

ARTÍCULO DE INVESTIGACIÓN

Aplicación web para la gestión automatizada de routers virtualizados: Un enfoque basado en Python

Web application for automated management of virtualized routers: a python-based approach

DOI: <https://doi.org/10.61154/metanoia.v12i1.4214>

Jean Pierre Salgado Benalcázar¹

Jorge Luis Garzón Izquierdo²

Rodrigo Fernando Morocho Román³

Joofre Antonio Honores Tapia⁴

¹**E-mail:** jsalgado2@utmachala.edu.ec **Afiliación:** Universidad Técnica De Machala. El Oro, Machala, Ecuador. **ORCID:** <https://orcid.org/0009-0008-1373-0797>

² **E-mail:** jgarzon3@utmachala.edu.ec **Afiliación:** Universidad Técnica De Machala. El Oro, Machala, Ecuador. **ORCID:** <https://orcid.org/0009-0008-2208-0255>

³ **E-mail:** rmorocho@utmachala.edu.ec **Afiliación:** Universidad Técnica De Machala. El Oro, Machala, Ecuador. **ORCID:** <https://orcid.org/0000-0003-0194-5033>

⁴ **E-mail:** jhonores@utmachala.edu.ec **Afiliación:** Universidad Técnica De Machala. El Oro, Machala, Ecuador. **ORCID:** <https://orcid.org/0000-0001-8612-3025>

Recibido: 01/09/2025

Revisado: 11/09/2025

Aprobado: 01/10/2025

Publicado: 01/01/2026

RESUMEN

La configuración manual de routers solía requerir bastante conocimiento técnico y además tomaba mucho tiempo para su correcta ejecución, lo que representa una limitación en entornos educativos y de simulación. Pensando en esto se desarrolló RouterApp, una aplicación web que busca automatizar estas tareas y hacer más sencillo el proceso de configuración por medio de una interfaz gráfica fácil de usar. Para organizar el proyecto se siguió el ciclo de mejora continua PDCA, permitiendo trabajar de manera iterativa y ajustando el avance según las necesidades detectadas y la retroalimentación recibida en cada etapa. La herramienta fue puesta a prueba en routers Cisco simulados en GNS3 y ejecutados en entornos virtualizados. Para el backend se utilizó Python junto con Flask y Netmiko, facilitando la automatización de comandos mediante conexiones SSH. Para la parte del frontend, se trabajó con tecnologías como JavaScript y Tailwind CSS, integrando también la visualización de topología de red. Con RouterApp es posible automatizar tareas claves como asignar direcciones IP, configurar interfaces, activar protocolos de enrutamiento, gestionar el ancho de banda y llevar un registro de los comandos utilizados. Cada función fue verificada comparando los resultados obtenidos automáticamente con los generados mediante configuraciones manuales, confirmando precisión, reducción de errores y una mejora significativa en los tiempos de configuración. La arquitectura modular aplicada asegura que el sistema sea escalable para futuras mejoras según las necesidades y experiencia de uso.

Descriptor: tecnología de la comunicación, aplicación informática, red informática.

ABSTRACT

Manual router configuration has historically required substantial technical expertise and has been time-consuming to execute correctly, which poses challenges in educational and simulation environments. Acknowledging this limitation, RouterApp was developed as a web application designed to automate these tasks and streamline the configuration process through a user-friendly graphical interface. The project followed the PDCA continuous improvement cycle, enabling iterative development and adjustments based on identified needs and feedback at each stage. RouterApp was tested on Cisco routers simulated in GNS3 and operated in virtualized environments. For the backend, Python, along with Flask and Netmiko, facilitated command automation over SSH connections. The frontend utilized technologies such as

JavaScript and Tailwind CSS, which also included network topology visualization. With RouterApp, users can automate essential tasks such as assigning IP addresses, configuring interfaces, activating routing protocols, managing bandwidth, and tracking utilized commands. Each feature has been validated by comparing the outcomes of automated processes with those obtained through manual configurations, demonstrating increased accuracy, reduced errors, and a notable enhancement in configuration speed. The implemented modular architecture ensures the system's scalability for future improvements based on evolving needs and user feedback.

Descriptors: communication technology, computer application, computer network.

INTRODUCCIÓN

La administración de infraestructura de red es un pilar fundamental para garantizar entornos digitales seguros y escalables, especialmente en servicios críticos que demandan alta disponibilidad (Hayudini, 2021). En el ámbito de redes, la automatización de configuraciones se ha convertido en una alternativa importante porque ayuda a reducir los errores humanos, acelera los procesos y hace más consistentes las operaciones (Brighenti et al., 2023). Además, el uso de entornos virtualizados y de plataformas automáticas ha demostrado ser de gran ayuda en la enseñanza, ya que permite desplegar configuraciones de manera ágil, aporta flexibilidad y fomenta la mejora continua en la gestión de redes (Santayadiputra et al., 2021).

El uso de simuladores también marcó un avance significativo en la educación de redes. Arias Figueroa et al (2018) señala que su aplicación en la enseñanza del direccionamiento IP facilita la comprensión práctica de los conceptos estudiantiles. Sin embargo, la configuración manual incrementa la posibilidad de cometer errores y pueden comprometer a la estabilidad y seguridad de la red (Zapata Diaz, 2022). A esto podemos sumar la necesidad de configurar los dispositivos por separados, lo que limita la escalabilidad demandando tiempo y recursos (Cañas, 2025). Ante esto la automatización se plantea como solución práctica para aplicar configuraciones rápidas y confiables (Romero Lovera, 2022).

Las herramientas como GNS3 o Cisco Packet Tracer han sido clave en la formación, ya que permiten poder practicar sin tener la necesidad de contar con equipos físicos (Tandazo Tandazo, 2022). Sin embargo, presentan limitaciones cuando se quiere trabajar con sistemas con automatización más avanzados o en escenarios de producción, para superar estas barreras lenguajes como Python y ocupando librerías como Netmiko dan la posibilidad de crear scripts que automatizan configuraciones de red, lo que mejora la eficiencia y reduce en gran medida

los errores (López & Ramírez, 2024; Sandoval Rivera, 2023). El inconveniente de estas soluciones surge porque se requiere de conocimientos técnicos avanzados, lo que minimiza su uso en ambientes educativos (Ramírez Paucar & Aguilar Mora, 2011).

Con la intención de ofrecer una alternativa mas accesible para todos, se desarrolló RouterApp, una aplicación web pensada para la automatización de configuraciones de routers virtualizados. La idea principal fue basarse en simplificar los procesos y tratar de reducir la complejidad técnica, para que cualquier usuario pueda usarla sin experiencia previa.

Tomando en cuenta esto, RouterApp se presentó como una herramienta innovadora que combina automatización, replicación para el escenario educativo y apoyar al aprendizaje practico en redes. Su arquitectura desarrollada en Python e integrada en un entorno virtual de GNS3, permitió obtener configuraciones mas precisas y reproducibles, minimizando errores humanos y reduciendo los tiempos de ejecución en cada comando. Además, la comparación frente al método manual de línea de comando (CLI), respaldo la confiabilidad y el uso del ciclo PDCA aseguro el proceso de mejora continua.

Este artículo describe el desarrollo de RouterApp y su aporte como herramienta para la gestión automatizada en entornos educativos y de simulación. Se detallan los fundamentos teóricos que guiaron su diseño, la metodología de validación y los resultados obtenidos comparando con configuraciones manuales mediante la línea de comando (CLI). Finalmente, se analiza su valor en la formación técnica para usuarios y su potencial de expansión hacia escenarios más complejos, tomando en cuenta tendencias actuales de automatización y digitalización de la gestión de redes.

METODOLOGÍA

Este estudio se enmarca dentro de una investigación aplicada, donde se combinó un enfoque experimental para evaluar el desempeño de la herramienta y un enfoque descriptivo para analizar y dar a conocer los resultados. La investigación se orientó a dos objetivos principales:

- El desarrollo de la herramienta tecnológica RouterApp.
- La comparación sistemática de eficiencia frente al método manual de configuración (CLI).

Para sustentar el desarrollo, se realizó un análisis documental muy detallado de artículos científicos, tesis y manuales técnicos sobre la automatización de redes, virtualización con GNS3, el uso de bibliotecas de Python y Netmiko.

La recolección de los datos se llevó a cabo mediante observación directa en un entorno de simulación controlado, donde se ejecutaron y midieron los dos escenarios:

- Configuración manual mediante CLI.
- Configuración automatizada con RouterApp.

Las variables críticas medidas fueron:

- Tiempo de ejecución: Cronometrado para cada tarea completada.
- Tasa de error: Registrada mediante el análisis de logs de comando y la verificación del estado de la configuración.

Ciclo PDCA

Con el propósito de garantizar la calidad del desarrollo y promover la mejora continua, se adoptó el ciclo PDCA (Planificar, Hacer, Verificar, Actuar) como metodología, donde se aplicaron cada una de sus fases de la siguiente manera:

Plan (Planificar): En esta fase se identificó como problema central la complejidad y la recurrencia de errores en la configuración manual en entorno de aprendizaje. En la Tabla 1 se definieron los requisitos funcionales de la herramienta, los criterios de validación (tiempo y errores) y se diseñó el entorno de simulación base.

Tabla 1 Actividades realizadas en la fase de planificación.

1	Determinar las configuraciones básicas a automatizar.
2	Diseñar una arquitectura modular que permitiera escalabilidad y mantenimiento.
3	Establecer criterios de comparación para los resultados.
4	También se definió el entorno de simulación en GNS3.

Fuente: Los autores

Do (Hacer): Se desarrollo la primera versión funcional (prototipo) de RouterApp, implementando la capacidad de establecer conexiones mediante SSH con routers Cisco virtualizados en GNS3 y teniendo la capacidad de ejecutar comandos básicos de configuración.

Check (Verificar): Se evaluó el prototipo en entornos simulados y se compararon los resultados obtenidos con RouterApp frente a los logrados manualmente mediante CLI. El

análisis incluyo tiempos de ejecución, equivalencia técnica y ocurrencia de errores. La Tabla 2 detalla las pruebas funcionales ejecutadas, las cuales abarcan operaciones clave como:

Tabla 2 *Pruebas funcionales ejecutadas para validar el comportamiento de RouterApp.*

1	Asignación de direcciones IP a interfaces específicas.
2	Configuración de interfaces (activación, descripción, velocidad, etc.).
3	Activación del protocolo OSPF, especificando la red, área y tipo de router.
4	Pruebas de conectividad (ping) desde los routers configurados hacia otros nodos.
5	Control de ancho de banda, mediante la configuración de políticas básicas de tráfico.
6	Validación del registro de comandos, para asegurar trazabilidad de acciones.

Fuente: Los autores

Se documentaron los tiempos de ejecución, la consistencia de los resultados, y se verificó que la salida de comandos desde RouterApp fuera equivalente a la emitida por el CLI.

Act (Actuar): Con base en los resultados, se aplicaron ajustes orientados a mejorar la estabilidad y la usabilidad de la herramienta. Entre las mejoras realizadas se incluyeron la optimización en la ejecución de comandos, validaciones para evitar parámetros inválidos y la incorporación de funciones adicionales que enriquecen la experiencia de uso. Estos elementos se resumen en la:

Tabla 3 *Módulos adicionales implementados en RouterApp.*

Visor de la topología	Muestra gráficamente los routers configurados y sus conexiones.
Exportación de logs	Permite descargar en formato .txt o .csv los comandos ejecutados.
Un sistema de alertas	Notifica al usuario sobre el éxito o falla de cada operación.

Fuente: Los autores (2025).

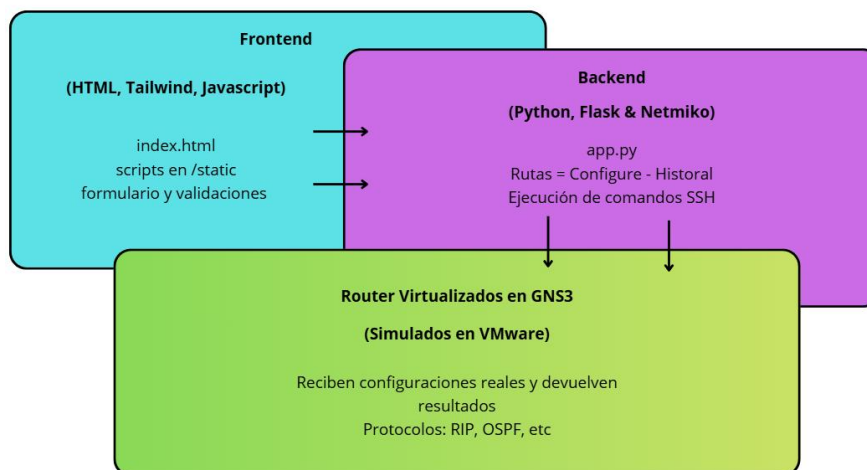
Estas mejoras fueron aplicadas de forma iterativa hasta alcanzar una versión estable que cumpliera con los objetivos de automatización, usabilidad y fidelidad técnica frente a las configuraciones tradicionales.

Desarrollo de la aplicación

El diseño de RouterApp se organizó en componentes que se comunican entre sí para lograr la automatización de configuraciones en routers virtualizados, arquitectura visualizable en la

figura 1.

Figura 1 *Arquitectura de RouterApp*



Fuente: Los autores

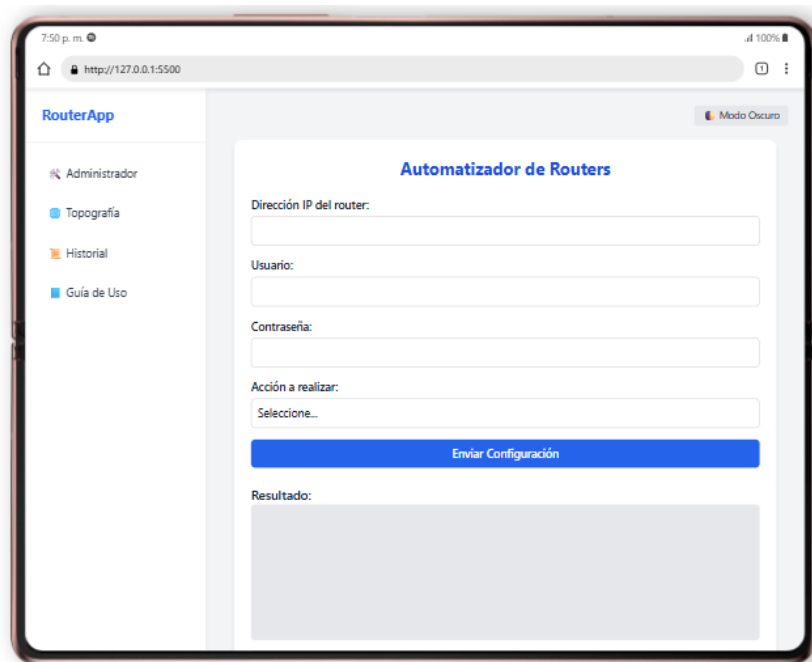
Backend

Desarrollado en Python utilizando el framework Flask y la biblioteca Netmiko para poder tener la conexión SSH con los routers Cisco virtualizados en GNS3. Su función es procesar las solicitudes del usuario, generar y ejecutar los comandos de configuración en los routers y gestionar la trazabilidad mediante un historial de logs (acciones) y así mismo funciones para poder descargar estos resultados y consultar el estado de los dispositivos.

Frontend

El frontend proporciona al usuario una interfaz web intuitiva que se organiza en cuatro opciones claves: Administrador, Topografía, Historial y Guía de uso.

- El módulo Administrador contiene un formulario que recoge las credenciales del router y la acción a ejecutar.
- Las opciones de configuración incluyen hostname, direccionamiento IP, DHCP, OSPF, RIP, rutas estáticas, control de ancho de banda y pruebas de conectividad.
- Se integraron scripts JavaScript externos en la carpeta static (ej. sendConfigure.js, cargarHistorial.js, cargarInterfaces.js) para modularizar las funciones: envío de configuraciones, renderizado de resultados y carga dinámica de opciones.
- Tailwind CSS se utilizó para garantizar un diseño responsivo con soporte de modo oscuro, y se añadió la librería vis-network para la visualización de la topología.

Figura 2 *Diseño Final de RouterApp.*

Fuente: Los autores

Módulo de visualización topología

Se implementó un visualizador en el apartado de Topografía, que se enlaza directo con el proyecto que se encuentre activo en GNS3. Esto ayuda a que el usuario observe en tiempo real la topología configurada y los cambios que se realizan usando RouterApp.

Gestión de historial y auditoría

Se agregaron funciones específicas para poder almacenar todas las configuraciones empleadas en un archivo local. En la interfaz gráfica, los usuarios pueden descargar, visualizar y limpiar este historial.

Validaciones y mejoras en usabilidad

La lógica del fronted no permite que se envíen los campos del formulario incompleto, calcula automáticamente los valores de wildcards para OSPF, también nos brinda menús desplegables de interfaces detectadas y permite elegir entre asignaciones manuales/automáticas de ancho de banda. Estas funcionalidades no solo ayudan a la comprensión de los procesos de las configuraciones, sino que nos permite reducir errores.

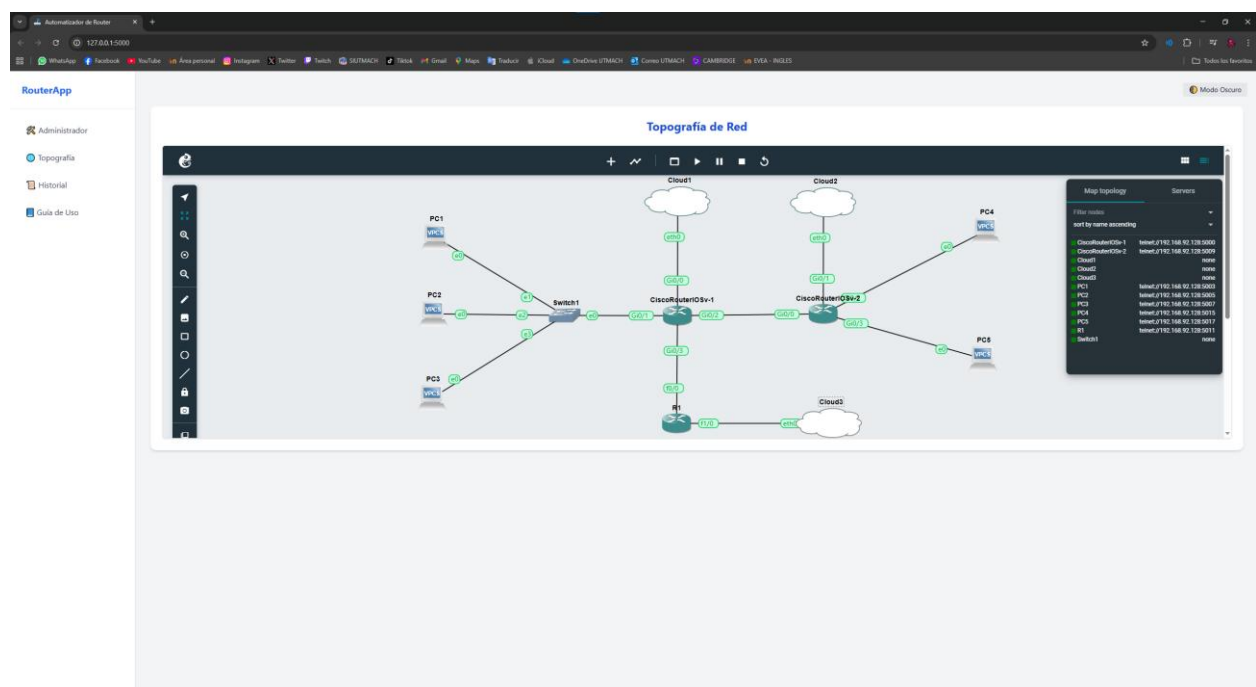
En conjunto, la aplicación quedó complementada bajo un esquema MVC simplificado, en donde Flask actúa como controlador, los scripts en static como capa de lógica de interacción, y index.html como vista.

RESULTADOS

Para la validación funcional de RouterApp, se plantea un escenario de red diseñado en GNS3 que se visualiza en la Figura 3, representativo de un laboratorio de configuración de routers. La topología estuvo conformada por:

- Tres estaciones de trabajo (PC1, PC2, PC3) conectadas a un switch de capa 2 (Switch1), representando clientes finales.
- Dos routers virtualizados (CiscoRouterIOSv-1, CiscoRouterIOSv-2) como elementos principales de enrutamiento, con múltiples interfaces interconectadas para el intercambio de tráfico.
- Tres segmentos de nube (Cloud1, Cloud2, Cloud 3) simulando conexiones externas y permitiendo pruebas de conectividad.

Figura 3 Topología de red empleada para la validación funcional de RouterApp.



Fuente: Los autores

Sobre esta estructura, se realizaron diez funciones de red consideradas básicas para entornos educativos y de operación inicial: asignación de hostname, pruebas de conectividad (ping), configuración de contraseña, protocolos de enrutamiento (OSPF, RIPv2), asignación de

dirección IP, activación/desactivación de interfaces, ajuste de ancho de banda, rutas estáticas y habilitación DHCP.

Cada función fue ejecutada en dos fases:

- Configuración manual mediante CLI tradicional.
- Configuración automática usando la aplicación RouterApp.

De esta forma se midió el tiempo requerido para configurar funciones comunes de red, comparado con el método manual por línea de comandos (CLI) frente a la aplicación automatizada RouterApp. La técnica empleada fue la observación directa cronometrada, complementada con registros sistemáticos de errores y las repeticiones que se ocupa en cada tarea.

Los resultados obtenidos se representan en la Tabla 4, donde se evidencia una reducción de tiempo y errores para diez funciones clave, con un promedio de 84,77 % usando RouterApp respecto al método manual por CLI. Destacando tareas como configuración de DHCP y configuración de contraseñas, que lograron reducciones del 95,49 % y 91,96 % respectivamente.

Tabla 4 Comparativa de tiempos de configuración y errores entre CLI y RouterApp

Función	Tiempo CLI (s)	Tiempo RouterApp (s)	Reducción (%)	Errores CLI	Errores RouterApp
Hostname	7,36	1,68	77,17%	1	0
Password	13,06	1,05	91,96%	1	0
OSPF	25,07	2,05	91,82%	2	0
RIPv1/2	16,46	2,1	87,24%	1	0
Ping	25,2	10,05	60,12%	1	0
Asignar IP Interfaz	25,58	5,5	78,5%	2	0
Habilitar/Deshabilitar interfaz	11,11	2,4	78,4%	1	0
Ancho de Banda	16,09	1,45	90,99%	2	0
DHCP	33,49	1,51	95,49%	2	0
Configurar Rutas Estáticas	19,01	2,09	89,01%	1	0

Fuente: Los autores

Se observo que, además de optimizar los tiempos de ejecución, el uso de RouterApp elimino por completo la aparición de errores, lo cual contrasta con la configuración manual por CLI donde se registraron fallos frecuentes como:

- Errores de sintaxis (al cambiar el nombre del host).
- Omisión de comandos (no shutdown, enable).
- Direcciones IP incompletas (falta de mascara de subred).
- Uso inadecuado de parámetros (por ejemplo, valores de ancho de banda).
- Configuraciones incompletas (como ausencia de default-router en DHCP).

Estas carencias no solo afectaron la validez de los resultados, sino que también incrementaron el tiempo requerido por la necesidad de correcciones o repeticiones.

Con esto podemos decir que los resultados demuestran el aporte innovador de RouterApp, al constituirse en una herramienta que no solo reduce significativamente los tiempos de configuración, sino que eliminan los errores humanos frecuentes en la enseñanza y practicas de redes. Esto a su vez refuerza su potencial como recurso educativo y apoyo en entorno de laboratorio y simulación.

DISCUSIÓN

Los resultados evidencian que RouterApp logró una reducción promedio del 84,77% en los tiempos de configuración respecto al método manual por CLI, alcanzando valores superiores al 90% en funciones críticas como la configuración de contraseñas, OSPF y DHCP. Este hallazgo coincide con lo señalado por López & Ramírez (2024), quienes destacan que la automatización mediante Python y Netmiko disminuye significativamente los tiempos de ejecución y la complejidad en la gestión de dispositivos de red. Durante las pruebas de configuración manual se repitieron varios errores comunes como la escritura de comandos con sintaxis incorrectas, olvidar pasos necesarios y no completar parámetros de configuración. Esto le da la razón a Zapata Díaz (2022), quien señalo que depender únicamente de la línea de comandos (CLI) aumenta la probabilidad de errores humanos y afecta a la estabilidad de la red.

En contraste RouterApp consiguió eliminar por completo estos errores, gracias a su sistema de validación de datos antes de aplicar las configuraciones en los dispositivos. Este resultado mostró que la aplicación no solo hacia mas eficiente el trabajo, sino que aportaba mas fiabilidad

a las configuraciones, reduciendo la necesidad de correcciones o repeticiones que retrasan las prácticas en escenarios de simulación y educativos.

Uno de los puntos a destacar fue justamente la reducción de errores al utilizar RouterApp. En el ámbito educativo resulta clave este aspecto, porque los estudiantes que recién inician al mundo de redes suelen cometer fallos frecuentes, ya sea al escribir comandos muy largos o al recordar los parámetros específicos para cada configuración. Tal como señalan Arias Figueroa et al. (2018), el uso de simuladores en la enseñanza de redes ayuda a reducir equivocaciones y refuerza la comprensión de los conceptos. Nuestros resultados van en la misma línea: RouterApp no reemplaza el aprendizaje con la CLI, pero sí lo hace más accesible al permitir que los estudiantes se enfoquen en la lógica de la configuración en lugar de memorizar cada instrucción. Esto coincide también con lo planteado por Santyadiputra et al. (2021), quienes destacan que las plataformas virtualizadas y autoajustables facilitan la práctica en redes, al mismo tiempo que reducen errores y apoyan a la competencia técnicas.

En el plano técnico, RouterApp logró integrarse de forma estable gracias a la combinación de Python con Flask y la librería Netmiko, lo que permitió establecer conexiones SSH confiables y automatizar comandos de red. Este desarrollo, al interactuar directamente con GNS3, ofrece un entorno mucho más cercano a la realidad en comparación con simuladores que solo imitan el comportamiento de los dispositivos. Según López y Ramírez (2024), el uso de Python y librerías de automatización en redes no solo agiliza la gestión de configuraciones, sino que también fomenta buenas prácticas al minimizar los errores humanos. De esta manera, los estudiantes y administradores de red no solo disponen de una herramienta que ahorra tiempo, sino también de un recurso que replica escenarios de producción, aportando un valor añadido en la formación práctica.

Cabe destacar que la validación de RouterApp se realizó en un entorno virtualizado con GNS3, que, si bien replica con alta confiabilidad el comportamiento de equipos físicos, representa una primera aproximación al problema. Futuras investigaciones deberían incluir pruebas con dispositivos reales, tal como sugiera Tandazo Tandazo (2022) y Cañas (2025) en sus estudios sobre las limitaciones de los entornos de simulación. También es recomendable ampliar la muestra de usuarios, donde se pueda incluir tanto estudiantes principiantes como profesionales en formación, permitiendo evaluar la usabilidad de la aplicación en distintos niveles de conocimientos.

Proyecciones futuras

Los resultados obtenidos nos permiten plantear algunas mejoras para RouterApp. Una de las principales es la adaptación a entornos con hardware físico, lo cual permitiría validar su funcionamiento en infraestructura real y no solo de simulación. Además, se considera relevante integrar nuevos protocolos o funcionalidades específicas, como configuraciones de seguridad o VPN.

Se recomienda a futuro el desarrollo de una versión desde el enfoque pedagógico. Este enfoque clave en RouterApp podría tener:

- Retroalimentación visual inmediata ante errores frecuentes
- Recursos de aprendizaje integrados directamente en la interfaz.

Finalmente se recomienda ampliamente evaluar el desempeño de la aplicación con grupos de usuarios de distintos niveles de experiencias, estudiantes principiantes hasta administradores de red en formación.

CONCLUSIONES

El desarrollo de RouterApp demostró que es posible optimizar las configuraciones de routers virtualizados, reduciendo de manera significativa los tiempos de ejecución y evitando errores que suelen surgir cuando se trabaja manualmente con el CLI. Los resultados que la herramienta nos dio fueron muy positivos ya que dieron a los usuarios una experiencia intuitiva, rápida y sencilla de adaptarse, ya sea en escenarios educativos o de formación de redes.

Es importante aclarar que la automatización que propone RouterApp no pretende remplazar el aprendizaje tradicional, sino servir como apoyo que guía al usuario validando sus acciones y previniendo fallos en el proceso de configuración de routers. Con esto permite la mejor comprensión de conceptos sin generar confusiones en el proceso de aprendizaje, resultando valioso cuando el tiempo es muy limitado y los errores pueden desmotivar o retrasar a un estudiante en particular.

La arquitectura flexible y el uso tecnologías como lo son Python, Flask y Netmiko, permitió integrar con éxito a RouterApp en entornos simulados mediante GNS3, ofreciendo una experiencia muy cercana a trabajo con equipos físicos. Con esto también se puede plantear a futuro mejoras relacionadas a la implementación con hardware real y ponerla a prueba con diversos grupos de usuarios de diferentes niveles de formación. Esto permitirá mejorar la

interfaz, pulir las funciones a distintos niveles de experiencia y consolidar tanto en escenarios educativos como profesionales, manteniendo una visión de mejora continua y estando a la vanguardia de las necesidades actuales de automatización de redes.

REFERENCIAS BIBLIOGRÁFICAS

- Hayudini, M. A. (2021). Network Infrastructure Management: Its importance to the organization. *Natural Sciences Engineering and Technology Journal*, 2(1), 62–67. <https://doi.org/10.37275/nasetjournal.v2i1.15>
- Bringhenti, D., Marchetto, G., Sisto, R., & Valenza, F. (2023). Automation for network security configuration: State of the art and research trends. *ACM Computing Surveys*, 56(3), 1–37. <https://doi.org/10.1145/3616401>
- Santiyadiputra, G. S., Listartha, I. M. E., & Saskara, G. A. J. (2021). The effectiveness of Automatic Network Administration (ANA) in network automation simulation at Universitas Pendidikan Ganesha. *Journal of Physics: Conference Series*, 1810(1), 012028. <https://doi.org/10.1088/1742-6596/1810/1/012028>
- Cañas, D. E. (2025). Eficiencia de tareas concurrentes utilizando lenguaje de programación Python en la automatización de redes. *Revista de Redes y Comunicaciones*, 10(1), 1–10. <https://doi.org/10.5377/ryr.v1i61.20636>
- Arias Figueroa, D., Díaz, J., & Gimson, L. (2018). Estudio de la influencia de métodos de simulación en la enseñanza del tema Direccionamiento IP. *TEyET*, (21), e08. <https://doi.org/10.24215/18509959.21.e08>
- Romero Lovera, J. A. (2022). Influencia del sistema automatizado de configuraciones sobre el proceso de gestión del cambio del área IP de una operadora móvil en Perú frente a la pandemia de COVID-19. *Industrial Data*, 25(1), 181–203. <https://doi.org/10.15381/idata.v25i1.22036>
- Zapata Díaz, J. A. (2022). Automatización de requerimientos y configuraciones en equipos de red del área de conectividad y seguridad perimetral de la empresa ARUS S.A. para mejora en eficiencia de tiempo mediante programación de scripts [Informe de práctica profesional, Universidad de Antioquia]. Repositorio Institucional Universidad de Antioquia. <http://bibliotecadigital.udea.edu.co/server/api/core/bitstreams/e93fe795-e9e6-4912-8d4c-65652df6d4a9/content>
- Tandazo Tandazo, J. C. (2022). Automatización de redes utilizadas para el Internet de las Cosas:

- Automatización de redes con Netmiko [Tesis de pregrado, Escuela Politécnica Nacional]. Repositorio Institucional EPN.
<https://bibdigital.epn.edu.ec/handle/15000/23212>
- López, J. A., & Ramírez, M. A. (2024). Configuración de una red LAN utilizando scripting en Python. Revista Elite, 6(1), 45–52.
<https://revistaelite.itsqmet.edu.ec/index.php/elite/article/view/87>
- Gracia Andrade, Y. P. (2025). Propuesta para la aplicación de metodología PDCA en el proceso de mejora continua en la producción de una empresa de alimentos [Tesis de pregrado, Universidad Politécnica Salesiana]. Repositorio Institucional UPS.
<https://dspace.ups.edu.ec/handle/123456789/29887>
- Sandoval Rivera, J. A. (2023). Automatización de redes utilizadas para el Internet de las Cosas: Análisis de metodologías para la automatización de redes para IoT [Tesis de pregrado, Escuela Politécnica Nacional]. Repositorio Institucional EPN.
<https://bibdigital.epn.edu.ec/handle/15000/24259>
- Ramírez Paucar, V. L., & Aguilar Mora, C. D. (2011). Compilación y automatización de un sistema de gestión de redes NOC con herramientas de código abierto [Tesis de pregrado, Universidad Técnica Particular de Loja]. Repositorio Institucional UTPL.
<https://dspace.utpl.edu.ec/handle/123456789/12996>
- Zamora Sánchez, R. A., & Panchana Cedeño, A. M. (2024). Gestión de relaciones con los clientes como estrategia de negocios del sector retail en la provincia de Manabí-Ecuador. Revista Social Fronteriza, 4(1), e41138.
[https://doi.org/10.59814/resofro.2024.4\(1\)138](https://doi.org/10.59814/resofro.2024.4(1)138)

©2026 por los autores. Este artículo es de acceso abierto y distribuido según los términos y condiciones de la licencia Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional (CC BY-NC-SA 4.0) (<https://creativecommons.org/licenses/by-nc-sa/4.0/>)