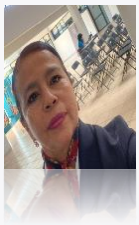
© FEGLININ
| Año 10, No 39, octubre - diciembre 2026 |
https://doi.org/10.70417/

ISSN 2594-2298

Prefijo DOI: 10.70417

NETEXPERT-XAI: DESARROLLO DE UN SISTEMA EXPERTO BASADO EN REGLAS PARA EL DIAGNÓSTICO EXPLICABLE DE FALLAS EN REDES LAN



NETEXPERT-XAI: A RULE-BASED EXPERT SYSTEM FOR EXPLAINABLE DIAGNOSIS OF LAN NETWORK FAILURES

Maestra. Ariopajita Rojo Lopez[1]
Mtra. Rocío Guadalupe Zozaya Salas[2]
Lic. Guadalupe Flores Díaz[3]
Lic. Minnie Ruth Fragoso Cobos[4]

RESUMEN

Los Sistemas Expertos constituyen una alternativa eficaz para representar conocimiento especializado y apoyar la toma de decisiones mediante reglas de producción (Russell & Norvig, 2021). En este trabajo se presenta NetExpert-XAI, un sistema experto para el diagnóstico explicable de fallas en redes de área local (LAN). La metodología se basó en los principios de la Ingeniería del Conocimiento, abarcando la adquisición y representación del conocimiento, el diseño de una base de conocimientos y la implementación de un motor de inferencia con encadenamiento hacia adelante utilizando Python y la biblioteca Experta (Experta Developers, s. f.). El sistema integra una interfaz de usuario, base de hechos, motor de inferencia y módulo de explicación, permitiendo generar diagnósticos justificados y recomendaciones técnicas conforme a los principios de la Inteligencia Artificial Explicable (Adadi & Berrada, 2018). Los resultados evidencian que el prototipo reproduce adecuadamente el razonamiento de especialistas en administración de redes y constituye una herramienta transparente y escalable para apoyar el diagnóstico.

ABSTRACT

Expert systems offer an effective alternative for representing specialized knowledge and supporting decision-making through production rules (Russell & Norvig, 2021). This paper presents NetExpert-XAI, an expert system for the explainable diagnosis of faults in local area networks (LANs). The methodology was based on knowledge engineering principles, encompassing knowledge acquisition and representation, the design of a knowledge base, and the implementation of a forward-chaining inference engine using Python and the Experta library (Experta Developers, n.d.). The system integrates a user interface, a fact base, an inference engine, and an explanation module, enabling the generation of justified diagnoses and technical recommendations in accordance with the principles of Explainable Artificial Intelligence (Adadi & Berrada, 2018). The results demonstrate that the prototype effectively replicates the reasoning of network administration specialists and serves as a transparent, scalable tool to support the diagnostic process.

Palabras clave: Sistemas Expertos, Ingeniería del Conocimiento, Redes LAN, Inteligencia Artificial Explicable, Forward Chaining.

Key Words: *Expert Systems, Knowledge Engineering, LAN Networks, Explainable Artificial Intelligence, Forward Chaining.*

[1] Mtra. Ariopajita Rojo López es docente del TECNM campus ITSJR. ariopajita.rl@sjuanrio.tecnm.mx
[2] Mtra. Rocío Guadalupe Zozaya Salas es docente del TECNM campus ITSJR. rocio.zs@sjuanrio.tecnm.mx
[3] Lic. Guadalupe Flores Diaz es docente del TECNM campus ITSJR. guadalupe.fd@sjuanrio.tecnm.mx
[4] Lic. Minnie Ruth Fragoso Cobos es docente del TECNM campus ITSJR. minnie.fc@sjuanrio.tecnm.mx

ISSN 2594-2298
Prefijo DOI: 10.70417

INTRODUCCIÓN

La Inteligencia Artificial (IA) ha experimentado un crecimiento significativo durante la última década debido al desarrollo de modelos de aprendizaje automático y aprendizaje profundo capaces de resolver problemas complejos en diversos dominios. Sin embargo, el incremento en la capacidad predictiva de estos modelos ha estado acompañado por una disminución en la interpretabilidad de sus decisiones, dando origen a los denominados modelos de "caja negra". Esta situación ha impulsado el desarrollo de la Inteligencia Artificial Explicable (Explainable Artificial Intelligence, XAI), cuyo objetivo consiste en proporcionar mecanismos que permitan comprender, justificar y validar el razonamiento seguido por un sistema inteligente para emitir una determinada conclusión (Adadi & Berrada, 2018; Arrieta et al., 2020).

Dentro de las técnicas de IA que ofrecen un razonamiento inherentemente interpretable, los Sistemas Expertos continúan representando una alternativa vigente para el desarrollo de sistemas de apoyo a la toma de decisiones. A diferencia de otros enfoques basados en aprendizaje automático, los Sistemas Expertos fundamentan su funcionamiento en la representación explícita del conocimiento mediante reglas de producción, las cuales permiten reproducir el razonamiento de especialistas humanos en dominios específicos. Esta característica facilita la validación del conocimiento, el mantenimiento de la base de conocimientos y la explicación del proceso de inferencia, aspectos esenciales en aplicaciones donde la transparencia y la confianza constituyen requisitos fundamentales (Giarratano & Riley, 2005; Jackson, 1999).

En el ámbito de las Tecnologías de la Información, el diagnóstico de fallas en redes de área local (Local Area Networks, LAN) representa una actividad que demanda experiencia técnica para identificar la causa de incidencias relacionadas con la conectividad física, la configuración lógica y la disponibilidad de servicios de red. Aunque existen herramientas de monitoreo y administración de infraestructura, muchas decisiones iniciales de diagnóstico continúan dependiendo del conocimiento del administrador de redes, quien debe analizar múltiples evidencias antes de determinar la causa probable del problema. Formalizar este conocimiento mediante un Sistema Experto permite conservar la experiencia acumulada, reducir la variabilidad en el diagnóstico y proporcionar recomendaciones consistentes durante el proceso de resolución de incidencias (Castillo, Gutiérrez & Hadi, 1997).

Con base en esta problemática, se desarrolló NetExpert-XAI, un Sistema Experto orientado al diagnóstico de fallas básicas en redes LAN mediante un enfoque basado en reglas de producción. El sistema fue diseñado siguiendo la metodología de Ingeniería del Conocimiento, contemplando las etapas de identificación del dominio, adquisición del conocimiento experto, representación formal mediante reglas IF–THEN, construcción de la base de conocimientos e implementación de un motor de inferencia utilizando la biblioteca Experta del lenguaje de programación Python. La estrategia de inferencia adoptada corresponde al encadenamiento hacia adelante (Forward Chaining), permitiendo que el sistema evalúe los hechos proporcionados por el usuario para activar automáticamente las reglas cuya condición sea satisfecha.

A diferencia de otros prototipos de diagnóstico centrados únicamente en emitir un resultado, NetExpert-XAI incorpora un módulo de explicación que informa al usuario la regla activada, el nivel de confianza asociado al diagnóstico y la recomendación técnica correspondiente.

ISSN 2594-2298
Prefijo DOI: 10.70417

Este enfoque fortalece la transparencia del proceso de razonamiento y alinea el desarrollo del sistema con los principios de la Inteligencia Artificial Explicable, favoreciendo tanto la comprensión del diagnóstico como la validación de la base de conocimientos por parte de especialistas en redes.

El objetivo de este trabajo consiste en diseñar e implementar un Sistema Experto basado en reglas de producción para el diagnóstico explicable de fallas en redes LAN, utilizando técnicas de Ingeniería del Conocimiento y un motor de inferencia desarrollado en Python mediante la biblioteca Experta. Asimismo, se presenta la arquitectura del prototipo, la organización modular del sistema, el proceso de implementación y la validación funcional realizada mediante diferentes escenarios de diagnóstico, demostrando la viabilidad de emplear Sistemas Expertos como herramientas de apoyo para la identificación de incidencias en infraestructura de redes.

METODOLOGÍA

El desarrollo de NetExpert-XAI se llevó a cabo siguiendo una metodología basada en los principios de la Ingeniería del Conocimiento, disciplina encargada de adquirir, representar e implementar el conocimiento especializado dentro de sistemas computacionales capaces de reproducir el razonamiento de un experto humano. Esta metodología permitió estructurar el proceso de desarrollo desde la identificación del problema hasta la validación funcional del prototipo, garantizando la separación entre el conocimiento del dominio y el mecanismo de inferencia (Giarratano & Riley, 2005; Castillo, Gutiérrez, & Hadi, 1997).

La investigación se desarrolló bajo un enfoque aplicado, debido a que busca resolver un problema específico relacionado con el diagnóstico de fallas en redes de área local (LAN), mediante la construcción de un sistema experto funcional. Asimismo, corresponde a un estudio de tipo descriptivo y de desarrollo tecnológico, ya que describe el proceso de construcción del sistema y presenta un prototipo operativo como resultado de la investigación. También se tomo en cuenta la metodología basada en proyectos que apoya perfectamente a este tipo de proyectos.

La metodología seguida estuvo integrada por seis etapas principales, las cuales se muestran en la Figura 1.

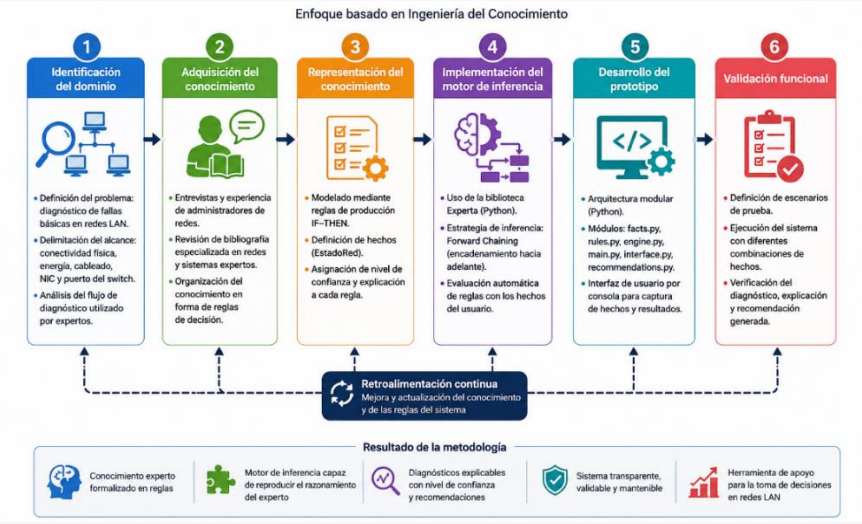


Figura 1. Metodología de desarrollo de NetExpert-XAI

ISSN 2594-2298
Prefijo DOI: 10.70417

Etapa 1. Identificación del dominio

Se definió como dominio de aplicación el diagnóstico de fallas básicas en redes LAN, considerando aquellas incidencias que normalmente representan el primer nivel de atención durante los procesos de soporte técnico. Para delimitar el alcance inicial del prototipo se seleccionaron problemas relacionados con la conectividad física, incluyendo la disponibilidad de energía eléctrica, el estado del cable de red, la operación de la tarjeta de red y la disponibilidad del puerto del switch. Esta delimitación permitió construir un modelo inicial suficientemente representativo y susceptible de ampliarse en futuras versiones.

Etapa 2. Adquisición del conocimiento

El conocimiento utilizado para construir la base de conocimientos fue obtenido mediante la experiencia profesional en administración y diagnóstico de redes, complementada con bibliografía especializada en redes de computadoras, administración de infraestructura y Sistemas Expertos. Posteriormente, dicho conocimiento fue organizado en forma de reglas de decisión siguiendo la lógica utilizada por un administrador de redes durante el proceso de diagnóstico.

Etapa 3. Representación del conocimiento

El conocimiento adquirido fue representado mediante reglas de producción del tipo IF–THEN, implementadas utilizando la biblioteca Experta para Python. Cada regla establece un conjunto de condiciones que deben cumplirse para activar un diagnóstico específico y emitir la recomendación técnica correspondiente. Los hechos del sistema fueron modelados mediante la clase EstadoRed, la cual almacena el estado de los componentes evaluados durante el proceso de inferencia.

Etapa 4. Implementación del motor de inferencia

El motor de inferencia fue desarrollado utilizando la biblioteca Experta, la cual implementa un mecanismo de razonamiento basado en Forward Chaining (encadenamiento hacia adelante). Durante la ejecución del sistema, los hechos declarados por el usuario son comparados automáticamente contra las reglas de producción almacenadas en la base de conocimientos. Cuando las condiciones de una regla son satisfechas, el motor activa la regla correspondiente y genera el diagnóstico asociado.

Etapa 5. Desarrollo del prototipo

El prototipo fue implementado siguiendo una arquitectura modular desarrollada en Python, organizando los componentes del sistema en archivos independientes de acuerdo con su responsabilidad funcional. La estructura del proyecto comprende los módulos encargados de representar los hechos (facts.py), almacenar las reglas de producción (rules.py), implementar el motor de inferencia (engine.py), gestionar la interacción con el usuario (main.py e interface.py) y administrar las recomendaciones técnicas (recommendations.py). Esta organización facilita el mantenimiento del código y permite incorporar nuevas reglas o módulos de diagnóstico sin modificar la arquitectura general del sistema.

Etapa 6. Validación funcional

La validación del prototipo se realizó mediante escenarios de prueba previamente definidos, simulando diferentes condiciones de operación de una red LAN. Cada escenario corresponde a una combinación específica de hechos suministrados al motor de inferencia, verificando

ISSN 2594-2298
Prefijo DOI: 10.70417

que el sistema activara la regla de producción esperada y generara el diagnóstico y la recomendación técnica correspondientes. Los escenarios implementados consideran tanto condiciones normales de operación como fallas relacionadas con la conectividad física, permitiendo comprobar el correcto funcionamiento del mecanismo de inferencia y del módulo de explicación.

Finalmente, el desarrollo del sistema se apoyó en diversas herramientas tecnológicas. El lenguaje de programación utilizado fue Python 3, mientras que la implementación del razonamiento basado en reglas se realizó mediante la biblioteca Experta. El desarrollo y las pruebas del prototipo se efectuaron utilizando Visual Studio Code como entorno de desarrollo integrado, favoreciendo una implementación modular y fácilmente extensible. La Tabla 1 resume las tecnologías utilizadas durante el desarrollo de NetExpert-XAI.

Tabla 1. Tecnologías utilizadas para el desarrollo de NetExpert-XAI

Componente	**Tecnología**
Lenguaje de programación	Python 3
Motor de inferencia	Experta
Paradigma	Sistemas Expertos basados en reglas
Estrategia de inferencia	Forward Chaining
Entorno de desarrollo	Visual Studio Code
Sistema Operativo	Windows 11
Arquitectura	Modular

Fuente: Elaboración propia

RESULTADOS

Implementación del prototipo

El sistema experto desarrollado, denominado NetExpert-XAI, fue implementado en el lenguaje Python utilizando la biblioteca Experta, la cual permite construir motores de inferencia basados en reglas de producción y encadenamiento hacia adelante (Forward Chaining). La arquitectura del sistema fue organizada en módulos independientes que facilitan su mantenimiento y escalabilidad.

La implementación considera una base de conocimientos formada por hechos (Facts), reglas de producción (Rules), un motor de inferencia (Knowledge Engine) y una interfaz de usuario desarrollada en consola que permite seleccionar distintos escenarios de diagnóstico. Esta organización modular favorece la reutilización del conocimiento y permite incorporar nuevas reglas sin modificar la estructura principal del sistema, característica fundamental de los Sistemas Expertos modernos (Russell & Norvig, 2021; Giarratano & Riley, 2005).

La versión actual implementa el diagnóstico de la capa física de una red LAN, considerando las fallas más comunes reportadas durante el proceso de adquisición del conocimiento con especialistas en administración de redes.

La figura 2 muestra la arquitectura que se consideró para el desarrollo del prototipo.

ISSN 2594-2298

Prefijo DOI: 10.70417

© FEGLININ

| Año 10, No 39, octubre - diciembre 2026 |
https://doi.org/10.70417/

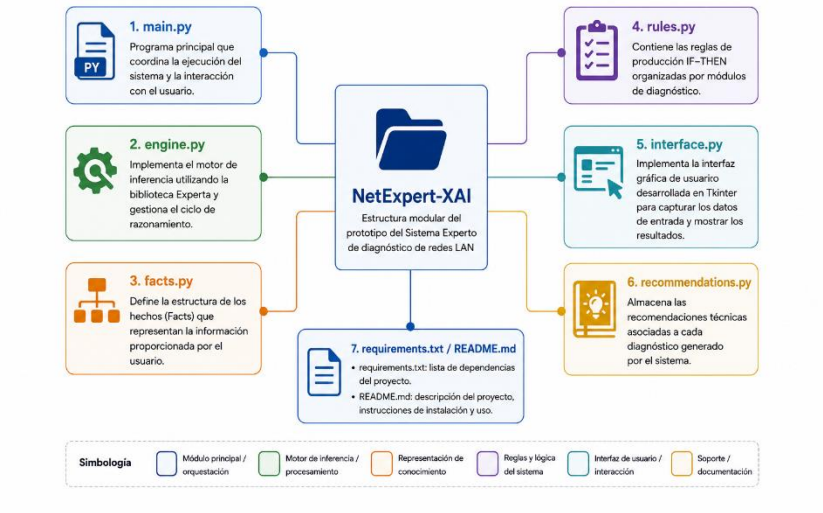


Figura 2. Organización de los módulos del proyecto

Base de conocimientos implementada

Como resultado de la ingeniería del conocimiento se implementó una base inicial integrada por cinco reglas correspondientes al diagnóstico de la capa física de una red local, como se puede observar en la tabla 2.

Tabla 2 Reglas utilizadas para el sistema experto

Regla	Condición	Diagnóstico	Acción recomendada
R1	Energía=SI, Cable=SI, Tarjeta=SI, Puerto=SI	Conexión correcta	No requiere acción
R2	Energía=NO	Ausencia de energía	Revisar alimentación
R3	Cable=NO	Cable desconectado	Revisar cable UTP
R4	Tarjeta=NO	Tarjeta de red deshabilitada	Reinstalar controlador
R5	Puerto=NO	Puerto del switch deshabilitado	Cambiar puerto o habilitarlo

Validación funcional

Para evaluar el funcionamiento del sistema se diseñaron diversos escenarios de prueba que representan las fallas más comunes de la capa física en una red LAN.

Cada escenario fue ejecutado mediante la interfaz desarrollada, verificando que el motor de inferencia activara únicamente la regla correspondiente y generara una explicación del razonamiento seguido.

Los resultados muestran que el sistema identifica correctamente la condición presente, emite un diagnóstico consistente con la base de conocimiento y proporciona recomendaciones específicas para la resolución del problema.

ISSN 2594-2298
Prefijo DOI: 10.70417

Tabla 3 Resultados de la validación funcional

Escenario	Resultado esperado	Resultado obtenido	Correcto
Conexión correcta	R1	R1	Sí
Sin energía	R2	R2	Sí
Cable desconectado	R3	R3	Sí
Tarjeta deshabilitada	R4	R4	Sí
Puerto del switch	R5	R5	Sí

La Figura 4 presenta la interfaz de usuario desarrollada en consola, la cual constituye el Shell del sistema experto. A través de esta interfaz el usuario puede seleccionar diferentes escenarios de prueba sin necesidad de modificar el código fuente. La información capturada es enviada automáticamente al motor de inferencia para ejecutar el proceso de diagnóstico correspondiente. Esta interfaz permite una interacción sencilla e intuitiva durante la validación del prototipo.

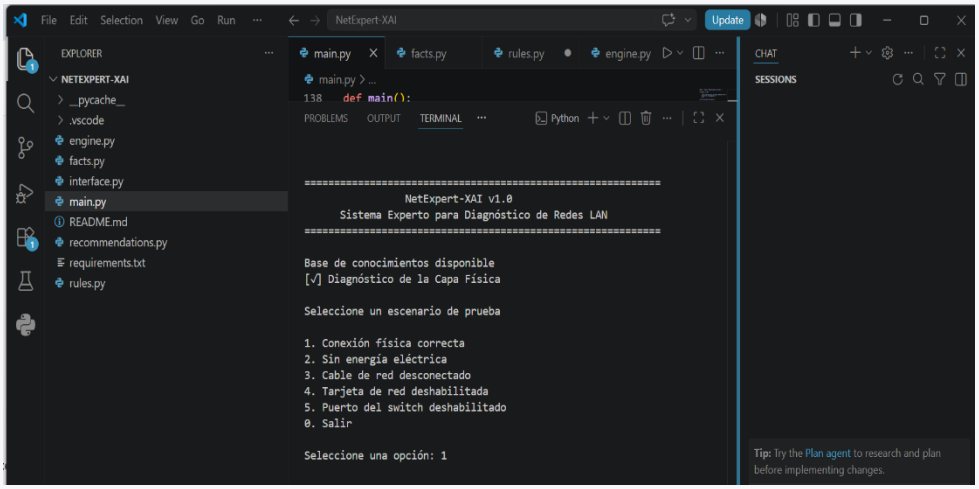


Figura 4 Interfaz de usuario

Fuente Elaboración propia

El resultado obtenido cuando todas las condiciones de la capa física son satisfactorias, como se observa en la figura 5. El sistema identifica correctamente que existe conectividad física entre los dispositivos de la red y activa la Regla R1. Además del diagnóstico, el motor de inferencia proporciona el nivel de confianza, la recomendación correspondiente y una explicación del razonamiento utilizado para llegar a dicha conclusión, fortaleciendo la transparencia del sistema.

ISSN 2594-2298

Prefijo DOI: 10.70417

© FEGLININ

| Año 10, No 39, octubre - diciembre 2026 |
https://doi.org/10.70417/

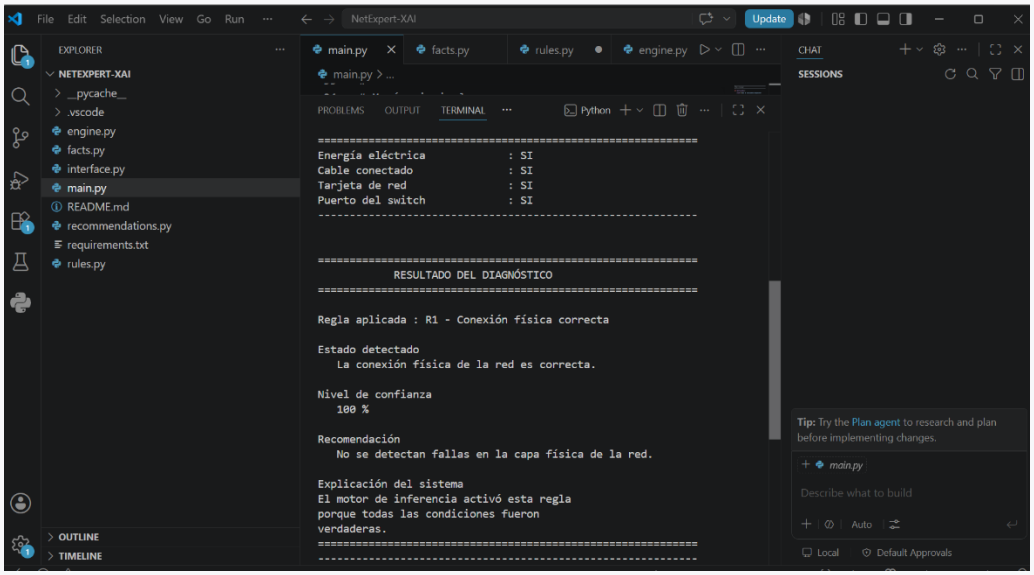


Figura 5 Resultado de la activación de la regla 1

Fuente: Elaboración propia

La Figura 6 presenta la ejecución de la Regla R3 correspondiente al escenario en el que el cable de red no se encuentra conectado. El motor de inferencia detecta automáticamente esta condición y genera un diagnóstico específico indicando la posible causa del problema. Asimismo, el sistema recomienda verificar las conexiones físicas o sustituir el cable si presenta daños, proporcionando información útil para la resolución de la falla.

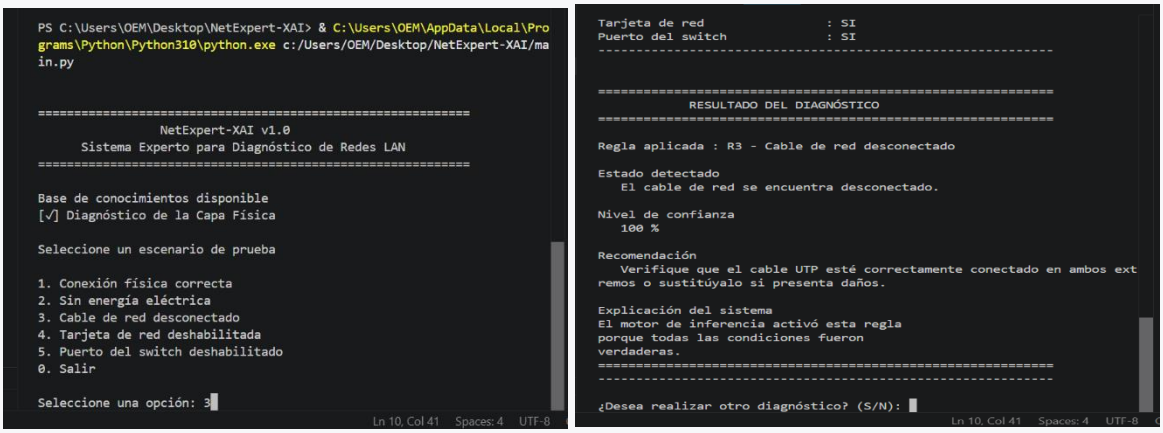


Figura 6 Resultado de la activación de la regla 3

Fuente: Elaboración propia

La Figura 7 ilustra el funcionamiento del sistema cuando la tarjeta de red presenta una falla o se encuentra deshabilitada. En este escenario se activa la Regla R4, identificando correctamente el origen del problema y generando una recomendación orientada a revisar el controlador del dispositivo, habilitar el adaptador de red o reemplazar el hardware en caso de una falla física. Este resultado evidencia la correcta representación del conocimiento experto dentro del sistema.

ISSN 2594-2298
Prefijo DOI: 10.70417

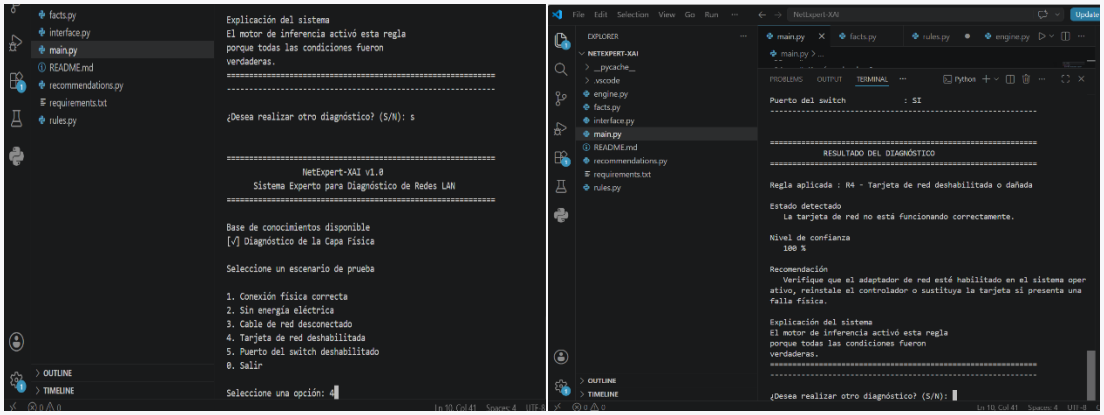


Figura 7 Resultado de la activación de la regla 4

Fuente: Elaboración propia

La Figura 8 muestra la activación de la Regla R5 cuando el puerto del switch no proporciona conectividad a la red. El sistema identifica la condición a partir de los hechos ingresados y genera un diagnóstico acompañado de recomendaciones para verificar la configuración del puerto o utilizar uno alternativo. Este escenario demuestra la capacidad del sistema experto para diferenciar entre distintas causas de pérdida de conectividad física.

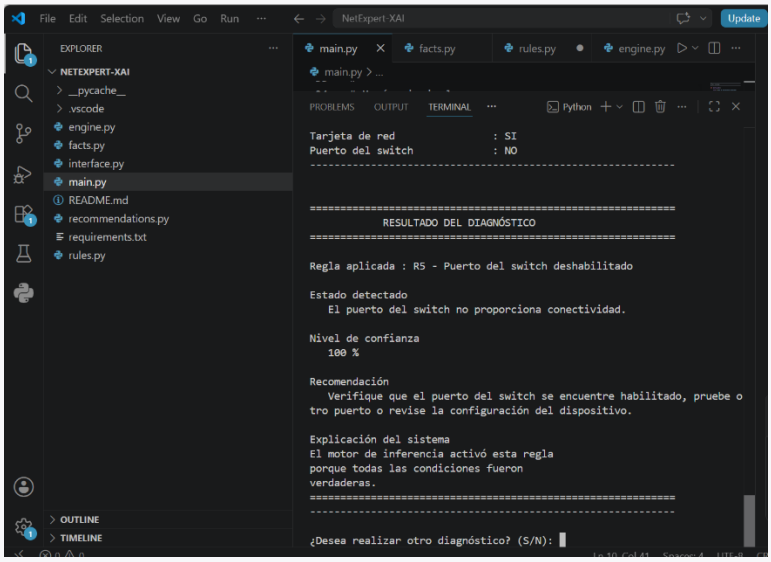


Figura 8 Resultado de la activación de la regla 5

Fuente: Elaboración propia

Las iteraciones se repiten hasta completar el número de reglas diseñadas para este sistema, la figura 9 muestra el flujo del proceso de inferencia (forward chaining), encadenamiento hacia adelante en el sistema experto NetExpert-XAI.

ISSN 2594-2298

Prefijo DOI: 10.70417

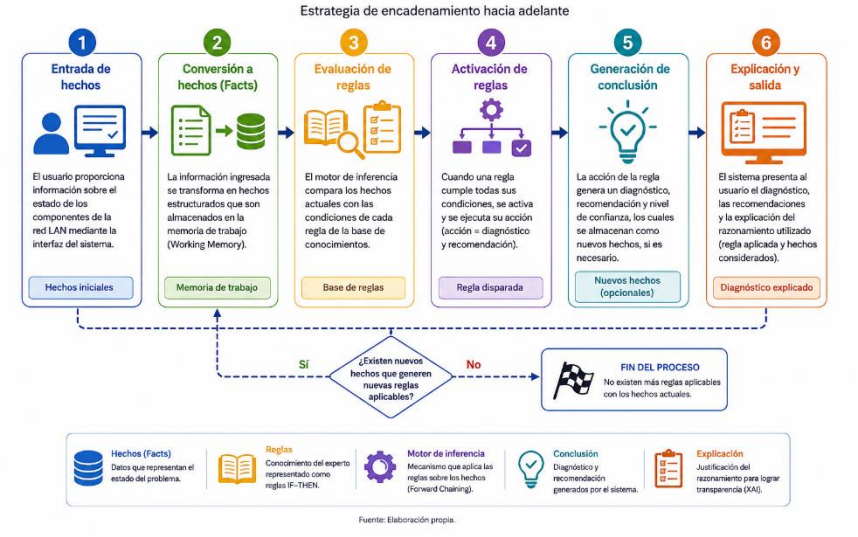


Figura 9 Flujo del proceso de inferencia en NetExpert-XAI

Fuente: Elaboración propia

DISCUSIÓN

Los resultados obtenidos demuestran que la metodología basada en reglas de producción permite reproducir de manera consistente el razonamiento utilizado por un especialista durante el diagnóstico inicial de fallas en redes LAN.

La utilización del motor de inferencia de Experta facilitó la implementación del encadenamiento hacia adelante, permitiendo seleccionar automáticamente la regla que satisface las condiciones declaradas por el usuario. Asimismo, la incorporación de explicaciones junto con cada diagnóstico incrementa la transparencia del sistema y fortalece uno de los principios fundamentales de la Inteligencia Artificial explicable (XAI), al permitir que el usuario comprenda las razones que justifican la decisión emitida.

Aunque la versión actual se limita al diagnóstico de la capa física, la arquitectura modular diseñada facilita la incorporación de nuevos hechos, reglas y niveles de diagnóstico correspondientes a las capas de enlace, red y transporte, lo que representa una línea natural de evolución del sistema.

CONCLUSIONES

El desarrollo del sistema experto NetExpert-XAI permitió comprobar que la Ingeniería del Conocimiento constituye una metodología adecuada para representar y automatizar el razonamiento utilizado por especialistas en el diagnóstico de fallas en redes LAN, Kurose, J. F., & Ross, K. W. (2022). La implementación de una base de conocimientos basada en reglas de producción, junto con un motor de inferencia de encadenamiento hacia adelante desarrollado mediante la biblioteca Experta, permitió generar diagnósticos consistentes, transparentes y acompañados de explicaciones que facilitan la comprensión de las decisiones emitidas por el sistema.

Los resultados obtenidos durante la validación funcional demostraron que el prototipo identifica correctamente las principales fallas de la capa física de una red local, activando únicamente la regla correspondiente a cada escenario de prueba. Asimismo, la incorporación

ISSN 2594-2298
Prefijo DOI: 10.70417

de recomendaciones específicas para cada diagnóstico incrementa el valor práctico del sistema como herramienta de apoyo para estudiantes y personal encargado de la administración básica de redes.

Una de las principales contribuciones de este trabajo consiste en integrar los principios de la Inteligencia Artificial Explicable (XAI) dentro de un sistema experto clásico, proporcionando no solo un diagnóstico, sino también una explicación del razonamiento seguido por el motor de inferencia, Arrieta et al. (2020). Esta característica favorece la transparencia, la confianza del usuario y el proceso de aprendizaje durante la resolución de problemas en redes de computadoras.

Como trabajo futuro, se propone ampliar la base de conocimientos para incorporar reglas correspondientes a las capas de enlace, red y transporte del modelo TCP/IP, así como integrar mecanismos de adquisición automática del conocimiento, factores de incertidumbre y una interfaz gráfica de usuario que permita mejorar la experiencia de interacción y facilite la utilización del sistema en entornos educativos y profesionales.

REFERENCIAS BIBLIOGRÁFICAS

Adadi, A., & Berrada, M. (2018). *Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). IEEE Access, 6,* 52138–52160. https://doi.org/10.1109/ACCESS.2018.2870052

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., García, S., Gil-López, S., Molina, D., Benjamins, R., Chatila, R., & Herrera, F. (2020). *Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. Information Fusion, 58,* 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

Castillo, E., Gutiérrez, J. M., & Hadi, A. S. (1997). *Expert Systems and Probabilistic Network Models*. Springer. https://doi.org/10.1007/978-1-4612-2290-2

Giarratano, J., & Riley, G. (2005). *Expert Systems: Principles and Programming* (4th ed.). Course Technology.

Jackson, P. (1999). *Introduction to Expert Systems* (3rd ed.). Addison-Wesley.

Python Software Foundation. (2025). *Python Language Reference (Version 3.13).* https://www.python.org/

Russell, S., & Norvig, P. (2021). *Artificial Intelligence: A Modern Approach* (4th ed.). Pearson.

Salvatore, M. (2023). *Experta: A Python expert system library*. GitHub. https://github.com/nilp0inter/experta

Kurose, J. F., & Ross, K. W. (2022). *Computer Networking: A Top-Down Approach* (8th ed.). Pearson.