



Universidad Nacional Mayor de San Marcos

Universidad del Perú. Decana de América

Facultad de Ingeniería Electrónica y Eléctrica

Escuela Profesional de Ingeniería Electrónica

**Implementación de un sistema de monitoreo de red
basado en Zabbix y en el protocolo SNMP para un ISP**

TRABAJO DE SUFICIENCIA PROFESIONAL

Para optar el Título Profesional de Ingeniero Electrónico

AUTOR

Frank RAMOS CLEMENTE

ASESOR

Rossina Isabel GONZALES CALIENES

Lima, Perú

2026



Reconocimiento - No Comercial - Compartir Igual - Sin restricciones adicionales

<https://creativecommons.org/licenses/by-nc-sa/4.0/>

Usted puede distribuir, remezclar, retocar, y crear a partir del documento original de modo no comercial, siempre y cuando se dé crédito al autor del documento y se licencien las nuevas creaciones bajo las mismas condiciones. No se permite aplicar términos legales o medidas tecnológicas que restrinjan legalmente a otros a hacer cualquier cosa que permita esta licencia.

Referencia bibliográfica

Ramos, F. (2026). *Implementación de un sistema de monitoreo de red basado en Zabbix y en el protocolo SNMP para un ISP*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería Electrónica y Eléctrica, Escuela Profesional de Ingeniería Electrónica]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	Frank Ramos Clemente
Tipo de documento de identidad	DNI
Número de documento de identidad	72956403
URL de ORCID	No Aplica
Datos de asesor	
Nombres y apellidos	Rossina Isabel Gonzales Calienes
Tipo de documento de identidad	DNI
Número de documento de identidad	09389306
URL de ORCID	https://orcid.org/0000-0002-6353-2712
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Jorge Elías Moscoso Sánchez
Tipo de documento	DNI
Número de documento de identidad	07206008
Miembro del jurado 1	
Nombres y apellidos	Wilbert Chavez Irazabal
Tipo de documento	DNI
Número de documento de identidad	08121733
Miembro del jurado 2	
Nombres y apellidos	Daniel Díaz Ataucuri
Tipo de documento	DNI
Número de documento de identidad	07139361
Datos de investigación	
Línea de investigación	ODS 9: INDUSTRIA, INNOVACIÓN E INFRAESTRUCTURA / Redes de Datos (26)
Grupo de investigación	No Aplica
Agencia de financiamiento	No Aplica

Ubicación geográfica de la investigación	Edificio: ON Empresas País: Perú Departamento: Lima Provincia: Lima Distrito: San Borja Latitud: -12.098733939203242 Longitud: -77.01172773479077
Año o rango de años en que se realizó la investigación	2023
URL de disciplinas OCDE	Telecomunicaciones https://purl.org/pe-repo/ocde/ford#2.02.05 Ingeniería de sistemas y comunicaciones https://purl.org/pe-repo/ocde/ford#2.02.04 Ingeniería eléctrica, Ingeniería electrónica https://purl.org/pe-repo/ocde/ford#2.02.01



UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
(Universidad del Perú, DECANA DE AMÉRICA)
FACULTAD DE INGENIERÍA ELECTRÓNICA Y ELÉCTRICA
Teléfono 619-7000 Anexo 4226
Calle Germán Amezaga 375 – Lima 1 – Perú



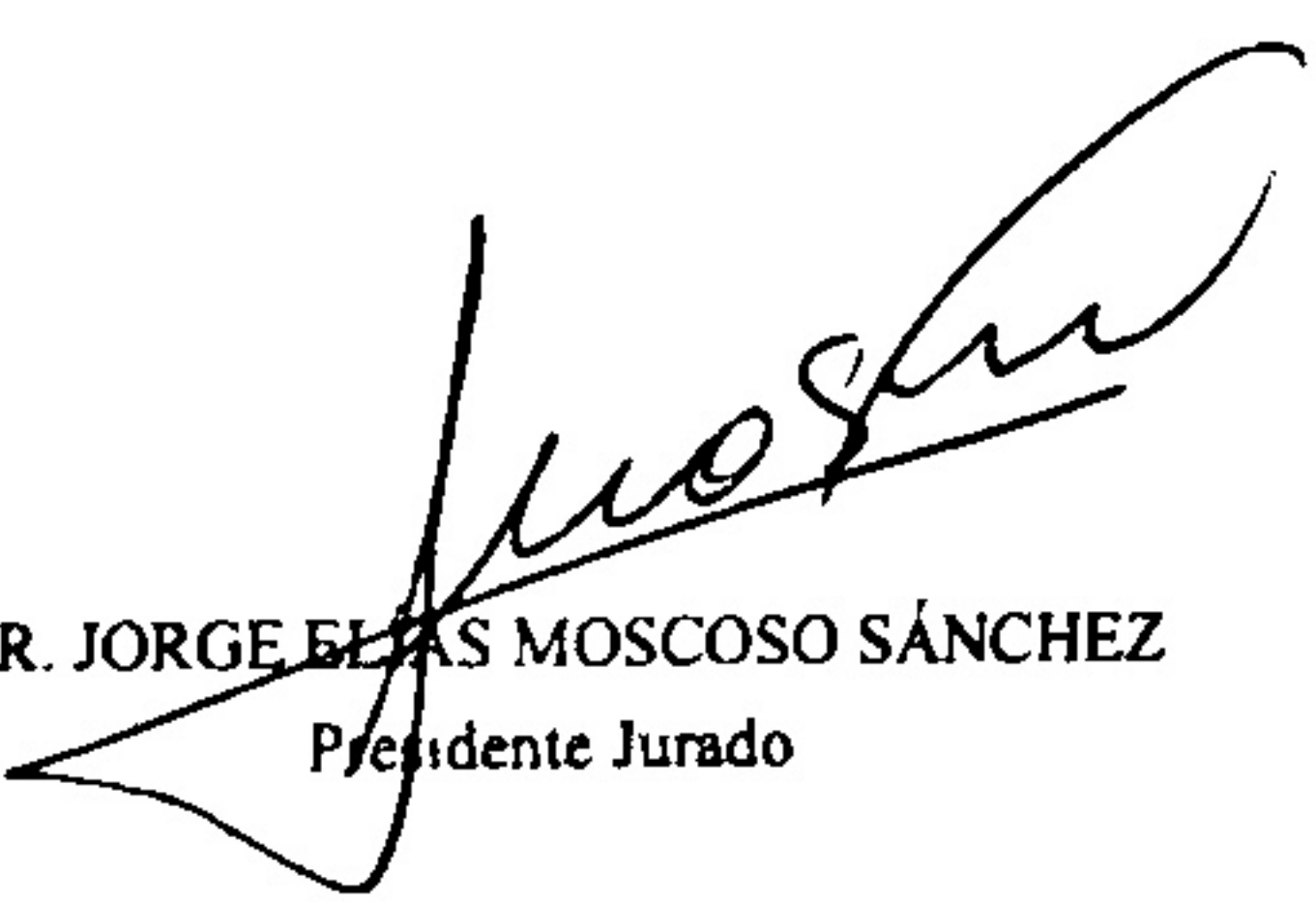
ACTA DE SUSTENTACIÓN
TRABAJO DE
SUFICIENCIA PROFESIONAL
ACTA N°50/FIEE-PPP/2026

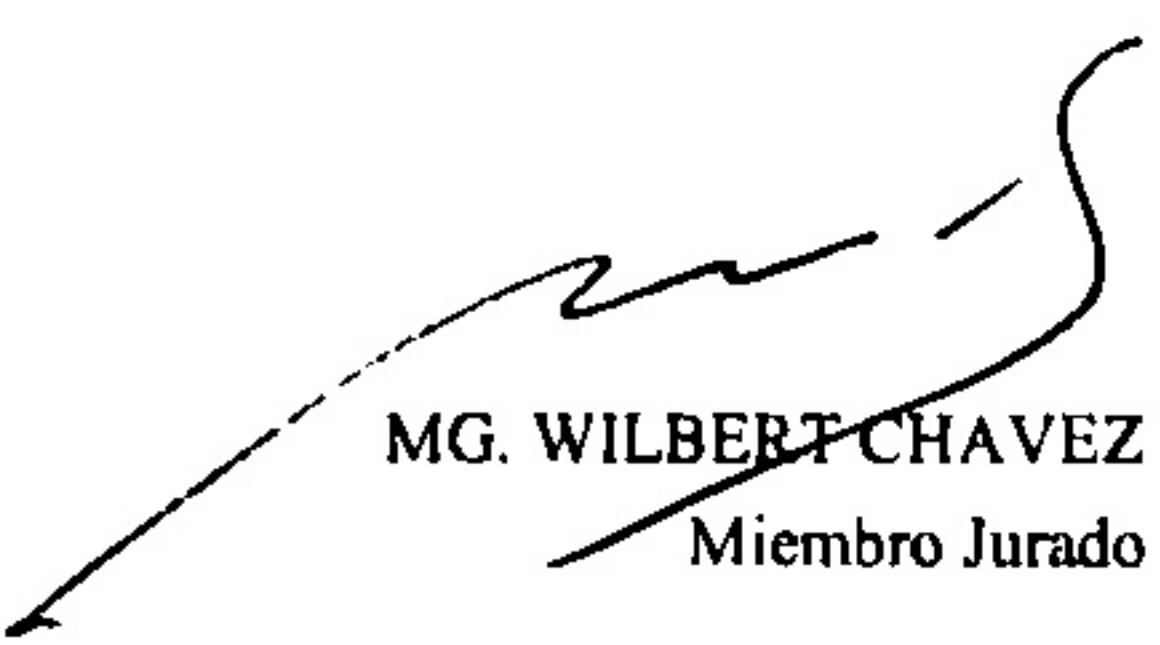
Los suscritos Miembros del Jurado, docentes permanentes de las Escuelas Profesionales de la Facultad de Ingeniería Electrónica y Eléctrica, reunidos en la fecha 17 de Julio del 2026, como presidente de Jurado el **DR. JORGE ELÍAS MOSCOSO SÁNCHEZ**, integrado por el Miembro de Jurado el **MG. WILBERT CHAVEZ IRAZABAL**, el Miembro de Jurado el **MG. DANIEL DÍAZ ATAUCURI** y Miembro Asesor el **ING. ROSSINA ISABEL GONZALES CALIENES**.

Después de escuchar la Sustentación Presencial del Trabajo de Suficiencia Profesional del Bachiller **FRANK RAMOS CLEMENTE** con código 14190022 que para optar el Título Profesional de Ingeniero Electrónico sustentó el Trabajo de Suficiencia Profesional titulado **IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO DE RED BASADO EN ZABBIX Y EN EL PROTOCOLO SNMP PARA UN ISP**.

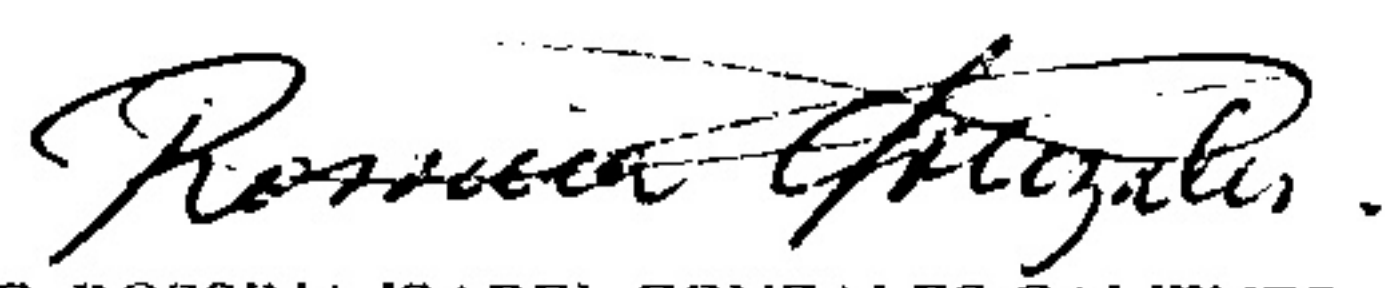
El jurado examinador procedió a formular las preguntas reglamentarias y, luego de una deliberación en privado, decidió otorgarle el calificativo de 17 (Diecisiete).

Ciudad Universitaria, 17 de julio del 2026


DR. JORGE ELÍAS MOSCOSO SÁNCHEZ
Presidente Jurado


MG. WILBERT CHAVEZ IRAZABAL
Miembro Jurado


MG. DANIEL DÍAZ ATAUCURI
Miembro Jurado


ING. ROSSINA ISABEL GONZALES CALIENES
Miembro Asesor



UNIVERSIDAD NACIONAL MAYOR DE
SAN MARCOS
Universidad del Perú, Decana de América

CERTIFICADO DE SIMILITUD

Yo, **ROSSINA ISABEL GONZALES CALIENES** en mi condición de asesora acreditada con el Dictamen N°52/FIEE-PPP/2026 del Trabajo de Suficiencia Profesional titulado **IMPLEMENTACIÓN DE UN SISTEMA DE MONITOREO DE RED BASADO EN ZABBIX Y EN EL PROTOCOLO SNMP PARA UN ISP**, presentado por el bachiller **FRANK RAMOS CLEMENTE**, para optar el título profesional de Ingeniero Electrónico. CERTIFICO que se ha cumplido con lo establecido en la Directiva de Originalidad y de Similitud de Trabajos Académicos, de Investigación y Producción Intelectual. Según la revisión, análisis y evaluación mediante la plataforma PIDS, el documento evaluado cuenta con el 2% de similitud, nivel PERMITIDO para continuar con los trámites correspondientes y para su publicación en el repositorio institucional.

Se emite el presente certificado en cumplimiento de lo establecido en las normas vigentes, como uno de los requisitos para la obtención del título profesional de Ingeniero Electrónico.

Lima, 17 de Julio del 2026

Firma del Asesor

DNI: 09389306

Nombres y apellidos del asesor:

ROSSINA ISABEL GONZALES CALIENES

DEDICATORIA

Este trabajo está dedicado a mi madre, Elizabeth Clemente, por ser el soporte de mi vida; a mi padre, Simeon Ramos; y a mis hermanos, porque este logro es el resultado de nuestro sacrificio familiar.

Finalmente, a la memoria de P. R. C., por su entrañable amistad durante aquellas largas noches de estudio, y por acompañarme y cuidarme siempre.

AGRADECIMIENTOS

Agradezco a Dios por su infinita bondad y por guiar mi camino hacia este logro.

A mi asesora, la Ing. Rossina Gonzales, por su excelente disposición y valiosa orientación a lo largo del desarrollo del presente trabajo.

A mi jefe, Juan Puyen, y a mi equipo de trabajo, por las facilidades brindadas y el apoyo constante para culminar este objetivo.

Finalmente, a la Universidad Nacional Mayor de San Marcos y a la Facultad de Ingeniería Electrónica y Eléctrica, por mi formación profesional.

RESUMEN

El presente trabajo describe el diseño, implementación y validación de un sistema de monitoreo proactivo para redes empresariales de un proveedor de servicios de internet, con el objetivo de reducir los tiempos de diagnóstico reactivo. La metodología que se usó incluye el despliegue del servidor Zabbix y el diseño de la red de gestión fuera de banda. Posteriormente, se configuró en los equipos de premisa de cliente (CPE), Cisco y MikroTik, el protocolo SNMPv2c y listas de control de acceso (ACL) para finalmente realizar una integración del sistema con la API de Telegram para notificaciones en tiempo real. Como resultado, tras la ejecución de los escenarios presentados, se evidenció que la plataforma recolecta correctamente métricas de *hardware* y emite las alertas correspondientes ante eventos como caídas de enlace, saturación de ancho de banda, entre otros. Se concluye que la solución presentada mejora la visibilidad de la red, garantiza la detección temprana de incidencias y favorece el cumplimiento de los acuerdos de nivel de servicio.

Palabras clave: SNMP, Zabbix, Monitoreo de red, proveedor de servicios de internet, telecomunicaciones.

ABSTRACT

This work describes the design, implementation, and validation of a proactive monitoring system for enterprise networks of an Internet service provider, with the objective of reducing reactive diagnostic times. The methodology used includes the deployment of the Zabbix server and the design of the Out-of-Band management network. Subsequently, the SNMPv2c protocol and access control lists (ACL) were configured on the Cisco and MikroTik customer premises equipment (CPE), to finally perform system integration with the Telegram API for real time notifications. As a result, following the execution of the presented scenarios, it was evidenced that the platform correctly collects *hardware* metrics and issues the corresponding alerts for events such as link failures, bandwidth saturation, among others. It is concluded that the presented solution improves network visibility, guarantees the early detection of incidents, and favors compliance with service level agreements.

Keywords: SNMP, Zabbix, Network monitoring, Internet service provider, telecommunications.

ÍNDICE GENERAL

RESUMEN	iv
ABSTRACT	v
ÍNDICE GENERAL	vi
LISTA DE FIGURAS	x
LISTA DE CUADROS	xiii
CAPITULO I	1
INTRODUCCIÓN	1
1.1 Objetivo del informe	1
1.2 Estructura del informe	1
CAPITULO II	3
INFORMACIÓN DEL LUGAR DONDE SE DESARROLLÓ LA ACTIVIDAD	3
2.1. Institución – Actividad que desarrolla	3
2.2. Periodo de duración de la actividad	3
2.3. Finalidad y objetivos de la entidad	3
2.4. Razón social	3
2.5. Dirección postal	3
2.6. Nombre y correo electrónico del profesional a cargo	4
CAPITULO III	5
DESCRIPCIÓN DE LA ACTIVIDAD	5

3.1 Organización de la actividad	5
3.2 Finalidad y objetivos de la Actividad	6
3.2.1 Finalidad	6
3.2.2 Objetivos	6
3.3 Problemática	7
3.3.1 Problema General	8
3.3.2 Problemas Específicos	8
3.3.3 Justificación e importancia de la investigación	9
3.4 Metodología	10
3.4.1 Bases teóricas	10
3.4.2 Marco conceptual	10
3.5 Procedimiento	16
3.5.1. Análisis y diseño de la red de monitoreo	16
3.5.2. Implementación del servidor NMS Zabbix en Ubuntu Server	18
3.5.3. Configuración del protocolo SNMP en equipos de borde	21
3.5.4 Integración de dispositivos y creación de alertas	22
3.5.5 Triggers y umbrales de alerta	23
3.5.6 Integración del canal de alertas mediante la API de Telegram	24
3.5.7. Simulación de fallos y evaluación de alertas	25
3.6 Resultados de la actividad	27
3.6.1. Escenario 1: Simulación de caída de enlace (CPE Cisco)	27
3.6.2. Escenario 2: Saturación de ancho banda (CPE MikroTik)	28
3.6.3. Escenario 3: Ataque DoS desde la red LAN (Seguridad)	30
CAPITULO IV	32
DISCUSION DE RESULTADOS	32
4.1 Descripción de la implementación	32

	viii
4.1.1 <i>Arquitectura de cómputo del servidor Zabbix</i>	32
4.1.2 <i>Escalabilidad del sistema.</i>	33
4.1.3 <i>Diseño de seguridad.</i>	36
4.2 Evaluación Económica	36
CAPITULO V	38
CONCLUSIONES	38
CAPITULO VI	39
RECOMENDACIONES	39
CAPITULO VII	40
REFERENCIAS BIBLIOGRÁFICAS	40
CAPITULO VIII	42
ANEXOS	42
Anexo A: Configuración de red y SNMP del CPE Cisco	42
Anexo B: Configuración de red y SNMP del CPE MikroTik	43
Anexo C: Extracto de script de integración de Zabbix con Telegram	44
Anexo D: Definición de umbrales críticos (<i>Triggers</i>)	45
Anexo E: Cuadro de identificadores de objetos (OIDs)	46
Anexo F: Creación y configuración de la base de datos MariaDB por consola	47
Anexo G: Configuración de conexión a la base de datos MariaDB	47
Anexo H: Creación del Bot de Telegram y obtención del API Token mediante BotFather.	48
Anexo I: Vinculación del Token en el Media Type de Zabbix.	49
Anexo J: Registro manual de host y asignación de credenciales SNMP en Zabbix.	49
Anexo K: Configuración de la regla de autodescubrimiento (Network Discovery) de equipos en la red.	50

Anexo L: Diseño y estructuración del mapa topológico en Zabbix	50
Anexo M: Topología lógica de la red del sistema de monitoreo implementada en el emulador.	51
Anexo N: Dashboard de monitoreo implementado en la interfaz web de Zabbix	52

LISTA DE FIGURAS

- Figura 1 Organigrama de la estructura funcional del área ASTC de ON Empresas (2023).* Fuente. Elaboración propia. 5
- Figura 2 Arquitectura del protocolo SNMP.* Fuente. Tomado de ManageEngine. 11
- Figura 3 Estructura jerárquica de una MIB.* Fuente. Tomado de Huawei Technologies Co., Ltd. 12
- Figura 4 Panel de control o dashboard de Zabbix.* Fuente. Tomado de Zabbix SIA. 13
- Figura 5 Configuración de cola simple en MikroTik para el control de ancho de banda.* Fuente. Elaboración propia a partir de interfaz de MikroTik. 14
- Figura 6 Configuración de **trigger** con histéresis en Zabbix para el monitoreo de CPU.* Fuente. Elaboración propia a partir de la interfaz de Zabbix. 15
- Figura 7 Configuración de macros en plantilla para la estandarización de umbrales.* Fuente. Elaboración propia a partir de la interfaz de Zabbix. 16
- Figura 8 Topología lógica de la red de monitoreo y enlaces de clientes corporativos.* Fuente. Elaboración propia. 17
- Figura 9 Configuración de parámetros de red estáticos mediante Netplan.* Fuente. Elaboración propia. 19
- Figura 10 Frontend principal del servidor Zabbix.* Fuente. Elaboración propia a partir de la interfaz web de Zabbix 20
- Figura 11 Registro manual del CPE Cisco en Zabbix.* Fuente. Elaboración propia 22

Figura 12 Panel de **triggers**, umbrales y nivel de severidad de alertas para el equipo MikroTik. Fuente. Elaboración propia a partir de Zabbix 23

Figura 13 Vinculación de Zabbix con Telegram vía el token API. Fuente. Elaboración propia. 24

Figura 14 Telegram notificando alertas emitidas por el servidor Zabbix. Fuente. Elaboración propia 25

Figura 15 Apagado administrativo de la subinterfaz de servicio G0/0.100. Fuente. Elaboración propia a partir de CLI de Cisco IOS. 27

Figura 16 Contenido del PDU SNMP Get-Response donde se observa el OID 1.3.6.1.2.1.2.2.1.8.11 con un valor de “2”, lo que indica que la subinterfaz G0/0.100 sí está apagada. Fuente. Elaboración propia a partir de Wireshark. 28

Figura 17 Notificación recibida en Telegram donde se observa el host afectado y el estado operacional actual. Fuente. Elaboración propia a partir de Telegram 28

Figura 18 Generación de tráfico en red LAN de cliente usando Iperf3 29

Figura 19 Gráfica de Zabbix mostrando cómo las políticas de tráfico hacen efecto en la capacidad de la red de cliente y cómo se alcanza el umbral de 45 Mbps. Fuente. Elaboración propia a partir de Zabbix. 29

Figura 20 Dashboard de Zabbix alerta de saturación. Fuente. Elaboración propia a partir de Zabbix. 30

Figura 21 Simulación del ataque UDP Flood con Iperf3 desde un host mostrando múltiples hilos de conexión y paquetes de tamaño mínimo. Fuente. Elaboración propia a partir de la herramienta Putty. 30

Figura 22 Gráfica de Zabbix evidenciando picos anómalos en el uso de CPU, tras iniciados los ataques UDP Flood. Fuente. Elaboración propia a partir de Zabbix. 31

Figura 23 Dashboard de Zabbix que muestra la alerta “High CPU utilization” en estado activo. Fuente. Elaboración propia a partir de Zabbix 31

Figura 24 **Comandos de inicialización, creación de usuario y asignación de permisos para la base de datos del sistema de monitoreo.** Fuente. Elaboración propia a partir de Ubuntu Server. 47

Figura 25 **Parámetros de conexión a la base de datos durante la configuración inicial de la interfaz web de Zabbix.** Fuente. Elaboración propia a partir de Zabbix. 48

Figura 26 **Proceso de creación del bot de notificaciones y generación del API Token mediante la plataforma BotFather.** Fuente. Elaboración propia a partir de Telegram. 48

Figura 27 **Configuración del tipo de medio (Media Type) de Telegram en el servidor Zabbix.** Fuente. Elaboración propia a partir de Zabbix. 49

Figura 28 **Configuración del host en Zabbix para el monitoreo del dispositivo mediante SNMP.** Fuente: Elaboración propia a partir de Zabbix. 49

Figura 29 **Configuración de la regla de autodescubrimiento en Zabbix mediante barrido de direcciones IP y consultas de identificadores SNMP.** Fuente. Elaboración propia a partir de Zabbix. 50

Figura 30 **Interfaz de configuración del mapa de red en Zabbix, mostrando la vinculación de elementos gráficos con los equipos monitoreados.** Fuente. Elaboración propia a partir de Zabbix. 50

Figura 31 **Topología lógica de la red en GNS3.** Fuente. Elaboración propia. 51

Figura 32. **Dashboard de monitoreo en interfaz web de Zabbix.** Fuente. Elaboración propia a partir de Zabbix. 52

LISTA DE CUADROS

<i>Cuadro 1</i> Direccionamiento IP de los dispositivos de la red de monitoreo	17
<i>Cuadro 2</i> Especificaciones técnicas de la VM Zabbix	33
<i>Cuadro 3</i> Requerimientos de hardware según el tamaño de la instalación en Zabbix	35
<i>Cuadro D1</i> Umbrales de alerta del sistema de monitoreo	45
<i>Cuadro E1</i> Identificadores de objetos estándar y propietarios más comunes	46

CAPITULO I INTRODUCCIÓN

En el entorno actual de las telecomunicaciones, la continuidad del servicio de internet juega un papel crítico en el sector corporativo. Históricamente, se ha visto que la gestión reactiva de la red aumenta el tiempo de solución de averías y compromete el cumplimiento de los acuerdos de nivel de servicio (SLA). Por otro lado, la visibilidad técnica sobre los equipos de premisa de cliente (CPE) se suele limitar a pruebas de conectividad básica, no existe visibilidad detallada sobre el estado del hardware y el comportamiento del tráfico. A partir de esta problemática, el presente trabajo de suficiencia profesional presenta el diseño y la simulación de un sistema de monitoreo proactivo para redes empresariales. El sistema propuesto se basa en el despliegue de la plataforma de código abierto Zabbix 7.0 y el uso del protocolo SNMPv2c. La viabilidad de esta implementación se justifica en su capacidad de extraer métricas (como uso de procesador, memoria RAM y saturación de interfaces) en *routers* Cisco y MikroTik, aunque también es aplicable a equipos de otras marcas. A su vez, el proyecto se apoya en los principios de segmentación de red, gestión fuera de banda (Out-of-Band), y en la automatización de los mensajes de alerta mediante notificaciones móviles integradas con Telegram.

1.1 Objetivo del informe

El presente trabajo de suficiencia profesional presenta el diseño y la simulación de un sistema de monitoreo proactivo para redes empresariales.

1.2 Estructura del informe

El documento está estructurado en seis capítulos principales. El Capítulo II trata sobre la empresa en donde se realizó la implementación del proyecto, su perfil corporativo y las actividades de la organización. El Capítulo III detalla la organización de la actividad, abarcando desde las configuraciones lógicas de la red hasta el despliegue

del servidor de monitoreo. El Capítulo IV presenta la discusión de la implementación, incluyendo el análisis de escalabilidad de los recursos informáticos. Posteriormente, los Capítulos V y VI muestran cada uno las conclusiones obtenidas de la validación del sistema y las recomendaciones de mejora que se pueden realizar al producto obtenido, respectivamente. Finalmente, se incluye las referencias bibliográficas y los anexos técnicos.

CAPITULO II

INFORMACIÓN DEL LUGAR DONDE SE DESARROLLÓ LA ACTIVIDAD

2.1. Institución – Actividad que desarrolla

ON Empresas – Operador de Telecomunicaciones

2.2. Periodo de duración de la actividad

El proyecto fue diseñado, implementado y puesto en marcha en el año 2023.

2.3. Finalidad y objetivos de la entidad

Según el sitio oficial de ON Empresas, en su sección “Quienes Somos”, es una compañía peruana del rubro de las telecomunicaciones que se orienta al sector B2B (segmento empresarial) y que se especializa en brindar soluciones de conectividad y transformación digital a las organizaciones.

Visión:

La visión de ON Empresas es brindar soluciones tecnológicas integrales a las empresas para que impulsen su transformación, acompañándolas en su crecimiento con innovación, agilidad y servicio especializado.

Misión:

La misión de ON Empresas es consolidarse y ser líder en el segmento B2B de telecomunicaciones, impulsando la evolución digital de las organizaciones con soluciones de alto valor.

2.4. Razón social

La razón social de la empresa es ON EMPRESAS S.A.C., con RUC 20552504641.

2.5. Dirección postal

Av. José Galvez Barrenechea Nro. 645

2.6. Nombre y correo electrónico del profesional a cargo

El proyecto estuvo bajo la supervisión de:

Profesional a cargo: Juan José Puyen Rodríguez

Cargo durante el proyecto: Supervisor del Área de Monitoreo.

Cargo actual: Coordinador de implementaciones (ON Negocios).

Correo corporativo: jpuyenr@on.pe

CAPITULO III

DESCRIPCIÓN DE LA ACTIVIDAD

3.1 Organización de la actividad

El desarrollo del presente trabajo de suficiencia profesional estuvo a cargo del autor, Frank Ramos Clemente. Si bien el despliegue original del sistema de monitoreo en ON Empresas requirió la participación de equipos multidisciplinarios (divididos en áreas que se encargaron de la infraestructura de servidores, configuración de red e integración con plataformas de notificación), para los fines de validación técnica de este documento, el autor diseñó, configuró y reprodujo la arquitectura de manera independiente en un entorno de simulación controlado.

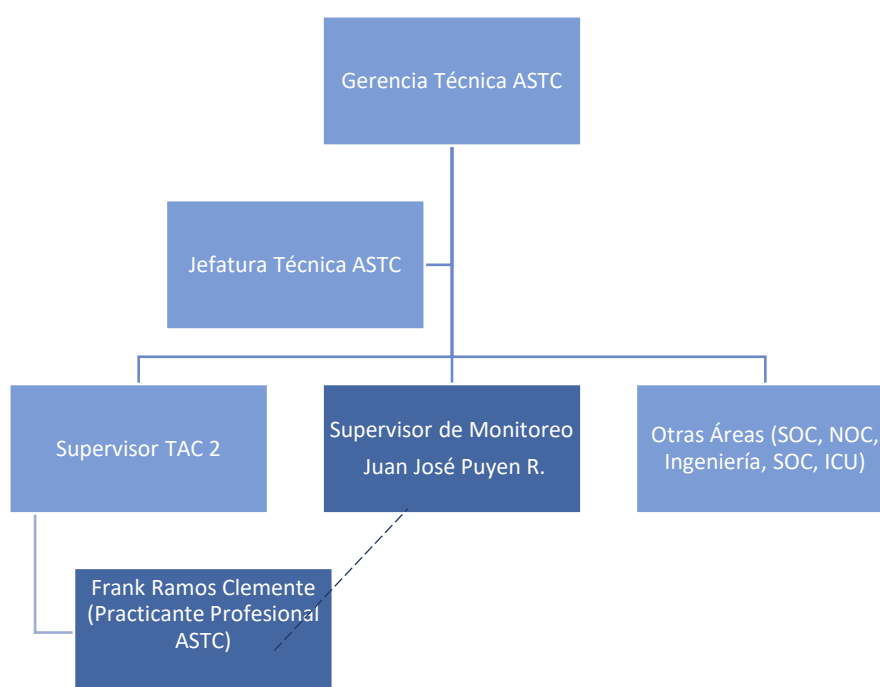


Figura 1 Organigrama de la estructura funcional del área ASTC de ON Empresas (2023). Fuente. Elaboración propia.

El desarrollo del trabajo se estructuró por bloques:

- **Evaluación y diseño:** Análisis de la problemática del soporte reactivo y diseño de la topología de red OOB, garantizando el aislamiento del tráfico de gestión.
- **Implementación:** Despliegue del servidor *network management system* (NMS) Zabbix 7.0 en un entorno Linux virtualizado, incluyendo la configuración del motor de base de datos y la integración de la API para la notificación de alertas del sistema.
- **Configuración:** Habilitación del protocolo SNMPv2c, definición de políticas de ancho de banda y restricción de accesos mediante listas de control de acceso (ACL) en los equipos CPE simulados.
- **Pruebas:** Ejecución de escenarios de pruebas y simulación de eventos, incluyendo caídas de enlace y saturación de tráfico, para validar la respuesta de los disparadores.

3.2 Finalidad y objetivos de la Actividad

3.2.1 Finalidad

La finalidad del presente trabajo de suficiencia profesional es ayudar a mejorar la continuidad de los servicios brindados al grupo de clientes corporativos *top*, denominación para los clientes corporativos de alto valor que tiene la empresa. Se busca mejorar la capacidad de respuesta que tiene el área de Monitoreo, pasando desde un soporte reactivo a uno proactivo, reduciendo así los tiempos de interrupción del servicio, velando por el cumplimiento de los SLA y, por consiguiente, una mejora en la experiencia del cliente.

3.2.2 Objetivos

3.2.2.1 Objetivo General

El objetivo general del presente trabajo es la implementación de un sistema de monitoreo de red basado en el *software* Zabbix y en el uso del protocolo SNMP para el monitoreo y diagnóstico proactivo de los servicios proporcionados a los clientes *top*; siendo más específicos, implementar una herramienta de supervisión integral de los equipos CPE.

3.2.2.2 Objetivos Específicos

Reconocer las variables y parámetros de estado, como el nivel de memoria, uso de la unidad central de procesamiento (CPU) y consumo de tráfico de red, que permitan monitorear el estado de los equipos CPE y analizar las causas de incidencias en los servicios proporcionados a los clientes.

Configurar adecuadamente el protocolo de administración SNMPv2c y las políticas de seguridad necesarias en los equipos CPE, se precisa el uso de "comunidades" para garantizar una comunicación segura.

Implementar la interfaz web de Zabbix, así como establecer la configuración necesaria en dicho *software* para el monitoreo en tiempo real y para la generación de alertas de incidencias

Verificar la operatividad de la herramienta creada mediante simulación de fallos y pruebas de conectividad, validando la correcta generación de gráficas y recepción de alertas.

3.3 Problemática

En el entorno empresarial actual, la continuidad operativa depende de una infraestructura de telecomunicaciones sólida. Las organizaciones contratan diversos servicios de conectividad corporativa bajo estrictos acuerdos de nivel de servicio. Para garantizar estos niveles de disponibilidad, los proveedores de servicios deben poseer la capacidad de detectar y diagnosticar incidencias de manera inmediata hasta el punto final del circuito, el equipo CPE. Sin embargo, las herramientas de gestión convencionales suelen limitarse a validar la conectividad básica o el caudal de tráfico. Esta visión superficial resulta insuficiente cuando se gestiona una diversidad de equipos de distintos fabricantes, impiden determinar con certeza la causa raíz de una interrupción: ya sea un corte en el medio físico (fibra óptica), una falta de energía en la sede del cliente o un bloqueo lógico del equipo.

En el caso de la empresa en estudio, el área de Monitoreo supervisa la totalidad de la red de clientes. Este departamento aplica protocolos de atención proactiva para toda su cartera, realizando contactos anticipados o gestionando el envío de cuadrillas técnicas ante la detección de caídas. No obstante, a pesar de contar con estos procedimientos establecidos, la eficacia de la gestión se veía limitada técnicamente

por la falta de visibilidad profunda sobre el estado de salud de los equipos de borde. Al no contar con alertas sobre parámetros críticos del *hardware* (CPU, memoria, entre otros) ni sobre el estado operativo de los servicios lógicos en el CPE, el personal de monitoreo carecía de la evidencia necesaria para anticiparse a fallos, lo que generaba puntos ciegos en el diagnóstico, especialmente críticos para la atención de los clientes *top*.

Para solventar esta carencia y potenciar la capacidad de respuesta, se seleccionó Zabbix (*software* de código abierto) como herramienta complementaria para la supervisión avanzada. Sin embargo, su despliegue y la integración con los equipos de borde representa un reto técnico significativo. No basta con una instalación básica; se requiere un conocimiento especializado para configurar correctamente el protocolo SNMP, dominar la sintaxis de los sistemas operativos que traen los distintos *vendors* (como IOS de Cisco, RouterOS de MikroTik o FortiOS de Fortinet), entender las distintas capas de red para aplicar seguridad, y manejar las múltiples opciones de *triggers* y plantillas que ofrece Zabbix. La dificultad radica en implementar en un sistema todos estos elementos de manera unificada, logrando una plataforma robusta que otorgue visibilidad granular y permita tomar decisiones preventivas basadas en el análisis integral del servicio (*hardware*, interfaces y conectividad).

3.3.1 Problema General

¿Cómo implementar un sistema de monitoreo de red basado en el protocolo SNMP y en el uso de Zabbix que permita supervisar y diagnosticar proactivamente el estado del servicio entregado en los equipos CPE, para optimizar la gestión de incidencias en los clientes *top* de la empresa?

3.3.2 Problemas Específicos

¿Qué variables y parámetros de estado operativo se requiere monitorear en los equipos de borde para poder identificar una incidencia por falla de *hardware* o por caída de enlace de fibra?

¿Cómo configuramos el protocolo de gestión y las políticas de acceso en los equipos CPE para lograr una extracción segura de los valores de las variables identificadas?

¿De qué manera implementamos una plataforma central, así como sus plantillas correspondientes, para una la obtención automática de datos y la generación de alertas de monitoreo cuando los valores sobrepasen cierto umbral?

¿Cómo validamos la operatividad del sistema implementado para garantizar que las gráficas y alertas de Zabbix muestren fielmente el estado real del servicio proporcionado al cliente?

3.3.3 Justificación e importancia de la investigación

Justificación práctica

El trabajo realizado tiene implicancia práctica directa y eficaz, ya que soluciona el problema mencionado que aquejaba al equipo de Monitoreo: La visibilidad limitada. Tras su implementación, se proporcionó al equipo técnico de una potente herramienta que les permite poder determinar la causa de una avería, que bien podría ser debido a una falla física (rotura de fibra) o una falla lógica (inhibición del equipo CPE, saturación de recursos). Finalmente, la puesta en marcha del sistema de monitoreo implementado tiene un impacto positivo inmediato en las actividades diarias; pues, permite que el diagnóstico o detección de averías sea más preciso, evitando el desplazamiento innecesario de personal cuando se trata de problemas que se pueden resolver remotamente, lo que facilita la restitución temprana del servicio, mejorando así su disponibilidad y la experiencia del cliente.

Justificación técnica

El trabajo realizado cumple un rol trascendental en la estandarización y modernización de la gestión de red de la empresa. Propicia el tránsito de un sistema de gestión de monitoreo básico a una estructura más avanzada, fundamentada en el aprovechamiento del protocolo SNMPv2c. La implementación del nuevo sistema de monitoreo demuestra que es posible la integración de sistemas heterogéneos (como el servidor Linux que aloja Zabbix con los distintos CPEs de la empresa) a través del uso de scripts de automatización, *templates*, y políticas de control de acceso, sofisticando así el nivel técnico de la operación y alineando a la organización en el empleo de las mejores prácticas de la industria de las telecomunicaciones.

3.4 Metodología

3.4.1 Bases teóricas

Parra y Oshita (2023) abordaron la problemática de la ausencia de supervisión ambiental en un nodo de telecomunicaciones de un ISP. Indicaron que algunas de las causas de la indisponibilidad o mal funcionamiento de los equipos de red se debe a condiciones térmicas o de humedad en el entorno. Frente a esto, implementaron un sistema de monitoreo basado en Zabbix que permitió la supervisión permanente de las condiciones ambientales a través de la recolección de datos captada por los sensores utilizados y con la correspondiente configuración para la generación de alertas de aviso al usuario. De esta manera, se facilitó la detección temprana de anomalías y, por ende, se contribuyó a la prevención de fallas físicas en equipos críticos. Concluyeron que su solución permitió garantizar condiciones ambientales adecuadas, reduciendo así los tiempos de indisponibilidad de la infraestructura de red del nodo, gracias a su capacidad preventiva como herramienta de monitoreo, lo que demuestra la utilidad de Zabbix en entornos de proveedores de servicios.

Mardiyono et al. (2020) implementaron una herramienta de monitoreo de red basada en Zabbix y un sistema de notificaciones de alerta inmediata mediante el uso de la aplicación Telegram. La solución, gracias al uso del protocolo SNMP, Zabbix Server y Zabbix Agent, fue aplicada en un entorno verdadero de interconexión de ISPs, con la que se pudo monitorear servidores y dispositivos de red críticos. Finalmente, concluyeron que integrando Zabbix junto a sistemas de notificación móviles se logra mejorar la eficiencia del monitoreo y, por lo tanto, la capacidad de respuesta ante incidentes en infraestructuras de telecomunicaciones.

González Sierra (2016) concluyó que el uso del protocolo estándar SNMP en entornos de proveedores de servicios de internet mejora la detección de fallas y facilita la administración de redes. La propuesta de su implementación se basó en el uso del protocolo mencionado en combinación de un *software* de código abierto, con lo cual se logró visualizar el tráfico en tiempo real, estado de enlaces, y dispositivos de red y se resolvió la problemática de la falta de visibilidad de la infraestructura de red en un ISP.

3.4.2 Marco conceptual

3.4.2.1 Protocolo Simple de Administración de Red (SNMP)

Es un protocolo estándar que permite el control y monitoreo de equipos de red y pertenece a la familia TCP/IP. SNMP opera en la capa de aplicación del modelo OSI (capa 7) y utiliza el protocolo UDP de capa de transporte, a través del puerto 161 realiza el envío de solicitudes o comandos (Get, Set, GetNext) y mediante el puerto 162 realiza la recepción de avisos o alertas no solicitadas (*traps*).

Su funcionamiento se basa en el modelo cliente – servidor, donde existe un gestor que envía mensajes de solicitud a un agente, el cual está ubicado en el dispositivo que se quiere monitorear. El agente responde a las solicitudes del gestor o también, por propia iniciativa, puede enviar notificaciones no solicitadas, *traps*.

En el trabajo realizado se utilizó SNMP en su versión SNMPv2c el cual es basado en cadenas de comunidad (*community strings*) que sirven como contraseñas que viajan en texto sin encriptar (Stallings, 2014) (Case et al., 1996).

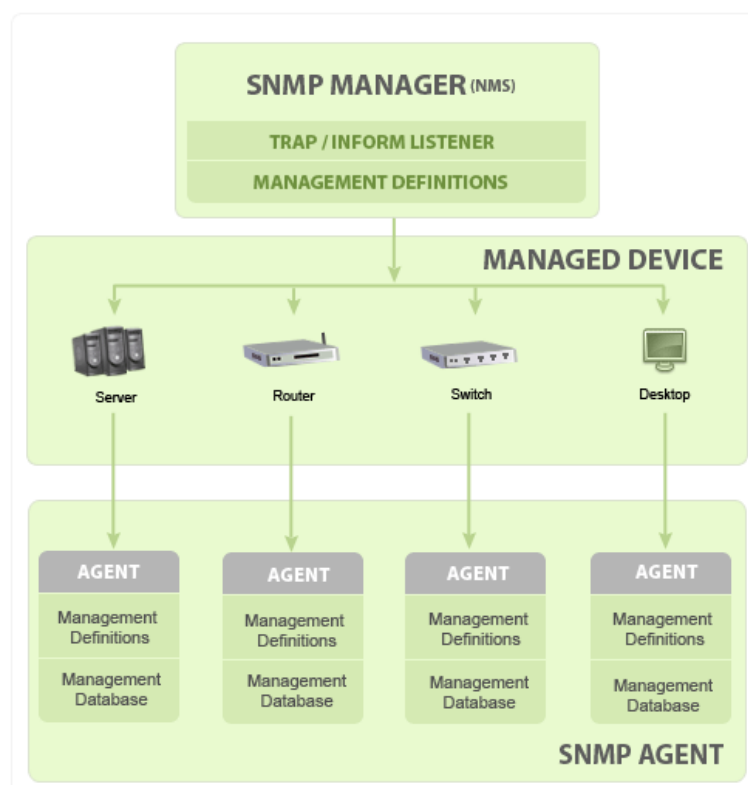


Figura 2 Arquitectura del protocolo SNMP. Fuente. Tomado de ManageEngine.

3.4.2.2 Management Information Base (MIB)

Según Presuhn (2002), en la RFC 3418, una *Management Information Base* (MIB) es una estructura lógica en forma de árbol invertido conformada por un conjunto de objetos gestionables pertenecientes a un dispositivo de red a través del protocolo SNMP. Cada uno de los objetos representa una variable propia del sistema y se identifican con un identificador único de objeto (OID). Estos son manejados por el agente SNMP y el gestor accede a ellos gracias a la estructura jerárquica del MIB a través de los OID. La arquitectura de una MIB está conformada por objetos estandarizados, los cuales están definidos por la IETF y ubicados en la rama *mgmt(2).mib(1)*, como por objetos privados propios de los distintos fabricantes ubicados en la rama *enterprises*, permitiendo así la supervisión de características comunes y particulares de cada dispositivo de red.

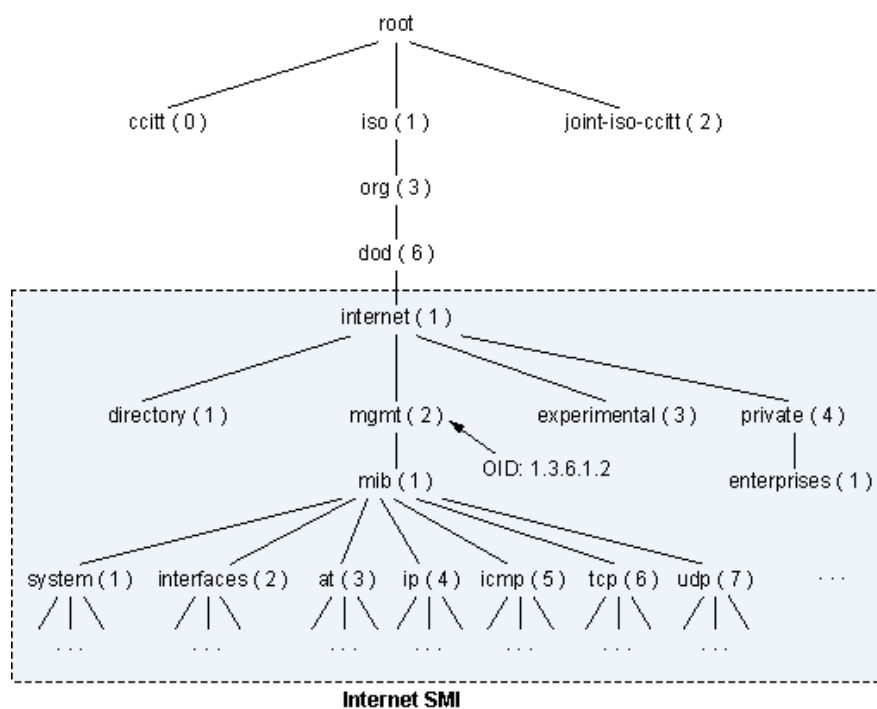


Figura 3 Estructura jerárquica de una MIB. Fuente. Tomado de Huawei Technologies Co., Ltd.

3.4.2.3 Sistema de monitoreo Zabbix.

Zabbix es un *software* de código abierto orientado al monitoreo de infraestructura tecnológica del sector empresarial; permite recolectar, almacenar y analizar métricas en tiempo real procedentes de equipos de red como servidores, *routers*, *switches*, máquinas virtuales, aplicaciones, entre otros. Tiene como característica importante su escalabilidad, lo cual permite monitorear muchos nodos y múltiples métricas desde una plataforma central. (Zabbix SIA, s. f.).

Su funcionamiento se basa en un modelo cliente-servidor, en donde el Zabbix Server es el elemento principal que se ocupa de tareas como procesamiento de datos recolectados, evaluar si estos cumplen determinadas condiciones o si sobrepasan algún umbral especificado y de generar las alertas de notificación. (Zabbix SIA, s. f.).

Zabbix tiene varias formas de recolectar datos; por ejemplo, a través del protocolo SNMP, vía agentes propios, consultas remotas, entre otros. Debido a esto, es posible su uso en entornos heterogéneos.

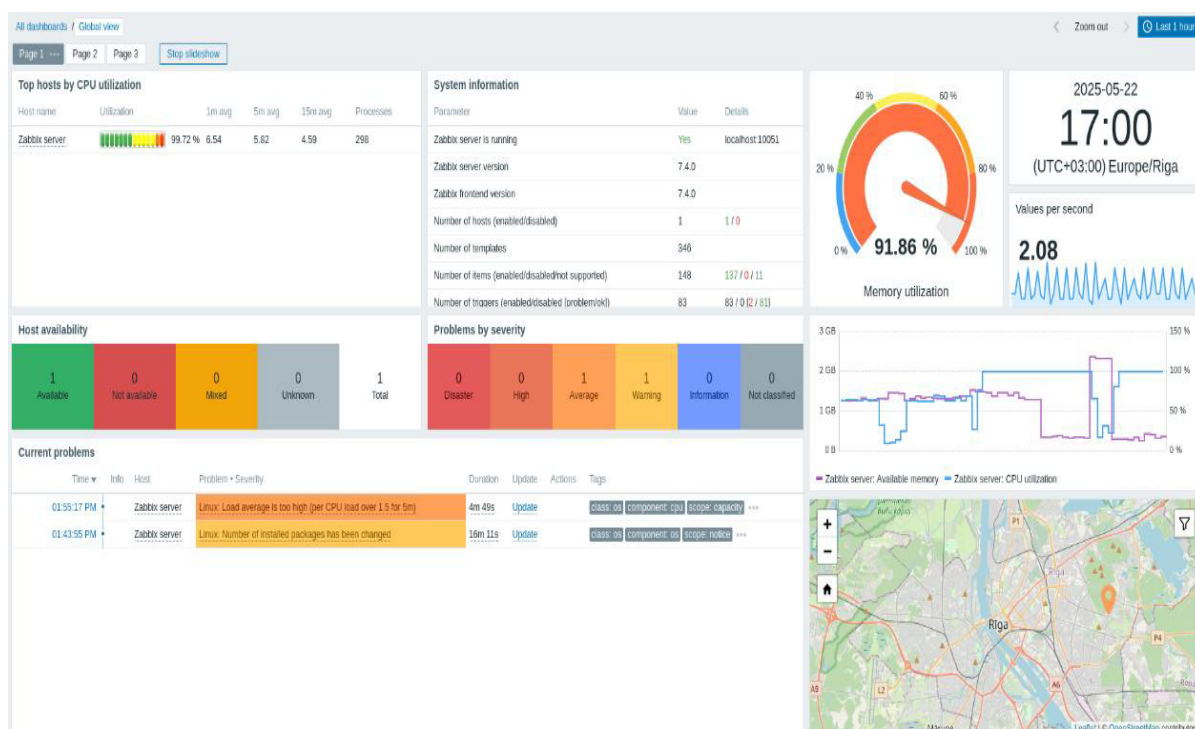


Figura 4 Panel de control o dashboard de Zabbix. Fuente. Tomado de Zabbix SIA.

3.4.2.4 Control de ancho de banda y limitación de tráfico (*Traffic Shaping*)

El control de ancho de banda consiste en restringir la tasa de transferencia máxima que una subred o dirección IP en específico pueden utilizar. Para el caso de los router MikroTik, esa característica se implementa haciendo uso de colas simples (*Simple Queues*). Esta función permite a los administradores de red definir un límite máximo (*Max Limit*) de tráfico, la velocidad de transmisión de datos se logra retrasando un búfer temporal o descartando paquetes cuando cliente supera su capacidad asignada. (MikroTik, s. f.).

Traffic Shapping es fundamental para garantizar una distribución apropiada de recursos de red por parte de los ISP. El sistema de monitoreo Zabbix puede calcular el porcentaje de uso de red de cada cliente y generar alertas proactivas de saturación.

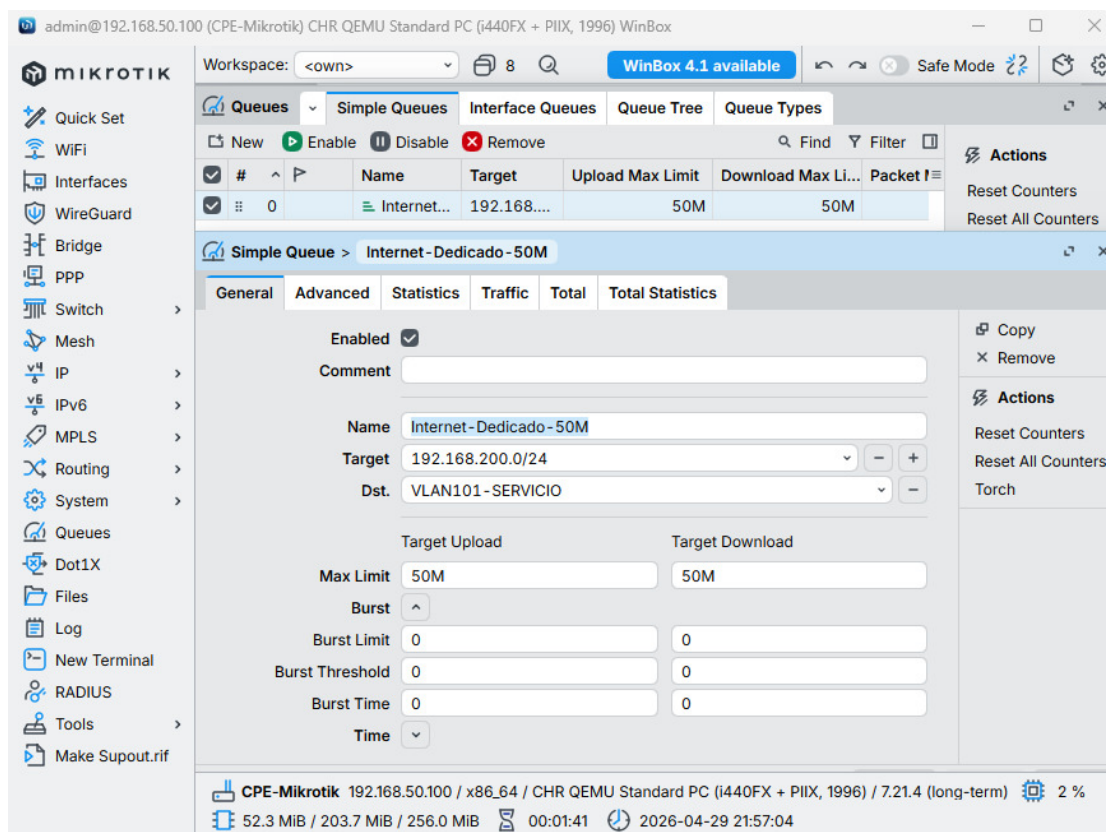


Figura 5 Configuración de cola simple en MikroTik para el control de ancho de banda. Fuente. Elaboración propia a partir de interfaz de MikroTik.

3.4.2.5. Disparadores (*Triggers*) e Histéresis

En los sistemas de monitoreo, como Zabbix, los *triggers* o disparadores son expresiones lógicas que evalúan los datos recopilados de los *hosts* agentes y lo comparan frente a un umbral establecido para determinar el estado actual de un componente. Si la métrica evaluada supera dicho umbral, el *trigger* cambia su estado lógico, generando un evento que representa un problema que permite disparar las alertas correspondientes (Zabbix SIA, s.f.).

En los entornos de red, las métricas como uso de CPU o ancho de banda, por ejemplo, suelen sufrir fluctuaciones, lo que ocasiona transiciones entre estados representan problemas (*problem*) y estados correctos (*ok*), a este fenómeno se le conoce como *flapping*. Se emplea histéresis como principio de control para definir los umbrales estrictos de activación de alarmas y los umbres más permisivos para su recuperación.

Por ejemplo, una alerta de saturación se puede activar cuando el tráfico sobrepase el 90% de la capacidad de la red, pero el problema no será considerado resuelto hasta que el tráfico descienda y se mantenga por debajo del 80% (Zabbix SIA, s.f.).

Para los ISP, la aplicación de histéresis es fundamental porque los operadores de red puedan atender solamente eventos de congestión, caída o degradación sostenida, desestimando las fluctuaciones momentáneas de tráfico.

The screenshot shows the Zabbix Trigger configuration page. The trigger is named "MikroTik: #1: High CPU utilization (PRUEBA TSP)". The event name is "MikroTik: #1: High CPU utilization (over {CPU.UTIL.CRIT}% for 1m)". The operational data is "Current utilization: {ITEM.LASTVALUE1}". The severity is set to "Warning". The problem expression is "min (/CPE-Mikrotik/system.cpu.util[hrProcessorLoad.1],1m) > {CPU.UTIL.CRIT}". The recovery expression is "max (/CPE-Mikrotik/system.cpu.util[hrProcessorLoad.1],1m) < 70". The OK event generation is set to "Expression". The PROBLEM event generation mode is set to "Single". The OK event closes is set to "All problems". The menu entry name is "Trigger URL". The menu entry URL is empty. The description is "CPU utilization is too high. The system might be slow to respond." The interface includes buttons for "Update", "Clone", "Delete", and "Cancel".

Figura 6 Configuración de *trigger* con histéresis en Zabbix para el monitoreo de CPU. Fuente. Elaboración propia a partir de la interfaz de Zabbix.

3.4.2.6 Macros de Usuario en Zabbix

Para la arquitectura de Zabbix, las macros de usuario son variables personalizadas que simplifican la gestión de configuración, mejorando la escalabilidad el mantenimiento del sistema de monitoreo. Estas variables se pueden configurar de manera global, a nivel de *template* o para un host individual y tienen la sintaxis `{$MACRO}` (Zabbix SIA, s. f.)

El uso de macros es una práctica común en ingeniería de los ISP, quienes gestionan un gran número de equipos de red. En lugar de configurar credenciales de forma

manual para cada dispositivo, por ejemplo, se utilizan macros como `{$SNMP_COMMUNITY}` para representar a las cadenas de comunidad del protocolo SNMP. Asimismo, con macros se pueden estandarizar umbrales de alerta como `{$CPU_UTIL. CRIT}`. En otras palabras, reducen drásticamente la carga administrativa y el margen de error humano permitir realizar cambios, y propagándolos automáticamente a los equipos asociados, modificando un único valor en la plantilla.

Macro	Effective value	Template value	Global value (configure)
<code>{CPU_UTIL.CRIT}</code>	90		
description			
<code>{FAN_CRIT_STATUS:'critical'}</code>	3		
description			
<code>{FAN_CRIT_STATUS:'shutdown'}</code>	4		
description			
<code>{FAN_WARN_STATUS:'notFunctioning'}</code>	6		
description			
<code>{FAN_WARN_STATUS:'warning'}</code>	2		
description			
<code>{ICMP_LOSS_WARN}</code>	20		
Warning threshold of ICMP packet loss in %.			
<code>{ICMP_RESPONSE_TIME_WARN}</code>	0.15		
Warning threshold of the average ICMP response time in seconds.			
<code>{IF_ERRORS.WARN}</code>	2		
Warning threshold of error packet rate. Can be used with interface name as context.			
<code>{IF_UTIL.MAX}</code>	90		
Used as a threshold in the interface utilization trigger.			

Update Clone Delete Delete and clear Cancel

Figura 7 Configuración de macros en plantilla para la estandarización de umbrales. Fuente. Elaboración propia a partir de la interfaz de Zabbix.

3.5 Procedimiento

3.5.1. Análisis y diseño de la red de monitoreo

Se basó el diseño de la arquitectura lógica de la red de monitoreo en la segmentación de tráfico, con el fin de garantizar gestión de los equipos y por temas de seguridad. Como se puede observar en la Figura 8, existe 2 planos: un plano de control para el acceso fuera de banda, OOB, y un plano de servicio o de datos, con el que los clientes acceden a internet. La separación del tráfico de administración con el de servicio permite al servidor Zabbix realizar consultas SNMP a través de una VLAN de gestión

dedicada; de esta manera, se asegura el acceso a los equipos CPE aun cuando este no tenga acceso a internet o ante saturaciones del enlace de servicio.

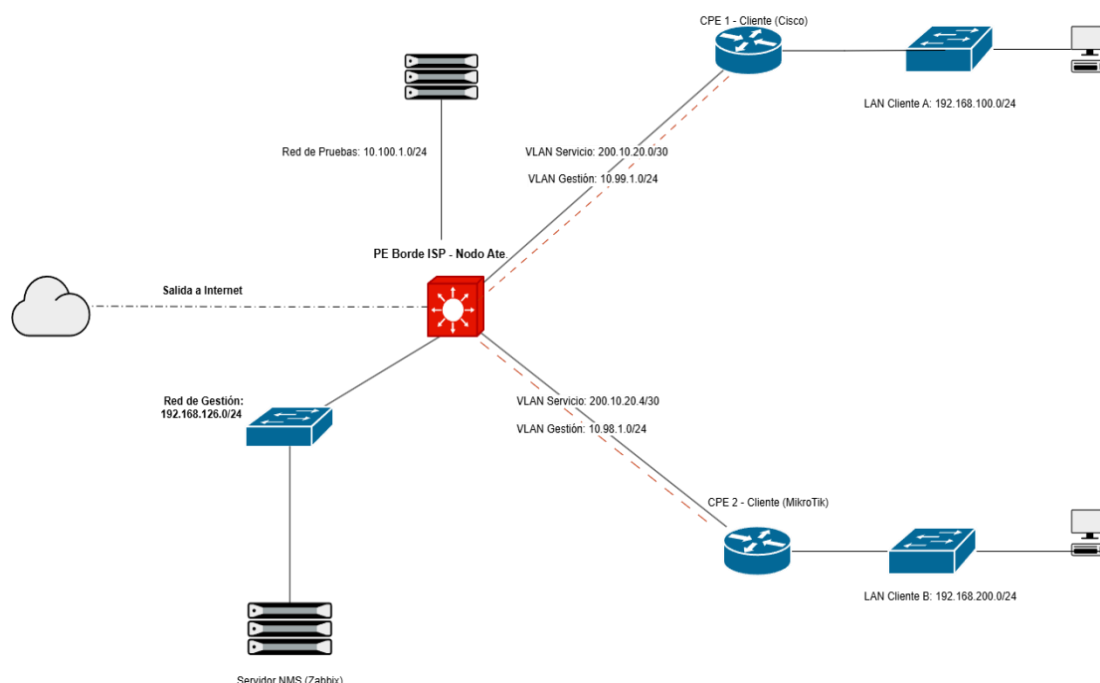


Figura 8 Topología lógica de la red de monitoreo y enlaces de clientes corporativos. Fuente. Elaboración propia.

El direccionamiento IP para cada uno de los elementos de la topología de red presenta en la figura anterior, se muestra en el siguiente cuadro:

Cuadro 1 Direccionamiento IP de los dispositivos de la red de monitoreo

Dispositivo	Interfaz Lógica / Rol	Dirección IP	Máscara de Subred
PE Borde ISP (Nodo Ate)	VLAN Servicio (Hacia Cisco)	200.10.20.1	255.255.255.252
	VLAN Gestión (Hacia Cisco)	10.99.1.1	255.255.255.0
	VLAN Servicio (Hacia MikroTik)	200.10.20.5	255.255.255.252
	VLAN Gestión (Hacia MikroTik)	10.98.1.1	255.255.255.0
	Red de Gestión (Gateway)	192.168.126.129	255.255.255.0
	Red de Pruebas (Gateway)	10.100.1.1	255.255.255.0

Dispositivo	Interfaz Lógica / Rol	Dirección IP	Máscara de Subred
Servidor NMS (Zabbix)	Interfaz LAN (Gestión)	192.168.126.10	255.255.255.0
Servidor de Pruebas	Interfaz LAN (Tráfico iPerf3)	10.100.1.10	255.255.255.0
CPE 1 (Cisco)	VLAN Servicio (WAN)	200.10.20.2	255.255.255.252
	VLAN Gestión (OOB)	10.99.1.2	255.255.255.0
	LAN Cliente A	192.168.100.1	255.255.255.0
CPE 2 (MikroTik)	VLAN Servicio (WAN)	200.10.20.6	255.255.255.252
	VLAN Gestión (OOB)	10.98.1.2	255.255.255.0
	LAN Cliente B	192.168.200.1	255.255.255.0

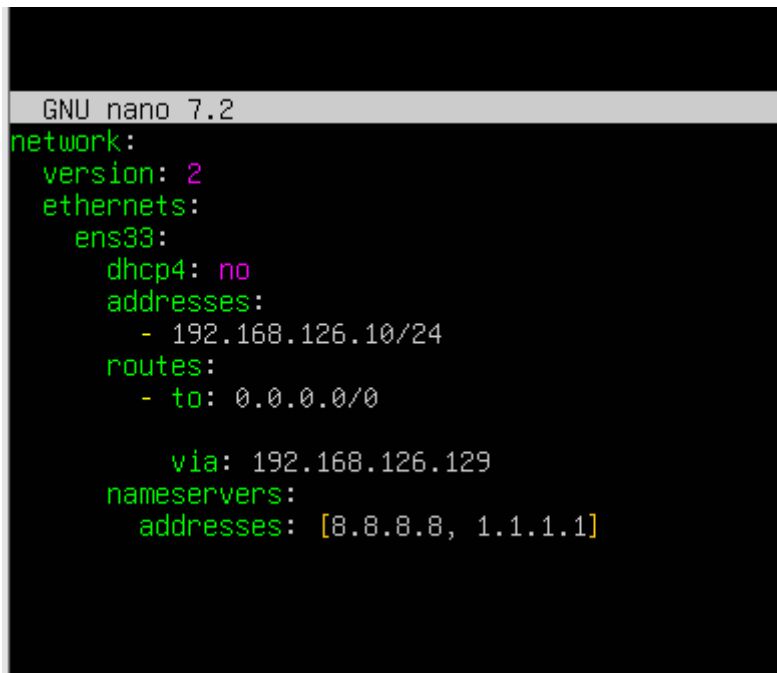
Fuente. Elaboración propia.

3.5.2. Implementación del servidor NMS Zabbix en Ubuntu Server

Se aprovisionó un servidor Linux dedicado utilizando el sistema operativo Ubuntu Server en su versión 24.04 LTS para en él alojar el sistema de gestión de red (NMS). Se decidió tomar esa distribución debido a que no cuenta con una interfaz gráfica, lo cual ayuda a optimizar los recursos de *hardware*. Asimismo, en cuanto a las características de *hardware* del servidor, se le asignó 40 GB como almacenamiento interno, 4GB de RAM y 2 núcleos en CPU.

3.5.2.1 Configuración de red y acceso.

Es necesario establecer una dirección IP estática en el servidor NMS para que pueda ser siempre accesible desde cualquier lugar de la red de gestión, para esto se hizo uso de la utilidad de Linux Netplan. Se editó el archivo de configuración YAML, colocando como dirección de red IPv4 la 192.168.126.10/24. Asimismo, para lograr una administración remota segura vía SSH, se instaló OpenSSH.



```
GNU nano 7.2
network:
  version: 2
  ethernets:
    ens33:
      dhcp4: no
      addresses:
        - 192.168.126.10/24
      routes:
        - to: 0.0.0.0/0

          via: 192.168.126.129
      nameservers:
        addresses: [8.8.8.8, 1.1.1.1]
```

Figura 9 Configuración de parámetros de red estáticos mediante Netplan.

Fuente. Elaboración propia.

3.5.2.2 Motor de base de datos

Todo el historial de monitoreo, las métricas SNMP que se obtienen de los equipos a censar, las alertas de caídas que obtiene Zabbix, se deben almacenar con un motor de base de datos relacional. En este sentido, se hizo uso de MariaDB, el cual es un sistema de gestión de bases de datos relacionales de código abierto de alto rendimiento.

Con el uso de lenguaje SQL, se creó una base de datos llamada “Zabbix” y se brindaron privilegios administrativos a un usuario local para la conexión interna del sistema.

3.5.2.3 Instalación y configuración de Zabbix

Se eligió implementar Zabbix en su versión 7.0 LTS debido a que es más reciente, estable y moderna. Luego de importar el repositorio oficial del desarrollador, se procedió los paquetes esenciales para el funcionamiento del sistema, los cuales son:

- **Zabbix-agent:** El cual es un *software* que normalmente se instala en los equipos a monitorear. Esta herramienta permitirá monitorear la propia salud del sistema.

- **Zabbix-server-mysql:** Quien se encargará de realizar las consultas SNMP y evaluarlas según los umbrales de alerta definidos.
- **Zabbix-frontend-php:** Paquete que contiene el código fuente lenguaje php con el que se dibuja la interfaz web, botones y gráficos de Zabbix.
- **Zabbix-apache-conf:** Para que la interfaz de Zabbix pueda mostrarse a los clientes web se usa Apache como servidor.
- **Zabbix-sql-scripts:** Con el que se cargarán las plantillas predefinidas de monitoreo que usa Zabbix.

Con estos binarios ya instalados, prosiguió inyectar el contenido del paquete Zabbix-sql-scripts, el cual construirá en la base de datos MariaDB tablas predeterminadas para almacenar los históricos, alertas y datos de usuarios.

Finalmente, se procedió a establecer la conexión entre el motor principal y la base de datos, además de configurar Ubuntu para que inicie los servicios automáticamente cada vez que el sistema operativo arranque.

3.5.2.4 Primeros pasos en el *frontend*

Haciendo uso de un navegador web, se procedió a configurar la interfaz gráfica de Zabbix, a través del asistente de configuración básico. Se validó también el correcto funcionamiento de todos los componentes que trae el sistema y se definió la zona horaria correcta, America/Lima, establecer este parámetro correctamente es fundamental para que haya coincidencia entre las marcas de tiempo que nos arrojará el sistema con la hora real en que suceden los eventos de los clientes.

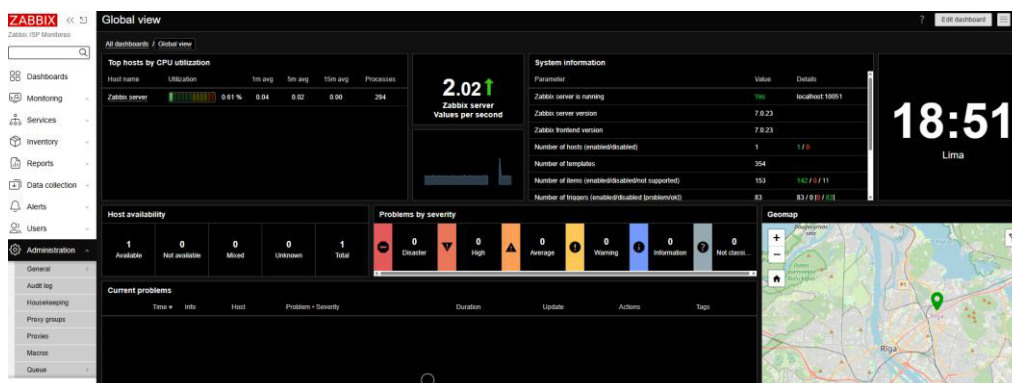


Figura 10 Frontend principal del servidor Zabbix. Fuente. Elaboración propia a partir de la interfaz web de Zabbix

3.5.3. Configuración del protocolo SNMP en equipos de borde

Para que Zabbix pueda extraer datos de los equipos CPE, se habilitó en estos últimos agentes SNMP. En el diseño de este sistema, se usó el protocolo SNMP en su versión v2c (SNMPv2c), el cual es basado en comunidades, es estándar y ofrece un mejor rendimiento.

3.5.3.1 Configuración en CPE Cisco

Una debilidad de las versiones 1 y 2 de SNMP es que usan cadenas de comunidad en texto plano, para enfrentar esto y como medida de seguridad adicional, se introdujo en el diseño el uso ACLs. De este modo, el CPE monitoreado solo responderá a consultas SNMP que provengan del servidor especificado, Zabbix, y denegará cualquier otro intento de consulta de dispositivos no identificados. Los comandos aplicados son:

```
! Crear la ACL y dar permiso únicamente al servidor Zabbix
access-list 10 permit 192.168.126.10

! Vinculacion de la ACL 10 con la comunidad SNMP
snmp-server community snmp_pro RO 10
snmp-server location Nodo-Ate-CPE-Corporativo
snmp-server enable traps entity-sensor threshold
```

3.5.3.2 Configuración en CPE MikroTik

Para los equipos MikroTik se usó tanto la herramienta Winbox, que ofrece una configuración gráfica, y también se trabajó por líneas de comando (CLI). Se habilitó el servicio SNMP y se estableció en la configuración que el *router* debía responder solo a las consultas provenientes del servidor Zabbix, como medida de seguridad.

```
/snmp community set [ find default=yes] name=snmp_pro
addresses=192.168.126.10/32
/snmpp set enabled=yes contact="Area_Monitoreo" location="Ate-
Sede-Cpe"
```

3.5.3.3 Configuración de enrutamiento de gestión

Para asegurar la conectividad bidireccional entre los equipos CPE y el servidor NMS se utilizó enrutamiento estático. Para el equipo CPE Cisco, se implementó una ruta

estática hacia la red de Zabbix, usando como salida la subinterfaz de gestión GigabitEthernet0/0.99 hacia el Gateway 10.99.1.1, el PE. Análogamente, en el CPE MikroTik, se añadió una ruta hacia la red 192.168.126.0/24 apuntando al Gateway de la VLAN de gestión (VLAN 98) el cual tiene como dirección 10.98.1.1

3.5.4 Integración de dispositivos y creación de alertas

Tras configurar los equipos de borde para trabajar como agentes SNMP, se procedió a integrarlos en la plataforma de monitoreo Zabbix. Se empleó 2 metodologías para este proyecto: registro manual y registro automático por autodescubrimiento.

3.5.4.1 Registro Manual

En este proyecto, se empleó este método para el registro del CPE Cisco. Se usó el *template* predeterminado que viene en Zabbix para dispositivos de este *vendor* Cisco IOS by SNMP. Esta plantilla contiene todos los identificadores de objetos necesarios para poder monitorear distintas características como uso de memoria, tiempo de encendido, uso de CPU, entre otros.

En la sección de configuración de *host*, se introdujo la dirección IP de gestión del equipo (10.99.1.2) de la subinterfaz asociada con la VLAN 99. Asimismo, se configuró la macro {\$SNMP_COMMUNITY} con el valor snmp_pro. El uso de macros permite realizar ágilmente cambios de seguridad de manera global.

Macro	Value	Description
{MEMORY_UTIL_MAX}	90	Used as a threshold in the interface utilization trigger.
{SNMP_COMMUNITY}	snmp_pro	description

Figura 11 Registro manual del CPE Cisco en Zabbix. Fuente. Elaboración propia

3.5.4.2 Registro por Autodescubrimiento

Para el CPE MikroTik se usó un método distinto de matrícula al sistema, el autodescubrimiento. Con esta función, el servidor Zabbix detecta automáticamente los nuevos *routers* conectados a la red y los agrega a la lista de dispositivos monitoreados.

En el presente proyecto, esta técnica se empleó como un proceso de 2 pasos. Primero se configuró una regla de descubrimiento (*discovery rule*) que permitiera buscar entre determinados segmentos de red a dispositivos que dieran una respuesta positiva al OID de sysName(1.3.6.1.2.1.1.5.0) a través de SNMP. Segundo, una vez que el servidor NMS detectara dispositivos que cumplieran la regla creada, se configuró Zabbix para que decidiera que hacer con ellos; esto es, agregarlos a la lista de *host* monitoreados y cargarles la plantilla MikroTik by SNMP.

3.5.5 Triggers y umbrales de alerta

El sistema implementado utilizó disparadores lógicos o *triggers*, incluidos en las plantillas oficiales de Zabbix, para garantizar alta disponibilidad en los servicios. Estos *triggers* son componentes que realizan evaluaciones a partir de los valores obtenidos por las consultas SNMP y de ser el caso, al superar cierto umbral y alcanzar una condición crítica, emiten una alerta. Estas son algunas de las condiciones más comunes que el sistema de monitoreo alerta:

- **Pérdida de *ping*:** Alerta cuando el equipo monitoreado deja de responder a las consultas ICMP de *ping*, no hay gestión.
- **Saturación de recursos de *hardware*:** Por defecto, al excederse el umbral del 90% de uso de CPU o de memoria, el sistema emite alertas. Esos usos elevados pueden estar relacionados con ataques DoS, fallos de *hardware* o simplemente saturación por uso de cliente.
- **Estado de interfaces.** Si una de las interfaces del dispositivo monitoreo cambia de estado; por ejemplo, cambia a Down, el sistema emite una alerta por ella.

Severity	Value	Name	Operational data	Expression	Status	Info	Tags
Warning	OK	ALERTA COMERCIAL: Client e saturando plan contratado d e 50 Mbps		<code>avg(CPE-Mikrotik/net.if.out[ifCOutOctets.11,3m])>45M</code>	Unknown	!	
Warning	OK	CPU discovery: MikroTik: #1: High CPU utilization	Current utilization (ITEM.LASTVALUE1)	<code>min(CPE-Mikrotik/system.cpu.util[hrProcessorLoad.1,5m])>(\$CPU_UTIL_CRIT)</code>	Enabled		scope: performance
Information	OK	Mikrotik by SNMP: MikroTik Device has been replaced		<code>last(CPE-Mikrotik/system.hw.serialnumber.#1)<last(CPE-Mikrotik/system.hw.serialnumber.#2) and length(last(CPE-Mikrotik/system.hw.serialnumber.#2))>0</code>	Unknown	!	scope: notice
Average	OK	Storage discovery: MikroTik Disk-131072: Disk space is critically low	Space used: (ITEM.LASTVALUE3) of (ITEM.LASTVALUE2) ((ITEM.LASTVALUE1))	<code>last(CPE-Mikrotik/vfs.fs.pused[hrStorageSize.131072])>(\$VFS_FS_PUSED_MAX_CRIT"Disk-131072") and ((last(CPE-Mikrotik/vfs.fs.total[hrStorageSize.131072])>last(CPE-Mikrotik/vfs.fs.used[hrStorageSize.131072]))<(\$VFS_FS_FREE_MIN_CRIT"Disk-131072") or timeleft(CPE-Mikrotik/vfs.fs.pused[hrStorageSize.131072],1h,100)<1d)</code>	Unknown	!	scope: availability scope: capacity
Warning	OK	Storage discovery: MikroTik Disk-131072: Disk space is low	Space used: (ITEM.LASTVALUE3) of (ITEM.LASTVALUE2) ((ITEM.LASTVALUE1))	<code>last(CPE-Mikrotik/vfs.fs.pused[hrStorageSize.131072])>(\$VFS_FS_PUSED_MAX_WARN"Disk-131072") and ((last(CPE-Mikrotik/vfs.fs.total[hrStorageSize.131072])>last(CPE-Mikrotik/vfs.fs.used[hrStorageSize.131072]))<(\$VFS_FS_FREE_MIN_WARN"Disk-131072") or timeleft(CPE-Mikrotik/vfs.fs.pused[hrStorageSize.131072],1h,100)<1d)</code>	Unknown	!	scope: availability scope: capacity
Information	OK	Mikrotik by SNMP: MikroTik Firmware has changed	Current value: (ITEM.LASTVALUE1)	<code>last(CPE-Mikrotik/system.hw.firmware.#1)<last(CPE-Mikrotik/system.hw.firmware.#2) and length(last(CPE-Mikrotik/system.hw.firmware.#2))>0</code>	Unknown	!	scope: notice

Figura 12 Panel de *triggers*, umbrales y nivel de severidad de alertas para el equipo MikroTik. Fuente. Elaboración propia a partir de Zabbix

3.5.6 Integración del canal de alertas mediante la API de Telegram

Con el fin de extender la utilidad de la herramienta obtenida, se empleó Telegram como medio de notificación móvil para el área.

Se aprovisionó un bot a través de @BotFather, herramienta oficial de Telegram, a quien se le brindó el nombre del bot a crear, Alertas Monitoreo – Zabbix, y también el usuario que identifica a dicho bot, el cual es “Monitoreo_ISP_bot”. Tras esto, se obtuvo el *token* de acceso al API HTTP de Telegram, el cual es necesario para que Zabbix pueda conectarse a la base de Telegram, autenticarse. Otro elemento que se debe extraer de Telegram es el ID del usuario que recibirá los mensajes de alerta; esto es, el ID, para esta implementación tiene como valor 8340365076 y se obtuvo a través de la herramienta @getmyid_bot de Telegram.

Zabbix trae por defecto la opción de notificación vía Telegram desactivada. Por ello, corresponde luego habilitar dicha opción, en la sección de tipos de medios (*media types*), e insertar el *token* obtenido previamente, como se observa en la Figura 13.

Media type

Media type Message templates 10 Options

Name Telegram

Type Webhook

Name	Value	Action
alert_message	{ALERT.MESSAGE}	Remove
alert_subject	{ALERT.SUBJECT}	Remove
api_chat_id	{ALERT.SENDTO}	Remove
api_parse_mode	<PLACE PARSE MODE>	Remove
api_token	8292509185:AAFAeDVpt_G80KtH	Remove
event_nseverity	{EVENT.NSEVERITY}	Remove
event_severity	{EVENT.SEVERITY}	Remove
event_source	{EVENT.SOURCE}	Remove
event_tags	{EVENT.TAGSJSON}	Remove
event_update_nseverity	{EVENT.UPDATE.NSEVERITY}	Remove
event_update_severity	{EVENT.UPDATE.SEVERITY}	Remove
event_update_status	{EVENT.UPDATE.STATUS}	Remove
event_value	{EVENT.VALUE}	Remove

Add

Script `const CLogger = function(serviceName) {...`

Timeout 30s

Process tags ☒

Figura 13 Vinculación de Zabbix con Telegram vía el token API. Fuente. Elaboración propia.

Finalmente, se procedió a señal en Zabbix a que usuarios se debía notificar o enviar alertas a través de este nuevo método de comunicación. Se eligió el usuario Admin y en medios se agregó Telegram para poder asociarlo con el ID de usuario recibido de @getmyid_bot. Para que el sistema inicie con el envío de alertas fue necesario habilitar la regla *Report problems to Zabbix administrators* dentro del menú de acciones de los disparadores.

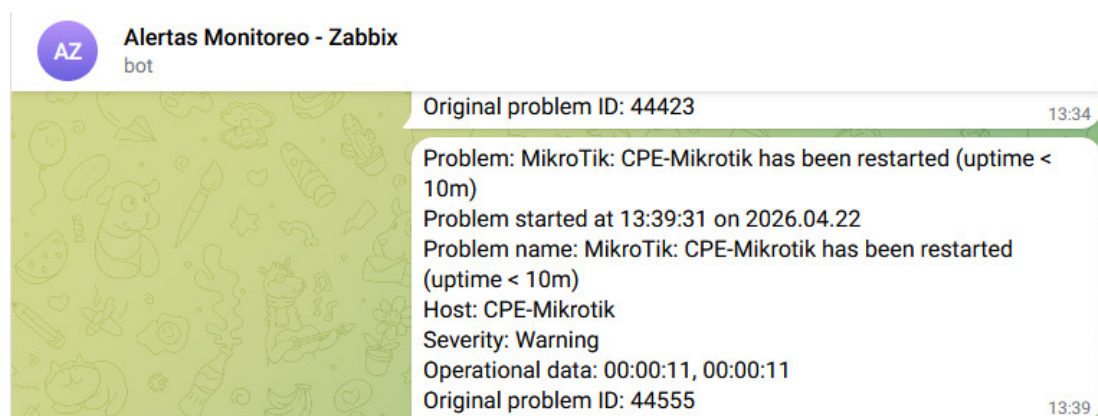


Figura 14 Telegram notificando alertas emitidas por el servidor Zabbix. Fuente. Elaboración propia

3.5.7. Simulación de fallos y evaluación de alertas

Para verificar el funcionamiento y eficacia del sistema NMS implementado, se diseñó una serie de pruebas de estrés. Se quiso comprobar la operación correcta de los umbrales o *thresholds* establecidos en los *triggers* de Zabbix, que estos disparen alertas cuando se llega a cierto nivel crítico. Se presentó 3 casuísticas o escenarios:

3.5.7.1 Escenario 1: Caída lógica del servicio (Disponibilidad)

Este escenario permitió mostrar cómo responde el sistema ante la caída del servicio de cliente, sin perder la gestión remota.

- **Metodología:** Se procedió a deshabilitar de forma administrativa la subinterfaz de servicio asociada a la VLAN 100 con ayuda del comando shutdown para simular un fallo en la red de acceso. Se aplicó esta prueba al equipo CPE-Cisco.
- **Objetivo:** Verificar la efectividad de la implementación con segmentación OOB. Como solo se apagó la subinterfaz de servicio, Zabbix pudo continuar con el monitoreo a través de SNMP en la red fuera de banda, VLAN 98. Con la emisión de la alerta correspondiente en el *dashboard* y en Telegram, se pudo

tener un diagnóstico preciso del estado del equipo y no un falso positivo que indicara pérdida por rotura del enlace de fibra óptica.

3.5.7.2 Escenario 2: Saturación de Ancho de Banda (Capacidad)

Esta segunda prueba permitió evaluar cómo responde el sistema de monitoreo frente al consumo elevado por parte del cliente, permitió mostrar cuan efectivas son las reglas de tráfico o políticas de QoS (calidad de servicio) que se aplican a los distintos equipos CPE de la empresa.

- **Metodología:** Se configuró una política de *Simple Queue* en el CPE MikroTik para que limitara el tráfico a 50Mbps tanto en *upstream* como en *downstream*. Asimismo, se utilizó la herramienta de inyección de tráfico IPerf3 en modo cliente en un dispositivo final ubicado en la LAN 192.168.200.0/24, se hizo la configuración necesaria en el cliente iperf para que este tráfico sea enviado hacia un servidor de pruebas exclusivo fuera de la red de cliente. Se configuró en Zabbix un *trigger* exclusivo llamado “ALERTA COMERCIAL: Cliente saturando plan contratado de 50 Mbps” que emita las notificaciones siempre que el promedio de consumo de tráfico se acercará al valor establecido como plan.
- **Objetivo:** Verificar la sensibilidad del *trigger* configurado cuando hay picos de consumo. Comprobar que toda vez que se cumpla la expresión lógica “ifHoutOctets > 45 Mbps por 3 min” el estado cambie a PROBLEM, emitiendo la alerta con el nombre ya mencionado y manteniendo la gestión del equipo por el monitoreo fuera de banda del equipo CPE.

3.5.7.3 Escenario 3: Ataque Dos desde la LAN (Seguridad)

El tercer escenario simuló la existencia de una vulnerabilidad en la red LAN de cliente, se asume la existencia de un *host* infectado el cual intenta realizar un ataque de DoS (denegación de servicio) enviando paquetes maliciosos al equipo CPE MikroTik buscando saturar sus recursos.

- **Metodología:** Para esta prueba se usó una vez más Iperf3, a quien se le configuró para envíe tráfico simultaneo a través de múltiples hilos usando paquetes del tamaño mínimo, 64 bytes. Con esta herramienta se busca realizar

un ataque *UDP Flood* o de inundación, saturando el número de paquetes por segundo que el *router* puede procesar.

- **Objetivo:** Demostrar que el *router* puede llegar saturar sus recursos de *hardware* cuando hay un exceso de procesamiento de cabeceras y de seguimiento de conexiones, el uso de CPU puede llegar a valores anormales o al 100% inclusive. El sistema de monitoreo Zabbix debe saber leer, a través del OID correspondiente, el uso de este recurso y debe emitir una alerta crítica de “*High CPU Utilization*” cuando se detecte alguna anomalía o un uso intensivo que supere el umbral establecido en el *template*.

3.6 Resultados de la actividad

En esta sección se muestra los resultados obtenidos a partir de la ejecución de cada uno de los escenarios propuestos en la etapa anterior. Las evidencias demuestran la capacidad de respuesta del sistema de monitoreo Zabbix ante fallos lógicos, estrés de recursos y saturación de ancho de banda en los equipos CPE de los clientes.

3.6.1. Escenario 1: Simulación de caída de enlace (CPE Cisco)

Este escenario permitió comprobar como Zabbix detecta, de forma casi instantánea, una caída o interrupción lógica del servicio, sin perder gestión del CPE, en este caso, del equipo Cisco.

```
CPE1CISCO#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
CPE1CISCO(config)#interface g0/0.100
CPE1CISCO(config-subif)#shutdown
```

Figura 15 Apagado administrativo de la subinterfaz de servicio G0/0.100.

Fuente. Elaboración propia a partir de CLI de Cisco IOS.

Posteriormente, se analizó el tráfico usando la herramienta Wireshark para verificar cómo fluye el intercambio de mensajes SNMP entre el servidor Zabbix y el equipo CPE, tal como se detalla en la Figura 16.

Finalmente, tras sucedido el evento, Zabbix realizó la notificación respectiva en su *dashboard* web y a través de Telegram (ver Figura 17).

*Standard input [Switch2 Ethernet1 to CPE-CISCO GigabitEthernet0/0]

Archivo Edición Visualización Ir Captura Analizar Estadísticas Telefonía Wireless Herramientas Ayuda

snmp

No.	Time	Source	Destination	Protocol	Length	Info
222	65.855955	192.168.126.10	10.99.1.2	SNMP	93	getBulkRequest 1.3.6.1.2.1.2.2.1.7.10
223	65.864787	10.99.1.2	192.168.126.10	SNMP	250	get-response 1.3.6.1.2.1.2.2.1.7.11 1.3.6.1.2.1.2.2.1.8.11 1.3.6.1.2.1.2.2.1.8.12
224	65.866686	192.168.126.10	10.99.1.2	SNMP	92	getBulkRequest 1.3.6.1.2.1.2.2.1.8
225	65.875254	10.99.1.2	192.168.126.10	SNMP	250	get-response 1.3.6.1.2.1.2.2.1.8.1 1.3.6.1.2.1.2.2.1.8.10
226	65.876691	192.168.126.10	10.99.1.2	SNMP	93	getBulkRequest 1.3.6.1.2.1.2.2.1.8.10
227	65.885966	10.99.1.2	192.168.126.10	SNMP	258	get-response 1.3.6.1.2.1.2.2.1.8.11 1.3.6.1.2.1.2.2.1.8.12
228	65.887949	192.168.126.10	10.99.1.2	SNMP	93	getBulkRequest 1.3.6.1.2.1.31.1.1.1.1

> Frame 227: Packet, 258 bytes on wire (2064 bits), 258 bytes captured (2064 bits) on interface -, id 0

> Ethernet II, Src: ca:01:08:73:00:08 (ca:01:08:73:00:08), Dst: 0c:ae:7f:e2:00:02 (0c:ae:7f:e2:00:02)

> 802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 99

> Internet Protocol Version 4, Src: 10.99.1.2, Dst: 192.168.126.10

> User Datagram Protocol, Src Port: 161, Dst Port: 52890

> Simple Network Management Protocol

version: v2c (1)

community: snmp_pro

> data: get-response (2)

> get-response

request-id: 1535699859

error-status: noError (0)

error-index: 0

> variable-bindings: 10 items

> 1.3.6.1.2.1.2.2.1.8.11: 2

Object Name: 1.3.6.1.2.1.2.2.1.8.11 (iso.3.6.1.2.1.2.2.1.8.11)

Value (Integer32): 2

> 1.3.6.1.2.1.2.2.1.8.12: 1

Object Name: 1.3.6.1.2.1.2.2.1.8.12 (iso.3.6.1.2.1.2.2.1.8.12)

Value (Integer32): 1

Figura 16 Contenido del PDU SNMP Get-Response donde se observa el OID 1.3.6.1.2.1.2.2.1.8.11 con un valor de “2”, lo que indica que la subinterfaz G0/0.100 sí está apagada. Fuente. Elaboración propia a partir de Wireshark.

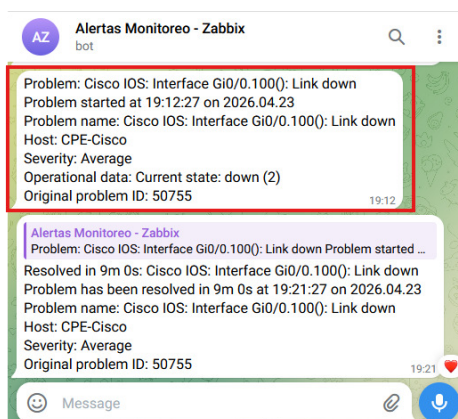


Figura 17 Notificación recibida en Telegram donde se observa el *host* afectado y el estado operacional actual. Fuente. Elaboración propia a partir de Telegram

3.6.2. Escenario 2: Saturación de ancho banda (CPE MikroTik)

Acorde a la metodología presentada en la sección 3.5, para este escenario se tuvo como objetivo validar cómo responde el sistema implementado Zabbix ante consumos de tráfico elevados en las redes de los clientes. Se verificó también la efectividad de las políticas de QoS configuradas y la sensibilidad de los *triggers*.

Se ejecutó la herramienta iperf3 en un *host* ubicado en la red 192.168.200.0/24 (LAN de cliente) para inyectar tráfico continuo hacia un servidor de pruebas ubicado en otra

red. Se validó con Zabbix cómo la regla de *simple queue* limitó la conexión de cliente a la capacidad contratada

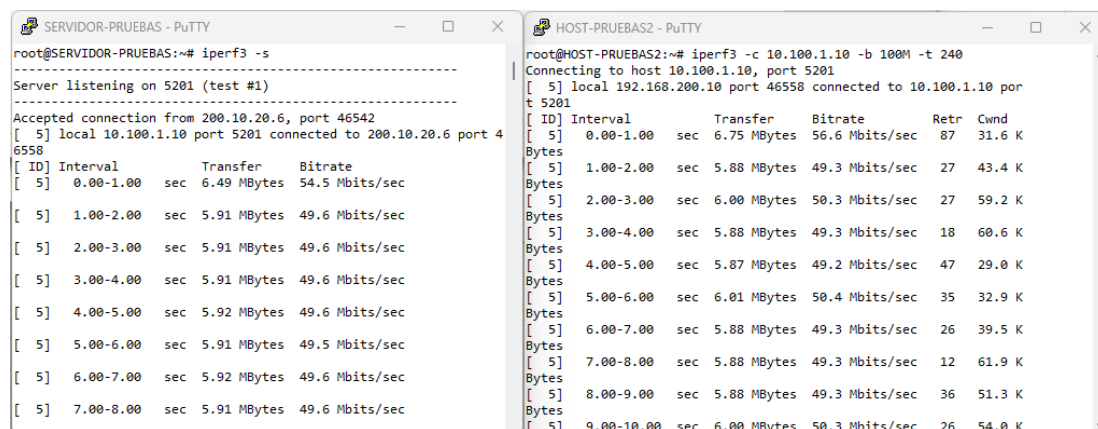


Figura 18 Generación de tráfico en red LAN de cliente usando Iperf3

Fuente. Elaboración propia a partir de la herramienta Putty.

La plataforma Zabbix recopiló datos de tráfico de la subinterfaz de servicio del CPE, con ayuda de las variables *ifOutOctets*. El grafico siguiente muestra como el consumo de cliente alcanzó el límite establecido por la política de *simple queue* y cómo se superó el umbral establecido en el *trigger* del sistema NMS.

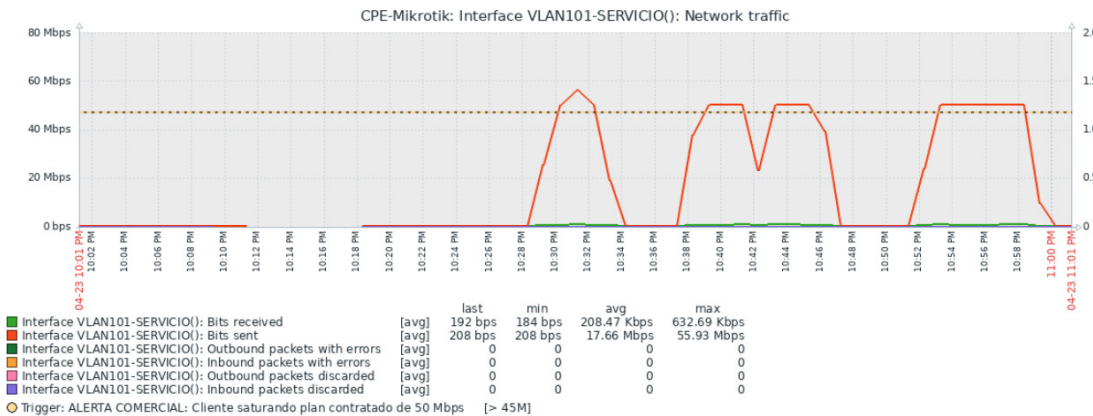


Figura 19 Gráfica de Zabbix mostrando cómo las políticas de tráfico hacen efecto en la capacidad de la red de cliente y cómo se alcanza el umbral de 45 Mbps. Fuente. Elaboración propia a partir de Zabbix.

El sistema Zabbix fue configurado para que siempre que se satisfaga la expresión $ifOutOctets > 45$ por un tiempo mínimo de 3 minutos, el *trigger* se dispare mostrando en el *dashboard* la alerta correspondiente a saturación del plan contratado. Se verificó la sensibilidad del *trigger* configurado, así como también se observó cómo ese enorme consumo no afectaba la gestión remota del CPE, lo que confirmó la eficacia de tener gestión OOB

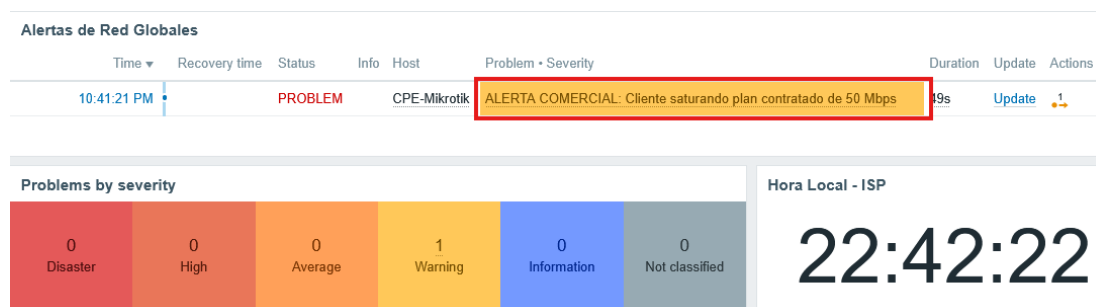


Figura 20 Dashboard de Zabbix alerta de saturación. Fuente. Elaboración propia a partir de Zabbix.

3.6.3. Escenario 3: Ataque DoS desde la red LAN (Seguridad)

El tercer escenario de pruebas planteado permitió mostrar la capacidad del sistema para detectar comportamientos anómalos en la red de cliente, que podrían estar asociados a vulnerabilidades de seguridad, para este trabajo de suficiencia profesional, se consideró un ataque de DoS ocurrida desde una LAN.

Según la metodología propuesta en la sección 3.5.6.3, para este escenario, se asumió la existencia de *host* infectado en la red local de cliente, este dispositivo inició un ataque de tipo *UDP Flood*. Para simular el ataque, se utilizó nuevamente la herramienta *Iperf3* a la cual se le configuró para establecer conexiones hacia un destino a través de múltiples hilos (exactamente 40) utilizando paquetes de tamaño mínimo (PDUs de 64bytes)

Con esta técnica se buscó estresar el *hardware* del equipo CPE MikroTik, proporcionándole un numero elevados de cabeceras por segundo por procesar y saturando su tabla de seguimiento de conexiones, *conntrack*.

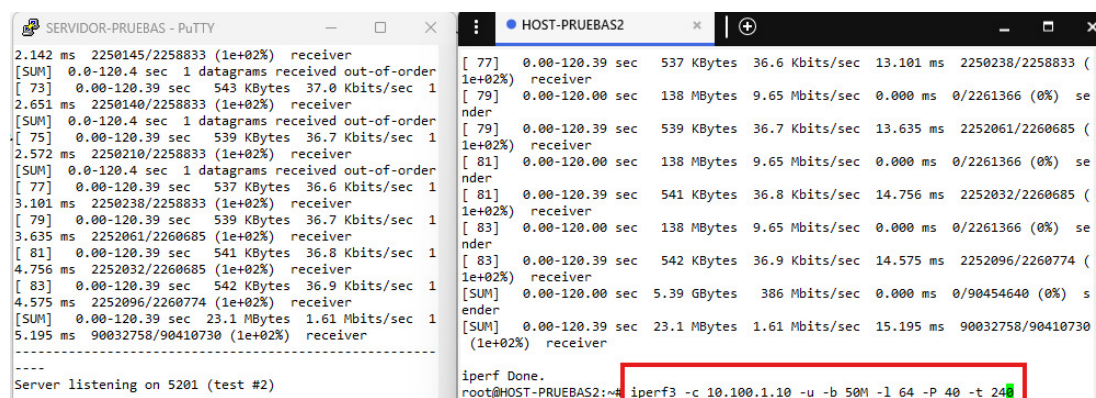


Figura 21 Simulación del ataque UDP Flood con Iperf3 desde un host mostrando múltiples hilos de conexión y paquetes de tamaño mínimo. Fuente. Elaboración propia a partir de la herramienta Putty.

Gracias a las consultas SNMP realizadas por el servidor Zabbix al OID de carga del procesador en el *router*, el sistema logró graficar cómo afectó el ataque de inundación de paquetes en los recursos del equipo. En la siguiente grafica se observa cómo, tras iniciado el ataque, el uso del CPU se dispara notoriamente, sobrepasando el umbral del *trigger* respectivo, que por defecto alerta cuando se tiene un uso mayor al 40%.

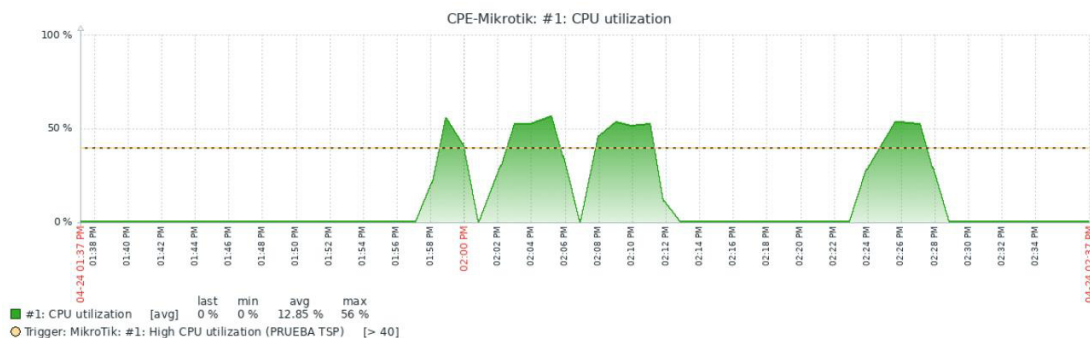


Figura 22 Gráfica de Zabbix evidenciando picos anómalos en el uso de CPU, tras iniciados los ataques UDP Flood. Fuente. Elaboración propia a partir de Zabbix.

Una vez detectado el incremento inusual y siempre que se satisfaga condición del *trigger*, el sistema emitió la alerta correspondiente, con esto se puede decir que la plataforma es capaz de notificar sobre la ocurrencia de eventos anómalos, que podrían estar relacionados con incidentes de seguridad. Al tener establecido el umbral del 40%, el servidor NMS permite a los operadores de turno poder tomar acciones proactivas antes de que ocurra una caída total del servicio de cliente.

Panel de Control - ISP

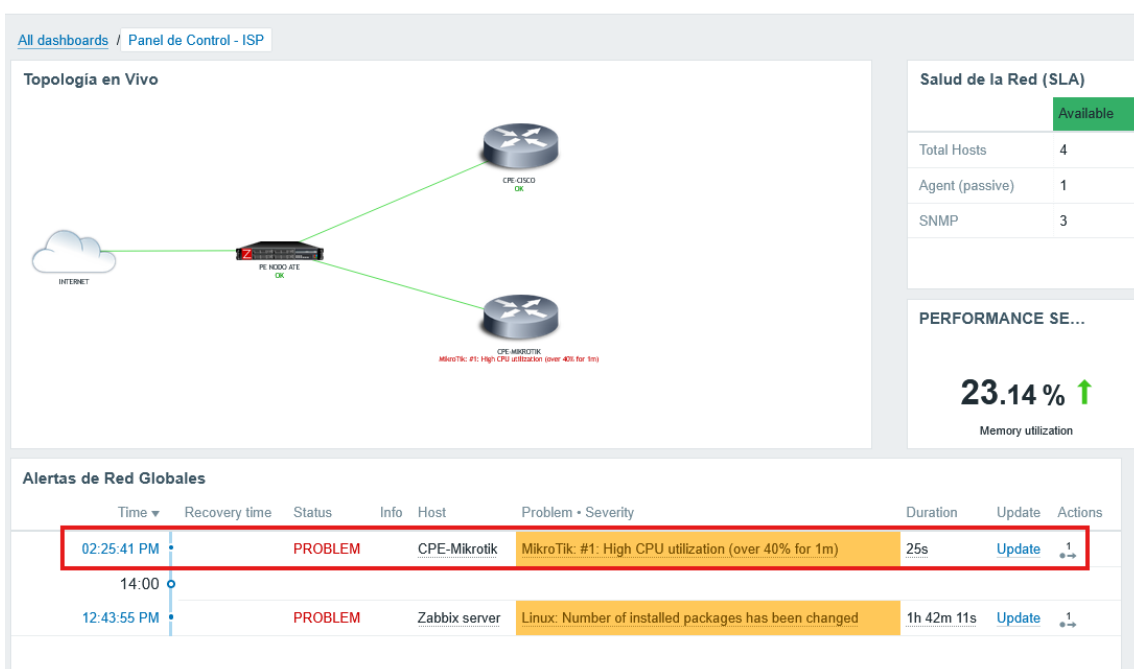


Figura 23 Dashboard de Zabbix que muestra la alerta "High CPU utilization" en estado activo. Fuente. Elaboración propia a partir de Zabbix

CAPITULO IV

DISCUSION DE RESULTADOS

4.1 Descripción de la implementación

La implementación del sistema de monitoreo se basó en una arquitectura cliente-servidor donde los equipos monitoreados, equipos de premisa a cliente o CPE, son dispositivos descentralizados y se accede a ellos remotamente para la recolección de datos a través del protocolo SNMP. Aunque la solución presentada en el actual trabajo de suficiencia profesional abarca desde la implementación del motor de base de datos hasta las notificaciones móviles, la presente discusión técnica se centra en la estandarización de la capa de red, las políticas de seguridad implementadas y en cómo se proyecta la escalabilidad técnica del sistema NMS.

4.1.1 Arquitectura de cómputo del servidor Zabbix

El corazón del sistema de monitoreo fue implementado siguiendo el modelo de virtualización de servidores; es decir, se alojó en una máquina virtual, se usó VMware como hipervisor. Se empleó esta configuración porque es más flexible al permitir tener control total sobre los recursos del sistema y al facilitar la escalabilidad. Asimismo, se utilizó para el servidor una distribución de Linux sin entorno gráfico, Ubuntu Server 24.04 LTS, lo que facilitó orientar el total de las capacidades del sistema hacia las consultadas SNMP y al manejo de la base de datos relacional, MariaDB.

La asignación de recursos para la máquina virtual que aloja Zabbix se realizó siguiendo criterios de carga de trabajo. Hay que tener en cuenta que el sistema está diseñado para supervisar clientes corporativos, clientes *top*, por lo que es fundamental realizar un dimensionamiento correcto a fin de garantizar la integridad de la base de datos y funcionamiento fluido del motor del servidor Zabbix. Los recursos asignados se detallan en el Cuadro 2.

Cuadro 2 Especificaciones técnicas de la VM Zabbix

Recurso	Capacidad	Justificación
Procesador (vCPU)	2 Cores	Para manejar procesos paralelos de Zabbix , recolección y recepción de datos.
Memoria RAM	4 GB	Permite mantener el caché de Zabbix en memoria y agilizar las consultas del motor <i>MariaDB</i>
Almacenamiento		
Interno	40 GB	Adecuado para mantener el histórico de datos(history) y las tendencias (trends) de las métricas de red.
Interfaz de Red	VMXNET3 (1 Gbps)	Interfaz virtual que garantiza el ancho de banda necesario para todas las consultas/respuestas SNMP

Fuente. Elaboración propia en base a los requerimientos de Zabbix 7.0

4.1.2 Escalabilidad del sistema.

En esta sección se realiza un análisis de la escalabilidad del sistema implementado, se quiere saber si según los recursos de *hardware* asignados el producto será capaz de soportar el incremento de la cartera de clientes en los próximos años. Para este fin, se consideró 2 variables:

4.1.2.1 Proyección de equipos a monitorear.

Teniendo en cuenta los registros de la cantidad de clientes *top* desde hace 3 años, podemos calcular la tasa de variación anual:

$$Tasa\ de\ crecimiento = \frac{Equipos\ Actuales - Equipos\ Anteriores}{Equipos\ Anteriores} \times 100\%$$

Considerando que se llegó a obtener 65 equipos CPE desde los 45 que habían originalmente, la tasa promedio anual de crecimiento ronda los 14.8%, con lo que se estima que en 3 años el ISP contará con alrededor 100 dispositivos en total para monitorear.

4.1.2.2 Procesamiento en NVPS

Según Zabbix LLC, la carga del sistema no recae exactamente en el número de equipos a monitorear, sino en los nuevos valores por segundo (NVPS) que el motor

de recolección de Zabbix debe procesar y escribir en su correspondiente base de datos.

Teniendo en cuenta el número de clientes proyectados para los próximos 3 años y considerando que se monitorizan 50 *items* por equipo con una frecuencia de actualización de 60 segundos, se puede calcular el valor de NVPS que se espera tener para ese tiempo:

$$NVPS = \frac{100 \text{ Equipos} \times 50 \text{ items}}{60 \text{ segundos}} \approx 83.3 \text{ valores por segundo}$$

Con el valor obtenido, se calcula la capacidad de almacenamiento en disco duro que el sistema requerirá. Según el desarrollador, el tamaño de la base de datos, para este trabajo MariaDB, depende de 3 variables: el almacenamiento del historial, de tendencias y de los eventos en la red. Usando el número de NVPS, con las fórmulas oficiales se puede encontrar el valor de cada una de dichas variables:

Almacenamiento de historial.

El sistema está configurado para mantener un historial por 30 días. Asumiendo que cada *ítem* ocupa aproximadamente 90 Bytes en MariaDB, se calcula:

$$\begin{aligned} \text{Almacenamiento de historial} &= 30 \text{ días} \times \frac{5000 \text{ items}}{60} \times 24 \times 3600 \times 90 \text{ bytes} \\ &\approx 19.4 \text{ GB} \end{aligned}$$

Almacenamiento de tendencias

Adicional al almacenamiento de historial de cada valor puntual de los *ítems*, Zabbix obtiene un promedio de esos datos por hora y los retiene por un periodo especificado, para este trabajo, se proyectó retener la información por 365 días. Se calculó el almacenamiento que ocupan las tendencias de la siguiente manera:

$$\begin{aligned} \text{Almacenamiento de tendencias} &= 365 \text{ días} \times \frac{5000}{3600} \times 24 \times 3600 \times 90 \text{ bytes} \\ &\approx 3.6 \text{ GB} \end{aligned}$$

Almacenamiento de eventos

Según Zabbix SIA, el espacio que ocupa un evento en almacenamiento es de alrededor 330 Bytes. Asumiendo el peor de los casos, cuando se registra 1 evento por segundo y dada la configuración del sistema para retener esta información por 30 días, se calcula:

$$\begin{aligned} \text{Almacenamiento de eventos} &= 30 \text{ días} \times 1 \text{ evento} \times 24 \times 3600 \times 330 \text{ bytes} \\ &\approx 0.85 \text{ GB} \end{aligned}$$

Si se realiza la suma de los datos obtenidos, de ocupación en disco al valor que ocupa la configuración de Zabbix (el cual suele ser de 10MB o menos según información oficial del fabricante), se obtiene que para un año de operación a capacidad máxima proyectada el uso de disco alcanza los 24.15GB, lo que quiere decir que se tendría una reserva de seguridad de alrededor de 40%. De esta manera, se garantiza que el sistema no tendrá caídas por falta de espacio en disco aun cuando se llegue al número de clientes proyectados, el aprovisionamiento de 40GB como almacenamiento interno del sistema es más que suficiente. Asimismo, el desarrollador indica que para instalaciones pequeñas (aquellas que monitorean no más 1000 métricas) el sistema requiere por lo menos de 2 vCores y de 8GB de RAM. Para la etapa de simulación del presente trabajo se asignó 4GB de RAM al servidor, lo cual fue suficiente para poder monitorear los equipos CPE de prueba sin problemas. Sin embargo, para el pase a producción y para poder satisfacer la demanda de los 100 equipos proyectados, será necesario cumplir con las recomendaciones técnicas del fabricante, ver Cuadro 3, y aumentar la capacidad de memoria RAM a 8GB, los cuales serán suficientes para poder acelerar el procesamiento y evitar cuellos de botella.

Cuadro 3 Requerimientos de hardware según el tamaño de la instalación en Zabbix

Tamaño de la instalación	Métricas monitorizadas	Núcleos CPU/vC PU	Memoria (GiB)	Base de datos soportada	Instancia de referencia (Amazon EC2)
Pequeña	1,000	2	8 (≈ 8.6 GB)	MySQL, MariaDB, PostgreSQL	Percona, m6i.large, m6g.large
Mediana	10,000	4	16 (≈ 17.2 GB)	MySQL, MariaDB, PostgreSQL	Percona, m6i.xlarge, m6g.xlarge

Grande	100,000	16	64 (≈ 68.7 GB)	MySQL, MariaDB, PostgreSQL, Oracle	Percona,	m6i.4xlarge m6g.4xlarge	/
Muy grande	1,000,000	32	96 (≈ 103 GB)	MySQL, MariaDB, PostgreSQL, Oracle	Percona,	m6i.8xlarge m6g.8xlarge	/

Fuente: Adaptado de Zabbix SIA (s. f.).

4.1.3 Diseño de seguridad.

Uno de los problemas al cual se enfrentó la implementación era el de lograr extraer los datos de forma segura desde los equipos de borde al servidor Zabbix. Como se sabe, SNMPv2c utiliza cadenas de comunidad que viajan en texto plano, por lo cual existía un riesgo inherente de interceptación.

Para enfrentar esta vulnerabilidad, se incorporó en el diseño una medida de seguridad y se aprovechó una característica ya existente de la red de los clientes. Los elementos fundamentales para el diseño de seguridad son:

Listas de control de acceso (ACL)

Se implementó reglas de ACL para que restrinjan el acceso a información SNMP relevante de los equipos CPE. Los equipos CPE están configurados para descartar consultas UDP que provengan por medio del puerto 161 que no sean de la IP del servidor Zabbix (192.168.126.10)

Segmentación fuera de banda (OOB)

Se aprovechó la existencia de VLANs gestión en la red, en este trabajo de suficiencia profesional se muestra las VLAN 98 y 99. Esta característica permitió crear reglas de enrutamiento en los equipos de la red para poder pasar el tráfico SNMP por VLANs dedicadas. De esta manera, el tráfico de monitoreo viaja por un canal virtual exclusivo, sin mezclarse con el tráfico de datos de cliente.

4.2 Evaluación Económica

Para la ejecución o puesta en marcha de la implementación no se requirió de inversión adicional por parte de la empresa, debido a que el despliegue del sistema se realizó usando la infraestructura de virtualización existente en el área técnica y, además, se

usó Zabbix, el cual es un software libre de monitoreo. Al implementar el proyecto, desde un punto de vista financiero, el beneficio directo de la empresa es de ahorro operativo, ya que se evita incurrir en gastos de licenciamiento de software propietarios de gestión de red. De igual forma, el beneficio indirecto sería la reducción de penalidades hacia el operador por incumplimiento de SLA, debido a que la detección temprana de averías o alertas permite reducir los tiempos de inactividad de los servicios proporcionados a los clientes.

CAPITULO V

CONCLUSIONES

Se logró implementar un sistema de monitoreo integral basado en Zabbix 7.0 y en el protocolo SNMPv2c. La puesta en marcha de este proyecto permitió monitorear de manera proactiva los servicios proporcionados a los clientes *top* de la empresa, brindó mayor visibilidad sobre los equipos CPE instalados.

Se identificó parámetros críticos de *hardware*, como uso de CPU, cantidad de memoria RAM y de estado de interfaces que sirvieron para el diagnóstico de fallas. El monitoreo de esas variables permitió analizar con mayor precisión las causas de incidencias en los servicios de clientes, las cuales pueden ir desde saturación de *hardware*, caídas lógicas de servicios hasta ataques externos.

Se logró implementar políticas de seguridad basadas en listas de control de acceso (ACL) y en el uso de cadenas de comunidad SNMP para una extracción segura y exclusiva de datos. Estas medidas, sumadas a la segmentación fuera de banda, protegen a los equipos frente al acceso no autorizado.

Se configuró la interfaz web de Zabbix para que permita visualizar fácilmente la ocurrencia de incidencias. Asimismo, la integración con Telegram facilitó el envío de notificaciones automáticas en tiempo real, lo que ayudó a la empresa a mejorar el cumplimiento de los SLA

Se verificó la operatividad del sistema de monitoreo a través de pruebas saturación de ancho de banda, caída de enlace y ataques UDP Flood. Se validó que las gráficas generadas eran acorde a las incidencias ocurridas. Con el análisis de escalabilidad, se confirmó que el sistema puede soportar el crecimiento proyectado por la empresa para los próximos años, siempre que se siga las recomendaciones del fabricante

CAPITULO VI

RECOMENDACIONES

Se recomienda a futuro realizar la migración a SNMPv3, el cual ofrece autenticación, cifrado e integridad. De esta manera, se eliminaría la vulnerabilidad que se tiene con versiones anteriores, donde los datos de monitoreo pueden ser capturados y leídos por personal no autorizado con herramientas como Wireshark, ya que viajan en texto plano por la red.

Se sugiere configurar un *cluster* de alta disponibilidad para el servidor Zabbix. Con esta acción, se buscaría tener un sistema de respaldo que permita darle continuidad al monitoreo aun cuando se tenga fallas críticas de *hardware*, en el hipervisor o en el mismo sistema operativo.

Se recomienda el uso de *proxies* de Zabbix para eventuales expansiones de la red a otras regiones. Con esta arquitectura, se optimizaría el uso de ancho de banda en los enlaces troncales, se tendría menos carga de trabajo en el servidor principal, se lograría una menor latencia y respuestas más rápidas, ya que la recolección de datos sería local en los puntos remotos.

Zabbix es capaz de recopilar datos de todo equipo que exponga información por SNMP. Se puede aprovechar esta característica para implementar sensores que recolecten datos de temperatura y humedad en los nodos del ISP (donde se ubican los equipos críticos, *provider edge*, PE) para prevenir fallas de *hardware* por condiciones térmicas inadecuadas en equipos fundamentales para la prestación de servicios.

CAPITULO VII

REFERENCIAS BIBLIOGRÁFICAS

- Case, J., McCloghrie, K., Rose, M., & Waldbusser, S. (1996). *Introduction to community-based SNMPv2 (RFC 1901)*. Internet Engineering Task Force. <https://doi.org/10.17487/RFC1901>
- Cisco Systems. (s. f.). *Cisco MIB Object Navigator*. <https://oidref.com/1.3.6.1.4.1.9>
- Huawei Technologies Co., Ltd. (2024). *MIB tree structure (Document ID EDOC1100035587)*. Huawei Support. <https://support.huawei.com/enterprise/en/doc/EDOC1100035587/21b78d21/mib-tree-structure>
- Internet Engineering Task Force. (1991). *Management Information Base for Network Management of TCP/IP-based internets: MIB-II (RFC 1213)*. <https://www.ietf.org/rfc/rfc1213.txt>
- Internet Engineering Task Force. (2000). *Host Resources MIB (RFC 2790)*. <https://www.ietf.org/rfc/rfc2790.txt>
- Internet Engineering Task Force. (2000). *Interfaces Group MIB (IF-MIB) (RFC 2863)*. <https://www.ietf.org/rfc/rfc2863.txt>
- ManageEngine. (s. f.). *What is SNMP? Monitoring, architecture, and how it works*. Recuperado el 9 de enero de 2026, de <https://www.manageengine.com/network-monitoring/what-is-snmp.html>
- McCloghrie, K., & Rose, M. (1991). *RFC 1213: Management Information Base for Network Management of TCP/IP-based internets: MIB-II*. Internet Engineering Task Force. <https://datatracker.ietf.org/doc/html/rfc1213>
- MikroTik. (s. f.). *MikroTik SNMP MIB*. <https://mikrotik.com>
- MikroTik. (s. f.). *Queues*. <https://help.mikrotik.com/docs/spaces/ROS/pages/328088/Queues>
- OIDRef. (s. f.). *SNMP OID database*. <https://oidref.com>
- Presuhn, R., et al. (2002). *Management Information Base (MIB) for the Simple Network Management Protocol (SNMP) (RFC 3418)*. Internet Engineering Task Force.

Stallings, W. (2014). *Data and computer communications* (10.^a ed.). Pearson.

Zabbix SIA. (s. f.). *About*. En *Zabbix documentation*.

<https://www.zabbix.com/documentation/current/en/manual/introduction/about>

Zabbix SIA. (s. f.). *Concepts*. En *Zabbix documentation*.

<https://www.zabbix.com/documentation/current/en/manual/concepts>

Zabbix SIA. (s. f.). *Features*. En *Zabbix documentation*.

<https://www.zabbix.com/documentation/current/en/manual/introduction/features>

Zabbix SIA. (s. f.). *Triggers*. En *Zabbix documentation*.

<https://www.zabbix.com/documentation/current/en/manual/config/triggers>

Zabbix SIA. (s. f.). *User macros*. En *Zabbix documentation*.

https://www.zabbix.com/documentation/current/en/manual/config/macros/user_macros

Zabbix SIA. (s. f.). *What's new*. En *Zabbix documentation*.

<https://www.zabbix.com/documentation/current/en/manual/introduction/whatsnew>

CAPITULO VIII

ANEXOS

Anexo A: Configuración de red y SNMP del CPE Cisco

```
CPE1CISCO# show running-config
Building configuration...
```

```
Current configuration : 1854 bytes
!
version 15.2
service timestamps debug datetime msec
service timestamps log datetime msec
no service password-encryption
!
hostname CPE1CISCO
!
no aaa new-model
no ip icmp rate-limit unreachable
no ip domain lookup
ip cef
no ipv6 cef
!
ip tcp synwait-time 5
!
policy-map LIMITE_50M_POLICY
 class class-default
   police 50000000 conform-action transmit exceed-action drop
!
interface GigabitEthernet0/0
 no ip address
 duplex full
 speed 1000
 media-type gbic
 negotiation auto
!
interface GigabitEthernet0/0.99
 description VLAN_GESTION_OOB_ZABBIX
 encapsulation dot1Q 99
 ip address 10.99.1.2 255.255.255.0
!
interface GigabitEthernet0/0.100
 description VLAN_DATOS_WAN
 encapsulation dot1Q 100
 ip address 200.10.20.2 255.255.255.252
 ip nat outside
```

```

ip virtual-reassembly in
service-policy input LIMITE_50M_POLICY
service-policy output LIMITE_50M_POLICY
!
interface GigabitEthernet1/0
description LAN_CLIENTE_CORPORATIVO
ip address 192.168.100.1 255.255.255.0
ip nat inside
ip virtual-reassembly in
negotiation auto
!
ip forward-protocol nd
no ip http server
no ip http secure-server
!
ip nat source list 1 interface GigabitEthernet0/0.100 overload
ip route 0.0.0.0 0.0.0.0 200.10.20.1
ip route 192.168.126.10 255.255.255.255 10.99.1.1
!
access-list 1 permit 192.168.100.0 0.0.0.255
access-list 10 permit 192.168.126.10
!
snmp-server community snmp_pro RO 10
snmp-server location Nodo-Ate-CPE-Corporativo
snmp-server enable traps entity-sensor threshold
!
line con 0
exec-timeout 0 0
privilege level 15
logging synchronous
line vty 0 4
login
transport input all
!
End

```

Anexo B: Configuración de red y SNMP del CPE MikroTik

```

[admin@CPE-MikroTik] > export
# Configuración exportada para Validación de Proyecto NMS
# system id = 1bU2rCBvUDG
#
/interface ethernet
set [ find default-name=ether1 ] disable-running-check=no
set [ find default-name=ether2 ] disable-running-check=no
set [ find default-name=ether3 ] disable-running-check=no
set [ find default-name=ether4 ] disable-running-check=no
set [ find default-name=ether5 ] disable-running-check=no
/interface vlan
add interface=ether1 name=VLAN98-GESTION vlan-id=98
add interface=ether1 name=VLAN101-SERVICIO vlan-id=101
/queue simple
add dst=VLAN101-SERVICIO max-limit=50M/50M name=Internet-Dedicado-50M target=192.168.200.0/24
/snmp community

```

```

add addresses=192.168.126.10/32 name=snmp_pro
/ip address
add address=10.98.1.2/24 interface=VLAN98-GESTION network=10.98.1.0
add address=200.10.20.6/30 interface=VLAN101-SERVICIO
network=200.10.20.4
add address=192.168.200.1/24 interface=ether2 network=192.168.200.0
/ip dns
set allow-remote-requests=yes servers=8.8.8.8,8.8.4.4
/ip firewall nat
add action=masquerade chain=srcnat out-interface=VLAN101-SERVICIO
/ip route
add dst-address=0.0.0.0/0 gateway=200.10.20.5
add dst-address=192.168.126.0/24 gateway=10.98.1.1
add dst-address=10.100.1.0/24 gateway=200.10.20.5
/snmpp
set contact=Area_Monitoreo enabled=yes location=Ate-Sede-CPE
/system identity
set name=CPE-MikroTik

```

Anexo C: Extracto de script de integración de Zabbix con Telegram

Se presenta el bloque principal de código *JavaScript* nativo de Zabbix que permite ver cómo se conecta con la API de Telegram.

```

Telegram.prototype.onEvent = function (alert) {
    Logger.log(Logger.INFO, 'Source: ' + alert.source + '; Event: '
+ alert.event);
    Logger.log(Logger.INFO, 'URL: ' +
this.params.url.replace(this.params.api_token, '<TOKEN>'));
    var response = this.request.jsonRequest('POST', this.params.url
+ this.params.api_token + '/sendMessage', this.data);

    if (this.request.getStatus() !== 200 ||
!CParamValidator.isType(response.ok, 'boolean') || response.ok !==
true) {
        Logger.log(Logger.INFO, 'HTTP code: ' +
this.request.getStatus());
        if (CParamValidator.isType(response.description,
'string')) {
            throw response.description;
        }
        else {
            throw 'Unknown error. Check debug log for more
information.';
        }
    }
    if (CParamValidator.isDefined(response.result.message_id) &&
this.getMessageID(this.data['chat_id'],
this.data['message_thread_id']) === null) {
        this.result.tags['__telegram_msg_id_' +
this.data['chat_id'] + (this.data['message_thread_id'] ? '_' +
this.data['message_thread_id'] : '')] = response.result.message_id;
    }
    return this.result;
}

```

};

Anexo D: Definición de umbrales críticos (*Triggers*)

Nota: Los umbrales de los *triggers* que se utilizaron fueron los estándares, aquellos que vienen de forma predeterminada en las plantillas de Zabbix 7.0 para equipos Cisco IOS y MikroTik RouterOS. Sin embargo, para algunos escenarios, se tuvo que modificar algunos umbrales con fines de verificación del sistema.

Cuadro D1 Umbrales de alerta del sistema de monitoreo

Componente	Condición del Disparador (<i>Trigger</i>)	Origen del Umbral	Severidad
Conectividad Base	ICMP <i>ping</i> : Pérdida de paquetes > 20%	Plantilla Oficial (Default)	High
Estado físico	Interface OperStatus cambia a Down	Plantilla Oficial (Default)	Disaster
<i>Hardware</i> (CPU)	Uso del procesador > 50% por 1 minuto	Personalizado (Validación)	Average
Tráfico de red	Saturación del enlace supera la política de 50 Mbps	Personalizado (Validación)	Warning
<i>Hardware</i> (RAM)	Falta de memoria libre (< 10% disponible)	Plantilla Oficial (Default)	Average

Fuente. Elaboración propia según la documentación encontrada en Zabbix 7.0 s.

Anexo E: Cuadro de identificadores de objetos (OIDs)

Cuadro E1 Identificadores de objetos estándar y propietarios más comunes

Categoría	Objeto	OID	Descripción
Sistema	sysDescr	1.3.6.1.2.1.1.1.0	Descripción del sistema (<i>hardware</i> y <i>software</i>)
Sistema	sysUpTime	1.3.6.1.2.1.1.3.0	Tiempo desde el último reinicio
Sistema	sysName	1.3.6.1.2.1.1.5.0	Nombre del dispositivo
Interfaces	ifNumber	1.3.6.1.2.1.2.1.0	Número total de interfaces
Interfaces	ifDescr	1.3.6.1.2.1.2.2.1.2.[i]	Nombre de la interfaz
Interfaces	ifSpeed	1.3.6.1.2.1.2.2.1.5.[i]	Velocidad de la interfaz (bps)
Interfaces	ifOperStatus	1.3.6.1.2.1.2.2.1.8.[i]	Estado operativo (1=up, 2=down)
Trafico	ifInOctets	1.3.6.1.2.1.2.2.1.10.[i]	Bytes recibidos
Trafico	ifOutOctets	1.3.6.1.2.1.2.2.1.16.[i]	Bytes enviados
Cpu	hrProcessorLoad	1.3.6.1.2.1.25.3.3.1.2.[i]	Carga de CPU(%)
Memoria	hrMemorySize	1.3.6.1.2.1.25.2.2.0	Memoria RAM total
Memoria	hrStorageUsed	1.3.6.1.2.1.25.2.3.1.6.[i]	Memoria utilizada
Cisco	cpmCPUTotal5minRev	1.3.6.1.4.1.9.9.109.1.1.1.1.8.[i]	Uso de CPU en 5 minutos

Categoría	Objeto	OID	Descripción
Cisco	CiscoMemoryPoolUsed	1.3.6.1.4.1.9.9.48.1.1.1.6.[i]	Memoria utilizada
MikroTik	mtxrHICpuLaoad	1.3.6.1.4.1.14988.1.1.3.10.0	Carga de CPU(%)
MikroTik	mtxrHIMemoryUsed	1.3.6.1.4.1.14988.1.1.3.11.0	Memoria RAM utilizada

Fuente. Elaboración propia basada en los RFC 2863,2790,1213 del IETF y en los MIBs propios de Cisco Systems y MikroTik. El valor [i] al final de algunos OIDs representa un índice de interfaz, el protocolo SNMP asigna este valor a toda interfaz física o lógica para diferenciarlos en sus consultas.

Anexo F: Creación y configuración de la base de datos MariaDB por consola

```
soporte@new-zabbix-server:~$ sudo mysql
[sudo] password for soporte:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 31
Server version: 10.11.14-MariaDB-0ubuntu0.24.04.1 Ubuntu 24.04

Copyright (c) 2000, 2018, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> create database zabbix character set utf8mb4 collate utf8mb4_bin;
Query OK, 1 row affected (0.000 sec)

MariaDB [(none)]> create user zabbix@localhost identified by 'zabbix';
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> grant all privileges on zabbix.* to zabbix@localhost;
Query OK, 0 rows affected (0.001 sec)

MariaDB [(none)]> set global log_bin_trust_function_creators = 1;
Query OK, 0 rows affected (0.000 sec)

MariaDB [(none)]> quit;
Bye
soporte@new-zabbix-server:~$
```

Figura 24 Comandos de inicialización, creación de usuario y asignación de permisos para la base de datos del sistema de monitoreo. Fuente. Elaboración propia a partir de Ubuntu Server.

Anexo G: Configuración de conexión a la base de datos MariaDB

ZABBIX

Welcome

Check of pre-requisites

Configure DB connection

Settings

Pre-installation summary

Install

Configure DB connection

Please create database manually, and set the configuration parameters for connection to this database. Press "Next step" button when done.

Database type: MySQL

Database host: localhost

Database port: 0 - use default port

Database name: zabbix

Store credentials in: Plain text | HashiCorp Vault | CyberArk Vault

User: zabbix

Password: *****

Database TLS encryption: Connection will not be encrypted because it uses a socket file (on Unix) or shared memory (Windows).

Back Next step

Figura 25 Parámetros de conexión a la base de datos durante la configuración inicial de la interfaz web de Zabbix. Fuente. Elaboración propia a partir de Zabbix.

Anexo H: Creación del Bot de Telegram y obtención del API Token mediante BotFather.

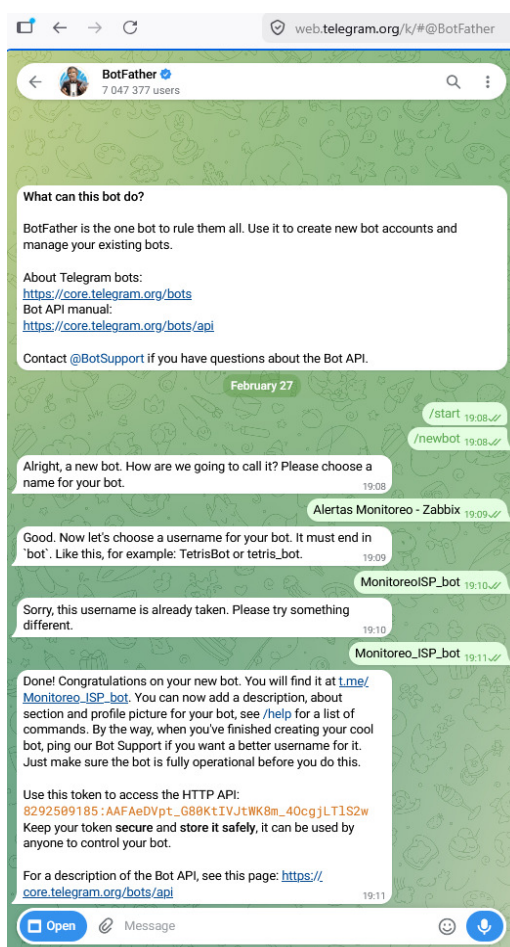


Figura 26 Proceso de creación del bot de notificaciones y generación del API Token mediante la plataforma BotFather. Fuente. Elaboración propia a partir de Telegram.

Anexo I: Vinculación del Token en el Media Type de Zabbix.

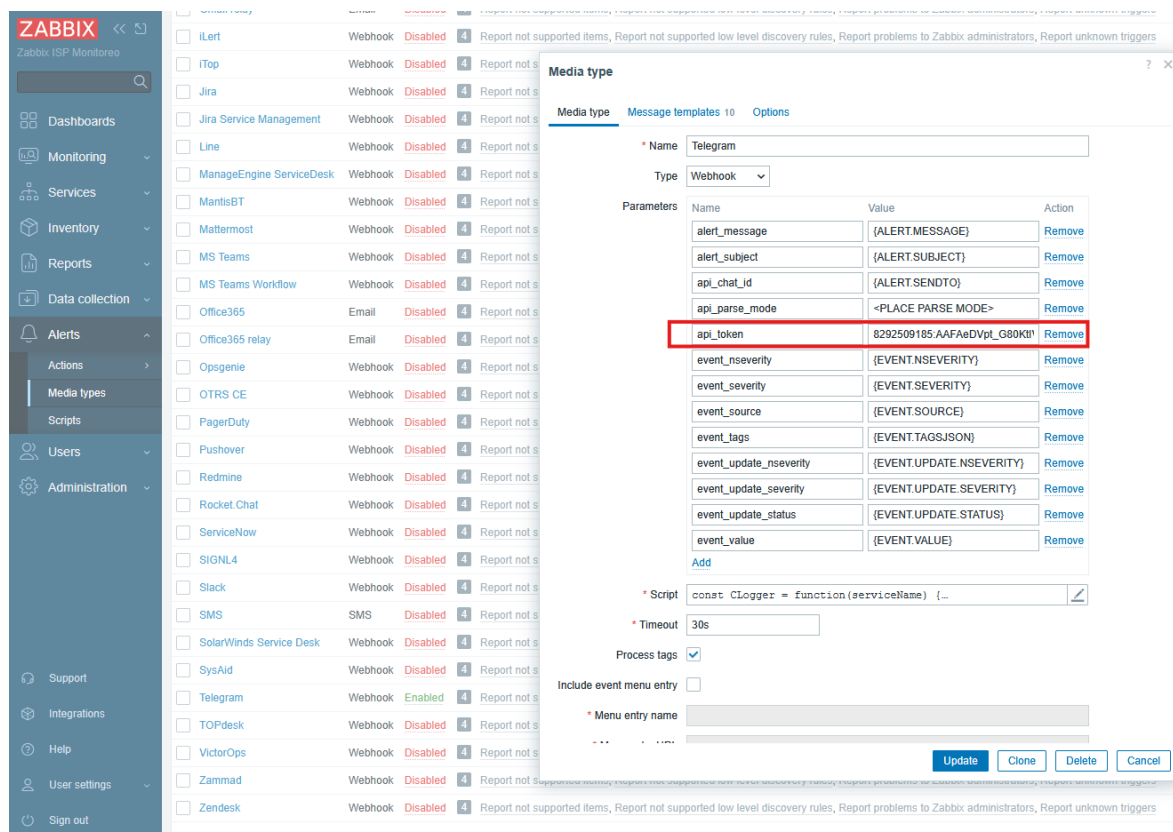


Figura 27 Configuración del tipo de medio (Media Type) de Telegram en el servidor Zabbix. Fuente. Elaboración propia a partir de Zabbix.

Anexo J: Registro manual de host y asignación de credenciales SNMP en Zabbix.

Host

Host IPMI Tags 1 Macros 2 Inventory Encryption Value mapping

* Host name CPE-Cisco

Visible name CPE-Cisco

Templates Name Action

Cisco IOS by SNMP Unlink Unlink and clear

type here to search Select

* Host groups Routers Cisco X type here to search Select

Interfaces Type IP address DNS name Connect to Port Default

SNMP 10.99.1.2 IP DNS 161 Remove

* SNMP version SNMPv2

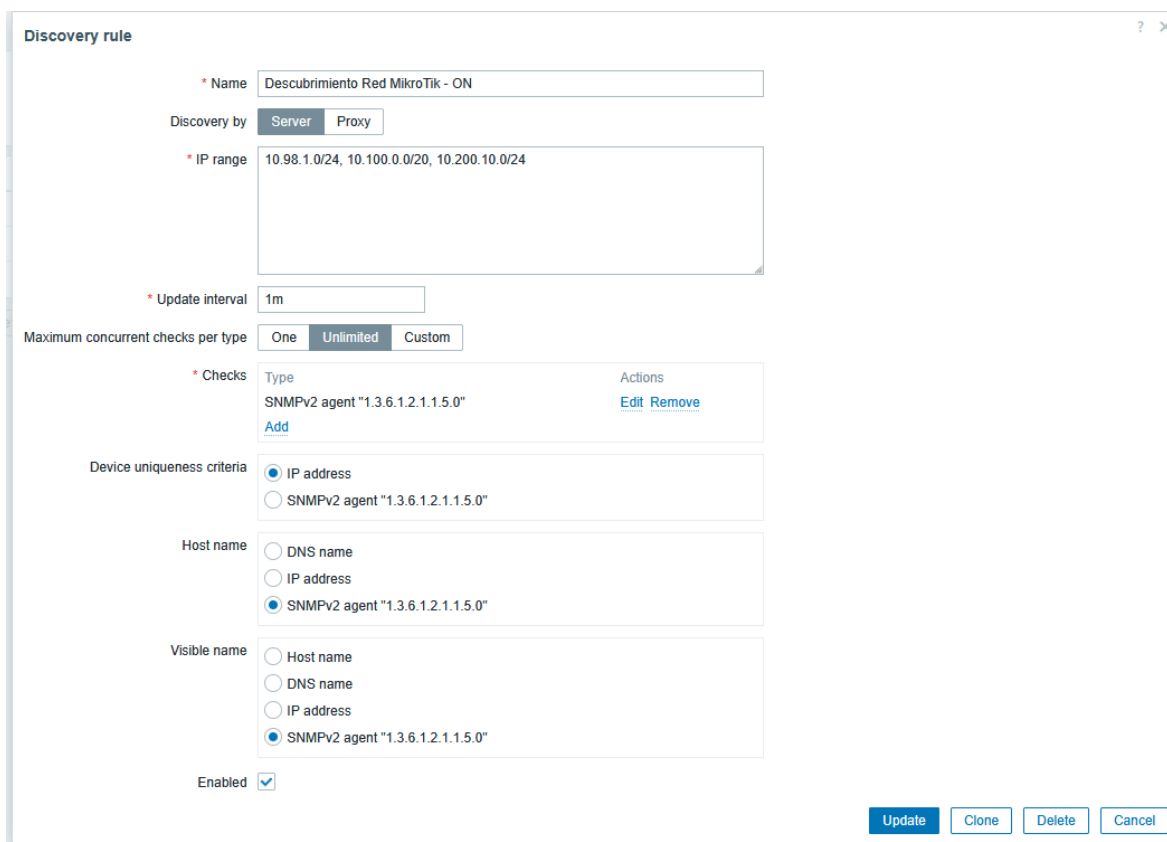
* SNMP community {\$SNMP_COMMUNITY}

Max repetition count 10

☒ Use combined requests

Figura 28 Configuración del host en Zabbix para el monitoreo del dispositivo mediante SNMP. Fuente: Elaboración propia a partir de Zabbix.

Anexo K: Configuración de la regla de autodescubrimiento (Network Discovery) de equipos en la red.



The screenshot shows the 'Discovery rule' configuration window in Zabbix. The rule is named 'Descubrimiento Red MikroTik - ON'. It is configured to discover by 'Server' using the IP range '10.98.1.0/24, 10.100.0.0/20, 10.200.10.0/24'. The update interval is set to '1m'. The maximum concurrent checks per type are set to 'Unlimited'. The checks are configured to use the 'SNMPv2 agent "1.3.6.1.2.1.1.5.0"' type. The device uniqueness criteria, host name, and visible name are all set to 'SNMPv2 agent "1.3.6.1.2.1.1.5.0"'. The rule is enabled. At the bottom, there are buttons for 'Update', 'Clone', 'Delete', and 'Cancel'.

Figura 29 Configuración de la regla de autodescubrimiento en Zabbix mediante barrido de direcciones IP y consultas de identificadores SNMP. Fuente. Elaboración propia a partir de Zabbix.

Anexo L: Diseño y estructuración del mapa topológico en Zabbix

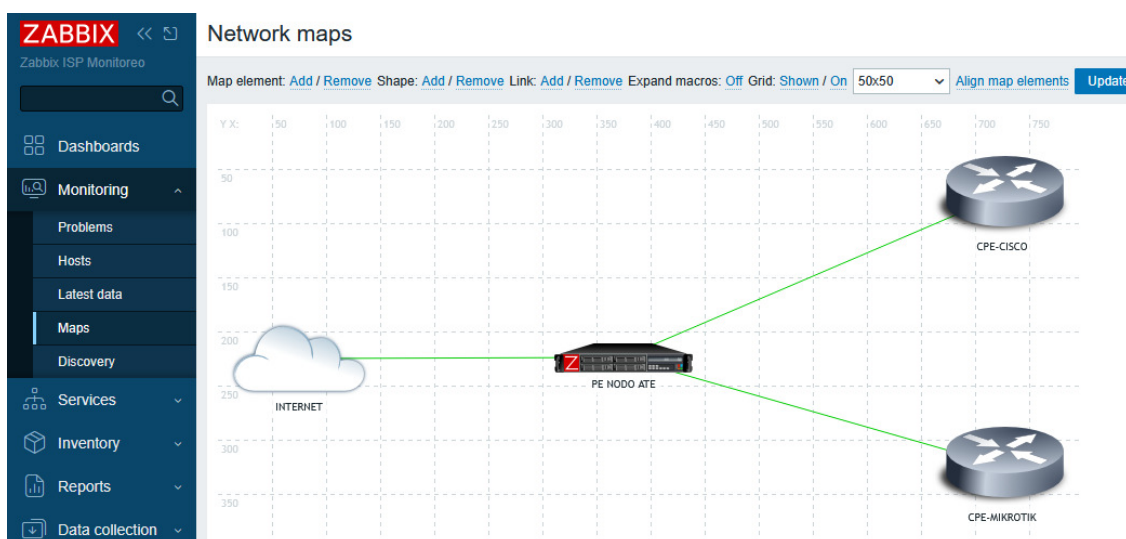


Figura 30 Interfaz de configuración del mapa de red en Zabbix, mostrando la vinculación de elementos gráficos con los equipos monitoreados. Fuente. Elaboración propia a partir de Zabbix.

Anexo M: Topología lógica de la red del sistema de monitoreo implementada en el emulador.

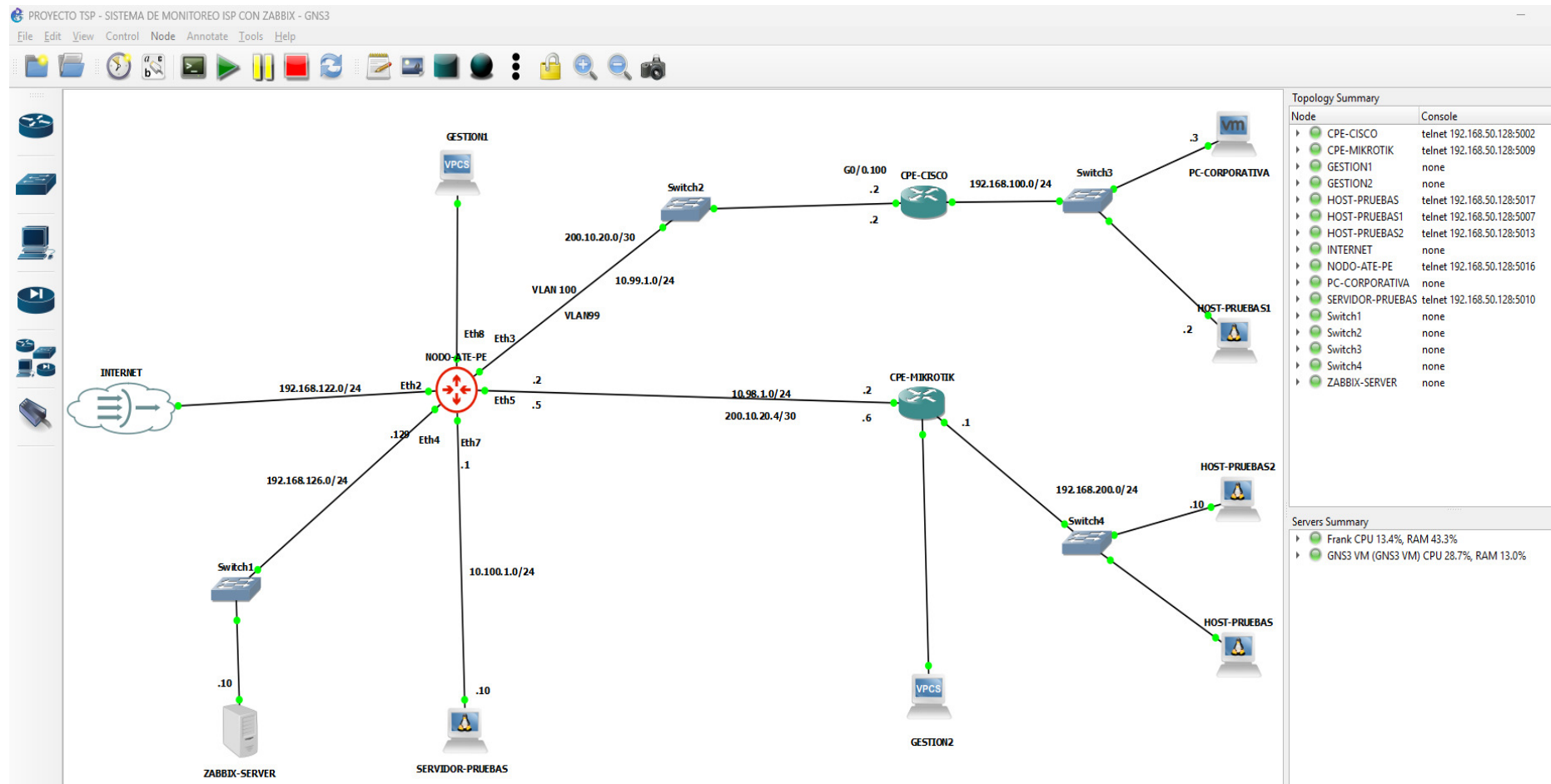


Figura 31 Topología lógica de la red en GNS3. Fuente. Elaboración propia.

Anexo N: Dashboard de monitoreo implementado en la interfaz web de Zabbix

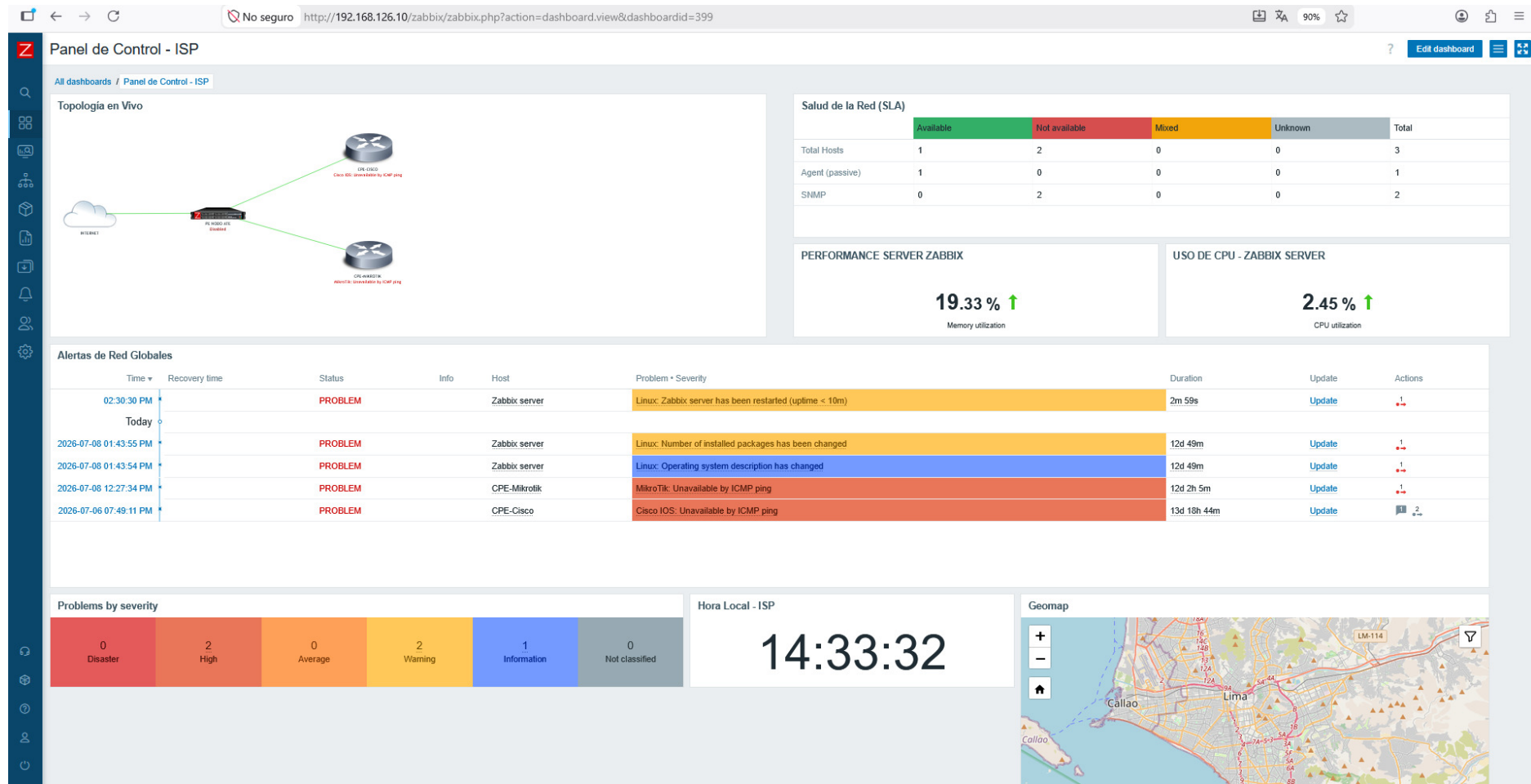


Figura 32. Vista del dashboard de monitoreo en interfaz web de Zabbix. Fuente. Elaboración propia a partir de Zabbix.