Desarrollo experimental de scripts para automatización de redes de datos, usando el lenguaje de programación Python y herramientas de Open Source

Azogue Jaque, Jofre Vinicio

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Tecnología Superior en Redes y Telecomunicaciones

Trabajo de Integración Curricular, previo a la obtención del título de Tecnólogo Superior en Redes y Telecomunicaciones

Ing. Caicedo Altamirano, Fernando Sebastián

29 de febrero del 2024

Latacunga

Reporte de Verificación de Contenido

Scan details

Scan time: **February 27th, 2024 at 20:33 UTC**

Total Pages: **39**

Total Words: **9600**

Plagiarism Detection

4.9%

Types of plagiarism		Words
Identical	0.7%	64
Minor Changes	0.9%	88
Paraphrased	3.3%	314
Omitted Words	0%	0

AI Content Detection

N/A

Text coverage
- AI text
- Human text

Plagiarism Results: (16)

¿Qué son los dispositivos de Red y para qué sirven? — **1.8%**

https://www.plotandesign.com/redes/dispositivos-de-red/

Especialistas en Redes de Computadores Inicio Instalación de Redes Ma...

DISPOSITIVOS DE CONEXIÓN – Redes — **1.2%**

https://redesietsb.school.blog/dispositivos-de-conexion/

Saltar al contenido Redes Sitio de aprendizaje de los Estudiantes de la Técnica en Sistemas ...

8 tipos de dispositivos de red y sus características — **0.7%**

https://blog.conzultek.com/dispositivos-de-red-caracteristicas

Menu Soluciones Empresariales Servicios Servicios en la nube Microsoft 365 Virtualizac...

01 Netmiko – La Herramienta de Automatización para Dispositivos de Re... — **0.5%**

https://community.cisco.com/t5/blogs-general/01-netmiko-la-herramienta-de-automatizaci%c3%b3n-para-dis...

...

.....

Ing. Caicedo Altamirano, Fernando Sebastián

Director

C.C: 1803935020

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Tecnología Superior en Redes y Telecomunicaciones

Certificación

Certifico que el trabajo de integración curricular: "**Desarrollo experimental de scripts para automatización de redes de datos, usando el lenguaje de programación Python y herramientas de Open Source**" fue realizado por el señor **Azogue Jaque, Jofre Vinicio,** el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizada en su totalidad por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que se lo sustente públicamente.

Latacunga, 29 de Febrero del 2024

Firma

Ing. Caicedo Altamirano, Fernando Sebastián

Director

C.C: 1803935020

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Tecnología Superior en Redes y Telecomunicaciones

Responsabilidad de Autoría

Yo, **Azogue Jaque, Jofre Vinicio**, con cédula de ciudadanía n°050389406-5, declaro que el contenido, ideas y criterios del trabajo de integración curricular: **Desarrollo experimental de scripts para automatización de redes de datos, usando el lenguaje de programación Python y herramientas de Open Source** es de mi autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 29 de Febrero del 2024

Firma

.....

Azogue Jaque, Jofre Vinicio

C.C: 0503894065

Departamento de Eléctrica, Electrónica y Telecomunicaciones

Carrera de Tecnología Superior en Redes y Telecomunicaciones

Autorización de Publicación

Yo **Azogue Jaque, Jofre Vinicio**, con cédula de ciudadanía n°0503894065, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de integración curricular: **Desarrollo experimental de scripts para automatización de redes de datos, usando el lenguaje de programación Python y herramientas de Open Source** en el Repositorio Institucional, cuyo contenido, ideas y criterios son de mi responsabilidad.

Latacunga, 29 de Febrero del 2024

Firma

.....

Azogue Jaque, Jofre Vinicio

C.C: 0503894065

Dedicatoria

El presente trabajo de titulación se lo dedico a mi madre y a mis hermanos, quienes han sido pilar fundamental e inquebrantable apoyo, contribuyendo de manera crucial para alcanzar este objetivo en mi vida. Agradezco a Dios por dotarme de salud y sabiduría, permitiéndome absorber valiosos aprendizajes a lo largo de este proceso.

Para mí, este logro adquiere un significado extraordinario en mi vida, brindándome la oportunidad de concretar un objetivo anhelado en mi carrera profesional.

AZOGUE JAQUE, JOFRE VINICIO

Agradecimiento

Iniciare agradeciendo a mi madre y mis hermanos, cuyo inquebrantable esfuerzo y apoyo fueron fundamentales para el éxito de mi carrera universitaria. Fueron ellos quienes me brindaron el apoyo necesario para superar momentos difíciles y desafíos aparentemente insuperables.

A la Universidad de las Fuerzas Armadas ESPE sede Latacunga, por abrir sus puertas y poder adquirir los conocimientos y aprendizaje de todos mis docentes, a quienes con mucho gusto les agradezco por impartir los conocimientos en el aula. Ante todo, quiero expresar un agradecimiento a mi tutor de trabajo de titulación al Ing. Fernando Caicedo Altamirano por su esfuerzo, paciencia, y motivación que me ha instruido para alcanzar la meta propuesta.

Muy agradecido de una u otra manera aquellos compañeros que fueron el apoyo incondicional por su amistad sincera.

AZOGUE JAQUE, JOFRE VINICIO

ÍNDICE DE CONTENIDOS

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

Resumen

Este trabajo de investigación pretende aportar soluciones a los retos a los que se enfrentan los departamentos de Tecnologías de la Información y la Comunicación (TIC) a la hora de configurar, gestionar y asegurar la infraestructura de los sistemas de comunicación en medio del rápido crecimiento de la Industria 4.0 y la Web 4.0. La creciente demanda de servicios virtualizados en la nube, el consumo masivo de ancho de banda y los diversos servicios multimedia han creado grandes desafíos para garantizar un buen servicio de red, como actualizaciones y mantenimiento físico y lógico, así como vulnerabilidades como ciberataques, especialmente de denegación de servicio (DoS). Centrarse en la automatización de redes utilizando la biblioteca Paramiko con el lenguaje de programación Python para acelerar la reconfiguración de los dispositivos de red tras una pérdida de servicio. Demostrar la eficacia de los scripts automatizados para reducir significativamente los tiempos de recuperación en comparación con los métodos manuales de reconfiguración y archivo de copias de seguridad. Se presenta una metodología en tres fases, incluyendo la topología de red para el desarrollo de scripts con Paramiko y la ejecución a través del protocolo SSH. Centrándonos con la implementación en una red compuesta por dispositivos de las marcas Mikrotik, Cisco y HPE, mostrando notables mejoras en los tiempos de recuperación y optimización del sistema. Los resultados indican una reducción del 99% en el tiempo de inactividad, lo que subraya el potencial de la automatización de la red para mejorar la eficiencia de los sistemas de comunicación

Palabras Claves: Automatización de redes, Python, Librería Paramiko, VS Code.

Abstract

This research paper aims to provide solutions to the challenges faced by Information and Communication Technology (ICT) departments in configuring, managing and securing the infrastructure of communication systems amidst the rapid growth of Industry 4.0 and Web 4.0. The growing demand for virtualised cloud services, massive bandwidth consumption and various multimedia services have created major challenges to ensure good network service such as physical and logical upgrades and maintenance, as well as vulnerabilities such as cyber-attacks, especially denial of service (DoS). Focusing on network automation using the Paramiko library with Python programming language to speed up the reconfiguration of network devices after a loss of service. Demonstrating the effectiveness of automated scripts in significantly reducing recovery times compared to manual reconfiguration and backup archiving methods. A three-phase methodology is presented, including network topology for script development with Paramiko and execution via the SSH protocol. Focusing with the implementation on a network composed of Mikrotik, Cisco and HPE branded devices, showing remarkable improvements in recovery times and system optimisation. Results indicate a 99% reduction in downtime, underlining the potential of network automation to improve the efficiency of communication systems.

Key Words: Network Automation, Python, Library Paramiko, VS Code.

Capítulo I

Introducción

Tema

DESARROLLO EXPERIMENTAL DE SCRIPTS PARA AUTOMATIZACIÓN DE REDES DE DATOS, USANDO EL LENGUAJE DE PROGRAMACIÓN PYTHON Y HERRAMIENTAS DE OPEN SOURCE.

Antecedentes

En los últimos años, las redes han sido la principal influencia a factor de desarrollo en medianas y grandes empresas u organizaciones debido al aumento constante de la complejidad de las redes y la demanda de servicios más rápidos y eficientes. La configuración manual y la gestión de redes tradicionales resultan tediosas, propensas a errores y consumen tiempo. (Islami et al, 2020)

Es por esta razón que muchos investigadores a enfocado a desarrollar trabajos de formas automatizadas para la gestión de trabajos con nuevos métodos más sencillos y eficientes.

Según la investigación de Mazin bajo el título “Performance Analysis on Network Automation Interaction with Network Devices Using Python”, describieron las diferencias entre la configuración manual y la automatización a través de secuencias de comandos. Se emplearon 36 dispositivos de red Cisco, abarcando diversas versiones de imagen IOS, como parte de la investigación. El estudio descubrió que la implementación con la automatización para la configuración a los dispositivos mejoraba la eficiencia y reducía el tiempo necesario. Los resultados de la investigación muestran el tiempo empleado cuando se utiliza la automatización con un tiempo de (120 segundos), a diferencia con la configuración manual que fue de (5797 segundos), dando que la automatización mediante scripts mejora con un 99% siendo más rápido y eficiente (2020).

El trabajo de investigación por Fahmi con el tema “Otomatisasi Jaringan Menggunakan Script Python Untuk Penyediaan Konfigurasi Internet Dan Manajemen Mikrotik”, explicaron la diferencia de la implementación de la automatización de redes mediante el uso de las bibliotecas Paramiko y Netmiko. Logrando la configuración a 10 dispositivos, incluyendo Router Mikrotik y Switches Cisco, empleando protocolos como SNMP, SSH y YANG. Los resultados de la investigación mencionaron que la librería Paramiko tiene un mejor rendimiento en el proceso de automatización mediane script para redes con un tiempo de 25 segundos en comparación con Netmiko. Concluyendo que el tiempo de proceso para enviar datos de scripts a los dispositivos Routers utilizando la biblioteca Netmiko es 4,14 veces más lento que cuando se utiliza Paramiko (2021).

La automatización con su configuración y gestión de redes se ha destacado como una solución eficiente y precisa, como lo han demostrado los autores anteriormente mencionados por el estudio positivo y herramientas que han solucionado los problemas de red y enfatizando la versatilidad de Python y sus bibliotecas especializadas.

Ha pesar de existir varios trabajos de investigación existe todavía una gran deficiencia en estudios que comprueben el rendimiento a equipos físicos y evalúen exhaustivamente la vulnerabilidad de estos sistemas. No obstante, los estudios revisados han destacado la eficiencia y precisión de las herramientas utilizadas para la automatización, evidenciando la capacidad de detectar, solucionar problemas y agilizar las operaciones en redes. Es crucial continuar avanzando en investigaciones que aborden esta brecha, brindando un enfoque más completo que valide la efectividad de la automatización en la práctica, su adaptabilidad a equipos físicos y su capacidad para mitigar vulnerabilidades en sistemas reales.

Planteamiento del problema

La gestión en las redes surgió a mediados de la década de los 90 y 2000, cuando la complejidad de las infraestructuras de red comenzó a aumentar exponencialmente con la expansión de dispositivos y servicios en línea. La falta de herramientas accesibles y

costeables para la automatización de redes generó dificultades en la escalabilidad, mantenimiento y seguridad de las infraestructuras (Ojeda, 2021).

Con la complejidad creciente de las infraestructuras de redes de datos, evidente desde mediados de la década del 2000, ha planteado desafíos significativos en su gestión y mantenimiento. El incremento exponencial en la cantidad de dispositivos conectados, la diversificación de servicios en línea y la demanda de una interconexión más amplia entre sistemas han intensificado la dificultad inherente a la gestión manual de estas redes.

La falta de herramientas efectivas de automatización ha generado retrasos en las actualizaciones críticas de la red, incrementando así las vulnerabilidades de seguridad. Además, la propensión a errores humanos ha resultado en interrupciones del servicio, impactando directamente la productividad operativa y generando costos adicionales de mantenimiento. Estas deficiencias han agudizado la capacidad de adaptación frente a las demandas dinámicas de las infraestructuras de red, limitando la capacidad de respuesta ante cambios y dificultando la optimización general del rendimiento de la red.

Al no solucionarse el problema presentado anteriormente, podría resultar en un aumento alarmante de vulnerabilidades de seguridad, exponiendo las redes a ataques cibernéticos y comprometiendo la integridad de los datos críticos. Con una carencia de adaptabilidad tecnológica limitada a la escalabilidad de las infraestructuras, dejando a las organizaciones rezagadas en un entorno competitivo en constante evolución.

Por lo expuesto, es necesario el desarrollo experimental de scripts para automatizar las redes utilizando Python. El uso de herramientas de código abierto emerge como un camino prometedor hacia la resolución de estas problemáticas. La capacidad de crear soluciones flexibles y adaptativas a través de la programación brinda una oportunidad única para superar las limitaciones de la gestión manual.

Justificación

Mediante el entorno empresarial altamente dinámico y tecnológicamente impulsado, la automatización de procesos se ha convertido en un pilar fundamental para la eficiencia y la seguridad de las operaciones. Ante la creciente complejidad de las infraestructuras de red, se hace imperativo abordar las deficiencias actuales en la gestión y actualización de estos sistemas

Al elaborar los scripts mediante herramientas de código abierto ofrece flexibilidad y escalabilidad, que valida su efectividad en entornos diversos para la automatización. La experimentación propuesta se centrará en la implementación y evaluación de estos scripts, priorizando la medición del tiempo de ejecución y la eficiencia en diferentes equipos de redes.

Los resultados tendrán un gran valor, ya que les permitirá estudiar y comprender mejor cómo funcionan los equipos en los entornos de red, que suelen ser complejos. La exactitud, velocidad y capacidad de procesamiento de estos scripts potenciarán la investigación al agilizar la recopilación, análisis y modelado de grandes cantidades de datos de forma más eficaz y detallada.

La importancia de los scripts para la automatización de redes utilizando Python y herramientas de código abierto representa un avance esencial en la gestión y optimización de una red. Por lo que sería una mejoraría en la eficiencia operativa y la seguridad de las redes, demostrando un compromiso con la innovación tecnológica de estos scripts para adaptarse y evolucionar frente a las demandas cambiantes del entorno tecnológico.

Objetivos

General

- Desarrollar scripts experimentales para automatización de redes de datos, usando el lenguaje de programación Python y herramientas de Open Source.

Específicos

- Analizar los requerimientos técnicos y herramientas para automatización de redes de datos para seleccionar las que serán utilizadas en el proyecto.
- Implementar una red de datos con características y protocolos de una red corporativa.
- Desarrollar scripts mediante el lenguaje de programación Python para aplicar configuraciones a los equipos de red.

Alcance

El presente trabajo de titulación se centra en la automatización de redes utilizando las bibliotecas Telnetlib, Paramiko, Netmiko y Napalm con el lenguaje de programación Python para acelerar la reconfiguración de los dispositivos de red tras una pérdida de servicio ocupando la herramienta Visual Studio Code con él propósito de elaborar los scripts dentro de los mismos que contendrán las configuraciones para el restablecimiento de la red. Llevando a cabo una exhaustiva investigación y la medición del tiempo de los diferentes tipos de configuración en distintos dispositivos de red, evaluando su comportamiento a los scripts desarrollados. Brindando soluciones eficaz y precisa para agilizar y optimizar estas tareas, aportando un enfoque práctico para la gestión más eficiente de una red.

Capítulo II

Marco Teórico

Redes de Datos

Las redes constituyen sistemas esenciales que facilitan la comunicación y el intercambio de recursos entre múltiples dispositivos. Estas redes, que pueden adoptar tanto configuraciones alámbricas como inalámbricas, presentan diversas formas y dimensiones. Su funcionalidad abarca la posibilidad de que los dispositivos accedan a Internet, compartan archivos, impriman documentos, y ofrecen una amplia gama de capacidades adicionales. (Computer Hope, 2023)

Tipos de Redes

En el ámbito de las redes informáticas, se reconocen distintos arquetipos que facilitan la interconexión de dispositivos, por lo que han posibilitado la comunicación en distintas áreas geográfica utilizando protocolos y tecnologías especializadas. Cada tipo de red representa características específicas, adaptándose a distintos contextos y necesidades tecnológicas.(Petryschuk, 2021)

- **PAN**

Se trata de una Red de Área Personal (PAN) que interconecta dispositivos individuales, tales como teléfonos celulares, computadoras portátiles o impresoras, ya sea mediante conexión alámbrica o inalámbrica, sin necesidad de acceso a Internet. Además, una PAN también puede conectarse a una red mayor a través de un dispositivo de puerta de enlace. (Bourgeois, 2016)

- **LAN**

La Red de Área Local (LAN) establece la conexión entre dispositivos ubicados dentro de un espacio restringido, como una vivienda, una oficina o una institución educativa. Su infraestructura puede emplear tanto cables como tecnología Wi-Fi, posibilitando a los usuarios compartir recursos como impresoras, archivos y acceso a Internet. La configuración y tamaño

de una LAN varían según la topología de la red, y un ejemplo ilustrativo sería la red de computadoras en una biblioteca. (Bourgeois, 2016)

- **CAN**

Con la Red de Área de Campus (CAN) establece la conexión entre dispositivos distribuidos en múltiples edificios dentro de un campus, como una universidad, un hospital o una base militar. Su implementación puede hacer uso de Ethernet u otras tecnologías, brindando una comunicación confiable y de alta velocidad. Con la CAN tiene la capacidad de integrarse con otras redes, como una WAN o Internet. (Bourgeois, 2016)

- **MAN**

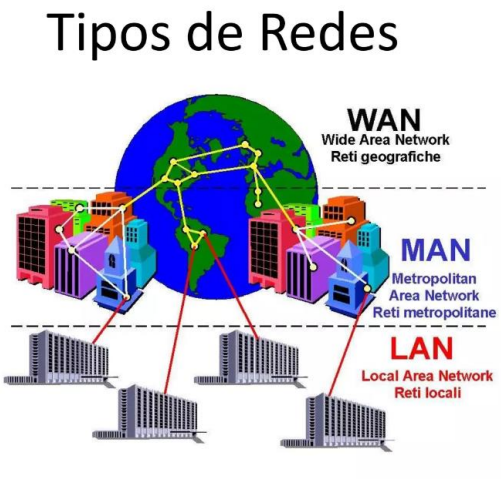
En la Red de Área Metropolitana (MAN) establece la conexión entre dispositivos ubicados en una ciudad, un pueblo o un área metropolitana. Su implementación puede hacer uso de cables de fibra óptica u otras tecnologías, proporcionando una comunicación de velocidad media con un coste moderado. La red MAN tiene la capacidad de integrarse con otras redes, como una WAN o a Internet. (Bourgeois, 2016)

- **WAN**

Con la Red de Área Extensa (WAN) establece la conexión entre dispositivos dispersos a lo largo de grandes distancias geográficas, abarcando desde un país hasta un continente o incluso a nivel mundial. Con su infraestructura puede emplear tecnologías como satélites, microondas u otros medios, brindando una comunicación con velocidades relativamente bajas y costos elevados. La Red WAN tiene la capacidad de integrarse con otras redes, tales como LAN o MAN. (Bourgeois, 2016)

Figura 1

Clasificación de los tipos de redes existentes



Nota. Tipos de redes que se maneja en la actualidad. Tomado de (Marrón, 2015)

Topologías

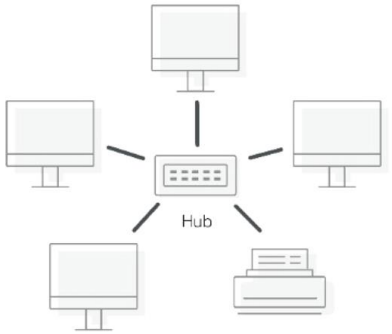
Las topologías de red son las formas de organizar los elementos y conexiones de una red de comunicaciones. Pueden ser físicas o lógicas, según muestren la colocación real de dispositivos y cables, o el flujo de datos y el encaminamiento de la información. Algunos tipos comunes de topologías de red son bus, anillo, estrella, malla y árbol. (Multicloud, 2018)

Topología en Estrella

La topología en estrella es una estructura de red común en la que cada nodo de la red se conecta a un dispositivo central, como un conmutador, concentrador o punto de acceso inalámbrico. Las topologías en estrella son comunes en redes de área local, presentan desafíos, como la dependencia crítica del funcionamiento del concentrador central y la limitación en el número de nodos que se pueden agregar debido a la capacidad del concentrador central. (Zenarmor, 2024)

Figura 2

Topología en estrella



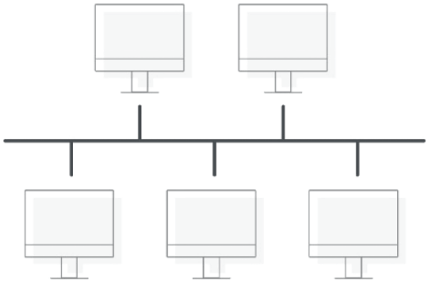
Nota. Topología en estrella: nodos conectados a central. Tomado de (Zenarmor, 2024)

Topología de Bus

La topología de bus implica la conexión de nodos en línea, donde toda la red se visualiza como un único cable. Esta topología facilita la adición, reemplazo o eliminación de dispositivos sin afectar a otros; presenta el inconveniente de que, si se rompe el cable, se desactiva toda la red. (Zenarmor, 2024)

Figura 3

Topología de Bus



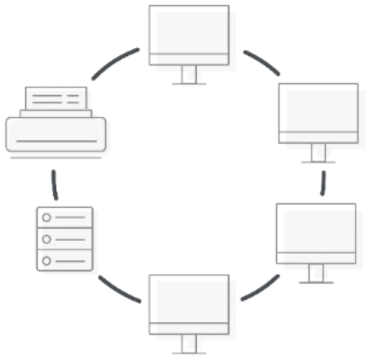
Nota. Topología en bus: nodos conectados a línea central. Tomado de (Zenarmor, 2024)

Topología en Anillo

La topología en anillo es otro modelo de red en el cual los paquetes se transmiten de un equipo a otro en un circuito cerrado. En este sistema, cada computadora examina los paquetes para determinar si están destinados a ella; en caso contrario, el paquete se envía a la siguiente computadora en el anillo. (Zenarmor, 2024)

Figura 4

Topología en anillo



Nota. Topología en anillo: nodos conectados en círculo cerrado. Tomado de (Zenarmor, 2024)

Topología de Malla

La topología de malla implica que cada nodo está interconectado con todos los demás nodos, permitiéndole enviar, recibir y retransmitir datos. Puede ser una red de malla completa o parcialmente vinculada, y no requiere un diseño físico específico, aunque cada nodo debe tener múltiples conexiones con otros nodos. A pesar de estas limitaciones, la topología de malla se vuelve más práctica con las redes WiFi. (Zenarmor, 2024)

Figura 5

Topología en malla

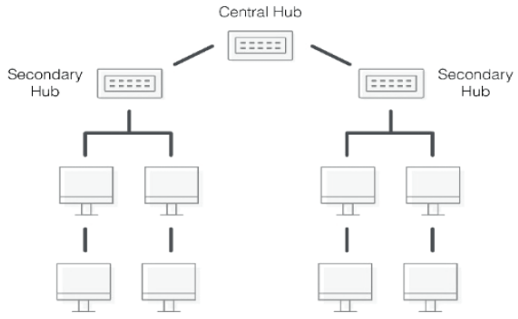
Nota. Topología en malla: nodos interconectados de manera redundante. Tomado de (Zenarmor, 2024)

Topología de árbol

La topología de árbol se considera una extensión de la topología de bus, representando una configuración híbrida que fusiona elementos de las topologías de estrella y bus. El nombre de esta topología se deriva de la forma en que los árboles organizan sus ramas. Esta topología se utiliza comúnmente en dispositivos en cascada, como en el caso de repetidores con varios puertos. Debido a su alta escalabilidad y flexibilidad, las topologías de árbol son populares en redes de área amplia (WAN). (Zenarmor, 2024)

Figura 6

Topología de árbol



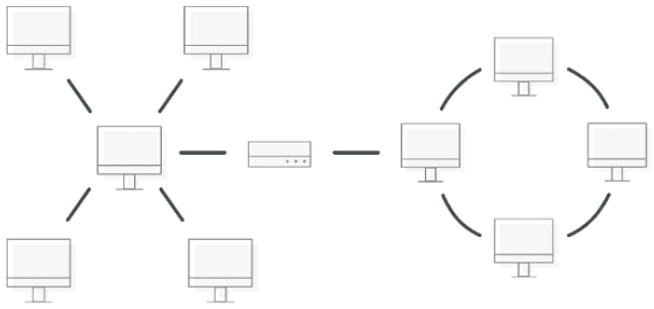
Nota. Topología de árbol: nodos jerárquicos conectados en estructura de árbol. Tomado de (Zenarmor, 2024)

Topología Hibrida

La arquitectura de red híbrida, integra elementos de diversas topologías, adaptándose a los requisitos específicos de la red. En la creación de configuraciones de este tipo suelen emplear múltiples topologías, fusionando las ventajas de cada una, siendo el anillo, estrella y el bus. Esta topología presenta desafíos y complicaciones potenciales, como una arquitectura complicada que dificulta la resolución de problemas y puede requerir considerables recursos para su establecimiento y mantenimiento, dependiendo de la configuración. A pesar de estas dificultades, la topología híbrida ofrece ventajas significativas al combinar las fortalezas de distintas topologías, resultando altamente adaptable y escalable cuando se implementa de manera adecuada. (Zenarmor, 2024)

Figura 7

Topología Hibrida



Nota. Topología hibrida: combinación de diferentes estructuras de red. Tomado de (Zenarmor, 2024)

Infraestructura de Red

La infraestructura de una red es el conjunto de sistemas que permite la comunicación entre la informática y el almacenamiento, las aplicaciones y/o los usuarios. Puede ser física o

lógica, dependiendo de si muestra la colocación real de dispositivos y cables, o el flujo de datos y el enrutamiento de la información. (Cisco, 2021) (VMware, 2023)

La infraestructura de una red puede ayudar a los administradores de red a optimizar el rendimiento, solucionar problemas y planificar el crecimiento futuro. Algunos ejemplos de infraestructura de una red son los enrutadores, los conmutadores, los cortafuegos y los recursos de software de red. (Rouse, 2023) (Rajdojr1o, 2022)

Figura 8

Diagrama de la Infraestructura de la red

Nota. Diagrama como está formado una infraestructura de red. Tomado de (Cisco, 2021)

Dispositivos de red

Diversos dispositivos de red desempeñan un papel crucial en el establecimiento y funcionamiento de una red, algunos destinados a facilitar conexiones, mientras que otros actúan como mejoradores del rendimiento. Esta amplia variedad de dispositivos contribuye significativamente a garantizar la transferencia segura de información en una red, adaptándose a las necesidades evolutivas de las organizaciones a lo largo del tiempo. (Schrader, 2023)

Router

Son dispositivos fundamentales en redes, establecen la conexión entre dos o más redes. Su aplicación común incluye la vinculación de una red doméstica u oficina (LAN) con Internet (WAN). La conexión LAN puede ser inalámbrica mediante Wi-Fi, convirtiendo al router

en un dispositivo inalámbrico. Se les conoce como puntos de acceso inalámbrico (WAP), facilitando la conectividad inalámbrica de dispositivos en la red. (Kanade, 2022)

Hubs, Bridges, and Switches

Son dispositivos de conectividad que posibilitan la conexión de varios dispositivos al enrutador y facilitan la transferencia de datos a todos los dispositivos en una red. Los switches, es un dispositivo multipuerto que mejora la eficiencia de la red. Los switches mantienen información de enrutamiento limitada sobre los nodos de la red interna, lo que permite conexiones a sistemas como concentradores o enrutadores. (Schrader, 2023)

Los bridges, actúan como dispositivos que conectan dos o más hosts o segmentos de red almacenando y transmitiendo tramas entre ellos. Estos dispositivos emplean direcciones MAC de hardware para llevar a cabo la transferencia de tramas y tienen la capacidad de reenviar o bloquear datos según las direcciones MAC de los dispositivos conectados a cada segmento. Los bridges tienen la capacidad de interconectar dos LAN físicas, creando así una LAN lógica más extensa. Su función principal es facilitar la comunicación eficiente entre diferentes partes de una red al dirigir y gestionar el tráfico de manera adecuada. (Schrader, 2023)

Los hubs sirven para conectar varios dispositivos de red de computadoras, operando exclusivamente en la capa física del modelo OSI. Estos dispositivos carecen de funciones de filtrado o direccionamiento de paquetes, y en su lugar, envían paquetes de datos a todos los dispositivos conectados. Un hubs también cumple el rol de repetidor, amplificando señales que pueden degradarse al viajar distancias extensas a través de cables. (Schrader, 2023)

Tarjeta de Interfaz de Red

La Tarjeta de Interfaz de Red (NIC) es un componente de hardware instalado en una computadora para posibilitar su conexión a una red. Esta unidad suele adoptar la forma de una placa de circuito o chip. En la mayoría de las computadoras modernas, las NIC están

integradas directamente en las placas base, proporcionando conectividad de red de manera inherente. (Schrader, 2023)

Firewall

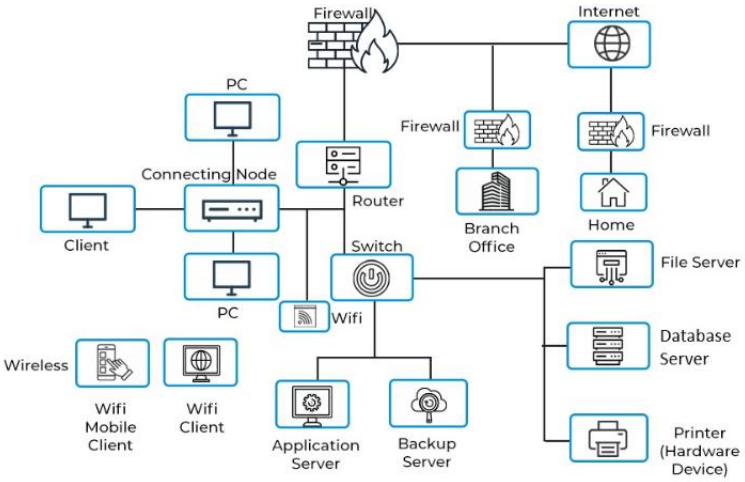
El firewall es un componente, ya sea de hardware o software, que se sitúa entre una computadora y el resto de la red, actuando como una barrera protectora contra posibles ataques o intrusiones de piratas informáticos. Su función principal de un firewall es permitir el tráfico de conexiones autorizadas, así como la transferencia de datos como correos electrónicos o páginas web legítimas, mientras bloquea de manera efectiva cualquier conexión no autorizada dirigida a una computadora o la propia LAN. (Kanade, 2022)

Puntos de Acceso Inalámbricos

Los Puntos de Acceso Inalámbricos (WAP), diseñados para establecer una Red de Área Local Inalámbrica (WLAN). Estos dispositivos funcionan de manera independiente y están equipados con una antena, un transmisor y un adaptador integrados. Operan en el modo de red de infraestructura inalámbrica, facilitando un punto de conexión entre las WLAN y una LAN Ethernet con cableado. Permitiendo la expansión de la red para admitir clientes adicionales en una empresa o en lugares con abundancia de personas. (Schrader, 2023)

Figura 9

Equipos que conforman en una red



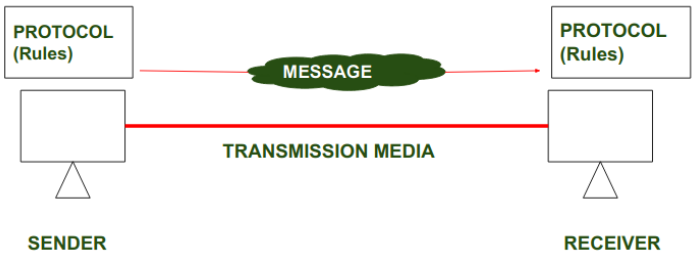
Nota. Dispositivos y elementos que conforman en una arquitectura de red. Tomado de (Kanade, 2022)

Protocolos

Un protocolo se caracteriza como un sistema o conjunto de procesos que simplifica la intercomunicación entre dispositivos en la red, específicamente a través de Internet. Para que la comunicación sea efectiva, es esencial que dos dispositivos sean conformes al mismo protocolo; en caso contrario, se hace necesario el empleo de una puerta de enlace que realice la traducción necesaria para posibilitar la comunicación. (Keary & Kiran, 2023)

Figura 10

Protocolo de comunicación



Nota. Diagrama de comunicación entre un emisor y el receptor. Tomado de (jagroopofficial , 2023)

Protocolo Telnet

Telnet es un protocolo de red que permite a los usuarios acceder y controlar ordenadores o servicios remotos mediante comandos basados en texto. Se basa en la idea de un terminal virtual que puede comunicarse con cualquier otro dispositivo. Telnet utiliza TCP como protocolo de capa de transporte y el puerto 23 como puerto por defecto. Telnet es uno de los protocolos de Internet más antiguos, pero no es seguro ni fiable, por lo que a menudo se sustituye por otras alternativas. (IBM, 2023)

Protocolo Secure Shell SSH

SSH es un protocolo de administración remota que habilita a los usuarios para acceder y supervisar sus servidores a través de internet. Desarrollado con el propósito de conectarse de forma segura entre el usuario y el servidor sustituyendo a Telnet. Este protocolo garantiza la encriptación de toda la información y proporciona un mecanismo de autenticación para los usuarios remotos. SSH puede utilizarse para el inicio de sesión remoto, la transferencia de archivos y la ejecución de comandos. (SSH, 1996)

Protocolo de Enrutamiento

El protocolo de enrutamiento desempeña un papel esencial en las redes de comunicación al dirigir eficientemente el flujo de datos entre dispositivos. Este sistema establece normas y algoritmos que permiten a los routers tomar decisiones informadas sobre la ruta más óptima para enviar paquetes desde la fuente hasta el destino. La optimización de la ruta se basa en consideraciones como la disponibilidad de enlaces, la congestión de la red y la velocidad de transmisión. Diversos protocolos, como OSPF, RIP o BGP, han sido diseñados para abordar distintos escenarios y requisitos de red, contribuyendo a la solidez y eficacia del proceso de enrutamiento en diversos entornos. (Bradley & Michael Barton , 2021) (Keary, Types of Routing Protocols – The Ultimate Guide, 2023)

Protocolo de Switching

El protocolo de switching desempeña un papel fundamental en la administración eficiente del tráfico de datos en entornos de redes de comunicación. Su función consiste en dirigir el flujo de información entre dispositivos como switches y bridges, con el propósito de optimizar la conectividad y reducir los tiempos de latencia. Mediante la aplicación de algoritmos y normativas específicas, este protocolo determina la manera en que los datos son transferidos desde su origen hasta su destino, teniendo en cuenta aspectos como la dirección MAC, el estado de los puertos y la carga de la red. Se emplean distintos tipos de switching, como el de

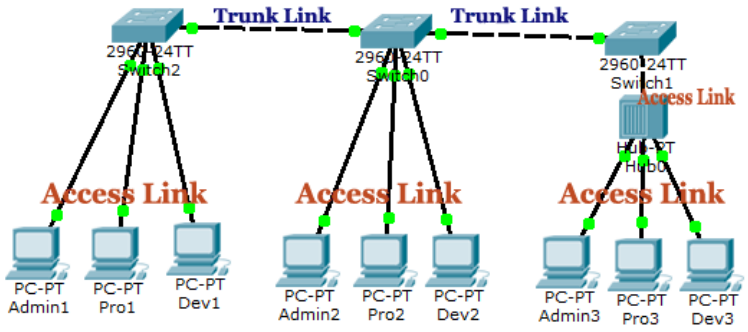
capa 2 y el de capa 3, para adaptarse a diversos escenarios y requisitos de red, contribuyendo así a la eficacia y rendimiento global del proceso de conmutación de datos. (Rodríguez, 2020)

Protocolo de registro de VLAN

El protocolo de Virtual LANs (VLANs) juega un papel crucial en la gestión eficiente de redes al dividir físicamente una red en subredes virtuales lógicas. Lo que posibilita la agrupación de dispositivos con características similares, independientemente de su ubicación física en la red. Las VLANs facilitan la organización y el control del tráfico al restringir la comunicación a dispositivos dentro de la misma VLAN, mejorando así la seguridad y el rendimiento de la red. (ComputerNetworkingNotes, 2023)

Figura 11

Switching con Vlans



Nota. Topología de switching con Vlans en Packet Trace. Tomado de (ComputerNetworkingNotes, 2023)

Automatización de Redes

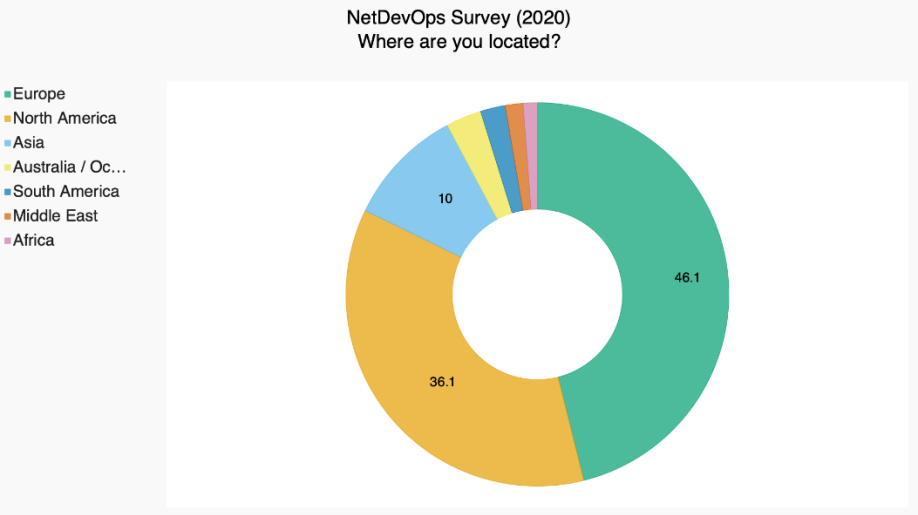
La automatización de redes se define como una metodología que sustituye los procesos manuales, desarrollados por seres humanos, mediante el uso de software. Este software se encarga de llevar a cabo tareas como configuración, administración, seguridad, pruebas, implementación y operaciones en dispositivos físicos y virtuales dentro de una red. Esta práctica puede ser implementada en diversos tipos de redes, incluyendo áreas locales, áreas extendidas, centros de datos, entornos en la nube y redes inalámbricas. Actualmente, muchas

empresas, centros de datos y proveedores de redes incorporan la automatización de redes en sus operaciones para gestionar y evitar procedimientos repetitivos, reducir costos operativos, prevenir errores humanos y, principalmente, mejorar significativamente el rendimiento, la disponibilidad y la confiabilidad de la red. (WMware, 2024) (Froehlich & Scarpati, 2024) (SolarWinds, 2024)

Tomando como referencia la encuesta de Netdevevops, se ha observado a nivel global un aumento en la implementación de la automatización de redes, como se ilustra en la figura 12. Se destaca que Sudamérica se encuentra en las etapas iniciales de adoptar esta tecnología para la automatización de redes, lo que indica que el desarrollo de esta tendencia a nivel mundial lleva aproximadamente dos décadas.

Figura 12

Regiones mundiales en la automatización de redes



Nota. Encuesta a nivel mundial de la automatización de redes. Tomado de (Garros, 2020)

Python

Python es un lenguaje de programación versátil y de alto nivel conocido por su legibilidad y sencillez. Permite a los desarrolladores escribir código claro y conciso, lo que lo convierte en una opción excelente tanto para principiantes como para programadores

experimentados. La amplia biblioteca estándar de Python y su activa comunidad contribuyen a su popularidad en diversos ámbitos, como el desarrollo web, el análisis de datos, la informática científica y la automatización. (Amazon Web Services, Inc, 2024)

Figura 13

Logotipo de Python en la automatización

Nota. Logotipo de Python en la automatización de redes. Tomado de (Gupta, 2023)

Herramientas y lenguajes de Automatización

Las herramientas de automatización desempeñan un papel esencial en simplificar y optimizar la gestión de infraestructuras de red. Por lo que existen varias herramientas para la automatización de redes. Estas herramientas las cuales son Terraform, Chef, Puppet, Salt y Ansible que resultan fundamentales al otorgar a los administradores de red las capacidades necesarias para organizar, automatizar y mantener eficientemente las operaciones de red, mejorando así la eficacia y confiabilidad del entorno de red. (DNSstuff, 2022)

Ansible

Es una plataforma de automatización que aborda la implementación de aplicaciones, la gestión de configuraciones, así como el monitoreo y control de la seguridad. Ansible opera mediante el lenguaje de programación Python con el lenguaje de configuración simple basado en YAML. Destacando por ser una herramienta sin agente, elimina la necesidad de instalar software adicional en las máquinas cliente. Esta característica permite a los usuarios enviar configuraciones directamente a dispositivos de red utilizando el protocolo SSH para la comunicación remota. (Study-ccna, 2024)

Terraform

Terraform se presenta como una herramienta de software de infraestructura como código de código abierto, cuya finalidad es posibilitar la creación, modificación y mejora segura y predecible de la infraestructura. Este software emplea un lenguaje denominado HCL, permitiendo que los desarrolladores automaticen las implementaciones de infraestructura y realicen cambios de manera segura en su configuración existente. Terraform también es compatible con JSON, un formato familiar para la mayoría de los desarrolladores. La herramienta utiliza conectores denominados proveedores para facilitar la comunicación entre Terraform y las API de diversas plataformas y servicios. (Koppenhaver, 2023)

Chef

Chef se posiciona como una herramienta de gestión de configuración altamente beneficiosa cuando se busca una solución consolidada en automatización que utiliza Ruby como lenguaje de configuración, respaldada por una comunidad sólida y un historial estable de estabilidad, con una barrera de entrada relativamente baja. Su capacidad de adaptación a tareas de infraestructura complejas la distingue, destacando frente a otras herramientas más idóneos para implementaciones de menor complejidad. Este software, diseñado para la automatización del cumplimiento, la configuración y la gestión de redes y servidores, requiere la instalación de un agente en los dispositivos de red bajo su administración. (Koppenhaver, 2023) (Study-ccna, 2024)

Salt

Salt, también conocido como SaltStack, se destaca como una herramienta de gestión de infraestructuras centrada en la comunicación eficiente entre numerosos sistemas. Su motor de ejecución remota establece una red segura y bidireccional para facilitar la interacción rápida con cada sistema, independientemente del tamaño de la infraestructura. Al adoptar un enfoque declarativo y basarse en Python con YAML como lenguaje de configuración. Su flexibilidad se refleja en la capacidad de interactuar con varios lenguajes y marcos, siendo simple, extensible

y determinista. Con una amplia gama de módulos compatibles, mientras que su modo sin agente permite una ejecución eficiente mediante la comunicación SSH con el lenguaje Python. (Koppenhaver, 2023)

Puppet

Diseñado para gestionar infraestructuras grandes. Puppet adopta un lenguaje de configuración propio casi similar a Chef. Al combinar la ejecución imperativa de tareas con un lenguaje declarativo, garantiza la coherencia entre numerosos sistemas. Sus sólidas funciones de informes y supervisión ayudan a identificar y resolver rápidamente los problemas. (Koppenhaver, 2023)

Figura 14

Herramientas para la automatización de redes

Nota. Herramientas utilizadas para la automatización de la infraestructura de red. Tomado de (Koppenhaver, 2023)

Tipos de librerías en la Automatización

La amplia variedad de librerías y módulos que son ampliamente utilizados para la automatización, especialmente en el ámbito de redes y sistemas facilitan el acceso a funcionalidades ofreciendo soluciones. Asimismo, Python disponibles para una variedad de aplicaciones, abarcando desde el desarrollo web, automatización de datos hasta el machine learning (ML). (Python Software Foundation, 2024)

Telnetlib

Telnetlib es una librería de Python diseñada para facilitar la conexión a dispositivos mediante el protocolo Telnet. La función principal es proporcionar una manera sencilla de establecer conexiones Telnet e interactuar con dispositivos remotos. Se destaca la librería por su capacidad para gestionar comunicaciones a través del protocolo Telnet, permitiendo a los desarrolladores realizar conexiones eficientes y llevar a cabo operaciones en dispositivos de manera programática. (Python Software Foundation, 2024)

Netmiko

La librería Netmiko de Python, específicamente diseñada para la automatización de redes. Su función principal es simplificar la interacción con dispositivos de red a través del protocolo SSH (Secure Shell). Esta librería facilita tareas repetitivas al ofrecer una interfaz consistente y abstrayendo las complejidades asociadas con la gestión de dispositivos de diferentes fabricantes. Netmiko permite a los desarrolladores crear scripts y herramientas de automatización que son independientes de la marca del dispositivo, brindando así flexibilidad y eficiencia en entornos de red heterogéneos. (Byers, 2021)

Figura 15

Logotipo de Netmiko

Nota. Logotipo de la librería de Netmiko. Tomado de (Byers, 2021)

Paramiko

La Librería Paramiko dedicada a la implementación del protocolo SSH (Secure Shell). Proporcionando una interfaz que permita la comunicación segura y transferencia de archivos

entre una máquina local y una remota. Paramiko destaca por su utilidad en situaciones donde se requiere establecer conexiones seguras para la ejecución de comandos y transferencia de datos, siendo especialmente valiosa en el ámbito de la automatización y administración remota de sistemas. Paramiko se convierte en una herramienta esencial para desarrolladores que buscan implementar conexiones seguras y transferencias de datos cifradas en sus proyectos. (Forcier, 2024)

NAPALM

Para la librería NAPALM está diseñada para ofrecer una interfaz unificada para dispositivos de red de diferentes fabricantes. Como objetivo principal es simplificar la automatización de redes al proporcionar una capa de abstracción que oculta las complejidades específicas del proveedor. NAPALM permite a los desarrolladores interactuar con dispositivos de red de manera consistente, independientemente del fabricante, facilitando tareas como la configuración, recolección de datos y gestión de dispositivos. NAPALM se convierte en una herramienta valiosa para aquellos que buscan lograr una automatización eficiente y escalable en entornos de red diversificados. (Alvarez, 2020)

Figura 16

Logotipo de NAPALM

Nota. Logotipo de la librería de NAPALM. Tomado de (Kochar, 2021)

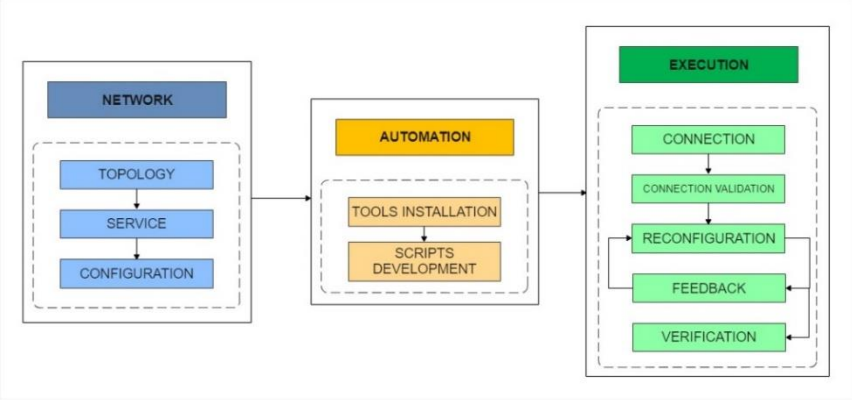
Capítulo III

Desarrollo del Tema

La metodología a utilizar se desglosa en tres etapas, como se ilustra en el esquema presentado en la figura 17. La primera etapa consiste en la creación de una topología de red LAN (Local Area Network) y WLAN (Wireless Local Area Network), la selección de los servicios y protocolos disponibles en la red y finalmente la configuración de los dispositivos. En la segunda etapa se desarrolla el proceso de automatización, que consiste en la programación de los scripts que contienen los parámetros de reconfiguración utilizando la librería Paramiko. Finalmente, en la tercera etapa de ejecución, los scripts se ejecutan a través de una conexión remota utilizando el protocolo SSH. Se aplican las reconfiguraciones correspondientes y finalmente se obtienen los datos de realimentación del proceso de configuración y la validación del correcto funcionamiento de la red.

Figura 17

Proceso de automatización con protocolo SSH utilizando la librería Paramiko



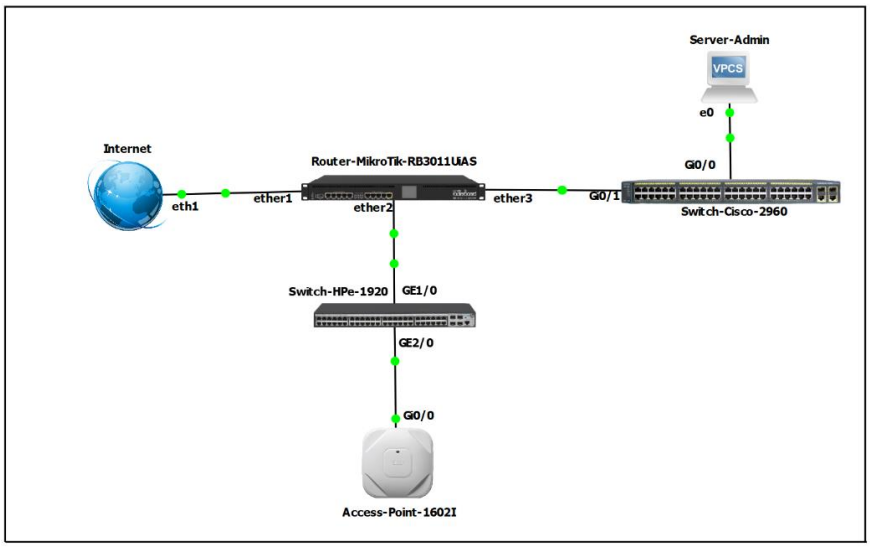
Red

La figura 18 muestra la topología de red propuesta, compuesta por un router Mikrotik RB3011UiAS, un switch Cisco Catalyst WS-C2960-48PST-L, un switch HPE OfficeConnect 1920 48G y un AP (Access Point) Cisco Aironet 1602I. Los procesos de automatización son gestionados por un ordenador con un Intel core i7 de 10ª generación con 8 GB de memoria

RAM. Estos dispositivos de red fueron seleccionados porque soportan configuración CLI y conexiones remotas vía protocolo SSH; además, al no ser de una sola marca, permiten validar el funcionamiento de la librería con dispositivos multimarca.

Figura 18

Topología de red propuesta para la investigación



Una vez establecida la topología, el siguiente paso fue determinar los protocolos y servicios a configurar en los dispositivos de red, teniendo en cuenta los protocolos básicos que debe mantener cualquier LAN empresarial.

Para el acceso a Internet se requieren rutas estáticas y NAT (Network Address Translation). También se tuvo en cuenta la segmentación de la red para reducir los dominios de difusión y la protección de la red frente a ataques internos y externos.

El router se configuró con una ruta estática predefinida y NAT (Network Address Translation) para permitir el acceso a Internet, direcciones IP en las interfaces físicas y VLAN, el servidor DHCP (Dynamic Host Configuration Protocol) habilitado y conjuntos básicos de cortafuegos para asegurar la red.

Por otro lado, los conmutadores y AP segmentan la red mediante VLAN (red de área local virtual). Se habilitaron enlaces troncales mediante el protocolo IEEE 802.1Q y puertos en

modo de acceso para la conexión de host. La red también se protegió de ataques internos mediante el protocolo de seguridad de puertos y el registro de direcciones MAC conocidas.

Se habilitaron 3 SSID diferentes en el punto de acceso, cada uno asignado a una VLAN distinta, y se activó el protocolo de seguridad WPA2 (Wi-Fi Protected Access 2) con autenticación de clave compartida PSK (Pre-Shared Key) y AES (Advanced Encryption Standard).

Automatización

La etapa de automatización consiste en el desarrollo de scripts, que contienen los comandos que permiten la reconfiguración de los equipos en la etapa de restablecimiento. La programación se realizó en la herramienta de desarrollo libre Visual Studio Code con el lenguaje de programación Python. Se decidió trabajar con la librería Paramiko, por su capacidad de fácil adaptabilidad con equipos de red en diferentes entornos de red, facilitando la comunicación segura vía SSH.

Desarrollo de Script

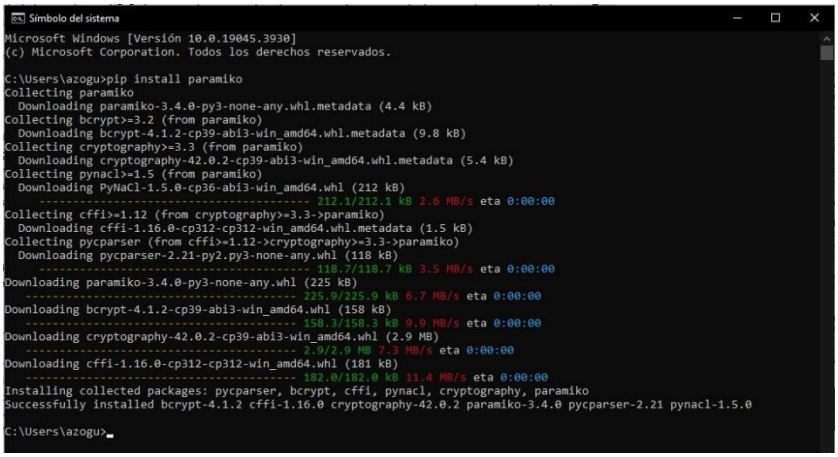
Para el desarrollo de los scripts de automatización, se analizaron entornos de desarrollo integrados (IDE) como PyCharm y Visual Studio Code (VS Code).

PyCharm, conocido por su robusto entorno de desarrollo Python y su completo conjunto de herramientas, proporciona un entorno intuitivo para escribir y depurar código (JetBrains, 2024). VS Code destaca por su flexibilidad y extensibilidad, ofreciendo un entorno ligero y personalizable que se adapta a las necesidades específicas de los desarrolladores (Extension, LMA, 2024). VS Code fue elegida como herramienta principal para el desarrollo de scripts. Esta elección se basó en la flexibilidad y extensibilidad ofrecidas, aspectos considerados esenciales para adaptarse a las necesidades específicas de automatización en dispositivos de red.

La figura 19 muestra el proceso de instalación de la librería Paramiko, que es la base para el desarrollo de los scripts ya que permite la comunicación entre el servidor y los equipos de red.

Figura 19

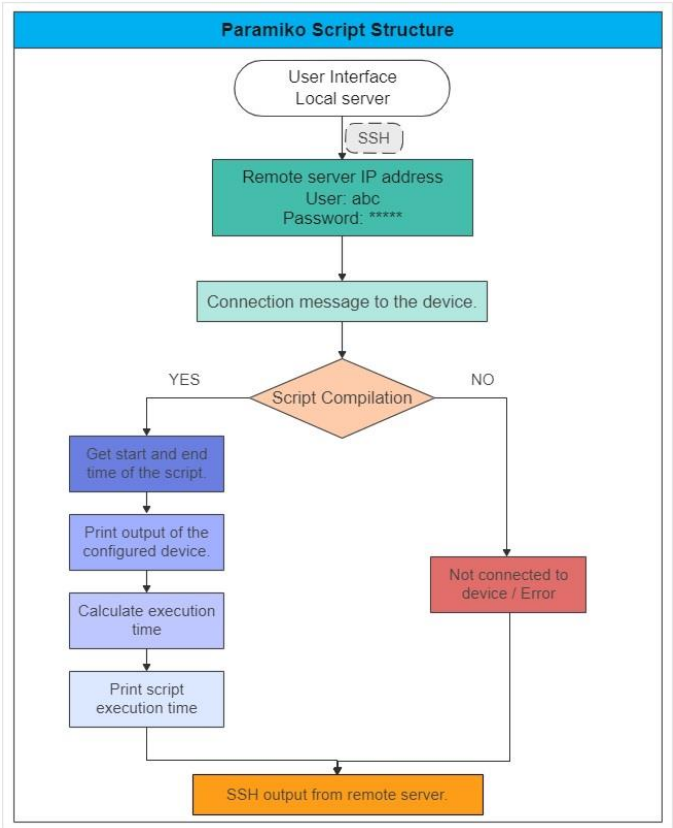
Instalación de la biblioteca Paramiko



Una vez instaladas y configuradas las herramientas de desarrollo, procedemos a la programación de los scripts, siguiendo la estructura mostrada en la figura 20.

Figura 20

Metodología para el desarrollo de Script

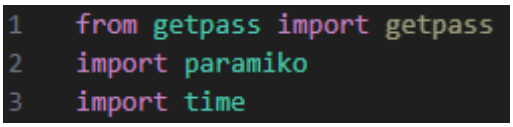


Comienza importando tres bibliotecas esenciales: getpass, paramiko y time. La librería getpass se utiliza para solicitar al usuario información sensible, como contraseñas, de forma segura, evitando que se muestren en la consola mientras se introducen.

La biblioteca Paramiko es esencial para establecer conexiones seguras a través del protocolo SSH, permitiendo la ejecución remota de comandos y la transferencia segura de archivos entre sistemas. Por último, la librería time proporciona funciones para medir y gestionar el tiempo, permitiendo introducir pausas en la ejecución del programa o medir la duración de determinadas operaciones como se muestra en la figura 21.

Figura 21

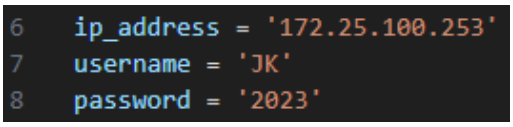
Importación de librerías (getpass, paramiko y time)



Se añaden las siguientes variables ip_address, username, y password ya que contienen información de conexión para interactuar con un dispositivo remoto vía SSH, como se muestra en la figura 22. Estos valores se utilizan junto con la biblioteca paramiko para establecer conexiones seguras.

Figura 22

Introducción de variables y credenciales



Para la comunicación con ordenadores a través de paramiko, se crea una instancia de la clase SSHClient llamada client. A continuación, se configura la política para manejar claves host desconocidas utilizando AutoAddPolicy(); esto permite añadir automáticamente nuevas claves al archivo de claves conocidas. A continuación, se establece la conexión SSH con el dispositivo remoto mediante el método connect, proporcionando la dirección ip=hostname, el

nombre de usuario username y la contraseña password. Por último, se invoca un Shell interactivo en el dispositivo remoto utilizando invoke_shell(), que facilita la ejecución de comandos y la interacción programática con el sistema a través de la conexión SSH establecida, como se muestra en la figura 23.

Figura 23

Autenticación de credenciales mediante autenticación remota con SSH

```
10  client = paramiko.SSHClient()
11  client.set_missing_host_key_policy(paramiko.AutoAddPolicy())
12  client.connect(hostname=ip_address,username=username,password=password)
13  devices_access = client.invoke_shell()
```

La instrucción print() se coloca para mostrar un mensaje informativo en la consola, como se muestra en la figura 24, indicando que la conexión del conmutador se ha establecido correctamente.

Figura 24

Mensaje de confirmación de conexión al Switch 2960

```
15  print("Connection established to Switch 2960")
```

Se introduce el código de la línea del temporizador, como se muestra en la figura 25, esto marca el inicio de la configuración para ejecutar los comandos de configuración. A la variable start_time se le asigna el valor devuelto por la función time.time(), que representa la hora actual en segundos como punto de referencia en el tiempo. Al capturar este valor al inicio de una sección de código, se crea una referencia temporal que permite medir la duración de la ejecución de ese segmento.

Figura 25

Inicio del temporizador del script

```
18  start_time = time.time()
```

Se ingresa el código devices_access para enviar una serie de instrucciones para la configuración al dispositivo. Los comandos ingresados, están destinados a la configuración de VLANs (Virtual LANs), como se muestra en la figura 26. Estos comandos son típicos en la configuración de infraestructuras de red para segmentar y organizar el tráfico en VLANs distintas.

Figura 26

Configuración al equipo mediante comandos

```
19  devices_access.send(b"configure terminal\n")
20  devices_access.send(b"vlan 11\n")
21  devices_access.send(b"name VLAN11\n")
22  devices_access.send(b"exit\n")
23  devices_access.send(b"vlan 12\n")
24  devices_access.send(b"name VLAN12\n")
25  devices_access.send(b"exit\n")
26  devices_access.send(b"vlan 13\n")
27  devices_access.send(b"name VLAN13\n")
28  devices_access.send(b"exit\n")
```

El código time.sleep(1), permitir que el dispositivo remoto procese las configuraciones previamente enviadas. Luego, se utiliza devices_access.recv(3500) para recibir datos de la salida del shell interactivo del dispositivo, capturando la respuesta. La función decode('ascii') se aplica para convertir estos datos binarios en una representación de texto ASCII legible. Finalmente, la salida decodificada se imprime en la consola con print(output.decode('ascii')), proporcionando visibilidad sobre la respuesta del dispositivo a las configuraciones. Finalizando con el cierra a la conexión SSH con client.close(), como se muestra en la figura 27.

Es imprescindible incorporar la línea de código end_time = time.time(), ya que marca el final del temporizador previamente iniciado con start_time.

Figura 27

Recepción de salida, cierre del cliente y registro del tiempo de finalización

```
42  time.sleep(1)
43  output = devices_access.recv(3500)
44  print(output.decode('ascii'))
45  client.close()
46
47  end_time = time.time()
```

La figura 28 muestra la ecuación utilizada para determinar el tiempo transcurrido para el proceso completo de ejecución del script. Este valor corresponde a la relación entre la hora de inicio y la hora de finalización de la ejecución del script.

Figura 28

Tiempo transcurrido de ejecución del script

```
56    execution_time = end_time - start_time
```

Para finalizar, se utiliza un mensaje que indica la configuración realizada por el script y el tiempo que se ha tardado en realizar la configuración, tal y como se ilustra en la figura 29.

Figura 29

Impresión del tiempo de ejecución del script a través de SSH al Switch Cisco

```
59    print(f"The run time of the Cisco PARAMIKO Switch is: {execution_time} seconds.")
```

Resultados

Se proponen tres escenarios para evaluar el proceso de restablecimiento de la red en caso de interrupción. El primero consiste en reconfigurar manualmente cada uno de los dispositivos. El segundo escenario consiste en utilizar un archivo con configuraciones de copia de seguridad que se aplican a los dispositivos, pero el proceso de conexión y carga del archivo sigue siendo manual. Finalmente, el tercer escenario consiste en que el servidor detecta una caída del sistema y ejecuta automáticamente el script que contiene los comandos de reconfiguración preestablecidos, sin intervención del administrador de red. La figura 30 muestra la red desplegada para las pruebas realizadas.

Figura 30

Automatización de equipos físicos

Configuración manual

En la reconfiguración manual de los dispositivos de red, el tiempo necesario para las configuraciones de enrutamiento, conmutación y puntos de acceso en diferentes dispositivos es considerablemente alto. El Router Mikrotik RB3011UiAS tarda 1440 segundos en configurar o restaurar el estado de operatividad. Del mismo modo, los Switches tienen tiempos significativamente largos, en el caso del Switch Cisco 2960, requiere 1680 segundos para restaurar y poner en marcha el equipo, así como el Switch HPe 1920, que tarda 1920 segundos en configurarse.

Archivos de copia de seguridad

Este método representa una alternativa más adecuada que el manual, ya que permite restablecer los dispositivos con relativa rapidez, pero sigue requiriendo la intervención del administrador, que debe acceder manualmente a la CLI de forma independiente para cada dispositivo y cargar las configuraciones preestablecidas en el archivo de copia de seguridad.

Sistema de automatización

Las figuras 31, 32, 33 y 34 muestran los tiempos de recuperación calculados automáticamente por el script para el router Mikrotik RB3011UiAS, el switch Cisco 2960, el switch HPe 1920 y el punto de acceso Cisco 1602I respectivamente.

Figura 31

Tiempo de ejecución del script al router Mikrotik RB3011UiAS

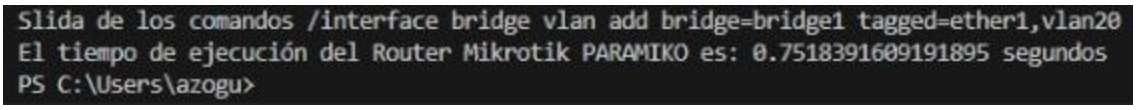


Figura 32

Tiempo de ejecución del script al Switch Cisco 2960

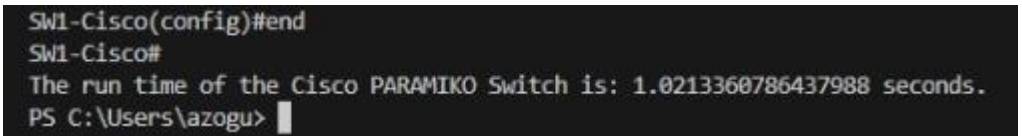


Figura 33

Tiempo de ejecución del script para el switch HPe 1920

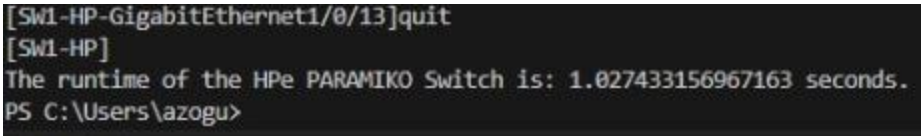
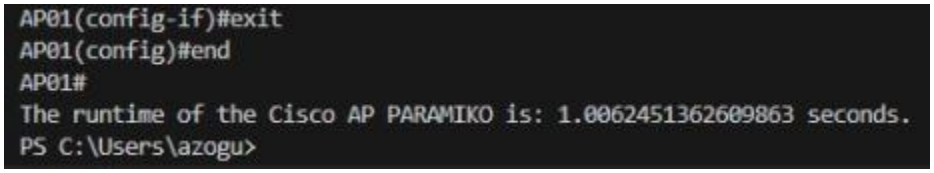


Figura 34

Tiempo de ejecución del script para el AP Cisco Aironet 1602I.



En la tabla 1 se muestran los tiempos medios obtenidos en cada uno de los escenarios descritos de forma individual para cada uno de los dispositivos que forman parte de la topología. El uso de ficheros de respaldo mejora considerablemente los tiempos de recuperación, reduciéndolos en un 90% respecto a la reconfiguración manual. Sin embargo, trabajando con un sistema automatizado, el tiempo empleado se reduce al 99%, optimizando el sistema y evitando interrupciones en los procesos administrativos y productivos; además no requiere la intervención del administrador de red. En la tabla 2 se muestra el tiempo medio transcurrido para el restablecimiento de toda la red, considerando los 4 dispositivos considerados.

Tabla 1

Tiempo de ejecución y restablecimiento de la red

Dispositivos	**Configuración Manual**	**Archivo de copia de seguridad**	**Sistema automatizado**
Mikrotik RB3011UiAS	1440 segundos	85 segundos	0.75 segundos
Cisco Switch 2960	1680 segundos	140 segundos	1.02 segundos
HPe Switch 1920	1920 segundos	175 segundos	1.03 segundos
Cisco AP 1602I	1140 segundos	125 segundos	1.01 segundos

Nota. Temporización por diferentes métodos de reinicio y configuración

Tabla 2

Tiempo total de funcionamiento y restablecimiento de la red

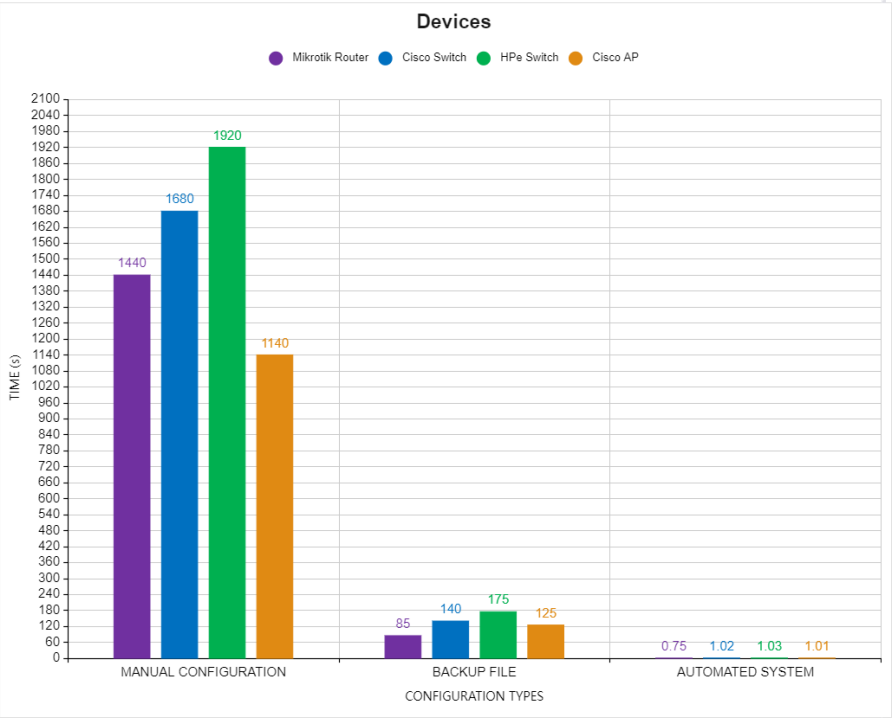
Dispositivos	Configuración Manual	Archivo de copia de seguridad	Sistema automatizado
Toda la red en servicio	6180 segundos	525 segundos	3.81 segundos

Nota. Tiempo total de los tres métodos de configuración

La figura 35 muestra las estadísticas detalladas de la restauración del servicio de red, aplicadas en tres escenarios diferentes, en los que intervienen cuatro dispositivos de red. Los datos obtenidos muestran que la automatización con script ha demostrado ser excepcionalmente rápida en las configuraciones de enrutamiento, conmutación y puntos de acceso en comparación con los métodos manuales y de copia de seguridad.

Figura 35

Comparación gráfica de los tipos de configuraciones



Capítulo IV

Conclusiones y recomendaciones

Conclusiones

- Se analizó un análisis técnico exhaustivo de diversas herramientas utilizadas en automatización de redes. La librería Paramiko fue seleccionada debido a su amplia funcionalidad y extensibilidad facilitando la automatización y gestión segura mediante conexiones SSH de manera efectiva y confiable.
- Se estableció una red de datos con equipos de enrutamiento, conmutación y red inalámbrica, con características corporativas que incluyen segmentación, medidas de seguridad y la implementación de diversos protocolos, como VLANs entre otras.
- Se llevó a cabo el desarrollo de scripts diseñados para la configuración de equipos de enrutamiento, conmutación y puntos de acceso para la red. Estos scripts se crearon para agilizar y simplificar el proceso de configuración, permitiendo una implementación rápida y eficiente de la infraestructura.

Recomendaciones

- Proponer y analizar una investigación a relleno en la parte de la automatización de redes con script, junto con software especializados, que puedan optimizar las configuraciones de la red, generando ir en creses a la par de la Web 4.0.
- Se sugiere cuando se trabaje con simuladores, entornos de red virtualizados o equipos físicos admitan la configuración mediante CLI el acceso remoto vía SSH o TELNET es fundamental para la automatización de estos equipos.
- Se recomienda trabajar con herramientas principalmente para el desarrollo del script que proporcione la facilidad, extensibilidad y adaptación a las necesidades específicas de la automatización se lo quiera realizar.

Bibliografía

Alvarez, M. (2020, Agosto 05). *NAPALM Network Automation Python: Introduction e Installation*. Retrieved Febrero 12, 2024, from Coding Networks Blog: https://codingnetworks.blog/napalm-network-automation-python-introduction-e-installation/

Amazon Web Services, Inc. (2024, Febrero 05). *¿Qué es Python?* 2024: https://aws.amazon.com/es/what-is/python/

Bourgeois, S. (2016, Octubre 06). *11 types of networks explained: VPN, LAN & more*. Retrieved Febrero 09, 2024, from Belden.com: https://www.belden.com/blogs/network-types

Bradley , M., & Michael Barton , H. J. (2021, Junio 25). *Top 5 Network Routing Protocols Explained*. Retrieved Febrero 12, 2024, from https://www.lifewire.com/top-network-routing-protocols-explained-817965

Byers, K. (2021, Septiembre 30). *Netmiko Library*. Retrieved Febrero 10, 2024, from https://pynet.twb-tech.com/blog/netmiko-python-library.html

Cisco. (2021, Marzo 08). *What Is Network Infrastructure?* Retrieved Febrero 01, 2024, from cisco.com: https://www.cisco.com/c/en/us/solutions/enterprise-networks/what-is-network-infrastructure.html

Computer Hope. (2023, Octubre 12). *Network*. Retrieved Febrero 10, 2024, from computerhope.com: https://www.computerhope.com/jargon/n/network.htm

ComputerNetworkingNotes. (2023, Septiembre 08). *VLAN Basic Concepts Explained with Examples*. Retrieved Febrero 15, 2024, from https://www.computernetworkingnotes.com/ccna-study-guide/vlan-basic-concepts-explained-with-examples.html

DataScientest. (2023, Mayo 09). *Ansible: la herramienta de automatización preferida por los DevOps*. Retrieved Febrero 04, 2024, from Datascientest.com:

https://datascientest.com/es/ansible-la-herramienta-de-automatizacion-preferida-por-los-devops

DNSstuff. (2022, Septiembre 02). *Best Network Automation Tools*. Retrieved Febrero 04, 2024, from Software Reviews, Opinions, and Tips - DNSstuff: https://www.dnsstuff.com/network-automation-tools

Extension, LMA. (2024). *Visual Studio Code Version 1.86*. Retrieved Enero 29, 2024, from visualstudio.com: https://code.visualstudio.com/

Fahmi, M., Maisyaroh, M., Komarudin, I., Faizah, S., & Fadhilah, I. (2021). Otomatisasi Jaringan Menggunakan Script Python Untuk Penyediaan Konfigurasi Internet Dan Manajemen Mikrotik. *BINA INSANI ICT JOURNAL, 8(1)*, 53. https://doi.org/https://doi.org/10.51211/biict.v8i1.1517

Forcier, J. (2024). *Welcome to paramiko! — paramiko documentation*. Retrieved Febrero 12, 2024, from https://www.paramiko.org/

Froehlich, A., & Scarpati, J. (2024). *network automation*. Retrieved Febrero 04, 2024, from techtarget.com: https://www.techtarget.com/searchnetworking/definition/network-automation

Garros, D. (2020, Octubre 31). *Netdevops-Survey*. Retrieved Febrero 05, 2024, from https://github.com/dgarros/netdevops-survey?tab=readme-ov-file

Gupta, R. (2023, Febrero 14). *Python installation with ansible, netmiko, paramiko, NAPALM*. Retrieved Febrero 2024, 07, from https://s4u.in/python-installation-with-ansible-netmiko-paramiko-napalm/

IBM. (2023, Marzo 24). *IBM documentation*. Retrieved Febrero 03, 2024, from Ibm.com: https://www.ibm.com/docs/en/aix/7.1?topic=protocols-telnet-protocol

Islami, M. F., Musa, P., & Lamsani, M. (2020). Implementation of Network Automation using Ansible to Configure Routing Protocol in Cisco and Mikrotik Router with Raspberry PI.

Array. Jurnal Ilmiah Komputasi, 19(2), 127-134. https://doi.org/https://doi.org/10.32409/jikstik.19.2.80

jagroopofficial , J. (2023, Junio 22). *Protocol and standard in computer networks*. GeeksforGeeks: https://www.geeksforgeeks.org/protocol-and-standard-in-computer-networks/

JetBrains. (2024). *"PyCharm: el IDE de Python para desarrolladores profesionales"*. Retrieved Enero 18, 2024, from jetbrains.com: https://www.jetbrains.com/es-es/pycharm/

Kanade, V. (2022, Febrero 10). *What is network hardware? Definition, architecture, challenges, and best practices*. Retrieved Febrero 02, 2024, from Spiceworks: https://www.spiceworks.com/tech/networking/articles/what-is-network-hardware/

Keary, T. (2023, Agosto 10). *Types of Routing Protocols – The Ultimate Guide*. Retrieved Febrero 13, 2024, from https://www.comparitech.com/net-admin/routing-protocol-types-guide/

Keary, T., & Kiran, A. (2023, Agosto 10). *9 types of network protocols & when to use them*. Retrieved Febrero 03, 2024, from Forbes: https://www.forbes.com/advisor/business/types-network-protocols/

Kochar, G. (2021, Agosto 17). *Python NAPALM For Network Automation*. Retrieved Febrero 12, 2024, from https://networkautomationlane.in/python-napalm-for-network-automation/

Koppenhaver, K. (2023, Febrero 07). *Comparing Ansible, Terraform, Chef, Salt, and Puppet for cloud-native applications*. Retrieved Febrero 07, 2024, from https://redpanda.com/blog/ansible-terraform-chef-salt-puppet-cloud

Marrón, S. S. (2015, Septiembre 13). *2. tipos de redes*. Retrieved Enereo 31, 2024, from SlideShare: https://es.slideshare.net/ssainzm/2-tipos-de-redes

Mazin, A., Rahman, R., Kassim, M., & Mahmud, A. R. (2021). Performance Analysis on Network Automation Interaction with Network Devices Using Python. *2021 IEEE 11th IEEE*

Symposium on Computer Applications \& Industrial Electronics (ISCAIE), 360-366. https://doi.org/10.1109/ISCAIE51753.2021.9431823

Multicloud. (2018, Noviembre 02). Retrieved Enero 31, 2024, from The Promise of the Multicloud World is Amazing: https://www.cisco.com/c/en/us/solutions/automation/network-topology.html

Ojeda, O. (2021, 12 22). *Título: Automatización de Redes con Python.* Repositorio Digital de la Universidad Central “Marta Abreu” de Las Villas, Facultad de Ingeniería Eléctrica, Departamento de Electrónica y Telecomunicaciones: https://dspace.uclv.edu.cu/items/4f8040f1-5db5-45cd-b0e5-37ad42bd918e

Petryschuk, S. (2021, Febrero 03). *11 types of networks: Understanding the differences. Auvik*. Retrieved Febrero 09, 2024, from Auvik Networks: https://www.auvik.com/franklyit/blog/types-of-networks/

Python Software Foundation. (2024). *La biblioteca estándar de Python*. Retrieved Febrero 08, 2024, from Python documentation.

Python Software Foundation. (2024). *telnetlib — Telnet client*. Retrieved Febrero 09, 2024, from https://docs.python.org/3/library/telnetlib.html

Rajdojr1o, R. (2022, Agosto 21). *Network infrastructure*. Retrieved Febrero 01, 2024, from GeeksforGeeks: https://www.geeksforgeeks.org/network-infrastructure/

Rodríguez, A. (2020, Agosto 22). *Routing y Switching: optimización de redes*. Retrieved Febrero 14, 2024, from https://www.tokioschool.com/noticias/routing-y-switching/#:~:text=El%20Switching%20se%20utiliza%20para,recursos%20compartidos%20a%20nivel%20interno.

Rouse, M. (2023, Octubre 30). *Network Infrastructure*. Retrieved Febrero 01, 2024, from Tecnnology Expert: https://www.techopedia.com/definition/16955/network-infrastructure

Schrader, D. (2023, Agosto 02). *Network Devices: Types, Functions and Best Practices for Security Management*. Retrieved Febrero 02, 2024, from netwrix.com: https://blog.netwrix.com/2019/01/08/network-devices-explained/

SolarWinds. (2024). *What is Network Automation?* Retrieved Febrero 04, 2024, from SolarWinds Worldwide: https://www.solarwinds.com/resources/it-glossary/network-automation

SSH. (1996). SSH - Secure login connections over the Internet. In P. o. Symposium. USENIX. Retrieved Febrero 04, 2024, from https://www.ssh.com/academy/ssh/protocol

Study-ccna. (2024). *Configuration Management Tools – Ansible, Chef, Puppet*. Retrieved Febrero 04, 2024, from study-ccna.com: https://study-ccna.com/configuration-management-tools-ansible-chef-puppet/

Terraform by HashiCorp. (2024). *Automate infrastructure on any cloud with Terraform*. Retrieved 04 de Febrero de 2024, from https://www.terraform.io/

VMware. (2023, Septiembre 11). *What is network infrastructure?* Retrieved Febrero 01, 2024, from VMware.com: https://www.vmware.com/es/topics/glossary/content/network-infrastructure.html

WMware. (2024). *What is network automation?* Retrieved Febrero 04, 2024, from vmware.com: https://www.vmware.com/topics/glossary/content/network-automation.html#:~:text=Network%20automation%20is%20the%20process,in%20c

Zenarmor. (2024). *What is Network Topology?* Retrieved Febrero 02, 2024, from zenarmor.com: https://www.zenarmor.com/docs/network-basics/what-is-network-topology