UACM

Universidad Autónoma
de la Ciudad de México

NADA HUMANO ME ES AJENO

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE TELECOMUNICACIONES

Implementación, administración y monitoreo de una red LAN utilizando software libre

T E S I S

QUE PARA OPTAR POR EL TÍTULO DE

LICENCIADOS EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y DE TELECOMUNICACIONES

P R E S E N T A N

ISEO SALVADOR MARTÍNEZ GONZÁLEZ

MIGUEL ÁNGEL ROJAS ROMERO

D I R E C T O R

MTRO. ENRIQUE CRUZ MARTÍNEZ

Ciudad de México, marzo de 2025.

UACM
Universidad Autónoma de la Ciudad de México
Nada humano me es ajeno

CA Coordinación Académica

SISTEMA BIBLIOTECARIO DE INFORMACIÓN Y DOCUMENTACIÓN

UNIVERSIDAD AUTÓNOMA DE LA CIUDAD DE MÉXICO
COORDINACIÓN ACADÉMICA

RESTRICCIONES DE USO PARA LAS TESIS DIGITALES

DERECHOS RESERVADOS©

Agradecimientos

Iseo

A mi padre, por ser mi inspiración desde siempre y por mostrarme, con su ejemplo, el camino hacia la ingeniería.
A mi madre, por ser mi guía en los momentos clave de la vida y por enseñarme la importancia de la perseverancia.
A mis hermanos, por estar siempre presentes, apoyándome y acompañándome en este viaje.
A Dafne, por ser mi apoyo constante y por recordarme que todo esfuerzo tiene su recompensa.
A mis abuelas, tíos y tías, por su amor y sus palabras de ánimo, que siempre me impulsaron a seguir adelante.
A mis profesores, por su paciencia, su guía y por compartir conmigo su conocimiento, que fue fundamental para alcanzar este logro.

Miguel

A mi familia, por ser mi pilar fundamental y brindarme siempre su apoyo incondicional.
A mis amigos y compañeros, por las experiencias compartidas y por hacer más llevadero este recorrido académico.
A mis profesores, por su dedicación y esfuerzo al transmitirme sus conocimientos, motivándome a superarme en cada etapa.
A todas las personas que me inspiraron con su ejemplo y que me mostraron que la perseverancia y el esfuerzo siempre dan frutos.

Resumen

En este proyecto, se diseña y simula una red de área local (LAN) que representa la infraestructura del Laboratorio de Matemáticas LAMAT y el Laboratorio de Enseñanza para la Ciencia (LACECI), ubicados en la Universidad Autónoma de la Ciudad de México (UACM), en el campus San Lorenzo Tezonco. La simulación se realizó en GNS3 y considera diversos dispositivos de comunicación esenciales, como conmutadores, enrutadores y estaciones de trabajo.

Como parte del proyecto, se implementa un sistema de monitoreo basado en Zabbix, un software de código libre configurado en una máquina virtual con sistema operativo Ubuntu en su versión LTS (Long Term Support). Este sistema es compatible con el virtualizador Oracle VirtualBox y permite supervisar métricas clave, como el uso de memoria RAM, el estado de las interfaces y puertos, así como el rendimiento del CPU. Para mejorar la visualización de los datos recopilados, se emplea Grafana, que facilita el análisis gráfico de la información, y se configura un sistema de notificaciones que alerta sobre fallos o problemas en la infraestructura.

Además, el sistema de monitoreo también se implementa en una Raspberry Pi 3, esta solución permite contar con un dispositivo de bajo costo y bajo consumo energético, ideal para la supervisión de redes pequeñas. Finalmente, se aplica el método de Isolation Forest para detectar anomalías en el comportamiento del uso del CPU y la memoria de una de las máquinas virtuales simuladas, lo que contribuye a una mejor gestión del rendimiento del sistema. Los resultados demuestran la eficacia del sistema de monitoreo propuesto, destacando su capacidad para identificar y alertar sobre posibles incidencias en tiempo real.

Palabras clave: Red LAN, TCP/IP, software libre, monitoreo, protocolo SNMP, Raspberry Pi.

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introducción

Hoy en día, es fundamental contar con control e información sobre el estado de una red para garantizar su correcto funcionamiento, independientemente de su tamaño o propósito, ya sea una red LAN, WAN o PAN. El monitoreo implica supervisar continuamente dispositivos como enrutadores, conmutadores, servidores y estaciones de trabajo, una tarea que puede complicarse en redes con múltiples ubicaciones y cientos de dispositivos.

El monitoreo permite a administradores y usuarios obtener una visión integral del estado de los dispositivos y del flujo de datos. Esto facilita la detección y resolución de problemas de manera oportuna, previene interrupciones, mejora el rendimiento y garantiza la seguridad de la infraestructura.

En este contexto, las herramientas de software libre se presentan como una solución eficiente y accesible. Ofrecen funciones como la recopilación de datos en tiempo real, el análisis de tráfico, identificación de anomalías y la generación de informes que permiten tomar decisiones correctivas. Su flexibilidad las convierte en opciones ideales para monitorear aspectos clave, como el uso de ancho de banda, el rendimiento de los dispositivos y la detección de ataques.

Por otro lado, el software privado, también conocido como software propietario, pertenece a individuos o empresas y está protegido por derechos de autor que limitan su uso. A diferencia del software libre, no permite el acceso ni la modificación de su código fuente y se comercializa generalmente bajo licencias de pago. Aunque estas restricciones buscan proteger la inversión en desarrollo, el software privado ofrece ventajas, como soporte técnico especializado y funciones exclusivas. Sin embargo, resulta ser más costoso y menos flexible que las alternativas de código abierto.

1.1. Antecedentes

El monitoreo del flujo de datos en una red es una actividad fundamental para garantizar el correcto funcionamiento de un sistema informático que depende de la conectividad entre diferentes dispositivos. Además, permite detectar y resolver problemas de rendimiento, seguridad, disponibilidad y calidad de servicio que pueden afectar la operatividad de una organización. Se presentan buenas prácticas y desafíos que se han encontrado en la implementación y evaluación del monitoreo de red en diversos contextos y escenarios.

En este caso, se propone utilizar una red simulada en GNS3[1], un software que permite diseñar y emular una red de área local con diversos dispositivos de comunicación esenciales, como conmutadores y estaciones de trabajo.

El sistema de monitoreo se implementa con Zabbix[2], una plataforma de software libre que ofrece múltiples funcionalidades para el monitoreo de red. El servidor de monitoreo se aloja en una máquina virtual creada con Oracle VirtualBox[3], un programa que permite ejecutar diversos sistemas operativos dentro de otro sistema operativo anfitrión. De esta manera, se pretende demostrar la viabilidad y eficacia de una solución de monitoreo de redes basada en software libre.

1.2. Planteamiento del problema

Actualmente, un gran número de universidades, así como de pequeñas y medianas empresas, no cuentan con un área especializada en redes o un centro de monitoreo (NOC, por sus siglas en inglés) que lleve un registro de los signos vitales, como el estado, almacenamiento, demanda y uso de los servidores, conmutadores, enrutadores y otros elementos que conforman su red local. Tampoco existe un área que realice un monitoreo diario de los servicios y aplicaciones que se ejecutan dentro de los dispositivos que componen su red local, lo que provoca que, al presentar un incidente o falla, les tome demasiado tiempo y esfuerzo identificar la posible causa de la misma.

1.3. Justificación

El monitoreo de red reduce el tiempo de inactividad, evita interrupciones y ahorra tiempo y recursos al resolver problemas. Proporciona datos valiosos sobre el uso de recursos, permite evitar

[1]GNS3 es un emulador de redes, el cual permite diseñar cualquier tipo y topología de red pues imita el hardware de los dispositivos de redes de forma virtual, utilizando los recursos de cómputo del host en el que se está ejecutando.

[2]Zabbix es una solución diseñada para monitorear la infraestructura de TI, adecuada tanto para pequeñas empresas con un número limitado de servidores como para corporaciones con una infraestructura de servidores más extensa.

[3]VirtualBox es una herramienta para la virtualización de sistemas operativos, proporcionando un entorno seguro para ejecutar diferentes sistemas y aplicaciones.

actualizaciones costosas e innecesarias, y ofrece ventajas como la detección temprana de problemas, mantenimiento proactivo, mejora del rendimiento, seguridad, planificación de capacidad, cumplimiento normativo, análisis de tendencias y decisiones informadas.

1.4. Objetivos

Garantizar el funcionamiento óptimo de la red mediante la detección de alertas relacionadas con problemas de rendimiento y seguridad, las cuales son canalizadas a los administradores de red para su resolución. Además, facilitar la planificación y la escalabilidad de la infraestructura a través del análisis exhaustivo de los datos obtenidos del monitoreo en el Centro de Datos, el Laboratorio de Cómputo para la Enseñanza de las Ciencias (LACECI) y el Laboratorio de Matemáticas (LAMAT).

1.5. Metodología

- Implementar un centro piloto de monitoreo en una red local simulada en GNS3, incluyendo servidores, conmutadores, enrutadores y estaciones de trabajo.
- Configurar y utilizar Zabbix en una máquina virtual con Ubuntu 22.04 LTS para supervisar el estado y rendimiento de la red y sus servicios en tiempo real.
- Desplegar el monitoreo en una Raspberry Pi 3, configurada como un servidor accesible y de fácil implementación, ideal para redes LAN pequeñas o pruebas demostrativas.
- Generar gráficas y configurar alertas por correo electrónico para notificar fallos o problemas en los dispositivos de la red.
- Aplicar un algoritmo de detección de anomalías, basado en los datos del servidor de monitoreo, para analizar el uso del disco y la CPU de una estación de trabajo.

1.6. Alcances y limitaciones

El principal alcance de este proyecto es la implementación de un sistema de monitoreo utilizando software libre, lo que permite un despliegue rápido y accesible en entornos de red. En el escenario emulado, se logró obtener datos clave, monitorear los signos vitales de los dispositivos, conocer la infraestructura de la red, detectar fallas y notificarlas a través del sistema de alertas implementado. Además, se probó la comunicación mediante SNMP en el entorno emulado, validando su funcionalidad con la herramienta Zabbix.

Sin embargo, una de las principales limitaciones encontradas fue la capacidad de la máquina anfitriona utilizada para la emulación. Debido a los recursos limitados de memoria y RAM, no fue posible agregar más equipos de red sin que el sistema se saturara. En el escenario implementado en la Raspberry Pi, se realizó la instalación y prueba del sistema, aunque no se logró evaluar su funcionamiento en dispositivos físicos que admitieran SNMP. Es importante señalar que la funcionalidad del protocolo SNMP sí fue validada en el entorno emulado.

Capítulo 2

Marco teórico

2.1. Redes de transmisión de información

Una red de transmisión de datos está formada por todos los dispositivos y programas que permiten conectar computadoras y aplicaciones, sin importar la distancia entre ellas. Estos componentes facilitan el envío y la recepción de datos entre los dispositivos y las aplicaciones conectadas.

La red de transmisión de información puede ser de diferentes tipos, dependiendo de su alcance y de la tecnología utilizada.

2.2. Clasificación de redes de acuerdo a su tecnología de transmisión

No existe una clasificación única que englobe todas las redes; no obstante, las clasificaciones por tecnología de transmisión y por escala son las que mejor se adaptan a su clasificación.

Los tres tipos de tecnología de transmisión que se utilizan mayormente en la actualidad son:

1. **Enlaces punto a punto:** Los enlaces punto a punto conectan dispositivos de red. Los paquetes enviados suelen atravesar dispositivos intermedios. Este tipo de transmisión, donde hay un solo emisor y un receptor, se conoce como unidifusión o unicast en inglés.

2. **Enlaces para difusión:** En los enlaces de difusión, todos los dispositivos de la red comparten el canal de comunicación. Los paquetes enviados por un dispositivo pueden ser recibidos por varios o todos los demás dispositivos en la red. Existen dos formas principales de difusión:

   a) **Broadcast:** Los paquetes son enviados a todos los dispositivos conectados al canal de comunicación. Cada dispositivo recibe el paquete, pero solo lo procesa si está dirigido

a él; de lo contrario, lo ignora.

b) **Multicast:** Los paquetes son enviados únicamente a un subconjunto específico de dispositivos en la red que forman parte de un grupo de multidifusión. Esto optimiza el uso del ancho de banda, ya que un solo paquete es replicado únicamente para los destinatarios que lo necesitan.

La figura 2.14 ilustra los tres principales tipos de transmisión de datos en redes.

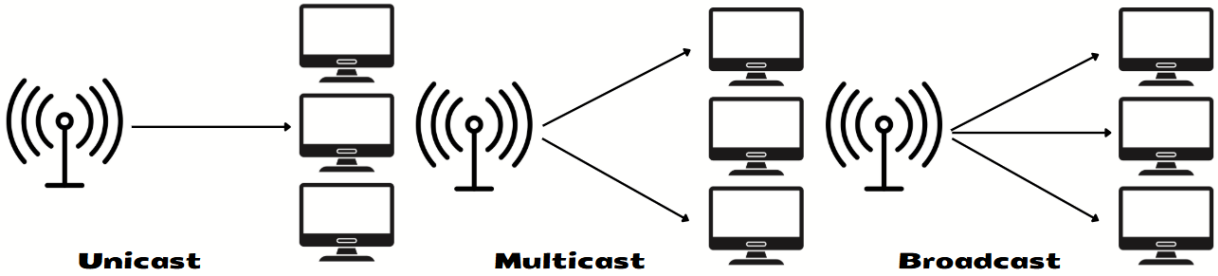


Figura 2.1: Tipos de transmisión: unicast, multicast y broadcast.

En el desarrollo de este proyecto, se utilizará principalmente la transmisión unicast, ya que Zabbix emplea este método para la comunicación entre su servidor y los agentes instalados en los dispositivos monitoreados, así como para las consultas SNMP dirigidas a equipos de red. Esto permite un monitoreo específico de cada dispositivo sin generar tráfico innecesario en la red. Sin embargo, en algunos casos específicos, la funcionalidad de broadcast podría utilizarse para el descubrimiento automático de dispositivos dentro de la red, mientras que multicast podría emplearse en la recepción de traps SNMP enviados por ciertos equipos configurados para notificar eventos a múltiples destinos.

2.3. Clasificación de redes de acuerdo a su alcance

La distancia es un factor importante para clasificar una red, ya que determina las tecnologías empleadas y las capacidades de conexión entre los dispositivos. Estas clasificaciones son fundamentales en el diseño y la implementación de sistemas de comunicación eficientes. La Tabla 2.1 muestra la clasificación de redes según la distancia entre los dispositivos. Según Tanenbaum y Wetherall [1], las redes de mayor alcance requieren infraestructuras tecnológicamente más avanzadas para garantizar la integridad de los datos transmitidos.

Tabla 2.1: Clasificación de las Redes por Distancia y Ubicación

Distancia	**Ubicación**	**Clasificación**
1 m	Metro cuadrado	Red de área personal (PAN)
10 m	Cuarto	Red de área local (LAN)
100 m	Edificio	Red de área local (LAN)
1 km	Campus	Red de área metropolitana (MAN)
10 km	Ciudad	Red de área metropolitana (MAN)
100 km	País	Red de área amplia (WAN)
1000 km	Continente	Red de área amplia (WAN)
10000 km	Planeta	Internet

2.3.1. Redes de área personal

Las redes de área personal (PAN) son fundamentales en el ámbito de la conectividad moderna, ya que permiten la comunicación e interacción entre dispositivos ubicados en la proximidad de una persona. Estas redes conectan dispositivos como teléfonos inteligentes, computadoras portátiles e impresoras, facilitando la transferencia de datos y recursos. Además, son útiles para dispositivos portátiles y tecnologías vestibles, como relojes inteligentes o sensores biomédicos. [2].

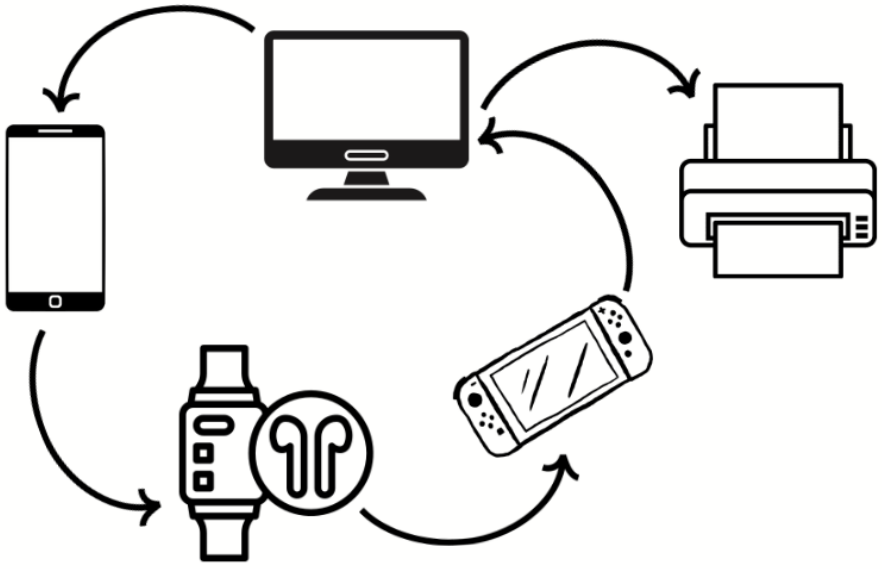


Figura 2.2: Dispositivos en una PAN

La Figura 2.2 ilustra un ejemplo de la conexión entre un teléfono móvil y un ordenador portátil mediante Bluetooth.

La Tabla 2.2 contrasta algunas características de las tecnologías PAN utilizadas en la actualidad. Es importante señalar que las velocidades y el alcance reales de las redes PAN pueden verse afectados por la versión específica de la tecnología utilizada, las condiciones del entorno operativo y las posibles interferencias presentes [3].

Tabla 2.2: Características de Tecnologías PAN

Tecnología	**Características**
Bluetooth	Frecuencias de operación: 2.4 GHz Ancho de banda: Hasta 3 Mbps (dependiendo de la versión y condiciones)
Wi-Fi	Frecuencias de operación: 2.4 GHz, 5 GHz Ancho de banda: Varios cientos de Mbps a varios Gbps
Infrarrojo (IR)	Frecuencias de operación: 300 GHz - 430 THz Ancho de banda: Hasta unos pocos Mbps

2.3.2. Redes de área local

Las redes de área local (LAN, por sus siglas en inglés, Local Area Networks) son sistemas de comunicación que conectan dispositivos a través de hardware privado, como cableado, conmutadores y servidores. Estas redes se limitan a distancias cortas, como edificios, oficinas o fábricas, y permiten el intercambio rápido y seguro de información entre los dispositivos conectados.

La Figura 2.3 ilustra un ejemplo de una LAN.

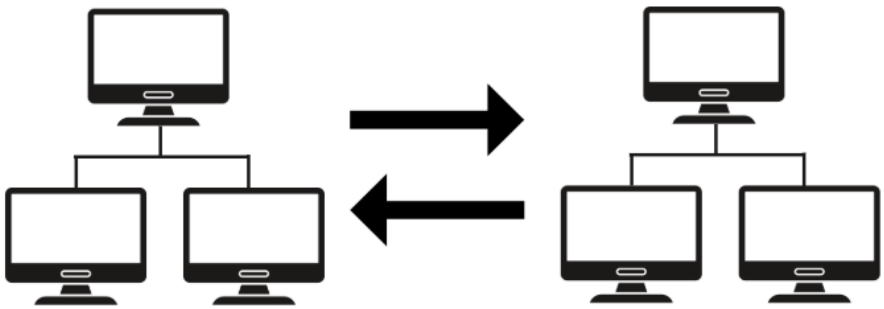


Figura 2.3: Red LAN

El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, por sus siglas en inglés, Institute of Electrical and Electronics Engineers) ha definido la familia de estándares 802 para las redes LAN, la cual abarca tanto redes cableadas como inalámbricas [4]. En la Tabla 2.3 se describen algunos de los estándares más relevantes.

Tabla 2.3: Estándares IEEE 802 y sus descripciones

Estándar	**Descripción**
802.1	Interconexión de redes
802.2	Control de enlace lógico
802.3	LAN en bus con CSMA/CD (Ethernet)
802.4	LAN en bus con testigo (Token Bus)
802.5	LAN en anillo con testigo (Token Ring)
802.6	Red de área metropolitana (MAN)
802.7	Grupo asesor para banda ancha
802.8	Grupo asesor para fibra óptica
802.9	Redes integradas de voz y datos
802.10	Seguridad en redes LAN
802.11	Redes locales inalámbricas (WiFi)
802.15	Redes de área personal (Bluetooth)
802.16	Redes metropolitanas inalámbricas (WiMAX)

En la mayoría de las redes de área local (LAN, por sus siglas en inglés, Local Area Network), siempre se utiliza un modo de transmisión o modulación, ya sea en banda base o banda ancha, un protocolo de acceso al medio como CSMA/CD (Carrier Sense Multiple Access with Collision Detection) o TDMA (Time Division Multiple Access), y un medio físico como cables de par trenzado, coaxiales o fibra óptica.

El medio físico más utilizado en la actualidad es el definido en el estándar EIA/TIA 568B (Electronic Industries Alliance/Telecommunications Industry Association), conocido como par trenzado no apantallado UTP (Unshielded Twisted Pair, por sus siglas en inglés). La categoría más común actualmente es la Cat 5e (Categoría 5 mejorada), que permite transmitir datos a velocidades de hasta 1 Gbps (Gigabit por segundo) y a frecuencias de hasta 100 MHz (Megahercios). También se emplean cables de mayor categoría, como el Cat 6 y Cat 6a, capaces de ofrecer velocidades de hasta 10 Gbps y operar a frecuencias de hasta 500 MHz, siendo ideales para aplicaciones de alta velocidad y redes modernas [5].

Las topologías más utilizadas en las redes LAN incluyen bus, anillo, estrella y malla, siendo esta última especialmente relevante en entornos donde se busca redundancia y resiliencia. Las velocidades de transmisión en las redes LAN pueden variar desde 10 Mbps (Megabits por segundo) hasta 10 Gbps, dependiendo del hardware y las tecnologías implementadas.

Actualmente, este tipo de redes tiende a integrar enlaces inalámbricos (WiFi, por sus siglas en inglés, Wireless Fidelity) para facilitar la movilidad en hogares, empresas y espacios públicos. Las redes inalámbricas se han convertido en un componente esencial de las LAN modernas, proporcionando conectividad sin necesidad de infraestructura cableada, lo que reduce costos y

aumenta la flexibilidad de implementación [5].

Generalmente, las redes LAN se interconectan entre sí mediante redes MAN (Metropolitan Area Network) o WAN (Wide Area Network), formando lo que se conoce como Internet, una red global que conecta dispositivos y sistemas en todo el mundo [1].

2.3.3. Redes de área metropolitana

Las redes de área metropolitana (MAN, por sus siglas en inglés, *Metropolitan Area Network*) son redes informáticas que conectan sistemas de comunicación, tales como ordenadores y dispositivos, en una extensa área geográfica, como una ciudad, localidades o un campus universitario. Estas redes integran servicios de datos, voz y video mediante medios de transmisión de alta velocidad, como fibra óptica, par trenzado u ondas de radio.

Las topologías físicas más utilizadas en las MAN incluyen anillo, malla y bus. Un ejemplo típico de una red MAN es la televisión por cable, que utiliza infraestructuras dedicadas para transmitir señales de video y datos. Además, las MAN permiten la interconexión de redes locales (LAN, por sus siglas en inglés, *Local Area Network*) mediante tecnologías como ATM (*Asynchronous Transfer Mode*). Estas redes también son ampliamente utilizadas para la conexión entre proveedores de servicios de Internet (ISP, por sus siglas en inglés, *Internet Service Provider*) y para enlazar múltiples redes institucionales o corporativas [5].

La Figura 2.4 ilustra un ejemplo de una MAN

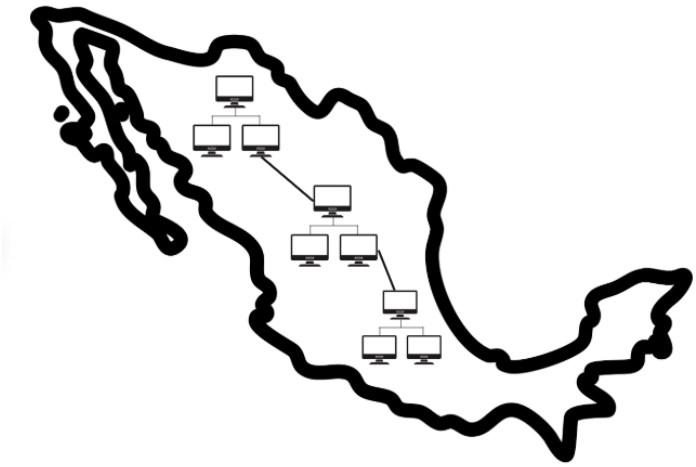


Figura 2.4: Red de área metropolitana (MAN).

2.3.4. Redes de área amplia

Las redes de área amplia (WAN, por sus siglas en inglés, *Wide Area Network*) son sistemas de comunicación que enlazan redes de área local y redes de área metropolitana a través de grandes

distancias geográficas, sin un límite predefinido en su alcance. Estas WAN dependen de la infraestructura de diversos proveedores de servicios de telecomunicaciones, empleando una variedad de tecnologías tanto cableadas (como fibra óptica) como inalámbricas (mediante enlaces satelitales y de microondas). Estas tecnologías emplean la conmutación de paquetes, un método de transmisión de datos donde la información se fragmenta en pequeños paquetes que viajan de forma independiente por la red y se reensamblan al llegar a su destino, optimizando así el uso del ancho de banda disponible [5].

En cuanto a las conexiones cableadas, la fibra óptica es el medio más utilizado en las WAN modernas debido a su capacidad para transmitir grandes volúmenes de datos a alta velocidad. La fibra óptica utiliza principalmente luz láser como fuente de emisión para la transmisión de datos, aunque también puede emplear LED (Diodos Emisores de Luz) en aplicaciones específicas. Este medio opera en un rango de frecuencias de aproximadamente 300 THz a 430 THz, lo que permite alcanzar anchos de banda de varios Gbps hasta cientos de Gbps, dependiendo de su modo de transmisión (multimodo o monomodo) [3].

Por otro lado, las tecnologías inalámbricas, como los enlaces satelitales y de microondas, operan en frecuencias que oscilan entre 1 GHz y 50 GHz. Estas conexiones ofrecen anchos de banda que varían desde Mbps (megabits por segundo) hasta Gbps, y son esenciales en regiones donde la implementación de infraestructura cableada no es viable [1].

La Figura 2.4 ilustra un ejemplo de una WAN

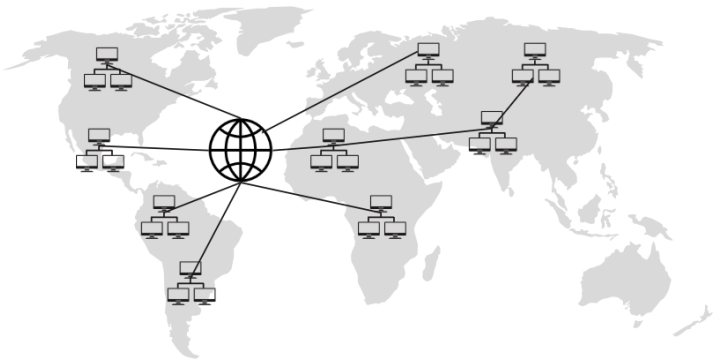


Figura 2.5: WAN utilizada para interconectar diferentes regiones.

En este proyecto se diseña y emula una red de área local conformada por distintos dispositivos de red conectados de forma alámbrica. En la implementación con la Raspberry Pi, se monitorean tanto las redes inalámbricas como las cableadas; sin embargo, ambas siguen siendo parte de la misma LAN.

2.4. Dispositivos de red

Los dispositivos de red son aquellos que permiten la comunicación entre los equipos que forman parte de una red informática. Estos dispositivos pueden clasificarse en diferentes tipos según su función y ubicación dentro de la red. Algunos ejemplos de dispositivos de red incluyen:

a) Enrutadores.

b) Conmutadores.

c) Hubs.

d) Repetidores.

e) Puentes.

f) Firewalls.

g) Servidores.

h) Tarjetas de red de dispositivos de usuario final, como PCs o teléfonos inteligentes.

i) Módems.

Estos dispositivos se ilustran en la Figura 2.6.

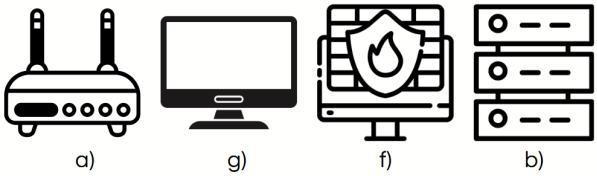


Figura 2.6: Ejemplos de dispositivos de red.

Los dispositivos de red son equipos físicos que permiten conectar computadoras y otros dispositivos para compartir recursos e intercambiar información. Además, realizan funciones esenciales como el direccionamiento, el enrutamiento, el control de errores, la seguridad y la optimización del ancho de banda. En el contexto de redes, cada dispositivo tiene dos tipos de direcciones que lo identifican y permiten su localización y acceso:

- Dirección física (MAC, por sus siglas en inglés, *Media Access Control*): Es única para cada interfaz de red y se utiliza para la comunicación a nivel de la capa de enlace de datos.
- Dirección lógica (IP, por sus siglas en inglés, *Internet Protocol*): Se asigna de manera dinámica o estática y se utiliza para identificar un dispositivo en la capa de red, lo que permite la comunicación entre diferentes subredes.

Los dispositivos de red también pueden clasificarse según el nivel o capa del modelo OSI (Open Systems Interconnection) en el que operan, desde la capa física hasta la capa de aplicación. En la Tabla 2.4 se describen algunos de los dispositivos más comunes y su funcionamiento.

Tabla 2.4: Descripción de dispositivos de red

Dispositivo	**Descripción**
Enrutador	Funciona en la capa de red (capa 3 del modelo OSI) y su principal objetivo es conectar diversas redes, identificando la ruta óptima para transmitir datos según la dirección IP de destino.
Conmutador	Trabaja en la capa de enlace de datos (capa 2 del modelo OSI) y conecta dispositivos dentro de una misma red local (LAN). Reenvía datos basándose en direcciones MAC.
Hub	Es un dispositivo de capa física (capa 1 del modelo OSI) que actúa como un punto central para conectar dispositivos. Reenvía datos a todos los dispositivos conectados sin discriminar el destinatario.
Repetidor	Amplifica o regenera señales debilitadas por la distancia, permitiendo extender el alcance de una red. Opera en la capa física.
Puente	Conecta dos segmentos de red que utilizan el mismo protocolo de nivel de enlace. Filtra y reenvía tramas según la dirección MAC de destino.
Firewall	Protege redes internas mediante el filtrado de tráfico basado en reglas predefinidas. Puede operar en varias capas del modelo OSI, incluyendo las capas de red, transporte y aplicación.
Módem	Convierte señales digitales en analógicas y viceversa, permitiendo la transmisión de datos a través de líneas telefónicas o redes de cable.
Servidor	Un dispositivo que ofrece servicios a otros dispositivos en la red, como almacenamiento, bases de datos o autenticación de usuarios.
Tarjeta de red	Es un componente de hardware instalado en los dispositivos para conectarlos a una red. Puede ser cableada o inalámbrica y opera en las capas física y de enlace.

En el desarrollo de este proyecto se utilizan conmutadores, enrutadores y tarjetas de red en las computadoras para las estaciones de trabajo en el ambiente simulado. Para los dispositivos de red monitoreados en la implementación física, se incluyen estaciones de trabajo, conmutadores y módems de acceso a Internet.

2.5. Topologías de red

El término "topología" se refiere, en esencia, a la forma en que está organizada una red. La topología de red describe cómo se conectan sus nodos o puntos. Existen diversas formas de estructurar una red, y elegir la más adecuada suele ser una de las decisiones clave en su planificación. Cada tipo de topología varía en costos de instalación y mantenimiento, rendimiento y nivel de confiabilidad.[6]

Algunas de las topologías más utilizadas se describen a continuación.

2.5.1. Topología bus

La topología bus es una forma de conectar varios dispositivos en una red de comunicaciones. Consiste en un cable principal, llamado bus, al que se conectan los nodos mediante adaptadores. El bus actúa como un medio compartido que transmite los datos entre los nodos. Cada nodo tiene una dirección única que le permite identificar los mensajes que están destinados a él. Esta dirección única asegura que los datos enviados a través del bus lleguen al nodo correcto sin interferencias. Además, permite a cada nodo comunicarse de manera eficiente con los demás en la red, garantizando así una transmisión de datos fluida y precisa. La topología bus tiene algunas ventajas, como su simplicidad, su bajo costo y su facilidad de instalación y ampliación. Sin embargo, también presenta algunos inconvenientes, como su baja velocidad, su limitada capacidad de transmisión y su baja fiabilidad. Si el bus se rompe o se desconecta, toda la red se interrumpe. Además, el bus puede sufrir interferencias o colisiones si varios nodos intentan transmitir al mismo tiempo. La representación visual se muestra en la figura 2.7

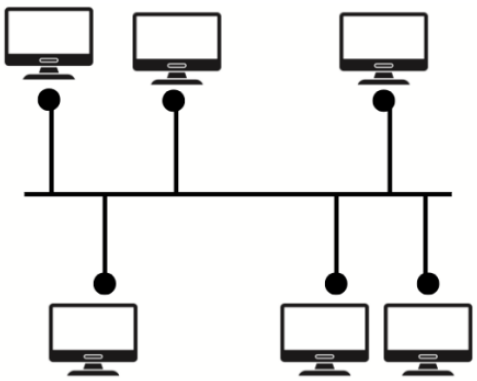


Figura 2.7: Topología bus

2.5.2. Topología estrella

La topología en estrella es un diseño de red donde todos los nodos están conectados directamente a un nodo central, el cual coordina la transmisión de datos entre ellos. Cada nodo cuenta

con una conexión dedicada al nodo central, lo que facilita la administración de la red y mejora su tolerancia a fallos. Si un enlace entre el nodo central y uno de los nodos se interrumpe, solo ese nodo queda fuera de la red, mientras que el resto de la red sigue funcionando de manera normal. Esta característica convierte a la topología estrella en una opción más fiable que la topología de bus.[6].

Entre las ventajas de esta configuración destacan la fácil detección y solución de problemas, su escalabilidad, que permite la adición de nuevos nodos sin afectar al resto de la red, y su alto rendimiento, ya que cada conexión es independiente. Sin embargo, también presenta algunas desventajas, como un mayor costo de instalación debido al número de cables necesarios y su dependencia total del nodo central: si este falla, toda la red deja de funcionar.

La representación visual de la topología en estrella se ilustra en la figura 2.8.

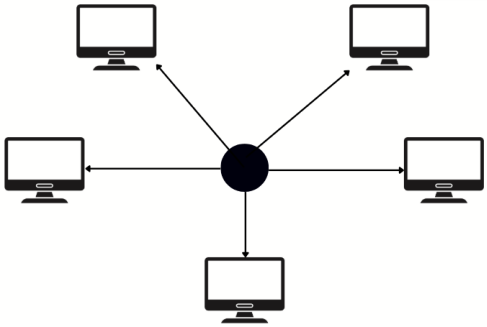


Figura 2.8: Topología estrella

2.5.3. Topología anillo

La topología de anillo es una forma de organizar los dispositivos de una red informática en un bucle circular, donde cada uno se conecta con dos vecinos. En esta configuración, los datos se mueven en una única dirección a través del anillo, pasando por cada dispositivo hasta alcanzar su destino. La topología de anillo tiene algunas ventajas, como la facilidad de instalación y la ausencia de colisiones, pero también presenta desventajas, como la dependencia de cada nodo y la baja velocidad de transmisión. La topología de anillo se relaciona con la topología de bus, que utiliza un cable común para todos los dispositivos, y con el paso de un nodo a otro, que es un método para controlar el acceso al medio de comunicación.

La representación visual de esta topología se muestra en la figura 2.9

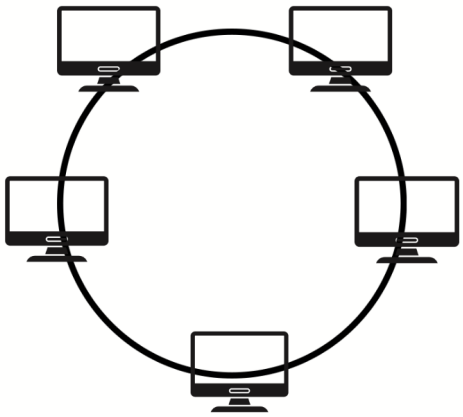


Figura 2.9: Topología anillo

2.5.4. Topología árbol

En la topología tipo árbol, cada nodo tiene un único nodo padre al que está conectado por un enlace, excepto el nodo raíz. Los nodos que no tienen hijos se denominan nodos hoja. La topología de árbol permite crear redes escalables y eficientes, ya que facilita el control del tráfico y la gestión de los recursos. No obstante, esta configuración tiene algunas desventajas, como la dependencia del nodo principal y la posibilidad de congestión si hay demasiados nodos en un mismo nivel.

La figura 2.10 proporciona una representación visual de la topología tipo árbol.

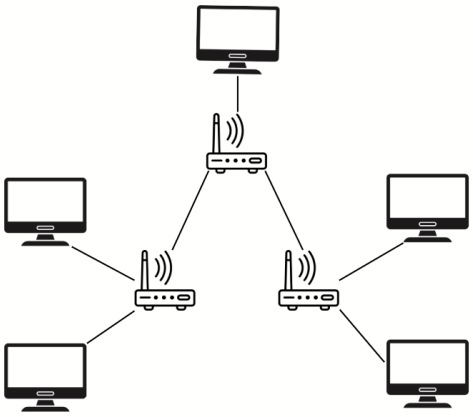


Figura 2.10: Topología arbol

2.5.5. Topología en malla

La topología en malla es una forma de conectar los nodos de una red de comunicaciones en la que cada nodo está conectado a todos los demás nodos. De este modo, se crea una ruta múltiple entre cualquier par de nodos, lo que incrementa la fiabilidad y el rendimiento de la

red [7]. La topología en malla también permite una mayor escalabilidad y flexibilidad, ya que se pueden agregar o eliminar nodos sin afectar al resto de la red. Sin embargo, la topología en malla presenta algunas desventajas, como el alto costo de instalación y mantenimiento, la complejidad de la configuración y el control, así como el mayor consumo de energía y ancho de banda.

La topología de malla puede ser bidimensional o tridimensional, dependiendo de la organización de los nodos.
La Figura 2.11 proporciona una representación visual de la interconexión de los nodos en malla.

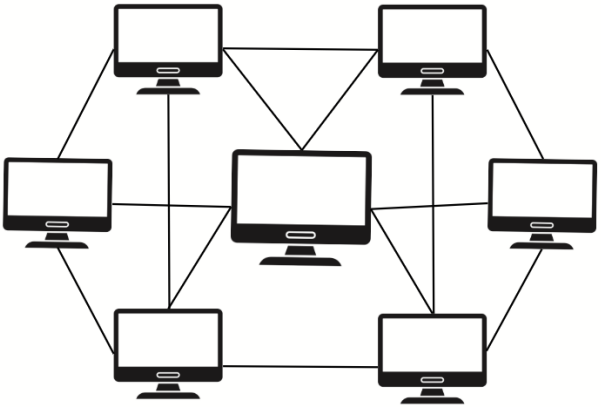


Figura 2.11: Topología malla

1. **Malla Bidimensional:** En una malla bidimensional, los nodos se organizan en un patrón de cuadrícula. Cada nodo está conectado a sus vecinos inmediatos: arriba, abajo, izquierda y derecha. Esto crea una red robusta donde la falla de un solo nodo no interrumpe la comunicación entre los demás. Sin embargo, esta topología puede requerir una gran cantidad de conexiones, lo que puede aumentar la complejidad y el costo.

2. **Malla Tridimensional:** En una malla tridimensional, los nodos se organizan en un patrón de cubo. Cada nodo está conectado a sus vecinos inmediatos en todas las tres dimensiones: arriba, abajo, izquierda, derecha, adelante y atrás. Esta topología proporciona aún más redundancia y robustez que la malla bidimensional, pero también requiere aún más conexiones.

Ambas topologías de malla proporcionan una alta redundancia y fiabilidad, ya que la falla de un nodo no afecta la comunicación entre los otros. Sin embargo, pueden ser más costosas y complicadas de implementar debido a la gran cantidad de conexiones requeridas. Por esta razón, se emplean principalmente en redes donde la fiabilidad es esencial.

2.5.6. Topología doble anillo

La topología de doble anillo es un método de conexión para redes en el que los dispositivos se enlazan a través de dos anillos de cable concéntricos, cada uno transmitiendo datos en direcciones

opuestas. Los dispositivos se conectan a ambos anillos y pueden enviar y recibir datos a través de cualquiera de ellos. Esta topología tiene la ventaja de ofrecer una mayor fiabilidad y redundancia que la topología simple de anillo, ya que si uno de los anillos se rompe, el otro puede seguir funcionando **techlib2023**. Además, permite una mayor velocidad de transmisión al poder utilizar ambos anillos simultáneamente. Sin embargo, también presenta algunas desventajas, como una mayor complejidad y costo de instalación y mantenimiento, así como una menor escalabilidad en comparación con otras topologías.

La Figura 2.12 proporciona una representación visual de la topología de doble anillo.

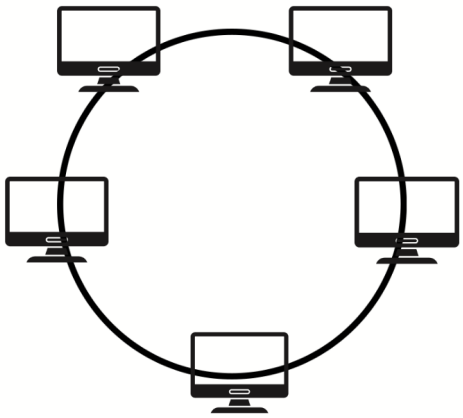


Figura 2.12: Topología doble anillo

2.5.7. Topología híbrida

La topología híbrida integra dos o más tipos de configuraciones de red, como estrella, anillo, bus o malla, adaptándose a diferentes necesidades y estructuras. Algunos ejemplos de redes híbridas son las redes inalámbricas, las redes de área local virtual (VLAN) y las redes de área amplia (WAN). Algunos ejemplos de redes híbridas son las redes inalámbricas, las redes de área local virtual (VLAN) y las redes de área amplia (WAN).

La Figura 2.13 proporciona una representación visual de la topología híbrida.

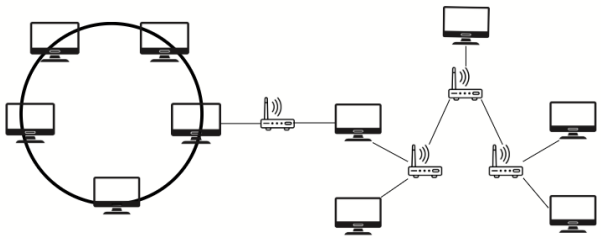


Figura 2.13: Topología híbrida

En este proyecto se implementa una combinación de topologías en estrella y en árbol para la

LAN emulada, con el objetivo de optimizar la conectividad y la gestión del tráfico de red. Las estaciones de trabajo están conectadas a un conmutador central. Además, este conmutador está conectado a otro conmutador que también cuenta con estaciones de trabajo, formando así una topología en árbol al conectar varios conmutadores entre sí, todos al mismo nivel. Uno de estos conmutadores está conectado a un enrutador que actúa como la puerta de enlace principal de la red, gestionando el tráfico hacia y desde otras redes, como Internet.

La combinación de estas topologías permite una escalabilidad eficiente, ya que se pueden agregar más dispositivos y conmutadores sin afectar significativamente el rendimiento de la red.

En la red física donde se implementó la Raspberry Pi, se mantiene la misma topología, con la diferencia de que el servicio de internet llega mediante un nodo a uno de los conmutadores. Además, se incluye una red inalámbrica que también proporciona acceso a Internet.

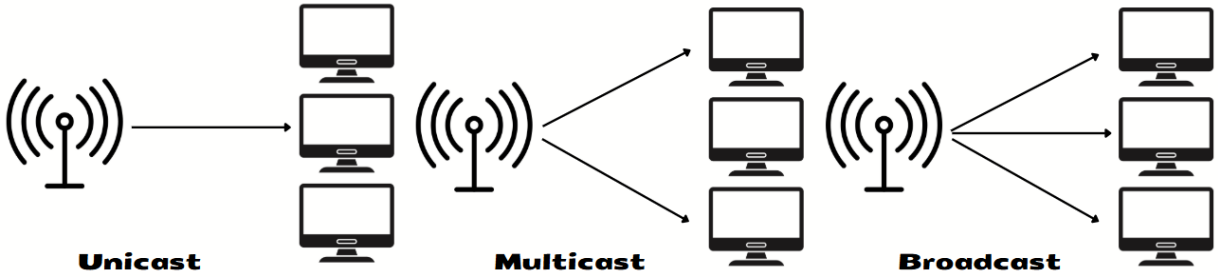


Figura 2.14: Tipos de transmisión: unicast, multicast y broadcast.

La topología resultante en la red física, con la inclusión de una red inalámbrica, se puede describir como una topología híbrida. Esta topología combina elementos de la topología en estrella, en árbol y de red inalámbrica, proporcionando flexibilidad y redundancia en la conectividad.

2.6. Clases de relaciones entre las redes

El término "relaciones entre redes" se refiere a dos conceptos diferentes sobre cómo una computadora utiliza los recursos de otra a través de la red. Existen dos tipos fundamentales de relaciones entre redes: "de igual a igual" y "cliente/servidor". [6]

- **De igual a igual:** En una red de igual a igual, las computadoras interactúan como pares y comparten recursos entre sí, por ejemplo: archivos, programas o dispositivos periféricos (impresoras o módems). Cada equipo gestiona sus propios recursos y establece su seguridad de forma autónoma. Además, cada equipo es responsable de configurar y mantener la seguridad de sus recursos. Finalmente, cada computadora debe poder acceder a los recursos compartidos por otras máquinas en la red de igual a igual, entendiendo su ubicación y los requisitos de seguridad necesarios para acceder a ellos.

- **Cliente/servidor:** Una relación de red cliente/servidor se distingue por la separación entre las computadoras que proporcionan los recursos de la red (servidores) y las que los utilizan (clientes o estaciones de trabajo). En una red cliente/servidor, los recursos, como archivos, aplicaciones y dispositivos compartidos, están centralizados y son administrados por un servidor. Las computadoras cliente acceden a estos recursos sin compartir sus propios datos con otras máquinas. Ninguna de estas computadoras comparte sus recursos con otros clientes o servidores; en cambio, las computadoras cliente solo consumen estos recursos. Los servidores en una red cliente/servidor son responsables de ofrecer y administrar los recursos compartidos de manera adecuada, así como de gestionar su seguridad.

En la tabla 2.5 se muestran algunas ventajas y desventajas de cada uno de los tipos de relaciones entre redes.

Tabla 2.5: Comparación de ventajas y desventajas entre redes de igual a igual y cliente/-servidor

Ventajas	**Desventajas**
De Igual a Igual	
Hardware económico	Bajo rendimiento del usuario
Fácil de administrar	Baja seguridad
No requiere un sistema operativo	Copias de seguridad difíciles de realizar
Redundancia integrada	Sin control de versiones
Cliente/Servidor	
Mayor seguridad	Requiere administración profesional
Mejor rendimiento	Uso de hardware costoso y exigente
Respaldo centralizado	
Mayor confiabilidad	

Estas relaciones establecen la estructura fundamental de una red. Para entenderlas mejor, se pueden comparar con diferentes enfoques de administración empresarial. Una red de igual a igual es similar a una empresa con una administración descentralizada, donde las decisiones se toman localmente y los recursos se gestionan según las necesidades inmediatas. En cambio, una red cliente/servidor se asemeja a una empresa con una administración centralizada, donde un grupo pequeño toma decisiones en un punto central. A menudo, ambos esquemas son apropiados, y muchas redes combinan características de ambos tipos.

Tanto las redes de igual a igual como las de cliente/servidor comparten ciertas capas de red. Ambos tipos necesitan una conexión física entre las computadoras y el uso de los mismos protocolos de red, entre otros aspectos. En este sentido, no hay diferencias entre los dos tipos de

relaciones de red. La diferencia radica en si los recursos compartidos de la red se distribuyen entre todas las computadoras o si se utilizan servidores de red centralizados. La mayoría de las redes actuales están instaladas bajo un esquema cliente/servidor.

En este proyecto, al utilizar un sistema de monitoreo con Zabbix, se utiliza una relación de red cliente/servidor. En este tipo de configuración, el servidor Zabbix centraliza la recopilación y gestión de datos de monitoreo, mientras que los agentes Zabbix instalados en los dispositivos cliente (como servidores, estaciones de trabajo y otros equipos de red) recopilan y envían datos al servidor Zabbix [8], [9].

2.7. Arquitectura y estandarización de redes

La arquitectura de red representa un sistema compuesto por múltiples elementos, donde cada uno desempeña una función específica e interactúa con los demás. En otras palabras, la arquitectura de la red divide los procesos de cada dispositivo en una serie de subprocesos que deben ejecutarse en cada elemento de la red. Uno de los elementos más importantes de la arquitectura de red es el **protocolo** de comunicación, que puede definirse como un conjunto formal de reglas para la interacción entre nodos de la red [10].

El desarrollo del Modelo de Interconexión de Sistemas Abiertos (OSI, Open System Interconnection) marcó un hito en la estandarización de las redes informáticas. Este modelo fue desarrollado a principios de los años 80 del siglo pasado y recogió toda la experiencia acumulada en aquel momento [11].

Las pilas de protocolos que se utilizan hoy en día se basan generalmente en la arquitectura del modelo OSI; sin embargo, cada conjunto de protocolos presenta características y diferencias específicas [12].

La arquitectura estándar de una red informática también define la distribución de protocolos entre los elementos de la red, como nodos finales (computadoras) y nodos de puerta de enlace (conmutadores y enrutadores). Los nodos de paso admiten solo un subconjunto limitado de funcionalidades de la pila de protocolos; realizan funciones de transporte, retransmitiendo el tráfico de la red entre nodos finales. A su vez, soportan toda la pila de protocolos, ya que necesitan proporcionar servicios de información, como los servicios web.

2.7.1. Protocolo y pila de protocolos

El modelo de capas múltiples en redes presenta características específicas que requieren la interacción de al menos dos entidades para que la comunicación sea posible. Esto implica coordinar el funcionamiento de diferentes niveles de herramientas de red que operan en sistemas separados. Por ejemplo, es necesario definir aspectos como el formato de las señales, el tamaño de los mensajes y los métodos para detectar y corregir errores.

Es necesario llegar a acuerdos en todos los niveles, desde las capas más bajas, que gestionan la transmisión de bits, hasta las más altas, que ofrecen servicios a los usuarios finales.

Un **protocolo** define las reglas para la comunicación entre módulos en diferentes nodos, mientras que una interfaz establece las reglas entre módulos dentro del mismo nodo.

El conjunto de protocolos que permite que los nodos se comuniquen entre sí se denomina *pila de protocolos*. En las capas inferiores, los protocolos suelen depender de una combinación de hardware y software, mientras que en las capas superiores se gestionan exclusivamente por software.

Un módulo o software que utiliza un protocolo se conoce como *entidad de protocolo*. La efectividad del protocolo depende de su diseño lógico y de la forma en que se implementa. Además, la calidad de la red también está relacionada con la claridad de las interfaces entre capas y con la distribución de las funciones entre los distintos protocolos de la pila.

Las entidades de protocolo en la misma capa intercambian mensajes siguiendo las reglas establecidas. Estos mensajes suelen incluir un encabezado que contiene información de control, así como los datos que se transmiten. A medida que los protocolos se vuelven más complejos, también aumentan las funciones que pueden desempeñar.

2.7.2. Modelo OSI

A principios de los años 80, varias organizaciones internacionales de estándares, como la Organización Internacional de Normalización (ISO) y el Sector de Normalización de Telecomunicaciones de la UIT (ITU-T), crearon el Modelo de Interconexión de Sistemas Abiertos (OSI). Este modelo ha sido crucial en el desarrollo de las redes de computadoras, ya que ha servido como un marco de referencia para la comunicación entre sistemas abiertos, es decir, aquellos que pueden intercambiar información con otros sistemas.[10]

El modelo OSI no contiene descripciones de ninguna pila de protocolos en específico; su objetivo es proporcionar una descripción general de las herramientas de interconectividad de redes, razón por la cual se le conoce como modelo de referencia.

El modelo OSI describe siete capas de intercomunicación en redes de conmutación de paquetes. Cada capa tiene un nombre estándar y funciones específicas que debe cumplir.

Estas capas son las siguientes:

1. **Capa física:** Se encarga de la transmisión de bits a través de un medio físico, como un cable, una fibra óptica o una onda electromagnética. Esta capa define las características eléctricas, mecánicas y funcionales del medio, así como los métodos de codificación y sincronización de los bits.

2. **Capa de enlace de datos:** Esta capa se encarga de detectar y corregir los errores que puedan ocurrir en la capa física, así como de controlar el flujo y el acceso al medio. Esta capa también puede dividirse en dos subcapas: la subcapa de control de enlace lógico (LLC), que

proporciona servicios a la capa de red, y la subcapa de control de acceso al medio (MAC), que gestiona el acceso al medio compartido por varios nodos.

3. **Capa de red:** Esta capa gestiona la conmutación y el enrutamiento de paquetes entre nodos no conectados directamente. También asigna direcciones lógicas, fragmenta y reensambla paquetes según sea necesario. Esta capa también puede proporcionar funciones como el control de congestión, el control de errores o la seguridad.

4. **Capa de transporte:** Esta capa se encarga de crear, mantener y finalizar las conexiones lógicas entre los procesos, además de segmentar y reensamblar los datos cuando sea necesario. También puede proporcionar servicios como el control de flujo, la corrección de errores, el reordenamiento de datos y la multiplexación.

5. **Capa de sesión:** Esta capa se encarga de iniciar, mantener y finalizar las sesiones, además de sincronizar y recuperar los datos en caso de fallos. También puede proporcionar servicios como el control del diálogo, la gestión de tokens y la seguridad.

6. **Capa de presentación** Se encarga de asegurar que los datos que se envían y reciben entre las aplicaciones sean compatibles, independientemente de las diferencias en el formato, la codificación o la representación. Para ello, la capa de presentación realiza operaciones como la conversión, la compresión, el cifrado y el descifrado de los datos. La capa de presentación también se encarga de gestionar la sintaxis y la semántica de los mensajes, así como de definir los formatos de intercambio de datos.

7. **Capa de Aplicación:** Esta capa se encarga de proveer servicios específicos necesarios para la comunicación entre usuarios o aplicaciones con otros sistemas. Define reglas y protocolos para acceder a recursos compartidos como archivos, impresoras o bases de datos, y para realizar funciones como correo electrónico, transferencia de archivos o navegación web.

Tabla 2.6: Funciones de las capas del modelo OSI

Capa	**Función**
7. Aplicación	Datos normalizados
6. Presentación	Interpretación de los datos
5. Sesión	Diálogos de control
4. Transporte	Integridad de los mensajes
3. Red	Encaminamiento
2. Enlace de datos	Detección de errores
1. Físico	Conexión de equipos

La representación del Modelo OSI, de mayor a menor nivel, se basa en la jerarquía de abstracción y funcionalidad de las capas. Este enfoque permite una comprensión gradual y estructurada de cómo funciona la comunicación en las redes de computadoras.

Al comenzar desde el nivel más alto (capa de aplicación) y descender hacia el nivel más bajo (capa física), los usuarios y desarrolladores pueden comprender, en primer lugar, los servicios de red ofrecidos a nivel de aplicación, como el correo electrónico y la navegación web. Luego, pueden adentrarse gradualmente en capas más fundamentales que manejan aspectos específicos de la comunicación, como el manejo de errores a nivel de enlace de datos y la transmisión de bits a nivel físico.

Además, esta representación permite una clara separación de responsabilidades entre las capas. Cada capa se enfoca en una tarea específica y no necesita preocuparse por los detalles de las capas superiores o inferiores. Esto facilita la modularidad, la interoperabilidad y el desarrollo de estándares de comunicación. La Tabla 2.6 presenta las principales funciones del modelo OSI, proporcionando una visión clara de cómo cada capa contribuye a la comunicación en red. Este modelo es fundamental para entender y diseñar redes de computadoras, ya que permite una estructura organizada y estandarizada para la transmisión de datos.

El modelo OSI es un modelo teórico que sirve para entender cómo funciona la comunicación entre sistemas abiertos y para facilitar el diseño e implementación de protocolos interoperables. Sin embargo, no todos los protocolos existentes siguen estrictamente este modelo; algunos pueden combinar o simplificar ciertas capas según sus necesidades.

2.8. Modelo del Protocolo de Control de Transmisión/-Protocolo de Internet "TCP/IP"

El modelo TCP/IP, que significa Protocolo de Control de Transmisión/Protocolo de Internet en inglés (Transmission Control Protocol/Internet Protocol), tiene una correspondencia directa con el modelo OSI. La Tabla 2.7 muestra la correspondencia general entre las capas del modelo TCP/IP y el modelo OSI.

Tabla 2.7: Comparación entre el modelo OSI y el modelo TCP/IP

Modelo OSI	Modelo TCP/IP
7. Capa de Aplicación	4. Aplicación
6. Capa de Presentación	
5. Capa de Sesión	
4. Capa de Transporte	3. Transporte
3. Capa de Red	2. Internet
2. Capa de Enlace de Datos	1. Acceso a la Red
1. Capa Física	

Las pilas de protocolos que se utilizan hoy en día se basan generalmente en la arquitectura del modelo OSI; sin embargo, cada conjunto de protocolos presenta características y diferencias específicas [12].

La pila de protocolos TCP/IP, originalmente diseñada para Internet, ofrece numerosas ventajas en la construcción de redes, especialmente en entornos WAN. Su capacidad para fragmentar paquetes es esencial en redes de gran tamaño, y su flexibilidad de direccionamiento la hace idónea para entornos heterogéneos. A pesar de estos desafíos, TCP/IP sigue siendo la pila de protocolos más utilizada en WAN y LAN en la actualidad [13].

2.8.1. Capas del Modelo TCP/IP

TCP/IP cuenta con cuatro capas principales que facilitan la transmisión de datos desde las aplicaciones hasta el nivel físico, permitiendo la comunicación entre dispositivos en redes heterogéneas:

1. **Capa de Aplicación:** Es la más cercana a los usuarios y aplicaciones. Proporciona servicios de red directamente a los programas de aplicación. Incluye protocolos como HTTP para la transferencia de páginas web, SMTP para el correo electrónico y FTP para la transferencia de archivos, entre otros. También maneja la combinación de servicios proporcionados por el sistema a las aplicaciones del usuario.

2. **Capa de Transporte:** La función principal de esta capa es proporcionar servicios de transporte de datos entre aplicaciones. ,

   - **TCP (Protocolo de Control de Transmisión):** Este es el protocolo más utilizado en esta capa. Su función principal es garantizar la entrega de datos, asegurando que lleguen sin errores y en el orden correcto.
   - **UDP (Protocolo de Datagrama de Usuario):** Es un protocolo que proporciona un servicio no orientado a la conexión, donde la entrega de datos no está garantizada,

pero ofrece mayor velocidad.

3. **Capa de Red:** Es responsable de la transferencia de datos entre sistemas finales (hosts) a través de una red. El protocolo más importante en esta capa es el Protocolo de Internet (IP), que enruta los paquetes de datos entre diferentes redes hasta su destino final. El funcionamiento del protocolo IP implica dividir el flujo de datos en fragmentos más pequeños, conocidos como paquetes. Cada paquete incluye una cabecera con las direcciones IP de origen y destino.

4. **Capa de Acceso a la Red:** También conocida como capa de enlace de datos, esta capa se encarga de la comunicación entre dispositivos adyacentes, garantizando que los datos se transmitan correctamente. Incluye tecnologías como Ethernet y Wi-Fi.

2.8.2. IPv4 e IPv6

IPv4

El Protocolo de Internet versión 4 (IPv4), creado en la década de los 70, utiliza direcciones de 32 bits, lo que permite un total teórico de aproximadamente 4.3 mil millones de direcciones únicas. Sin embargo, el aumento acelerado en el número de dispositivos conectados, como smartphones, computadoras y dispositivos del Internet de las Cosas (IoT), ha generado un agotamiento significativo de estas direcciones disponibles.

Para mitigar este problema, se han implementado tecnologías como el Network Address Translation (NAT), que permite a múltiples dispositivos compartir una única dirección IPv4 pública, y la asignación dinámica de direcciones IP mediante servidores DHCP. Estas soluciones han prolongado la vida útil de IPv4, aunque no resuelven la cuestión de fondo.

El agotamiento de las direcciones IPv4 ha sido uno de los principales impulsores del desarrollo y adopción de IPv6, que ofrece un espacio de direcciones significativamente mayor (128 bits), asegurando escalabilidad a largo plazo para la creciente demanda de conectividad global. A pesar de ello, la transición a IPv6 ha sido lenta, en gran parte debido a los costos y la complejidad de migrar infraestructuras establecidas que aún dependen de IPv4.

IPv6

IPv6, propuesto en 1994 en el RFC 1752, emplea direcciones de 128 bits, ampliando las direcciones IP. Sus mejoras incluyen seguridad, rendimiento y manejo de paquetes, como indica el RFC 1827. IPv6 ofrece direccionamiento multicast y anycast, optimiza la distribución de datos y actualiza protocolos como DHCP, DNS, ICMP y TCP para mejorar la interoperabilidad y el rendimiento de la red.

Tabla 2.8: Comparativa de IPv4 con IPv6

Característica	**IPv4**	**IPv6**
Dirección	32 bits (4 bytes)	128 bits (16 bytes)
Formato de dirección	Decimal (192.168.100.1)	Hexadecimal (2001:0db8:85a3:0000:0000:8a2e)
Direcciones totales	Aproximadamente 4.3 mil millones	340 trillones de trillones de trillones
Cabecera	20-60 bytes	40 bytes fijos
Fragmentación	Realizada por hosts y enrutadores	Realizada solo por hosts
Broadcast	Sí	No aplica
Seguridad	Depende de las aplicaciones	IPSec obligatorio
Asignación de direcciones	Manual o DHCP	Autoconfiguración o DHCPv6

La Tabla 2.8 compara IPv4 e IPv6. El RFC 1933 describe la transición de IPv4 a IPv6 mediante el plan SIT (Simple IPo6 Transition), que implica la actualización a IPv6 coexistiendo con IPv4; posteriormente, se realiza el reemplazo completo, facilitando la transición sin interrupciones y minimizando costos. La pila TCP/IP se adapta a tecnologías como Ethernet, Token Ring, PPP y ATM. Aunque no se alinea estrictamente con el modelo OSI, su capa de interfaz conecta diversas tecnologías, mejorando la interoperabilidad.

Interoperabilidad en TCP/IP

En el modelo TCP/IP, la interoperabilidad entre diferentes sistemas y redes se logra mediante conceptos clave como el **flujo**, los **segmentos** y los **datagramas**, que describen las unidades de datos transmitidas entre dispositivos.

- **Flujo:** Representa un conjunto continuo de datos enviados desde una aplicación de origen hacia una aplicación de destino. Este concepto es gestionado por el protocolo TCP, que asegura un transporte confiable mediante la retransmisión de datos perdidos y el control de congestión.
- **Segmentos:** Son las unidades de datos que TCP genera al dividir un flujo en partes manejables para su transmisión. Cada segmento contiene un encabezado con información necesaria para el reensamblado y el control de errores en el destino.
- **Datagramas:** Son las unidades de datos manejadas por el protocolo IP. Cada datagrama incluye un encabezado que contiene información de direccionamiento, así como el segmento

de datos a ser transmitido. Los datagramas pueden llegar desordenados o incluso perderse, ya que IP proporciona un servicio de entrega no confiable.

Estos conceptos trabajan conjuntamente para garantizar la interoperabilidad en las comunicaciones, permitiendo que los datos viajen a través de redes heterogéneas y alcancen su destino de manera eficiente.

2.9. Protocolos de redes

Los protocolos de red, como IP (Internet Protocol), TCP (Transmission Control Protocol), UDP (User Datagram Protocol), HTTP (HyperText Transfer Protocol) y FTP (File Transfer Protocol), establecen cómo los dispositivos se comunican, definiendo el formato, tamaño, orden y control de los datos según la capa OSI (Open Systems Interconnection). Facilitan el intercambio de información entre sistemas heterogéneos, garantizando comunicaciones seguras y confiables [10].

A continuación, se presenta un resumen de los protocolos involucrados en el desarrollo de este proyecto:

- **IP (Internet Protocol):** Este protocolo establece las reglas que permiten la comunicación entre dispositivos conectados a una red, identificándolos mediante una dirección IP única. Existen dos versiones principales: IPv4, que utiliza direcciones de 32 bits (aproximadamente 4.3 mil millones de direcciones), e IPv6, que emplea direcciones de 128 bits (alrededor de 3.4×10^{38} direcciones posibles). IPv6 fue desarrollado para superar las limitaciones de IPv4 ante el creciente número de dispositivos conectados. Aunque ambos protocolos coexisten, IPv6 ofrece ventajas significativas, como una mayor capacidad, mejor seguridad y soporte para la autoconfiguración. IP divide los datos en paquetes, les asigna direcciones de origen y destino, y los envía por la ruta más adecuada. Sin embargo, no garantiza la entrega ni el orden correcto de los paquetes, funciones que complementan otros protocolos como TCP [12].

- **TCP (Transmission Control Protocol):** Este protocolo orientado a la conexión asegura la transmisión confiable de datos entre dispositivos. Utiliza un proceso conocido como "triple apretón de manos"para establecer la comunicación entre el emisor y el receptor. TCP garantiza que los datos lleguen completos y en el orden correcto, proporcionando integridad en las transmisiones. Además, utiliza direcciones IP y puertos para identificar dispositivos y aplicaciones específicas, ofreciendo una abstracción de la red mediante sockets.

- **UDP (User Datagram Protocol):** Este protocolo, a diferencia de TCP, no requiere establecer una conexión previa, lo que lo hace más rápido y eficiente. Sin embargo, no garantiza la entrega ni la integridad de los datos, características que lo hacen ideal para

aplicaciones en tiempo real, como el streaming de vídeo, consultas DNS (Domain Name System) y conexiones VPN (Virtual Private Network). Su simplicidad reduce la sobrecarga en la red, optimizando la velocidad de transmisión.

- **HTTP (HyperText Transfer Protocol):** Este protocolo facilita la transferencia de datos entre navegadores y servidores web, utilizando un modelo de petición-respuesta. HTTP es fundamental para la navegación web, ya que permite la visualización de páginas y recursos en línea. Sin embargo, debido a sus limitaciones de seguridad, ha evolucionado hacia HTTPS (HyperText Transfer Protocol Secure), que incorpora cifrado mediante SSL/TLS (Secure Sockets Layer/Transport Layer Security).

- **FTP (File Transfer Protocol):** Este protocolo permite la transferencia de archivos entre dispositivos conectados a Internet sin restricciones de tamaño. FTP es ampliamente utilizado en la gestión de servidores web, ya que facilita la carga, descarga y organización de archivos. Aunque es un estándar antiguo, sigue siendo funcional; sin embargo, sus limitaciones en materia de seguridad han llevado al desarrollo de alternativas como FTPS (FTP Secure) y SFTP (SSH File Transfer Protocol), que incorporan cifrado y autenticación avanzada.[6].

2.10. Protocolo de administración de redes

Estos protocolos de administración de redes son esenciales para garantizar el correcto funcionamiento y la eficiencia de las redes informáticas. A través de estos protocolos, se puede supervisar, ajustar, administrar y mejorar el rendimiento y la seguridad de la red, así como identificar y solucionar posibles incidencias o errores. Entre los protocolos de administración de redes más utilizados se encuentran SNMP, ICMP, Telnet, SSH y NetFlow [10].

El protocolo ICMP ofrece herramientas para optimizar y asegurar la comunicación entre dispositivos. Una de sus herramientas es la detección de dispositivos activos mediante el envío de mensajes ICMP, ya que es posible verificar si un dispositivo está en línea y responde a las peticiones enviadas, lo cual es esencial tanto para resolver problemas de conectividad como para identificar equipos no autorizados que podrían representar una amenaza a la seguridad de la red [14]. Además, ICMP se utiliza para medir la latencia de la red, proporcionando datos sobre el tiempo que tarda en enviarse la información entre dos puntos; información crítica para localizar y solucionar cuellos de botella que puedan estar afectando el rendimiento de la red [14].

Otra función importante de ICMP es la identificación de rutas de red. Herramientas como traceroute [1] aprovechan este protocolo para trazar la trayectoria que siguen los paquetes de datos,

[1]Es una herramienta de diagnóstico de redes. Esta herramienta permite determinar la ruta que un paquete de datos sigue desde su origen hasta su destino en una red IP

revelando el camino y los nodos por los que pasan. Esto permite comprender mejor el flujo del tráfico y optimizar las rutas según sea necesario [14].

A continuación, se enlistan las versiones del protolo SNMP:

- **SNMP (Simple Network Management Protocol):** Es un estándar para la gestión y el monitoreo de redes de comunicación. Permite a los administradores de red obtener información sobre el estado y el rendimiento de los dispositivos conectados a la red, así como modificar su configuración y enviar instrucciones. El protocolo SNMP se basa en un modelo cliente-servidor, donde el cliente es una aplicación que envía consultas al servidor, que es un agente instalado en el dispositivo gestionado. El agente responde a las consultas del cliente con datos o acciones. SNMP emplea la MIB (Management Information Base), que define los objetos que pueden ser consultados o modificados. El protocolo SNMP incluye varias versiones, siendo la más reciente SNMPv3, que incorpora mecanismos de seguridad y autenticación [12]. Dado que este es el protocolo utilizado para administrar una red, más adelante, en el desarrollo del proyecto, se explicará en detalle su funcionamiento.

  - **SNMPv1:** Es la versión original del protocolo que define la estructura y el formato de los mensajes, así como el protocolo de transporte (UDP, User Datagram Protocol). SNMPv1 solo soporta la autenticación basada en una contraseña comunitaria, la cual se envía en texto plano y puede ser interceptada fácilmente.
  - **SNMPv2:** Es una versión mejorada de SNMPv1, que introduce nuevos tipos de mensajes, como los de respuesta masiva (GetBulk) y los de notificación (Inform). SNMPv2c también mejora el rendimiento y la eficiencia del protocolo, aunque mantiene el mismo mecanismo de seguridad que SNMPv1.
  - **SNMPv3:** Es la versión más reciente y segura del protocolo, que incorpora mecanismos de autenticación, cifrado y control de acceso. SNMPv3 permite a los gestores establecer sesiones seguras con los agentes, protegiendo la confidencialidad, la integridad y la disponibilidad de los datos.

En este proyecto se utiliza la versión 2 del protocolo SNMP, ya que los dispositivos Cisco emulados se configuran con la versión pública, que corresponde a SNMPv2. Esta versión mejora el rendimiento y la eficiencia del protocolo, permitiendo una gestión efectiva de los dispositivos de red. Aunque SNMPv2 no incluye los mecanismos de seguridad avanzados de SNMPv3, sigue siendo ampliamente utilizado debido a su compatibilidad y facilidad de implementación.

2.10.1. Análisis de Paquetes con Wireshark

Wireshark es una herramienta de código abierto utilizada para la captura y análisis de tráfico en redes de comunicación. Permite examinar paquetes en tiempo real, proporcionando detalles

sobre protocolos, direcciones IP, puertos y datos transportados. Su uso es fundamental para la depuración de redes, detección de anomalías y auditoría de seguridad [15].

Wireshark soporta cientos de protocolos de red y permite aplicar filtros avanzados para facilitar la inspección del tráfico. Además, ofrece la posibilidad de reconstruir flujos de comunicación y extraer información específica de los paquetes analizados.

2.10.2. Arquitectura SNMP

El Protocolo Simple de Administración de Red (SNMP, acrónimo de Simple Network Management Protocol), establece un marco estructural que organiza las estaciones de gestión y los elementos de red. Las estaciones de gestión operan aplicaciones encargadas de monitorear y controlar los elementos de red. Estos elementos, que incluyen anfitriones, como computadoras o servidores que actúan como origen y destino de las transferencias de datos, puertas de enlace, que son dispositivos que conectan diferentes redes y permiten la comunicación entre sistemas heterogéneos, y servidores de terminales, están equipados con agentes que ejecutan las operaciones solicitadas por las estaciones. SNMP facilita la transferencia de datos operativos entre las estaciones de gestión y los agentes instalados en los componentes de la red [6].

2.10.3. Especificación del protocolo SNMP

El protocolo de gestión de red es un protocolo de aplicación que permite inspeccionar o modificar las variables de la MIB (Management Information Base), una base de datos virtual utilizada para gestionar dispositivos en una red de comunicaciones [16]. La interacción entre los componentes del protocolo se realiza mediante el intercambio de mensajes. Cada mensaje se representa de manera completa e independiente en un único datagrama UDP (User Datagram Protocol), siguiendo las reglas básicas de codificación de ASN (Autonomous System Number), un identificador único asignado a cada red que participa en el enrutamiento de paquetes de datos en Internet. Dicho número es esencial para el funcionamiento de la red, ya que permite a los enrutadores y otros dispositivos de red determinar la ruta óptima para enviar datos [10]. Un mensaje incluye un identificador de versión, un nombre de comunidad SNMP y una unidad de datos del protocolo (PDU, Protocol Data Unit).

Una entidad de protocolo recibe mensajes en el puerto UDP 161 del host asociado para todos los mensajes, excepto aquellos que contienen notificaciones no solicitadas (es decir, los que no incluyen la Trap-PDU). Las notificaciones no solicitadas deben ser recibidas en el puerto UDP 162 para su procesamiento posterior.

Es obligatorio que todas las implementaciones de SNMP soporten los cinco PDU:

1. **GetRequest-PDU:** Es una solicitud enviada por el administrador para obtener información del agente.

2. **GetNextRequest-PDU:** Se utiliza para solicitar la siguiente variable en el Árbol de Información de Administración (MIB).
3. **GetResponse-PDU:** Es la respuesta del agente que devuelve la información solicitada al administrador.
4. **SetRequest-PDU:** Es una solicitud del administrador para modificar la información en el agente.
5. **Trap-PDU:** Es una notificación no confirmada enviada por el agente para informar al administrador sobre eventos de red.

2.11. Gestión de redes

Es el proceso de supervisar y controlar los elementos de una red. Los elementos de la red incluyen hosts, puertas de enlace, servidores de terminales y otros dispositivos similares. Cada dispositivo cuenta con un agente de gestión que se encarga de realizar las funciones de administración de red solicitadas por las estaciones de operaciones de red. Estas funciones pueden incluir la recopilación de estadísticas, la configuración de parámetros de red y la detección y notificación de condiciones de error.

2.12. Identificación de fallos en interfaces

La implementación de herramientas de gestión permite identificar problemas en las interfaces de red antes de que afecten la operación. Por ejemplo, un enrutador puede notificar automáticamente al administrador sobre una falla en una interfaz, evitando que los usuarios reporten problemas al Centro de Operaciones de Red (NOC, Network Operations Center). Un NOC es una ubicación centralizada donde se monitorean y gestionan las redes de computadoras, telecomunicaciones o satélites las 24 horas del día, los 7 días de la semana. Los NOC son esenciales para mantener la disponibilidad y el rendimiento de la red, ya que permiten a los equipos de TI detectar y resolver problemas rápidamente, optimizar el rendimiento de la red y garantizar una conectividad continua [17]. Un administrador puede analizar el tráfico de la red y detectar anomalías, como un aumento en los errores de verificación de suma en los paquetes enviados. Este indicador puede señalar un fallo inminente, lo que permite tomar medidas preventivas, como reemplazar una tarjeta de interfaz antes de que ocurra una interrupción.

2.13. Seguimiento a asignación de recursos

El monitoreo continuo de patrones de tráfico puede ayudar a los administradores a optimizar el uso de los recursos de la red. Por ejemplo, reubicar servidores entre segmentos de una LAN

podría reducir significativamente el volumen de tráfico que atraviesa múltiples segmentos. Además, supervisar la congestión en los enlaces permite anticipar la necesidad de una mayor capacidad de ancho de banda antes de que dicha congestión afecte el rendimiento.

2.14. Monitoreo de hosts

El administrador de la red puede realizar revisiones periódicas para asegurarse de que todos los hosts de la red están funcionando y en estado operativo. Una vez más, el administrador de la red puede abordar proactivamente un problema, como un host inactivo, antes de que el usuario lo reporte.

2.15. Identificación de intrusos

Un administrador de red podría ser notificado cuando el tráfico de la red provenga o esté dirigido a una fuente sospechosa, como un host o un número de puerto específico. De manera similar, un administrador de red podría identificar (y, en muchos casos, filtrar) la presencia de ciertos tipos de tráfico, como paquetes con enrutamiento de origen o un gran número de paquetes SYN (Synchronize), que son una parte integral del protocolo TCP (Transmission Control Protocol), utilizado para establecer conexiones entre un cliente y un servidor en una red. Estos paquetes SYN dirigidos a un host específico son indicativos de ciertos tipos de ataques de seguridad. Las áreas proporcionan un marco estructurado para la gestión de la red, lo que permite a los administradores de red abordar eficazmente los desafíos de mantener una red funcionando de manera eficiente y segura. [18]

2.16. Administración de una red

La administración de una red consiste en el conjunto de actividades, métodos, procedimientos y herramientas que se utilizan para operar, controlar y mantener una red de comunicaciones.

Los objetivos principales de la administración de una red son garantizar la disponibilidad, el rendimiento, la seguridad y la calidad del servicio de los recursos y servicios de la red, así como optimizar el uso y el costo de los mismos. Para lograr estos objetivos, se requiere realizar las siguientes funciones básicas de la administración de una red, las cuales se enlistan en la tabla 2.9.

Tabla 2.9: Funciones de la Gestión de Redes

Función	**Descripción**
Configuración	Establecer parámetros y políticas para el funcionamiento de la red y sus componentes.
Monitoreo	Observar el estado y comportamiento de la red, incluyendo tráfico, carga y errores.
Diagnóstico	Identificar y analizar problemas como pérdida de paquetes o ataques.
Solución de Problemas	Aplicar acciones correctivas, como reconfiguración o reinicio, para resolver problemas detectados.
Contabilidad	Registrar y reportar el uso de recursos y servicios por parte de usuarios y aplicaciones.
Auditoría	Verificar el cumplimiento de normas y estándares de seguridad y protocolos.
Planificación	Diseñar cambios y mejoras para adaptar la red a las necesidades presentes y futuras.

2.17. GNS3

El simulador de redes GNS3 (Graphical Network Simulator-3) es una herramienta de software diseñada para crear y emular redes de computadoras de manera virtual. Permite a los usuarios replicar la infraestructura de redes complejas sin la necesidad de hardware físico, lo cual facilita la experimentación y el aprendizaje en la configuración de redes. Esta plataforma es ampliamente utilizada tanto en el ámbito educativo como profesional, proporcionando un entorno seguro y accesible para probar configuraciones y realizar prácticas de redes.

Una de las principales características de GNS3 es su capacidad para emular dispositivos de red de diversas marcas, como enrutadores, conmutadores y cortafuegos, lo que permite la creación de topologías de redes avanzadas. Además, la interfaz gráfica de usuario de GNS3 facilita la creación de estas topologías mediante una sencilla operación de arrastrar y soltar, lo que reduce la barrera de entrada para los usuarios menos experimentados. Otra característica importante de GNS3 es su compatibilidad con máquinas virtuales, lo que amplía las posibilidades de pruebas y simulaciones al integrar diferentes sistemas operativos dentro de la misma red emulada [19].

.

2.18. Servidor de monitoreo

Un servidor de monitoreo es un dispositivo encargado de supervisar el estado y el rendimiento de otros equipos o sistemas informáticos. Su función es alertar a los administradores o respon-

sables de la red en caso de que se produzca algún fallo, anomalía o amenaza que pueda afectar el funcionamiento normal de los servicios o aplicaciones. Un servidor de monitoreo puede utilizar diferentes protocolos y herramientas para recoger y analizar los datos de los elementos monitorizados, como, por ejemplo, SNMP, ICMP, WMI y NetFlow, entre otros. Además, puede ofrecer una interfaz gráfica que facilite la visualización y el control de los parámetros más relevantes, así como la generación de informes y estadísticas extraídas de archivos de registro (logs).

2.19. Zabbix

Zabbix es una solución de monitoreo distribuido de código abierto desarrollada por Alexei Vladishev [20]. Zabbix monitorea parámetros clave de redes, servidores, aplicaciones, servicios y más, ofreciendo una amplia gama de funcionalidades para garantizar la salud y el rendimiento de los sistemas de TI.

Las funcionalidades principales de Zabbix incluyen la recopilación de datos en tiempo real, la verificación de disponibilidad y rendimiento, soporte para protocolos como SNMP, IPMI[2] y JMX[3]. Zabbix destaca por su flexibilidad en las notificaciones, ya que permite configurar alertas por correo electrónico para prácticamente cualquier evento, lo que facilita una respuesta rápida ante cualquier problema que pueda surgir en los servidores [20]. Además, Zabbix cuenta con sólidas capacidades de generación de informes y visualización de datos, lo que lo convierte en una herramienta valiosa para la planificación de la capacidad y la toma de decisiones informadas [20].

Arquitectura

Zabbix está compuesto por varios componentes de software principales.

- **Servidor:** El servidor de Zabbix funciona como el núcleo central al que los agentes envían información sobre disponibilidad, integridad y estadísticas. Actúa como el depósito principal donde se almacenan todas las configuraciones, así como los datos estadísticos y operativos [20].
- **Almacenamiento en base de datos:** Toda la información de configuración y los datos recopilados por Zabbix se guardan en una base de datos [20].
- **Interfaz web:** Zabbix proporciona una interfaz basada en la web para acceder fácilmente desde cualquier dispositivo [20].

[2]IPMI: (Intelligent Platform Management Interface) es un estándar de gestión remota de hardware que permite a los administradores monitorear, administrar y controlar servidores y otros dispositivos de hardware de manera remota [21].

[3]JMX: (Java Management Extensions) es una tecnología de gestión y monitoreo utilizada en entornos de desarrollo Java que permite la supervisión de recursos como memoria, CPU y tiempo de ejecución, así como la gestión remota de aplicaciones [22].

- **Proxy:** El proxy de Zabbix puede recopilar datos de rendimiento y disponibilidad en representación del servidor Zabbix. Aunque su uso es opcional, puede ser muy útil para distribuir la carga en un solo servidor Zabbix [20].
- **Agente:** Los agentes de Zabbix se despliegan en los objetivos de monitoreo para vigilar activamente los recursos y aplicaciones locales, enviando los datos recopilados al servidor Zabbix. Hay dos tipos de agentes disponibles: el agente Zabbix (ligero, multiplataforma y escrito en C) y el agente Zabbix 2 (muy flexible, fácilmente ampliable con complementos y escrito en Go) [20].
- **Flujo de Datos:** En Zabbix, comprender el flujo de datos es esencial. Para recolectar información, primero se debe configurar un host, y para establecer un disparador, es indispensable tener un elemento creado. Por ejemplo, si se desea recibir una alerta sobre una alta carga de CPU en un servidor específico, primero debes crear una entrada de host para ese servidor, seguida de un elemento para monitorear la CPU, un disparador que se active ante una carga alta y finalmente una acción para notificar por correo electrónico. Aunque este procedimiento puede parecer complicado, el uso de plantillas lo simplifica. Esta estructura permite una configuración altamente adaptable [20].

2.20. Grafana

Grafana es una plataforma de observabilidad y monitoreo de código abierto desarrollada por Grafana Labs. Grafana facilita la consulta, visualización, alerta y comprensión de métricas, independientemente de dónde se encuentren almacenadas. Es ampliamente utilizada para crear, explorar y compartir dashboards visuales que facilitan una cultura basada en datos [23].

Las funcionalidades principales de Grafana incluyen la capacidad de recopilar datos en tiempo real, verificar la disponibilidad y el rendimiento, y soportar una amplia gama de fuentes de datos como Prometheus, Graphite, InfluxDB, Elasticsearch y muchos más. Grafana destaca por su flexibilidad en la creación de alertas y notificaciones, permitiendo configurar alertas basadas en correo electrónico, Slack y otros canales para cualquier evento, lo que facilita una respuesta rápida ante cualquier problema que pueda surgir en los sistemas monitoreados [24].

Además, Grafana cuenta con sólidas capacidades de generación de informes y visualización de datos, lo que la convierte en una herramienta valiosa para la planificación de la capacidad y la toma de decisiones informadas. Grafana también soporta la integración con herramientas de gestión de incidentes y automatización, lo que mejora la eficiencia operativa y la resolución de problemas [23].

2.20.1. Arquitectura

Grafana está compuesto por varios componentes principales de software:

- **Servidor de Grafana:** Actúa como el núcleo del sistema, gestionando la autenticación de usuarios, la configuración de dashboards y la comunicación con las fuentes de datos.
- **Fuentes de datos:** Grafana soporta una amplia variedad de fuentes de datos, lo que permite la integración con múltiples sistemas de monitoreo y bases de datos.
- **Dashboards:** Los dashboards en Grafana son altamente personalizables y permiten la visualización de datos en tiempo real mediante gráficos, tablas y otros widgets.
- **Alertas:** Grafana permite configurar alertas basadas en reglas definidas por el usuario, enviando notificaciones a través de diversos canales cuando se cumplen ciertas condiciones.

Grafana utiliza una arquitectura basada en microservicios, lo que permite una escalabilidad horizontal y una alta disponibilidad. Los componentes de Grafana pueden ejecutarse de manera independiente y en paralelo, lo que facilita la gestión y el despliegue en entornos distribuidos [25].

2.21. Raspberry Pi

La Raspberry Pi utilizada en este proyecto es el modelo 3, un dispositivo de bajo costo y tamaño compacto, ideal para aplicaciones de monitoreo en redes. Este dispositivo se caracteriza por su capacidad de ser configurado de forma eficiente en proyectos de monitoreo continuo como el que se presenta en este proyecto [26].

Entre las principales características del modelo Raspberry Pi 3 se encuentran:

- **Procesador:** Quad-core ARM Cortex-A53 a 1.2 GHz, que ofrece un rendimiento adecuado para realizar tareas múltiples de monitoreo, gestión y administración de redes de manera eficiente.
- **Memoria RAM:** 1 GB LPDDR2, suficiente para la ejecución fluida de servicios como Zabbix, junto con otros servicios de monitoreo requeridos para el análisis de redes.
- **Conectividad inalámbrica:** Wi-Fi Dual Band (2.4GHz y 5GHz) IEEE 802.11 b/g/n/ac, y Bluetooth 4.2 BLE, lo que permite flexibilidad en las conexiones, tanto alámbricas como inalámbricas.
- **Conectividad por cable:** Gigabit Ethernet (300 Mbps máximo teórico), ideal para entornos de red que requieren conexiones de alta velocidad.
- **Almacenamiento:** Utiliza una tarjeta microSD como almacenamiento primario para el sistema operativo y los datos necesarios para el proyecto. En este caso, se utilizó una tarjeta de 64 GB.

- **Consumo energético:** El bajo consumo de energía de la Raspberry Pi la convierte en una opción adecuada para proyectos de monitoreo a largo plazo, como en este estudio, donde se requiere monitoreo continuo y confiable.

Además de estas características, la Raspberry Pi modelo 3 ofrece los siguientes puertos y capacidades de expansión, lo que permite integrarse fácilmente con otros dispositivos y periféricos:

- **Puertos:**
  - 4 puertos USB 2.0 para conectar dispositivos adicionales, como teclados, ratones o adaptadores USB.
  - 1 puerto HDMI que permite la conexión a pantallas externas para la visualización local.
  - 1 puerto MicroUSB para alimentación, que simplifica la alimentación del dispositivo mediante adaptadores comunes.
  - 1 puerto RJ45 para Ethernet, proporcionando una conexión física estable y confiable en entornos de red donde se requiere una conexión por cable.
  - 1 toma de auriculares/vídeo compuesto, que ofrece opciones para conexiones audiovisuales alternativas.
  - 1 puerto GPIO de 40 pines que facilita la integración con otros dispositivos electrónicos o sensores.
  - 1 puerto CSI para cámara Raspberry Pi, útil para aplicaciones de visión o captura de imágenes.
  - 1 puerto DSI para pantalla táctil, que permite la conectividad con pantallas táctiles compatibles para una mejor interacción.

La Raspberry Pi, con sus características de bajo consumo, conectividad avanzada y flexibilidad en la expansión, se convierte en una plataforma perfecta para implementar una solución de monitoreo como Zabbix en este proyecto.

2.22. Isolation Forest

Isolation Forest es un algoritmo de machine learning creado específicamente para detectar anomalías en grandes conjuntos de datos. Funciona construyendo múltiples árboles de decisión de forma aleatoria, donde las observaciones anómalas se aíslan con menos particiones que los datos normales. Este enfoque permite detectar de manera eficiente valores atípicos sin necesidad de un conjunto de datos previamente etiquetado, lo que lo convierte en una opción versátil en entornos de monitoreo de sistemas y seguridad informática.

A diferencia de otros métodos de detección de anomalías, Isolation Forest no se enfoca en modelar la distribución normal de los datos, sino en identificar aquellas instancias que requieren un menor número de particiones para su aislamiento. Este principio de diseño le otorga una ventaja en términos de eficiencia computacional, permitiendo analizar grandes volúmenes de información en tiempo real con un bajo costo computacional.

En el contexto de monitoreo de servidores y redes, Isolation Forest se ha utilizado con éxito para identificar comportamientos inusuales en métricas de rendimiento, como el uso de CPU y memoria. Su integración con plataformas como Zabbix facilita la detección temprana de eventos anómalos, mejorando la capacidad de respuesta ante posibles fallos en la infraestructura de TI [27].

En el marco teórico, se destaca la importancia de comprender los fundamentos de las redes de transmisión de información, su clasificación según tecnología, alcance y topologías, así como la relevancia de los dispositivos de red y los protocolos que las sustentan.

La selección de software libre, como Zabbix para monitoreo y Grafana para visualización de datos, permite supervisar de manera precisa y analizar en tiempo real los recursos y el estado de la red. Al integrar modelos estandarizados como OSI y TCP/IP, así como protocolos esenciales como SNMP, se facilita la comunicación entre los componentes, asegurando una administración centralizada y eficiente.

En la administración de una LAN, identificar fallos, gestionar recursos y detectar intrusiones son tareas importantes que garantizan la seguridad y estabilidad de la red. Esto, complementado con herramientas de visualización y monitoreo, fortalece la capacidad de responder rápidamente ante problemas.

En conclusión, el uso de software libre no solo democratiza el acceso a herramientas avanzadas de redes, sino que también fomenta la independencia tecnológica y la optimización de recursos.

Capítulo 3

Desarrollo

El desarrollo de este proyecto se basa en la emulación de la LAN de los laboratorios de Enseñanza de la Ciencia (LACECI) y de Matemáticas (LAMAT), ubicados en el edificio C del plantel San Lorenzo Tezonco de la Universidad Autónoma de la Ciudad de México, con el objetivo de evaluar y probar la solución de monitoreo Zabbix. En la fase inicial, se realiza la emulación de dispositivos de red, como conmutadores y enrutadores Cisco, lo que permite probar el protocolo SNMP en un entorno virtualizado. Sin embargo, una de las limitaciones de esta fase es que en el laboratorio físico no se dispone de los mismos dispositivos de red que se simulan, lo que impide realizar pruebas del protocolo SNMP en la LAN real. La segunda parte del proyecto consiste en la implementación de Zabbix, que se instala inicialmente en una máquina virtual con Ubuntu Server, una distribución de Linux adecuada para entornos de servidor. Finalmente, para llevar esta solución al laboratorio físico, se implementa Zabbix en una Raspberry Pi utilizando el sistema operativo Raspbian, lo que permitirá ofrecer una solución de monitoreo accesible y eficiente en el entorno real del laboratorio.

3.1. Emulación de la Red en GNS3

La emulación de la LAN presente en los laboratorios LACECI y LAMAT se realizó utilizando GNS3. Esta emulación incluye los siguientes elementos de red:

- 1 Enrutador Cisco 7200
- 2 Conmutadores multicapa Cisco C3725
- 1 Servidor Windows
- 1 Servidor Ubuntu
- 15 Estaciones de trabajo

Los servidores se emulan mediante máquinas virtuales dentro de la red creada en GNS3. La topología de la emulación es híbrida, combinando elementos de las topologías en *estrella* y *árbol*. En este diseño, las estaciones de trabajo están conectadas a un conmutador central, caracterizando la parte de la red como una topología en estrella. Sin embargo, los conmutadores están interconectados de manera que forman una estructura que recuerda a una topología en árbol. Aunque una topología en árbol tradicional incluye una jerarquía clara de dispositivos, en este caso la interconexión de los conmutadores no sigue una jerarquía estricta, ya que todos los conmutadores están distribuidos sin un nivel centralizado de control. Este diseño híbrido optimiza la conectividad y la gestión del tráfico de red, aprovechando las ventajas de ambas topologías: la facilidad de administración de la estrella y la expansión escalable de la estructura en árbol.

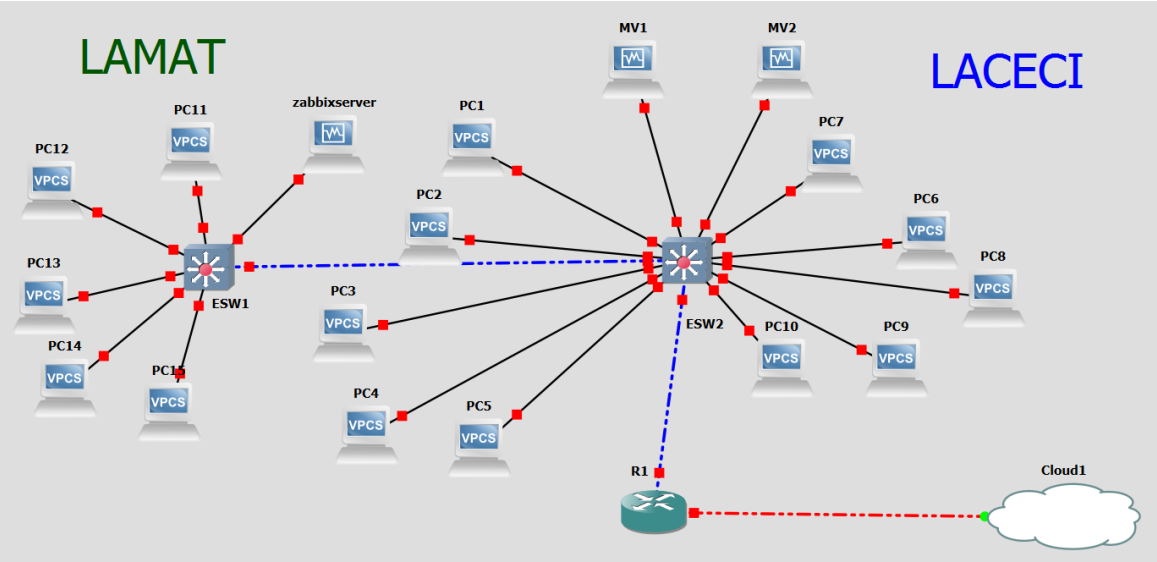


Figura 3.1: Topología implementada en GNS3

El conmutador 1, ubicado en LAMAT, conecta cinco ordenadores con el conmutador 2, ubicado en LACECI, al que están conectados 12 ordenadores. Al primer conmutador también se conecta la máquina virtual *zabbixserver*, que ejecuta el servidor de monitoreo Zabbix con el sistema operativo Ubuntu Server.

La máquina virtual 1, que ejecuta el sistema operativo Windows Server 2019, se utiliza para obtener datos de monitoreo mediante el uso del agente Zabbix y para visualizar el *front-end*[1] del servidor de monitoreo.

La configuración personalizada de Zabbix y su integración con el *dashboard*[2] de Grafana se realiza desde la máquina virtual 2 con sistema operativo Kali Linux. Este entorno permite realizar

[1]El *front-end* se refiere a la interfaz de usuario en informática, sirviendo como herramienta para interactuar y monitorear sistemas. Funciona como una aplicación o panel de control que permite gestionar el monitoreo, configurar alertas y supervisar el rendimiento y estado de las máquinas virtuales.

[2]Un *dashboard* es una representación visual de datos e indicadores clave de rendimiento que permite a los usuarios monitorear y analizar información de manera rápida y eficiente.

pruebas de comunicación con el enrutador, el conmutador y otros elementos que forman parte de la simulación de la red LAN. De esta manera, es posible evaluar la eficacia de las alertas y monitoreos configurados en Zabbix, asegurando que la red LAN esté bajo constante supervisión para detectar y resolver cualquier incidente.

En la maquina virtual 2 se ejecuta el agente de monitoreo de Zabbix, que permite el monitoreo constante del host en el que opera, proporcionando datos en tiempo real sobre su estado y rendimiento.

El enrutador Cisco, modelo 7200, se emula utilizando una imagen IOS, la cual se carga y configura en el software GNS3.

```
c7200-advipservicesk9-mz.152-4.S5.image
```

En este enrutador, se configura una dirección IP estática y se implementa NAT en la interfaz Gigabit Ethernet 1/0 para permitir que la red emulada tenga acceso a Internet a través de la máquina anfitriona, funcionando como puerta de enlace en la red LAN. Además, se habilita el envío de paquetes mediante SNMP para permitir su monitoreo a través de este protocolo con Zabbix.La configuración detallada de este enrutador se presenta en la primera parte del apéndice de este proyecto.

La efectividad de esta conexión se verifica mediante la ejecución de la instrucción `ping` desde la máquina virtual 1 hacia la dirección IP de Google, 8.8.8.8. Esta prueba confirma la correcta configuración y funcionamiento de la conexión a Internet desde la máquina virtual emulada.

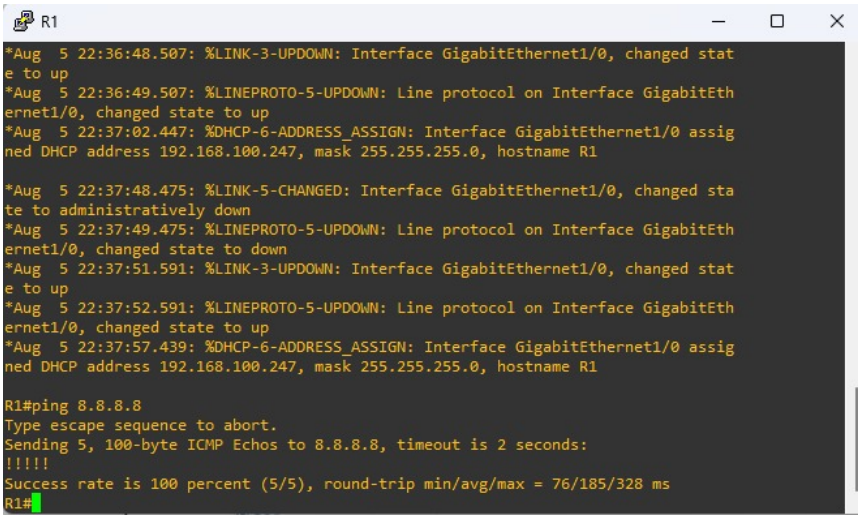


Figura 3.2: Ejecución de la instrucción ping a la dirección IP 8.8.8.8

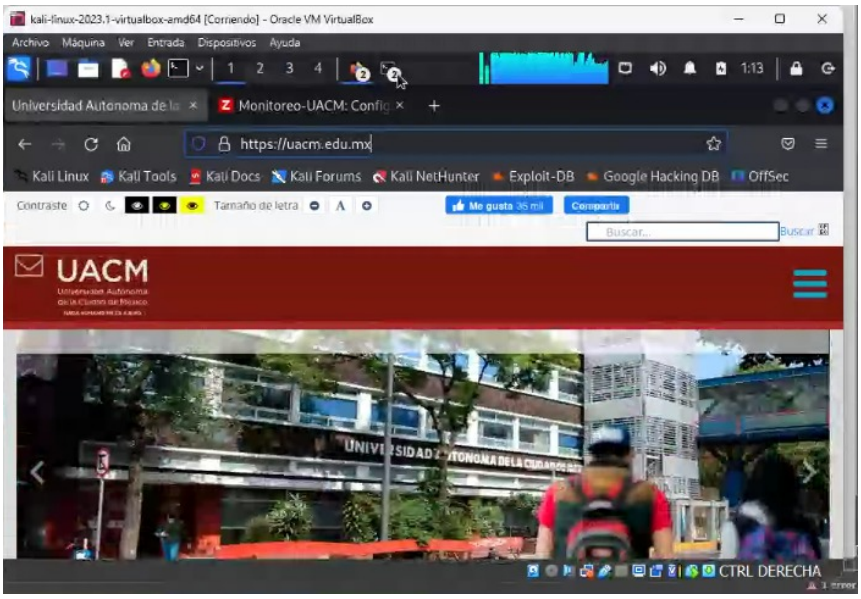


Figura 3.3: Funcionamiento de la conexión a Internet desde la máquina virtual

La elección del enrutador Cisco 7200 y de los conmutadores multicapa Cisco C3725, se debe a su capacidad para admitir el protocolo SNMP. Cabe destacar que esta familia de enrutadores se seleccionó específicamente por su compatibilidad con SNMP. Sin embargo, existen otros modelos y series que también pueden configurarse y utilizarse en entornos de simulación como GNS3.

3.1.1. Direccionamiento IP utilizado

La red LAN emulada se configura para que opere bajo el segmento de direcciones IPv4: 192.168.1.0/24, donde se asignan las siguientes IPs mostradas en la tabla 3.1

Tabla 3.1: Tabla de Dispositivos y sus IPs

Dispositivo	**IP**
Enrutador 1 (R1)	192.168.1.1
Conmutador 1 (ESW1)	192.168.1.2
Conmutador 2 (ESW2)	192.168.1.12
Máquina virtual (Zabbix)	192.168.1.30
Máquina virtual 1 (Windows Server)	192.168.1.8
Máquina virtual 2 (Kali Linux)	192.168.1.4

En este caso, se asignan direcciones IP a los conmutadores **Conmutador 1 (ESW1)** con la IP 192.168.1.2 y **Conmutador 2 (ESW2)** con la IP 192.168.1.30, debido a que son dispositivos multicapa. Los conmutadores multicapa, además de realizar funciones básicas de conmutación en

la Capa 2 del modelo OSI, tienen la capacidad de operar en la Capa 3, lo que les permite llevar a cabo tareas como el enrutamiento entre VLANs y la asignación de direcciones IP para su gestión.

La configuración de direcciones IP en los conmutadores es necesaria para que puedan interactuar con otros dispositivos en la red y ser administrados de forma remota a través de protocolos como SSH o SNMP.

Mientras que a los hosts se les asignaron las direcciones IPs de la tabla 3.2.

Tabla 3.2: Tabla de PCs y sus IPs

Tabla de PCs y sus IPs	
PC	**IP**
1	192.168.1.3
2	192.168.1.5
3	192.168.1.6
4	192.168.1.7
5	192.168.1.9
6	192.168.1.10
7	192.168.1.11
8	192.168.1.13
9	192.168.1.14
10	192.168.1.15
11	192.168.1.16
12	192.168.1.17
13	192.168.1.18
14	192.168.1.19
15	192.168.1.20

Tras configurar la red y asignar direcciones IP a los servidores correspondientes, es de vital importancia verificar la conectividad entre ellos. Para ello, se utiliza el protocolo ICMP empleando la instrucción ping.

Esto verifica la conectividad entre los dispositivos, asegurando el correcto funcionamiento del monitoreo y la red.

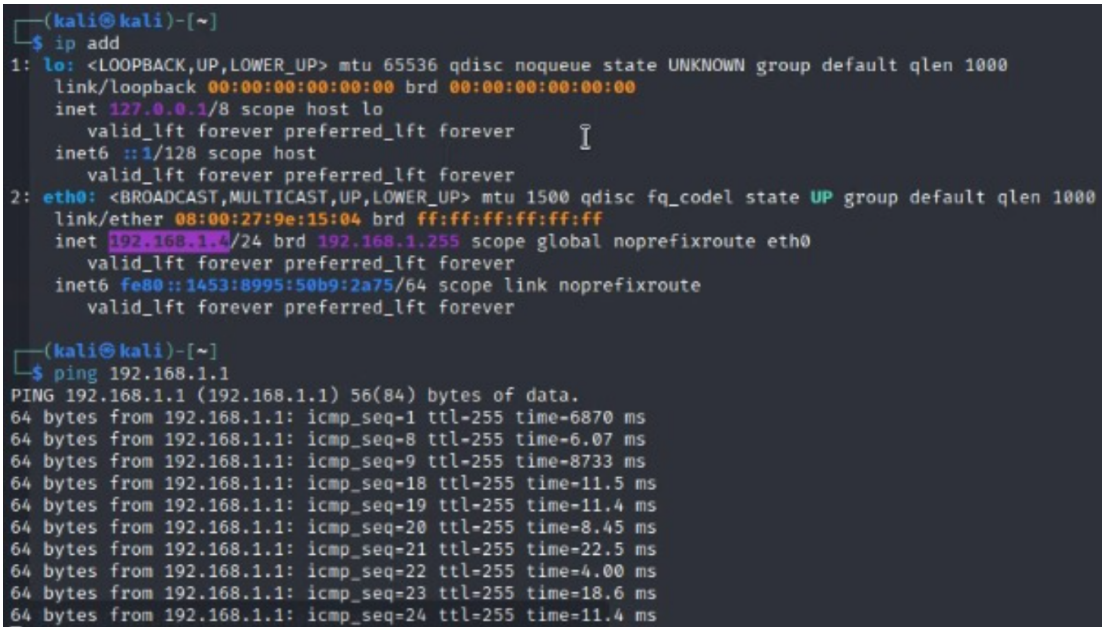


Figura 3.4: Ping de servidor 3 al enrutador 1

El conmutador 1, que actúa como el punto central de la red y se encarga de unir las estaciones de trabajo entre ambos laboratorios, tiene asignada la dirección IP 192.168.1.2 y se conecta al enrutador a través del puerto FastEthernet f1/0. El enrutador, con la dirección IP 192.168.1.1, actúa como la puerta de enlace predeterminada. Esta configuración estratégica no solo mejora la comunicación entre los servidores y el acceso a recursos compartidos, sino que también incrementa la seguridad de la red al centralizar el flujo de tráfico mediante puntos de control definidos.

La red local está configurada para asegurar una comunicación fluida entre el servidor de monitoreo y todos los dispositivos de la red.

3.2. Implementación y configuración del servidor de monitoreo

3.2.1. Requisitos de instalación del servidor Zabbix

Para implementar Zabbix como servidor de monitoreo, se requieren los siguientes recursos mínimos:

- **Memoria RAM:** Al menos 128 MB de memoria física para un funcionamiento adecuado. Para asegurar un desempeño óptimo y escalable, se recomienda disponer de una cantidad mayor de memoria RAM.
- **Espacio en disco:** El servidor Zabbix requiere al menos 256 MB de espacio libre en disco para su instalación y operación. La cantidad exacta de espacio en disco adicional dependerá

del número de dispositivos que se deseen monitorear y de la configuración específica del monitoreo.

- **Sistema operativo:** Zabbix es compatible con varios sistemas operativos basados en Linux, incluidos Ubuntu, CentOS y Debian, entre otros.

La instalación y configuración del servidor de monitoreo Zabbix se realiza en una máquina virtual, la cual se crea utilizando VirtualBox, que permite ejecutar varios sistemas operativos en un solo equipo físico. La máquina virtual se crea con las siguientes características:

- 2048 MB de memoria RAM
- 1 CPU virtual
- 100 GB de memoria ROM
- Sistema operativo Ubuntu Server, versión 22.04 LTS.

En la máquina virtual se configuran el servidor de monitoreo Zabbix, su frontend y el agente, utilizando los siguientes recursos:

Tabla 3.3: Lista de Componentes y Tecnologías

Componente	**Tecnología**
Base de datos	PostgreSQL
Servidor web	Nginx
Versión de PHP	8.1

La figura 3.5 muestra la arquitectura implementada para el servidor de monitoreo de redes Zabbix.

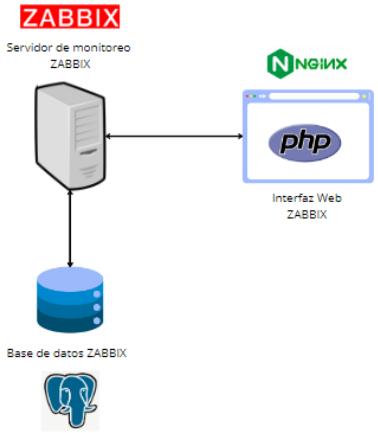


Figura 3.5: Arquitectura Zabbix

Para la configuración del servidor, se opta por utilizar Nginx como servidor web y PostgreSQL como sistema de gestión de bases de datos. Esta elección se basa en varias razones. En primer lugar, Nginx es conocido por su alto rendimiento y capacidad para manejar gran cantidad de conexiones concurrentes con un uso eficiente de recursos, lo que lo hace ideal para aplicaciones modernas y de alta demanda. En cuanto a PostgreSQL, destaca por ser una base de datos robusta, compatible con transacciones complejas y con un fuerte enfoque en la integridad de los datos.

Por otra parte, se decidió no utilizar Apache y MySQL, ya que, aunque son opciones ampliamente usadas, presentan algunas diferencias clave. Apache, aunque versátil, tiende a consumir más recursos bajo alta carga en comparación con Nginx. Por su parte, MySQL, aunque rápido en muchas implementaciones, puede no ser tan adecuado como PostgreSQL en entornos que requieren operaciones transaccionales complejas y una mayor extensibilidad.[28]

Además, la elección de Nginx y PostgreSQL busca diferenciar este proyecto al emplear tecnologías que son menos comunes en proyectos similares, que generalmente optan por Apache y MySQL.

3.2.2. Instalación y Configuración de Nginx y PHP

Zabbix requiere un servidor web y soporte para PHP para proporcionar su interfaz de usuario y procesar las solicitudes del frontend. En este caso, utilizamos Nginx como servidor web y PHP como lenguaje de programación. Para realizar la instalación, se ejecutan las siguientes instrucciones:

- sudo apt install nginx
- sudo apt install php
- sudo apt install php8.1.2 php8.1.2-mbstring php8.1.2-xml php8.1.2-bcmath php8.1.2-gd php8.1.2-common php8.1.2-zip php8.1.2-pgsql

PHP 8.1.2 se selecciona por ser compatible con las versiones más recientes de Zabbix, ofreciendo mejoras en el rendimiento y la seguridad.

La Figura 3.13 ilustra el proceso de instalación de PHP. Además, se verifica la versión de PHP instalada para asegurar la compatibilidad:

La figura 3.7 muestra la versión actual de PHP instalada, que en este caso es la versión 8.1.2:

Tras instalar los componentes necesarios, se configura Nginx para garantizar la integración adecuada con PHP. Esto incluye la configuración de Nginx para habilitar el procesamiento de contenido PHP mediante PHP-FPM, asegurando una integración eficiente.

El servidor Nginx gestiona las solicitudes HTTP y entrega la interfaz web, mientras que PHP procesa el código backend, llevando a cabo las operaciones y comunicándose con la base de datos para gestionar la información obtenida del monitoreo.

```
iseo@iseo:/etc/zabbix$ sudo apt install php
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
  fontconfig-config fonts-dejavu fonts-dejavu-core fonts-dejavu-extra libdeflate0 libfontconfig1 libgd3 libjbig0
  libjpeg-turbo8 libjpeg8 libnginx-mod-http-geoip2 libnginx-mod-http-image-filter libnginx-mod-http-xslt-filter
  libnginx-mod-mail libnginx-mod-stream libnginx-mod-stream-geoip2 libonig5 libtiff5 libwebp7 libxpm4 nginx
  nginx-common nginx-core
Utilice «sudo apt autoremove» para eliminarlos.
Se instalarán los siguientes paquetes adicionales:
  libapache2-mod-php8.1 php8.1 php8.1-cli php8.1-common php8.1-opcache php8.1-readline
Paquetes sugeridos:
  php-pear
Se instalarán los siguientes paquetes NUEVOS:
  libapache2-mod-php8.1 php php8.1 php8.1-cli php8.1-common php8.1-opcache php8.1-readline
0 actualizados, 7 nuevos se instalarán, 0 para eliminar y 255 no actualizados.
```

Figura 3.6: Proceso de instalación de PHP

```
iseo@iseo:/etc/zabbix$ php -v
PHP 8.1.2-1ubuntu2.17 (cli) (built: May  1 2024 10:10:07) (NTS)
Copyright (c) The PHP Group
Zend Engine v4.1.2, Copyright (c) Zend Technologies
    with Zend OPcache v8.1.2-1ubuntu2.17, Copyright (c), by Zend Technologies
iseo@iseo:/etc/zabbix$
```

Figura 3.7: Verificación de la versión PHP 8.1.2

3.2.3. Configuración de NGINX

La figura 3.8 ilustra el proceso de configuración de NGINX para Zabbix en este apartado es necesario se configurar la dirección IP y el puerto en los cuales Zabbix atiende y responde a las peticiones a través de su interfaz web.

```
iseo@zabbixserver: /etc/php/
 GNU nano 6.2        /etc/nginx/nginx.conf
events {
        worker_connections 768;
        # multi_accept on;
}

http {

        ##
        # Basic Settings
        ##

        sendfile on;
        tcp_nopush on;
        types_hash_max_size 2048;
        charset utf-8;
        # server_tokens off;

        # server_names_hash_bucket_size 64;
        # server_name_in_redirect off;

        include /etc/nginx/mime.types;
        default_type application/octet-stream;

^G Help      ^O Write Out  ^W Where Is   ^K Cut
^X Exit      ^R Read File  ^\ Replace    ^U Paste
```

Figura 3.8: Configuración de nginx para Zabbix

La figura 3.8 muestra un fragmento de la configuración en donde se configura el charset UTF-

8, que significa 'Formato de Transformación Unicode de 8 bits'. Configurar el charset UTF-8 en NGINX garantiza que el servidor interprete y envíe los caracteres correctamente a los navegadores y dispositivos, evitando así problemas de interpretación o la aparición de caracteres extraños. Este estándar es parte del conjunto más amplio de Unicode [3].

3.2.4. Instalación y configuración de la base de datos

Zabbix utiliza una base de datos para almacenar los datos de monitoreo. En este caso, se utilizó PostgreSQL como el gestor de base de datos. La instalación se realiza con la siguiente instrucción: [29].

- sudo apt install postgresql postgresql-contrib

PostgreSQL es conocido por su robustez y escalabilidad, lo que lo convierte en una opción adecuada para manejar grandes volúmenes de datos de monitoreo generados por Zabbix. Su compatibilidad con Zabbix permite una integración eficiente y un rendimiento óptimo del sistema de monitoreo [30].

Configuración de base de datos PostgreSQL

De acuerdo con el manual de instalación de Zabbix, es fundamental crear un usuario con los permisos necesarios para acceder y realizar operaciones en la base de datos. La Figura 3.9 ilustra un fragmento del proceso de configuración de la base de datos destinada a almacenar los datos obtenidos del monitoreo utilizando PostgreSQL. Este proceso incluye la creación del usuario, la definición de la contraseña, la configuración de la codificación UTF-8, así como la ejecución de la instrucción `GRANT` para otorgar los permisos necesarios al usuario Zabbixuser en la tabla Zabbixdb.

Es importante garantizar que el usuario tenga los privilegios adecuados para realizar operaciones de lectura y escritura, lo cual es esencial para el correcto funcionamiento del sistema de monitoreo. La configuración de la codificación UTF-8 asegura que la base de datos pueda manejar una amplia variedad de caracteres, lo cual es particularmente importante en entornos multilingües [31].

3.2.5. Descargar e instalar Zabbix

Para la descarga e instalación del software Zabbix, se ejecutan las siguientes instrucciones:

```
wget https://repo.Zabbix.com/Zabbix/5.4/ubuntu/pool/main/z/Zabbix-release

/Zabbix-release_5.4-1+ubuntu22.04_all.deb
```

[3] Es un esfuerzo internacional para crear un único conjunto de códigos para todos los caracteres de texto, lo que facilita la compatibilidad y la comunicación.

```
iseo@zabbixserver:~$ sudo -i -u postgres
postgres@zabbixserver:~$ psql
psql (14.9 (Ubuntu 14.9-0ubuntu0.22.04.1))
Type "help" for help.

postgres=# CREATE DATABASE zabbixdb WITH ENCODING 'UTF8';
ERROR:  database "zabbixdb" already exists
postgres=# CREATE USER zabbixuser WITH PASSWORD 'zabbix';
CREATE ROLE
postgres=# ALTER ROLE zabbixuser SET client_encoding TO 'UTF8';
ALTER ROLE
postgres=# ALTER ROLE zabbixuser SET default_transaction_isolation TO 'read committed';
ALTER ROLE
postgres=# ALTER ROLE zabbixuser SET timezone TO 'UTC';
ALTER ROLE
postgres=# GRANT ALL PRIVILEGES ON DATABASE zabbixdb TO zabbixuser;
GRANT
postgres=#
```

Figura 3.9: Creación de base de datos en postgresql.

`sudo dpkg -i Zabbix-release_5.4-1+ubuntu22.04_all.deb`

Estas instrucciones se obtuvieron de la página web oficial de Zabbix. La primera instrucción descarga el paquete de instalación de Zabbix desde el repositorio oficial utilizando wget [32]. La segunda instrucción utiliza la función "dpkg"para instalar el paquete descargado en el sistema. Este proceso garantiza que se instale la versión correcta de Zabbix en Ubuntu 22.04 [33].

La instalación del paquete Zabbix-release es fundamental, ya que agrega el repositorio oficial de Zabbix a la lista de fuentes de paquetes del sistema. Esto permite que Zabbix y sus dependencias se instalen mediante el gestor de paquetes de Ubuntu, lo que facilita la actualización y el mantenimiento del software. Una vez que se cuenta con el repositorio de Zabbix en el sistema, se procedió a instalar el servidor Zabbix con las siguientes instrucciones [34].

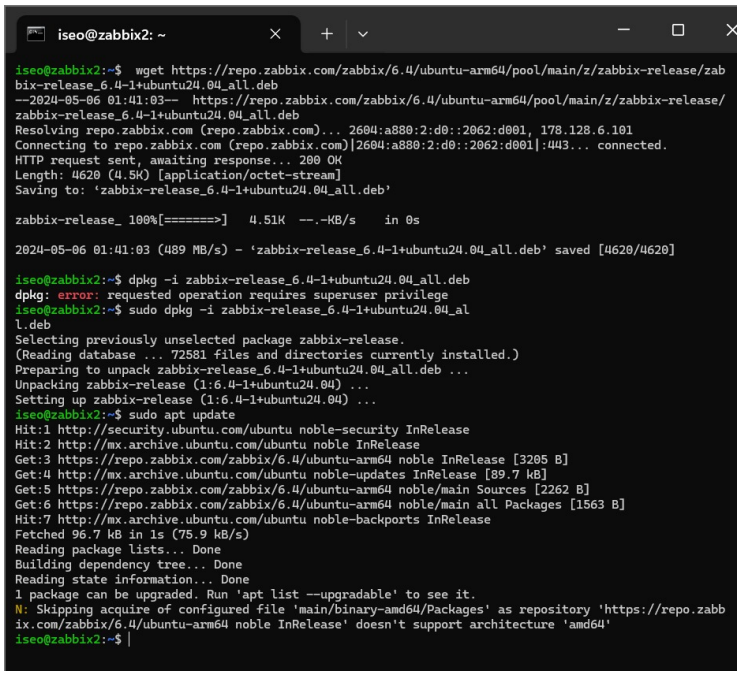


Figura 3.10: Instalación repositorio Zabbix

3.3. Instalación del Servidor Zabbix

Una vez que se cuenta con el repositorio de Zabbix en el sistema, se procede a instalar el servidor Zabbix con las siguientes instrucciones [32].

```
sudo apt update

sudo apt install Zabbix-server-pgsql Zabbix-frontend-php Zabbix-apache-conf

Zabbix-sql-scripts Zabbix-agent
```

La Figura 3.10 muestra el proceso de instalación realizado. La instrucción `sudo apt update` actualiza la lista de paquetes disponibles y sus versiones, asegurando que se instalen las versiones más recientes de los paquetes necesarios. Posteriormente, la instrucción `sudo apt install` se utiliza para instalar los siguientes componentes [33].

- **Servidor Zabbix**: Es el núcleo del sistema de monitoreo, encargado de recopilar y almacenar datos de monitoreo.
- **Frontend PHP de Zabbix**: Proporciona una interfaz web para interactuar con el servidor Zabbix, visualizar datos y configurar el sistema.
- **Configuración de NGINX para Zabbix**: Ajusta el servidor web Nginx para proporcionar la interfaz web de Zabbix.
- **Scripts SQL de Zabbix**: Incluyen las definiciones de la base de datos necesarias para almacenar los datos de monitoreo.
- **Agente de Zabbix**: Se instala en los dispositivos a monitorear y envía datos al servidor Zabbix.

3.3.1. Importación de Datos y Esquema

En este paso, se procede a importar y configurar el esquema inicial en la base de datos PostgreSQL [30].

La instrucción utilizada es la siguiente **mineriacongaby2024**:

- `zcat /usr/share/doc/Zabbix-sql-scripts/postgresql/create.sql.gz | psql -U Zabbix -d Zabbix`

La instrucción `zcat` descomprime el archivo `create.sql.gz`, y la instrucción `psql` ejecuta el contenido del archivo descomprimido en la base de datos `Zabbix` utilizando el usuario `Zabbix`.

3.3.2. Configuración del servidor Zabbix

Se debe editar el archivo de configuración del servidor Zabbix para especificar los detalles de la base de datos. Este archivo se encuentra en la ruta [30].

`/etc/Zabbix/Zabbix_server.conf`

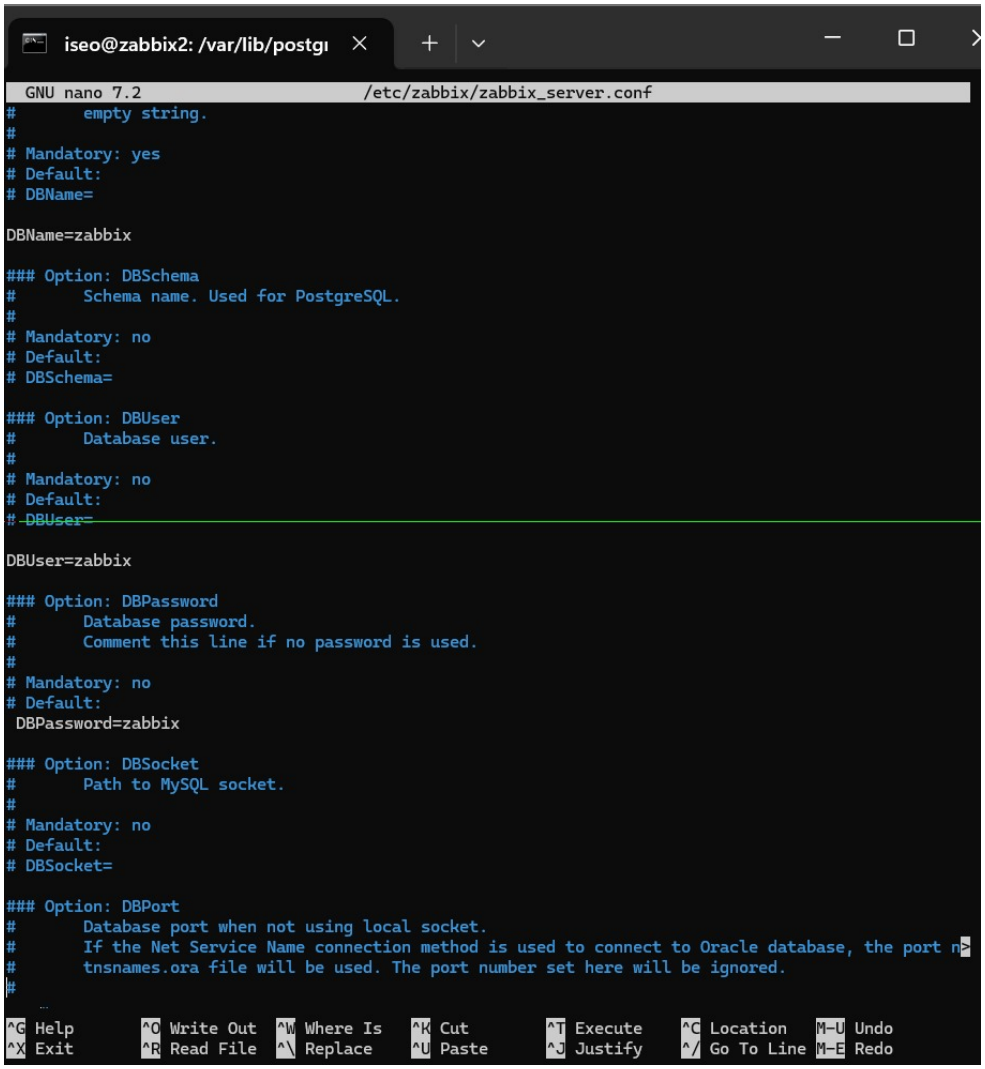


Figura 3.11: Configuración Base de Datos

Como se puede observar en la figura 3.11, dentro del archivo mencionado se configuraron los parámetros de acuerdo con los que se crearon en la base de datos PostgreSQL [30].

- DBName=Zabbix
- DBUser=Zabbix
- DBPassword=contraseña

3.3.3. Reinicio y activación del servidor Zabbix

Por último, se utilizaron las siguientes instrucciones para reiniciar el servidor Zabbix y habilitar el servicio, de modo que se inicie cada vez que el servidor Ubuntu se encienda [32].

- sudo systemctl restart Zabbix-server Zabbix-agent nginx
- sudo systemctl enable Zabbix-server Zabbix-agent nginx

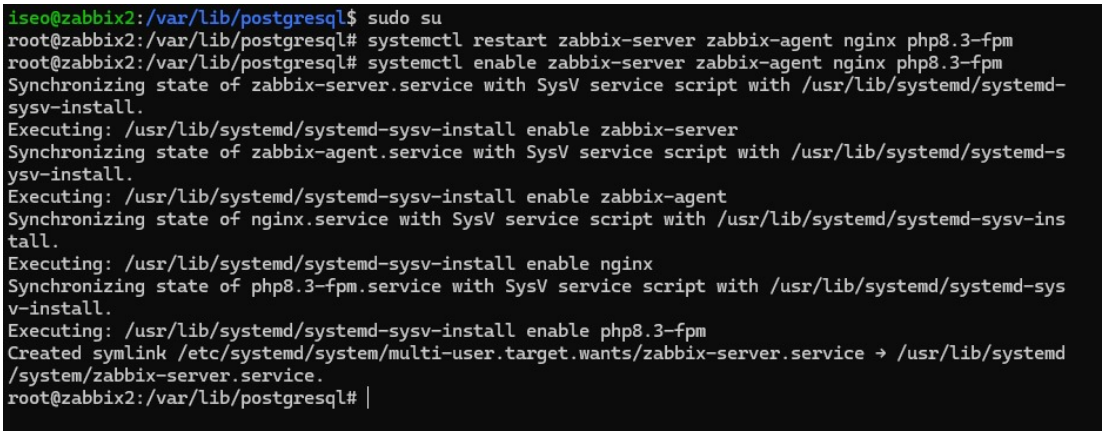


Figura 3.12: Reinicio de agente Zabbix

3.3.4. Instalación en la interfaz web del servidor

Para acceder a la interfaz web de Zabbix, se debe ingresar la IP del servidor, en este caso, 192.168.1.30, utilizando el puerto 8080 a través de cualquier navegador. Al entrar por primera vez después de la instalación mediante línea de comandos, se inicia el script de configuración de Zabbix. Este script es una guía de varios pasos para configurar Zabbix, incluyendo la conexión a la base de datos y la configuración de parámetros del servidor desde una interfaz gráfica [14].

Figura 3.13: Interfaz de bienvenida Zabbix

Verificación de pre-requisitos

En la interfaz de verificación de requisitos previos para la instalación de Zabbix, es fundamental asegurarse de que todos los componentes estén marcados como "OK" lo cual garantiza que todos los elementos necesarios estén configurados correctamente antes de continuar con el proceso de instalación [30].

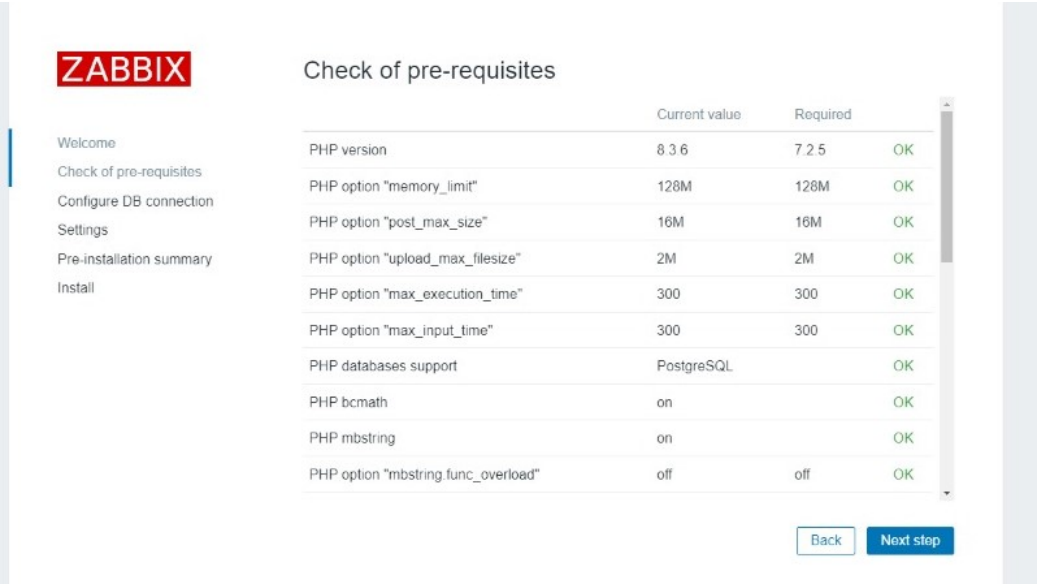


Figura 3.14: Prerrequisitos de instalación

La figura 3.14 muestra la lista de prerrequisitos para la instalación de Zabbix.

Configuración de conexión a la base de datos

En este paso, se verifican las configuraciones previamente realizadas durante el proceso de instalación mediante línea de comandos, y se proporciona la información necesaria para establecer la conexión con la base de datos, la cual fue definida en el archivo de configuración [30].

Se deben introducir los siguientes detalles:

- **Nombre de la base de datos:** Se especifica el nombre de la base de datos de Zabbix.
- **Usuario de la base de datos:** Se proporciona el nombre de usuario utilizado para acceder a la base de datos.
- **Contraseña:** Se establece la contraseña asociada al usuario de la base de datos.

Una vez completada esta configuración, se procede con el siguiente paso del proceso de instalación de Zabbix [30].

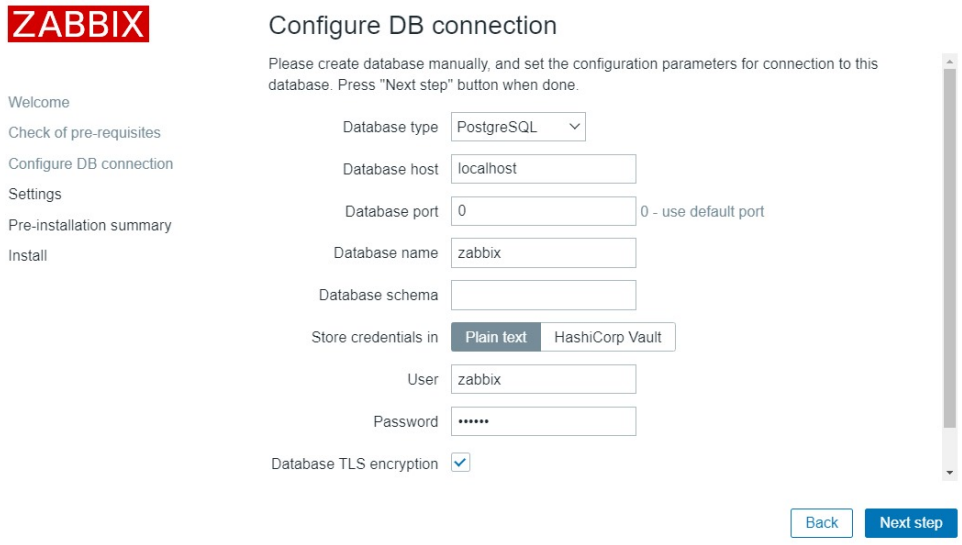


Figura 3.15: Conexión a la base de datos.

La figura 3.15 muestra la verificación de la configuración para la conexión a la base de datos.

Información del servidor Zabbix

En este apartado, se puede configurar un nombre para el servidor y la interfaz web, con el fin de distinguirlo en caso de contar con varios servidores de monitoreo [30].

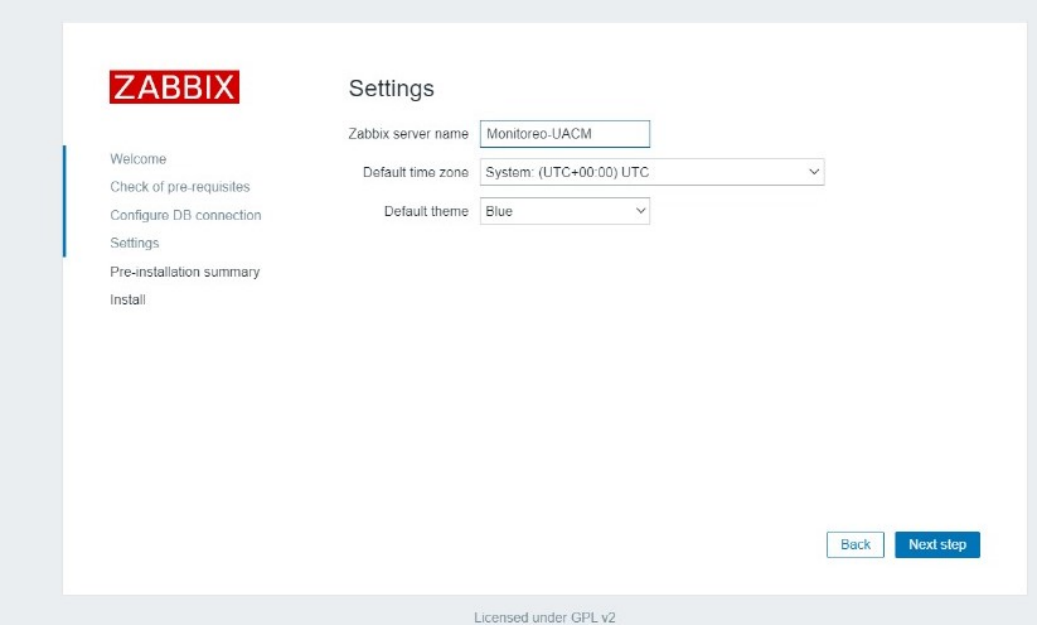


Figura 3.16: Interfaz web de Zabbix durante el proceso de configuración

La figura 3.16 muestra la asignación del nombre para el servidor, que fue: MONITOREO UACM.

Resumen de la instalación

Antes de proceder con la instalación, se presenta una interfaz resumida de preinstalación. En esta etapa, se verifica que todas las configuraciones sean las predeterminadas o estén acordes a lo que se ha configurado previamente para poder continuar con la instalación [30].

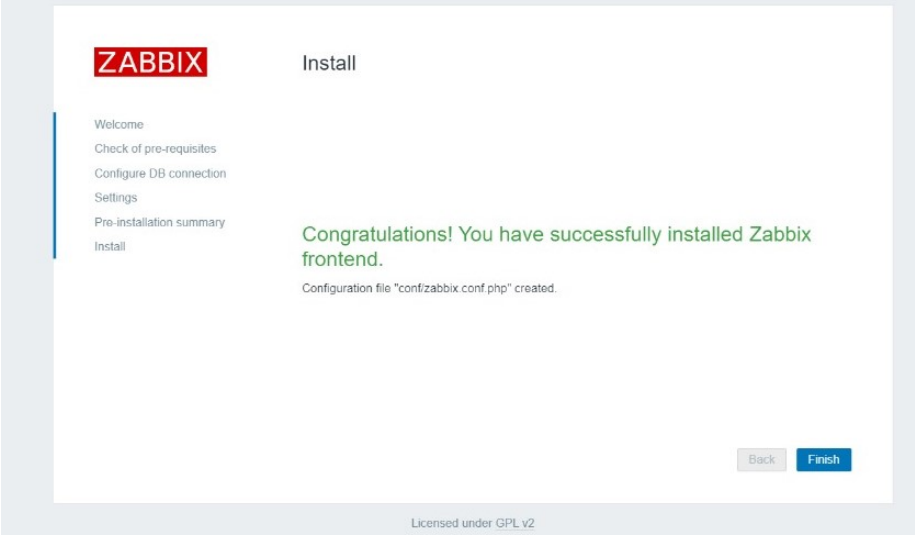


Figura 3.17: Configuración Zabbix

Una vez finalizada la instalación, la interfaz nos muestra un mensaje de que la instalación se ha completado, como se muestra en la figura 3.17.

Inicio de sesión

Una vez que se complete la instalación de Zabbix, se despliega automáticamente la interfaz de inicio de sesión. En este punto, ya es posible acceder a la interfaz web del servidor utilizando el nombre de usuario "Admin"‘ y la contraseña "Zabbix"‘ [30].

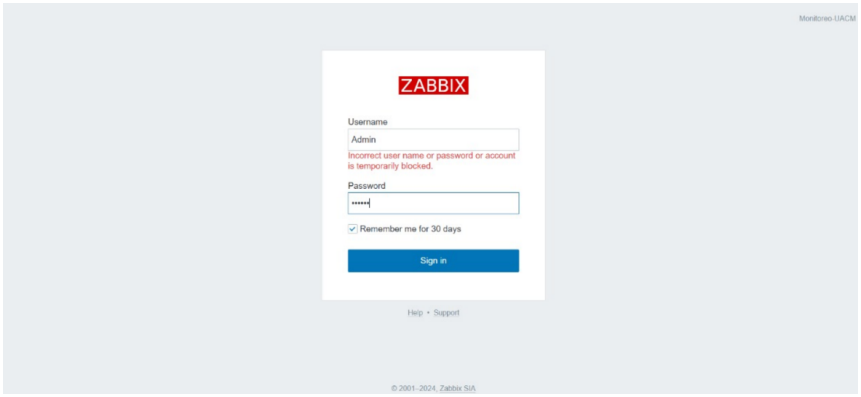


Figura 3.18: Inicio de sesión por el portal de Zabbix

La figura 3.18 muestra el ingreso de usuario y contraseña para iniciar sesión en la interfaz web del servidor Zabbix.

Finalmente, en la figura 3.19 se muestra la interfaz web del servidor con su vista global y opciones de configuración, listas para realizar las configuraciones de monitoreo de red correspondientes.

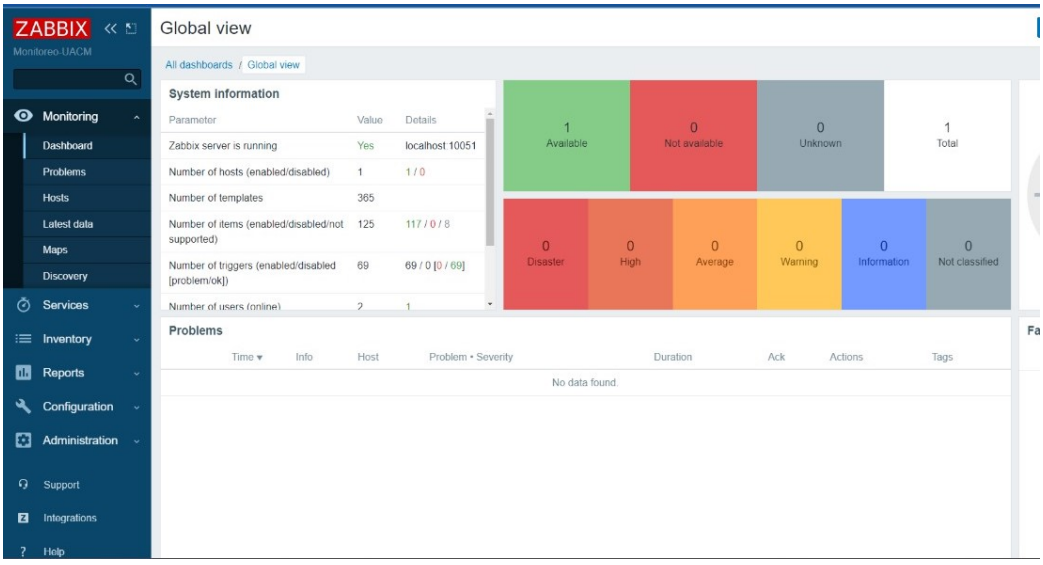


Figura 3.19: Panel de control Zabbix

También se mostrará información sobre los parámetros de la memoria RAM y el uso de la CPU del servidor de monitoreo, los cuales están establecidos de manera predeterminada [30].

3.4. Monitoreo de la LAN con Zabbix

3.4.1. Descubrimiento de equipos por medio del protocolo ICMP

La integración de ICMP con Zabbix facilita la supervisión continua de la disponibilidad y el rendimiento de los dispositivos conectados a la red [14].

Para realizar esta configuración en Zabbix, se llevó a cabo el siguiente proceso:

1. En la subsección de **Descubrimiento**, seleccionar la opción que indica crear regla de descubrimiento.

2. Se configuró el rango de direcciones IP para el descubrimiento, abarcando todas las direcciones posibles dentro del segmento de red 192.168.1.0 - 192.168.1.254. El intervalo de actualización se definió en 5 minutos, lo cual es una frecuencia adecuada que evita la saturación del sistema.

   Estas comprobaciones Zabbix utilizará para el descubrimiento, entre otros.

3. En las **comprobaciones**, se seleccionó la de ICMP ping.

Una vez configurada la regla de descubrimiento, nos dirigimos a la sección de equipos (hosts) para establecer que cada host descubierto mediante esta regla sea registrado en un grupo específico de hosts. Este proceso se ilustra en la figura 3.20 Los parámetros solicitados por el formulario son los siguientes:

1. Nombre de la acción: Se asignó el nombre de "Descubrimiento de red por ICMP ping.

2. Añadir a un grupo de plantillas, en este caso "Discovery check"

3. Habilitar la acción creada.

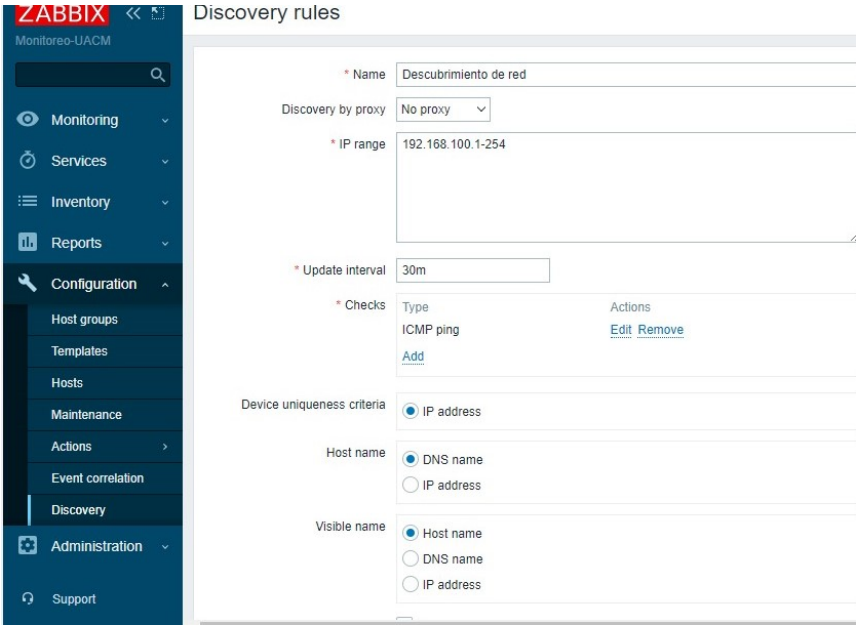


Figura 3.20: Configuración para el descubrimiento de equipos por medio del protocolo ICMP en Zabbix.

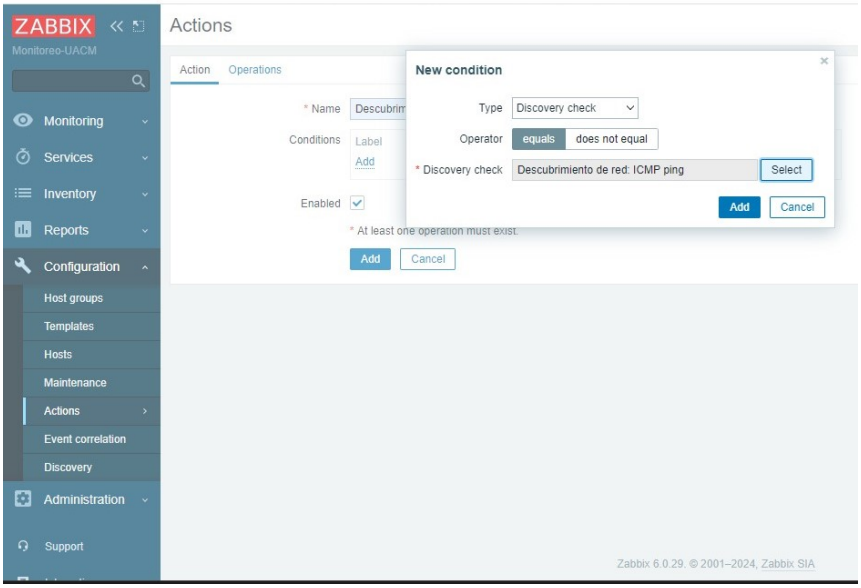


Figura 3.21: Activación de acción para registrar hosts mediante el descubrimiento de equipos por ICMP.

Al utilizar la validación ICMP *ping*, se envía un mensaje de solicitud de *eco* [4] mediante el protocolo ICMP a una dirección IP determinada; en este caso, cualquier dirección dentro del

[4]Un mensaje de *eco* en ICMP es una solicitud de respuesta utilizada para verificar la conectividad entre dispositivos en una red.

rango asignado. Si el host de destino es accesible y está en funcionamiento, este devolverá un mensaje de respuesta de *eco*.

En la figura 3.22 se observa un ejemplo de la ejecución de la instrucción *ping* para verificar la conexión con las direcciones IP 192.168.1.1 y 192.168.1.2.

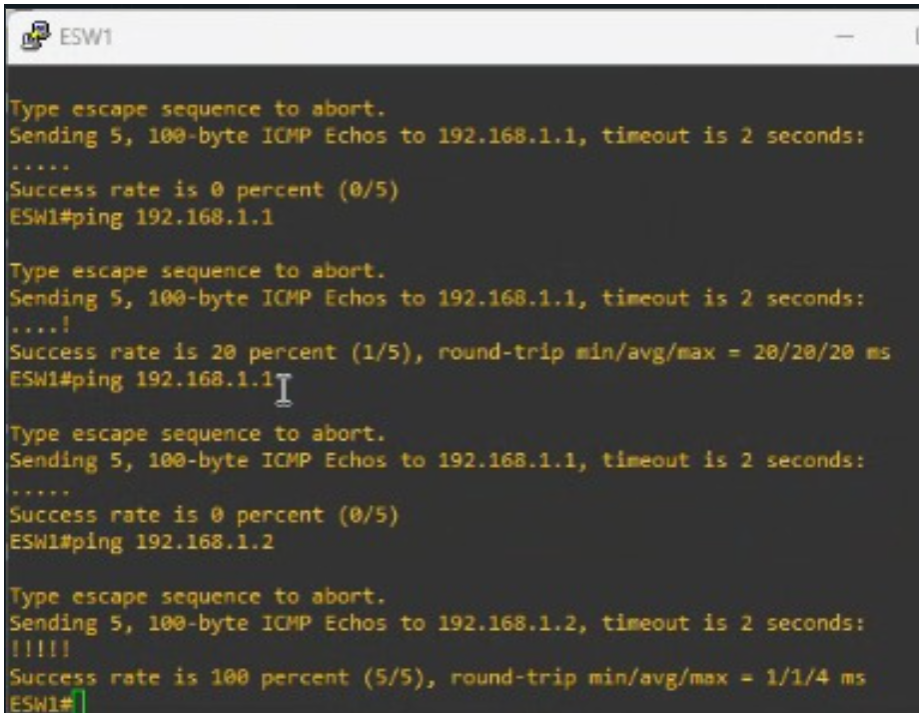


Figura 3.22: Ping ICMP entre el conmutador y el enrutador.

3.4.2. Activación del protocolo SNMP en el enrutador

La habilitación del protocolo SNMP (Simple Network Management Protocol, por sus siglas en inglés) permite la supervisión y gestión remota de dispositivos de red. SNMP es un estándar de gestión de red ampliamente utilizado que facilita la recopilación de datos sobre el rendimiento y la supervisión del estado de los dispositivos de red [35].

En esta sección, se detalla el proceso de activación del protocolo SNMP en un enrutador.

En esta configuración se establece una comunidad con permisos de solo lectura, mediante la instrucción *Gigabit Ethernet-server community public RO*. La gestión de eventos y distintos tipos de alertas SNMP se activa con las líneas *Gigabit Ethernet-server enable traps*. Por último, se define el host receptor de las notificaciones SNMP; este se define con la instrucción *Gigabit Ethernet-server host 192.168.1.4 version 2c public udp-port 161*, lo cual asegura que las alertas lleguen a la dirección IP del servidor de monitoreo.

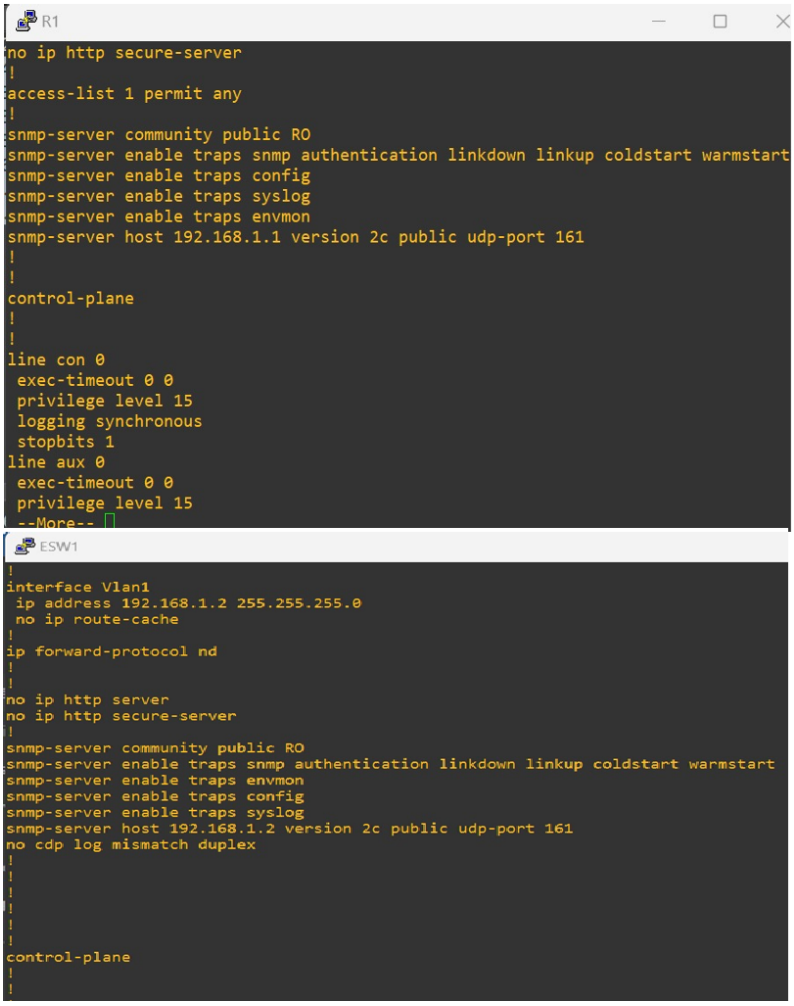


Figura 3.23: Configuración de SNMP en enrutador 1 y Configuración de red en conmutador

En la figura 3.23 se muestra la configuración del conmutador 1, donde se utiliza la interfaz VLAN1, configurada con la dirección IP 192.168.1.2 y la máscara de subred 255.255.255.0.
El protocolo SNMP se encuentra, al igual que el enrutador 1, configurado con la comunidad *public* en modo de solo lectura (RO). La comunidad 'public' de solo lectura fue seleccionada para limitar el acceso a consultas de monitoreo, reduciendo así los riesgos de seguridad, y permitiendo la habilitación de SNMP traps para eventos como *linkdown*, *linkup*, *coldstart* y *warmstart*[5].

La instrucción `Gigabit Ethernet-server community public RO` establece una comunidad SNMP de solo lectura (RO) llamada `public`. Esta configuración permite a los administradores de red consultar información del enrutador, como datos de tráfico y el estado de las interfaces, sin otorgarles permisos para realizar cambios en la configuración.

[5]Un "trap" en SNMP es un mensaje enviado de forma automática desde un dispositivo gestionado hacia el servidor de monitoreo para notificar eventos o condiciones específicas, como cambios en el estado del dispositivo. Los traps son útiles para alertar de situaciones como caídas de enlace (*linkdown*), establecimiento de enlace (*linkup*), o reinicios del dispositivo (*coldstart* y *warmstart*).

La línea `snmp-server host 192.168.1.1 version 1c public udp-port 161` define el host de destino para recibir notificaciones SNMP. En este caso, el host con dirección IP `192.168.1.1` recibirá dichas notificaciones utilizando la versión `1c` del protocolo SNMP y la comunidad `public` para la autenticación. Además, se especifica el puerto UDP 161, que es el puerto estándar para el protocolo SNMP.

Esta configuración permite al enrutador enviar alertas SNMP al host especificado, facilitando así la supervisión remota del dispositivo. El tráfico de monitoreo es gestionado por el enrutador, que retransmite la información relevante al servidor de monitoreo para su análisis.

Es importante mencionar que la configuración SNMP `public` utilizada es común en configuraciones predeterminadas. Sin embargo, en un entorno de producción, se recomienda emplear un identificador único y seguro para evitar accesos no autorizados a los dispositivos.

3.5. Agente de monitoreo Zabbix

3.5.1. Instalación del Agente de Zabbix en Windows

La instalación de agentes en dispositivos objetivo es fundamental para el monitoreo con Zabbix, ya que permite la recolección y envío de datos sobre el rendimiento y estado al servidor de Zabbix.

3.5.1 Proceso de instalación del agente de Zabbix en Windows

El agente recoge métricas clave, como el uso de CPU, memoria y espacio en disco.
Los pasos para su instalación en Windows son:

1. **Descargar el instalador del agente de Zabbix:** Desde el sitio oficial de Zabbix (`https://www.Zabbix.com`) descargar la versión más reciente del instalador del agente para Windows, asegurándose de seleccionar la arquitectura correcta (32 o 64 bits) según el sistema operativo.

2. **Ejecutar el instalador:** Abrir el archivo descargado y seguir el asistente de instalación. Durante este proceso, se solicitarán parámetros específicos de configuración.

3. **Configurar el agente:**

   - **Dirección del servidor Zabbix:** Introducir la dirección IP o nombre de dominio del servidor Zabbix al que el agente enviará los datos.
   - **Hostname:** Especificar el nombre del host (dispositivo) que será monitoreado, el cuál debe coincidir con el configurado en la interfaz de Zabbix para asegurar una identificación correcta.

- **Puerto de comunicación:** El puerto por defecto es 10050, pero puede cambiarse si se requiere.

4. **Iniciar el servicio de Zabbix Agent:** Una vez completada la instalación, iniciar el servicio del agente para que comience a recopilar y enviar datos al servidor Zabbix. Esto puede hacerse automáticamente al finalizar la instalación o manualmente a través de la consola de servicios de Windows (`services.msc`).

5. **Verificación de la instalación:** Finalmente, comprobar en la interfaz de Zabbix que el host Windows aparece como *activo* y que se están recibiendo los datos de monitoreo.

Instalar el agente de Zabbix en Windows permite integrar estos dispositivos en el monitoreo centralizado, facilitando la supervisión en tiempo real de métricas críticas. La elección de Windows es esencial debido a su amplia utilización en entornos industriales, lo que asegura un monitoreo eficiente y una gestión proactiva de TI. El agente se instala y configura en Windows, listo para enviar datos al servidor central, lo que permite una supervisión detallada del dispositivo.

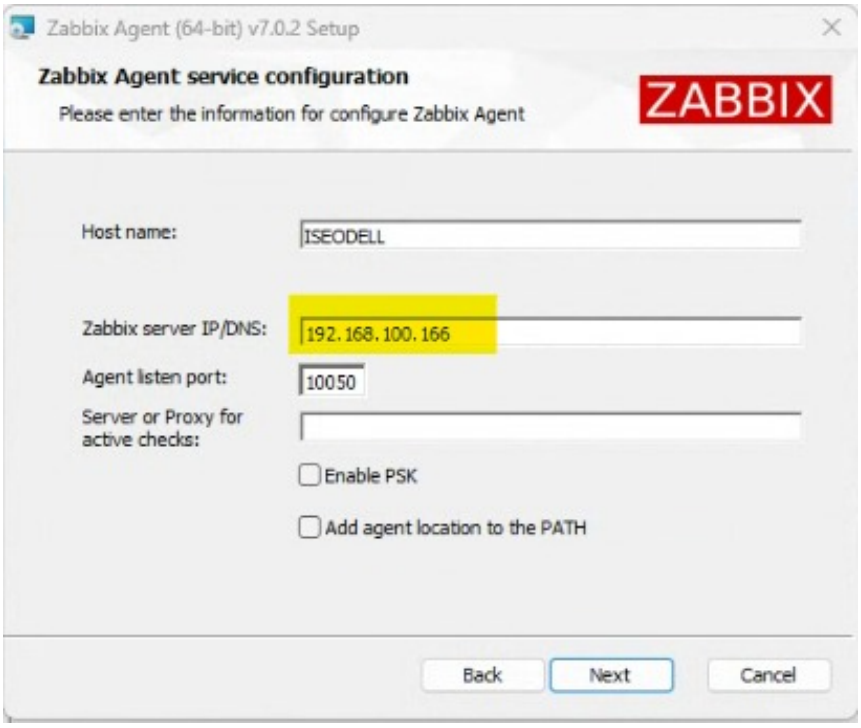


Figura 3.24: Proceso de instalación de agente Zabbix en servidro Windows

Como se puede observar en la figura 3.24, se establecieron los siguientes parámetros: el nombre de host **ISEODELL**, la dirección IP **192.168.1.66** y, por último, el puerto **10050** que es el predeterminado para establecer la comunicación entre el servidor y el agente Zabbix.

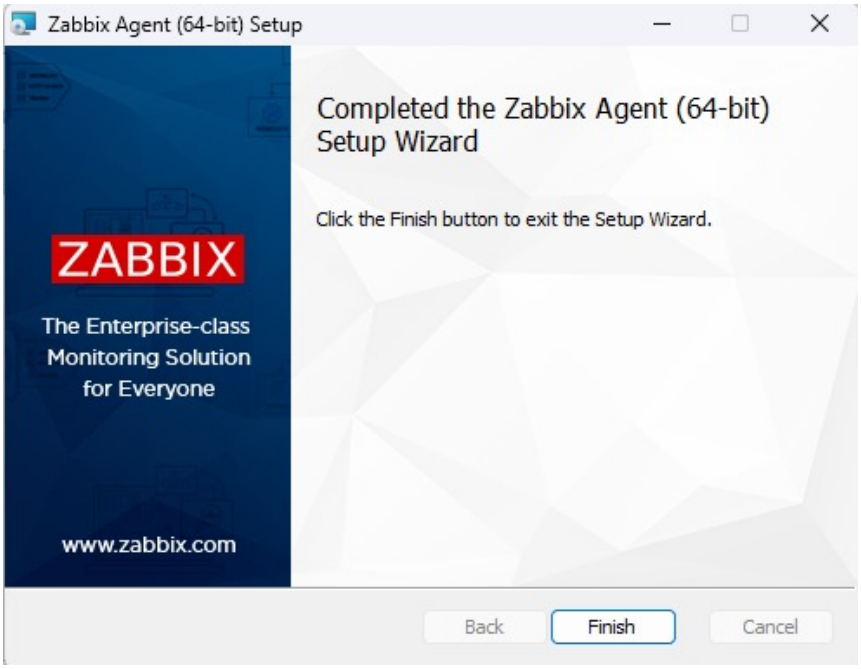


Figura 3.25: Finalización de proceso de instalación

Una vez instalado, podemos verificar a nivel del sistema operativo que el agente de Zabbix está habilitado correctamente. En la figura 3.26 se visualiza la configuración del firewall, específicamente las reglas que determinan el tráfico permitido y restringido. Las columnas presentadas incluyen detalles como el nombre de la regla, el grupo al que pertenece, el perfil de red aplicable, si está habilitada, la acción que realiza y el programa asociado. También se detallan los protocolos de red, como TCP, y los puertos correspondientes que se utilizan para la comunicación de datos.

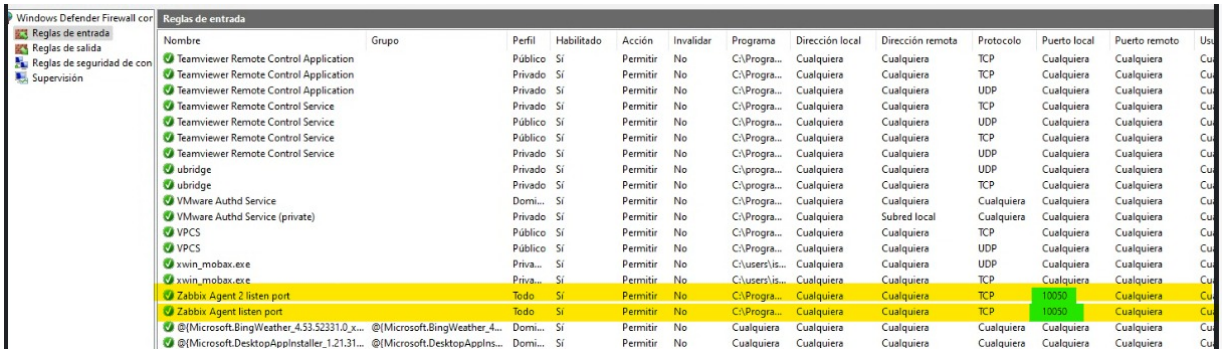


Figura 3.26: Reglas aplicadas al firewall tras la instalación del agente Zabbix

3.6. Instalación y Configuración de SNMP en Ubuntu

La figura 3.27 muestra el proceso de configuración para habilitar la recepción de traps SNMP en el servidor Zabbix. Se modificó el archivo de configuración del servidor Zabbix y se descargó un script que se utiliza para recibir y procesar traps SNMP en el sistema. Este procedimiento es parte de la configuración avanzada de monitoreo en Zabbix, permitiendo administrar eventos generados por dispositivos SNMP.

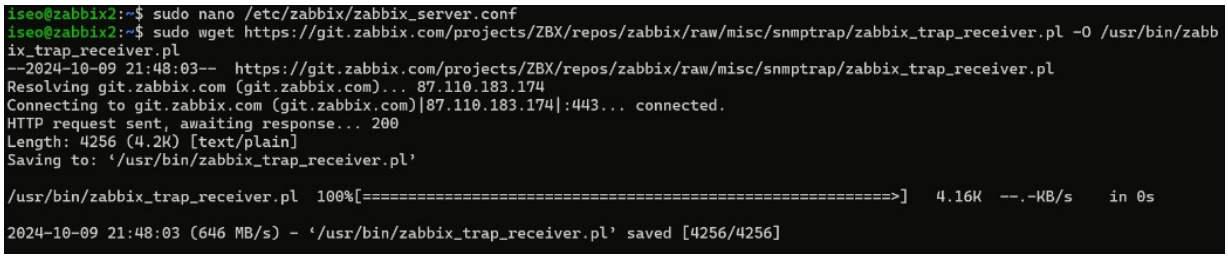


Figura 3.27: Configuración para habilitar y procesar traps SNMP.

Los valores que deben configurarse se detallan a continuación:

1. **SNMPTrapperFile:**
   - Archivo temporal utilizado para pasar datos desde el "demonio" SNMP trap al servidor.
2. **StartSNMPTrapper:**
   - Debe estar en 1 para que el proceso sea iniciado.
3. **ListenIP:**
   - En esta opción se define el rango de las IPs desde las que se va a obtener información.

En la figura 3.28 se detalla la instalación de paquetes durante este proceso:

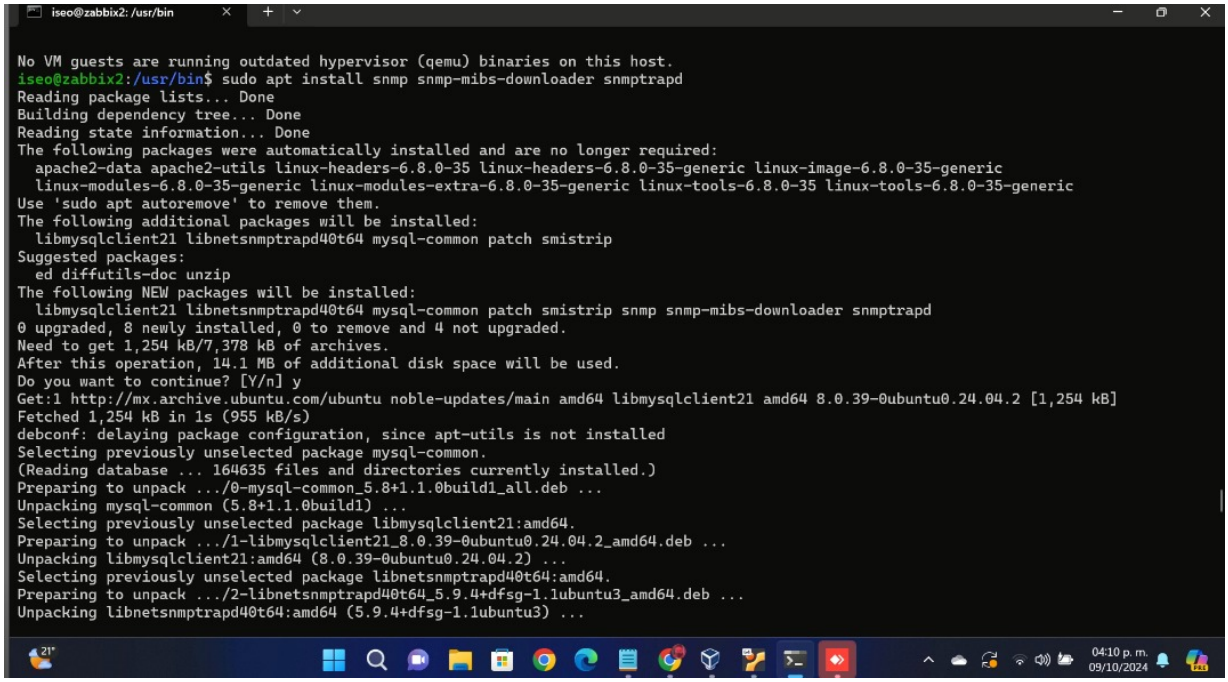


Figura 3.28: Instalación de paquetes en un entorno Linux.

- La instrucción utilizada es *sudo apt install snmp snmp-mibs-downloader snmptrapd*, que tiene como objetivo instalar los paquetes relacionados con SNMP (Simple Network Management Protocol).
- **snmp:** Es una herramienta para interactuar con dispositivos habilitados para SNMP.
- **snmp-mibs-downloader:** Descarga los archivos MIB (Management Information Base), que definen las jerarquías y objetos gestionables a través de SNMP.
- **snmptrapd:** Un "demonio" [6] que recibe y procesa los traps SNMP (mensajes de alerta generados por dispositivos de red).
- **libnetsnmptrapd40:** Es una biblioteca relacionada con la recepción de traps SNMP.

En la figura 3.29 se muestra una terminal donde se gestiona el servicio *snmptrapd*, que es un "demonio" que administra traps SNMP (Protocolo de Administración Simple de Red). La instrucción ejecutada para verificar el estado del servicio indica que se encuentra activo y funcionando desde el 9 de octubre de 2024.

[6]Programa que se ejecuta en segundo plano en un sistema operativo para realizar tareas o servicios sin la intervención directa de un usuario.

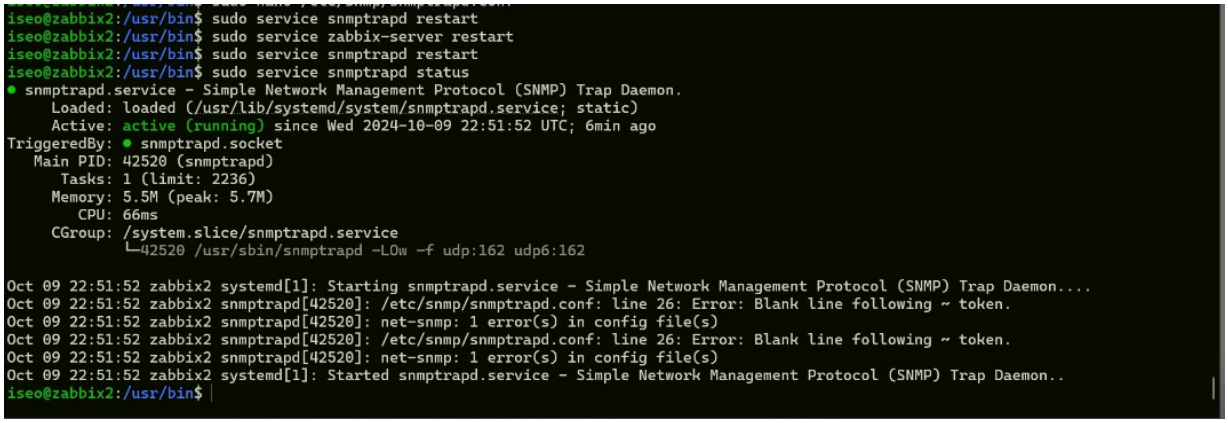


Figura 3.29: Instalación del servicio *snmptrapd* en un entorno Linux.

3.7. Creación de Hosts y Configuración de Plantillas en Zabbix

El proceso para crear un host en Zabbix implica completar los campos requeridos en la pestaña“Hosts”. Esto incluye información como el nombre del host, que debe coincidir con la configuración del agente de Zabbix en el servidor, y la dirección IP del dispositivo a monitorear. Esta configuración permite asociar correctamente el dispositivo físico o virtual con el sistema de monitoreo y garantizar que Zabbix pueda recolectar y procesar los datos relevantes. Además, asignar las plantillas adecuadas a los hosts permite automatizar la recolección de métricas y la supervisión, facilitando la configuración de alertas y la gestión de eventos.

Para llevar a cabo este proceso, es necesario completar los campos requeridos en la pestaña Host, como el nombre del host, que debe coincidir con la configuración del agente de Zabbix en el servidor, la dirección IP del servidor y otros detalles.

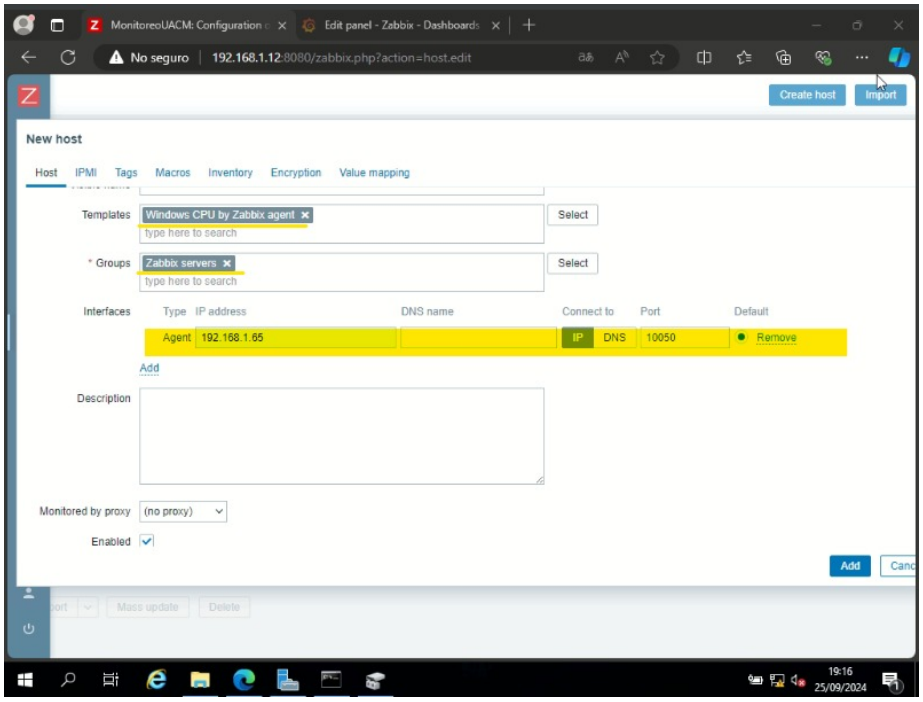


Figura 3.30: Configuración de host

Al agregar el host, se debe realizar un ping hacia la dirección IP que se encuentra en el mismo segmento; en este caso, 192.168.1.65. Al hacerlo, es posible visualizar en la figura 3.31 que se establece una comunicación efectiva con el agente previamente instalado.

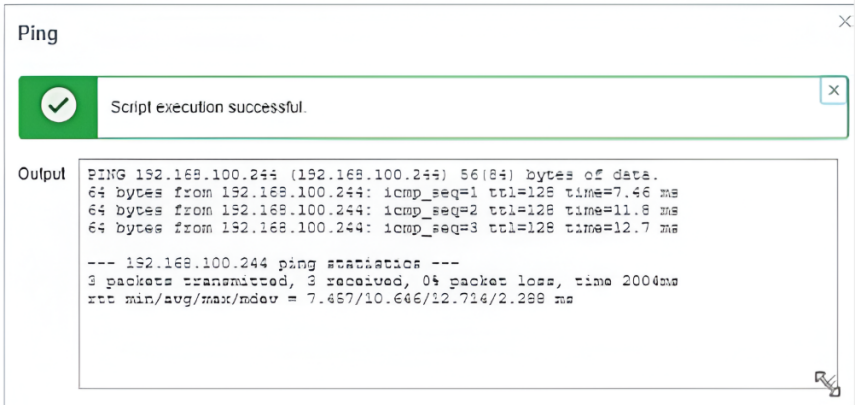


Figura 3.31: Comunicación con el agente mediante ping

Agregar una plantilla (template) adecuada: En Zabbix, un *template* es un conjunto preconfigurado de elementos de monitoreo, como ítems, gráficas, disparadores y acciones, que se puede aplicar a múltiples dispositivos con características similares. Utilizar un template facilita la configuración y estandarización del monitoreo, ya que permite aplicar configuraciones comunes de manera rápida y uniforme a varios hosts.

Para agregar plantillas, se debe acceder a la pestaña **Templates** en la interfaz de Zabbix y seleccionar la plantilla adecuada para el dispositivo en cuestión. De este modo, el agente de Zabbix en el host Windows utilizará las configuraciones de monitoreo definidas en la plantilla seleccionada, optimizando la implementación y el mantenimiento del sistema de monitoreo.

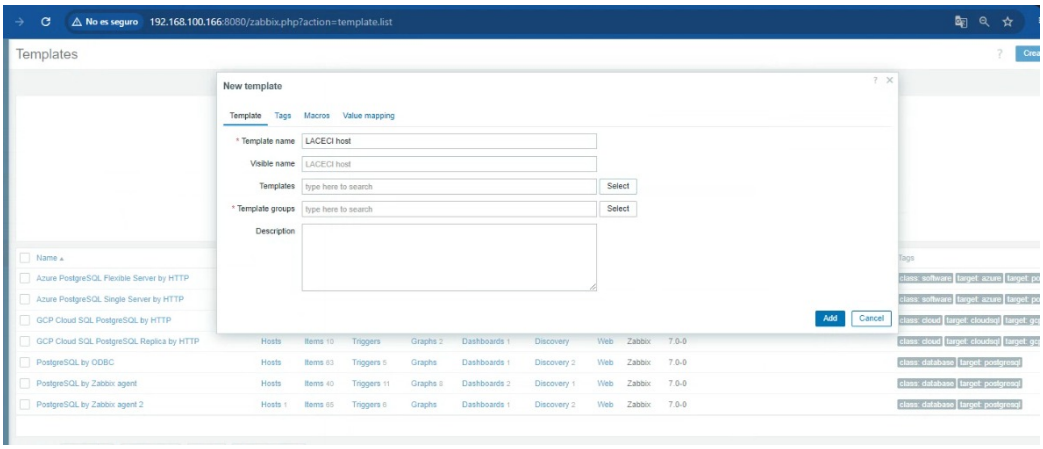


Figura 3.32: Creación de template.

Para monitorear un elemento específico en Zabbix, es necesario definir el *item* que se desea supervisar. Los *items* son elementos de monitoreo que permiten recopilar datos específicos sobre el estado y rendimiento de un dispositivo o sistema. Además, es crucial establecer un *disparador* que permita identificar y alertar sobre cualquier comportamiento anómalo o fuera de los parámetros establecidos.

Un *disparador* (o trigger, en inglés) en Zabbix es una condición lógica definida sobre uno o varios *items*, diseñada para detectar situaciones anormales o problemas en un sistema. Cuando un *disparador* se activa debido a que el valor del *item* supera un umbral o condición predefinida, Zabbix genera una alerta que puede notificarse a los administradores. Esto facilita la detección temprana de problemas y permite una respuesta proactiva en la gestión de la infraestructura.

Zabbix permite configurar una diversidad de *items* para supervisar distintos aspectos de redes o sistemas. Por ejemplo, se puede monitorear la utilización de la CPU, el uso de la memoria, el espacio disponible en disco, la carga de red, el estado de servicios específicos, el tiempo de actividad de los dispositivos, entre otros.

La figura 3.33 representa la configuración de un nuevo ítem en Zabbix. Para configurar un ítem, es necesario:

- **Seleccionar el tipo de ítem**: En este paso, se seleccionó "Simple check", lo que indica que se realizará una verificación básica.
- **Definir el tipo de información**: Aquí, se especificó que el ítem recopilará datos numéricos. En Zabbix, se pueden configurar diferentes tipos de datos según lo que se necesite medir, como números enteros, cadenas de texto, valores booleanos, entre otros.

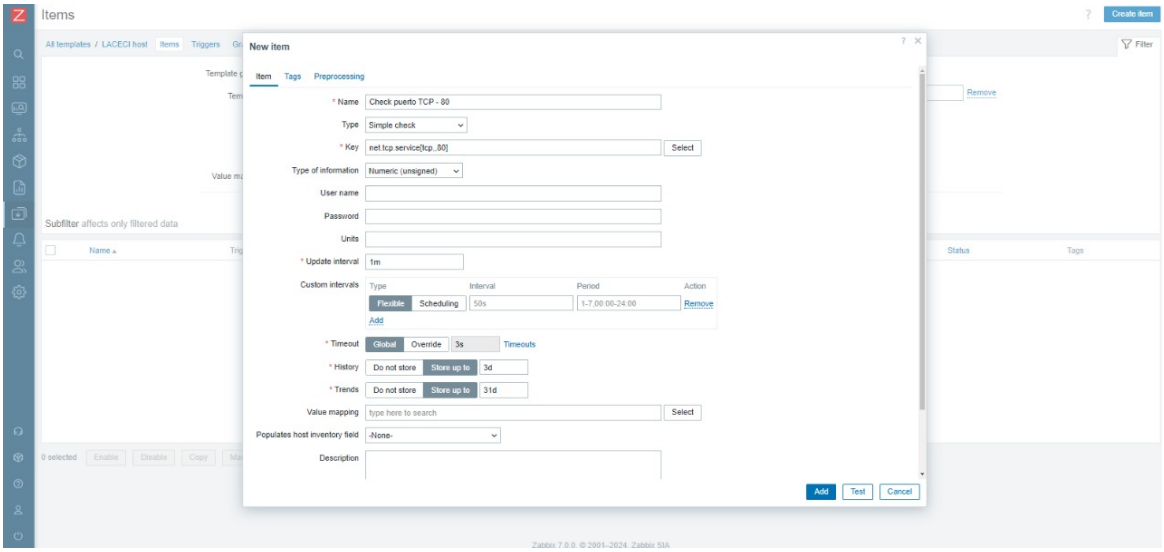


Figura 3.33: Creación de item

- **Establecer el intervalo de actualización**: Este parámetro determina la frecuencia con la que Zabbix realiza la recolección de datos para este ítem en particular. Para este caso se definió, un intervalo de actualización de 1 minuto significa que Zabbix verificará el estado del puerto TCP 80 cada minuto.

Este ítem se configuró para asegurar la disponibilidad del servicio web, ya que monitorea el puerto TCP 80. Esto permite detectar problemas en tiempo real, garantizando que los usuarios puedan acceder al contenido sin interrupciones.

Al completar estos pasos de configuración, Zabbix comenzará la recolección de datos del agente que se encuentra en el servidor.

3.8. Configuración de usuarios en Zabbix

La implementación de roles definidos en Zabbix constituye una estrategia para fortalecer la seguridad y optimizar la eficiencia en la administración de redes. Mediante la asignación de roles específicos, se establece un sistema de control de acceso que permite la definición precisa de quiénes tienen la facultad de visualizar o alterar determinados segmentos de Zabbix, limitando esta capacidad a los administradores. Este mecanismo de control no solo salvaguarda la información crítica, sino que también previene modificaciones no autorizadas que podrían comprometer la funcionalidad del sistema.

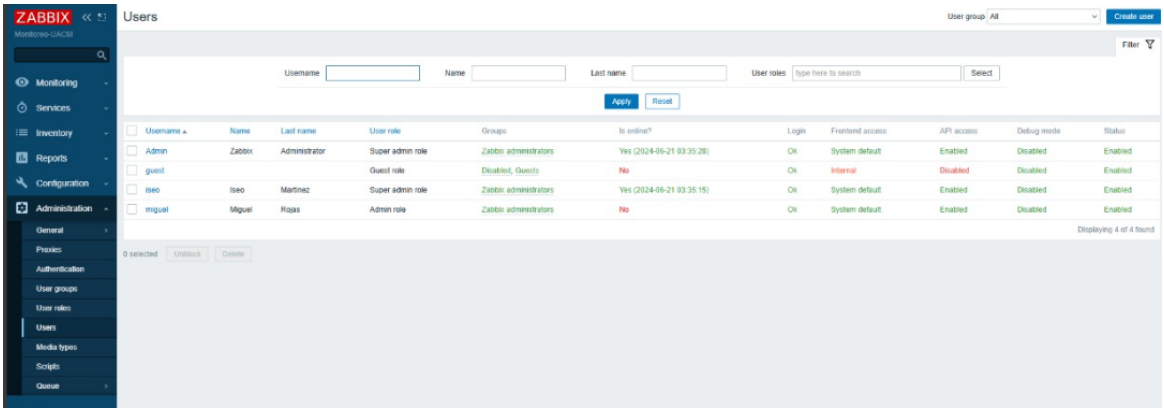


Figura 3.34: Usuarios conectados en Zabbix

Como se muestra en la figura **??**, se visualizan los usuarios conectados en Zabbix, lo cual permite a los administradores monitorear las sesiones activas y realizar auditorías sobre las acciones realizadas en el sistema.

Al clasificar a los usuarios según sus responsabilidades y concederles niveles de acceso equivalentes, se facilita considerablemente la gestión de permisos y se garantiza uniformidad en las operaciones de monitoreo. Esta metodología resulta particularmente beneficiosa en contextos donde múltiples usuarios requieren acceso concurrente a la información.

Por otro lado, la capacidad de monitorear la presencia de usuarios en línea otorga al administrador una herramienta para la auditoría y el seguimiento meticuloso de las acciones ejecutadas en el sistema en tiempo real. Dicha visibilidad es crucial para identificar cualquier actividad atípica y para fundamentar decisiones estratégicas en materia de seguridad de la red. En conjunto, estas medidas contribuyen a un entorno de red más seguro y eficiente, donde la integridad y la disponibilidad de los datos se mantienen como prioridades fundamentales.

3.9. Conexión de Zabbix con Grafana

Grafana se ha integrado con Zabbix para llevar a cabo una supervisión integral de redes y aplicaciones. Esta integración permite visualizar los datos de monitoreo recopilados por Zabbix en tiempo real, facilitando la interpretación y configuración de métricas críticas.

El proceso de integración consiste en añadir Zabbix como fuente de datos en Grafana, lo que permite importar y visualizar las métricas configuradas en Zabbix a través de paneles personalizables.

Para establecer esta conexión, se realizaron los siguientes pasos:

3.9.1. Preparación del entorno

```
sudo apt-get update
```

Instalación de las dependencias **Adduser** [7] y **libfontconfig1** [8] son paquetes necesarios para el correcto funcionamiento de Grafana [23].

```
sudo apt-get install -y adduser libfontconfig1
```

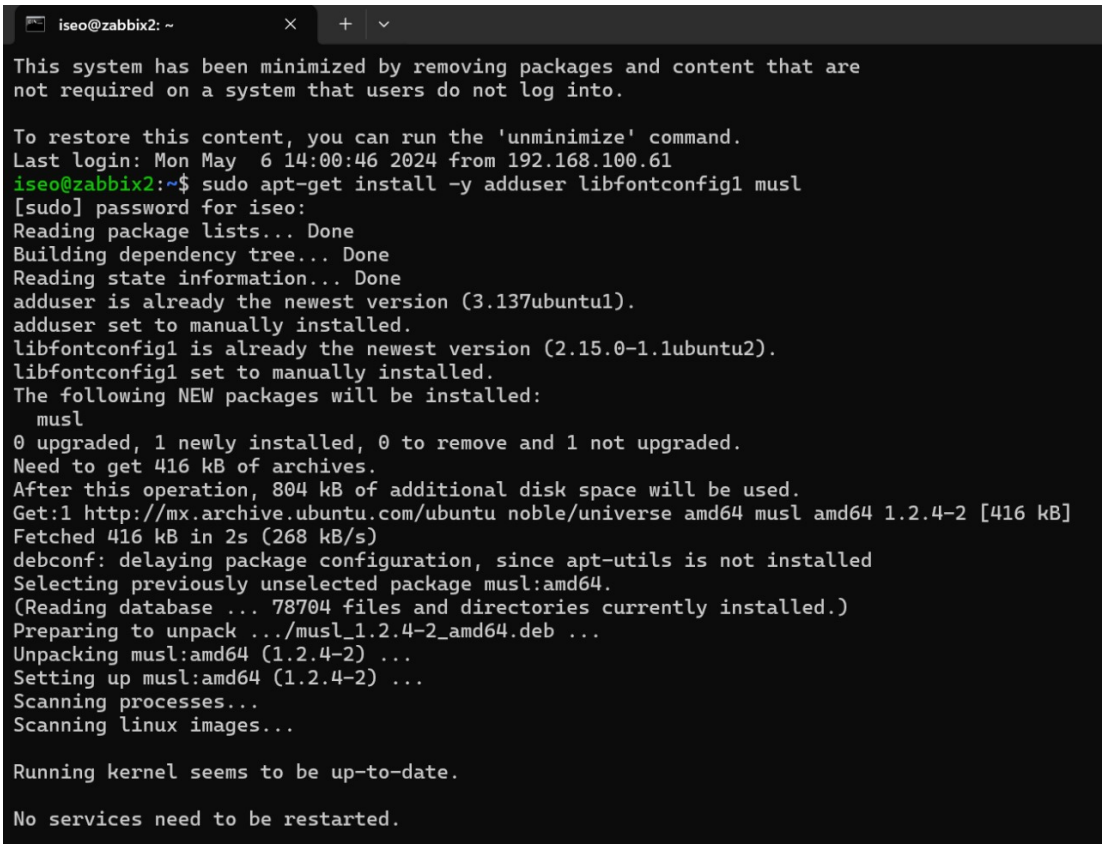


Figura 3.35: Instalación de Grafana.

También es necesario instalar los siguientes repositorios:

- **apt-transport-https:** Este paquete permite al sistema de gestión de paquetes **apt** transferir archivos e índices de paquetes a través de conexiones HTTPS. Es necesario porque se está añadiendo un repositorio que está alojado en HTTPS.
- **curl:** Es una herramienta que permite transferir datos desde o hacia un servidor. Se utiliza para añadir la clave GPG [9] del repositorio de Grafana.

```
sudo apt-get install -y apt-transport-https curl
```

[7] Add user facilita la gestión de usuarios.

[8] Libfontconfig1 asegura que las fuentes se manejen correctamente en el sistema.

[9] GPG, por sus siglas en ingles Privacy Guard, es una herramienta para el cifrado y la firma digital de información. Es de código abierto, implementa el estándar OpenPGP, proporcionando un método confiable para la encriptación de comunicaciones y la seguridad de datos.

Se añadió la clave GPG del repositorio de Grafana con la instrucción:

```
curl https://packages.grafana.com/gpg.key | sudo apt-key add -
```

También se añade el repositorio de Grafana a la lista de repositorios, como se observa en la figura 3.36:

```
echo "deb https://packages.grafana.com/oss/deb stable main" | sudo tee -a /etc/apt/
sources.list.d/grafana.list
```

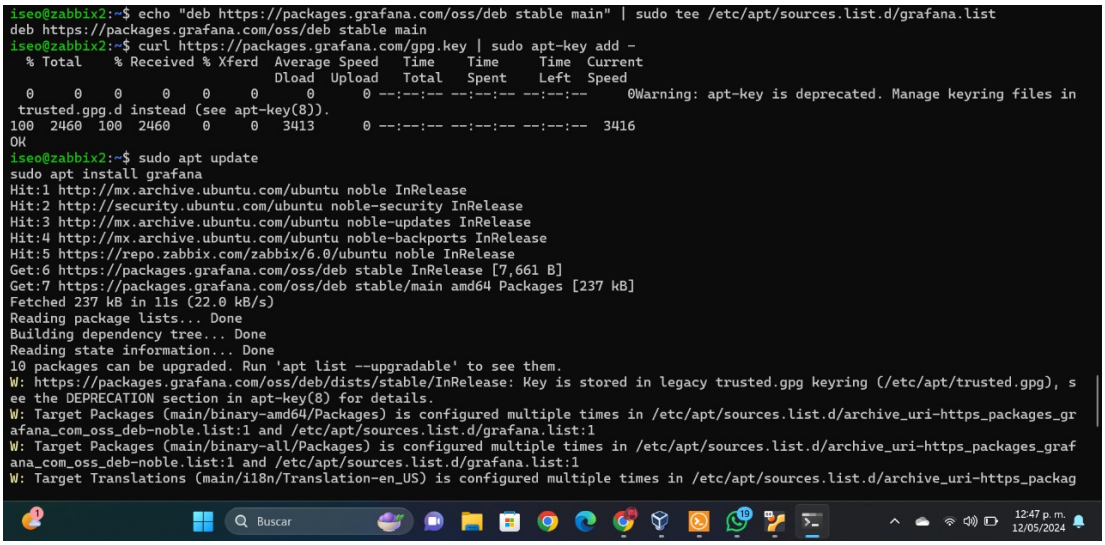


Figura 3.36: Repositorio Grafana.

Se actualizó el sistema nuevamente para que reconozca el nuevo repositorio:

```
sudo apt-get update
```

3.9.2. Instalación de Grafana

Para la instalación de Grafana, se ejecutó la instrucción:

```
sudo apt-get install grafana
```

Una vez instalado, se debe iniciar el servicio con la siguiente instrucción:

```
sudo systemctl start grafana-server
```

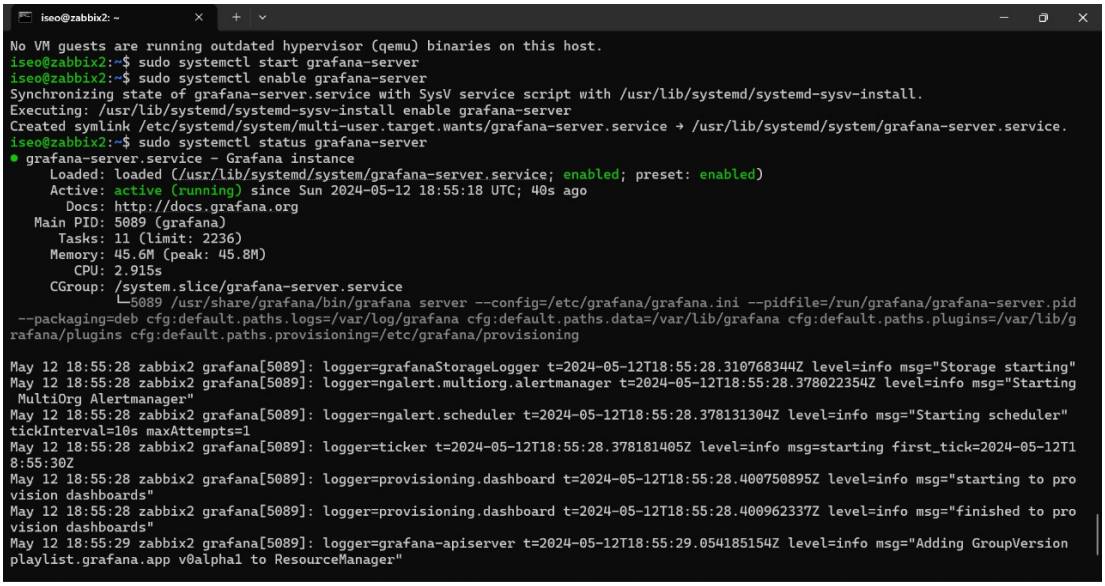


Figura 3.37: Inicialización de Grafana.

Para iniciar Grafana automáticamente al arrancar el sistema, se habilita con la instrucción:

```
sudo systemctl enable grafana-server
```

Al finalizar el proceso de instalación y configuración de Grafana, es posible acceder a la interfaz gráfica mediante un navegador web en otro servidor. Para ello, basta con introducir la dirección IP 192.168.1.30 del servidor donde se instaló Grafana, seguida del puerto 3000. La dirección correcta sería http://192.168.1.30:3000. Al introducir esta dirección en el navegador, se redirige a la página de inicio de sesión de Grafana. Los datos predeterminados para el inicio de sesión son 'admin' tanto para el nombre de usuario como para la contraseña. Tras iniciar sesión, se puede comenzar a explorar las distintas funcionalidades que Grafana ofrece para la visualización de datos [23].

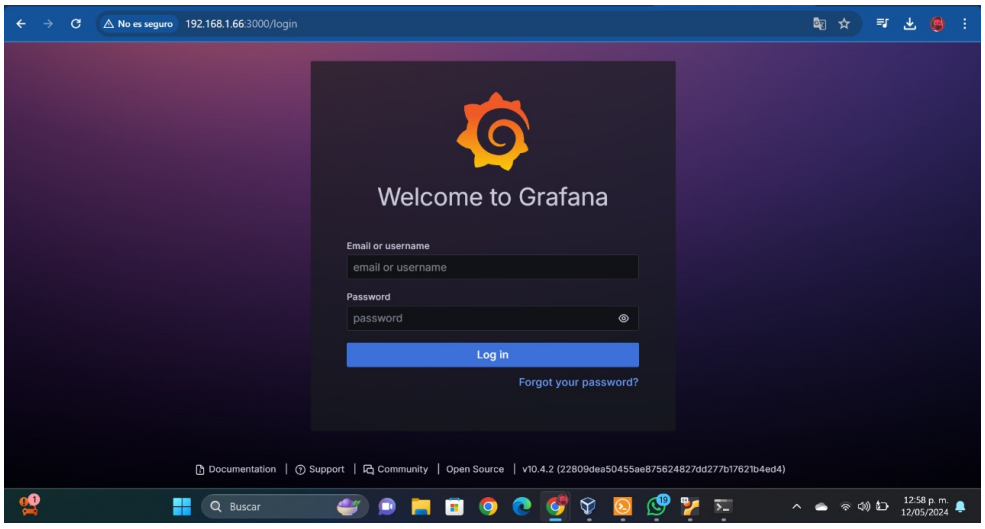


Figura 3.38: Interfaz gráfica de Grafana.

Realizar esta integración entre Grafana y Zabbix permite complementar las capacidades de monitoreo de ambos sistemas, aprovechando las características de cada software para lograr una recopilación de información más completa y funcional.

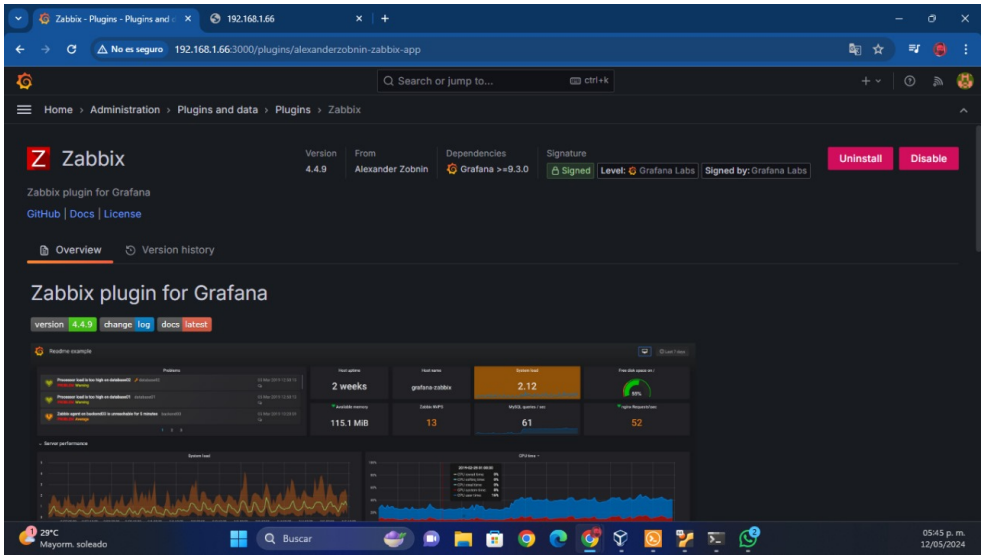


Figura 3.39: Interfaz gráfica de Grafana, para la integración de Zabbix.

3.10. Integración de Zabbix en Grafana

A continuación, se describen los pasos para integrar Zabbix en Grafana:

1. **Instalación de Grafana**: Una vez instalado Grafana, se accede a su interfaz web mediante la dirección IP y el puerto configurado, utilizando un navegador web. Esto permite interactuar con el panel de control de Grafana.

2. **Instalación de Complementos**: En el panel de control de Grafana, se debe seleccionar la opción **Configuration** y luego **Plugins**. Esto abre una serie de opciones para agregar complementos a Grafana, donde se busca y selecciona el complemento de Zabbix.

3. **Configuración de Zabbix como DataSource**: Una vez instalado el complemento de Zabbix, se procede a configurarlo como una fuente de datos (DataSource) en Grafana. En esta sección de configuración, se debe ingresar un nombre para la fuente de datos (campo **Name**) y la URL o dirección IP de Zabbix junto con el puerto correspondiente (campo **Connection**), lo que establece la conexión entre Grafana y Zabbix a través de la API.

4. **Exploración y Visualización de Datos**: Con la conexión establecida correctamente, se pueden explorar los datos de Zabbix y visualizar diferentes paneles de información. Esto permite crear dashboards personalizados para monitorear y analizar en tiempo real los datos provenientes de Zabbix de manera efectiva.

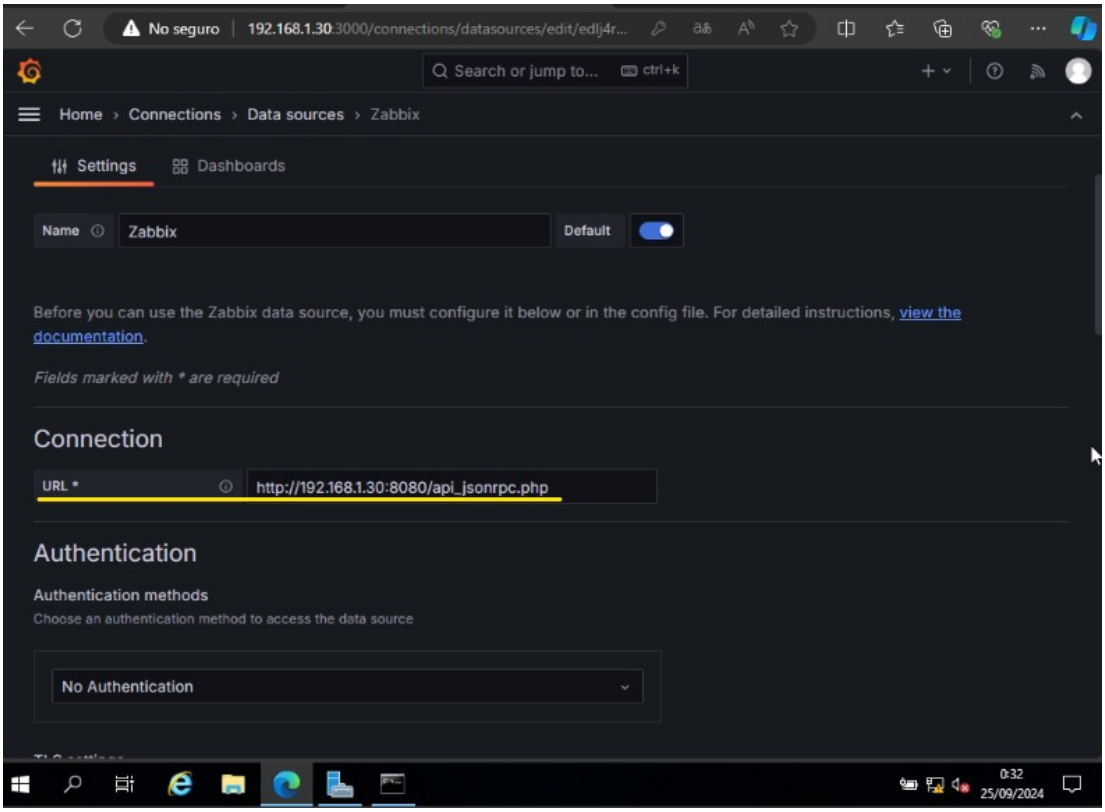


Figura 3.40: Integración de Zabbix a Grafana mediante URL.

Además de la integración con Grafana, también configuramos Zabbix para enviar alertas por correo electrónico. Esto nos permitió recibir notificaciones inmediatas al detectar un incidente; por ejemplo, cuando un valor excede cierto umbral, lo que activa el envío de la alerta por correo electrónico [30].

La integración de Grafana y Zabbix proporciona una solución de monitoreo completa y efectiva, que nos permite mantener los servidores en funcionamiento de manera óptima.

3.11. Configuración de notificaciones Telegram en Zabbix

Para mejorar la capacidad de monitoreo y alerta de Zabbix, es posible configurar notificaciones a través de Telegram. Telegram es una aplicación de mensajería instantánea que permite enviar mensajes rápidos y seguros, lo que la convierte en una excelente opción para recibir alertas de monitoreo en tiempo real [36].

A continuación, se presentan los pasos necesarios para esta configuración, garantizando que las alertas críticas lleguen directamente a un dispositivo móvil o a una computadora a través de Telegram.

1. **Creación de bot en Telegram:** Para establecer un bot en Telegram, es necesario iniciar una conversación con BotFather desde la aplicación telegram siguiendo las instrucciones para generar un nuevo bot. Una vez completado el proceso, se te otorgará un token HTTP API que deberás conservar, ya que será esencial para ajustar los parámetros dentro de Zabbix más adelante.

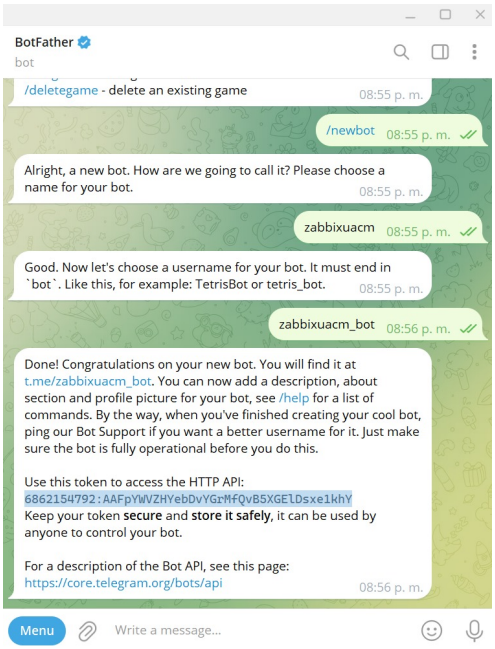


Figura 3.41: Creación de boot en Telegram

2. **Obtener ID de Telegram:** Para adquirir el ID de Telegram, se debe crear un grupo y añadir el bot creado en el paso anterior. Posteriormente, se incorpora a MyIDBot al mismo grupo para obtener el ID único del grupo.

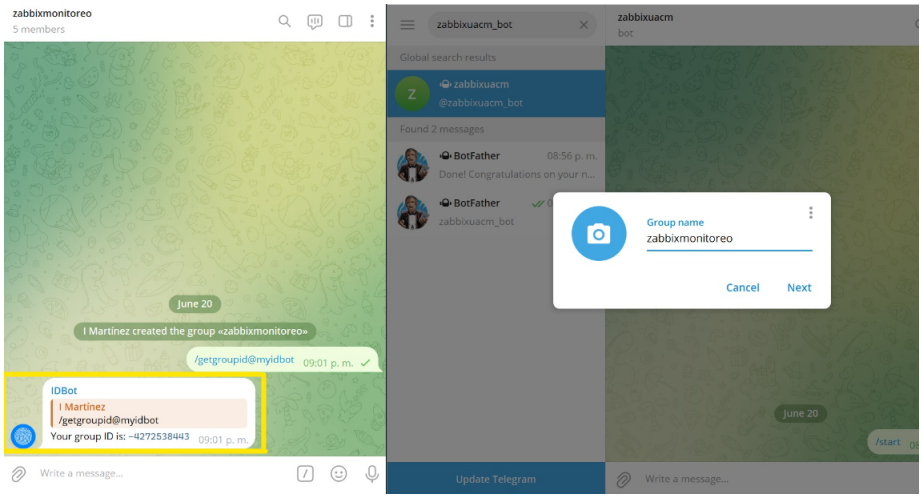


Figura 3.42: Creación de grupo y ID en Telegram

3. **Configuración del tipo de medio en Zabbix:** En cuanto a la definición del tipo de medio en Zabbix, se accede al menú de administración y en el apartado de 'Media Types' se establece el medio correspondiente a Telegram utilizando el token y el ID del grupo que se generó previamente.

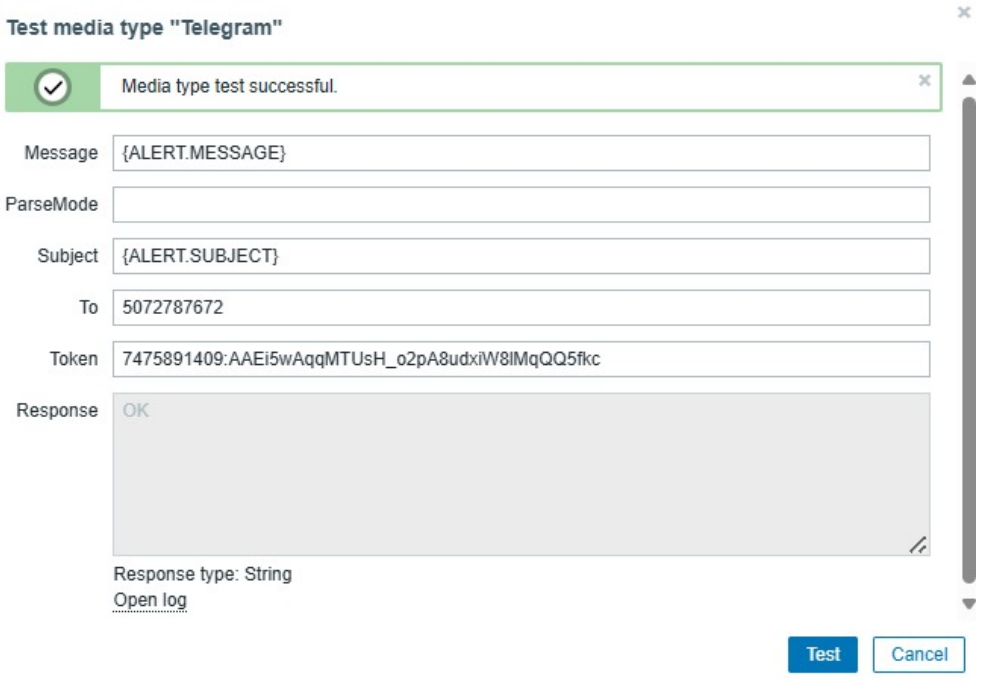


Figura 3.43: Creación de medio en Telegram

4. **Configuración de usuarios en Zabbix:** Para asignar el medio al usuario administrador y determinar las acciones en Zabbix, se debe definir el medio para el usuario admin y designa el tipo de medio de Telegram con el ID del grupo. Acto seguido, formular una acción de alerta en la configuración de Zabbix para fijar las condiciones y las operaciones de notificación para el envío de mensajes a través de Telegram.

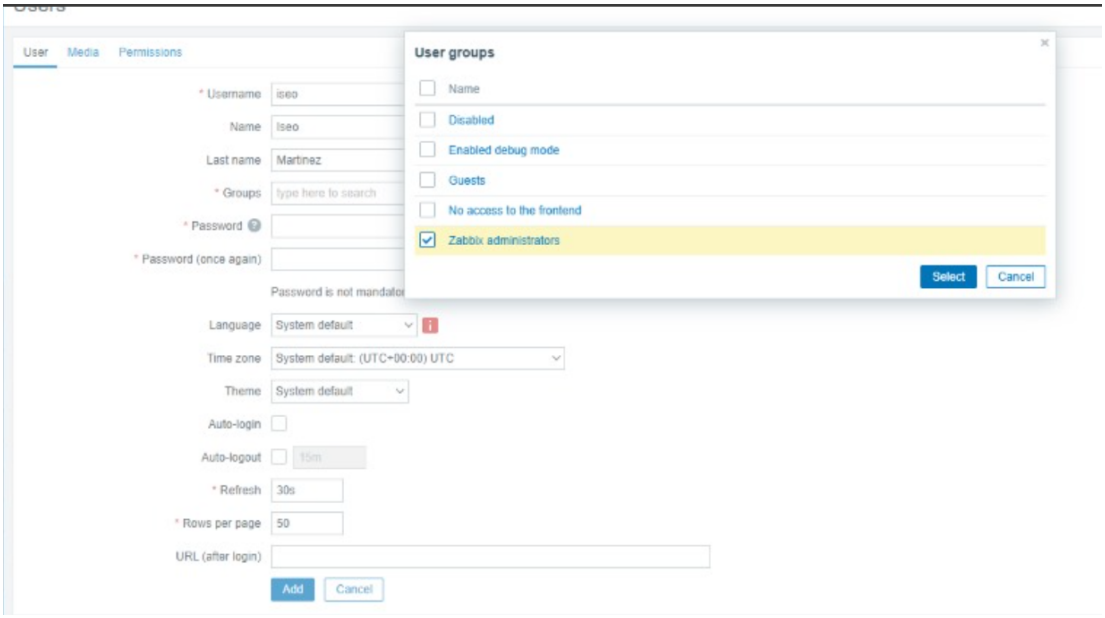


Figura 3.44: Configuración de usuarios en Zabbix

5. **Pruebas de configuración:** Finalmente, para verificar la integración, ejecuta una prueba que active una alerta y verifica que los mensajes se reciban correctamente en el grupo de Telegram, lo que confirmará el funcionamiento adecuado de la prueba efectuada.

3.12. Instalación y configuración de Zabbix en Raspberry Pi

La implementación de Zabbix en una Raspberry Pi se presenta como una solución de monitoreo confiable, de rápido despliegue y accesible, que ofrece beneficios significativos en términos de rendimiento y eficiencia. Esto permite aprovechar el bajo consumo energético (menos de 5 W) y el tamaño compacto de la Raspberry Pi, convirtiéndola en una opción económica para utilizar como hardware de monitoreo en una LAN.

En Zabbix, se pueden configurar diversos parámetros de monitoreo, lo que optimiza la gestión de recursos y mejora la eficiencia operativa. Además, el uso de la Raspberry Pi como servidor de Zabbix permite a los usuarios experimentar con la infraestructura de monitoreo sin incurrir en altos costos, lo que fomenta la adopción de prácticas de monitoreo y la mejora continua en la gestión de sistemas [30].

La instalación del servidor de monitoreo Zabbix se llevó a cabo utilizando la misma arquitectura empleada en la emulación, dado que el sistema operativo Raspbian es una distribución de Linux. Por esta razón, se omitirán las instrucciones de instalación y configuración, ya que se han mencionado anteriormente.

La versión del sistema operativo Raspbian con el que opera la Raspberry Pi es la versión 12, que se puede verificar ejecutando la siguiente instrucción en la terminal [37]:

```
cat /etc/os-release
```

La figura 3.45 muestra la ejecución de la instrucción antes mencionada, la cual nos proporciona la versión del software instalado y asegura que estamos trabajando con un entorno compatible para la implementación de Zabbix.

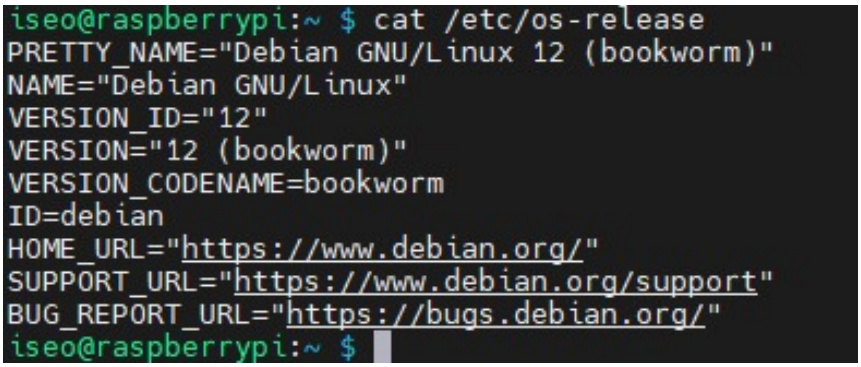


Figura 3.45: Ejecución de la instrucción para verificar la versión de Raspbian.

La implementación de Zabbix en una Raspberry Pi nos brinda la posibilidad de monitorear una red LAN con una tecnología de bajo costo, efectiva y segura. Esto permite una gestión más eficiente de los recursos de la red y ofrece una mayor visibilidad sobre el estado de los dispositivos conectados [30].

Instalación del servidor, la interfaz y el agente de Zabbix

Como ya se mencionó anteriormente, la instalación es una réplica de la realizada en la máquina virtual de la emulación. Por simplicidad, no se presentan los pasos ni la ejecución de las instrucciones; solo se verifican los recursos instalados en la Raspberry con las siguientes instrucciones [37]:

La figura 3.46 muestra información sobre la versión del servidor Zabbix instalada en el sistema mediante la instrucción.

```
zabbix_server --version
```

Esta instrucción permite conocer la versión exacta del servidor, que en este caso es la 7.0.0, y verificar la revisión específica y la fecha de compilación. También permite identificar la licencia bajo la cual se distribuye el software, que es la GNU Affero General Public License (AGPL) versión 3, otorgando libertad para modificar y redistribuir el programa bajo los términos de dicha licencia. Además, la instrucción permite ver con qué versión de OpenSSL se compiló el servidor, en este caso la 3.0.9, y con qué versión está funcionando actualmente, que es la 3.0.13 [30].

Figura 3.46: Ejecución de la instrucción para verificar la versión de Zabbix dentro del sistema de Raspberry.

Después de validar la versión del servidor Zabbix, es importante verificar que el agente de Zabbix se haya instalado y esté funcionando correctamente con la siguiente instrucción:

```
sudo systemctl status zabbix-agent
```

que muestra el estado detallado del servicio del agente. Al ejecutar esta instrucción, se debe observar que el estado aparezca en color verde y como ejecutando, lo cual indica que el servicio está cargado y en ejecución. Además, el estado activo confirma que el agente está configurado para iniciarse automáticamente con el sistema. Esta información valida que la instalación y configuración del agente fueron exitosas [30].

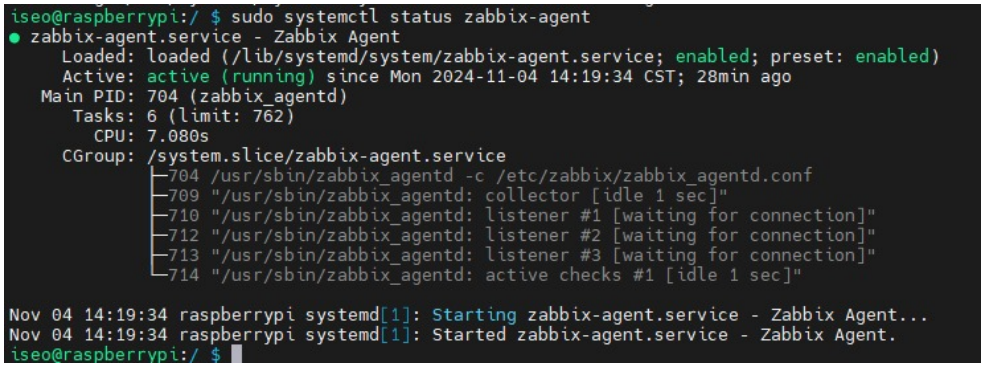


Figura 3.47: Estatus de Zabbix en Raspberry.

Configuración de una dirección IP en Raspberry Pi

La configuración de una dirección IP estática en la Raspberry Pi es fundamental para optimizar su funcionamiento dentro del laboratorio en el que se ubicará físicamente para monitorear la red. Al asignar una IP fija, se garantiza una conexión estable y se facilita el acceso remoto, el descubrimiento de redes y la configuración de servicios. Asimismo, la implementación de medidas de seguridad se simplifica, y la administración del dispositivo se vuelve más eficiente [37].

La Raspberry Pi cuenta con dos interfaces activas: wlan0 (inalámbrica) y eth0 (Ethernet). Solo la interfaz eth0 tiene asignada una dirección IPv4 (172.17.133.250). Esta dirección IP se configuró específicamente en la Raspberry Pi para proporcionar acceso a Internet y permitir conexiones remotas a través de SSH, facilitando así la instalación de herramientas como Zabbix y Grafana. Además, la IP 172.17.133.250 pertenece al mismo segmento de red que los demás equipos del laboratorio, lo que garantiza que todos los dispositivos puedan comunicarse entre sí sin problemas dentro de la misma subred (172.17.133.0/24) [37].

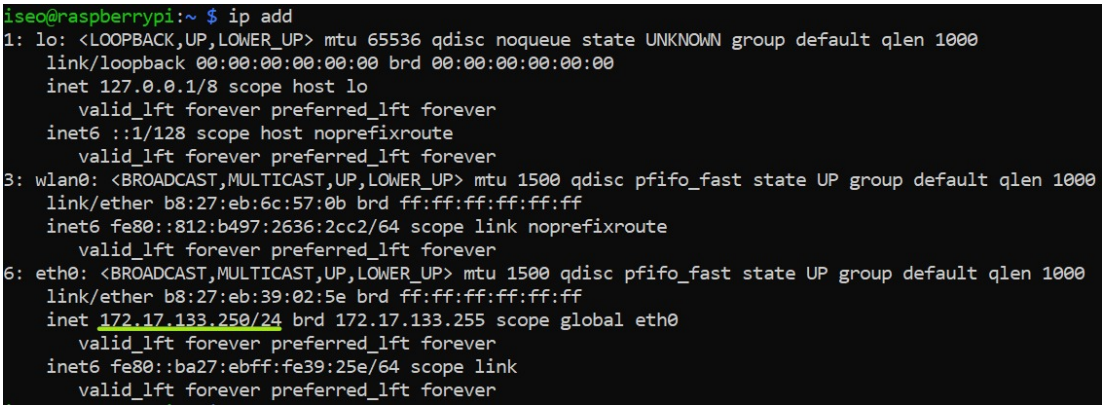


Figura 3.48: Configuración de IP 172.17.133.250

Una vez configurada la IP estática 172.17.133.250 en la Raspberry Pi, se procedió a comprobar la conexión mediante SSH. La conexión se realizó con éxito, permitiendo el acceso remoto al dispositivo, como se muestra en la figura 3.49.

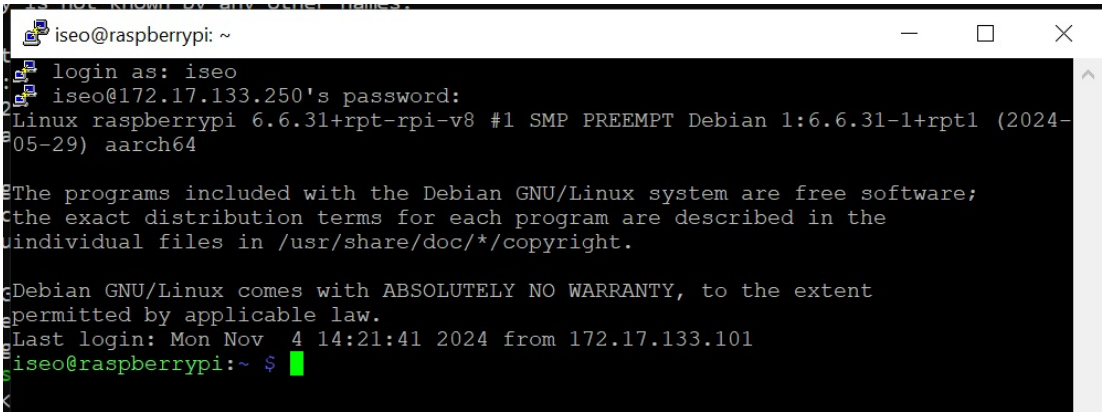


Figura 3.49: Comprobación de conexión remota mediante SSH.

3.12.1. Configuración de IPv4 estática en Raspberry Pi con el segmento de Red del Laboratorio LACECI

Para configurar una IP estática en la Raspberry Pi, es necesario realizar ciertas configuraciones utilizando la terminal. En primer lugar, se debe abrir el archivo */etc/network/interfaces* con el editor de texto *nano*. Por ejemplo [37]:

sudo nano /etc/network/interfaces

Una vez abierto el archivo, se deben agregar las siguientes líneas [37]:

- *auto eth0*
- *iface eth0 inet static*
- *address 172.17.133.250*
- *netmask 255.255.255.0*
- *gateway 172.17.133.1*
- *dns-nameservers 8.8.8.8 8.8.4.4*

como se muestra en la figura 3.50

```
iseo@raspberrypi:~ $ cd /etc/network
iseo@raspberrypi:/etc/network $ cat interfaces
# interfaces(5) file used by ifup(8) and ifdown(8)
# Include files from /etc/network/interfaces.d:
source /etc/network/interfaces.d/*
auto eth0
iface eth0 inet static
    address 172.17.133.250
    netmask 255.255.255.0
    gateway 172.17.133.254
    dns-nameservers 172.17.102.14
iseo@raspberrypi:/etc/network $
```

Figura 3.50: Configuración de IP estática.

Para entender la configuración, es importante mencionar lo siguiente:

- *auto eth0* activa la interfaz de ethernet.
- *iface eth0* inet static configura la IP estática.
- *address* es la IP estática.
- *netmask* es la máscara de subred.
- *gateway* es la IP del enrutador.
- *dns-nameservers* son los servidores DNS.

Después de realizar estas modificaciones, es importante guardar los cambios en el archivo. Posteriormente, se debe reiniciar el servicio de red con la siguiente instrucción [37]:

sudo systemctl restart networking

Esta instrucción permite reiniciar el servicio de red y validar que la dirección IP configurada se aplique correctamente.

Paquetes instalados para la implementación de Zabbix en raspberry Pi.

La figura 3.51 muestra la salida de la instrucción.

```
dpkg -l | grep Zabbix
```

que lista los paquetes relacionados con Zabbix instalados en el sistema. Los paquetes instalados corresponden a:

1. **Zabbix-agent:** El agente de Zabbix.
2. **Zabbix-frontend-php:** La interfaz web en PHP.
3. **Zabbix-nginx-conf:** Configuraciones para Nginx y PHP-FPM.
4. **Zabbix-release:** El paquete de configuración del repositorio de Zabbix.
5. **Zabbix-server-pgsql:** El servidor de Zabbix para PostgreSQL.
6. **Zabbix-sql-scripts:** Scripts SQL utilizados por Zabbix.

Los paquetes incluyen tanto el servidor como el agente, junto con las herramientas necesarias para la interfaz web y la integración con PostgreSQL [30].

```
iseo@raspberrypi:~ $ dpkg -l | grep zabbix
ii  zabbix-agent                  1:7.0.0-1+debian12              arm64        Zabbix net
work monitoring solution - agent
ii  zabbix-frontend-php            1:7.0.0-1+debian12              all          Zabbix net
work monitoring solution - PHP front-end
ii  zabbix-nginx-conf              1:7.0.0-1+debian12              all          Zabbix net
work monitoring solution - nginx and php-fpm configuration for front-end
ii  zabbix-release                 1:7.0-2+debian12                all          Zabbix off
icial repository configuration
ii  zabbix-server-pgsql            1:7.0.0-1+debian12              arm64        Zabbix net
work monitoring solution - server (PostgreSQL)
ii  zabbix-sql-scripts             1:7.0.0-1+debian12              all          Zabbix net
work monitoring solution - sql-scripts
iseo@raspberrypi:~ $
```

Figura 3.51: Paquetes instalados para implementar Zabbiz en Raspberry.

Una vez que se obtiene acceso exitoso a la interfaz de Zabbix a través del navegador, se procede a la instalación de tres agentes de monitoreo Zabbix en tres equipos del laboratorio, todos con sistema operativo Linux. Estos servidores se configuran con los siguientes nombres de host:

Tabla 3.4: Asignación de hostname a los equipos del laboratorio

Hostname	Dirección IP
EquipoLaceci	172.17.133.34
EquipoLaceci2	172.17.133.36
EquipoLaceci3	172.17.133.32

Esta configuración permitió iniciar el monitoreo de cada uno de estos equipos a través de la plataforma Zabbix.

Al igual que se configura Zabbix en la Raspberry Pi, también se realizan las configuraciones necesarias en los equipos del laboratorio para que puedan ser monitoreados correctamente. Dentro del archivo de configuración del agente de Zabbix en cada uno de los equipos Linux, se modifican varios parámetros clave. Se ajusta el valor de **Server** para que apunte a la dirección IP de la Raspberry Pi, 172.17.133.250, donde se encuentra Zabbix. Además, se configura el puerto de escucha con **ListenPort**=10050, que es el puerto predeterminado para el agente de Zabbix. Asimismo, se establece **ServerActive**=172.17.133.250 para permitir la comunicación activa con el servidor. Finalmente, se asigna el nombre de host correspondiente a cada equipo, colocando **Hostname**=EquipoLaceci, de modo que cada dispositivo sea identificado correctamente dentro del sistema de monitoreo.

```
# Mandatory: yes, if StartAgents is not explicitly set to 0
# Default:
# Server=

Server=172.17.133.250

### Option: ListenPort
#       Agent will listen on this port for connections from the server.
#
# Mandatory: no
# Range: 1024-32767
# Default:
ListenPort=10050
#
# Mandatory: no
# Default:
# ServerActive=

ServerActive=172.17.133.250

### Option: Hostname
#       List of comma delimited unique, case sensitive hostnames.
#       Required for active checks and must match hostnames as configured on the server.
#       Value is acquired from HostnameItem if undefined.
#
# Mandatory: no
# Default:
# Hostname=

Hostname=EquipoLaceci

### Option: HostnameItem
#       Item used for generating Hostname if it is undefined. Ignored if Hostname is defined.
#       Does not support UserParameters or aliases.
#
# Mandatory: no
# Default:
# HostnameItem=system.hostname
```

Figura 3.52: Archivo de configuración del agente Zabbix.

3.12.2. Instalación y Configuración de Grafana para Monitoreo con Zabbix

Una vez finalizada la configuración de Zabbix y sus agentes, se procede a la instalación y configuración de Grafana para la visualización avanzada de datos. En la figura 3.53 se muestra el estado del sistema de la Raspberry Pi, monitoreado a través de esta plataforma integrada, lo que permite obtener una visión más detallada de su rendimiento y funcionamiento.

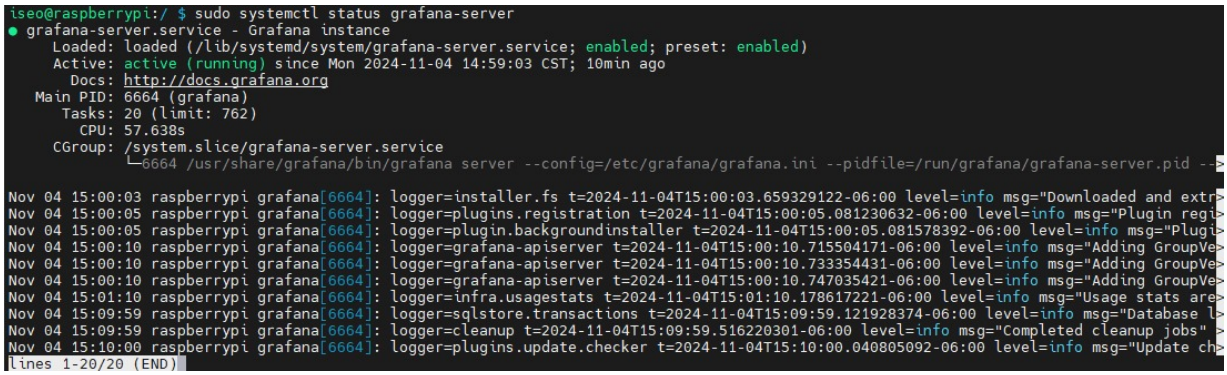


Figura 3.53: Estatus de Grafana instalado y configurado en la Raspberry Pi.

La implementación del sistema de monitoreo y gestión se lleva a cabo mediante una combinación de herramientas de monitoreo de software libre. La simulación de la topología en "GNS3" valida las configuraciones antes de aplicarlas en entornos reales, garantizando la funcionalidad y la conectividad mediante pruebas de comunicación entre las máquinas virtuales. Adicionalmente, la instalación y configuración de Zabbix, desde su base de datos en PostgreSQL hasta la integración con servicios como Grafana y Telegram, destaca por la flexibilidad y escalabilidad para cubrir necesidades de monitoreo en tiempo real.

Por otro lado, la implementación de SNMP y la configuración de traps en diversos dispositivos refuerzan la capacidad de supervisar infraestructuras. Es importante mencionar que la Raspberry Pi se integra como un nodo clave para aplicar el monitoreo mediante la instalación de agentes de Zabbix en las máquinas del laboratorio de "LAMAT", demostrando cómo las soluciones de uso libre se adaptan eficientemente a infraestructuras académicas.

Después de realizar todas las configuraciones necesarias en la Raspberry Pi, incluyendo la instalación y configuración de Zabbix y Grafana, así como la asignación de una dirección IP estática en el segmento de red del laboratorio, se procedió a implementar físicamente la Raspberry Pi en el laboratorio LAMAT.

La figura 3.54 muestra la conexión de la Raspberry Pi a un conmutador del laboratorio LAMAT. Este mismo conmutador también está conectado a las estaciones de trabajo que se pueden observar al fondo de la imagen.

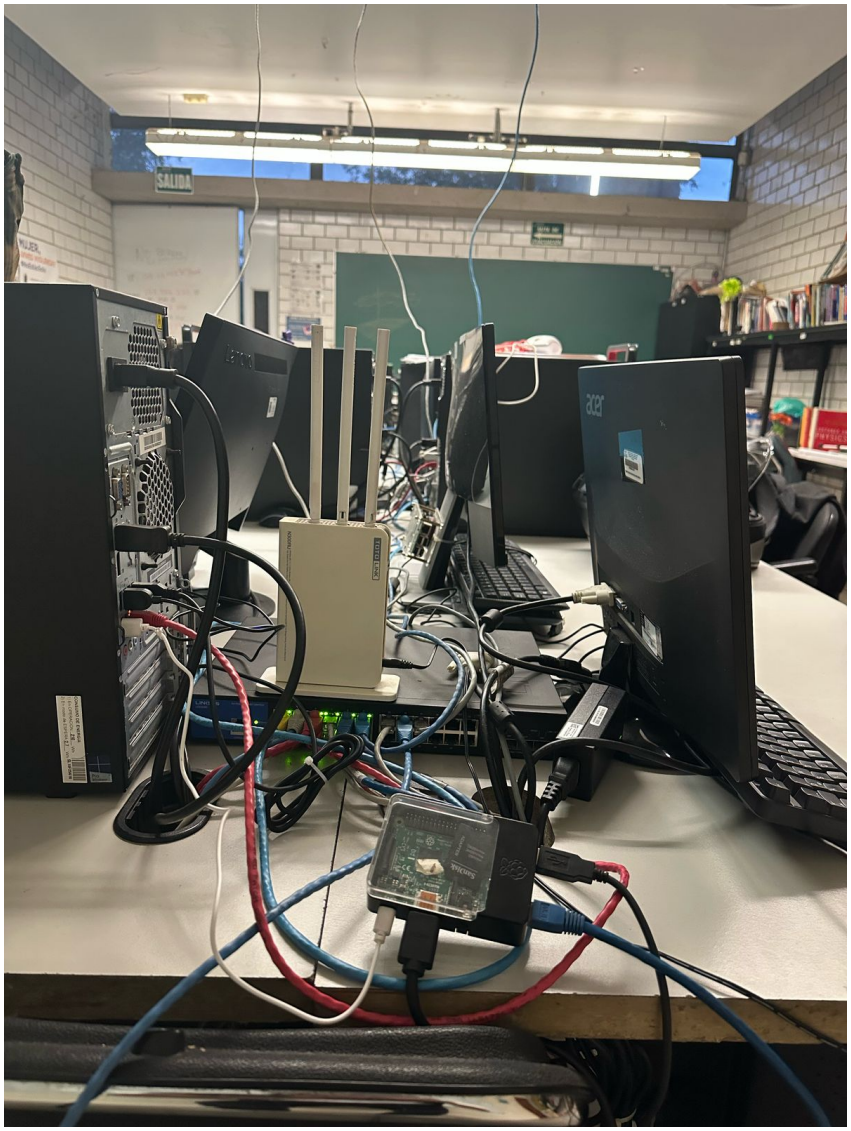


Figura 3.54: Implementación de la Raspberry Pi en el laboratorio LAMAT.

Una vez implementada la Raspberry Pi por medio de una estación de trabajo del laboratorio LAMAT, se puede observar la interfaz de Zabbix desde los equipos del laboratorio. A través de la consola, se verifica el correcto funcionamiento de Zabbix. La siguiente figura 3.55 muestra que Zabbix está activo y funcionando correctamente desde la Raspberry Pi.

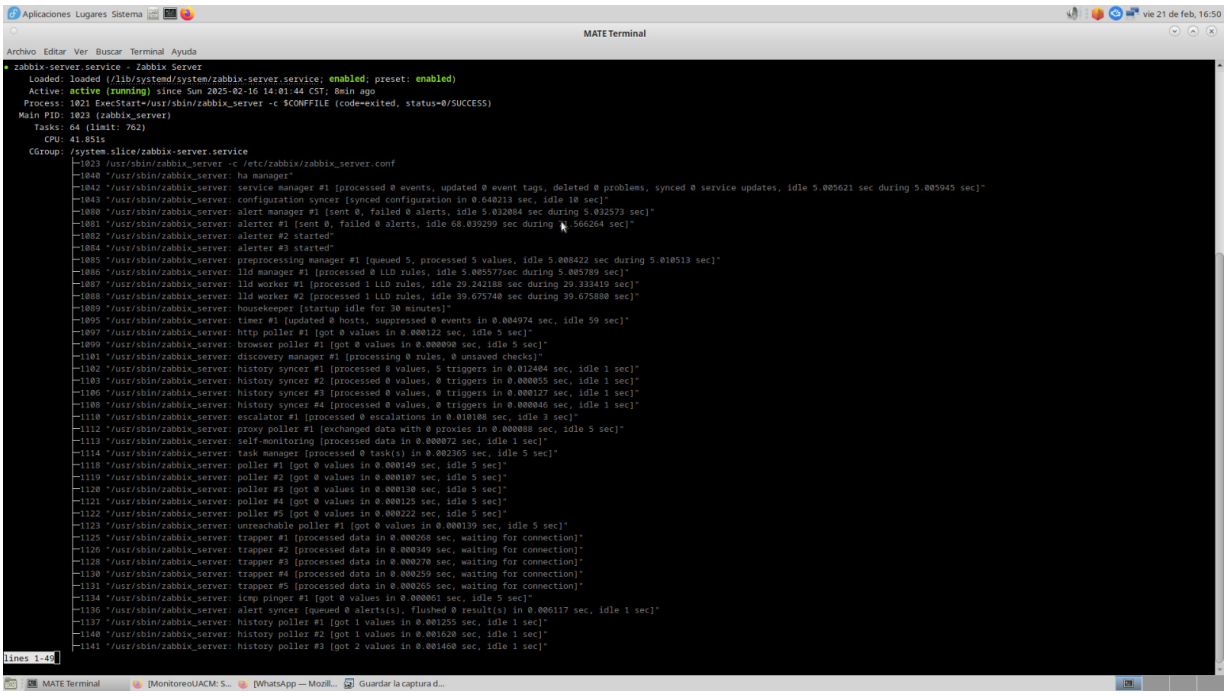


Figura 3.55: Estado actual del servicio Zabbix en ejecución desde la Raspberry Pi, mostrando procesos activos y consumo de recursos en tiempo real.

En este capítulo, se ha realizado con éxito la simulación de la red, replicando el entorno real de trabajo de la LAN de los laboratorios. Se llevó a cabo la instalación y configuración del servidor de monitoreo, así como la integración de Grafana con Zabbix para la visualización de datos. Además, se implementaron notificaciones tanto en el entorno simulado como en la implementación física, utilizando la Raspberry Pi como dispositivo de monitoreo. Esta aproximación ha permitido verificar el funcionamiento de la infraestructura en ambos entornos, asegurando su efectividad.

En el siguiente capítulo, se realizará primero el análisis de los resultados obtenidos en el ambiente simulado, para posteriormente abordar el análisis de los resultados de la Raspberry Pi implementada en los laboratorios. Este enfoque permitirá evaluar el rendimiento y la efectividad del sistema tanto en un entorno controlado como en uno físico.

Capítulo 4

Análisis de resultados

4.1. Descubrimiento de hosts

En la figura 4.1 se muestra un listado de los hosts detectados a través de la regla de descubrimiento ejecutada por el servidor Zabbix dentro del entorno de simulación en GNS3. En la columna "availability" se observa que algunos hosts presentan un estado en rojo, indicando que, aunque se detecta la interfaz de red, estos no cuentan con el agente de Zabbix instalado.

Esta situación se debe a que las máquinas virtuales predeterminadas de GNS3 están diseñadas principalmente para la simulación de dispositivos de red, sin contar con un sistema operativo completo. Como resultado, presentan limitaciones en cuanto a recursos y configuraciones, lo que dificulta la instalación de software adicional, como el agente de monitoreo de Zabbix.

Además, las imágenes predeterminadas proporcionadas en GNS3 no incluyen las bibliotecas y dependencias necesarias para la ejecución de herramientas de monitoreo, lo que impide la recopilación de métricas detalladas sobre el uso de CPU, memoria y otros parámetros esenciales del sistema. Esto representa una barrera para la supervisión en tiempo real de estos hosts dentro del entorno de simulación.

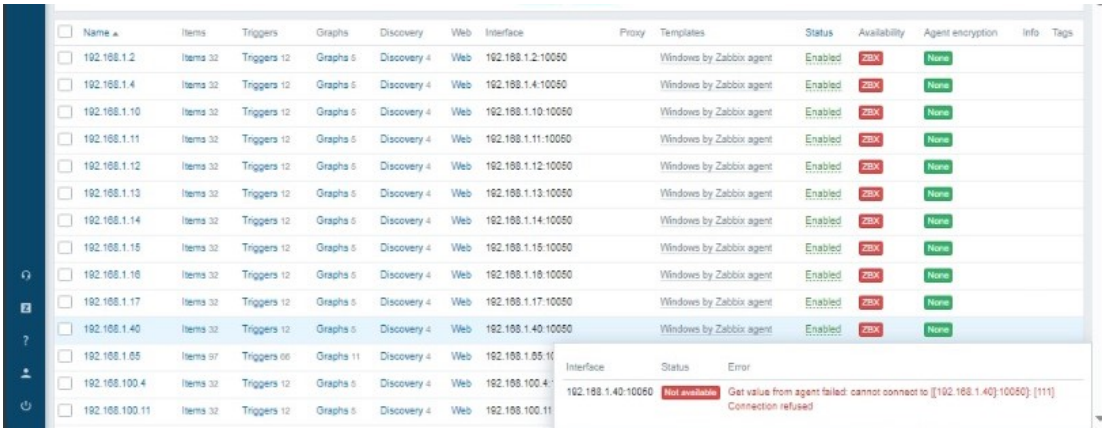


Figura 4.1: Máquinas virtuales sin la instalación de un agente Zabbix

Por lo tanto, para contrastar el monitoreo entre un host con agente y otro sin agente, se realizó la instalación del agente de monitoreo de Zabbix en una máquina virtual con el sistema operativo Windows Server 2019, garantizando así la compatibilidad con el agente de Zabbix. Como se muestra en la figura, cuando una máquina virtual cuenta con el agente de Zabbix instalado, el estado de "availability" cambia a color verde, indicando que el host está correctamente monitoreado por el agente de Zabbix. La tabla 4.1 compara algunas métricas destacables entre equipos que cuentan con agente y aquellos que no.

La implementación de un agente de monitoreo en un host permite una recolección de datos más detallada y precisa en comparación con un host que no tiene agente. En el caso de un host con el agente de Zabbix instalado, es posible monitorear métricas avanzadas del sistema operativo, como el uso de CPU, memoria, almacenamiento, actividad de red y registros de eventos, lo cual permite una supervisión integral del rendimiento y la detección temprana de problemas. Estos datos se presentan de manera estructurada y detallada en la interfaz de Zabbix, con alertas configurables basadas en umbrales específicos.

Tabla 4.1: Comparación de monitoreo con y sin agente de Zabbix

Métrica	**Con agente**	**Sin agente**
Uso de CPU	Sí	No
Uso de memoria	Sí	No
Estado del servicio HTTP	Sí	Parcial
Latencia ICMP (Ping)	Sí	Sí
Registro de eventos	Sí	No

En contraste, para hosts que no cuentan con un agente, el monitoreo se restringe a datos que son externamente accesibles, como la disponibilidad de red (ICMP), la condición de los puertos y la respuesta de servicios particulares. Esto es útil para comprobar la conectividad y el estado general del host, pero la información proporcionada es menos detallada y podría no descubrir problemas internos más a fondo.

Este enfoque de monitoreo sin agente es particularmente útil en dispositivos de red, como los equipos de Cisco. Dado que estos dispositivos no están diseñados para instalar software adicional, el monitoreo se realiza mediante SNMP o consultas de red específicas, lo que permite obtener métricas generales de rendimiento y disponibilidad. Sin embargo, para supervisar servidores y sistemas operativos completos (como en el caso de Windows Server 2019 con el agente de Zabbix instalado), la instalación del agente proporciona una cobertura de monitoreo más completa.

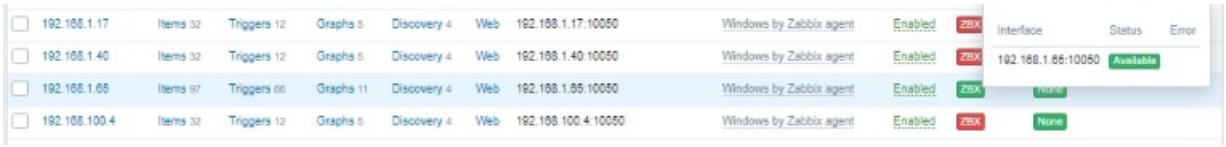


Figura 4.2: Máquina virtual con agente Zabbix instalado

En la figura 4.3, se presenta la tabla de **dispositivos descubiertos** en la red simulada, junto con información sobre el tiempo de actividad e inactividad de cada uno.

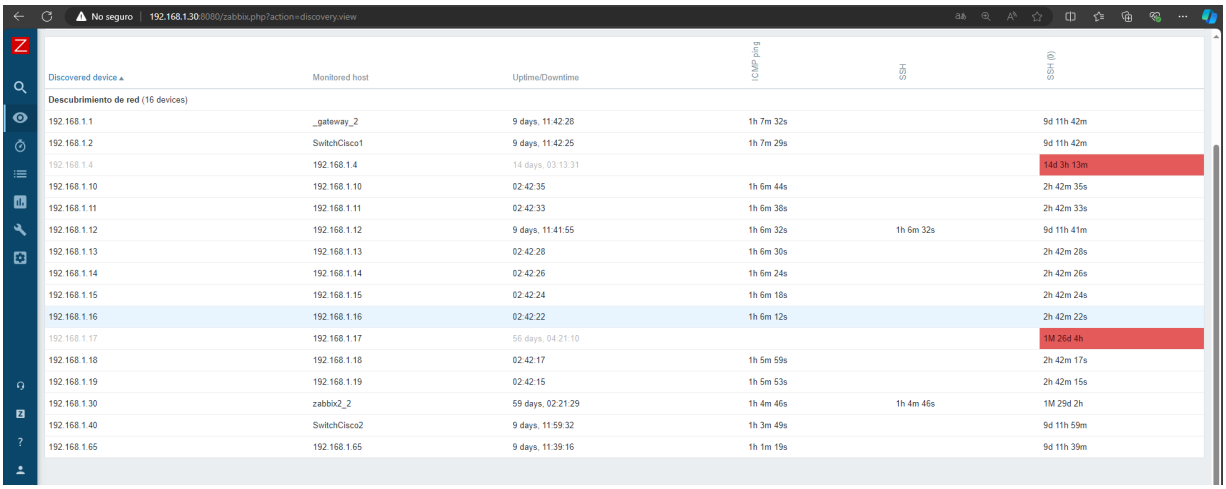


Figura 4.3: Dispositivos descubiertos en la simulación.

Análisis:

1. **Dispositivos y sus direcciones IP:**

   - Se listan un total de 16 dispositivos, con sus respectivas direcciones IP, como 192.168.1.1, 192.168.1.2, 192.168.1.4, entre otros.
   - La columna "Monitored host" muestra el nombre del host o dispositivo que Zabbix está monitoreando, por ejemplo, gateway2, conmutadorCisco1, entre otros.

2. **Tiempo de actividad y de inactividad**

   Dispositivos con conectividad estable:

   - Algunos dispositivos, como 192.168.1.1 (gateway) y 192.168.1.2 (conmutador Cisco 1), tienen un tiempo de actividad de 9 días, 11 horas y 42 minutos, lo que indica un funcionamiento constante y estable.
   - El dispositivo con dirección IP 192.168.1.12 también tiene un "uptime" similar de 9 días, 11 horas y 41 minutos.

Dispositivos con conectividad inestable o problemas:

- El dispositivo 192.168.1.4 tiene un tiempo de inactividad considerable, con 14 días, 3 horas y 13 minutos de inactividad, lo que sugiere que el dispositivo no está conectado o ha estado caído durante un período prolongado.
- Otro dispositivo con alta inactividad es Zabbix2_2 (con dirección IP 192.168.1.17), que ha estado inactivo durante 1 mes y 26 días.

3. **Dispositivos con tiempos de actividad bajos:**

- Varios dispositivos como 192.168.1.10, 192.168.1.11, y 192.168.1.13, entre otros, muestran tiempos de actividad más cortos, alrededor de 2 horas y 42 minutos, lo que podría indicar que se reiniciaron recientemente o que se conectaron recientemente a la red.

A continuación, se presenta la figura 4.4, que ilustra el descubrimiento de hosts desde una máquina virtual del laboratorio LAMAT, implementada con una Raspberry Pi. En esta imagen se destaca cómo se detectan y monitorean otras máquinas conectadas en el mismo segmento de red. En particular, se monitorean las máquinas conectadas mediante cable en el segmento de red 17.17.133.x, mientras que para la red inalámbrica se utiliza el segmento 192.168.0.x.

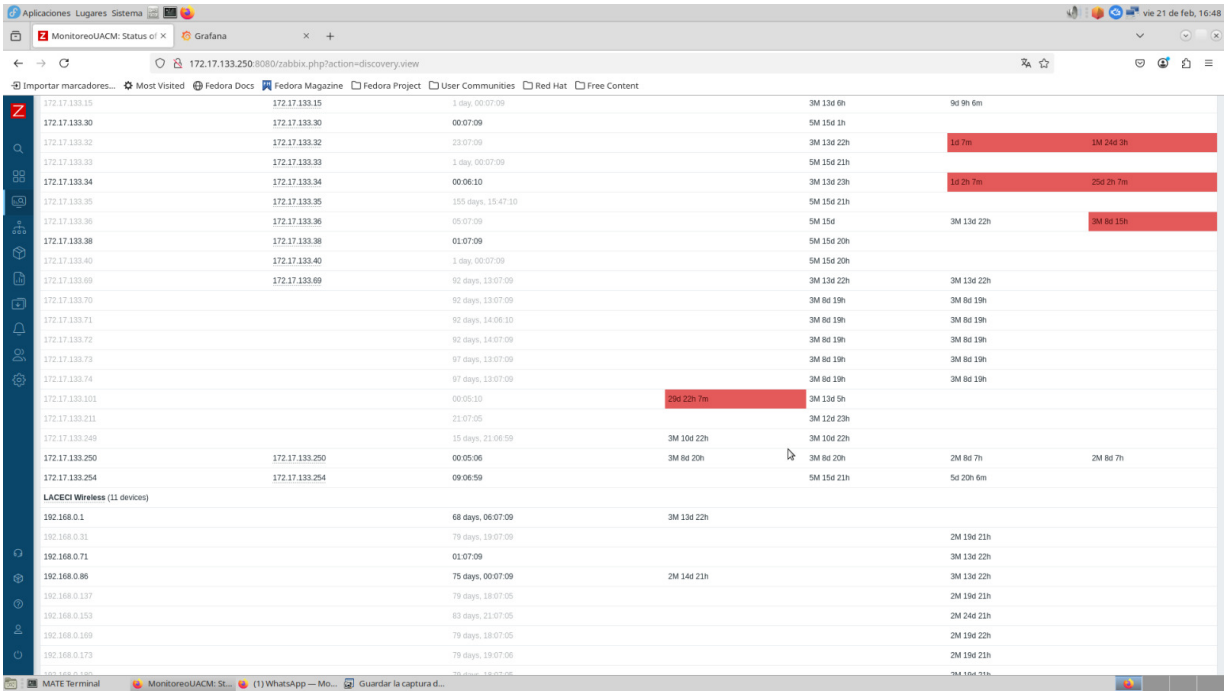


Figura 4.4: Descubrimiento de hosts desde una máquina virtual del laboratorio LAMAT.

4.2. Análisis de la topología de la red simulada en Zabbix

En este apartado, se muestra una representación de la red implementada en el simulador, destacando su estado en dos situaciones: cuando todos los dispositivos operan correctamente y cuando se presentan problemas de conectividad.

4.2.1. Topología de la Red en Estado Normal

La Figura 4.5 representa la topología de la red en Zabbix cuando todos los dispositivos están operativos y no se reportan problemas de conectividad. Esta simulación incluye diversos elementos clave, como PCs, conmutadores, enrutadores y servidores.

Los dispositivos sin problemas reportados son:

- Todos los dispositivos muestran un estado *OK*, indicando que están correctamente conectados y reportan su estado mediante el agente de Zabbix.
- Tanto los conmutadores como el enrutador aseguran la conectividad y el tráfico fluido entre los diferentes segmentos de la red.
- Los servidores (Linux y Windows) se encuentran operativos, lo que garantiza que los servicios de red funcionan sin interrupciones.

Estos estados permiten analizar el rendimiento de la red sin interferencias, lo que sirve como referencia para evaluar su estabilidad.

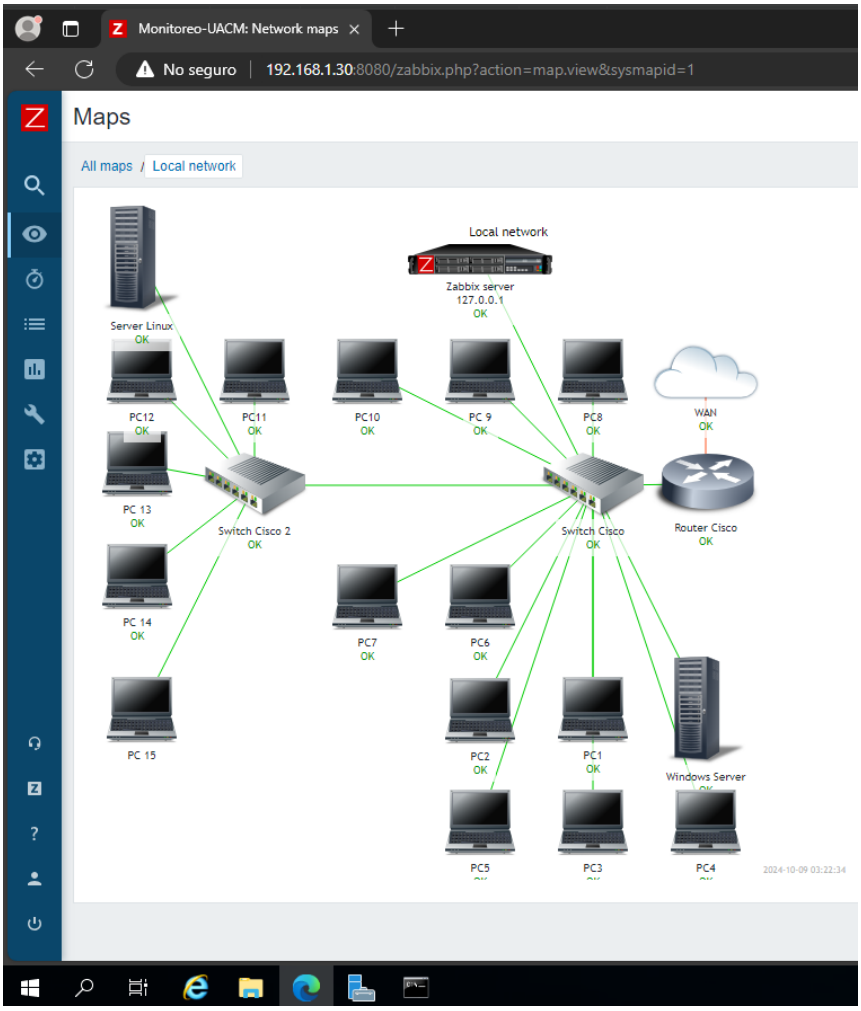


Figura 4.5: Topología de red simulada en Zabbix sin problemas reportados.

4.2.2. Topología de la Red con Problemas de Conectividad

La Figura 4.6 muestra la misma red simulada, pero en un estado en el que varios dispositivos presentan problemas de conectividad. En este escenario, se pueden observar alertas generadas por Zabbix debido a la falta de respuesta de algunos agentes en los dispositivos monitoreados.

Problemas detectados:

- Nueve PCs y el servidor Linux no responden al agente de Zabbix, mostrando la alerta "Zabbix agent is not available (for 3m)". Esto indica fallos de conectividad o que dichos dispositivos están inactivos.
- La interfaz `Fa1/1(30)` del conmutador principal Cisco se encuentra en estado "Link down", lo que provoca desconexiones en el segmento de red que conecta. Este problema

es crítico, ya que impacta en múltiples dispositivos que dependen de esta conexión.

- Algunos dispositivos, como el servidor Windows y varias PCs (PC3, PC4 y PC5), permanecen en estado *OK*, indicando que aún tienen conectividad y reportan correctamente.

Estos problemas simulan fallos comunes en redes reales, permitiendo probar la respuesta y el alcance de Zabbix para el diagnóstico y monitoreo de red. La identificación de estos problemas permite tomar decisiones informadas para restaurar la conectividad y asegurar un rendimiento óptimo.

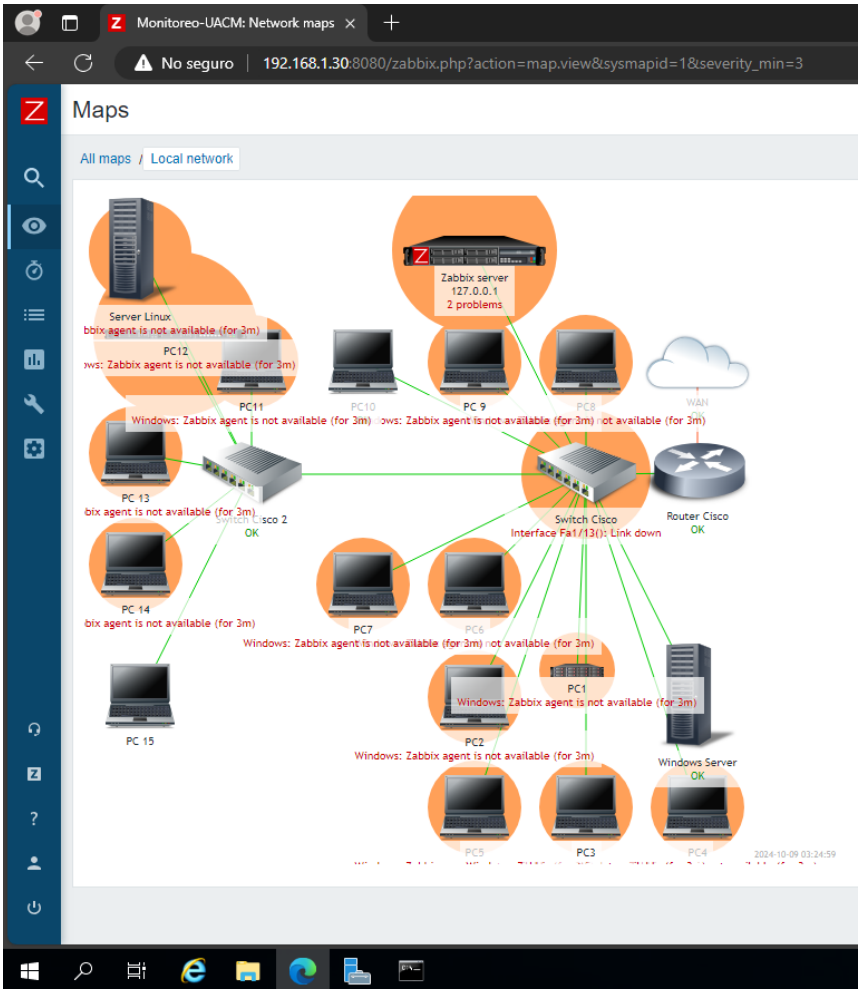


Figura 4.6: Topología de red simulada en Zabbix con problemas reportados.

Este análisis demuestra las capacidades de Zabbix para detectar y reportar fallos de red en tiempo real. Al simular tanto un estado de operación normal como uno con problemas, se evidenció cómo el software permite monitorear cada dispositivo, identificando rápidamente el origen del fallo y los dispositivos afectados.

Zabbix se destaca por su habilidad para diferenciar entre dispositivos conectados y desconectados, así como por su capacidad para notificar fallos en tiempo real, lo que facilita una respuesta rápida ante problemas en la red.

4.2.3. Limitaciones de las Máquinas Virtuales de GNS3

Durante la investigación, se encontró que las máquinas virtuales de GNS3 que vienen por defecto no permiten instalar agentes de monitoreo. Esto se debe a las siguientes razones:

a) **Configuración Predeterminada:** Las máquinas virtuales de GNS3 están configuradas para emular dispositivos de red y no están optimizadas para instalar software adicional como agentes de monitoreo.

b) **Recursos Limitados:** Estas máquinas virtuales suelen tener recursos limitados (CPU, memoria, almacenamiento), lo que dificulta instalar y ejecutar agentes de monitoreo que requieren más capacidad.

c) **Compatibilidad:** Los agentes de monitoreo pueden no ser compatibles con la configuración específica de las máquinas virtuales de GNS3.

4.2.4. Métricas Obtenidas

Servidor Ubuntu:

El servidor monitoreado por Zabbix presenta, en general, un buen desempeño, con estabilidad en el procesamiento de valores y en los procesos del sistema operativo. Sin embargo, el aumento en el uso del espacio de almacenamiento y la inactividad de la caché podrían requerir ajustes futuros para evitar posibles limitaciones de almacenamiento y mejorar la eficiencia de la memoria.

La siguiente figura 4.7 muestra el portal de monitoreo Zabbix. Se observan diversas métricas relacionadas con el rendimiento y uso de recursos del servidor supervisado por Zabbix, correspondientes a un análisis de desempeño en un período determinado, el cual puede ser modificado de acuerdo con las métricas que se necesiten visualizar.

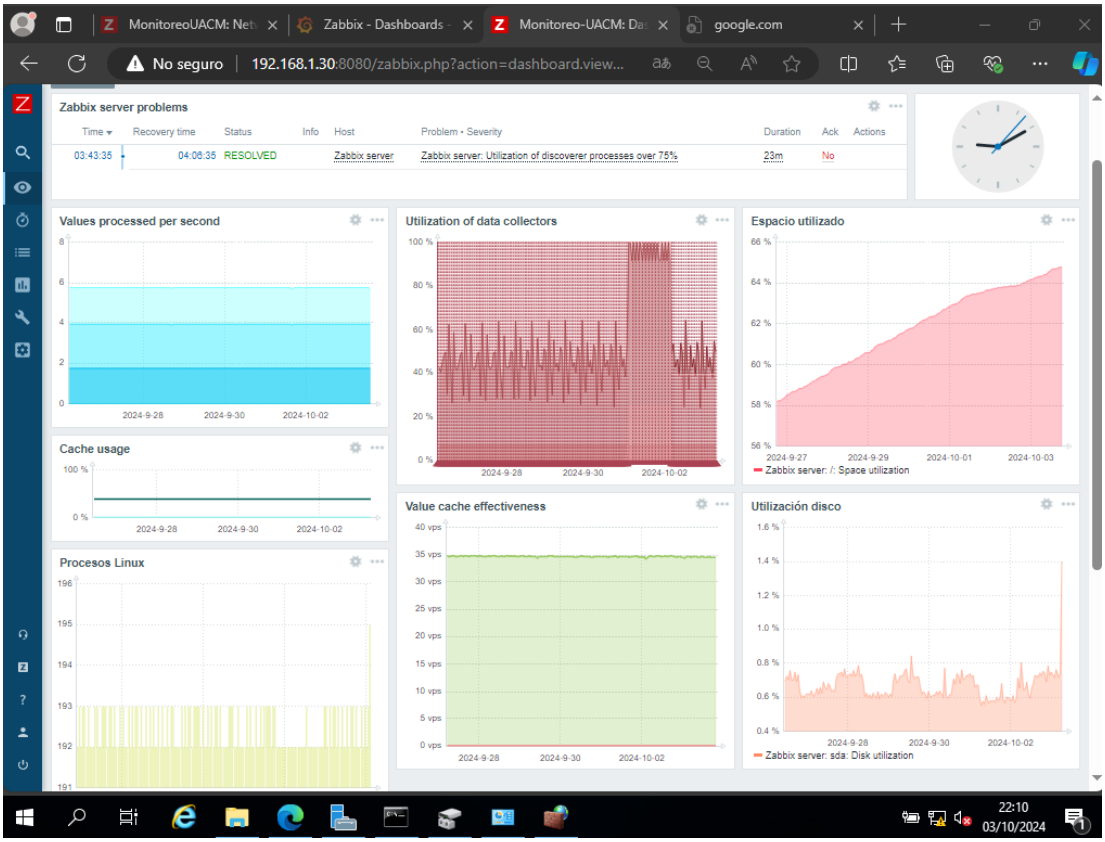


Figura 4.7: Métricas de rendimiento y uso de recursos del servidor Zabbix.

a) **Valores procesados por segundo:** Se observa que esta métrica es estable, procesando entre 3 y 4 valores por segundo sin caídas de rendimiento.

b) **Utilización de los recolectores de datos:** Hay variabilidad considerable en el uso de recolectores, atribuida a cambios en la carga del sistema o consultas, sin afectar la disponibilidad.

c) **Espacio utilizado:** El uso del espacio del servidor aumenta progresivamente al 64 %, indicando un posible crecimiento en registros o archivos, requiriendo evaluación para prevenir problemas futuros de almacenamiento.

d) **Uso de caché:** La caché permanece estática al 0 %, sugiriendo un uso no óptimo o mala configuración, afectando el rendimiento.

e) **Efectividad del caché de valores:** La efectividad del caché de valores se mantiene estable en alrededor de 35 valores por segundo (vps). Este comportamiento indica que el sistema está funcionando eficientemente con un número constante de aciertos de caché, sin cambios significativos en este aspecto.

f) **Procesos Linux:** El número de procesos Linux en el servidor ha tenido pequeñas variaciones entre 192 y 196 procesos a lo largo del tiempo. Esta métrica refleja esta-

bilidad en el sistema operativo, lo cual es un buen indicador de que el servidor no ha experimentado picos ni sobrecarga de procesos.

g) **Utilización del disco:** La métrica muestra una utilización muy baja del disco, variando ligeramente entre el 0.6 % y 1.2 % del total. Esto sugiere que, a pesar del aumento en el espacio utilizado (reflejado en otra gráfica), la carga del disco en términos de operaciones de lectura/escritura se mantiene baja.

La figura 4.8 identifica áreas de mejora como el uso de la caché y la estabilidad de la recolección de datos. También se monitorean picos en el uso del disco y procesos Linux para prevenir cuellos de botella en periodos de alta carga.

a) **Valores procesados por segundo:** Se observa un patrón repetitivo en la cantidad de valores procesados, que varía entre 3 y 8 por segundo en un corto tiempo. Esto sugiere un constante monitoreo en las cargas intermitentes del sistema para detectar posibles picos de actividad.

b) **Utilización de los recolectores de datos:** Notar una disminución repentina en la utilización de los recolectores de datos, llegando a casi el 0 %. Este comportamiento puede indicar una posible interrupción en la recolección de datos o una actividad temporalmente reducida. En el caso de detectar dichas disminuciones, revisar la estabilidad de los recolectores y su capacidad para recuperar el ritmo normal de operación.

c) **Espacio utilizado:** Identificar un incremento progresivo del espacio utilizado en el servidor, subiendo desde el 64.80 % hasta alrededor del 64.84 %. Este crecimiento, aunque no drástico, debe ser monitoreado para evitar futuras limitaciones de almacenamiento.

d) **Uso de caché:** Observar que el uso de caché se mantiene en 0 %, lo cual sugiere que no se está utilizando la caché de manera eficiente. Evaluar si la configuración de la caché es correcta o si existe alguna oportunidad para optimizar su uso, dado que una caché inactiva puede impactar en el rendimiento.

e) **Efectividad del caché de valores:** La efectividad del caché oscila entre 20 y 40 valores por segundo. Aun así, la tasa de aciertos sigue siendo buena.

f) **Procesos Linux:** Detectar una variación significativa en la cantidad de procesos en el servidor Linux, con un pico de hasta 206 procesos. Esto sugiere que, en algún punto, el sistema incrementó su carga de procesos, lo que puede deberse a la ejecución de múltiples tareas simultáneas.

g) **Utilización del disco:** Observar que la utilización del disco muestra picos intermitentes, con un aumento gradual que llega a 3.5 %. Aunque el uso general del disco sigue siendo bajo, los picos podrían ser señal de actividad intensa en el disco, como escrituras o lecturas masivas de datos.

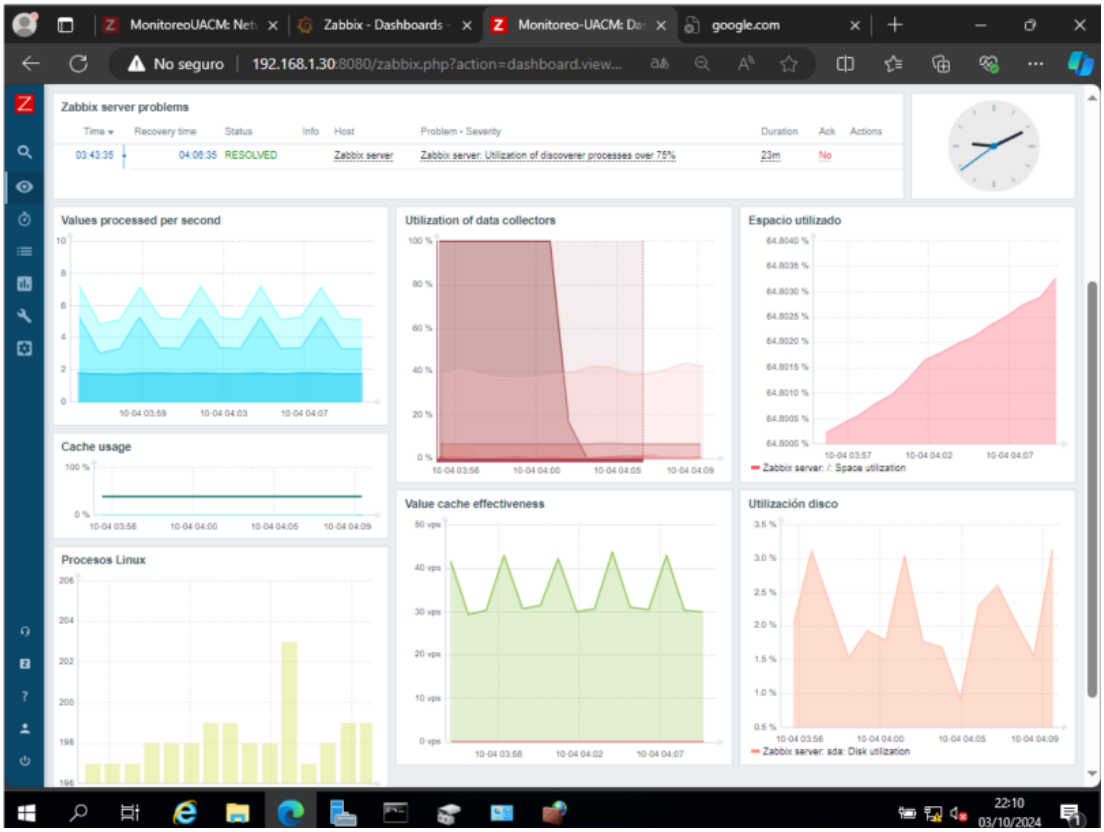


Figura 4.8: Métricas de rendimiento y uso de recursos del servidor Zabbix con periodo de tiempo diferente.

La principal diferencia entre ambas imágenes radica en el lapso de tiempo monitoreado. Esta diferencia temporal permite comparar cómo han evolucionado las métricas en un periodo determinado.

La siguiente tabla 4.2 muestra una comparativa de las métricas clave en ambos periodos de tiempo, destacando los valores registrados en cada uno:] En ambas figuras 4.7 y 4.8 se visualizan las mismas métricas clave que permiten evaluar el rendimiento del servidor. Estas métricas incluyen los valores procesados por segundo, la utilización de los recolectores de datos, el uso del espacio en disco, la efectividad del caché de valores, el uso de caché y la cantidad de procesos en el sistema Linux.

La principal diferencia entre ambas imágenes radica en el lapso de tiempo monitoreado. Esta diferencia temporal permite comparar cómo han evolucionado las métricas en un periodo determinado.

La siguiente tabla 4.2 muestra una comparativa de las métricas clave en ambos periodos de tiempo, destacando los valores registrados en cada uno:

Tabla 4.2: Comparativa de métricas del servidor entre las dos lapsos de tiempo

Métrica	Figura4.7	Figura4.8
Valores procesados por segundo	2-6 vps	3-8 vps
Utilización de recolectores	40-80 %	0-80 %
Espacio utilizado	64.60 % - 64.80 %	64.80 % - 64.84 %
Uso de caché	0 %	0 %
Efectividad del caché de valores	20-30 vps	20-40 vps
Procesos Linux	192-194 procesos	198-206 procesos
Utilización del disco	0.5 %-1.6 %	1.5 %-3.5 %

4.2.5. Monitoreo del Servidor Windows

La figura 4.9 muestra un conjunto de gráficos de monitoreo en Zabbix relacionados con el rendimiento del sistema Windows.

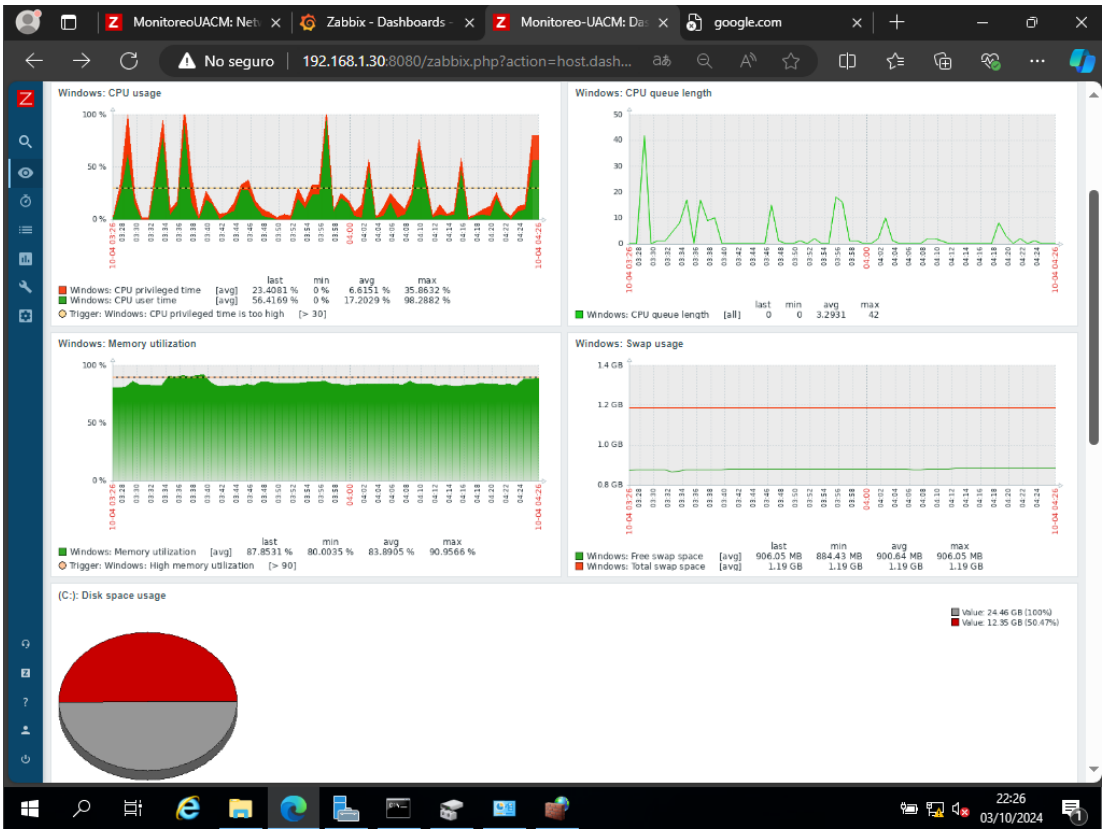


Figura 4.9: Métricas de rendimiento y uso de recursos del servidor Windows.

El análisis de cada sección de las gráficas presentadas se detalla a continuación:

a) Uso de CPU (*Windows: CPU usage*):

La gráfica de uso de CPU está dividida en dos métricas:

- CPU privileged time (rojo): Representa el tiempo en que la CPU está ejecutando instrucciones de sistema (kernel). El valor promedio es del 23.4081 %, con un máximo alcanzando picos de hasta un 55.4616 %.
- CPU user time (verde): Representa el tiempo en que la CPU ejecuta instrucciones de usuario. El promedio es del 12.2029 %, y el máximo llega al 98.2882 %.

Hay picos de uso de CPU notables durante el periodo monitoreado, lo que sugiere que las tareas o procesos de usuario y de sistema consumen la CPU en determinados momentos. Específicamente, un disparador indica que el tiempo privilegiado es muy alto cuando supera el 30 %, lo cual ha ocurrido varias veces.

b) Longitud de la cola de CPU (*Windows: CPU queue length*):

- Esta gráfica muestra cuántos procesos están en espera para ser atendidos por la CPU.
- El valor promedio es bajo 3.2931, pero ha alcanzado un máximo de 42, lo que sugiere que en algunos momentos ha habido una saturación en la capacidad de la CPU para procesar tareas, aunque la mayor parte del tiempo está bajo control.

c) Utilización de memoria (*Windows: Memory utilization*):

- La gráfica de utilización de memoria muestra que el uso promedio de la memoria RAM es muy alto, alrededor del 83.809 %, con un valor máximo que ha alcanzado el 90.9566 %.
- El disparador indica que se emite una alerta si la utilización de la memoria supera el 90 %, lo cual ocurrió en momentos durante el monitoreo.

d) Uso de memoria swap (*Windows: Swap usage*):

- Esta gráfica muestra la utilización del espacio de intercambio (swap), el cual es usado cuando la RAM está llena.
- La cantidad total de espacio de swap es 1.19 GB, y el uso ha sido muy bajo, permaneciendo en 0 MB durante casi todo el tiempo, lo cual es positivo ya que indica que no se ha tenido que utilizar swap, lo que implicaría una sobrecarga en la memoria RAM.

e) Uso de espacio en disco (*C Disk space usage*):

- Esta gráfica en forma de pastel muestra el uso del disco en la unidad C.

- Hay un total de 24.46 GB de espacio, de los cuales 12.36 GB (50.47 %) están ocupados, mientras que el resto está disponible. Esto sugiere que el espacio en disco no es un problema grave por el momento, ya que hay suficiente capacidad restante.

El sistema está experimentando picos periódicos en el uso de la CPU, particularmente en el "CPU privileged time", lo cual está relacionado con procesos o servicios del sistema operativo. Asimismo, se observa una alta utilización de la memoria RAM, la cual se aproxima al 90 %) en distintos momentos. Aunque el espacio de swap no ha sido utilizado, esto indica que la memoria no ha llegado al punto de sobrecarga.

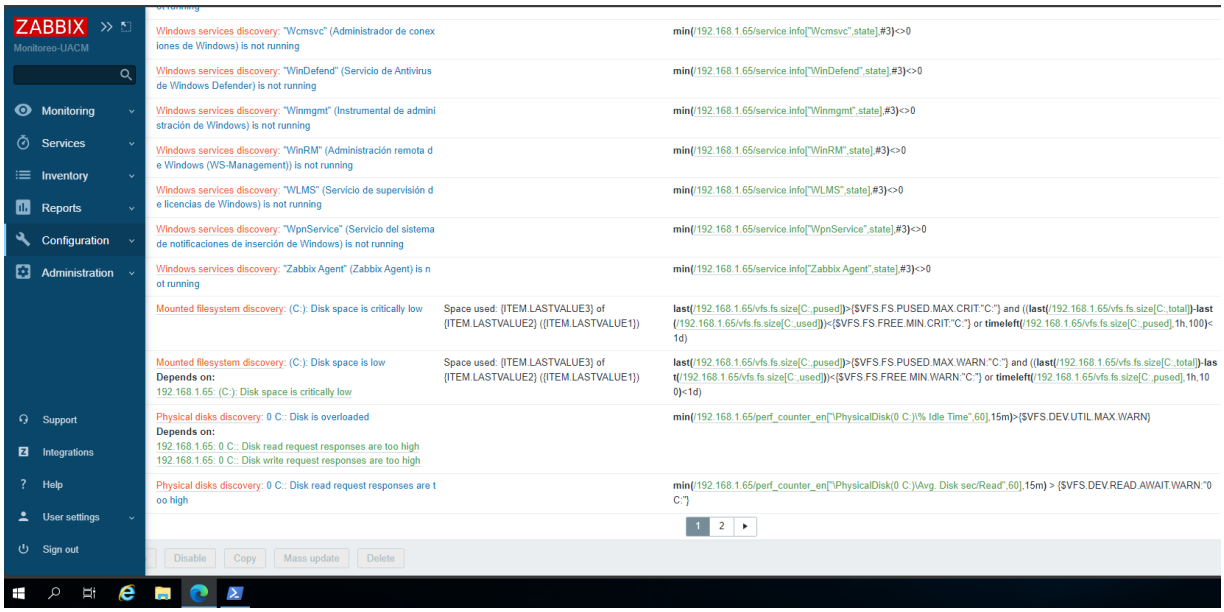


Figura 4.10: Alertas del host con Windows encontradas con Zabbix.

a) **Descubrimiento de servicios de Windows:**

- Aparecen múltiples servicios de Windows que no están corriendo (e.g., Wcmsvc, WinDefend, WpnService, etc.).
- Estas alertas indican que ciertos servicios esenciales de Windows han dejado de funcionar o no están activos, lo que puede generar problemas en la administración del sistema y en la seguridad.
- Todos estos están bajo un criterio de monitoreo que indica si están activos o no.

b) **Descubrimiento de sistemas de archivos montados:**

- Se detecta que el espacio en el disco C es críticamente bajo y otra alerta que indica que está cerca de quedarse sin espacio. Ambas advertencias dependen del dispositivo con la IP 192.168.1.65.

- Esta situación es crítica ya que podría afectar el rendimiento y causar fallos en los servicios que dependan de este disco si no se libera espacio o se aumenta la capacidad.

c) **Descubrimiento de discos:**

- Hay un disco con sobrecarga en el servidor con la dirección IP 192.168.1.65.
- Se muestran problemas relacionados con las altas latencias en las respuestas de lectura y escritura del disco. Las respuestas de lectura y escritura del disco C están excediendo los tiempos máximos permitidos, lo que puede indicar un rendimiento deficiente.
- Este tipo de sobrecarga puede tener un impacto negativo en el rendimiento general del sistema y en las aplicaciones que dependan del acceso a ese disco.

Acciones realizadas:

Se revisaron los servicios de Windows mencionados para determinar si es necesario que estén en ejecución o si algunos pueden ser desactivados. Aquellos que sean críticos deben reiniciarse. Liberar espacio en el disco C o realizar una expansión del almacenamiento es fundamental para evitar interrupciones en el sistema. Monitorear el rendimiento del disco en la IP afectada (192.168.1.65) para investigar si hay alguna causa adicional.

Prototipos de disparadores

La figura 4.11 muestra la sección de **Trigger Prototypes** en Zabbix, una herramienta para el monitoreo de redes y servidores. Los **trigger prototypes** son plantillas que definen condiciones para generar alertas basadas en valores de rendimiento o del estado del sistema. En este caso, los disparadores están monitoreando el rendimiento de los discos en un servidor específico (192.168.1.65), generando advertencias sobre la sobrecarga del disco y los altos tiempos de respuesta en las solicitudes de lectura y escritura.

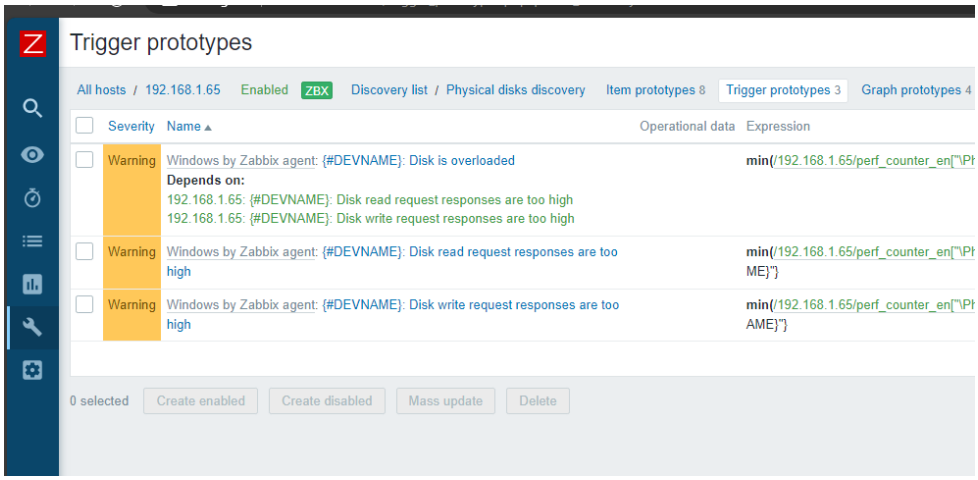


Figura 4.11: Plantilla que monitorea el rendimiento de discos.

Como se mencionó la figura anterior muestra el rendimiento de discos en el host con la IP 192.168.1.65.

En este caso hay tres alertas activas que están marcadas como advertencias.

a) **Disco sobrecargado:**

- Se refiere a que el disco está sobrecargado, lo que podría indicar problemas de rendimiento relacionados con la entrada/salida.
- La expresión de este trigger está evaluando el tiempo de inactividad del disco (% Idle Time) y comparándolo con un valor de advertencia establecido `$VFS.DEV.UTIL.MAX.WARN`. Esto sugiere que el disco está siendo utilizado intensamente, lo que puede causar sobrecarga.

b) **El tiempo de respuesta de las operaciones de lectura y escritura en disco es demasiado alto:**

- Esta alerta indica que los tiempos promedio de respuesta para las operaciones en disco han superado los umbrales configurados. Para las lecturas en disco (Avg. Disk sec/Read), el tiempo de espera ha sobrepasado el umbral definido en {$VFS.DEV.READ.AWAIT.WARN}. De manera similar, para las escrituras en disco (Avg. Disk sec/Write), se ha excedido el umbral de advertencia configurado en {$VFS.DEV.WRITE.AWAIT.WARN}.

Posibles causas:

- **Altas solicitudes de entrada/salida:** Los tiempos de respuesta altos y el disco sobrecargado podrían deberse a un gran volumen de solicitudes de lectura/escritura.
- **Configuración inadecuada de recursos:** Si el sistema está utilizando más recursos de disco de los disponibles, se puede sobrecargar el sistema. **Correcciones aplicadas:**
- **Monitorear el uso de disco:** Verificar qué procesos o aplicaciones están utilizando intensivamente el disco.
- **Ajustar los umbrales de advertencia:** Si los valores de advertencia están demasiado bajos, revisar y ajustar los umbrales de acuerdo a la capacidad del sistema.
- **Optimización de procesos:** Revisión de alguna aplicación o proceso en el servidor está generando demasiadas operaciones de lectura/escritura y optimizar.

4.3. Monitoreo del enrutador Cisco

Para un funcionamiento óptimo del enrutador Cisco, es esencial monitorear las métricas de temperatura. Zabbix facilita este seguimiento al ofrecer gráficas detalladas en tiempo real y registrar las variaciones clave. La figura 4.12 muestra el seguimiento de la temperatura del dispositivo *enrutador Cisco*.

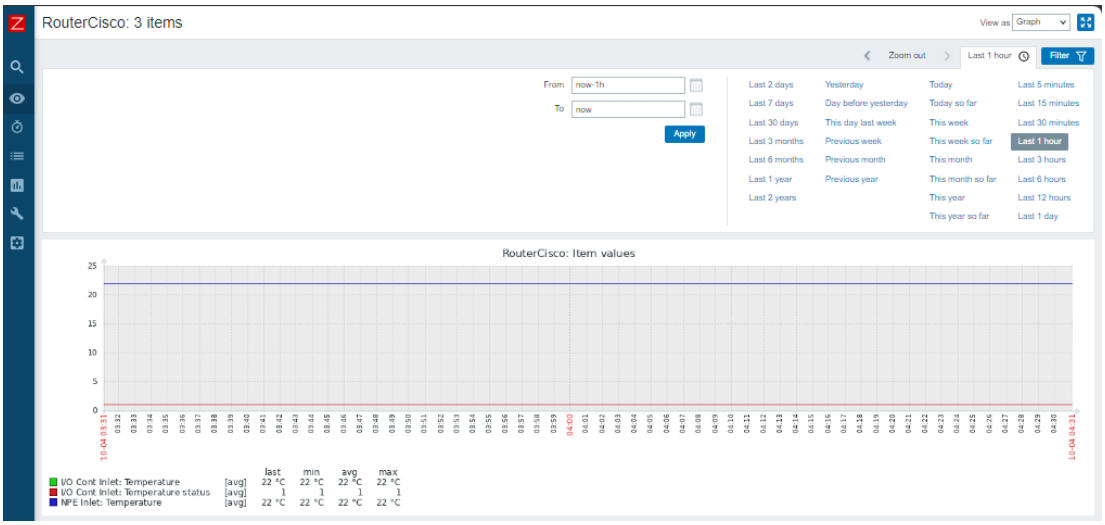


Figura 4.12: Métricas de temperatura del dispositivo enrutador Cisco.

Detalles observados:

a) Ítems monitoreados

b) **V0 Cont Inlet:** Temperature – Temperatura del enrutador.,

- Última lectura: 22°C.
- Valor mínimo: 22°C.
- Valor promedio: 22°C.
- Valor máximo: 22°C.

c) **V0 Cont Inlet:** Temperature status – Estado de la temperatura en el inlet.,

- Última lectura: 1.
- Valor mínimo, promedio, y máximo: 1.

d) **NPE Inlet:** Temperature – Temperatura del módulo de procesamiento de red (NPE) del equipo.

- Última lectura: 22°C.
- Valor mínimo: 22°C.
- Valor promedio: 22°C.

- Valor máximo: 22°C.

e) **Gráfica:**

- La gráfica no muestra mucha variación, lo que sugiere que la temperatura ha sido estable durante el intervalo de tiempo seleccionado (última hora).
- Los valores permanecen en 22°C (línea verde para V0 Cont Inlet y línea azul para NPE Inlet).

La temperatura del dispositivo *enrutador Cisco* está estable y dentro del rango adecuado (22 °C), sin signos de sobrecalentamiento. La figura 4.13 ilustra el monitoreo de la interfaz de red Gi1/0/0 (*enrutador Cisco*), centrándose en el tráfico en bits por segundo (bps).

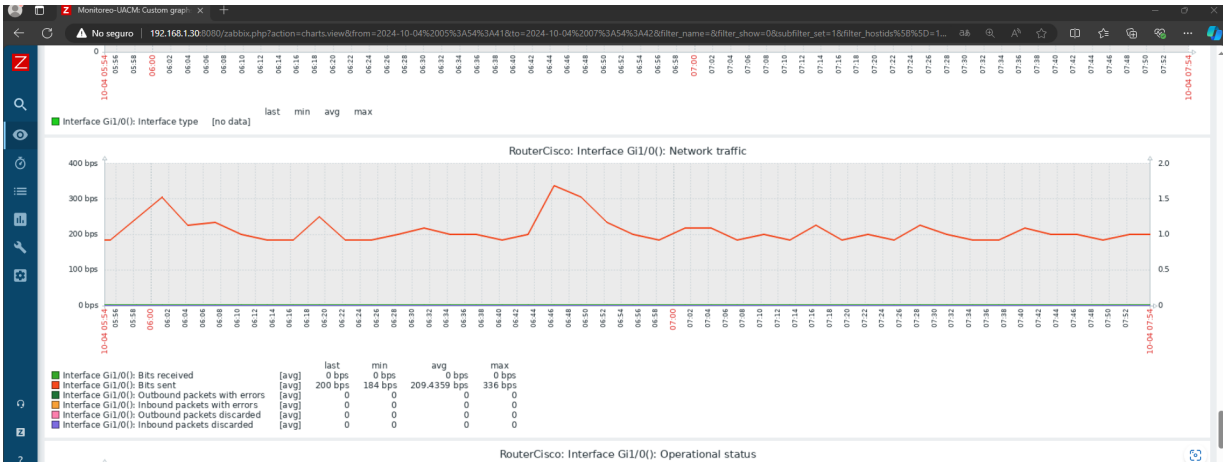


Figura 4.13: Interfaz de red Gi1/0/0 del dispositivo enrutador Cisco.

Análisis

a) **Tráfico de red en bits por segundo (bps):**,

- Se observa una variabilidad en el tráfico de la interfaz, alcanzando picos cercanos a los 336 bps y mínimos alrededor de los 184 bps.
- El tráfico promedio de red es de 209.43 bps, lo que refleja una actividad moderada en la interfaz.
- La gráfica refleja una variación constante entre los picos, pero sin ninguna anomalía significativa como caídas abruptas o interrupciones.

b) **Métricas específicas de la interfaz Gi1/0/0:**

- **Bits recibidos:** No parece haber datos disponibles para este valor (0 bps).
- **Bits enviados:** El tráfico se centra en los bits enviados, con un promedio de 209.43 bps, lo cual indica que el dispositivo está enviando más datos de los que recibe.

- **Errores en paquetes entrantes/salientes:** Tanto los paquetes entrantes como los salientes no presentan errores, lo que sugiere una conexión estable y sin problemas de transmisión.
- **Paquetes descartados:** Ningún paquete ha sido descartado, ni en la dirección entrante ni saliente, lo que también es una señal de que la red está operando de manera eficiente.

El tráfico de esta interfaz es bajo y estable, sin errores ni pérdidas de paquetes, lo que indica un funcionamiento correcto. Zabbix posee disparadores automáticos que generan notificaciones al exceder ciertos umbrales, permitiendo detectar problemas críticos como falta de espacio en disco o sobrecarga de unidades. La figura 4.10 muestra estos disparadores, y la figura 4.14 presenta alertas sobre el estado de varios componentes, como uso de CPU, temperatura, pérdida de paquetes ICMP y tiempo de respuesta en dispositivos Cisco.

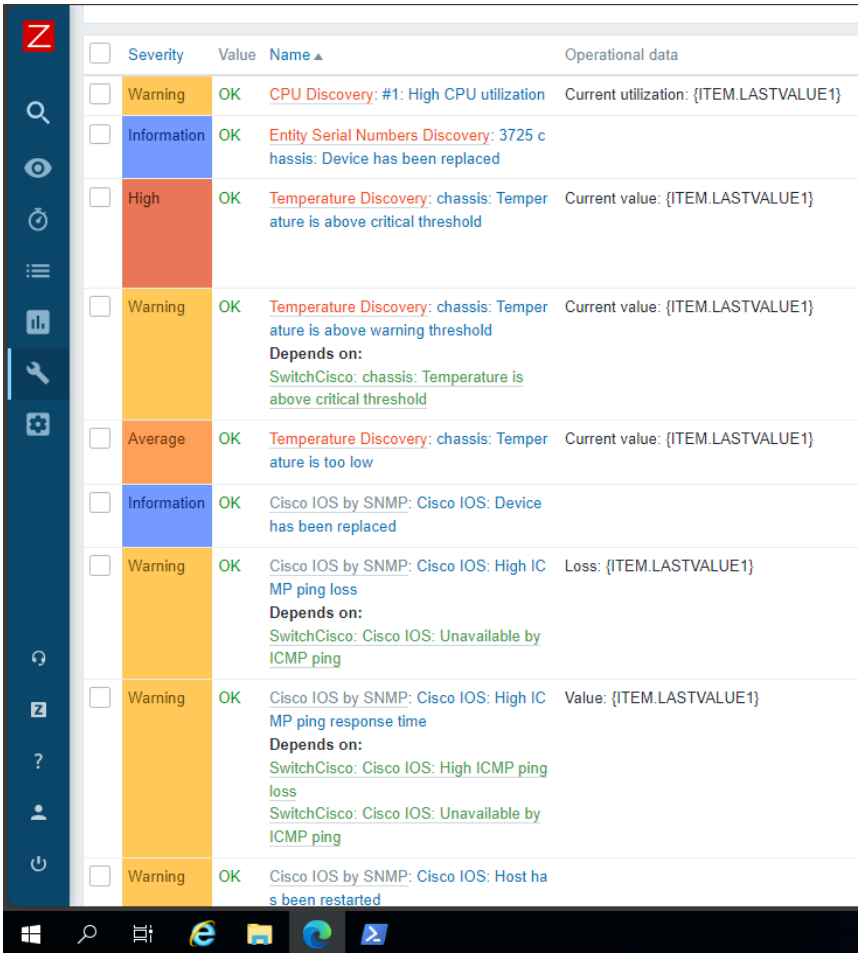


Figura 4.14: Monitoreo de distintas alertas en el enrutador Cisco.

a) Alta **utilización de CPU** en un dispositivo Cisco, marcado como una advertencia.

b) **Temperatura** crítica y por encima del umbral, con alertas en nivel alto y advertencia.

c) Pérdida de *paquetes ICMP* y alto tiempo de respuesta de ping, que podrían indicar problemas de red o congestión.

La figura 4.15 muestra la sección, específicamente el apartado de **gráficas** relacionado con un dispositivo denominado **conmutadorCisco**. Este módulo permite visualizar el estado de diferentes parámetros del dispositivo, como la utilización de CPU, memoria y tráfico de red en distintas interfaces (Fa1/0, Fa1/1, etc.). Estas gráficas proporcionan una representación visual del rendimiento y la actividad de las interfaces de red, lo que facilita la identificación de posibles cuellos de botella o problemas de tráfico.

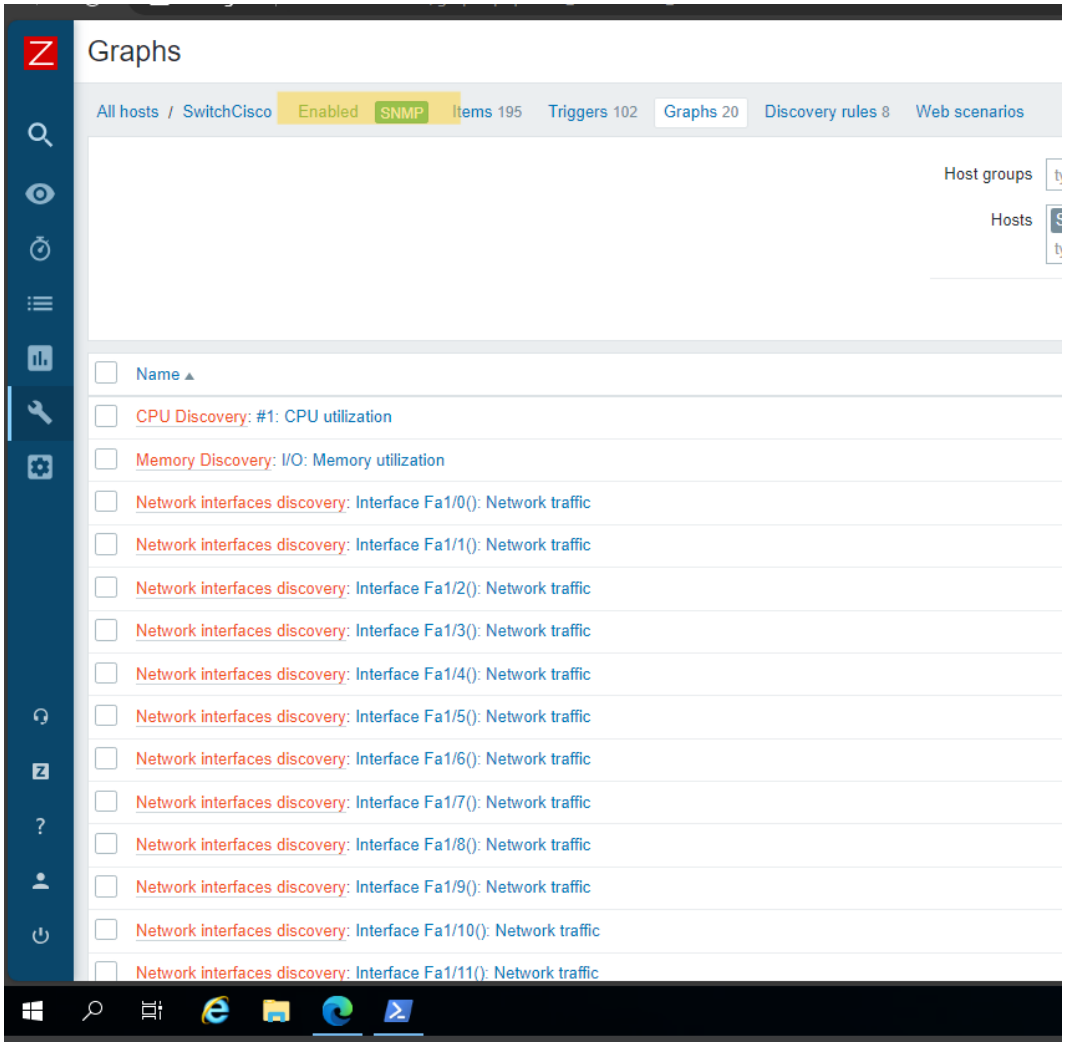


Figura 4.15: Estado de monitoreo del dispositivo conmutador Cisco.

Las gráficas están agrupadas bajo el protocolo SNMP (Simple Network Management Protocol), lo que indica que Zabbix está utilizando este protocolo para recopilar datos en tiempo real desde el dispositivo Cisco.

Ahora analizamos los datos mostrados en la tabla 4.3 que muestra el monitoreo del dispositivo enrutador Cisco, específicamente de la interfaz Gi1/0: bits enviados. En esta tabla se registran los bits enviados a través de la interfaz mencionada, con información de la marca de tiempo y el valor de los bits transmitidos en diferentes intervalos de tiempo.

Tabla 4.3: Monitoreo de bits enviados de la interfaz Gi1/0

Marca de tiempo	**Bits enviados**
2024-10-04 08:10:22	184
2024-10-04 08:07:22	248
2024-10-04 08:04:22	184
2024-10-04 08:01:23	224
2024-10-04 07:58:22	200
2024-10-04 07:55:22	224
2024-10-04 07:52:22	200
2024-10-04 07:49:22	184
2024-10-04 07:46:22	200
2024-10-04 07:43:22	200
2024-10-04 07:40:22	216
2024-10-04 07:37:22	184
2024-10-04 07:34:22	184
2024-10-04 07:31:22	200
2024-10-04 07:28:22	224
2024-10-04 07:25:22	184
2024-10-04 07:22:22	200
2024-10-04 07:19:22	184
2024-10-04 07:16:22	224
2024-10-04 07:13:22	184

Puntos clave sobre la información observada en la tabla anterior:

a) **Fecha y hora:** Los datos se registran en intervalos de tiempo de tres minutos o menos, comenzando el 4 de octubre de 2024 a las 07:13:22 y finalizando a las 08:10:22.

b) **Valores de bits enviados:**

- El valor más bajo registrado es de 184 bits (que aparece varias veces, como a las 07:13:22, 07:31:22, 07:37:22).
- El valor más alto registrado es 248 bits, a las 08:07:22.

- Se observan variaciones en los valores de bits enviados, con varios valores intermedios, como 224, 200, y 216 bits.

c) **Comportamiento:** Los valores de los bits enviados muestran variaciones a lo largo del tiempo, aunque no parece seguir un patrón claro de incremento o decremento constante. La variación observada es típica en redes donde la cantidad de datos enviados dependen del tráfico de la red y de la actividad en el uso del sistema.

El monitoreo de la interfaz Gi1/0 de este enrutador Cisco muestra un tráfico relativamente bajo (los valores oscilan entre 184 y 248 bits enviados) y las variaciones son esperables en función de la actividad de la red.

4.4. Monitoreo web con Zabbix

La configuración de monitoreo web en Zabbix es una funcionalidad avanzada que permite supervisar de manera integral el rendimiento y la disponibilidad de sitios y aplicaciones web.
Este tipo de monitoreo es importante para asegurar que los servicios web estén operando de manera óptima y para identificar problemas antes de que afecten la operación del sitio o de la aplicación. Entre las ventajas de configurar el monitoreo web en Zabbix se incluyen la capacidad de realizar pruebas de disponibilidad y tiempo de respuesta, lo que permite detectar caídas del servicio y tiempos de carga elevados. Además, Zabbix puede simular el comportamiento de un usuario, realizando acciones como iniciar sesión y navegar por las páginas, lo que proporciona una visión detallada del rendimiento desde la perspectiva del usuario.

Otra ventaja es la generación de alertas automáticas en caso de que se detecten problemas, lo que permite reaccionar rápidamente y resolver cualquier incidencia.

Zabbix también ofrece informes detallados y gráficos sobre el rendimiento del sitio web, lo que ayuda a identificar tendencias y cuellos de botella que puedan estar afectando a la página monitoreada. Además, el monitoreo web en Zabbix es altamente configurable, permitiendo ajustar los parámetros de monitoreo según las necesidades específicas de cada sitio o aplicación.

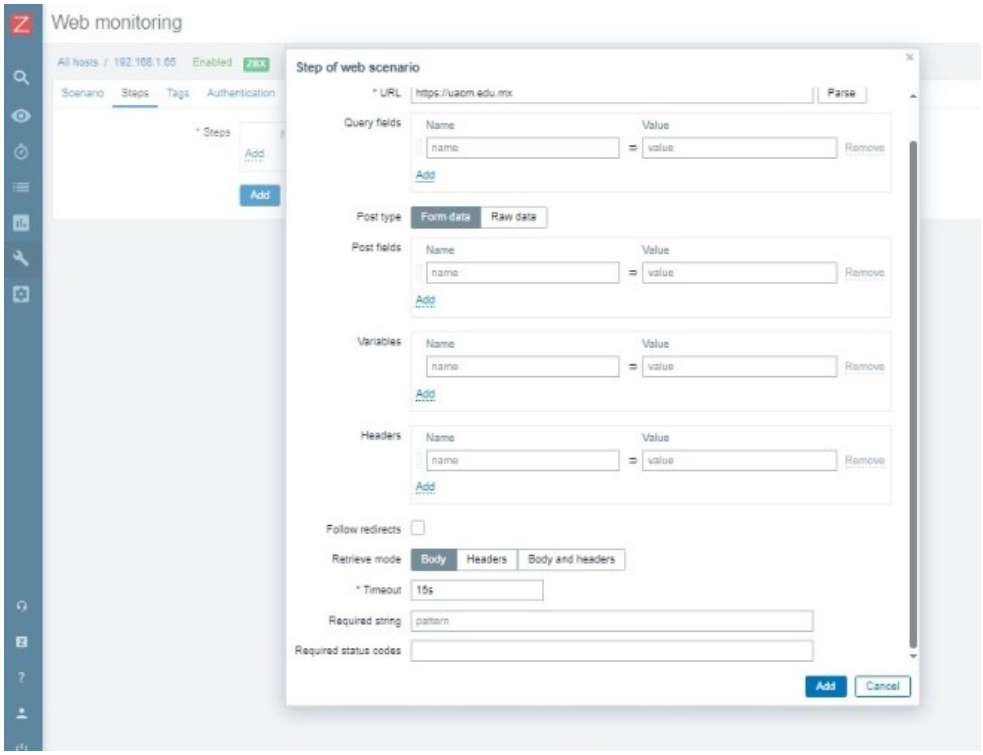


Figura 4.16: Configuración del monitoreo web

A continuación, se presenta un análisis detallado de una captura de pantalla de la herramienta de monitoreo Zabbix. La figura ilustra el estado actual de los hosts monitoreados, identificando problemas específicos relacionados con la resolución de dominios. Este análisis tiene como objetivo identificar las posibles causas de los errores detectados y proporcionar recomendaciones para su resolución.

La figura 4.17 muestra una pantalla de monitoreo web en la herramienta Zabbix.

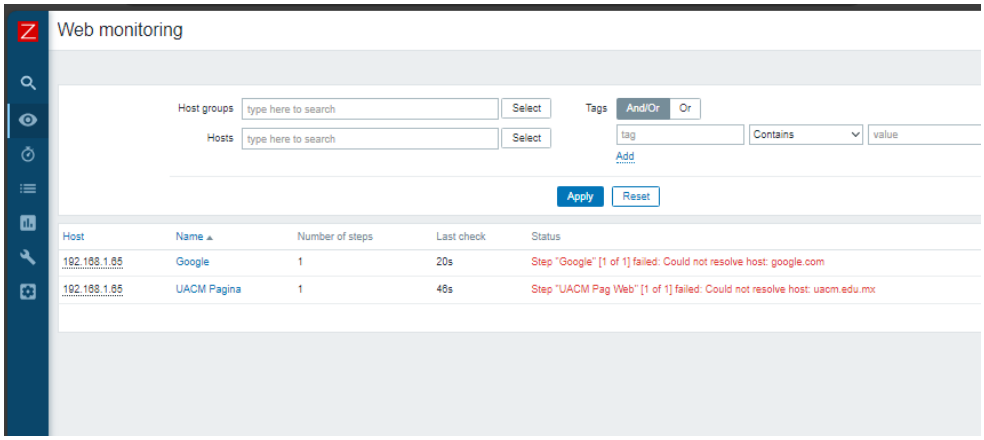


Figura 4.17: Estado del monitoreo web

El estado que se refleja es:

a) Páginas monitoreadas:

- wwww. google.com.mx desde la IP: 192.168.1.65
- www.uacm.edu.mx desde la IP: 192.168.1.65

b) Problema reportado: Ambos hosts están mostrando errores al intentar resolver los dominios:

- Para Google, aparece el error: Could not resolve host: google.com. Esto indica que Zabbix no pudo resolver el nombre de dominio google.com a una dirección IP, lo que sugiere un problema en el servidor DNS o en la conectividad hacia ese servidor.
- Para UACM Página Web, el error es similar: Could not resolve host: uacm.edu.mx. De nuevo, Zabbix no pudo convertir este nombre de dominio en una dirección IP.

c) Última comprobación:

- El host de Google fue comprobado hace 20 segundos.
- El host de UACM fue comprobado hace 48 segundos.

d) Número de pasos:

- Ambas pruebas tienen 1 paso, lo que indica que la verificación es básica, probablemente solo haciendo una solicitud simple al host para verificar su disponibilidad.

Posibles causas:

- **Problemas de DNS:** Es probable que el servidor o la red desde la que se ejecuta Zabbix esté experimentando problemas para resolver los nombres de dominio. Revisar la configuración de los servidores DNS sería un buen primer paso.
- **Problemas de conectividad a Internet:** Si el servidor Zabbix no tiene conexión a Internet o hay fallos en el enrutamiento, esto podría explicar por qué no se pueden resolver estos nombres de dominio.
- **Errores en la configuración de Zabbix:** Si la configuración de Zabbix para la prueba de estos hosts es incorrecta (por ejemplo, si se ha definido un servidor DNS erróneo), también podría causar estos errores.

Solución aplicada:

- Verificación de la configuración de DNS en el servidor donde corre Zabbix.
- Revisión de conectividad a Internet.

- Revisión de cambios recientes en la configuración de red o DNS que pudieran estar afectando la resolución de nombres.

Una vez corregidos los inconvenientes que afectaban la estabilidad de la conexión y el tiempo de respuesta de la página web, se logró obtener datos sobre su desempeño. Las gráficas presentadas en la figura 4.18 muestran el monitoreo de la velocidad de descarga y el tiempo de respuesta de la página web "UACM Página". La velocidad se mantuvo alta y estable después de un breve período inicial, mientras que el tiempo de respuesta presentó un pico inicial que posteriormente se normalizó.

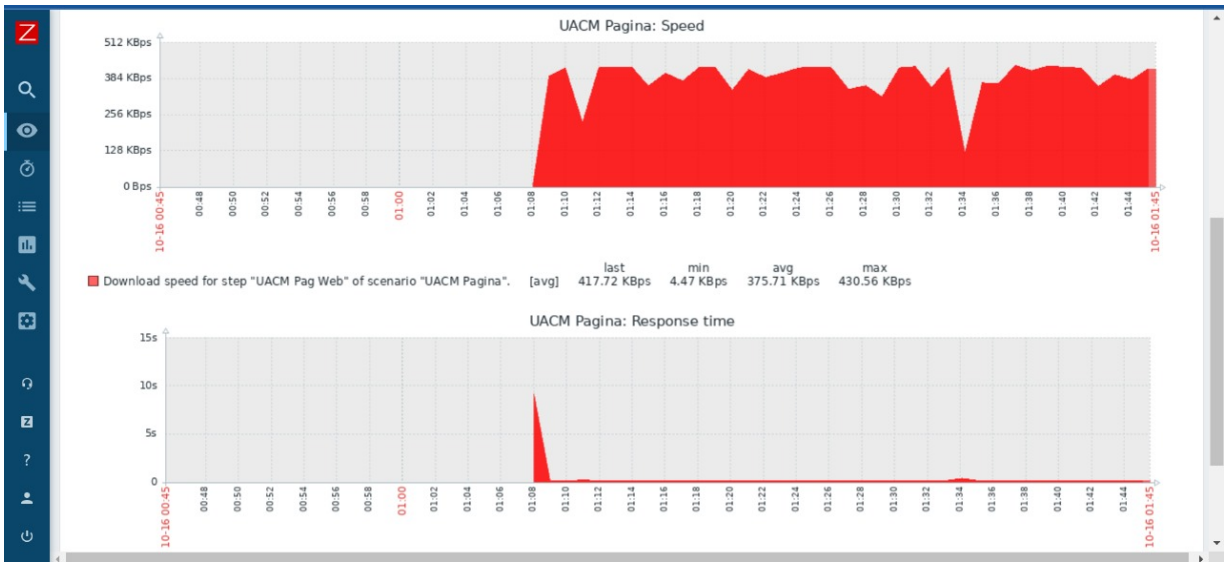


Figura 4.18: Graficas que analizan el rendimiento de la página web UCAM.

a) **Gráfica superior (UACM Pagina: Speed):**

- Esta gráfica muestra la velocidad de descarga en kilobytes por segundo (KBps) a lo largo del tiempo.
- El rango de velocidades oscila entre 0 y 512 KBps.
- Se observa un aumento significativo de la velocidad después de los primeros segundos, estabilizándose entre los 300 y 450 KBps con algunos cambios menores.
- La leyenda en la parte inferior de la gráfica indica que la velocidad promedio es de 375.71 KBps, la velocidad mínima fue de 4.47 KBps, y la máxima alcanzada es de 430.56 KBps.

b) **Gráfica inferior (UACM Pagina: Response time):**

- Muestra el tiempo de respuesta en segundos de la página web.
- El tiempo máximo de respuesta parece estar en el rango de 15 segundos, pero la mayor parte de la gráfica muestra tiempos de respuesta cercanos a 0, con un pico significativo al inicio, alrededor del segundo 10.

- Después de este pico inicial, el tiempo de respuesta vuelve a ser muy bajo.

La figura 4.19 muestra que el monitoreo de la página web titulada "UACM Página" mide la velocidad de descarga y el tiempo de respuesta de la misma.

Figura 4.19: Velocidad de descarga y tiempo de respuesta de la página web UACM.

a) **Velocidad:**

- La velocidad promedio de descarga para la página es de 330.99 KBps.
- La velocidad máxima registrada es de 442.87 KBps y la mínima de 4.47 KBps, lo cual indica variaciones importantes durante el periodo medido.

b) **Tiempo de respuesta:**

- El tiempo promedio de respuesta de 123.78 ms indica una buena capacidad del servidor para manejar solicitudes dentro de un rango aceptable para la mayoría de aplicaciones web.

c) **Código de respuesta:**

- El código HTTP es 200, lo que significa que la página fue accesible correctamente durante la monitorización.

d) **Gráfica de velocidad:**

- La gráfica muestra variaciones en la velocidad de descarga desde las 21:46 (del 15 de octubre de 2024) hasta las 01:46 (del 16 de octubre de 2024). Las caídas en la gráfica podrían indicar momentos de carga pesada o interrupciones breves en el servicio.

e) **Período de monitoreo:**

- El monitoreo cubre un periodo de aproximadamente 4 horas, desde el 15 de octubre de 2024 a las 21:46 hasta el 16 de octubre de 2024 a las 01:46 hrs.

4.5. Análisis de paquetes del protocolo SNMP

El análisis de paquetes del protocolo SNMP constituye un componente clave en el monitoreo de redes, ya que permite evaluar en detalle el estado y el rendimiento de los dispositivos de red.

Durante la implementación del sistema de monitoreo, se utilizó Wireshark para capturar los paquetes SNMP intercambiados entre el servidor Zabbix y los dispositivos gestionados. Este análisis permitió verificar el estado de los dispositivos, identificar posibles problemas de comunicación y medir la eficiencia de las consultas SNMP.

Zabbix es capaz de identificar en tiempo real cualquier anomalía, lo que resulta esencial para la administración eficiente de la red. En este contexto, se realizó un análisis de las consultas y respuestas SNMP, con el objetivo de evaluar el rendimiento de la red y detectar posibles retrasos, pérdidas de paquetes o errores de comunicación.

Se observó que el protocolo SNMP permitió una transmisión efectiva de la información sobre el estado de los dispositivos, incluidos enrutadores y conmutadores. A continuación, se presentan los resultados obtenidos del análisis de tráfico SNMP, destacando las diferencias en el rendimiento y la disponibilidad de los dispositivos monitoreados, especialmente en aquellos con y sin el agente de Zabbix instalado.

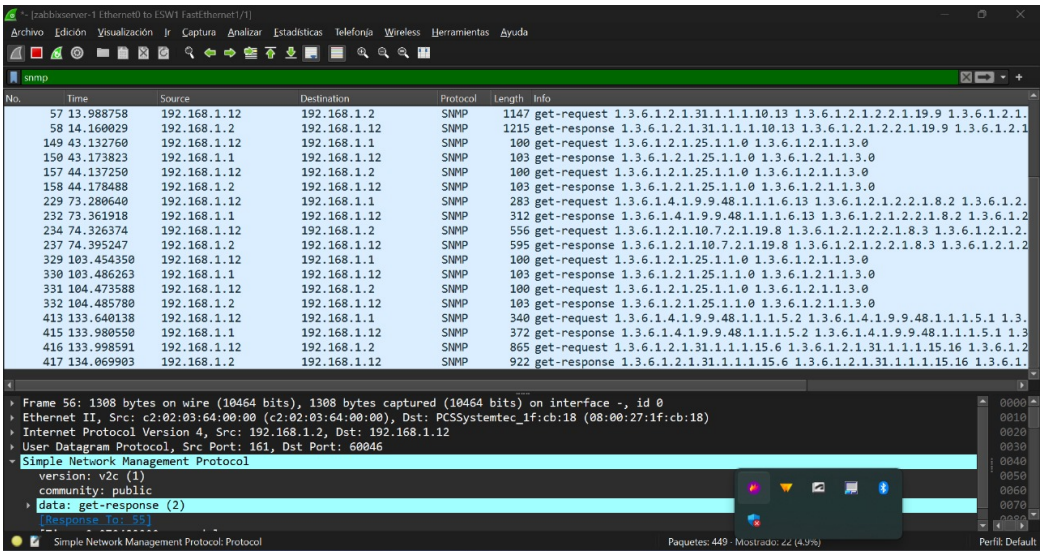


Figura 4.20: Captura de paquetes SNMP en Wireshark.

En la figura 4.20 se muestra una captura de tráfico de red SNMP obtenida mediante la herramienta Wireshark. La captura fue realizada en un entorno donde se monitorea la actividad de la red entre un servidor Zabbix y un dispositivo con la dirección IP `192.168.1.12`. En este caso, Wireshark nos permite observar las solicitudes y respuestas del protocolo

SNMP (Simple Network Management Protocol) que se utilizan para gestionar y supervisar dispositivos de red.

Descripción de los Paquetes Capturados

En la captura se observan principalmente dos tipos de paquetes:

- **get-request**: Estos paquetes representan las solicitudes enviadas desde el servidor Zabbix (dirección IP `192.168.1.2`) al dispositivo monitoreado (`192.168.1.12`) para obtener información específica mediante OID (Object Identifier), identificador único usado en SNMP para definir objetos específicos.
- **get-response**: Estos paquetes corresponden a las respuestas del dispositivo hacia el servidor, devolviendo la información solicitada por el servidor Zabbix.

Cada paquete incluye un OID específico que hace referencia a diferentes métricas o propiedades del dispositivo monitoreado. Por ejemplo, algunos OID comunes capturados incluyen valores para el estado del sistema, el uso de la CPU, la memoria y otros datos relevantes para el rendimiento del dispositivo.

La captura revela un tráfico constante de `get-request` y `get-response`, lo cual indica un monitoreo activo y en tiempo real del dispositivo. A continuación, se detallan algunos aspectos clave observados en la 4.20:

- **Frecuencia de Consultas**: Los paquetes se envían con una alta frecuencia, lo que sugiere un intervalo de monitoreo corto. Este enfoque es adecuado para una supervisión proactiva, ya que permite al servidor Zabbix detectar rápidamente cualquier cambio en el estado del dispositivo.
- **Versión de SNMP**: La versión del protocolo que se está utilizando es SNMP v2c, como se indica en la sección inferior de la captura. Esta versión es adecuada para entornos donde se requiere seguridad básica, dado que permite el uso de una comunidad pública o privada, en este caso configurada como `public`.
- **Tamaño de los Paquetes**: Los paquetes varían en tamaño, desde aproximadamente 100 hasta 1308 bytes. Esta variabilidad depende de la cantidad de datos solicitados y el tipo de respuesta que envía el dispositivo. Paquetes de mayor tamaño pueden indicar respuestas que incluyen datos más detallados o múltiples valores OID en una sola respuesta.
- **Interfaz de Usuario (UDP)**: Los paquetes utilizan el protocolo UDP en el puerto 161, lo cual es característico de SNMP. Este protocolo es liviano y de baja latencia, ideal para consultas rápidas y regulares, aunque carece de las garantías de entrega de TCP.

Interpretación de los Datos Recibidos

Los datos obtenidos de esta captura SNMP son fundamentales para la administración de la red, ya que permiten monitorear en tiempo real la disponibilidad y el rendimiento de los dispositivos. La respuesta de cada `get-request` proporciona métricas clave para el servidor Zabbix, que puede utilizar esta información para activar alertas en caso de detectar valores anómalos o fuera de rango en alguna de las métricas monitoreadas.

Además, este tipo de monitoreo no invasivo es especialmente útil para dispositivos de red que no permiten la instalación de agentes de monitoreo, como enrutadores y conmutadores. Gracias al protocolo SNMP, es posible obtener un conjunto básico pero crucial de datos sobre el rendimiento y el estado sin necesidad de software adicional en el dispositivo.

En la figura 4.21 se muestra un análisis detallado de un paquete de solicitud SNMP capturado mediante Wireshark. Este paquete específico corresponde a una solicitud `get-request` que el servidor de monitoreo Zabbix (IP `192.168.1.12`) envía al dispositivo de red ubicado en la dirección IP `192.168.1.2`. Este tipo de solicitudes se emplea para obtener información de los dispositivos de red a través de objetos identificados por OID (Object Identifiers).

Detalles del Paquete SNMP

En el paquete capturado se destacan las siguientes características principales:

- **Dirección IP de Origen**: `192.168.1.2`. Esta es la IP del dispositivo de red que responde a la solicitud enviada por el servidor de monitoreo.
- **Dirección IP de Destino**: `192.168.1.12`. Representa la dirección IP del servidor Zabbix, que es quien envía la solicitud de consulta `get-request`.
- **Puerto de Destino**: `161`. Este es el puerto estándar utilizado por SNMP para la comunicación con dispositivos de red. El uso de este puerto confirma que el tráfico corresponde a mensajes de monitoreo SNMP.
- **Protocolo de Capa de Transporte**: UDP (User Datagram Protocol). SNMP utiliza UDP debido a su baja sobrecarga y latencia, características importantes para el monitoreo en tiempo real.
- **Longitud del Paquete**: 66 bytes. Esto indica que el paquete contiene una consulta breve, optimizada para minimizar el tráfico de red.

Estructura de la Solicitud SNMP

En la parte inferior de la figura, se puede observar el contenido hexadecimal y ASCII del paquete. La solicitud incluye varios elementos clave:

- **Versión de SNMP**: v2c. Esta versión es ampliamente utilizada debido a su simplicidad y soporte para comunidades de seguridad.
- **Nombre de la Comunidad**: `public`. Este campo indica el nombre de la comunidad de SNMP, una medida de autenticación básica en SNMPv2c. En este caso, la comunidad es pública, lo que implica un nivel de acceso sin restricciones.
- **OID Consultado**: `1.3.6.1.2.1.25.1.1.0`. Este identificador hace referencia a un objeto específico que almacena información relevante sobre el dispositivo monitorizado. Los OID proporcionan acceso a datos específicos, tales como estadísticas de rendimiento o información de estado.

Este paquete SNMP forma parte de un monitoreo continuo, en el que el servidor Zabbix envía solicitudes `get-request` para recopilar datos en tiempo real de los dispositivos de red. La inclusión de la versión `v2c` y el nombre de la comunidad `public` facilita el acceso sin complicaciones a los datos básicos del dispositivo, aunque limita la seguridad debido a la falta de autenticación avanzada.

Este tipo de captura verifica la comunicación efectiva entre el servidor de monitoreo y los dispositivos de red. Además, permite observar la estructura interna de las solicitudes y respuestas SNMP, lo cual es importante para identificar posibles problemas de configuración o conectividad.

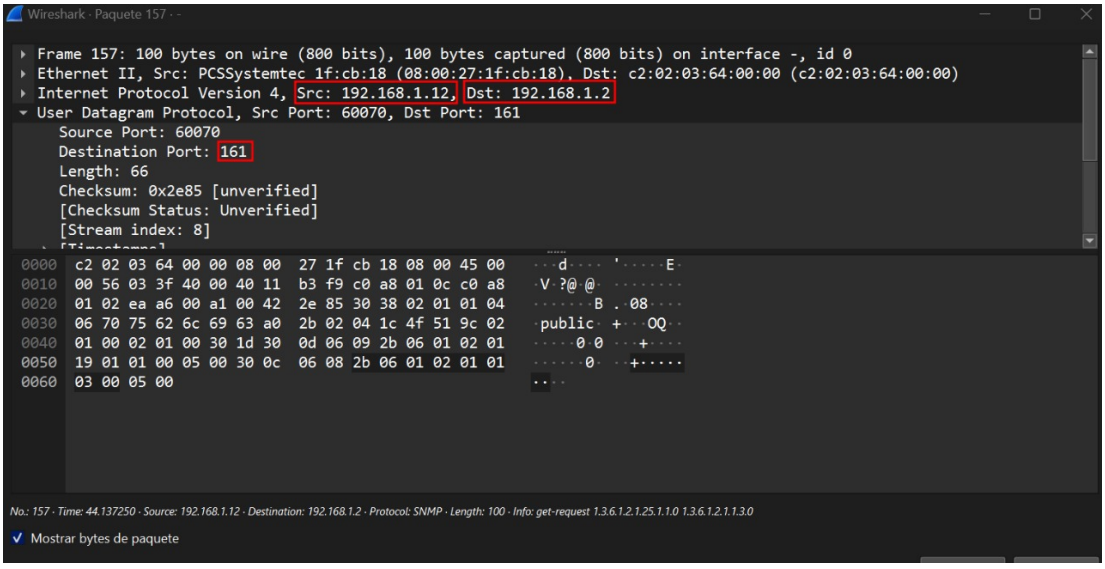


Figura 4.21: Análisis de un paquete SNMP en Wireshark

4.6. Detección y análisis de eventos críticos con traps SNMP

En el monitoreo de redes en donde se utiliza el protocolo SNMP, los *traps* representan una funcionalidad clave para la detección proactiva de eventos críticos en la infraestructura. A diferencia de las solicitudes tradicionales de SNMP (como *get-request* o *get-response*), los *traps* son mensajes generados automáticamente por los dispositivos monitoreados hacia el servidor de monitoreo, sin necesidad de que este último los solicite. Esta característica permite que el servidor Zabbix reciba notificaciones instantáneas de eventos relevantes, como fallos de conexión, sobrecarga de CPU o problemas de memoria, tan pronto como estos ocurren.

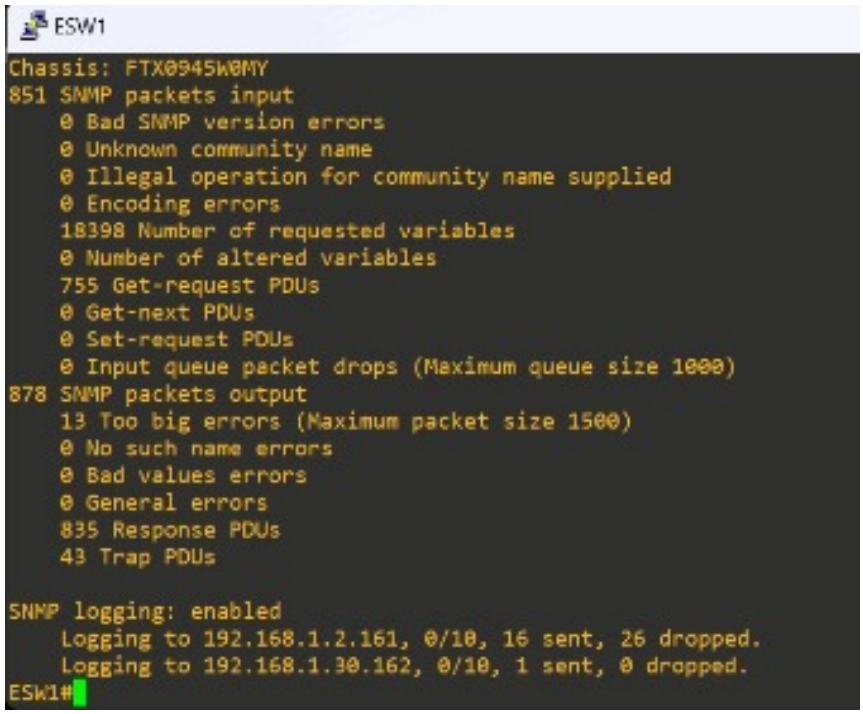


Figura 4.22: Estadísticas del protocolo SNMP.

La figura 4.22 muestra la terminal de un dispositivo identificado como ESW1. Se observan estadísticas y errores relacionados con el protocolo SNMP (Simple Network Management Protocol), en donde destacan lo siguiente:

a) **Entrada de paquetes SNMP:**

- 851 paquetes SNMP recibidos.
- Se listan varios tipos de errores, como:
- Errores por versión incorrecta de SNMP.

- Nombre de comunidad SNMP desconocido.
- Operación ilegal para el nombre de la comunidad.
- Errores de codificación.
- 18398 variables solicitadas.
- 755 PDUs (Protocol Data Units) de Get-request recibidos y 0 de Set-request.

b) **Salida de paquetes SNMP:**

- 878 paquetes enviados, con 13 errores de tamaño excesivo (paquetes demasiado grandes, máximo de 1500 bytes).
- Otros errores incluyen:
- Errores por nombres inexistentes.
- Valores incorrectos.
- Errores generales.
- Enviados: 435 PDUs de respuesta y 43 Trap PDUs.

c) **Logging SNMP:**

- El registro de SNMP está habilitado.
- Se está registrando en dos direcciones IP:
- 192.168.1.2.161, con 16 paquetes enviados, 26 descartados.
- 192.168.1.30.162, con 1 paquete enviado, ninguno descartado.

La figura 4.23 muestra un registro de eventos de SNMP traps generado por el dispositivo Cisco IOS, con un formato de *fallback*[1] visto desde la interfaz web de Zabbix.

[1]Fallback significa que el sistema está utilizando un formato alternativo, probablemente más básico, para mostrar los traps SNMP.

Figura 4.23: Registro de eventos de SNMP traps.

En este trap se envia la siguiente información:

a) **Tipo de mensaje (TRAP):** Se está registrando una acción SNMP, lo que significa que el dispositivo está enviando una notificación no solicitada a un gestor SNMP.

b) **Información de encabezado SNMP:**

- **Versión:** La versión del protocolo SNMP utilizada es la 1.
- Request ID, Error Index: La Request ID es 45 y el Error Index es 0, lo que indica que no hay errores en la operación.
- **Notification Type:** Tipo de notificación es TRAP.
- **Mensaje ID:** 0.
- **Recibido desde:** UDP [192.168.1.2]:49706 ->[192.168.1.30]:162. Aquí se indica la dirección IP del dispositivo emisor y el puerto de destino en el cual el trap fue recibido (162, puerto por defecto para SNMP traps).
- **Comunidad:** La comunidad SNMP es "public"(esto puede implicar una configuración de seguridad por defecto, que debería revisarse).
- **Uptime:** El tiempo de actividad del dispositivo.

c) **Variables:**

- OID (Identificadores de objetos): Los OIDs indicados representan diferentes variables dentro del dispositivo. Aquí hay algunos ejemplos de OIDs decodificados:
- 1.3.6.1.2.1.1.3.0: Valor del tiempo de actividad del sistema (7650289, lo que indica que el dispositivo ha estado activo aproximadamente 2 horas y 7 minutos).
- 1.3.6.1.6.3.1.1.5.2: Se refiere a un reinicio, lo que significa que el dispositivo ha sido reiniciado (en frío).
- 1.3.6.1.2.1.1.6.0: Normalmente, esto corresponde a la ubicación del sistema.
- 1.3.6.1.2.1.1.7.0 y 1.3.6.1.2.1.43.11.1.1.7.2: Otros valores relacionados con el sistema o estadísticas de la impresora.

d) **Hora y eventos:** Los registros muestran eventos con diferencia de segundos (05:06:12 y 05:06:15), lo que sugiere que los traps se generaron de manera casi simultánea.

En la figura 4.24 se muestran las instrucciones ejecutadas en el modo de configuración del enrutador Cisco, específicamente relacionadas con la habilitación de traps SNMP. Estas instrucciones configuran el enrutador para que envíe notificaciones SNMP sobre eventos relevantes.

- Cambios en la configuración del dispositivo.
- Problemas ambientales.
- Altos niveles de uso de la CPU.
- Intentos fallidos de autenticación SNMP.

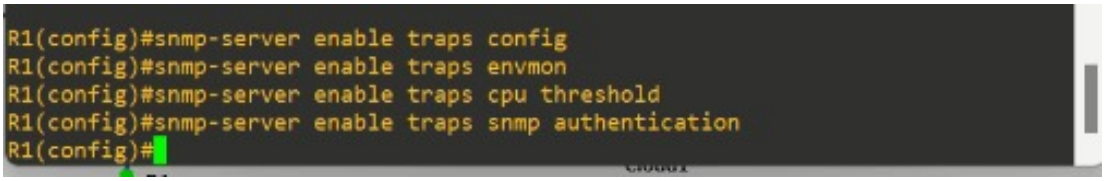


Figura 4.24: Instrucciones ejecutadas en un enrutador Cisco para la habilitación del protocolo SNMP.

Dicha configuración se ejecutó para el monitoreo del dispositivo, permitiendo notificaciones automáticas en caso de que ocurra algún problema o se realicen cambios importantes en la configuración.

a) **snmp-server enable traps config:** Esta instrucción habilita el envío de notificaciones SNMP cuando se realizan cambios en la configuración del rutador. Si se modifica la configuración, el enrutador enviará una notificación SNMP al gestor.

b) **snmp-server enable traps envmon:** Habilita notificaciones relacionados con el monitoreo ambiental del enrutador. Esto incluye aspectos como la temperatura, la fuente de alimentación. Cualquier cambio o problema con los sensores ambientales activará una alerta SNMP.

c) **snmp-server enable traps cpu threshold:** Esta instrucción habilita notificaciones cuando la utilización de la CPU del enrutador alcanza un umbral determinado. Si el uso de la CPU es demasiado alto, se enviará una notificación SNMP al gestor para alertar sobre la condición.

d) **snmp-server enable traps snmp authentication:** Habilita notificaciones SNMP relacionados con fallos de autenticación SNMP. Si alguien intenta acceder al dispositivo utilizando credenciales SNMP incorrectas o no autorizadas, el enrutador enviará una notificación para informar sobre el intento fallido.

La figura 4.25 muestra el tráfico de red en Wireshark, relacionado con el protocolo SNMP.

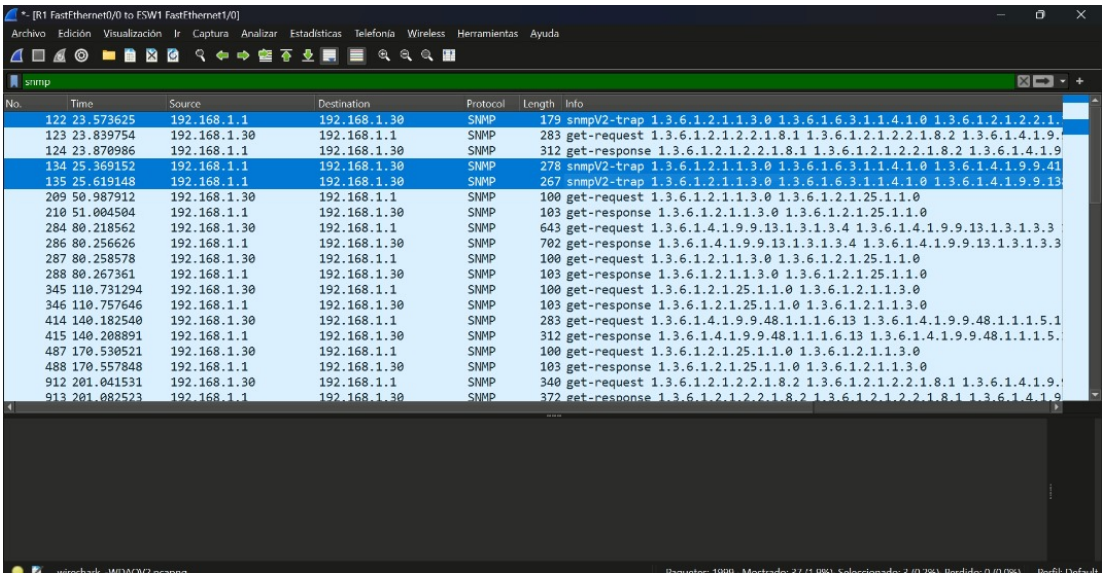


Figura 4.25: Tráfico de red en Wireshark.

Columnas mostradas:

- **No:** El número de paquete capturado, que permite identificar el orden de los paquetes en el tiempo.
- **Time:** El tiempo transcurrido en segundos desde el inicio de la captura hasta que se capturó el paquete.
- **Source:** La dirección IP de origen que envió el paquete.
- **Destination:** La dirección IP de destino que recibió el paquete.

- **Protocol:** El protocolo utilizado, que en este caso es SNMP.
- **Length:** El tamaño del paquete en bytes.
- **Info:** Información sobre el tipo de mensaje SNMP capturado (traps, get-request, get-response, etc.) seguido de los OIDs[2] correspondientes.

Análisis de los paquetes destacados:

a) **Paquetes de tipo *snmpV2-trap*:**

- **Origen:** 192.168.1.1
- **Destino:** 192.168.1.30
- Estos paquetes representan alertas (traps) SNMP enviados desde el dispositivo con IP 192.168.1.1 hacia el servidor de gestión SNMP con IP 192.168.1.30.
- Los traps SNMPv2 son notificaciones enviadas por el dispositivo cuando ocurre algún evento relevante, como un cambio en la configuración o un evento ambiental.
- El OID que aparece en el campo Info (1.3.6.1.2.1.1.3.0) se refiere al uptime del sistema, que indica el tiempo que lleva encendido el dispositivo en milisegundos.

b) **Paquetes de tipo *get-request* y *get-response*:**

- Estos paquetes son peticiones SNMP (get-request) y sus respectivas respuestas (get-response).
- El origen 192.168.1.30 (servidor SNMP) solicita información al dispositivo con IP 192.168.1.1. El dispositivo responde con los valores solicitados.
- El OID (Object Identifier, por sus siglas en inglés) de las peticiones y respuestas (por ejemplo, 1.3.6.1.2.1.2.2.1.8 o 1.3.6.1.2.1.1.5.0) indican el tipo de información específica que está siendo consultada. Estos OIDs pueden corresponder, entre otros, al estado de las interfaces de red, a los nombres de host, y a otras características de los dispositivos.

Observaciones:

- **Origen y destino:** El enrutador identificado con la IP 192.168.1.1 está enviando notificaciones SNMP al servidor de gestión con IP 192.168.1.30. Además, el servidor SNMP está haciendo peticiones de información (get-request) y recibiendo respuestas (get-response).
- **SNMPv2-Trap:** Las notificaciones en SNMPv2 se utilizan para alertar al servidor SNMP sobre eventos relevantes en el dispositivo, como cambios de configuración, problemas de red, o reinicios del sistema.

[2]Los OIDs son identificadores únicos que permiten a los sistemas de gestión de red realizar consultas y recibir información específica de dispositivos a través de SNMP

- **Comunicaciones activas:** Hay un intercambio continuo de información entre el dispositivo y el servidor SNMP. El servidor solicita información específica y el enrutador responde.

La figura 4.26 muestra el análisis de un paquete SNMP, específicamente una alerta o **trap** SNMP **SNMPv2**.

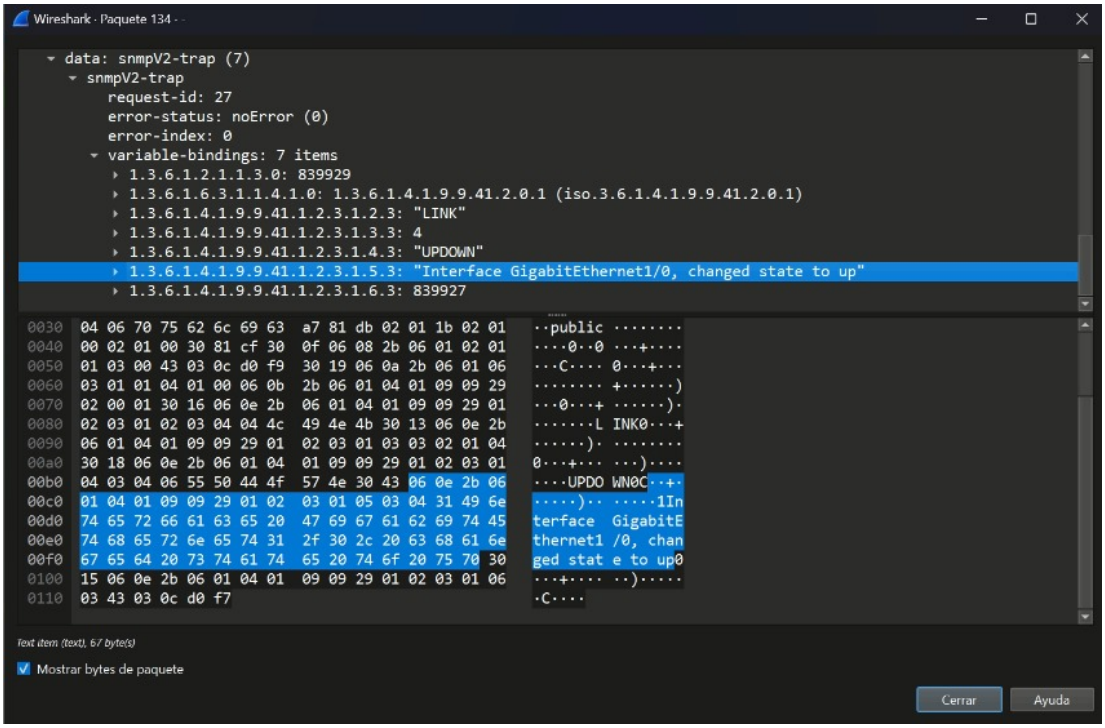


Figura 4.26: Análisis de alertas SNMPv2.

a) **Protocolo SNMPv2-Trap:** Este es un mensaje enviado por un dispositivo para notificar un evento significativo (en este caso, parece ser un cambio en el estado de una interfaz).

b) **Identificadores de objetos (OID):** Se observan varios OIDs como 1.3.6.1.6.3.1.1.5.3, que en el sistema SNMP representa un objeto específico. Cada uno de estos OIDs corresponde a un atributo específico en la información que el dispositivo está reportando.

c) **Texto visible:** Se muestra que la interfaz GigabitEthernet1/0 ha cambiado su estado a "arriba", lo que indica que la interfaz está activa nuevamente.

d) **Mensajes clave:**

- **OID 1.3.6.1.6.3.1.1.5.3:** Representa una notificación genérica, conocida como "enlazar", que indica que una interfaz de red ha vuelto a estar operativa.
- **Texto asociado:** interfaz GigabitEthernet1/0, "cambió el estado a arriba" (changed state to up), confirmando el evento de activación de la interfaz.

La figura 4.27 presenta un análisis de traps SNMPv2.

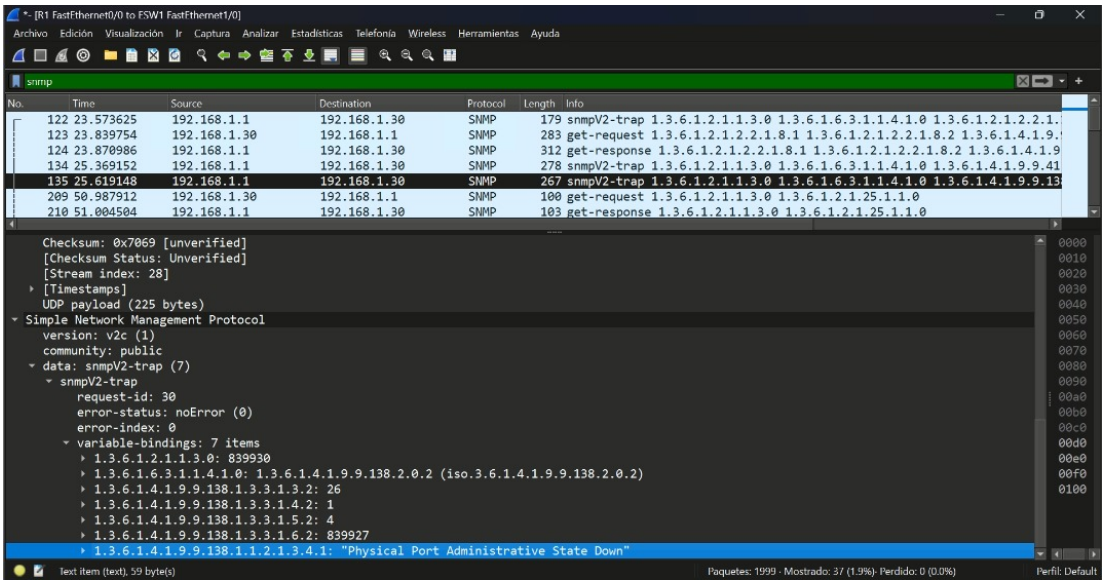


Figura 4.27: Continuación de análisis de alertas SNMPv2.

Detalles del paquete:

a) **Protocolo:** El protocolo es SNMPv2c.

b) **Fuente:** La dirección IP 192.168.1.1 envía una alerta SNMP hacia la IP de destino 192.168.1.30.

c) **Trap SNMP:**

- Tipo de mensaje: *snmpV2-trap*.
- Request ID: 30.
- Error status: noError (0), lo que indica que no hay errores en el mensaje.
- Error index: 0.

OID (Identificadores de objetos) El trap incluye varios OIDs:

- **1.3.6.1.4.1.9.9.138.1.2.3.4:** Este OID está asociado con la descripción "Estado administrativo del puerto físico inactivo - (Physical Port Administrative State Down)", que indica que el puerto físico ha sido deshabilitado administrativamente. Esto significa que el puerto ha sido desactivado manualmente o por alguna configuración.

Mensaje: El mensaje indica que el estado administrativo del puerto físico ha cambiado a "abajo", es decir, que la interfaz física fue desactivada.

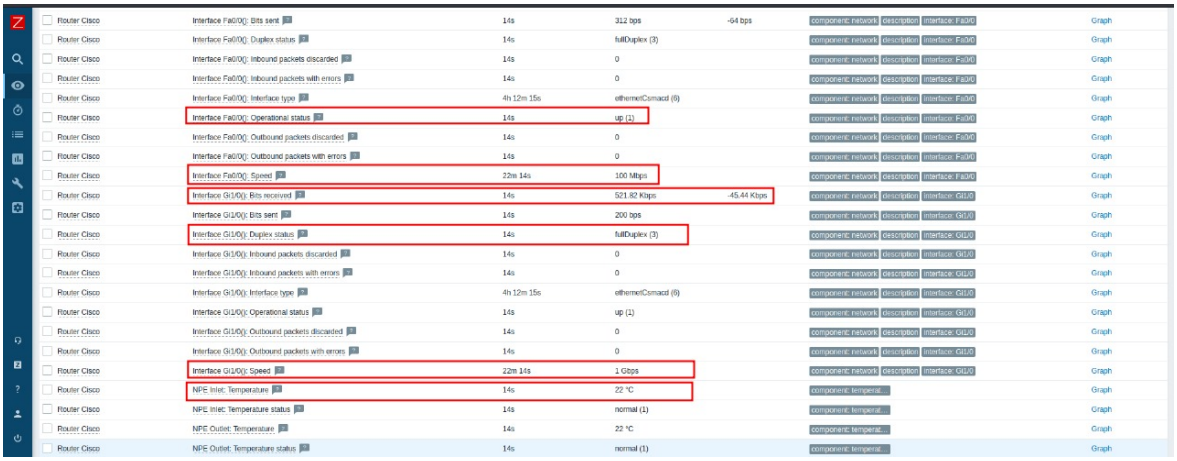


Figura 4.28: Elementos de monitoreo

Analizar el uso de la CPU en dispositivos de red, como enrutadores, es fundamental para garantizar su óptimo funcionamiento. La gráfica en la figura 4.4 permite observar el porcentaje de utilización de la CPU, con un intervalo entre 0 % y 100 %, reflejando la capacidad operativa del hardware durante diferentes períodos de tiempo. Comparar estos porcentajes entre el enrutadorA-CPU y el enrutadorB-CPU facilita la identificación de patrones de uso y posibles cuellos de botella.

La figura 4.29 muestra dos resultados del monitoreo de un enrutador, obtenidos a través de Zabbix.

Figura 4.29: Gráficas del consumo de recursos

- **Gráfica superior:** Representa la utilización de CPU del router, con un porcentaje en el eje Y y el tiempo en el eje X. Parece que el CPU se mantiene en niveles bajos, sin picos de uso significativos.
- **Gráfica inferior:** Muestra el tráfico de red en una interfaz específica (Fa0/1). La escala en el eje Y indica la cantidad de datos transmitidos o recibidos, pero no se observan picos de tráfico importantes.

Ambas gráficas muestran valores estables, lo que indica que no hay problemas evidentes de alto consumo de CPU ni de congestión en la red. Sin embargo, la zona sombreada a la derecha podría indicar una falta de datos recientes o un período de inactividad en la recopilación de métricas.

La figura 4.30 muestra el monitoreo del tráfico de red de una interfaz en Zabbix.

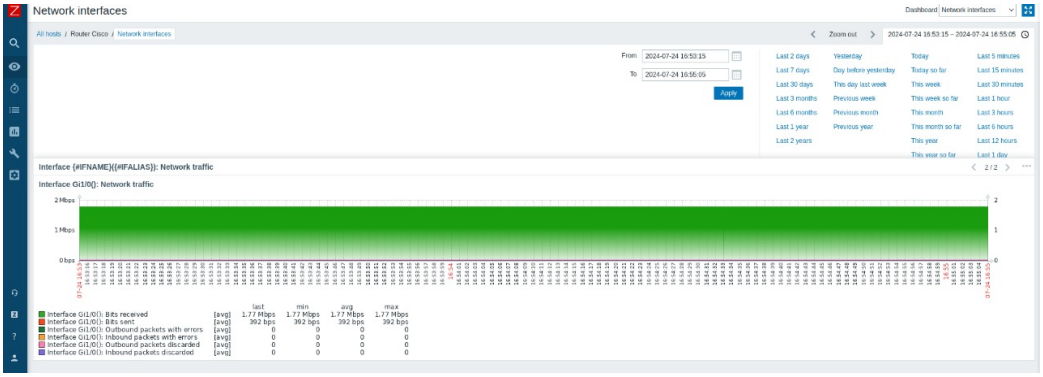


Figura 4.30: Interfaz de monitoreo y tráfico de red

En Zabbix, el monitoreo de problemas es indispensable para la administración eficaz de la red y para optimizar la gestión de alertas. Detectar problemas implica identificar estados críticos, como la falta de respuesta a pings SNMP, lo cual puede indicar un servidor apagado o desconexiones.

En la figura 4.30 se observa lo siguiente:

- **Gráfica principal:** Representa el tráfico de la interfaz Gi1/0. Se observa una carga de datos estable en la interfaz, sin grandes cambios. Asimismo, la intensidad del color verde sugiere un tráfico constante, sin interrupciones significativas.
- **Tabla inferior:** Presenta métricas detalladas del tráfico, incluyendo paquetes recibidos, transmitidos y errores. No se observan errores ni paquetes descartados, lo que indica que la interfaz está funcionando correctamente.

El tráfico en la interfaz es estable y sin anomalías. No hay indicios de pérdida de paquetes ni errores, lo que sugiere un funcionamiento óptimo de la conexión en esta interfaz específica.

En el enrutador, la Figura 4.31 nos muestra la falta de respuesta a ping. Esto puede deberse a varias razones, como problemas de conectividad, configuración incorrecta del enrutador o fallos en la red.

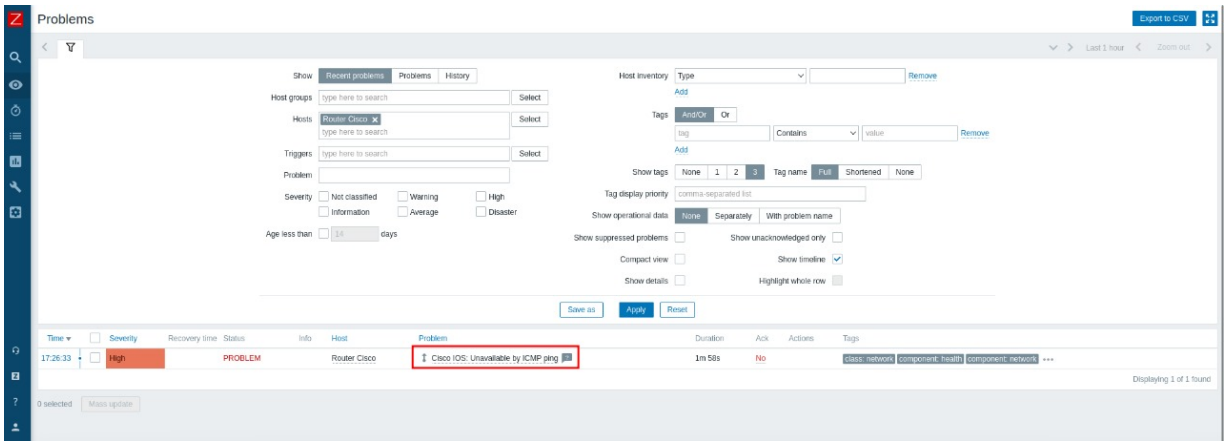


Figura 4.31: Alerta en enrutador por falta de respuesta a pings por el protocolo ICMP

La figura 4.32 muestra la sección de “Problemas” en Zabbix, donde se registran eventos relacionados con los dispositivos monitoreados.

El enrutador Cisco experimentó una interrupción temporal debido a un reinicio, lo que ocasionó la pérdida de respuesta al ICMP durante 6 minutos.

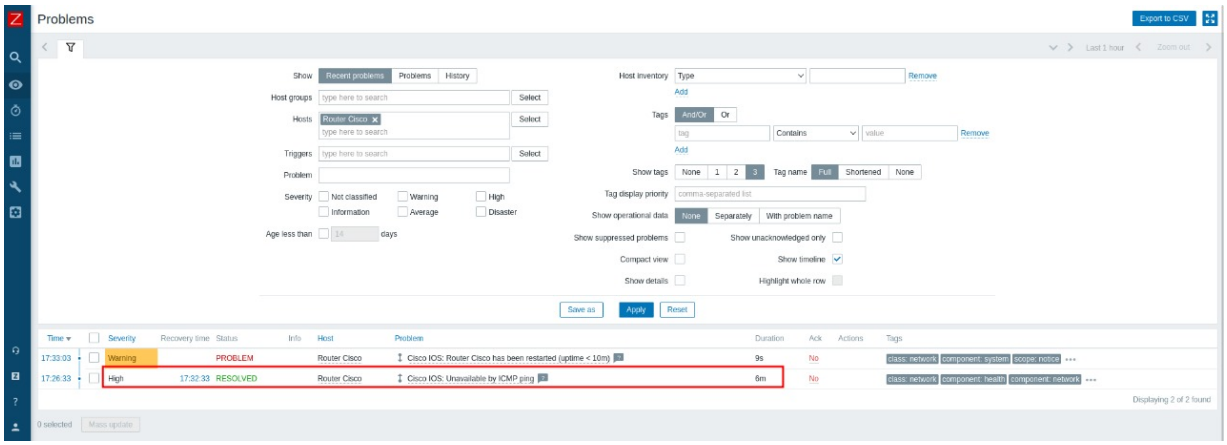


Figura 4.32: Detección de enrutador con estado de resuelto"

Al analizar un reporte de disponibilidad, se implica examinar las barras verdes para determinar los períodos de operatividad y las barras rojas para identificar incidencias. Realizar este análisis permite:

- Monitorear la fiabilidad, observando la continuidad operativa del sistema.
- Detectar problemas, reconociendo patrones de interrupciones que requieren atención.
- Organizar el mantenimiento, programando acciones preventivas para optimizar la funcionalidad.

- Evaluar el rendimiento, utilizando los datos para fundamentar decisiones estratégicas y mejorar la gestión del sistema.

En el informe de disponibilidad generado por Zabbix, la Figura 4.33 muestra el monitoreo de la disponibilidad durante un período específico. Este informe proporciona una visión detallada del tiempo de actividad y las interrupciones del sistema.

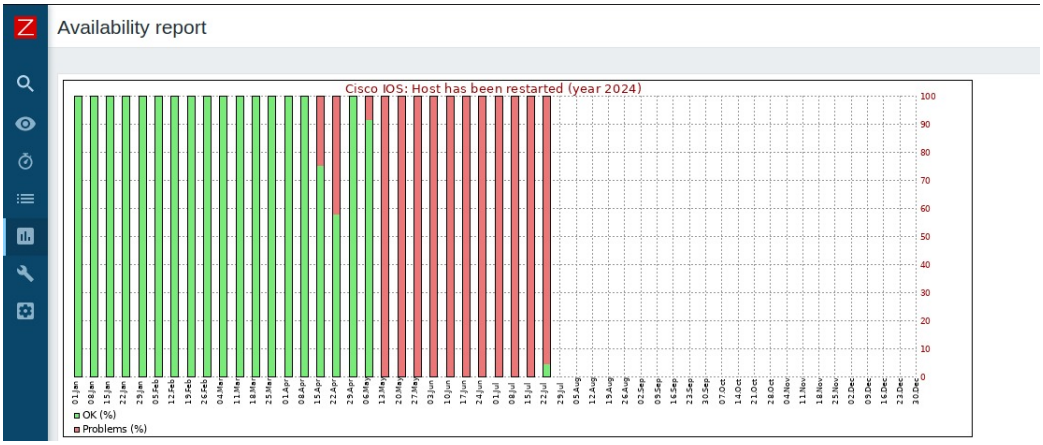


Figura 4.33: Monitorear la disponibilidad en periodo de tiempo

Los 100 *triggers* (o "disparadores" en español) más activos en la herramienta de monitoreo Zabbix permiten observar el estado de cada uno para determinar si se encuentra en condición "OK" o "Problema" como se observa en la figura 4.34. Cada situación se describe para identificar problemas específicos, como un "tiempo de respuesta de ping ICMP alto" o que "el host ha sido reiniciado". Además, se evalúa la severidad de los problemas mediante una escala de colores que va desde verde para situaciones "No clasificadas" hasta rojo para aquellas consideradas "Altas". Se contabilizan los cambios de estado de los disparadores para medir la frecuencia de las alteraciones en su condición.

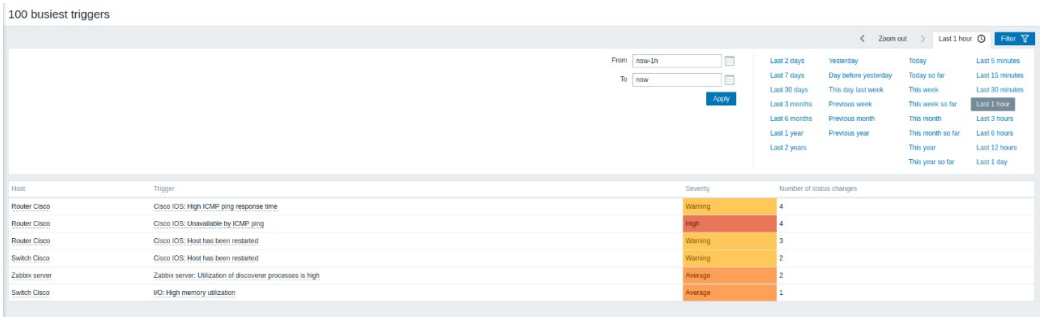


Figura 4.34: Estado de dispositivos

La figura 4.35 del panel de monitoreo de Zabbix ayuda a identificar y comprender las métricas clave para optimizar el rendimiento del sistema. El monitoreo de la temperatura

de entrada del NPE tiene como objetivo prevenir el sobrecalentamiento y garantizar la operatividad del equipo. La observación de los bits enviados y recibidos permite evaluar la eficiencia de la transmisión de datos a través de la interfaz Gi1/0. Finalmente, la utilización de la CPU es fundamental para asegurar que el procesamiento de datos se realice dentro de los parámetros deseados.

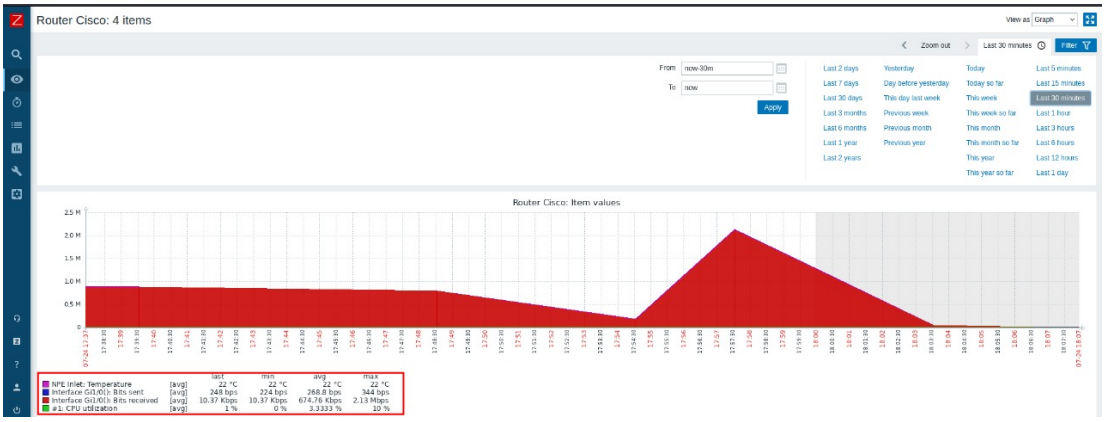


Figura 4.35: Grafica de rendimiento y estado de interfaz en Zabbix

Una vez identificados los eventos críticos mediante traps SNMP, Zabbix envía notificaciones inmediatas a los administradores, como se detalla en la siguiente sección.

4.7. Sistema de notificaciones vía correo electrónico

Zabbix nos permite configurar el envío de alertas y notificaciones utilizando varios métodos, como correo electrónico, mensajes de texto (SMS), Slack e incluso sistemas CRM (Customer Relationship Management) para gestionar incidencias y el seguimiento de eventos críticos de manera efectiva.

En este proyecto se implementó un sistema de notificaciones vía correo electrónico, en el cual se definió a los administradores de la red como destinatarios específicos de las alertas. Se establecieron umbrales para activarlas y se personalizaron los mensajes para asegurar una comunicación efectiva de la información relevante.

La integración con el correo electrónico garantiza la entrega confiable y puntual de las notificaciones, cumpliendo con los estándares de disponibilidad y los tiempos de respuesta necesarios para gestionar los incidentes de manera efectiva y mantener la continuidad operativa de la red.

Por ejemplo, se pueden establecer alertas para informar si un servidor está operativo o si ha dejado de funcionar. Además, es posible recibir notificaciones sobre el descubrimiento de nuevos dispositivos en la red, lo cual facilita la gestión y el seguimiento de los recursos.

Las figuras representan las notificaciones generadas por Zabbix. La función principal es detectar dispositivos en la red, según lo indicado por la regla de descubrimiento **Descubrimiento de red**. Cada dispositivo tiene una dirección IP única; en este caso, es 192.168.1.40. El estado "UP" confirma que el dispositivo está activo y operando sin interrupciones durante el tiempo indicado. Asimismo, el servicio **ICMP ping**, esencial para verificar la conectividad en la red, muestra un estado "UP", asegurando su funcionamiento continuo por el mismo período.

Figura 4.36: Notificación de servidor activo

La notificación de Zabbix presentada indica que se ha realizado un descubrimiento de red, identificando un dispositivo con dirección IP 192.168.1.1 y DNS *gateway*, que representa la puerta de enlace predeterminada. El estado operativo del dispositivo y su servicio ICMP ping es "UP", lo que significa que están funcionando correctamente. Ambos, el dispositivo y su servicio, han estado activos durante 45 minutos y 46 segundos.

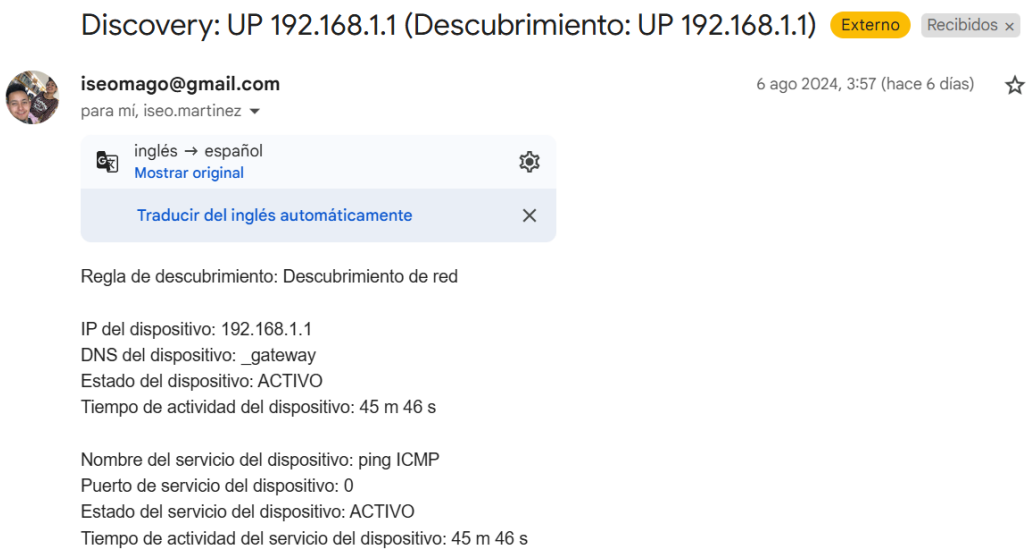


Figura 4.37: Notificación del estado del enrutador como activo

La ausencia de un DNS específico sugiere que no se ha asignado un nombre de dominio al dispositivo. El estado es *inactivo* indica que el dispositivo y su servicio de ICMP ping no están operativos. Además, el tiempo de actividad registrado de 13 segundos refleja la duración de este estado *inactivo*. El puerto 0, generalmente reservado para propósitos especiales, no está en uso.

Figura 4.38: Notificación del estado del enrutador como inactivo

4.8. Sistema de notificaciones vía telegram

Con el fin de lograr una comunicación más fluida entre el servidor de monitoreo y los administradores de la red, se implementó un sistema de notificaciones a través de Telegram, en el cual se designa a los administradores de la red como destinatarios específicos de las alertas. Se establecen umbrales para activarlas y se personalizan los mensajes para garantizar una comunicación efectiva de la información relevante.

En esta integración, se establecieron alertas para informar si un servidor está operativo o ha dejado de funcionar, así como para el descubrimiento de nuevos dispositivos en la red, lo cual facilita la gestión y el seguimiento de los recursos disponibles en ella.

La integración con Telegram garantizará la entrega confiable y puntual de las notificaciones, lo que contribuye a cumplir con los estándares de disponibilidad y con los tiempos de respuesta necesarios para gestionar los incidentes de manera efectiva, así como para mantener la continuidad operativa de la red.

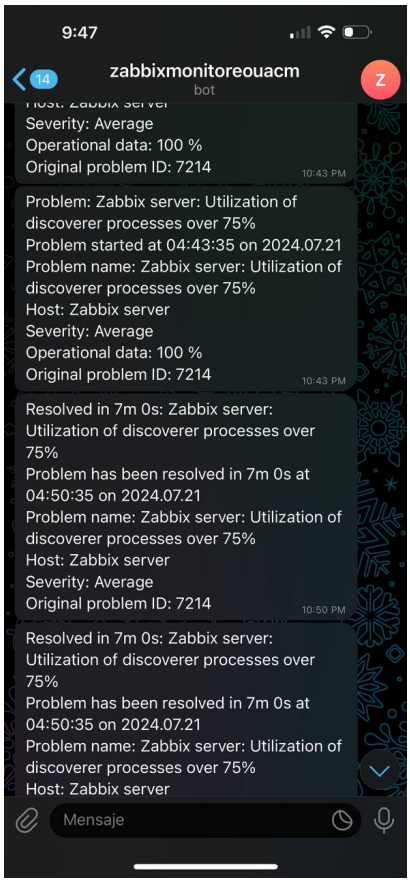


Figura 4.39: Notificaciones via telegram

La figura 4.39 muestra varias notificaciones enviadas por un bot de monitoreo de Zabbix a través de la aplicación Telegram. Estas notificaciones alertan sobre problemas detectados

en un servidor de Zabbix, específicamente relacionados con la ütilización de los procesos de descubrimiento que han superado un umbral crítico; en este caso, el 75 %.

A continuación, se detallan los elementos presentes en las notificaciones:

- **Fecha y hora del problema:** Cada notificación incluye la fecha y hora exactas del inicio y resolución del problema. Por ejemplo, uno de los problemas se reporta el 21 de julio de 2024 a las 04:43:35.
- **ID del problema:** Las alertas incluyen un identificador único (ID) para cada problema reportado, como el 7214, que permite un seguimiento fácil y preciso de cada incidente.
- **Mensaje de alerta:** Describir el problema, como Utilization of discoverer processes over 75 % (Utilización de los procesos de descubrimiento por encima del 75 %).
- **Datos operativos:** Mostrar el porcentaje de utilización, por ejemplo, Operational data: 100 %.
- **Severidad:** Cada alerta incluye una calificación de severidad para indicar el nivel de urgencia del problema. En este caso, la severidad promedio (average), lo que sugiere que el problema es moderadamente importante pero no crítico.
- **Tiempo de resolución:** Las notificaciones también indican el tiempo que tomó resolver el problema. En el ejemplo, la alerta informa que el problema fue resuelto en 7 minutos.

Las notificaciones en esta figura representan alertas automáticas generadas por el sistema Zabbix para monitorear la salud de un servidor. Los elementos clave incluyen la fecha y hora del incidente, la descripción del problema, el nivel de severidad y el tiempo de resolución. Esta información es útil para ilustrar cómo las herramientas de monitoreo, como Zabbix, proporcionan visibilidad en tiempo real sobre el rendimiento y la disponibilidad de los servidores, mejorando así la capacidad de respuesta ante posibles fallos.

Durante la implementación de las notificaciones vía Telegram en Zabbix, surgieron inicialmente una serie de problemas que impidieron la correcta entrega de alertas a los administradores de la red, como se observa en la Figura 4.40. Un ejemplo de esto fue la falla en la notificación sobre la caída del dispositivo con la IP 192.168.1.15, que estuvo inactivo durante un período prolongado sin que el mensaje de alerta llegara a su destinatario. Sin embargo, tras realizar los ajustes necesarios en la configuración, se lograron resolver estos inconvenientes y actualmente el sistema de notificaciones vía Telegram está funcionando correctamente. Esto garantiza que las alertas se entreguen de manera oportuna, permitiendo una respuesta rápida ante incidentes críticos y mejorando significativamente la capacidad de gestión de la infraestructura de red.

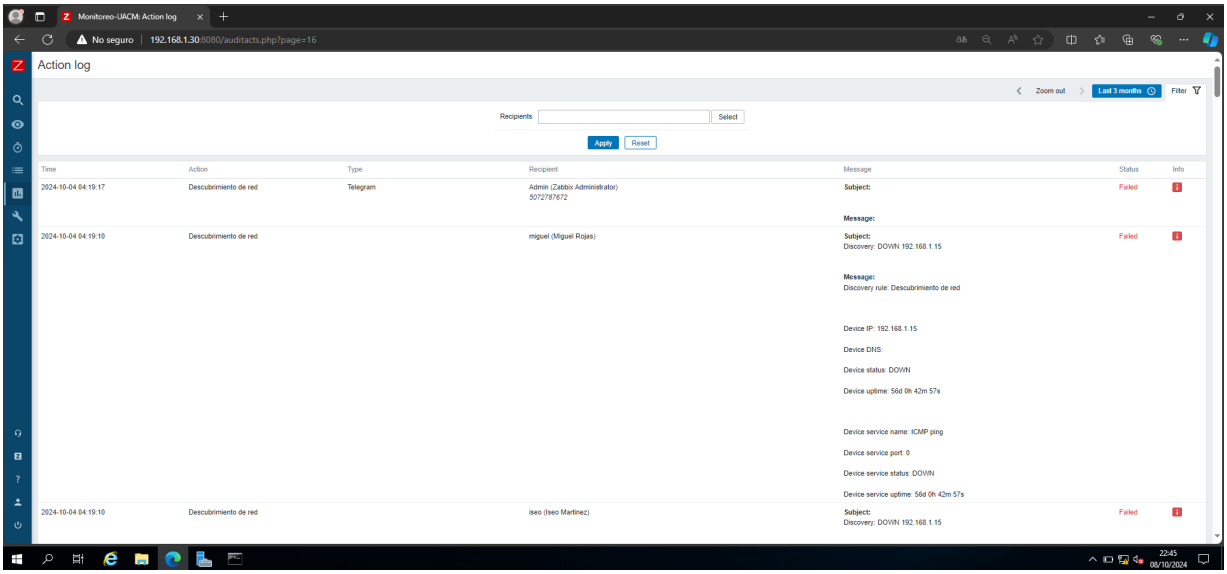


Figura 4.40: Registro de acciones de Zabbix.

La Figura 4.40 muestra el registro de acciones de Zabbix, donde se puede observar el historial de notificaciones enviadas y los problemas que se presentaron durante la implementación inicial.

a) **Detalles de las Acciones**

Tipo de acción:

- La primera acción utiliza un canal de notificación a través de Telegram, y está destinada al usuario administrador de Zabbix, etiquetado como Admin (Zabbix Administrator) con el número de teléfono 5072787672.
- La segunda acción no especifica un canal de notificación, pero está dirigida a un usuario llamado Miguel Rojas.
- **Estado:** Ambas acciones fallaron, lo que indica que los mensajes de alerta no pudieron ser entregados correctamente a los destinatarios.

b) **Mensajes:**

- **Contenido de los mensajes:** El mensaje se refiere al descubrimiento de que el dispositivo con la IP 192.168.1.15 está DOWN, lo que significa que está inactivo o desconectado.
- El dispositivo ha estado en este estado por un tiempo considerable, con un tiempo de actividad previo de 56 días, 0 horas, 42 minutos y 57 segundos.
- Se indica que el servicio que se estaba utilizando para comprobar la conectividad del dispositivo es ICMP ping, que es comúnmente usado para verificar la disponibilidad de dispositivos de red. El puerto del servicio es 0, lo cual es normal en el caso de ICMP, ya que no utiliza un puerto específico.

Las acciones intentaron enviar notificaciones a los administradores, pero fallaron. El dispositivo identificado por la IP 192.168.1.15 ha estado fuera de servicio, y esto ha sido registrado en el proceso de descubrimiento de red. Esto sugiere que puede haber problemas con la red o con la configuración de notificaciones en Zabbix, ya que las alertas críticas no llegaron a su destinatario, lo que puede retrasar la respuesta a incidentes importantes.

4.9. Dashboards Grafana

Para mejorar la visualización de los datos recolectados mediante el protocolo SNMP, se configuró un *dashboard* (o "panel" en español) en Grafana que permite a los administradores de red monitorear en tiempo real el estado y rendimiento de diversos dispositivos de la infraestructura. Grafana proporciona una interfaz gráfica intuitiva y altamente personalizable, lo cual facilita la interpretación de datos complejos y promueve una toma de decisiones más rápida y fundamentada.

Para ello, también se debe configurar la API[3]. La figura 4.41 muestra el proceso de autenticación en el servidor Zabbix a través de la API JSON-RPC, usando *curl*[4] en la terminal. El servidor ha respondido con éxito, proporcionando un token de autenticación que puede ser utilizado para realizar otras operaciones mediante la API.

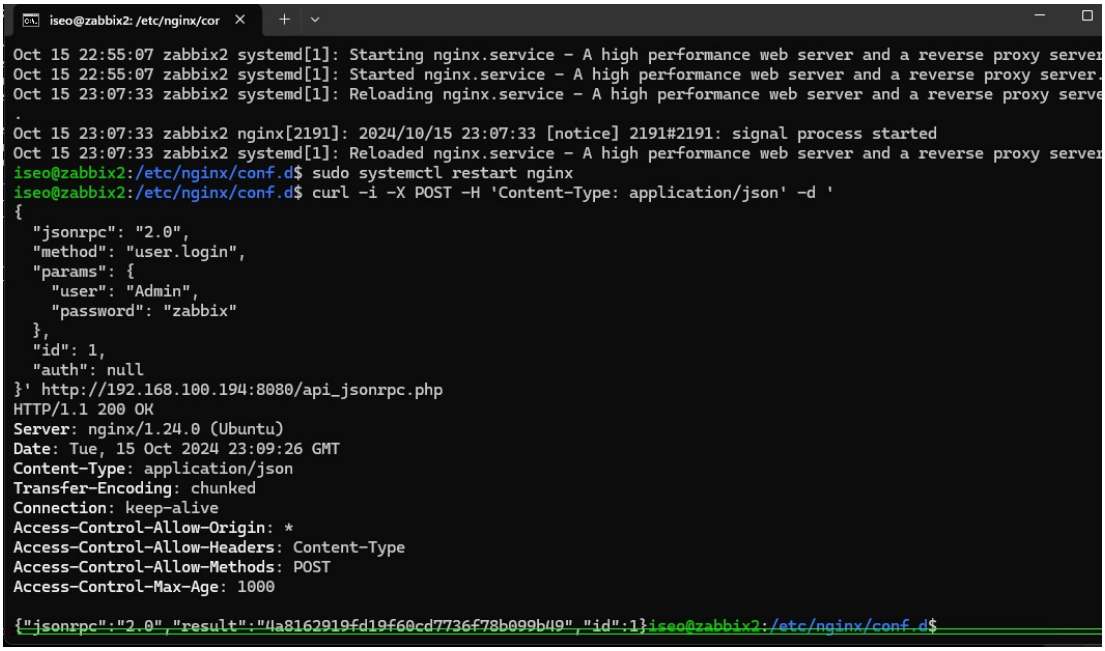


Figura 4.41: Autenticación en servidor Zabbix por medio de la API.

[3]Application Programming interfaz o interfaz de Programación de Aplicaciones) es un conjunto de definiciones y protocolos que permite que diferentes aplicaciones se comuniquen entre sí.

[4]*curl* es una herramienta de línea de instrucciones utilizada para transferir datos desde, utilizando una variedad de protocolos, como HTTP, HTTPS, FTP

Los *dashboards* o paneles de visualización creados en Grafana integran datos obtenidos a través de Zabbix y muestran métricas clave, tales como el uso de CPU, memoria, ancho de banda y estado de interfaces en dispositivos de red, incluidos enrutadores y conmutadores de Cisco.

Cada dashboard proporciona información clave de diferentes áreas del sistema, como:

- **Rendimiento del servidor:** CPU, memoria, discos y uso de red.
- **Estado de servicios y aplicaciones:** Disponibilidad y latencia.
- **Monitoreo de red:** Latencia, paquetes y disponibilidad de interfaces.
- **Estado general del sistema:** Resumen de alertas y problemas críticos.

La figura 4.60 muestra un panel de control en Grafana integrado con Zabbix, diseñado para monitorear diferentes aspectos de la infraestructura de red y servidores.

Figura 4.42: Dashboard 1 de Grafana integrado con Zabbix.

4.9.1. Análisis del dashboard general

a) **Gráficas de uso de disco y red:**

- En la parte superior izquierda, se observa una gráfica que muestra la tasa de lectura y escritura de discos, junto con la cantidad de bits enviados y recibidos a través de la red. Esto permite monitorear el rendimiento de las interfaces de red (como enp0s3 y sda) y detectar posibles cuellos de botella en el sistema.

b) **Estado del servidor Ubuntu:**

- En el centro izquierdo, hay una sección que muestra "el estado de salud" del servidor Ubuntu. Se incluyen métricas como la utilización del CPU, la memoria,

la carga promedio y el estado del servidor Zabbix (tendencia de caché y uso de la memoria del servidor de monitoreo). Esta gráfica ayuda a evaluar el uso de los recursos y a detectar anomalías en el rendimiento del servidor.

c) **Estado del enrutador Cisco:**

- A la derecha, se observa una gráfica del tráfico de bits a través de las interfaces del enrutador Cisco y otra con el estado de componentes como la fuente de alimentación y la respuesta ICMP. Esta es importante para monitorear la estabilidad y el rendimiento de los dispositivos de red críticos.

d) **Paneles de resumen:**

- En el centro inferior, los paneles muestran resumenes del espacio en disco usado (8.45 GiB), el tiempo de arranque del sistema Linux, el número de CPUs (1), y el número total de procesos (192).

e) **Rendimiento del conmutador Cisco:**

- En la parte inferior central, muestra el uso del procesador y de la memoria del conmutador Cisco.

f) **Métricas de Windows Server:**

- En la parte inferior derecha, se incluyen métricas de un servidor Windows, como la utilización de la CPU y el espacio de intercambio, así como el tráfico de red.

La figura 4.43 muestra un panel de control que es útil para monitorear la estabilidad y el rendimiento general del servidor Ubuntu, asegurando que no haya sobrecargas ni problemas significativos en cuanto a los recursos principales (CPU, memoria, carga del sistema).

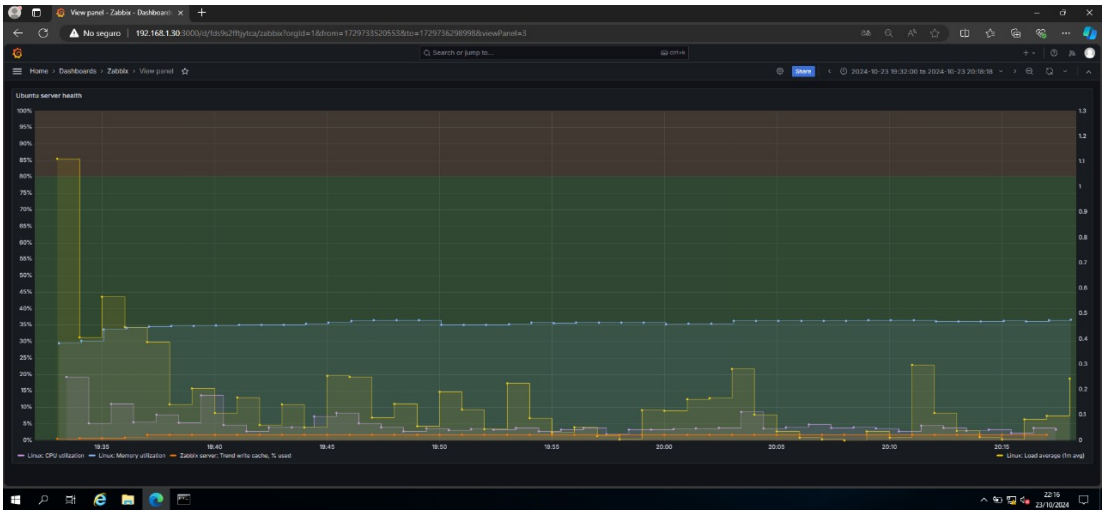


Figura 4.43: Panel para el monitoreo general del servidor Ubuntu.

a) **Métrica de Utilización de CPU (Linux CPU utilization):**

- Representada por la línea amarilla, en la figura 4.43 muestra el uso de la CPU a lo largo del tiempo. Al inicio se ve un pico elevado que llega casi al 80 %, pero rápidamente baja a niveles estables, manteniéndose por debajo del 10 % en la mayor parte del tiempo monitoreado. Lo que indica que el servidor tuvo un momento de alta carga al comienzo del intervalo, pero luego regresó a un estado normal y controlado.

b) **Utilización de Memoria (Linux Memory utilization):**

- La línea azul en la figura 4.43 muestra el uso de la memoria en el servidor. Este valor se mantiene constante alrededor del 30 %, lo que indica que el servidor no ha experimentado grandes variaciones en el uso de la memoria, siendo relativamente estable durante todo el período monitoreado.

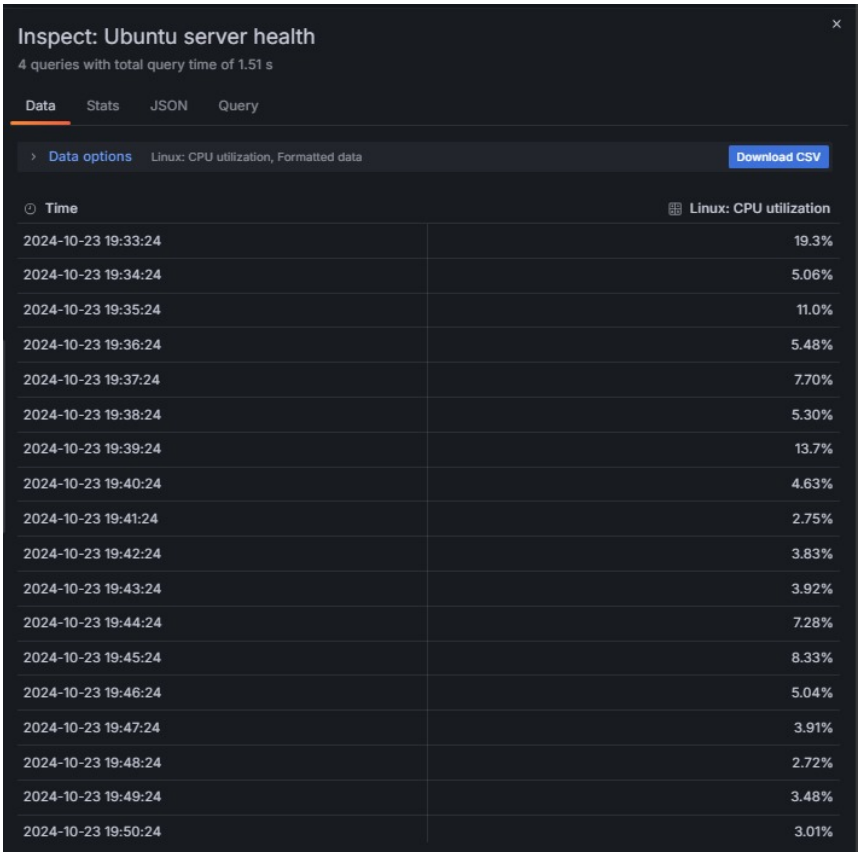
c) **Tendencia del Caché de Escritura de Zabbix (Zabbix server: Trend write cache):**

- Esta métrica, representada por las barras de color beige, en la figura 4.43 refleja el uso del caché de escritura del servidor Zabbix. Los valores se mantienen bajos con cambios ligeramente en torno al 0 % - 5 %, lo que sugiere que no hay una gran carga en el sistema de monitoreo en términos de la escritura en el caché.

d) **Promedio de carga (Linux Load average):**

- Representado por la línea verde, en la figura 4.43 muestra el promedio de carga del servidor se muestra en el eje derecho. El promedio de carga en Linux mide cuántos procesos están activos y esperando ser ejecutados por la CPU. El valor se mantiene bajo, alrededor de 0.5 o menos, lo que es un indicador de que el servidor no está sobrecargado y los procesos no presentan problemas.

La figura 4.44 presenta una tabla que monitorea la *salud del servidor Ubuntu* en relación con el uso del CPU, registrada el 23 de octubre de 2024, desde las 19:33:24 hasta las 19:50:24. La figura mencionada muestra las variaciones en el uso del CPU durante el período especificado, proporcionando una visión detallada del rendimiento del servidor.



Time	Linux: CPU utilization
2024-10-23 19:33:24	19.3%
2024-10-23 19:34:24	5.06%
2024-10-23 19:35:24	11.0%
2024-10-23 19:36:24	5.48%
2024-10-23 19:37:24	7.70%
2024-10-23 19:38:24	5.30%
2024-10-23 19:39:24	13.7%
2024-10-23 19:40:24	4.63%
2024-10-23 19:41:24	2.75%
2024-10-23 19:42:24	3.83%
2024-10-23 19:43:24	3.92%
2024-10-23 19:44:24	7.28%
2024-10-23 19:45:24	8.33%
2024-10-23 19:46:24	5.04%
2024-10-23 19:47:24	3.91%
2024-10-23 19:48:24	2.72%
2024-10-23 19:49:24	3.48%
2024-10-23 19:50:24	3.01%

Figura 4.44: Datos de monitoreo sobre el uso de CPU.

a) **Rango de utilización de CPU:** En la utilización del CPU se observan cambios entre un mínimo de 2.72 % y un máximo de 19.3 %.

b) **Picos de uso:**

- El valor más alto, 19.3 %, ocurre al inicio del registro (19:33:24).
- También hay otro pico de 13.7 % en el minuto 19:39:24.

c) **Estabilidad:** La utilización de la CPU disminuye en general después del pico inicial, con variaciones moderadas y estables en valores más bajos, alrededor del 3-8 % hacia el final del registro.

Una gráfica muestra la carga del CPU durante casi dos horas, revelando patrones de uso y facilitando el análisis de rendimiento. La figura 4.45 ofrece un análisis de dos horas del "Linux: CPU utilization" con datos cada segundo.

Figura 4.45: Variación de carga en el CPU del servidor.

a) **Picos de uso:** Se observan picos altos al inicio y en varios momentos, alcanzando hasta aproximadamente un 13 %. Estos picos representan periodos de mayor carga o demanda en el procesador.

b) **Patrones de variación:** Hay variaciones frecuentes, con el uso de la CPU subiendo y bajando en forma de picos y valles, lo que sugiere que el servidor experimenta momentos alternados de carga alta y baja.

c) **Tendencia general:** Después de los primeros picos, el uso tiende a disminuir ligeramente y estabilizarse en un rango más bajo, de aproximadamente entre 2.5 % y 5 %.

La figura 4.46 exhibe el "tráfico de red" y "uso de disco" el 23 de octubre de 2024, incluyendo las tasas de escritura/lectura, así como los datos enviados/recibidos por la interfaz enp0s3.

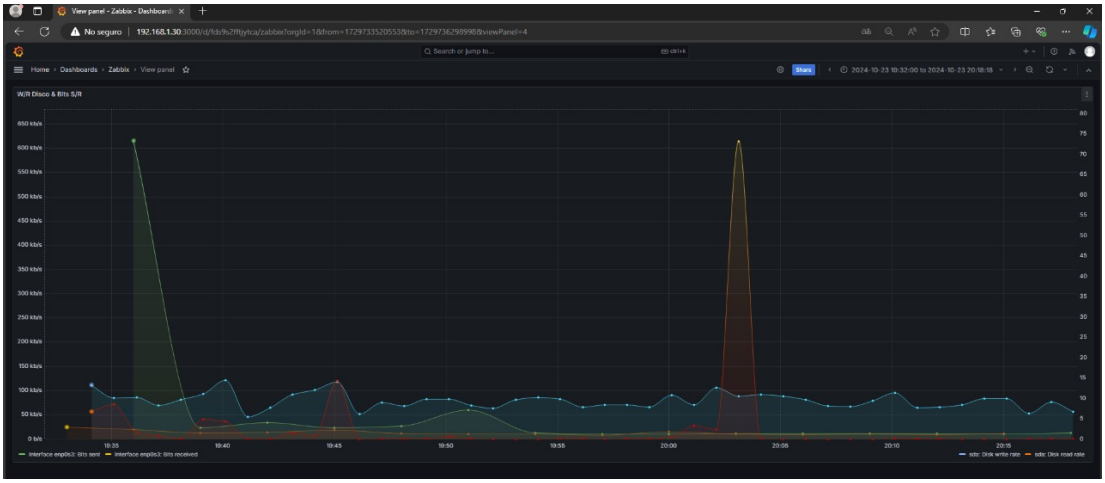


Figura 4.46: Gráfica sobre el tráfico de red y el uso de disco.

a) **Tráfico de Red (Bits enviados y recibidos):**

- La línea verde en la figura 4.46 representa los bits enviados y muestra un pico significativo que alcanza aproximadamente 600 kb/s alrededor de las 19:35 hrs.
- Después ese pico, el tráfico enviado cae rápidamente y permanece bajo hasta casi el final del periodo, donde se observa un aumento similar.
- La línea amarilla en la figura 4.46 muestra los bits recibidos, que también tienen un pico pero a un nivel menor que los enviados. Luego, se mantiene estable en valores bajos.

b) **Uso de Disco (Escritura y Lectura):**

- La línea azul en la figura 4.46 muestra la tasa de escritura en el disco (sdb2), manteniéndose en un rango bajo y estable, con ligeros cambios alrededor de 5-10 unidades.
- La línea roja n la figura 4.46 muestra la tasa de lectura del disco, que permanece cercana a cero, indicando actividad mínima de lectura.

La figura 4.47 detalla las métricas del enrutador Cisco.

Figura 4.47: Métricas del enrutador Cisco.

a) **Interfaz Gi0/0, Bits Sent y Bits Received:**

- En la figura 4.47, los bits enviados se representan con una línea amarilla y los bits recibidos con una línea azul en la interfaz GigabitEthernet 0/0.
- El tráfico enviado muestra un pico inicial y luego una disminución progresiva, mientras que el tráfico recibido permanece en niveles bajos y estables durante el periodo observado.

b) **AC Power Supply, Power Supply Status:** En la figura 4.47, se observa una línea morada que permanece en un estado estable, indicando que la fuente de alimentación está operativa sin cambios o interrupciones visibles.

c) **Activo/Inactivo Used Memory y NPE Init Temperature:**

- En la misma figura 4.47, la memoria usada (línea azul claro) y la temperatura inicial de la NPE (línea naranja) se mantienen constantes, lo que indica estabilidad del sistema a lo largo del tiempo monitoreado.

La figura 4.48 muestra el monitoreo del conmutador Cisco en Grafana.

Figura 4.48: Métricas del conmutador Cisco.

a) **Processor, Used Memory:** En la figura 4.48, la línea verde muestra la memoria utilizada por el procesador, que se mantiene en valores estables alrededor de 19.8 MiB, con ligeras fluctuaciones. Esta estabilidad indica que el uso de memoria es constante y que no existen picos significativos que sugieran una sobrecarga en el procesamiento de tareas.

b) **Cisco IOS, ICMP Response Time:** La figura 4.48 presenta una línea amarilla que refleja el tiempo de respuesta ICMP (ping) en milisegundos, con fluctuaciones que incluyen picos y caídas a lo largo del tiempo. Estos cambios en la latencia están relacionados con variaciones en la conectividad de red, lo cual podría deberse a fluctuaciones en el tráfico o a problemas intermitentes en la red.

c) **Internal Chassis Fan Status:** En la figura 4.48, la línea naranja ilustra el comportamiento del ventilador dentro del switch, mostrando subidas y bajadas regulares que reflejan cambios en la velocidad de este componente. La variabilidad observada sugiere que el ventilador ajusta su velocidad en respuesta a cambios de temperatura o carga en el sistema, lo que es común para mantener la temperatura interna dentro de los límites adecuados.

Observaciones:

- La memoria utilizada se mantiene estable, lo que indica que el sistema no tiene problemas de memoria.
- La latencia ICMP muestra picos frecuentes, lo que podría señalar problemas de conectividad o congestión en la red.
- La actividad del ventilador sugiere que el sistema ajusta su enfriamiento de forma dinámica, posiblemente debido a cambios en la carga o temperatura del dispositivo.

Es importante señalar que, en un entorno virtual, los valores relacionados con componentes físicos como el **ventilador** o la **temperatura** no corresponden a hardware real, ya que dichos dispositivos no existen dentro de una máquina virtual. En este caso, los valores mostrados para el estado del ventilador o la temperatura son simulados por el entorno virtual.

La figura 4.49 muestra el monitoreo del servidor Windows.

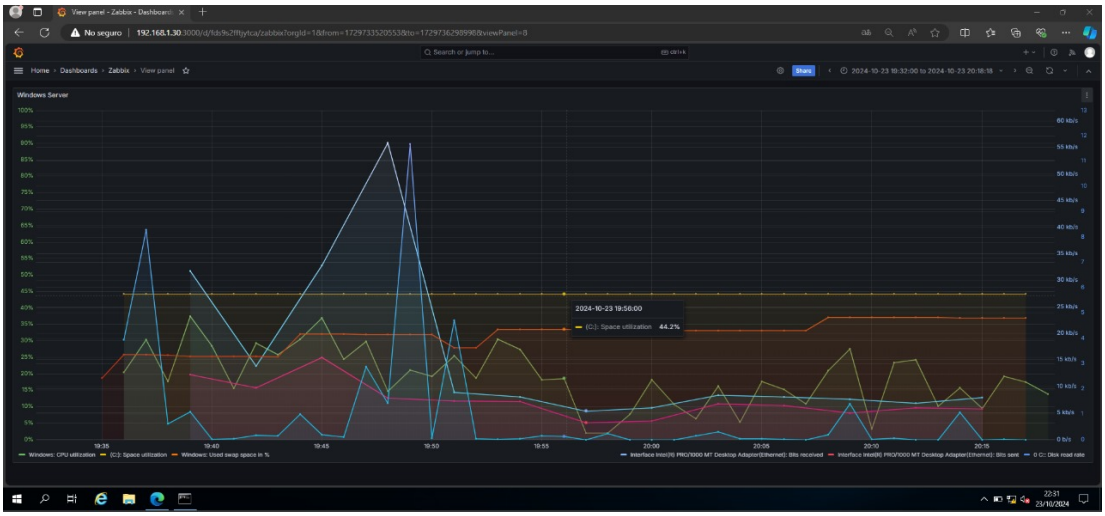


Figura 4.49: Métricas del servidor Windows.

a) **Windows, utilización de CPU:** En la figura 4.49, la línea verde muestra el uso del CPU del servidor en porcentaje. Se observan picos en la utilización de la CPU que alcanzan hasta el 70 % en ciertos momentos, aunque también hay intervalos durante los cuales el uso es bajo, lo que indica una variabilidad en la carga del procesador a lo largo del tiempo.

b) **C, utilización de espacio:** La figura 4.49 presenta una línea naranja que refleja la utilización del espacio en disco de la unidad C del servidor en porcentaje. Este indicador se mantiene estable alrededor del 44.2 %, lo que sugiere que no hay cambios significativos en el uso del disco durante el período monitoreado, lo que podría indicar un uso constante y eficiente del almacenamiento disponible.

c) **Windows, espacio de intercambio swap en %:** En la figura 4.49, la línea roja representa el uso del espacio de intercambio (swap) en porcentaje. La métrica muestra valores bajos y estables, lo que sugiere que el sistema no requiere mucho espacio de intercambio, probablemente debido a que la memoria RAM es suficiente para las tareas en curso y el sistema no ha tenido que recurrir a la memoria virtual de manera significativa.

d) **interfaz Intel(R) PRO/1000 MT Desktop Adapter (Ethernet), Bits recibi-**

dos y Bits enviados: En la figura 4.49, las líneas azul claro y amarilla muestran el tráfico de red entrante (Bits Received) y saliente (Bits Sent), respectivamente. Ambas métricas exhiben picos y variaciones, con algunos momentos de mayor actividad, aunque en general los valores son bajos, lo que sugiere un uso moderado de la red en el servidor durante el período observado.

e) **Tasa de lectura en disco:** En la figura 4.49, la línea celeste refleja la tasa de lectura en disco en kilobytes por segundo. Esta métrica se mantiene en niveles bajos, lo que indica que no se están llevando a cabo operaciones intensivas de lectura en el disco durante el período monitoreado, lo cual es un comportamiento esperado si las tareas realizadas no requieren acceso constante al almacenamiento.

Observaciones:

- El uso de CPU es variable, con algunos picos, lo que podría indicar procesos intensivos que se ejecutan de manera intermitente.
- La utilización del espacio en disco es estable, sin variaciones importantes.
- El uso del swap es mínimo, lo cual sugiere que el servidor tiene suficiente memoria RAM para manejar sus tareas actuales.
- El tráfico de red muestra actividad moderada, pero sin indicios de congestión.
- La tasa de lectura en disco es baja, indicando poca demanda de lectura en el almacenamiento.

La figura 4.50 muestra el estado del servidor Ubuntu.

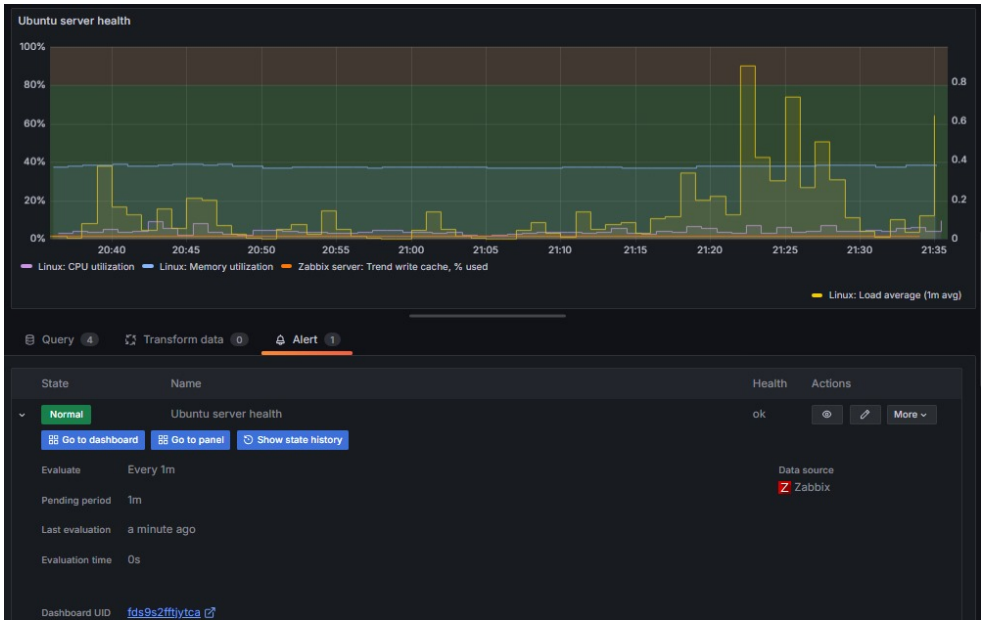


Figura 4.50: Métricas del servidor Ubuntu.

a) **Utilización de CPU:** Representada en color morado en la figura 4.50, esta línea muestra la carga de trabajo del CPU en diferentes momentos. Se observa una línea relativamente estable, indicando que la CPU mantiene un uso constante sin grandes variaciones.

b) **Utilización de Memoria:** La línea de color azul en la figura 4.50, indica el uso de memoria en el servidor Ubuntu. La gráfica muestra un comportamiento estable, sin picos significativos en el consumo de memoria.

c) **Caché de escritura del servidor Zabbix:** Representada en color naranja en la figura 4.50, muestra el porcentaje de uso de caché de escritura en el servidor Zabbix. Se observan algunos picos a lo largo del tiempo, lo que podría indicar momentos de actividad elevada en la escritura de datos.

d) **Promedio de Carga en 1 Minuto:** En la figura 4.50 la línea de color amarillo, muestra el promedio de carga del sistema en intervalos de un minuto. Los picos en esta línea representan momentos en los que el sistema experimentó una mayor carga, aunque en general la línea parece mantenerse dentro de niveles controlados.

En la sección inferior, se observa una alerta configurada para el estado de salud del servidor Ubuntu, donde el estado se encuentra en "Normal", lo que indica que las métricas monitoreadas están dentro de los parámetros esperados. Cabe mencionar que la frecuencia de muestreo es de un minuto.

La figura 4.51 muestra una representación de monitoreo en formato de **mapa de calor** para evaluar el estado del servidor Ubuntu.

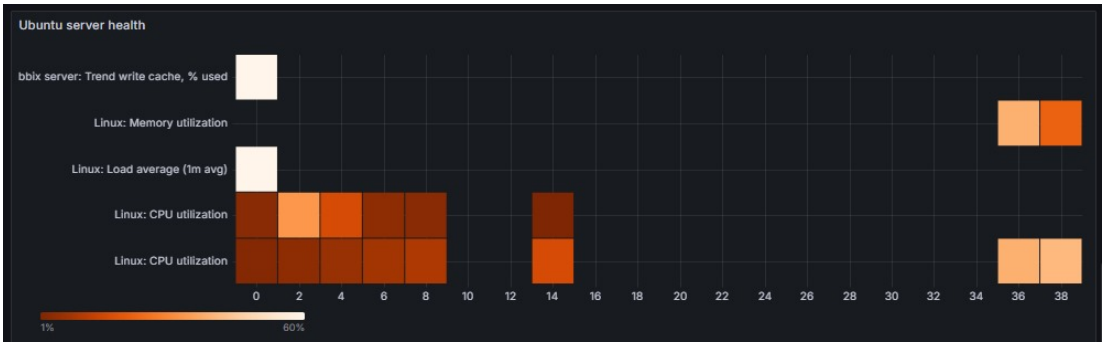


Figura 4.51: Monitoreo del servidor Ubuntu mediante un mapa de color.

- **Caché de escritura del servidor Zabbix (% utilizado):** Esta métrica, en la parte superior, muestra el porcentaje de caché de escritura en uso en el servidor Zabbix. Los bloques al final muestran un tono más claro, indicando un uso moderado.
- **Utilización de Memoria:** Indica el uso de memoria en el sistema Ubuntu. La actividad se encuentra en la última columna, con un uso moderado.

- **Promedio de Carga en 1 Minuto:** Representa la carga promedio en el sistema en intervalos de un minuto. Los valores muestran colores claros y oscuros, lo cual sugiere variaciones en la carga.
- **Utilización de CPU:** La utilización del CPU tiene dos registros. En los primeros bloques, hay un uso de CPU más intenso, representado en tonos oscuros, y otros momentos con menor actividad. Este patrón indica periodos de carga intensa seguidos de tiempos de menor actividad.

En este tipo de representación, los colores más oscuros indican un uso más alto de recursos, mientras que los más claros sugieren un menor uso. En resumen, el servidor Ubuntu parece haber experimentado momentos de carga variable, con períodos de mayor uso en la CPU y un uso moderado de la memoria y caché de escritura.

La figura 4.52 muestra una tabla de monitoreo del servidor Windows que presenta diversas métricas, utilizando una escala de colores para indicar el nivel de utilización.

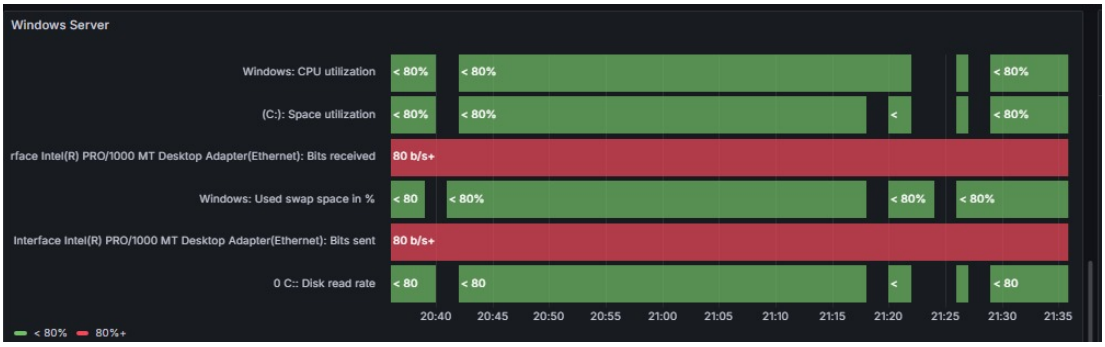


Figura 4.52: Tabla de monitoreo para el servidor Windows.

- **Utilización del CPU:** Representada en verde (<80 %), lo que indica que el uso de CPU está en niveles bajos y dentro de un rango seguro durante el periodo monitoreado.
- **Utilización del Espacio en Disco (C):** También representada en verde, indicando que el espacio en el disco C está por debajo del 80 % y no se observan problemas de almacenamiento.
- **interfaz de Red:** Las métricas de la interfaz para bits recibidos y enviados en la tarjeta de red muestran secciones en rojo (80 b/s), lo que sugiere un tráfico de red relativamente alto en algunos momentos, por encima del umbral del 80 %. Esto podría indicar una actividad de red considerable o un uso intensivo de la conexión.
- **Uso del Espacio de Swap:** La utilización del espacio de swap también se mantiene en verde (<80 %), indicando que el servidor no está recurriendo intensivamente a esta memoria adicional.

- **Velocidad de Lectura en Disco (C):** El acceso de lectura en disco se mantiene dentro de niveles seguros (<80 %), lo que indica que no hay operaciones intensivas de lectura en disco.

El servidor Windows presenta un uso eficiente de la CPU, memoria y el almacenamiento, sin sobrecargas significativas. Sin embargo, la actividad de red (tanto en bits recibidos como en enviados) muestra niveles elevados en ciertos momentos. Este análisis permite confirmar que el rendimiento general del servidor es estable.

La figura 4.53 muestra un tablero de monitoreo del servidor Windows, destacando métricas de rendimiento. Entre los valores monitoreados, se observa el uso de CPU (11.5 %), la utilización del espacio en disco (44.2 %), la tasa de transferencia de datos de red, tanto entrante como saliente, y el uso del espacio de intercambio (45.6 %). Estas métricas indican un servidor con buen rendimiento y un bajo uso de recursos en general.

Figura 4.53: Tablero de monitoreo para el servidor Windows.

El análisis de monitoreo realizado con Grafana ha demostrado ser una herramienta eficaz para supervisar el rendimiento y el estado de los servidores en la infraestructura de red. Este sistema permite una visibilidad completa y en tiempo real de las métricas clave de uso de CPU, memoria, almacenamiento y red, lo cual es fundamental para garantizar la estabilidad y la eficiencia de los servicios.

La infraestructura de monitoreo implementada permite anticiparse a problemas potenciales mediante alertas configurables y una representación visual clara del uso de recursos, lo que facilita la detección temprana de anomalías. Además, el uso de mapas de calor y gráficas optimiza el análisis, ya que permite observar patrones de uso y carga en diferentes momentos.

Los resultados muestran que los recursos disponibles son suficientes para las cargas de trabajo actuales y que el monitoreo continuo permitirá ajustar los recursos de manera oportuna en caso de cambios en la demanda. Esto no solo optimiza el uso de la infraestructura de TI, sino que también mejora la capacidad de respuesta ante posibles incidencias.

Por lo mencionado anteriormente, Grafana ha demostrado ser una herramienta eficaz para la gestión de infraestructuras de red. Su capacidad de monitoreo en tiempo real, junto con

las alertas y la facilidad de interpretación visual de los datos, permite mantener un control efectivo y tomar decisiones informadas que aseguren un rendimiento óptimo de los servidores. Estas características facilitan la gestión proactiva de la infraestructura, mejorando la capacidad de respuesta ante posibles incidencias y optimizando el uso de los recursos disponibles.

4.10. Paneles de monitoreo Zabbix en la Raspberry Pi

En esta sección se analizan los paneles de monitoreo en Zabbix, que se encuentra instalado en la Raspberry Pi. Para llevar a cabo este proceso, se configuraron diversos parámetros del sistema operativo y se utilizaron plantillas prediseñadas de Zabbix para monitorear el rendimiento del hardware de la Raspberry Pi, como el uso de CPU, memoria, espacio en disco y tráfico de red.

La figura 4.54 muestra la ventana de creación de una condición en Zabbix dentro de la configuración de acciones. En este caso, se define una condición basada en la severidad del disparador (Trigger severity), que permite especificar el nivel de gravedad de los eventos que activarán la acción. Los operadores disponibles incluyen opciones como igual, diferente, mayor o igual que, y menor o igual que. Además, se pueden seleccionar diferentes niveles de severidad, como información, advertencia, promedio, alta y desastre. Esto permite ajustar las alertas de acuerdo con la criticidad de los eventos.

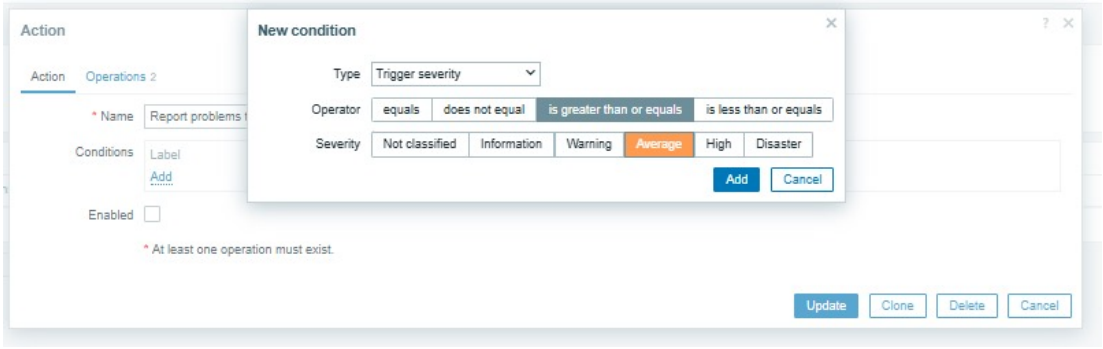


Figura 4.54: Creación de condición del disparador en Zabbix.

La figura 4.55 muestra el **panel de control global de Zabbix**, donde se monitorean diversos aspectos del sistema en tiempo real. En la sección superior, se visualiza la lista de los hosts principales según el uso de CPU, incluyendo el servidor Zabbix y la Raspberry Pi. A la derecha, se presentan las estadísticas del sistema, como la versión del servidor,

el número de hosts y disparadores habilitados, y la tasa de datos procesados. También se exhiben métricas de disponibilidad de hosts, problemas clasificados por severidad y los problemas detectados, tales como la desincronización de la hora del sistema o el uso elevado de CPU.

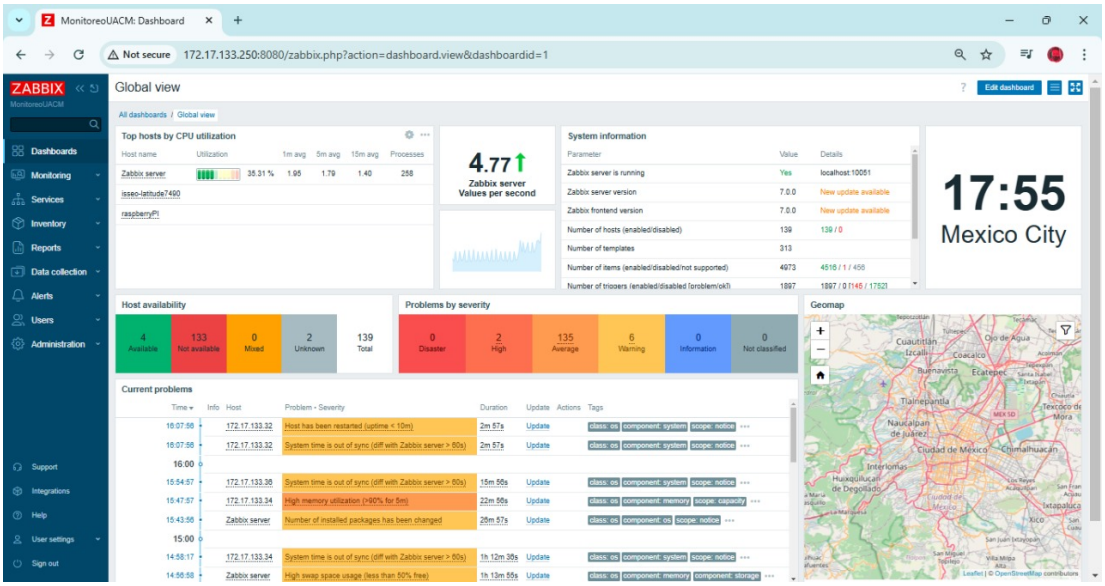


Figura 4.55: Panel de control de Zabbix.

La figura 4.56 muestra en la sección central una lista de los problemas detectados, donde se observan columnas como la hora del evento (time), severidad (severity), hora de recuperación (recovery time), estado (status), dispositivo afectado (host), descripción del problema (problem) y duración del problema (duration).

Al lado izquierdo del menú se encuentran las opciones principales de Zabbix, como tableros (dashboards), monitoreo (monitoring), inventario (inventory), entre otras. En la parte superior, hay filtros para personalizar la búsqueda de problemas según criterios específicos, como severidad, grupos de host y otros atributos del host.

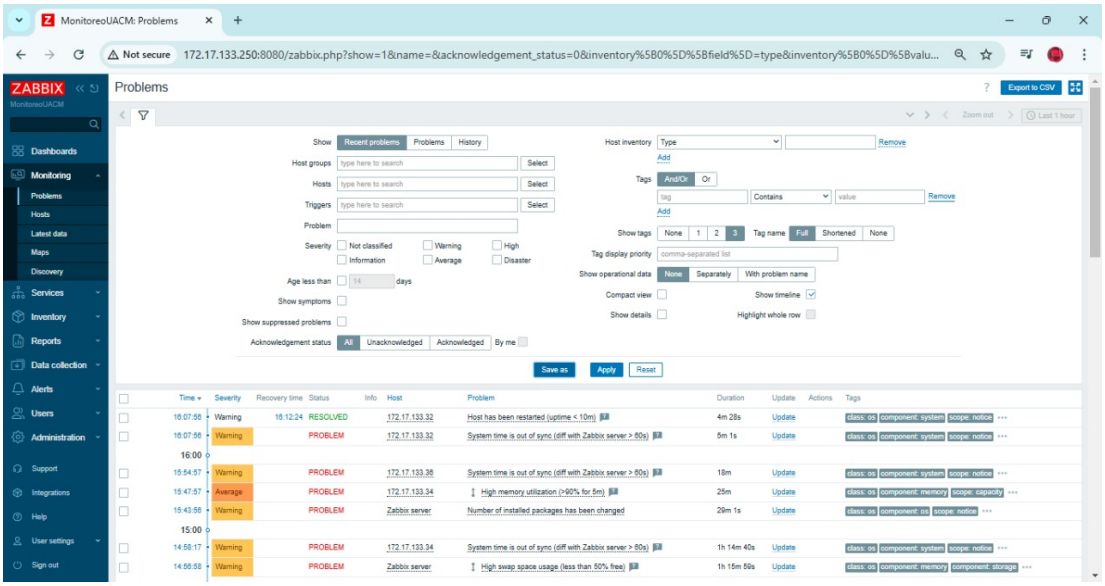


Figura 4.56: Monitoreo de problemas en Zabbix.

En la figura anterior, se muestran varios dispositivos (hosts) con diferentes problemas. A continuación, el análisis de los problemas específicos para cada dispositivo que se visualiza:

La siguiente tabla (4.4) muestra los problemas detectados en los diferentes hosts monitoreados, incluyendo la severidad y la duración de cada problema.

Tabla 4.4: Problemas detectados en los hosts

Host	**Problema**	**Severidad**	**Duración**
172.17.133.32	El host se ha reiniciado y la hora del sistema está desincronizada con el servidor Zabbix por más de 90 segundos.	Advertencia	4m 28s (reinicio), 5m 1s (desincronización)
172.17.133.34	La hora del sistema está desincronizada con el servidor Zabbix por más de 90 segundos.	Advertencia	18m
Zabbix server	Alta utilización de memoria, superior al 90 % durante 5 minutos.	Media	25m
Zabbix server	El número de paquetes instalados ha cambiado.	Advertencia	29m 1s
172.17.133.34	La hora del sistema está desincronizada con el servidor Zabbix por más de 90 segundos.	Advertencia	1h 14m 40s
172.17.133.34	Alto uso de memoria de intercambio, con menos del 50 % libre.	Advertencia	1h 15m 59s

La figura 4.57 muestra el estado de descubrimiento de varios dispositivos (hosts), por ejem-

plo, la red LACECI y LACECI Wireless. Cada fila representa un dispositivo identificado por su dirección IP, y la información se organiza en columnas que indican la última vez que el dispositivo fue detectado, junto con un indicador de tiempo en color rojo que señala la duración del último evento o problema en cada dispositivo.

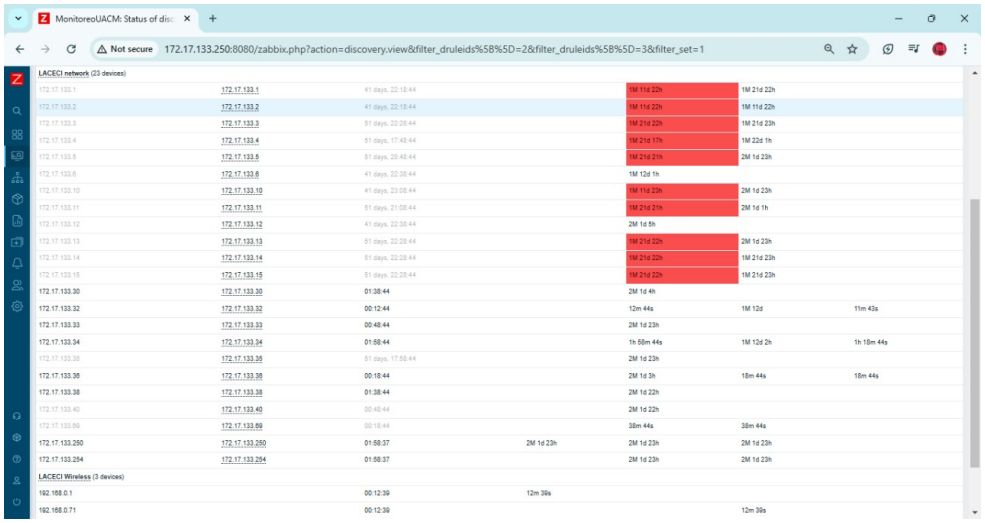


Figura 4.57: Estado de descubrimiento de dispositivos en Zabbix.

Se observa que los dispositivos 172.17.133.32, 172.17.133.34 y 172.17.133.36 son los únicos que presentan información en la última columna. Esto indica que estos dispositivos tienen instalado el agente de monitoreo de Zabbix, lo que permite una recolección de datos más detallada y en tiempo real sobre su estado y rendimiento.

También se presentan dos reglas de descubrimiento llamadas **LACECI Wireless** y **LACECI Network**, cada una configurada con diferentes criterios para la detección de dispositivos.

- **LACECI Wireless:** Esta regla está configurada para detectar los dispositivos del segmento de red 192.168.0.1 al 192.168.0.254 y lo realiza por medio del protocolo ICMP al verificar si tiene respuesta por ping, si existe una conexión por SSH abierta o sesión telnet activa.
- **LACECI Network:** Esta regla está configurada para detectar todos los dispositivos dentro de la red en el segmento 172.17.133.0. En este caso, identifica un total de 23 dispositivos dentro de ese rango de direcciones IP. Esta regla verifica que dispositvos tienen instalado el agente de Zabbix. En este caso, identifica 3 dispositivos: 172.17.133.32, 172.17.133.34 y 172.17.133.36. Estos son los dispositivos que muestran información adicional en la última columna gracias a la instalación del agente de monitoreo.

Estas reglas permiten distinguir entre los dispositivos que tienen instalado el agente de Zabbix y aquellos que no lo tienen.

La figura 4.58 muestra un panel de monitoreo del rendimiento del sistema del dipositivo Zabbix server, dividido en secciones clave: carga del sistema, uso de CPU, uso de memoria y uso de intercambio (swap). En la sección **Linux: carga del sistema (System load)**, se observa que la carga promedio es baja, con valores promedio de carga en 1, 5 y 15 minutos por debajo de 1, lo que indica que el sistema no está sobrecargado y que los 4 núcleos disponibles no están saturados. En la sección **Linux: uso de CPU (CPU usage)**, se evidencia un bajo uso de CPU, con un promedio de tiempo de uso de la CPU (CPU user time) de aproximadamente 1.48 % y un tiempo del sistema de CPU (CPU system time) de 1.18 %, mientras que otras métricas, como el tiempo de espera y el tiempo de interrupción, son insignificantes o nulas. **Linux: uso de memoria (Memory usage)** indica que el sistema tiene 3.71 GB de memoria total, con 2.69 GB disponibles en promedio, sugiriendo un uso de memoria estable y controlado. Finalmente, en **Linux: uso de intercambio (Swap usage)**, se muestra un total de 2 GB de espacio swap, de los cuales están libres en su totalidad, lo que indica que no se está utilizando el área de intercambio. En general, el sistema presenta un rendimiento estable y un bajo uso de recursos, sin señales de sobrecarga o cuellos de botella.

Figura 4.58: Monitoreo de rendimiento en Zabbix.

La figura 4.59 muestra información sobre el uso del espacio en el disco de los host monitoreados, específicamente el sistema de archivos ext4. En la gráfica de pastel Cuadro de **utilización del espacio (Space utilization chart)**, se observa que el espacio total es de 915.32 GB, de los cuales 27 GB (2.95 %) están en uso y 841.76 GB (91.96 %) están disponibles, lo que indica que el almacenamiento aún cuenta con espacio disponible.En la sección

inferior, **la gráfica de uso del espacio (Space usage graph)** muestra el uso de espacio en porcentaje durante la última hora, según se presenta en la figura, con un valor constante de 3.1077 %, muy por debajo de los umbrales de advertencia (80 %) y crítico (90 %). Esto señala que el sistema tiene un amplio espacio de almacenamiento disponible y no enfrenta riesgos de falta de espacio.

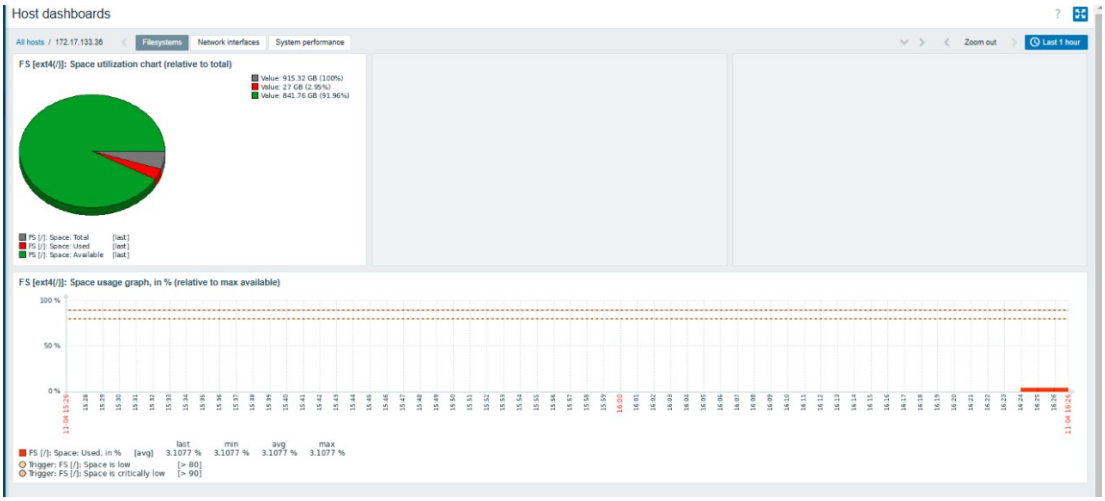


Figura 4.59: Estado del sistema de archivos en Zabbix.

4.11. Paneles de monitoreo Grafana en la Raspberry Pi

En esta sección se presentan los paneles de monitoreo configurados en Grafana para la Raspberry Pi. Estos paneles permiten visualizar métricas del rendimiento y estado de diversos dispositivos, proporcionando una interfaz intuitiva para el análisis en tiempo real. A través de estos dashboards, es posible monitorear el uso de CPU, memoria, carga del sistema, tráfico de red y otros parámetros clave. Además, Grafana facilita la personalización de gráficos y alertas, lo que permite detectar anomalías y tomar decisiones proactivas para garantizar un funcionamiento óptimo de los dispositivos.

La figura 4.60 muestra el panel de monitoreo de Grafana, que visualiza métricas de rendimiento y estado de algunos dispositivos.

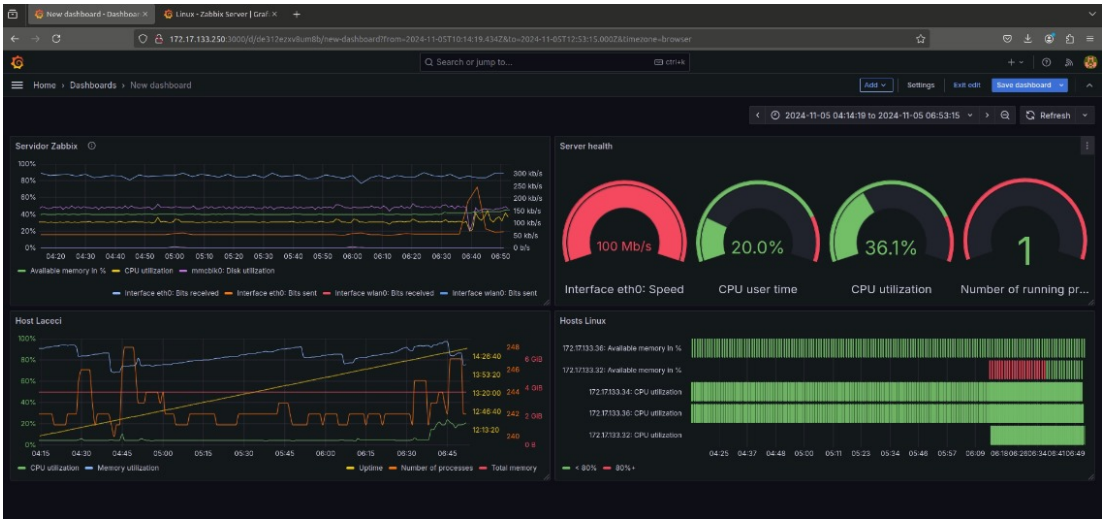


Figura 4.60: Panel de monitoreo en Grafana.

a) Servidor Zabbix:

- Memoria disponible
- Uso de CPU
- Uso del disco mmcblk0
- Actividad en las interfaces de red (eth0 y wlan0), mostrando bits recibidos y enviados.

b) Salud del servidor:

- Velocidad de la interfaz eth0: muestra 100 Mb/s.
- Uso de CPU: al 20 %.
- Utilización de CPU total: al 36.1 %.
- Número de procesos en ejecución: muestra 1, posiblemente destacando procesos críticos.

c) Host Laceci:

- Utilización de CPU y memoria.
- Tiempo de actividad.
- Número de procesos y memoria total.
- Esto da una visión detallada del rendimiento y la estabilidad del host en específico, variaciones de uso en CPU y memoria.

d) Hosts Linux:

- Lista de diferentes hosts en la red (con IPs como 172.17.133.32, 172.17.133.33, etc.)

- Memoria disponible.
- Utilización de CPU.

La figura 4.61 muestra una gráfica en la que se observan datos del **Servidor Zabbix** en Grafana, monitoreando diferentes indicadores de rendimiento.

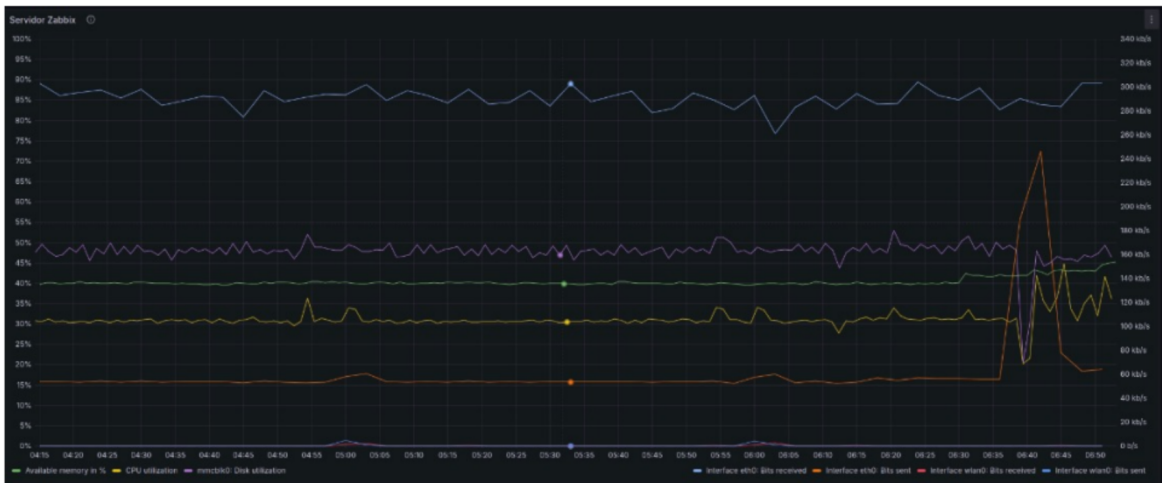


Figura 4.61: Métricas del servidor Zabbix en Grafana.

a) Memoria Disponible:

- Representada en verde en el gráfico, se mantiene entre 45 % y 50 %, lo cual indica una estabilidad en el uso de memoria del servidor.
- Utilización de la CPU:
- La línea amarilla muestra que la utilización de la CPU oscila alrededor del 30 % - 35 %, con algunos picos, lo que sugiere un uso moderado de la CPU.

b) Uso del Disco (mmcblk0):

- Representada en púrpura, permanece estable en torno al 45 % - 50 %, indicando un uso de disco relativamente constante sin picos importantes.

c) Actividad de la Red (eth0 y wlan0):

- La línea azul en la figura 4.61 (bits recibidos en eth0) muestra un tráfico estable, manteniéndose alrededor de 300 kb/s.
- La línea naranja en la figura 4.61 (bits enviados en wlan0) muestra un aumento drástico en un momento específico, alcanzando los 200 kb/s, seguido de una disminución rápida.
- Las otras métricas de tráfico en eth0 y wlan0 no presentan cambios relevantes.

Con base en la gráfica, se identifica que el servidor tiene un rendimiento estable en cuanto a memoria y disco, con un uso controlado del CPU. Sin embargo, el pico en el tráfico de

la interfaz wlan0 es un punto de interés que podría indicar una carga elevada en la red durante un período específico.

La obtención de datos para el monitoreo se realizó de manera fluida. Sin embargo, el proceso de configuración de los paneles presentó cierta lentitud al momento de su creación, lo cual se atribuye a las limitaciones en los recursos de la Raspberry Pi utilizada. A pesar de esta demora, no se observaron afectaciones en el uso de la CPU o la memoria del sistema, ya que los valores monitoreados no mostraron picos significativos en estos recursos.

La figura 4.62 muestra una gráfica de igual forma que la **Servidor Zabbix** pero la diferencia entre esta gráfica y la anterior radica en el tiempo de monitoreo. Es decir, mientras que la gráfica anterior mostraba un monitoreo detallado a corto plazo (horas) con una estabilidad general en el rendimiento y un pico específico en el tráfico de la interfaz wlan0, esta gráfica abarca un período más prolongado (días), lo que permite observar cambios en su comportamiento a largo plazo. En esta segunda gráfica, destacan dos eventos específicos: un incremento en la memoria disponible el 31 de octubre, lo cual indica una posible optimización del sistema, y un aumento en el tráfico de red el 4 de noviembre en la interfaz eth0, que alcanzó casi 800 kb/s.

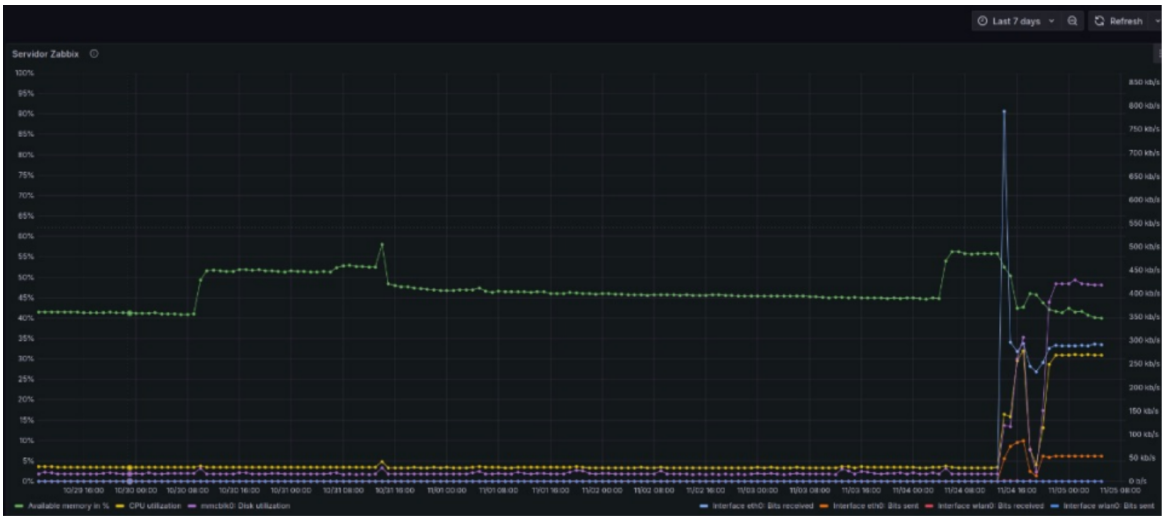


Figura 4.62: Métricas del servidor Zabbix a lo largo de varios días en Grafana.

a) **Memoria Disponible:**

- Representada en verde en la figura 4.62, la memoria disponible inicialmente está alrededor del 40 % y sube a aproximadamente 55 % el 31 de octubre, manteniéndose en ese nivel presentando cambios menores. Esto sugiere una liberación de memoria o una optimización en el sistema que ocurrió en esa fecha.

b) **Utilización de CPU:**

- La línea amarilla en la figura 4.62 muestra que el uso de CPU permanece bajo (alrededor del 5 %) durante todo el periodo. Esto indica que el servidor no experimenta una carga significativa en la CPU.

c) **Uso del Disco mmcblk0:**

- La línea morada en la figura 4.62 refleja una estabilidad en el uso del disco, con un nivel muy bajo (aproximadamente 5 %). Esto indica que el servidor no tiene una actividad alta de escritura o lectura en este disco.

d) **Actividad de la Red en eth0 y wlan0:**

- En la segunda mitad del gráfico (4 de noviembre), se observa un pico notable en la interfaz eth0 identificada con una línea azul en la figura 4.62 en los bits recibidos, alcanzando casi 800 kb/s, seguido de una rápida disminución.
- Posteriormente, hay cambios en el tráfico tanto en eth0 como en wlan0, pero en niveles más bajos (200 - 400 kb/s). Este comportamiento indica un aumento temporal en la demanda de red, debido a algún tipo de proceso o transferencia de datos que ocurrió en esa fecha.

La liberación de memoria el 31 de octubre y el pico de tráfico en eth0 el 4 de noviembre son los eventos más destacados del período observado. La estabilidad en el uso de la CPU y del disco sugiere que el servidor está funcionando dentro de los límites normales, sin sobrecargas significativas.

Tabla 4.5: Comparación entre Métricas en horas y en días

Aspecto	**Métricas en horas**	**Métricas en días**
Intervalo de Monitoreo	Horas	Días
Duración Observada	Corto plazo	Largo plazo
Memoria Disponible	Alrededor del 85 %, estable	Incremento a 55 % el 31 de octubre
CPU Utilización	Baja y estable (aprox. 5 %)	Baja y estable (aprox. 5 %)
Disco Utilización	Estable y baja (aprox. 5 %)	Estable y baja (aprox. 5 %)
Actividad en Red	Pico en `wlan0` (160 kb/s)	Pico en `eth0` (800 kb/s) el 4 de noviembre
Tipo de Análisis	Variaciones inmediatas	Tendencias a largo plazo

En la figura 4.63 se observa el monitoreo del **Host Laceci** durante un periodo de 24 horas.

Figura 4.63: Monitoreo del Hosts Laceci en Grafana.

a) **Utilización de Memoria:**

- Representada en la línea azul en la figura 4.63, la utilización de memoria muestra un comportamiento variable. Alrededor de las 15:00, la memoria se encontraba en niveles altos (aproximadamente 95 %). Luego, experimenta una serie de variaciones, especialmente después de las 16:00, donde desciende drásticamente. Durante la noche, se puede ver un patrón de picos y caídas regulares.
- La memoria total del sistema se mantiene constante en 3.71 GB, lo cual es mostrado en la línea roja en la figura 4.63 horizontal al nivel superior de la gráfica de valores en el eje derecho.

b) **Tiempo de Actividad:**

- Indicada en naranja en la figura 4.63, la línea del tiempo de actividad parece mostrar un patrón consistente hasta las 15:00, donde su valor desciende considerablemente. Luego se estabiliza y permanece constante hasta las 0 horas, cuando aumenta de nuevo.
- Este descenso y aumento pueden indicar reinicios o cambios en el sistema.

c) **Número de Procesos:**

- Representado en verde en la figura 4.63, el número de procesos tiene varios picos pequeños, lo cual indica momentos de actividad elevada del sistema. Estos picos son más pronunciados después de las 16:00 y aumentan especialmente al final del periodo monitoreado, alcanzando niveles inusualmente altos alrededor de las 07:00.

La gráfica indica una posible inestabilidad en el sistema o en la ejecución de tareas programadas que afectan la utilización de la memoria y el número de procesos. El patrón de picos

regulares en la memoria y el tiempo de actividad podría estar relacionado con ciclos de trabajo específicos o con tareas automáticas que se ejecutan periódicamente. El aumento, al final, en el número de procesos puede ser indicativo de una carga de trabajo en incremento o de un proceso específico que consume recursos.

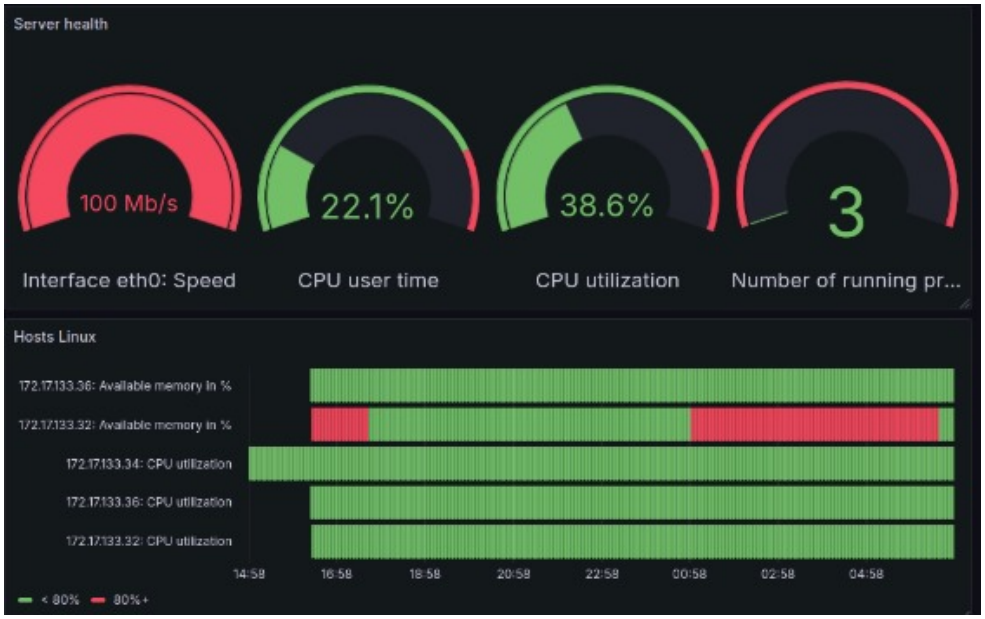


Figura 4.64: Panel de monitoreo del servidor en Grafana.

La figura 4.64 muestra un panel de monitoreo del servidor, con algunos indicadores como la velocidad de la interfaz eth0, el tiempo de uso del CPU, la utilización del CPU y el número de procesos en ejecución. La **velocidad de la interfaz** en eth0 está en 100 Mb/s y aparece en rojo, lo que podría indicar una sobrecarga o que se ha excedido el umbral configurado. **El tiempo de uso del CPU** es del 22.1 %, mientras que **la utilización del CPU** está en un 38.6 %, ambos en niveles aceptables. En cuanto a los **hosts Linux**, se visualiza el uso de memoria y CPU de varios hosts en la red. Las barras verdes indican niveles normales de recursos, mientras que las secciones rojas muestran que el umbral del 80 % ha sido superado en ciertos momentos, lo que podría señalar períodos de alta carga en esos hosts específicos.

4.12. Isolation Forest con Zabbix

En este proyecto se ha implementado Isolation Forest en conjunto con Zabbix para detectar anomalías en el uso de CPU y memoria de un servidor. El análisis de los resultados obtenidos permite evaluar la efectividad del algoritmo en la identificación de comportamientos atípicos y su impacto en la supervisión de la infraestructura.

El modelo de Isolation Forest fue configurado para procesar los datos obtenidos a través de la API de Zabbix, lo que permite detectar eventos fuera de lo común sin la necesidad de un conjunto de entrenamiento previamente etiquetado. Las anomalías se visualizaron en gráficos, donde los valores inusuales se marcaron en rojo, facilitando así la interpretación de los datos.

Las gráficas obtenidas resaltan visualmente las anomalías detectadas y muestran tendencias en el uso de recursos. Finalmente, el sistema envía un correo electrónico con las gráficas adjuntas y un análisis textual que incluye el número de anomalías encontradas, tanto para la CPU como para la memoria.

4.12.1. Análisis de las gráficas obtenidas

Análisis del uso de Memoria

En la figura 4.65, se muestra el análisis del uso de memoria de la máquina virtual 1. Se puede observar una tendencia general estable en el uso de memoria entre los días 11 y 15 de octubre de 2024, con un valor promedio alrededor de 39-40 unidades. No obstante, en este intervalo de tiempo se identifican algunas anomalías aisladas (marcadas con cruces rojas), las cuales podrían ser indicativas de picos en el consumo de memoria. Posteriormente, se aprecia una caída abrupta del uso de memoria el día 15, seguida por un incremento significativo y repentino el día 16. Esta variación drástica hacia el final del período monitoreado, que incluye múltiples anomalías, sugiere un comportamiento inesperado o un posible problema en el sistema, como un error de software o un proceso que consume memoria de forma descontrolada.

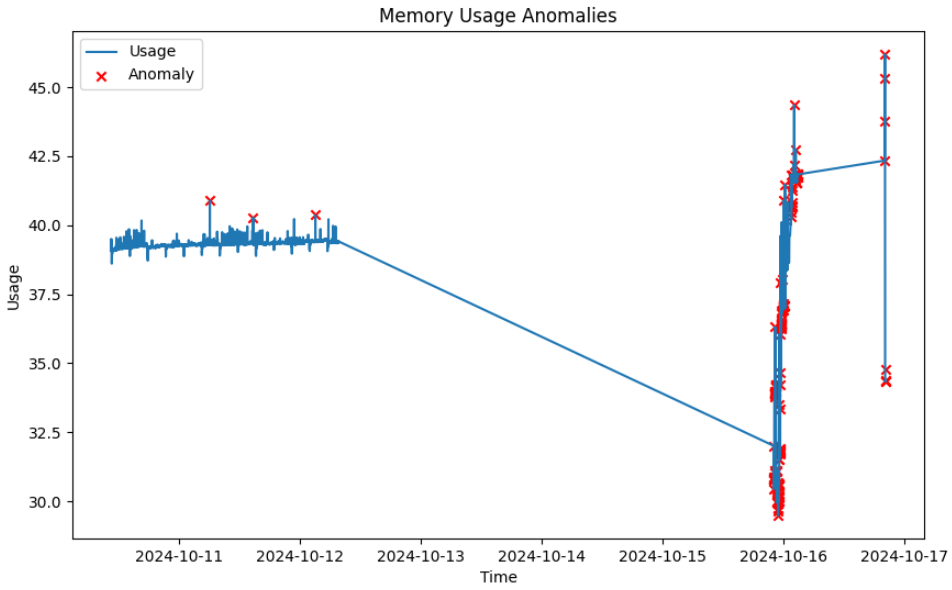


Figura 4.65: Anomalías en uso de memoria.

En la gráfica titulada *Anomalías en uso de memoria*, se observa el comportamiento del uso de la memoria a lo largo del tiempo. La línea azul de la figura 4.65 muestra el uso normal de la memoria, mientras que las cruces rojas indican las anomalías detectadas por el modelo de *Isolation Forest*, el cual utiliza un enfoque de inteligencia artificial para identificar comportamientos atípicos en los datos.

Análisis del uso de CPU

En la Figura 4.66, se muestra el comportamiento del uso de la CPU a lo largo del tiempo. La línea azul representa el uso de la CPU, mientras que las cruces rojas marcan las anomalías identificadas mediante el modelo de *Isolation Forest*, que utiliza inteligencia artificial para detectar eventos fuera de lo común en los datos.

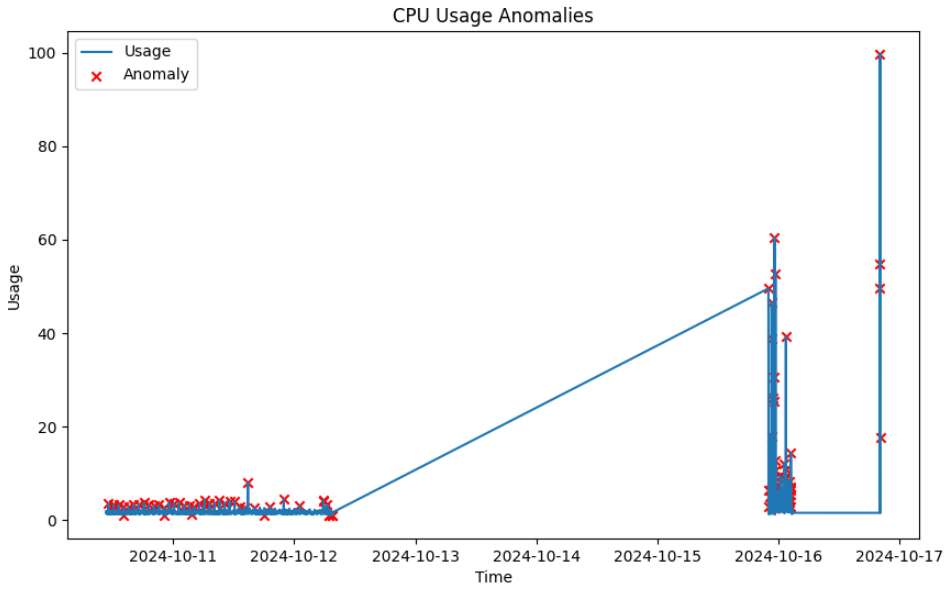


Figura 4.66: Anomalías detectadas en el uso de CPU.

Características observadas:

- **Estabilidad inicial:** Entre los días 2024-10-11 y 2024-10-13, el uso de la memoria es estable, alrededor de los 40 MB, con solo una anomalía detectada, lo que sugiere un comportamiento mayormente normal durante este período.
- **Caída abrupta y recuperación:** A partir del 2024-10-13, se observa una caída repentina del uso de memoria de aproximadamente 40 MB a 30 MB. Esto podría estar relacionado con un reinicio del sistema o la liberación masiva de recursos. Sin embargo, a partir del 2024-10-15, se registra un incremento abrupto del uso de memoria, acompañado por múltiples anomalías, lo que sugiere un comportamiento anómalo en la gestión de la memoria.
- **Aumento súbito:** El incremento repentino en el uso de la memoria, acompañado por una gran cantidad de anomalías, puede deberse a un problema de fuga de memoria (*memory leak*) o a una sobrecarga en los procesos que consumen memoria.

Posibles causas: La caída brusca y el rápido aumento podrían deberse a eventos internos del sistema, como reinicios o sobrecargas de procesos. Las anomalías detectadas por el modelo de *Isolation Forest* resaltan estas situaciones como comportamientos atípicos que deben investigarse para determinar su causa y mitigar cualquier problema subyacente.

Características observadas:

- **Período de baja actividad:** Entre los días 2024-10-11 y 2024-10-12, el uso del CPU es bajo, con algunos picos ocasionales que se marcan como anomalías. Esto sugiere que, aunque el sistema no está sobrecargado, ciertos eventos puntuales son identificados como fuera de lo común.
- **Incremento súbito:** A partir del 2024-10-12, se observa un aumento considerable en el uso del CPU, superando el 60 % de capacidad. Durante este periodo, se detectan varias anomalías, lo que podría ser indicativo de una sobrecarga o de la ejecución de procesos intensivos.
- **Cambios y picos:** A lo largo de los días posteriores, el uso del CPU muestra variaciones importantes y picos que son sistemáticamente marcados como anomalías, lo cual sugiere que el sistema está experimentando un comportamiento irregular, posiblemente no sostenible sin una intervención.

Posibles causas: El incremento repentino y las variaciones podrían estar relacionados con la ejecución de tareas intensivas, como procesos de cálculo complejos, o con la sobrecarga del sistema debido a múltiples aplicaciones en funcionamiento. Las anomalías detectadas podrían ser resultado de un uso ineficiente de los recursos o de procesos mal optimizados.

Uso de inteligencia artificial: El modelo de *Isolation Forest* funciona como un filtro de inteligencia artificial que identifica patrones anómalos en el uso de la CPU. Para mejorar la precisión y la capacidad predictiva de este análisis, podrían integrarse técnicas avanzadas de IA, como el aprendizaje profundo (*deep learning*) o modelos basados en series temporales como *Long Short-Term Memory* (LSTM). Estas técnicas permitirían no solo detectar anomalías, sino también predecir eventos futuros antes de que se conviertan en problemas críticos.

Automatización y Envío de Reportes

El sistema envía reportes con análisis y gráficas por correo, facilitando la supervisión remota. Esta funcionalidad podría mejorarse mediante la integración con sistemas de alerta más avanzados que permitan una respuesta automática ante la detección de ciertas anomalías críticas.

4.13. Implicaciones del uso de IA y futuras mejoras

El algoritmo *Isolation Forest* utilizado para la detección de anomalías es una técnica de inteligencia artificial orientada a la identificación de valores atípicos en grandes volúmenes de datos. Este método permite identificar comportamientos inesperados y optimizar los recursos en sistemas complejos, como los servidores monitorizados por Zabbix, donde pueden ocurrir cambios en el rendimiento debido a múltiples factores. Sin embargo, este método puede ser optimizado y complementado con otras técnicas de IA, como las redes neuronales recurrentes (RNN) o modelos de predicción basados en series temporales, los cuales podrían proporcionar un análisis más profundo y predictivo del comportamiento del sistema.

Una de las limitaciones actuales del sistema es que se enfoca en la detección de anomalías puntuales y no considera patrones temporales complejos. Para mejorar esta solución, sería recomendable incorporar modelos que puedan aprender tendencias a largo plazo en los datos de uso de recursos. Además, el ajuste dinámico del umbral de anomalías basado en el contexto operativo del sistema podría reducir la cantidad de falsos positivos y mejorar la precisión del mismo. Esto permitiría un sistema de monitoreo proactivo que no solo detecte anomalías, sino que también prediga problemas antes de que ocurran.

4.14. Evaluación de la Implementación Virtual y Física de Zabbix

Para evaluar el desempeño y las limitaciones del sistema de monitoreo, se realizaron dos implementaciones: una virtual, utilizando máquinas virtuales y dispositivos simulados, y otra física, basada en una Raspberry Pi conectada a equipos reales en laboratorios. La siguiente tabla presenta una comparación entre ambas aproximaciones, destacando sus características, ventajas y restricciones.

En la evaluación del sistema de monitoreo, se implementaron dos enfoques: una solución virtual utilizando máquinas virtuales y simulaciones de red, y una solución física basada en hardware real con Raspberry Pi y equipos de laboratorio. A continuación, en la Tabla 4.6, se presenta un resumen comparativo de ambas implementaciones, resaltando sus diferencias en infraestructura, rendimiento, escalabilidad y limitaciones.

Los resultados muestran que la implementación virtual permite probar configuraciones en un entorno controlado sin necesidad de hardware adicional; sin embargo, está limitada por los recursos de la máquina anfitriona, lo que impide simular un tráfico de red robusto y una mayor cantidad de dispositivos. En cambio, la implementación física con Raspberry Pi ofrece un monitoreo más realista y fiable, aunque presenta restricciones en la cantidad de dispositivos disponibles en el laboratorio. Para redes pequeñas, la Raspberry Pi resulta una

alternativa económica y funcional; no obstante, en redes más grandes, sería recomendable considerar hardware con mayor capacidad de procesamiento y almacenamiento.

Tabla 4.6: Comparación entre la implementación virtual y física

Criterio	**Implementación Virtual**	**Implementación Física**
Infraestructura	VirtualBox y GNS3 para simulación	Raspberry Pi y equipos en LACECI y LAMAT
Sistemas operativos	Ubuntu Server, Windows	Ubuntu Server
Software de monitoreo	Zabbix y Grafana en máquinas virtuales	Zabbix y Grafana en Raspberry Pi
Monitoreo con SNMP	Configurado en dispositivos simulados	Configurado en equipos físicos
Descubrimiento de hosts	Red virtual simulada	Red física del laboratorio
Agente Zabbix	Instalado en máquinas virtuales	Instalado en estaciones físicas
Limitaciones	- Recursos limitados de la máquina anfitriona - Tráfico de red insuficiente para pruebas reales	- Solo 25 dispositivos disponibles - Raspberry Pi puede requerir mejoras en hardware
Rendimiento	Depende de la máquina anfitriona	Depende del hardware de Raspberry Pi y equipos
Escalabilidad	Limitada por la máquina anfitriona	Puede ampliarse con más dispositivos físicos
Realismo	Limitado por la simulación	Representación fiel de un entorno real
Interacción con hardware	Simulación de dispositivos	Conexión con equipos reales
Costo	Depende del equipo anfitrión	Alternativa económica para redes pequeñas

4.15. Recomendaciones y trabajo a futuro

La implementación de las Raspberry Pi en el centro de datos que alberga las páginas web y servidores de la universidad representa una oportunidad para mejorar la eficiencia y escalabilidad del sistema de monitoreo. A continuación, se presentan recomendaciones

clave para optimizar el funcionamiento del sistema y asegurar su adaptabilidad a futuros requerimientos.

Recomendaciones:

a) **Optimización de recolectores de datos:**

- Ajustar la frecuencia de recolección según la actividad de las métricas para mejorar la eficiencia.
- Probar métodos de recolección (pull/push) para identificar el más adecuado según la carga de los dispositivos.

b) **Gestión del almacenamiento:**

- Optimizar el uso de caché para evitar consumo excesivo de almacenamiento.
- Implementar políticas de retención de datos en Zabbix y Grafana para conservar solo la información relevante.

c) **Automatización de alertas:**

- Mantener actualizadas las alertas en Telegram y correo electrónico, basadas en tendencias y patrones detectados.
- Asegurar que los responsables estén atentos a las notificaciones para una respuesta rápida.

d) **Análisis de patrones:**

- Usar Grafana para identificar tendencias y ciclos de carga que requieran intervención.
- Evaluar herramientas adicionales para predecir necesidades futuras de recursos.

e) **Documentación y mejora continua:**

- Documentar configuraciones y procedimientos en Zabbix y Grafana para facilitar futuras actualizaciones.
- Realizar revisiones periódicas del rendimiento y ajustar configuraciones según el crecimiento del sistema.

Conclusiones

El monitoreo y análisis con las herramientas configuradas ha demostrado ser efectivo en redes LAN de pequeña escala, validando su aplicabilidad en infraestructuras similares. Estas herramientas facilitan la supervisión de la red, optimizando su gestión y permitiendo la detección temprana de fallos y toma de decisiones basadas en métricas precisas.

La simulación en GNS3 y VirtualBox permitió evaluar el monitoreo de dispositivos virtuales en un entorno controlado. Aunque hubo limitaciones, como la imposibilidad de instalar agentes en GNS3, los resultados fueron satisfactorios y contribuyeron a entender el comportamiento de la red. La inclusión de sistemas operativos como Windows y Ubuntu permitió identificar problemas de configuración y rendimiento, similares a los encontrados en los laboratorios LACECI y LAMAT de la UACM.

El algoritmo Isolation Forest, integrado en Zabbix, permitió detectar anomalías en el uso de recursos del servidor, anticipando problemas y mejorando la capacidad de respuesta ante situaciones inesperadas.

La implementación en una Raspberry Pi 3 resultó ser una solución eficiente y de bajo costo para entornos pequeños, ideal para redes con presupuestos limitados, ofreciendo un buen equilibrio entre rendimiento y costo.

En general, la combinación de estas herramientas ofrece una alternativa escalable para el monitoreo de redes LAN. Sin embargo, algunas limitaciones, como la dependencia de la configuración inicial y la necesidad de ajustes manuales, persisten. Como trabajo futuro, se podrían explorar mejoras en la automatización y en la integración de inteligencia artificial para optimizar la detección de anomalías.

Apéndice

- Archivo de configuración del enrutador Cisco:

  A continuación se presenta la configuración de ejecución (*running-config*) del enrutador R1, obtenida mediante la intrucción:

  `show running-config`

```
R1#show running-config
Building configuration...

Current configuration : 5191 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
!
hostname R1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
no ip icmp rate-limit unreachable
ip cef
!
ip name-server 8.8.8.8
no ipv6 cef
!
multilink bundle-name authenticated
ip tcp synwait-time 5
!
interfaz FastEthernet0/0
 ip address 192.168.1.1 255.255.255.0
 ip nat inside
 duplex full
!
interfaz GigabitEthernet1/0
 mac-address e4b3.1812.4120
 ip address dhcp
 ip nat outside
 negotiation auto
!
```

```
ip default-gateway 192.168.122.208
ip nat inside source list 1 interfaz GigabitEthernet1/0 overload
ip forward-protocol nd
!
no ip http server
no ip http secure-server
access-list 1 permit any
!
snmp-server community public RW
snmp-server enable traps snmp authentication linkdown linkup coldstart
    warmstart
% rest of the snmp-server configurations
!
control-plane
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
 stopbits 1
line vty 0 4
 login
!
end

R1#
```

- Archivo de configuración del conmutador Cisco (ESW1):

  A continuación se presenta la configuración de ejecución (*running-config*) del conmutador ESW1, obtenida mediante el comando `show running-config`:

```
ESW1#show running-config
Building configuration...

Current configuration : 6776 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
no service dhcp
!
hostname ESW1
!
boot-start-marker
boot-end-marker
!
no aaa new-model
memory-size iomem 5
```

```
no ip routing
no ip icmp rate-limit unreachable
no ip cef
!
ip name-server 8.8.8.8
!
multilink bundle-name authenticated
!
interfaz FastEthernet0/0
 description *** Unused for Layer2 Etherconmutador ***
 no ip address
 no ip route-cache
 shutdown
 duplex auto
 speed auto
!
interfaz FastEthernet1/0
 duplex full
 speed 100
!
interfaz Vlan1
 ip address 192.168.1.2 255.255.255.0
 no ip route-cache
!
snmp-server community public RO
snmp-server enable traps snmp authentication linkdown linkup coldstart
    warmstart
snmp-server host 192.168.1.2 version 2c public udp-port 161
snmp-server host 192.168.1.30 version 2c public
!
control-plane
!
banner exec ^C

**************************************************************
This is a normal enrutador with a conmutador module inside (NM-16ESW)
It has been pre-configured with hard-coded speed and duplex

To create vlans use the command "vlan␣database" in exec mode
After creating all desired vlans use "exit" to apply the config

To view existing vlans use the command "show␣vlan-conmutador␣brief"
**************************************************************

^C
alias configure va macro global trace add_vlan $v
alias configure vd macro global trace del_vlan $v
alias exec vl show vlan-conmutador brief
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
```

```
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
end

ESW1#
```

- Archivo de configuración del conmutador Cisco (ESW2):

  A continuación se presenta la configuración de ejecución (*running-config*) del conmutador ESW2, obtenida mediante el comando `show running-config`:

```
ESW2#show running-config
Building configuration...

Current configuration : 3543 bytes
!
version 12.4
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
no service dhcp
!
hostname ESW2
!
boot-start-marker
boot-end-marker
!
no aaa new-model
memory-size iomem 5
no ip routing
no ip icmp rate-limit unreachable
no ip cef
!
ip name-server 8.8.8.8
!
multilink bundle-name authenticated
!
interfaz FastEthernet0/0
 description *** Unused for Layer2 Etherconmutador ***
 no ip address
 no ip route-cache
 shutdown
 duplex auto
 speed auto
!
interfaz FastEthernet1/0
 duplex full
 speed 100
!
interfaz Vlan1
 ip address 192.168.1.40 255.255.255.0
 no ip route-cache
```

```
!
ip default-gateway 192.168.1.1
ip forward-protocol nd
!
snmp-server community public RO
snmp-server enable traps snmp authentication linkdown linkup coldstart
    warmstart
snmp-server enable traps envmon
snmp-server enable traps config
snmp-server enable traps syslog
snmp-server host 192.168.1.40 version 2c public udp-port 161
!
banner exec ^C

**************************************************************
This is a normal enrutador with a conmutador module inside (NM-16ESW)
It has been pre-configured with hard-coded speed and duplex

To create vlans use the command "vlan␣database" in exec mode
After creating all desired vlans use "exit" to apply the config

To view existing vlans use the command "show␣vlan-conmutador␣brief"
**************************************************************

^C
alias configure va macro global trace add_vlan $v
alias configure vd macro global trace del_vlan $v
alias exec vl show vlan-conmutador brief
!
line con 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line aux 0
 exec-timeout 0 0
 privilege level 15
 logging synchronous
line vty 0 4
 login
!
end

ESW2#
```

- Configuración de cada VPCS en GNS3:

  Esta configuración se aplica a cada PC virtual (VPCS) en GNS3. La única modificación necesaria es cambiar la dirección IP según sea el caso para cada dispositivo.

```
# This is the configuration for PC14
#
# Uncomment the following line to enable DHCP
# dhcp
# or the line below to manually setup an IP address and subnet mask
# ip 192.168.1.1 255.0.0.0
```

```
#

set pcname PC14
```

- Código python para detección de anomalías utilizando el isolation forest:

```
import requests
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.ensemble import IsolationForest
from email.mime.text import MIMEText
from email.mime.multipart import MIMEMultipart
from email.mime.base import MIMEBase
from email import encoders
import smtplib

# Configuraci n de Zabbix
ZABBIX_API_URL = "http://192.168.100.194:8080/api_jsonrpc.php"
ZABBIX_USER = "Admin"
ZABBIX_PASSWORD = "Zabbix"

def zabbix_login():
    """Inicia sesi n en la API de Zabbix y devuelve el token de autenticaci n
        ."""
    headers = {'Content-Type': 'application/json'}
    payload = {
        "jsonrpc": "2.0",
        "method": "user.login",
        "params": {
            "user": ZABBIX_USER,
            "password": ZABBIX_PASSWORD
        },
        "id": 1,
        "auth": None
    }
    response = requests.post(ZABBIX_API_URL, headers=headers, json=payload)
    return response.json()['result']

def get_history(auth_token, host_id, item_id):
    """Obtiene el historial de datos de Zabbix para un host e  tem
        espec ficos."""
    headers = {'Content-Type': 'application/json'}
    payload = {
        "jsonrpc": "2.0",
        "method": "history.get",
        "params": {
            "output": "extend",
            "history": 0,
            "itemids": item_id,
            "hostids": host_id,
            "sortfield": "clock",
            "sortorder": "DESC",
            "limit": 3000
        },
        "auth": auth_token,
```

```
        "id": 2
    }
    response = requests.post(ZABBIX_API_URL, headers=headers, json=payload)
    return response.json()['result']

def preprocess_data(data):
    """Convierte los datos recibidos de Zabbix en un DataFrame de Pandas."""
    df = pd.DataFrame(data)
    df['clock'] = pd.to_datetime(df['clock'].astype(int), unit='s')  #
        Convertir tiempo
    df['value'] = pd.to_numeric(df['value'])  # Asegurar que los valores sean
        num ricos
    return df[['clock', 'value']]

def detect_anomalies(df):
    """Detecta anomal as usando Isolation Forest."""
    model = IsolationForest(contamination=0.05)  # Establecer un umbral de
        contaminaci n
    df['anomaly'] = model.fit_predict(df[['value']])
    return df

def plot_anomalies(df, title):
    """Genera un gr fico de anomal as y lo guarda como imagen."""
    plt.figure(figsize=(10, 6))
    plt.plot(df['clock'], df['value'], label='Uso')

    anomalies = df[df['anomaly'] == -1]  # Filtrar las anomal as detectadas
    plt.scatter(anomalies['clock'], anomalies['value'], color='red', label='
        Anomal a', marker='x')

    plt.title(title)
    plt.xlabel('Tiempo')
    plt.ylabel('Uso')
    plt.legend()
    plt.savefig(f'{title}.png')
    plt.close()

def send_email(subject, body, attachments):
    """Env a un correo con un asunto, cuerpo y archivos adjuntos."""
    sender_email = "iseomago@gmail.com"
    receiver_email = "iseo.martinez@estudiante.uacm.edu.mx"
    password = "vcto␣ozdi␣kysv␣hdzj"  # Se recomienda usar variables de entorno
         para la contrase a.

    # Crear el mensaje
    msg = MIMEMultipart()
    msg['From'] = sender_email
    msg['To'] = receiver_email
    msg['Subject'] = subject
    msg.attach(MIMEText(body, 'plain'))

    # Adjuntar archivos
    for file in attachments:
        with open(file, 'rb') as attachment:
            part = MIMEBase('application', 'octet-stream')
```

```
            part.set_payload(attachment.read())
            encoders.encode_base64(part)
            part.add_header('Content-Disposition', f'attachment; filename={file
                }')
            msg.attach(part)

    # Enviar el correo
    with smtplib.SMTP('smtp.gmail.com', 587) as server:
        server.starttls()
        server.login(sender_email, password)
        server.sendmail(sender_email, receiver_email, msg.as_string())

if __name__ == "__main__":
    # Obtener token de autenticaci n de Zabbix
    auth_token = zabbix_login()

    # Configuraci n de Zabbix
    host_id = "10084"
    cpu_item_id = "42269"
    memory_item_id = "42270"

    # Obtener historial de CPU y memoria
    cpu_history = get_history(auth_token, host_id, cpu_item_id)
    memory_history = get_history(auth_token, host_id, memory_item_id)

    # Procesar y detectar anomal as
    cpu_df = preprocess_data(cpu_history)
    cpu_df = detect_anomalies(cpu_df)
    plot_anomalies(cpu_df, 'Anomal as de uso de CPU')

    memory_df = preprocess_data(memory_history)
    memory_df = detect_anomalies(memory_df)
    plot_anomalies(memory_df, 'Anomal as de uso de memoria')

    # Enviar los resultados por correo electr nico
    send_email(
        subject="Informe de anomal as de CPU y Memoria",
        body="Se han detectado anomal as en el uso de CPU y memoria. Las
            im genes se adjuntan.",
        attachments=['Anomal as de uso de CPU.png', 'Anomal as de uso de
            memoria.png']
    )
```

Bibliografía

[1] A. S. Tanenbaum y D. J. Wetherall, *Redes de Computadoras*, 5th. Upper Saddle River, NJ: Prentice Hall, 2011.

[2] Cloudflare, *Las redes de área personal (PAN) en la conectividad moderna*, Accedido: 23 de enero de 2023, 2023. Disponible en: `https://www.cloudflare.com/es-es/`.

[3] J. F. Kurose y K. W. Ross, *Computer Networking: A Top-Down Approach*, 7th. Boston: Pearson, 2017.

[4] I. of Electrical y E. E. (IEEE), *Familia de estándares 802 para redes LAN*, Accedido: 23 de enero de 2023, 2023. Disponible en: `https://standards.ieee.org/`.

[5] W. Stallings, *Redes de Computadoras con Protocolos de Internet y Tecnología*. Pearson, 2021.

[6] A. Palacios, *Apuntes de Clase de Topología y Diseño de Redes*, Documento físico proporcionado por el Dr. Alfredo Palacios en la UACM, 2022.

[7] Lifeder, *Topología de Malla*, Accedido: 23 de enero de 2023, 2023. Disponible en: `https://www.lifeder.com/topologia-de-malla/`.

[8] Zabbix, *Zabbix Monitoring System*, Accedido: 23 de enero de 2023, 2023. Disponible en: `https://www.zabbix.com/`.

[9] Zabbix, *Time Offset in Zabbix*, Accedido: 23 de enero de 2023, 2023. Disponible en: `https://www.zabbix.com/documentation/current/manual/maintenance/time_offset`.

[10] W. Stallings, *Comunicaciones y Redes de Computadoras*, 6th. Madrid, España: Prentice Hall, 2020, ISBN: 978-8420529868.

[11] ISO/IEC, *ISO/IEC 7498-1:1984 - Open Systems Interconnection (OSI) - Basic Reference Model: The Basic Model*, Accedido: 23 de enero de 2023, International Organization for Standardization (ISO) e International Electrotechnical Commission (IEC), 1984.

[12] J. Dordogne, *Redes Informáticas: Nociones Fundamentales*, 8th. Francia: Ediciones ENI, 2022, ISBN: 978-2409038266.

[13] J. Postel, *Internet Protocol*, RFC 791, Internet Engineering Task Force (IETF), Septiembre 1981, 1981. Disponible en: `https://www.ietf.org/rfc/rfc791.txt`.

[14] ServerSpace, *Cómo Configurar Zabbix Server y Frontend en Ubuntu 22.04*, Accedido: 23 de enero de 2023, 2023. Disponible en: `https://www.serverspace.co/blog/how-to-configure-zabbix-server-and-frontend-on-ubuntu-22-04/`.

[15] W. Foundation, *Wireshark - Network Protocol Analyzer*, Accedido: 23 de enero de 2023, 2025. Disponible en: `https://www.wireshark.org/`.

[16] M. Miller, *Gestión de Redes con SNMP*, 3rd. New York, USA: Wiley, 1999, ISBN: 978-0471297268.

[17] Cisco, *What is a Network Operations Center (NOC)?* Accedido: 23 de enero de 2023, 2023. Disponible en: `https://www.cisco.com/c/en/us/solutions/collateral/enterprise-networks/network-operations-center/nb-06-noc-edu.html`.

[18] P. Cabantous, *Redes Informáticas - Guía Práctica para la Gestión, Seguridad y Supervisión*, 1st. Francia: Ediciones ENI, 2024, ISBN: 978-2409045325.

[19] G. Team, *GNS3: A Network Simulator*, Accedido: 23 de enero de 2023, 2023. Disponible en: `https://www.gns3.com/`.

[20] A. Vladishev, *Zabbix: La Solución de Monitoreo Distribuido de Código Abierto de Clase Empresarial*. Riga, Letonia: Zabbix SIA, 2005.

[21] I. Corporation, *Especificación de la Interfaz de Gestión de Plataforma Inteligente*, Versión 2.0, 2004.

[22] S. Microsystems, *Especificación de Java Management Extensions (JMX)*, Versión 1.4, 2006.

[23] G. Labs, *Grafana: La Plataforma de Observabilidad Abierta y Componible*, Accedido: 23 de enero de 2023, 2023. Disponible en: `https://grafana.com/`.

[24] G. Labs, *Soluciones de Monitoreo de Grafana*, Accedido: 23 de enero de 2023, 2023. Disponible en: https://grafana.com/solutions/monitoring/.

[25] G. Labs, *Arquitectura de Loki*, Accedido: 23 de enero de 2023, 2023. Disponible en: https://grafana.com/docs/loki/latest/get-started/architecture/.

[26] R. P. Foundation, *Raspberry Pi 3 Model B+*, Accedido: 23 de enero de 2023, 2023. Disponible en: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/.

[27] J. J. Estévez Pereira, «Detección de anomalías en la red empleando técnicas de machine learning,» *Repositorio Universitario de la Universidad de La Coruña*, 2020. Disponible en: https://ruc.udc.es/dspace/bitstream/handle/2183/26827/J.J_Est%C3%A9vez_Pereira_2020_Detecci%C3%B3_de_anomal%C3%ADas_en_la_red.pdf.

[28] J. C. A. Quiñonez, K. D. M. Lara y A. C. S. Chávez, «Manual de Instalación, Configuración de un Sistema de Gestión y Monitoreo de Redes Informáticas para Pequeñas y Medianas Empresas en El Salvador, utilizando software libre,» Accedido: 23 de enero de 2023, Tesis doct., Universidad Tecnológica de El Salvador, 2017. Disponible en: http://biblioteca.utec.edu.sv/siab/virtual/tesis/941001022.pdf.

[29] T. IT, *Monitorizar PostgreSQL con Zabbix 6*, Accedido: 23 de enero de 2023, 2023. Disponible en: https://www.tutorialesit.com/monitorizar-postgresql-con-zabbix-6.

[30] Zabbix, *Documentación de Zabbix 6.4*, Accedido: 23 de enero de 2023, 2023. Disponible en: https://www.zabbix.com/documentation/current/manual.

[31] Cyberciti.biz, *Cómo Configurar Nginx para Enviar Charset UTF-8 en Unix / Linux*, Accedido: 23 de enero de 2023, 2021. Disponible en: https://www.cyberciti.biz/faq/howto-nginx-webserver-send-charset-utf-8-under-unix-linux/.

[32] TecAdmin, *Cómo Instalar y Configurar el Servidor Zabbix en Ubuntu 22.04*, Accedido: 23 de enero de 2023, 2023. Disponible en: https://tecadmin.net/install-zabbix-on-ubuntu-22-04/.

[33] Shapehost, *Instalación de Zabbix en Ubuntu 22.04*, Accedido: 23 de enero de 2024, 2024. Disponible en: https://www.shapehost.com/knowledgebase/instalacion-de-zabbix-en-ubuntu-22-04.

[34] idroot, *Cómo Instalar Zabbix en Ubuntu 22.04 LTS*, Accedido: 23 de enero de 2024, 2024. Disponible en: `https://idroot.us/install-zabbix-ubuntu-22-04/`.

[35] C. Systems, *Configuración del protocolo SNMP en un router Cisco*, Accedido: 23 de enero de 2023, 2022. Disponible en: `https://www.cisco.com/c/es_mx/support/docs/ip/simple-network-management-protocol-snmp/7282-12.html`.

[36] Z. Team, *Integración de Zabbix con Telegram*, Accedido: 23 de enero de 2023, 2023. Disponible en: `https://www.zabbix.com/documentation/current/manual/config/notifications/media/telegram`.

[37] R. P. Foundation, *Raspberry Pi OS Documentation*, Accedido: 23 de enero de 2023, 2023. Disponible en: `https://www.raspberrypi.com/software/`.