mayor complejidad en la gestión de la información, así como en aumentos de latencia y una mayor dificultad en el momento del monitoreo de la red para la gestión de los datos que están viajando, SD-WAN aborda estos desafíos que están pasando los departamentos de redes mejorando la conectividad y reduciendo de manera significativa los costos operativos para la gestión y monitoreo de la red (Pérez & Gualoto, 2022) .

Sin embargo, la falta de escalabilidad, procesos manuales y tareas repetitivas propensas a errores pueden generar que la administración y gestión de la red pierda eficiencia. Por lo cual incorporar programabilidad de nuevas funcionalidades a través de APIs con el fin de obtener

información personalizada que podría agilizar la toma de decisiones sin tener que afectar el ambiente de producción que se encuentra desplegado a través de una red SD-WAN.

Una API (Interfaz de Aplicación Programable) es un software que permite a otras aplicaciones acceder a sus datos e interactuar con ellos, permite definir un conjunto de reglas que describen una aplicación y así interactuar con otra, se usa para controlar, administrar, orquestar y generar telemetría y consiste en un conjunto de reglas que describen como una aplicación puede interactuar con otra aplicación y el mecanismo que permite que esa interacción tenga efecto (G. Salazar & Marrone, 2021). Ayudan a simplificar el diseño y desarrollo de nuevas aplicaciones y servicios con lo cual se logra una flexibilidad en la integración y gestión de las ya existentes.

## 1.3. Objetivos

### *1.3.1. Objetivo General*

Desarrollar un sistema automatizado basado en APIs que optimice la administración y gestión de una red SD-WAN tradicional, a través de un entorno emulado.

### *1.3.2. Objetivos Específicos*

- Efectuar una investigación teórica con relación a las redes SD-WAN, entornos de automatización y la aplicación de APIs en su infraestructura.
- Identificar los requerimientos de una red SD-WAN y sus posibles mejoras aplicando APIs.
- Integrar APIs en un sistema SD-WAN emulado, detallando sus componentes.
- Realizar pruebas de funcionamiento en un entorno controlado de una red SD-WAN y la aplicación de APIs en aquella infraestructura.

## 1.4. Alcance

El presente proyecto tiene como finalidad el desarrollo de un sistema automatizado que mediante el uso de APIs se logre optimizar la administración y la gestión en una red SD-WAN

tradicional, esto dentro de un entorno emulado que permita controlar cada uno de los elementos que forman parte de la infraestructura de la red, para lo cual la interacción de la API con el administrador se realizara mediante herramientas como Python o mediante el uso del Bash de Linux, debido a que permiten escribir scripts que facilitan la creación de flujos de trabajo personalizados para procesar datos JSON y manejar respuestas de las API, dado que la gestión de la red está enfocada a usuarios avanzados por lo cual no cuenta con una interfaz amigable.

El proceso metodológico que se ha considerado para el presente proyecto es el de PMBOK el cual consta de cinco etapas las cuales ayudaran a cumplir con el objetivo general, considerando las siguientes:

En primera instancia se estableció la etapa de inicio, donde se efectuará una investigación teórica preliminar sobre aspectos de redes SD-WAN y el concepto de entornos de automatización de los sistemas, conceptualización de una API en relación con el contexto de redes de nueva generación y la aplicabilidad de APIs como una posible mejora en la infraestructura, en base a tesis de grado, sitios web, artículos de revistas indexados, libros y varios.

Como siguiente etapa, planificación, se realizará un análisis de los requerimientos de la red SD-WAN actual, incluyendo dispositivos, protocolos, configuraciones y flujos de trabajo, en base a la identificación de los requerimientos se procederá al desarrollo de la API y del sistema automatizado de la red SD-WAN, utilizando las herramientas y tecnologías adecuadas dentro de la emulación.

Posteriormente, en la etapa de ejecución, se procederá a integrar la API en un sistema SD-WAN emulado, detallando sus componentes y se realizara la implementación del sistema automatizado en el entorno de la red SD-WAN, adicional se procederá a generar un plan de monitoreo que permita analizar el correcto funcionamiento de la red SD-WAN integrada con

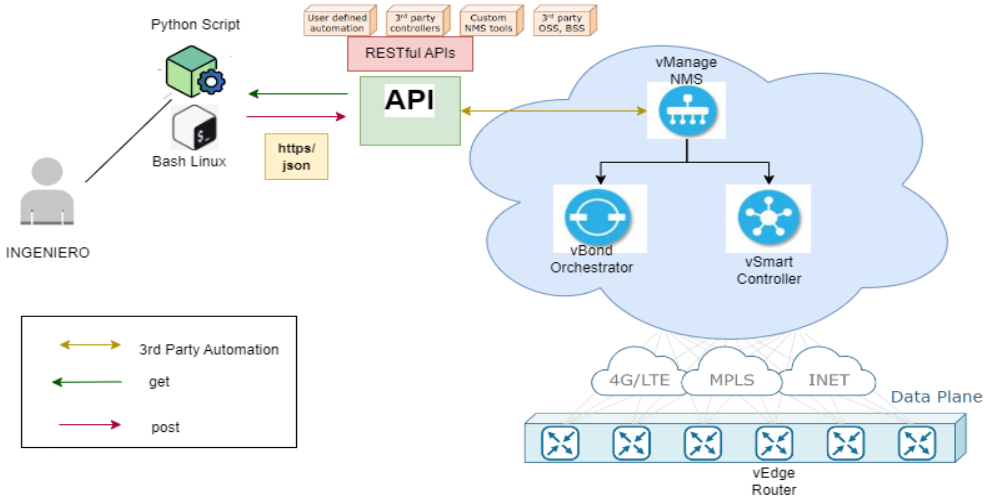
la API. Esto se realizará para cumplir con los requerimientos funcionales y no funcionales, asegurando su uso y escalabilidad.

A continuación, en la etapa de desempeño, con el fin de garantizar el correcto funcionamiento del sistema automatizado, se llevará a cabo pruebas separadas tanto para la red SD-WAN como para la API, con el propósito de verificar el desempeño individual de cada componente. Una vez concluidas estas pruebas individuales, se procederá a realizar pruebas de integración para evaluar la interacción entre los distintos componentes del sistema.

Para finalizar, en la etapa de cierre, se identificarán los resultados obtenidos con un análisis detallado de las respuestas del sistema y se evaluará la factibilidad de la solución propuesta.

**Figura 1**

*Topología de la red SD-WAN/API*




*Nota.* El gráfico representa la topología de una red SD-WAN y una API conectada al plano de gestión vManage para su automatización. Adaptado de (Cisco, 2020); NetworkAcademy.io, 2021)

## 1.5. Justificación

La automatización se ha convertido en uno de los enfoques principales de las nuevas tecnologías, para realizar tareas repetitivas de manera automática en base a programabilidad, sin necesidad o con poca intervención humana. El presente proyecto surge como una propuesta de mejorar las condiciones de administración y gestión de una red SD-WAN mediante un proceso de automatización en base a la programabilidad de la red en la capa de control, la cual se puede realizar mediante APIs y controladores que administran la red y dispositivos de hardware. Permitiendo la flexibilidad y dinamismo en la red lo que facilita la administración, nuevas implementaciones, crecimientos o la salida de nuevos servicios de manera rápida (L. Salazar, 2022).

En topologías actuales, se tienen enrutamientos ya establecidos para diferentes servicios, por lo que cualquier afectación, incidente o requerimiento como por ejemplo cambio de equipo o actualización, que implique variaciones significativas, se debe realizar en cada uno de los equipos, lo que constituye una actividad extenuante, larga, con alto costo operativo y de dinero. Lo que con una red SDN se pueden realizar dichos cambios de manera dinámica, flexible y sin costo operacional, como los realizados en (Jia et al., 2020) a través de diferentes flujos, calculados según la selección de ruta por SDN.

Al aplicar SND en una red WAN (SD-WAN) se obtiene una solución para que el transporte de comunicaciones, sensibles a las latencias e intermitencias, mejore en su modo de funcionamiento y facilita la administración de profesionales de TI porque su gestión está basada en software, y su arquitectura de funcionamiento se centraliza, reduciendo los costos operativos y de mantenimiento mensual (A. Salazar, 2022).

Según (Teldat, 2019) el crecimiento en SD-WAN será constante en los próximos años. Lejos de ser una moda pasajera, se está instituyendo como la forma actual de entender las redes corporativas, por ello se definen estrategias claras para no perder participación de mercado o

disminuir sus ganancias debido a la visibilidad y control sobre el tráfico interno es de vital importancia, por lo cual se requieren nuevos mecanismos como la implementación de APIs. En 2023, se preveía que los ingresos por infraestructura SD-WAN superen los 13 mil millones de dólares estadounidenses a nivel mundial, con una tasa de crecimiento anual compuesta (CARG) del 31,8% entre 2022 y 2027 (Stadista, 2023).

SD-WAN tiene un gran potencial para mejorar la visibilidad y el control de las operaciones de red y su enfoque centralizado y de software la hace más flexible, rentable, escalable y rápida ya que funciona como una red superpuesta, lo que significa que está construida sobre una red física subyacente existente que consta de enrutadores y conmutadores. La diferencia clave es que SD-WAN introduce dispositivos de borde en cada ubicación para facilitar el control y la gestión de la red, esta red consta de tres componentes clave: dispositivos perimetrales (Edge Devices), controlador (Controller) y orquestador (Orchestrator) (Lee, 2025).

SD-WAN Orchestrator está en la parte superior y se comunica con los controladores SD-WAN que están implementados en el centro de datos o la nube. Los Controladores SD-WAN se comunican con los Dispositivos de Borde SD-WAN que están instalados en cada sucursal. Todos se comunican con la ayuda de API, las API en dirección sur se utilizan para la comunicación entre controladores y dispositivos perimetrales. Las API en dirección norte se utilizan para la comunicación entre aplicaciones y la infraestructura de red. Además, las API East-West se utilizan para facilitar la comunicación entre las mismas entidades, como entre dos controladores (Sharma et al., 2023).

Los controladores utilizan las API para fines de comunicación y monitoreo, son esenciales para los controladores y controladores distribuidos, estas API proporcionan interoperabilidad y compatibilidad entre diferentes controladores, lo que aumenta la robustez del sistema y reduce la probabilidad de fallas comunes (Kreutz et al., 2015).

El controlador SD-WAN utiliza APIs para comunicarse con entidades externas. Estos controladores también pueden proporcionar interfaces tradicionales, sin embargo, estas interfaces tradicionales son para compatibilidad con versiones anteriores, ya que es posible que algunos proveedores de servicios o empresas no estén listos para usar APIs. Independientemente de que interfaces tradicionales proporcionen compatibilidad con versiones anteriores, el camino a seguir para la integración SD-WAN y la forma preferida de integración con entidades externas son las API (Azhar, 2021). Esta integración mediante API será uno de los aspectos importantes al adoptar un proveedor específico sobre otro y será fundamental en la adopción de SD-WAN.

Desde la perspectiva de un proveedor de servicios, la automatización significa tener API en el orquestador. Estas API ayudarán a administrar el marco de políticas comerciales de cada cliente a través del orquestador del proveedor de servicios, lo que facilita la instalación, configuración, operación y administración de la WAN de un cliente a través del panel de software (Yadav, 2021).

En la actualidad, dentro del Ecuador, se ve como cada vez más organizaciones han comenzado a usar SD-WAN, como un medio para aprovechamiento de la red y de los nuevos servicios (Cusco et al., 2022). Por ello a partir de las condiciones anteriormente planteadas y mencionadas, se establece la necesidad de analizar e investigar maneras en como automatizar una red SD-WAN mediante el uso de API a fin de mejorar el rendimiento de la red y sobre todo la programabilidad de esta, de manera que se optimice la administración y gestión de la red SD-WAN generando un salto de calidad en su infraestructura.

# CAPÍTULO II: Fundamentación Teórica

En este capítulo se abordarán conceptos e información teórica preliminar que permita comprender aspectos relevantes para el desarrollo del tema planteado, sustentados en investigaciones previas y actuales. Entre los ejes principales que se analiza se encuentra la red SD-WAN, la cual, según Cisco (2024a), es un enfoque definido por software de la gestión de la red de área amplia o WAN. De igual manera se examinan las principales ventajas de su implementación, arquitectura e infraestructura, definición de las APIs y su aporte dentro de las redes SD-WAN, así como su integración a entornos o plataformas de automatización, beneficios de automatización para una administración eficiente de red, entornos de automatización, algunos requerimientos importantes para su implementación y tendencias de la automatización de redes SD-WAN en base al uso APIs.

## 2.1. Redes SD-WAN

Una red SD-WAN, o Red de Área Amplia Definida por Software, es una tecnología que aplica principios de SDN (Software Defined Networking) a las WAN tradicionales que se basan en una red estática centrada en hardware, y la transforma en una WAN ágil y flexible con base en el software, separando de manera eficaz la gestión del tráfico de red de la infraestructura física de transporte que subyace, en pocas palabras permite gestionar y optimizar el tráfico de red de manera centralizada, a fin de dirigir el tráfico a través de la red de manera más eficiente en tiempo real (Vargas, 2020).

La tecnología SD-WAN se encuentra dentro de la tipología de un servicio de red extensa que actúa sobre la capa lógica de una red WAN, en otros términos, constituye un programa de gestión donde parte del hardware de red se ve virtualizado, de forma más específica, SD-WAN suministra una red de superposición virtual donde es posible una conectividad generada e inducida entre diversas interfaces de red (Moreno, 2021).

Entre las principales características de una SD-WAN, se destacan las siguientes:

- **Agnosticismo de la conexión**: No requiere de un tipo especifico de enlace para el funcionamiento, lo cual genera un entorno de alta disponibilidad.
- **Selección dinámica de rutas:** Elige de manera dinámica el mejor camino para la transmisión de datos en función de la calidad de la conexión local, y a veces dependiendo de la ruta entre la fuente y el destino.
- **Soporte de servicios adicionales:** Incluye elementos para la optimización y mejora en cuestiones de seguridad, a fin de garantizar los requisitos de rendimiento par aplicaciones sensibles, al mejorar y optimizar la experiencia de los usuarios finales.
- **Automatización de la red:** Permite un enrutamiento de tráfico más inteligente y seguro en la WAN, lo que deriva en que la red responda automáticamente ante fallos, sin la necesidad de intervención humana.

SD-WAN presenta varias ventajas en comparación a tecnologías tradicionales como MPLS, incluyendo un menor costo, una configuración de WAN más sencilla, fácil acceso a los servicios en la nube y uso eficiente de los recursos de la red WAN (Gordeychik et al., 2018).

Como se muestra en la Tabla 1, existen algunas diferencias entre la red tradicional WAN y una SD-WAN, respecto a integración, administración, extensión, operación y conductores, por ende, las WAN tradicionales se basan en una integración que requiere una administración manual dispositivo por dispositivo mientras que SD-WAN adopta un enfoque con gestión automatizada desde un controlador centralizado. Adicionalmente, las WAN tradicionales son difíciles de modificar, abordan los problemas solo cuando surgen y se centran en la infraestructura de la red, por el contrario, las redes SD-WAN ofrecen mayor flexibilidad en modificación, se anticipa a fallos antes de que afecten a la red y prioriza el rendimiento de aplicaciones clave y para necesidades empresariales.

**Tabla 1**

*Comparativa entre una red WAN y una red SD-WAN.*

| | **WAN Tradicional** | **WAN Definido por Software** |
|---|---|---|
| **Integración** | Centrado en el Hardware | Centrado en el Software |
| **Administración** | Manual caja a caja | Automatizada |
| **Extensión** | Cerrado | Programable a través de API REST |
| **Operaciones** | Reactivo | Predictivo |
| **Conductores** | Intención de Red | Intención Empresarial |

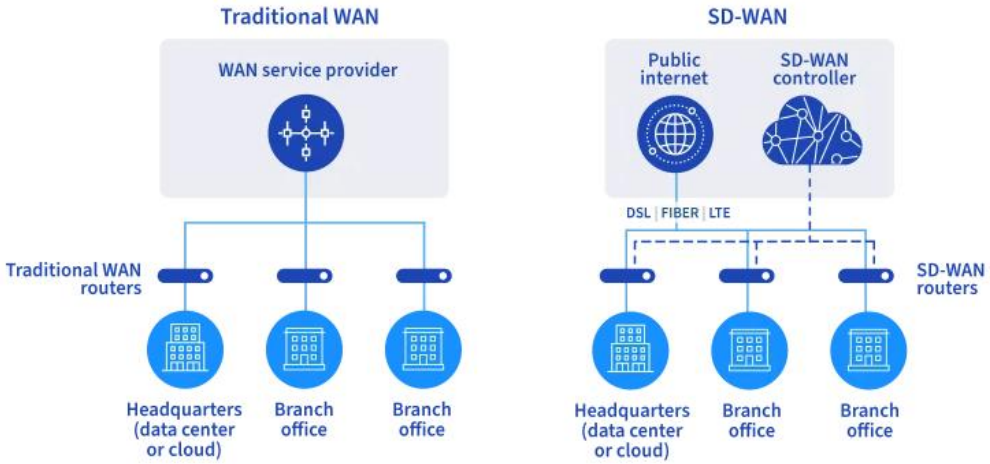
*Nota.* Adaptado de (NetworkAcademy.io, 2021b).

Cabe recalcar que en cada una de las ventajas previamente mencionas existen algunas especificaciones importantes como son:

- El bajo costo es independiente de la red de acceso, ya sea mediante MPLS u otras tecnologías relacionadas.
- En cuestión de rendimiento y flexibilidad, existe una mejora debido al uso eficiente del ancho de banda en las aplicaciones, adicional presenta beneficios en lo que respecta a seguridad.
- Fácil acceso a los servicios de la nube como se muestra en la Figura 2, debido a que se enfoca en simplificar las operaciones basándose en una automatización.

**Figura 2**

*Esquema de una red WAN frente a* una *SD-WAN*




*Nota.* Infraestructura de una red WAN tradicional versus una red SD-WAN, vista de una manera sencilla. Adaptado de (Gervasi, 2023).

### *2.1.1. Proveedores y soluciones SD-WAN en el mercado*

Debido al crecimiento y a las oportunidades que ofrece actualmente el mercado SD-WAN, algunos fabricantes han optado por crear soluciones basadas en esta tecnología, cada proveedor o fabricante presenta su propio conjunto de soluciones y herramientas, anualmente la organización de investigación Gartner proporciona un análisis exhaustivo y de evaluación de varios proveedores en el mercado WAN-Edge (Sol, 2020).

**Figura 3**

*Cuadro Mágico para SD*-WAN


*Nota*. Principales empresas que brindan los servicios de SD-WAN, con relación a cuatro factores importantes. Adaptado de (Fortinet, 2023).

En la Figura 3 se aprecia una representación gráfica de Gartner, llamado cuadrante mágico para la infraestructura de WAN-Edge, publicado en septiembre del 2023, donde identificar a detalle el lugar de cada fabricante que está inmerso en el desarrollo de redes SD-WAN. Este cuadrante mágico se compone de cuatro secciones: leaders, challengers, visionaries y niche players, estos cuadrantes se basan en dos cosas: la integridad de la visión y la capacidad de ejecución. Como se logra visualizar en la Figura 3, los principales proveedores de soluciones SD-WAN (lideres) son Cisco, Fortinet, VMware, HPE(Aruba), Versa Networks y Palo Alto Networks. Otros proveedores como Juniper Networks son visionarios, Huawei aún se considera como retador, mientras otros proveedores, como Barracuda, Shophos entre otros son actores especializados en soluciones SD-WAN (Yadav, 2021).

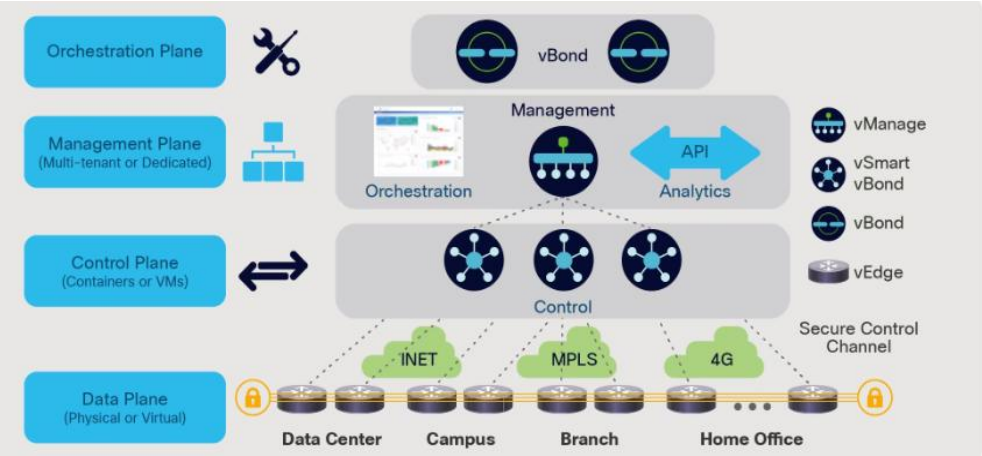
### *2.1.2. Arquitectura y componentes principales de una red SD-WAN.*

SD-WAN utiliza una arquitectura de red abstracta, compuesta por el plano de control y el plano de reenvió(datos). Esta arquitectura traslada la estructura de control a una ubicación central, como la sede de una organización. Así, la red puede gestionarse remotamente sin necesidad de TI en cada ubicación local.

En una red SD-WAN, los planos de gestión, control y datos están separados físicamente, a diferencia de las arquitecturas tradicionales como se muestra en la Figura 4. Esto permite que cada plano trabaje de manera independiente, lo que hace más eficiente el realizar sus tareas y responsabilidad. Según (Cisco, 2020b), el plano de gestión y el plano de control se encuentran en la nube de Internet y el plano de datos continúa siendo un componente físico que conecta los dispositivos de endpoint y/o servidores a través de una nueva interfaz, que es cualquier medio que le permite conectarse a Internet. El plano de datos obtiene información de decisiones y se encarga de enviar los paquetes a un nodo especifico debido a que está ubicado directamente en el nodo o sucursal deseada.

**Figura 4**

*Componentes de la solución SD-*WAN *de CISCO*




*Nota.* Detalle de los elementos que conforman una SD-WAN desde el Plano de Datos hasta el Plano de Orquestación. Adaptado de (Cisco, 2020b)

En la Figura 4 se observan los planos en los que se divide la solución propuesta por el fabricante de CISCO para una SD-WAN. Este tipo de redes se componen por tres planos como se mencionó anteriormente, pero en esta solución existe un nuevo plano el de orquestación, que autentica todos los nodos existentes, encuentra nuevos y mantiene conectados a la red.

Dentro de la arquitectura de una red SD-WAN existen algunos componentes o equipamientos de red, como se muestra en la Figura 4 en el plano gestión y orquestación se encuentra en **vManage** y el **vBond**, en el plano de control el **vSmart** y en el plano de datos el **vEdge**, cada uno cumple una función específica dentro de la red que se especificara a continuación:

- **Cisco vManage**

Es un panel de control centralizado que facilita la configuración, administración y monitoreo automáticos de redes de superposición. Este componente se usa en el nivel de gestión como una interfaz de usuario, permitiendo a los administradores y operadores de red, configurar, resolver problemas, supervisar y alertar sobre eventos no deseados en la red para lograr administrar de forma centralizada todos los aspectos del ciclo de vida de la red, para lo cual recopila datos sobre la conexión de los dispositivos vEdge (Cisco, 2020b).

- **Cisco vSmart**

Se podría considerar como el cerebro de la red, se encuentra en el plano de control y es el responsable de mantener una tabla y políticas de enrutamiento centralizadas, así como de calcular y distribuir rutas al resto de la red. Este controlador establece conexiones SSL seguras a todos los demás componentes de la red, también ejecutan OMP (Overlay Managemnet Protocol) para intercambiar información de enrutamiento, seguridad y políticas. El motor de políticas centralizado en los controladores vSmart proporciona construcciones de políticas para manipular la información de enrutamiento, el control de acceso, la segmentación, las extranets y el encadenamiento de servicios (Cisco, 2020b).

- **Cisco vEdge**

Son dispositivos IP completamente funcionales, responsables de crear las conexiones de red y supervisar el tráfico, estos pueden ser de distintas formas, ya sean físicos o virtuales. Realizan tareas estándar como BGP (Border Gateway Protocolo), OSPF (Open Shortest Path First), ACLs (Listas de Control de Acceso), QoS (Calidad de Servicio) y diversas políticas de enrutamiento y también gestionan la comunicación de superposición. Estos enrutadores establecen una conectividad segura con todos los componentes de control y conectan sesiones IPsec con otros enrutadores vEdge en la red WAN (Cisco, 2020b).

- **Cisco vBond**

Se encuentra a un nivel centralizado, estos controladores son los principales responsables de brindar servicios que facilitan la actualización inicial al realizar la autenticación y autorización de todos los elementos en la red. El orquestador vBond también proporciona información sobre como cada uno de los componentes se conecta a otros componentes. Este dispositivo desempeña un papel importante para facilitar la comunicación con los dispositivos que se encuentran detrás de la Traducción de Direcciones de Red (NAT) (Cisco, 2020b).

### *2.1.3. Protocolos utilizados en SD-WAN*

Al igual que en las redes WAN, en las redes SD-WAN se establecen protocolos y mecanismos de comunicación o de red entre los dispositivos los cuales se detallan a continuación:

#### 2.1.3.1. Overlay Management Protocol (OMP).

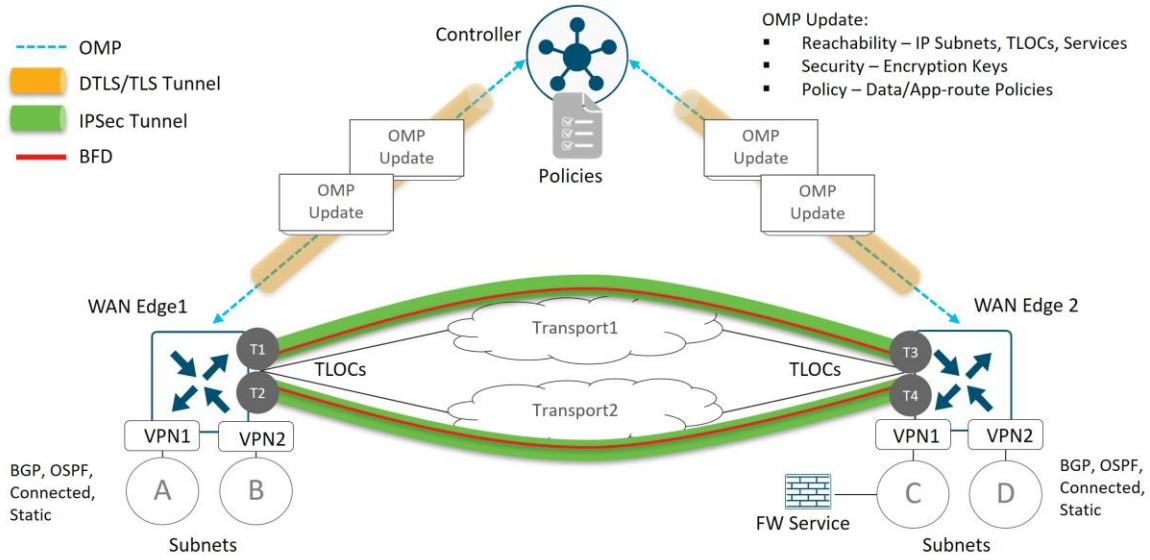
El Protocolo de Gestión de Superposición (OMP) se utiliza para gestionar la red superpuesta en una SD-WAN. Este protocolo facilita el intercambio seguro de información del plano de control entre los routers de borde WAN y los controladores vSmart, tal como se muestra en la **¡Error! No se encuentra el origen de la referencia.**. La información del plano d

e control que se transmite incluye prefijos de ruta, rutas de siguiente salto, claves criptográficas e información de políticas. Si no se define ninguna política, el comportamiento predeterminado de OMP permite una topología de malla completa entre los routers de borde WAN o vEdge, lo que significa que cada uno de estos routers pueden conectarse directamente con sus pares (Cisco, 2024c). Este protocolo anuncia tres tipos de rutas:

- Las rutas OMP son prefijos que se aprenden en el sitio local y se redistribuyen en OMP para que puedan transportarse a través de la superposición de la red. Estas rutas anuncian varios atributos, entre ellos la información de ubicación de transporte (TLOC), comparable a una dirección IP de siguiente salto BGP para la ruta, además de otros atributos como origen, originador, preferencia, ID de sitio, etiqueta y VPN. Una ruta OMP solo se instala en la tabla de reenvió si el TLOC al que está asociada se encuentra activo(Cisco, 2024c).
- Las rutas TLOC son puntos lógicos de terminación de túnel en los routers WAN Edge que se conectan a una red de transporte. Una ruta TLOC se identifica de manera única y se representa mediante una tripleta, que consiste en la dirección IP del sistema, el color del enlace y la encapsulación(Cisco, 2024c).
- Las rutas de servicio representan servicios como cortafuegos, IPS, optimización de aplicaciones, etc., que se conectan a la red de sitio local WAN Edge y se pueden usar como un tipo de encadenamiento de servicios en otros sitios, adicionalmente, estos caminos incluyen VPN. En este tipo de actualización, se envían etiquetas VPN para que los controladores vSmart sepan que VPN están recibiendo servicio en un sitio remoto(Cisco, 2024c).

**Figura 5**

*Diagrama Overlay Management Protocol (OMP)*




*Nota.* Adaptado de (Cisco, 2024c)

### 2.1.3.2. Bi-directional Forwarding Detection (BFD).

La detección de reenvió bidireccional es un mecanismo usado por los routers WAN Edge para sondear y medir el rendimiento de los enlaces de transporte. También determina la ruta de mejor rendimiento basándose en el resultado de las sondas BFD, proporcionando información sobre latencia, fluctuación (jitter) y perdida en todos los enlaces de transporte, con estos datos, los vSmart son capaces de determinar la mejor ruta y comunicar las mismas con los routers vEdge para garantizar la comunicación de los diferentes nodos (Cuaical, 2023).

En el apartado de guías de alta disponibilidad, Juniper (2022) explica que el protocolo BFD es un mecanismo de saludo (handshake) utilizado para detectar fallas en una red, estos "handshakes" se intercambian entre el plano de control y el plano de datos a intervalos específicos, lo que permite realizar diferentes mediciones. Por lo tanto, BFD representa una contribución significativa para la solución SD-WAN, ya que es compatible con una amplia gama de redes y topologías.

Cisco (2024b) señala que, una vez configurada una sesión BFD y establecidos los tiempos de saludo, se envían paquetes de control que funcionan de manera similar a lo paquetes de saludo utilizados en los protocolos de enrutamiento IGP. Estos paquetes ayudan a identificar la actividad de los nodos necesarios. La principal diferencia entre IGP y BFD es que BFD opera a una mayor velocidad, lo que permite una detección más rápida de fallas en la red.

El protocolo BFD se utiliza para detectar fallas en las rutas de reenvió dentro de la red de transporte. Aunque BFD es eficaz en la detección de estas fallas, la responsabilidad de implementar medidas para evitar la pérdida de paquetes recae en el protocolo de enrutamiento. Aunque BFD puede operar en cualquier capa de los protocolos de enrutamiento, en la práctica, por las limitaciones de las versiones de IOS de los equipos, se usa en la capa 3. En caso de que varios protocolos de enrutamiento compartan el mismo enlace, BFD no establece sesiones individuales para cada uno, en su lugar, configura una única sesión para un protocolo y comparte la información con los demás (Cuaical, 2023).

### *2.1.4. Ingenieria de tráfico y seguridad en SD-WAN.*

La ingenieria de tráfico es importante para la disponibilidad y fiabilidad de la red. Al implementarla, se optimiza el uso de los recursos de la red, lo que no solo mejora la eficiencia, sino que también fortalece la red frente fallos de los enlaces o nodos. El controlador SD-WAN tiene que orquestar el tráfico para que el rendimiento de la WAN medido por la supervisión pueda coincidir con los requisitos del servicio en cada instante (Troia et al., 2021).

La aplicación de ingenieria de tráfico consta de dos componentes principales:

- Clasificación de servicios: clasifica los paquetes en la entrada vEdge en una clase de servicio especifica, como VoIP, transmisión de video, etc., para evaluar los requisitos del servicio.
- Implementación de políticas: aplica políticas SD-WAN asociadas a la clase de servicio para seleccionar las superposiciones más adecuadas en cada momento.

Con respecto a la seguridad SD-WAN Edge utiliza autenticación y cifrado para proteger completamente el tráfico de extremo a extremo. Los expertos en seguridad informática pueden supervisar la calidad de las conexiones y garantizar que todas las comunicaciones cumplen las políticas de seguridad y fiabilidad de la empresa. SD-WAN Edge proporciona cifrado de extremo a extremo a través de cualquier tipo de red subyacente, incluida Internet, explotando tecnologías como IPSEC VPN, IKEv2 con certificado, cifrado de extremo a extremo mediante AES256, claves compartidas y PKI (Troia et al., 2021).

### *2.1.5. Casos de uso y aplicaciones de SD-WAN*

Durante la pandemia de COVID-19, las empresas de la India y del mundo en general tuvieron que adaptarse al trabajo remoto para garantizar la seguridad de sus empleados, lo que provocó un aumento de la demanda de soluciones SD-WAN. Con la necesidad de conectividad de red y seguridad para los empleados remotos, SD-WAN demostró ser una solución fiable y eficaz, la reciente implementación de la tecnología 5G en la India ha impulsado la demanda de soluciones SD-WAN. En la India, se espera que el mercado de SD-WAN crezca a una CAGR del 25,6% entre 2020 y 2025. A nivel mundial, se prevé que el mercado de SD-WAN crezca de 3.400 millones de USD en 2022 a 13.700 millones de USD en 2027, con una CAGR del 31,9% durante el periodo de previsión (Sharma et al., 2023).

A nivel mundial los sistemas críticos como educación, salud y sector energético a través del uso de aplicaciones basadas en la nube y otras tecnologías que dieron soporte a este tipo de servicios para el desenvolvimiento desde la casa debido a los confinamientos por la pandemia priorizo el uso de SD-WAN. Esta coyuntura permitió el despliegue de SD-WAN con mayor rapidez, entre algunos de los usos y aplicaciones se encuentra la conectividad de sucursales y oficinas remotas, optimización de aplicaciones en la nube, seguridad distribuida, mejora de la experiencia del usuario para aplicaciones en tiempo real, consolidación de redes multi sitio, soporte para infraestructuras hibridas, automatización y orquestación de redes, etc.

## 2.2. APIs en una red SD-WAN

Una API (Interfaz de Programación de Aplicaciones) es un software que permite a otras aplicaciones acceder a sus datos e interactuar con ellos. Es utilizada para controlar, administrar, orquestar y generar telemetría a través de un intercambio de solicitudes y respuestas, permitiendo él envió de información en tiempo real. Las APIs establecen un conjunto de reglas que describen como una aplicación puede interactuar con otra y son altamente personalizables según los dispositivos disponibles en el mercado. Estas interfaces se desarrollan o aplican para sistemas específicos, operando mediante una clave pre compartida única, conocida como API key (G. Salazar & Marrone, 2021).

### *2.2.1. Rol de las APIs en la administración y automatización de redes SD-WAN.*

SD-WAN proporciona interoperabilidad, lo que significa que puede funcionar y coexistir con la infraestructura de red existente, lo que permite una implementación adecuada y una interoperabilidad usando las APIs disponibles, debido a que las APIs permiten la flexibilidad y escalabilidad de las redes SD-WAN al facilitar la comunicación entre los dispositivos de red heterogéneos y las plataformas de gestión basadas en la nube, como vManage en el ecosistema de Cisco (Cisco, 2019).

Las APIs cumplen con un rol importante en la administración y automatización de las redes SD-WAN al proporcionar un medio estándar para la interacción programática entre los distintos componentes de la red y las plataformas de gestión. En una arquitectura SD-WAN, las APIs permiten la integración de soluciones de gestión y orquestación, habilitando funciones como la configuración automatizada de rutas, la provisión de políticas de seguridad y la monitorización del rendimiento de las aplicaciones en tiempo real, lo que facilita no solo la configuración inicial, sino también el ajuste dinámico de los parámetros de operación a medida que cambian las condiciones de la red (Scarpitta et al., 2021).

En la actualidad la capacidad de automatización programática antes mencionada es particularmente valiosa en escenarios de redes complejas y distribuidas, donde las aplicaciones empresariales pueden estar alojadas en múltiples ubicaciones y en diferentes nubes, lo que requiere un enfoque flexible y escalable para la gestión de la red (G. Salazar & Marrone, 2021).

### *2.2.2. Principales APIs usadas en entornos SD-WAN.*

Entre las principales APIs usadas en entornos SD-WAN se encuentran RESTful APIs, APIs REST, NETCONF, gRPC, la cuales permiten una comunicación eficiente entre controladores y dispositivos de red, optimizando la configuración como la monitorización. Otras APIs como RESTCONF y SOAP son fundamentales al momento de integrar diferentes servicios y tecnologías dentro de un entorno SD-WAN.

**Tabla 2**

*Principales APIs usadas en* entornos *SD-WAN*

| Nombre de la API | Descripción |
|---|---|
| **REST API (Cisco vManage)** | API utilizado por el controlador Cisco vManage para la gestión centralizada de la red, configuración y monitorización. |
| **NETCONF** | Protocolo de configuración de red que utiliza YANG como modelo de datos, permitiendo la configuración automatizada y segura. |
| **RESTCONF** | Variante simplificada de NETCONF que usa HTTP para realizar operaciones de configuraciones de red con modelos YANG. |
| **gRPC(Google Remote Procedure Call)** | API utilizado para la comunicación eficiente entre controladores y dispositivos en arquitecturas SD-WAN con baja latencia. |
| **SOAP API (Simple Object Access Protocol)** | Protocolo basado en XML que permite la comunicación entre aplicaciones de red, usado para monitoreo y configuración. |
| **JSON-RPC** | Protocolo ligero que utiliza JSON para invocar funciones remotas en entornos SD-WAN, facilitando la integración con aplicaciones |

*Nota*. Adaptado Cisco (2019) y Landázuri & Verdesoto (2020).

Como se muestra en la Tabla 2 existen algunos tipos de APIs que son ampliamente adoptadas por su simplicidad y compatibilidad con diversas plataformas, otras que son muy

usadas son las RESTful APIs, debido a que permiten realizar operaciones de configuración y monitorización mediante protocolos HTTP, facilitando de esta manera la interacción entre controladores y dispositivos de borde (Cisco, 2019).

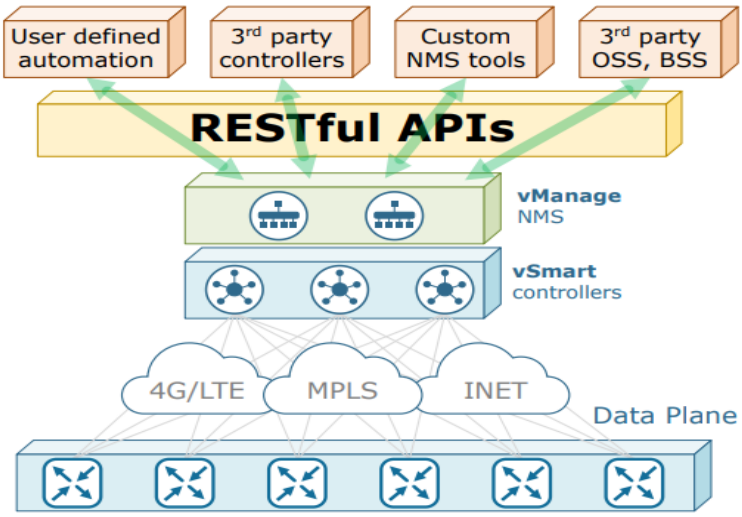
### *2.2.3. Ejemplos y requerimientos de la implementación de APIs en SD-WAN.*

La implementación exitosa de APIs en un entorno SD-WAN requiere de una infraestructura de red que soporte tanto controladores centrales como dispositivos de borde compatibles con las APIs estandarizadas, adicionalmente requiere de una integración fluida con plataformas de orquestación y automatización como Ansible o Terraform y contar con mecanismos de autenticación y seguridad robustos para proteger las comunicaciones API, especialmente en entornos milti-tenant o con acceso a internet (Landázuri & Verdesoto, 2020).

Como se muestra en la Figura 6, un ejemplo destacado de la implementación de APIs en SD-WAN es la solución Cisco vManage, que utiliza una API REST para permitir la gestión centralizada de múltiples dispositivos y la automatización de tareas operativas críticas.

**Figura 6**

*Solución Cisco SD-WAN REST APIs*




*Nota.* Estructura de RESTful API y su acople a vManage. Adaptado de (NetworkAcademy.io, 2021a).

### *2.2.4. Evolución de las APIs y nuevas posibilidades*

Vargas (2020) señala que la evolución de las APIs en el contexto de SD-WAN ha sido impulsada por la necesidad de simplificar la gestión de redes cada vez más complejas y distribuidas. Inicialmente, las APIs permitían operaciones básicas de configuración, pero hoy en día soportan funciones avanzadas como la automatización completa del flujo de trabajo, el análisis predictivo basado en datos de telemetría y la integración con plataformas de inteligencia artificial.

En el futuro, se espera que las APIs evolucionen para ofrecer capacidades de programación aún más sofisticadas, habilitando la automatización impulsada por eventos y la integración con soluciones de redes autónomas basadas en inteligencia artificial y machine learning (Cisco, 2019).

## 2.3. Automatización en Redes SD-WAN

La automatización de una red SD-WAN consiste en la capacidad de gestionar y configurar la red de manera programada, reduciendo al mínimo la necesidad de intervención manual. Lo que permite automatizar tareas como el provisionamiento, la gestión de políticas, el escalado, el diagnóstico y la resolución de problemas, mejorando los tiempos de respuesta de la red ante diferentes escenarios que se puedan presentar al interactuar con otros dispositivos.

### *2.3.1. Beneficios de la automatización en la administración de redes SD-WAN*

La automatización en redes SD-WAN optimiza las operaciones al reducir las tareas manuales, incrementando la eficiencia de administración y minimizando errores humanos. Mediante el uso de APIs, se facilita la administración centralizada, permitiendo la implementación automática de políticas de configuración y monitoreo, lo cual es clave para mantener altos niveles de rendimiento y seguridad sin necesidad de intervención constante (Yadav, 2021).

Esta capacidad resulta esencial para redes complejas que requieren adaptaciones continuas y permite una administración eficiente al simplificar la configuración de múltiples dispositivos distribuidos, logrando una mayor eficiencia operativa y la gestión de redes de expansión sin un aumento significativo en el esfuerzo humano (Azhar, 2021).

Entre los beneficios destacan:

- Las aplicaciones uniformes de políticas de seguridad.
- Optimización dinámica del tráfico de red.
- Reducción de costos operativos a través de una configuración y monitoreo centralizados.

Además, la automatización permite una respuesta proactiva, anticipando y resolviendo problemas antes de que afecten el rendimiento de la red.

### *2.3.2. Herramientas y tecnologías para la automatización de redes*

Las herramientas clave para la automatización de redes SD-WAN incluyen plataformas de gestión como Cisco vManage, que permite administrar la infraestructura de manera centralizada a través de APIs RESTful, y soluciones ampliamente adoptadas como Ansible y Terraform, que permiten automatizar despliegues, configuraciones y la ejecución de scripts personalizados, cabe destacar también Python herramienta muy útil para el llamado de APIs (NetworkAcademy.io, 2021a).

Ansible utiliza Playbooks para aplicar configuraciones simultáneamente en múltiples dispositivos, mientras que Terraform facilita la infraestructura como código (IaC) para crear, configurar y gestionar entornos SD-WAN de manera rápida y eficiente. De igual manera, bibliotecas de Python como Netmiko y NAPALM simplifican la conexión y gestión de dispositivos de red mediante scripts personalizados, proporcionando una integración fluida y flexible en el control de la red (G. Salazar & Marrone, 2021).

### *2.3.3. Plataformas y entornos de automatización*

Existen diversas plataformas y entornos que permiten una administración y monitorización centralizada, estas plataformas integran APIs RESTful que permiten la orquestación remota y automatizada, así como la implementación de políticas específicas para el tráfico de red y la seguridad. La elección de plataformas y entornos depende de la arquitectura de la red y de la necesidad de compatibilidad con otros servicios en la nube o infraestructura local (Azhar, 2021).

En la Tabla 3, se presenta algunas de las plataformas y entornos más usados con relación a las redes SD-WAN.

**Tabla 3**

*Plataformas/Entornos para la* automatización *en redes SD-WAN.*

| **Plataforma/Entorno** | **Descripción** | **Características Principales** | **Aplicaciones en SD-WAN** |
|---|---|---|---|
| **Cisco DevNet** | Plataforma de Cisco para desarrollo y pruebas en entronos de red. | Sandboxes para simulación, integracion con APIs de Cisco, soporte para desarrollo con Python y Ansible. | Desarrollo de soluciones personaliadas, pruebas de automatización con APIs de vManage, validación de configuraciones antes de producción. |
| **Ansible** | Herramienta de automatización de TI de código abierto. | Basada en YAML (Playbooks), facil integración con SD-WAN, automatización de configuraciones y tareas repetitivas. | Configuración de dispositivos, despliegue de políticas de red, automatización de flujos de trabajo. |
| **Terraform** | Plataforma de IaC (Infraestructura como Código) de Hashicorp. | Define la infraestructura en archivos de configuración, soporta proveedores como Cisco, AWS, Azure. | Aprovisionamiento y configuración de infraestructura SD-WAN, gestión de recursos en la nube y locales |
| **Cisco vManage** | Plataforma de administración centralizada para Cisco SD-WAN. | APIs RESTful, monitorización en tiempo real, gestion de políticas y configuración centralizada. | Administración de la infraestructura SD-WAN, integración con herramientas de automatizacion, gestión de telemetría. |

| Plataforma/Entorno | Descripción | Características Principales | Aplicaciones en SD-WAN |
|---|---|---|---|
| **Vmware VeloCloud** | Solución SD-WAN de Vmware, centrada en redes distribuidas. | Administración en la nube, integración con herramientas de monitoreo, optimización de aplicaciones. | Gestión centralizada de redes distribuidas, optimización de la experiencia de usuario final, configuración de seguridad y políticas de red. |
| **Postman** | Herramienta de colaboración para pruebas de API. | Testing de APIs REST, automatización de llamadas API, creación de scripts y pruebas integradas. | Pruebas de APIs en entornos SD-WAN, integracion de APIs en flujos de trabajo de automatización. |
| **Python SDK para Cisco SD-WAN** | SDK de Python proporcionado por Cisco para interactuar con vManage. | Simplificación de llamadas API en Python, automatización de tareas de red, compatibilidad con CI/CD. | Desarrollo de scripts personalizados para configuración y monitorización de dispositivos SD-WAN, integración con pipelines de DevOps. |

*Nota.* Descripción de ciertas plataformas y entornos para la automatización de redes SD-WAN. Adaptado de NetworkAcademy.io (2021a) y Yadav (2021).

### *2.3.4. Integración de APIs con herramientas de automatización*

El uso de APIs en conjunto con herramientas de automatización modifica el concepto de WAN en la administración de redes, permitiendo de esta manera una gestión escalable y más ágil. En soluciones como Cisco vManage, las APIs REST y RESTCONF permiten que plataformas de automatización o leguajes de programación como Python se comuniquen directamente con la infraestructura de la red y de esta manera llevar a cabo configuraciones, monitoreo y ajustes en instantes, lo que conlleva a mejorar las demandas operativas y reduciendo la necesidad intervenciones manuales (Azhar, 2021).

Esta integración facilita la coordinación eficiente de los dispositivos SD-WAN, permitiendo automatizar tareas complejas en redes distribuidas en base a flujos de trabajo estandarizados y fáciles de utilizar, lo que permite acortar tiempos de respuesta y por ende agiliza el despliegue de la red. Existen herramientas de orquestación, como Ansible o

Terraform a las cuales se les puede integrar estas APIs y que son compatibles con ciertas infraestructuras SD-WAN, por ejemplo, al usar Ansible, las APIs permiten que la configuración sea dinámica y presente un monitoreo continuo de los dispositivos, lo cual es de gran ayuda debido a que optimiza la red y responder de inmediato a cambios que puedan ocurrir con respecto al tráfico y demanda que se presente en la SD-WAN (G. Salazar & Marrone, 2021).

Por otro lado, el uso de las APIs RESTful como las de Cisco vManage, facilita la interacción entre el sistema SD-WAN y otras plataformas de monitoreo y administración. Lo que permite la recopilación de telemetría, configuración automática de políticas, y la capacidad de responder de forma ágil ante incidentes, derivando en una gestión proactiva y optimizada de la red (NetworkAcademy.io, 2021a).

### *2.3.5. Tendencias en la Automatización de Redes SD-WAN mediante APIs*

Las tendencias actuales en la automatización de redes SD-WAN en base a las APIs está avanzando hacia una administración más predictiva y autónoma, aprovechando la actualidad de la inteligencia artificial y del Machine Learning para anticipar y resolver problemas de manera automática. Esta integración es más analítica y permite que dichas redes se ajusten a configuraciones en tiempo real y con reducida intervención humana, optimizando de esta manera la eficiencia y el rendimiento operativo (Cisco, 2019).

La implementación de APIs facilitará la interoperabilidad entre distintas plataformas lo cual será un punto clave para alcanzar una administración predictiva de la red, además, la adopción de modelos de seguridad Zero-Trust asegura la conectividad en entornos multi-cloud, permitiendo una respuesta más segura y adaptativa a las cargas de trabajo cambiantes. Estos desarrollos están transformando a las redes SD-WAN en infraestructuras resilientes y escalables, facilitando una gestión de red proactiva y preparada para las crecientes demandas de conectividad distribuida en las empresas (G. Salazar & Marrone, 2021).

# CAPÍTULO III: Desarrollo de la Solución

En este capítulo se presenta una descripción más detallada de la solución propuesta, partiendo de un análisis de requerimientos técnicos de una red SD-WAN, al considerar aspectos como los dispositivos que la conforman, protocolos de comunicación involucrados, configuraciones básicas y flujos de trabajo operativos, permitiendo tener una visión clara del entorno a ser automatizado. En base al análisis previamente expuesto llevar a cabo el desarrollo de una solución de automatización mediante el uso de la API de vManage que permite la gestión de la red de manera eficiente, para lo cual se utilizará herramientas y tecnologías compatibles con entornos virtualizados como lo son GNS3 como plataforma de emulación y Cisco Viptela como solución por su alta compatibilidad con APIs REST.

## 3.1. Requerimientos Técnicos y Operativos de una Red SD-WAN.

En las redes WAN tradicionales, los enrutadores tomaban decisiones de reenvío basadas en un conjunto limitado de información. Donde los protocolos de enrutamiento tradicionales generalmente consideran solo el ancho de banda del enlace y el estado del enlace. Sin embargo, tener múltiples-diferentes transportes WAN conectados a un sitio remoto y desear usarlos de manera activa requiere un proceso de reenvío de enrutamiento más complejo. Con SD-WAN, los enrutadores de borde ahora pueden confiar en el plano de control y gestión centralizando la información, lo que permite tener un auxilio al momento de reenviar el tráfico. Un ejemplo práctico es del GPS, el cual ayuda a los conductores a evitar los típicos retrasos en los viajes, de la misma manera SD-WAN ayuda a los enrutadores a evitar fluctuaciones, pérdida de paquetes y latencia en la red.

Ahora si bien SD-WAN mejora las condiciones de enrutamiento más complejas también requiere de la automatización de procesos, y es ahí donde las APIs intervienen debido a que permiten a IT (Tecnología de Infraestructura) integrar de forma fluida su entorno operativo existente con SD-WAN, ofreciendo un nivel alto de flexibilidad con el fin de entregar

nuevas capacidades, por ejemplo, las herramientas empresariales de Gestión de Servicios de IT, pueden usarse para automatizar tickets de problemas si se detectan fallas en la WAN, mientras que herramientas de monitoreo de red pueden agregar datos de la WAN con datos del resto de la red. Esto proporciona una vista más holística del sistema, lo cual facilita la identificación y posible solución de problemas de manera más rápida. Las APIs también pueden ser usadas por los distintos proveedores de manera que puedan integrarlas a sus servicios de operaciones y facturación, monitoreando redes de clientes y el uso de estas, a la vez que orquestan cambios a gran escala en muchos nodos. En la actualidad la integración de las APIs a una red SD-WAN ha pasado de ser un plus a ser un requisito debido a que sin ellas una SD-WAN moderna estaría limitada y perdería gran parte de su valor estratégico, lo que significa un retraso en la búsqueda de una agilidad alineada con las necesidades de negocio y de seguridad de las redes actuales.

### *3.1.1. Requerimientos de Software para el Entorno de Despliegue*

Para el desarrollo y posterior implementación de la solución propuesta, se establecen algunos requerimientos de software contemplados para un despliegue adecuado, tomando en cuenta que la emulación se realizará en un entorno virtual controlado.

**Tabla 4**

*Especificaciones técnicas de despliegue vs emulación*

| **Componente (Cisco)** | **Recurso** | **Requerimientos de Despliegue (Producción)** | **Recursos de Emulación (Trabajo de grado)** |
|---|---|---|---|
| **vManage** | vCPUs /RAM /Storange | 16 vCPUs / 32 GB / 500 GB | 2 a 4 vCPUs / 10 GB / 30 GB |
| **vSmart** | vCPUs /RAM /Storange | 2 vCPUs / 4 GB / 20 GB | 2 vCPUs / 4 GB / 8 GB |
| **vBond** | vCPUs /RAM /Storange | 2 vCPUs / 4 GB / 20 GB | 1 vCPUs / 1 GB / 8 GB |
| **vEdge (Router)** | vCPUs /RAM /Storange | 1 vCPUs / 2 GB / 2 GB | 1 vCPUs / 1 GB / 2 GB |
| **Instancia Python** | vCPUs /RAM | 1 vCPUs / 2 GB | 1 vCPUs / 2 GB |

En la Tabla 4 se detalla las especificaciones mínimas para un despliegue en entornos reales de producción y los estipulados para el presente trabajo de grado, información proporcionada por la propia página de documentación de Cisco (Cisco, 2025).

Los valores establecidos para el presente trabajo de grado se obtuvieron de recomendaciones a través de personas de la comunidad de GNS3 que habían probado un despliegue similar en proyectos implementados por su autoría, se tomó en cuentas estos criterios debido a que la emulación también se desarrolló en GNS3(Augusto, 2018).

Es necesario tener cierto criterio en el uso del sistema operativo en el cual se pretende llevar a cabo la emulación debido a que existe una gran variedad y funcionan de forma distinta, lo que puede afectar el rendimiento de la emulación debido a que los recursos se gestionan con relación a este, entre los sistemas operativos más usados se encuentran: Linux, macOS y Windows.

Para el desarrollo del presente trabajo de grado se ha optado por elegir, como sistema operativo una distribución de Linux en este caso Ubuntu (versión 24.04 LTS al momento de realizar este trabajo de grado), debido a que tanto esta distribución como Debian optimizan recursos y baja el impacto en el rendimiento del sistema contrario a Windows que usa de manera excesiva los recursos del propio sistema operativo, y no permite una flexibilización y personalización de este, dato importante al realizar ciertas simulaciones. Adicionalmente en Windows es necesario desplegar una máquina virtual GNS3_VM con el fin de lograr emular las controladoras y los nodos de acceso, lo que consumiría aún más recursos.

### *3.1.2. Requisitos Operativos de la Red*

La operabilidad de la red SD-WAN mediante el uso de APIs establece algunos requerimientos que se detallan a continuación, de los cuales algunos son esenciales y otros se los detalla como parte del procedimiento para un posterior análisis en el trabajo de grado que se desarrolla.

**Tabla 5**

*Requerimientos Operativos*

| Requerimiento | Descripción operativa | Principales artefactos/Endpoints vManage |
|---|---|---|
| On-boarding automático de dispositivos (ZTP/PnP) | Cada nuevo vEdge se incorpora a la superposición sin intervención manual: obtiene IP, verifica firma, recibe UUID/serial y se registra en vBond, vManage y vSmart | POST/divice/action/device/ provision y POST /edge/network-device |
| Gestión centralizada de configuración | Crear plantillas y aplicarlas masivamente.<br>Versionar/guardar (configs) y realizar rollback<br>Exponer todo mediante REST para integrarlo con scripts Python. | POST /template/device/attach y Swagger "On-board API Docs" |
| Establecimiento y gestión de túneles seguros | El panel de control debe orquestar túneles Ipsec/DTLS auto descubiertos entre Transport Locators (TLOCs) y mantener sesiones BFD/OMP activas. | Configuración vía platillas; Métricas vía GET /statistics/tunnel |
| Ruteo dinámico consiente de aplicaciones (Application-Aware Routing) | Medir latencia, perdida y jitter por BFD y redirigir flujos si un enlace viola el SLA definido. | POST /policy/central; métricas en: GET /statistics/app-route |
| Monitorización y telemetría en tiempo real | Ofrecer paneles y API para:<br>• Estado BFD/OMP de cada túnel.<br>• CPU/RAM de dispositivos.<br>• Eventos y logs.<br>• Exportar en JSON, XML o PDF. | GET /device/realtime/monitor, GET /events |
| Políticas de segmentación y servicio (VPN/VRF) | Crear hasta 256 VPNs lógicas, aplicar ACL/QoS Y aislar tráfico entre segmentos, todo gestionado vía API/plantillas. | POST /template/policy/vpn |
| Detección y respuesta a fallo | Detección menor a 1 segundo con BFD<br>Conmutación automática a enlace de respaldo.<br>Registro de evento. | Get /alarms + syslog/ REST webhook |
| Escalabilidad mínima de laboratorio. | RAM de 32 GB del host soportará, al menos: 1vManage, 1vSmart, 1vBond y 4-5 vEdge(2vCPU y 2GB cada uno) | Fichas vManage/vEdge |

Cómo se logra visualizar en la Tabla 5 existen algunos requisitos para la operatividad de una red SD-WAN mediante el uso de APIs, criterios que se deben tomar en cuenta al momento de diseñar y ejecutar los scripts con las configuraciones que se pretende indexar a la

red. Entre los cuales se menciona la gestión centralizada y el establecimiento de túneles seguros IPsec precisamente entre los vEdges.

### *3.1.3. Elección de Plataforma de Emulación*

Una vez seleccionado el sistema operativo en el cual se va a desarrollar el trabajo de grado, se procede a realizar un análisis comparativo de ciertas plataformas de emulación, como se presenta en la siguiente tabla:

**Tabla 6**

*Características de* Plataformas *de Emulación*

| Plataforma | Ventajas | Desventajas |
|---|---|---|
| **GNS3** | Interfaz intuitiva.<br>Compatible con imágenes de routers reales (IOS, VyOS, etc).<br>Permite la integración con scripts de Python y herramientas externas.<br>Software libre y de uso gratuito para la academia. | Puede consumir muchos recursos dependiendo el sistema operativo en el que esté corriendo.<br>Requiere de una configuración inicial |
| **EVE-NG** | Emulación Profesional<br>Escalabilidad y soporte para múltiples vendors.<br>Soporte en laboratorios complejos. | Requiere de una licencia Pro para el uso de ciertas funciones.<br>Requiere de un mayor tiempo de aprendizaje. |
| **Packet Tracer** | Uso sencillo, óptimo para prácticas básicas. | No soporta la emulación de redes SD-WAN reales ni la integración de APIs |
| **Minunet** | Ideal para pruebas de redes simples y de SDN, muy ligero. | Presenta algunas limitaciones en el desarrollo de topologías de SD-WAN completas. |

En base a ciertos criterios descritos en la Tabla 6 como escalabilidad, compatibilidad, integración, licencias y requerimientos de hardware se ha seleccionado GNS3 como la plataforma para llevar a cabo el presente trabajo de grado, debido a que esta plataforma o software de simulación permite: emular redes SD-WAN, ejecutar e importar imágenes de sistemas operativos de red, y sobre todo la ejecución de scripts de automatización realizados

en Python. Adicionalmente es compatible con sistemas operativos como Windows, macOS y con cualquier distribución Linux, sobre todo las distribuciones Ubuntu y Debian (GNS3, 2025).

## 3.2. Arquitectura de la Red SD-WAN Automatizada

La siguiente sección se centra en la arquitectura de la solución SD-WAN en la que se basará el desarrollo del presente trabajo de grado, proporcionando ciertos elementos y conexiones que forman parte de la estructura de una red SD-WAN básica y que para el presente estudio contará con un factor adicional denominado automatización.

### *3.2.1. Componentes Lógicos: Planos, Capas, Equivalencia con el Modelo OSI.*

A continuación, se detallan los componentes lógicos, sus funciones y protocolos considerados en la arquitectura sobre la cual se basa el desarrollo del presente trabajo de grado. Adicionalmente se ha considerado una comparación con el modelo OSI con el fin de facilitar la comprensión del funcionamiento de cada capa.

**Tabla 7**

*Modelo Lógico de SD-WAN: Componentes y Mapeo OSI*

| Capa/Plano | Componentes en la Arquitectura | Funciones | Equivalencia Modelo OSI | Protocolos |
|---|---|---|---|---|
| Plano de Gestión | vManage NMS | Configuración y gestión centralizada.<br>Provisión y despliegue de dispositivos. | Capa 5 (Sesión)<br>Capa 6 (Presentación)<br>Capa 7 (Aplicación) | NETCONF, TLS/DTLS, RBAC y ACL |
| Plano de Control | vSmart Controller | Distribución de políticas de enrutamiento y seguridad.<br>Toma de decisiones de ruta para el tráfico. | Capa 3 (Red)<br>Capa 4 (Transporte)<br>Capa 5 (Sesión) | OMP, DTLS, TLS |
| Plano de Orquestación | vBond Orchestador | Autenticación inicial de componentes.<br>Establece conectividad Inicial. | Capa 5 (Sesión)<br>Capa 7 (Aplicación) | DTLS, TLS |
| Plano de Datos | vEdge Router | Reenvió de paquetes de datos de usuario.<br>Establecimiento de túneles seguros. | Capa 2 (Enlace de datos)<br>Capa 3 (Red), Capa 4 (Transporte) | IP, Ipsec, GRE, DTLS |

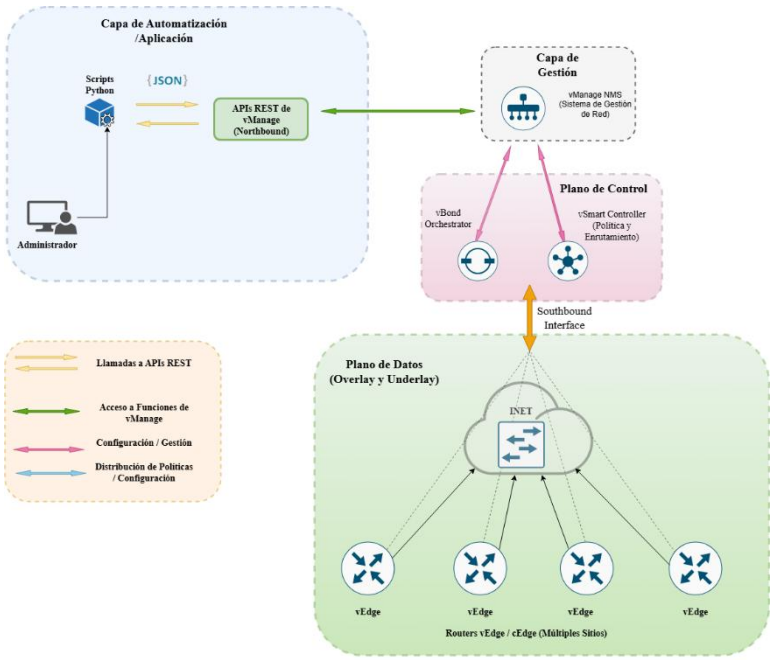
| Capa/Plano | Componentes en la Arquitectura | Funciones | Equivalencia Modelo OSI | Protocolos |
|---|---|---|---|---|
| Capa de Automatización/ Integración | Python Script, API (RESTful APIs), etc. | Interacción Programática. Automatizar configuraciones y operaciones. | Capa 7 (Aplicación) | TLS, RESTful APIs, NETCONF |

La Tabla 7 presenta una vista detallada de la arquitectura de red del sistema, estructurada en cinco planos importantes como son: Gestión, Control, Orquestación, Datos y Automatización, cada uno con sus componentes y funciones. Estos componentes forman parte esencial de la red que será emulada debido a que cada uno de ellos cumple funciones específicas como se ha detallado en la tabla, cabe recalcar que los protocolos que se mencionan son parte de las redes SD-WAN por ende contemplan un aspecto importante de la misma, por último, se detalla una equivalencia no tan llamativa con el modelo OSI pero que permitirá la comprensión de cómo se estructura cada plano o capa SD-WAN con redes tradicionales.

### *3.2.2. Diagrama de Arquitectura de la Solución*

**Figura 7**

*Arquitectura de la* Solución



En la Figura 7 se logra apreciar la arquitectura del sistema propuesto, organizada en capas jerárquicas. En la parte superior izquierda se observa la Capa de Automatización dentro de la cual se ubican el Administrador de la Red, Scripts de Python, formato JSON y APIs REST que permite la comunicación con la Capa de Gestión(vManage), la cual interactúa con el Plano de Control que incluye el vBond y el vSmart. El plano de Datos se encuentra conectado al plano de control por medio de una interfaz Southbound, representa el underlay/overlay con elementos como los vEdge, routers y edges en múltiples sitios, lo cual facilita el acceso a funciones de gestión, configuración, políticas y cumplimiento a través de llamadas a APIs REST.

### *3.2.3. Diseño y Topología de Prueba*

Una vez analizados cada uno de los componentes lógicos presentes en la arquitectura SD-WAN expuesta en la Figura 7, se procederá a elegir la solución y el diseño de la topología que será usada para la implementación de la red y posteriormente para su automatización.

#### 3.2.3.1. Selección de la Solución SD-WAN.

Para elegir una solución se ha optado por analizar tres opciones de las que existen actualmente, para lo cual se presenta una tabla especificando algunas características importantes de cada solución.

**Tabla 8**

*Comparativa de diferentes* soluciones *SD-WAN*

| Característica | Viptela SD-WAN | Fortigate SD-WAN | Meraki SD-WAN |
|---|---|---|---|
| Despliegue | Opciones en sitio y basadas en la nube. | Opciones en sitio y basadas en la nube. | Basado solo en la nube. |
| Escalabilidad | Alta para grandes empresas, y servicios de proveedores. | Escalable para empresas medianas y pequeñas. | Escalabilidad limitada, usada para emprensas medianas. |
| Seguridad | Ofrece funciones de seguridad elevadas (IPsec), para incluir redes de | Ofrece funciones de seguridad elevadas, incluyendo inspección SSL y control de aplicaciones. | Provee seguridad básica como un firewall y vpn. |

| Característica | Viptela SD-WAN | Fortigate SD-WAN | Meraki SD-WAN |
|---|---|---|---|
| | confianza y permite la segmentación. | | |
| Gestión | Centralizada, manejada desde el dashboard del vManage. | Centralizada, manejada desde el FortiManager dashboard. | Centralizada, manejada desde el Meraki dashboard. |
| Enrutamiento | Soporta protocolos de enrutamiento dinámico como OSPF y BGP. | Soporta protocolos de enrutamiento dinámico como OSPF y BGP. | Soporta protocolos de enrutamiento dinámico como OSPF y BGP. |
| Optimización | Incluye optimización de la WAN como la optimización TCP y la compresión de datos. | Incluye optimización de la WAN como la optimización TCP y la compresión de datos. | No tiene optimización de la WAN. |
| Integración de APIs en SD-WAN | Ideal para redes sencillas y complejas con automatización y escalabilidad. | Falta de features avanzadas para enterprises grandes. | Documentación requiere suscripción para detalles completos. |

*Nota.* Adaptado de (Cuaical, 2023)

En la Tabla 8, se presentan algunas especificaciones de tres soluciones de SD-WAN, y como se logra apreciar se analiza algunas características importantes de cada una, en base a lo cual se podrá elegir la que mejor se acople para el desarrollo del presente trabajo de grado. Para el presente trabajo de grado se ha optado por elegir la solución de Viptela SD-WAN debido a su capacidad de escalabilidad, permite opciones en sitio y basadas en la nube, al tipo de cifrado en este caso con IPsec usado para el tráfico de datos, la integración de algunos tipos de redes de transporte, y sobre todo que es óptimo en la integración de APIs para automatizar algunos procesos mediante llamadas API.

### 3.2.3.2. Selección de la topología de Red para la Simulación.

**Tabla 9**

*Comparación de Topologías SD-WAN*

| Aspecto | Tipo Estrella | Full Mesh | Hub and Spoke | Point to Point | Hibrida |
|---|---|---|---|---|---|
| **Escalabilidad** | Fácil de agregar sitios al hub | Conexiones crecen exponencialmente, | Fácil de añadir spokes. | No escala bien con muchos sitios. | Flexible para crecimiento. |

| Aspecto | Tipo Estrella | Full Mesh | Hub and Spoke | Point to Point | Hibrida |
|---|---|---|---|---|---|
| | central sin aumentar complejidad drásticamente. | limitando redes grandes. | | | |
| **Complejidad de Gestión** | Configuración centralizada, ideal para automatización con APIs. | Requiere más enlaces, ancho de banda y hardware. | Baja a media. Centralizada. | Simple para pares. | Requiere planificación mixta. |
| **Costo** | Menos enlaces y recursos requeridos. | Mas enlaces, ancho de banda y hardware necesario. | Recursos centralizados. | Bajo por par, alto para muchos. | Según la combinación. |
| **Redundancia y Fiabilidad** | Dependiente del centro, falla este y afecta toda la red. | Múltiples rutas directas proporcionan redundancia total. | Baja a media. Depende del Hub. | Sin alternativas directas. | Beneficios de múltiples tipos. |
| **Aplicación de APIs** | Uso básico para monitoreo y configuración del centro. | APIs gestionan todos los túneles y rutas. Orquestación avanzada. | Extensivo en SD-WAN. | APIs establecen y monitorean un enlace directo. | APIs integran múltiples topologías. Ideal para redes dinámicas. |
| **Automatización (APIs)** | APIs configuran el centro. Ideal para redes pequeñas con un solo punto de control. | APIs deben manejar todas las rutas y políticas entre sitios. | APIs centralizadas, gestionan hub and spokes. | APIs configuran un solo enlace directo. Ideal para automatización mínima. | APIs combinan estrategias. Flexible, pero exige integración. |

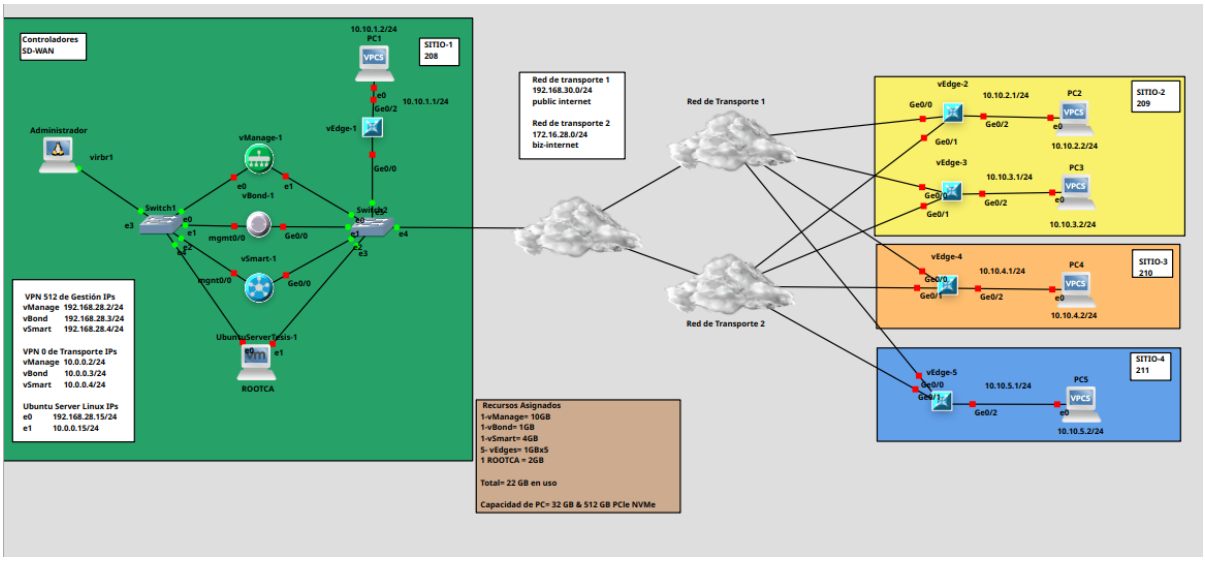
En base a la información recopilada en la Tabla 9, para el presente trabajo de grado se realizará la simulación de la red en base a las topologías tanto de Estrella como Full Mesh, ya que presentan características importantes para uso de APIs y posteriormente la automatización de la red, debido a que existiría un equilibrio entre un uso intermedio de las APIs y una gestión eficiente al momento de ejecutar los scripts.

Estas características permitirán que la simulación no requiera un uso excesivo de recursos, además de evidenciar los beneficios de automatizar la red a través de APIs.

A continuación, en la Figura *8* se presenta un diseño de la topología planteada en donde se evidencia los planos de gestión, control y datos con sus respectivos elementos y conexiones.

**Figura 8**

*Topología de la red SD-WAN*




### 3.3. Implementación y Configuración de la Red SD-WAN

Para la implementación y configuración de la Red SD-WAN se precisa algunas consideraciones con relación a la solución seleccionada para su despliegue en este caso Viptela, que posteriormente se solventarán paso a paso. De igual manera se define un direccionamiento IPv4 como se muestra en la Tabla 10.

**Tabla 10**

*Direccionamiento IPv4*

| Dispositivo | Interfaz | System IP | Site ID | Dirección | Gateway | VPN | Organization-Name |
|---|---|---|---|---|---|---|---|
| **vManage** | eth0 | 1.1.1.1 | 208 | 192.168.28.2/24 | 192.168.28.1 | 512 | benavides-sdwan-tesis |
| | eth1 | | | 10.0.0.2/24 | 10.0.0.1 | 0 | |
| **vBond** | eth0 | 1.1.1.2 | 208 | 192.168.28.3/24 | 192.168.28.1 | 512 | benavides-sdwan-tesis |
| | ge0/0 | | | 10.0.0.3/24 | 10.0.0.1 | 0 | |
| **vSmart** | eth0 | 1.1.1.3 | 208 | 192.168.28.4/24 | 192.168.28.1 | 512 | benavides-sdwan-tesis |
| | ge0/0 | | | 10.0.0.4/24 | 10.0.0.1 | 0 | |
| | e0 | | | 192.168.28.15/24 | 192.168.28.1 | | |

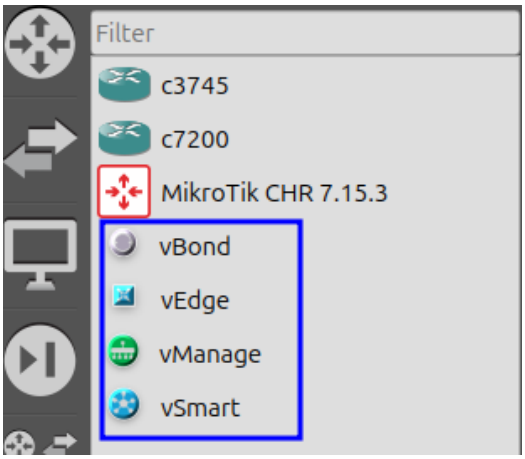
| Dispositivo | Interfaz | System IP | Site ID | Dirección | Gateway | VPN | Organization-Name |
|---|---|---|---|---|---|---|---|
| **Ubuntu-Server ROOT-CA** | e1 | | | 10.0.0.15/24 | 10.0.0.1 | | |
| **vEdge-1** | ge0/0 | 1.1.1.4 | 208 | 10.0.0.5/24 | 10.0.0.1 | 0 | benavides-sdwan-tesis |
| | eth0 | | | dhcp-cliente | | 512 | |
| | ge0/2 | | | 10.10.5.1/24 | | 1 | |
| **vEdge-2** | ge0/0 | 2.2.2.2 | 209 | 192.168.30.2/24 | 192.168.30.1 | 0 | benavides-sdwan-tesis |
| | ge0/1 | | | 172.16.28.2/24 | 172.16.28.1 | 0 | benavides-sdwan-tesis |
| | eth0 | | | dhcp-cliente | | 512 | |
| | ge0/2 | | | 10.10.1.1/24 | | 1 | |
| **vEdge-3** | ge0/0 | 23.23.23.23 | 209 | 192.168.30.3/24 | 192.168.30.1 | 0 | benavides-sdwan-tesis |
| | ge0/1 | | | 172.16.28.3/24 | 172.16.28.1 | 0 | benavides-sdwan-tesis |
| | eth0 | | | dhcp-cliente | | 512 | |
| | ge0/2 | | | 10.10.3.1/24 | | 1 | |
| **vEdge-4** | ge0/0 | 3.3.3.3 | 210 | 192.168.30.4/24 | 192.168.30.1 | 0 | benavides-sdwan-tesis |
| | ge0/1 | | | 172.16.28.4/24 | 172.16.28.1 | 0 | benavides-sdwan-tesis |
| | eth0 | | | dhcp-cliente | | 512 | |
| | ge0/2 | | | 10.10.4.1/24 | | 1 | |
| **vEdge-5** | ge0/0 | 35.35.35.35 | 210 | 192.168.30.5/24 | 192.168.30.1 | 0 | benavides-sdwan-tesis |
| | ge0/1 | | | 172.16.28.5/24 | 172.16.28.1 | 0 | benavides-sdwan-tesis |
| | eth0 | | | dhcp-cliente | | 512 | |
| | ge0/2 | | | 10.10.5.1/24 | | 1 | |
| **VPCS 1** | e0 | | | 10.10.1.2/24 | 10.10.1.1/24 | 1 | |
| **VPCS 2** | e0 | | | 10.10.2.1/24 | 10.10.2.2/24 | 1 | |
| **VPCS 3** | e0 | | | 10.10.3.1/24 | 10.10.3.2/24 | 1 | |
| **VPCS 4** | e0 | | | 10.10.4.1/24 | 10.10.4.2/24 | 1 | |
| **VPCS 5** | e0 | | | 10.10.5.1/24 | 10.10.5.2/24 | 1 | |

### *3.3.1. Despliegue de Controladores*

El despliegue de cada uno de los controladores requiere de algunos pasos y configuraciones, los cuales se detallan en el trabajo de (Cuaical, 2023), para el caso del presente trabajo de grado será de gran ayuda como guía con el fin de tener una configuración correcta y sin ningún inconveniente.

**Figura 9**

*Adhesión de equipos a GNS3*




Para el despliegue de la red es necesario contar con cada uno de los equipos que forman parte de la solución SD-WAN como se observa en la Figura 9, si es posible acceder a cada equipo desde GNS3-client significa que se han cargado correctamente.

Con respecto a las capacidades de cada uno de los equipos según (Cisco, 2025), se recomienda un valor de RAM mínimo como se presenta en la Tabla 11, cabe mencionar que estos datos son para entornos de despliegue físicos (producción), sin embargo, para un despliegue en la nube o local estos valores pueden variar con relación a criterios de diseño propios.

**Tabla 11**

*Valores mínimos en entornos de despliegue físicos*

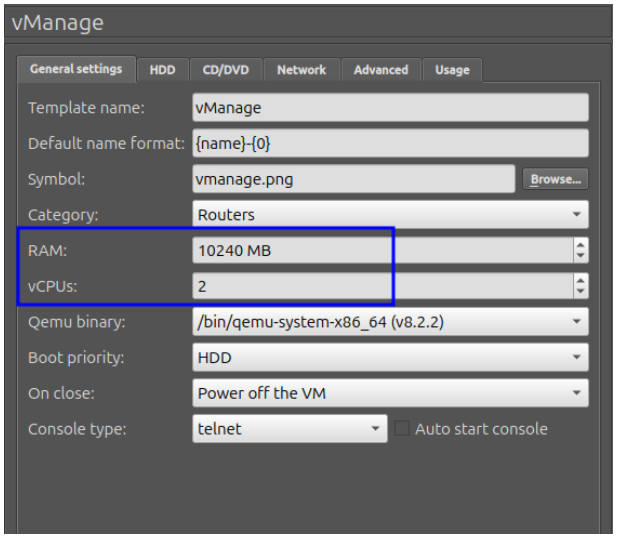
| Componente | Numero de dispositivos Edge | vCPU | RAM |
|---|---|---|---|
| **vManage** | Menor a 250 | 16 | 32 GB |
| **vSmart** | De 1 a 50 | 2 | 4 GB |
| **vBond** | De 1 a 50 | 2 | 4 GB |
| **vEdge** | Cada uno | 2 | 2 GB |

En el presente trabajo de grado se empleará una configuración con valores menores como se detallan en la Tabla 4, debido a que la topología presenta una estructura no tan amplia,

y dado las características del dispositivo en el que se ha desarrollado la emulación. Cabe mencionar que las configuraciones con variación de valores han sido probadas por varios miembros de la comunidad de GNS3 (Augusto, 2018).

**Figura 10**

*Configuraciones Físicas de vManage*




En la Figura 10, se observa el panel de configuraciones del vManage, en el apartado de General settings se localizan dos configuraciones importantes, para el presente trabajo de grado y con relación a las capacidades del equipo en el cual se levanta la emulación, se ha optado por asignar los valores de 10 GB de RAM y de 2 en los vCPUs, los demás campos se establecen por defecto.

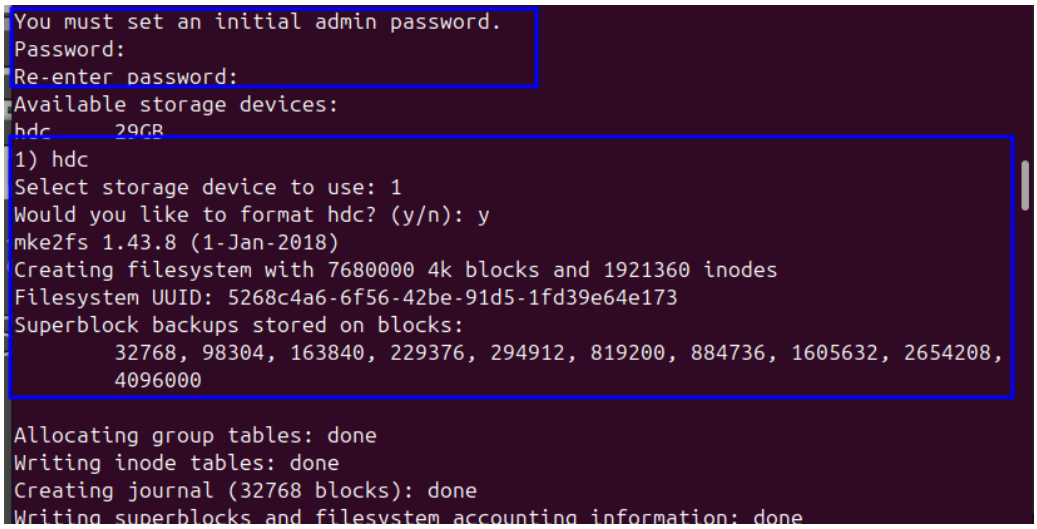
Tal como en el vManage, se realiza las configuraciones de RAM y vCPUs en el vBond, vSmart y los vEdges como se muestra en el Anexo 2, Anexo 3 y Anexo 4. Estas configuraciones son importantes ya que permiten el funcionamiento adecuado de cada uno de los dispositivos y en general de la emulación, con el fin de evitar una sobrecarga de las capacidades del equipo en el cual se levanta la topología de la red SD-WAN.

Una vez realizadas las configuraciones físicas de los cuatro dispositivos de manera adecuada, se procede a iniciar cada uno de ellos a fin de realizar las configuraciones lógicas,

es importante considerar que al arrancar cada uno de los dispositivos es necesario esperar un tiempo prudente hasta que el sistema se marque como system ready, caso contrario el sistema no permitirá ingresar las credenciales de manera correcta y por ende no será posible realizar la autenticación, en general las credenciales por defecto son (login: admin, password: admin).

**Figura 11**

*Inicio y proceso de carga de vManage*

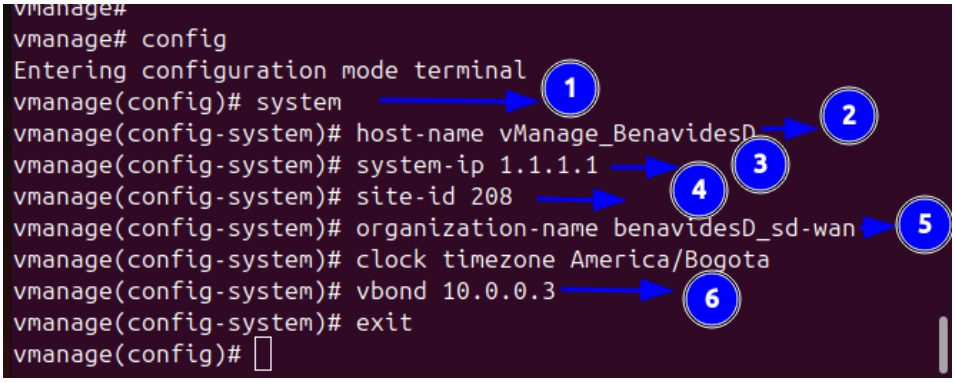



El la Figura 11 se puede observar que el primer dispositivo para configurar será el vManage, en primera instancia se solicita asignar un password de administración, una vez establecido dicho password se procede asignar el espacio de almacenamiento que en este caso es de 29GB; sin embargo suele existir más opciones, en ese caso es importante seleccionar el de mayor capacidad debido a que el vManage es el encargado de recolectar los datos de cada uno de los dispositivos o nodos, y si no se seleccionada correctamente a futuro puede causar algunos errores de rendimiento. Cabe recalcar que este paso es muy importante debido a que sin estas configuraciones el dispositivo no podrá iniciar el sistema al no contar con una unidad de almacenamiento establecida, y si se desea modificar el espacio es imposible cambiarlo mientras el dispositivo este activo.

### 3.3.1.1. Configuraciones de sistema e implementación de túneles.

**Figura 12**

*Configuraciones de sistema de vManage*



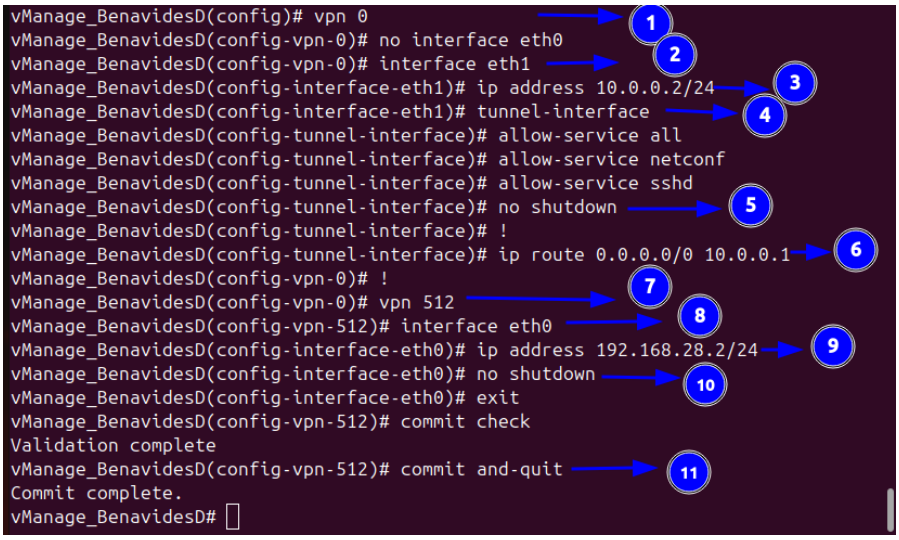

Después de establecer el almacenamiento adecuado e ingresar la nueva contraseña, se procede a realizar las configuraciones de sistema de la controladora como se observa en la Figura 12. Detallando los siguientes pasos:

1. Ingreso a configuraciones de sistema.
2. Nombre de dispositivo
3. Identidad única del equipo (system-ip)
4. Sitio o ubicación del dispositivo (site-id)
5. Nombre de la organización (importante)
6. Dirección IP del vBond (importante)

Una vez realizadas las configuraciones básicas del sistema se procede a configurar parámetros más específicos que permiten el funcionamiento y la conexión adecuada de los dispositivos.

**Figura 13**

*Configuración de interfaces y túnel en vManage.*




En la Figura 13 se observa el modo de configuración del vManage donde se señala algunos comandos específicos para lograr modificar interfaces y VPN en el entorno de red, a continuación, se detallan algunas de estas configuraciones:

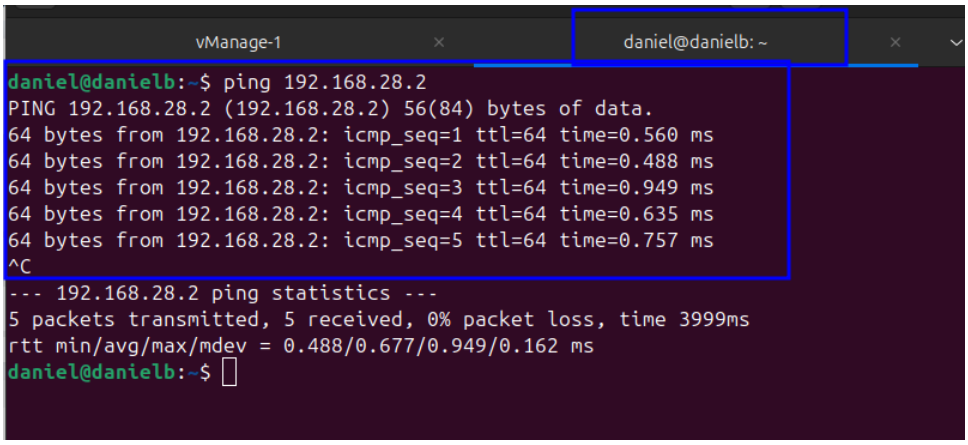
- Paso 1 y 2: configuración de VPN 0 e ingreso la interfaz **eth1** asignada para la VPN 0 o denominada de transporte.
- Paso 3,4 y 5: asignación de la IP 10.0.0.2 con mascara de subred /24, una vez asignada la IP se ingresa submodo tunnel-inteface para habilitar algunas funcionalidades como netconf o ssh, con el fin de asegurar que el dispositivo pueda formar túneles seguros con otros dispositivos o nodos de la red SD-WAN, finalmente se habilita la interfaz “no shutdown”.
- El paso 6 configura una ruta por defecto con el fin de lograr una comunicación con los dispositivos en este caso por el Gateway 10.0.0.1/24.
- En el paso 7 y 8 se visualiza la configuración de la VPN 512 o denominada de gestión en la interfaz **eth0**, asignándole la IPv4 192.168.28.2/24 mediante la cual se podrá ingresar posteriormente al portal web del vManage.

- Finalmente se valida y guarda las configuraciones para abandonar el apartado de configuraciones del sistema mediante el comando commit check y commit and-quit, lo que mostrara un mensaje de validación y uno de configuraciones completas.

Después de validar las configuraciones es posible acceder al portal web del vManage por medio de la IP 192.168.28.2/24 configurada en la VPN 512, para verificar la conectividad desde el dispositivo donde se está corriendo la emulación hacia el vManage se genera un ping como se observa en la Figura 14.

**Figura 14**

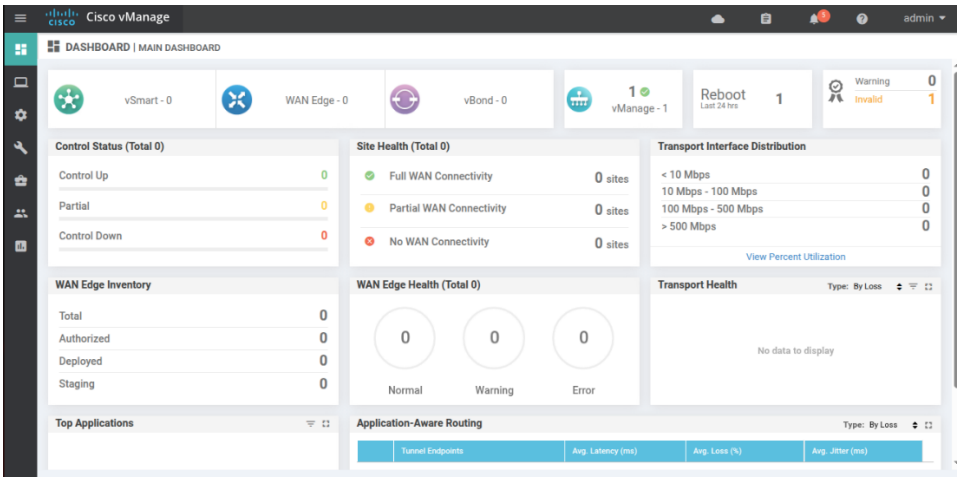
*Ping de verificación de conexión entre PC física y el vManage*




Una vez verificada la conectividad, en el navegador de preferencia se ingresa la IP de gestión configurada en el vManage y de esta manera se accede al portal de administración, por ende, se observa el panel de control como se muestra en la Figura 15. En el panel es factible observar los dispositivos presentes en el cluster de control, al igual que los enlaces de transporte activos de cada router de acceso. Sin embargo, como aún no se ha desplegado los demás dispositivos se puede observar que solo el vManage se encuentra marcado con un visto verde y el número 1, mientras el vBond, vSmart y vEdges se encuentran marcados con el numero 0 al no estar habilitados.

**Figura 15**

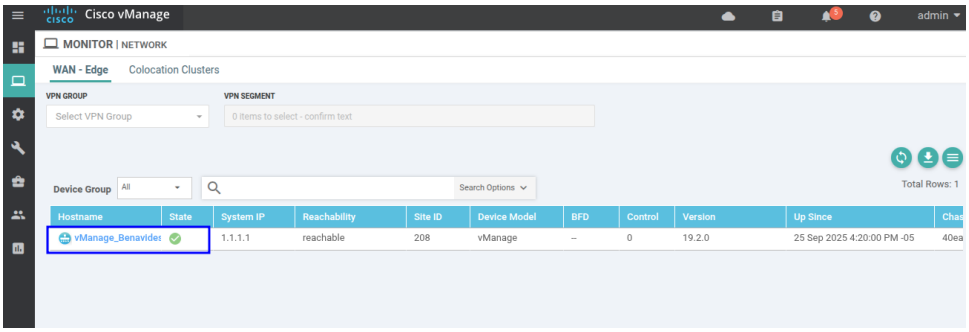
*Vista del panel de control de vManage desde el portal web.*




Al costado izquierdo del panel de control se localiza una barra de opciones en la cual se ubica el apartado de monitor y una vez desplegado se localiza la opción de red (network), en esta pestaña se logra apreciar que el vManage configurado se encuentra en estado activo en la lista de dispositivos, adicional se muestra información como, dirección IP del sistema, numero de chasis del dispositivo y el estado de este como se observa en la Figura 16.

**Figura 16**

*Verificación del vManage activo en la lista de dispositivos de la red.*

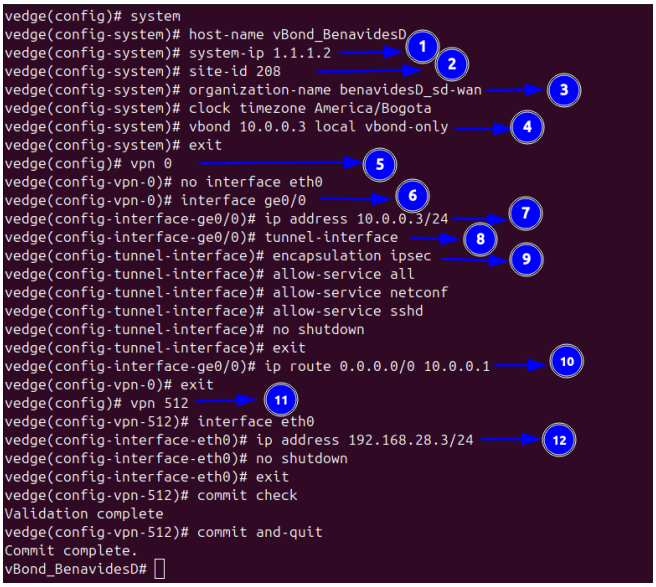



Una vez habilitado el vManage se procede a realizar el despliegue tanto del vBond como del vSmart, equipos son de vital importancia ya que el vBond es el encargado de añadir nodos a la red SD-WAN autenticándolos y asignándoles recursos y el vSmart recolecta las rutas de todos los nodos para compartirlas entre ellos a fin de mantener una comunicación

efectiva, y de esta manera lograr ejecutar políticas del vManage por medio de la red superpuesta.

**Figura 17**

*Configuraciones de sistema, interfaces y túnel en vBond*




En la Figura 17, se logra observar cada una de las configuraciones que se implementan en el vBond mediante la consola del equipo, pasos que se detallan a continuación:

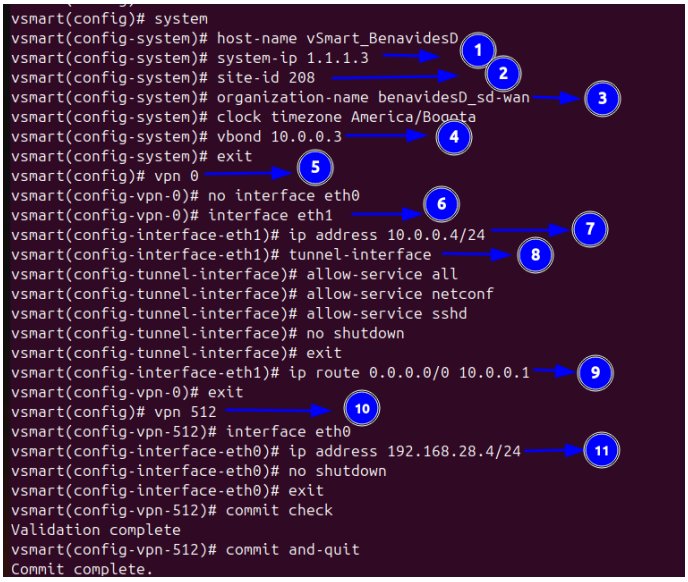
- Paso 1, 2 y 3: configuración de la identidad única del equipo en este caso 1.1.1.2, sitio o ubicación del equipo la misma que en el vManage, nombre de la organización importante ser la misma en cada dispositivo.
- Paso 4: un poco diferente con relación a los demás dispositivos, se asigna una dirección IP en este caso 10.0.0.3, sin embargo, esta dirección se configura como local dentro del vBond, debido a que este dispositivo es el encargado de las autenticaciones de los equipos en la red SD-WAN.
- Paso 5, 6, 7, 8 y 9: en estos pasos se realiza la configuración de la VPN 0 en la interfaz **ge0/0** asignándole la IP 10.0.0.3/24, una vez configurada se ingresa al sub-modo tunnel-interface para habilitar algunas funcionalidades, como la activación del **encapsulation ipsec**, comando que activa la transmisión segura

de los datos debido a que el dispositivo que se configura como vBond-only es un router vEdge.

- Paso 10, 11 y 12 detalla la activación de una ruta por defecto para 0.0.0.0/0 10.0.0.1 para la comunicación, para finalizar se configura la VPN 512 de gestión con la dirección IP 192.168.28.3/24, se verifica y se guarda los cambios con el comando commit.

**Figura 18**

*Configuraciones de sistema, interfaces y túnel en vSmart*

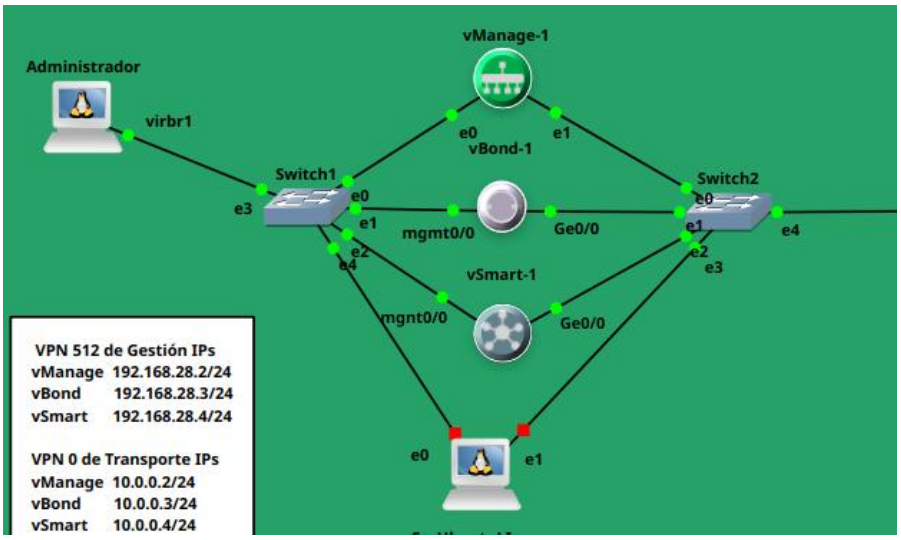



En la Figura 18 se logra observar las configuraciones del vSmart mediante la consola del equipo, cabe destacar que son los mismos pasos expuestos en la Figura 12 y Figura 13, sin embargo, se realiza algunos cambios como el nombre de host, el system ip, dirección IP de la interfaz **eth1** en la VPN 0 10.0.0.4/24 y la IP de la interfaz **eth0** en la VPN 512 192.168.28.4/24.

Una vez configurados cada uno de los controladores, se procede a verificar conectividad entre ellos para lo cual se procede a conectarlos formando una pequeña topología como se observa en la Figura 19, solventando las configuraciones de comunicación entre los controladores.

**Figura 19**

*Esquema de los controladores SD-WAN Viptela*




A pesar de que el vBond y el vSmart se encuentren en la misma red SD-WAN debido a las configuraciones realizadas anteriormente requieren de un paso adicional y muy importante. La configuración y montaje de certificados con los cuales se prioriza la seguridad IPsec para el plano de control y que solo los dispositivos permitidos se encuentren dentro de la red.

### 3.3.1.2. Montaje de certificados.

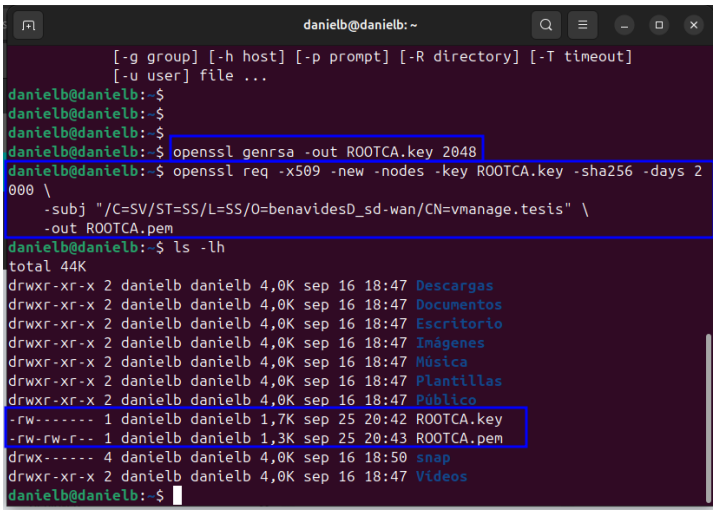
La implementación de los certificados no es una tarea sencilla y requiere del portal web del vManage como la consola de comandos para generarlos, sin embargo, es posible realizarlo desde la consola shell del vManage o desde un servidor conectado a la topología de los controladores, en el presente trabajo de grado se usará la opción 2, para lo cual se ha optado por un servidor Ubuntu 22.04 denominado ROOTCA, cabe recalcar que puede ser cualquier servidor Linux. Para la creación y asociación de dichos certificados se aplicará los siguientes pasos:

1. Se ingresa al servidor ROOTCA en el cual se procede a generar un archivo denominado **ROOTCA.key** que es una llave de 2048 bits y cuenta con un tiempo

de uso. Posteriormente mediante otra línea de comando se establece la organización que firma el certificado, dicho nombre es importante y debe coincidir con el name-organization configurado en cada uno de los dispositivos.

**Figura 20**

*Líneas de comandos para la creación de certificados.*

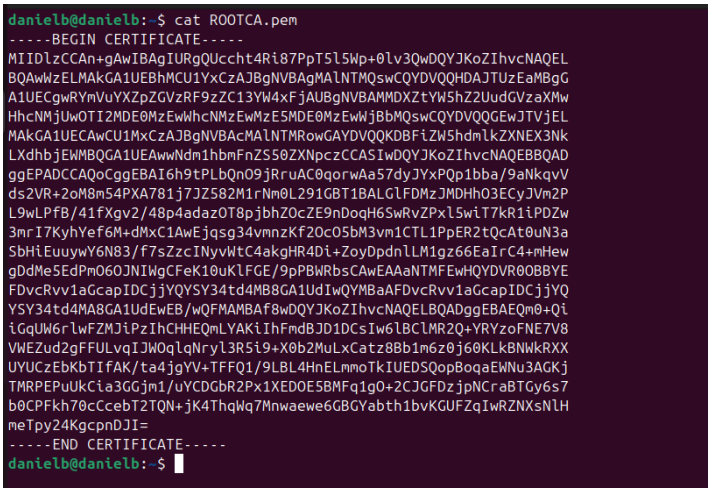



Una vez efectuada la creación de los certificados con los parámetros de configuración seleccionados se accede a la carpeta donde se han almacenado los archivos y se visualiza mediante el comando ls -lh que los dos archivos (.key) y (.pem) se hayan creado correctamente como se muestra en la Figura 20.

2. Una vez generados tanto la llave como el certificado se accede al archivo ROOTCA.pem mediante el comando **cat**, esto mostrará el contenido del archivo el cual se debe copiar desde la línea BEGIN CERTIFICATE hasta la línea END CERTIFICATE como se observa en la Figura 21, importante considerar este paso debido a que si no se copia correctamente el archivo no será entendible para el dispositivo donde se cargará.

**Figura 21**

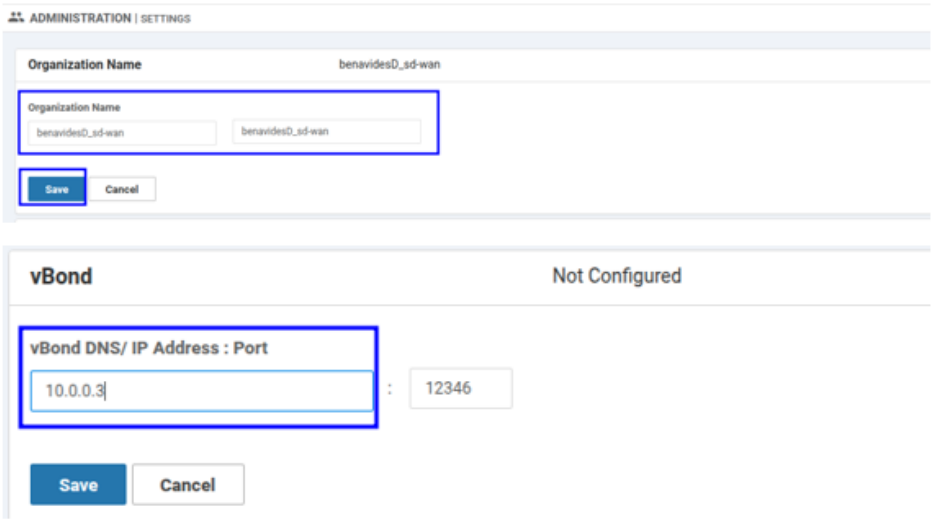
*Contenido del certificado ROOTCA.pem*




3. Copiado el archivo es necesario configurar el nombre de organización y la dirección IP del vBond, para lo cual se accede al panel de control del vManage y a las configuraciones de administración como se observa en la Figura 22.

**Figura 22**

*Configuración del vBond en el plano de control.*

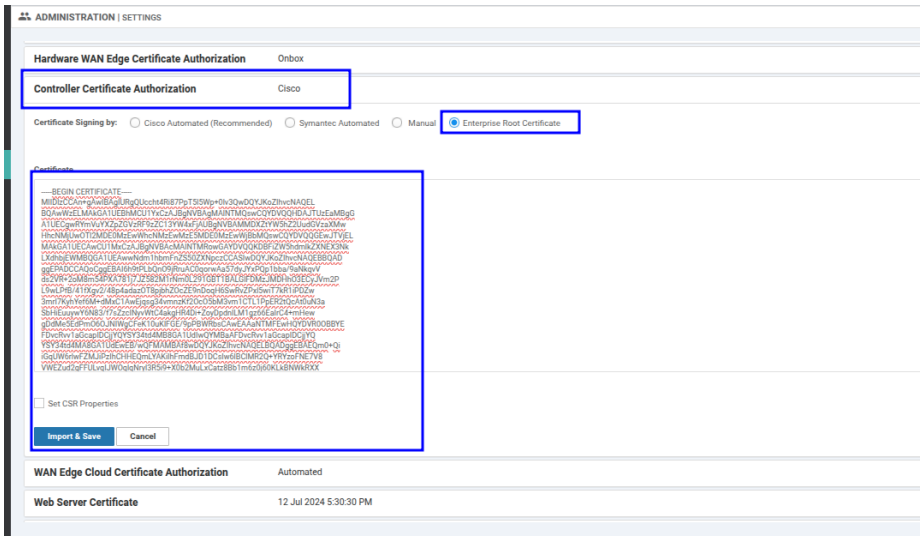



4. Configurados los parámetros de organización y dirección IP del vBond se procede a copiar el contenido del archivo ROOTCA.pem en el espacio dentro de la opción

Enterprise Root Certificate que se muestra en la Figura 23. Para posteriormente ser firmado.

**Figura 23**

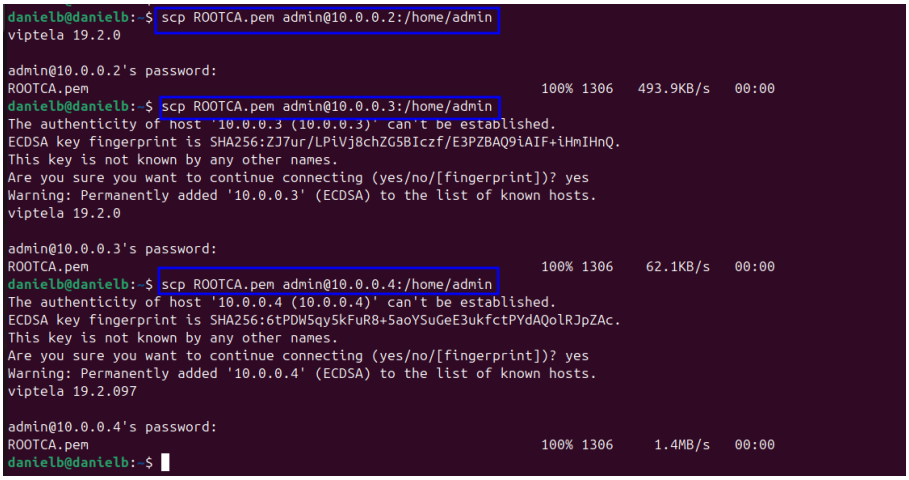
*Montaje del Certificado en el panel de control de vManage*




5. El archivo ROOTCA.pem debe encontrarse en cada uno de los controladores, para lo cual desde el servidor ROOTCA, y mediante el comando scp se procede a copiar el archivo en la ubicación de administración de cada dispositivo como se observa en la Figura 24.

**Figura 24**

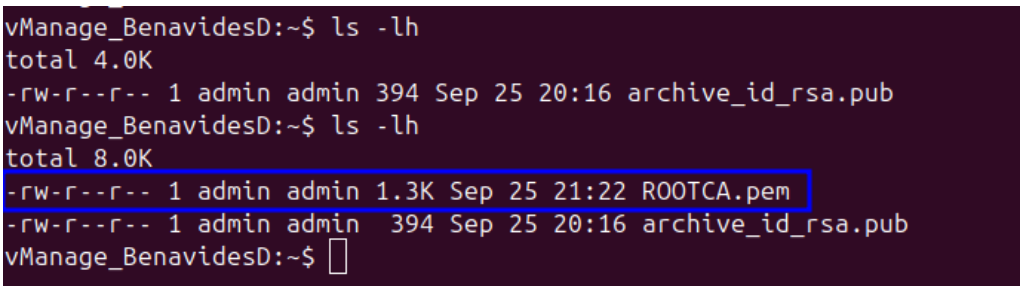
*Copia del certificado en los controladores.*




Posteriormente se procede a verificar que el archivo se encuentre cargado en cada dispositivo, mediante el comando ls -lh como se observa en la Figura 25.

**Figura 25**

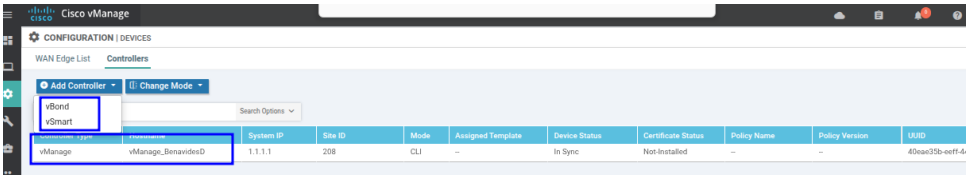
*Verificación del archivo ROOTCA.pem en cada controlador*




6. Una vez cargado el archivo en cada controlador se procede a registrar el vBond y el vSmart, para lo cual es necesario ingresar al panel de control del vManage en la pestaña WAN Edge List apartado Controllers.

**Figura 26**

*Adhesión de vBond y vSmart al vManage*




Como se observa en la Figura 26 el vManage se encuentra registrado, pero no habilitado, por lo cual es necesario registrar el vBond y el vSmart para posteriormente habilitar los tres controladores.

7. Para el registro del vBond y del vSmart es necesario seguir unos pasos sencillos:
    - Ingreso de la dirección IP configurada en la VPN 0 tanto del vBond como del vSmart.
    - Ingreso del nombre de usuario y la clave configurada en cada uno de los dispositivos, es decir la clave solicitada mediante consola al iniciar los controladores.
    - No modificar la casilla Generate CSR y finalmente añadir.

**Figura 27**

*Registro de vBond*

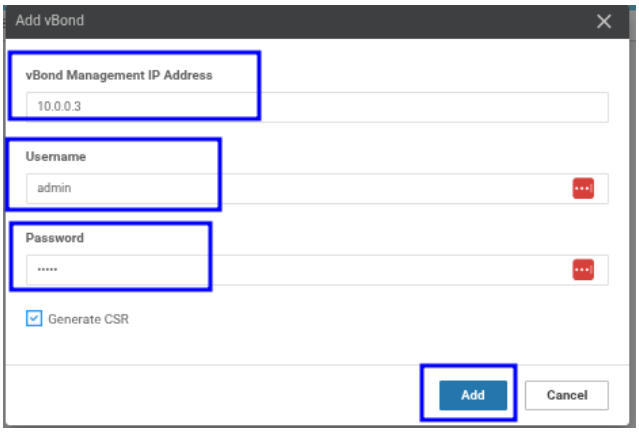
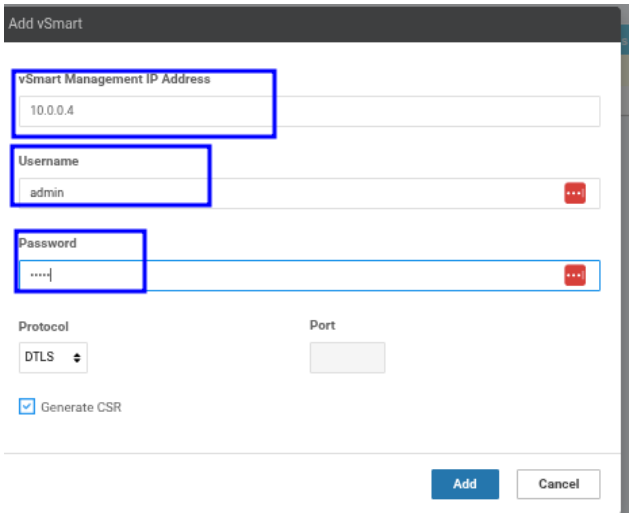


**Figura 28**

*Registro de vSmart*

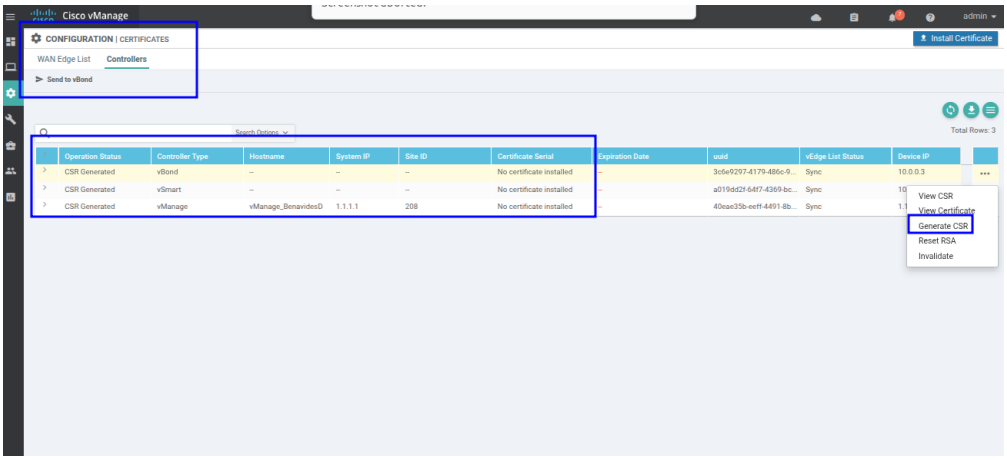



En la Figura 27 y Figura 28 se observa los datos para el registro tanto del vBond como del vSmart, es importante configurar correctamente las direcciones IP, y los dos parámetros de seguridad para llevar a cabo un registro exitoso.

8. Una vez registrados ambos dispositivos con sus direcciones IP, se procede a generar un archivo CSR, dicho archivo es una solicitud de firma de certificado, el cual es muy importante debido a que con esta solicitud se podrá autenticar tanto el vBond como el vSmart en el vManage y lograr activarlos como controladores de la red SD-WAN.

**Figura 29**

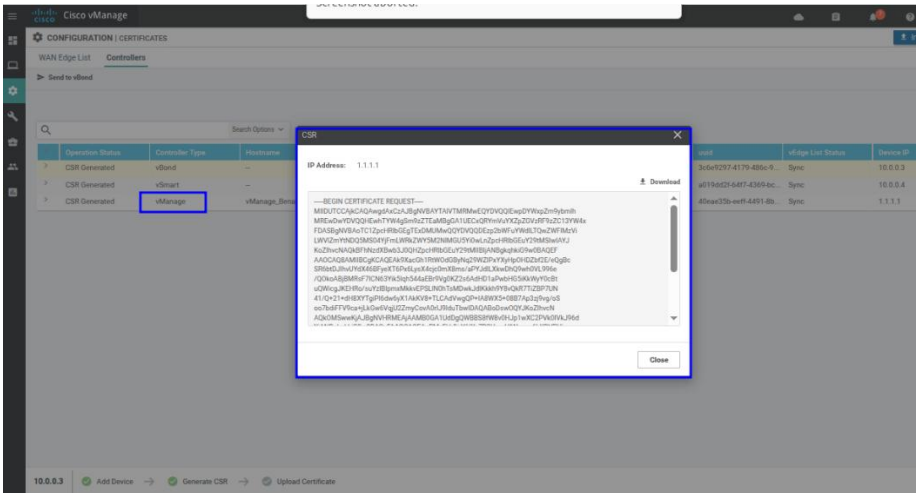
*Verificación de registro de controladores.*




En la Figura 29, se observa que los tres controladores se encuentran registrados en el panel de control del vManage, sin embargo, aún no se encuentran activados y por ende algunos datos en especial del vBond y del vSmart no se muestran en las celdas respectivas.

**Figura 30**

*Generación del archivo CSR para cada controlador*

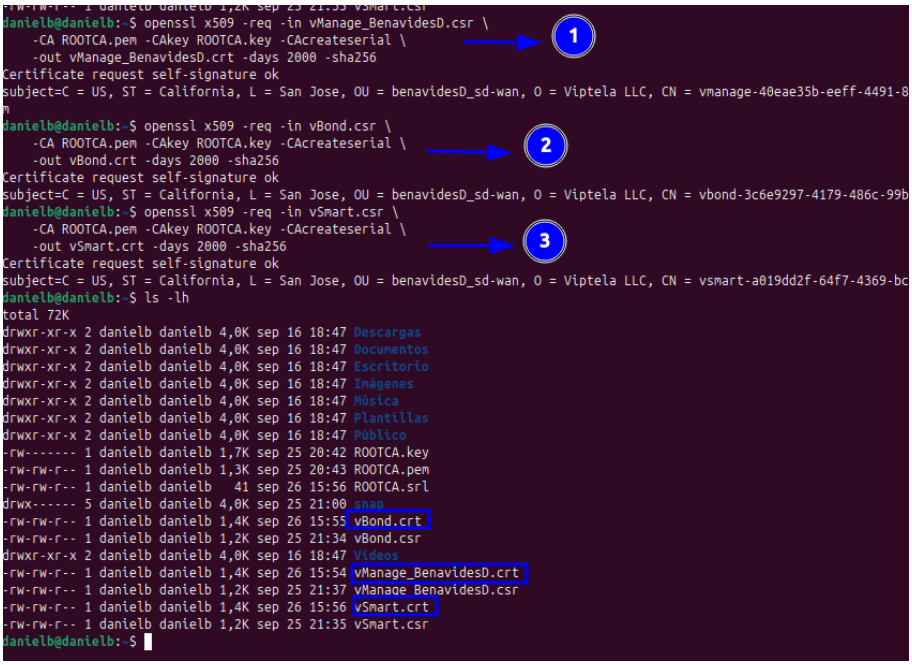



El archivo CSR es un requisito para cada uno de los controladores incluyendo el propio vManage como se presenta en la Figura 30, es necesario descargar los tres archivos para posteriormente firmarlos con el ente certificador ROOTCA.

9. Descargados los archivos CSR se procede a copiarlos al ROOTCA y una vez ahí se procede a firmarlos en base a los archivos ROOTCA.pem y la llave ROOTCA.key, es importante especificar ciertos parámetros como el tiempo de duración de los certificados y el nombre de la organización esto con el fin de que los certificados y los dispositivos se carguen correctamente en la controladora.

**Figura 31**

*Firma de los archivos CSR y creación de los certificados CRT*

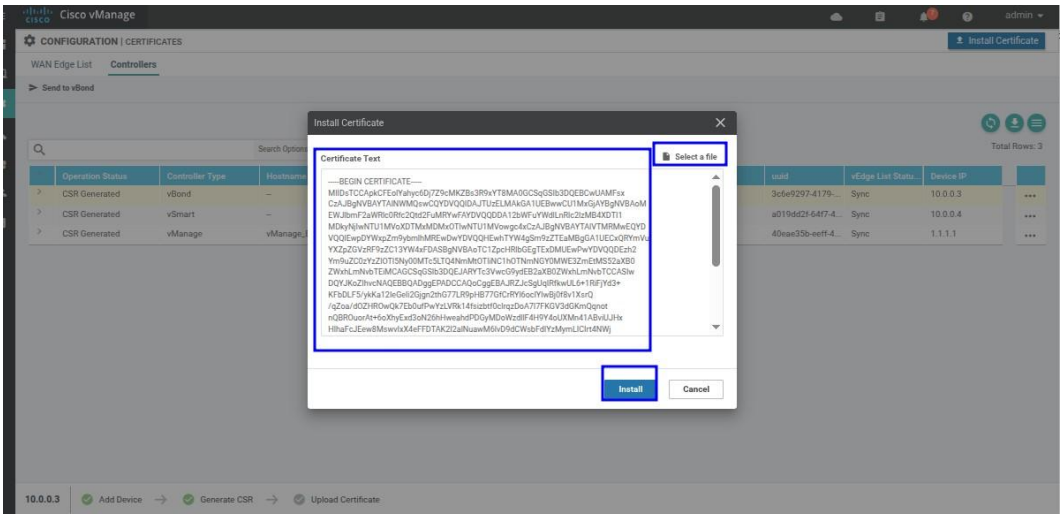



Como se logra apreciar en la Figura 31, se crean tres archivos (certificados CRT) firmados respectivamente por el ROOTCA, con el comando ls -lh se logra visualizar todos los archivos y certificados creados en el servidor ROOTCA.

10. Para cargar cada uno de los certificados en los dispositivos, es necesario visualizar el contenido de cada archivo mediante el comando **cat** o el comando **nano** dependiendo el que se encuentre en el sistema, se copia el contenido del archivo o también es posible pasarlo a un dispositivo USB para cargarlo en la controladora como un archivo certificado.

**Figura 32**

*Carga de los certificados firmados en cada dispositivo.*

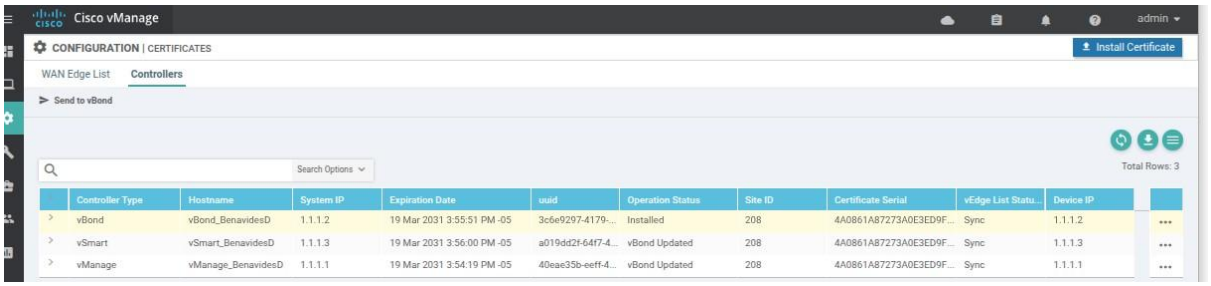



Para lograr cargar los certificados de manera correcta, es necesario seguir algunos pasos: en el panel de control del vManage, seleccionar el apartado de controladoras y seleccionar la opción “Install Certificate”, esto direccionara a una nueva ventana como se observa en la Figura 32, en este espacio se procede a pegar el contenido del certificado o también permite cargar el archivo firmado. Posteriormente, se debe seleccionar la opción “install”, y de esta manera el certificado se instalará de manera adecuada.

Una vez firmado y cargado el certificado es necesario esperar unos segundos hasta que sea verificado y validado, pasado ese tiempo y después de actualizar el panel de controladores es posible verificar el estado de cada uno de los dispositivos y comprobar que están activados correctamente como se muestra en la **Figura *33***, ya se encuentran activados los campos de estatus de operación, serial del certificado y el site ID.

**Figura 33**

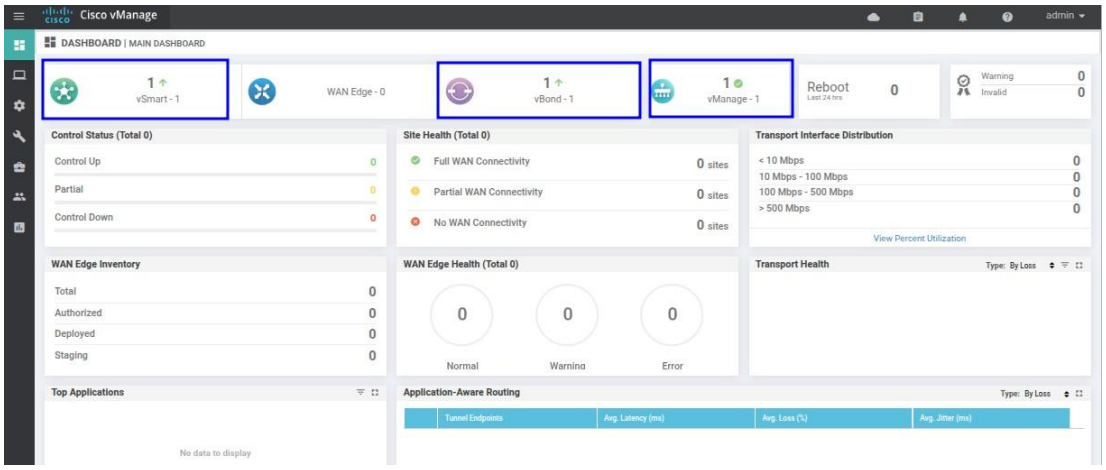
*Estado de los dispositivos cargados los respectivos certificados.*




Una vez validados y aprobados los dispositivos estos serán visibles en el panel principal del vManage, como se observa en Figura 34, los tres controladores ya están habilitados y reconocidos en la red SD-WAN.

**Figura 34**

*Vista de los controladores SD-WAN activados.*




### 3.3.1.3. Despliegue de vEdges

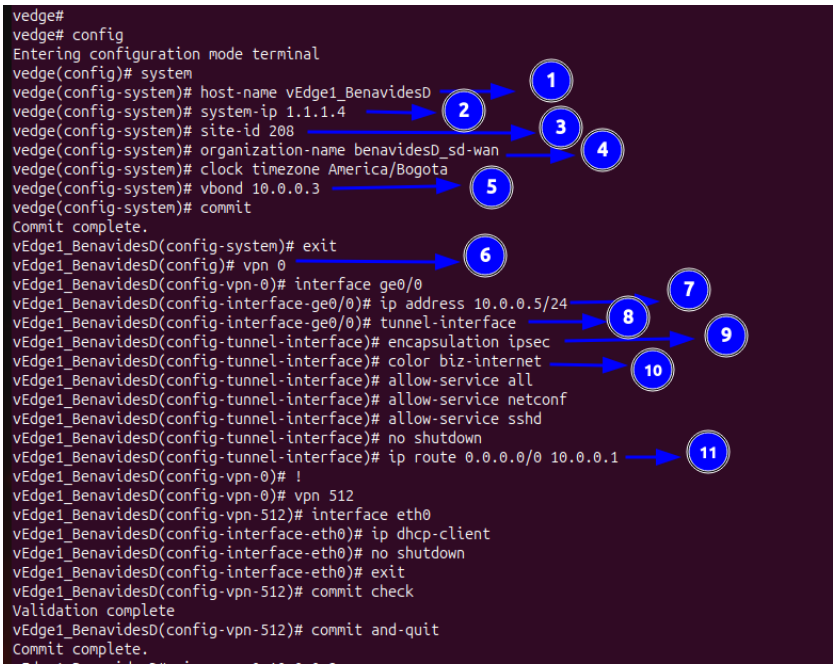
El despliegue y configuración de los dispositivos vEdge requiere de un proceso extra debido a la topología un poco compleja y que abarca ciertos vínculos de transporte con relación al plano de control que por lo general solo cuenta con un enlace para transporte. Por ende, es necesario una configuración un poco más detallada con el fin de asegurar un acceso apropiado a la red superpuesta.

La configuración del vEdge conlleva algunos pasos: primero se implementa las configuraciones básicas o iniciales como las presentadas en la Figura 12, posteriormente se requiere la activación de la comunicación a través del túnel IPsec mediante la encapsulación IPsec y finalmente se asigna un color de enlace de transporte con el fin de evidenciar el uso del protocolo TLOC.

Al igual que las controladoras para realizar las configuraciones dentro del vEdge, se requiere ingresar a través de consola, debido a que aún no se han asignado direcciones IP en las interfaces, para lo cual se requiere inicialmente las configuraciones básicas.

**Figura 35**

*Configuraciones de vEdge desde la consola.*




En la **¡Error! No se encuentra el origen de la referencia.**, se observa cada una de las c onfiguraciones que se debe realizar en cada uno de los dispositivos vEdge que se encuentren en la topología, algunas líneas de comando son repetitivas con relación a las configuraciones de las controladoras (vManage, vBond y vSmart) lo cual facilita la comprensión y configuración que se detalla a continuación:

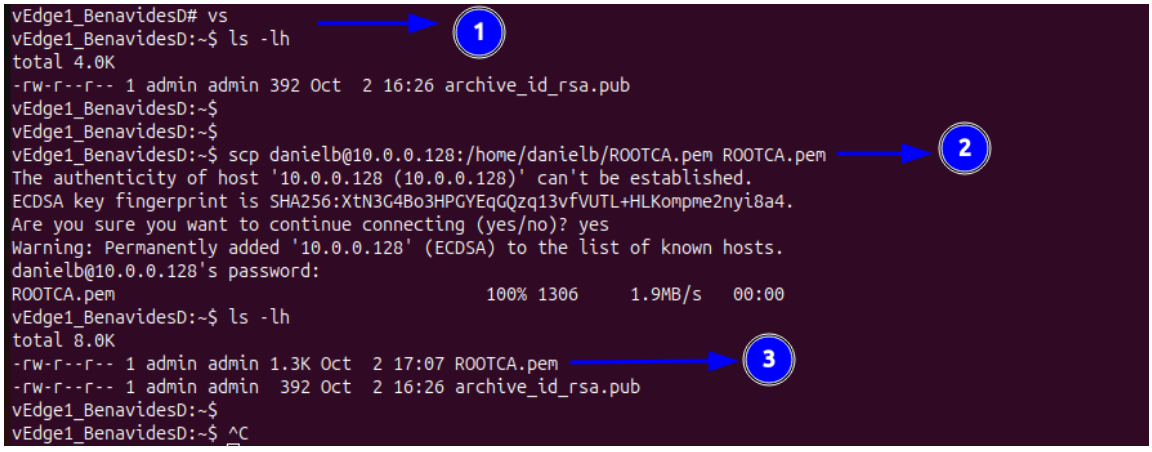
1. Cambio del host- name con el objetivo de distinguir los distintos dispositivos y que no existan confusiones al momento de configurarlos.
2. Configuración de system-ip, previamente establecido en la Tabla 10 con el fin de identificar cada nodo.
3. Ingreso del site-id, en base a criterios propios de la topología.
4. Nombre de la organización, uno de los pasos importantes debido a que si no se configura idéntico a como se encuentra en las controladoras no será posible agregar el nodo a la red.
5. Ingreso de la dirección del vBond, otro paso importante para la autenticación del dispositivo con la red.
6. Configuración de la VPN 0 o denominada de transporte.
7. Asignación de una dirección IPv4 en la interfaz de cada dispositivo vEdge para permitir la comunicación entre ellos.
8. Ingreso a la activación del túnel IPsec mediante el comando “tunnel-interface”: es una de las características de las rede SD-WAN que permite crear una conexión segura entre los diferentes dispositivos de la red.
9. En las configuraciones del túnel se activa la “encapsulación IPsec”, necesaria para cifrar y encriptar la información que se transmite a través de la VPN 0 denominada de transporte.
10. Finalmente se asigna un color de transporte dependiendo la configuración de la red, esta asignación permite el uso del protocolo TLOC el cual contine la información de cada uno de los end-point, adicionalmente se activa una ruta por defecto en base al comando ip route 0.0.0.0/0 10.0.0.1 lo que permite que el dispositivo sea capaz de salir a los demás dispositivos que se encuentran en la red.

Al igual que en los controladores es necesario realizar algunas configuraciones para activar los dispositivos en el vManage mediante la activación de certificados, sin embargo, el proceso para los vEdges es un poco distinto como se detalla en los siguientes pasos:

1. Como punto de partida se copia en cada vEdge el certificado raíz ROOTCA.pem que para el presente trabajo de grado se creó en el servidor certificador ROOTCA.

**Figura 36**

*Carga del certificado root CA en cada vEdge.*

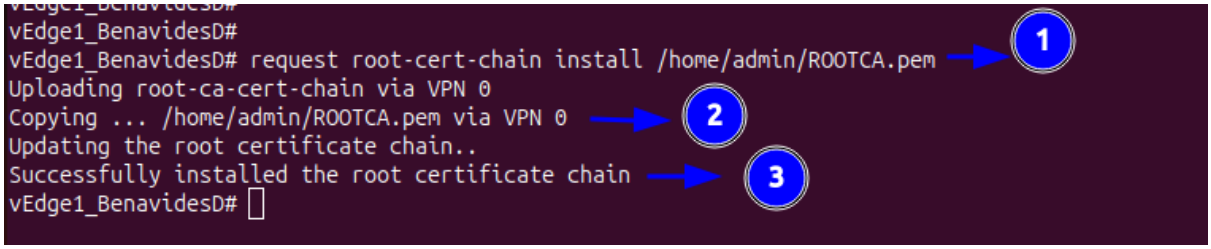



Como se observa en la **¡Error! No se encuentra el origen de la referencia.**, una vez d entro de cada vEdge a través de la consola se ejecuta el comando ls -lh para enlistar los archivos de configuración, ahora mediante una comunicación ssh se solicita mediante el comando scp copiar el archivo ROOTCA.pem desde la entidad certificadora en este caso el servidor ROOTCA, y de esta manera en cada vEdge se replica este paso para contar con el certificado necesario para registrar cada uno de los dispositivos en la red SD-WAN.

2. Una vez cargado el certificado en cada dispositivo vEdge se procede a instalarlo dentro del mismo.

**Figura 37**

*Instalación del certificado en cada vEdge*

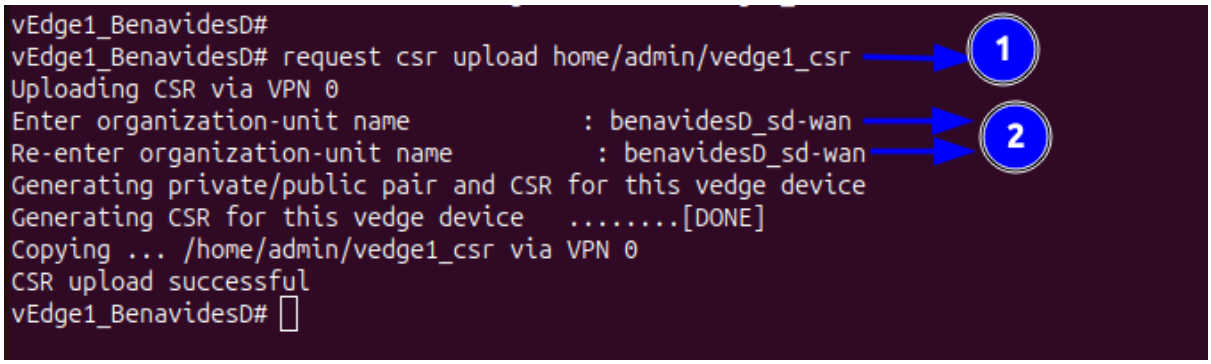



En la **¡Error! No se encuentra el origen de la referencia.**, se observa que la i nstalación es sencilla y requiere de la ejecución del comando **request root-cert-chain install**, posteriormente se copia y se carga el archivo vía VPN 0, finalmente se observa un mensaje de que la instalación fue exitosa.

3. Si bien el archivo CSR en los controladores vManage, vBond y vSmart se creaba y descargaba desde la sección controladores dentro del panel de control, en los vEdge se genera el CSR desde la propia consola mediante línea de código.

**Figura 38**

*Creación del archivo CSR en el vEdge*



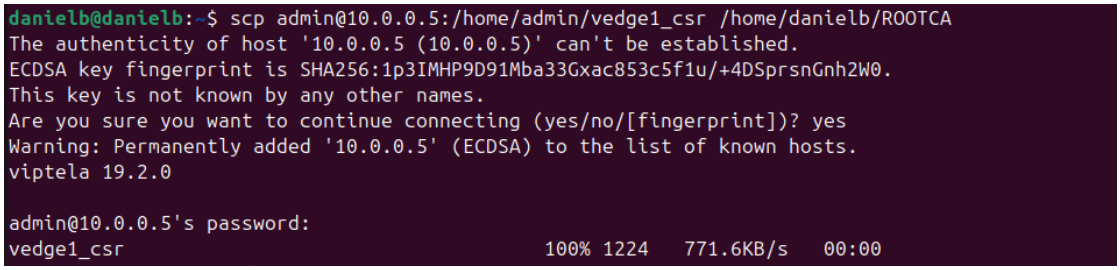

En la **¡Error! No se encuentra el origen de la referencia.**, se muestra cómo crear el a rchivo CSR a través de la consola del vEdge, este paso es muy importante debido a que al

momento de crear el archivo se solicita el ingreso y verificación del nombre de la organización, dato muy importante para que cada vEdge sea reconocido por la red, y de esta manera el archivo CSR se crea para posteriormente ser firmado.

4. Al igual que en los controladores es necesario firmar el archivo CSR generado en cada vEdge, con lo cual se requiere copiar el archivo en este caso **vedge1_csr** en el servidor ROOTCA.

**Figura 39**

*Copia del archivo CSR en el servidor ROOTCA.*

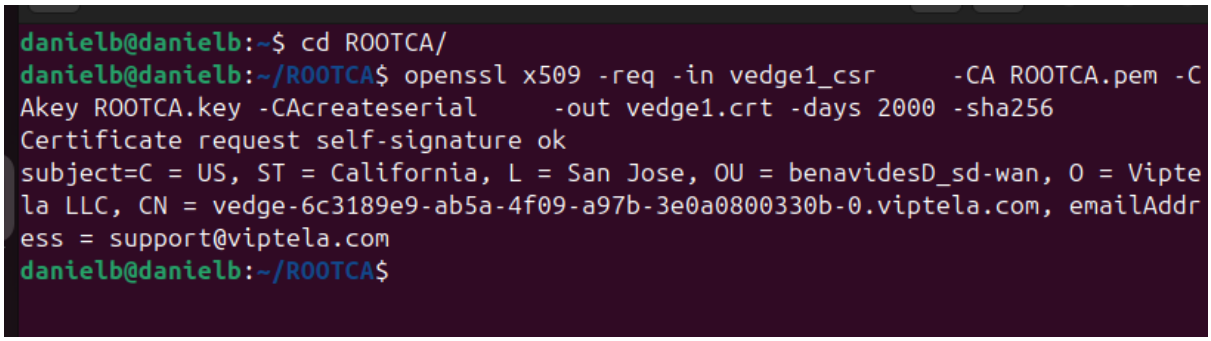



En la **¡Error! No se encuentra el origen de la referencia.**, se muestra como desde el s ervidor ROOTCA y a través de SSH se copia el archivo CSR generado en el vEdge de manera exitosa mediante consola de comandos, proceso que se replicara para cada uno de vEdge que formen parte de la red SD-WAN.

5. En el servidor ROOTCA, se precede a firmar el archivo CSR y generar el certificado CRT, importante para lograr autenticar los dispositivos vEdge en el panel de control del vManage.

**Figura 40**

*Firma y generación del certificado CRT*



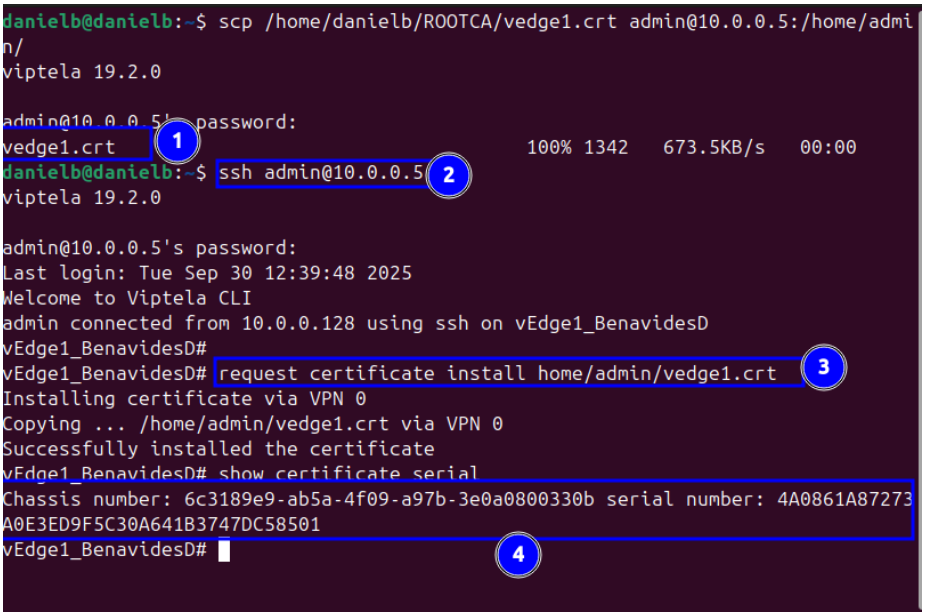

La firma del certificado CRT requiere de elementos clave que se encuentran ya cargados dentro del servidor ROOTCA, como la llave KEY y el archivo ROOTCA.pem, archivos con los cuales se firman todos los dispositivos de la solución Viptela para lograr la sincronía y seguridad de la red. En la **¡Error! No se encuentra el origen de la referencia.**, se o bserva la línea de comando con la cual se obtiene el certificado CRT, y en la cual se agregan ciertos parámetros importantes como son:

- El tiempo de duración del certificado en días: -days 2000
- El cifrado usado: -sha256
- El nombre de la organización: OU= benavidesD_sd-wan (importante).

6. Con el certificado CRT generado para cada vEdge, es necesario copiarlo a cada dispositivo, con el fin de instalar el certificado en cada uno de ellos, debido a que dicho certificado contiene el número de chasis y el número de serie indispensable para registrar y habilitar cada uno de los end point.

**Figura 41**

*Montaje e instalación del certificado CRT en el vEdge*

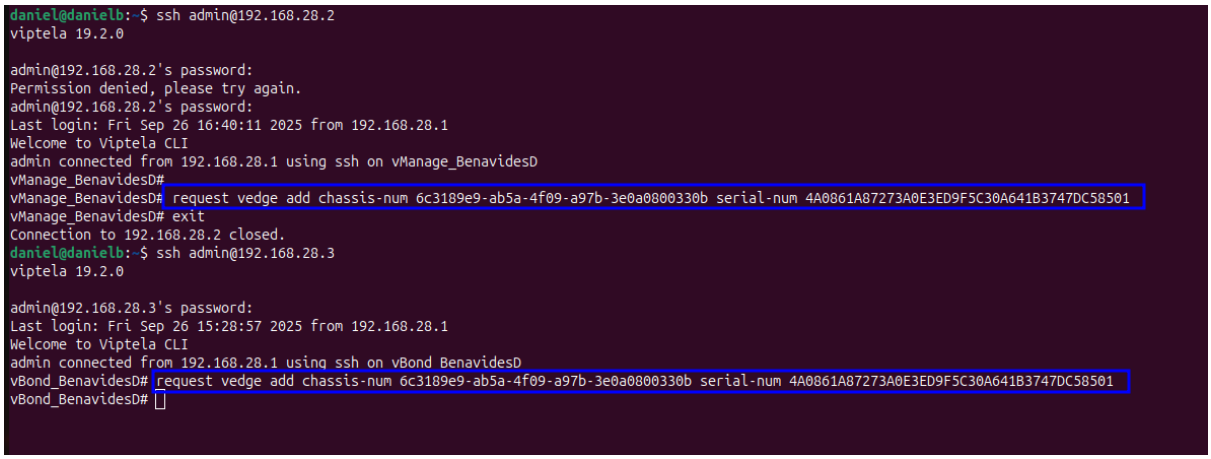



En este caso mediante la comunicación SSH se copia el archivo CRT desde el servidor certificador ROOTCA hacia cada uno de los vEdge, y posteriormente se conecta al dispositivo mediante la línea de comando **ssh** **admin@10.0.0.5** esto con el objetivo de ingresar a las configuraciones de cada vEdge para habilitar el certificado mediante el comando **request certificate install** como se observa en la **¡Error! No se encuentra el origen de la referencia.**. U na vez instalado el certificado mediante el comando **show certificate serial** se logra obtener los datos de numero de chasis y serial que permitirán habilitar el vEdge en el panel de control de vManage.

7. A pesar de tener instalado los certificados en el vEdge, es necesario realizar una configuración adicional mediante la línea de comando en este caso a través de SSH desde el servidor ROOTCA se ingresa al vManage y al vBond a través de la dirección IP de administración configuradas previamente en las interfaces asignadas para lograr cargar el número de chasis y el numero serial del vEdge.

**Figura 42**

*Registro de vEdge en vManage y vBond*

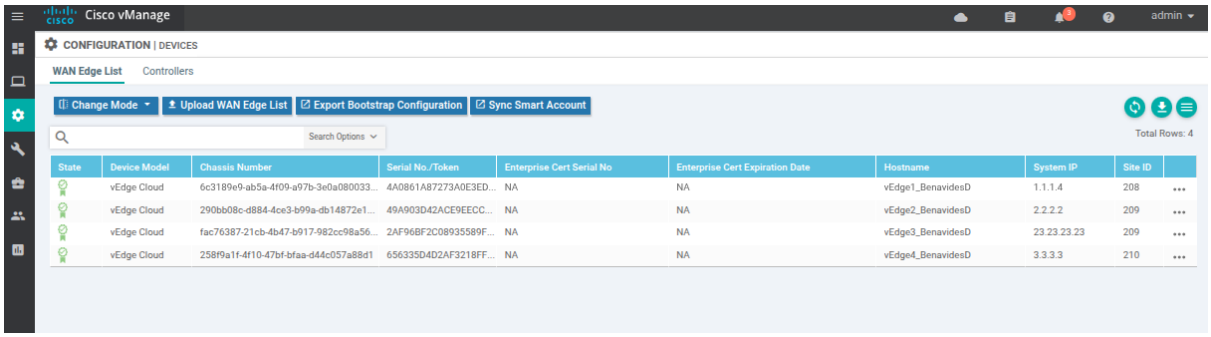



Como se observa en la **¡Error! No se encuentra el origen de la referencia.**, los datos d el vEdge que se obtienen instalando el certificado se agregan en el vManage y el vBond respectivamente, mediante el comando **request vedge add chassis-num "numero" serial-num "numero2"**, con el objetivo de registrar, autenticar y habilitar el vEdge dentro de la red, y que posteriormente se visualizara dentro del panel de control del vManage.

Una vez completada toda la configuración previamente detallada, se observa en la lista de dispositivos vEdge dentro del panel de controladoras que el dispositivo ya se encuentra habilitado dentro de la red SD-WAN, este proceso es necesario replicarlo para cada uno de los vEdges.

**Figura 43**

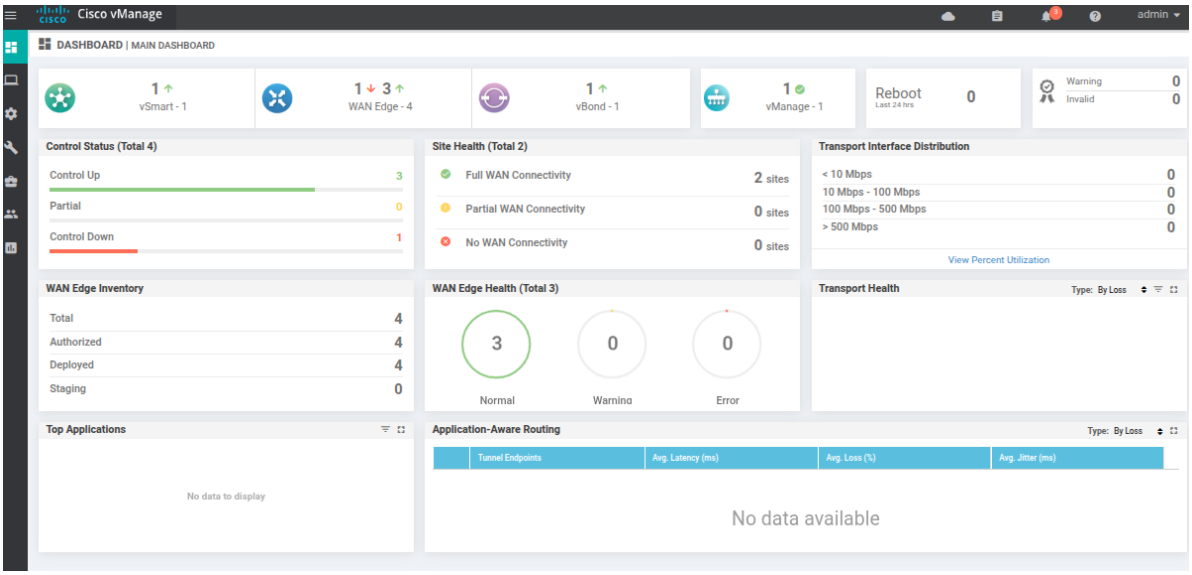
*Dispositivos vEdge cargados en vManage*



En la **¡Error! No se encuentra el origen de la referencia.**, se observa como los d ispositivos vEdge ya se encuentra habilitados en el vManage, y conforme se registra cada vEdge la lista de dispositivos se va actualizando hasta registrar cada uno de los dispositivos que formaran parte de la red.

**Figura 44**

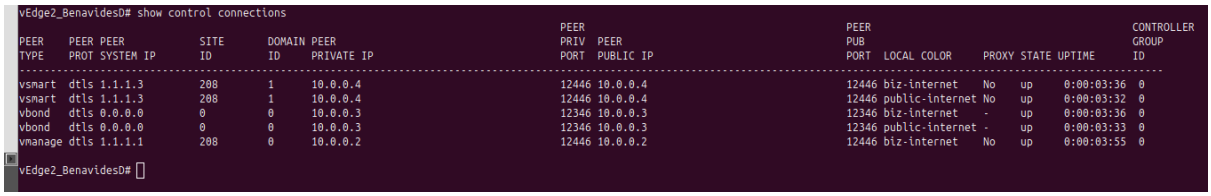
*Visualización desde el panel de control principal.*




Una vez cargado cada uno de los vEdges y verificado que ya se integraron al vManage, al momento de ingresar a la interfaz del vManage mediante la dirección https://192.168.28.2:8443 como se muestra en la **¡Error! No se encuentra el origen de la r eferencia.**, se logra observar que el plano de control ya ha detectado el dispositivo ingresado y el estado de este ha pasado de 0 a 1, lo que permite que el panel de control comience a monitorearlo de una manera segura mediante el protocolo DTLS, al momento de ingresar nuevos dispositivos vEdge el marcado del estado seguirá incrementando, y cuando algún dispositivo vEdge este en estado inactivo se mostrará con una flecha roja y cuando este activo con una flecha verde.

**Figura 45**

*Conexiones del vEdge*



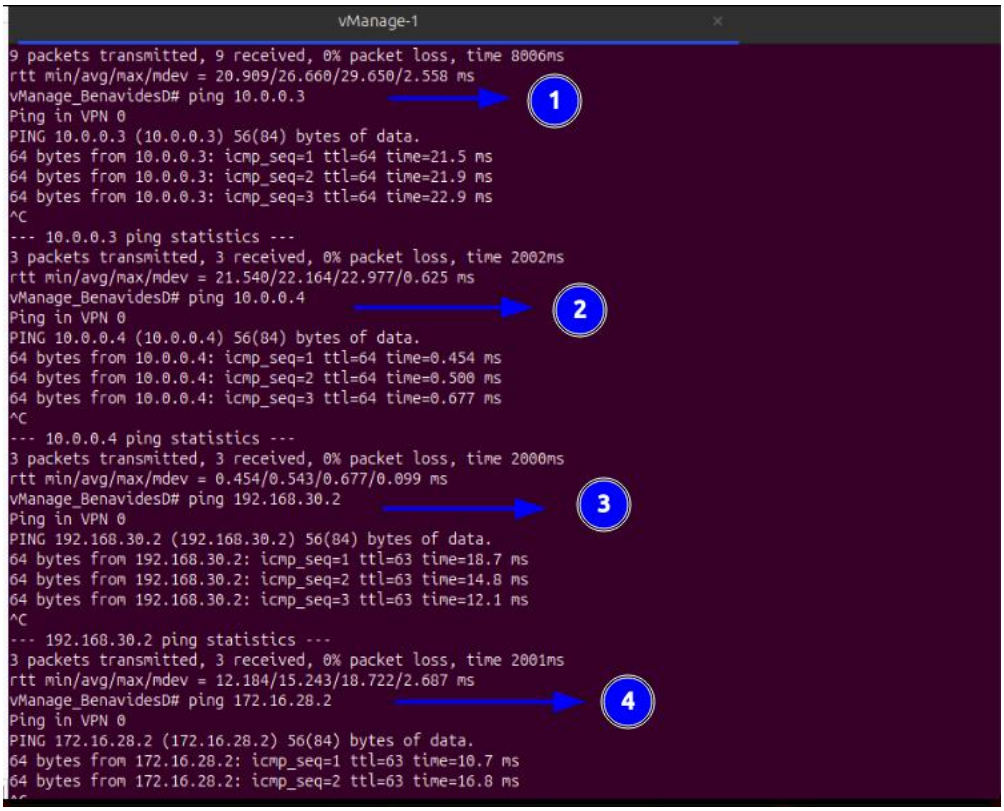

En la **¡Error! No se encuentra el origen de la referencia.**, se realizó la verificación d e conectividad del panel de control en el **vEdg2_BenavidesD**, mediante la salida del comando **show control connections**, donde se logra visualizar que el dispositivo ha establecido túneles DTLS consistentes en dirección a los controladores vManage, vBond y vSmart. Se logró identificar en la columna **state** que se encuentran en "up" a través de los colores public-internet y biz-internet lo que demuestra una capacidad de transporte multiple (multi-homing) del dispositivo, característica que también estará presente en los vEdge de los otros sitios, lo que asegura redundancia necesaria para la gestión de la red SD-WAN.

### *3.3.1. Validación Inicial de Conectividad (Pruebas ICMP)*

Una vez configurados y registrados cada uno de los dispositivos que conforman la red SD-WAN, se procede a realizar pruebas de conectividad entre las controladoras y los vEdges.

**Figura 46**

*Conectividad entre dispositivos de la red SD-WAN*




Para validar que los controladores de la red SD-WAN implementada se encuentren operando de manera adecuada, se generó tráfico ICMP a través de la VPN de transporte 0 desde el vManage hacia cada uno de los controladores como se puede observar en la Figura 46, a continuación, se detalla cada uno de los tráficos generados y los elementos involucrados:

1. Tráfico ICMP generado entre el vManage (10.0.0.2) y el Vbond (10.0.0.3) con respuesta exitosa.
2. Tráfico ICMP generado entre el vManage y el vSmart (10.0.0.4) con respuesta exitosa.
3. Tráfico ICMP generado entre el vManage y el vEdge 2 (192.168.30.2) a través de la Red de Transporte 1, con respuesta exitosa.
4. Tráfico ICMP generado entre el vManage y el vEdge 2 (172.16.28.2) a través de la Red de Transporte 2, con respuesta exitosa.

Este proceso se realizó entre el vManage y cada uno de los vEdge que conforman la red para validar la conectividad entre sitios y entre controladores.

Una vez validada la convergencia de la red SD-WAN es posible configurar dispositivos finales como VPCS para generar tráfico ICMP a través de una VPN numerada de 1-511 que son consideradas como VPN de servicios.

**Figura 47**

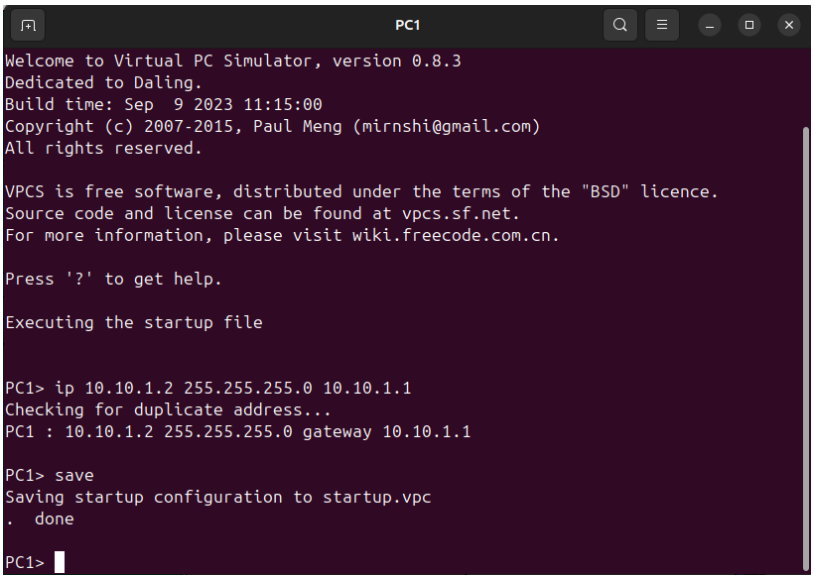
*Configuración de VPN 1 en el vEdge.*

```
vEdge1_BenavidesD# config
Entering configuration mode terminal
vEdge1_BenavidesD(config)# vpn 1
vEdge1_BenavidesD(config-vpn-1)# interface ge0/2
vEdge1_BenavidesD(config-interface-ge0/2)# no ip address 10.10.5.1/24
vEdge1_BenavidesD(config-interface-ge0/2)# ip address 10.10.1.1/24
vEdge1_BenavidesD(config-interface-ge0/2)# no shutdown
vEdge1_BenavidesD(config-interface-ge0/2)# exit
vEdge1_BenavidesD(config-vpn-1)# commit check
Validation complete
vEdge1_BenavidesD(config-vpn-1)# commit and-quit
Commit complete.
```

En la Figura 47 se logra apreciar la configuración del a VPN 1 de servicio en la interfaz de cada uno de los vEdge respectivamente con el direccionamiento presentado en la Tabla 10 a través de la cual se pueden comunicar los usuarios conectados a cada uno de los vEdges en cada uno de los sitios.

**Figura 48**

*Configuración de VPCs para cada sitio.*

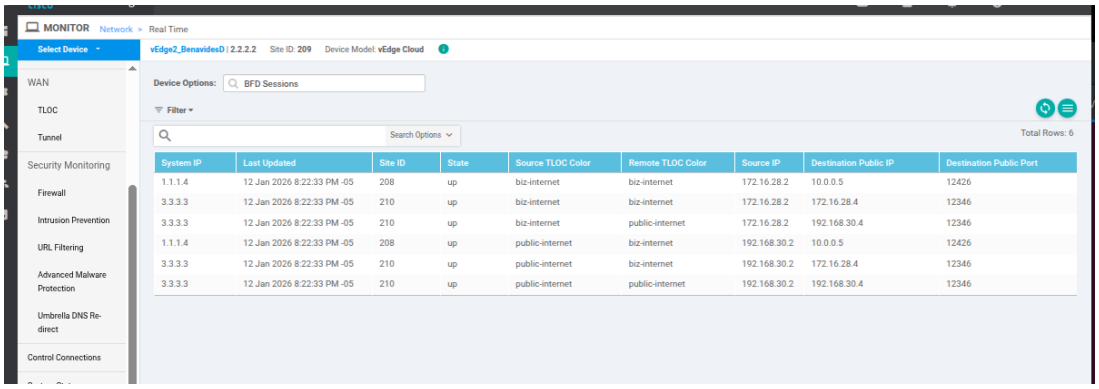



En la Figura 48, se puede observar la configuración de VPCS para el presente proyecto desde los cuales se enviará tráfico ICMP mediante la VPN 1, con lo cual se comunicarán cada uno de los dispositivos finales configurados en cada sitio, este tráfico permitirá analizar algunos aspectos de la red SD-WAN a través del panel de control del vManage y posteriormente mediante código Python a través de la API de Cisco, la configuración de las VPCS restantes se realiza conforme los datos presentados en la Tabla 10 de direccionamiento.

A continuación, se presentan algunos estados operativos dentro de la arquitectura SD-WAN implementada, con lo cual se verificará la actuación de la red con el fin de interconectar distintos sitios. A través de las herramientas de monitoreo presentes en el panel de control del vManage se identifican sesiones BFD, intercambio de rutas OMP y la conectividad ICMP, permitiendo de esta manera evidenciar la comunicación adecuada entre los dispositivos vEdge en diferentes sitios, mostrando la operatividad de la red con el objetivo de tener una administración eficiente.

**Figura 49**

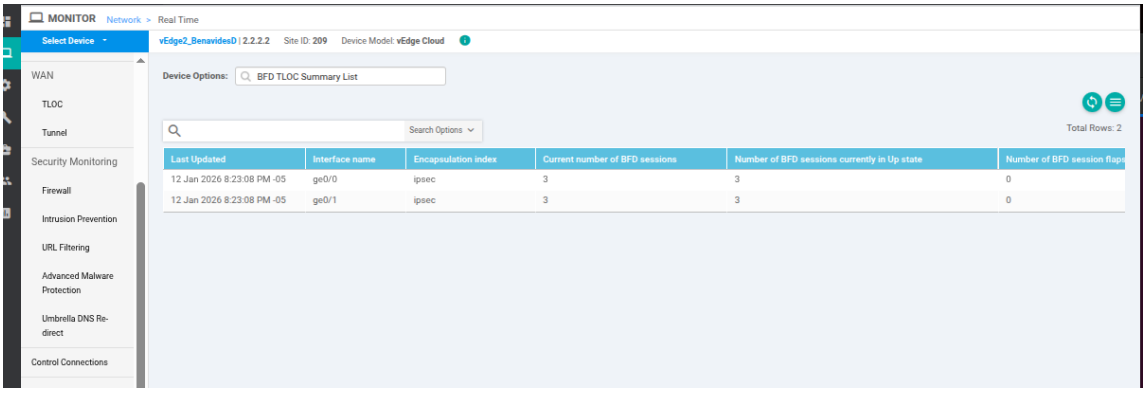
*Estado de sesiones BFD*




En la Figura 49, se observa una tabla en la cual se segmenta a detalle cada una de las sesiones BFD, se logra visualizar seis sesiones activas que van dirigidas a los sitios 208 y 2010 (identificados por sus IPs de sistema 1.1.1.4 y 3.3.3.3), estos datos muestran con quien exactamente se está comunicando el router, adicional muestra a detalle los colores de los TLOCS (biz-internet y public-internet) junto con las IP públicas de origen y destino y el estado en que se encuentra cada una de las sesiones lo que confirma que la SD-WAN este operando correctamente a través de múltiples rutas.

**Figura 50**

*Resumen de TLOCs y BFD*

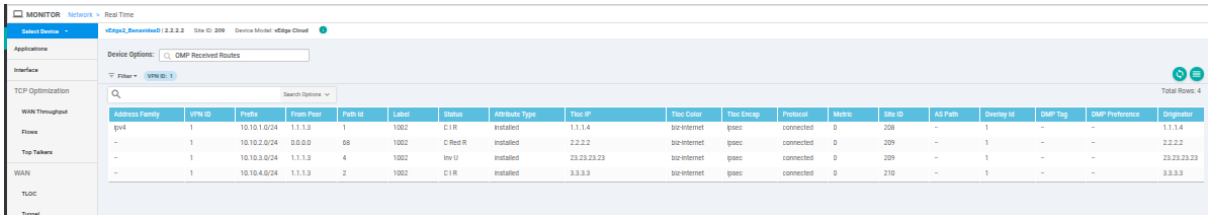



Continuando con la comprobación de conectividad dentro del panel de control del vManage es posible dar una visión rápida de la salud de las redes de transporte (WAN) levantadas, en la Figura 50 se logra visualizar una tabla donde se presentan dos interfaces

físicas usando encapsulación IPsec (ge0/0 y la ge0/1), cada interfaz marca 3 sesiones BFD activas y en estado “Up” con cero números de caídas, indicando que la conectividad física y de los túneles básicos hacia otros sitios se encuentran estables.

**Figura 51**

*Rutas OMP*

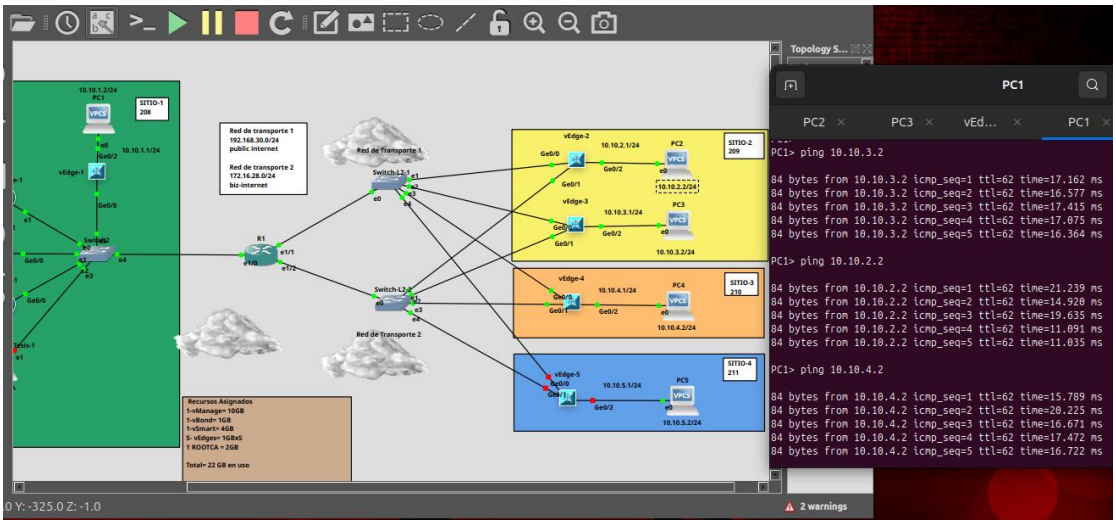



Dentro de la comunicación en una red SD-WAN el protocolo OMP es considerado el cerebro que distribuye la información de enrutamiento, en la **¡Error! No se encuentra el o rigen de la referencia.** se observa la opción de **OMP Received Routes** específicamente para la VPN 1 de servicio configurada en cada vEdge de cada sitio, se aprecia como el vEdge ha aprendido rutas para 4 redes distintas, para cada ruta se especifica el TLOC de salida (IP y color de transporte), lo que confirma  que el estado es de “Installed”, y por ende demuestra que todas estas rutas se encuentran habilitadas en la tabla de enrutamiento del dispositivo y prestas a enviar tráfico hacia otros sitios de la red SD-WAN.

Una vez comprobada la conectividad entre controladores (vManage, vBond y vSmart) y los vEdges , es posible generar tráfico ICMP desde los dispositivos finales VPCs conectados en cada uno de los vEdges, esto permitirá verificar la comunicación entre sitios a través de la VPN 1 de servicio configurada para el presente trabajo de grado.

**Figura 52**

*Prueba de conectividad ICMP entre sitios.*




En la Figura 52, se observa envió de tráfico ICMP desde la VPCs 1 conectada al vEdge-1 en el sitio 208 con dirección a las VPCs de los distintos sitios establecidos en la red SD-WAN con respuesta positiva lo que confirma que la conectividad entre sitios fue exitosa y por ende existe comunicación estable.

## 3.4. Desarrollo de una Solución de Automatización Mediante el Uso de la API de vManage.

En este apartado, se describe el proceso de automatización implementada en el lenguaje de programación Python, orientado a mejorar la administración de la red Cisco SD-WAN (Viptela) implementada en el presente trabajo de grado, la propuesta se basa en el consumo de las APIs REST proporcionada por el controlador vManage que forma parte del plano de gestión de la arquitectura SD-WAN y el cual permite la interacción de forma centralizada con los distintos elementos que conforman la red.

La solución desarrollada permite automatizar y gestionar diversas tareas, como el monitoreo del estado de los equipos, la obtención de información operativa y la verificación de parámetros de funcionamiento, entre otros que por lo general son realizadas de forma manual por el administrador de la red a través de la interfaz de control del vManage, para ello, desde

los scripts de Python se establece una comunicación al vManage a través de las APIs facilitando la gestión centralizada y reduciendo la dependencia de configuraciones manuales y aprovechando la integración con los controladores vSmart y vBond, responsables del plano de control y de la orquestación dentro de la arquitectura Cisco SD-WAN.

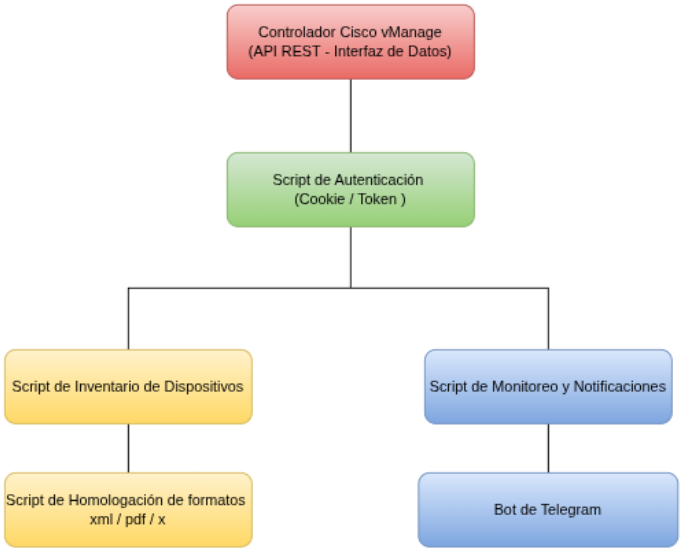
En base a este enfoque, se evidencia como el uso de APIs y la automatización basada en software aportan a tener una administración más eficiente de la red y alineándose con los principios de paradigma Software Defined Networking (SDN) y demostrando aplicación práctica en un entorno de emulación orientado al desarrollo del presente trabajo de grado.

### *3.4.1. Diseño de la Solución de Automatización y Diagramas de Flujo.*

El desarrollo de la solución de automatización en el presente trabajo de grado se fundamenta en una arquitectura basada en servicios independientes permitiendo una gestión segregada de los recursos y procesos, lo que podría favorecer el mantenimiento del sistema y su escalabilidad debido a que cada componente puede ser modificado o reestructurado sin afectar el funcionamiento de la solución general.

**Figura 53**

*Diagrama de bloques de la capa de automatización del sistema*



En la **¡Error! No se encuentra el origen de la referencia.**, se presenta el diagrama de b loques de la capa de automatización de la solución propuesta, el cual describe la estructura lógica desarrollada para la administración eficiente de la red SD-WAN, esta capa es el punto central de la propuesta debido a que se encarga de la interacción con el controlador Cisco vManage mediante el consumo de servicios API REST. La comunicación parte desde el script de autenticación que permite el acceso de manera segura a los recursos brindados por el controlador, partiendo de este script se habilita la ejecución de los scripts funcionales de inventario de dispositivos y monitoreo de túneles, encargados de la obtención y supervisión del estado operativos de la red propuesta. La información obtenida a través de estos elementos es usada para generar archivos en formato JSON, XML, PDF para el caso de inventarios, para el caso de monitoreo establece notificaciones automáticas mediante un Bot en la plataforma de Telegram, ofreciendo una contribución más eficiente y centralizada con respecto a la gestión de la red desde la capa de automatización.

Cada uno de los scripts desarrollados en Python e implementados en la red SD-WAN, corresponde a un módulo de automatización que está orientado a una función operativa especifica, entre las cuales se encontrarían: autenticación con la API de vManage, recopilación de información de los dispositivos, generación de reportes de auditoría y envió de notificaciones en tiempo real. Esta estructura lo que busca es descartar puntos únicos de fallo, debido a que la ejecución de cada script no dependería directamente de los demás. De igual forma, cada script gestiona su propio tiempo de duración incluyendo la autenticación frente a la API de vManage y el manejo de sesiones necesarias para una comunicación adecuada con el controlador.

Esta división de funcionalidades permite mejorar los tiempos de respuesta del sistema y simplifica tareas de depuración, debido a que los errores pueden ser identificados y corregidos de manera rápida, característica de la automatización.

En el contexto de facilitar la comprensión de la operatividad de la solución propuesta en el presente trabajo de grado, se han diseñado diagramas de flujo que describen de manera coherente la lógica de ejecución para cada uno de los scripts de automatización instaurados. A continuación, se detalla cada uno de ellos:

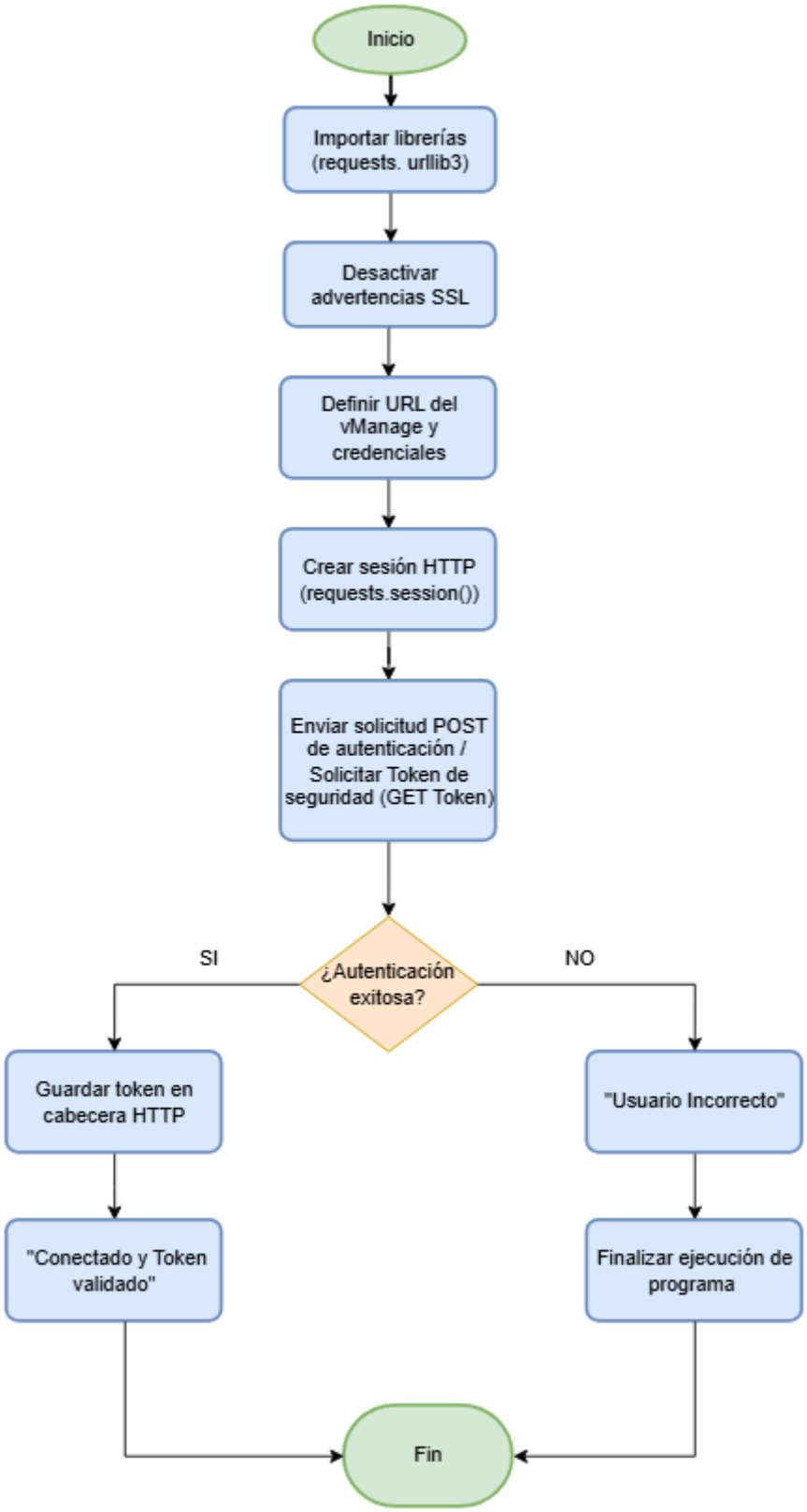
#### 3.4.1.3. Diagrama de Flujo de Autenticación.

El diagrama de la **¡Error! No se encuentra el origen de la referencia.** corresponde al s cript de autenticación, el cual detalla la secuencia de pasos necesarias para establecer la comunicación inicial con el controlador Cisco vManage y habilitar el consumo de los servicios expuestos por su API REST, los cuales son requeridos por script posteriores de la solución.

El proceso parte de la importación de las librerías necesarias para la ejecución del script de Python, a continuación, crea una sesión HTTP segura seguida del envió de las credenciales de acceso al controlador vManage mediante el consumo de la API REST, el sistema valida la respuesta del servidor, en caso de que la autenticación sea exitosa obtiene un token de seguridad del tipo XSRF que es incorporado en la cabecera de las siguientes peticiones API que realizaran los siguientes scripts. Este proceso permite asegura un acceso seguro y controlado a los servicios de administración de la red SD-WAN a través de los scripts de Python, en caso de que la autenticación fallase el script termina su ejecución bloqueada el acceso no autorizado y asegurando de esta manera la integridad del sistema.

**Figura 54**

*Diagrama de Flujo para el Script de Autenticación*

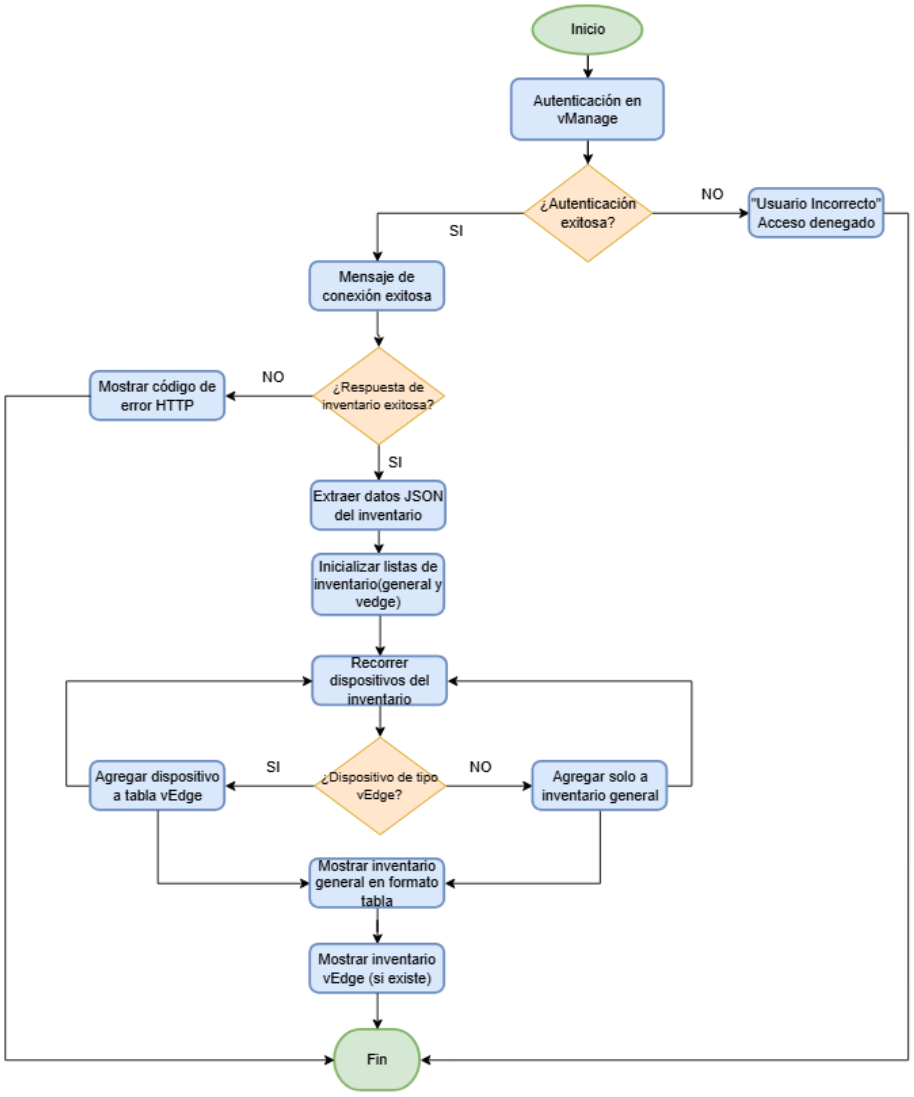


### 3.4.1.4. Diagrama de Flujo de Inventario de Dispositivos y Homologación de Formatos.

En este apartado se presenta dos diagramas de flujo el primero del script de inventario de dispositivos el cual está orientado a la consulta, procesamiento, organización de información relevante en la gestión y auditoria de la red SD-WAN **¡Error! No se encuentra el origen de l a referencia.**, y el segundo diagrama es un apartado para la homologación de formatos lo cual facilita la obtención de reportes en formatos adicionales a JSON tradicional en este tipo de comunicación, entre los formatos adicionales se encuentran xml, pdf y xlsx como se observa en la **¡Error! No se encuentra el origen de la referencia.**. Sin embargo, para una compresión a decuada se lo ha dividido en dos scripts.

**Figura 55**

*Diagrama de Flujo Inventario de Dispositivos*




La **¡Error! No se encuentra el origen de la referencia.**, presenta el diagrama de flujo d el proceso que desarrolla el sistema automatizado para la gestión del inventario de dispositivos SD-WAN, como se logra observar el proceso comienza con la autenticación segura en el controlador vManage, usando las credenciales de acceso y el token XSRF. Establecida y validad la sesión, el sistema realiza una consulta al inventario de dispositivos a través de la API REST, para posteriormente procesar la información obtenida y clasifica los equipos en dos categorías: la de inventario general y la de inventario específico de dispositivos vEdge para

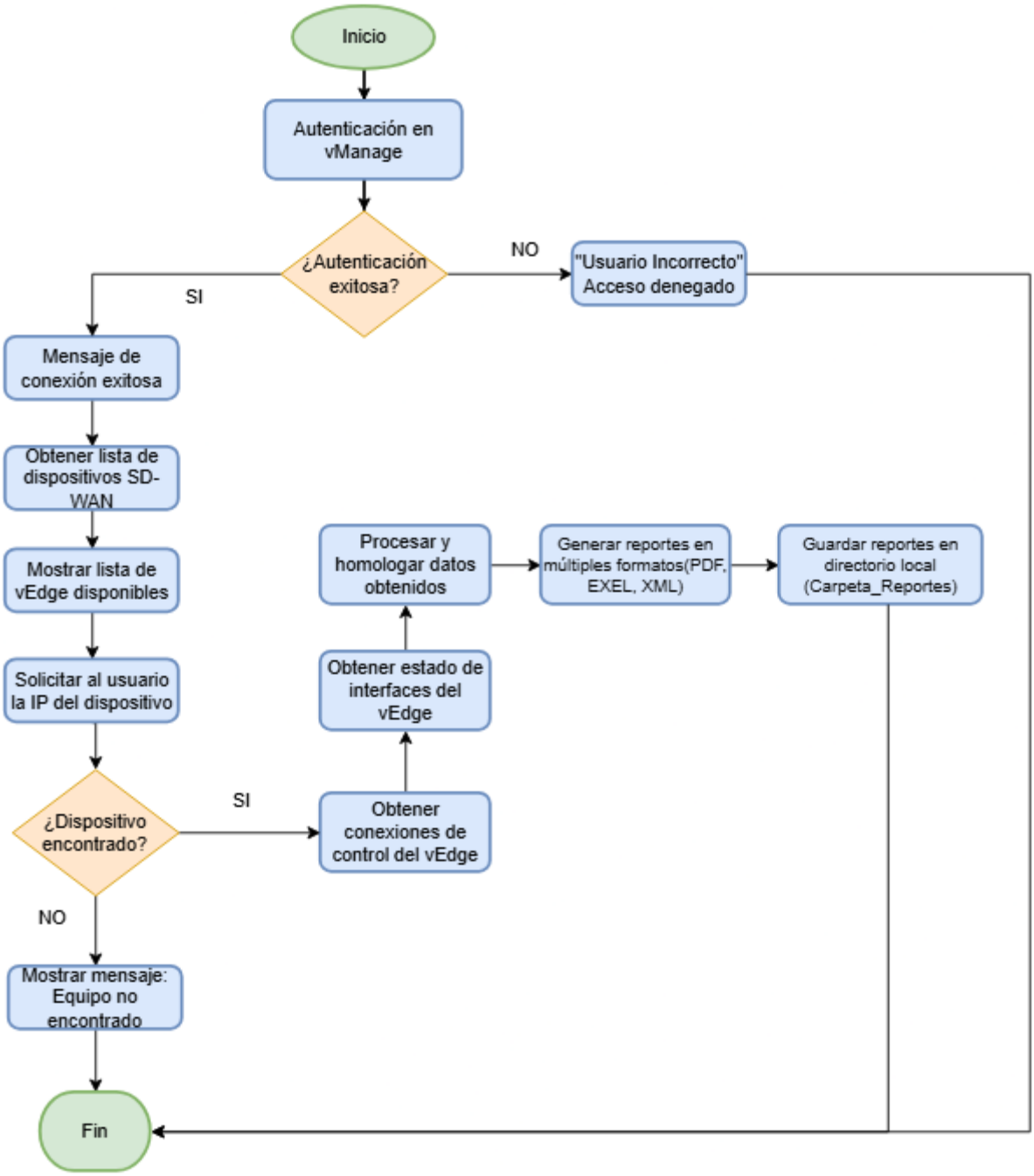
finalmente presentar los resultados en un formato tabular que facilita su análisis y posterior interpretación.

Cabe especificar que el diagrama de flujo presentado enfatiza exclusivamente las etapas de obtención y procesamiento del inventario de dispositivos, excluyendo el proceso de autenticación debido a que estos ya fueron abordados y documentados anteriormente.

Continuando en este apartado **¡Error! No se encuentra el origen de la referencia.** m uestra un segundo diagrama de flujo que representa el script con la sección de homologación y exportación de información de los vEdges, este proceso parte con una sesión previamente autenticada en el controlador vManage para posteriormente obtener y filtrar los dispositivos de tipo vEdge, concediendo al usuario la facultad de elegir un dispositivo específico para su análisis. Una vez seleccionado el vEdge se recopilan datos de conexiones de control y estados de las interfaces, dicha información es procesada y homologada para finalmente ser presentada en formatos XML, XLSX Y PDF respectivamente, facilitando el análisis e interoperabilidad de los datos obtenidos.

**Figura 56**

*Diagrama de Flujo-Homologación de Formatos*



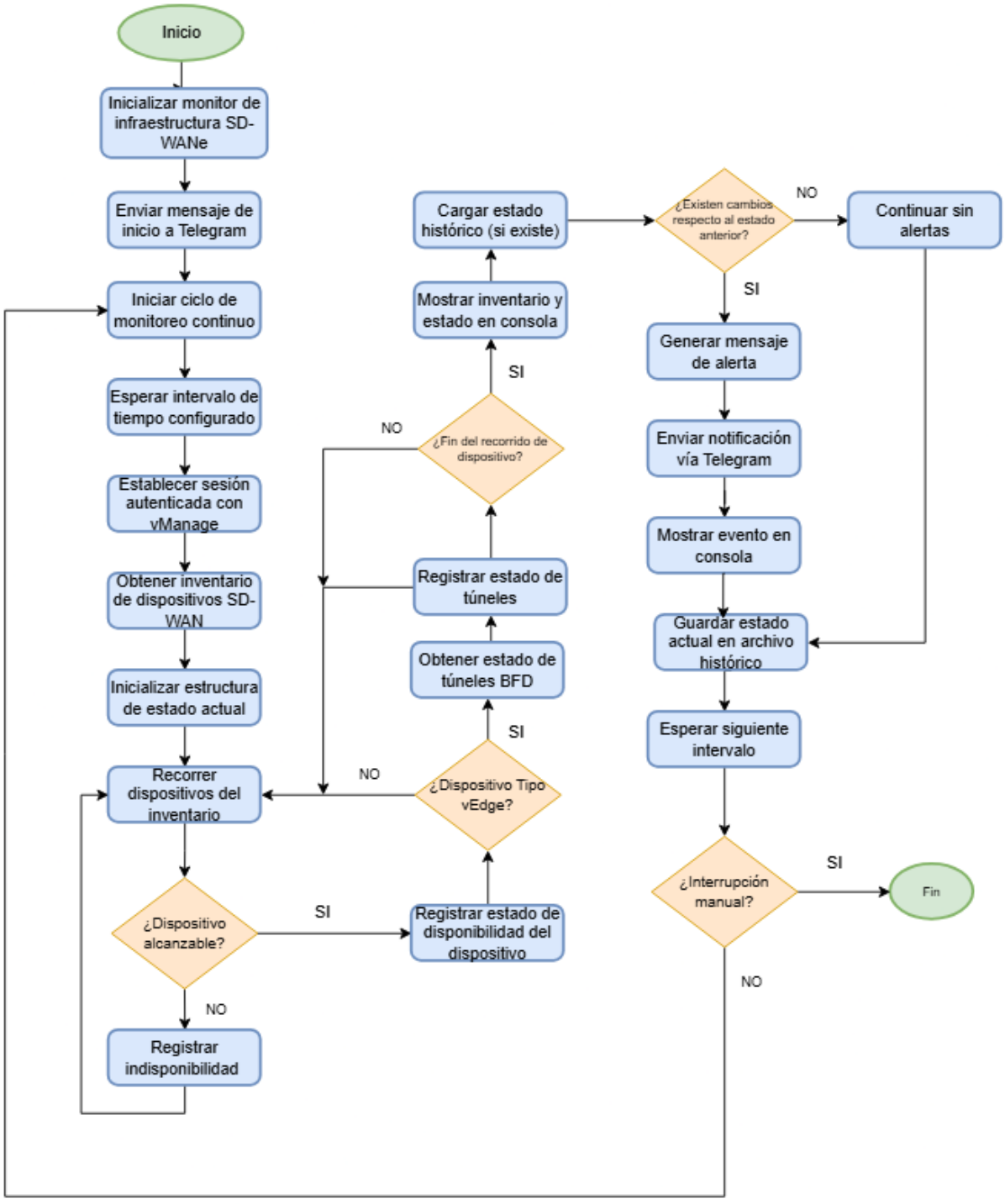

### 3.4.1.5. Diagrama de Flujo de Monitoreo y Notificaciones.

El diagrama de flujo de monitoreo y notificaciones de la **¡Error! No se encuentra el origen de la referencia.** describe el funcionamiento del script de monitoreo de dispositivos y

túneles BFD, y establece un flujo de ejecución cíclico que admite la revisión continua del estado de los dispositivos y de los túneles BFD generando notificaciones automáticas ante eventos relevantes.

**Figura 57**

*Diagrama de flujo de Monitoreo y Notificaciones*



En la **¡Error! No se encuentra el origen de la referencia.**, se describe a priori el f uncionamiento general del sistema de automatizado de monitoreo de la red SD-WAN, el proceso parte con la inicialización del monitor y envió de una notificación a Telegram indicando que el servicio o comunicación se encuentra activo y el sistema entra en un ciclo continuo de ejecución, en el cual espera un cierto tiempo configurado antes de inicia con la recolección de la información.

Posterior al tiempo de espera, el sistema establece una sesión autenticada con el vMamage y obtiene el inventario de los dispositivos de la red SD-WAN para posteriormente verificar que son alcanzable, en caso de identificar un equipo vEdge hace una pausa para analizar el estado de los túneles BFD. Completado el recorrido y verificados los dispositivos el sistema ejecuta un análisis comparativo de los estados actual y anterior para identificar variaciones o cambios relevantes, si se encuentra alguna variación el sistema genera una alerta y procede a enviar una notificación inmediata por Telegram. Finalmente, el estado actual reemplaza al estado anterior para una posterior comparación debido a que el sistema retoma el periodo de espera, sosteniendo un monitoreo continuo y autónomo que puede ser finalizado solo de manera manual por el administrador de la red.

Cabe acotar que cada uno de estos diagramas de flujo sirven como una guía conceptual para posteriormente ser implementada en el lenguaje Python como parte fundamental de la solución planteada.

### *3.4.2 Entorno de Desarrollo y Librerías*

En esta sección se describe el entorno de desarrollo usado en la implementación de la solución de automatización para la red SD-WAN en base a la descripción de las principales librerías utilizadas durante el desarrollo de los scripts en lenguaje Python, el emplear un entorno controlado y con librerías especificas ayuda a garantizar la ejecución correcta de los procesos

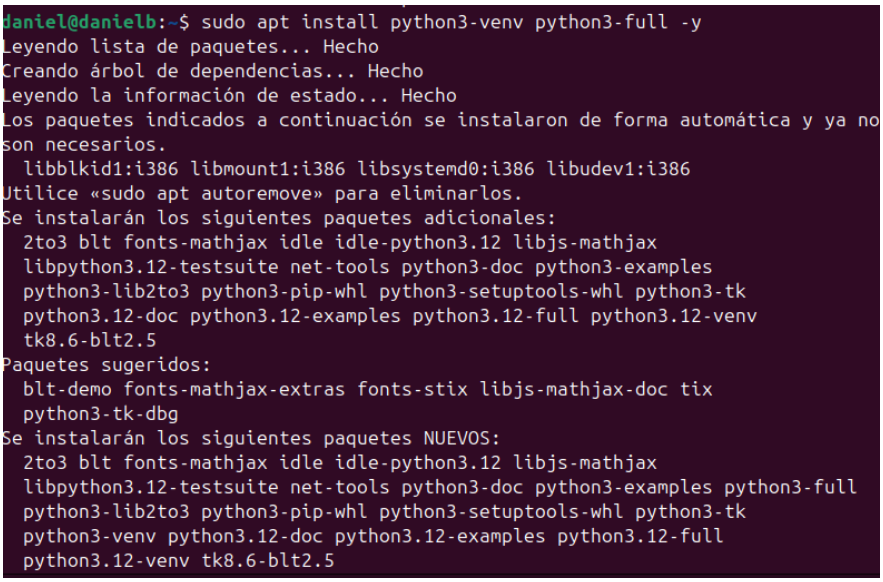
de autenticación, inventario, homologación de formatos y monitoreo, así como la interacción adecuada con la API REST del controlador vManage.

Adicional, se detallan dependencias de software necesarias en la ejecución de la solución al presentar algunas funciones específicas de cada librería dentro del sistema, para finalmente presentar una estructura de carpetas y archivos que forman parte del proyecto, con el objetivo de evidenciar la organización de los scripts y de esta manera facilitar la comprensión del diseño para la solución implementada para futuras replicas en otros entornos de laboratorio o producción.

La solución fue desarrollada usando el lenguaje de programación Python en su versión 3.12 en un entorno de laboratorio basado en sistema Linux, sin embargo, fue necesaria la instalación y uso de un entorno virtual Python como se observa en la Figura 58, con el fin de permitir la instalación de librerías requeridas y evitar conflictos entre las versiones manejadas por el sistema operativo lo que permitió aislar el proyecto y evitar posibles problemas de ejecución, los cuales se presentaron inicialmente al intentar ejecutar determinadas librerías.

**Figura 58**

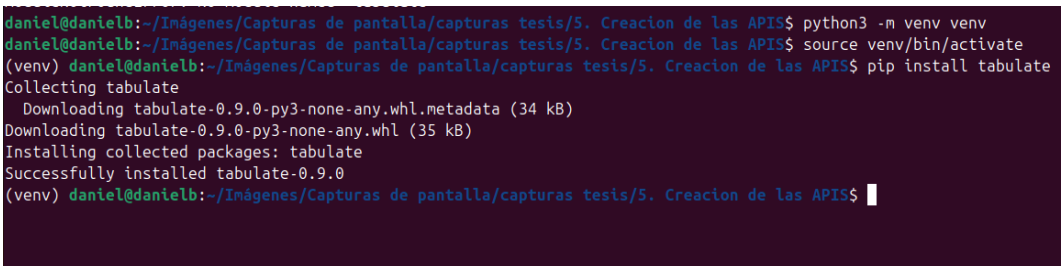
*Instalación de entorno virtual en Python*



Una vez instalado el entorno virtual se crea una carpeta mediante el comando **python3 -m venv venv** en la cual almacenan todas las configuraciones realizadas, finamente se ejecuta el entorno mediante la línea de comandos **source venv/bin/actívate,** como se observa en la Figura 59, con lo cual el entrono queda establecido para la ejecución de los distintos scripts, es importante tomar cuenta que no se debe cambiar el nombre de la carpeta donde se encuentran los scripts, debido a que el entono no reconocerá dicha carpeta y generara un error en la ejecución.

**Figura 59**

*Despliegue de entorno virtual en Python*




Con el objetivo de detallar las dependencias o librerías de software utilizadas en la implementación de los scripts de automatización en el desarrollo de la solución, la **Tabla *12*** resume las principales.

**Tabla 12**

*Principales librerías utilizadas en el desarrollo de la solución*

| Librería | Versión | Tipo | Función dentro de la solución |
|---|---|---|---|
| requests | 2.32.5 | Externa | Consumo de la API REST del controlador Cisco vManage mediante solicitudes HTTP |
| urllib3 | 2.6.3 | Externa | Gestión de conexiones HTTP seguras y manejo de certificados SSL |
| httpx | 0.28.1 | Externa | Soporte para solicitudes HTTP avanzadas y comunicación eficiente |

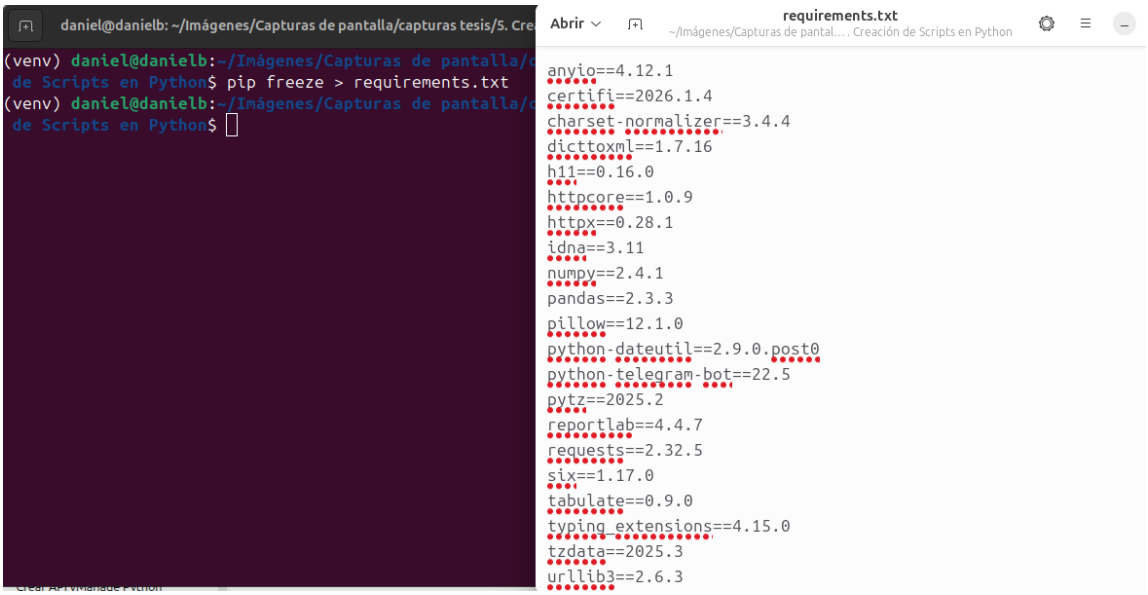
| Librería | Versión | Tipo | Función dentro de la solución |
|---|---|---|---|
| python-telegram-bot | 22.5 | Externa | Envío de notificaciones automáticas sobre el estado de la red SD-WAN |
| pandas | 2.3.3 | Externa | Procesamiento y organización de información obtenida desde la API |
| numpy | 2.4.1 | Externa | Soporte para operaciones numéricas y estructuración de datos |
| dicttoxml | 1.7.16 | Externa | Conversión de datos para la generación de reportes |
| reportlab | 4.4.7 | Externa | Generación de reportes de auditoría en formato PDF |
| tabulate | 0.9.0 | Externa | Presentación tabular de información de inventario |
| json | Nativa | Estándar | Procesamiento de respuestas de la API en formato JSON |

Las librerías expuestas en la **Tabla *12*** corresponden a las dependencias principales usadas en el desarrollo de los scripts de autenticación, inventario y monitoreo, estas permiten la comunicación con la API REST del controlador Cisco vManage, el procesamiento de la información obtenida y la generación de reportes y notificaciones automáticas aportando gran valor a la administración de la red SD-WAN adquiriendo una gestión más eficiente.

Con el fin de facilitar la replicabilidad del presente trabajo de grado las dependencias de software fueron centralizadas en un archivo de configuración, este archivo obtuvo mediante el comando **pip freeze > requirements.txt**, en la Figura 60 se puede observar el contenido del archivo requirements.txt que detalla las librerías necesarias para la ejecución adecuada de los scripts desarrollados.

**Figura 60**

*Listado de librerías requirements.txt*

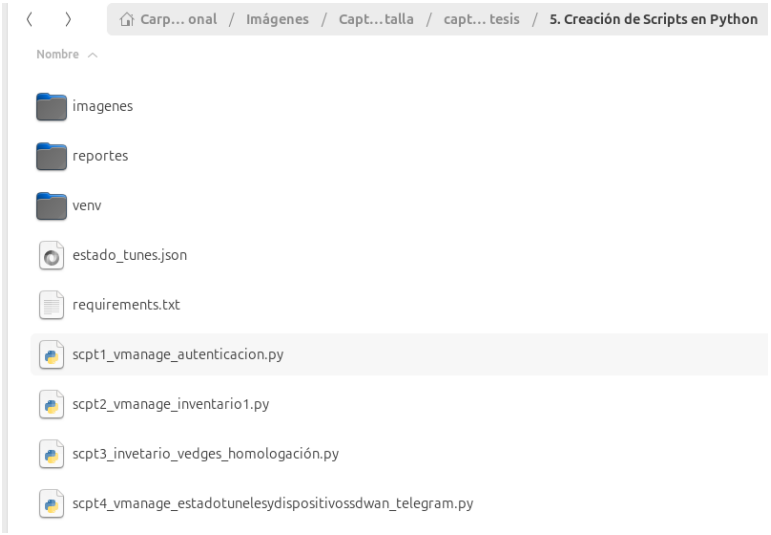



La Figura 61 presenta la estructura de carpetas y archivos de la solución referente al aparatado de automatización para el presente trabajo de grado, en la cual se observa la organización de los scripts desarrollados, archivos de configuración y los recursos auxiliares necesarios para la ejecución de la solución. A continuación, se detalla un pequeño listado los elementos:

1. Carpeta de reportes: Almacena reportes generados en los diferentes formatos.
2. Carpeta venv: Configuraciones del entorno virtual de Python.
3. Estado_tunes.json: Archivo de estado anterior para la generación de alertas.
4. Requtiments.txt: Archivo de Librerías usadas en la solución.
5. Scpt1_vmanage_autenticacion.py: script de autenticación.
6. Scpt2_vmanage_invetario1.py: script de inventario
7. Scpt3_vmanage_vedges_honologacion.py: script de inventario y homologación de formatos.
8. Scpt4_vmanage_estadotunelesydispositivossdwan_telegram.py: script de monitoreo y notificaciones vía Telegram.

**Figura 61**

*Estructura de carpetas capa de Automatización*



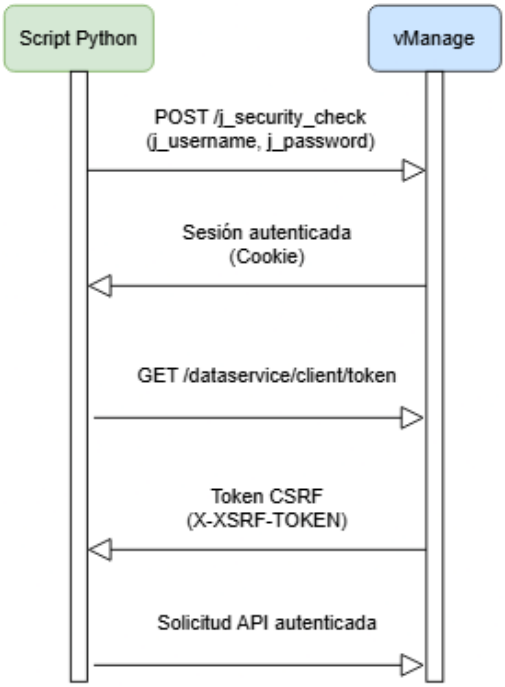

### *3.4.3 Autenticación y Gestión de Sesiones en vManage*

La seguridad es el componente más crítico en la interacción con la API de Cisco SD-WAN, a diferencia de las APIs más sencillas, vManage requiere de un proceso de autenticación de dos pasos nada complicado pero importante a fin de prevenir ataques de falsificación de peticiones en sitios cruzados (CSRF). Con el objetivo de habilitar este acceso seguro a través de las APIs REST se implementó un mecanismo de autenticación basado en sesiones y el uso de un token CSRF(Cross-Site Request Forgery Token) generado dinámicamente por el vManage, esta solución dio paso a establecer una comunicación confiable entre el script desarrollado en Python y la red SD-WAN a través de la controladora vManage, a fin de garantizar que todas las solicitudes realizadas se ejecuten en torno a un contexto autenticado y protegido.

Este proceso de autenticación y gestión de sesión es factible representarlo mediante un diagrama de secuencia como se observa en la **Figura *62***, el cual describe la interacción entre el cliente (script Python) y el servidor vManage.

**Figura 62**

*Diagrama de secuencia del proceso de autenticación.*



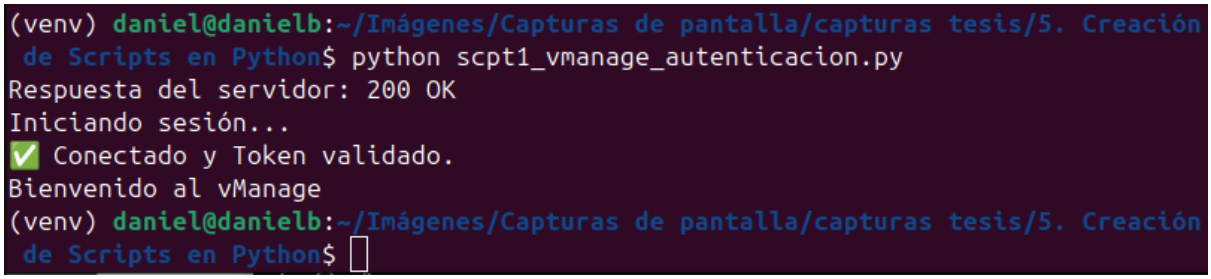

En el diagrama de secuencia presentado en la Figura 62, se observa el proceso de autenticación entre el script desarrollado en Python y el endpoint de autenticación del vManage, para lo cual se realiza una solicitud HTTP de tipo POST al endpoint /j_security_check, si las credenciales son válidas el vManage responde con una cookie de sesión que debe ser almacenada y enviada en el encabezado de las peticiones subsiguientes a fin de mantener el estado de la conexión. Desde la versión 19.2 del vManage en adelante no basta con la cookie, también se requiere realizar una petición GET adicional al endpoint /dataservice/client/token para autorizar el consumo de los servicios REST, el token obtenido es incorporado en la cabecera X-XSRF-TOKEN de la sesión activa.

Con la sesión y el token configurados de manera correcta, la solución queda habilitada para ejecutar solicitudes autenticadas a la API de vManage, permitiendo las consultas de estado de la red SD-WAN y el monitoreo de sus dispositivos y componentes, la continuación de este

procedimiento en el diagrama de secuencia se representa a través de la **Solicitud API autenticada** lo que simboliza un acceso seguro y controlado a la red SD-WAN.

**Figura 63**

*Validación Autenticación y conexión con la API de vManage*




En la **Figura *63***, se evidencia la ejecución exitosa del script de autenticación **scpt1_vmanage_autenticacion.py** en la terminal de comandos de una maquina Ubuntu Linux y dentro del entorno virtual de Python (venv) a fin de establecer una gestión adecuada de dependencias, este procedimiento forma parte del proceso de desarrollo de scripts orientados a la automatización de una infraestructura de red SD-WAN en el presente trabajo de grado.

Durante el proceso de ejecución el script interactúa con la API del controlador Cisco vManage, obteniendo como respuesta un **HTTP 200 OK**, que confirma la disponibilidad del servicio para posteriormente establecer una sesión autenticada que se valida mediante la obtención del token de seguridad, que se evidencia en la imagen a través de un mensaje de confirmación. La ejecución correcta del script valida el funcionamiento de la autenticación, constituyendo la base para la ejecución de scripts posteriores adecuados para la recolección y análisis de información de los dispositivos de la red.

### *3.4.4 Implementación del Consumo de Endpoints y Lógica de Automatización.*

En este apartado se implementó el consumo de los distintos endpoints proporcionados por el controlador vManage de la solución SD-WAN escogida para el presente trabajo de grado, con el principal objetivo de automatizar el monitoreo del estado de la red y generar notificaciones en tiempo real, para lo cual se emplearon los servicios REST, los cuales permiten

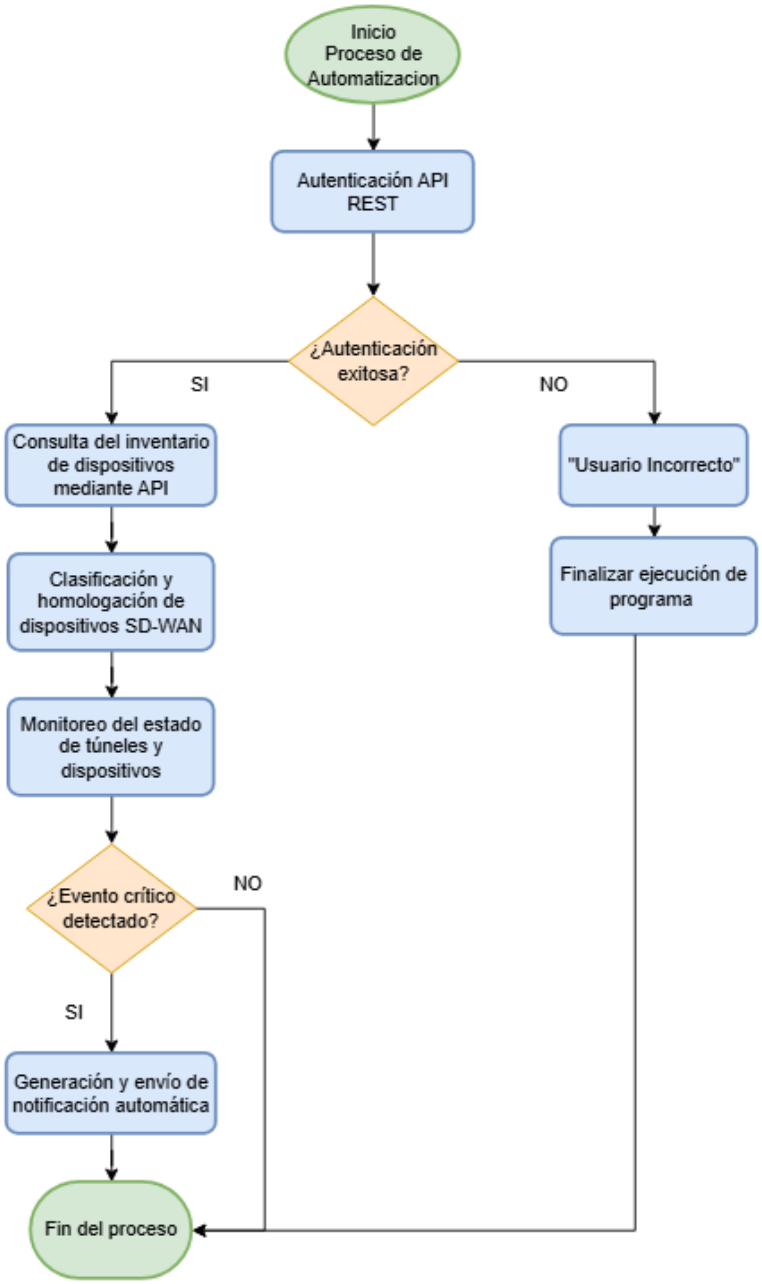
la interacción del sistema con scripts de Python de manera programática, facilitando la obtención de información sobre el estado operativo de los dispositivos y los túneles de comunicación.

El proceso de automatización inicia con el establecimiento de la sesión autenticada a través de la API REST con el controlador vManage para tener acceso a los recursos disponibles de la red SD-WAN, validad la sesión el sistema ejecuta solicitudes HTTP del tipo GET para realizar consultas de información relacionada con el estado de los dispositivos, túneles de comunicación, métricas operativas y la disponibilidad de los enlaces. Una vez obtenida la información esta es procesada a través de la lógica de automatización, la cual evalúa los resultados y determina si existen condiciones que precisen la generación de alertas automáticas.

A fin de representar de manera más clara y a través de una secuencia lógica el proceso de automatización implementado, en la **Figura *64*** se expone el diagrama de flujo correspondiente al consumo de los endpoints y la lógica de automatización desarrollada, este diagrama resume las etapas principales de la solución propuesta partiendo de la autenticación con el controlador SD-WAN hasta llegar al monitoreo y la generación de notificaciones en tiempo real.

**Figura 64**

*Diagrama de flujo de la lógica de automatización del sistema.*




Como parte de la automatización de la red se integró un mecanismo de notificaciones para enviar mensajes informativos a través de Telegram al personal encargado de la red en este caso el administrador, dicho proceso se ejecuta de manera automática y sin intervención manual, lo que contribuye un factor importante al momento de gestionar la red permitiendo una reacción oportuna ante eventos imprevistos en la misma. En la **Tabla *13*** se presenta los

principales endpoints usados durante la implementación, así como su función dentro del sistema desarrollado.

De esta manera, la lógica de automatización desarrollada e implementada permite centralizar la supervisión del estado de la red SD-WAN y responder de forma inmediata ante cambios en las condiciones operativas y reduciendo el tiempo de detección y respuesta frente a posibles fallos comunes en cualquier tipo de red.

**Tabla 13**

*Endpoints de la API de vManage usados.*

| Proceso | Método | Endpoint (recurso) | Scripts | Propósito Técnico |
|---|---|---|---|---|
| Autenticación | POST | /j_security_check | Todos | Autenticación inicial mediante envío de credenciales (j_username y j_password). |
| Seguridad | GET | /dataservice/client/token | Todos | Obtención del token CSRF necesario para validar operaciones de seguridad en la sesión.c |
| Inventario | GET | /dataservice/device | 2,3,4 | Recupera el inventario completo de dispositivos, incluyendo System IP, Hostname y estado de alcanzabilidad. |
| Control | GET | /dataservice/device/control/connections | 3 | Obtiene el estado detallado de las conexiones de control (vSmart/vBond) para un dispositivo específico.c |
| Interfaces | GET | /dataservice/device/interface | 3 | Extrae estadísticas de tráfico (Rx/Tx) y estado operativo de las interfaces de red de un vEdge. |
| Túneles BFD | GET | /dataservice/device/bfd/sessions | 4 | Consulta el estado de los túneles BFD para monitorear la estabilidad del plano de datos en tiempo real. |

En la Tabla 13 se detallan los procesos asociados al método HTTP empleado, a los recursos consumido, los scripts en los cuales se anexaron y el propósito técnico de cada uno de los endpoints usados dentro de la solución desarrollada e implementada, el proceso de autenticación se realiza en base al endpoint /**j_security_check** el cual permite validar las credenciales de acceso y así establecer una sesión inicial con el controlador, para posteriormente usar el endpoint **/dataservice/client/token** que permite la obtención del token CSRF necesario para autorizar las solicitudes REST realizadas durante una sesión activa.

Cabe mencionar que los servicios REST usados en la comunicación con el vManage se basan en el protocolo HTTP, el cual contempla y se encuentra estandarizado por el Internet Engeerieng Task Force (IETF) lo cual garantiza interoperabilidad y compatibilidad entre sistemas.

Establecida la sesión se detalla los endpoints relacionados a los procesos de inventario, control, interfaces y de túneles BFD, los cuales utilizan el método GET para obtener información de cada uno de los dispositivos como Sytem IP, Hostname y el estado de alcanzabilidad, adicional permiten identificar de manera detallada las conexiones de control que se encuentran establecidas a través del vBond y el vSmart. Con relación a las interfaces y túneles BFD es necesario el uso del endpoint **/dataservice/device/interface** que permite revisar el estado operativo de las interfaces de los dispositivos y **/dataservice/devise/bdf/sessions**, para realizar consultas de estado de los túneles y monitorear su comportamiento y estabilidad.

### *3.4.5 Validación de Resultados y Pruebas de Funcionamiento.*

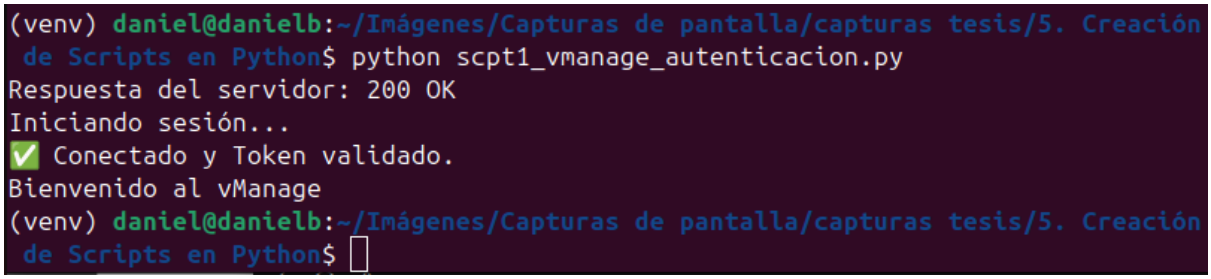
Con el propósito de comprobar el funcionamiento adecuado del sistema desarrollado se realizaron diferentes pruebas de funcionamiento, operabilidad y conectividad en un entorno de laboratorio controlado, emulación de la red SD-WAN en GNS3 y el desarrollo de los scripts en un entorno virtual de Python, para posteriormente ejecutar los scripts como parte de la solución de automatización propuesta en el presente trabajo de grado. Las pruebas permitieron

cotejar que los procesos de autenticación, consumo de endpoints y ejecución de la lógica de automatización se realizaran de manera adecuada y cumpliendo con los objetivos planteados en el presente capitulo.

Primero se validó el proceso de autenticación con el controlador SD-WAN debido a que este paso es la base de la comunicación entre la lógica de automatización y la red con lo cual se estableció una sesión usando las credenciales de validación y el token de seguridad requerido para el consumo de los servicios API REST, la correcta autenticación se evidencio a través de respuestas exitosas del tipo HTTP 200(OK), cuya validación y mensaje confirmo que la sesión fue creada de manera correcta y que el acceso a los recursos del controlador se encuentren habilitados, como se muestra en la **Figura *65***.

**Figura 65**

*Verificación del proceso de autenticación*

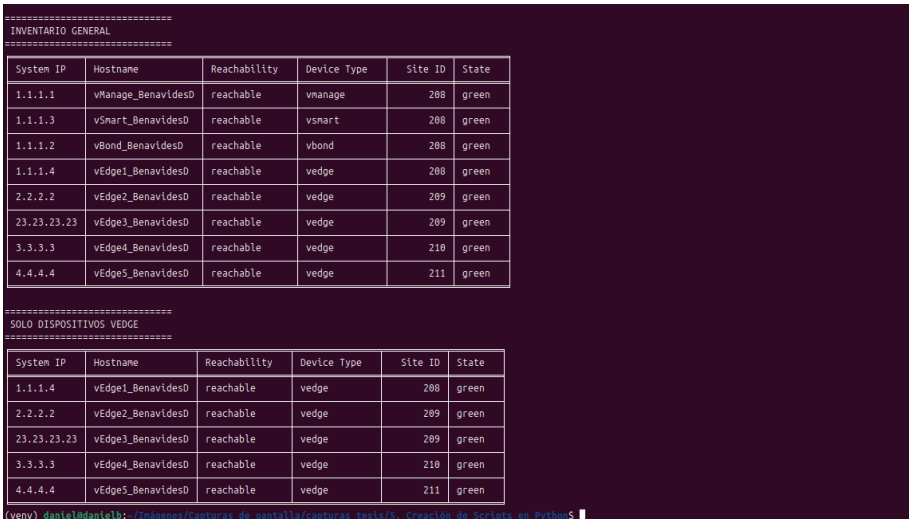



Una vez verificado el proceso de autenticación se procede a realizar pruebas relacionadas al consumo de los endpoints implementados para el monitoreo de la red, mediante solicitudes HTTP del tipo GET desde los scripts propuestos, el sistema logro recabar información correspondiente al estado operativo de los dispositivos, túneles y enlaces de comunicación, estos datos fueron posteriormente procesados por la lógica de automatización implementada a través de los scripts de Python, permitiendo su organización y análisis en favor de la supervisión del entrono SD-WAN, la **Figura *66*** presenta un ejemplo de los resultados obtenidos tras la ejecución de las consultas. Y en la **Figura *67*** se muestra cómo es posible solicitar ciertos datos de algún dispositivo en específico y mostrarlos a través de reportes en

formatos distintos al presentado mediante consola, esto se ampliara y analizara posteriormente en el Capítulo 4 de Análisis de Resultados.

**Figura 66**

*Verificación del proceso de consultas e inventario de dispositivos*



```
============================
 INVENTARIO GENERAL
============================
```

| System IP | Hostname | Reachability | Device Type | Site ID | State |
|---|---|---|---|---|---|
| 1.1.1.1 | vManage_BenavidesD | reachable | vmanage | 208 | green |
| 1.1.1.3 | vSmart_BenavidesD | reachable | vsmart | 208 | green |
| 1.1.1.2 | vBond_BenavidesD | reachable | vbond | 208 | green |
| 1.1.1.4 | vEdge1_BenavidesD | reachable | vedge | 208 | green |
| 2.2.2.2 | vEdge2_BenavidesD | reachable | vedge | 209 | green |
| 23.23.23.23 | vEdge3_BenavidesD | reachable | vedge | 209 | green |
| 3.3.3.3 | vEdge4_BenavidesD | reachable | vedge | 210 | green |
| 4.4.4.4 | vEdge5_BenavidesD | reachable | vedge | 211 | green |

```
============================
 SOLO DISPOSITIVOS VEDGE
============================
```

| System IP | Hostname | Reachability | Device Type | Site ID | State |
|---|---|---|---|---|---|
| 1.1.1.4 | vEdge1_BenavidesD | reachable | vedge | 208 | green |
| 2.2.2.2 | vEdge2_BenavidesD | reachable | vedge | 209 | green |
| 23.23.23.23 | vEdge3_BenavidesD | reachable | vedge | 209 | green |
| 3.3.3.3 | vEdge4_BenavidesD | reachable | vedge | 210 | green |
| 4.4.4.4 | vEdge5_BenavidesD | reachable | vedge | 211 | green |

```
(venv) daniel@danielb:~/Imágenes/Capturas de pantalla/capturas tesis/6. Creación de Scripts en Python$
```

**Figura 67**

*Verificación del proceso consulta y presentación de información*

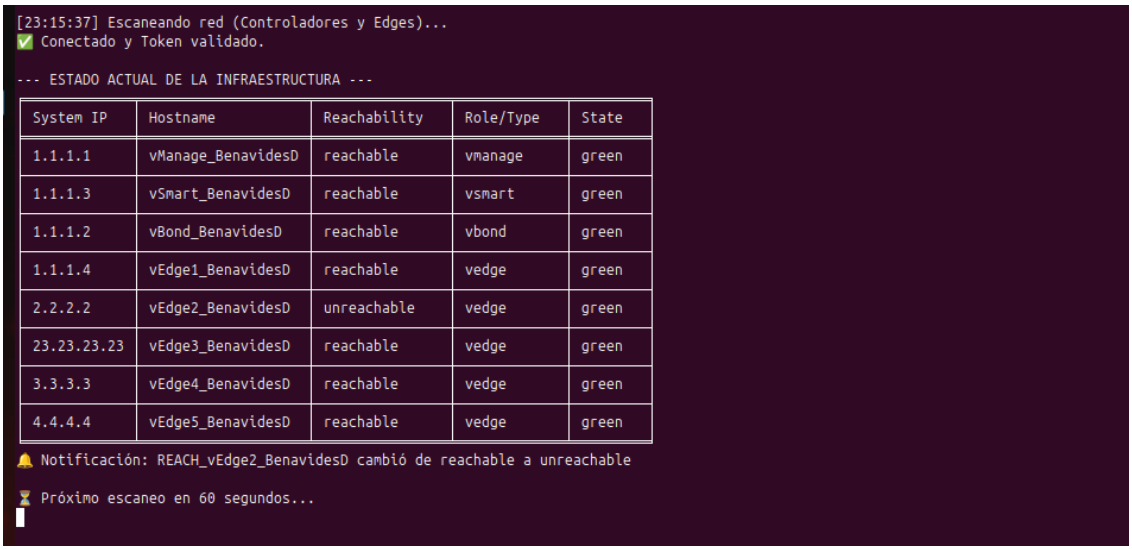
```
(venv) daniel@danielb:~/Imágenes/Capturas de pantalla/capturas tesis/5. Creación de Scripts en Python$ python scpt3_invetario_vedges_homologación.py
✅ Autenticación exitosa.

--- Dispositivos vEdge de la Red ---
+-------------+-------------------+
| IP          | Hostname          |
+=============+===================+
| 1.1.1.4     | vEdge1_BenavidesD |
+-------------+-------------------+
| 2.2.2.2     | vEdge2_BenavidesD |
+-------------+-------------------+
| 23.23.23.23 | vEdge3_BenavidesD |
+-------------+-------------------+
| 3.3.3.3     | vEdge4_BenavidesD |
+-------------+-------------------+
| 4.4.4.4     | vEdge5_BenavidesD |
+-------------+-------------------+

Ingrese System IP: 2.2.2.2
✅ Reportes generados en 'reportes'.
(venv) daniel@danielb:~/Imágenes/Capturas de pantalla/capturas tesis/5. Creación de Scripts en Python$
```

Adicionalmente, se evaluó el funcionamiento del script de notificaciones que forma parte del proceso de automatización del sistema, durante las pruebas se logró verificar que ante la detección de eventos relevantes en el estado de la red, el sistema genera y envía notificaciones de manera automática al administrador de la red dicho comportamiento

demuestra que la solución es capaz de alertar oportunamente sobre posibles incidencias y automatizando este proceso para que no exista la necesidad de la intervención manual, este proceso se evidencia en la Figura 68, donde se logra apreciar cómo se enlistan los dispositivos de la red y al mismo tiempo detecta el cambio de estado de un dispositivo de activo a desactivado.

**Figura 68**

*Verificación del proceso de monitoreo y notificación.*




Los resultados obtenidos y observados a través de las figuras expuestas anteriormente y durante las diferentes pruebas realizadas muestran que el sistema desarrollado cumple con el propósito de automatizar el monitoreo del entorno SD-WAN, al permitir la adquisición de información relevante de manera precisa y directa mejorando la capacidad de respuesta ante eventos operativos. En base a estas pruebas y validaciones se concluye que la solución presentada e implementada funciona de manera correcta dentro del escenario de pruebas propuesto y que se encuentra alineada con los objetivos planteados en el presente trabajo de grado.

# CAPÍTULO IV: Análisis de Resultados

El presente capítulo tiene como finalidad analizar los resultados obtenidos tras la implementación del sistema de automatización para el monitoreo de una red SD-WAN, los resultados presentados corresponden a las pruebas realizadas en un entorno de laboratorio y permiten evaluar el cumplimiento de los objetivos definidos para el desarrollo de la solución.

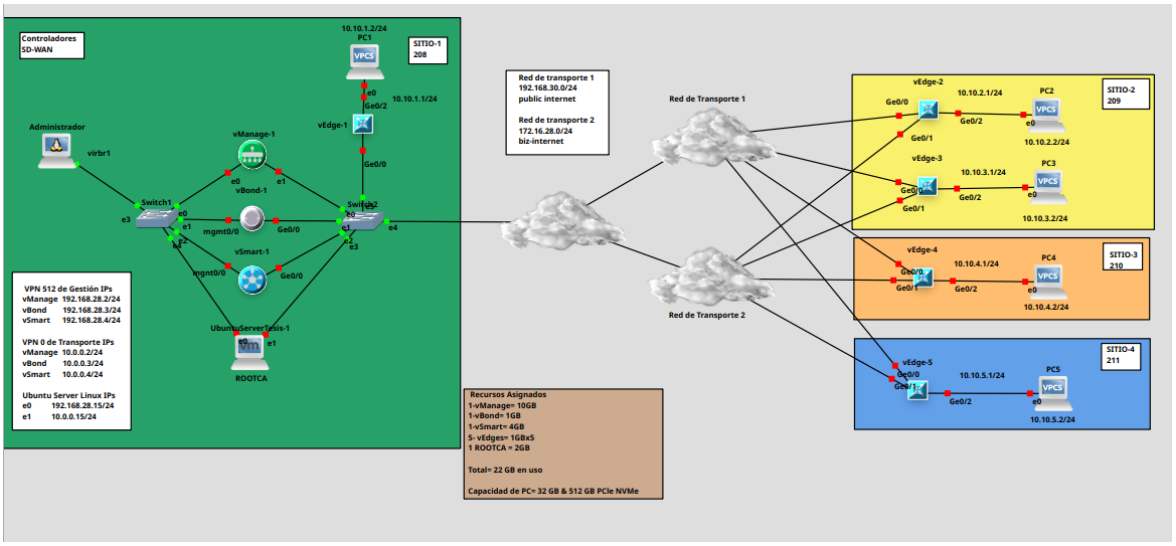
## 4.1. Entorno de pruebas

El proceso de validación del sistema propuesto se llevó a cabo en un entorno de laboratorio controlado en un sistema operativo Ubuntu Linux, el cual permitió simular el funcionamiento de una red SD-WAN basada en el controlador vManage. Dicho entorno fue utilizado para verificar la correcta ejecución de los scripts desarrollados dentro de un entorno virtual de Python, así como el consumo de los endpoints necesarios para el monitoreo del estado de la red, el entrono se logra observar en la **Figura *69***.

Las pruebas se realizaron considerando escenarios básicos de operación, enfocados principalmente en la autenticación contra el controlador, la consulta del estado de los dispositivos y túneles, y la generación de notificaciones automáticas ante eventos relevantes. Este enfoque permitió evaluar el comportamiento general del sistema sin introducir variables externas que pudieran afectar los resultados.

**Figura 69**

*Topología de red Cisco SD-WAN en GNS3*



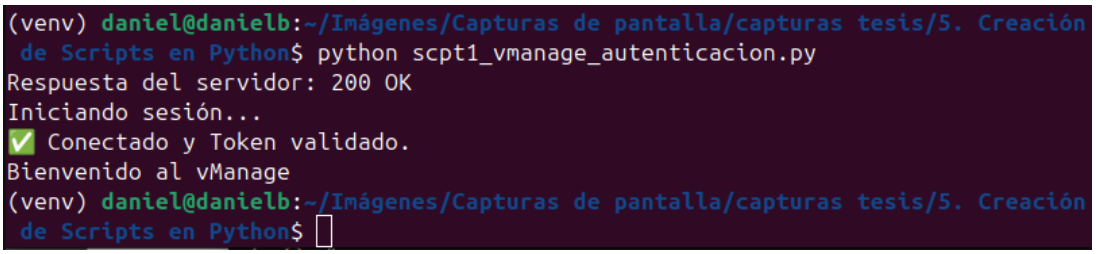

En la **Figura *69*** se detalla el entorno de simulación usado para el desarrollo de presente trabajo de grado cumpliendo con la arquitectura de red establecida en la Figura 7, en la imagen se puede observar que la topología en el área de controladores se encuentra conectada a un administrador de red a través de una interfaz de red virbr1 preestablecida con anterioridad, dicha interfaz de encuentra conectada mediante la VPN 512 denominada de gestión a la maquina física con sistema operativo Ubuntu Linux, la cual ejerció de administrador y desde donde se ejecutaron cada uno de los scripts de automatización desarrollados posteriormente.

### 4.2. Resultados de la Autenticación y Consumo de Endpoints

Como primer paso del proceso de automatización, se validó el establecimiento de una sesión autenticada con el controlador SD-WAN vManage como se observa en la **Figura *70***, esta etapa es fundamental, ya que habilita el acceso a los distintos recursos expuestos a través de los servicios REST, durante las pruebas realizadas, el sistema logró autenticarse correctamente, recibiendo respuestas HTTP del tipo 200 (OK), lo que confirma que las credenciales y el mecanismo de sesión funcionan de manera adecuada.

**Figura 70**

*Visualización del 200 OK de autenticación y conexión*




Validada la autenticación, se procede al consumo de los endpoints implementados que se detallan en la **Tabla *14*** para la obtención de información relacionada con el estado operativo de la red SD-WAN.

**Tabla 14**

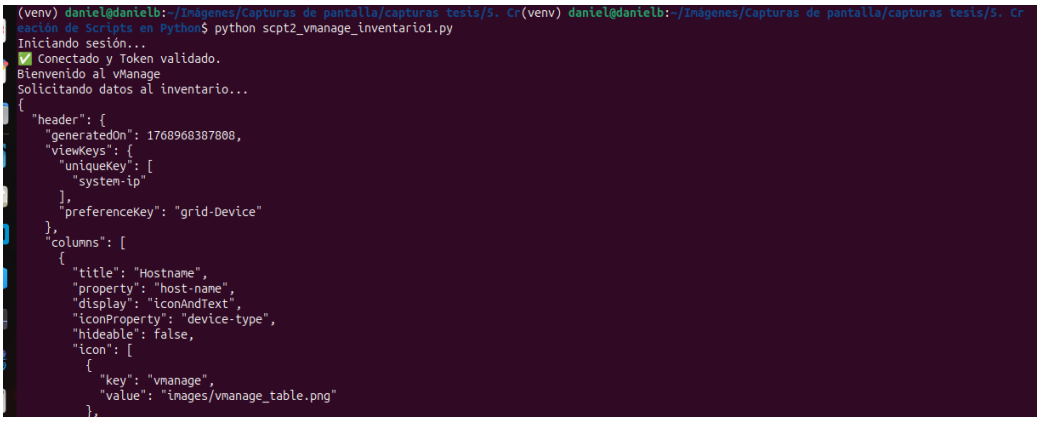
*Endpoints utilizados*

| **Endpoint** | **Método** | **Función** |
|---|---|---|
| /j_security_check | POST | Autenticación con vManage |
| /dataservice/device | GET | Obtención del inventario de dispositivosC |
| /dataservice/device/tunnel | GET | Consulta del estado de túneles |

Los resultados obtenidos evidencian que las solicitudes HTTP de tipo GET permiten recuperar información relevante sobre los dispositivos y túneles como se aprecia en la **Figura *71*** , datos utilizados por la lógica de automatización para su posterior análisis.

**Figura 71**

*Respuesta de un endpoint (formato JSON)*

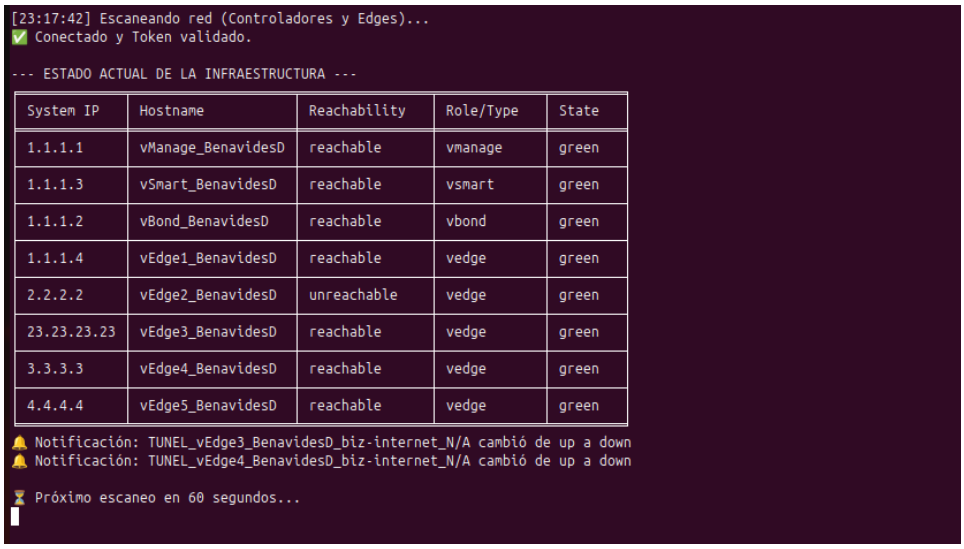



## 4.3. Resultados del Monitoreo Automatizado y Sistema de Notificaciones

Una vez validada la autenticación y el acceso a los endpoints, se procedió a evaluar el funcionamiento del monitoreo automatizado. En esta etapa, el sistema ejecuta de manera secuencial las consultas configuradas, permitiendo obtener información actualizada sobre el estado de los dispositivos, accesibilidad, tipo de dispositivo, enlaces y túneles de la red SD-WAN, estos resultados demuestran que la lógica de automatización permite identificar el estado operativo de los elementos monitoreados, facilitando la detección temprana de posibles fallos o degradaciones en la conectividad. Este proceso se realiza sin intervención manual, lo que contribuye a una gestión más eficiente de la red como se observa en la **Figura *72*** .

**Figura 72**

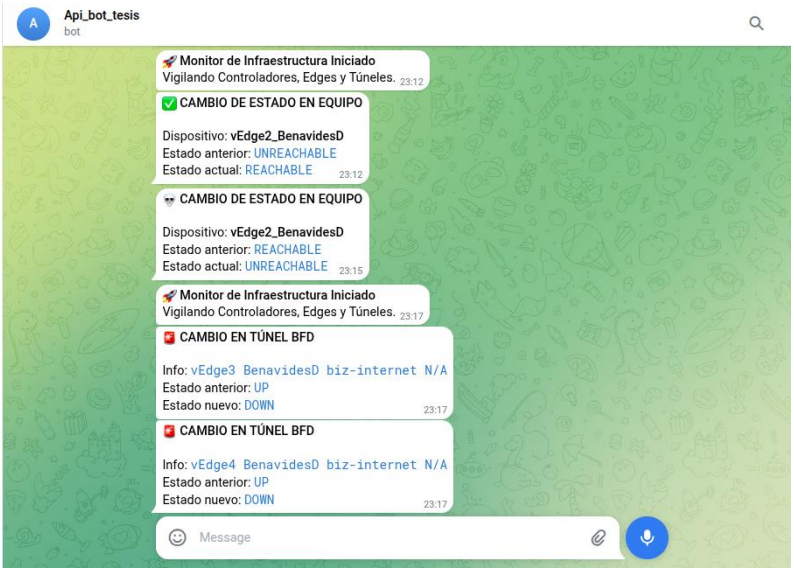
*Visualización de datos mediante el script de monitoreo*




Para validar la eficacia del monitoreo se optó por provocar una desconexión intencional en uno de las redes de transporte y adicional suspender momentáneamente uno de los dispositivos vEdge de la sucursal virtual, el resultado fue gratificante debido a que en menos de un minuto el script detecto anomalías en el sistema, y presentándolas como notificación a través de un bot de Telegram dende se visualiza un cambio de estado de activado a desactivado con respecto al dispositivo vEdge, y de una alteración en el túnel BFD identificado por el color asignado dentro del TLOC prestablecido en cada uno de los dispositivos como se observa en la **Figura** ***73***.

**Figura 73**

*Mensaje de alerta a través de un Bot en Telegram*

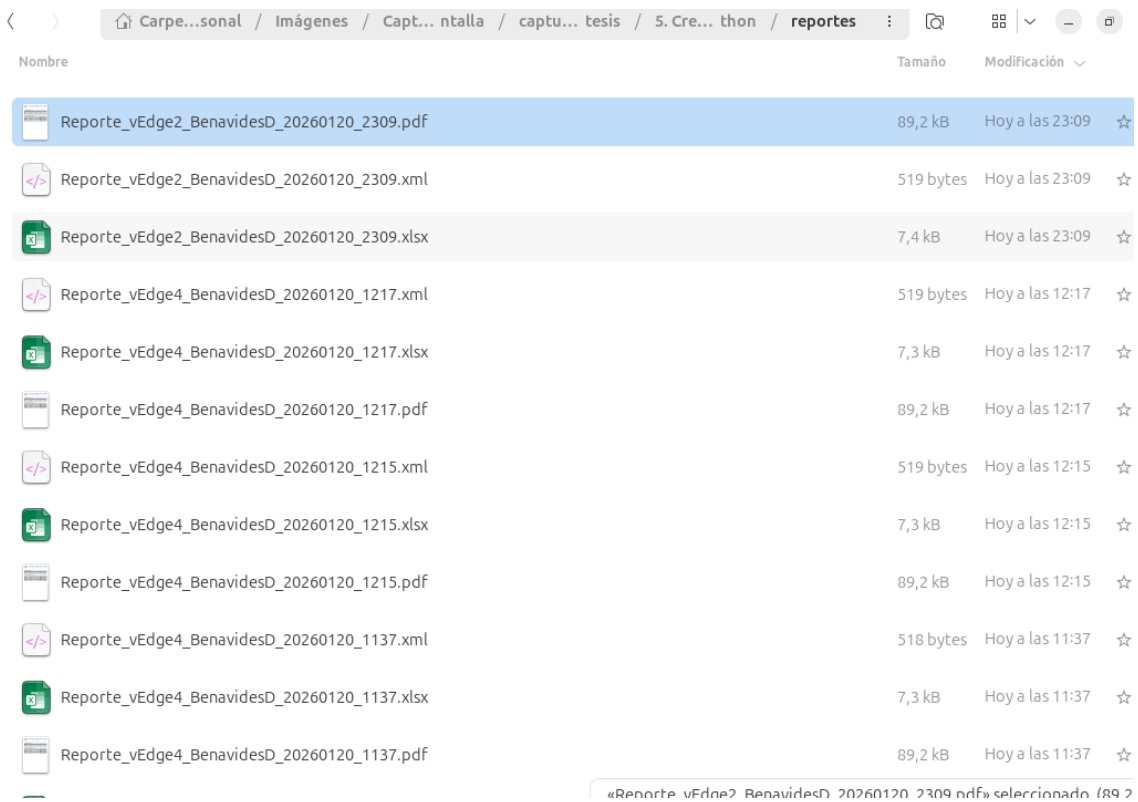



## 4.4. Resultados de la Auditoria e Inventario Automatizado

Como parte del proceso de automatización propuesto, se implementó un mecanismo de auditoría e inventario automatizado de los dispositivos que conforman la red SD-WAN. La información recopilada es procesada automáticamente por el sistema y mostrada a manera de reportes en tres formatos distintos al presentado a través de consola, los formatos establecidos fueron PDF, XML Y XLSX esto con el fin de lograr una visualización más concreta lo que evita la necesidad de realizar consultas manuales a través de la interfaz gráfica del vManage, para almacenar los archivos en los formatos mencionados se crea una carpeta denominada reportes como se evidencia en la **Figura *74***, guardando los archivos en los tres formatos mencionados al momento de seleccionar el dispositivo a través del monitor consola.

**Figura 74**

*Carpeta de reportes*




Este proceso permite obtener de manera centralizada información relevante sobre los equipos gestionados por el controlador vManage, facilitando la supervisión y el control del estado general de la infraestructura, durante las pruebas realizadas, el sistema fue capaz de consultar de forma exitosa el inventario de dispositivos registrados en el controlador, obteniendo datos como el identificador del equipo, tipo de dispositivo, estado operativo entre otros como se observa en la **Figura *75***.

**Figura 75**

*Reporte de dispositivos en formato PDF*

Reporte_vEdge2_BenavidesD_20260120_2309.pdf

## Informe Estado de vEdges en la SD-WAN

*Reporte generado: 20/01/2026 23:09*

### 1. CONEXIONES DE CONTROL

| Peer Type | Peer IP | Site ID | State |
|---|---|---|---|
| vsmart | 10.0.0.4 | 208 | up |
| vsmart | 10.0.0.4 | 208 | up |
| vmanage | 10.0.0.2 | 208 | up |

### 2. ESTADO DE INTERFACES Y PAQUETES

| VPN | Interface | IP Address | Rx (kbps) | Tx (kbps) | Rx Pkts | Tx Pkts |
|---|---|---|---|---|---|---|
| 0 | ge0/0 | 192.168.30.2/24 | 11 | 11 | 33741 | 32870 |
| 0 | ge0/1 | 172.16.28.2/24 | 13 | 13 | 40531 | 41318 |
| 0 | system | 2.2.2.2/32 | 0 | 0 | 0 | 0 |
| 1 | ge0/2 | 10.10.2.1/24 | 0 | 0 | 6 | 14 |

Los resultados obtenidos evidencian que la auditoría e inventario automatizado permiten mantener una visión actualizada de la red, contribuyendo a una administración más ordenada y eficiente enfoque que resulta muy útil en entornos donde se gestionan múltiples dispositivos, debido a que se reduce el tiempo requerido para la recopilación de información y minimiza la posibilidad de errores humanos.

**Tabla 15**

*Información obtenida de inventario y reporte*

| Parámetro | Descripción |
|---|---|
| System IP | Identificador único de cada equipo |
| Hostname | Nombre del dispositivo |
| Accesibilidad | Reachable/unreachable |
| Tipo de dispositivo | Contolador/ vEdge |
| Site ID | Identificador del sitio |
| Estado | Up/Down |
| Estado de interfaces | Interfaces activas |
| Estado de paquetes | Rx/Tx |

Adicionalmente, tanto el inventario como los reportes generados pueden ser utilizados como base para futuras tareas de monitoreo y validación del estado de la red, así como para la detección temprana de inconsistencias o cambios en la infraestructura. En este sentido, la automatización de la auditoría representa un aporte significativo al proceso de gestión de redes SD-WAN, es necesario aclarar que el script scpt3_invetario_vedges_homologación.py contiene una parte del inventario y se enfoca en la generación de los reportes, esto con el objetivo de especificar que puede ser utilizado antes o después del monitoreo y notificaciones de la red.

## 4.5. Análisis General de los Resultados

En base a los resultados obtenidos, se puede afirmar que la solución desarrollada cumple con los objetivos planteados en el presente trabajo de titulación. La correcta integración de los servicios REST con la lógica de automatización permitió centralizar el monitoreo del estado de la red SD-WAN y responder de manera oportuna ante eventos relevantes, en la **Tabla *16*** se especifica un resumen de los resultados del sistema en base a los scripts.

**Tabla 16**

*Resumen de resultados del sistema*

| Script | Estado | Comentario |
|---|---|---|
| Autenticación | Correcto | Sesión establecida |
| Monitoreo | Correcto | Datos obtenidos |
| Notificación | Correcto | Alerta generada |
| Reporte | Correcto | Archivos almacenados |

Si bien el sistema fue evaluado en un entorno de laboratorio y con un enfoque practico, los resultados evidencian que la solución es funcional y puede ser considerada como una base para futuras implementaciones en entornos reales, incorporando mejoras y ampliaciones según las necesidades de la red.

En este capítulo se presentaron los resultados obtenidos a partir de la implementación y validación del sistema propuesto, los cuales sirven como base para el desarrollo de las conclusiones y recomendaciones expuestas en el capítulo siguiente.

# CONCLUSIONES Y RECOMENDACIONES

## Conclusiones

Luego de realizar una investigación preliminar con relación a las redes SD-WAN, entornos de automatización y la aplicación de APIs en su infraestructura, se concluye que es uno de los temas más interesantes en la actualidad dentro del mundo de las redes, debido a que la transición de una gestión basada en CLI hacia una arquitectura programable mediante APIs REST de Cisco vManage es técnica y operativamente superior, proporcionando una base de datos estructurada que facilita la interoperabilidad con herramientas externas como Python y

Telegram dado que muchos procesos son repetitivos lo que da paso a buscar soluciones de automatización.

Se identificó que los procesos de auditoría manual y monitoreo visual son los puntos críticos que restan eficiencia a la red. La aplicación de APIs permitió automatizar estos requerimientos, logrando una estandarización total en la generación de reportes y eliminando el factor del error humano en la recolección de métricas operativas, cumpliendo con la condición de posibles mejoras debido a que la respuesta de las solicitudes mediante la automatización fue más específica y detallada a diferencia de consultas de manera manual a través de la interfaz de control del vManage debido a que requiere una búsqueda general.

La integración exitosa de la solución en un entorno emulado en GNS3 demostró la viabilidad de un modelo de desarrollo modular y desacoplado. Esta arquitectura permitió que los scripts de Python interactuaran con el controlador de forma centralizada sin afectar el tráfico de producción, validando la escalabilidad del sistema propuesto. Este es un punto importante que se debe tener en cuenta debido a que intentar concatenar todos los scripts en uno solo puede generar problemas de ejecución y posteriormente errores en aspectos que corresponden a una sola sección.

Las pruebas operativas confirmaron la eficacia de la automatización al ser capaz de enviar notificaciones de forma más eficaz mediante un alerta, que a través del panel de control del vManage , dependiendo la programación del script donde se utilizó una variable para el lapso de las alertas, variable netamente configurable y que se puede adaptar a las condiciones de la red y del administrador, los resultados evidenciaron que la red automatizada es capaz de generar alertas proactivas y reportes precisos en base a las condiciones de la red, lo que permite efectividad en su gestión cumpliendo satisfactoriamente con los parámetros de funcionamiento esperados.

**Recomendaciones**

Se recomienda que, para implementaciones en redes de producción real, los scripts desarrollados no se ejecuten desde una estación de trabajo local, sino que se migren a un servidor dedicado o a una arquitectura basada en contenedores (como Docker) alojada en la nube o en un centro de datos. Esto garantizaría que el módulo de monitoreo y el bot de Telegram mantengan una disponibilidad del 99.9%, operando de manera ininterrumpida e independiente del estado de la computadora personal del administrador de la red.

Para elevar el nivel de seguridad de la solución, se sugiere implementar mecanismos avanzados para la gestión de credenciales, tales como el uso de variables de entorno o gestores de secretos. Esta medida ayudaría a proteger la información sensible utilizada durante el proceso de autenticación con el controlador.

Como trabajo futuro, se recomienda ampliar la solución incorporando métricas adicionales y mecanismos de análisis que permitan evaluar el comportamiento de la red a lo largo del tiempo. Esto abriría la posibilidad de evolucionar hacia un modelo de monitoreo más completo, orientado a la prevención de fallos.

Se sugiere explorar la integración de la solución desarrollada con frameworks de automatización de redes, lo que permitiría optimizar tareas de configuración y gestión en redes SD-WAN de mayor escala.

## REFERENCIAS BIBLIOGRÁFICAS

Augusto, G. (2018, May 31). *Community | GNS3*. https://gns3.com/community/featured/viptela-lab-in-gns3-is-it-possib

Azhar, I. (2021). *Testing of SD-WAN Vendors APIs For Service Providers Integration*. https://openrepository.aut.ac.nz/handle/10292/14259

Cisco. (2019). *Cisco SD-WAN Cloud scale architecture*. https://www.cisco.com/go/trademarks

Cisco. (2020a). *Cisco SD-WAN Introduction Part 1*. Yasser Auda. https://learningnetwork.cisco.com/s/article/cisco-sd-wan-introduction-part-1

Cisco. (2020b). *SD-WAN Solution - Cisco Software-Defined WAN for Secure Networks White Paper - Cisco*. https://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/sd-wan/white-paper-c11-741640.html

Cisco. (2024a). *What Is SD-WAN? - Software-Defined WAN (SDWAN) - Cisco*. https://www.cisco.com/c/en/us/solutions/enterprise-networks/sd-wan/what-is-sd-wan.html?dtid=osscdc000283

Cisco. (2024b, February 2). *Comprender la relación del protocolo BFD con el routing con reconocimiento de aplicaciones - Cisco*. https://www.cisco.com/c/es_mx/support/docs/routers/sd-wan/221604-understand-bfd-protocol-relationship-wit.html

Cisco. (2024c, August 1). *Design Zone for Branch/WAN - Cisco Catalyst SD-WAN Design Guide*. Https://Www.Cisco.Com/c/En/Us/Td/Docs/Solutions/CVD/SDWAN/Cisco-Sdwan-Design-Guide.Html#SDWANRouting. https://www.cisco.com/c/en/us/td/docs/solutions/CVD/SDWAN/cisco-sdwan-design-guide.html#SDWANRouting

Cisco. (2025, December 1). *Cisco Catalyst SD-WAN Control Components Compatibility Matrix and Recommended Computing Resources - Recommended Computing Resources for Cisco SD-WAN Controller Release 20.5.x (On-Prem Deployment) [Cisco SD-WAN] - Cisco*. https://www.cisco.com/c/en/us/td/docs/routers/sdwan/release/notes/compatibility-and-server-recommendations/ch-server-recs-20-5.html

Cuaical, A. (2023). *Testbed para el estudio de la tecnología SD-WAN en la Universidad Técnica del Norte*. https://repositorio.utn.edu.ec/handle/123456789/15043

Cusco, W., Cabrera, J., & Lugo, J. (2022). Análisis de las tecnologías SD-WAN usadas en Ecuador. *Dominio de Las Ciencias, ISSN-e 2477-8818, Vol. 8, Nº. Extra 2, 2022 (Ejemplar Dedicado a: Mayo Especial 2022), 886 Págs.*, *8*(2), 870–886. https://doi.org/10.23857/dc.v8i2.2789

ECLAC. (2022). *Digital technologies for a new future*. www.cepal.org/apps

Fortinet. (2023). *Cuadrante Mágico™ de Gartner® de 2023 para SD-WAN: Fortinet es nombrado líder*. https://www.fortinet.com/lat/resources/analyst-reports/gartner-wan-edge

Gervasi, P. (2023, July 26). *SD-WAN Best Practices | Kentik Blog*. https://www.kentik.com/blog/sd-wan-best-practices/

GNS3. (2025). *Getting Started with GNS3 | GNS3 Documentation*. https://docs.gns3.com/docs/. https://docs.gns3.com/docs/

Gordeychik, S., Kolegov, D., & Nikolaev, A. (2018). *SD-WAN Internet Census*. https://arxiv.org/abs/1808.09027v2

Jia, W. K., Chou, Y. Y., & Chen, Y. C. (2020). QoS Improvement of VoIP over SDN. *2020 IEEE 17th Annual Consumer Communications and Networking Conference, CCNC 2020*. https://doi.org/10.1109/CCNC46108.2020.9045152

Juniper. (2022). *MTU de medios y MTU de protocolo | Junos OS | Juniper Networks*. https://www.juniper.net/documentation/mx/es/software/junos/interfaces-fundamentals/topics/topic-map/media-mtu.html

Kreutz, D., Ramos, F. M. V., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2015). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE*, *103*(1), 14–76. https://doi.org/10.1109/JPROC.2014.2371999

Landázuri, M., & Verdesoto, P. (2020). *Desarrollo de una guía de laboratorio para el uso de Devnet en entornos académicos*. https://dspace.udla.edu.ec/handle/33000/13088

Lee, J. (2025, December 4). *¿Qué es SD-WAN?* https://www.trendmicro.com/es_es/what-is/what-is-zero-trust/sd-wan.html.

Moreno, S. (2021). *COMPARACIÓN DE ASPECTOS OPERATIVOS Y ECONÓMICOS ENTRE SD-WAN Y MPLS PARA ESTABLECER LA MEJOR OPCIÓN DE UNA EMPRESA CORPORATIVA A NIVEL NACIONAL E INTERNACIONAL*.

NetworkAcademy.io. (2021a). *RESTful APIs | NetworkAcademy.io*. https://www.networkacademy.io/ccie-enterprise/sdwan/cisco-sd-wan-rest-apis

NetworkAcademy.io. (2021b). *What is SD-WAN? | NetworkAcademy.io*. https://www.networkacademy.io/ccie-enterprise/sdwan/what-is-sd-wan

Pérez, F., & Gualoto, J. (2022). *Simulación de una red sd-wan mediante equipos de networking en gns3.* http://bibdigital.epn.edu.ec/handle/15000/22097

Salazar, A. (2022). Diseño de una red SD-WAN para la transformación de las comunicaciones empresariales en las entidades financieras. *Repositorio Institucional - UTP*. http://repositorio.utp.edu.pe/handle/20.500.12867/7756

Salazar, G., & Marrone, L. (2021). *Hybrid Networking SDN y SD-WAN: Interoperabilidad de arquitecturas de redes tradicionales y redes definidas por software en la era de la digitalización*. https://doi.org/10.35537/10915/129910

Salazar, L. (2022). Revisión de literatura redes definidas por software para manejo de VoIP. *Repository.Udistrital.Edu.Co*. https://repository.udistrital.edu.co/handle/11349/29230

Sarmiento, G. (2023). *Uso de APIs y programabilidad en SDWAN Viptela*. PUCE - Quito. https://repositorio.puce.edu.ec/handle/123456789/32520

Scarpitta, C., Ventre, P., Lombardo, F., Salsano, S., & Blefari, N. (2021). *EveryWAN-An Open Source SD-WAN solution*. 7–8.

Sharma, N., Sharma, A., & Ahlawat, N. (2023). SD-WAN: The Future of Networking. *Researchgate.Net*, *11*. https://doi.org/10.22214/ijraset.2023.51475

Sol, M. B. del. (2020). *Diseño y despliegue de escenarios de red sobre un entorno de pruebas virtualizado SD-WAN basado en tecnología Viptela*. https://oa.upm.es/id/eprint/68356

Stadista. (2023). *Global SD-WAN market revenue 2027 | Statista*. Lionel Sujay Vailshery. https://www.statista.com/statistics/895974/worldwide-sd-wan-infrastructure-revenue/#statisticContainer

Teldat. (2019). *SD-WAN en América Latina - Teldat*. Fernando Castro. https://www.teldat.com/es/blog/sd-wan-america-latina-wan-de-calidad-segurida-disponibilidad/

Troia, S., Zorello, L., & Maier, G. (2021). SD-WAN: how the control of the network can be shifted from core to edge. *Ieeexplore.Ieee.Org*. https://ieeexplore.ieee.org/abstract/document/9492375/

Vargas, J. L. (2020). *Evolución de red en sucursales a SD-WAN*. https://openaccess.uoc.edu/handle/10609/116646

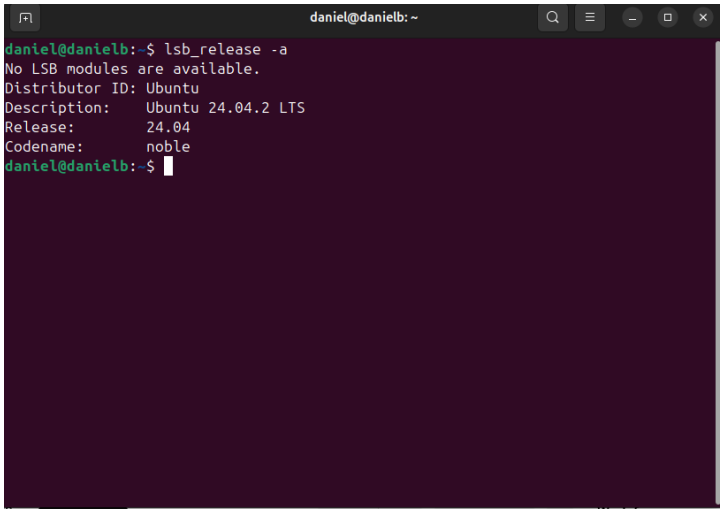
Yadav, S. (2021). *SD-WAN Service Analysis, Solution, and its Applications*. https://era.library.ualberta.ca/items/2613b784-8aa6-498c-accb-b8bb86462b58

# ANEXOS

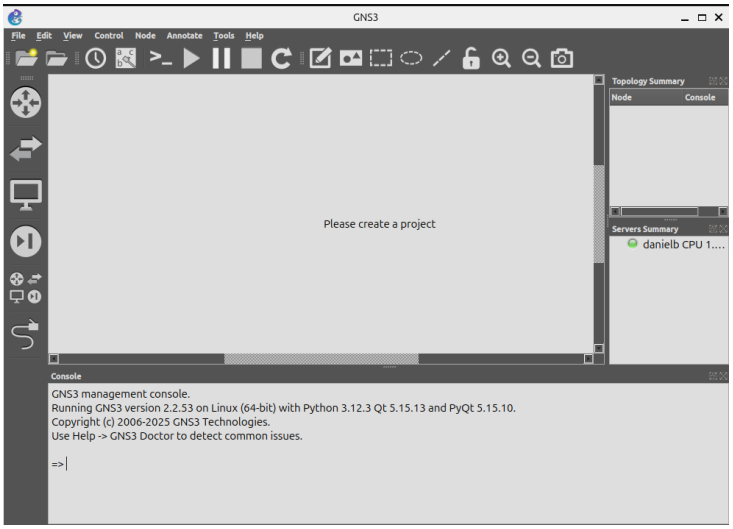
## Anexo 1: Especificaciones para el entorno de Emulación

En el presente anexo se presenta imágenes del entorno GNS3 y de las características del sistema operativo en el cual se va a desarrollar la solución propuesta en el presente trabajo de grado.

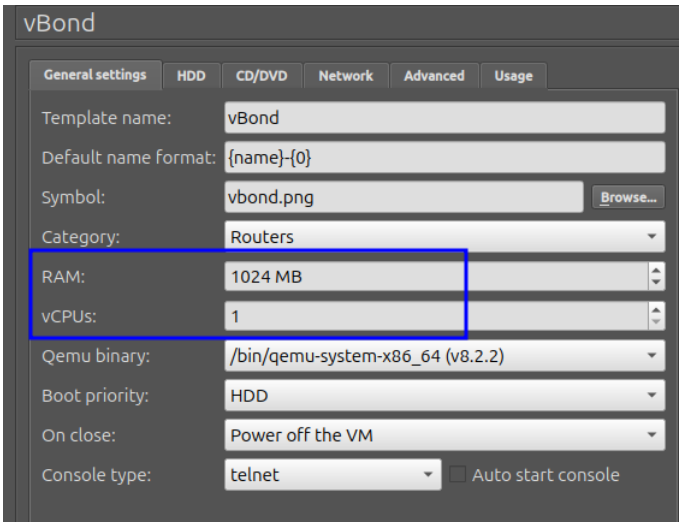
1. Imagen de características de sistema operativo.



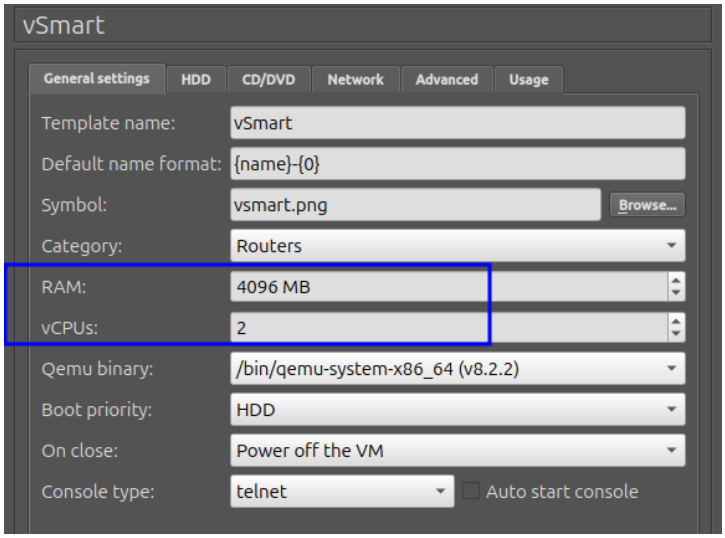

2. Imagen de entorno GNS3 para la emulación de la red.



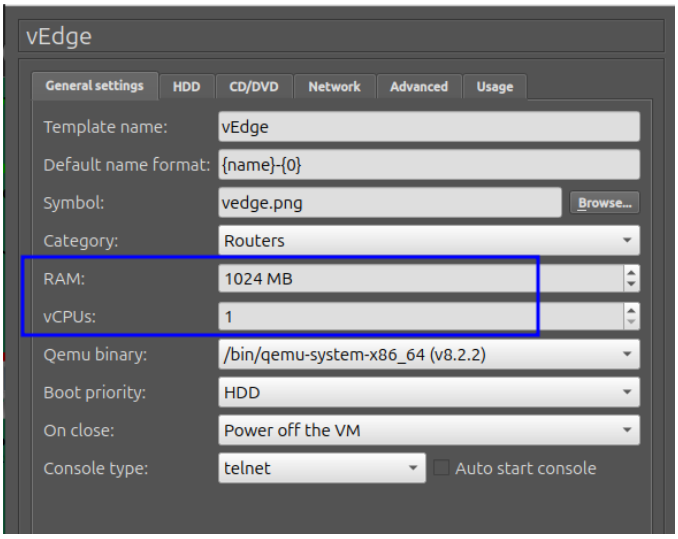
## Anexo 2: Configuraciones Físicas de vBond




## Anexo 3: Configuraciones Físicas de vSmart

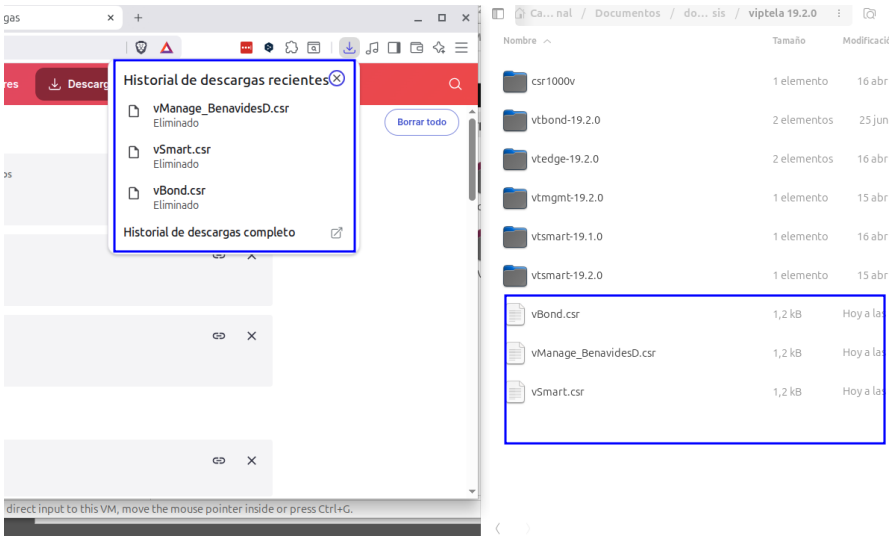


## Anexo 4: Configuraciones Físicas de los vEdges



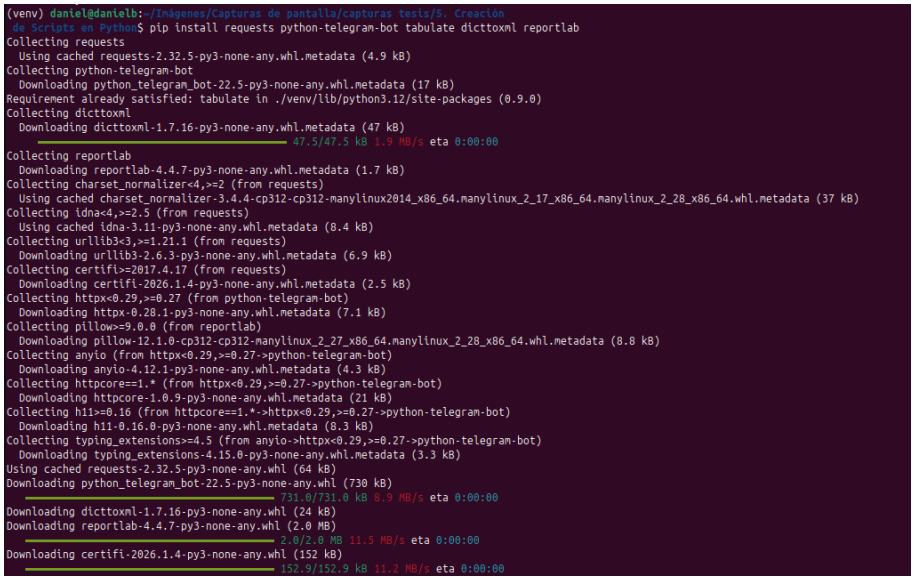

## Anexo 5: Descarga de archivos CRT desde vManage

En este anexo se muestra los archivos crt de cada controlador descargados desde el vManage, dichos archivos son necesarios ya que al firmarlos en el servidor certificador ROOTCA se crean los certificados crt con los cuales es posible resgistar y habilitar cada uno de los controladores dentro de la red SD-WAN.



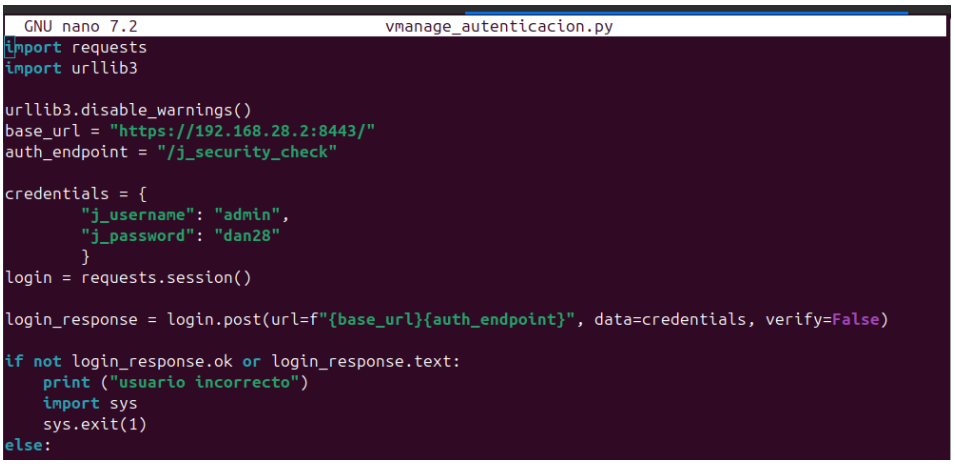
## Anexo 6: Instalación de Librerías

En este anexo se evidencia el proceso de instalación de librerías necesarias para ejecutar los scripts, dichas librerías se instalan mediante el comando **pip install "libreria"** dentro del entorno virtual creado en Python.

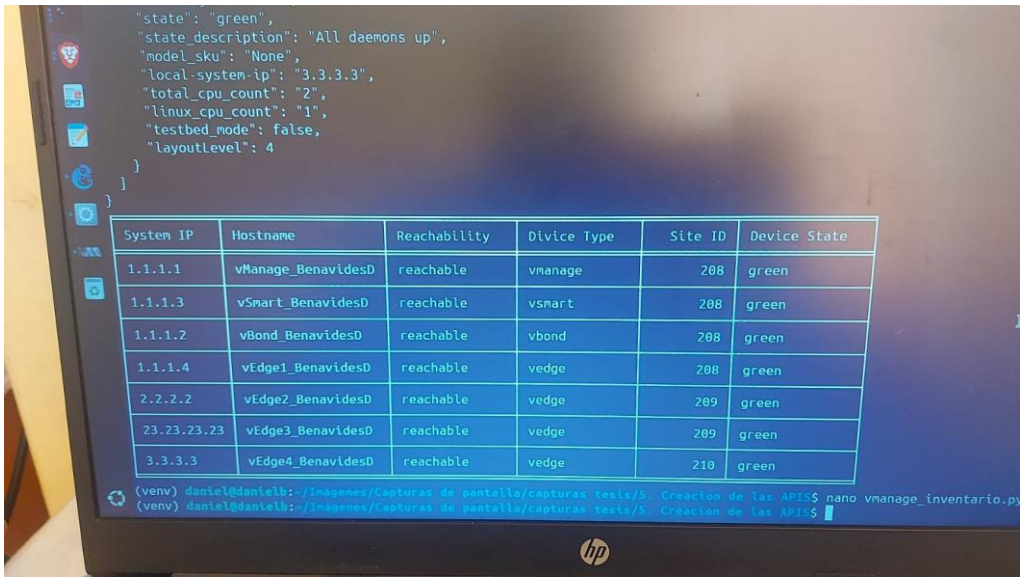



## Anexo 7: Estructura base para los scripts en Python

El anexo presenta la estructura base para la mayoría de scripts en Python, siendo este el de autenticación y comunicación a través de la API REST del vManage.

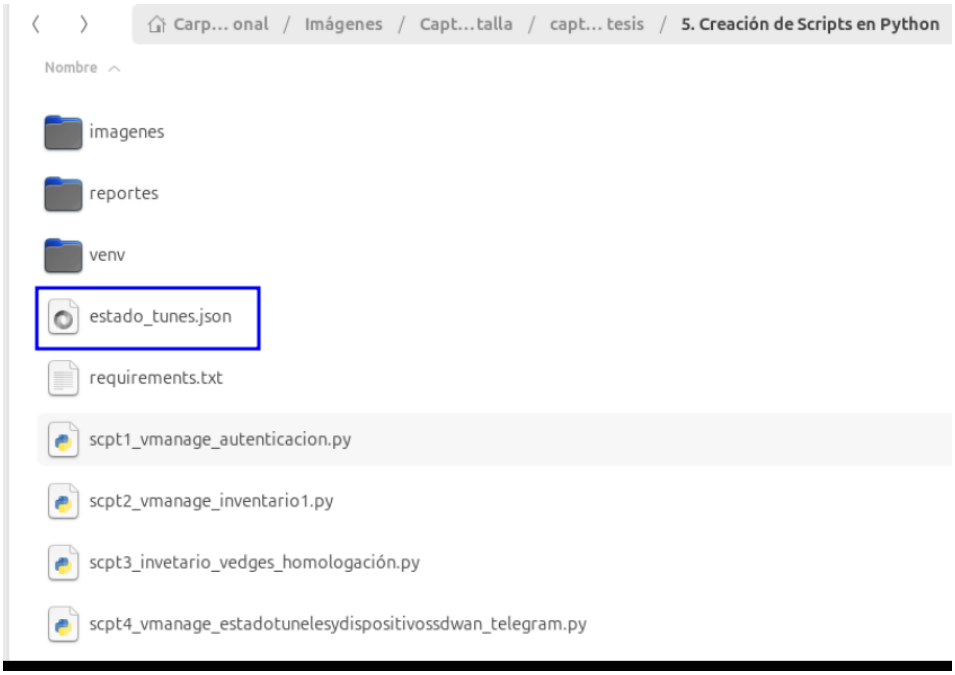


## Anexo 8: Primeras pruebas de funcionamiento de Scripts




## Anexo 9: Visualización de archivo para monitoreo

En el presente anexo se presenta evidencia de la creación del archivo que almacena el estado anterior de los dispositivos, para posteriormente ser comparado con la nueva lectura en el script de monitoreo y notificaciones.



**Anexo 10: Repositorio GitHub**

A continuación, se presenta un enlace que redirige a al repositorio de GitHub, donde se encuentran almacenados los scripts de programación en lenguaje Python de las automatizaciones implementadas en la red SD-WAN, presentados para un posterior uso y análisis.

Repositorio: https://github.com/BMDaniell/Automatizacion-SD-WAN-mediante-APIs.git