Universidad Nacional Mayor de San Marcos
Universidad del Perú. Decana de América
Facultad de Ingeniería Electrónica y Eléctrica
Escuela Profesional de Ingeniería de Telecomunicaciones

Implementación de Zabbix para mejorar operatividad del área NOC y detección de incidentes en un proyecto regional de banda ancha

TRABAJO DE SUFICIENCIA PROFESIONAL
Para optar el Título Profesional de Ingeniero de Telecomunicaciones

AUTOR

André Alexander JAULIS CONDE

ASESOR

Mg. Jesus Otto VILLANUEVA NAPURI

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

Jaulis, A. (2026). *Implementación de Zabbix para mejorar operatividad del área NOC y detección de incidentes en un proyecto regional de banda ancha*. [Trabajo de suficiencia profesional de pregrado, Universidad Nacional Mayor de San Marcos, Facultad de Ingeniería Electrónica y Eléctrica, Escuela Profesional de Ingeniería de Telecomunicaciones]. Repositorio institucional Cybertesis UNMSM.

Metadatos complementarios

Datos de autor	
Nombres y apellidos	André Alexander Jaulis Conde
Tipo de documento de identidad	DNI
Número de documento de identidad	72895413
URL de ORCID	No Aplica
Datos de asesor	
Nombres y apellidos	Mg. Jesús Otto Villanueva Napuri
Tipo de documento de identidad	DNI
Número de documento de identidad	25726864
URL de ORCID	https://orcid.org/0000-0002-3380-4991
Datos del jurado	
Presidente del jurado	
Nombres y apellidos	Dr. Celso Ysidro Geronimo Huaman
Tipo de documento	DNI
Número de documento de identidad	09191500
Miembro del jurado 1	
Nombres y apellidos	Dra. Teresa Esther Nuñez Zuñiga
Tipo de documento	DNI
Número de documento de identidad	08006778
Miembro del jurado 2	
Nombres y apellidos	Dr. Jorge Elías Moscoso Sánchez
Tipo de documento	DNI
Número de documento de identidad	08596442
Datos de investigación	
Línea de investigación	Comunicación/redes y sistemas distribuidos
Grupo de investigación	No Aplica
Agencia de financiamiento	No Aplica

Ubicación geográfica de la investigación	Edificio / institución: YOFC PERÚ S.A.C. (Network Operations Center) País: Perú Departamento: Lima Provincia: Lima Distrito: San Isidro Calle: Av. Enrique Canaval y Moreira N.º 480, Int. 1500 Latitud: -12.097462955447915 Longitud: -77.02070588881912
Año o rango de años en que se realizó la investigación	julio 2025 - diciembre 2025
URL de disciplinas OCDE	Telecomunicaciones https://purl.org/pe-repo/ocde/ford#2.02.05

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°79/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 29 de Octubre del 2026, como Presidente de Jurado la **DRA. TERESA ESTHER NUÑEZ ZUÑIGA,** integrado por el Miembro de Jurado el **DR. JORGE ELÍAS MOSCOSO SÁNCHEZ,** el Miembro de Jurado el **DR. CELSO YSIDRO GERONIMO HUAMAN** y Miembro Asesor el **MG. JESUS OTTO VILLANUEVA NAPURI.**

Después de escuchar la Sustentación Presencial del Trabajo de Suficiencia Profesional del Bachiller **ANDRÉ ALEXANDER JAULIS CONDE** con código 12190067 que para optar el Título Profesional de Ingeniero de Telecomunicaciones sustentó el Trabajo de Suficiencia Profesional titulado **IMPLEMENTACIÓN DE ZABBIX PARA MEJORAR OPERATIVIDAD DEL AREA NOC Y DETECCIÓN DE INCIDENTES EN UN PROYECTO REGIONAL DE BANDA ANCHA.**

El jurado examinador procedió a formular las preguntas reglamentarias y, luego de una deliberación en privado, decidió otorgarle el calificativo de 16 (Dieciséis).

Ciudad Universitaria, 29 de Octubre del 2026

DRA. TERESA ESTHER NUÑEZ ZUÑIGA
Presidente Jurado

DR. CELSO YSIDRO GERONIMO HUAMAN
Miembro Jurado

DR. JORGE ELÍAS MOSCOSO SÁNCHEZ
Miembro Jurado

MG. ANA CORONEL CHAMORRO
Miembro Asesor

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

CERTIFICADO DE SIMILITUD

Yo, **JESUS OTTO VILLANUEVA NAPURI** en mi condición de asesor acreditado con el Dictamen N°80/FIEE-PPP/2026 del Trabajo de Suficiencia Profesional titulado **IMPLEMENTACIÓN DE ZABBIX PARA MEJORAR OPERATIVIDAD DEL AREA NOC Y DETECCIÓN DE INCIDENTES EN UN PROYECTO REGIONAL DE BANDA ANCHA,** presentado por el bachiller ANDRÉ ALEXANDER JAULIS CONDE, para optar el título profesional de Ingeniero de Telecomunicaciones. 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 03% 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 de Telecomunicaciones.

Lima, 29 de Octubre del 2026

Firma del Asesor

DNI: 25726864

Nombres y apellidos del asesor:
JESUS OTTO VILLANUEVA NAPURI

DEDICATORIA

Este trabajo se lo dedico a mis padres y familiares por su apoyo, a mi pareja por ser el motor y motivo para superarme cada día y finalmente a todos los involucrados académicamente en este trabajo.

AGRADECIMIENTOS

Un gran agradecimiento a la empresa por facilitarme los recursos necesarios para realizar este trabajo

RESUMEN

La empresa YOFC PERÚ S.A.C. dependía del sistema Nokia NSP NFM-P para supervisar su red, bajo un modelo de filtrado manual de alarmas para ubicar routers sin gestión en sus cuatro regiones operativas. Esto fragmentaba la información, exigía revisión constante y carecía de mecanismos automatizados de detección de incidentes afectando la eficiencia del NOC.

El presente trabajo buscó implementar un sistema de monitoreo integral basado en Zabbix para reforzar el desempeño del NOC en la detección y gestión de incidentes bajo una arquitectura centralizada alineada con la estructura regional que incorporó routers, gestores auxiliares, sistema de energía y dispositivos de seguridad física.

La implementación comprendió Zabbix 7.2 sobre Ubuntu 22.04 con PostgreSQL, la organización jerárquica de host por región, la carga masiva de dispositivos, triggers ICMP de disponibilidad y la integración con Telegram para alertamiento por región, junto con dashboards del estado de la red en tiempo real.

Los resultados mostraron mejora en la detección de incidentes, con el paso de un modelo reactivo a un esquema centralizado y automatizado. Unificar la información mejoro la visibilidad operativa y estandarizó la gestión de incidentes, dejando un sistema escalable y alineado con buenas prácticas de gestión de redes.

Palabras clave: Zabbix; monitoreo de red; detección de incidentes; NOC; alertamiento automatizado,

ABSTRACT

YOFC PERÚ S.A.C. relied on the Nokia NSP NFM-P system to monitor its network, under a manual alarm-filtering model used to locate unmanaged routers across its four operational regions. This fragmented information required constant review, and lacked automated incident detection mechanisms, affecting the efficiency of NOC.

This work aimed to implement a comprehensive monitoring system based on Zabbix to strengthen NOC performance in incident detection and management, under a centralized architecture aligned with the existing regional structure, incorporating routers, auxiliary managers, power systems, and physical security devices.

The implementation included Zabbix 7.2 on Ubuntu 22.04 with PostgreSQL, the hierarchical organization of hosts by region, bulk device loading, ICMP availability triggers, and Telegram integration for region-based alerting, along with dashboards showing real-time network status.

The results showed improved incident detection, moving from a reactive model to a centralized, automated scheme. Unifying the information improved operational visibility and standardized incident management, resulting in a scalable system aligned with good network management practices.

Keywords: Zabbix; network monitoring; incident detection; NOC; automated alerting.

INDICE GENERAL

CARÁTULA i

RESUMEN iv

ABSTRACT v

INDICE GENERAL vi

LISTA DE FIGURAS xi

LISTA DE TABLAS xiii

CAPÍTULO I: INTRODUCCIÓN 1

1.1 Objetivos del informe 1

1.2 Estructura del informe 3

CAPITULO II: INFORMACIÓN DEL LUGAR DONDE SE DESARROLLÓ LA ACTIVIDAD 5

2.1 Institución donde se desarrolló la actividad 5

2.2 Periodo de duración de la actividad 5

2.3 Finalidad y objetivos de la entidad 6

2.4 Razón social 6

2.5 Dirección postal 6

2.6 Nombre y correo electrónico del profesional a cargo 6

CAPITULO III: DESCRIPCIÓN DE LA ACTIVIDAD 7

3.1 Organización de la actividad 7

3.2 Finalidad y objetivos de la Actividad 8

3.2.1 Finalidad 8

3.2.2 Objetivo 8
3.3 Problemática 9
3.3.1 Problema General 11
3.3.2 Problemas Específicos 11
3.3.3 Justificación e importancia de la investigación 12
3.4 Metodología 12
3.4.1 Bases teóricas 13
3.4.2 Marco conceptual 25
3.5 Procedimiento 29
3.5.1 Diagnóstico del Entorno Pre-implementación 30
3.5.2 Planificación de la Implementación 32
3.5.2.1 Definición de Requerimientos. 32
3.5.2.2 Selección de Tecnologías. 33
3.5.2.3 Dimensionamiento de la Infraestructura. 34
3.5.2.4 Diseño de la Estructura de Monitoreo. 35
3.5.2.5 Estrategia de Despliegue. 36
3.5.2.6 Consideraciones de Seguridad y Acceso. 37
3.5.2.7 Diseño Conceptual de la Arquitectura. 38
3.5.3 Configuración de Zabbix para el Monitoreo de Red 40
3.5.3.1 Configuración de ICMP Ping.. 41
3.5.4 Organización de la Estructura de la red en Zabbix 44
3.5.4.1 Creación de Grupos de Hosts. 45
3.5.4.2 Carga masiva de Hosts en Zabbix. 47
3.5.5 Configuración del Sistema de Alertas 49
3.5.5.1 Definición de Niveles de Alerta. 49
3.5.6 Configuración de Telegram en Zabbix 51
3.5.6.1 Creación del Bot y Obtención del Token API . 51

3.5.6.2 Configuración de Grupos por región. 52

3.5.6.3 Configuración de usuarios en Zabbix por región. 53

3.5.6.4 Implementación del Script de Envío. 54

3.5.6.5 Personalización de Mensajes. 55

3.5.6.6 Arquitectura de Gestión de Alertas. 57

3.5.6.7 Implementación de Dashboards. 60

3.5.6.8 Arquitectura de Dashboards. 60

3.5.6.9 Configuración de Dashboards Regionales. 61

3.5.6.10 Implementación Técnica. 62

3.5.7 Implementación de Monitoreo SNMP para los Enlaces de la red de Transporte 63

3.5.7.1 Identificación de OIDs y Diseño del Mecanismo de Descubrimiento. 64

3.5.7.2 Creación de los tres Master Items. 65

3.5.7.3 Configuración de la Discovery Rule y Filtros. 66

3.5.7.4 Item Prototypes, Trigger Prototypes y Encapsulamiento en Template. 67

3.5.7.5 Creación del Template Nokia IXR - Interface Monitor. 68

3.5.7.6 Automatización de la Asignación Masiva del template Nokia IXR - Interface Monitor mediante API. 68

3.5.7.7 Diferenciación de Alertas Telegram (Host vs. Enlace). 69

3.5.7.8 Optimización de la Base de Datos con TimescaleDB . 70

3.6 Resultado de la actividad 70

3.6.1 Evidencia del Entorno Configurado 71

3.6.2 Sistema de Alertas Implementado 72

3.6.3 Personalización de Plantillas de Mensaje 75

3.6.4 Dashboards Desplegados 77

3.6.5 Estructura Organizacional Implementada 81

3.6.6 Impacto en el rendimiento operativo del NOC 81
3.6.7 Resultados del Monitoreo SNMP de Enlaces 85
CAPÍTULO IV: DISCUSIÓN DE RESULTADOS 89
4.1 Descripción de la implementación 89
4.1.1 Transformación del Modelo Operativo del NOC 89
4.1.1.1 Evaluación de las proposiciones teóricas orientadoras. 91
4.1.2 Impacto en Indicadores Operativos 92
4.1.2.1 Estadísticos descriptivos de la ventana de detección y del tiempo de notificación. 94
4.1.2.2 Análisis de correlación entre la hora del evento y la ventana de detección. 95
4.1.3 Evaluación del cumplimiento de SLA y disponibilidad del servicio 97
4.1.4 Evaluación del desempeño de la infraestructura 98
4.1.5 Discusión Crítica de la Arquitectura Implementada 102
4.1.6 Contraste con Buenas Prácticas 103
4.1.6.1 Comparación con Otras Implementaciones de Zabbix en Telecomunicaciones. 105
4.1.7 Modelo de implementación de monitoreo centralizado 107
4.1.8 Limitaciones del estudio 109
4.2 Evaluación Económica 111
CAPÍTULO V: CONCLUSIONES 121
5.1 Conclusión General 121
5.2 Cumplimiento del Objetivo General 122
5.3 Implicaciones del caso 123
5.4 Impacto en la Gestión de Incidentes 124
5.5 Centralización y Mejora en la Visibilidad Operativa 124
5.6 Evaluación Técnica de la Solución Implementada 125

5.7 Impacto Económico y Estratégico 126

5.8 Aporte Profesional y Académico 126

5.9 Alineamiento con los Objetivos de Desarrollo Sostenible (ODS) 127

5.10 Aporte del Modelo de Implementación Propuesto 128

CAPÍTULO VI: RECOMENDACIONES 129

6.1 Fortalecimiento del Monitoreo mediante Protocolos Avanzados 129

6.2 Implementación de Alta Disponibilidad del Servidor Zabbix (Alta prioridad – continuidad operativa crítica) 130

6.3 Implementación de Correlación Avanzada de Eventos (Prioridad media – optimización operativa) 131

6.4 Optimización de Dashboards mediante Indicadores SLA (Prioridad media – mejora de gestión) 131

6.5 Capacitación Continua del Personal Técnico (Prioridad media – sostenibilidad operativa) 132

6.6 Evaluación Periódica del Desempeño del Sistema (Alta prioridad – control y mejora continua) 135

6.7 Evolución estratégica del sistema de monitoreo (Prioridad baja – consolidación y crecimiento a largo plazo) 135

6.8 Sostenibilidad técnica y transferencia de conocimiento 136

6.9 Ampliación de la Validación Cuantitativa del MTTD (Prioridad media – control y mejora continua) 137

6.10 Implementación de Proxies Zabbix en Regiones Remotas (Prioridad media – escalabilidad y continuidad operativa) 138

CAPÍTULO VII: BIBLIOGRAFÍA 141

CAPITULO VIII: ANEXOS 146

8.1 ANEXO A: Archivo JSON de carga masiva de hosts 146

8.2 ANEXO B: Manual básico de operación del sistema 149

8.3 ANEXO C: Script de Asignación Masiva del template Nokia IXR - Interface Monitor 150

LISTA DE FIGURAS

Figura 1 Arquitectura de supervisión (modelo fragmentado). 32

Figura 2 Esquema Conceptual y sus componentes. 39

Figura 3 Modelo jerárquico de organización de hosts en Zabbix. 46

Figura 4 Carga del archivo json en sistema Zabbix. 47

Figura 5 Estructura general de la media Type de Telegram. 55

Figura 6 Message Template. 57

Figura 7 Template Nokia IXR – Interface Monitor. 68

Figura 8 Ejemplo del Resumen de ejecución del script de asignación masiva para una región. 68

Figura 9 Instalación de la plataforma sobre Ubuntu Server 22.04 LTS. 71

Figura 10 Instalación de Zabbix sobre Ubuntu Server 22.04 LTS. 72

Figura 11 Instalación de la base de datos PostgreSQL. 72

Figura 12 Creación del bot y Token. 73

Figura 13 Prueba de funcionamiento del token del bot. 73

Figura 14 Grupos regionales creados en Telegram. 74

Figura 15 Usuarios regionales creados en Telegram. 75

Figura 16 Test de envío de alerta. 75

Figura 17 Mensaje de Evento critico en Telegram. 76

Figura 18 Mensaje de Fin de Evento en Telegram. 76

Figura 19 Vista del Dashboard Corporativo. 78

Figura 20 Vista del Dashboard Regional Arequipa. 80

Figura 21 Ejemplo de Widget de tendencia de uso de memoria de disco del servidor Zabbix (4 parámetros monitoreados). 86

Figura 22 Tendencia de Utilización de CPU en router. 99

Figura 23 Tráfico de entrada y Salida en interfaz de red. 100

Figura 24 Errores y descartes en interfaz. 101

Figura A1 Prompt para archivo json de importación masiva de hosts. 146

Figura B1 Captura de Manual de Usuario para personal NOC. 150

LISTA DE TABLAS

Tabla 1 Comparación de herramientas de monitoreo de red 18

Tabla 2 Limitaciones identificadas 30

Tabla 3 Selección de Tecnología 34

Tabla 4 Parámetros de configuración ICMP 41

Tabla 5 Métricas de disponibilidad generadas 42

Tabla 6 Umbrales de alerta configurados 43

Tabla 7 Estructura jerárquica de monitoreo en Zabbix 45

Tabla 8 Comparación de métodos de incorporación de hosts 48

Tabla 9 Niveles de alerta configurados 50

Tabla 10 Configuración de grupos Telegram 52

Tabla 11 Usuarios configurados en Zabbix 53

Tabla 12 Ejemplo de ciclo de alerta crítica para un host 59

Tabla 13 OIDs IF-MIB utilizados para el monitoreo de enlaces 64

Tabla 14 Diagrama de arquitectura en capas 65

Tabla 15 Configuración básica de cada Master Item 66

Tabla 16 Configuración de filtros de la Discovery Rule 66

Tabla 17 Trigger Prototypes configurados para el monitoreo de enlaces 67

Tabla 18 Comparación de mensajes de Alerta de Telegram diferenciados: HOST vs. LINK 69

Tabla 19 Comparación de mensajes de Recuperación de evento de Telegram diferenciados: HOST vs. LINK 69

Tabla 20 Registro de eventos reales de indisponibilidad y ventana de detección 82

Tabla 21 Comparación cualitativa del proceso de detección de incidentes (antes y después de la implementación) 85

Tabla 22 Incidentes de infraestructura identificados y resueltos 86

Tabla 23 Dimensionamiento final de memoria caché del servidor Zabbix 87

Tabla 24 Resultado de la compresión TimescaleDB sobre la tabla de historial principales durante la primera semana de implementación 88

Tabla 25 Comparación estructural del modelo de monitoreo 90

Tabla 26 Registro de eventos reales y tiempo de notificación vía Telegram 92

Tabla 27 Análisis estadístico descriptivo de los indicadores de detección y notificación 95

Tabla 28 Comparación del modelo de monitoreo en función de buenas prácticas de gestión de servicios TI (ITIL e ISO/IEC 20000) 104

Tabla 29 Resumen comparativo de la implementación Zabbix de YOFC PERU SAC con Hrvatski Telekom, O2 Czech Republic y LMT Latvia 105

Tabla 30 Fases del modelo de implementación de monitoreo centralizado propuesto 107

Tabla 31 Comparación de costos estimados 112

Tabla 32 Estimación de recursos utilizados en la implementación 113

Tabla 33 Estimación de Retorno de la Inversión (ROI) dentro del primer año 116

Tabla 34 Proyección de costos evitados por licenciamiento (horizonte 3–5 años) 117

Tabla 35 Estimación de ahorro de horas-hombre 118

Tabla 36 Comparación económica entre la solución implementada y alternativas comerciales 119

Tabla 37 Estructura del Plan de Capacitación 133

Tabla 38 Plan de implementación en 6 fases 139

CAPÍTULO I:
INTRODUCCIÓN

1.1 Objetivos del informe

En el contexto actual de las telecomunicaciones, la continuidad operativa de los servicios depende directamente de la capacidad de supervisión y gestión proactiva de la infraestructura de red. En organizaciones que administran redes IP/MPLS distribuidas geográficamente, la eficiencia del Centro de Operaciones de Red (Network Operations Center – NOC) constituye un factor crítico para garantizar altos niveles de disponibilidad, confiabilidad y calidad del servicio.

YOFC PERÚ S.A.C., filial de Yangtze Optical Fibre and Cable Joint Stock Limited, desarrolla sus operaciones en el marco del proyecto regional de banda ancha adjudicado por el Estado peruano en el año 2019, cuyo objetivo es reducir la brecha digital en zonas rurales del país. En virtud de esta concesión, la empresa tiene presencia operativa en las regiones de Arequipa, San Martín, La Libertad y Áncash, brindando acceso a internet de alta velocidad a instituciones públicas estratégicas tales como hospitales, comisarías, centros educativos y plazas públicas. La naturaleza del servicio prestado exige altos estándares de disponibilidad, dado que la conectividad impacta directamente en servicios esenciales para la población.

En el marco de las políticas públicas de telecomunicaciones del Perú, el desarrollo de proyectos de conectividad en zonas rurales se encuentra alineado con el Plan Nacional de Telecomunicaciones y con las iniciativas del Estado orientadas al cierre de la brecha digital. En este contexto, el Proyecto Regional de Banda Ancha, adjudicado mediante mecanismos de inversión pública, tiene como finalidad garantizar el acceso universal a servicios de telecomunicaciones en localidades de difícil acceso.

A ello se suma el marco regulatorio establecido por el Ministerio de Transportes y

Comunicaciones (MTC), a través del Decreto Supremo N.º 003-2021-MTC, que promueve la expansión de infraestructura digital como eje estratégico para el desarrollo social y económico del país. Por otra parte, el Organismo Supervisor de Inversión Privada en Telecomunicaciones (OSIPTEL), mediante la Resolución de Consejo Directivo N.º 138-2020-CD-OSIPTEL, establece estándares de calidad y continuidad del servicio que deben ser cumplidos por los operadores.

En este escenario, la supervisión eficiente de la infraestructura de telecomunicaciones es una necesidad operativa interna y un requisito fundamental para garantizar el cumplimiento de las obligaciones regulatorias, la sostenibilidad de los proyectos de conectividad y la provisión de servicios de calidad a la población beneficiaria.

La operación de esta infraestructura implica la supervisión continua de una red compuesta por routers de transporte y acceso, sistemas de energía, plataformas de gestión y dispositivos de seguridad física distribuidos territorialmente en las cuatro regiones concesionadas. Esta dispersión geográfica incrementa la complejidad operativa y demanda un modelo de monitoreo robusto, centralizado y eficiente.

Previo a la implementación desarrollada en el presente Trabajo de Suficiencia Profesional, la supervisión de la infraestructura se realizaba mediante el sistema NSP NFM-P de Nokia y gestores independientes para energía y seguridad física. Si bien estas herramientas permitían obtener información técnica relevante, la gestión de incidentes dependía de procesos manuales de filtrado y consultas fragmentadas. Esta dinámica generaba demoras en la detección temprana de fallas, dependencia de revisión constante y ausencia de una visión consolidada en tiempo real del estado global de la red.

Esta situación evidenció una necesidad operativa concreta: contar con una plataforma centralizada capaz de consolidar en tiempo real la información de los routers, gestores auxiliares, sistemas de energía y dispositivos de seguridad física, permitiendo al personal del NOC actuar de manera inmediata ante cualquier degradación del servicio y mejorar el desempeño operativo en la detección temprana de incidentes.

En respuesta a esta problemática, se planteó la implementación de un sistema de monitoreo con base en Zabbix, herramienta de código abierto ampliamente utilizada

en entornos empresariales por su escalabilidad, flexibilidad y capacidad de personalización según *Olups, R. (2010)*. La solución propuesta contempló la supervisión de equipos de red y la integración de notificaciones automatizadas mediante el soporte de Telegram, el diseño de dashboards corporativos y regionales, y la estructuración lógica de la infraestructura en función de la arquitectura operativa existente en las cuatro regiones.

1.2 Estructura del informe

El presente trabajo documenta el proceso de análisis, diseño, implementación y evaluación de la solución propuesta, evidenciando la aplicación de competencias profesionales propias de la Ingeniería Electrónica y de Telecomunicaciones. En el desarrollo de la solución se aplicaron competencias relacionadas con la gestión de redes IP/MPLS y administración de infraestructura tecnológica, las cuales, según *Stallings (2018)*, son elementos fundamentales para garantizar la disponibilidad y continuidad operativa de las redes modernas.

De forma complementaria, el diseño de la arquitectura de monitoreo centralizado y la estructuración lógica de dispositivos por regiones se fundamentan en los principios de diseño jerárquico y escalabilidad de redes descritos por *Oppenheimer (2011)*, quien destaca la importancia de implementar arquitecturas organizadas y eficientes en entornos distribuidos geográficamente.

De igual manera, la integración de plataformas tecnológicas, automatización de alertas y consolidación de dashboards corporativos responde a las buenas prácticas de gestión de servicios TI propuestas por ITIL *(Axelos, 2019)*, donde se enfatiza la detección temprana de incidentes, la automatización de procesos operativos y la mejora continua como factores esenciales para fortalecer la gestión de infraestructuras críticas.

Este compromiso institucional cobra un sentido concreto en el caso de YOFC PERÚ S.A.C. La magnitud de esta necesidad puede dimensionarse con cifras recientes: según el Instituto Nacional de Estadística e Informática (INEI, 2025), durante el primer trimestre de 2025 el 58.9% de los hogares del país contó con acceso a Internet, mientras

que en el ámbito rural esa proporción cae a apenas 20.5%, un contraste que, pese a los avances registrados en los últimos años, sigue siendo significativo. La red que se supervisa aquí no conecta oficinas corporativas: conecta hospitales, comisarías, centros educativos y plazas públicas en localidades donde, muchas veces, esa conexión es la única vía de acceso a servicios básicos. Por eso cada minuto que el sistema de monitoreo logra recortar en la detección de una falla termina teniendo una lectura que trasciende lo técnico. Sostener la disponibilidad de esta red aporta a la meta 9.c de la Agenda 2030, que busca ampliar el acceso a las tecnologías de la información y la comunicación (ONU, 2015). Y en la medida en que esta infraestructura forma parte de un proyecto pensado explícitamente para reducir la brecha digital en zonas rurales, el trabajo también se conecta con el Objetivo de Desarrollo Sostenible 10, referido a la reducción de las desigualdades, e indirectamente favorece la continuidad de servicios de salud y educación en esas mismas localidades.

Por todo esto, el presente trabajo no se limita a resolver un problema operativo del NOC. Combina conocimiento técnico, competencias propias de la ingeniería de telecomunicaciones y una infraestructura que, en la práctica, sostiene el acceso a servicios esenciales de poblaciones que dependen de ella.

CAPITULO II:
INFORMACIÓN DEL LUGAR DONDE SE DESARROLLÓ LA ACTIVIDAD

2.1 Institución donde se desarrolló la actividad

La actividad descrita en el presente Trabajo de Suficiencia Profesional se desarrolló en la empresa YOFC PERÚ S.A.C., filial en el Perú de la corporación internacional Yangtze Optical Fibre and Cable Joint Stock Limited Company (YOFC), empresa especializada en la fabricación y provisión de soluciones integrales de fibra óptica y sistemas de telecomunicaciones.

YOFC PERÚ S.A.C. participa en la ejecución de proyectos regionales de conectividad, orientados principalmente al despliegue de infraestructura de fibra óptica y a la implementación de redes de transporte y acceso en distintas regiones del territorio nacional. Su operación se encuentra alineada con los objetivos de expansión de infraestructura digital y fortalecimiento del ecosistema de telecomunicaciones en el país.

En este contexto organizacional, el área del Network Operations Center (NOC) cumple un rol estratégico en la supervisión continua de la infraestructura desplegada, garantizando la disponibilidad, estabilidad y calidad del servicio ofrecido.

2.2 Periodo de duración de la actividad

La actividad profesional documentada en el presente trabajo se desarrolló durante el periodo comprendido entre julio y diciembre del año 2025.

2.3 Finalidad y objetivos de la entidad

YOFC PERÚ S.A.C. tiene como finalidad principal el diseño, despliegue y mantenimiento de infraestructura de fibra óptica, así como la provisión de servicios de conectividad de alta velocidad a instituciones públicas y privadas.

Uno de sus objetivos estratégicos consiste en contribuir al cierre de la brecha digital en localidades rurales y zonas de difícil acceso, mediante la implementación de soluciones tecnológicas que permitan el acceso a servicios de telecomunicaciones confiables y sostenibles. De esta manera, la empresa promueve el desarrollo social, educativo y económico de las comunidades beneficiadas.

2.4 Razón social

La razón social de la empresa es: YOFC PERU S.A.C

2.5 Dirección postal

La dirección fiscal y administrativa de la empresa es:

Av. Enrique Canaval y Moreira N.º 480, Int. 1500, distrito de San Isidro, Lima – Perú.

2.6 Nombre y correo electrónico del profesional a cargo

El nombre del profesional a cargo es Dany Alvarado Llatance y su correo electrónico es: dany.alvarado@yofc.com

CAPITULO III:
DESCRIPCIÓN DE LA ACTIVIDAD

3.1 Organización de la actividad

La actividad profesional se desarrolló en el área del Network Operations Center (NOC) de YOFC PERÚ S.A.C., unidad responsable de la supervisión continua de la infraestructura de red desplegada en las distintas regiones operativas del país.

El NOC administra cuatro regiones, cada una estructurada en redes de transporte y acceso, lo que implica la gestión de múltiples routers, enlaces lógicos entre los nodos de transporte, sistemas de energía y dispositivos de seguridad física asociados a la infraestructura de telecomunicaciones.

La organización del trabajo se estructuró en seis fases técnicas claramente definidas: diagnóstico del entorno existente, planificación de la arquitectura de monitoreo, implementación del servidor Zabbix, integración de sistemas complementarios, configuración de notificaciones automatizadas y desarrollo de paneles de visualización regionales.

Cada fase fue ejecutada considerando la operatividad permanente del NOC, garantizando que la implementación no afectara la continuidad del servicio.

El desarrollo de estas seis fases se extendió a lo largo de aproximadamente 4 meses y la coordinación con el personal del NOC se mantuvo constante durante todo el proceso, de modo que cada avance pudiera contrastarse con la operación real de la red antes de pasar a la siguiente fase.

Así mismo con el fin de preservar la confidencialidad de la información estratégica de la organización, ciertos datos relacionados con la operación interna, métricas específicas y configuraciones detalladas han sido anonimizados o presentados de

manera general a lo largo de este trabajo. Esta decisión responde a políticas de seguridad de la empresa y no afecta la validez técnica ni los resultados del presente trabajo.

3.2 Finalidad y objetivos de la Actividad

3.2.1 Finalidad

La finalidad de la actividad consistió en fortalecer la capacidad operativa del NOC mediante la implementación de una plataforma centralizada de monitoreo que permitiera consolidar, en una única interfaz web, la supervisión de los routers y de los demás componentes críticos de la infraestructura tecnológica de YOFC PERÚ S.A.C.

La solución planteada buscó proporcionar visibilidad en tiempo real del estado de la red y de sus sistemas de soporte, incorporando mecanismos de alerta automatizados vía Telegram para los routers, así como Dashboard regionales diseñados para optimizar la gestión de incidencias. También se contempló la supervisión de gestores auxiliares, equipos de energía y sistemas de seguridad física, con el objetivo de reducir puntos ciegos operativos y mejorar la capacidad de respuesta ante fallas.

3.2.2 Objetivo

Implementar un sistema integral de monitoreo basado en Zabbix que permita:

- Centralizar en una única plataforma la información de routers, gestores auxiliares, sistemas de energía y dispositivos de seguridad física de las cuatro regiones operativas, garantizando visibilidad integral en tiempo real.
- Diseñar e implementar un sistema de notificaciones automatizada vía Telegram que permita la detección inmediata y el escalamiento oportuno de incidentes críticos
- Estructurar la configuración de Zabbix conforme la arquitectura multicapa de

la red (acceso y transporte), organizando el monitoreo bajo criterios regionales y funcionales.

- Desarrollar dashboards corporativos y regionales que faciliten el análisis histórico de eventos y visualización estratégica para la toma de decisiones operativas del NOC.

3.3 Problemática

En el contexto operativo de YOFC PERÚ S.A.C., el Network Operations Center (NOC) tiene la responsabilidad de supervisar de manera continua la infraestructura de telecomunicaciones desplegada en cuatro regiones operativas del país, cada una compuesta por redes de transporte y acceso, routers de agregación, enlaces troncales, sistemas de energía y dispositivos de seguridad física asociados a la continuidad del servicio.

Previo a la implementación desarrollada en el presente Trabajo de Suficiencia Profesional, la supervisión de la infraestructura se realizaba mediante el sistema NSP NFM-P de Nokia para la gestión de routers, mientras que los sistemas de energía y seguridad física eran monitoreados a través de gestores independientes. Si bien estas herramientas cumplían funciones específicas de administración y consulta, no estaban diseñadas para ofrecer una visualización integral, centralizada y proactiva del estado global de la infraestructura.

Esta fragmentación tecnológica generaba una serie de limitaciones operativas relevantes:

- La identificación de routers sin gestión requería procesos manuales de filtrado de alarmas.

- No existía una vista consolidada que permitiera correlacionar eventos entre red, energía y seguridad.

- La detección de fallas dependía en gran medida de la revisión constante por parte del operador.

- No se contaba con herramientas que facilitaran la supervisión regional comparativa.
- La notificación de incidentes no estaba automatizada de forma estructurada para todos los escenarios críticos.

En consecuencia, el modelo de operación del NOC tendía a ser reactivo más que preventivo. La ausencia de un sistema de monitoreo centralizado con capacidad de alertamiento dinámico implicaba que algunos incidentes pudieran detectarse con retraso, especialmente en escenarios de caída simultánea de múltiples dispositivos o fallas asociadas a la infraestructura de soporte energético.

Además, la dispersión de información en múltiples plataformas incrementaba la carga cognitiva del personal, obligando a mantener abiertas varias interfaces de gestión de manera simultánea. Esta situación impactaba el rendimiento operativo y generaba riesgos asociados a errores humanos, omisión de eventos críticos y tiempos de diagnóstico prolongados.

En términos de ingeniería de telecomunicaciones, la inexistencia de una herramienta capaz de consolidar métricas de disponibilidad, generar historiales estructurados de incidentes y ofrecer visualización en tiempo real es una debilidad en la arquitectura de supervisión de red. En entornos donde la disponibilidad del servicio es un “indicador clave de desempeño” (KPI - Key Performance Indicator), esta limitación puede afectar la percepción de confiabilidad, el cumplimiento de “acuerdos de nivel de servicio” (SLA - Service Level Agreement) y la capacidad de escalabilidad futura.

Asimismo, considerando que la empresa orienta sus operaciones al cierre de la brecha digital de conectividad en zonas rurales, cualquier interrupción prolongada de los servicios impacta directamente en las actividades de instituciones educativas, entidades públicas y comunidades beneficiarias, lo que incrementa la responsabilidad técnica del NOC.

En este escenario, la problemática no se limitaba a la carencia de una herramienta moderna de monitoreo, sino a la necesidad de transformar el modelo de supervisión hacia un enfoque integral, automatizado y orientado a la detección temprana de

incidentes, capaz de proporcionar visibilidad transversal de todos los componentes críticos de la infraestructura.

Por tanto, la implementación de un sistema centralizado basado en Zabbix se planteó como una respuesta estratégica para optimizar el rendimiento operativo del NOC, reducir tiempos de detección y mejorar la capacidad de respuesta ante eventos que comprometieran la continuidad del servicio.

3.3.1 Problema General

En el contexto operativo de YOFC PERÚ S.A.C., donde el Network Operations Center (NOC) supervisa infraestructura distribuida en múltiples regiones, redes de transporte y acceso, y sistemas de soporte asociados, surge la siguiente interrogante central:

¿Cuáles son los factores críticos que determinan el éxito de la transición de un modelo de monitoreo de red reactivo hacia uno proactivo en entornos de telecomunicaciones distribuidos geográficamente, considerando como caso de aplicación la infraestructura preexistente de YOFC PERÚ S.A.C.?

3.3.2 Problemas Específicos

¿Cómo consolidar en una plataforma unificada la información proveniente de múltiples herramientas de monitoreo de red, energía y seguridad física, garantizando visibilidad integral en tiempo real?

¿Cómo diseñar e implementar un sistema de notificaciones automatizadas que permita la detección inmediata y el escalamiento oportuno de incidentes críticos en la infraestructura de telecomunicaciones?

¿Cómo estructurar la configuración de Zabbix para adaptarla a la arquitectura multicapa de la red (acceso y transporte), considerando la organización regional y los flujos operativos existentes del NOC?

¿Cómo mejorar la capacidad de análisis histórico y la visualización estratégica de eventos mediante dashboards regionales que faciliten la toma de decisiones operativas?

3.3.3 Justificación e importancia de la investigación

La implementación de un sistema de monitoreo integral representa una solución estratégica para el fortalecimiento operativo del Network Operations Center de YOFC PERÚ S.A.C. Sin dejar de lado las limitaciones técnicas identificadas en el diagnóstico, el principal impacto se evidenciaba en la capacidad de respuesta ante incidentes y mejora en la eficiencia del proceso de supervisión.

En el día a día operativo, la ausencia de una plataforma centralizada dificultaba la toma de decisiones en tiempo real, incrementando los tiempos de diagnóstico y afectando la continuidad del servicio. Esta situación adquiere mayor relevancia considerando que la infraestructura supervisada soporta servicios críticos en zonas rurales, donde la disponibilidad del servicio tiene un impacto directo en sectores como educación, salud y seguridad.

En este contexto, la implementación de Zabbix permite optimizar los procesos de monitoreo y también fortalecer la gestión basada en indicadores, mejorar la trazabilidad de eventos y proporcionar evidencia técnica para el cumplimiento de acuerdos de nivel de servicio (SLA).

3.4 Metodología

El presente Trabajo de Suficiencia Profesional se desarrolló bajo un enfoque metodológico de tipo aplicativo–tecnológico, orientado a la solución de una problemática operativa concreta en el entorno productivo del Network Operations Center (NOC) de YOFC PERÚ S.A.C.

El trabajo se enmarca dentro de un proceso de intervención tecnológica estructurada, que comprende cinco fases:

1. Diagnóstico técnico del modelo de monitoreo existente.
2. Identificación de brechas operativas respecto a buenas prácticas de gestión de redes.
3. Diseño de una arquitectura de monitoreo centralizada.
4. Implementación progresiva de la solución.
5. Evaluación del impacto en indicadores operativos.

El enfoque adoptado es descriptivo y transformacional, ya que implicó la modificación directa del entorno operativo, incorporando: automatización, integración de plataformas y estructuración lógica de supervisión.

La metodología se apoyó en fundamentos teóricos provenientes de la gestión de: redes, administración de servicios TI y modelos de desempeño en telecomunicaciones.

3.4.1 Bases teóricas

1. Gestión de Redes de Telecomunicaciones

La gestión de redes es un conjunto de procesos orientados a asegurar la disponibilidad, rendimiento, confiabilidad y seguridad de una infraestructura de telecomunicaciones. Según *Stallings (2018)*, la administración de redes comprende cinco áreas funcionales conocidas como "modelo FCAPS":

- Fault Management (Gestión de fallas)
- Configuration Management (Gestión de configuración)
- Accounting Management (Gestión de contabilidad)
- Performance Management (Gestión de desempeño)
- Security Management (Gestión de seguridad)

Dentro de este modelo, la gestión de fallas y la gestión de desempeño son fundamentales en la operación de un NOC, ya que permiten detectar eventos anómalos, correlacionar alarmas y mantener continuidad del servicio.

Oppenheimer (2011) señala que una red empresarial distribuida requiere mecanismos de supervisión proactiva capaces de identificar degradaciones antes de que se conviertan en interrupciones críticas. La falta de visibilidad centralizada incrementa la probabilidad de fallas no detectadas oportunamente.

En este sentido, la problemática identificada en el entorno del NOC evidenciaba una brecha respecto a los principios teóricos de gestión estructurada de redes.

La clasificación funcional FCAPS es ampliamente mencionada en la literatura técnica y ha sido formalizada por la Unión Internacional de Telecomunicaciones (ITU-T) en la recomendación M.3400, donde se establecen los lineamientos para la gestión integrada de redes de telecomunicaciones. Este modelo proporciona un marco estructurado para la administración de fallas, configuración, contabilidad, desempeño y seguridad, y asume que la gestión de fallas (Fault Management) es más efectiva cuando está automatizada. Entonces, ¿qué sucede cuando la automatización se implementa de manera incremental? El caso YOFC permite evaluar el impacto de una implementación por fases (ICMP → SNMPv3) sobre la efectividad de la gestión de fallas.

2. Network Operations Center (NOC) y Supervisión Continua

El Network Operations Center es la unidad central de supervisión en organizaciones de telecomunicaciones. Según *García-Zaballos et al. (2022)*, un NOC debe garantizar:

- Monitoreo en tiempo real
- Respuesta estructurada ante incidentes
- Escalamiento definido
- Registro histórico de eventos

- Análisis de tendencias

La literatura técnica coincide en que la efectividad de un NOC depende directamente de la calidad de las herramientas de monitoreo implementadas. Un entorno fragmentado reduce la capacidad de correlación de eventos y afecta la eficiencia del personal operativo.

Un NOC moderno no solo observa alarmas; interpreta patrones, anticipa riesgos y mantiene indicadores de desempeño medibles.

3. Gestión de Incidentes bajo el Marco ITIL

El marco ITIL (*Axelos, 2019*) define la gestión de incidentes como el proceso cuyo objetivo es restaurar el servicio normal lo más rápido posible, minimizando el impacto en el negocio.

En este contexto, se identifican métricas fundamentales:

- MTTD (Mean Time to Detect), Tiempo medio para detectar.
- MTTR (Mean Time to Repair), Tiempo medio de reparación.
- Disponibilidad porcentual del servicio

La reducción del MTTD depende de sistemas automatizados de detección. Cuando la identificación de fallas es manual, el tiempo promedio de detección es significativo.

La implementación de plataformas de monitoreo centralizadas se alinea directamente con las recomendaciones ITIL para mejorar la eficiencia operativa.

(Axelos, 2019) sostiene que la reducción del MTTD es clave para la gestión de incidentes. El caso YOFC permite contrastar empíricamente esta afirmación, respondiendo a la pregunta: ¿qué factores (automatización, segmentación regional, notificación) contribuyen más a la reducción del MTTD en un contexto real?

4. SLA y Métricas de Disponibilidad

Los Acuerdos de Nivel de Servicio (SLA) establecen compromisos formales respecto a disponibilidad, latencia y continuidad. Según *Hiles (2010)*, la medición objetiva de

disponibilidad requiere herramientas capaces de registrar eventos de caída y recuperación con precisión temporal.

La disponibilidad se expresa comúnmente como:

Disponibilidad (%) = (Tiempo total – Tiempo de indisponibilidad) / Tiempo total × 100

Sin sistemas automatizados de monitoreo, la medición de estos indicadores carece de confiabilidad estadística.

5. Sistemas de apoyo operativo OSS en Telecomunicaciones

Los OSS (Operations Support Systems) son plataformas que permiten administrar redes, servicios y recursos. Sin embargo, muchos sistemas propietarios priorizan la configuración sobre la visualización ejecutiva.

Según *Camarillo y García-Martín (2014)*, la evolución de redes IP/MPLS exige herramientas complementarias que permitan correlación de eventos y visualización estratégica.

La integración de soluciones open source como Zabbix responde a esta necesidad de complementar los OSS tradicionales con monitoreo dinámico.

6. Arquitectura de Monitoreo y Protocolos

El monitoreo de red se apoya en protocolos como: **SNMP (Simple Network Management Protocol)** Protocolo simple de gestión de red.

Definido en RFC 1157 y actualizado en RFC 3411–3418, permite recopilar información estructurada de dispositivos de red.

El protocolo SNMP ha evolucionado hacia versiones más seguras como el SNMPv3, definido en el conjunto de estándares RFC 3411–3418, que introducen mecanismos de autenticación y cifrado que permiten su implementación en entornos empresariales críticos. Esta evolución responde a la necesidad de garantizar la integridad y confidencialidad de la información de monitoreo en infraestructuras distribuidas.

ICMP (Internet Control Message Protocol) Protocolo de mensajes de control de red

Definido en RFC 792, permite verificar conectividad y disponibilidad mediante eco-respuesta. Según *Comer (2018)*, la combinación de protocolos de sondeo y generación de traps permite establecer esquemas de supervisión híbridos.

Técnicamente, el monitoreo basado en ICMP presenta ventajas en términos de simplicidad y bajo impacto en la red, ya que permite verificar la disponibilidad de los dispositivos sin requerir configuración adicional. Sin embargo, su principal limitación radica en la falta de visibilidad sobre el estado interno de los equipos.

En contraste, el monitoreo mediante SNMP permite acceder a métricas detalladas como: uso de CPU, consumo de memoria, estado de interfaces y estadísticas de tráfico, lo que resulta fundamental para la gestión de desempeño en redes MPLS. Según *Stallings (2018)*, los sistemas de monitoreo que combinan mecanismos de sondeo básico (ICMP) con recolección de métricas estructuradas (SNMP) ofrecen un enfoque más completo para la supervisión de redes empresariales.

En este sentido, la implementación desarrollada en el presente trabajo utilizó ICMP como primera capa de detección de disponibilidad de equipos, complementada posteriormente con monitoreo SNMP para la supervisión del estado de enlaces de transporte, quedando abierta para fases futuras la incorporación de métricas de desempeño de dispositivo (CPU, memoria, tráfico de interfaces).

7. Plataformas de código abierto (Open Source) en Entornos Críticos

El uso de software open source en telecomunicaciones ha crecido debido a su flexibilidad, escalabilidad y reducción de costos. Sin embargo, la literatura sobre adopción de open source en telecomunicaciones es escasa. El caso YOFC permite explorar los trade-offs entre costo (cero licencias) y complejidad de implementación (necesidad de scripts personalizados, ajustes de TimescaleDB).

Zabbix (*Zabbix LLC, 2023a*) es una plataforma empresarial que permite:

- Monitoreo distribuido
- Alertamiento automatizado

- Paneles (Dashboards) personalizados
- Escalabilidad horizontal

Su arquitectura cliente-servidor y su capacidad de integración con API (Application Programming Interface) o Interfaz de programación de aplicaciones, la hacen adecuada para entornos multicapa.

En el contexto de la gestión de redes, existen diversas herramientas de monitoreo ampliamente utilizadas tanto en entornos empresariales como en operadores de telecomunicaciones. La selección de una plataforma adecuada depende de múltiples factores, incluyendo costos, escalabilidad, facilidad de implementación y capacidades de integración. Las principales características de estas herramientas se resumen en la Tabla 1.

Tabla 1 **Comparación de herramientas de monitoreo de red**

Herramienta	Licenciamiento	Escalabilidad	SNMP/API
Zabbix	Código abierto	Alta	Completo
Nagios Core	Código abierto	Media	Completo (plugins)
PRTG	Propietario	Media	Completo
SolarWinds	Propietario	Alta	Completo

Fuente. Elaboración propia, basada en Stallings (2018), Mongkolluksamee et al. (2010) y documentación oficial.

El análisis comparativo evidencia que Zabbix ofrece un balance entre costo, escalabilidad y capacidades técnicas, destacando por su modelo open source y su capacidad de integración con múltiples protocolos. A diferencia de soluciones propietarias como PRTG o SolarWinds, Zabbix permite una implementación sin costos de licenciamiento, lo cual resulta especialmente relevante en entornos de gran

escala.

En comparación con Nagios, Zabbix presenta además una mayor integración nativa y menor dependencia de configuraciones externas, lo que facilita su administración en entornos operativos complejos como un NOC.

Estado del Arte

En el ámbito de la gestión de redes de telecomunicaciones, la evolución de los sistemas de monitoreo ha experimentado una transición significativa desde modelos tradicionales basados en herramientas aisladas hacia plataformas integradas capaces de proporcionar visibilidad en tiempo real, automatización de alertas y análisis histórico de eventos. Esta evolución responde directamente a la creciente complejidad de las infraestructuras modernas, especialmente en entornos IP/MPLS distribuidos geográficamente, donde la supervisión eficiente constituye un factor crítico para garantizar la disponibilidad del servicio.

Diversos estudios en el área de gestión de redes coinciden en que la implementación de soluciones de monitoreo centralizado en centros de operaciones de red (NOC) permite mejorar la capacidad de detección de incidentes y reducir los tiempos de respuesta operativa *(Narayan et al. 2023)*. Ello es consistente con lo señalado en la literatura técnica, donde la transición de modelos reactivos hacia enfoques proactivos de monitoreo representa una tendencia consolidada en la industria de las telecomunicaciones (*Stallings, 2018)*.

En cuanto a las herramientas disponibles, Zabbix, Nagios y PRTG Network Monitor se posicionan como algunas de las soluciones más utilizadas en entornos empresariales. Zabbix se destaca por su arquitectura escalable, su capacidad de integración con múltiples tecnologías y su soporte para diferentes métodos de monitoreo, incluyendo ICMP, SNMP y agentes. Estas características permiten consolidar la información en Dashboard personalizados y automatizar la generación de alertas, lo cual resulta fundamental en entornos donde la supervisión debe ser continua y centralizada.

Por su parte, Nagios ha sido ampliamente adoptado debido a su flexibilidad basada en plugins; sin embargo, en entornos de gran escala su configuración puede volverse

compleja y requerir una mayor intervención manual para la integración de múltiples componentes, dado que la flexibilidad que lo caracteriza se traduce en procesos de configuración extensos que demandan experiencia técnica especializada *(Mongkolluksamee et al., 2010)*. En contraste, PRTG Network Monitor ofrece una implementación más sencilla y una interfaz gráfica intuitiva, pero su modelo de licenciamiento puede representar una limitación en infraestructuras extensas, particularmente en organizaciones con crecimiento progresivo del número de dispositivos monitoreados.

La literatura técnica coincide en que las soluciones de monitoreo más eficientes son aquellas que integran múltiples capas de supervisión. Bajo esa lógica, el monitoreo basado en ICMP permite verificar la disponibilidad de los dispositivos, mientras que protocolos como SNMP proporcionan métricas detalladas relacionadas con el desempeño de la red, tales como uso de CPU, consumo de memoria y tráfico en interfaces. Según *Stallings (2018)*, la combinación de estos enfoques permite construir sistemas de monitoreo más robustos, capaces de detectar fallas y anticipar comportamientos anómalos mediante el análisis de tendencias.

En línea con lo anterior, los marcos de referencia en gestión de servicios de tecnologías de la información, como ITIL *(Axelos, 2019)*, enfatizan la importancia de implementar sistemas de monitoreo proactivos que permitan identificar incidentes antes de que estos impacten a los usuarios finales. Este enfoque se encuentra alineado con el modelo FCAPS definido por la *ITU-T (2012)*, en el cual la gestión de fallas (Fault Management) es uno de los pilares fundamentales para garantizar la continuidad del servicio.

En términos de resultados, diversas implementaciones de sistemas de monitoreo centralizado han reportado mejoras significativas en los tiempos de detección de incidentes, reduciendo el Mean Time to Detect (MTTD) desde intervalos de varios minutos en entornos manuales hasta valores cercanos a tiempo real en sistemas automatizados. Esta reducción del MTTD se traduce directamente en una mayor disponibilidad del servicio y en una mejora en el cumplimiento de acuerdos de nivel de servicio (SLA).

En este contexto, la implementación de Zabbix desarrollada en el presente trabajo se alinea con las tendencias actuales en monitoreo de redes, al proponer una solución

centralizada, automatizada y escalable que integra diferentes componentes de la infraestructura tecnológica, incluyendo routers, sistemas de energía y seguridad física. Esta integración mejora la visibilidad operativa, optimiza los tiempos de detección de incidentes y fortalece la gestión del NOC, posicionando la solución dentro de las buenas prácticas internacionales en gestión de redes de telecomunicaciones en concordancia con los lineamientos del modelo *FCAPS (ITU-T, 2012)* y los principios de gestión de incidentes propuestos por ITIL *(Axelos, 2019).*

La adopción de Zabbix en operadores de telecomunicaciones de gran escala respalda empíricamente estas tendencias. Hrvatski Telekom, filial de Deutsche Telekom en Croacia, opera una de las implementaciones de mayor envergadura documentadas hasta la fecha: cerca de 17,000 hosts y 15 millones de ítems monitoreados, con 800,000 triggers activos y un flujo de 32,000 nuevos valores por segundo, soportado por una arquitectura distribuida de 30 servidores proxy y una base de datos de 4 TB optimizada con TimescaleDB que alcanza una compresión cercana al 75% (Osmanagić, 2025). LMT, el operador móvil más grande de Letonia, con más de 1.5 millones de usuarios y 1,660 estaciones base, opera múltiples instancias de Zabbix (versiones 5.0 y 7.0) para supervisar más de 3,000 hosts y cerca de 500,000 ítems, combinando bases de datos MySQL, PostgreSQL y TimescaleDB (Zabbix, 2025). Por su parte, O2 Czech Republic, el mayor operador de telecomunicaciones de la República Checa, reemplazó su plataforma empresarial IBM Netcool por Zabbix tras un proyecto ejecutado en pocos meses, logrando una migración completa que redujo costos de licenciamiento sin sacrificar la capacidad de procesamiento de eventos que exige una red de esa escala (initMAX, 2025a).

Estos tres casos, aunque operan en órdenes de magnitud considerablemente mayores a los del presente trabajo, coinciden en un punto: ninguno reporta públicamente una cifra de MTTD comparable a la obtenida aquí. En el ámbito nacional, (Abanto Taipe & Pareja Abarca, 2025) documentan en su Trabajo de Suficiencia Profesional la implementación de una plataforma de monitoreo proactivo con Zabbix orientada a reducir el tiempo de inactividad en el NOC de la empresa peruana Pronet System SAC, tampoco con una cifra específica de MTTD. Otros trabajos han documentado previamente la viabilidad técnica de esta integración; Mohd Faris Mohd Fuzi (2022) describe una implementación de monitoreo de red con Zabbix y notificaciones push vía Telegram, confirmando la madurez y adopción de este mecanismo de alertamiento

en distintos contextos operativos. En un caso distinto, la integración de Zabbix con Telegram documentada por Rahayu et al. (2025), en un caso aplicado a la supervisión de infraestructura escolar, reporta una reducción del tiempo de reconocimiento de incidentes (MTTA) de 1-2 horas a 10-20 minutos, con una ventana de detección promedio de 2 minutos 30 segundos y notificación de recuperación en 20 segundos evidencia adicional de que la automatización del canal de notificación, más que la sola centralización de la información, es lo que impacta directamente los tiempos de respuesta.

A diferencia de los casos de mayor escala revisados, que no publican una cifra de MTTD, el presente trabajo aporta una medición concreta: 182 segundos de ventana de detección y 3.1 segundos de tiempo de notificación, obtenidos sobre 30 eventos reales de producción (Tabla 20 y Tabla 27). Esta cuantificación explícita, poco frecuente en la literatura de casos consultada, constituye en sí misma un aporte diferencial del presente trabajo frente al estado del arte revisado.

8. Monitoreo de Interfaces de Red mediante el estándar IF-MIB

El estándar IF-MIB (Interfaces Group MIB), definido en el *RFC 2863 (McCloghrie & Kastenholz, 2000)*, especifica una tabla de objetos que describe cada interfaz de red presente en un dispositivo SNMP. Los objetos más relevantes para la supervisión del estado de enlaces son ifDescr, que contiene la descripción textual de la interfaz; ifAdminStatus, que indica el estado configurado por el operador (1=up, 2=down); e ifOperStatus, que refleja el estado operativo real del enlace físico (1=up, 2=down). La combinación de ambos estados permite distinguir entre un enlace caído de forma no intencional (Admin=up, Oper=down, condición de falla real) y un puerto deshabilitado deliberadamente por el operador (Admin=down), lo cual resulta fundamental para evitar falsas alarmas en entornos operativos.

Esta distinción tiene valor práctico significativo en redes de transporte gestionadas mediante plataformas OSS como Nokia NSP NFM-P, donde el cambio de color de un enlace en la topología puede indicar una condición de alarma sin especificar si el puerto fue inhabilitado intencionalmente o perdió conectividad por una falla física. El monitoreo directo de ifAdminStatus e ifOperStatus mediante SNMP proporciona una visibilidad más precisa y diferenciada del estado de cada enlace.

9. Low-Level Discovery (LLD) en Zabbix

El mecanismo de Low-Level Discovery (LLD) o Descubrimiento a bajo nivel de Zabbix, permite hallar automáticamente entidades dentro de un host supervisado, tales como interfaces de red, sistemas de archivos o procesos, y generar de forma dinámica los ítems, triggers (desencadenantes) y gráficos correspondientes a cada entidad descubierta, sin intervención manual por dispositivo *(Zabbix LLC, 2023a)*.

El flujo de procesamiento de una "regla de descubrimiento" (Discovery Rule) con base en un walk SNMP permite extraer listas completas de datos y comprende tres etapas principales: un Master Item realiza una consulta walk sobre la tabla SNMP de interés y almacena el resultado completo; la Discovery Rule aplica un preprocesamiento que transforma ese resultado en un arreglo JSON con una entrada por entidad descubierta; y los Item Prototypes y Trigger Prototypes definen la lógica de monitoreo que se instancia automáticamente para cada entidad que supera los filtros configurados. Este enfoque reduce significativamente la carga de configuración en redes de gran escala, donde la creación manual de ítems por cada interfaz de cada dispositivo resultaría impracticable *(Zabbix LLC, 2023a)*.

10. La extensión de código abierto TimescaleDB y la Compresión de Series Temporales

TimescaleDB es una extensión de código abierto para PostgreSQL especializada en el almacenamiento y consulta eficiente de datos de series temporales *(Timescale Inc., 2024a)*. Su arquitectura se basa en el concepto de hypertables, que son tablas particionadas automáticamente en bloques temporales denominados chunks, permitiendo que las consultas acotadas a un intervalo de tiempo accedan únicamente a los chunks relevantes en lugar de escanear toda la tabla.

Adicionalmente, TimescaleDB incorpora un mecanismo de compresión columnar nativa que, mediante políticas configurables, convierte automáticamente los chunks más antiguos de un formato de fila a un formato columnar comprimido. En datos de monitoreo de red, donde un mismo valor puede permanecer constante durante largos periodos (por ejemplo, una interfaz que permanece en estado operativo durante días), este mecanismo logra tasas de reducción del orden del 90 al 95 por ciento del espacio en disco *(Timescale Inc., 2024a)*. Esta capacidad resulta especialmente relevante en

entornos con alta tasa de ingesta de datos, donde el crecimiento sin control del historial puede comprometer la disponibilidad del servidor de monitoreo.

Síntesis: Proposiciones teóricas orientadoras del estudio de caso

La revisión de literatura precedente presenta los fundamentos conceptuales de FCAPS, ITIL, NOC, SNMP/ICMP, LLD y TimescaleDB de manera individual. Sin embargo, estos marcos no explicitan por sí solos qué combinación de factores resulta determinante para que una organización de telecomunicaciones logre efectivamente la transición de un modelo de monitoreo reactivo a uno proactivo, pregunta central del presente trabajo (sección 3.3.1). A partir de la problematización de dichos marcos, se formulan las siguientes proposiciones teóricas, que el caso de YOFC PERÚ S.A.C. permite evaluar:

Proposición N°1. La centralización de la gestión de fallas (Fault Management, dentro del modelo FCAPS) en una plataforma única que constituye una condición necesaria, pero no suficiente para reducir la fragmentación informativa en un NOC que administra infraestructura distribuida geográficamente (Stallings, 2018; Oppenheimer, 2011).

Proposición N°2. La reducción del tiempo promedio de detección de incidentes (MTTD), enfatizada por ITIL como factor crítico de la gestión de incidentes (Axelos, 2019), depende principalmente de la sustitución de la revisión manual por mecanismos de alertamiento automatizado, más que de la sola centralización de la información en un tablero único.

Proposición N°3. La segmentación jerárquica por criterios geográficos y funcionales, recomendada por Oppenheimer (2011) para arquitecturas de monitoreo empresarial, mejora la escalabilidad y la trazabilidad operativa en redes de transporte distribuidas, en comparación con esquemas de alertamiento no diferenciados.

Proposición N°4. La combinación complementaria de mecanismos de sondeo básico (ICMP) y de recolección estructurada de métricas por dispositivo (SNMP) señalada por Stallings (2018), ofrece mayor capacidad de diagnóstico diferenciado distinguiendo indisponibilidad total de degradación parcial de un enlace que el uso

exclusivo de un único mecanismo de sondeo.

3.4.2 Marco conceptual

El marco conceptual del presente trabajo se fundamenta en categorías técnicas propias de la gestión de redes de telecomunicaciones, administración de servicios TI y supervisión operativa de infraestructuras distribuidas. A continuación, se desarrollan los conceptos centrales que sustentan la implementación realizada.

1. Network Operations Center (NOC)

El Network Operations Center (NOC) es la unidad organizacional responsable de la supervisión, control y gestión continua de una infraestructura de telecomunicaciones. Según *García-Zaballos et al. (2022)*, un NOC tiene como objetivo garantizar la disponibilidad del servicio mediante la monitorización en tiempo real, el análisis de eventos y la respuesta estructurada ante incidentes.

Desde la perspectiva de la ingeniería de redes, el NOC cumple una función reactiva ante fallas y debe operar bajo un enfoque preventivo y proactivo, apoyado en herramientas de supervisión centralizadas capaces de generar visibilidad transversal de la infraestructura.

En el contexto del presente trabajo, el NOC constituye el entorno operativo donde se evidenció la necesidad de optimizar la detección temprana de incidentes.

2. Gestión de Incidentes

La gestión de incidentes es definida por el marco ITIL como el proceso orientado a restaurar la operación normal del servicio lo más rápido posible, minimizando el impacto negativo en el negocio *(Axelos, 2019).*

Un incidente se entiende como cualquier evento que interrumpe o reduce la calidad de un servicio. La eficiencia de la gestión de incidentes depende directamente de la capacidad de detección automática, correlación de eventos y escalamiento oportuno.

La implementación de sistemas de monitoreo centralizados impacta directamente en

la reducción del MTTD (Mean Time to Detect) y el MTTR (Mean Time to Repair), indicadores clave en la operación de redes críticas.

3. Disponibilidad del Servicio

La disponibilidad es un indicador de desempeño que mide el porcentaje de tiempo durante el cual un sistema permanece operativo y accesible. Según *Stallings (2018)*, en redes empresariales la disponibilidad es uno de los principales parámetros de evaluación de calidad del servicio.

La fórmula comúnmente aceptada para su cálculo es:

Disponibilidad (%) = (Tiempo total – Tiempo de indisponibilidad) / Tiempo total × 100

En entornos de telecomunicaciones, niveles de disponibilidad superiores al 99.5% son considerados estándares mínimos en servicios críticos.

La ausencia de mecanismos automatizados de monitoreo compromete la precisión en la medición de este indicador.

4. Service Level Agreement (SLA)

Un Service Level Agreement (SLA) es un acuerdo formal que define los niveles de servicio comprometidos entre proveedor y cliente. *Hiles (2010)* sostiene que los SLA deben basarse en métricas objetivas y verificables, como disponibilidad, latencia y tiempos de respuesta ante incidentes.

El cumplimiento de SLA requiere herramientas que registren eventos de caída y recuperación con precisión temporal y capacidad de auditoría.

La implementación de Zabbix contribuye a la generación de evidencia técnica necesaria para respaldar dichos compromisos contractuales.

5. Gestión de Fallas (Fault Management)

Dentro del modelo FCAPS propuesto por ISO y desarrollado por *Stallings (2018)*, la gestión de fallas tiene como objetivo detectar, aislar y corregir anomalías en la red.

Este proceso implica:

- Identificación del evento
- Notificación automática
- Registro histórico
- Análisis de causa raíz

En ausencia de una plataforma centralizada, el proceso de fault management se vuelve manual y dependiente de la revisión humana.

6. Monitoreo Proactivo

El monitoreo proactivo se define como la supervisión continúa orientada a identificar condiciones anómalas antes de que generen interrupciones del servicio. *Oppenheimer (2011)* destaca que las redes empresariales modernas requieren monitoreo preventivo basado en métricas y alertas automatizadas.

El modelo proactivo reduce la probabilidad de fallas catastróficas y mejora la estabilidad operativa.

7. Arquitectura Multicapa de Red

Las redes IP/MPLS suelen estructurarse en capas funcionales:

- Capa de acceso
- Capa de distribución
- Capa de transporte o core

Según *Comer (2018)*, la segmentación en capas permite optimizar el rendimiento y la escalabilidad, pero también exige esquemas de monitoreo organizados lógicamente para evitar ambigüedades en la detección de fallas.

La arquitectura multicapa del entorno analizado requirió una organización estructurada dentro de Zabbix.

8. SNMP (Simple Network Management Protocol)

SNMP es un protocolo de aplicación definido inicialmente en el RFC 1157 y posteriormente actualizado en RFC 3411–3418. Permite la recolección estructurada de métricas desde dispositivos de red mediante el uso de agentes y objetos MIB (Management Information Base).

Comer (2018) señala que SNMP constituye el estándar predominante para la supervisión de dispositivos IP en redes empresariales.

9. ICMP (Internet Control Message Protocol)

ICMP, definido en el RFC 792, es utilizado para verificar conectividad y disponibilidad mediante mensajes de eco (ping).

En sistemas de monitoreo, ICMP se emplea como mecanismo básico de verificación de estado activo/inactivo.

10. Plataforma Open Source

El software open source se caracteriza por permitir acceso al código fuente, facilitando personalización y escalabilidad. Según *Zabbix LLC (2023a)*, las plataformas open source empresariales permiten reducir costos de licenciamiento sin comprometer funcionalidad avanzada.

En entornos donde la escalabilidad y flexibilidad son requerimientos estratégicos, este modelo ofrece ventajas competitivas.

11. Estado Administrativo y Operativo de Interfaces (ifAdminStatus / ifOperStatus)

En el contexto del monitoreo de redes basado en SNMP, el estado administrativo (ifAdminStatus) representa la intención configurada por el operador sobre una interfaz de red, mientras que el estado operativo (ifOperStatus) refleja la condición real del enlace físico en un momento dado *(McCloghrie & Kastenholz, 2000)*.

La evaluación conjunta de ambos parámetros permite clasificar el estado de un enlace en tres condiciones operativas relevantes: enlace activo (Admin=up, Oper=up), falla

real de enlace (Admin=up, Oper=down) y puerto deshabilitado intencionalmente (Admin=down).

12. Hypertable

Una hypertable es una abstracción de almacenamiento implementada por TimescaleDB sobre PostgreSQL, que organiza los datos de una tabla temporal en particiones automáticas basadas en intervalos de tiempo denominadas chunks.

Esta estructura permite que las operaciones de lectura, escritura y eliminación de datos sean significativamente más eficientes que sobre una tabla convencional de gran tamaño, al limitar el acceso a los chunks correspondientes al intervalo temporal consultado *(Timescale Inc., 2024a).*

3.5 Procedimiento

El procedimiento seguido para la implementación del sistema de monitoreo Zabbix en el Network Operations Center (NOC) de YOFC PERÚ S.A.C. se desarrolló bajo un enfoque estructurado de ingeniería, basado en el análisis del entorno operativo existente, la identificación de brechas técnicas y la planificación progresiva de una arquitectura centralizada de supervisión.

Desde el aspecto metodológico, el procedimiento adoptó principios propios de la gestión de infraestructura tecnológica, considerando aspectos de disponibilidad, escalabilidad, eficiencia operativa y alineación con buenas prácticas de monitoreo establecidas en la literatura técnica *(Stallings, 2018; Axelos, 2019).*

El proceso se estructuró en cuatro etapas principales:

1. Diagnóstico del entorno pre-implementación.
2. Planificación y diseño de la solución.
3. Implementación técnica y configuración.
4. Organización estructural y optimización operativa.

Cada etapa fue desarrollada procurando no afectar la continuidad operativa del NOC, dado que la red supervisada se encontraba en funcionamiento permanente.

3.5.1 Diagnóstico del Entorno Pre-implementación

Antes de la implementación de Zabbix, la supervisión de la infraestructura se realizaba mediante el sistema NSP NFM-P de Nokia, complementado con herramientas independientes para la gestión de sistemas de energía y seguridad física. Si bien estas plataformas permitían la administración técnica de los dispositivos, el análisis del entorno evidenció limitaciones operativas que afectaban la eficiencia del proceso de monitoreo. Estas limitaciones se sintetizan en la Tabla 2.

Tabla 2 **Limitaciones identificadas**

Dimensión	Situación observada	Impacto operativo
Fragmentación	Existencia de 8 plataformas separadas (4 regiones × 2 redes)	Alta carga cognitiva para el operador
Supervisión energética	Monitoreo realizado en sistemas independientes	Falta de correlación con eventos de red
Alertamiento	Dependencia de filtrado manual de alarmas	Incremento del tiempo de detección (MTTD)
Visualización	Ausencia de Dashboard consolidados	Baja visibilidad global de la infraestructura
Análisis histórico	Información distribuida en múltiples plataformas	Dificultad para auditoría y seguimiento

Fuente. Elaboración propia.

La infraestructura estaba compuesta por:

- 4 regiones operativas
- 2 subredes por región (Transporte y Acceso)
- Sistemas de energía asociados
- Sistemas de seguridad física

Cada región requería acceso independiente a su plataforma Nokia correspondiente, lo que implicaba múltiples sesiones activas y consumo elevado de recursos de estación de trabajo.

Desde la perspectiva de gestión de redes (modelo FCAPS), la organización presentaba debilidades particularmente en:

- Gestión de fallos
- Gestión de rendimiento

Esto se traducía en un modelo predominantemente reactivo

Con el propósito de representar gráficamente la estructura de supervisión existente antes de la implementación de Zabbix, se elaboró un esquema conceptual que resume la arquitectura de monitoreo fragmentada del NOC.

Como se observa en la Figura 1, cada región operativa (San Martin, Arequipa, La Libertad y Ancash) contaba con plataformas independientes para la gestión y visualización de la disponibilidad de los routers de las redes de transporte y acceso mediante el sistema NSP NFM-P de Nokia.

Paralelamente, la disponibilidad de los sistemas de energía y seguridad física eran supervisados mediante herramientas distintas, sin integración ni correlación centralizada de eventos.

Esta disposición generaba una estructura de monitoreo distribuida y desarticulada, donde la información se encontraba dispersa en múltiples plataformas, dificultando la obtención de una visión global del estado de la infraestructura.

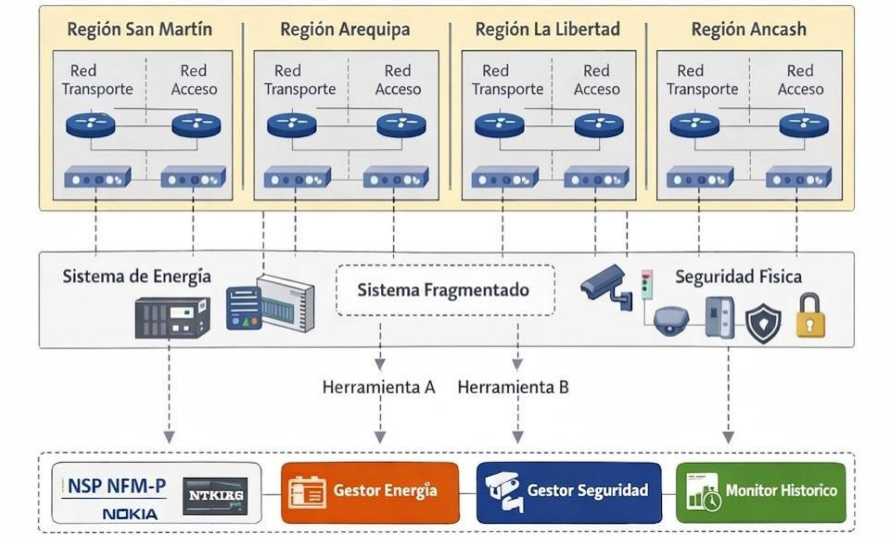


Figura 1 **Arquitectura de supervisión (modelo fragmentado)**. Fuente. Elaboración propia

3.5.2 Planificación de la Implementación

La planificación se desarrolló siguiendo principios de diseño de arquitectura de monitoreo recomendados en entornos empresariales *(Oppenheimer, 2011)*, priorizando:

- Centralización
- Escalabilidad
- Modularidad
- Bajo impacto en producción.

3.5.2.1 Definición de Requerimientos. Se establecieron requerimientos funcionales y no funcionales.

Requerimientos Funcionales

- Monitoreo ICMP de routers Nokia

- Supervisión de gestores auxiliares
- Monitoreo de disponibilidad de equipos de energía
- Monitoreo de dispositivos de seguridad física
- Generación de alertas automatizadas vía Telegram
- Dashboards regionales personalizados
- Monitoreo SNMP de enlaces de transporte mediante descubrimiento automático (Low-Level Discovery)

Requerimientos No Funcionales

- Escalabilidad
- Bajo consumo de recursos
- Alta disponibilidad
- Facilidad de administración

3.5.2.2 Selección de Tecnologías. La selección tecnológica se realizó bajo criterios de:

- Estabilidad
- Comunidad de soporte
- Documentación técnica
- Compatibilidad.

Considerando estos criterios, en la Tabla 3 se presenta la matriz de selección de tecnología.

Tabla 3 **Selección de Tecnología**

Componente	Tecnología Seleccionada	Justificación Técnica
Plataforma de monitoreo	Zabbix 7.2	Escalable, open source
Sistema operativo	Ubuntu 22.04 LTS	Soporte extendido
Base de datos	PostgreSQL	Estabilidad y rendimiento
Servidor web	Apache	Compatibilidad comprobada

Fuente. Elaboración propia.

Zabbix 7.2 fue seleccionado por su arquitectura cliente-servidor, soporte SNMP, ICMP y API, además de su capacidad de escalamiento horizontal *(Zabbix LLC, 2023a)*.

3.5.2.3 Dimensionamiento de la Infraestructura. El dimensionamiento consideró:

- Número de hosts proyectados
- Intervalo de polling
- Retención histórica
- Crecimiento estimado

Se realizó cálculo estimado de carga según el criterio de estimación en las recomendaciones de dimensionamiento para entornos de producción publicadas en la documentación oficial de Zabbix *(Zabbix LLC, 2023a):*

Carga = Número de hosts × Intervalo de chequeo × Métricas generadas

El servidor principal de la empresa cumplía con los requerimientos de CPU, RAM y almacenamiento para soportar la carga estimada en la fase inicial de monitoreo ICMP.

Sin embargo, al escalar la solución mediante monitoreo SNMP a 1,493 hosts el dimensionamiento original resultó insuficiente, lo que motivó ajustes posteriores en la configuración de memoria caché y en la capacidad de almacenamiento del servidor, documentados más adelante.

3.5.2.4 Diseño de la Estructura de Monitoreo

El diseño de la estructura de monitoreo constituyó una etapa fundamental en la planificación de la solución, ya que una organización lógica inadecuada puede limitar la escalabilidad del sistema y dificultar la operación diaria del NOC.

De acuerdo con *Oppenheimer (2011)*, una arquitectura de monitoreo efectiva debe reflejar la topología real de la red y la estructura organizacional de la operación, permitiendo una correlación clara entre eventos técnicos y su ubicación lógica.

En función de esta premisa, se definió una estructura jerárquica basada en dos criterios principales: segmentación regional y clasificación por tipo de componente.

Esta decisión respondió a la necesidad de mantener coherencia entre la organización física de la red y su representación lógica dentro de la plataforma de monitoreo.

La estructura definida consideró:

- Cuatro regiones operativas, alineadas con la distribución geográfica de la infraestructura.
- Dos redes por región: acceso y transporte, diferenciadas por su función dentro de la arquitectura IP/MPLS.
- Cuatro categorías de componentes por cada región y red: Componentes de Red (routers), Gestores Auxiliares, Componentes de Energía y Componentes de Seguridad Física

Esta organización permitió:

- Facilitar la visualización regional de eventos.
- Aplicar políticas de alertamiento diferenciadas.

- Simplificar la administración de hosts.
- Preparar el sistema para futuras expansiones de red.

Desde la perspectiva del modelo FCAPS *(Stallings, 2018)*, esta estructuración fortalece la gestión de fallas y la gestión de desempeño, al permitir identificar rápidamente el contexto operativo de cada evento.

3.5.2.5 Estrategia de Despliegue. La estrategia de despliegue del sistema de monitoreo fue diseñada bajo un enfoque progresivo y controlado, con el objetivo de minimizar riesgos operativos y garantizar la continuidad de los servicios supervisados. Según las buenas prácticas de gestión de servicios TI establecidas por ITIL (Axelos, 2019), la implementación de cambios en entornos productivos debe realizarse de manera gradual, validando cada etapa antes de avanzar a la siguiente.

Bajo este enfoque, el despliegue se estructuró en las siguientes fases secuenciales:

1. Instalación del servidor Zabbix y configuración de la base de datos.
2. Implementación de la interfaz web y verificación de conectividad interna.
3. Creación de grupos de hosts y aplicación de plantillas base.
4. Carga masiva de hosts mediante archivos JSON.
5. Integración del sistema de notificaciones con Telegram.
6. Desarrollo y validación de Dashboard regionales.
7. Ampliación del monitoreo mediante SNMPv3 y descubrimiento automático (LLD) para enlaces de transporte, incluyendo la optimización de la base de datos con TimescaleDB

Esta estrategia permitió:

- Validar funcionalidad por etapas.
- Identificar y corregir errores tempranamente.

- Evitar interrupciones en la operación del NOC.
- Facilitar la adaptación progresiva del personal al nuevo sistema.
- Mecanismo de mantenimiento automatizado integrado a la plataforma con TimescaleDB.

3.5.2.6 Consideraciones de Seguridad y Acceso. La configuración de seguridad y control de accesos fue considerada un componente crítico del diseño del sistema, dado que el NOC opera con información sensible relacionada con la infraestructura de telecomunicaciones.

De acuerdo con la norma ISO/IEC 27001 (ISO, 2013), el control de acceso debe basarse en el principio de privilegio mínimo, garantizando que cada usuario tenga acceso únicamente a la información necesaria para el desempeño de sus funciones.

En este sentido, se definió una política de usuarios y permisos que contempló:

- Acceso regional restringido para operadores locales.
- Visualización exclusiva de dashboards correspondientes a cada región.
- Acceso administrativo global reservado para el administrador central del sistema.
- Separación entre usuarios operativos y usuarios de administración.

Esta configuración permitió:

- Proteger la información crítica.
- Evitar modificaciones no autorizadas.
- Mantener trazabilidad de accesos.
- Garantizar gobernabilidad del sistema de monitoreo.

En la práctica operativa, este enfoque fortaleció la seguridad del NOC sin comprometer la eficiencia del personal.

3.5.2.7 Diseño Conceptual de la Arquitectura. Antes de fijar la tecnología concreta y dimensionar la infraestructura, fue necesario trazar primero una visión arquitectónica del sistema: un boceto conceptual que sirviera de referencia para evaluar cada decisión posterior en función de principios de diseño, y no solo de la disponibilidad inmediata de herramientas (Oppenheimer, 2011).

a) Principios de diseño

El diseño conceptual se apoyó en cuatro principios rectores:

1. Centralización con autonomía regional. La arquitectura concentra la visualización en un único punto, pero cada región conserva margen de operación propio para recolectar sus datos y generar sus alertas. Este equilibrio responde a una necesidad concreta del NOC: contar con una vista global sin perder capacidad de reacción local.

2. Separación de capas funcionales. El sistema se organiza en cuatro capas: una capa de adquisición, encargada de recoger datos desde los dispositivos mediante pollers, proxies o agentes; una capa de procesamiento, que almacena, normaliza y analiza esa información; una capa de presentación, con los dashboards y reportes dirigidos a cada perfil de usuario; y una capa de notificación, responsable de comunicar los eventos al personal operativo.

3. Tolerancia a fallos de comunicación. Dado que las regiones se encuentran distribuidas geográficamente y pueden sufrir cortes de conectividad WAN, el diseño contempla mecanismos de almacenamiento local y sincronización diferida como salvaguarda ante esas interrupciones.

4. Escalabilidad horizontal. El crecimiento en el número de dispositivos monitoreados no debería exigir un rediseño del sistema, sino la incorporación progresiva de nodos de procesamiento adicionales.

Conviene precisar que, dentro del alcance efectivamente implementado en este trabajo (descrito a partir de la sección 3.5.7), la recolección de datos se realiza de forma directa desde el servidor central, sin proxies regionales intermedios.

La incorporación de proxies, contemplada aquí únicamente como principio de diseño, se retoma como propuesta concreta de evolución del sistema en la sección 6.10.

b) Representación conceptual

El esquema conceptual (ver Figura 2) resume los componentes previstos en este diseño y sus relaciones:

1. Servidor central Zabbix: nodo principal que aloja la base de datos, procesa la información y expone la interfaz web.

2. Proxies regionales (componente conceptual, no implementado en la fase actual): nodos intermedios que recolectarían datos localmente en cada región y los transmitirían al servidor central de forma asíncrona.

3. Dispositivos monitoreados: routers Nokia, gestores auxiliares, sistemas de energía y dispositivos de seguridad física, agrupados por región y función.

4. Capa de notificación: integración con Telegram, segmentada por región.

5. Capa de visualización: dashboards corporativos y regionales, con distintos niveles de agregación según el perfil del usuario.

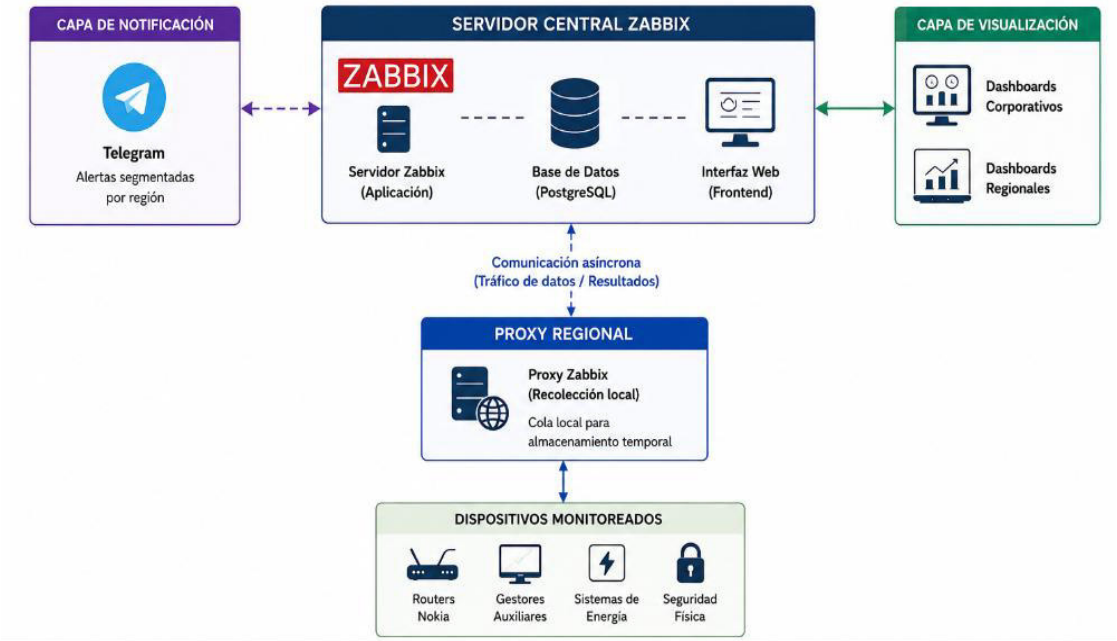


Figura 2 **Esquema Conceptual y sus componentes.** Fuente. Elaboración propia

c) Justificación de la arquitectura conceptual

Este diseño recoge los requerimientos funcionales y no funcionales identificados en el diagnóstico inicial y se apoya en el modelo FCAPS (Stallings, 2018), particularmente en las áreas de gestión de fallas y gestión de desempeño. La separación por capas busca, además, que cada componente pueda evolucionar de forma independiente. El proceso de incorporar una nueva funcionalidad en la capa de notificación, por ejemplo, no debería obligar a modificar la capa de adquisición ni afectar la operación del resto del sistema.

3.5.3 Configuración de Zabbix para el Monitoreo de Red

La fase de configuración del sistema Zabbix fue el eje central de la implementación, ya que en esta etapa se definieron los parámetros que permitirían supervisar de manera efectiva la disponibilidad de los dispositivos de red distribuidos en las distintas regiones operativas.

En la ingeniería de telecomunicaciones, la configuración de un sistema de monitoreo no se limita a habilitar pruebas de conectividad; implica establecer criterios técnicos coherentes con los niveles de desempeño esperados, la topología de red y los indicadores de calidad de servicio definidos por la organización.

De acuerdo con *Stallings (2018)*, un sistema de gestión de redes debe equilibrar tres factores fundamentales:

- Precisión en la detección de fallas
- Minimización de tráfico adicional de supervisión
- Reducción de falsos positivos

Bajo este enfoque, la configuración aplicada en Zabbix fue evaluada considerando el comportamiento típico de las redes IP/MPLS desplegadas en la empresa, las condiciones de latencia propias de enlaces regionales y la necesidad de evitar alarmas innecesarias que pudieran saturar la operación del NOC.

La supervisión inicial se basó en monitoreo ICMP, mecanismo fundamental para verificar disponibilidad básica de hosts, conforme a lo definido en el RFC 792 *(Postel, 1981)*. Este método permite determinar el estado activo/inactivo de un dispositivo sin requerir credenciales SNMP ni agentes instalados, lo cual lo convierte en una solución eficiente para una primera capa de verificación de estado.

3.5.3.1 Configuración de ICMP Ping. La Tabla 4 presenta los parámetros técnicos configurados en la plantilla "Template Module ICMP Ping", utilizada como base para la supervisión de routers Nokia.

Tabla 4 **Parámetros de configuración ICMP**

Parámetro	Valor Configurado	Justificación Técnica
Paquetes enviados	3	Balance entre precisión y bajo tráfico
Intervalo entre paquetes	25 ms	Detecta microcortes sin sobrecarga
Timeout	500 ms	Adecuado para enlaces regionales
Tamaño de paquete	56 bytes	Estándar ICMP sin congestión
Intervalo de actualización	50 segundos	Frecuencia real de sondeo del ítem icmpping (intervalo flexible activo las 24 horas, los 7 días)

Fuente. Elaboración propia

Los parámetros mostrados en la Tabla 4 fueron seleccionados tras evaluar el comportamiento de latencia promedio en las redes regionales.

El envío de tres paquetes por ciclo permite reducir la probabilidad de falsos positivos asociados a pérdidas momentáneas de paquetes. Según *Comer (2018)*, el uso de

múltiples solicitudes ICMP mejora la confiabilidad estadística del diagnóstico de conectividad.

El intervalo de 25 milisegundos entre paquetes responde a un criterio de eficiencia, ya que permite detectar microinterrupciones sin generar tráfico significativo en enlaces de transporte.

El tiempo máximo de espera (timeout) de 500 milisegundos fue considerado apropiado para enlaces regionales que pueden presentar latencias variables debido a distancia geográfica y condiciones de infraestructura.

Bajo el enfoque de gestión de fallas (Fault Management), estos parámetros permiten una detección temprana sin generar alarmas innecesarias que afecten la operación del NOC.

Métricas generadas por ICMP

La Tabla 5 resume las métricas obtenidas automáticamente por la plantilla utilizada.

Tabla 5 **Métricas de disponibilidad generadas**

Métrica	Descripción	Aplicación Operativa
Packet Loss (%)	Porcentaje de pérdida de paquetes	Indica degradación
Average Response Time (ms)	Latencia promedio	Detecta congestión
Host Availability	Estado activo/inactivo	Identifica caída total

Fuente. Elaboración propia

La métrica de pérdida de paquetes funciona como un indicador temprano de

degradación de servicio. Una pérdida sostenida superior al 20% puede reflejar congestión, errores físicos o inestabilidad del enlace.

La latencia promedio permite identificar incrementos anómalos en tiempos de respuesta, lo cual puede anticipar saturación o degradación progresiva.

El estado de disponibilidad (Host Availability) activa el trigger “Host Down”, el cual veremos más adelante y representa el nivel crítico de falla, ya que indica interrupción total del servicio.

Estas métricas permiten medir indirectamente el indicador de disponibilidad definido en SLA, reforzando la capacidad de auditoría del sistema.

Triggers configurados

La Tabla 6 muestra los umbrales establecidos para la generación automática de alertas.

Tabla 6 **Umbrales de alerta configurados**

Evento	Umbral	Duración	Tipo de impacto
Alta pérdida	>20%	5 min	Degradación
Alta latencia	>100 ms	5 min	Congestión
Host Down	3 fallos consecutivos	Inmediato	Crítico

Fuente. Elaboración propia.

Los umbrales configurados fueron evaluados considerando la estabilidad promedio de la red y recomendaciones técnicas para entornos empresariales *(Stallings, 2018)*.

El criterio de 5 minutos consecutivos permite evitar activaciones por fluctuaciones transitorias, garantizando que la alerta represente una condición real sostenida.

El disparador de tres fallos consecutivos fue definido como condición crítica, ya que

indica pérdida total de conectividad.

Este esquema de triggers permite equilibrar sensibilidad y estabilidad operativa, reduciendo falsos positivos y evitando sobrecarga en el canal de notificaciones

La configuración implementada demuestra que el sistema no fue desplegado bajo parámetros arbitrarios, sino sustentado en criterios técnicos coherentes con estándares de monitoreo de redes IP.

La combinación de parámetros ICMP, métricas generadas y umbrales configurados permitió establecer una primera capa de supervisión robusta orientada a la detección temprana de fallas, reduciendo el tiempo promedio de identificación de incidentes y fortaleciendo el rendimiento operativo del NOC.

3.5.4 Organización de la Estructura de la red en Zabbix

La organización lógica de los dispositivos dentro de Zabbix fue un componente estratégico del diseño de la solución, ya que una estructura inadecuada puede comprometer la escalabilidad, dificultar la administración y limitar la claridad operativa del NOC.

De acuerdo con *Oppenheimer (2011)*, un sistema de monitoreo eficiente debe reflejar la topología real de la red y su distribución organizacional, permitiendo una correlación clara entre eventos técnicos y su ubicación geográfica o funcional.

En el entorno analizado, la infraestructura se encontraba distribuida en cuatro regiones operativas, cada una con redes de transporte y acceso, además de sistemas de soporte energético y seguridad física. Por tanto, replicar esta estructura en la plataforma de monitoreo era fundamental para mantener coherencia entre la arquitectura física y su representación lógica. Esta correspondencia no es un detalle menor ya que sienta las bases sobre las cuales se apoyan las etapas posteriores del proyecto, como la segmentación regional del alertamiento y la construcción de dashboards diferenciados.

3.5.4.1 Creación de Grupos de Hosts. Se diseñó una estructura jerárquica basada en dos niveles principales de segmentación:

1. Segmentación geográfica (regiones operativas).
2. Segmentación funcional (tipo de red y tipo de componente).

La Tabla 7 resume la organización implementada.

Tabla 7 **Estructura jerárquica de monitoreo en Zabbix**

Nivel 1	Nivel 2	Nivel 3
Región San Martin	Transporte / Acceso	Componentes de Red, Gestores Auxiliares, Componentes de Energía,
Región Arequipa	Transporte / Acceso	Componentes de Red, Gestores Auxiliares, Componentes de Energía, Seguridad Física
Región La Libertad	Transporte / Acceso	Componentes de Red, Gestores Auxiliares, Componentes de Energía, Seguridad Física
Región Ancash	Transporte / Acceso	Componentes de Red, Gestores Auxiliares, Componentes de Energía, Seguridad Física

Fuente. Elaboración propia

Fundamentación de la Estructura

Esta organización responde a principios de modularidad y escalabilidad. Según *Stallings (2018)*, en sistemas de gestión de red es recomendable segmentar los recursos monitoreados de acuerdo con su función y ubicación para facilitar la identificación de

fallas y reducir ambigüedades en el análisis.

La segmentación regional permite:

- Identificar rápidamente el ámbito geográfico del incidente.
- Aplicar políticas diferenciadas de acceso.
- Generar dashboards específicos por región.

La segmentación funcional permite:

- Diferenciar fallas de red de fallas energéticas.
- Aplicar triggers específicos según tipo de dispositivo.
- Configurar notificaciones selectivas (por ejemplo, Telegram solo para routers).

Desde la perspectiva del modelo FCAPS, esta estructura fortalece principalmente la gestión de fallas (Fault Management) y la gestión de desempeño (Performance Management). La Figura 3 muestra cómo la arquitectura fragmentada previa fue transformada en una estructura centralizada con organización lógica coherente. Cada región se representa como un contenedor principal, dentro del cual se segmentan redes y tipos de componentes.

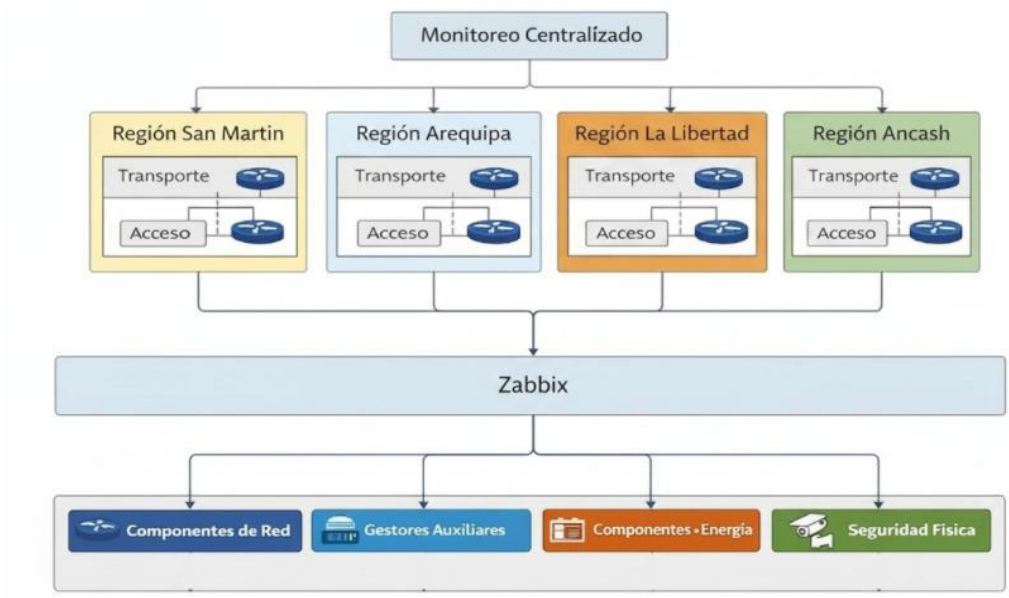


Figura 3 **Modelo jerárquico de organización de hosts en Zabbix.** Fuente. Elaboración propia

Esta representación facilita la correlación visual de eventos y mejora la eficiencia en la supervisión diaria del NOC.

3.5.4.2 Carga masiva de Hosts en Zabbix. La incorporación de dispositivos en el sistema de monitoreo representó un desafío operativo debido al volumen de routers y equipos asociados en las distintas regiones.

La creación manual de hosts, si bien es funcional para entornos pequeños, resulta ineficiente y propensa a errores en escenarios con decenas o cientos de dispositivos. Según *Zabbix LLC (2023a)*, la importación estructurada mediante archivos JSON constituye una práctica recomendada para despliegues empresariales.

Por este motivo, se optó por la carga masiva mediante importación en formato JSON, siguiendo el esquema oficial de Zabbix y usando como método de creación y agilización de sus líneas de código para su contenido el uso de IA (Ver Anexo A). El proceso de importación se evidencia en la Figura 4, donde se muestra la interfaz de Zabbix utilizada para la carga estructurada de hosts

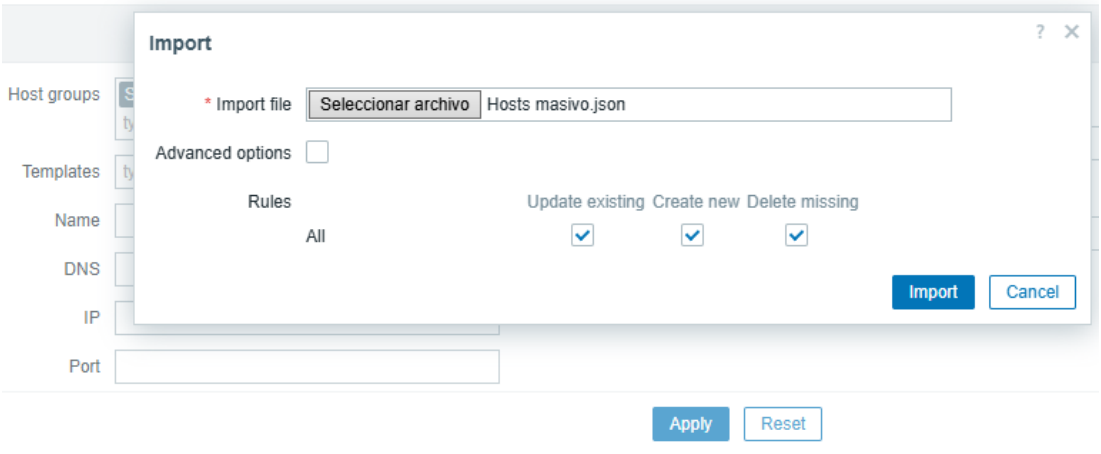


Figura 4 **Carga del archivo json en sistema Zabbix.** Fuente. Captura de pantalla de la interfaz de Zabbix para importación de hosts

Justificación Técnica de la Automatización

La automatización permitió:

- Reducir tiempos de configuración.
- Evitar errores tipográficos.
- Mantener uniformidad en plantillas.
- Facilitar futuras expansiones.

Se realizó una estimación comparativa del tiempo requerido la cual podemos observar en la Tabla 8.

Tabla 8 **Comparación de métodos de incorporación de hosts**

Método	Tiempo estimado por host	Tiempo total (100 hosts)	Riesgo de error
Manual	1–2 minutos	100–120 minutos	Alto
Importación JSON	2 segundos (total)	1 minuto	Bajo

Fuente. Elaboración propia

El uso de importación masiva permitió una reducción superior al 99% en el tiempo estimado de configuración. Además, disminuyó la probabilidad de inconsistencias en nombres, asignación de plantillas o direcciones IP.

Desde la perspectiva de eficiencia operativa, esta decisión contribuyó a acelerar la implementación sin afectar la precisión de la configuración.

Impacto Operativo de la Organización y Automatización

La correcta organización jerárquica y la automatización de carga generaron reducción

del tiempo de despliegue, mejora en la claridad visual del sistema, simplificación del mantenimiento y facilidad para escalar el monitoreo ante nuevas incorporaciones.

Estas decisiones técnicas no fueron meramente operativas, sino estratégicas, ya que permitieron establecer una arquitectura sostenible y alineada con el crecimiento proyectado de la red.

3.5.5 Configuración del Sistema de Alertas

La configuración del sistema de alertas constituyó un componente crítico dentro de la implementación, ya que el monitoreo sin alertamiento automatizado no garantiza reducción efectiva del tiempo promedio de detección (MTTD).

De acuerdo con ITIL *(Axelos, 2019)*, la eficiencia en la gestión de incidentes depende directamente de la capacidad del sistema para generar notificaciones oportunas, priorizadas y dirigidas al personal responsable.

En este contexto, el sistema de alertas fue diseñado bajo tres principios fundamentales:

- Clasificación por severidad
- Diferenciación entre degradación e indisponibilidad total
- Activación automática de respuesta operativa

3.5.5.1 ***Definición de Niveles de Alerta.*** La plantilla Template Module ICMP Ping incorpora tres niveles predeterminados de severidad mostrados a continuación en la Tabla 9, los cuales fueron analizados y mantenidos tras evaluación técnica.

Tabla 9 **Niveles de alerta configurados**

Severidad	Nombre del Trigger	Descripción Operativa
Warning	High ICMP ping loss	Pérdida elevada de paquetes
Warning	High ICMP ping response time	Latencia elevada sostenida
High	Unavailable by ICMP ping	Host completamente inalcanzable

Fuente. Elaboración propia

Interpretación Técnica de los Niveles de Alerta

Alertas tipo *Warning*

Las alertas clasificadas como "Warning" representan condiciones de degradación del servicio. No implican caída total, pero sí constituyen indicadores tempranos de posibles fallas inminentes.

Desde la perspectiva de gestión proactiva *(Oppenheimer, 2011),* la detección de pérdida de paquetes o incremento sostenido de latencia permite anticipar:

- Congestión de enlace
- Saturación de equipo
- Problemas físicos intermitentes
- Inestabilidad en rutas

Estas alertas cumplen una función preventiva, permitiendo intervención antes de que se produzca una interrupción total.

Alertas tipo *High*

La severidad *High* corresponde a la condición crítica de indisponibilidad total del host.

En este escenario:

- El router no responde a ICMP.
- Se pierde gestión remota.
- Se activa protocolo inmediato de troubleshooting.

Esta alerta marca el inicio formal del proceso de diagnóstico, alineado con el flujo de gestión de incidentes definido en ITIL *(Axelos, 2019)*.

Desde el punto de vista operativo, esta clasificación permite priorizar recursos y evitar que eventos menores saturen la atención del equipo.

3.5.6 Configuración de Telegram en Zabbix

La integración del sistema de monitoreo con Telegram fue implementada con el objetivo de reducir el tiempo promedio de detección (MTTD) y mejorar la inmediatez en la recepción de alertas.

En entornos de NOC distribuidos, depender exclusivamente de la visualización en consola puede retrasar la reacción ante eventos críticos. Según *Stallings (2018)*, los sistemas de gestión de red modernos deben incorporar mecanismos de notificación externos que permitan movilidad y disponibilidad del personal operativo.

Telegram fue seleccionado por:

- Disponibilidad multiplataforma (web y móvil)
- Bajo costo (gratuito)
- Facilidad de integración vía API
- Alta confiabilidad en entrega de mensajes

3.5.6.1 Creación del Bot y Obtención del Token API. La creación del bot se realizó mediante la herramienta oficial @BotFather de Telegram, utilizando el comando /newbot.

El proceso permitió obtener un Token API, el cual actúa como credencial de autenticación para que el servidor Zabbix pueda enviar solicitudes HTTP hacia la API de Telegram conforme a lo especificado en la documentación oficial de la API de bots (Telegram Messenger LLP, 2024).

Considerando la seguridad, este token es un elemento sensible y debe almacenarse de forma protegida, ya que permite acceso al canal de notificaciones.

3.5.6.2 Configuración de Grupos por región. Para mantener la coherencia con la estructura regional definida en Zabbix, se crearon grupos de Telegram independientes para cada región operativa detallados en la Tabla 10.

Tabla 10 **Configuración de grupos Telegram**

Grupo Telegram	Chat_ID	Bot Asociado
NOC_Alerts_San Martin	Identificador del grupo De Telegram	NotificacionesBot
NOC_Alerts_Arequipa	Identificador del grupo De Telegram	NotificacionesBot
NOC_Alerts_La Libertad	Identificador del grupo De Telegram	NotificacionesBot
NOC_Alerts_Ancash	Identificador del grupo De Telegram	NotificacionesBot

Fuente. Elaboración propia

Fundamentación de la Segmentación Regional

La separación de grupos permitió:

- Evitar saturación de mensajes irrelevantes.
- Dirigir alertas exclusivamente al equipo responsable.
- Reducir ruido operativo.

- Mejorar tiempos de reacción.

En síntesis, puede concluirse que en NOC con infraestructura geográficamente distribuida, la segmentación regional del alertamiento es un factor crítico para la eficiencia operativa, superior a la centralización sin diferenciación regional.

3.5.6.3 Configuración de usuarios en Zabbix por región. En el servidor Zabbix se implementó una estructura de usuarios alineada con la división regional. Dicha estructura se muestra en la Tabla 11.

Tabla 11 **Usuarios configurados en Zabbix**

Usuario	Rol	Alcance
Admin	Administrador	Visibilidad global
region_san martin	Usuario	Región San Martin
region_arequipa	Usuario	Región Arequipa
region_la libertad	Usuario	Región La Libertad
region_ancash	Usuario	Región Ancash

Fuente. Elaboración propia

Esta configuración se alineó con el principio de mínimo privilegio (ISO/IEC 27001; ISO, 2013), garantizando que cada operador tuviera acceso únicamente a la información correspondiente a su región.

Considerando la gobernanza del sistema, esta segmentación reduce riesgos de modificación indebida y mejora trazabilidad.

3.5.6.4 Implementación del Script de Envío.

La comunicación entre Zabbix y Telegram se estableció mediante un script personalizado que utiliza el comando curl para realizar solicitudes HTTP hacia la API de Telegram.

El script recibe tres parámetros:

- Chat ID
- Asunto de la alerta
- Mensaje detallado

El flujo de envío se estructura de la siguiente manera: Primero, un trigger en Zabbix cambia a estado de problema cuando la condición configurada se cumple, sea esta la pérdida de paquetes ICMP de un host, la caída de un enlace de transporte o cualquier otra condición evaluada por el motor del trigger. Ese cambio genera un evento que Zabbix contrasta contra las acciones configuradas para determinar si corresponde notificar y hacia qué destinatarios en función de la severidad del evento y la región a la que pertenece el host o enlace afectado.

Una vez identificada la acción correspondiente, Zabbix ejecuta el Media Type asociado, que en este caso invoca al script personalizado descrito anteriormente. El script, ya en ejecución, recibe los tres parámetros mencionados y construye con ellas la solicitud HTTP hacia la API de Telegram, enviándola mediante una petición POST dirigida al endpoint del bot.

Finalmente, los servidores de Telegram procesan esa solicitud y entregan el mensaje al grupo correspondiente, completando el ciclo de notificación en cuestión de segundos desde que el trigger cambio de estado.

Desde la perspectiva de arquitectura, este mecanismo funciona como un puente entre el sistema de monitoreo y la plataforma de mensajería, permitiendo alertamiento en tiempo real sin requerir infraestructura adicional. La estructura general de este script se presenta en la Figura 5.

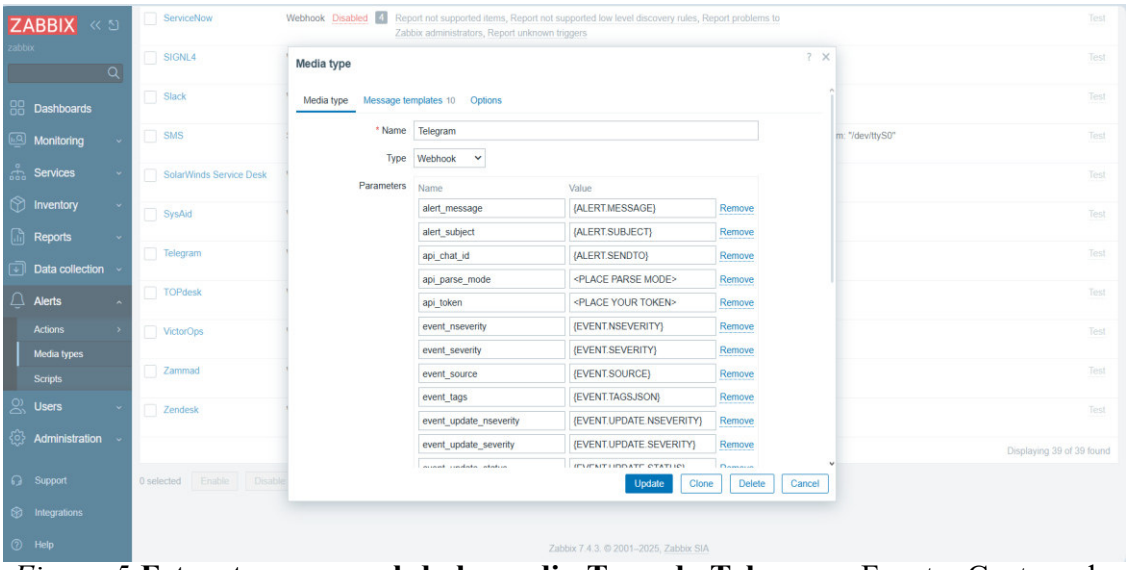


Figura 5 **Estructura general de la media Type de Telegram.** Fuente. Captura de pantalla del apartado media type de Telegram en Zabbix

Impacto Operativo del Sistema de Alertas

La integración con Telegram permitió:

- Reducir dependencia exclusiva de consola NOC.
- Aumentar inmediatez de notificación.
- Mejorar movilidad del personal.

En términos prácticos, se transformó el modelo de supervisión de pasivo a activo, fortaleciendo la capacidad de respuesta ante eventos críticos.

3.5.6.5 Personalización de Mensajes. La personalización de los mensajes de notificación fue un elemento clave dentro del diseño del sistema de alertas, ya que la efectividad del alertamiento depende de la rapidez de envío y de la claridad y estructuración de la información presentada al operador.

De acuerdo con *Axelos (2019)*, en la gestión de incidentes es fundamental que las notificaciones contengan información suficiente para iniciar el diagnóstico sin necesidad de acceder inmediatamente a la plataforma de monitoreo. Un mensaje

ambiguo o incompleto puede generar retrasos innecesarios en la respuesta operativa.

Bajo este enfoque, la plantilla de notificación fue diseñada con los siguientes criterios:

- Claridad estructural
- Identificación visual rápida
- Información técnica esencial
- Jerarquización por severidad
- Minimización de ambigüedades

Diseño Estructural del Mensaje

La plantilla fue configurada en el apartado Media Type de Zabbix, utilizando variables dinámicas que permiten incluir información contextual del evento.

El mensaje fue estructurado en formato organizado, incluyendo:

- Severidad del evento
- Nombre del host
- Dirección IP
- Región
- Fecha y hora
- Estado actual
- Descripción técnica
- Tiempo de ocurrencia

El uso de emojis fue incorporado estratégicamente como mecanismo de identificación visual inmediata, especialmente en dispositivos móviles. Un ejemplo del resultado final de esta plantilla se observa en la Figura 6.

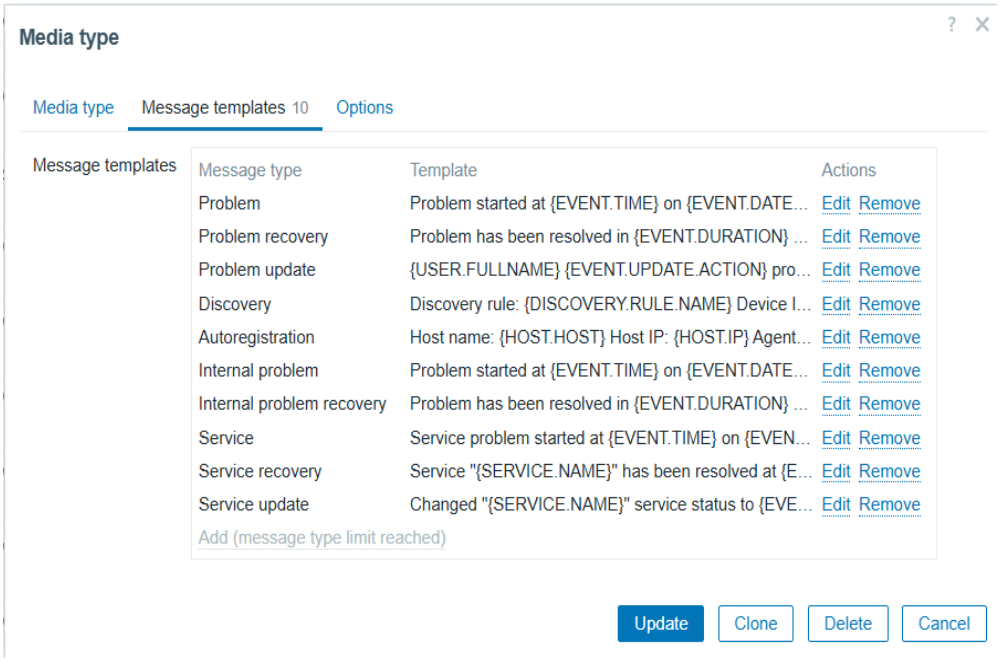


Figura 6 **Message Template.** Fuente. Captura de pantalla del Message Template en Zabbix

3.5.6.6 Arquitectura de Gestión de Alertas. La arquitectura de gestión de alertas implementada en Zabbix fue diseñada bajo un enfoque estructurado de distribución regional, priorización por severidad y cierre automático del ciclo del incidente.

En entornos NOC, el alertamiento no debe limitarse al envío de notificaciones; debe formar parte de un flujo de gestión integral que contemple:

- Generación del evento
- Clasificación por severidad
- Enrutamiento inteligente
- Notificación contextual
- Confirmación
- Cierre automático
- Registro histórico

De acuerdo con ITIL *(Axelos, 2019)*, el ciclo de vida de un incidente comprende desde su detección hasta su resolución y documentación. La arquitectura implementada en Zabbix permite cubrir estas fases mediante la configuración de triggers, actions, media types y mensajes de recuperación.

Flujo de Gestión Implementado

El flujo configurado puede representarse en las siguientes etapas:

1. Se activa un trigger por condición ICMP.
2. Zabbix evalúa la severidad.
3. Se ejecuta una acción asociada.
4. La acción verifica condiciones:
    - Severidad
    - Región
    - Estado de mantenimiento
5. Se envía notificación al grupo Telegram correspondiente.
6. Cuando el evento se resuelve, se genera mensaje automático de recuperación.

Este mecanismo asegura que cada región reciba exclusivamente las alertas correspondientes a su ámbito operativo, evitando la saturación informativa y reduciendo el ruido operacional.

Segmentación Regional del Alertamiento

La distribución regional de alertas fue diseñada con el objetivo de mantener aislamiento informativo entre equipos técnicos la cual es resumida en la Tabla 12.

Tabla 12 **Ejemplo de ciclo de alerta crítica para un host**

Campo	Evento de Problema	Evento de Recuperación
Host	Nodo Region_1	Nodo Region_1
Fecha	Inicio del incidente	Fin del incidente
Duración	No se incluye en el mensaje	Duración del incidente
Severidad	Alta	No se incluye en el mensaje

Fuente. Elaboración propia

La tabla evidencia que el sistema notifica la caída del equipo y su recuperación. Este aspecto es fundamental desde la perspectiva de gestión de incidentes, ya que permite:

- Confirmar restablecimiento del servicio.
- Cerrar formalmente el evento.
- Actualizar estadísticas de disponibilidad.
- Reducir incertidumbre operativa.

La inclusión automática de mensajes de recuperación completa el ciclo del incidente, evitando que los operadores mantengan estados ambiguos respecto a la situación del equipo.

Beneficios Operativos de la Arquitectura de Alertas

La arquitectura implementada permitió:

- Reducción del tiempo promedio de detección (MTTD).
- Segmentación inteligente por región.
- Eliminación de alertas irrelevantes.
- Registro histórico estructurado.

- Mejora en trazabilidad de eventos.

A nivel sistémico, se pasó de un modelo reactivo basado en consulta manual a un modelo activo basado en notificación automatizada.

3.5.6.7 Implementación de Dashboards. La implementación de dashboards en Zabbix respondió a la necesidad de transformar datos técnicos dispersos en información visual estructurada, permitiendo una interpretación rápida del estado de la infraestructura.

Según *Few (2013)*, la visualización efectiva de datos debe facilitar la comprensión inmediata, reducir la carga cognitiva y permitir identificar patrones y anomalías sin análisis complejo.

En el contexto del NOC, los dashboards cumplen una función estratégica al:

- Resumir el estado general.
- Priorizar incidentes.
- Facilitar toma de decisiones.
- Permitir seguimiento histórico.

3.5.6.8 Arquitectura de Dashboards. Se diseñó una arquitectura jerárquica compuesta por:

- Un Dashboard corporativo principal.
- Cuatro dashboards regionales especializados.

Esta arquitectura responde a un modelo de doble nivel:

- Nivel estratégico (corporativo)
- Nivel operativo (regional)

Dashboard Corporativo

El Dashboard principal actúa como centro de mando unificado, proporcionando:

- Estado general por regiones.
- Indicadores agregados.
- Tendencias globales.
- Top problemas críticos.

Widgets Implementados

- Estado del servidor Zabbix (CPU, RAM, DB).
- Resumen por región (semáforo de estado).
- Top 5 incidentes críticos.
- Estadísticas de alertas.
- Métricas de desempeño agregadas.

Este Dashboard está orientado a supervisores y administradores, permitiendo evaluación global sin necesidad de ingresar a cada región.

3.5.6.9 Configuración de Dashboards Regionales. Cada región dispondrá de un Dashboard especializado configurado específicamente. Cada región cuenta con un Dashboard especializado, diseñado para soporte operativo diario.

Estructura estandarizada

Cada Dashboard regional incluye:

- Widget de problemas por severidad.
- Widget de problemas críticos (High).
- Lista de hosts caídos.

- Lista de Enlaces caídos
- Historial de recuperación.
- Indicadores de disponibilidad.

Fundamentación de Diseño

El widget de problemas por severidad utiliza codificación por colores:

- Rojo: High
- Naranja: Average
- Amarillo: Warning

Este esquema visual responde a principios de diseño cognitivo que priorizan percepción rápida del riesgo.

El widget de problemas actuales filtra exclusivamente severidad High, evitando distracción con eventos menores. La actualización cada 30 segundos garantiza información prácticamente en tiempo real sin generar sobrecarga excesiva.

3.5.6.10 Implementación Técnica. La correcta funcionalidad de los dashboards dependió de:

- Organización lógica de grupos de hosts.
- Configuración adecuada de permisos.
- Filtrado correcto por región.
- Exclusión de mantenimientos programados.

Se estableció una política de retención de datos de 90 días, permitiendo:

- Análisis histórico.
- Identificación de patrones recurrentes.

- Evaluación de disponibilidad regional.

El balance entre retención histórica y rendimiento fue definido considerando capacidad de almacenamiento y carga de base de datos.

Impacto Integral de Alertas y Dashboards

La combinación de alertamiento automatizado, segmentación regional, mensajes personalizados y dashboards jerárquicos consolidó un modelo operativo del NOC proactivo, visual y medible, superando las limitaciones del esquema fragmentado y reactivo previo a la implementación.

3.5.7 Implementación de Monitoreo SNMP para los Enlaces de la red de Transporte

La supervisión de enlaces de transporte en la plataforma Nokia NSP NFM-P opera principalmente a través de la visualización de la topología de red, donde el cambio de color de un enlace señala la existencia de alguna condición de alarma. No obstante, esta modalidad presenta limitaciones operativas de fondo: el indicador visual no distingue si el enlace cayó por una falla física, si fue inhabilitado deliberadamente, o si se trata de una degradación parcial sin pérdida total de tráfico. Ante esa ambigüedad, el personal del NOC debía revisar la topología de manera periódica y manual para determinar el origen y severidad de cada condición, lo que introduce dependencia de la constancia del operador y retrasos en la detección.

El monitoreo SNMP basado en la lectura directa de ifAdminStatus e ifOperStatus resuelve esta limitación de raíz: permite determinar con precisión si un enlace caído responde a una falla física no prevista o a una acción administrativa intencional, generando alertas diferenciadas para cada escenario. Esta capacidad de discriminación elimina la ambigüedad operativa del modelo previo y permite al NOC responder de forma específica y proporcional a cada tipo de evento. A diferencia del monitoreo ICMP —orientado a confirmar si un equipo responde al ping—, este componente supervisa el estado individual de cada enlace lógico entre nodos, con una granularidad que el NFM-P no entregaba de forma automatizada *(McCloghrie & Kastenholz, 2000; Zabbix LLC, 2023a)*.

3.5.7.1 Identificación de OIDs y Diseño del Mecanismo de Descubrimiento.

La fase exploratoria consistió en la ejecución de consultas snmpwalk sobre una muestra representativa de equipos, con el objetivo de identificar los Object Identifiers (OID) de la rama estándar IF-MIB (1.3.6.1.2.1.2.2.1) necesarios para representar el estado administrativo y operativo de cada interfaz (Postel & Harrenstien, 1990). Los OIDs identificados en esta fase se resumen en la Tabla 13.

Tabla 13 **OIDs IF-MIB utilizados para el monitoreo de enlaces**

OID	Nombre IF-MIB	Tipo de Dato
1.3.6.1.2.1.2.2.1.2	ifDescr	DisplayString
1.3.6.1.2.1.2.2.1.7	ifAdminStatus	INTEGER (1=up, 2=down)
1.3.6.1.2.1.2.2.1.8	ifOperStatus	INTEGER (1=up, 2=down)

Fuente. Elaboración propia basado en RFC 2863.

El protocolo SNMPv3 fue configurado bajo el nivel de seguridad authPriv, combinando autenticación MD5 y cifrado DES, conforme al modelo de seguridad USM (User-based Security Model) definido en el RFC 3414.

Este esquema de autenticación y cifrado responde a los principios generales de seguridad en redes de datos (Stallings, 2017).

EL siguiente paso es configurar dentro de Zabbix el sistema que descubre automáticamente los puertos de cada router y crea el monitoreo correspondiente.

La arquitectura se organiza en cuatro capas funcionales, descritas de la capa más cercana al dispositivo a la capa de notificación como podemos ver en la Tabla 14.

Tabla 14 **Diagrama de arquitectura en capas**

Capa	Configuración en Zabbix	Descripción
1.Adquisición	Master Items (3)	Le pregunta al router su lista completa de puertos y estados. Es la única consulta real que se hace al equipo
2.Descubrimiento y filtrado	Discovery Rule	Analiza la lista de puertos y aplica cualquier filtro configurado.
3.Extracción de valores	Item Prototypes	Para cada puerto descubierto, crea automáticamente los ítems de Admin Status y Oper Status, y los triggers de alerta.
4.Notificación	Telegram + Trigger Prototypes	Los Trigger Prototypes evalúan las combinaciones de los valores extraídos para generar los mensajes de problemas en los enlaces

Fuente. Elaboración propia

3.5.7.2 Creación de los tres Master Items. Necesitamos tres consultas separadas al router y para ellos utilizaremos los OIDs vistos previamente en la Tabla 13.

La configuración básica de cada Master Item lo detallamos en la Tabla 15.

Tabla 15 **Configuración básica de cada Master Item**

Master item	Name	Update Interval (minutos)	Aplicación
1	SNMP walk ifDescr	5	Identificación del puerto/enlace
2	SNMP walk ifAdminStatus	1	Estado configurado por el operador
3	SNMP walk ifOperStatus	1	Estado real de la señal de fibra

Fuente. Elaboración propia

Sondear el Master item SNMP walk ifDescr cada 5 minutos es más que suficiente para detectar un cambio estructural (una interfaz nueva o un puerto retirado) sin sobrecargar innecesariamente al poller ni al propio router con un walk completo.

3.5.7.3 Configuración de la Discovery Rule y Filtros. Se identificaron tres formatos distintos de "ifDescr" coexistentes en un mismo equipo (interfaces lógicas IP, puertos físicos Ethernet y grupos LAG), lo cual determinó el diseño de una expresión de filtrado de doble condición, combinada mediante el operador lógico "And", a fin de seleccionar exclusivamente los puertos físicos de enlace entre nodos (prefijo to_). La configuración final de estos filtros se detalla en la Tabla 16.

Tabla 16 **Configuración de filtros de la Discovery Rule**

Macro	Operador	Expresión Regular	Función
{#IFDESCR}	matches	^[0-9]+/[0-9]+/[0-	Selecciona puertos físicos con destino

		9]+,.*to_	to_
{#IFDESCR}	does not match	to_FSU|to_Switch-PoE	Excluye enlaces de gestión

Fuente. Elaboración propia

3.5.7.4 Item Prototypes, Trigger Prototypes y Encapsulamiento en Template. A partir de cada interfaz descubierta, se generaron de forma automática dos Dependent Items (Admin Status y Oper Status) y dos Trigger Prototypes, diferenciando entre una condición de falla real (Link DOWN) y una condición de apagado intencional (Port DISABLED). Los Trigger Prototypes configurados se resumen en la Tabla 17.

Tabla 17 **Trigger Prototypes configurados para el monitoreo de enlaces**

Trigger	**Severidad**	**Expresión (Simplificada)**
Link DOWN	High	min(operstatus,#3) =2 and last(adminstatus)=1
Port DISABLED	Warning	last(adminstatus)=2

Fuente. Elaboración propia

El uso de la función min() con ventana de 3 valores, en lugar de last(), respondió a la necesidad de evitar falsos positivos de recuperación ante eventos de flapping (inestabilidad transitoria del enlace), detectados durante la fase de validación.

3.5.7.5 Creación del Template Nokia IXR - Interface Monitor. La configuración de los Mater Items, Discovery Rules, Item Prototypes y Trigger Prototypes fueron guardadas en un Template denominado "Nokia IXR - Interface Monitor" como se puede ver en la Figura 7, permitiendo su aplicación estandarizada sobre múltiples hosts sin configuración manual individual.

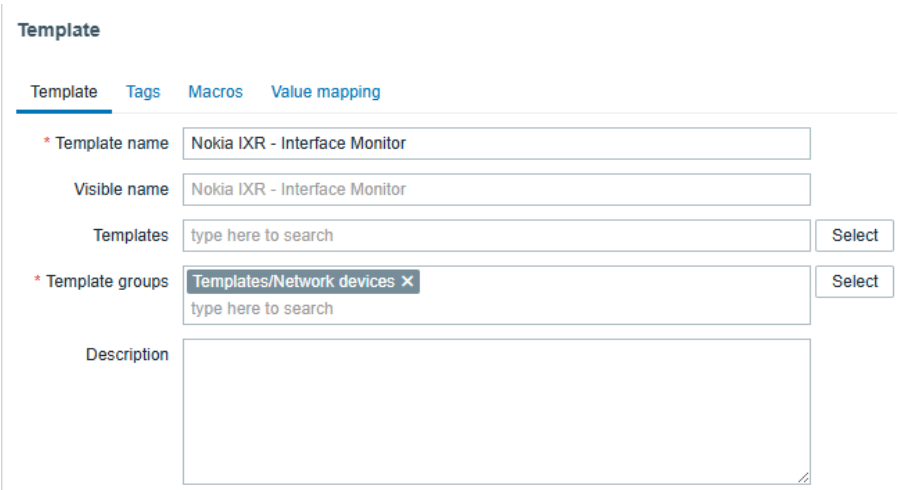


Figura 7 **Template Nokia IXR – Interface Monitor.** Fuente. Captura de pantalla del Template en Zabbix.

3.5.7.6 Automatización de la Asignación Masiva del template Nokia IXR - Interface Monitor mediante API. Dado que Zabbix 7.2 no permite la creación masiva de interfaces SNMP desde la interfaz web, se desarrolló un script en Python (Ver Anexo C) que mediante la API REST de Zabbix automatiza la creación de la interfaz SNMPv3 y la asignación del Template a la totalidad de hosts de un grupo. Un ejemplo del resultado de la ejecución del script, lo podemos ver en la Figura 8**.**

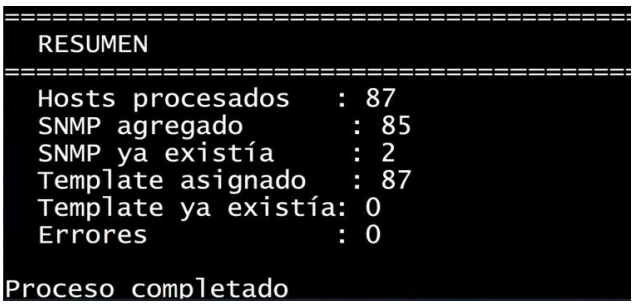


Figura 8 **Ejemplo del Resumen de ejecución del script de asignación masiva para una región.** Fuente. Captura de pantalla de la cantidad de interfaces SNMP creadas y templates asignados del script.

3.5.7.7 Diferenciación de Alertas Telegram (Host vs. Enlace). Se tuvo la necesidad de modificar el script del Media Type Telegram para diferenciar entre alertas de disponibilidad de equipos y alertas sobre el estado de enlace, así como la recuperación del evento mediante la inspección del nombre del evento y la construcción de un encabezado diferenciado con formato HTML. El resultado final de las alertas lo podemos ver en la Tablas 18 y 19.

Tabla 18 **Comparación de mensajes de Alerta de Telegram diferenciados: HOST vs. LINK**

Campo	Alerta de Enlace	Alerta de Host
Subject	LINK DOWN	HOST DOWN
Message	Host Link Start Time Severity	Host Start Time Severity

Fuente: Elaboración propia.

Tabla 19 **Comparación de mensajes de Recuperación de evento de Telegram diferenciados: HOST vs. LINK**

Campo	Alerta de Enlace	Alerta de Host
Subject	LINK UP	HOST UP
Message	Host Link Duration	Host Duration Resolved

Resolved

Fuente: Elaboración propia.

3.5.7.8 Optimización de la Base de Datos con TimescaleDB. Como consecuencia del incremento en el volumen de datos generado por el monitoreo de 1,493 hosts (283.44 valores/segundo), se implementó la extensión TimescaleDB sobre PostgreSQL, mediante la conversión de las tablas de historial y tendencias en hypertables particionadas temporalmente, junto con políticas de compresión automática segmentadas por itemid. La configuración de particionamiento temporal empleó la función set_integer_now_func() para definir el criterio de avance de chunks (Timescale Inc., 2024b).

3.6 Resultado de la actividad

La implementación del sistema de monitoreo basado en Zabbix permitió consolidar una plataforma centralizada, automatizada y segmentada para la supervisión de la infraestructura del NOC.

Los resultados obtenidos se presentan en términos de:

- Evidencia técnica de implementación.
- Funcionamiento operativo validado.
- Mejora estructural del modelo de monitoreo.
- Impacto en la detección temprana de incidentes.
- Optimización del flujo de gestión de alertas.

3.6.1 Evidencia del Entorno Configurado

Instalación Completa de la Plataforma

Se logró la instalación y puesta en producción de Zabbix 7.2, Ubuntu Server 22.04 LTS, PostgreSQL como motor de base de datos y Apache como servidor web conforme a la documentación oficial de la distribución (Ubuntu, 2024).

La elección tecnológica respondió a criterios de estabilidad y compatibilidad, siguiendo la documentación oficial de *Zabbix LLC (2023b)*.

Resultado obtenido:

- Plataforma estable y operativa.
- Integración correcta entre servidor, base de datos y frontend.
- Sin errores críticos durante la instalación.
- Validación exitosa de conectividad y acceso web.

La instalación y puesta en producción del servidor se documenta en la Figura 9

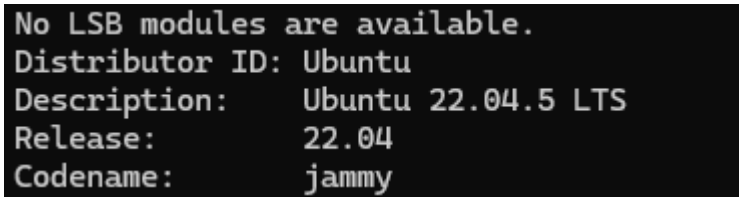


Figura 9 **Instalación de la plataforma sobre Ubuntu Server 22.04 LTS.** Fuente. Elaboración propia

La instalación y puesta en producción de la plataforma Zabbix se evidencia en la Figura 10.

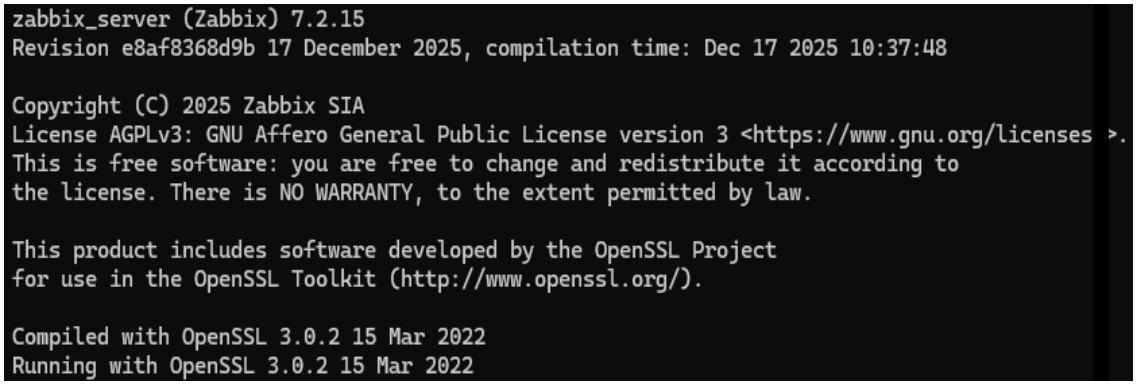


Figura 10 **Instalación de Zabbix sobre Ubuntu Server 22.04 LTS.** Fuente. Elaboración propia

La Figura 11 muestra la instalación de la base de datos PostgreSQL, realizada siguiendo el procedimiento oficial de configuración (PostgreSQL Global Development Group, 2024).



Figura 11 **Instalación de la base de datos PostgreSQL.**

3.6.2 Sistema de Alertas Implementado

Bot de Telegram Funcional

Se creó y configuró el bot "NotificacionesBot", obteniendo su Token API y validando su funcionamiento mediante pruebas de envío y recepción de mensajes.

Resultados validados:

- Comunicación exitosa entre Zabbix y Telegram.
- Recepción de alertas en tiempo real.
- Confirmación de mensajes de recuperación.
- Correcta segmentación por región.

La Figura 12 evidencia el proceso de creación del bot de Telegram y la generación del

Token API necesario para la integración con Zabbix

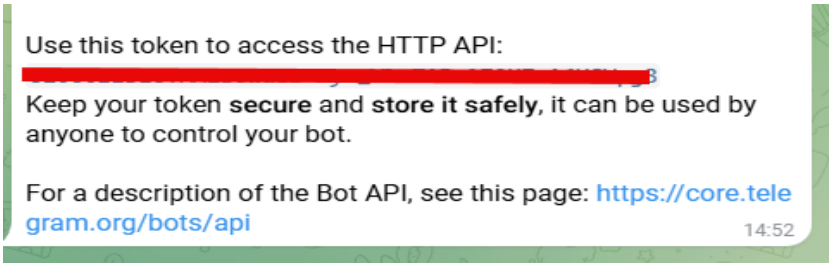


Figura 12 **Creación del bot y Token.** Fuente. Elaboración propia

La Figura 13 presenta la prueba de funcionamiento de los mensajes al bot de Telegram gracias al token brindado.

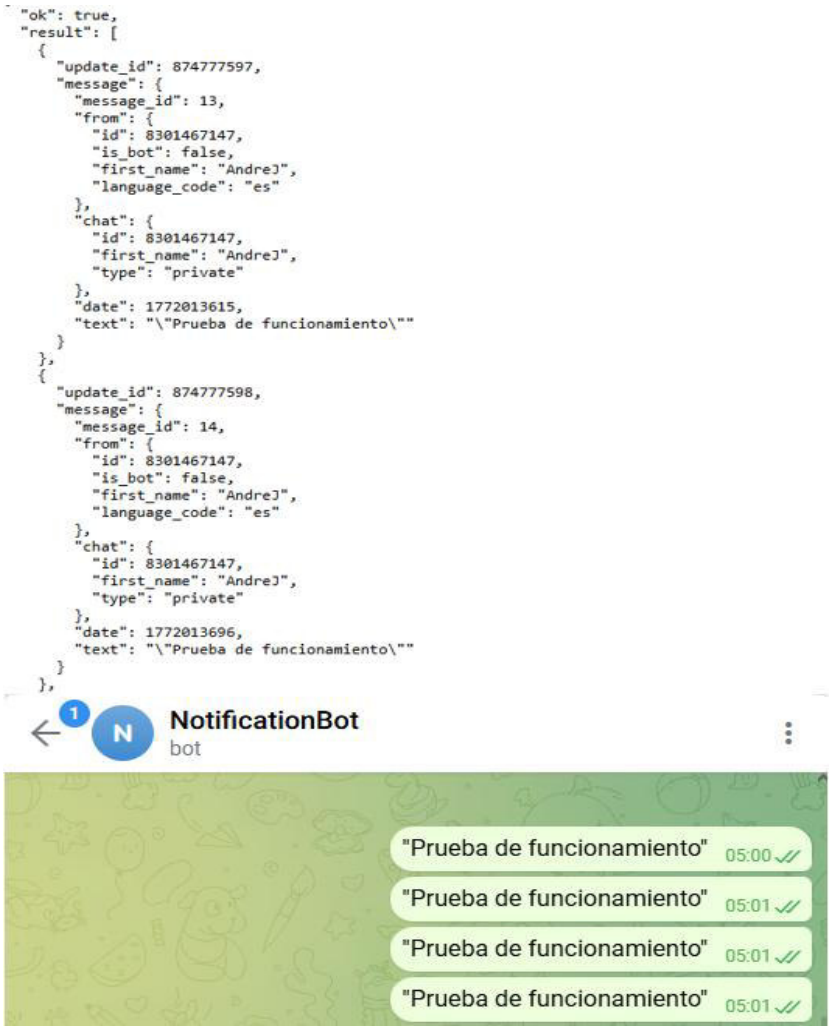


Figura 13 **Prueba de funcionamiento del token del bot.** Fuente. Elaboración propia

Segmentación Regional de Alertas

Se configuraron cuatro grupos de Telegram (ver Figura 14), uno por cada región operativa.

Resultado:

- Eliminación de saturación informativa.
- Cada equipo regional recibe únicamente sus eventos.
- Reducción de ruido operacional.

Esto permitió mejorar la organización interna del NOC y reducir tiempos de identificación de responsabilidad.



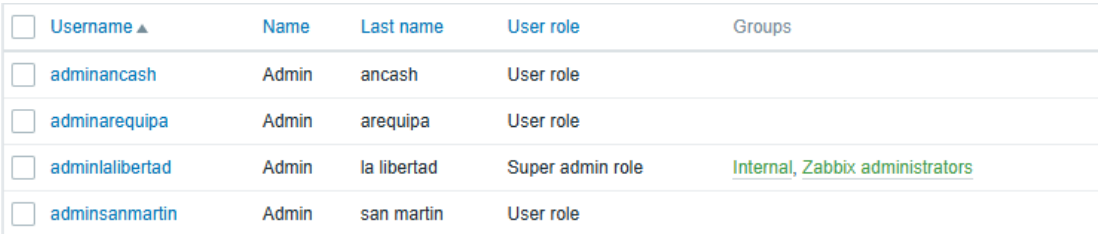
Figura 14 **Grupos regionales creados en Telegram.** Fuente. Elaboración propia

Usuarios Regionales en Zabbix

Se crearon usuarios específicos por región, aplicando el principio de mínimo privilegio (ver Figura 15).

Resultado:

- Acceso restringido por región.
- Separación clara de responsabilidades.
- Gobernanza del sistema fortalecida.



Username ▲	Name	Last name	User role	Groups
adminancash	Admin	ancash	User role	
adminarequipa	Admin	arequipa	User role	
adminlalibertad	Admin	la libertad	Super admin role	Internal, Zabbix administrators
adminsanmartin	Admin	san martin	User role	

Figura 15 **Usuarios regionales creados en Telegram.** Fuente. Elaboración propia

Script de Envío Validado

El script desarrollado para integración con Telegram fue probado bajo el test de la misma plataforma Zabbix como se evidencia en la Figura 16.



Figura 16 **Test de envío de alerta.** Fuente. Elaboración propia

3.6.3 Personalización de Plantillas de Mensaje

Se modificaron las plantillas predeterminadas de Zabbix para incluir únicamente información relevante para el análisis inmediato del incidente. Estos fueron probados bajo el criterio de:

- Evento crítico (Ver Figura 17)

- Evento de recuperación (Ver Figura 18)

Resultado:

- 100% de mensajes entregados durante pruebas.
- Sin pérdida de notificaciones.
- Correcta interpretación de variables dinámicas.
- Reducción del contenido irrelevante.
- Mejora en legibilidad.
- Identificación visual rápida.

Esto permitió disminuir el tiempo de interpretación del evento.

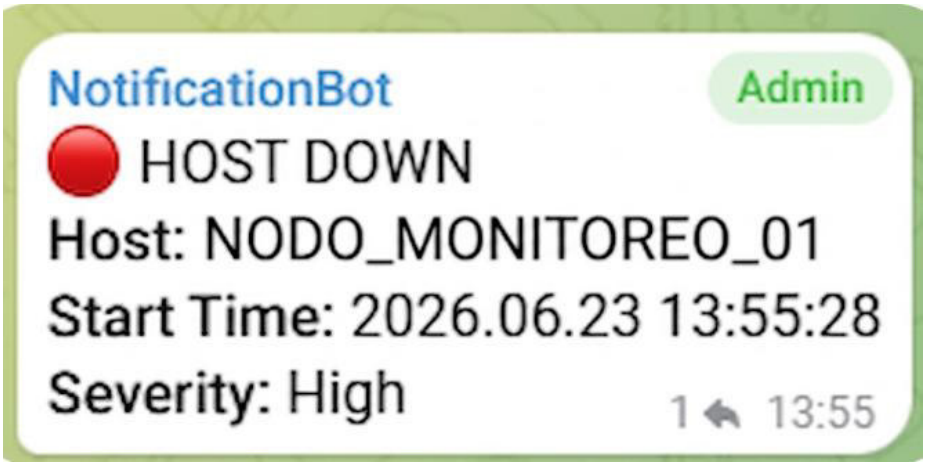


Figura 17 **Mensaje de Evento critico en Telegram.** Fuente. Elaboración propia

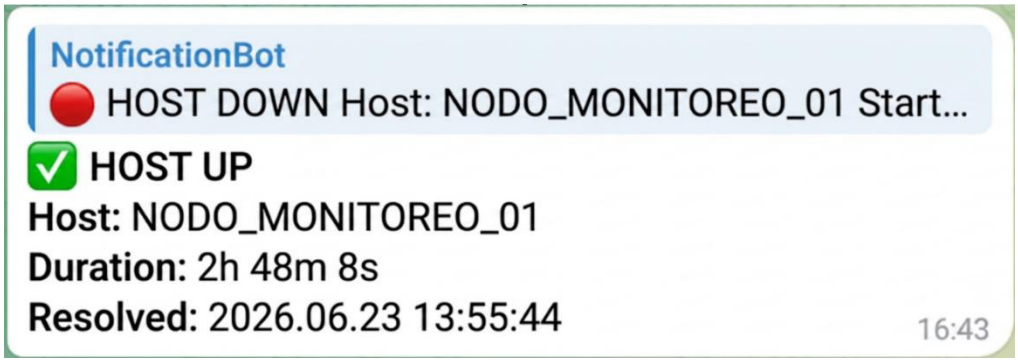


Figura 18 **Mensaje de Fin de Evento en Telegram.** Fuente. Elaboración propia

3.6.4 Dashboards Desplegados

Dashboard Corporativo

Se implementó un Dashboard principal (Ver Figura 19) con:

- Estado del servidor.
- Resumen global de las regiones.
- Top problemas críticos (Nodos y Enlaces).
- Métricas agregadas.

Resultado:

- Visión consolidada en un solo panel.
- Evaluación ejecutiva inmediata.
- Mejora en supervisión estratégica.

Entre los widgets desplegados destaca el panel de información del sistema, que resume en tiempo real el estado agregado de la plataforma: número total de hosts activos, plantillas aplicadas, ítems habilitados y triggers en ejecución, junto con el indicador NVPS (New Values Per Second) que refleja la carga real de ingesta de datos sobre el servidor. Este panel permite al supervisor verificar, sin necesidad de consultar cada región por separado, que la infraestructura de monitoreo está operando dentro de parámetros normales antes de revisar el detalle de los incidentes activos.

Complementan este panel dos widgets de estado agregado uno para problemas de enlaces y otro para problemas de hosts que distinguen visualmente, mediante un código de colores por severidad, si la atención del NOC debe dirigirse hacia la disponibilidad de los equipos o hacia el estado de los enlaces de transporte entre ellos. Debajo de estos widgets, dos tablas de "problemas actuales" listan cada incidente activo con su host, severidad, duración y opciones de actualización directa, permitiendo al operador gestionar el evento sin salir del dashboard corporativo.

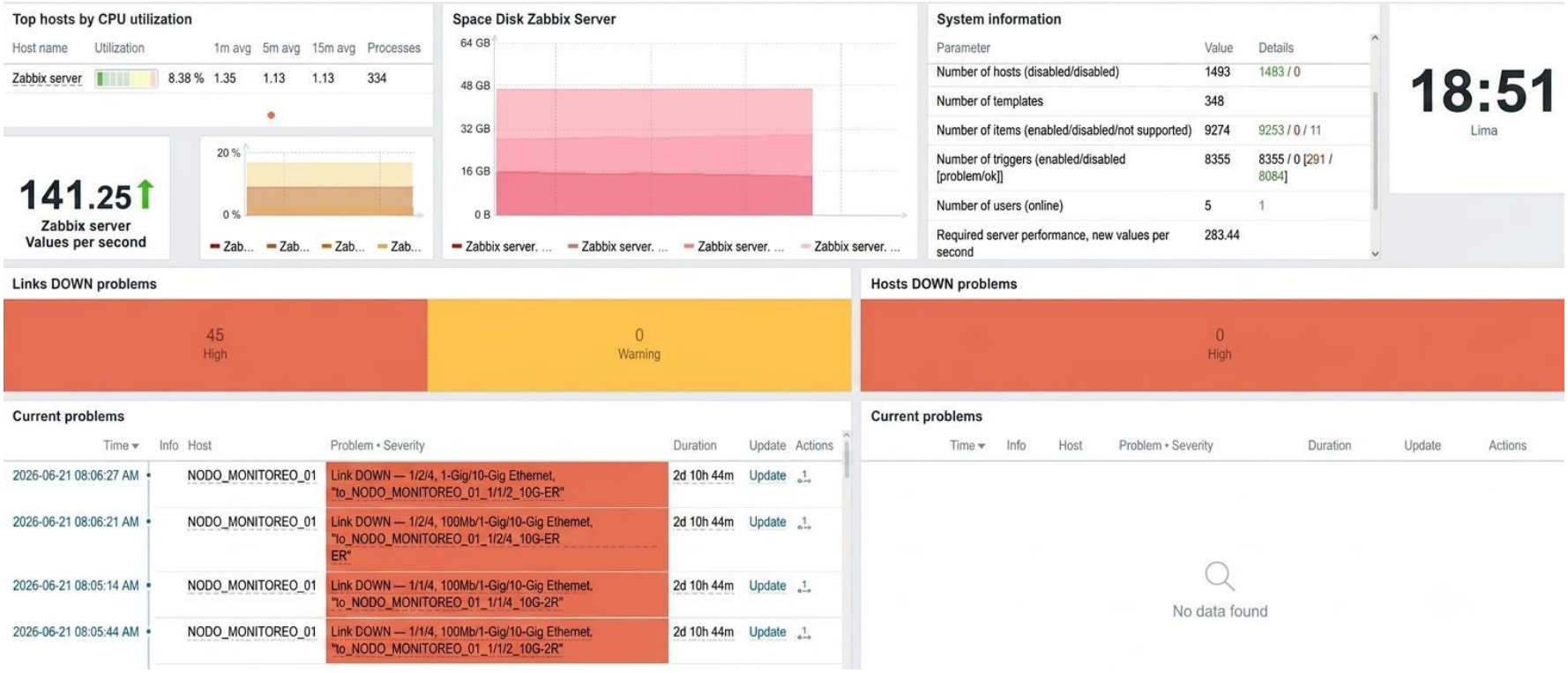


Figura 19 **Vista del Dashboard Corporativo.** Fuente. Captura de pantalla de la plataforma Zabbix ocultando información privada

Dashboards Regionales

Cada región cuenta con Dashboard específico que incluye:

- Problemas por severidad.
- Lista de hosts caídos.
- Historial de recuperación.

Resultado:

- Mejora en enfoque operativo.
- Identificación inmediata de equipos sin gestión.
- Reducción de dependencia de filtrado manual.

A diferencia del dashboard corporativo, los dashboards regionales separan explícitamente los problemas de la red de transporte de los problemas de la red de acceso, replicando la misma distinción funcional definida desde la etapa de planificación de la arquitectura. Cada uno de estos widgets muestra, además del conteo agregado de incidentes por severidad, una tabla detallada con los enlaces afectados, identificados por su nombre de interfaz y el nodo al que pertenecen, lo que permite al operador regional ubicar el problema sin necesidad de interpretar previamente la topología completa de la región.

Esta separación por tipo de red dentro de un mismo dashboard regional resultó especialmente útil durante los primeros días de operación posteriores a la ampliación del monitoreo SNMP, cuando el volumen de enlaces bajo supervisión creció de forma considerable y era necesario distinguir con rapidez si un incremento en los problemas reportados correspondía a la red de transporte o a la de acceso.

Lo anterior lo podemos confirmar a través de la Figura 20.

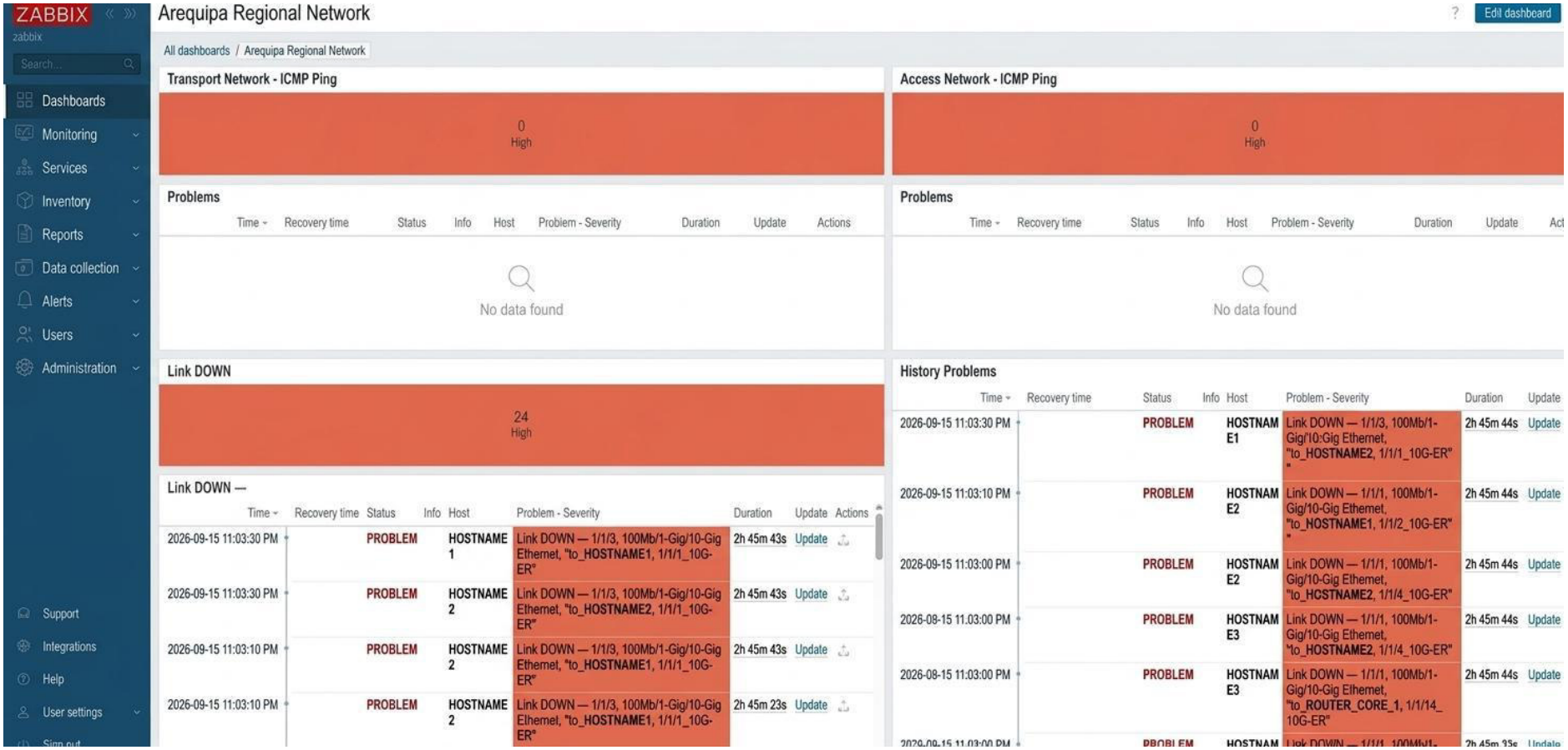


Figura 20 **Vista del Dashboard Regional Arequipa**. Fuente. Captura de pantalla de la plataforma Zabbix ocultando información privada y tomando como referencia la región de Arequipa

3.6.5 Estructura Organizacional Implementada

Se creó estructura lógica completa en Zabbix con:

- Grupos de Routers.
- Gestores Auxiliares.
- Componentes de Energía.
- Seguridad Física.

Resultado:

- Representación coherente de arquitectura real.
- Mejor correlación de eventos.
- Escalabilidad futura garantizada.

3.6.6 Impacto en el rendimiento operativo del NOC

La implementación del sistema de monitoreo basado en Zabbix fortaleció la capacidad operativa del Network Operations Center (NOC), proporcionando mecanismos automatizados de supervisión, alertamiento y visualización en tiempo real. La integración de los diferentes componentes de la infraestructura dentro de una única plataforma facilitó la gestión de incidentes y optimizó los procesos de detección y respuesta operativa.

Desde el punto de vista funcional, la solución implementada hizo posible centralizar la supervisión de routers, gestores auxiliares, sistemas de energía y dispositivos de seguridad física, mejorando la capacidad del personal técnico para identificar eventos críticos y priorizar incidentes de acuerdo con su impacto operativo. Asimismo, la automatización de notificaciones mediante Telegram estableció un esquema de alertamiento inmediato segmentado por región, fortaleciendo los procesos de

escalamiento y atención de fallas.

Para evaluar el impacto de la solución implementada en la capacidad de detección de incidentes, se utilizó como referencia el indicador Mean Time to Detect (MTTD), el cual representa el tiempo promedio transcurrido entre la ocurrencia de una falla y su identificación por el sistema de monitoreo. Este indicador es ampliamente utilizado en la gestión de servicios de tecnologías de la información para medir la eficiencia de los procesos de supervisión *(Axelos, 2019)*.

En el entorno previo a la implementación, la detección de incidentes dependía principalmente de procesos manuales de revisión y filtrado de alarmas, lo que impedía contar con registros sistemáticos y precisos de los tiempos de detección. Debido a ello, la evaluación se basó en observaciones operativas y en la experiencia del personal del NOC, sin que fuera posible establecer una base histórica estructurada para un análisis estadístico formal.

Con la implementación de Zabbix, se analizó una muestra de 30 eventos reales de indisponibilidad registrados sobre la base de datos de producción del servidor, lo que permitió cuantificar los tiempos de detección del sistema. La ventana de detección, medida entre la última lectura de disponibilidad registrada y la creación del problema por parte de Zabbix, arrojó un promedio de 182 segundos sobre la muestra analizada (ver Tabla 20), resultado coherente con el criterio de tres fallos consecutivos configurado para el disparador de indisponibilidad total (ver Tabla 6). Estos resultados son consistentes con lo reportado en la literatura especializada sobre monitoreo automatizado, donde la centralización y automatización del alertamiento contribuyen a la reducción del MTTD y a la mejora de la disponibilidad operativa de los servicios, aun cuando la comparación directa frente al modelo previo no fue posible por la ausencia de registros históricos estructurados en dicho entorno. Esta comparación cualitativa se sintetiza en la Tabla 21.

Tabla 20 **Registro de eventos reales de indisponibilidad y ventana de detección**

N° de evento	Hora de Ultima	Hora Creación	Ventana de

	lectura “UP”	del problema	detección (s)
1	14:42:10	14:45:42	212
2	22:22:32	22:26:33	241
3	12:18:54	12:23:23	269
4	11:30:28	11:33:55	207
5	11:46:42	11:50:05	203
6	07:50:22	07:54:20	238
7	08:56:53	09:00:42	229
8	10:32:39	10:36:39	240
9	15:06:15	15:10:30	255
10	08:13:36	08:16:52	196
11	06:28:02	06:30:59	177
12	01:00:35	01:03:25	170
13	08:25:20	08:28:29	189
14	21:10:51	21:14:34	223
15	22:47:14	22:51:07	233
16	04:17:28	04:21:19	231
17	09:13:34	09:17:09	215
18	05:16:38	05:20:09	211
19	06:45:21	06:48:38	197
20	01:03:31	01:06:53	202
21	10:56:09	11:00:05	236

22	20:07:16	20:08:45	89
23	17:47:24	17:48:33	69
24	10:30:45	10:32:00	75
25	14:11:30	14:13:21	111
26	00:41:38	00:42:59	81
27	05:08:32	05:12:02	210
28	10:47:34	10:48:43	69
29	14:18:20	14:19:46	86
30	15:55:11	15:56:47	96
Promedio			**182**

Fuente. Elaboración propia basado en eventos reales de indisponibilidad registrado en la base de datos del servidor Zabbix.

Este resultado es consistente con el diseño del disparador de disponibilidad total, el cual exige tres fallos consecutivos del chequeo ICMP (Tabla 6) sobre un ítem configurado con un intervalo de actualización de 50 segundos (Tabla 4), produciendo una ventana mínima teórica de 150 segundos, el promedio empírico obtenido sobre la muestra de producción (182 segundos) se ubica ligeramente por encima de este mínimo teórico, diferencia atribuible al tiempo de espera hasta el siguiente ciclo de sondeo programado tras la falla real del equipo, así como a la variabilidad del procesamiento del poller de Zabbix bajo carga.

Esta configuración responde a un criterio deliberado de equilibrio entre sensibilidad de detección y minimización de falsos positivos por pérdida transitoria de paquetes (Stallings, 2018; Comer, 2018)

Tabla 21 **Comparación cualitativa del proceso de detección de incidentes (antes y después de la implementación)**

Modelo	Método de detección	Características operativas
Modelo previo (Pre-implementación)	Revisión manual de alarmas	Dependencia de filtrado y supervisión continua
Modelo actual (Post-implementación)	Alertamiento automatizado (trigger+notificación de Telegram)	Alertamiento automatizado y visualización en tiempo real

Fuente. Elaboración propia. Nota: La comparación presentada es de carácter cualitativo y no incluye valores cuantitativos ante la ausencia de registros históricos estructurados en el entorno previo.

La tabla anterior resume comparativamente ambos modelos de detección, evidenciando el contraste entre la supervisión manual previa y el esquema automatizado implementado. Si bien la ausencia de registros históricos estructurados impidió realizar un análisis estadístico formal del MTTD, el registro de eventos reales del sistema implementado demuestra una mejora en la capacidad de detección y respuesta operativa del NOC.

3.6.7 Resultados del Monitoreo SNMP de Enlaces

La implementación del monitoreo SNMP permitió extender la supervisión de enlaces a los 1493 hosts ya monitoreados vía ICMP, validando la viabilidad del enfoque automatizado para redes de transporte regionales de gran tamaño. Durante la fase de despliegue se identificaron y resolvieron dos incidentes críticos de infraestructura no anticipados en la planificación inicial: agotamiento de la memoria caché de configuración (CacheSize) y saturación del disco del servidor por crecimiento no controlado del historial. Los incidentes se resumen en la Tabla 22.

Tabla 22 **Incidentes de infraestructura identificados y resueltos**

Incidente	Causa	Solución implementada
Agotamiento de CacheSize	Memoria insuficiente (32 MB por defecto) para 1,493 hosts	Incremento a 256 MB
Rápido incremento de uso de disco duro por historial	Memoria insuficiente para uso del cache y grandes volúmenes de datos	Aumento del cache para historial e implementación del TimescaleDB

Fuente. Elaboración propia

Como resultado complementario, se implementó un esquema de monitoreo proactivo del propio servidor Zabbix (self-monitoring), mediante widgets de tendencia de disco y caché (Ver Figura 21), además de triggers de alerta al 80% de uso, transformando la detección de estos incidentes de reactiva a proactiva para futuras incorporaciones de hosts. También se verifica la disminución de uso de espacio al implementar las soluciones.

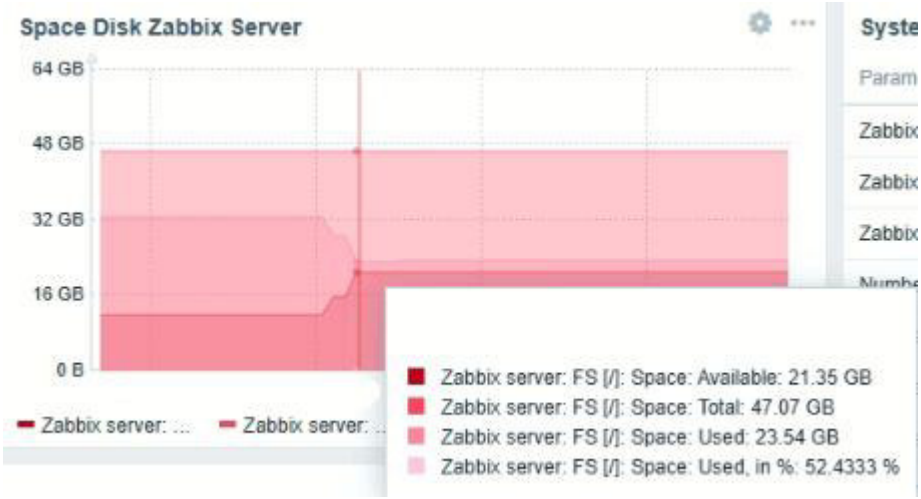


Figura 21 **Ejemplo de Widget de tendencia de uso de memoria de disco del servidor Zabbix (4 parámetros monitoreados).** Fuente. Captura de pantalla desde Zabbix.

Como resultado directo de la resolución del incidente de agotamiento de memoria caché descrito en la Tabla 22, se procedió a recalcular y ajustar de manera integral los cinco parámetros de caché del servidor Zabbix, considerando la carga proyectada para los 1,493 hosts monitoreados mediante SNMP. La Tabla 23 presenta el dimensionamiento final configurado para cada parámetro, contrastando los valores por defecto de la plataforma con los valores ajustados tras el análisis de capacidad.

Tabla 23 **Dimensionamiento final de memoria caché del servidor Zabbix**

Parámetro	Valor por defecto (MB)	Nuevo valor configurado (MB)
CacheSize	32	256
ValueCacheSize	8	256
HistoryCacheSize	16	128
HistoryIndexCacheSize	4	64
TrendCacheSize	4	64

Fuente. Elaboración propia.

La extensión TimescaleDB fue configurada con políticas de compresión automática segmentadas por itemid, con umbrales diferenciados de 7 días para las tablas de historial y 30 días para las tablas de tendencias.

Tras la ejecución de la primera compresión sobre el historial acumulado se registró una reducción del espacio ocupado en disco consistente con lo documentado en la literatura especializada para cargas de monitoreo de red con alta repetitividad de valores (*Timescale Inc., 2024a; initMAX, 2025b)* y cuyo resultado podemos observar en la Tabla 24.

Tabla 24 **Resultado de la compresión TimescaleDB sobre la tabla de historial principales durante la primera semana de implementación**

Tabla	Espacio pre compresión	Espacio Post compresión	Reducción	Reducción acumulada
History	4,387 MB	212 MB	95%	4,175 MB
History_uint	5,537 MB	239 MB	96%	9,473 MB
trends	244 MB	82 MB	66%	9,635 MB
Total	**10,168 MB**	**533 MB**	**95%**	**9,635 MB**

Fuente. Elaboración propia.

CAPÍTULO IV:
DISCUSIÓN DE RESULTADOS

4.1 Descripción de la implementación

4.1.1 Transformación del Modelo Operativo del NOC

La implementación del sistema de monitoreo basado en Zabbix generó una transformación estructural en el modelo de supervisión del Network Operations Center (NOC), modificando la dinámica operativa desde un esquema predominantemente reactivo hacia un modelo proactivo y automatizado.

Previo a la implementación, la supervisión dependía del filtrado manual de alarmas en plataformas independientes, lo cual implicaba:

- Fragmentación de información.
- Mayor carga cognitiva.
- Dependencia de revisión constante.
- Posibles retrasos en la detección.

Tras la implementación, se estableció:

- Monitoreo centralizado.
- Alertamiento automatizado.
- Segmentación regional.
- Visualización estructurada.

Para evidenciar esta transformación, se presenta la siguiente comparación de la Tabla 25.

Tabla 25 **Comparación estructural del modelo de monitoreo**

Dimensión	Modelo Previo	Modelo Implementado
Arquitectura	Fragmentada	Centralizada
Detección	Manual	Automática
Notificación	Dependía de revisión	Telegram en tiempo real
Visualización	Múltiples consolas	Dashboard unificado
Segmentación	No estructurada	Regional y funcional
Cierre de incidentes	Manual	Automático

Fuente. Elaboración propia.

La comparación evidencia una transición clara hacia un modelo alineado con buenas prácticas de gestión de redes. Según *Stallings (2018)*, la gestión de fallas efectiva requiere automatización, correlación y visibilidad consolidada.

El modelo previo carecía de integración y dependía de intervención humana constante, lo que incrementaba la probabilidad de retrasos en la detección. En contraste, el modelo implementado reduce la dependencia manual y mejora la consistencia operativa.

Desde el marco ITIL *(Axelos, 2019)*, la automatización del alertamiento impacta directamente en la fase de identificación del incidente, reduciendo el tiempo promedio de detección (MTTD).

4.1.1.1 Evaluación de las proposiciones teóricas orientadoras.

A partir de los resultados descritos en el Capítulo III, es posible evaluar el grado de cumplimiento de las proposiciones formuladas en la sección 3.4.1.

Respecto a la proposición N°1, la evidencia recogida confirma la proposición de manera parcial: la consolidación de ocho plataformas independientes en un tablero único redujo efectivamente la fragmentación informativa, tal como establece el modelo FCAPS. Sin embargo, esta centralización por sí sola no garantizó la detección temprana de incidentes; fue necesaria su combinación con alertamiento automatizado (P2), lo que confirma el carácter de condición necesaria pero no suficiente planteado inicialmente.

Respecto a proposición N°2, los resultados son consistentes con la proposición: la incorporación de triggers ICMP y notificaciones automatizadas vía Telegram permitió identificar eventos de indisponibilidad en ventanas de pocos minutos, según el registro real de eventos analizado (ver Tabla 20), mientras que en el modelo previo basado en revisión manual sobre plataformas centralizadas de gestión (NSP NFM-P), la detección dependía de la frecuencia de revisión del operador. Esto sugiere que la automatización del alertamiento, más que la centralización en sí constituye el factor determinante de la reducción del MTTD.

Respecto a proposición N°3, la segmentación regional y funcional implementada mostró beneficios consistentes con la proposición: la distribución de alertas por región eliminó la saturación informativa reportada en el diagnóstico inicial y permitió una asignación más clara de responsabilidades operativas.

Respecto a proposición N°4, la incorporación posterior del monitoreo SNMP mediante Low-Level Discovery confirmó esta proposición: el monitoreo ICMP por sí solo no permitía distinguir un puerto deshabilitado intencionalmente (Admin=down) de una falla real de enlace (Admin=up, Oper=down), distinción que solo fue posible mediante la incorporación de SNMP, validando empíricamente la complementariedad señalada por la literatura.

En conjunto, la evaluación de estas proposiciones aporta una lectura teórica adicional a los resultados: la transición de un modelo reactivo a uno proactivo en el caso

analizado dependió de la combinación progresiva de varios factores tecnológicos: centralización, automatización, segmentación organizacional y ampliación del alcance del protocolo de monitoreo, hallazgo que es consistente con el carácter multifactorial señalado en el problema general de investigación

4.1.2 Impacto en Indicadores Operativos

Como se evidenció en la Tabla 21, el modelo implementado sustituyó la revisión manual por un mecanismo de alertamiento automatizado, lo que se reflejó en el registro real de eventos analizado.

La implementación fortaleció la capacidad de supervisión y detección de incidentes en el entorno del NOC. En el modelo previo, la identificación de eventos dependía de factores como la frecuencia de revisión de alarmas, la carga operativa del personal y la necesidad de supervisar múltiples plataformas de manera simultánea.

Con el sistema implementado, la configuración de triggers ICMP y el envío automatizado de notificaciones mediante Telegram dieron lugar a un mecanismo de alertamiento inmediato, reduciendo la dependencia de procesos manuales y optimizando la capacidad de respuesta operativa. Sobre una muestra de 30 eventos reales registrados en producción el tiempo transcurrido entre la creación del problema en Zabbix y el envío efectivo de la notificación a Telegram registró un promedio de 3.1 segundos, con un máximo de 4 segundos (ver Tabla 26), lo que confirma de forma cuantitativa la inmediatez atribuida al mecanismo de alertamiento.

Tabla 26 **Registro de eventos reales y tiempo de notificación vía Telegram**

N° de evento	Hora de creación del problema	Hora de envío a Telegram	Tiempo de notificación (s)
1	14:45:42	14:45:45	3

2	22:26:33	22:26:36	3
3	12:23:23	12:23:26	3
4	11:33:55	11:33:58	3
5	11:50:05	11:50:08	3
6	07:54:20	07:54:23	3
7	09:00:42	09:00:45	3
8	10:36:39	10:36:42	3
9	15:10:30	15:10:33	3
10	08:16:52	08:16:55	3
11	06:30:59	06:31:02	3
12	01:03:25	01:03:29	4
13	08:28:29	08:28:32	3
14	21:14:34	21:14:37	3
15	22:51:07	22:51:10	3
16	04:21:19	04:21:22	3
17	09:17:09	09:17:12	3
18	05:20:09	05:20:12	3
19	06:48:38	06:48:41	3
20	01:06:53	01:06:56	3
21	11:00:05	11:00:08	3
22	20:08:45	20:08:48	3

23	17:48:33	17:48:37	4
24	10:32:00	10:32:03	3
25	14:13:21	14:13:25	4
26	00:42:59	00:43:02	3
27	05:12:02	05:12:05	3
28	10:48:43	10:48:46	3
29	14:19:46	14:19:49	3
30	15:56:47	15:56:50	3

Fuente. Elaboración propia a partir de registro real de eventos extraído de la base de datos de producción del servidor Zabbix.

4.1.2.1 Estadísticos descriptivos de la ventana de detección y del tiempo de notificación

. Con el propósito de complementar los valores promedio ya reportados en las Tablas 20 y 26, se calcularon los estadísticos descriptivos correspondientes a la ventana de detección y al tiempo de notificación sobre la muestra de 30 eventos reales (ver Tabla 27)

Para la ventana de detección la distribución muestra una mediana de 205.0 segundos, ligeramente superior a la media de 182.0 segundos, lo que sugiere la presencia de un grupo de eventos con tiempos de detección notablemente más cortos (entre 69 y 111 segundos) que empujan el promedio hacia abajo. El 75% de los eventos se detectó en un tiempo igual o menor a 230.5 segundos, mientras que el 25% inferior se ubicó por debajo de 125.75 segundos, con una desviación estándar muestral de 63.76 segundos que confirma una dispersión considerable dentro de la muestra.

En cuanto al tiempo de notificación a Telegram, el comportamiento es mucho más uniforme: de los 30 eventos analizados, 27 (90%) se notificaron en 3 segundos y los 3 restantes (10%) en 4 segundos, con una desviación estándar de apenas 0.31 segundos.

Esta consistencia confirma que el componente de notificación opera con un margen de variación mínimo, independientemente del tipo o la duración del evento de origen.

Tabla 27 **Análisis estadístico descriptivo de los indicadores de detección y notificación**

Estadístico	Ventana de detección (Tabla 20, s)	Tiempo de notificación (Tabla 26, s)
Media	182	3.1
Mediana	205	3
Desviación estándar (muestral)	63.76	0.31
Mínimo	69	3
Máximo	269	4
Percentil 25 (Q1)	125.75	-
Percentil 75 (Q3)	230.5	-

Fuente. Elaboración propia. Nota: Los cuartiles se calcularon con el procedimiento estándar descrito para conjuntos de datos continuos (Navidi, 2015; Field, 2018).

4.1.2.2 Análisis de correlación entre la hora del evento y la ventana de detección.

Con el fin de descartar un posible efecto de la carga horaria del servidor sobre el desempeño del poller de Zabbix, se evaluó si la ventana de detección guarda relación con el momento del día en que ocurre el evento. Para ello se calculó el coeficiente de correlación de Pearson entre la hora del evento expresada en minutos desde la medianoche, tomando como referencia la última lectura de disponibilidad registrada antes de la falla (columna "Hora de última lectura UP" de la Tabla 20) y la ventana de detección en segundos, sobre los 30 registros disponibles.

El coeficiente obtenido fue r = −0.037, un valor que indica una correlación prácticamente nula entre ambas variables. Esto sugiere que la capacidad de detección del sistema se mantiene estable a lo largo de las 24 horas del día, sin que la franja horaria en la que ocurre la falla incida de manera relevante sobre el tiempo que toma identificarla. El resultado es consistente con el diseño del disparador de indisponibilidad total, que depende únicamente del intervalo de sondeo ICMP (50 segundos) y del umbral de tres fallos consecutivos configurado en la Tabla 6, ambos parámetros fijos e independientes de la hora del día o de la carga del servidor.

En conjunto, la reducción de la ventana de detección, la inmediatez del mecanismo de notificación vía Telegram y la estabilidad temporal confirmada mediante el análisis de correlación tienen implicancias directas en aspectos relacionados con la gestión de incidentes, tales como:

- Fortalecimiento de la detección temprana de fallas.
- Optimización del proceso de respuesta operativa.
- Mejora de la visibilidad sobre eventos críticos.
- Contribución al cumplimiento de SLA operativos.

Desde el enfoque de gestión de servicios de TI, la automatización del monitoreo representa uno de los factores más relevantes para fortalecer la eficiencia del proceso de gestión de incidentes, especialmente en entornos distribuidos donde la supervisión manual genera limitaciones operativas.

Narayan et al. (2023), en una revisión sistemática sobre detección de anomalías en entornos de monitoreo con Nagios y Zabbix, señalan que la integración de mecanismos automatizados de alertamiento incluyendo enfoques basados en inteligencia artificial contribuye a reducir los tiempos de identificación de incidentes en infraestructuras de gran escala, especialmente en entornos con alta densidad de eventos y múltiples nodos supervisados simultáneamente. Cabe precisar que dicho estudio se enmarca en escenarios donde se incorporan modelos de aprendizaje automático para el análisis de logs, condición que difiere del presente trabajo, en el cual la detección se basa en triggers configurados sobre umbrales definidos sin componente de inteligencia

artificial. No obstante, estos resultados se alinean con los principios establecidos en ITIL *(Axelos, 2019)*, que enfatizan la importancia de la automatización y la detección temprana en la gestión de incidentes, así como con el modelo FCAPS, en el cual la gestión de fallas depende directamente de la capacidad de monitoreo continuo.

En este sentido, la solución implementada responde a una necesidad específica del entorno analizado y se encuentra alineada con las prácticas recomendadas internacionalmente para la supervisión de infraestructuras críticas de telecomunicaciones.

Narayan et al. (2023) reportaron una reducción del 25% en el MTTD mediante la integración de inteligencia artificial en el monitoreo con Zabbix. En este caso, sin la aplicación de IA el análisis de una muestra de 30 eventos reales de producción registró una ventana de detección promedio de 182 segundos (ver Tabla 20). Si bien no es posible establecer una comparación porcentual directa con el estudio de Narayan et al. dada la ausencia de una línea base cuantitativa en el modelo previo de la implementación, el orden de magnitud obtenido es consistente con la tendencia reportada en la literatura. Esta convergencia sugiere que la automatización del alertamiento, incluso sin componentes de IA, puede generar mejoras sustanciales en la detección temprana, siempre que esté acompañada de una segmentación regional efectiva y una infraestructura de base de datos optimizada.

A diferencia del estudio de Narayan et al., el presente caso evidencia que la escalabilidad del sistema (1,493 hosts) introduce requerimientos de infraestructura subyacente (base de datos, memoria) para absorber el incremento de datos que no son tratados en la literatura revisada, lo que constituye un aporte empírico.

4.1.3 Evaluación del cumplimiento de SLA y disponibilidad del servicio

Con el objetivo de evaluar el impacto de la implementación del sistema de monitoreo en la continuidad del servicio, se realizó un análisis de la disponibilidad de los routers supervisados, tomando como referencia los acuerdos de nivel de servicio (SLA) establecidos para la operación de la red.

La disponibilidad se define como la proporción de tiempo en la que un sistema se encuentra operativo respecto al tiempo total observado, y se calcula mediante la siguiente expresión:

Disponibilidad (%)=((Tiempo total −Tiempo de indisponibilidad)/ Tiempo total)×100

En el entorno previo a la implementación, la ausencia de un sistema de monitoreo centralizado impedía contar con registros precisos del tiempo de indisponibilidad, por lo que no era posible calcular la disponibilidad de manera objetiva. La identificación de fallas dependía de procesos manuales, lo cual limitaba la trazabilidad de los eventos.

Posteriormente, con la implementación de Zabbix, se logró registrar de manera automática los eventos de indisponibilidad de los dispositivos, permitiendo estimar la disponibilidad del servicio por región.

4.1.4 Evaluación del desempeño de la infraestructura

Además del monitoreo de disponibilidad implementado mediante ICMP, se incorporó monitoreo basado en el protocolo SNMPv3 para evaluar el estado de los enlaces de transporte de la red, descrito en la sección 3.5.7, lo cual permitió ampliar la visibilidad operativa más allá de la disponibilidad básica de los equipos.

Las métricas de desempeño de dispositivos (CPU, memoria, tráfico de interfaces y errores de transmisión) descritas a continuación constituyen una extensión natural de este mismo enfoque SNMP, aplicable a futuro sobre el conjunto completo de dispositivos de la red. Es decir, corresponden a la capacidad futura del sistema, no a la implementación actual.

Métricas consideradas

Las principales métricas de desempeño relevantes en el contexto del NOC son:

- Utilización de CPU de los routers
- Uso de memoria

- Tráfico de interfaces (ingreso y egreso)
- Errores de transmisión (CRC)
- Paquetes descartados

Estas métricas permiten identificar condiciones de sobrecarga o degradación antes de que se produzca una falla total del servicio.

Análisis de comportamiento

El monitoreo de estas variables permite detectar patrones como:

- Incremento progresivo del uso de CPU
- Saturación de enlaces
- Pérdida de paquetes
- Comportamientos anómalos en interfaces

Lo cual facilita la toma de decisiones preventivas y la planificación de capacidad.

La Figura 22 muestra las variaciones del uso de CPU a lo largo del tiempo, evidenciando un comportamiento dinámico con picos que alcanzan valores cercanos al umbral de saturación.

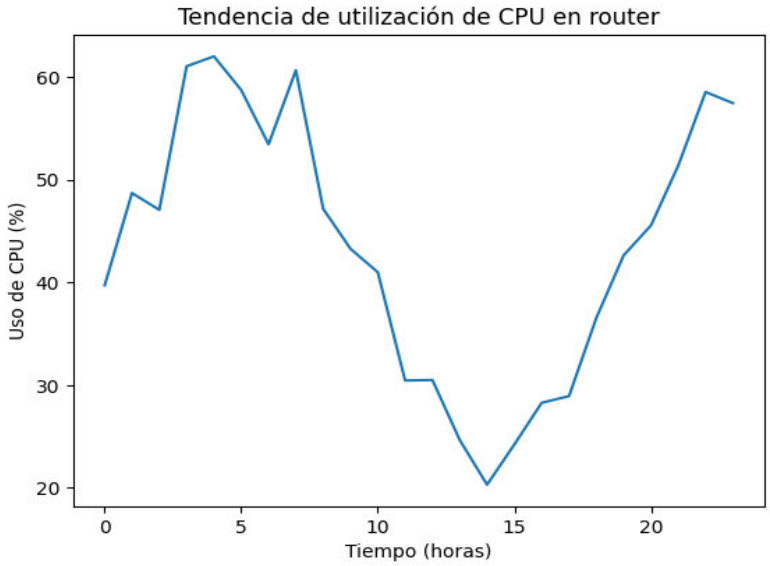


Figura 22 **Tendencia de Utilización de CPU en router.** Fuente. Elaboración propia

Este tipo de comportamiento es característico de entornos donde la carga de procesamiento depende del volumen de tráfico y de la cantidad de sesiones activas en el dispositivo.

Se observa que, aunque el uso promedio se mantiene en niveles aceptables, existen incrementos periódicos que podrían comprometer el rendimiento del router si se mantienen de forma sostenida. En este contexto, el monitoreo continuo de la CPU permite identificar tendencias de crecimiento que podrían anticipar la necesidad de optimización o ampliación de recursos.

Con un enfoque de gestión proactiva, estos datos permiten establecer umbrales de alerta preventiva, evitando que el dispositivo alcance estados críticos que afecten la calidad del servicio.

La Figura 23 presenta la evolución del tráfico de entrada y salida en una interfaz de red, mostrando un comportamiento oscilante que refleja variaciones en la demanda del servicio. Se observa una correlación entre ambas curvas, lo que indica un flujo de datos bidireccional estable.

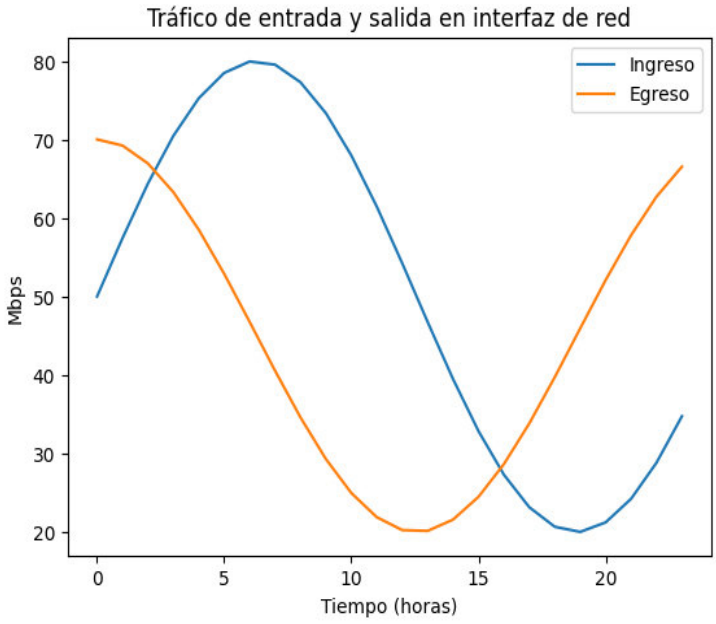


Figura 23 **Tráfico de entrada y Salida en interfaz de red.** Fuente. Elaboración propia

Sin embargo, se identifican periodos en los que el tráfico alcanza valores elevados, lo

cual podría generar congestión si se aproxima a la capacidad máxima del enlace. Este tipo de análisis resulta fundamental para la planificación de capacidad, ya que permite determinar si la infraestructura actual es suficiente para soportar la demanda.

De manera complementaria, el monitoreo de tráfico permite detectar anomalías, como picos inesperados, que podrían estar asociados a fallas, configuraciones incorrectas o incluso eventos de seguridad.

La Figura 24 muestra la cantidad de errores y paquetes descartados en una interfaz de red, evidenciando la presencia de eventos que afectan la calidad de la transmisión. Aunque los valores se mantienen relativamente bajos, la aparición recurrente de errores puede ser indicativa de problemas físicos en el enlace, interferencias o degradación de la señal.

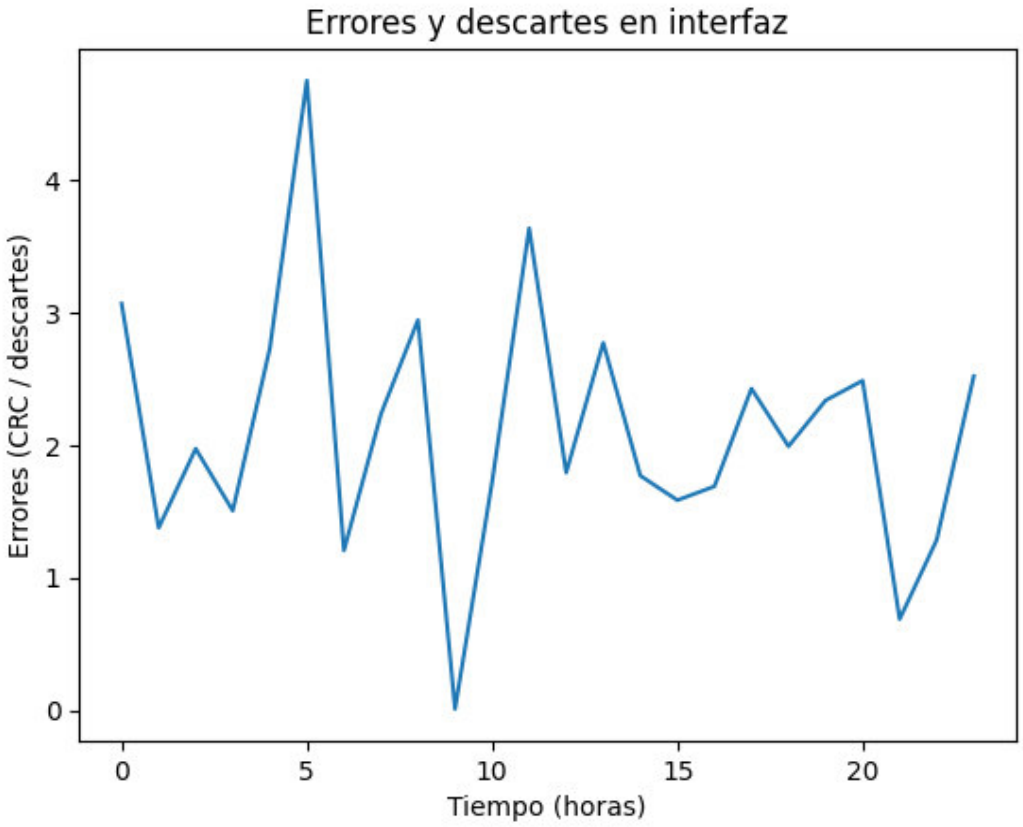


Figura 24 **Errores y descartes en interfaz.** Fuente. Elaboración propia

Este tipo de métricas resulta clave para la gestión de la calidad del servicio, ya que permite identificar fallas que no necesariamente generan una interrupción total, pero que sí impactan en el desempeño de la red.

El análisis de errores complementa el monitoreo de disponibilidad, proporcionando una visión más completa del estado de la infraestructura y permitiendo la detección temprana de problemas que podrían escalar a incidentes mayores.

En conjunto, el análisis de estas métricas permite evolucionar el sistema de monitoreo desde un enfoque reactivo basado en la detección de fallas hacia un modelo proactivo orientado a la gestión del desempeño. Esto se alinea con el modelo FCAPS, integrando tanto la gestión de fallas como la gestión de rendimiento, lo cual fortalece la capacidad operativa del NOC.

4.1.5 Discusión Crítica de la Arquitectura Implementada

Aunque la implementación demostró mejoras sustanciales, es necesario analizarla críticamente.

Fortalezas

- Uso de plataforma open source escalable.
- Automatización efectiva de alertas.
- Segmentación regional adecuada.
- Visualización estructurada.
- Bajo costo de implementación.

Limitaciones

- El monitoreo SNMP implementado se concentra en el estado de los enlaces sin embargo no cubre aún métricas de desempeño de dispositivo (CPU, memoria, errores de interfaz).

- No se configuró alta disponibilidad del servidor Zabbix.
- No se integró correlación avanzada de eventos.
- Dependencia de un único servidor.

Estas limitaciones no invalidan la implementación, pero representan oportunidades de mejora futura. Adicionalmente son desarrolladas con mayor detalle en la sección 4.1.8 del presente capítulo, donde se explica también el fundamento de cada una en el contexto del trabajo realizado.

Los resultados obtenidos en la presente implementación son consistentes con lo reportado en la literatura técnica en el ámbito de la gestión de redes y monitoreo centralizado. Diversos autores destacan que la automatización de alertas y la centralización de la supervisión contribuyen a fortalecer los procesos de detección y gestión de incidentes en infraestructuras críticas de telecomunicaciones.

De acuerdo con *Stallings (2018)*, los sistemas de monitoreo centralizado permiten mejorar la visibilidad operativa y optimizar la capacidad de respuesta ante eventos de red mediante mecanismos automatizados de supervisión y correlación de alarmas. *Axelos (2019)*, a través del marco ITIL, enfatiza que la automatización del incident management constituye un factor fundamental para fortalecer la continuidad operativa y reducir la dependencia de procesos manuales.

En este contexto, la solución implementada en el presente trabajo se alinea con las buenas prácticas internacionales de gestión de redes, al integrar monitoreo centralizado, alertamiento automatizado y visualización en tiempo real dentro de un entorno operativo distribuido.

4.1.6 Contraste con Buenas Prácticas

Para analizar el nivel de alineación de la solución implementada con estándares internacionales, se realizó una comparación basada en buenas prácticas de gestión de servicios de TI, tomando como referencia el marco ITIL *(Axelos, 2019)* y los lineamientos de gestión definidos en la norma ISO/IEC 20000 *(ISO, 2018)*, los cuales

establecen principios para la gestión eficiente de incidentes, monitoreo y disponibilidad del servicio. La Tabla 28 contrasta la situación previa a la implementación con la situación actual respecto a cinco buenas prácticas clave de gestión de servicios TI.

Tabla 28 **Comparación del modelo de monitoreo en función de buenas prácticas de gestión de servicios TI (ITIL e ISO/IEC 20000)**

Buenas Prácticas	Situación Previa	Situación Implementada
Gestión proactiva	No	Sí
Fault Management (FCAPS)	Parcial	Mejorado
Segmentación operativa	No estructurada	Formalizada
Automatización	Limitada	Implementada
Visualización ejecutiva	No	Sí

Fuente. Axelos (2019) e ISO (2018).

Al contrastar ambos escenarios se evidencia que la implementación realizada se alinea con estándares internacionales de gestión de servicios TI. En particular, la transición hacia un modelo de monitoreo centralizado y automatizado refleja la adopción de principios fundamentales de ITIL relacionados con la gestión proactiva de incidentes y la mejora continua del servicio *(Axelos, 2019)*. La estructuración del monitoreo y la estandarización de procesos también siguen los lineamientos establecidos por la norma ISO/IEC 20000, que promueve la eficiencia operativa, la trazabilidad y la garantía de disponibilidad del servicio *(ISO, 2018)*. Esta alineación valida técnicamente la solución implementada, posicionándola dentro de un marco reconocido de buenas prácticas a nivel internacional

La implementación introdujo una nueva herramienta tecnológica y redefinió el modelo

de supervisión del NOC, incorporando la automatización, la segmentación y la visualización estratégica.

El sistema implementado mejora el desempeño operativo al reducir tiempos de detección, fortalecer la gestión regional y proporcionar una arquitectura centralizada coherente con buenas prácticas internacionales.

Esa alineación con buenas prácticas internacionales es parcial: las limitaciones de la arquitectura ya identificadas en la sección 4.1.5, particularmente la ausencia de alta disponibilidad del servidor y de correlación avanzada de eventos constituyen brechas puntuales respecto de los lineamientos de continuidad y gestión de incidentes establecidos por ITIL e ISO/IEC 20000 y son retomadas como líneas prioritarias de evolución en el Capítulo VI (Recomendaciones).

4.1.6.1 Comparación con Otras Implementaciones de Zabbix en Telecomunicaciones.

Además del contraste contra marcos normativos como ITIL e ISO/IEC 20000, resulta pertinente comparar los resultados obtenidos con implementaciones de Zabbix documentadas en otros operadores de telecomunicaciones, identificando similitudes, diferencias y lecciones transferibles. La Tabla 29 resume esta comparación frente a tres casos de escala considerablemente mayor: Hrvatski Telekom, O2 Czech Republic y LMT Latvia.

Tabla 29 **Resumen comparativo de la implementación Zabbix de YOFC PERU SAC con Hrvatski Telekom, O2 Czech Republic y LMT Latvia**

Aspecto	Hrvatski Telekom	O2 Czech Republic	LMT Latvia	YOFC PERÚ S.A.C. (presente caso)
Escala	17,000 hosts,	Migración	3,000 hosts, ~500,000	1,493

	15M ítems, 800k triggers	completa desde IBM Netcool	ítems	hosts
Arquitectura	30 proxies regionales, TimescaleDB (4 TB, ~75% compresión)	Plataforma unificada con visualizaciones propias	Múltiples instancias (5.0 y 7.0), MySQL/PostgreSQL/TimescaleDB	Servidor central único, sin proxies
Alcance de monitoreo	Red completa del operador	Redes de internet y telefonía	Estaciones base, servicios móviles e ISP	Routers Nokia, enlaces de transporte, energía, seguridad física
Notificaciones	No especificado en la fuente consultada	No especificado en la fuente consultada	Dashboards Zabbix	Telegram segmentado por región
MTTD/MTTA reportado	No especificado cuantitativamente	No especificado cuantitativamente	No especificado cuantitativamente	182 segundos (medido)

Fuente. Elaboración propia a partir de initMAX (2025a), Osmanagić (2025) y Zabbix (2025).

Los tres casos de referencia confirman que Zabbix escala sin dificultad a redes de telecomunicaciones considerablemente mayores a la de YOFC PERÚ S.A.C., lo que respalda la elección de la plataforma y su margen de crecimiento futuro. La diferencia más relevante frente a los tres operadores es arquitectónica: Hrvatski Telekom opera con 30 proxies regionales y LMT con múltiples instancias distribuidas, mientras que el presente caso funciona sobre un servidor único sin proxies regionales, limitación ya reconocida en la sección 4.1.8 y abordada como línea de evolución en la recomendación 6.10. Resulta significativo que, incluso en implementaciones de la escala de Hrvatski Telekom, la literatura consultada no reporte una cifra explícita de MTTD; esto sugiere que la cuantificación del tiempo de detección 182 segundos, sobre una muestra real de 30 eventos es uno de los elementos que distingue este trabajo frente a los casos revisados, sin que ello descarte que otros operadores simplemente no hayan publicado ese dato.

4.1.7 Modelo de implementación de monitoreo centralizado

Como consecuencia de la experiencia acumulada a lo largo de las etapas descritas en el presente capítulo, es posible sistematizar el proceso seguido en un modelo conceptual de seis fases (ver Tabla 30), aplicable como referencia para futuras implementaciones de monitoreo centralizado en organizaciones con infraestructura de telecomunicaciones distribuida geográficamente. Este modelo no pretende ser una metodología rígida, sino una síntesis del recorrido real del proyecto, útil tanto para replicar la experiencia como para anticipar los puntos críticos que suelen surgir al escalar este tipo de soluciones.

Tabla 30 **Fases del modelo de implementación de monitoreo centralizado propuesto**

Fase	Objetivo
1. Diagnóstico	Evaluar la fragmentación de herramientas de supervisión existentes y la dependencia de procesos manuales de

	detección.
2. Diseño	Definir una arquitectura jerárquica de monitoreo basada en segmentación regional y funcional, alineada con la topología real de la infraestructura.
3. Implementación base	Desplegar el monitoreo ICMP como primera capa de disponibilidad, junto con el esquema inicial de alertamiento.
4. Escalamiento	Incorporar monitoreo SNMPv3 mediante Low-Level Discovery, automatizando la asignación de configuración a través de la API del sistema.
5. Optimización	Implementar mecanismos de compresión de series temporales (TimescaleDB) y monitoreo proactivo del propio servidor de supervisión.
6. Validación	Medir el impacto sobre indicadores como el MTTD y la disponibilidad del servicio, documentando las lecciones aprendidas del proceso.

Fuente. Elaboración propia.

Cabe precisar que estas fases no se ejecutaron de forma estrictamente lineal. En la práctica, el escalamiento hacia SNMPv3 y la optimización mediante TimescaleDB ocurrieron de manera casi simultánea, forzados por el crecimiento acelerado del número de hosts monitoreados una vez superada la etapa de validación inicial con ICMP. Esta característica es consistente con el enfoque de investigación-acción adoptado en el desarrollo del proyecto, donde el diagnóstico y el ajuste de la solución se dieron de forma iterativa y no como una secuencia cerrada de pasos.

El valor de este modelo radica en que condensa, en una estructura replicable, los aprendizajes obtenidos frente a problemas que normalmente no se anticipan en la fase de planificación, como el agotamiento de memoria caché o la saturación de disco

descritos en la sección 3.6.2, y que suelen manifestarse recién cuando el sistema opera bajo carga real. Esta síntesis se retoma en las conclusiones del presente trabajo.

4.1.8 Limitaciones del estudio

El modelo presentado en la sección anterior sintetiza el aporte metodológico del presente trabajo; corresponde ahora reconocer las condiciones que acotan su alcance.

Todo trabajo de implementación tecnológica desarrollado en un entorno productivo real opera bajo restricciones que condicionan el alcance de sus resultados. Reconocer estas limitaciones de forma explícita delimita con precisión el contexto en que son válidos y señala las condiciones necesarias para su replicación o extensión. Las limitaciones identificadas en el presente trabajo se agrupan en cuatro categorías.

a) Alcance de la validación: entorno de producción único

La implementación fue desplegada y validada sobre la infraestructura de una sola organización y dentro de ella sobre un conjunto de equipos de un fabricante específico (Nokia 7250). Los resultados obtenidos incluyendo la lógica de filtrado de interfaces mediante el prefijo to_, los parámetros de configuración SNMPv3 y las tasas de compresión de TimescaleDB son representativos de ese entorno particular. Su aplicabilidad directa a redes con equipos de otros fabricantes, versiones de firmware distintas o topologías de diferente naturaleza requeriría una fase de adaptación previa no contemplada en este trabajo. Esta limitación responde al carácter aplicado del presente trabajo, cuyo objetivo es documentar una solución real implementada en condiciones concretas, no desarrollar un marco genérico de aplicación universal.

b) Ausencia de línea base cuantitativa previa

La evaluación del impacto en la capacidad de detección de incidentes se sustenta en pruebas funcionales y en la comparación cualitativa entre el modelo previo y el implementado. No fue posible establecer una medición cuantitativa formal del tiempo promedio de detección (MTTD) antes de la implementación, dado que el modelo anterior no contaba con registros históricos estructurados ni mecanismos de

timestamping automático de eventos. Esta ausencia impide realizar una comparación estadística rigurosa entre ambos escenarios y restringe el análisis de impacto al plano descriptivo y funcional. La limitación es inherente a entornos donde la herramienta de monitoreo preexistente no generaba trazabilidad automatizada de incidentes, situación frecuente en organizaciones en proceso de maduración de sus capacidades operativas *(Axelos, 2019)*. No obstante, sí se dispone de un registro real de eventos generados por el sistema implementado durante el periodo de operación (Tabla 20 y Tabla 27), el cual permite verificar empíricamente los tiempos de detección y notificación de Zabbix, aun cuando no sea posible establecer una comparación cuantitativa directa frente al modelo previo por la razón ya expuesta.

c) Cobertura parcial de métricas de desempeño

El monitoreo implementado cubre disponibilidad de equipos (ICMP) y estado de enlaces de transporte (SNMPv3 sobre IF-MIB). No se incorporaron en la fase actual métricas de desempeño de dispositivo, tales como utilización de CPU, consumo de memoria, estadísticas de tráfico por interfaz o errores de transmisión. Estas métricas, relevantes para una gestión integral bajo el modelo FCAPS *(Stallings, 2018),* fueron identificadas como extensión natural del sistema en la sección de recomendaciones. Su exclusión responde a una decisión de priorización técnica: en el contexto operativo del NOC de YOFC PERÚ S.A.C., la necesidad más urgente era contar con detección automatizada de indisponibilidad y estado de enlaces, dejando el monitoreo de desempeño detallado como fase evolutiva del sistema.

d) Proyecciones de almacenamiento basadas en periodo de observación limitado

Las proyecciones de crecimiento de almacenamiento a 12 meses y las tasas de compresión de TimescaleDB reportadas se calcularon sobre el historial acumulado durante el periodo de implementación, que no cubre un año completo de operación. Factores como la variación estacional en la carga de la red, la incorporación de nuevos grupos de hosts o cambios en la frecuencia de polling podrían afectar los valores proyectados. Las cifras presentadas constituyen estimaciones orientativas fundamentadas en las tasas observadas y en la literatura especializada *(Timescale Inc., 2024a; initMAX, 2025b)*, y no deben interpretarse como garantías de comportamiento futuro sin un seguimiento periódico del sistema.

e) Arquitectura de servidor único sin redundancia

La solución implementada opera sobre un único servidor Zabbix sin configuración de alta disponibilidad. Esta condición implica que cualquier interrupción del servidor de monitoreo ya sea por mantenimiento programado, falla de hardware o actualización de software detiene temporalmente la capacidad de supervisión de toda la red, incluyendo la generación de alertas hacia Telegram. En términos prácticos, el sistema de monitoreo se convierte en un punto único de falla para la visibilidad operativa del NOC.

Esta limitación fue una decisión consciente de alcance, no una omisión técnica. Durante el periodo de implementación, la prioridad operativa fue establecer una plataforma funcional, centralizada y validada sobre la infraestructura existente de la organización, aprovechando el servidor virtual corporativo disponible sin requerir inversión adicional en hardware. La configuración de alta disponibilidad mediante esquemas activo-pasivo con replicación de base de datos PostgreSQL representa una fase evolutiva del sistema cuya viabilidad depende de la disponibilidad de infraestructura adicional por parte de la organización, y se encuentra detallada como recomendación de alta prioridad en la sección 6.2 del presente trabajo. Desde la perspectiva de la gestión de servicios TI, *Axelos (2019)* señala que la implementación incremental de capacidades en entornos productivos, priorizando la funcionalidad inicial sobre la redundancia, es una práctica aceptable cuando los recursos disponibles son limitados, siempre que los riesgos asociados sean identificados y gestionados de forma explícita, como se hace en el presente apartado.

4.2 Evaluación Económica

La evaluación económica de la solución implementada se aborda bajo un enfoque de costo-beneficio, considerando tanto los costos evitados por el uso de software open source como los beneficios operativos derivados de la mejora en los tiempos de detección de incidentes. Este enfoque permite analizar el impacto económico de la solución en términos de inversión directa y en la optimización de la disponibilidad del

servicio, con los consecuentes beneficios asociados.

Aunque la solución implementada no implicó costos de licenciamiento (Zabbix, Ubuntu y PostgreSQL son open source), sí debe analizarse el impacto económico desde una perspectiva más amplia. Dicha comparación de costos se presenta en la Tabla 31.

Tabla 31 **Comparación de costos estimados**

Concepto	Solución Open Source	Solución Propietaria (estimada)
Licenciamiento	$0	Miles de USD anuales
Infraestructura	Reutilizada	Similar
Soporte	Comunidad	Contrato adicional
Escalabilidad	Sin costo adicional	Licencias por nodo

Fuente. Elaboración propia.

En cuanto a los costos de implementación, estos se concentraron principalmente en recursos internos de la organización, incluyendo horas-hombre destinadas a la instalación, configuración y pruebas del sistema, así como el uso de infraestructura existente. La naturaleza open source de Zabbix permitió evitar costos de licenciamiento, lo que representa una ventaja significativa frente a soluciones propietarias. Se estima que el esfuerzo de implementación comprendió actividades técnicas equivalentes a un proyecto de mediana complejidad, sin requerir inversión adicional en hardware ni contratación de licencias.

Discusión Económica

El uso de software open source permitió eliminar costos recurrentes de licenciamiento. Además, la reducción del tiempo de detección puede traducirse en:

- Menor duración de interrupciones.
- Mayor disponibilidad demostrable.
- Mejor imagen institucional.
- Mayor competitividad en contratos.

Aunque no se realizó un análisis financiero cuantitativo exhaustivo, el impacto cualitativo es significativo.

A mediano plazo, las soluciones propietarias de monitoreo suelen implicar costos recurrentes de licenciamiento que se incrementan en función del número de dispositivos monitoreados. En contraste, la implementación de Zabbix permite proyectar un ahorro acumulado significativo en un horizonte de 3 a 5 años, especialmente en entornos con crecimiento progresivo de la infraestructura.

Adicionalmente, la reducción del tiempo medio de detección (MTTD) tiene un impacto económico indirecto relevante. Al disminuir el tiempo de identificación de incidentes, se reduce la duración de las interrupciones del servicio, lo cual contribuye al cumplimiento de los acuerdos de nivel de servicio (SLA) y evita posibles penalizaciones contractuales. También mejora la calidad percibida del servicio y fortalece la competitividad de la organización. La estimación de recursos utilizados en la implementación se detalla en la Tabla 32.

Tabla 32 **Estimación de recursos utilizados en la implementación**

Concepto	Estimación
Duración aproximada del proyecto	4 meses
Horas-hombre (Implementación ICMP)	Aproximadamente 80 horas
Horas-hombre (Implementación SNMP)	Aproximadamente 120 horas
Personal involucrado	2 personas

Concepto	Estimación
Capacitación al personal NOC	2 horas
Infraestructura utilizada	Servidor virtual corporativo
Sistema operativo	Ubuntu 22.04 LTS
Base de datos	PostgreSQL
Costo de licenciamiento	0 USD (software open source)

Fuente. Elaboración propia.

La implementación del sistema aprovechó infraestructura tecnológica existente de la organización, evitando inversiones adicionales en hardware dedicado. El uso de herramientas open source como Zabbix, Ubuntu y PostgreSQL permitió eliminar costos asociados a licenciamiento de software propietario.

En términos operativos, el proyecto requirió en la fase inicial aproximadamente 80 horas-hombre y aproximadamente 120 horas-hombre adicionales, dedicadas al diseño del mecanismo de descubrimiento automático, la automatización de la asignación masiva mediante scripts en Python sobre la API de Zabbix, y la resolución de tres incidentes críticos de infraestructura (agotamiento de memoria caché, saturación de disco y fallo silencioso de los jobs de compresión de TimescaleDB). El costo total estimado del proyecto, considerando ambas fases (200 horas-hombre), asciende a S/ 6,000.00 (≈USD 1,620), muy por debajo del costo de licenciamiento anual de una solución propietaria equivalente. Todas estas fases fueron distribuidas durante un periodo de cuatro meses, considerando actividades de instalación, configuración, pruebas funcionales, integración de alertamiento y capacitación básica al personal del NOC. La capacitación tuvo una duración aproximada de dos horas, debido a que el sistema implementado presenta una interfaz intuitiva y el personal ya contaba con experiencia previa en plataformas similares de monitoreo

Retorno de la Inversión (ROI)

Dado que la solución implementada se basa en herramientas open source (Zabbix, Ubuntu, PostgreSQL) sin costos de licenciamiento, y que la implementación fue ejecutada íntegramente por el autor como parte de sus funciones dentro del NOC, el único componente de inversión cuantificable corresponde a las horas-hombre invertidas (200 horas, según Tabla 32). Por políticas de confidencialidad de la empresa, no es posible utilizar la tarifa horaria real del personal; por ello, para fines exclusivamente ilustrativos del cálculo, se asume una tarifa horaria interna referencial de S/ 30.00 por hora, equivalente aproximadamente a USD 8.00.

Bajo este supuesto, la inversión total asciende a S/ 6,000.00 (aproximadamente USD 1620.00).

Como beneficio cuantificable, se considera el costo evitado de licenciamiento de una solución comercial equivalente (ver "Costos Evitados por Licenciamiento" más adelante), estimado en un rango de USD 19,704 a USD 143,328 anuales para una infraestructura de escala similar.

Tomando el valor menor del rango (USD 19,704 anuales) ya que PRTG comparte características similares a Zabbix, cubre las necesidades actuales y considerando que dicho tier cubre 10,000 sensores (aproximadamente 1,000 dispositivos, cifra cercana a la escala de la infraestructura analizada) tenemos como beneficio del primer año:

$$ROI = ((Beneficio - Costo) / Costo) \times 100$$

$$ROI = ((19{,}704 - 1{,}620) / 1{,}620) \times 100 \approx 1116\%$$

Periodo de recuperación de la inversión (payback) = Costo / Beneficio mensual

$$Payback \approx 1620 / (19{,}704/12) \approx 0.987 \text{ meses (aproximadamente 1 mes)}$$

Estos resultados se resumen en la Tabla 33.

Tabla 33 **Estimación de Retorno de la Inversión (ROI) dentro del primer año**

Concepto	Valor estimado
Costo de implementación (200 h-hombre × S/30)	S/ 6,000.00 (≈USD 1620)
Beneficio anual estimado (licenciamiento evitado)	USD 19,704
ROI estimado	1116%
Periodo de recuperación (payback)	≈ 1 mes

Fuente. Elaboración propia.

Cabe precisar que la licencia PRTG 10000 utilizada como referencia cubre hasta 10,000 sensores, equivalentes aproximadamente a 1,000 dispositivos monitoreados con una carga típica de 5 a 10 sensores por equipo. Para los 1,493 hosts de la infraestructura de la empresa, esta licencia resultaría insuficiente y requeriría una edición de mayor capacidad (PRTG Enterprise, con precio bajo cotización). La cifra empleada en el cálculo de ROI se mantiene como referencia conservadora para fines comparativos, no como cotización exacta para el volumen real de dispositivos.

Costos Evitados por Licenciamiento

La adopción de Zabbix como plataforma de monitoreo eliminó la necesidad de licenciamiento recurrente propio de soluciones comerciales como PRTG Network Monitor o SolarWinds NPM. Según el precio de lista oficial publicado por *Paessler AG (2026)* para la licencia PRTG 10000 (para más de 1000 dispositivos) el costo de licenciamiento anual asciende a USD 19,704. Para SolarWinds Network Performance Monitor, dado que el fabricante no publica precios de lista oficiales y opera bajo modalidad de cotización personalizada, se toma como referencia el valor reportado en su página web, estimado en USD 8 x mes x host, haciendo un total estimado anual de USD 143,328 *(SolarWinds Worldwide, LLC., 2026)*.

Proyectando este costo evitado a un horizonte de 3 a 5 años, periodo de vida útil razonable para una plataforma de monitoreo antes de requerir actualización mayor, el ahorro acumulado estimado, costos evitados, se presenta en la Tabla 34.

Tabla 34 **Proyección de costos evitados por licenciamiento (horizonte 3–5 años)**

Horizonte	PRTG Costo evitado	SolarWinds Costo evitado
1 año	USD 19,704	USD 143,328
3 años	USD 59,112	USD 429,984
5 años	USD 98,520	USD 716,640

Fuente. Elaboración propia, estimación basada en rangos referenciales de mercado.

Estos valores no consideran el costo adicional que las soluciones propietarias suelen aplicar por escalamiento (licenciamiento por sensor o por nodo adicional), por lo que el ahorro real podría ser mayor conforme crezca el número de dispositivos monitoreados.

Ahorro de Horas-Hombre

El ahorro en horas-hombre se evidencia en dos frentes operativos:

a) **Carga masiva de hosts mediante JSON:** La incorporación manual de 100 hosts requería entre 300 y 500 minutos, frente a 10–15 minutos mediante importación JSON. Esto representa un ahorro estimado de entre 4.8 y 8.1 horas-hombre por cada 100 dispositivos incorporados al sistema. Es decir, en entornos de monitoreo a gran escala, la automatización de la configuración (frente a métodos manuales) es un habilitador crítico para la viabilidad del proyecto, reduciendo el tiempo de implementación en más del 95%.

b) **Consolidación de la supervisión operativa:** Previo a la implementación, el operador del NOC debía revisar de forma independiente las 8 plataformas de monitoreo (4 regiones × 2 redes), además de los gestores de energía y seguridad física. Asumiendo, de manera referencial, un tiempo promedio de revisión de 5 minutos por plataforma y una frecuencia de 6 revisiones por turno de 12 horas, el tiempo diario dedicado a supervisión manual ascendía aproximadamente a 4 horas por operador. Con la consolidación en un dashboard único, este tiempo se redujo a aproximadamente 1 hora diaria, lo que representa un ahorro estimado de 3 horas-hombre por operador por turno, equivalente a un ahorro mensual cercano a 90 horas-hombre por operador.

Estas cifras corresponden a estimaciones ilustrativas basadas en supuestos operativos razonables, y no en mediciones cronometradas formales, dado que no existía un registro histórico previo a la implementación (limitación ya señalada en la sección 3.6.6).

La Tabla 35 sintetiza los dos ahorros estimados descritos anteriormente, correspondientes a la carga masiva de hosts y a la consolidación de la supervisión operativa diaria del NOC.

Tabla 35 **Estimación de ahorro de horas-hombre**

Concepto	Ahorro estimado (horas/hombre)
Carga masiva de hosts	4.8 – 8.1
Supervisión diaria del operador	3 x turno

Fuente. Elaboración propia, basada en supuestos operativos declarados

Comparación Económica frente a Soluciones Comerciales

Con el fin de dimensionar comparativamente el impacto económico de la solución implementada frente a alternativas comerciales equivalentes, la Tabla 36 contrasta los principales criterios de licenciamiento, costo de implementación, escalabilidad,

soporte técnico y tiempo de despliegue entre Zabbix, PRTG y SolarWinds, considerando precios de lista vigentes para infraestructuras de escala similar a la analizada (1,493 hosts).

Tabla 36 **Comparación económica entre la solución implementada y alternativas comerciales**

Criterio	Zabbix	PRTG	SolarWinds
Licenciamiento anual	USD 0	USD 19,704 (suscripción anual; USD 1642/mes según plan, ~ 1000 dispositivos)	USD 143,328 (licencia USD8 hosts x mes)
Cobertura	1,493 hosts	~1,000 dispositivos (10,000 sensores)	1,493 hosts
Costo de implementación	200 h-hombre internas (≈USD 1620)	Variable; usualmente incluye consultoría externa	Variable; usualmente incluye consultoría externa
Costo por escalamiento	Sin costo adicional por dispositivo/sensor	Costo adicional por sensor o múltiples instancias	Costo adicional por nodo
Soporte	Comunidad open source + documentación oficial	Soporte comercial incluido	Soporte comercial incluido
Tiempo de	4 meses	Referencial: 1–2 meses (mayor velocidad por	Referencial: 1–2

implementación	soporte comercial dedicado)	meses

Fuente. Elaboración propia.

La comparación evidencia que, si bien las soluciones propietarias pueden ofrecer tiempos de implementación más cortos gracias a soporte comercial dedicado, el costo recurrente de licenciamiento y escalamiento representa una desventaja significativa frente al modelo open source adoptado, especialmente en organizaciones con crecimiento progresivo del número de dispositivos monitoreados, como es el caso de YOFC PERÚ S.A.C.

Este cálculo corresponde a una estimación referencial basada en supuestos declarados, y no en cifras contables auditadas, dado que la implementación no generó desembolsos directos de capital ni contratación de terceros.

La solución implementada representa una alternativa económicamente sostenible para entornos de telecomunicaciones distribuidos geográficamente, especialmente en organizaciones que requieren plataformas escalables sin incurrir en altos costos de licenciamiento.

La implementación demandó aproximadamente 200 horas-hombre distribuidas entre actividades de instalación, configuración, validación y pruebas operativas durante un periodo de cuatro meses

CAPÍTULO V:
CONCLUSIONES

5.1 Conclusión General

El Network Operations Center de YOFC PERÚ S.A.C. supervisaba, antes de este trabajo, una infraestructura de telecomunicaciones distribuida en cuatro regiones mediante ocho plataformas independientes y procesos manuales de filtrado de alarmas. Esa fragmentación no era un simple inconveniente operativo: retrasaba la detección de fallas, dependía por completo de la constancia del operador de turno y dejaba sin registro sistemático el tiempo real que tomaba identificar un incidente. El presente Trabajo de Suficiencia Profesional abordó esa problemática mediante la implementación de un sistema de monitoreo centralizado basado en Zabbix, que consolidó routers, gestores auxiliares, sistemas de energía y dispositivos de seguridad física en una única plataforma de supervisión.

El resultado se sostiene en evidencia concreta y no solo en la percepción de mejora. Sobre una muestra de 30 eventos reales de indisponibilidad extraídos de la base de datos de producción, el sistema implementado detecta una falla en un promedio de 182 segundos y notifica ese hallazgo a Telegram en 3.1 segundos adicionales en promedio, con un máximo de 4 segundos. Frente a un modelo previo sin registros estructurados de tiempos de detección, esta capacidad de respuesta casi inmediata transforma de raíz la forma en que el NOC identifica y comunica sus incidentes.

Ese resultado es significativo porque no depende de una única mejora aislada, sino de la combinación de varios factores trabajando en conjunto: la centralización de la información redujo la fragmentación, pero fue la automatización del alertamiento la que efectivamente acortó los tiempos de detección; la segmentación regional evitó que esa automatización se tradujera en saturación de mensajes; y la ampliación posterior

hacia monitoreo SNMP de enlaces añadió una capacidad de diagnóstico que el monitoreo ICMP por sí solo no podía ofrecer. En conjunto, la actividad desarrollada demuestra la aplicación práctica de principios de gestión de redes, automatización de procesos y diseño estructural de monitoreo, alineados con el modelo FCAPS y con los lineamientos de gestión de incidentes de ITIL. Estos cuatro factores: centralización, automatización del alertamiento, segmentación regional y ampliación progresiva del monitoreo constituyen, en conjunto, la respuesta al problema general planteado en la sección 3.3.1 sobre los factores críticos que determinan el éxito de la transición de un modelo de monitoreo reactivo hacia uno proactivo en entornos de telecomunicaciones distribuidos geográficamente.

5.2 Cumplimiento del Objetivo General

El objetivo general planteado consistió en implementar un sistema de monitoreo integral basado en Zabbix que permitiera mejorar la capacidad operativa del NOC en la detección temprana de incidentes.

Evidencias obtenidas:

- Fortalecimiento de la capacidad de detección de incidentes, mediante la transición desde un modelo de revisión manual hacia alertamiento automatizado con tiempos de notificación de pocos segundos confirmados mediante el análisis de una muestra de 30 eventos reales registrados en producción.
- Automatización del alertamiento mediante integración con Telegram.
- Centralización de información previamente fragmentada.
- Segmentación regional estructurada.
- Visualización estratégica mediante dashboards jerárquicos.

La implementación permitió que los eventos críticos sean detectados y notificados en tiempo real, eliminando la dependencia exclusiva de revisión manual y disminuyendo

el riesgo de omisión de incidentes.

Estos hallazgos son consistentes con las proposiciones teóricas orientadoras formuladas a partir de FCAPS, ITIL y los principios de diseño de arquitecturas de monitoreo distribuido (Stallings, 2018; Axelos, 2019; Oppenheimer, 2011), evaluadas con mayor detalle en la sección 4.1.1.1.

Aportes técnicos específicos: Además de cumplir el objetivo general, la implementación dejó cuatro contribuciones concretas que merece la pena destacar de forma explícita. La primera es el modelo de implementación de seis fases. La segunda es el script en Python desarrollado sobre la API REST de Zabbix (sección 3.5.7.6), que automatiza la creación de interfaces SNMPv3 y la asignación masiva de templates, resolviendo una limitación que la interfaz web de Zabbix 7.2 no cubre de forma nativa. La tercera es la optimización de la base de datos mediante TimescaleDB (sección 3.5.7.8), que logró una reducción cercana al 95% en el espacio de almacenamiento del historial, validando la viabilidad de extender PostgreSQL con herramientas open source para volúmenes de datos de monitoreo a gran escala. La cuarta es la diferenciación de alertas por tipo de evento en el script de Telegram (sección 3.5.7.7), que distingue entre disponibilidad de host y estado de enlace tanto en el mensaje de alerta como en el de recuperación, reduciendo la ambigüedad que antes obligaba al operador a interpretar cada notificación caso por caso.

5.3 Implicaciones del caso

- **Modelo FCAPS:** La implementación muestra que la gestión de fallas (Fault Management) y la gestión de desempeño (Performance Management) no pueden implementarse de manera aislada; la escalabilidad del sistema depende de la optimización de la base de datos (una consideración no contemplada en el modelo FCAPS estándar).
- **ITIL:** Se evidencia que la implementación de ITIL en entornos con recursos limitados puede requerir un enfoque incremental (ICMP → SNMPv3), en lugar de una implementación completa desde el inicio.

- **Adopción de open source en telecomunicaciones:** Se demuestra que el ahorro en licencias puede compensar la inversión en horas-hombre y que la comunidad open source (Zabbix, TimescaleDB) proporciona un soporte suficiente para entornos críticos, siempre que se cuente con personal capacitado.

5.4 Impacto en la Gestión de Incidentes

La arquitectura implementada fortaleció la fase de identificación dentro del ciclo de gestión de incidentes, al garantizar que cada evento crítico active automáticamente una notificación contextualizada dirigida al equipo correspondiente.

La integración de alertas segmentadas por región permitió reducir la saturación informativa y mejorar la eficiencia operativa, mientras que los mensajes de recuperación automática completaron el ciclo de gestión, proporcionando trazabilidad y cierre formal de cada incidente.

Desde el enfoque de desempeño operacional, la reducción del MTTD es uno de los logros más relevantes del proyecto, ya que la rapidez en la detección es un factor determinante para minimizar el impacto de las interrupciones en la disponibilidad del servicio.

5.5 Centralización y Mejora en la Visibilidad Operativa

La consolidación de múltiples plataformas en una única herramienta de monitoreo representó una mejora estructural significativa en la gestión del NOC. La arquitectura jerárquica implementada, basada en segmentación regional y funcional, permitió una representación coherente de la infraestructura física dentro del entorno lógico de supervisión.

La incorporación de dashboards corporativos y regionales fortaleció la capacidad de

supervisión estratégica y operativa, permitiendo identificar rápidamente patrones de comportamiento, incidentes críticos y estados de disponibilidad. Esta visualización estructurada reduce la carga cognitiva del operador y facilita la toma de decisiones en tiempo real.

5.6 Evaluación Técnica de la Solución Implementada

Desde el aspecto tecnológico, la solución implementada demostró un adecuado nivel de estabilidad, funcionalidad y cumplimiento de los objetivos planteados, al establecer un sistema de monitoreo centralizado, automatizado y alineado con las necesidades operativas del NOC.

La elección de Zabbix 7.2 sobre Ubuntu 22.04 con PostgreSQL resultó apropiada en términos de compatibilidad, rendimiento y escalabilidad, permitiendo desplegar una arquitectura robusta sin incurrir en costos de licenciamiento, lo cual favorece su sostenibilidad a largo plazo.

De forma paralela, la configuración de triggers ICMP con parámetros evaluados técnicamente permitió establecer una primera capa confiable de supervisión de disponibilidad, mientras que la implementación de acciones y media types consolidó un flujo eficiente de distribución de alertas, garantizando la notificación oportuna de incidentes.

Es importante precisar que la implementación se enfocó en resolver prioritariamente la problemática de fragmentación del monitoreo y la detección tardía de incidentes, estableciendo una base funcional sólida para la gestión centralizada del NOC. La incorporación de mecanismos avanzados como alta disponibilidad del servidor, monitoreo SNMP de desempeño de dispositivo (CPU, memoria, errores de interfaz), integración con sistemas OSS y correlación de eventos fue considerada como parte de una fase evolutiva del sistema, habiéndose ya alcanzado un primer nivel de madurez en el monitoreo SNMP de enlaces (sección 3.5.7) orientada a su maduración progresiva.

Este enfoque por etapas es consistente con prácticas de ingeniería en entornos de telecomunicaciones, donde las soluciones son implementadas de manera incremental para garantizar estabilidad operativa, control del sistema y validación progresiva de funcionalidades antes de incorporar niveles adicionales de complejidad.

5.7 Impacto Económico y Estratégico

La solución implementada generó un impacto económico positivo al eliminar costos asociados a licencias propietarias y optimizar el tiempo operativo del personal del NOC.

La reducción en el tiempo de detección contribuye indirectamente a disminuir la duración de interrupciones y mejorar el cumplimiento de acuerdos de nivel de servicio (SLA), fortaleciendo la posición competitiva de la empresa en proyectos de conectividad.

Estratégicamente, la implementación consolida la capacidad de la organización para demostrar control y trazabilidad sobre su infraestructura crítica, elemento fundamental en entornos donde la confiabilidad del servicio es determinante.

5.8 Aporte Profesional y Académico

El desarrollo del presente Trabajo de Suficiencia Profesional evidencia la aplicación práctica de competencias propias de la Ingeniería Electrónica y de Telecomunicaciones, tales como:

- Diseño de arquitecturas de supervisión.
- Integración de plataformas tecnológicas.
- Automatización de procesos operativos.
- Gestión de redes IP.

- Evaluación de desempeño operativo.
- Diagnóstico y resolución de incidentes de infraestructura bajo escenarios de escalamiento.

La actividad desarrollada y la instalación de una herramienta implicó un proceso de análisis, diseño, implementación y validación estructurada, con impacto directo en la operación empresarial.

En consecuencia, el proyecto evidencia la capacidad profesional para diseñar e implementar soluciones tecnológicas orientadas a la mejora de procesos críticos en entornos de telecomunicaciones.

5.9 Alineamiento con los Objetivos de Desarrollo Sostenible (ODS)

La implementación desarrollada en este trabajo no tuvo como propósito medir impacto social, pero sí puede leerse en función de él. El fortalecimiento de la capacidad de detección temprana de incidentes y la reducción del tiempo de indisponibilidad de la red apuntan, de manera directa, al Objetivo de Desarrollo Sostenible 9, en tanto se consolida una plataforma de monitoreo escalable construida sobre herramientas de código abierto, capaz de sostener infraestructura crítica de telecomunicaciones sin las barreras de costo que suelen limitar este tipo de despliegues.

Hay además una segunda lectura, quizás menos evidente pero igual de relevante. La red que se supervisa forma parte del Proyecto Regional de Banda Ancha orientado a cerrar la brecha digital en zonas rurales del Perú, descrito en el Capítulo I. Cada mejora en la disponibilidad del servicio termina beneficiando, en última instancia a las poblaciones que dependen de esa conectividad, entre ellas instituciones educativas y de salud pública que ven reducido el riesgo de quedarse sin comunicación durante periodos prolongados. En ese sentido, el trabajo se vincula también con el Objetivo 10, referido a la reducción de las desigualdades.

Vale precisar que estos vínculos no se plantean como resultados medidos ni como objetivos formales del proyecto, sino como una condición habilitante: una red más

disponible y mejor monitoreada no garantiza por sí sola el desarrollo de las comunidades que conecta, pero sí es un requisito necesario para que ese desarrollo pueda sostenerse en el tiempo.

5.10 Aporte del Modelo de Implementación Propuesto

Como resultado adicional del presente trabajo, se sistematizó la experiencia de implementación en un modelo de seis fases (diagnóstico, diseño, implementación base, escalamiento, optimización y validación), descrito en la sección 4.1.7. Este modelo constituye un aporte metodológico que trasciende el caso particular de YOFC PERÚ S.A.C., pudiendo servir de referencia para organizaciones con infraestructura de telecomunicaciones distribuida que busquen transitar de un esquema de supervisión fragmentado hacia uno centralizado y automatizado.

CAPÍTULO VI:
RECOMENDACIONES

Las recomendaciones formuladas se organizan en función de su nivel de prioridad, considerando su impacto en la operación del NOC y su viabilidad de implementación. Se distinguen recomendaciones de alta prioridad, orientadas a garantizar la continuidad operativa; recomendaciones de optimización, enfocadas en mejorar el desempeño del sistema; y recomendaciones estratégicas, vinculadas a la evolución tecnológica a largo plazo.

6.1 Fortalecimiento del Monitoreo mediante Protocolos Avanzados

(Prioridad media – optimización operativa)

Habiéndose implementado el monitoreo SNMPv3 a enlaces de transporte (sección 3.5.7), se recomienda extender este mismo enfoque de Low-Level Discovery hacia métricas de desempeño a nivel de dispositivo, permitiendo la recolección de información relacionada con:

- Uso de CPU y memoria de routers.
- Utilización de interfaces.
- Estadísticas de errores.
- Tráfico entrante y saliente.
- Estado de módulos internos.

Si bien el monitoreo ICMP implementado cumple adecuadamente la función de detección de indisponibilidad, la incorporación de monitoreo SNMP permitiría evolucionar desde un esquema de supervisión de disponibilidad hacia un modelo de supervisión de desempeño integral, fortaleciendo la gestión bajo el enfoque FCAPS (Fault, Configuration, Accounting, Performance, Security).

Esta ampliación contribuiría a detectar degradaciones antes de que se conviertan en interrupciones, reforzando el carácter preventivo del NOC.

6.2 Implementación de Alta Disponibilidad del Servidor Zabbix (Alta prioridad – continuidad operativa crítica)

Se recomienda diseñar e implementar una arquitectura de alta disponibilidad (HA) para el servidor Zabbix, con el objetivo de eliminar el punto único de falla actualmente existente. Entre las alternativas técnicas se sugieren:

- Configuración de servidor Zabbix en modo activo-pasivo.
- Replicación de base de datos PostgreSQL.
- Implementación de balanceador de carga.
- Virtualización con snapshots automatizados.
- Implementación en infraestructura redundante.

La disponibilidad del sistema de monitoreo es crítica, ya que una falla en la plataforma impediría la detección de incidentes en la red productiva.

Desde una perspectiva de ingeniería, el sistema de supervisión debe contar con niveles de disponibilidad iguales o superiores a la infraestructura que monitorea.

6.3 Implementación de Correlación Avanzada de Eventos (Prioridad media – optimización operativa)

Se recomienda implementar mecanismos de correlación de eventos que permitan identificar incidentes raíz y evitar tormentas de alertas derivadas de una misma causa primaria.

Por ejemplo:

- Si un nodo de transporte cae, los nodos dependientes podrían generar múltiples alertas secundarias.
- Mediante correlación inteligente, el sistema podría priorizar la alerta raíz.

Esta mejora reduciría el ruido operacional y optimizaría la priorización de incidentes críticos.

6.4 Optimización de Dashboards mediante Indicadores SLA (Prioridad media – mejora de gestión)

Se recomienda incorporar indicadores formales de cumplimiento de SLA en los dashboards corporativos, incluyendo:

- Disponibilidad mensual por región.
- Tiempo promedio de recuperación (MTTR).
- Tiempo promedio de detección (MTTD).

Estos indicadores permitirían:

- Medición objetiva del desempeño.
- Evaluación continua del NOC.

- Presentación ejecutiva ante gerencia.
- Sustento técnico ante auditorías.

6.5 Capacitación Continua del Personal Técnico (Prioridad media – sostenibilidad operativa)

La sostenibilidad del sistema de monitoreo no depende únicamente de la robustez técnica de la plataforma, sino también del nivel de dominio que el personal del NOC alcance sobre ella. Un plan de capacitación estructurado, con módulos progresivos y ejercicios prácticos, reduce la dependencia de soporte externo y prepara al equipo para responder con criterio propio ante incidentes cada vez más variados (Axelos, 2019).

Objetivo general: Dotar al personal del NOC de las competencias técnicas necesarias para operar, administrar y mantener el sistema de monitoreo Zabbix de manera autónoma.

Objetivos específicos: El plan busca que el personal sea capaz de:

- Comprender la arquitectura del sistema de monitoreo y su relación con la infraestructura de red.
- Interpretar correctamente los dashboards y las alertas generadas por el sistema.
- Realizar diagnósticos iniciales de incidentes con las herramientas ya disponibles.
- Configurar nuevos hosts y ajustar parámetros de monitoreo según las necesidades operativas.
- Generar reportes de disponibilidad orientados a la medición de SLA.
- Aplicar los procedimientos de escalamiento definidos ante incidentes críticos.

Estructura propuesta: La Tabla 37 muestra el plan organizado en seis módulos progresivos, con una carga total estimada de 36 horas.

Tabla 37 **Estructura del Plan de Capacitación**

Modulo	Duración (h)	Contenido	Ejercicio Practico
1.Fundamentos del Sistema de Monitoreo	4	Cubre la arquitectura de Zabbix, sus componentes y el flujo de datos bajo los protocolos ICMP y SNMP	El participante identifica en el sistema real los distintos componentes y cómo se relacionan entre sí
2.Operación del Sistema	6	Aborda la navegación de la interfaz, la interpretación de dashboards y la gestión de hosts y grupos	Consiste en resolver un incidente simulado apoyándose en el dashboard regional
3.Gestión de Alertas	6	Trata la configuración de triggers, acciones y media types, junto con la gestión de periodos de mantenimiento	El participante configura una alerta nueva para un host específico y verifica que funcione
4.Diagnóstico de Incidentes	8	Desarrolla metodologías de troubleshooting y correlación de eventos	El ejercicio pide identificar la causa raíz de un incidente simulado a partir de varias alertas relacionadas
5.Administración	8	Profundiza en la gestión de	Como práctica, el

Avanzada		templates, Low-Level Discovery y administración de usuarios y permisos	participante crea un template para un tipo de dispositivo nuevo y lo aplica a un grupo de hosts
6.Generación de Reportes y SLA	4	Enseña a construir reportes personalizados, calcular disponibilidad e interpretar los indicadores resultantes	El ejercicio final es generar el reporte mensual de disponibilidad de una región concreta
Total	**36**		

Fuente. Elaboración propia.

Cronograma de implementación: Se plantea un despliegue en cinco fases: diseño del programa y materiales durante el primer mes; capacitación inicial cubriendo los Módulos 1 a 3 en el segundo mes; capacitación avanzada con los Módulos 4 a 6 en el tercer mes; una fase de evaluación y retroalimentación en el cuarto mes; y, a partir de ahí, sesiones de refuerzo con frecuencia trimestral.

Impacto esperado: De implementarse, este plan permitiría reducir la dependencia de soporte externo para incidencias y configuraciones rutinarias, mejorar los tiempos de respuesta gracias a una interpretación más rápida de las alertas, aprovechar de forma más completa las funcionalidades avanzadas de Zabbix y sostener el conocimiento técnico incluso ante rotación de personal.

Cabe precisar que esta propuesta amplía, sin sustituirla, la capacitación básica ya impartida durante la implementación (descrita en la sección 4.2), la cual tuvo una duración de dos horas orientada a un manejo funcional inmediato de la plataforma. El plan aquí descrito corresponde a una segunda etapa, de carácter más técnico y

especializado.

6.6 Evaluación Periódica del Desempeño del Sistema (Alta prioridad – control y mejora continua)

Se recomienda realizar auditorías técnicas periódicas del sistema de monitoreo para:

- Validar tiempos de detección.
- Revisar configuración de triggers.
- Ajustar umbrales de alerta.
- Verificar capacidad del servidor.
- Analizar crecimiento de base de datos.

La mejora continua debe formar parte de la cultura operativa del NOC.

6.7 Evolución estratégica del sistema de monitoreo (Prioridad baja – consolidación y crecimiento a largo plazo)

A mediano y largo plazo, se recomienda consolidad el sistema implementado como plataforma oficial de supervisión corporativa, integrando progresivamente todos los componentes críticos de la organización bajo una única arquitectura de monitoreo y evaluar su integración con plataformas OSS (Operational Support Systems) y herramientas de gestión de tickets, de modo que el ciclo de gestión de incidentes quede cerrado no solo en la detección sino en su registro y seguimiento formal.

Esta línea de evolución comprende:

- Generación automática de tickets ante alertas críticas e integración con sistemas ITSM.

- Enlace con herramientas de gestión de cambios y generación automática de reportes SLA.
- Planificación de la expansión de capacidad de almacenamiento y evaluación de una migración de arquitectura distribuida, conforme crezca el número de hosts monitoreados.

Las recomendaciones representan líneas naturales de evolución tecnológica y fortalecimiento estructural del modelo de monitoreo.

El sistema implementado sienta una base sólida, escalable y técnicamente coherente; sin embargo, su madurez dependerá de la capacidad de la organización para adoptar un enfoque de mejora continua, integración progresiva y actualización permanente de sus prácticas operativas.

Desde el aspecto de la ingeniería del sistema, la sostenibilidad tecnológica radica en la implementación y en su evolución estructurada.

6.8 Sostenibilidad técnica y transferencia de conocimiento

Con el fin de garantizar la continuidad operativa del sistema de monitoreo implementado, se han establecido una serie de acciones orientadas a asegurar su sostenibilidad en el tiempo, así como la transferencia de conocimiento al personal encargado de su operación.

En primer lugar, se realizó un proceso de capacitación dirigido al personal del NOC, con una duración aproximada de dos horas apoyado en un manual de operación elaborado específicamente para este sistema (véase Anexo B). Esta capacitación abordó aspectos fundamentales del uso de la plataforma, incluyendo la interpretación de dashboards, gestión de alertas y procedimientos básicos de diagnóstico. La corta duración de la capacitación se justifica por la naturaleza intuitiva del sistema, así como por la experiencia previa del personal con herramientas similares, como Centreon.

En cuanto a la administración del sistema, se ha designado un esquema de

responsabilidad compartida entre el autor del presente trabajo y un miembro del área de tecnologías de la información encargado de la gestión del servidor. Ambos cuentan con credenciales de acceso y la capacidad de intervenir en caso de fallas o requerimientos de mantenimiento.

Respecto a la gestión de actualizaciones, se ha establecido un plan de mantenimiento periódico con frecuencia mensual, el cual permite mantener el sistema actualizado y seguro. Como medida de mitigación de riesgos, se cuenta con una copia de respaldo de la base de datos, lo que permite realizar pruebas de actualización en un entorno controlado antes de su implementación en el sistema principal.

Adicionalmente, las actualizaciones se ejecutan en ventanas de mantenimiento programadas, generalmente durante fines de semana y en horarios de baja carga operativa, con una duración aproximada de una hora, minimizando así el impacto sobre las operaciones del NOC.

Cabe destacar que, debido a políticas internas de la organización, la documentación detallada de procedimientos no puede ser compartida públicamente. No obstante, se garantiza que existen lineamientos internos que respaldan la operación del sistema.

6.9 Ampliación de la Validación Cuantitativa del MTTD (Prioridad media – control y mejora continua)

El análisis presentado en la sección 3.6.6 se apoyó en una muestra de 30 eventos reales de indisponibilidad, suficiente para caracterizar el comportamiento del sistema durante el periodo de implementación, pero todavía acotada frente al volumen de eventos que el servidor acumula en producción. Se recomienda ampliar esta validación bajo los siguientes lineamientos:

- Incrementar el tamaño de la muestra a un mínimo de 100 eventos, cubriendo al menos 30 días continuos de operación, con el fin de reducir la influencia de casos atípicos sobre el promedio y aproximar el análisis a una base estadísticamente más sólida.

- Calcular el MTTD de forma segmentada por región operativa, además del valor agregado, lo que permitiría identificar si existen diferencias de desempeño atribuibles a la latencia propia de cada enlace regional, a la configuración particular de los triggers o a la carga de trabajo del poller en cada franja horaria.
- Ampliar el análisis de distribución iniciado en la sección 4.1.2.1, donde se reportaron los percentiles P25, P50 y P75 sobre la muestra actual de 30 eventos, incorporando los percentiles P90 y P95 sobre la muestra ampliada de al menos 100 eventos. Estos percentiles superiores resultan más sensibles para capturar comportamientos extremos como ventanas de detección inusualmente largas que los cuartiles ya calculados, cuya cobertura se limita al 75% central de la distribución.
- Documentar el procedimiento de extracción bajo el mismo esquema ya empleado para las Tablas 20 y 26 (consulta a la tabla events, muestreo estratificado por día y exclusión justificada de valores atípicos), de manera que el análisis ampliado resulte auditable y reproducible.

La ejecución de esta ampliación requiere acceso directo a la base de datos de producción durante un periodo de observación más extenso del que permitió el cronograma del presente trabajo, por lo que se plantea como línea de continuidad para el equipo del NOC, una vez consolidada la operación del sistema bajo la configuración descrita en el Capítulo III.

6.10 Implementación de Proxies Zabbix en Regiones Remotas (Prioridad media – escalabilidad y continuidad operativa)

La arquitectura conceptual descrita en la sección 3.5.2.7 contempló, entre sus principios rectores, la incorporación de proxies regionales como mecanismo para reforzar la tolerancia a fallos de comunicación. Sin embargo, la implementación efectiva documentada en el Capítulo III recolecta todos los datos directamente desde el servidor central, sin nodos intermedios. Se recomienda evaluar el despliegue de proxies Zabbix en cada una de las cuatro regiones operativas como siguiente paso natural de esa visión arquitectónica.

Contexto y justificación: Actualmente, todo el tráfico de monitoreo, tanto ICMP como SNMP, atraviesa la red WAN hacia el servidor central. Esto implica que una interrupción de conectividad entre una región y el servidor deja sin visibilidad a esa región mientras dure el corte, y que la carga de procesamiento del servidor central crece de forma directa con cada host nuevo que se incorpora, sin importar en qué región se encuentre. Un proxy Zabbix por región permitiría, en cambio, recolectar y almacenar los datos localmente incluso ante una caída temporal del enlace, sincronizando con el servidor central en cuanto la comunicación se restablece.

Beneficios esperados: La incorporación de proxies traería, principalmente, una reducción del tráfico WAN, ya que el proxy agrupa y comprime los datos antes de enviarlos; tolerancia a fallos de red, al conservar localmente la información recolectada durante un corte de conectividad; una menor carga sobre el servidor central, que pasaría a mantener una sola conexión por proxy en lugar de una conexión directa por cada host; una administración de firewalls más simple, al requerir solo una regla de salida por proxy hacia el servidor; y una mejora general en la escalabilidad del sistema, que podría crecer agregando proxies adicionales sin comprometer el rendimiento del nodo central.

Arquitectura propuesta: Se plantea desplegar un proxy en modo activo en cada una de las cuatro regiones (Arequipa, San Martín, La Libertad y Áncash), con base de datos local SQLite para el almacenamiento temporal de los datos recolectados. Cada proxy asumiría la responsabilidad exclusiva de los hosts correspondientes a su propia región, delegando en el servidor central únicamente la consolidación y visualización.

Plan de implementación: La Tabla 38 muestra el despliegue que se organizaría en seis fases:

Tabla 38 **Plan de implementación en 6 fases**

Fase	Actividad	Responsable	Plazo
1	Evaluación inicial de la infraestructura disponible en	Equipo NOC	Semana 1-2

	cada región		
2	Instalación y configuración de proxies en un entorno de pruebas controlado	Administrador Zabbix	Semana 3-4
3	Pruebas de conectividad y rendimiento sobre en el entorno	Equipo NOC	Semana 5-6
4	Despliegue en una región piloto	Administrador Zabbix	Semana 7-8
5	Periodo de monitoreo y ajuste de la configuración piloto	Equipo NOC	Semana 9-10
6	Despliegue progresivo en las tres regiones restantes	Administrador Zabbix	Semana 11-14

Fuente. Elaboración propia.

Consideraciones técnicas: La viabilidad de esta propuesta depende de tres factores puntuales: que cada proxy pueda establecer comunicación TCP con el servidor central por el puerto 10051, que el almacenamiento local de cada proxy se dimensione considerando tanto el volumen de datos de los hosts de su región como el tiempo máximo esperado de interrupción de conectividad, y que el estado de cada proxy quede a su vez monitoreado desde el servidor central, de modo que una falla del propio proxy no pase inadvertida.

Conviene precisar que esta propuesta no sustituye la recomendación de alta disponibilidad del servidor central descrita en la sección 6.2: un esquema de proxies regionales reduce la dependencia del enlace WAN y descarga al servidor de procesamiento, pero no elimina el punto único de falla que representa el propio servidor Zabbix central, riesgo que sigue requiriendo una solución independiente de replicación activo-pasivo.

CAPÍTULO VII:
BIBLIOGRAFÍA

[1]Abanto Taipe, J. D., & Pareja Abarca, G. (2025). *Implementación de una plataforma de monitoreo proactivo para reducir el tiempo de inactividad en el NOC utilizando Zabbix para la empresa Pronet System* [Trabajo de suficiencia profesional, Universidad Peruana de Ciencias Aplicadas]. Repositorio Académico UPC. https://hdl.handle.net/10757/686054

[2]Axelos. (2019). *ITIL® foundation: ITIL 4 edition*. The Stationery Office.

[3]Blumenthal, U., & Wijnen, B. (2002). *User-based security model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)* (RFC 3414). IETF. https://datatracker.ietf.org/doc/html/rfc3414

[4]Camarillo, G., & García-Martín, M. (2014). *IP multimedia subsystem (IMS): Merging the Internet and the cellular worlds* (3rd ed.). Wiley.

[5]Comer, D. E. (2018). *Internetworking with TCP/IP: Principles, protocols, and architecture* (6th ed.). Pearson.

[6]Few, S. (2013). *Information dashboard design: Displaying data for at-a-glance monitoring* (2nd ed.). Analytics Press.

[7]Field, A. (2018). *Discovering statistics using IBM SPSS statistics* (5th ed.). SAGE Publications.

[8]García Zaballos, A., Dalio, M., Garran, J., Iglesias Rodriguez, E., Puig Gabarró, P., & Martínez Garza Fernández, R. (2022). *Estructuración de un centro de operación de redes (NOC)*. https://doi.org/10.18235/0004520

[9]Harrington, D., Presuhn, R., & Wijnen, B. (2002). *An architecture for describing*

Simple Network Management Protocol (SNMP) management frameworks (RFC 3411). IETF. https://datatracker.ietf.org/doc/html/rfc3411

[10]Hiles, A. (2010). *Service level agreements: Measuring cost and quality in service relationships*. Rothstein Associates.

[11]initMAX s.r.o. (2025a). *How we replaced Netcool with Zabbix and our own visualization at O2.* https://www.initmax.com/case-study/how-we-replaced-netcool-with-zabbix-at-o2/

[12]initMAX s.r.o. (2025b). *Implementation of TimescaleDB in Zabbix: Benefits, key tables, and installation*. https://www.initmax.com/wiki/implementation-of-timescaledb-in-zabbix/

[13]International Organization for Standardization. (2013). *ISO/IEC 27001:2013 information technology — Security techniques — Information security management systems — Requirements*. https://www.iso.org/standard/54534.html

[14]International Organization for Standardization. (2018). *ISO/IEC 20000-1:2018 information technology — Service management*. https://www.iso.org/standard/70636.html

[15]International Telecommunication Union. (2012). *Recommendation ITU-T M.3400: TMN management functions*. https://www.itu.int/rec/T-REC-M.3400

[16]McCloghrie, K., & Kastenholz, F. (2000). *The interfaces group MIB (RFC 2863).* IETF. https://datatracker.ietf.org/doc/html/rfc2863

[17]Mohd Faris Mohd Fuzi. (2022). *Integrated network monitoring using Zabbix with push notification via Telegram*. ResearchGate. https://www.researchgate.net/publication/359759373_Integrated_Network_Monitoring_using_Zabbix_with_Push_Notification_via_Telegram

[18]Mongkolluksamee, S., Pongpaibool, P., & Issariyapat, C. (2010). *Strengths and Limitations of Nagios as a Network Monitoring Solution*. ResearchGate.

https://www.researchgate.net/publication/267366269_Strengths_and_Limitations_of_Nagios_as_a_Network_Monitoring_Solution

[19]Narayan, A., Lakshmi, B., Gopal, H., & Vardhan, V. (2023). *AI-Driven Anomaly Detection in Nagios and Zabbix Logs*. ResearchGate. https://www.researchgate.net/publication/393973718_AI-Driven_Anomaly_Detection_in_Nagios_and_Zabbix_Logs

[20]Navidi, W. (2015). *Statistics for engineers and scientists* (4th ed.). McGraw-Hill Education.

[21]Oppenheimer, P. (2011). *Top-down network design* (3rd ed.). Cisco Press. https://www.ciscopress.com/store/top-down-network-design-9781587202834

[22]Organización de las Naciones Unidas. (2015). *Transformar nuestro mundo: la Agenda 2030 para el Desarrollo Sostenible* (A/RES/70/1). https://www.un.org/sustainabledevelopment/es

[23]Osmanagić, A. (2025). *Beyond the noise: Automated RCA & scalable reporting in Zabbix: A Croatian Telecom case study* [Conference presentation]. Zabbix Summit 2025. https://assets.zabbix.com/files/events/2025/zabbix_summit_2025/slides/d2/8_Aldin_Osmanagic.pdf

[24]Paessler AG. (2026). *PRTG Network Monitor pricing*. https://www.paessler.com/pricing

[25]Postel, J. (1981). *Internet control message protocol (RFC 792).* IETF. https://datatracker.ietf.org/doc/html/rfc792

[26]Postel, J., & Harrenstien, K. (1990). *Simple network management protocol (SNMP) (RFC 1157).* IETF. https://datatracker.ietf.org/doc/html/rfc1157

[27]PostgreSQL Global Development Group. (2024). *PostgreSQL documentation*. https://www.postgresql.org/docs/

[28]Rahayu, P. D., Dahlan, A. A., & Vitria, R. (2025). Monitoring real-time server dan

router berbasis Zabbix dengan notifikasi alert Telegram: Studi pra-pasca di SDIT Raudhatul As-Salimy Gobah. *Intellect: Indonesian Journal of Learning and Technological Innovation, 4*(1), 107–116. https://doi.org/10.57255/intellect.v4i1.1372

[29]SolarWinds Worldwide, LLC. (2026). *SolarWinds pricing.* https://www.solarwinds.com/pricing

[30]Stallings, W. (2017). *Network security essentials: Applications and standards* (6th ed.). Pearson. https://www.pearson.com/en-us/subject-catalog/p/network-security-essentials/P200000003482

[31]Stallings, W. (2018). *Foundations of modern networking: SDN, NFV, QoE, IoT, and cloud* (2nd ed.). Pearson. https://www.pearson.com/en-us/subject-catalog/p/foundations-of-modern-networking/P200000003479

[32]Telegram Messenger LLP. (2024). *Telegram Bot API documentation*. https://core.telegram.org/bots/api

[33]Timescale Inc. (2024a). *TimescaleDB documentation — Compression*. https://docs.timescale.com/use-timescale/latest/compression/

[34]Timescale Inc. (2024b). *TimescaleDB documentation — set_integer_now_func()*. https://docs.timescale.com/api/latest/hypertable/set_integer_now_func/

[35]Ubuntu. (2024). *Ubuntu Server 22.04 LTS documentation*. Canonical Ltd. https://ubuntu.com/server/docs

[36]Zabbix LLC. (2023a). *Zabbix 7.2 manual*. https://www.zabbix.com/documentation/7.2/en/manual

[37]Zabbix LLC. (2023b). *Zabbix installation from packages*. https://www.zabbix.com/documentation/7.2/en/manual/installation

[38]Zabbix LLC. (2023c). *Zabbix media types and alerting configuration*. https://www.zabbix.com/documentation/7.2/en/manual/config/notifications

[39]Zabbix LLC. (2023d). *Template module ICMP ping documentation*.

https://www.zabbix.com/documentation/7.2/en/manual/config/templates

[40]Zabbix. (2025, July 23). *Keeping Latvia connected with Zabbix and LMT*. Zabbix Blog. https://blog.zabbix.com/keeping-latvia-connected-with-zabbix-and-lmt/30834/

CAPITULO VIII:
ANEXOS

8.1 ANEXO A: Archivo JSON de carga masiva de hosts

En el marco del TSP, se empleó inteligencia artificial (IA) como herramienta de apoyo para la generación automatizada de archivos de configuración en Zabbix. A continuación, en la Figura A1, se documenta un ejemplo del prompt utilizado, el resultado obtenido y las consideraciones técnicas relevantes.

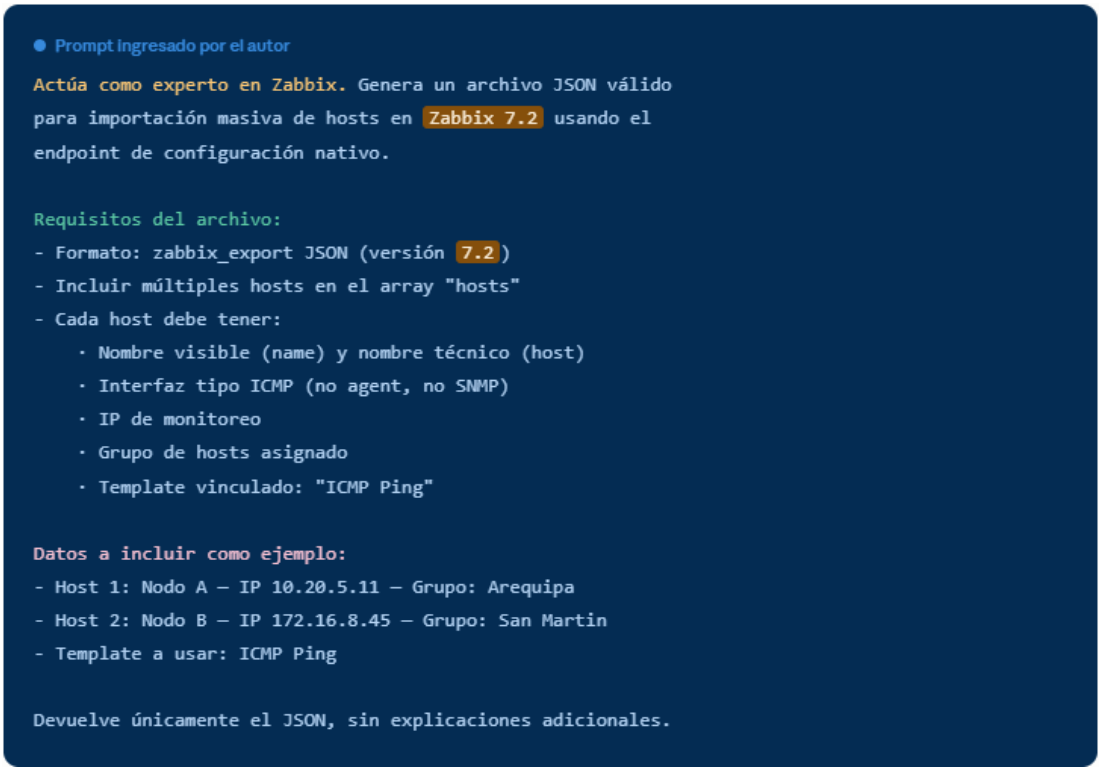


Figura A1 **Prompt para archivo json de importación masiva de hosts.** Fuente. Elaboración propia.

Archivo JSON generado:


```
{
  "zabbix_export": {
    "version": "7.2",
    "date": "Fecha de carga al Sistema Zabbix",
    "groups": [
      { "uuid": "a1b2c3d4e5f6a1b2c3d4e5f6a1b2c3d4", "name": "Arequipa" },
      { "uuid": "b2c3d4e5f6a1b2c3d4e5f6a1b2c3d4e5", "name": "San Martin" }
    ],

    "hosts": [
      {
        "host":       "nodo-a",
        "name":       "Nodo A",
        "description": "Nodo de red - Arequipa",
        "status":     "0",
        "interfaces": [{
          "default": "1",  "type": "AGENT",
          "useip":  "1",  "ip":   "10.20.5.11",
          "dns":   "",   "port": "10050",
          "interface_ref": "if1"
```

```
 }],
 "groups":    [{ "name": "Arequipa" }],
 "templates": [{ "name": "ICMP Ping" }],
 "tags": [
   { "tag": "region", "value": "arequipa" },
   { "tag": "tipo",   "value": "nodo"     }
 ]
},
{

 "host":        "nodo-b",
 "name":        "Nodo B",
 "description": "Nodo de red - San Martin",
 "status":      "0",
 "interfaces": [{
   "default": "1",  "type": "AGENT",
   "useip":  "1",  "ip":   "172.16.8.45",
   "dns":   "",   "port": "10050",
   "interface_ref": "if1"
 }],
 "groups":    [{ "name": "San Martin" }],
```

```
      "templates": [{ "name": "ICMP Ping" }],
      "tags": [
        { "tag": "region", "value": "san-martin" },
        { "tag": "tipo",   "value": "nodo"      }
      ]
    }
  ]
 }
}
```


Consideraciones Técnicas:

Los grupos indicados en el prompt se crean automáticamente si no existen al momento de la importación.

El campo “uuid” en los grupos debe ser único para cada grupo. Se recomendó dejarlo para que Zabbix lo asigne automáticamente.

Para la carga masiva en la implementación, fue necesario colocar los datos de los hosts en una hoja de cálculo y usar la IA para transformarlos al formato JSON válido.

El archivo generado por la IA fue verificado contra la documentación oficial de Zabbix 7.2 antes de ser usado en la implementación.

8.2 ANEXO B: Manual básico de operación del sistema

Una muestra de este manual se presenta en la Figura B1.

El presente manual tiene como finalidad orientar al personal del NOC en el uso básico del sistema de monitoreo Zabbix.

CONTENIDO

1. Acceso al sistema

- Ingresar a la URL del servidor Zabbix
- Autenticarse con usuario y contraseña

2. Visualización de dashboards

- Seleccionar dashboard regional
- Identificar hosts en estado crítico

3. Gestión de alertas

- Revisar notificaciones en Telegram
- Validar estado del host en Zabbix

4. Acciones ante incidentes

- Verificar conectividad
- Escalar según procedimiento NOC

Figura B1 **Captura de Manual de Usuario para personal NOC.** Fuente. Elaboración propia

8.3 ANEXO C: Script de Asignación Masiva del template Nokia IXR - Interface Monitor

```
#!/usr/bin/env python3
"""
Configura interfaz SNMPv3 y asigna template a hosts Nokia en Zabbix
via API de Zabbix 7.x
```

```
Requisito: pip install requests

Uso: python3 zabbix_snmp_setup.py


SEGURIDAD: Las credenciales y datos sensibles se obtienen de variables de entorno

o se deben configurar manualmente antes de su uso.

"""

import requests

import os

import sys


#
============================================================

# CONFIGURACION - Leer desde variables de entorno o definir aquí

#
============================================================


# Credenciales de Zabbix (obligatorio)

ZABBIX_URL = os.getenv("ZABBIX_URL",
"http://[IP_DEL_SERVIDOR_ZABBIX]/zabbix/api_jsonrpc.php")

ZABBIX_USER = os.getenv("ZABBIX_USER", "[USUARIO_ZABBIX]")

ZABBIX_PASS = os.getenv("ZABBIX_PASS", "[CONTRASENA_ZABBIX]")

# Configuración de grupos y templates
```

```
GROUP_NAME = os.getenv("GROUP_NAME", "[NOMBRE_DEL_GRUPO_EN_ZABBIX]")

TEMPLATE_NAME = os.getenv("TEMPLATE_NAME", "[NOMBRE_DEL_TEMPLATE_EN_ZABBIX]")


# Credenciales SNMPv3 (obligatorio)

SNMP_USER = os.getenv("SNMP_USER", "[USUARIO_SNMP]")

SNMP_AUTHPASS = os.getenv("SNMP_AUTHPASS", "[CLAVE_AUTENTICACION_SNMP]")

SNMP_PRIVPASS = os.getenv("SNMP_PRIVPASS", "[CLAVE_PRIVACIDAD_SNMP]")


#
============================================================

# Verificar que las credenciales no estén vacías (seguridad)

#
============================================================


def check_credentials():

    """Verifica que las credenciales obligatorias estén configuradas."""

    required = [

        ("ZABBIX_URL", ZABBIX_URL),

        ("ZABBIX_USER", ZABBIX_USER),
```

```
        ("ZABBIX_PASS", ZABBIX_PASS),
        ("SNMP_USER", SNMP_USER),
        ("SNMP_AUTHPASS", SNMP_AUTHPASS),
        ("SNMP_PRIVPASS", SNMP_PRIVPASS),
    ]
    missing = []
    for name, value in required:
        if value.startswith("[") and value.endswith("]") or not value:
            missing.append(name)

    if missing:
        print(f"ERROR: Las siguientes variables no están configuradas: {', '.join(missing)}")
        print("Configúralas como variables de entorno o edita el script.")
        print("\nEjemplo de variables de entorno:")
        print("  export ZABBIX_URL='http://IP/zabbix/api_jsonrpc.php'")
        print("  export ZABBIX_USER='admin'")
        print("  export ZABBIX_PASS='tu_contraseña'")
        print("  export SNMP_USER='usuario_snmp'")
        print("  export SNMP_AUTHPASS='clave_auth'")
        print("  export SNMP_PRIVPASS='clave_priv'")
        print("  export GROUP_NAME='Mi Grupo de Red'")
```

```
        print("  export TEMPLATE_NAME='Mi Template SNMP'")
        sys.exit(1)


#
============================================================

# Cliente API de Zabbix

#
============================================================


session = requests.Session()

_id = 1

_token = None


def api_call(method, params):
    global _id, _token
    payload = {"jsonrpc": "2.0", "method": method, "params": params, "id": _id}
    headers = {"Content-Type": "application/json"}
    if _token:
        headers["Authorization"] = f"Bearer {_token}"


    try:
        resp = session.post(ZABBIX_URL, json=payload, headers=headers,
```

```
timeout=30)
        resp.raise_for_status()
    except requests.exceptions.RequestException as e:
        raise Exception(f"Error de conexión a Zabbix: {e}")

    result = resp.json()
    if "error" in result:
        raise Exception(f"API error en '{method}': {result['error']['data']}")
    _id += 1
    return result["result"]


def login():
    global _token
    print("  Autenticando en Zabbix...")
    _token = None
    token = api_call("user.login", {"username": ZABBIX_USER, "password":
ZABBIX_PASS})
    _token = token
    print("  Autenticado correctamente")


def logout():
    api_call("user.logout", {})
```

```
def get_group_id(group_name):
    result = api_call("hostgroup.get", {
        "filter": {"name": [group_name]},
        "output": ["groupid", "name"]
    })
    if not result:
        raise Exception(f"Grupo '{group_name}' no encontrado")
    gid = result[0]["groupid"]
    print(f"  Grupo encontrado: '{group_name}' (ID: {gid})")
    return gid


def get_template_id(template_name):
    result = api_call("template.get", {
        "filter": {"name": [template_name]},
        "output": ["templateid", "name"]
    })
    if not result:
        raise Exception(f"Template '{template_name}' no encontrado")
    tid = result[0]["templateid"]
    print(f"  Template encontrado: '{template_name}' (ID: {tid})")
```

```
    return tid


def get_hosts_in_group(group_id):
    result = api_call("host.get", {
        "groupids": [group_id],
        "output": ["hostid", "host", "name"],
        "selectInterfaces": ["interfaceid", "type", "ip"],
    })
    print(f"  Hosts encontrados en el grupo: {len(result)}")
    return result


def has_snmp_interface(host):
    for iface in host.get("interfaces", []):
        if iface["type"] == "2":  # 2 = SNMP
            return True
    return False


def get_agent_ip(host):
    for iface in host.get("interfaces", []):
        if iface["type"] == "1":  # 1 = Zabbix Agent
            return iface["ip"]
```

```
    return ""


def add_snmp_interface(host):
    ip = get_agent_ip(host)
    if not ip:
        raise Exception("No se encontró IP del agente para crear interfaz SNMP")

    # Configuración SNMPv3 con seguridad máxima (authPriv)
    return api_call("hostinterface.create", {
        "hostid": host["hostid"],
        "main": 1,
        "type": 2,  # SNMP
        "useip": 1,
        "ip": ip,
        "dns": "",
        "port": "161",
        "details": {
            "version": 3,           # SNMPv3
            "bulk": 1,              # Bulk requests habilitado
            "securityname": SNMP_USER,
            "securitylevel": 2,     # 2 = authPriv (autenticación + cifrado)
```

```
            "authprotocol": 0,      # 0 = MD5 (considerar migrar a SHA)
            "authpassphrase": SNMP_AUTHPASS,
            "privprotocol": 0,      # 0 = DES (considerar migrar a AES)
            "privpassphrase": SNMP_PRIVPASS,
            "contextname": "",
        }
    })

def assign_template(host_id, template_id):
    current = api_call("host.get", {
        "hostids": [host_id],
        "selectParentTemplates": ["templateid"]
    })
    existing = current[0].get("parentTemplates", [])
    for t in existing:
        if t["templateid"] == template_id:
            return False  # Template ya asignado
    templates = existing + [{"templateid": template_id}]
    api_call("host.update", {"hostid": host_id, "templates": templates})
    return True
#
============================================================
```

```
# Función principal
#
============================================================


def main():
    print("=" * 55)
    print("  Zabbix SNMPv3 Setup - Nokia IXR Interfaces")
    print("=" * 55)

    # Verificar credenciales antes de continuar
    check_credentials()

    try:
        login()
        group_id = get_group_id(GROUP_NAME)
        template_id = get_template_id(TEMPLATE_NAME)
        hosts = get_hosts_in_group(group_id)
        print(f"\n  Procesando {len(hosts)} hosts...\n")

        ok_snmp = ok_template = skip_snmp = skip_tmpl = 0
        errors = []
```

```
for host in hosts:
    name = host.get("name") or host.get("host", "[NODO_SIN_NOMBRE]")
    hid = host["hostid"]

    # Agregar interfaz SNMP si no existe
    if has_snmp_interface(host):
        print(f"  [SKIP] {name} - SNMP ya configurado")
        skip_snmp += 1
    else:
        try:
            add_snmp_interface(host)
            print(f"  [OK] {name} - Interfaz SNMP agregada")
            ok_snmp += 1
        except Exception as e:
            print(f"  [ERROR] {name} - SNMP: {e}")
            errors.append(f"{name} (SNMP): {e}")
            continue

    # Asignar template
    try:
        if assign_template(hid, template_id):
```

```
            print(f"  [OK] {name} - Template asignado")
            ok_template += 1
        else:
            print(f"  [SKIP] {name} - Template ya asignado")
            skip_tmpl += 1
    except Exception as e:
        print(f"  [ERROR] {name} - Template: {e}")
        errors.append(f"{name} (Template): {e}")

# Resumen final
print(f"\n{'='*55}")
print("  RESUMEN")
print(f"{'='*55}")
print(f"  Hosts procesados    : {len(hosts)}")
print(f"  SNMP agregado       : {ok_snmp}")
print(f"  SNMP ya existía     : {skip_snmp}")
print(f"  Template asignado   : {ok_template}")
print(f"  Template ya existía : {skip_tmpl}")
print(f"  Errores             : {len(errors)}")
if errors:
    print("\n  Detalle de errores:")
```

```
            for err in errors:
                print(f"    - {err}")


        logout()
        print("\n  Proceso completado exitosamente ✓")
    except Exception as e:
        print(f"\n  Error fatal: {e}")
        sys.exit(1)


if __name__ == "__main__":
    main()
```