UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Automatización de redes: caso práctico ancho de banda y establecimiento de roles

LUZON CHIRIGUAYO JOEL ABRAHAM
INGENIERO EN TECNOLOGIAS DE LA INFORMACION

LOPEZ TELLO JEREMY ISAAC
INGENIERO EN TECNOLOGIAS DE LA INFORMACION

MACHALA
2024

UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

Automatización de redes: caso práctico ancho de banda y establecimiento de roles

LUZON CHIRIGUAYO JOEL ABRAHAM
INGENIERO EN TECNOLOGIAS DE LA INFORMACION

LOPEZ TELLO JEREMY ISAAC
INGENIERO EN TECNOLOGIAS DE LA INFORMACION

MACHALA
2024

UTMACH

FACULTAD DE INGENIERÍA CIVIL

CARRERA DE TECNOLOGÍAS DE LA INFORMACIÓN

PROPUESTAS TECNOLÓGICAS

Automatización de redes: caso práctico ancho de banda y establecimiento de roles

LUZON CHIRIGUAYO JOEL ABRAHAM
INGENIERO EN TECNOLOGIAS DE LA INFORMACION

LOPEZ TELLO JEREMY ISAAC
INGENIERO EN TECNOLOGIAS DE LA INFORMACION

MOROCHO ROMAN RODRIGO FERNANDO

MACHALA
2024

CERTIFICADO DE ANÁLISIS
magister

Compilatio-Lopez_Jeremy_y_Luzon_Joel-20250204

1%
Textos sospechosos

1% **Similitudes**
< 1% similitudes entre comillas
0% entre las fuentes mencionadas
0% **Idiomas no reconocidos**

Nombre del documento: Compilatio-Lopez_Jeremy_y_Luzon_Joel-20250204.pdf
ID del documento: e7c5bc135b004653b7606eb45a601252bada0b2a
Tamaño del documento original: 2,74 MB
Autores: []

Depositante: RODRIGO FERNANDO MOROCHO ROMAN
Fecha de depósito: 4/2/2025
Tipo de carga: interface
fecha de fin de análisis: 4/2/2025

Número de palabras: 13.936
Número de caracteres: 107.366

Ubicación de las similitudes en el documento:

Fuentes con similitudes fortuitas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	**dspace.ups.edu.ec** | Diseño e implementación de un prototipo didáctico para el mo... http://dspace.ups.edu.ec/bitstream/123456789/21805/4/UPS-GT003605.pdf	< 1%		Palabras idénticas: < 1% (10 palabras)
2	**repositorio.puce.edu.ec** https://repositorio.puce.edu.ec/bitstreams/667c0d85-39d9-44b4-9318-ac0c33a0ee9c/download	< 1%		Palabras idénticas: < 1% (32 palabras)
3	**www.hindawi.com** | Intelligent Fault-Tolerant Mechanism for Data Centers of Cloud ... https://www.hindawi.com/journals/mpe/2022/2379643/	< 1%		Palabras idénticas: < 1% (16 palabras)
4	**www.mdpi.com** | Network Function Virtualization and Service Function Chaining Fra... https://www.mdpi.com/1999-5903/14/2/59	< 1%		Palabras idénticas: < 1% (14 palabras)
5	**bibliotecadigital.udea.edu.co** https://bibliotecadigital.udea.edu.co/bitstream/10495/15082/1/TovarJuliana_2020_OptimizacionC...	< 1%		Palabras idénticas: < 1% (14 palabras)

Fuentes mencionadas (sin similitudes detectadas)

Estas fuentes han sido citadas en el documento sin encontrar similitudes.

1 https://www.sciencedirect.com/science/article/pii/S266591742300048X?via=ihub
2 https://openaccess.uoc.edu/bitstream/10609/73586/6/rsalesgTFG0118memoria.pdf
3 https://doi.org/10.46550/amormundi.v4i4.211

CLÁUSULA DE CESIÓN DE DERECHO DE PUBLICACIÓN EN EL REPOSITORIO DIGITAL INSTITUCIONAL

Los que suscriben, LUZON CHIRIGUAYO JOEL ABRAHAM y LOPEZ TELLO JEREMY ISAAC, en calidad de autores del siguiente trabajo escrito titulado Automatización de redes: caso práctico ancho de banda y establecimiento de roles, otorgan a la Universidad Técnica de Machala, de forma gratuita y no exclusiva, los derechos de reproducción, distribución y comunicación pública de la obra, que constituye un trabajo de autoría propia, sobre la cual tienen potestad para otorgar los derechos contenidos en esta licencia.

Los autores declaran que el contenido que se publicará es de carácter académico y se enmarca en las dispociones definidas por la Universidad Técnica de Machala.

Se autoriza a transformar la obra, únicamente cuando sea necesario, y a realizar las adaptaciones pertinentes para permitir su preservación, distribución y publicación en el Repositorio Digital Institucional de la Universidad Técnica de Machala.

Los autores como garantes de la autoría de la obra y en relación a la misma, declaran que la universidad se encuentra libre de todo tipo de responsabilidad sobre el contenido de la obra y que asumen la responsabilidad frente a cualquier reclamo o demanda por parte de terceros de manera exclusiva.

Aceptando esta licencia, se cede a la Universidad Técnica de Machala el derecho exclusivo de archivar, reproducir, convertir, comunicar y/o distribuir la obra mundialmente en formato electrónico y digital a través de su Repositorio Digital Institucional, siempre y cuando no se lo haga para obtener beneficio económico.

LUZON CHIRIGUAYO JOEL ABRAHAM
0705790442

LOPEZ TELLO JEREMY ISAAC
0705935179

DEDICATORIA

López Tello Jeremy Isaac

El presente trabajo de titulación, en primer lugar, se lo dedico a Dios por darme la sabiduría y la tranquilidad en este proceso muy importante de mi vida; a mis padres Iván López, Melania Tello por ser esas personas incondicionales que siempre están a mi lado; a mi hermano Iván López, que es mi compañero y mi mejor amigo, el cual está siempre aconsejándome y dándome los ánimos cuando a veces desfallezco; a mi enamorada Lucia Romero porque ella ha compartido muchos momentos junto a mí, quien también ha sabido darme esa tranquilidad y aconsejarme en los momentos que más lo necesito. Gracias a todos porque han sido una motivación para no rendirme y seguir de pie para alcanzar mi meta.

Luzón Chiriguayo Joel Abraham

A Dios por darme la fortaleza, paciencia y determinación para alcanzar esta meta, a mi familia, en especial a mis padres Segundo Luzón, Gina Chiriguayo, y mis hermanos Luis, Coraima, Abigail, quienes con su amor, apoyo incondicional y consejos han sido mi mayor inspiración. A mis amigos, por su compañía y palabras de aliento en los momentos difíciles. A todos aquellos que de una u otra forma han formado parte de este camino y han ayudado en mi crecimiento personal y profesional.

AGRADECIMIENTO

López Tello Jeremy Isaac

Agradecer a Dios, quien me da la fortaleza para seguir adelante, y a mi familia en general, que me han sabido guiar para superarme en esta etapa de mi vida, con sus consejos y motivación. A mis tutores de tesis, Ing. Rodrigo Morocho y Ing. Nancy Loja, por su invaluable apoyo, paciencia y guía a lo largo de este camino, su compromiso, conocimiento y consejos fueron fundamentales para la realización de este trabajo, gracias por compartir su tiempo y ayudarme a crecer tanto de manera académica como personalmente, su dedicación y pasión por la enseñanza han sido una inspiración para mí, y también agradecer a mi compañero de tesis Joel Luzón, quien ha sido una pieza fundamental en la realización de este trabajo, su esfuerzo, compromiso y dedicación han sido clave para superar cada desafío que encontramos en el camino, más allá del trabajo académico, valoro profundamente su amistad, compañerismo y apoyo incondicional. Ha sido un verdadero privilegio compartir este proceso contigo, y estoy seguro de que este logro es el resultado de nuestro esfuerzo conjunto.

Luzón Chiriguayo Joel Abraham

En primer lugar, agradezco a Dios, quien me ha dado la fortaleza, sabiduría y la perseverancia necesaria para superar cada desafío a lo largo de este camino. A mi familia, especialmente a mis padres, Segundo Luzón, Gina Chiriguayo y a mis hermanos Luis, Coraima y Abigail, por ser mi mayor fuente de apoyo y motivación, por su amor incondicional, sus consejos y por enseñarme el valor del esfuerzo y la dedicación. A mis tutores de tesis, Ing. Rodrigo Morocho e Ing. Nancy Loja, por su paciencia, orientación y valiosos aportes que han sido fundamentales para la culminación de este trabajo. A mis amigos, quienes con su apoyo y compañía hicieron más ameno este proceso. Finalmente, pero no menos importante, a mi compañero de tesis Jeremy López, por su compromiso, esfuerzo y amistad a lo largo de este proceso. Su dedicación y trabajo en equipo han sido clave para lograr este objetivo juntos.

RESUMEN

La automatización de redes se ha convertido en una herramienta crucial para optimizar la administración de infraestructuras tecnológicas; su implementación permite mejorar la eficiencia operativa y gestionar los recursos de manera más efectiva. En este trabajo, centrado en redes y programación, se llevó a cabo un caso práctico orientado a automatizar la gestión del ancho de banda y la asignación de roles en un entorno de red emulado. Esto responde a la necesidad de reducir la intervención manual en la configuración de redes, lo que se traduce en un ahorro de tiempo y una disminución de errores humanos.

Se utilizó un entorno de red basado en Python, VMware, GNS3, Lubuntu y routers Cisco C7200, aplicando la metodología PDCA (Plan-Do-Check-Act). Con este enfoque, fue posible gestionar el tráfico de red de manera más eficiente y asignar roles de forma dinámica, lo que resultó en una administración más precisa y en una reducción significativa del tiempo necesario para la configuración en comparación con los métodos manuales tradicionales.

Los resultados obtenidos demuestran que la automatización no solo mejora la estabilidad y el rendimiento de la red, sino que también optimiza el uso del ancho de banda y facilita su administración. Además, esta estrategia reduce la carga operativa y contribuye a una gestión más eficiente de las infraestructuras tecnológicas. Como línea de trabajo futuro, se sugiere explorar la integración de inteligencia artificial para mejorar la toma de decisiones en la configuración de redes y ampliar la automatización a infraestructuras más complejas, lo que permitiría un mayor nivel de eficiencia y adaptación a diferentes escenarios tecnológicos.

PALABRAS CLAVE

Automatización de redes, gestión de ancho de banda, asignación de roles, optimización de recursos, entorno emulado.

ABSTRACT

Network automation has become a crucial tool for optimizing the management of technological infrastructures; its implementation improves operational efficiency and enables more effective resource management. This study, focused on networking and programming, presents a practical case aimed at automating bandwidth management and role assignment in an emulated network environment. This approach addresses the need to reduce manual intervention in network configuration, resulting in time savings and a decrease in human errors.

A network environment was implemented using Python, VMware, GNS3, Lubuntu, and Cisco C7200 routers, following the PDCA (Plan-Do-Check-Act) methodology. This approach made it possible to manage network traffic more efficiently and assign roles dynamically, leading to more precise administration and a significant reduction in configuration time compared to traditional manual methods.

The results demonstrate that automation not only enhances network stability and performance but also optimizes bandwidth usage and simplifies its management. Additionally, this strategy reduces the operational workload and contributes to a more efficient management of technological infrastructures. As a future research direction, integrating artificial intelligence for improved decision-making in network configuration and extending automation to more complex infrastructures is suggested. This would enable a higher level of efficiency and adaptability to different technological scenarios.

KEYWORDS

Network automation, bandwidth management, role assignment, resource optimization, emulated environment.

ÍNDICE DE CONTENIDO

ÍNDICE DE TABLAS

ÍNDICE DE FIGURAS

Glosario

OSPF: Es un protocolo de enrutamiento dinámico basado en el estado de enlace que se utiliza en redes IP. OSPF calcula las rutas más óptimas para el tráfico de datos utilizando el algoritmo de Dijkstra y se adapta automáticamente a los cambios en la topología de la red.

IP:(Internet Protocol) Es un protocolo fundamental en la suite TCP/IP que permite el envío y recepción de datos a través de redes. Se encarga de asignar direcciones IP únicas a dispositivos y de fragmentar los datos en paquetes para su transmisión.

IEEE: (Institute of Electrical and Electronics Engineers)
Es una organización internacional encargada de desarrollar estándares en tecnologías relacionadas con la electricidad, la electrónica y las telecomunicaciones. Ejemplo: IEEE 802.3 para Ethernet e IEEE 802.11 para redes Wi-Fi.

ROUTER: Es un dispositivo de red que conecta diferentes redes y dirige los paquetes de datos entre ellas, utilizando tablas de enrutamiento para determinar el mejor camino para alcanzar su destino.

SWITCH: Es un dispositivo de red que conecta múltiples dispositivos dentro de una misma red local (LAN), opera en la capa 2 (enlace de datos) del modelo OSI y utiliza direcciones MAC para enviar datos al destino correcto.

VLAN: (Virtual Local Area Network) Es una tecnología que permite dividir una red física en múltiples redes lógicas independientes. Esto mejora la seguridad y el rendimiento al segmentar el tráfico de red según necesidades específicas.

REDES DEFINIDAS POR SOFTWARE: Es un enfoque de redes que separa el plano de control (gestión) del plano de datos (transferencia) para facilitar la administración centralizada y la automatización de las redes.

AUTOMATIZACIÓN: Es el proceso de realizar tareas o configuraciones de manera automática mediante herramientas o scripts, minimizando la intervención manual y aumentando la eficiencia y precisión en la gestión de redes

HSRP: (Hot Standby Router Protocol) Es un protocolo desarrollado por Cisco que proporciona alta disponibilidad en redes. Permite configurar un grupo de routers redundantes, de manera que uno actúe como principal y los demás estén en espera para asumir el rol en caso de fallo.

ETHERNET: Es una tecnología estándar para redes locales (LAN) que define protocolos para la transmisión de datos mediante cables. Utiliza un método de acceso conocido como CSMA/CD (Carrier Sense Multiple Access with Collision Detection).

INTRODUCCIÓN

La automatización de redes es un campo de creciente relevancia teórica y práctica en la ingeniería informática y de telecomunicaciones. A nivel mundial, la falta de automatización en la gestión del ancho de banda y la asignación de roles en redes computacionales representa un desafío significativo para la eficiencia y el desempeño organizacional. Este problema es particularmente crítico en América Latina y, específicamente, en Ecuador, donde la ausencia de sistemas automatizados limita la capacidad de las organizaciones para ofrecer servicios de calidad y mantener una infraestructura tecnológica sostenible.

El proyecto de investigación tiene como objetivo desarrollar un sistema automatizado que gestione de manera eficiente el ancho de banda y la asignación de roles, utilizando un entorno de red emulado. Este sistema se espera que mejore la eficiencia operativa y el rendimiento de la red, asegurando una mejor calidad de servicio y experiencia de usuario. La metodología adoptada incluye un análisis exhaustivo de las tecnologías existentes en automatización de redes y asignación de roles, seguido de la implementación y evaluación de un sistema prototipo. La implementación de técnicas avanzadas de automatización, como el aprendizaje automático y el análisis predictivo, promete superar las limitaciones de los enfoques manuales tradicionales, ofreciendo soluciones más adaptables y eficientes.

Los beneficiarios directos de este proyecto serán la comunidad académica y las organizaciones que buscan mejorar la eficiencia, seguridad y calidad de sus redes. Con la automatización de la gestión del ancho de banda y la asignación de roles, se espera una mejora significativa en la conectividad, haciéndola más rápida, confiable y segura. Este proyecto, por tanto, no solo aborda un problema técnico, sino que también contribuye al desarrollo tecnológico y a la competitividad de las organizaciones en un entorno globalizado y en constante evolución.

I. Declaración y formulación del problema

Declaración del problema

A nivel mundial, la falta de automatización de funciones de red se ve reflejada en la carga administrativa que tienen los encargados de infraestructura, razón por la cual se considera abordar casos prácticos como la gestión del ancho de banda y la asignación de roles en redes digitales, debido a que es una problemática que afecta a la eficiencia y el desempeño de las organizaciones que no cuentan con un proceso automatizado para gestionar el tráfico de red y distribuir adecuadamente los recursos, los cuales producen desequilibrios en el uso de ancho de banda, limitando la capacidad para lograr mantener una infraestructura digital robusta.

La ausencia de sistemas automatizados en la gestión del ancho de banda y la asignación de roles adecuados impiden ofrecer servicios de calidad y dificultan la sostenibilidad de la infraestructura tecnológica. Como se menciona en [1], la automatización de las empresas en la actualidad constituye un pilar para la competitividad; esto ayuda a que no se intervenga en los procesos de manera manual y se efectúe de manera automática.

La falta de automatización en la gestión del ancho de banda y la asignación de roles puede afectar la calidad del servicio brindado a los usuarios de cualquier institución. Es por ello que, para evitar que la calidad del servicio se vea comprometida, se propone implementar soluciones automatizadas que optimicen el uso del ancho de banda y mejoren la gestión de roles, elevando así la experiencia digital para todos los usuarios.

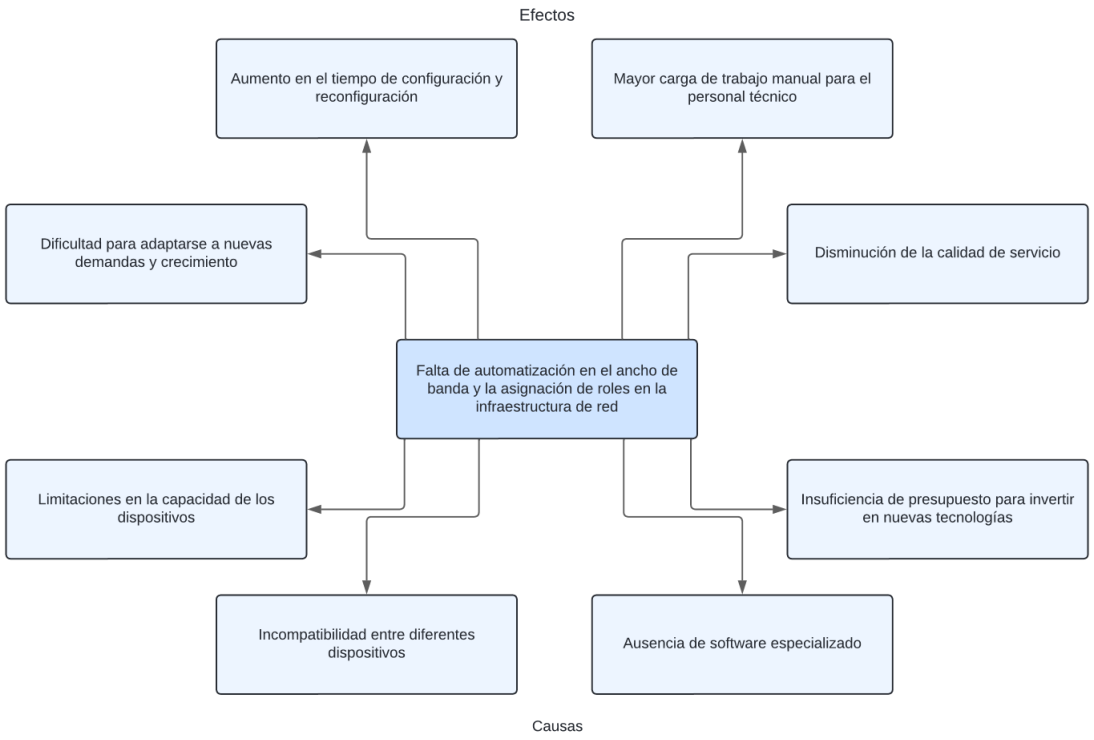


Figura 1: Árbol de Problema

Formulación del problema

- ¿Cómo puede implementarse de manera efectiva la automatización en redes de computadoras para optimizar el uso del ancho de banda y garantizar un establecimiento eficiente de roles, considerando las necesidades específicas de la infraestructura, el flujo de datos y los protocolos de comunicación?

Problema principal

- ¿Cómo lograr una automatización eficiente en redes informáticas que maximice la utilización del ancho de banda y defina roles de manera óptima?

II. Objeto de estudio y campo de acción

Objeto de estudio

- Se centra en la red digital misma, donde se aplicarán las soluciones de automatización para optimizar la asignación de ancho de banda y roles. Este enfoque permite analizar la realidad específica de la infraestructura de red seleccionada y desarrollar intervenciones concretas para mejorar su funcionamiento y eficiencia.

Campo de acción

- Se centra en las disciplinas específicas de la ingeniería de redes y la informática. Estas disciplinas abordarían la investigación, diseño, implementación y evaluación de soluciones tecnológicas para optimizar la asignación de ancho de banda y roles en la red digital seleccionada.

III. Objetivos

3.1. Objetivo general

- Automatizar redes de computadoras para la gestión del ancho de banda y el establecimiento de roles mediante la configuración de un entorno de red emulado.

3.2. Objetivos específicos

- Investigar tecnologías y metodologías de automatización de redes, con un enfoque en la gestión del ancho de banda y la asignación de roles.
- Diseñar una topología de red para la automatización de ancho de banda y la asignación de roles.
- Configurar la topología de red diseñada, implementando herramientas que ayuden a la automatización.
- Evaluar la funcionalidad del sistema de automatización mediante el uso de la herramienta Wireshark, para comprobación de la reducción en los tiempos de respuesta de la red automatizada en comparación con la configuración manual vía consola.

IV. Hipótesis y variables

Hipótesis principal

- La automatización de las funciones como la gestión del ancho de banda y asignación de roles disminuirá los tiempos de respuesta de la red en relación con la configuración vía consola.

Variables y dimensionamiento

Tabla 1: Variables y Dimensionamiento

Categoría	Variables	Indicadores
Variable independiente	Sistema de automatización de redes	Sistema de automatización de red, utilizando algoritmos de gestión de ancho de banda y establecimiento de roles.
Variable dependiente	Carga administrativa	Mayor capacidad para manejar el tráfico de red, mayor precisión en la asignación de roles.

V. Justificación

La conectividad en la actualidad es muy importante en varios aspectos, tanto en la productividad, comunicación y operaciones comerciales; sin embargo, todos estos no garantizan un rendimiento óptimo, ya que no puede ofrecer una equidad en ancho de banda. Además, en el paradigma actual de las redes, la configuración y gestión suelen hacerse manualmente a través de la línea de comandos, lo que puede ser propenso a errores humanos y limitaciones en la adaptabilidad a los cambios en el tráfico de red. Por esta razón se ha pensado en la automatización de ancho de banda en redes computacionales para gestionar las redes y se puedan distribuir de una mejor manera.

La importancia de esta se basa en permitir que las empresas puedan optimizar sus recursos de red, asegurando un uso más eficiente del ancho de banda disponible y que estos se vayan adaptando a los cambios del tráfico de red, para así obtener una mejor calidad de servicio y experiencia de usuario. Además, la implementación de roles dentro de una red mejora la seguridad al limitar el acceso solo a usuarios autorizados y dispositivos confiables para disminuir los intrusos y fugas de datos. La innovación radica en la integración de técnicas avanzadas de automatización en la gestión del ancho de banda y la asignación de roles, implementando aprendizaje automático y análisis predictivo para toma de decisiones, superando las limitaciones de los enfoques manuales utilizados tradicionalmente.

Los beneficiarios directos son la comunidad académica, empresas u organizaciones que buscan mejorar la eficiencia, seguridad y calidad de sus redes, obteniendo una mejora en la conectividad, siendo más rápida, confiable y segura.

La motivación surge de la necesidad de adaptarse a un panorama tecnológico en constante evolución, donde el volumen y la diversidad de tráfico de red plantean desafíos significativos para una administración efectiva de los recursos disponibles.

1. CAPÍTULO II. MARCO TEÓRICO

1.1. Antecedentes de la investigación

La revisión bibliográfica de la investigación se realizó usando la metodología de Revisión Sistemática de Literatura (SRL: Systematic Review of the Literature).

La SRL pretende recopilar información de publicaciones relacionadas con la investigación con criterios de elegibilidad predefinidos para responder las interrogantes de la investigación y una correcta estructura en la redacción del informe. Esta estrategia logra realizar el estudio y puede proporcionar una mirada transparente, exhaustiva y estructurada de la literatura especializada[2], esto nos permite obtener una respuesta clara a una pregunta de investigación [3].

a) Preguntas de investigación

Tabla 2: Preguntas de investigación

Pregunta de investigación	Descripción
¿La automatización de la optimización del ancho de banda y asignación de roles en un entorno de red disminuirá los tiempos de respuesta?	Identificar si la implementación de automatización en la gestión del ancho de banda y la asignación de roles puede mejorar significativamente los tiempos de respuesta en la red.
¿Las tecnologías utilizadas para la automatización de redes garantizarán resolver los problemas asociados a la configuración de la optimización del ancho de banda y la asignación de roles?	Evaluar la eficacia de las tecnologías actuales de automatización en la resolución de problemas específicos relacionados con la configuración del ancho de banda y la asignación de roles.

b) Palabras clave y cadenas de búsqueda

Para realizar la búsqueda de artículos científicos de revistas relevantes, se puede obtener información de diferentes repositorios científicos y bases de datos, los cuales son:

Google Scholar, Scopus, IEEE Xplore y Web of Science.

Después de buscar en las diferentes bases de datos científicas, se define la cadena, la cual nos permite buscar por títulos, palabras claves, resumen y textos generales:

"Network automation" AND "bandwidth optimization".

"Automatización de redes" Y "optimización de ancho de banda".

c) Criterios de inclusión y exclusión

Como criterios de inclusión fueron tomados en cuenta artículos científicos, estudios primarios, estudios que sean publicados a partir del año 2020 en adelante, estudios relacionados al tema propuesto; como criterios de exclusión se tomó en cuenta lo siguiente: estudios secundarios, estudios publicados antes del 2020, estudios duplicados. En la Tabla 3 se pueden observar detalladamente los criterios de inclusión y exclusión que se usaron para realizar esta investigación.

Tabla 3: Criterios de inclusión y exclusión

#	**Criterios de inclusión**
1	Estudios primarios
2	Estudios publicados a partir de año 2020
3	Estudios relacionados a la automatización de redes
4	Estudios relacionados a emulación de redes en GNS3
5	Estudios relacionados con el ancho de banda de una red
6	Estudios relacionados con la asignación de roles
#	**Criterios de exclusión**
1	Estudios secundarios
2	Estudios publicados antes del 2020
3	Estudios duplicados
4	Estudios que no estén redactados en inglés o español
5	Estudios que no se encuentren disponibles de manera gratuita
6	Estudios cuya orientación no esté alineada a nuestras palabras claves

d) Proceso y resultados de la búsqueda

El proceso de la búsqueda se realizó mediante el uso de palabras claves y cadenas de búsqueda en las diferentes bases de datos bibliográficas, las cuales son: En MDPI, Google Scholar, Scopus, IEEE Xplore y Web of Science, en la Figura 2 se puede visualizar el diagrama de procesos de búsqueda.

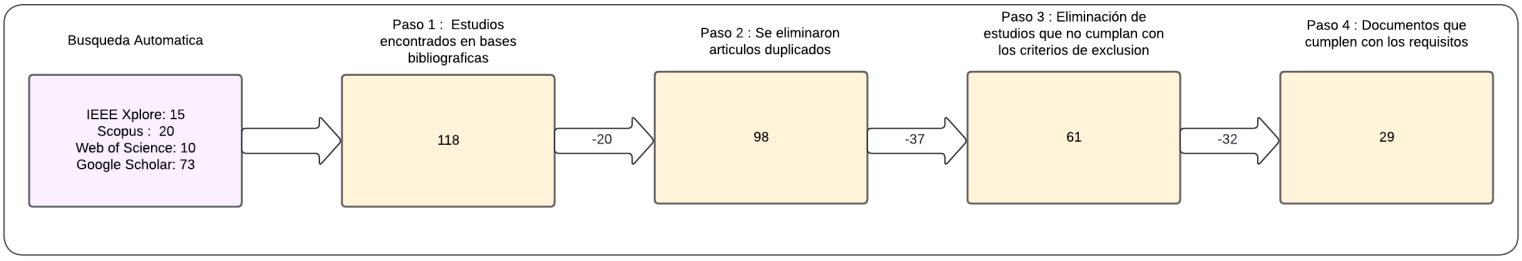


Figura 2: Diagrama de proceso de búsqueda

1.2. Antecedentes históricos

Según [4], las redes de datos como se las conoce actualmente surgieron en los años 1970 como un resultado de los requerimientos de las primeras redes militares y de la aparición del internet. Fueron concebidas como funcionamiento basado en conmutación de paquetes, donde cada componente funciona de manera inteligente, con la capacidad de la toma de decisiones autónomas.

[5] Indica que se requieren redes autónomas para ampliar la capacidad de gestión de una red, lo que permite satisfacer algunos indicadores como la confiabilidad y la latencia. Esto requiere mayores gastos e ineficiencia a medida que se expande el tamaño de la red; es por ello por lo que la automatización de una red reduce la carga administrativa y por ende reduce los costos.

En la actualidad las redes de datos están compuestas por switches y routers, que posibilitan las comunicaciones entre clientes y servidores, ya sean físicos o virtuales; así se conforma una red compleja de administrar. Para alcanzar las funcionalidades deseadas se requiere configurar de manera manual a través de comandos. Esto limita la eficiencia y la escalabilidad en las redes.

Los avances tecnológicos han traído consigo muchas ventajas, entre ellas la automatización de redes que se realiza a través de programación y control centralizado de los recursos de red, lo que permite a los administradores de red optimizar el ancho de banda y gestionar los roles de manera más eficiente y flexible.

1.3. Antecedentes teóricos

Para entender de mejor manera los antecedentes teóricos, se elaboró un mapa temático, según la figura 3, sobre los temas relevantes.

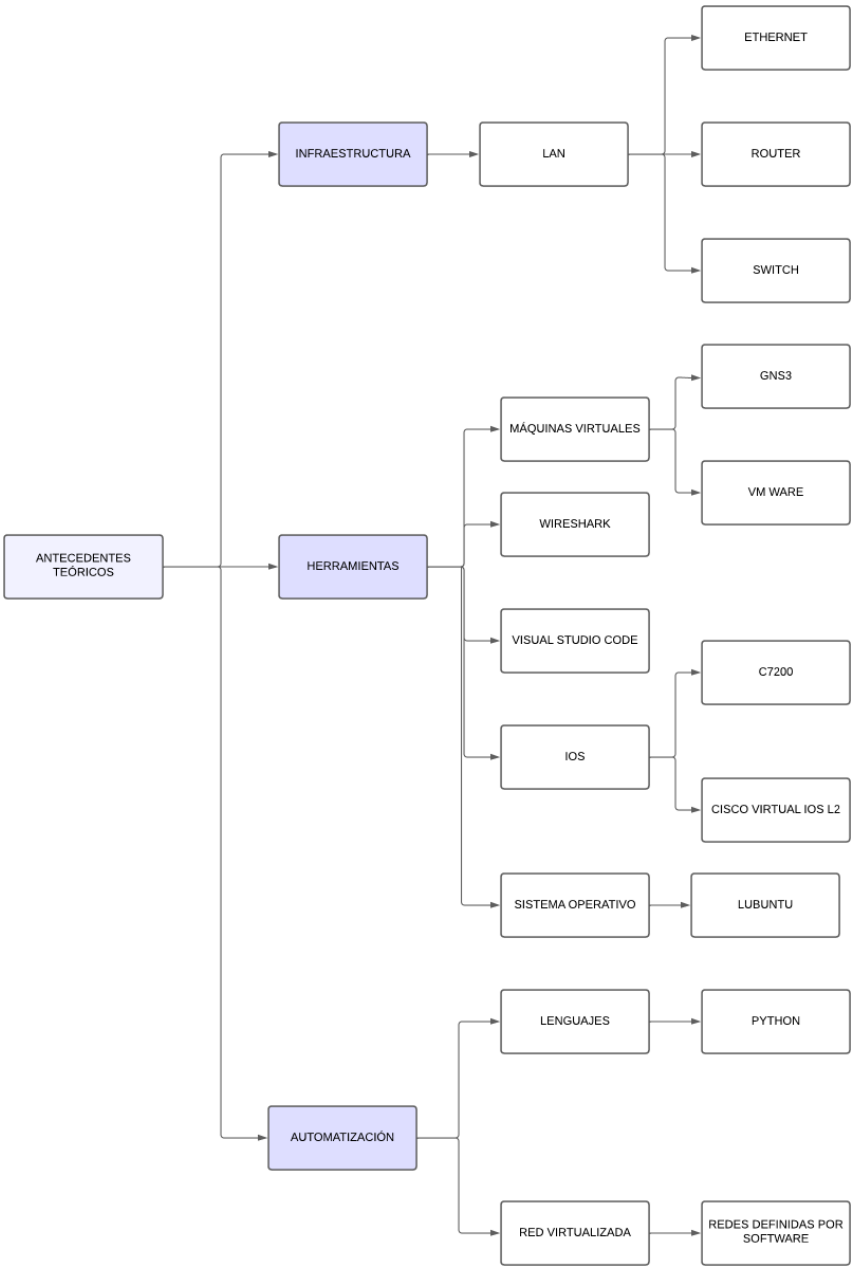


Figura 3: Mapa conceptual de Antecedentes Teóricos

1.3.1. Infraestructura

La infraestructura de red abarca todos los componentes físicos y lógicos que permiten conectar dispositivos y transmitir datos en una red informática. Esta infraestructura es vital para que cualquier sistema de comunicaciones funcione correctamente, sirviendo como la base para construir y operar tanto redes locales pequeñas (LAN) como grandes redes globales de Internet [6].

1.3.1.1.LAN

Una red LAN es una red de computadoras que abarca un área geográfica limitada, como una casa, oficina o edificio. Su principal función es conectar dispositivos como computadoras, impresoras y otros equipos, permitiendo la comunicación y el intercambio de datos entre ellos. Además, [7] menciona que se usa esta red para reducir costos y aprovechar los beneficios de banda ancha, evitando cableados adicionales.

Ethernet

Ethernet es una tecnología de red ampliamente empleada. Permite la interconexión de dispositivos dentro de una red LAN. Originada por Xerox en los años 70, Ethernet se ha convertido en el estándar predominante para redes con cableado [8].

Switch

Los switches son dispositivos clave en la gestión de una red de área local. Conectan múltiples dispositivos, como computadoras e impresoras, y facilitan la comunicación entre ellos. A diferencia de los hubs, los switches envían datos directamente al dispositivo de destino, lo que mejora la eficiencia y reduce el tráfico. Son esenciales para el funcionamiento de redes, permitiendo su expansión y escalabilidad con alta velocidad y confiabilidad [9].

Router

Es un dispositivo que administra y dirige el tráfico de datos entre redes, permitiendo que varios dispositivos compartan una única conexión a Internet. Puede ser tanto inalámbrico como cableado y se utiliza en hogares y negocios. Además de distribuir la conexión a Internet, los routers asignan direcciones IP, configuran redes locales y proporcionan seguridad a través de firewalls [10].

1.3.2. Herramientas

Máquinas virtuales

La virtualización permite ejecutar múltiples sistemas operativos y aplicaciones en un mismo servidor físico, los cuales proporcionan un sistema aislado y controlado, el cual es esencial para probar configuraciones de red. Según [11], el utilizar máquinas virtuales puede mejorar

significativamente la utilización de recursos como los servidores físicos y así reducir el costo de construir centros de datos. Por otro lado, [12] indica que la virtualización es el proceso de crear una versión simulada de recursos de hardware o software, lo que permite administrar múltiples sistemas y aplicaciones dentro de un mismo hardware.

GNS3

Según [13], nos dice que GNS3 es un simulador de redes que permite la emulación, configuración, prueba y análisis de diversas redes basadas en IP, así como la resolución de problemas que se presenten en las mismas.

En este trabajo se usará GNS3 como herramienta principal para simular redes. GNS3 permitirá crear un entorno de red virtualizado que emule realistamente los dispositivos y topologías presentes en una infraestructura empresarial. Este entorno será fundamental para el desarrollo y la prueba de scripts de automatización destinados a la gestión de ancho de banda y la asignación de roles de red. Según [14] GNS3 es una herramienta ampliamente utilizada para emular escenarios de red reales en un entorno controlado lo que conlleva a realizar pruebas sin afectar a infraestructuras reales.

VMware

VMware es una herramienta de virtualización de código abierto, permite a los usuarios ejecutar varios sistemas operativos al mismo tiempo en una sola máquina física, creando entornos virtuales independientes para cada uno. Esto facilita el desarrollo, prueba y despliegue de aplicaciones en diversas plataformas sin la necesidad de hardware adicional. Según [15] VMware utiliza un hipervisor de tipo 2 con host el cual ese necesario para virtualizar redes en la nube, estos tipos de hipervisor puede ejecutar varios sistemas operativos en simultaneo lo que ayuda a poder realizar diferentes pruebas.

WireShark

Como señala [16], esta herramienta es un analizador de protocolos de código abierto. En este trabajo, se utilizará WireShark para analizar los datos capturados del tráfico generado por el protocolo ICMP.

Visual Studio Code

Es un editor de código fuente gratuito y compatible con múltiples plataformas, diseñado para ofrecer una experiencia de desarrollo eficiente y robusta. Incluye funciones como depuración integrada, gestión de versiones con Git, resaltado de sintaxis, autocompletado de código, fragmentos y refactorización, permitiendo trabajar con una amplia variedad de lenguajes de programación y tecnologías[17].

IOS

IOS (Internetwork Operating System) está relacionado con la infraestructura y la operación de redes. Son cruciales en la gestión y mantenimiento de redes de comunicación modernas. La importancia de las redes IOS radica en la capacidad de ofrecer un marco estandarizado que logra permitir la interoperabilidad y la eficiencia en la transferencia de datos.

Citando a [18], la implementación de una manera adecuada de este software nos permite reducir significativamente el tiempo de inactividad y así lograr mejorar la confiabilidad en el sistema de nuestros dispositivos, lo que ayuda de manera eficiente a las operaciones empresariales de nuestras infraestructuras de red.

C7200

Los routers de la serie 7200 de Cisco son routers de alta capacidad que están diseñados con el fin de estar en funcionamiento en redes empresariales o de proveedores de internet porque estos necesitan un rendimiento bueno y con una alta disponibilidad. También están construidos para soportar grandes volúmenes de datos y manejar variedad de aplicaciones y servicios de red [19].

Cisco Virtual IOS L2

Cisco Virtual IOS (Internetwork Operating System) es una plataforma de virtualización de software la cual permite emular funcionalidades de los dispositivos de red correspondientes de Cisco. El término L2 hace referencia a la capa 2 del modelo OSI (Open Systems Interconnection). La capa 2 en el modelo OSI, según [20], es la encargada del enmarcado, control de acceso, direccionamiento MAC, control de velocidad de datos y la corrección de errores recibida desde la capa física. En el contexto de Cisco Virtual IOS, L2 incluye

funcionalidades como el direccionamiento MAC, la conmutación de tramas Ethernet, el control de flujo y la detección y la corrección de errores.

Sistema Operativo

Un sistema Operativo es un conjunto de programas que sirven para gestionar los recursos de hardware y software de una computadora, que proporciona servicios esenciales para la ejecución de aplicaciones y así facilitando la interacción entre usuario y máquina, según [21] los sistemas operativos modernos constan de una serie de herramientas para el procesamiento de información gráfica lo que nos ayuda a realizar acciones de una manera más fácil y efectiva. Su capacidad para gestionar recursos de manera eficiente y proporcionar una interfaz amigable con el usuario final, es crucial para el funcionamiento de dispositivos electrónicos y sistemas informáticos de manera general.

Lubuntu

Lubuntu es una distribución ligera de Linux basada en Ubuntu, esta deseñada para ser rápida y eficiente en el uso de recursos. Según un estudio realizado por [22], las distribuciones ligeras como Lubuntu, son ideales para equipos con recursos limitados, ya que optimizan el uso de memoria y procesador mediante el empleo de entornos de escritorio como LXQT.

1.3.3. Automatización de Redes

[23] indica que la automatización de redes es una metodología en la que el software logra configurar, aprovisionar, administrar y probar automáticamente los dispositivos de red. Otra definición de la automatización de redes es la eliminación de tareas manuales repetibles y su reemplazo por tareas programadas automatizadas usando software con herramientas y scripts.

La automatización pretende optimizar la administración y la operación de una infraestructura de red, reduciendo el tiempo y los errores asociados a las tareas que se realizan manualmente, lo que permite una mayor escalabilidad en la gestión de la red, reduciendo la carga administrativa.

Tipos de Lenguajes

Dentro de la informática existen varios tipos de lenguaje, entre ellos el lenguaje de programación que nos permite interpretar instrucciones y codificaciones que tienen como finalidad crear aplicaciones, programas, websites y plataformas[24]. Mientras que por otro lado también tenemos el lenguaje de marcado que simplemente tiene como función principal la transportación de datos, y es un lenguaje que nos permite definir etiquetas y así lograr tener una información estructurada jerárquicamente[25].

Python

Un lenguaje de programación introducido en el mercado por Guido Van Rossum en 1991. Las ventajas que ofrece permiten su uso para crear scripts de automatización. Algunas razones para usar este lenguaje son: facilidad de uso, ya que tiene muchos asistentes que lo hacen fácil de usar, flexibilidad, es ampliamente utilizado por la comunidad y proporciona una gran cantidad de bibliotecas.

[26] describe que el desarrollo del lenguaje de programación Python surgió como un pasatiempo; por eso este lenguaje pretende ser de fácil uso y tener un sin número de funciones que pueden ayudar a desarrollar una aplicación que permita automatizar cualquier entorno. La filosofía del diseño de Python se basa en la legibilidad de código, lo que convierte en una opción aceptable para personas que empiezan a desarrollar aplicaciones.

Red Virtualizada

La virtualización de red es una tecnología que permite abstraer los recursos físicos de una red y los convierte en recursos virtuales que estos se manejan de una manera más flexible y eficiente; de acuerdo con [27], las redes virtuales ahorrarían costos operativos a empresas de servicios.

Redes Definidas por Software

Como señala [28], es un tipo de arquitectura de red que facilita la administración y control centralizado mediante software. En lugar de depender del hardware tradicional, la SDN cuenta con un controlador centralizado que tiene una visión global de la red y puede configurar y gestionar dinámicamente los dispositivos de red, optimizando el rendimiento y

la eficiencia, incrementando la flexibilidad y facilitando la implementación de políticas y modificaciones de red.

1.4. Antecedentes contextuales

Este trabajo consiste en la emulación de pruebas de automatización de ancho de banda y la asignación de roles, utilizando herramientas que nos permitan automatizar una red por medio de scripts, el cual ayudará a la reducción de la carga administrativa empresarial, ya que automatizará estos procesos.

2. CAPITULO II. DESARROLLO DEL PROTOTIPO

2.1. Definición del prototipo

En la figura 4 se puede observar la topología que se empleará para hacer las pruebas respectivas para la automatización de la infraestructura. Los switches de capa 2 conectan dispositivos finales y segmentan la red en VLANs (VLAN 10 y VLAN 20), mejorando la seguridad y eficiencia. También están interconectados a 2 routers, los cuales tienen configuraciones conectadas al router principal que nos permitirán la salida hacia internet.

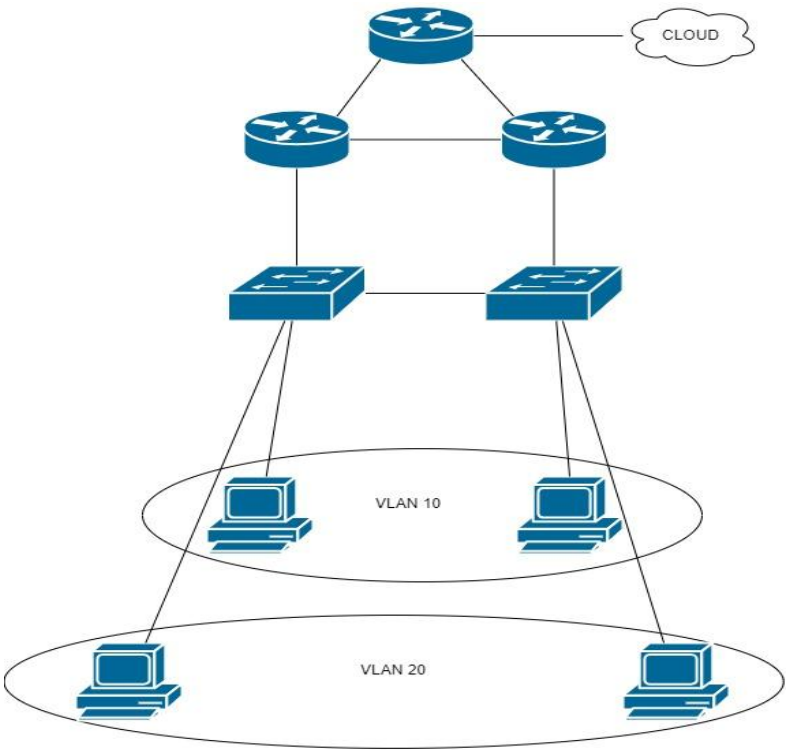


Figura 4: Topología Planteada

2.2.Metodología de desarrollo de prototipo

2.2.1. Enfoque, alcance y diseño de investigación

El proyecto adopta un enfoque cuantitativo, debido a que se utilizarán métricas de rendimiento de la red y datos de uso de ancho de banda para desarrollar y evaluar el sistema automatizado. Se recopilarán y analizarán datos empíricos para medir la eficiencia operativa, rendimiento y tiempos de respuesta del sistema en comparación con los métodos tradicionales de gestión de redes. Este enfoque permitirá obtener resultados precisos y replicables que demostraran el impacto de la automatización en la gestión de redes.

El alcance de este estudio se centrará en la automatización de redes mediante el uso de tecnologías como SDN (Software-Defined Networking). El estudio se llevará a cabo de manera emulada, donde se evaluarán parámetros específicos como el uso de ancho de banda, tiempos de respuesta y la eficiencia en la asignación de roles. El diseño de investigación es cuasiexperimental, ya que se implementará el sistema automatizado en un entorno de red de prueba, asignando tareas específicas de gestión de ancho de banda y roles en escenarios emulados.

2.2.2. Unidades de análisis

Población

La población de estudio se define como el conjunto de datos capturados durante las pruebas realizadas mientras la red está en funcionamiento. Esto incluye todo el tráfico generado y transmitido a través de la red durante los períodos de prueba. El enfoque se centra en la observación y análisis exhaustivo del tráfico de red en condiciones operativas reales, con el objetivo de comprender mejor el rendimiento, la seguridad y otros aspectos relevantes del sistema.

2.2.3. Muestra

La muestra de estudio se basa en la captura y análisis del tráfico generado exclusivamente por el protocolo ICMP durante las pruebas realizadas en la red en funcionamiento.

2.2.4. Técnicas e instrumentos de recopilación de datos (requisitos)

Se utilizará el análisis de paquetes de red como técnica principal para recopilar datos durante las pruebas realizadas en la red en funcionamiento, donde Wireshark será el instrumento principal utilizado para la captura y análisis de paquetes de red.

2.2.5. Técnicas de procesamiento y análisis de datos para la obtención de resultados

Se emplearán varias técnicas para analizar los datos capturados mediante Wireshark durante las pruebas de red en funcionamiento. Estas técnicas serán filtradas de paquetes, análisis estadísticos, análisis de flujo de datos y visualización de datos.

2.2.6. Metodología o métodos específicos

La metodología por usar es la PDCA (Plan- Do- Check-Act) o conocida como ciclo de Deming. Esta metodología se puede implementar para la automatización de redes. Según [29], esta metodología ayuda a la eficiencia de los procesos, lo cual es muy favorable para la implementación de recursos que se necesiten emplear porque ayuda a la reducción de tiempos.

PDCA

- **Planificar**

1. Analizar la infraestructura de red que se empleará y determinar la automatización del ancho de banda y la asignación de roles.
2. Seleccionar las herramientas para la automatización.

3. Desarrollar un plan de implementación.

- **Hacer**

1. Configurar un entorno de prueba emulado en GNS3
2. Realizar el direccionamiento de la topología de red.
3. Programar los scripts en Python para la gestión del ancho de banda y la asignación de roles utilizando las tecnologías seleccionadas.

- **Verificar**

1. Obtener los datos de tiempos de respuesta sobre el rendimiento de la red con la automatización.

- **Actuar**

1. Realizar ajustes necesarios a los scripts y la configuración de red.
2. Implementar mejoras.
3. Documentar los resultados.

2.2.7. Herramientas y/o Materiales

Tabla 4: Herramientas y Materiales

Categoría	Herramienta
Software	• VirtualBox • Wireshark • GNS3 • Python • XML • Visual Studio Code
Hardware	• Laptops / Pc
Sistemas / imágenes	• Imagen router C7200 • Imagen Switch capa 2 Cisco Virtual IOS L2

2.3. Desarrollo de prototipo

2.3.1. Metodología PDCA

2.3.1.1. Fase 1: Planificar

En esta fase se ha realizado una infraestructura de red para determinar cómo se automatizará la gestión de ancho de banda y la asignación de roles. Además, se evalúan rutas redundantes y la segmentación mediante VLANs para optimizar la distribución del tráfico y mejora de la seguridad. Se ha escogido GNS3 como la herramienta especializada en la simulación y automatización de redes, la cual permitirá la configuración y la gestión automática de los dispositivos de red. En la figura 5 se puede observar la topología de red realizada en GNS3 que representa una configuración de múltiples routers y switches interconectados.

A continuación, se describirán los componentes y las conexiones que tiene la topología:

Tabla 5: Conexiones de la topología

Dispositivo	**Interfaz**	**Conexión a**	**IP / VLAN**
Routers			
R1	G4/0	Cloud	192.168.80.0
	G2/0	R2	192.168.40.0
	G3/0	R3	192.168.70.0
	G4/0	R3	192.168.50.0
R2	G3/0	R3	192.168.70.0
	G1/0	SW1	-
R3	G1/0	SW2	-
Switches			
SW1	Gi1/0	R2	-
	Gi0/2	SW2	-
	Gi0/0	SW2	-
	Gi0/1	PC1	VLAN 10
SW2	Gi1/0	R3	-
	Gi0/2	SW1	-
	Gi0/0	SW1	-
	Gi0/0	PC2	VLAN 10
	Gi0/1	PC3	VLAN 20
PCs			
PC1		SW1 Gi0/1	VLAN 10
PC2		SW2 Gi0/0	VLAN 10
PC3		SW2 Gi0/1	VLAN 20
PC4		SW2 Gi0/1	VLAN 20

En esta topología podemos ver una estructura de red robusta y eficiente, adecuada para la automatización y la gestión de ancho de banda, proporcionando redundancia, segmentación y una clara separación de niveles de red.

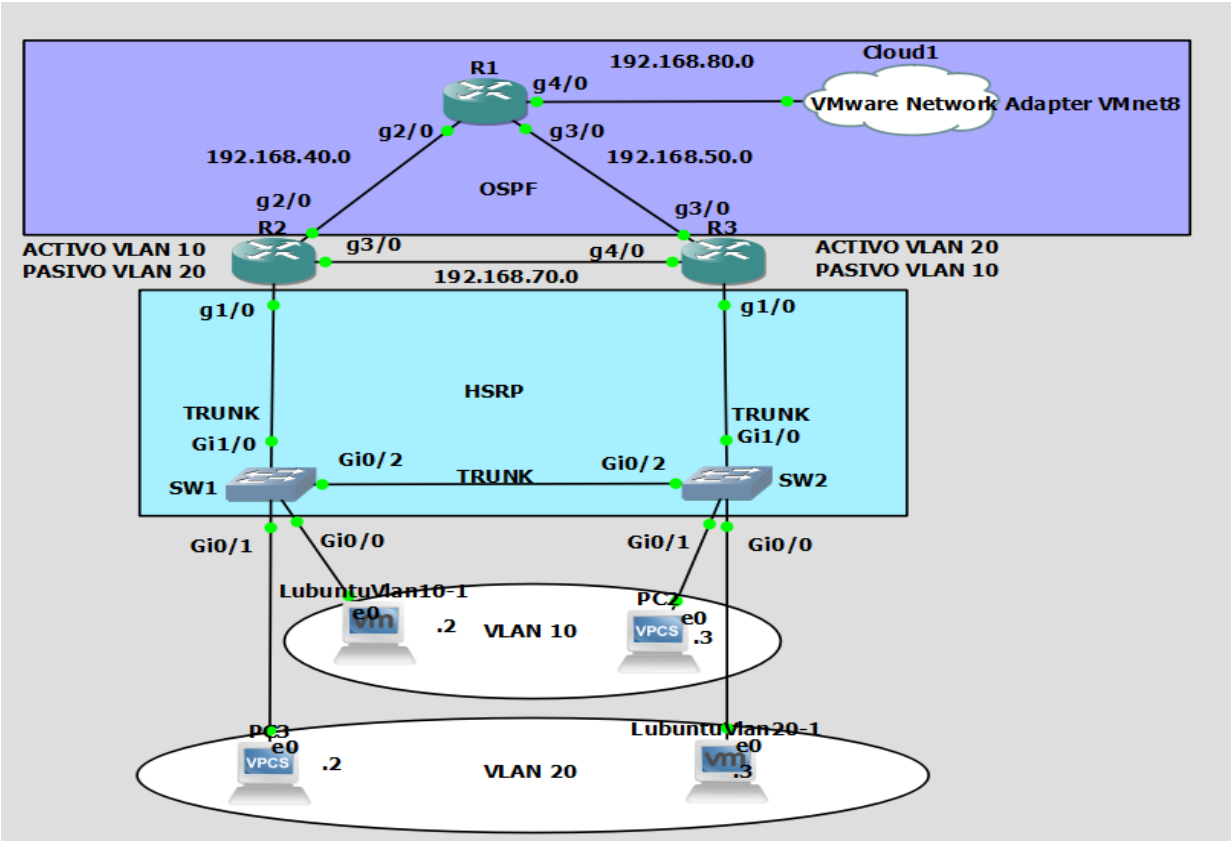


Figura 5: Topología de Red diseñada en GNS3

2.3.1.2. Fase 2: Hacer

Una vez finalizada la fase de planificación, donde se obtuvo un diseño detallado de la topología de red, se procede a la fase de ejecución, conocida como la fase de "hacer". En esta etapa, se comienza con la configuración de la topología utilizando GNS3. Esta herramienta permite crear una simulación precisa de la red planificada, facilitando la implementación y prueba de las configuraciones antes de su despliegue en el entorno de producción.

Configuración del Router R1:

En la figura 6 podemos observar la configuración inicial del Router como el nombre respectivo del dispositivo (R1).

En la figura 7 se observa la configuración de la FastEthernet y las GigabitEthernet, las cuales permiten la conexión con el router R2 y el router R3, además se puede visualizar que la

interfaz Gi4/0 está configurada con DHCP, la cual asigna una IP automática dado que esta tiene configurado un adaptador de red que simula una interfaz loopback.

En la figura 8 podemos ver la configuración del OSPF, donde se visualiza el identificador del router que es el 1.1.1.1 que van hacia las redes directamente conectadas 40.0, 50.0.

```
Building configuration...

Current configuration : 2417 bytes
!
upgrade fpd auto
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname R1
!
boot-start-marker
boot-end-marker
!
!
!
no aaa new-model
no ip icmp rate-limit unreachable
!
!
!
!
!
!
ip name-server 192.168.137.1
ip cef
no ipv6 cef
!
multilink bundle-name authenticated
!
```

Figura 6: Configuración Inicial del Router

```
!
interface FastEthernet0/0
 no ip address
 duplex half
!
interface FastEthernet1/0
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface FastEthernet1/1
 no ip address
 shutdown
 duplex auto
 speed auto
!
interface GigabitEthernet2/0
 ip address 192.168.40.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 negotiation auto
!
interface GigabitEthernet3/0
 ip address 192.168.50.1 255.255.255.0
 ip nat inside
 ip virtual-reassembly in
 negotiation auto
!
interface GigabitEthernet4/0
 ip address dhcp
 ip nat outside
 ip virtual-reassembly in
 negotiation auto
!
```

Figura 7: Configuración de las Interfaces

```
router ospf 1
 router-id 1.1.1.1
 network 192.168.40.0 0.0.0.255 area 0
 network 192.168.50.0 0.0.0.255 area 0
 network 192.168.137.0 0.0.0.255 area 0
 default-information originate
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
no cdp log mismatch duplex
!
!
!
control-plane
!
!
!
mgcp profile default
!
!
!
gatekeeper
 shutdown
```

Figura 8: Configuración OSPF

```
R1#
R1#show ip ro
R1#show ip route
Codes: L - local, C - connected, S - static, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, su - IS-IS summary, L1 - IS-IS level-1, L2 - IS-IS level-2
       ia - IS-IS inter area, * - candidate default, U - per-user static route
       o - ODR, P - periodic downloaded static route, H - NHRP, l - LISP
       + - replicated route, % - next hop override

Gateway of last resort is 192.168.80.2 to network 0.0.0.0

S*    0.0.0.0/0 [254/0] via 192.168.80.2
      192.168.40.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.40.0/24 is directly connected, GigabitEthernet2/0
L        192.168.40.1/32 is directly connected, GigabitEthernet2/0
      192.168.50.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.50.0/24 is directly connected, GigabitEthernet3/0
L        192.168.50.1/32 is directly connected, GigabitEthernet3/0
      192.168.80.0/24 is variably subnetted, 2 subnets, 2 masks
C        192.168.80.0/24 is directly connected, GigabitEthernet4/0
L        192.168.80.129/32 is directly connected, GigabitEthernet4/0
R1#
*Feb  2 11:40:54.087: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on GigabitEthernet2/0 from LOADING to FULL, Loading Done
*Feb  2 11:40:54.595: %OSPF-5-ADJCHG: Process 1, Nbr 3.3.3.3 on GigabitEthernet3/0 from LOADING to FULL, Loading Done
R1#
```

Figura 9 Configuración rutas estáticas

```
R1#show ip interface brief
Interface                  IP-Address      OK? Method Status                Protocol
FastEthernet0/0            unassigned      YES NVRAM  administratively down down
FastEthernet1/0            unassigned      YES NVRAM  administratively down down
FastEthernet1/1            unassigned      YES NVRAM  administratively down down
GigabitEthernet2/0         192.168.40.1    YES NVRAM  up                    up
GigabitEthernet3/0         192.168.50.1    YES NVRAM  up                    up
GigabitEthernet4/0         192.168.80.129  YES DHCP   up                    up
Serial5/0                  unassigned      YES NVRAM  administratively down down
Serial5/1                  unassigned      YES NVRAM  administratively down down
Serial5/2                  unassigned      YES NVRAM  administratively down down
Serial5/3                  unassigned      YES NVRAM  administratively down down
Serial5/4                  unassigned      YES NVRAM  administratively down down
Serial5/5                  unassigned      YES NVRAM  administratively down down
Serial5/6                  unassigned      YES NVRAM  administratively down down
Serial5/7                  unassigned      YES NVRAM  administratively down down
R1#
R1#
R1#
```

Figura 10 Verificación de las direcciones IP

Configuración del router R2

El Router 2 se configura de forma similar al Router 1, pero con subinterfaces para conectarse a las VLAN. En la Figura 11, se muestra la configuración de la interfaz GigabitEthernet1/0, encapsulada en la VLAN 10, con **standby 1** configurado como puerta de enlace para esta VLAN, haciendo que R2 sea el router activo en VLAN 10 y pasivo en VLAN 20. En la Figura 12, se detalla la configuración OSPF, donde se conectan directamente las redes 10.0, 20.0, 40.0 y 70.0, cumpliendo con la teoría de OSPF que requiere declarar las direcciones conectadas entre dispositivos.

```
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface GigabitEthernet1/0
 no ip address
 negotiation auto
!
interface GigabitEthernet1/0.10
 encapsulation dot1Q 10
 ip address 192.168.10.9 255.255.255.0
 standby version 2
 standby 1 ip 192.168.10.254
 standby 1 priority 200
 standby 1 preempt
!
interface GigabitEthernet1/0.20
 encapsulation dot1Q 20
 ip address 192.168.20.9 255.255.255.0
 standby version 2
 standby 2 ip 192.168.20.254
!
interface GigabitEthernet2/0
 ip address 192.168.40.2 255.255.255.0
 negotiation auto
!
interface GigabitEthernet3/0
 ip address 192.168.70.1 255.255.255.0
 negotiation auto
!
interface GigabitEthernet4/0
 no ip address
 shutdown
 negotiation auto
```

Figura 11: Configuración Router

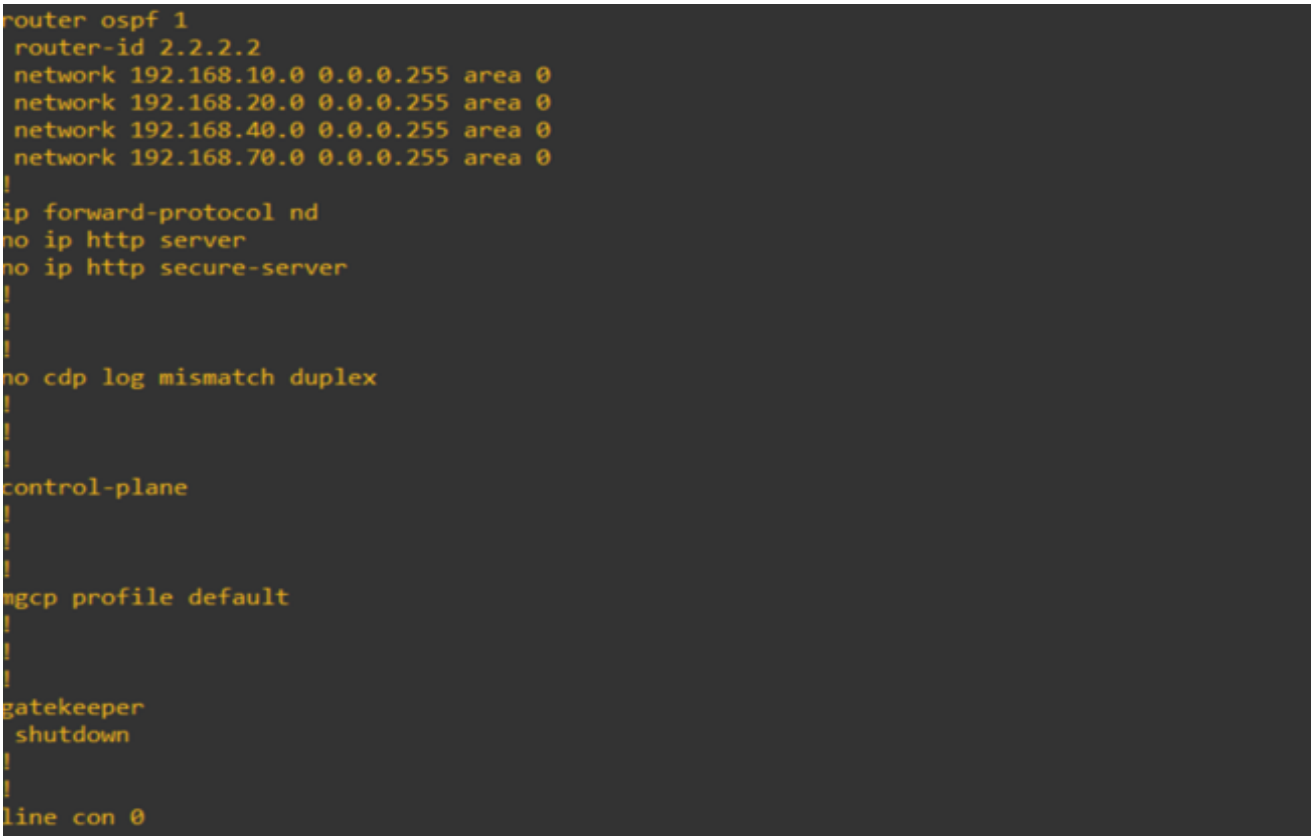


Figura 12: Configuración OSPF

Configuración del router R3

En la Figura 13, podemos observar la configuración de la interfaz GigabitEthernet1/0 de Router 3, que está encapsulada en la VLAN 20. Esta configuración incluye el comando standby 2 con la dirección IP de la puerta de enlace de la VLAN 20, lo que convierte a R3 en el router activo para la VLAN 20 utilizando priority 200. y router pasivo para la VLAN 10. En la Figura 14, se visualiza la configuración del protocolo OSPF en Router 3, donde están directamente conectadas las subredes con las IPs 10.0, 20.0, 50.0 y 70.0, lo que permite la interconexión entre ellas.

```
!
interface FastEthernet0/0
 no ip address
 shutdown
 duplex half
!
interface GigabitEthernet1/0
 no ip address
 negotiation auto
!
interface GigabitEthernet1/0.10
 encapsulation dot1Q 10
 ip address 192.168.10.10 255.255.255.0
 standby version 2
 standby 1 ip 192.168.10.254
!
interface GigabitEthernet1/0.20
 encapsulation dot1Q 20
 ip address 192.168.20.10 255.255.255.0
 standby version 2
 standby 2 ip 192.168.20.254
 standby 2 priority 200
 standby 2 preempt
!
interface GigabitEthernet2/0
 no ip address
 shutdown
 negotiation auto
!
interface GigabitEthernet3/0
 ip address 192.168.50.2 255.255.255.0
 negotiation auto
!
interface GigabitEthernet4/0
 ip address 192.168.70.2 255.255.255.0
 negotiation auto
```

Figura 13: Configuración de Interfaz

```
router ospf 1
 router-id 3.3.3.3
 network 192.168.10.0 0.0.0.255 area 0
 network 192.168.20.0 0.0.0.255 area 0
 network 192.168.50.0 0.0.0.255 area 0
 network 192.168.70.0 0.0.0.255 area 0
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
!
!
no cdp log mismatch duplex
!
!
!
control-plane
!
!
!
mgcp profile default
!
!
!
gatekeeper
 shutdown
!
!
line con 0
 exec-timeout 0 0
```

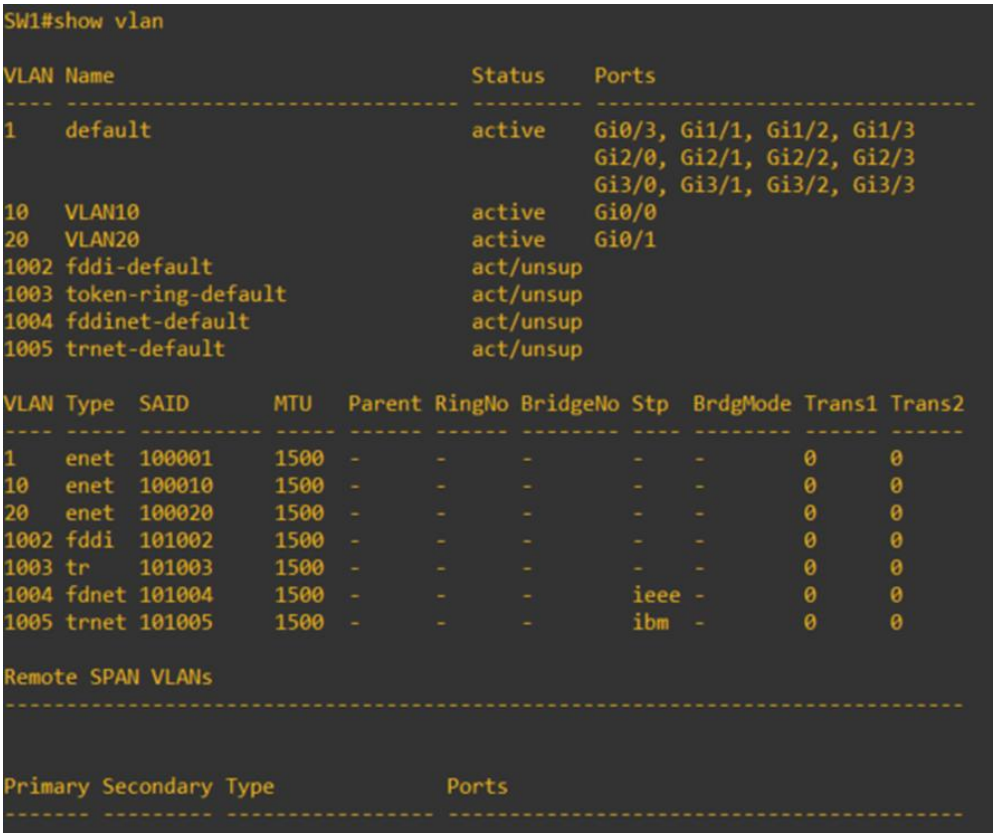
Figura 14: Verificación OSPF

Configuración del Switch SW1

Para la configuración del SW1, las interfaces 1/0 y 0/2 deben estar en modo trunk para que permita el envío de paquetes tanto del SW1 y del SW2, así como se puede visualizar en la configuración en la figura 15. Además, también se configuran las VLANs 10 y 20 tal como se ve en la figura 16.

```
!
interface GigabitEthernet0/0
 switchport access vlan 10
 switchport mode access
 media-type rj45
 negotiation auto
!
interface GigabitEthernet0/1
 switchport access vlan 20
 switchport mode access
 media-type rj45
 negotiation auto
!
interface GigabitEthernet0/2
 switchport trunk allowed vlan 10,20
 switchport trunk encapsulation dot1q
 switchport mode trunk
 media-type rj45
 negotiation auto
!
interface GigabitEthernet0/3
 media-type rj45
 negotiation auto
!
interface GigabitEthernet1/0
 switchport trunk allowed vlan 10,20
 switchport trunk encapsulation dot1q
 switchport mode trunk
 media-type rj45
 negotiation auto
!
interface GigabitEthernet1/1
 media-type rj45
```

Figura 15: Verificación interfaces modo troncal SW1



```
SW1#show vlan

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Gi0/3, Gi1/1, Gi1/2, Gi1/3
                                                 Gi2/0, Gi2/1, Gi2/2, Gi2/3
                                                 Gi3/0, Gi3/1, Gi3/2, Gi3/3
10   VLAN10                            active    Gi0/0
20   VLAN20                            active    Gi0/1
1002 fddi-default                      act/unsup
1003 token-ring-default                act/unsup
1004 fddinet-default                   act/unsup
1005 trnet-default                     act/unsup

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
10   enet  100010     1500  -      -      -        -    -        0      0
20   enet  100020     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
1004 fdnet 101004     1500  -      -      -        ieee -        0      0
1005 trnet 101005     1500  -      -      -        ibm  -        0      0

Remote SPAN VLANs
------------------------------------------------------------------------------


Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------
```

Figura 16: Verificación de VLANS SW1

Configuración del Switch SW2

La configuración es similar al SW1; las interfaces 1/0 y 0/2 deben estar en modo trunk para que permita el envío de paquetes tanto del SW2 y del SW1, así como se puede visualizar en la configuración en la Figura 17. Así mismo se configuran las VLANs 10 y 20, como podemos ver en la Figura 18.

```
interface GigabitEthernet0/0
 switchport access vlan 10
 switchport mode access
 media-type rj45
 negotiation auto
!
interface GigabitEthernet0/1
 switchport access vlan 20
 switchport mode access
 media-type rj45
 negotiation auto
!
interface GigabitEthernet0/2
 switchport trunk allowed vlan 10,20
 switchport trunk encapsulation dot1q
 switchport mode trunk
 media-type rj45
 negotiation auto
!
interface GigabitEthernet0/3
 media-type rj45
 negotiation auto
!
interface GigabitEthernet1/0
 switchport trunk allowed vlan 10,20
 switchport trunk encapsulation dot1q
 switchport mode trunk
 media-type rj45
 negotiation auto
!
interface GigabitEthernet1/1
 media-type rj45
 negotiation auto
!
interface GigabitEthernet1/2
 media-type rj45
```

Figura 17: Verificación interfaces modo troncal SW1

```
SW2#show vlan

VLAN Name                              Status    Ports
---- --------------------------------- --------- -------------------------------
1    default                           active    Gi0/3, Gi1/1, Gi1/2, Gi1/3
                                                 Gi2/0, Gi2/1, Gi2/2, Gi2/3
                                                 Gi3/0, Gi3/1, Gi3/2, Gi3/3
10   VLAN10                            active    Gi0/0
20   VLAN20                            active    Gi0/1
1002 fddi-default                      act/unsup
1003 token-ring-default                act/unsup
1004 fddinet-default                   act/unsup
1005 trnet-default                     act/unsup

VLAN Type  SAID       MTU   Parent RingNo BridgeNo Stp  BrdgMode Trans1 Trans2
---- ----- ---------- ----- ------ ------ -------- ---- -------- ------ ------
1    enet  100001     1500  -      -      -        -    -        0      0
10   enet  100010     1500  -      -      -        -    -        0      0
20   enet  100020     1500  -      -      -        -    -        0      0
1002 fddi  101002     1500  -      -      -        -    -        0      0
1003 tr    101003     1500  -      -      -        -    -        0      0
1004 fdnet 101004     1500  -      -      -        ieee -        0      0
1005 trnet 101005     1500  -      -      -        ibm  -        0      0

Remote SPAN VLANs
------------------------------------------------------------------------------


Primary Secondary Type              Ports
------- --------- ----------------- ------------------------------------------
```

Figura 18: Verificación de VLANS SW2

Tras completar la configuración inicial de los routers y switches, se procede al desarrollo e implementación de los scripts que serán utilizados para automatizar la gestión de la red.

Script de escaneo (get_connected_devices) identifica los dispositivos conectados en la red.

```
from scapy.all import ICMP, IP, sr1, conf
import time
import os
import warnings
from scapy.all import *

warnings.filterwarnings("ignore", ".*MAC address to reach
destination not found.*")

def get_connected_devices(network):
    """
    Obtiene las IPs de los dispositivos conectados a la red
utilizando nmap.
    """
    # Verificar si el script tiene permisos para usar sudo
    if os.geteuid() != 0:
        print("Este script necesita privilegios de sudo para
ejecutar nmap correctamente.")

    active_ips = []

    # Iterar sobre el rango de IPs de la subred
    for i in range(1, 12):
        ip = f"{network}.{i}"

        # Crear un paquete ICMP para hacer ping a la IP
        packet = IP(dst=ip)/ICMP()

        # Enviar el paquete y esperar una respuesta
        response = sr1(packet, timeout=0.3, verbose=False)

        # Si recibimos una respuesta, significa que el host está
activo
        if response:
            active_ips.append(ip)
    return active_ips

def calculate_bandwidth_per_device(total_bandwidth_kbps,
num_devices):
    """
    Calcula el ancho de banda por dispositivo.
    """
    if num_devices == 0:
        return 0
    return total_bandwidth_kbps / num_devices  # Convertir a kbps
```

Este script tiene como objetivo escanear una red local para identificar los dispositivos activos y calcular el ancho de banda disponible por dispositivo. Primero, utiliza la librería scapy para enviar paquetes ICMP a un rango de direcciones IP dentro de una subred, determinando cuáles dispositivos están activos en la red. Luego, a partir del número de dispositivos detectados y un ancho de banda total previamente definido, calcula el ancho de banda que se asignará a cada dispositivo. Este proceso permite monitorear la red y distribuir eficientemente los recursos de red, asegurando un uso adecuado del ancho de banda disponible.

Script para calcular el ancho de banda asignable por dispositivo utilizando calculate_bandwith_per_device.

```
import telnetlib
import time
from network_monitor import calculate_bandwidth_per_device
def set_qos_bandwidth_per_vlan(
    router_ip, vlan_interfaces,  upload_kbps, username=None,
password=None
):
    """
    Configura QoS en el router Cisco para limitar el ancho de banda de
descarga y subida por VLAN.
    """
    download_kbps = upload_kbps
    try:
        # Conectar al router vía Telnet
        tn = telnetlib.Telnet(router_ip)
        if username and password:
            tn.read_until(b"Username:", timeout=5)
            tn.write(username.encode("ascii") + b"\n")
            tn.read_until(b"Password:", timeout=5)
            tn.write(password.encode("ascii") + b"\n")
        tn.read_until(b">", timeout=5)
        # Entrar en modo privilegiado
        tn.write(b"enable\n")
        if password:
            tn.read_until(b"Password:", timeout=5)
            tn.write(password.encode("ascii") + b"\n")
        tn.read_until(b"#", timeout=5)
        # Entrar en modo de configuración global
        tn.write(b"configure terminal\n")
        tn.read_until(b"(config)#", timeout=5)
        # Configurar política para limitar la descarga (input)
        tn.write(f"policy-map DOWNLOAD_POLICY\n".encode("ascii"))
        tn.read_until(b"(config-pmap)#", timeout=5)
        tn.write(b"class class-default\n")
        tn.read_until(b"(config-pmap-c)#", timeout=5)
```

```
        normal_boost = int(int(download_kbps * 0.1) / 8)
        excess_boost = int(normal_boost * 2)
        tn.write(f"police {download_kbps*1000} {normal_boost}
{excess_boost} conform-action transmit exceed-action
drop\n".encode("ascii"))
        tn.read_until(b"(config-pmap-c-police)#", timeout=5)
        tn.write(b"exit\n")
        tn.read_until(b"(config-pmap-c)#", timeout=5)
        tn.write(b"exit\n")
        tn.read_until(b"(config-pmap)#", timeout=5)
        tn.write(b"exit\n")
        tn.read_until(b"(config)#", timeout=5)
        # Configurar política para limitar la subida (output)
        tn.write(f"policy-map UPLOAD_POLICY\n".encode("ascii"))
        tn.read_until(b"(config-pmap)#", timeout=5)
        tn.write(b"class class-default\n")
        tn.read_until(b"(config-pmap-c)#", timeout=5)
        tn.write(f"shape average {upload_kbps*1000}\n".encode("ascii"))
        tn.read_until(b"(config-pmap-c)#", timeout=5)
        tn.write(b"exit\n")
        tn.read_until(b"(config-pmap)#", timeout=5)
        tn.write(b"exit\n")
        tn.read_until(b"(config)#", timeout=5)
        # Aplicar las políticas a las interfaces VLAN
        for interface in vlan_interfaces:
            print(f"Aplicando QoS a {interface}")
            # Aplicar política de descarga (input)
            tn.write(f"interface {interface}\n".encode("ascii"))
            tn.read_until(b"(config-if)#", timeout=5)
            tn.write(f"service-policy input
DOWNLOAD_POLICY\n".encode("ascii"))
            tn.read_until(b"(config-if)#", timeout=5)
            # Aplicar política de subida (output)
            tn.write(f"service-policy output
UPLOAD_POLICY\n".encode("ascii"))
            tn.read_until(b"(config-if)#", timeout=5)
            tn.write(b"exit\n")
            tn.read_until(b"(config)#", timeout=5)
        # Guardar la configuración
        tn.write(b"write memory\n")
        tn.read_until(b"#", timeout=5)
        # Cerrar sesión
        tn.write(b"exit\n")
        tn.close()

    except Exception as e:
        pass

    print(f"QoS configurado en {router_ip}")
# Ejecución desde la consola
if __name__ == "__main__":
    router_ip = input("Ingrese la IP del router: ")
    vlan_interfaces = input(
        "Ingrese las interfaces VLAN separadas por comas (ejemplo:
GigabitEthernet0/1,GigabitEthernet0/2): "
```

```
    ).split(",")
    username = input("Ingrese el nombre de usuario: ")
    password = input("Ingrese la contraseña: ")
    total_bandwidth_mbps = float(input("Ingrese el ancho de banda total en
Mbps: "))
    bandwidth_kbps = total_bandwidth_mbps * 1000  # Convertir Mbps a Kbps
    # Configurar QoS
    set_qos_bandwidth_per_ip(
        router_ip, vlan_interfaces, bandwidth_kbps, username, password
    )
```

Configura políticas de QoS (Quality of Service) en un router Cisco, asignando límites de ancho de banda para las interfaces de VLAN. Este script asegura una distribución controlada del ancho de banda, especialmente en entornos donde las VLAN están segmentadas.

- Función principal: set_qos_bandwidth_per_vlan(router_ip, vlan_interfaces, upload_kbps, username, password)
    - Se conecta al router vía Telnet y configura las políticas de QoS.
    - Define dos políticas principales:
        1. Descarga (input): Limita el tráfico entrante mediante police.
        2. Subida (output): Limita el tráfico saliente mediante shape average.
    - Aplica estas políticas a las interfaces de VLAN especificadas.
- Permite gestionar varias VLAN desde un solo punto de configuración.
- Garantiza que cada VLAN tenga un ancho de banda específico, evitando el uso excesivo por parte de ciertos dispositivos o aplicaciones.

Script donde las políticas de QoS se configuran en los routers secundarios mediante set_qos_bandwidth_per_vlan, distribuyendo el ancho de banda de forma dinámica.

```
import time
from network_monitor import get_connected_devices,
calculate_bandwidth_per_device
from network_qos import set_qos_bandwidth_per_vlan

# Configuración básica
network_a = "192.168.10.0/24"
network_b = "192.168.20.0/24"
total_bandwidth = 200  # Ancho de banda total en Kbps
username = "cisco"  # Usuario Telnet
password = "cisco"  # Contraseña Telnet
sub_routers = ["192.168.50.2", "192.168.40.2"]
excepted_ips = [
    "192.168.10.9",
    "192.168.10.10",
    "192.168.10.254",
    "192.168.20.9",
    "192.168.20.10",
    "192.168.20.254",
]

sub_router_host_interfaces = ["GigabitEthernet 1/0"]

if __name__ == "__main__":
    print("Inicio del monitoreo y configuración de QoS dinámico.")

    processed_devices_count = 0

    while True:
        print("Escaneando redes...")
        # Escanear red y obtener dispositivos conectados
        #scanned_devices = get_connected_devices(network_a)
        scanned_devices = get_connected_devices(network_b)
        devices= []
        for device in scanned_devices:
            if device not in excepted_ips:
                devices.append(device)
                print(f"   {device}")

        if processed_devices_count == len(devices):
            print("Sin cambios")
            print("Hora de escaneo:", time.strftime("%H:%M:%S"))
            print("Reintentando en 30s...")
            time.sleep(30)
            continue

        print(f"Dispositivos conectados: {len(devices)}")
        processed_devices_count = len(devices)
        # Calcular ancho de banda por dispositivo
        if devices:
            bandwidth_kbps = calculate_bandwidth_per_device(
                total_bandwidth, len(devices)
            )
```

```
            print(f"Ancho de banda por dispositivo:
{bandwidth_kbps:.2f} kbps")

            for sub_router in sub_routers:
                print(f"Configurando QoS en el sub-router
{sub_router}")
                set_qos_bandwidth_per_vlan(
                    sub_router,
                    sub_router_host_interfaces,
                    int(bandwidth_kbps),
                    username,
                    password,
                )

            print("Configuración de QoS en dispositivos
completada.")
            print("Hora de escaneo:", time.strftime("%H:%M:%S"))
        else:
            print("No se detectaron dispositivos. No se configura
QoS.")

        print("Reintentando en 30s...")
        time.sleep(30)
```

Monitorea continuamente las redes definidas, identifica los dispositivos conectados y configura dinámicamente políticas de QoS en sub-routers, adaptando la asignación de ancho de banda según los cambios en la red.

- **Monitoreo dinámico:**
  - Escanea las redes en intervalos regulares de tiempo (30 segundos).
  - Detecta cambios en los dispositivos conectados y ajusta las configuraciones en consecuencia.
- **Cálculo del ancho de banda:**
  - Utiliza la función calculate_bandwidth_per_device para distribuir equitativamente el ancho de banda disponible entre los dispositivos detectados.
- **Configuración en sub-routers:**
  - Usa set_qos_bandwidth_per_vlan para aplicar las políticas de QoS en routers secundarios, asegurando un control granular del tráfico en las VLAN.
- Dispositivos críticos definidos en excepted_ips se excluyen de las políticas de QoS para evitar interrupciones en su conectividad.

Script de asignación de roles.

```
import subprocess
import telnetlib
import time
import os
from scapy.all import ICMP, IP, sr1, conf
import warnings
from scapy.all import *

warnings.filterwarnings("ignore", ".MAC address to reach
destination not found.*")

# Archivo para registrar los usuarios creados
USERS_FILE = "usuarios_creados.txt"
range_to_scan = 12

# Función para leer el archivo y obtener los usuarios existentes
def read_existing_users():
    if not os.path.exists(USERS_FILE):
        return {}
    with open(USERS_FILE, "r") as file:
        users = {}
        for line in file:
            user, ip = line.strip().split(", ")
            users[user] = ip
        return users

# Función para sobrescribir el archivo con los usuarios actuales
def overwrite_user_file(users):
    with open(USERS_FILE, "w") as file:
        for username, ip in users.items():
            file.write(f"{username}, {ip}\n")

# Función para agregar un usuario al archivo
def add_user_to_file(username, ip, users):
    users[username] = ip
    overwrite_user_file(users)

# Función para eliminar un usuario del archivo
def remove_user_from_file(username, users):
    if username in users:
        del users[username]
        overwrite_user_file(users)

# Función para asignar la vista según la VLAN
def assign_view_by_vlan(ip):
    # Asignación de vistas según la VLAN
    if ip.startswith("192.168.10."):
        return "root"  # Vista root para VLAN 10
    elif ip.startswith("192.168.20."):
        return "SOPORTE"  # Vista SOPORTE para VLAN 20
    return None
```

```
# Función para crear un usuario en el router Cisco
def create_user_on_router(router_ip, username, password, view):
    try:
        # Conectar al router
        tn = telnetlib.Telnet(router_ip)
        tn.read_until(b"Username:", timeout=5)
        tn.write("cisco".encode("ascii") + b"\n")
        tn.read_until(b"Password:", timeout=5)
        tn.write("cisco".encode("ascii") + b"\n")

        tn.read_until(b">", timeout=5)
        # Entrar en modo privilegiado
        tn.write(b"enable\n")

        tn.read_until(b"Password:", timeout=5)
        tn.write("cisco".encode("ascii") + b"\n")

        tn.read_until(b"#", timeout=5)
        # Entrar en modo de configuración global
        command = "configure terminal\n"
        print(f"{router_ip}: {command}", end="")
        tn.write(command.encode("ascii"))
        tn.read_until(b"(config)#", timeout=5)

        # Crear usuario
        command = f"username {username} view {view} secret
{password}\n"
        print(f"{router_ip}: {command}", end="")
        tn.write(command.encode("ascii"))
        tn.read_until(b"(config)#", timeout=5)

        # Guardar la configuración
        command = "end\n"
        print(f"{router_ip}: {command}", end="")
        tn.write(command.encode("ascii"))
        tn.read_until(b"#", timeout=5)
        command = "write memory\n"
        print(f"{router_ip}: {command}", end="")
        tn.write(command.encode("ascii"))
        tn.read_until(b"#", timeout=5)
        tn.close()

        print(f"Usuario {username} creado con vista {view} en
{router_ip}")
    except Exception as e:
        print(f"Error al crear el usuario {username} en
{router_ip}: {e}")
```

```
# Función para eliminar un usuario del router Cisco
def delete_user_from_router(router_ip, username, password):
    try:
        # Conectar al router
        tn = telnetlib.Telnet(router_ip)
        tn.read_until(b"Username:", timeout=5)
        tn.write("cisco".encode("ascii") + b"\n")
        tn.read_until(b"Password:", timeout=5)
        tn.write("cisco".encode("ascii") + b"\n")
        tn.read_until(b"#", timeout=5)

        # Entrar en modo de configuración global
        tn.write(b"configure terminal\n")
        tn.read_until(b"(config)#", timeout=5)

        # Eliminar usuario
        tn.write(f"no username {username}\n".encode("ascii"))
        tn.read_until(b"(config)#", timeout=5)

        # Salir de la configuración
        tn.write(b"exit\n")
        tn.read_until(b"#", timeout=5)

        tn.write(b"exit\n")
        tn.close()

        print(f"Usuario {username} eliminado del router
{router_ip}")
    except Exception as e:
        print(f"Error al eliminar el usuario {username} en
{router_ip}: {e}")

# Función para crear un usuario para un dispositivo en la red
def create_user_for_device(router_ip, ip, username, password,
users):
    # Asignar la vista según la VLAN
    view = assign_view_by_vlan(ip)
    if view:
        # Crear el usuario en el router
        create_user_on_router(router_ip, username, password, view)
        # Registrar el usuario en el archivo
        add_user_to_file(username, ip, users)
    else:
        print(f"Dispositivo con IP {ip} no tiene una VLAN asignada
adecuada.")

# Función para eliminar un usuario si el dispositivo ya no está en
la red
def delete_user_if_device_offline(router_ip, username, ip,
devices, users):
```

```
# Verificar si la IP está en la lista de dispositivos conectados
    if ip not in devices:
        # Eliminar el usuario del router
        delete_user_from_router(router_ip, username, password)
        # Eliminar el usuario del archivo
        remove_user_from_file(username, users)

def get_connected_devices(network):
    """
    Obtiene las IPs de los dispositivos conectados a la red
utilizando nmap.
    """
    # Verificar si el script tiene permisos para usar sudo
    if os.geteuid() != 0:
        print("Este script necesita privilegios de sudo para
ejecutar nmap correctamente.")

    active_ips = []

    # Iterar sobre el rango de IPs de la subred
    for i in range(1, range_to_scan + 1):
        ip = f"{network}.{i}"

        # Crear un paquete ICMP para hacer ping a la IP
        packet = IP(dst=ip)/ICMP()

        # Enviar el paquete y esperar una respuesta
        response = sr1(packet, timeout=0.3, verbose=False)

        # Si recibimos una respuesta, significa que el host está
activo
        if response:
            active_ips.append(ip)

    return active_ips

# Función principal que gestiona la creación de usuarios
def manage_users(router_ip, network_a, network_b, username,
password):
    # Leer los usuarios ya creados
    existing_users = read_existing_users()
    sub_routers = ["192.168.50.2", "192.168.40.2"]
    # Detectar dispositivos en las redes
    print("Escaneando dispositivos en VLAN 10...")
    devices_a = get_connected_devices(network_a)
    print("Escaneando dispositivos en VLAN 20...")
    devices_b = get_connected_devices(network_b)

    print("Listado de dispositivos...")
    for device in devices_a + devices_b:
        print(device)
```

```
    # Procesar los dispositivos de la VLAN 10
    for ip in devices_a:

        # Comprobar si el usuario ya ha sido creado
        new_user = f"user_{ip.replace('.', '_')}"
        if new_user not in existing_users:
            for sub_router in sub_routers:
                create_user_for_device(sub_router, ip, new_user,
password, existing_users)

    # Procesar los dispositivos de la VLAN 20
    for ip in devices_b:
        # Comprobar si el usuario ya ha sido creado
        new_user = f"user_{ip.replace('.', '_')}"
        if new_user not in existing_users:
            for sub_router in sub_routers:
                create_user_for_device(sub_router, ip, new_user,
password, existing_users)

    # Verificar si algún usuario ya no está en la red
    all_devices = devices_a + devices_b
    for username, ip in list(existing_users.items()):
        for sub_router in sub_routers:
            delete_user_if_device_offline(sub_router, username,
ip, all_devices, existing_users)

# Ejecución del script
if __name__ == "__main__":
    username = "cisco"
    password = "cisco"


    # Rango de IPs para VLAN 10 y VLAN 20
    network_a = "192.168.10"
    network_b = "192.168.20"
    while True:
        start_time = time.time()
        print("Inicio del monitoreo y configuración de QoS
dinámico.")
        print("Hora de escaneo:", time.strftime("%H:%M:%S"))

        # Llamar a la función para gestionar los usuarios
        manage_users(network_a, network_b, username, password)

        end_time = time.time()
        print("Fin del monitoreo y configuración de QoS
dinámico.")
        print("Tiempo total de ejecución:", end_time - start_time,
"segundos")

        print("Reinicio en 30s...")
        time.sleep(30)
```

El propósito de este script es automatizar la administración de usuarios en enrutadores Cisco, fundamentado en los dispositivos vinculados a dos subredes distintas. Emplea Telnet para establecer y eliminar usuarios en el enrutador, otorgando distintos niveles de acceso en función de la VLAN a la que se encuentra el dispositivo. Los equipos en la red "192.168.10" se establecen con la vista "root"; en cambio, los equipos en "192.168.20" se establecen con la vista "SOPORTE", que solo les brinda la posibilidad de visualizar. Los usuarios se guardan en un documento de texto "usuarios_creados.txt", en el que se anotan las IPs de los dispositivos junto con sus nombres de usuario correspondientes.

El script escanea continuamente las dos subredes para detectar dispositivos activos utilizando paquetes ICMP (ping). Los dispositivos que responden al ping son considerados activos y se les crea un usuario en el router si aún no existe uno para ellos. La creación del usuario incluye la asignación de la vista correspondiente según la IP del dispositivo (si está en “192.168.10”, se le asigna “root”; si está en “192.168.20”, se le asigna “SOPORTE”). Además, si algún dispositivo se desconecta, el script elimina automáticamente el usuario asociado en el router y en el archivo de usuarios.

El ciclo de ejecución del script se repite cada 30 segundos, asegurando que los usuarios en el router se mantengan actualizados según los dispositivos activos en la red. Si un dispositivo deja de estar conectado, el usuario correspondiente es eliminado tanto del router como del archivo de registro. Este proceso se lleva a cabo de manera continua, lo que facilita la administración dinámica de los usuarios sin intervención manual.

2.4.Ejecución de prototipo

2.4.1. Fase 3: Verificación

Verificación de las conexiones entre diferentes dispositivos de la topología.

En esta fase, verificaremos las configuraciones realizadas previamente. Para ello, llevaremos a cabo pruebas de conectividad mediante el uso del comando PING desde el PC físico hacia los diferentes dispositivos presentes en nuestra topología de red.

Ping hacia la 192.168.40.0:

```
C:\Users\ivanl>ping 192.168.40.1

Haciendo ping a 192.168.40.1 con 32 bytes de datos:
Respuesta desde 192.168.40.1: bytes=32 tiempo=12ms TTL=255
Respuesta desde 192.168.40.1: bytes=32 tiempo=15ms TTL=255
Respuesta desde 192.168.40.1: bytes=32 tiempo=15ms TTL=255
Respuesta desde 192.168.40.1: bytes=32 tiempo=15ms TTL=255

Estadísticas de ping para 192.168.40.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 12ms, Máximo = 15ms, Media = 14ms

C:\Users\ivanl>ping 192.168.40.2

Haciendo ping a 192.168.40.2 con 32 bytes de datos:
Respuesta desde 192.168.40.2: bytes=32 tiempo=42ms TTL=254
Respuesta desde 192.168.40.2: bytes=32 tiempo=46ms TTL=254
Respuesta desde 192.168.40.2: bytes=32 tiempo=46ms TTL=254
Respuesta desde 192.168.40.2: bytes=32 tiempo=47ms TTL=254

Estadísticas de ping para 192.168.40.2:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 42ms, Máximo = 47ms, Media = 45ms
```

Figura 19: Verificación de salida

Ping hacia la 192.168.50.0

```
C:\Users\ivanl>ping 192.168.50.1

Haciendo ping a 192.168.50.1 con 32 bytes de datos:
Respuesta desde 192.168.50.1: bytes=32 tiempo=12ms TTL=255
Respuesta desde 192.168.50.1: bytes=32 tiempo=15ms TTL=255
Respuesta desde 192.168.50.1: bytes=32 tiempo=15ms TTL=255
Respuesta desde 192.168.50.1: bytes=32 tiempo=15ms TTL=255

Estadísticas de ping para 192.168.50.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 12ms, Máximo = 15ms, Media = 14ms

C:\Users\ivanl>ping 192.168.50.2

Haciendo ping a 192.168.50.2 con 32 bytes de datos:
Respuesta desde 192.168.50.2: bytes=32 tiempo=42ms TTL=254
Respuesta desde 192.168.50.2: bytes=32 tiempo=46ms TTL=254
Respuesta desde 192.168.50.2: bytes=32 tiempo=46ms TTL=254
Respuesta desde 192.168.50.2: bytes=32 tiempo=46ms TTL=254

Estadísticas de ping para 192.168.50.2:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 42ms, Máximo = 46ms, Media = 45ms
```

Figura 20: Verificación de salida

Ping hacia la 192.168.70.0

```
C:\Users\ivanl>ping 192.168.70.1

Haciendo ping a 192.168.70.1 con 32 bytes de datos:
Respuesta desde 192.168.70.1: bytes=32 tiempo=62ms TTL=254
Respuesta desde 192.168.70.1: bytes=32 tiempo=62ms TTL=254
Respuesta desde 192.168.70.1: bytes=32 tiempo=62ms TTL=254
Respuesta desde 192.168.70.1: bytes=32 tiempo=62ms TTL=254

Estadísticas de ping para 192.168.70.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 62ms, Máximo = 62ms, Media = 62ms

C:\Users\ivanl>ping 192.168.70.2

Haciendo ping a 192.168.70.2 con 32 bytes de datos:
Respuesta desde 192.168.70.2: bytes=32 tiempo=42ms TTL=254
Respuesta desde 192.168.70.2: bytes=32 tiempo=46ms TTL=254
Respuesta desde 192.168.70.2: bytes=32 tiempo=46ms TTL=254
Respuesta desde 192.168.70.2: bytes=32 tiempo=46ms TTL=254

Estadísticas de ping para 192.168.70.2:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 42ms, Máximo = 46ms, Media = 45ms
```

Figura 21: Verificación de salida

Ping hacia la 192.168.10.1

```
C:\Users\ivanl>ping 192.168.10.1

Haciendo ping a 192.168.10.1 con 32 bytes de datos:
Respuesta desde 192.168.10.1: bytes=32 tiempo=109ms TTL=62
Respuesta desde 192.168.10.1: bytes=32 tiempo=77ms TTL=62
Respuesta desde 192.168.10.1: bytes=32 tiempo=61ms TTL=62
Respuesta desde 192.168.10.1: bytes=32 tiempo=62ms TTL=62

Estadísticas de ping para 192.168.10.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 61ms, Máximo = 109ms, Media = 77ms

C:\Users\ivanl>ping 192.168.10.2

Haciendo ping a 192.168.10.2 con 32 bytes de datos:
Respuesta desde 192.168.10.2: bytes=32 tiempo=97ms TTL=62
Respuesta desde 192.168.10.2: bytes=32 tiempo=62ms TTL=62
Respuesta desde 192.168.10.2: bytes=32 tiempo=77ms TTL=62
Respuesta desde 192.168.10.2: bytes=32 tiempo=62ms TTL=62

Estadísticas de ping para 192.168.10.2:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 62ms, Máximo = 97ms, Media = 74ms
```

Figura 22: Verificación de salida

Ping hacia la 192.168.20.1

```
C:\Users\ivanl>ping 192.168.20.1

Haciendo ping a 192.168.20.1 con 32 bytes de datos:
Respuesta desde 192.168.20.1: bytes=32 tiempo=106ms TTL=62
Respuesta desde 192.168.20.1: bytes=32 tiempo=77ms TTL=62
Respuesta desde 192.168.20.1: bytes=32 tiempo=77ms TTL=62
Respuesta desde 192.168.20.1: bytes=32 tiempo=61ms TTL=62

Estadísticas de ping para 192.168.20.1:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 61ms, Máximo = 106ms, Media = 80ms

C:\Users\ivanl>ping 192.168.20.2

Haciendo ping a 192.168.20.2 con 32 bytes de datos:
Respuesta desde 192.168.20.2: bytes=32 tiempo=69ms TTL=62
Respuesta desde 192.168.20.2: bytes=32 tiempo=62ms TTL=62
Respuesta desde 192.168.20.2: bytes=32 tiempo=77ms TTL=62
Respuesta desde 192.168.20.2: bytes=32 tiempo=77ms TTL=62

Estadísticas de ping para 192.168.20.2:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 62ms, Máximo = 77ms, Media = 71ms
```

Figura 23: Verificación de salida

Verificación del funcionamiento del script de la gestión de ancho de banda automatizada.

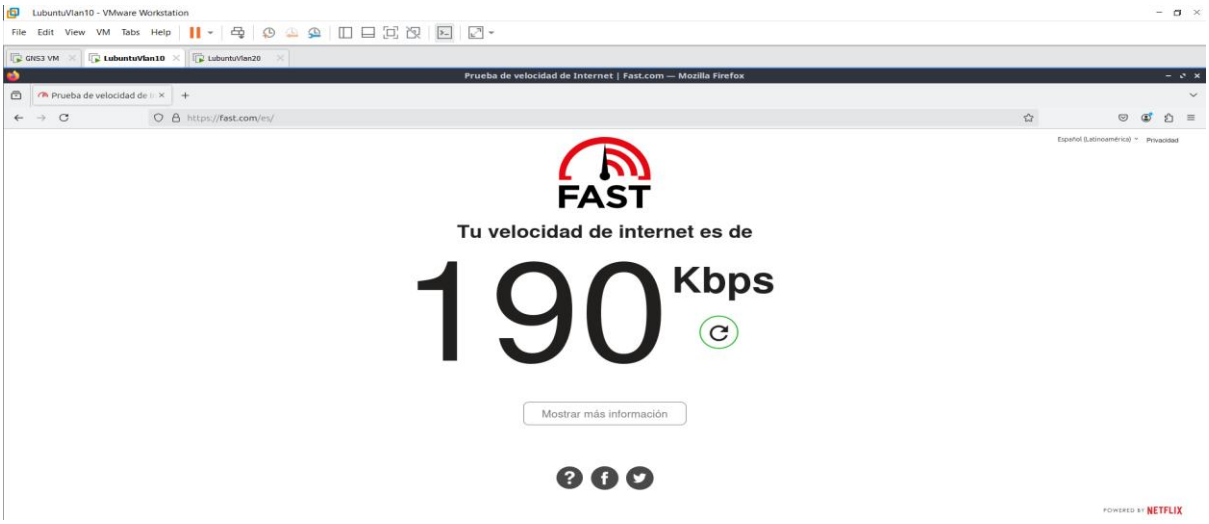


Figura 24: Prueba de velocidad sin aplicar la gestión de ancho de banda.

En la Figura 24 se observa la realización de una prueba de velocidad previa a la implementación de la gestión de ancho de banda, obteniendo un resultado de 190 Kbps; sin embargo, le damos un rango de más menos 10 Kbps según las varias pruebas realizadas y para el script dejaremos un promedio de 200 Kbps.

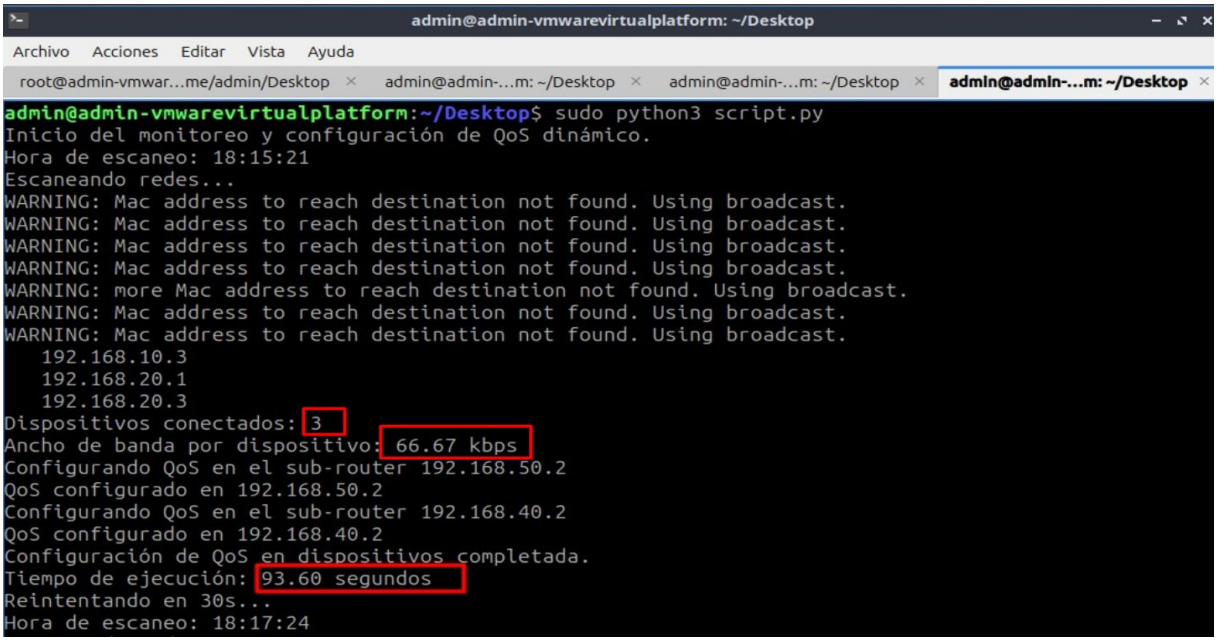


Figura 25: Ejecución del Script

En la Figura 25, se observa la ejecución del script que detecta tres dispositivos activos en la red. Con esta información, el ancho de banda total disponible se distribuye de manera equitativa entre los dispositivos identificados, lo que resulta en la asignación de 66.67 Kbps para cada uno.

Figura 26: Prueba de velocidad aplicada la automatización

En la Figura 26, se confirma que la automatización de la gestión del ancho de banda se implementó correctamente en los dispositivos conectados. Como resultado, se realizó una prueba de velocidad que arrojó un valor de 60 Kbps, evidenciando la correcta distribución de los recursos de la red.

Verificación del funcionamiento del script de asignación de roles.

```
admin@admin-vmwarevirtualplatform: ~/Desktop
admin@admin-vmwarevirtualplatform:~/Desktop$ sudo python3 codigo
[sudo] password for admin:
/home/admin/Desktop/codigo:2: DeprecationWarning: 'telnetlib' is deprecated and slated for removal in Python 3.13
  import telnetlib
Inicio del monitoreo y configuración de QoS dinámico.
Hora de escaneo: 12:35:26
Escaneando dispositivos en VLAN 10...
WARNING: Mac address to reach destination not found. Using broadcast.
WARNING: Mac address to reach destination not found. Using broadcast.
WARNING: Mac address to reach destination not found. Using broadcast.
WARNING: Mac address to reach destination not found. Using broadcast.
WARNING: more Mac address to reach destination not found. Using broadcast.
WARNING: Mac address to reach destination not found. Using broadcast.
WARNING: Mac address to reach destination not found. Using broadcast.
WARNING: more Mac address to reach destination not found. Using broadcast.
Escaneando dispositivos en VLAN 20...
Listado de dispositivos...
192.168.10.3
192.168.10.9
192.168.10.10
192.168.20.1
192.168.20.3
192.168.20.9
192.168.20.10
192.168.50.2: configure terminal
192.168.50.2: username user_192_168_10_3 view root secret cisco
192.168.50.2: end
192.168.50.2: write memory
Usuario user_192_168_10_3 creado con vista root en 192.168.50.2
192.168.40.2: configure terminal
192.168.40.2: username user_192_168_10_3 view root secret cisco
```

Figura 27: Ejecución de script asignación de roles

En la Figura 27, se están enlistando 7 dispositivos que serán configurados, tanto los de la VLAN 20 como SOPORTE y VLAN 10 como ROOT.

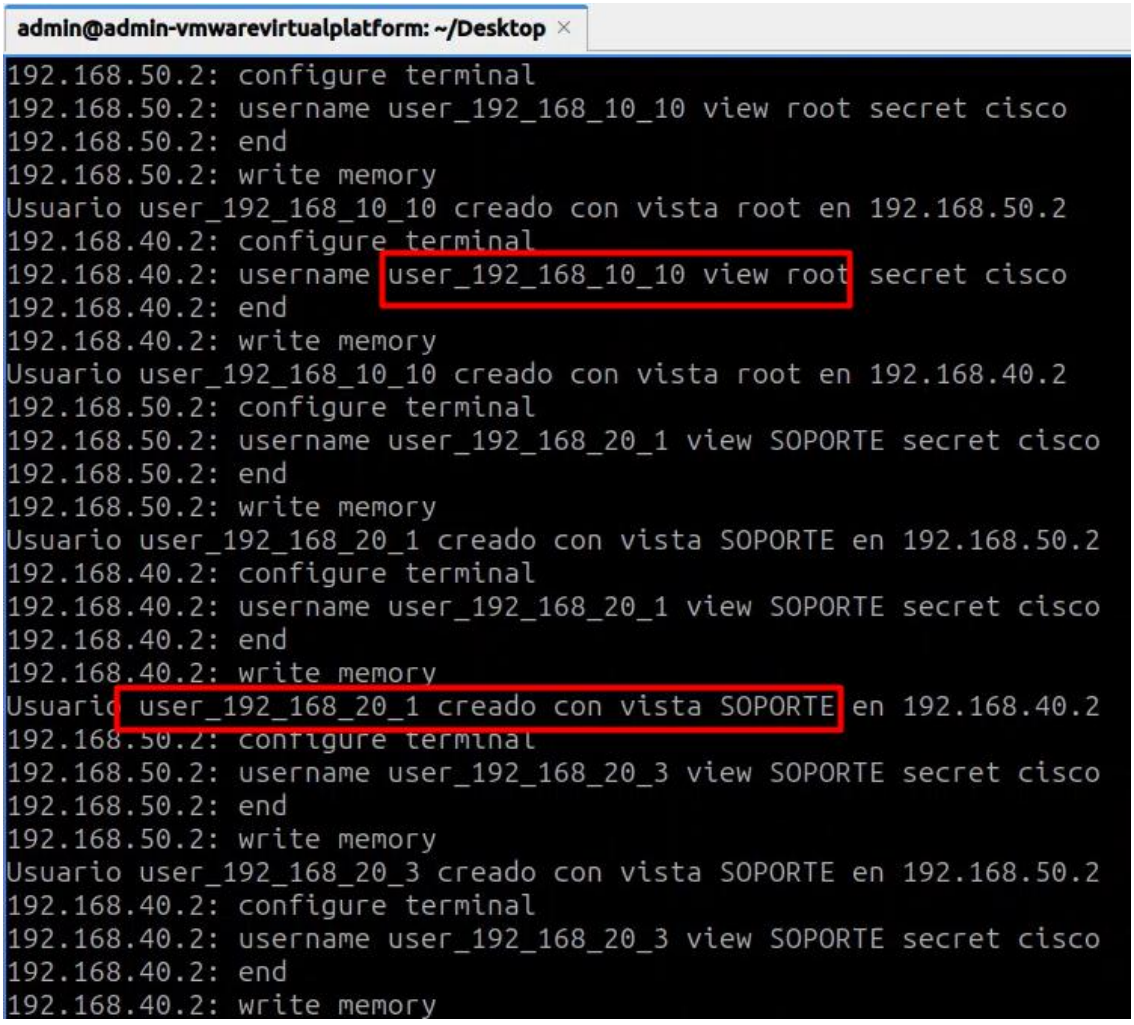


Figura 28; Asignación de roles a nuevos dispositivos

Se evidencia en la Figura 28 cómo se están configurando los nuevos dispositivos según la VLAN a la que pertenecen, asignándoles los roles de SOPORTE y ROOT según como corresponden.

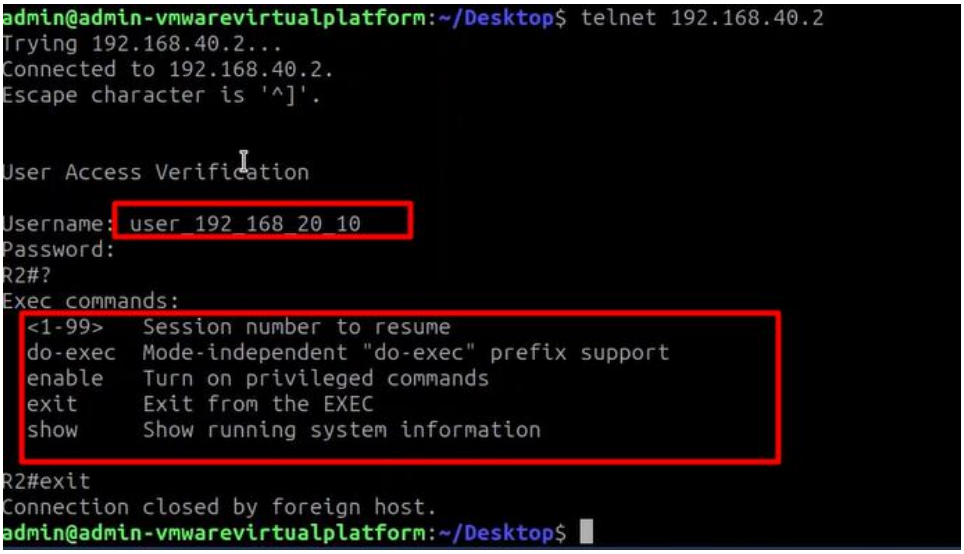


Figura 29: Verificación de la Asignación del rol SOPORTE

En la Figura 29, se accede al router 20.10 a través de Telnet para verificar la implementación del rol asignado. Esto se evidencia al revisar los comandos disponibles en el router, observándose que únicamente tiene acceso al comando **show**, lo cual confirma la correcta configuración del rol.

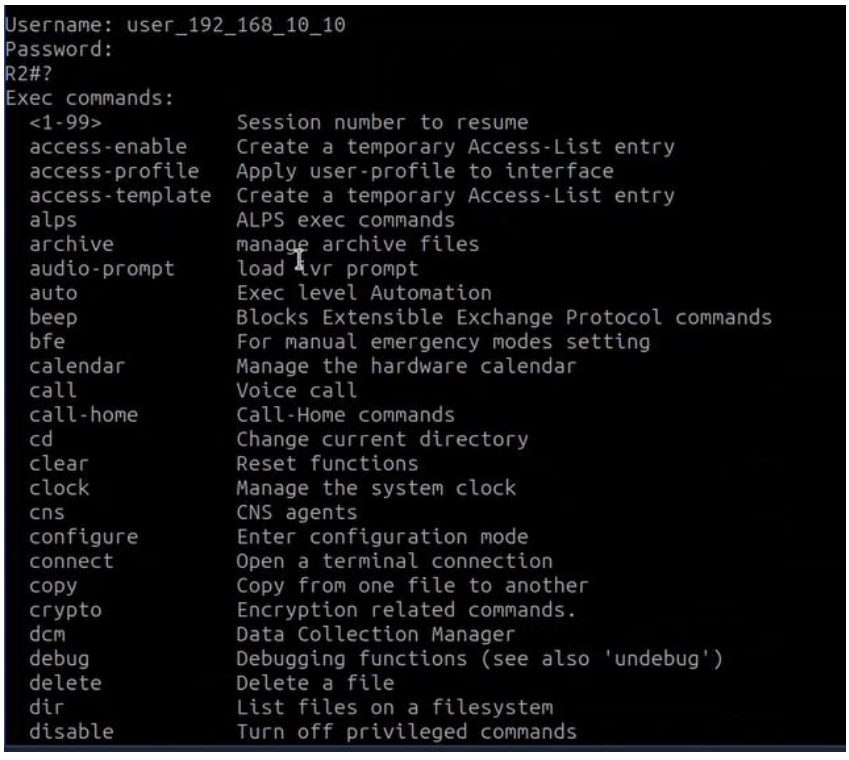


Figura 30: Verificación de la Asignación del rol ROOT

De igual manera que se verificó la asignación del rol SOPORTE, se revisa si el rol ROOT ha sido aplicado correctamente. En la Figura 30, se observa que el rol ROOT dispone de acceso a un conjunto ampliado de comandos, lo que confirma su asignación y las capacidades adicionales que conlleva.

3. CAPITULO III. EVALUACIÓN DELPROTOTIPO

3.1. Plan de evaluación

3.1.1. Objetivo

Evaluar la efectividad del sistema automatizado en la gestión del ancho de banda y la asignación de roles para asegurar que cumple con los objetivos de optimización de recursos y reducción de la carga administrativa.

3.1.2. Criterios de evaluación

- **Eficiencia en el uso de ancho de banda:** Medir la capacidad del sistema para optimizar el ancho de banda disponible.
- **Tiempo de respuesta de la red:** Comparar los tiempos de respuesta con o sin automatización.
- **Precisión en la asignación de roles:** Verificar si la asignación automática de roles es consistente y cumple con las reglas de acceso establecidas.
- **Reducción de la carga administrativa:** Evaluar el impacto en la reducción de tareas manuales de configuración y gestión.

3.1.3. Cronograma de Actividades

Tabla 6: Cronograma de actividades del plan de evaluación

Actividad	Semana 13					Semana 14				
	1	2	3	4	5	1	2	3	4	5
Configuración inicial del entorno de simulación en GNS3 y herramientas (Wireshark, scripts).										
Validación del entorno: pruebas básicas de conectividad y monitoreo de tráfico en Wireshark.										
Realización de pruebas de ancho de banda (escenarios: carga baja, media y alta).										
Análisis preliminar de los datos de ancho de banda recopilados.										

Actividad	Semana 13					Semana 14				
	1	2	3	4	5	1	2	3	4	5
Realización de pruebas de latencia con múltiples dispositivos activos.										
Validación de la asignación de roles (dispositivos y usuarios según políticas).										
Recopilación de métricas sobre la reducción de la carga administrativa.										
Revisión y análisis de resultados obtenidos (ancho de banda, latencia, roles, etc.).										
Realización de pruebas finales para verificar mejoras tras ajustes.										
Revisión del informe y preparación de recomendaciones finales.										
Presentación del informe de evaluación final										

3.1.4. Métricas de rendimiento

En el contexto de la automatización de redes, la evaluación del rendimiento es un aspecto fundamental para verificar la eficiencia, fiabilidad y efectividad de las configuraciones implementadas. Estas métricas nos permiten analizar el impacto de los métodos manuales a comparación de las soluciones automatizadas; a continuación, en la tabla 7 se especifica cada una de las métricas que implementaron.

Tabla 7: Métricas de Evaluación

Métrica 1: Tiempo de configuración manual	Tiempo requerido para realizar la configuración manual en redes, incluyendo ajustes del ancho de banda y asignación de roles.
Métrica 2: Tiempo de configuración automática	Tiempo necesario para implementar la configuración de redes utilizando herramientas o scripts de automatización.

Métrica 3: Precisión de la automatización	Medida que evalúa la exactitud de los resultados obtenidos mediante la configuración automatizada.
Métrica 4: Precisión de configuración manual	Medida que evalúa la exactitud de los resultados obtenidos mediante la configuración manual.
Métrica 5: Latencia	Retraso observado en el envío de paquetes ICMP.

3.1.5. Metodología de evaluación

- **Entorno de prueba:** Utilizar un entorno simulado en GNS3 que emule las condiciones de la red real.
- **Procedimiento de prueba:** Implementar un protocolo de prueba estandarizado que incluya:
  - **Pruebas de ancho de banda:** Ejecutar una serie de pruebas de carga para verificar que el sistema distribuye el ancho de banda de forma óptima.
  - **Pruebas de latencia**: Medir la latencia con múltiples dispositivos activos en la red.
  - **Validación de asignación de roles**: Verificar que cada dispositivo y usuario se asigne correctamente a roles según las políticas establecidas.
- **Herramientas de análisis:** Utilizar Wireshark para capturar y analizar los datos de tráfico en la red.

3.1.6. Etapas de evaluación

- **Preparación:** Configurar el entorno de simulación en GNS3, configurar Wireshark para monitorear y ajustar el sistema automatizado para capturar métricas precisas.
- **Ejecutar pruebas:** Realizar una serie de pruebas para revisar el funcionamiento correcto de los scripts sobre los dispositivos finales.
- **Revisión de resultados:** Analizar los datos obtenidos y compararlos con los valores de referencia.

- **Documentación y ajustes:** Documentar todos los resultados y realizar ajustes en los scripts o configuraciones si los resultados no cumplen con los objetivos esperados.

3.1.7. Informe de evaluación

- **Resumen de resultados:** Incluir gráficos y tablas que muestren las métricas de rendimiento obtenidas.
- **Análisis de desempeño:** Describir en qué áreas el sistema cumplió o no con los objetivos planteados.
- **Recomendaciones:** Incluir ajustes recomendados para mejorar el sistema de automatización según los resultados obtenidos.

3.2. Resultados de evaluación

Tras realizar las ejecuciones, se llevaron a cabo pruebas en diferentes escenarios, donde se asignaron usuarios finales a distintas VLANs. Cada escenario representa una cantidad específica de dispositivos o usuarios finales conectados en la topología de red. Estos escenarios fueron diseñados con el propósito de evaluar el impacto de la configuración, analizar los tiempos necesarios para su implementación y verificar la precisión, así como los posibles errores asociados a los procesos de configuración.

Escenario 1: Configuración con 3 dispositivos conectados

En este escenario inicial, se simula un entorno pequeño con solo tres dispositivos conectados. Este caso representa una configuración básica, como la que podría encontrarse en una red doméstica o en una pequeña oficina. El objetivo principal de este escenario es evaluar el tiempo y la precisión de las configuraciones manuales y automáticas en un entorno de baja complejidad.

Escenario 2: Configuración con 5 dispositivos conectados

El segundo escenario incrementa la cantidad de dispositivos a cinco, simulando un entorno de tamaño medio; este caso podría reflejar las necesidades de una pequeña empresa o una red con usuarios moderados. En este escenario, la complejidad aumenta ligeramente debido a la mayor cantidad de dispositivos que requieren configuración, lo que permite analizar cómo la automatización gestiona eficientemente una carga de trabajo más significativa en comparación con la configuración manual.

Escenario 3: Configuración con 7 dispositivos conectados

En este último escenario, se simula un entorno más grande con siete usuarios finales conectados. Este escenario se asemeja a la configuración de una red en una oficina mediana o en un entorno educativo, donde se requiere la configuración de varios dispositivos simultáneamente. Aquí se evalúa la escalabilidad de las herramientas de automatización y su capacidad para mantener tiempos consistentes y reducir errores, incluso en configuraciones más complejas.

Tabla 8: Resultados de evaluación asignación del ancho de banda

Escenarios	Configuración Manual (min)	Configuración Automática (min)	Precisión de Automatización	Precisión de configuración manual	Latencia
Escenario 1	40	1.52	96%	80%	17,75
Escenario 2	60	1.55	93%	75%	28,75
Escenario 3	120	1.56	98%	67%	53,75

Tabla 9: Resultados de la evaluación asignación de roles

Escenarios	Configuración Manual (min)	Configuración Automática (min)	Precisión de Automatización	Precisión de configuración manual	Latencia
Escenario 1	35	3,42	98 %	85 %	70,75
Escenario 2	45	5,72	97 %	75 %	84,00
Escenario 3	60	8,61	99 %	88 %	39,75

A partir de los datos obtenidos en la evaluación, podemos observar que tenemos una latencia buena; se puede observar en los Anexos 1 y 2 los resultados de latencia. Se realizan gráficos que representen los datos obtenidos de manera más detallada para cada una de las asignaciones.

Evaluación asignación ancho de banda Gráfica

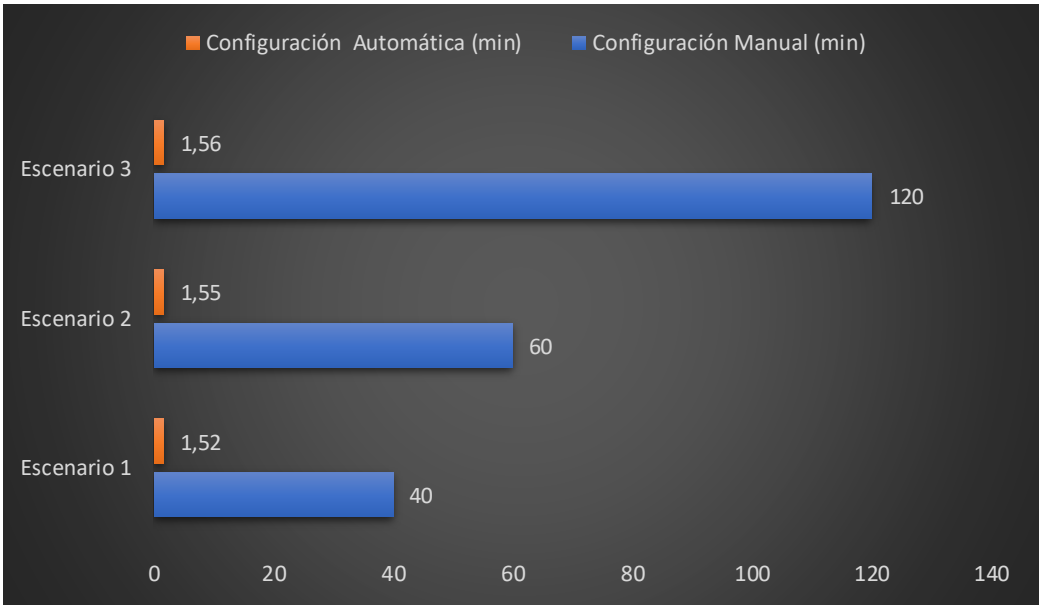


Figura 31: Comparación de Tiempo Configuración Manual y Automatizada en la asignación de ancho de banda

En la Figura 27 se presenta la evaluación del tiempo requerido para realizar las configuraciones en redes, comparando métodos manual y automático; los datos recopilaron información de los 3 escenarios.

La configuración manual representada por las barras de color azul es un método que podemos identificar que demanda significativamente mayor tiempo en todos los escenarios analizados. En el caso del escenario 3, es el que más tiempo logro alcanzar por el número de dispositivos que están conectados; se observa que la configuración manual alcanza un punto máximo con un total de aproximadamente 220 minutos abarcando todos los escenarios.

La configuración automática representada por las barras de color naranja es un método que demuestra notablemente la eficiencia en los tiempos de configuración, manteniendo tiempos muy bajos en comparación con el método manual. En cada uno de los escenarios, el tiempo de configuración es casi imperceptible en relación con el tiempo de configuración manual.

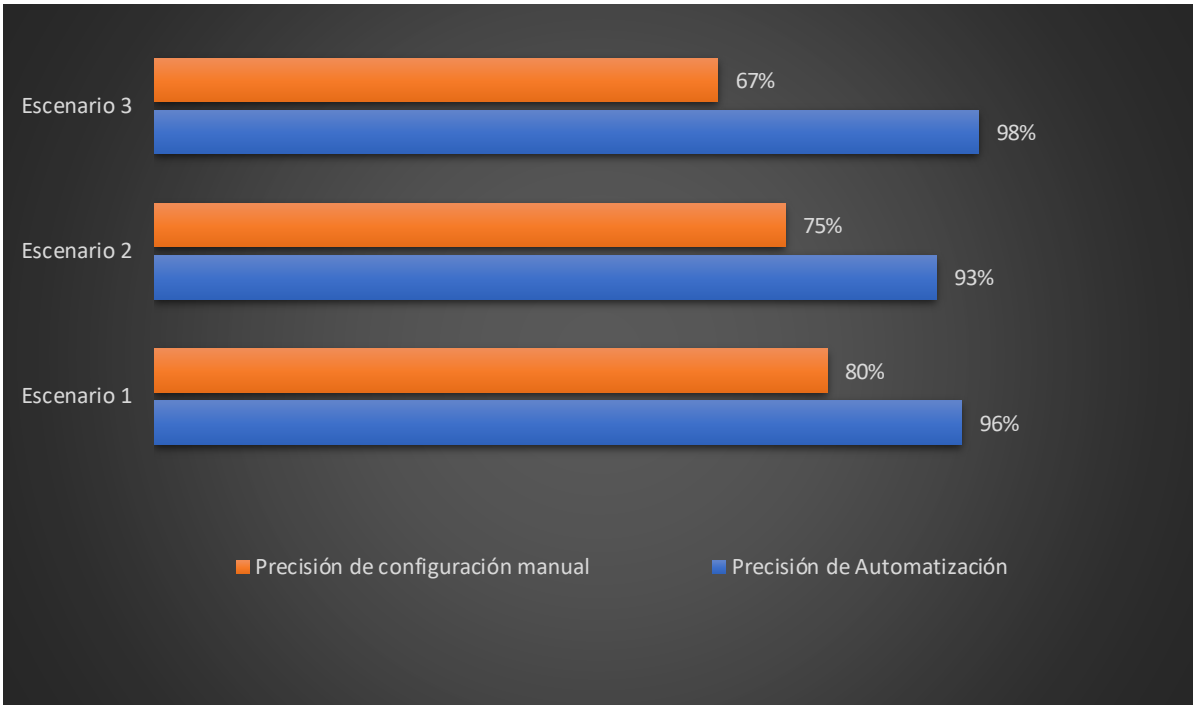


Figura 32: Comparación entre precisión de automatización y configuración manual en la asignación de ancho de banda

En la Figura 28 se presenta una comparación en tres escenarios diferentes entre la precisión de la configuración manual y la precisión de la automatización en la asignación del ancho de banda. La configuración manual está representada por el color naranja, mientras que la precisión de la automatización se representa con el color azul . La precisión de configuración manual indica el porcentaje de aciertos durante la configuración realizada por el administrador de la red, mientras que la precisión de automatización refleja el porcentaje de éxito alcanzado mediante la ejecución de scripts en la topología.

En el Escenario 1, se observa que la precisión de la configuración manual es del 80%, lo que significa que, aunque la configuración se realizó con éxito, existe un margen de error del 20% debido a fallas en los comandos que se fueron corrigiendo durante el desarrollo y las pruebas en comparación a la precisión de la automatización alcanzó un 96%, demostrando que este método es altamente confiable y casi no presenta fallas.

En el Escenario 2, la precisión de la configuración manual disminuye al 75%, lo que refleja un mayor desafío al configurar un número más elevado de dispositivos, este incremento en la complejidad reduce la optimización del tiempo durante la configuración. Por otro lado, la precisión de la automatización se mantiene alta, con un 93%, lo que indica que, a pesar del aumento en el número de dispositivos, la automatización sigue siendo muy efectiva.

Finalmente, en el Escenario 3, la precisión de la configuración manual registra su nivel más bajo, con un 67%, lo que sugiere que, a medida que aumenta la cantidad de dispositivos, también lo hace la probabilidad de errores. Sin embargo, la precisión de la automatización alcanza su punto más alto, con un 98%, lo que confirma que este método es extremadamente eficaz y consistente, incluso en entornos más complejos.

Evaluación asignación de roles Gráfica

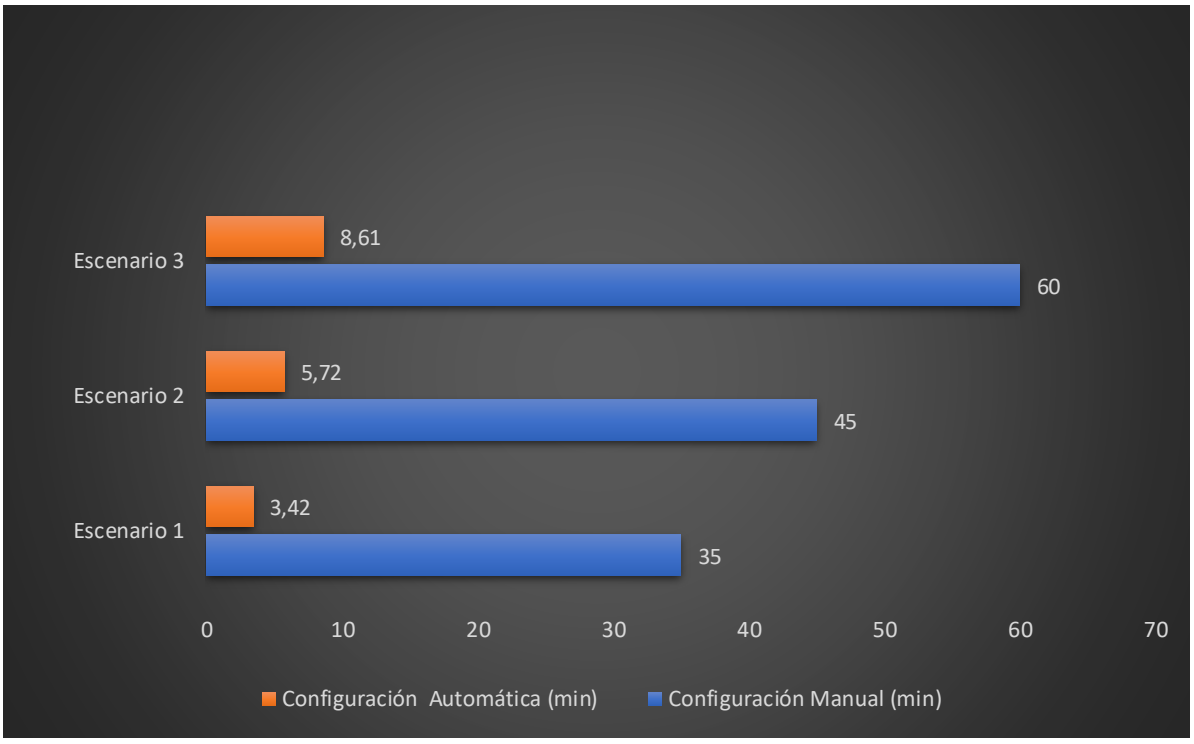


Figura 33: Comparación de Tiempo Configuración Manual y Automatizada en la asignación roles

En la Figura 29, compara los tiempos necesarios para realizar configuraciones manuales y automáticas en la asignación de roles en tres escenarios distintos. Se observa que la

configuración automática es significativamente más rápida que la manual en todos los casos, lo que destaca la eficiencia de los procesos automatizados en términos de tiempo.

En el Escenario 1, la configuración manual toma un promedio de 35 minutos, mientras que la configuración automática requiere solo 3.42 minutos. Se muestra que la automatización reduce significativamente el tiempo necesario para completar la tarea, ahorrando más del 90% del tiempo en este caso.

En el Escenario 2, la configuración manual toma 45 minutos, y la automática disminuye este tiempo a 5.72 minutos. Aunque el tiempo de configuración automática es mayor en comparación con el Escenario 1, sigue representando una reducción considerable respecto al método manual, con una mejora del orden del 87%.

En el Escenario 3, el tiempo requerido para la configuración manual es el más alto de los tres escenarios, alcanzando 60 minutos, mientras que la configuración automática toma 8.61 minutos. Este último valor es el mayor entre las configuraciones automáticas de los tres escenarios; la automatización aún implica una mejora significativa, ahorrando alrededor del 85% del tiempo.

Como se puede apreciar en los 3 escenarios, se evidencia que la configuración automática es mucho más eficiente en los tiempos de configuración; este ahorro de tiempo es importante para agilizar procesos y garantizar la efectividad.

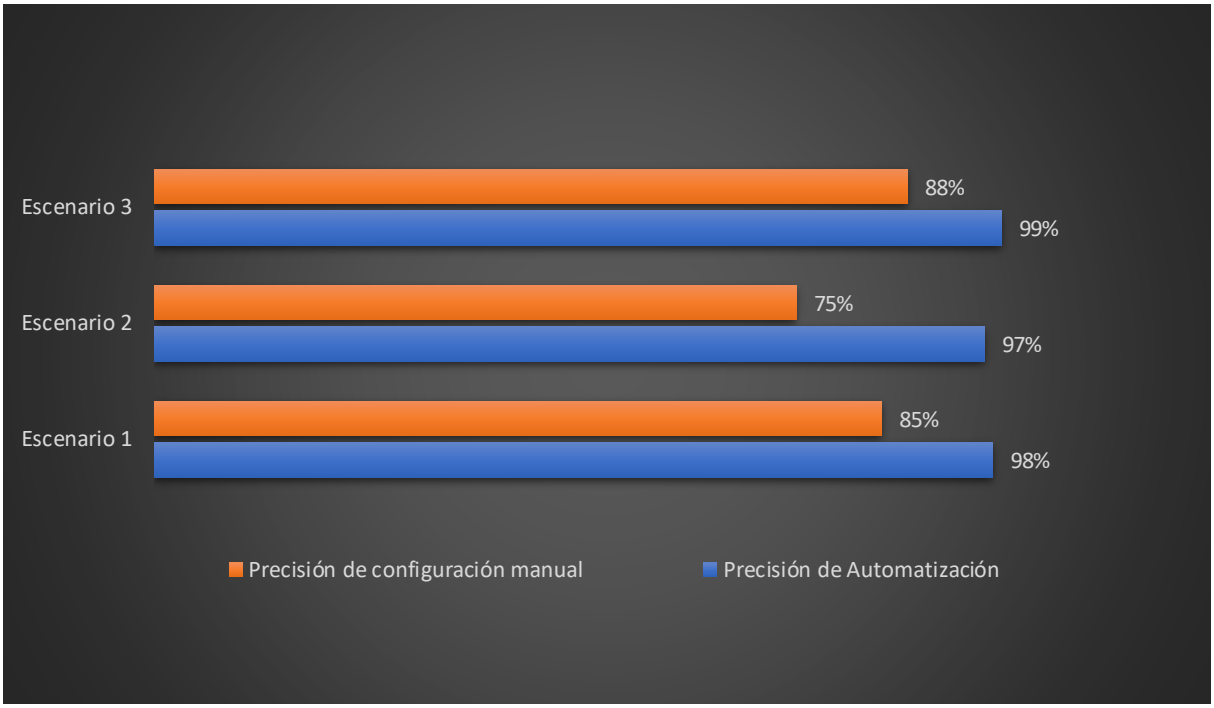


Figura 34: Comparación entre precisión de automatización y configuración manual en la asignación de roles

La Figura 30 compara la precisión entre la configuración manual y la automatización en la asignación de roles para tres escenarios diferentes. Los resultados están expresados en porcentajes, donde las barras naranjas representan la precisión de la configuración manual y las barras azules muestran la precisión de la configuración automatizada.

En el Escenario 1, la precisión de la configuración manual alcanza el 85%, mientras que la automatización logra una precisión del 98%. Esto evidencia una ventaja significativa de la automatización, que supera la configuración manual en un 13%, lo cual puede ser crucial en tareas que requieran alta exactitud.

En el Escenario 2, la diferencia es aún más pronunciada: la precisión manual disminuye al 75%, mientras que la automatización mantiene un nivel alto con un 97%. Este escenario resalta una clara superioridad de la automatización en términos de confiabilidad, con una mejora de 22 puntos porcentuales respecto al método manual.

En el Escenario 3, la configuración manual muestra una precisión del 88%, mientras que la automatización alcanza un 99%. La configuración manual mejora en este escenario en comparación con los anteriores; la automatización sigue destacándose con una diferencia positiva de 11 puntos porcentuales.

En términos generales, el análisis de los gráficos demuestra que la configuración automatizada no solo es más eficiente en tiempo, como se evidencia en la Figura 30, sino que también es mucho más precisa en cada uno de los escenarios evaluados.

4. CONCLUSIONES

- La revisión exhaustiva de tecnologías como SDN, Python y plataformas como GNS3 y Cisco IOS permitió identificar enfoques efectivos para la gestión de ancho de banda y la asignación de roles. Estas herramientas ofrecen una reducción significativa de la carga administrativa y optimización del desempeño de la red.
- La topología diseñada integra mecanismos avanzados como VLANs para segmentación y redundancia, garantizando flexibilidad, escalabilidad y un control eficiente del tráfico. Las pruebas iniciales en entornos emulados demostraron la viabilidad del diseño al facilitar la automatización.
- La implementación de scripts y herramientas en un entorno simulado permitió automatizar tareas críticas, como la distribución dinámica del ancho de banda y la asignación de roles. Esto no solo incrementó la precisión en la gestión de recursos, sino que también redujo tiempos de configuración y errores humanos.

5. RECOMENDACIONES

- Ampliar el conocimiento del equipo en tecnologías como SDN, scripting avanzado en Python y uso de GNS3 para simular escenarios complejos. Esto asegurará la sostenibilidad y evolución del sistema automatizado.
- Realizar pruebas adicionales que simulen condiciones de tráfico extremo para evaluar y optimizar la gestión del ancho de banda bajo diferentes escenarios.
- Incorporar rutinas automatizadas para la detección y solución de problemas en la red, como alertas en caso de fallos o congestionamiento.

- Registrar cada detalle del diseño y configuración, incluyendo scripts, políticas implementadas y resultados de pruebas. Esto facilitará futuras modificaciones y transferencias de conocimiento.
- Garantizar que la infraestructura diseñada pueda soportar el crecimiento en el número de dispositivos y usuarios sin comprometer el rendimiento de la red.

6. REFERENCIAS BIBLIOGRÁFICAS

[1] P. A. Quinteros, M. C. Zurita, N. C. Zambrano, y E. L. Manchay, «Automatización de los procesos industriales», J. Bus. Entrep. Stud., vol. 4, n.o 2, Art. n.o 2, jul. 2020, doi: 10.37956/jbes.v4i2.82.

[2] B. Perdomo, O. G. Martinez, y I. B. Barreto, «Competencias digitales en docentes universitarios: una revisión sistemática de la literatura», EDMETIC, vol. 9, n.o 2, Art. n.o 2, jul. 2020, doi: 10.21071/edmetic.v9i2.12796.

[3] W. Mengist, T. Soromessa, y G. Legese, «Ecosystem services research in mountainous regions: A systematic literature review on current knowledge and research gaps», Sci. Total Environ., vol. 702, p. 134581, feb. 2020, doi: 10.1016/j.scitotenv.2019.134581.

[4] R. Blanchet, S. Pérez, y H. Facchini, Estudio y Simulación de Redes Definidas por Software y Automatización de Red. 2021.

[5] S. Arzo, C. Serugunda, F. Granelli, R. Bassoli, M. Devetsikiotis, y F. Fitzek, «A Theoretical Discussion and Survey of Network Automation for IoT: Challenges and Opportunity», IEEE Internet Things J., vol. 8, pp. 12021-12045, ago. 2021, doi: 10.1109/JIOT.2021.3075901.

[6] D. Cordero Guzmán y G. Ramón Poma, «Modelo tecnológico e infraestructura informática de un campus virtual para el contexto universitario», Rev. Científica Tecnológica UPSE RCTU, vol. 8, n.o 2, pp. 48-58, dic. 2021, doi: 10.26423/rctu.v8i2.627.

[7] «IEEE Standard for Information Technology–Telecommunications and Information Exchange between Systems - Local and Metropolitan Area Networks–Specific Requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications», IEEE Std 80211-2020 Revis. IEEE Std 80211-2016, pp. 1-4379, feb. 2021, doi: 10.1109/IEEESTD.2021.9363693.

[8] M. Máté, C. Simon, y M. Maliosz, «Asynchronous Time-Aware Shaper for Time-Sensitive Networking», J. Netw. Syst. Manag., vol. 30, n.o 4, p. 76, sep. 2022, doi: 10.1007/s10922-022-09688-y.

[9] P. D. Bol, R. Lunardi, B. de França, y W. Cordeiro, «Modular switch deployment in programmable forwarding planes with switch (de)composer», en Proceedings of the SIGCOMM '21 Poster and Demo Sessions, en SIGCOMM '21. New York, NY, USA: Association for Computing Machinery, ago. 2021, pp. 30-32. doi: 10.1145/3472716.3472856.

[10] Z. Fan et al., «Monolithically Integrated 8 × 8 Transmitter-Router Based on Tunable V-Cavity Laser Array and Cyclic Arrayed Waveguide Grating Router», IEEE Photonics J., vol. 14, n.o 4, pp. 1-8, ago. 2022, doi: 10.1109/JPHOT.2022.3191946.

[11] W. Zhang, X. Chen, y J. Jiang, «A multi-objective optimization method of initial virtual machine fault-tolerant placement for star topological data centers of cloud systems», Tsinghua Sci. Technol., vol. 26, n.o 1, pp. 95-111, feb. 2021, doi: 10.26599/TST.2019.9010044.

[12] J. Singh y N. K. Walia, «A Comprehensive Review of Cloud Computing Virtual Machine Consolidation», IEEE Access, vol. 11, pp. 106190-106209, 2023, doi: 10.1109/ACCESS.2023.3314613.

[13] R. Ramirez, C.-Y. Huang, y S.-H. Liang, «5G Digital Twin: A Study of Enabling Technologies», Appl. Sci., vol. 12, n.o 15, Art. n.o 15, ene. 2022, doi: 10.3390/app12157794.

[14] M. A. A. Mamun, M. Li, y B. K. Pramanik, «Development of Delay-Tolerant Networking Protocols for Reliable Data Transmission in Space Networks: A Simulation-Based Approach», IEEE Access, vol. 12, pp. 178642-178658, 2024, doi: 10.1109/ACCESS.2024.3501676.

[15] «High availability of kernel-based virtual machine using nested virtualization - ScienceDirect». Accedido: 23 de enero de 2025. [En línea]. Disponible en: https://www.sciencedirect.com/science/article/pii/S266591742300048X?via%3Dihub

[16] G. Jain y Anubha, «Application of SNORT and Wireshark in Network Traffic Analysis», IOP Conf. Ser. Mater. Sci. Eng., vol. 1119, n.o 1, p. 012007, mar. 2021, doi: 10.1088/1757-899X/1119/1/012007.

[17] B. Chen, N. Mustakin, A. Hoang, S. Fuad, y D. Wong, «VSCuda: LLM based CUDA extension for Visual Studio Code», en Proceedings of the SC '23 Workshops of The International Conference on High Performance Computing, Network, Storage, and Analysis, en SC-W '23. New York, NY, USA: Association for Computing Machinery, nov. 2023, pp. 11-17. doi: 10.1145/3624062.3624064.

[18] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, y M. Ayyash, «Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications», IEEE Commun. Surv. Tutor., vol. 17, n.o 4, pp. 2347-2376, 2015, doi: 10.1109/COMST.2015.2444095.

[19] D. Chefrour, «One-Way Delay Measurement From Traditional Networks to SDN: A Survey», ACM Comput. Surv., vol. 54, n.o 7, p. 156:1-156:35, jul. 2021, doi: 10.1145/3466167.

[20] S. Mahmood, S. M. Mohsin, y A. Akber, «Network Security Issues of Data Link Layer: An Overview», mar. 2020, doi: 10.1109/iCoMET48670.2020.9073825.

[21] A. Giatsintov, K. Mamrosenko, y P. Bazhenov, «Architecture of the Graphics System for Embedded Real-Time Operating Systems», Tsinghua Sci. Technol., vol. 28, n.o 3, pp. 541-551, jun. 2023, doi: 10.26599/TST.2022.9010028.

[22] Raul Sales Giner, «Comparativa de distribuciones GNU/LINUX». [En línea]. Disponible en: https://openaccess.uoc.edu/bitstream/10609/73586/6/rsalesgTFG0118memoria.pdf

[23] S. Pérez, H. Facchini, A. Dantiacq, B. Roberti, y F. Hidalgo, Plataformas de Automatización de Red. 2022.

[24] B. Heim et al., «Quantum programming languages», Nat. Rev. Phys., vol. 2, n.o 12, pp. 709-722, dic. 2020, doi: 10.1038/s42254-020-00245-7.

[25] A. Salman, «Similarity Matching of XML Schema», Karaelmas Fen Ve Mühendis. Derg., vol. 10, n.o 1, Art. n.o 1, 2020, doi: 10.7212/zkufbd.v10i1.1516.

[26] A. Rawat, «A Review on Python Programming», Int. J. Res. Eng. Sci. Manag., vol. 3, n.o 12, Art. n.o 12, dic. 2020.

[27] H. U. Adoga y D. P. Pezaros, «Network Function Virtualization and Service Function Chaining Frameworks: A Comprehensive Review of Requirements,

Objectives, Implementations, and Open Research Challenges», Future Internet, vol. 14, n.o 2, Art. n.o 2, feb. 2022, doi: 10.3390/fi14020059.

[28] L. M. A. Fariño, J. F. A. Pizarro, M. J. Infante, A. R. T. Reyes, y B. M. M. Morán, «SDN Redes definidas por Software usando MiniNet», Rev. Científica Tecnológica UPSE, vol. 9, n.o 1, Art. n.o 1, jun. 2022, doi: 10.26423/rctu.v9i1.489.

[29] Silvana Maria Aparecida Viana Santos, Camila Sabino de Araujo, Camilo Eduardo do Nascimento, Elzo Brito dos Santos Filho, y Luciene Carneiro da S. O. Timoteo, «CICLO PDCA APLICADO À EDUCAÇÃO», Rev. Amor Mundi, vol. 4, n.o 4, ago. 2023, doi: https://doi.org/10.46550/amormundi.v4i4.211.

7. ANEXOS

Anexo 1 – Resultado de las métricas de evaluación de Latencia capturados por Wireshark

No.	Tiempo (ms)	Destino	Protocolo	Latencia (ms)
2	1.024.136.733	192.168.40.1	ICMP	14
4	2.025.475.773	192.168.40.1	ICMP	16
6	3.025.242.626	192.168.40.1	ICMP	19
9	505.639.723	192.168.40.1	ICMP	22
11	6.056.791.365	192.168.40.1	ICMP	24
13	8.128.366.981	192.168.40.1	ICMP	27
15	913.041.288	192.168.40.1	ICMP	30
20	12.137.145.031	192.168.40.1	ICMP	34
21	13.184.221.714	192.168.40.1	ICMP	36
25	15.187.104.672	192.168.40.1	ICMP	39
26	16.192.654.112	192.168.40.1	ICMP	40
31	19.241.726.662	192.168.40.1	ICMP	44
33	21.312.630.859	192.168.40.1	ICMP	48
41	25.321.472.259	192.168.40.1	ICMP	52
43	27.392.288.948	192.168.40.1	ICMP	55
50	32.512.172.527	192.168.40.1	ICMP	60
61	40.682.541.255	192.168.40.1	ICMP	65
66	4.477.795.411	192.168.40.1	ICMP	70
67	45.824.444.772	192.168.40.1	ICMP	71
69	47.872.398.718	192.168.40.1	ICMP	77
73	49.919.951.265	192.168.40.1	ICMP	79
74	50.944.156.295	192.168.40.1	ICMP	82
75	51.968.564.133	192.168.40.1	ICMP	85
80	54.993.875.232	192.168.40.1	ICMP	90