

PONTIFICIA UNIVERSIDAD CATÓLICA DEL ECUADOR  
FACULTAD DE HÁBITAT, INFRAESTRUCTURA Y  
CREATIVIDAD  
CARRERA DE INGENIERÍA EN TECNOLOGÍAS DE LA  
INFORMACIÓN



Proyecto de Titulación

Diseño e implementación de un prototipo para el monitoreo y gestión automatizada de servicios de TI para pequeñas y medianas empresas

AUTOR: Matheo Ismael Bravo Valencia

TUTOR: Charles Escobar

QUITO DM, ENERO DE 2026

## **DEDICATORIA**

---

Este trabajo está dedicado con todo mi amor a mis padres Miguel y Angeline, quienes han sido los cimientos de mi vida y mi mayor ejemplo de perseverancia. Gracias por cada sacrificio silencioso, por creer en mí cuando yo dudaba y por enseñarme que con esfuerzo y honestidad todo es posible. Este logro es, en gran medida, fruto de su incansable trabajo y amor infinito.

A mis hermanos, Estefanía, Miguel y Benjamín. Gracias por estar siempre a mi lado, por sus consejos, por las risas que aligeraron la carga y por ser mi apoyo incondicional en cada paso de este camino.

A mis sobrinos, Olivia, Emilia, Daphne y Samuel quienes con su inocencia y alegría han sido mi motor y mi inspiración para buscar ser un mejor profesional y persona cada día. Espero que este logro les sirva de ejemplo para que nunca dejen de perseguir sus propios sueños.

Y a mí mismo, por no rendirme nunca.

## **AGRADECIMIENTOS**

---

Agradezco a la Pontificia Universidad Católica del Ecuador por haberme brindado los conocimientos y las herramientas necesarias para formarme como Ingeniero en Tecnologías de la Información.

A mis docentes, por compartir su experiencia y guiarme con exigencia y calidad humana a lo largo de la carrera.

Sin embargo, mi gratitud más profunda y sentida es para mi familia, el pilar que sostuvo este sueño:

A mis padres, porque no existen palabras suficientes para agradecer todo lo que han hecho por mí. Gracias por brindarme la oportunidad de estudiar, por sus desvelos, por su apoyo moral y económico, y por ser mi refugio seguro en los momentos difíciles. Este título también lleva sus nombres.

A mis hermanos, por su paciencia y comprensión durante mis largas horas de estudio frente al computador. Gracias por alentarme a seguir adelante, por celebrar mis pequeñas victorias y por recordarme siempre de lo que soy capaz.

A mis sobrinos, gracias por regalarme sus sonrisas y esos momentos de desconexión tan necesarios, recordándome que la felicidad está en las cosas simples.

Finalmente, a mis amigos y colegas, por el compañerismo, el intercambio de conocimientos y por hacer de esta etapa universitaria una experiencia inolvidable.

## RESUMEN

---

En la actualidad, las pequeñas y medianas empresas (PYMES) enfrentan grandes desafíos para acceder a infraestructuras tecnológicas eficientes debido a limitaciones presupuestarias y falta de personal especializado. La gestión manual de servidores y la ausencia de monitoreo proactivo derivan frecuentemente en interrupciones de servicios críticos, afectando la productividad y generando pérdidas económicas.

El presente proyecto de titulación propone el diseño e implementación de un prototipo de entorno virtualizado enfocado en el monitoreo y gestión automatizada de servicios de TI. La solución se basa en el uso de herramientas de código abierto, utilizando Proxmox VE como hipervisor de tipo 1 para la administración eficiente de máquinas virtuales y contenedores, optimizando así los recursos de hardware disponibles.

Para la gestión operativa, se implementó Zabbix como sistema de monitoreo centralizado, capaz de detectar fallos y métricas de rendimiento en tiempo real. Esta herramienta se integró mediante Webhooks con n8n, un orquestador de automatización, permitiendo la ejecución de flujos correctivos autónomos. Se desarrollaron casos de prueba funcionales, incluyendo el auto-reinicio de servicios ante caídas y el escalado vertical automático de recursos (vCPU y RAM) en respuesta a picos de demanda.

Los resultados demuestran que es posible desplegar una infraestructura autogestionada, estable y escalable utilizando hardware de recursos moderados. La automatización redujo significativamente los tiempos de respuesta ante incidentes, validando que este prototipo es una solución técnica y económicamente viable para modernizar la gestión de TI en las PYMES.

**Palabras clave:** Virtualización, Automatización, Proxmox VE, Zabbix, n8n, PYMES, Monitoreo de TI.

## ABSTRACT

---

Currently, small and medium-sized enterprises (SMEs) face significant challenges in accessing efficient technological infrastructures due to budgetary limitations and a lack of specialized personnel. Manual server management and the absence of proactive monitoring frequently lead to critical service interruptions, affecting productivity and causing economic losses.

This degree project proposes the design and implementation of a virtualized environment prototype focused on the monitoring and automated management of IT services. The solution is based on open-source tools, using Proxmox VE as a Type 1 hypervisor for the efficient administration of virtual machines and containers, thereby optimizing available hardware resources.

For operational management, Zabbix was implemented as a centralized monitoring system capable of detecting failures and performance metrics in real-time. This tool was integrated via Webhooks with n8n, an automation orchestrator, allowing the execution of autonomous corrective workflows. Functional test cases were developed, including the auto-restart of services upon failure and the automatic vertical scaling of resources (vCPU and RAM) in response to demand spikes.

The results demonstrate that it is possible to deploy a self-managed, stable, and scalable infrastructure using moderate resource hardware. Automation significantly reduced response times to incidents, validating that this prototype is a technically and economically viable solution for modernizing IT management in SMEs.

**Keywords:** Virtualization, Automation, Proxmox VE, Zabbix, n8n, SMEs, IT Monitoring.

## TABLA DE CONTENIDOS

---

<b>1. INTRODUCCIÓN</b> .....	<b>14</b>
<b>1.1. JUSTIFICACIÓN</b> .....	<b>14</b>
<b>1.2. Alcance</b> .....	<b>15</b>
<b>1.3. Planteamiento del problema</b> .....	<b>15</b>
<b>1.4. Objetivos</b> .....	<b>16</b>
1.4.1. Objetivo General.....	16
1.4.2. Objetivos Específicos.....	16
<b>2. Marco Teórico</b> .....	<b>17</b>
<b>2.1. Virtualización</b> .....	<b>17</b>
<b>2.2. Tipos de virtualización</b> .....	<b>17</b>
2.2.1. Virtualización de servidores.....	17
2.2.2. Virtualización de almacenamiento .....	17
2.2.3. Virtualización de red.....	17
<b>2.3. Hipervisor</b> .....	<b>18</b>
<b>2.4. Tipos de hipervisores</b> .....	<b>18</b>
2.4.1. Hipervisor de tipo 1 .....	18
2.4.2. Hipervisor de tipo 2 .....	18
<b>2.5. Máquinas virtuales</b> .....	<b>18</b>
<b>2.6. Contenedores de Linux</b> .....	<b>18</b>
<b>2.7. Herramientas para virtualización</b> .....	<b>19</b>
2.7.1. Proxmox .....	19
2.7.2. VMware ESXi .....	19
2.7.3. Hyper-v.....	19
<b>2.8. Herramientas de monitoreo</b> .....	<b>19</b>
2.8.1. Zabbix.....	19
2.8.2. Nagios.....	20
2.8.3. Grafana – Prometheus.....	20
<b>2.9. Herramientas de administración</b> .....	<b>20</b>
2.9.1. Zentyal.....	20
2.9.2. Active directory .....	20
<b>2.10. Herramientas de automatización</b> .....	<b>20</b>
2.10.1. n8n.....	20
2.10.2. Ansible .....	21
<b>2.11. Herramientas de almacenamiento</b> .....	<b>21</b>

2.11.1. Samba.....	21
2.11.2. TrueNAS .....	21
2.11.3. NextCloud.....	21
<b>2.12. Comunicación entre sistemas .....</b>	<b>21</b>
2.12.1. API .....	21
2.12.2. WebHook.....	22
<b>3. Análisis y diseño.....</b>	<b>22</b>
<hr/>	
<b>3.1. Marco Referencial para el diseño del laboratorio virtual.....</b>	<b>22</b>
3.1.1. ITIL v4 .....	23
<b>3.2. Marco metodológico para el análisis y selección de herramientas .....</b>	<b>27</b>
3.2.1. Criterios de evaluación.....	27
3.2.1.1. Costos .....	27
3.2.1.2. Escalabilidad.....	27
3.2.1.3. Facilidad de uso .....	27
3.2.1.4. Compatibilidad .....	28
3.2.1.5. Capacidad de automatización.....	28
3.2.2. Método de análisis .....	28
<b>3.3. Metodología para el prototipado.....</b>	<b>28</b>
<b>3.4. Estructura de la implementación.....</b>	<b>29</b>
3.4.1. Identificación de objetivos y requerimientos .....	29
3.4.1.1. Hardware mínimo necesario.....	29
3.4.1.2. Plataforma de virtualización y gestión.....	29
3.4.1.3. Definición de servicios esenciales para la PYME.....	30
<b>4. Implementación de un prototipo funcional para el monitoreo y gestión automatizada de servicios de TI. ....</b>	<b>30</b>
<hr/>	
<b>4.1. Identificación de los servicios de tecnologías de la información (TI) necesarios para el funcionamiento de una PYME.....</b>	<b>30</b>
4.1.1. Servicios de red.....	31
4.1.2. Servicio de almacenamiento.....	32
4.1.3. Servicio de soporte de TI – Mesa de ayuda (help desk).....	32
4.1.4. Servicios web.....	33
4.1.5. Servicio de base de datos .....	33
4.1.6. Servicio de gestión de usuarios .....	34
4.1.7. Servicio de monitoreo .....	34
4.1.8. Servicio de gestión y automatización .....	35
<b>4.2. Diseño técnico de la arquitectura del laboratorio virtual .....</b>	<b>35</b>

<b>4.3. Comparación y selección de herramientas .....</b>	<b>35</b>
4.3.1. Selección hipervisor.....	35
4.3.1.1. Criterio y pesos .....	36
4.3.1.2. Matriz AHP - Hipervisor.....	36
4.3.2. Selección herramienta de monitoreo .....	36
4.3.2.1. Criterios y peso .....	36
4.3.2.2. Matriz AHP – Sistema de monitoreo .....	37
4.3.3. Selección herramienta automatización .....	37
4.3.3.1. Criterios y pesos.....	37
4.3.3.2. Matriz AHP – Automatización.....	37
4.3.4. Descripción general de la arquitectura .....	38
4.3.5. Componentes de la arquitectura .....	38
4.3.6. Diseño de máquinas virtuales y contenedores .....	39
4.3.7. Gestión de la red del laboratorio virtual .....	40
4.3.8. Diagrama general de la arquitectura.....	41
<b>4.4. Implementación del laboratorio virtual.....</b>	<b>41</b>
4.4.1. Infraestructura física del laboratorio.....	41
4.4.2. Instalación hipervisor.....	42
4.4.3. Configuración inicial proxmox .....	45
4.4.4. Creación de máquinas virtuales.....	47
4.4.4.1. Creación VM Zentyal.....	48
4.4.4.2. Implementación Zentyal.....	52
4.4.4.3. Creación VM TrueNAS .....	57
4.4.4.4. Implementación TrueNAS .....	58
4.4.4.5. Creación VM NextCloud .....	62
4.4.4.6. Implementación VM NextCloud .....	63
4.4.4.7. Creación VM HelpDesk GLPI .....	64
4.4.4.8. Implementación VM HelpDesk.....	64
4.4.5. Creación de contenedores .....	65
4.4.5.1. Creación contenedor Zabbix .....	66
4.4.5.2. Creación CT n8n.....	68
4.4.5.3. Creación contenedor BDD .....	68
4.4.5.4. Creación contenedor nginx - Load Balancer.....	69
4.4.5.5. Implementación contenedor Load Balancer.....	69
4.4.5.1. Creación contenedor página web.....	69
4.4.5.2. Implementación CT Web Server .....	70
4.4.6. Resumen Creación e Implementación VMs y CTs .....	71

<b>4.5. Implementación del monitoreo.....</b>	<b>73</b>
4.5.1. Instalación Zabbix.....	73
4.5.2. Instalaciones agentes Zabbix.....	75
4.5.2.1. Agentes en CTs.....	75
4.5.3. Página monitoreo personalizada.....	78
4.5.3.1. Zabbix API .....	78
4.5.3.2. Página web.....	79
<b>4.6. Implementación de la automatización .....</b>	<b>81</b>
Flujos n8n.....	81
4.6.1. Flujo activación de servicios .....	81
4.6.2. Flujo auto escalado vertical.....	92
4.6.2.1. Flujo .....	93
4.6.3. Flujo auto reinicio servicios .....	96
4.6.3.1. Auto inicio de página web.....	96
<b>4.7. Pruebas.....</b>	<b>97</b>
4.7.1. Pruebas Flujo 1 .....	97
4.7.1.1. Resumen Flujo 1 .....	98
4.7.2. Pruebas Flujo 2 .....	98
4.7.2.1. Escalado Vertical (vCPUs).....	98
4.7.2.2. Escalado Vertical (RAM).....	100
4.7.2.3. Resumen flujo autoescalado.....	102
4.7.3. Pruebas flujo 3 .....	102
4.7.3.1. Flujo página web.....	102
<b>4.8. Resumen pruebas.....</b>	<b>103</b>
<b>5. Conclusiones y Recomendaciones .....</b>	<b>103</b>
<hr/>	
<b>5.1. Conclusiones.....</b>	<b>104</b>
<b>5.2. Recomendaciones.....</b>	<b>105</b>
<b>6. Referencias Bibliográficas.....</b>	<b>107</b>
<hr/>	

## ÍNDICE DE TABLAS

Tabla 1 Matriz AHP – Hipervisor.....	36
Tabla 2 Matriz AHP - Sistema de monitoreo .....	37
Tabla 3 Matriz AHP - Automatización .....	38
Tabla 4 Características del servidor .....	41
Tabla 5 Características del procesador .....	41
Tabla 6 Características de la memoria RAM.....	42
Tabla 7 Configuración de almacenamiento .....	42
Tabla 8 Detalle VMs.....	48
Tabla 9 Comandos VM GLPI-HelpDesk .....	64
Tabla 10 Detalle Contenedores .....	65
Tabla 11 Comandos Instalación node.js.....	71
Tabla 12 Comandos creación estructura página web.....	71
Tabla 13 IDs de servicios.....	72
Tabla 14 Comandos API Zabbix.....	79
Tabla 15 Límites escalado de servicios.....	92
Tabla 16 Comandos para estresas CPU y RAM .....	98
Tabla 17 Resumen pruebas flujos n8n .....	103

## ÍNDICE DE FIGURAS

Ilustración 1 Diagrama general de la arquitectura .....	41
Ilustración 2 Ventoy .....	43
Ilustración 3 Instalación proxmox.....	44
Ilustración 4 Primer ingreso Proxmox.....	45
Ilustración 5 Revisión componentes.....	45
Ilustración 6 Script Post install .....	46
Ilustración 7 Repositorio enterprise proxmox .....	46
Ilustración 8 Script en proxmox.....	47
Ilustración 9 Deshabilitar repositorio enterprise.....	47
Ilustración 10 Resumen Post Install .....	47
Ilustración 11 ISO zentyal .....	49
Ilustración 12 Carga de ISOs en proxmox.....	49
Ilustración 13 Creación de VM.....	49
Ilustración 14 Asignar nombre a VM.....	50
Ilustración 15 Selección ISO para VM.....	50
Ilustración 16 Asignación Almacenamiento.....	51
Ilustración 17 Asignación vCPUs .....	51
Ilustración 18 Asignación RAM .....	51
Ilustración 19 Selección Interfaz de Red.....	51
Ilustración 20 Resumen de VM Zentyal.....	52
Ilustración 21 Zentyal Setup Inicial .....	53
Ilustración 22 Servicios Zentyal.....	54
Ilustración 23 Dashboard Zentyal .....	54
Ilustración 24 Revisión servicios Zentyal .....	55
Ilustración 25 IP estática Zentyal .....	55
Ilustración 26 IP estática en Router.....	56
Ilustración 27 Rango IPs Zentyal DHCP.....	56
Ilustración 28 Dominio Zentyal .....	57
Ilustración 29 Resumen VM TrueNAS .....	58
Ilustración 30 Login TrueNAS .....	59
Ilustración 31 Creación disco para VMs y CTs.....	60
Ilustración 32 Configuración Discos.....	60
Ilustración 33 Creación Pool TrueNAS.....	61
Ilustración 34 Agregar almacenamiento NFS en proxmox .....	61
Ilustración 35 Disco TrueNAS en ProxMox .....	62
Ilustración 36 Resumen VM NextCloud .....	62
Ilustración 37 Selección NextCloud.....	63
Ilustración 38 Aplicaciones NextCloud.....	63
Ilustración 39 Resumen VM GLPI.....	64
Ilustración 40 Zabbix Helper Scripts.....	66
Ilustración 41 Creación contenedor Zabbix.....	66
Ilustración 42 Selección almacenamiento CT .....	67
Ilustración 43 Resumen CT Zabbix .....	67

Ilustración 44 Resumen CT n8n.....	68
Ilustración 45 Resumen CT Base de datos .....	68
Ilustración 46 Resumen CT Load Balancer.....	69
Ilustración 47 Ubicación template CTs .....	70
Ilustración 48 Creación CT PáginaWeb .....	70
Ilustración 49 Resumen instalación node.js.....	71
Ilustración 50 Resumen Servicios levantados .....	72
Ilustración 51 Discos en TrueNAS.....	72
Ilustración 52 Creación Base de datos para zabbix.....	73
Ilustración 53 Creación estructura base de datos para zabbix .....	74
Ilustración 54 Resumen instalación zabbix .....	74
Ilustración 55 Agente para CT BDD.....	75
Ilustración 56 Comandos instalación agente .....	75
Ilustración 57 Agregar servidor en agente.....	76
Ilustración 58 Agregar host en Zabbix .....	76
Ilustración 59 Información Host en Zabbix.....	77
Ilustración 60 Servicios registrados en Zabbix.....	77
Ilustración 61 Zabbix API Token.....	78
Ilustración 62 Pruebas API Zabbix .....	78
Ilustración 63 Dashboard página web .....	80
Ilustración 64 Dashboard - Información importante.....	80
Ilustración 65 Menú de navegación .....	80
Ilustración 66 Dashboard n8n .....	81
Ilustración 67 Flujo reactivación de servicios .....	82
Ilustración 68 Acción Inicio Flujo .....	82
Ilustración 69 Configuración primer Trigger .....	83
Ilustración 70 Crear media type Zabbix .....	83
Ilustración 71 Parámetros alerta.....	84
Ilustración 72 Creación accion en Zabbix .....	84
Ilustración 73 Asignación a todos los hosts.....	85
Ilustración 74 Webhook test n8n.....	85
Ilustración 75 Prueba Webhook .....	86
Ilustración 76 Revisión werbhook test en n8n.....	86
Ilustración 77 Contenido webhook .....	87
Ilustración 78 Servidor Discord .....	87
Ilustración 79 WebHook Discord.....	88
Ilustración 80 Configuración nodo discord .....	88
Ilustración 81 Mensaje de alerta .....	89
Ilustración 82 Creación usuario proxmox .....	89
Ilustración 83 ProxMox API Token .....	90
Ilustración 84 Nodo SSH.....	90
Ilustración 85 Script para iniciar servicios .....	91
Ilustración 86 Varios resultados para un nodo.....	91
Ilustración 87 Mensaje reitento.....	92

Ilustración 88 Flujo autoescalado vertical .....	93
Ilustración 89 Configuración media type flujo 2 .....	94
Ilustración 90 Trigger Auto Escalado .....	94
Ilustración 91 Nodo Code .....	95
Ilustración 92 Nodo HTTP Request .....	95
Ilustración 93 Mensaje discord flujo 2 .....	96
Ilustración 94 Alerta Zabbix página web .....	96
Ilustración 95 Flujo autoinicio node.js .....	97
Ilustración 96 Caída de un servicio .....	97
Ilustración 97 Revisión en Zabbix .....	97
Ilustración 98 Alerta flujo 1 en Discord .....	98
Ilustración 99 CPU estresado .....	99
Ilustración 100 Después del flujo - aumento cores .....	100
Ilustración 101 Mensaje discord autoescalado .....	100
Ilustración 102 Estado CT previo comando estrés RAM .....	100
Ilustración 103 Comando para estresar RAM .....	101
Ilustración 104 Alto uso de RAM .....	101
Ilustración 105 Notificación en discord flujo 2 .....	101
Ilustración 106 Verifcaición flujo 2 en ProxMox .....	102
Ilustración 107 Parar servicio web .....	102
Ilustración 108 Mensaje en discord flujo 3.1 .....	103
Ilustración 109 Verificación estado servicio web flujo 3.1 .....	103

# CAPÍTULO I: INTRODUCCIÓN

## 1. INTRODUCCIÓN

### 1.1. JUSTIFICACIÓN

En la actualidad muchas instituciones y empresas pequeñas se enfrentan a muchos problemas cuando intentan acceder a ambientes tecnológicos donde puedan desarrollar, implementar y administrar sus propios servicios de TI los cuales se han convertido en elementos indispensables para las operaciones diarias de cualquier empresa. Estos problemas pueden deberse a distintos factores como presupuestos reducidos, falta de conocimiento técnico o simplemente la complejidad que tiene la implementación de todos los servicios. Por esta situación, en este proyecto de titulación, se propone la creación de un prototipo de un entorno virtualizado, el cual se presenta como una opción viable por ser económica y escalable, además de tener ventajas como el control absoluto sobre su propia infraestructura sin depender de terceros. Esto se logra con ayuda de la virtualización, en específico la de tipo 1 (bare metal), la cual permite crear y administrar múltiples máquinas virtuales y/o contenedores directamente sobre la infraestructura física, que puede ser un computador tradicional con recursos básicos, lo que también demuestra que se pueden optimizar los recursos disponibles siendo los principales el procesamiento, memoria y almacenamiento, lo que resulta también en ahorro en el costo total de la implementación. Asimismo, al tener toda la infraestructura centralizada se simplifican las tareas de administración, control y monitoreo de los sistemas.

Otro de los problemas que se presentan en este contexto es que la gestión manual de los servidores (cuando existe una gestión porque no siempre es así) dificulta la detección temprana de incidentes, la optimización de recursos y la planificación del crecimiento de la infraestructura. Como consecuencia, las acciones correctivas suelen aplicarse de forma reactiva, después de que ocurren los fallos, lo que puede resultar en pérdidas económicas, de información y riesgos de interrupción en los servicios críticos de la organización.

Al utilizar virtualización se tienen varios beneficios no solo económicos, sino que también se puede maximizar el aprovechamiento de los recursos físicos y simplificar la administración de sistemas, mientras que el monitoreo proporciona información en tiempo real sobre el estado, rendimiento de los servicios y la disponibilidad general del sistema. Además de todos los beneficios mencionado, con la autogestión de los servicios se pretende que el entorno en sí sea autosuficiente, permitiendo que la operación y servicios de la empresa estén disponibles la mayor parte del tiempo.

También se debe mencionar que durante el desarrollo de este proyecto de titulación se presentaron varios inconvenientes en proveedores de servicios cloud, esto nos demuestra que las empresas como Amazon, azure, cloudfare, etc. Son vulnerables de algún modo.

Por todos estos motivos mencionados anteriormente este prototipo se presenta como solución para las PYMEs ya que ofrece entornos flexibles, escalables, seguros y de fácil mantenimiento, sin tener que realizar una inversión inicial alta, ni contar con un gran personal para el monitoreo y gestión del entorno. El uso de entornos virtualizados combinados con herramientas de monitoreo y gestión automatizado es una solución eficaz y accesible para enfrentar estos desafíos.

Por estas razones, el presente proyecto que propone el diseño e implementación de un prototipo de monitoreo y gestión automatizada de servicios de TI, enfocado a las necesidades y capacidades de las pequeñas y medianas empresas busca demostrar que es posible establecer una infraestructura tecnológica confiable, escalable y eficiente, sin requerir altos costos de implementación ni personal altamente especializado.

Finalmente, esta investigación aportará conocimiento práctico sobre la integración de tecnologías de virtualización, automatización y monitoreo, constituyendo una base sólida para futuras iniciativas que impulsen la transformación digital y la modernización tecnológica de las PYMEs.

## **1.2. Alcance**

Este trabajo de titulación se enfoca en la creación de un entorno virtualizado, funcional y replicable, destinado principalmente a pequeñas empresas que requieren gestionar y monitorear su infraestructura de TI sin incurrir en grandes inversiones. El laboratorio incluirá servicios esenciales para las operaciones diarias de una PYME en general como DNS, DHCP, servidor web, bases de datos, herramientas de monitoreo, etc. Este entorno será implementado en hardware con recursos moderados, demostrando su viabilidad y escalabilidad. Si bien el enfoque principal es para las pequeñas y medianas empresas, este laboratorio podría también ser utilizado con fines más allá de estos como startups que necesitan una infraestructura sólida durante el comienzo de sus funciones, de igual forma se podría utilizar como un entorno de pruebas y desarrollo para DevOps. Considerando el mundo tan tecnológico en el que nos encontramos sería interesante también aplicarlo a nivel educativo en instituciones educativas técnicas, universidades o programas de capacitación profesional.

## **1.3. Planteamiento del problema**

Las pequeñas y medianas empresas dependen cada vez más de la infraestructura tecnológica y sus servicios de TI para que sus operaciones no presenten interrupciones y se pueda garantizar la disponibilidad de la información. Sin embargo, muchas de ellas no poseen los recursos económicos, humanos y técnicos necesarios para implementar soluciones robustas de administración y monitoreo de servidores. Estas deficiencias generan entornos de TI con estructuras mal diseñadas, difíciles de mantener y con una alta vulnerabilidad ante fallas y/o sobrecargas de los sistemas. Todo esto afecta la disponibilidad de los servicios e incluso en caídas completas de ciertos servicios lo que directamente perjudica la productividad de las

empresas y puede resultar en que los clientes pierdan confianza o que se tengan pérdidas económicas.

La mayoría de las veces estos problemas son identificados cuando las fallas son tan severas que todos los miembros de la empresa empiezan a sufrir los efectos de estos problemas por lo que las soluciones suelen reactivas que derivan en pequeños parches temporales que no solucionan los errores de raíz. Por todo lo mencionado, surge la necesidad de diseñar e implementar un prototipo que permita automatizar el monitoreo y la gestión de los servidores en un entorno virtualizado, de manera que el sistema pueda detectar fallas o sobrecargas, notificar oportunamente a los responsables y tomar acciones básicas de recuperación. Esta propuesta ayudará a mejorar la disponibilidad y rendimiento de los servicios de TI en las PYMEs optimizando la administración sin que se requiera una gran inversión inicial en infraestructura o personal.

## **1.4. Objetivos**

### **1.4.1. Objetivo General**

Diseñar e implementar un prototipo funcional para el monitoreo y gestión automatizada de servicios de TI en un entorno virtualizado, con el fin de optimizar la administración, disponibilidad y rendimiento de los servicios tecnológicos en una pequeña o mediana empresa (PYME).

### **1.4.2. Objetivos Específicos**

- Identificar los servicios de tecnologías de la información (TI) necesarios para el funcionamiento de una PYME, con el propósito de determinar los componentes críticos que deben ser monitorizados y gestionados dentro del entorno virtual.
- Diseñar la arquitectura del laboratorio virtual, definiendo la estructura de máquinas virtuales, contenedores y servicios de TI, de acuerdo con las necesidades de una PYME.
- Implementar un entorno virtual de pruebas que integre máquinas virtuales, contenedores y servicios esenciales, permitiendo la simulación de un ecosistema TI representativo de una infraestructura empresarial.
- Realizar pruebas de validación del prototipo, simulando escenarios de caída y sobrecarga de servidores, con el fin de evaluar la efectividad del sistema en la detección, notificación y gestión automatizada de incidentes.

# **CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA**

## **2. Marco Teórico**

### **2.1. Virtualización**

El concepto general de virtualización es la abstracción de recursos físicos para poder utilizarlos en varias máquinas virtuales utilizando un hipervisor. Amazon define la virtualización de la siguiente forma. *“Es un proceso que permite a una computadora compartir sus recursos de hardware con varios entornos separados de forma digital. Cada entorno virtualizado se ejecuta dentro de los recursos asignados, como la memoria, la potencia de procesamiento y el almacenamiento.”* (¿Qué Es La Virtualización? - Explicación de La Virtualización de La Computación En La Nube - AWS, 2022)

### **2.2. Tipos de virtualización**

#### **2.2.1. Virtualización de servidores**

*La virtualización de servidores es un proceso que particiona un servidor físico en múltiples servidores virtuales. Es una forma eficaz y rentable de utilizar los recursos del servidor y de implementar los servicios de TI en una organización.* (¿Qué Es La Virtualización? - Explicación de La Virtualización de La Computación En La Nube - AWS, 2022)

#### **2.2.2. Virtualización de almacenamiento**

*La virtualización del almacenamiento utiliza todo el almacenamiento físico de datos y crea una gran unidad de almacenamiento virtual que se puede asignar y controlar mediante un software de administración.* (¿Qué Es La Virtualización? - Explicación de La Virtualización de La Computación En La Nube - AWS, 2022)

#### **2.2.3. Virtualización de red**

*“Cualquier red de computadoras dispone de elementos de hardware como conmutadores, enrutadores y firewalls. (...) La virtualización de la red es un proceso que combina todos estos recursos de red para centralizar las tareas administrativas. Los administradores pueden ajustar y controlar estos elementos virtualmente sin tocar los componentes físicos, lo que simplifica enormemente la administración de la red.”* (¿Qué Es La Virtualización? - Explicación de La Virtualización de La Computación En La Nube - AWS, 2022)

## 2.3. Hipervisor

*Un hipervisor es un software que permite que varias máquinas virtuales (VM), cada una con su propio sistema operativo (SO), se ejecuten en un servidor físico. El hipervisor agrupa y asigna recursos informáticos físicos según las necesidades de la máquina virtual, lo que permite eficiencia, flexibilidad y escalabilidad (Susnjara & Smalley - IBM, 2024)*

## 2.4. Tipos de hipervisores

### 2.4.1. Hipervisor de tipo 1

*Se ejecuta directamente en el hardware físico de la computadora subyacente, interactuando directamente con su unidad central de procesamiento (CPU), memoria y almacenamiento físico. Por este motivo, los hipervisores de tipo 1 también se denominan hipervisores bare metal o hipervisores nativos. Un hipervisor de tipo 1 reemplaza al sistema operativo host. (Susnjara & Smalley - IBM, 2024)*

### 2.4.2. Hipervisor de tipo 2

*No se ejecuta directamente en el hardware subyacente. Se ejecuta más bien como una aplicación en un sistema operativo. Sin embargo, debido a que un hipervisor de tipo 2 debe acceder a los recursos de cómputo, memoria y red a través del sistema operativo host, introduce problemas de latencia que pueden afectar el rendimiento. (Susnjara & Smalley - IBM, 2024)*

## 2.5. Máquinas virtuales

Denominados durante este proyecto como VMs según Google son “Una versión digitalizada de un ordenador físico. Las máquinas virtuales pueden ejecutar programas y sistemas operativos, almacenar datos, conectarse a redes y cumplir otras funciones informáticas. Sin embargo, una máquina virtual usa recursos totalmente virtuales en lugar de componentes físicos.” (*¿Qué Es Una Máquina Virtual? Usos Y Ventajas de Las Máquinas Virtuales | Google Cloud, 2025*)

## 2.6. Contenedores de Linux

Usualmente denominados LXC según QindelGroup “Son una tecnología de virtualización ligera que permite la ejecución y el aislamiento de aplicaciones y servicios en entornos Linux. A diferencia de las máquinas virtuales tradicionales, que emulan todo un sistema operativo, los contenedores Linux comparten el mismo kernel del sistema operativo anfitrión, lo que los hace más eficientes y rápidos.” (*¿Qué Son Los Contenedores Linux O LXC (Linux Containers)? - Qindel: Consultoría IT, 2023*)

## **2.7. Herramientas para virtualización**

### **2.7.1. Proxmox**

*Proxmox Virtual Environment es una potente plataforma de virtualización de servidores de código abierto que gestiona dos tecnologías de virtualización: KVM (Máquina Virtual Basada en Kernel) para máquinas virtuales y LXC para contenedores, con una única interfaz web. Además, integra herramientas listas para usar para configurar la alta disponibilidad entre servidores, el almacenamiento definido por software, las redes y la recuperación ante desastres. (Proxmox, 2019)*

### **2.7.2. VMware ESXi**

*ESXi de VMware es un hipervisor de tipo 1 o coloquialmente bare-metal que se ejecuta directamente en un servidor físico instalado y se puede utilizar sea cual sea el sistema operativo. El software vSphere se utiliza para la administración. (Todo Lo Que Necesita Saber Sobre ESXi: Ventajas Y Aplicaciones, 2022)*

### **2.7.3. Hyper-v**

*Hyper-V es la tecnología de hipervisor de nivel empresarial de Microsoft integrada en Windows Server y Windows. Proporciona funcionalidades de virtualización de hardware que permiten a las organizaciones crear, administrar y ejecutar máquinas virtuales a escala. Como hipervisor de tipo 1, Hyper-V se ejecuta directamente en el hardware informático, lo que proporciona un rendimiento casi nativo y un aislamiento sólido para cargas de trabajo virtualizadas. (microsoft, 2025)*

## **2.8. Herramientas de monitoreo**

*Las herramientas de monitorización son un sistema informático que se utiliza para seguir y supervisar en tiempo real el rendimiento y la funcionalidad de las redes, infraestructuras, sistemas y aplicaciones de una empresa. Estas proporcionan información crítica que puede ayudar a identificar y solucionar problemas antes de que afecten a la productividad o la seguridad. (Santander Universidades, 2024)*

### **2.8.1. Zabbix**

*Zabbix es un software que monitorea numerosos parámetros de una red y la salud y la integridad de servidores, máquinas virtuales, aplicaciones, servicios, bases de datos, sitios web, la nube y más. (2 Qué Es Zabbix, 2025)*

### **2.8.2. Nagios**

*Nagios es un software de código abierto diseñado para la monitorización continua de sistemas, redes e infraestructuras de TI. Su función principal es supervisar equipos y servicios con el objetivo de alertar a los administradores cuando el comportamiento de los mismos no sea el esperado. (godaddy, 2024)*

### **2.8.3. Grafana – Prometheus**

*Durante la explotación de los sistemas informáticos es fundamental monitorizar su funcionamiento para poder valorar su rendimiento y facilitar el diagnóstico de cualquier problema. Una de las soluciones de monitorización más utilizadas hoy en día consiste en combinar Prometheus (una base de datos de series temporales) con Grafana (una interfaz web), para poder definir los dashboards que visualicen gráficamente las métricas de nuestros sistemas. (Introducción a La Monitorización Con Prometheus Y Grafana, 2023)*

## **2.9. Herramientas de administración**

### **2.9.1. Zentyal**

*Zentyal es una distribución de GNU/Linux, basada en Ubuntu Server, que funciona como una solución de servidor unificada y de código abierto para pequeñas y medianas empresas (PYMES). Diseñada para ser una alternativa a Windows Server, integrando servicios esenciales como controlador de dominio (compatible con Active Directory), servidor de correo, servidor de archivos (Samba), firewall, VPN, DHCP, DNS, y gestión de groupware (agenda, contactos), todo gestionable desde una interfaz web intuitiva. Zentyal S.L. (s. f.). Zentyal,*

### **2.9.2. Active directory**

*Active Directory (AD) es una base de datos y un conjunto de servicios que conectan a los usuarios con los recursos de red que necesitan para realizar su trabajo. La base de datos (o el directorio) contiene información crítica sobre su entorno, incluidos los usuarios y las computadoras que hay y quién puede hacer qué. (Quest Software, 2026)*

## **2.10. Herramientas de automatización**

### **2.10.1. n8n**

*n8n es una herramienta de automatización del flujo de trabajo que te permite programar tareas y mover datos entre muchas de las aplicaciones, herramientas, plataformas y servicios que tu equipo utiliza a diario. (Hostiner, Boada, 2025)*

## **2.10.2. Ansible**

*Ansible es un motor open source que automatiza una gran cantidad de procesos informáticos, como la preparación de la infraestructura, la gestión de la configuración, la implementación de las aplicaciones y la organización de los sistemas. (Ansible: ¿Qué Es Y Cómo Funciona?, 2024)*

## **2.11. Herramientas de almacenamiento**

### **2.11.1. Samba**

*Samba es una suite de software que proporciona interoperabilidad fluida entre los sistemas operativos Windows, Linux y Unix en un entorno de red. Permite que estos sistemas compartan archivos, impresoras y otros recursos, lo que permite a los usuarios de diferentes plataformas colaborar y acceder a recursos compartidos de forma transparente. (What Is Samba?, 2023)*

### **2.11.2. TrueNAS**

*Es un sistema operativo de almacenamiento abierto diseñado para ayudar a las empresas a gestionar, almacenar y compartir datos en toda la red. Basado en un sistema de archivos OpenZFS, el sistema permite a las empresas facilitar el almacenamiento de bloques, objetos, archivos y aplicaciones en entornos en la nube o locales. (TrueNAS CORE, 2026)*

### **2.11.3. NextCloud**

*Nextcloud es un potente software de nube no comercial para el autoalojamiento y el cloud computing adecuado tanto para fines privados como empresariales. El software es una bifurcación, es decir, una rama de desarrollo paralela del proyecto OwnCloud. OwnCloud y Nextcloud son alternativas de código abierto a los servicios de almacenamiento en la nube comerciales como Amazon AWS, Google Drive o iCloud. (¿Qué Es Nextcloud? La Solución En La Nube, 2025)*

## **2.12. Comunicación entre sistemas**

### **2.12.1. API**

*Las API son mecanismos que permiten a dos componentes de software comunicarse entre sí mediante un conjunto de definiciones y protocolos. Por ejemplo, el sistema de software del instituto de meteorología contiene datos meteorológicos diarios. La aplicación meteorológica de su teléfono “habla” con este sistema a través de las API y le muestra las actualizaciones meteorológicas diarias en su teléfono. (Amazon Web Services, 2026)*

## 2.12.2. WebHook

*Un webhook consiste en una comunicación ligera basada en eventos que envía datos automáticamente entre las aplicaciones a través del protocolo HTTP. Los webhooks, que se activan por eventos específicos, automatizan la comunicación entre las interfaces de programación de aplicaciones (API) y se pueden utilizar para ejecutar flujos de trabajo, como en los entornos de GitOps. (¿Qué Es Un Webhook Y Para Qué Sirve? | Automatización Integral Empresarial de La Mano de Red Hat, 2024)*

## 2.13. Conceptos Zabbix

### 2.13.1. Media type

*Son los canales de entrega utilizados para enviar notificaciones y alertas desde Zabbix. (Zabbix LLC, s.f.)*

### 2.13.2. Trigger

*Las expresiones de iniciador permiten definir un umbral del estado de los datos "aceptable". Por lo tanto, si los datos entrantes superan lo aceptable, un iniciador se "activa" o cambia su estado a PROBLEMA. (Zabbix LLC, s.f.)*

## **CAPÍTULO 3: ANÁLISIS Y DISEÑO DEL LABORATORIO VIRTUAL**

### 3. Análisis y diseño

#### 3.1. Marco Referencial para el diseño del laboratorio virtual

El diseño de este laboratorio virtual se fundamenta en un marco de referencia reconocidos internacionalmente por la gestión y gobernanza de tecnologías de la información. Este marco proporciona lineamientos para estructurar procesos orientados a garantizar la continuidad operativa de los servicios propuestos, así como establecer controles adecuados que aseguren una operación eficiente, confiable y segura dentro del entorno tecnológico. Dado que este proyecto está orientado al **monitoreo, gestión** automatizada y operación de servicios de TI en entornos virtualizados, se ha tomado como referencia el marco de referencia ITILv4 el cual según (Aranda, 2019), *“Proporciona la guía que necesitan las organizaciones para abordar los nuevos desafíos de la administración de servicios y utilizar el potencial de la tecnología moderna. Está diseñado para garantizar un sistema flexible, coordinado e integrado para el gobierno y la gestión efectiva de los servicios habilitados para TI.”*

ITIL v4 resulta interesante para este proyecto porque aborda prácticas directamente relacionadas con la supervisión de infraestructura, la atención de incidentes, la disponibilidad de servicios y la automatización de procesos operativos.

Se tiene que enfatizar en que **este marco no se implementa completamente** dentro del prototipo, pero se utiliza como referencia teórica para fundamentar y justificar las prácticas de monitoreo, operación y seguridad consideradas en el diseño del laboratorio virtual.

### **3.1.1. ITIL v4**

Según el Departamento de Consultoría (2020), que nos da una pequeña explicación sobre que es ITIL, según ellos *“ITIL es una guía de buenas prácticas para la gestión de servicios de tecnologías de la información (TI). La guía ITIL ha sido elaborada para abarcar toda la infraestructura, desarrollo y operaciones de TI y gestionarla hacia la mejora de la calidad del servicio.”*

Aunque este prototipo no implementa todas las prácticas que ITIL recomienda, si se utilizan varias de sus recomendaciones como guía conceptual para el diseño del laboratorio virtual, especialmente lo enfocado a monitoreo, operación, continuidad y automatización de servicios. Principalmente el laboratorio uso como referencia los temas abordados en ITILv4 capítulos 5.2 Prácticas de gestión de servicio en el cual se desarrollan temas como:

- Gestión de activos de TI
- Monitoreo y gestión de eventos
- Gestión de problemas
- Gestión de la configuración del servicio
- Gestión de la continuidad del servicio.

En el capítulo 5.3 también se abordan temas que aportan al desarrollo de este proyecto de titulación como la gestión de infraestructura y plataforma.

A continuación, se describen las prácticas más relevantes para este proyecto:

#### **a) Gestión de Activos de TI**

Esto nos permite identificar, registrar y controlar todos los componentes que forman parte del entorno tecnológico: máquinas virtuales, contenedores, almacenamiento, servicios, configuraciones y software.

Específicamente nos permite:

- Control de las VMs y CTs.
- Asignación adecuada de recursos (CPU, RAM y almacenamiento).

Esta gestión garantiza que los recursos sean utilizados de forma eficiente, evitando sobrecargos o desperdicio de su capacidad.

## **b) Monitoreo y gestión de eventos**

Esta práctica es la que más se relaciona con los objetivos de este proyecto, según ITIL: *“El propósito de la práctica de monitoreo y gestión de eventos es observar sistemáticamente los servicios y componentes del servicio, y registrar e informar cambios seleccionados de estado identificados como eventos. Esta práctica identifica y prioriza infraestructura, servicios, procesos comerciales y eventos de seguridad de la información, y establece la respuesta adecuada a estos eventos, incluida la respuesta a condiciones que podrían conducir a posibles fallas o incidentes”* (AXELOS, 2019)

En este laboratorio se implementa n8n como herramienta de automatización para responder a eventos críticos y otras herramientas para el monitoreo, esto nos permite:

- Enviar alertas automáticas cuando un servicio presenta degradación o falla.
- Ejecutar acciones correctivas sin intervención humana (por ejemplo, reiniciar un servicio, levantar una VM o liberar recursos).
- Activar flujos automatizados basados en reglas (por ejemplo: CPU > 90% → crear alerta → reiniciar proceso → registrar acción).

Esta integración con herramientas de monitoreo y automatización crea un entorno autogestionado el cual es el objetivo principal del proyecto.

## **c) Gestión de disponibilidad**

Según el documento de ITIL describen el objetivo de este tema como *“El propósito de la práctica de gestión de disponibilidad es garantizar que los servicios entreguen niveles acordados de disponibilidad para satisfacer las necesidades de los clientes y usuarios.”* (AXELOS, 2019)

Aquí ITIL menciona dos conceptos fundamentales que son métricas que permiten medir la estabilidad del servicio y la rapidez con la que se recupera antes fallas, estos son:

- **MTBF:** Tiempo medio entre fallas. Mide la frecuencia con la que falla el servicio.
- **MTRS:** Tiempo medio para restaurar el servicio. Mide la rapidez con la que se restablece el servicio después de una falla.

(AXELOS, 2019)

Cuando combinamos monitoreo, métricas y automatización, el laboratorio se orienta a mejorar la disponibilidad de los servicios y a reducir significativamente los tiempos de inactividad.

#### **d) Gestión de la configuración del servicio**

Aunque esta práctica no se enfoca en su totalidad con la configuración completa de los servicios, al tener como marco de referencia a ITIL en el que se propone garantizar que la información sobre los componentes del servicio, sus configuraciones y las relaciones entre ellos se encuentre identificada, documentada y actualizada. Se realizará una pequeña configuración de los servicios seleccionados. Para ello utilizaremos las buenas prácticas que recomienda ITIL como:

- Documentación estructurada de máquinas virtuales, contenedores y servicios desplegados.
- Registro de versiones, cambios y configuraciones relevantes dentro del entorno virtual.
- Identificación de dependencias entre servicios (por ejemplo: el monitoreo depende de la disponibilidad de los agentes instalados en cada VM).

#### **e) Gestión de la continuidad del servicio**

Uno de los principales objetivos de este proyecto de titulación es garantizar que los servicios de TI puedan continuar operando incluso ante fallas graves o interrupciones inesperadas. Según ITIL, *“El propósito de la práctica de gestión de la continuidad del servicio es garantizar que la disponibilidad y el desempeño de un servicio se mantengan en niveles suficientes en caso de un desastre.” (AXELOS, 2019)*

En este entorno virtual se implementan de la siguiente forma:

- Snapshots y backups periódicos en Proxmox para proteger el estado de las máquinas virtuales.
- Procedimientos de restauración rápida ante fallas.
- Segmentación de servicios en múltiples VMs o CTs para evitar puntos únicos de falla.
- Integración de automatización con n8n, que puede ejecutar acciones de continuidad como:
  - Encender nodos o VMs auxiliares.
  - Reiniciar servicios críticos en caso de caída.
  - Ejecutar scripts de recuperación remota.
  - Enviar notificaciones automáticas cuando un servicio esencial deja de estar disponible.

Estas características permiten que el laboratorio responda rápidamente ante incidentes y mantenga la continuidad de los servicios, alineándose con las buenas prácticas recomendadas por ITIL.

#### **f) Gestión de infraestructura y plataforma**

En el documento de ITIL nos explican que este tema tiene como objetivo, *“Supervisar la infraestructura y las plataformas utilizadas por una organización. Cuando se realiza adecuadamente, esta práctica permite monitorear las soluciones tecnológicas disponibles para la organización, incluida la tecnología de proveedores de servicios externos.”* (AXELOS, 2019).

En esta práctica se incluye la planificación, implementación, mantenimiento y optimización de la infraestructura tecnológica que alojará los servicios de TI. Esto incluye tanto elementos físicos como virtuales, que son: servidor, hipervisores, VMs, CTs, etc.

En el laboratorio virtual este tema se refleja en:

- La utilización de Proxmox VE como plataforma de virtualización principal.
- La definición clara de recursos asignados a cada VM y CT.

Así mismo, se integra n8n como herramienta de operación automatizada que permite:

- Ejecutar tareas repetitivas de mantenimiento (limpieza de logs, reinicios programados).
- Monitorear el estado de la infraestructura.
- Automatizar acciones sobre VMs y/ CTs.

#### **g) Gestión del catálogo de servicios**

En este subcapítulo ITIL indica que se deben definir de manera clara y documentada los servicios que la organización ofrece, sus características, requisitos y dependencias. El objetivo de esto es asegurar que los usuarios y administradores comprendan qué servicios están disponibles y cómo se deben gestionar.

En el proyecto este tema se aplica documentando los distintos servicios desplegados dentro del laboratorio virtual como son:

- Servicios de red (DNS, DHCP).
- Servicios de monitoreo.
- Servicios Web.

- Servicios de base de datos y almacenamiento.
- Servicios de control de usuarios (Active directory).
- Flujos de automatización implementados con n8n.

Contar con este catálogo facilita la gestión operativa del entorno y también permite que los flujos automatizados puedan asociarse directamente a servicios específicos.

## **3.2. Marco metodológico para el análisis y selección de herramientas**

Para el diseño del laboratorio virtual fue necesario seleccionar un conjunto de herramientas que permitan cumplir con los objetivos de monitoreo, gestión automatizada y operación de servicios de TI en un entorno virtualizado. Con el fin de realizar una selección objetiva, estructurada y alineada con los requerimientos del proyecto, se definió un marco metodológico para el análisis y selección de herramientas.

Este proceso se desarrolla previo a la fase de implementación y tiene como propósito fundamentar las decisiones técnicas presentadas en este proyecto. Para la selección de herramientas se realizará un análisis en el que se consideren tanto aspectos técnicos como operativos y económicos, priorizando soluciones que puedan ser adoptadas por pequeñas y medianas empresas y que permitan su integración con procesos de automatización y autogestión.

### **3.2.1. Criterios de evaluación**

Se establecieron los siguientes criterios para evaluar las herramientas, estas permiten comparar de forma objetiva las opciones.

#### **3.2.1.1. Costos**

Este criterio evalúa si la herramienta es de código abierto o de pago ya sea suscripción o un solo pago, se priorizan opciones que minimicen la inversión inicial sin que se comprometa la funcionalidad.

#### **3.2.1.2. Escalabilidad**

Aquí se analiza la capacidad de la herramienta para crecer en función de los servicios, usuarios o carga de trabajo, debe garantizar un rendimiento estable mientras el laboratorio se expanda.

#### **3.2.1.3. Facilidad de uso**

Se debe considerar la complejidad de la instalación, configuración y la curva de aprendizaje de las herramientas, esto es importante ya que el proyecto está orientado a entornos donde el personal de TI es limitado.

#### **3.2.1.4. Compatibilidad**

Este criterio se utiliza para comprobar el nivel de integración que tienen las herramientas con entornos virtualizados, así como su compatibilidad con sistemas operativos linux y con los otros servicios que se conformarán el entorno virtual.

#### **3.2.1.5. Capacidad de automatización**

Analiza la posibilidad de automatizar tareas mediante APIs, generación de eventos e integración con herramientas de automatización como n8n, este criterio es indispensable para cumplir con el objetivo de autogestión del proyecto.

### **3.2.2. Método de análisis**

Para el análisis y selección de herramientas se utilizará el método AHP que según (Noé González & PrevenControl, 2019), *“Es un método de decisión multicriterio que nos ayuda a seleccionar entre distintas alternativas en función de una serie de criterios o variables de selección, normalmente jerarquizadas, y que suelen entrar en conflicto entre sí.”*

Este método nos permite tener una estructura con los objetivos y los criterios de evaluación a los cuales asignaremos pesos relativos según su importancia y se compararán con alternativas similares.

Dentro de este método existen pasos que seguiremos para aplicarlos en este proyecto como son:

- a) Definición del objetivo de decisión, que consiste en seleccionar las herramientas más adecuadas para el laboratorio virtual.
- b) Identificación de los criterios de evaluación establecidos en la sección 3.2.1.
- c) Asignación de pesos a cada criterio en función de su relevancia para los objetivos del proyecto.
- d) Comparación de las alternativas tecnológicas respecto a cada criterio.
- e) Obtención de una puntuación global que permite determinar la herramienta más adecuada.

*(Peterka, 2024)*

### **3.3. Metodología para el prototipado**

Para el desarrollo del laboratorio virtual propuesto en este proyecto de titulación se utiliza la metodología de prototipado evolutivo, el cual consiste en construir el sistema en iteraciones, permitiendo realizar ajustes continuos conforme se valida su funcionamiento. Esta metodología es adecuada ya que permite probar cada etapa del prototipo antes de avanzar a la siguiente. Este enfoque propicio construir el laboratorio de forma incremental, es decir, primera la infraestructura base VMs y CTs, posteriormente desplegar los servicios, luego las herramientas de monitoreo y finalizar con los componentes de automatización. Al trabajar con esta metodología se reducen riesgos y se detectan fallas antes de continuar con los siguientes pasos garantizando que cada fase funcione correctamente.

### **3.4. Estructura de la implementación**

#### **3.4.1. Identificación de objetivos y requerimientos**

La primera etapa del prototipado evolutivo consiste en identificar los requerimientos fundamentales para la construcción del laboratorio virtual. Aquí establecemos los elementos técnicos necesarios para el funcionamiento estable de la infraestructura y permite definir los recursos mínimos que aseguran el monitoreo, la operación de los servicios y la integración posterior de procesos de automatización.

##### **3.4.1.1. Hardware mínimo necesario**

El proyecto tiene como uno de sus objetivos optimizar los recursos del servidor o computador que se usará, esto implica aprovechar al máximo la capacidad del hardware. Con esto en mente y que tampoco se busca una gran inversión. El hardware mínimo que se recomienda son 16Gb de RAM, un procesador CPU de 4 núcleos, de preferencia multihilo ya que permite generar más VMs y CTs y un disco de preferencia un SSD de 512Gb.

Estos recursos los podemos encontrar en cualquier computador medio-bajo en la actualidad por lo que no es necesario inicialmente adquirir infraestructura nueva. Sin embargo, si la PYME considera necesario más capacidad ya que quieren tener varios backups, replicación, etc. Se lo puede incluir sin ningún problema, además la herramienta seleccionada proxmox, nos permite hacer un clúster que permite asegurar alta disponibilidad en nuestros servicios.

##### **3.4.1.2. Plataforma de virtualización y gestión**

En esta etapa se define el software base que controlará todo el hardware mencionado anteriormente. Se ha seleccionado Proxmox VE como el hipervisor principal debido a su capacidad para gestionar dos tecnologías diferentes en una misma interfaz:

**Máquinas Virtuales (KVM):** Para servicios que necesitan un sistema operativo completo y aislado, como el controlador de dominio o el almacenamiento masivo.

**Contenedores (LXC):** Para servicios ligeros que comparten el núcleo del sistema, permitiendo ahorrar memoria RAM y mejorar la velocidad de respuesta en herramientas como el monitoreo y la automatización.

### **3.4.1.3. Definición de servicios esenciales para la PYME**

Para que el prototipo sea representativo de una infraestructura empresarial real, se han seleccionado servicios que cubren las áreas operativas más comunes de una pequeña y mediana empresa. Estos servicios se han categorizado según su función dentro del laboratorio:

- **Infraestructura y Control:** Se define el uso de un controlador de dominio (Zentyal) para centralizar la gestión de usuarios y servicios de red básicos.
- **Productividad y Datos:** Se incluye el almacenamiento compartido y la colaboración mediante TrueNAS y NextCloud, asegurando que la información sea accesible y esté respaldada.
- **Soporte Técnico:** Se establece una mesa de ayuda con GLPI para organizar la atención a los incidentes de los usuarios.
- **Disponibilidad Web:** Se integran servidores Nginx y un balanceador de carga para demostrar cómo mantener la continuidad de la página institucional ante una alta demanda.
- **Gestión y Automatización:** Finalmente, se definen los pilares del proyecto: Zabbix para la vigilancia constante y n8n como el motor encargado de ejecutar las acciones correctivas automáticas.

## **CAPÍTULO 4: IMPLEMENTACIÓN DEL LABORATORIO VIRTUAL**

### **4. Implementación de un prototipo funcional para el monitoreo y gestión automatizada de servicios de TI.**

#### **4.1. Identificación de los servicios de tecnologías de la información (TI) necesarios para el funcionamiento de una PYME.**

Las pequeñas y medianas empresas (PYMES) dependen cada vez más de los servicios de tecnologías de TI para garantizar la continuidad de sus operaciones diarias, la seguridad de

la información y la eficiencia de sus procesos internos. A diferencia de las grandes organizaciones, las PYMES suelen contar con recursos limitados, por lo que requieren infraestructuras de TI que sean funcionales, escalables y fáciles de administrar, sin incurrir en altos costos de implementación o mantenimiento.

Con este contexto, la identificación de los principales servicios de TI es un parte fundamental para la implementación del laboratorio propuesto, ya que esto nos permite definir qué componentes deben ser desplegados, monitoreados y gestionados de manera automatizada dentro del entorno virtualizado. Se debe mencionar que estos servicios no son todos los que necesitan las PYMEs ya que esto puede variar dependiendo del tipo de empresa que sea, sin embargo, estos servicios representan una base mínima necesaria para el correcto funcionamiento tecnológico de una PYME y como se mencionó antes esta base puede ser ampliada según la necesidad de cada empresa.

A continuación, se describen los principales servicios de TI los cuales se consideran la base para el funcionamiento de una PYME y que forman parte del prototipo implementado en este proyecto de titulación.

#### **4.1.1. Servicios de red**

Los servicios de red son fundamentales para el flujo de la información interna de la empresa. Dentro de estos servicios tenemos dos muy importantes que son el sistema de nombres de dominio (**DNS**) y el protocolo de configuración dinámica de host (**DHCP**).

##### **A) DNS:**

Son servidores en los cuales se asocia una dirección IP con un nombre de dominio, según (Schneider, 2019), *“El Sistema de Nombres de Dominio (DNS) es un sistema de nombres jerárquico y descentralizado para ordenadores, servicios, etc., conectados a internet o a una red privada . El DNS funciona de forma similar a una guía telefónica: asigna nombres de dominio como [www.univention.de](http://www.univention.de) a direcciones IP numéricas (78.47.199.152) y viceversa. (...). Un servidor DNS en una red privada también es responsable de la resolución de nombres. Conoce todas las direcciones IP y nombres de los dispositivos.”*

##### **B) DHCP:**

El servicio de DHCP que tiene como función principal la asignación dinámica de direcciones IPs a los distintos hosts. Según (Schneider, 2019), *“El servidor DHCP distribuye direcciones IP libres de un conjunto específico o asigna direcciones estáticas a los clientes y los identifica mediante su dirección MAC (Control de Acceso al Medio, identificador único asignado a un controlador de*

*interfaz de red).*” En el mismo artículo se menciona otra funcionalidad importante del DHCP que es *“El servidor DHCP también determina la validez de una dirección IP. Si el tiempo de concesión expira mientras un cliente sigue activo, intenta renovarlo automáticamente. Los usuarios normalmente no notan este intercambio entre el servidor y el cliente.”* (Schneider, 2019)

#### **4.1.2. Servicio de almacenamiento**

En el contexto de una PYME la información fluye diariamente y todos los días se generan nuevos archivos, documentos, etc. Todos estos deben ser almacenados por seguridad y deben poder ser compartidos entre equipos de trabajo. En este caso según nextcloud nos menciona algo muy importante en relación con la seguridad de la información. *“En la era de Google Drive, Dropbox y compañía, almacenar tus datos en línea nunca ha sido tan fácil. Con las violaciones de privacidad reportadas a diario”*. Así mismo mencionan razones por las cuales es mejor tener un control absoluto sobre la información, sobre todo la privacidad. *“Tus datos pueden ser fácilmente objeto de abuso, igual que el poder. Siempre. Piensa en un político local enfadado porque descubriste su fraude. Un policía al que le gusta tu novio. O un empresario con buenos contactos que busca comprar tu tienda a bajo precio.”* (Self-Hosted Cloud Collaboration Platform for Home Users - Nextcloud, 2025)

Otro aspecto fundamental en este proyecto es que se requiere de un almacenamiento centralizado para poder escalar los demás servicios, para ellos se utilizará la herramienta de TrueNas, la cual ofrece la capacidad de configurarse como un disco compartido en el entorno virtual.

#### **4.1.3. Servicio de soporte de TI – Mesa de ayuda (help desk)**

Otro de los servicios necesarios para las operaciones diarias es la implementación de un servicio de soporte de TI mediante una mesa de ayuda (Help Desk), el cual actúa como un canal formal y centralizado de comunicación entre los usuarios y el departamento de tecnologías de la información. Diariamente se presenta problemas en equipos, sistemas, configuraciones o simplemente desconocimiento del usuario, para solucionar esto es necesario una forma organizada de comunicación con el departamento de tecnologías. Cuando estos problemas no son gestionados mediante un proceso estructurado, suelen resolverse de manera informal, lo que dificulta su seguimiento, priorización y documentación, generando retrasos en la atención y afectando la productividad de la organización.

Dentro del laboratorio propuesto en este proyecto, este servicio es un componente estratégico y que podría integrarse con las herramientas de monitoreo y gestión. Detectando

eventos importantes y generando tickets de forma automática, lo que permite una respuesta rápida y eficiente ante estas fallas.

#### **4.1.4. Servicios web**

Este servicio nos permite alojar aplicaciones internas o sistemas de gestión como de inventarios que utiliza la empresa en sus operaciones diarias, la disponibilidad de este servicio es esencial para garantizar el acceso continuo a dichas aplicaciones.

Según IBM, *“Los servicios web son aplicaciones web que permiten aumentar la flexibilidad de los procesos empresariales al integrarse con aplicaciones que de otra forma no se comunicarían. El programa de préstamo de biblioteca interna de la biblioteca local es un buen ejemplo del concepto de los servicios web y su evolución.” (WebSphere Application Server, 2025)*

#### **4.1.5. Servicio de base de datos**

El servicio de base de datos es uno de los componentes más importantes dentro de la infraestructura de TI de una PYME, ya que es el encargado de almacenar, organizar y gestionar toda la información crítica que utilizan los distintos sistemas de la empresa. En este servicio se guarda información como datos de clientes, proveedores, usuarios, inventarios, registros de ventas, credenciales de acceso y configuraciones de los sistemas.

En la mayoría de las empresas, los servicios web, sistemas administrativos, aplicaciones internas y herramientas de monitoreo dependen directamente de una base de datos para funcionar correctamente. Por esta razón, la disponibilidad y estabilidad del servicio de base de datos es clave para garantizar la continuidad operativa de la organización.

Según Oracle, *“Una base de datos es una colección organizada de información estructurada, normalmente almacenada de forma electrónica en un sistema informático. Una base de datos está controlada por un sistema de gestión de bases de datos (DBMS), que permite crear, leer, actualizar y eliminar datos de forma eficiente” (Oracle, 2024).*

Dentro del laboratorio virtual propuesto en este proyecto, el servicio de base de datos se implementa de manera aislada en una máquina virtual o contenedor dedicado. Esta separación permite mejorar la seguridad, el rendimiento y la facilidad de administración, ya que evita que otros servicios interfieran directamente con la base de datos. Además, al estar virtualizado, se

facilita la realización de copias de seguridad (backups), restauraciones rápidas y monitoreo del uso de recursos como CPU, memoria y almacenamiento.

#### **4.1.6. Servicio de gestión de usuarios**

El servicio de gestión de usuarios es fundamental dentro de la infraestructura de TI de una PYME, ya que permite administrar de forma centralizada las cuentas de los usuarios, sus credenciales y los permisos de acceso a los distintos servicios y recursos del sistema. Este servicio ayuda a mantener el orden, la seguridad y el control dentro del entorno tecnológico de la empresa.

En muchas organizaciones este aspecto suele no considerarse ya que se piensa es algo que no es importante cuando la organización es pequeña, sin embargo, puede provocar problemas como accesos no autorizados, uso indebido de información o dificultad para revocar permisos cuando un empleado deja la empresa. Por esta razón, contar con un servicio centralizado de autenticación es una buena práctica recomendada.

Según Microsoft, *“El departamento de TI de la organización necesita una forma de controlar a qué pueden acceder los usuarios y a qué no, para que las funciones y los datos confidenciales estén restringidos solo a las personas y los dispositivos que necesiten trabajar con ellos.”* (*¿Qué Es La Administración de Identidad Y Acceso (IAM)? | Seguridad de Microsoft, 2025*). Esto significa que mediante un servicio de gestión de usuarios se pueden definir roles, políticas de acceso y niveles de permisos según el cargo o función del usuario.

#### **4.1.7. Servicio de monitoreo**

El monitoreo es uno de los principales servicios de este proyecto, ya que permite supervisar en tiempo real el estado, rendimiento y disponibilidad de los servicios de TI. Su objetivo principal es detectar fallos, sobrecargas o comportamientos anómalos antes de que afecten gravemente a las operaciones de la empresa.

Para las PYMEs, el monitoreo automatizado es especialmente importante porque, en la mayoría de los casos, no cuentan con personal de TI dedicado a revisar constantemente el estado de los servidores. Sin un sistema de monitoreo, los problemas suelen detectarse solo cuando el servicio ya dejó de funcionar.

Para IBM el monitoreo consiste en *“El proceso de seguimiento, análisis y gestión del rendimiento, la disponibilidad y el estado de los componentes de backend de la pila tecnológica de una empresa.”* (IBM, 2023).

En este laboratorio virtual el servicio de monitoreo se encargará de:

- Supervisar el uso de CPU, memoria y almacenamiento.
- Verificar la disponibilidad de servicios como DNS, web y base de datos.

- Detectar caídas o fallos de máquinas virtuales y contenedores.
- Generar alertas automáticas cuando se superan ciertos umbrales.

Esta información servirá para reaccionar ante fallos y para planificar el crecimiento de la infraestructura, optimizar recursos y mejorar la estabilidad de los servicios.

#### **4.1.8. Servicio de gestión y automatización**

La gestión y automatización de tareas administrativas permiten reducir la intervención manual en la administración de la infraestructura de TI automatizando tareas repetitivas y respuestas ante eventos específicos, como reinicios de servicios, notificaciones de fallos o ejecución de respaldos, optimizando así la operación del entorno de TI con un menor esfuerzo humano.

Para redhat la automatización permite que *“Los equipos de TI pueden automatizar los procesos complejos para aumentar la eficiencia, la productividad y la flexibilidad, a la vez que reducen los costos y los errores humanos.”* (La Automatización: Qué Es Y Sus Ventajas | Red Hat, 2022).

En el laboratorio virtual propuesto, este servicio se integra con las herramientas de monitoreo para ejecutar acciones automáticas cuando ocurre un evento, por ejemplo:

- Reiniciar un servicio cuando se detecta una caída.
- Enviar notificaciones automáticas por correo o mensajes.
- Ejecutar scripts de mantenimiento.
- Registrar eventos y acciones realizadas de forma automática.

#### **4.2. Diseño técnico de la arquitectura del laboratorio virtual**

El diseño de la arquitectura del laboratorio virtual es una parte fundamental dentro del desarrollo del prototipo propuesto, ya que define cómo se organizan los componentes, cómo se distribuyen los servicios y de qué manera interactúan entre sí dentro del entorno virtualizado. Este diseño busca garantizar un funcionamiento estable, seguro y fácil de administrar, considerando las limitaciones de recursos que suelen tener las pequeñas y medianas empresas.

#### **4.3. Comparación y selección de herramientas**

##### **4.3.1. Selección hipervisor**

Para una PYME, no solo se busca que el programa sea potente, sino que no sea caro y que no consuma todos los recursos del servidor. En esta tabla comparamos tres opciones conocidas para ver cuál nos da más funciones (como copias de seguridad y manejo de contenedores) sin tener que pagar licencias costosas.

#### 4.3.1.1. Criterio y pesos

- **Costos (35%):** Crítico para una PYME.
- **Soporte de Contenedores LXC (25%):** Esencial para optimizar los recursos del servidor.
- **Facilidad de Gestión (20%):** Interfaz web unificada.
- **Funcionalidades Gratuitas (20%):** Backups, Cluster, HA sin pagar licencias.

#### 4.3.1.2. Matriz AHP - Hipervisor

Criterios	Peso (%)	Proxmox VE		VMware ESXi		MS Hyper- V	
Calificación	Total	Calificación	Total	Calificación	Total		
<b>Costos / Licenciamiento</b>	35%	5	1.75	1	0.35	3	1.05
<b>Eficiencia (Soporte LXC)</b>	25%	5	1.25	2	0.50	2	0.50
<b>Facilidad de Gestión</b>	20%	4	0.80	5	1.00	4	0.80
<b>Funcionalidades Free</b>	20%	5	1.00	2	0.40	3	0.60
<b>PUNTUACIÓN FINAL</b>	100%		4.80		2.25		2.95

*Tabla 1 Matriz AHP – Hipervisor*

#### 4.3.2. Selección herramienta de monitoreo

El éxito del monitoreo depende de la capacidad de la herramienta para centralizar métricas de diversas fuentes sin elevar excesivamente la complejidad administrativa. El criterio principal es obtener una solución que integre alertas y visualización en una sola plataforma, facilitando la detección temprana de anomalías que posteriormente serán gestionadas de forma automatizada.

##### 4.3.2.1. Criterios y peso

- **Centralización (30%):** Capacidad de ver todo en un solo lugar sin instalar plugins externos.
- **Curva de Aprendizaje (25%):** Facilidad para implementar en corto tiempo.
- **Capacidad de Alerta (25%):** Flexibilidad para enviar datos a n8n.
- **Consumo de Recursos (20%):** Impacto en el servidor.

#### 4.3.2.2. Matriz AHP – Sistema de monitoreo

Criterios	Peso (%)	Zabbix		Prometheus		Nagios Core	
Calificación	Total	Calificación	Total	Calificación	Total		
Solución "Todo en Uno"	30%	5	1.50	3	0.90	2	0.60
Curva de Aprendizaje	25%	4	1.00	2	0.50	2	0.50
Integración (Alertas/API)	25%	5	1.25	5	1.25	3	0.75
Consumo de Recursos	20%	4	0.80	4	0.80	5	1.00
<b>PUNTUACIÓN FINAL</b>	100%		4.55		3.45		2.85

*Tabla 2 Matriz AHP - Sistema de monitoreo*

#### 4.3.3. Selección herramienta automatización

El componente de gestión automatizada es el diferenciador de este prototipo, ya que permite la transición de un monitoreo pasivo a una gestión activa. Se comparan herramientas capaces de orquestar flujos de trabajo basados en eventos, priorizando aquellas que ofrezcan una interfaz visual para facilitar el mantenimiento y que operen localmente (self-hosted) para garantizar la privacidad de los datos institucionales y reducir la dependencia de servicios externos en la nube.

##### 4.3.3.1. Criterios y pesos

- **Capacidad Self-Hosted (30%):** Instalarse localmente (privacidad y velocidad).
- **Interfaz Visual (30%):** Facilidad para diseñar flujos lógicos (Low-code).
- **Gestión de Eventos (20%):** Capacidad de reaccionar a Webhooks al instante.
- **Costos Operativos (20%):** Suscripciones mensuales.

##### 4.3.3.2. Matriz AHP – Automatización

Criterios	Peso (%)	n8n		Zapier		Ansible	
-----------	----------	-----	--	--------	--	---------	--

Calificación	Total	Calificación	Total	Calificación	Total		
<b>Instalación Local (Privacidad)</b>	30%	5	1.50	1	0.30	5	1.50
<b>Interfaz Visual (Low-Code)</b>	30%	5	1.50	5	1.50	2	0.60
<b>Gestión de Eventos</b>	20%	5	1.00	5	1.00	3	0.60
<b>Costos Operativos</b>	20%	5	1.00	1	0.20	5	1.00
<b>PUNTUACIÓN FINAL</b>	100%		5.00		3.00		3.70

*Tabla 3 Matriz AHP - Automatización*

La arquitectura propuesta se basa en los resultados obtenidos de la tabla comparativa anterior por lo que se realizará con en el uso de virtualización mediante Proxmox VE, utilizando máquinas virtuales (VMs) y contenedores (CTs) para alojar los distintos servicios de TI. Esta separación permite un mejor control de los recursos, facilita el monitoreo y mejora la disponibilidad de los servicios.

#### **4.3.4. Descripción general de la arquitectura**

La arquitectura del laboratorio virtual está diseñada bajo un enfoque centralizado, donde un servidor físico actúa como host principal ejecutando el hipervisor Proxmox VE, aunque se debe mencionar que el hipervisor soporta clustering con varios nodos para asegurar alta disponibilidad. Sobre este hipervisor se despliegan máquinas virtuales y contenedores que alojan los servicios esenciales de TI requeridos por una PYME.

El diseño prioriza la separación de servicios, donde cada servicio de TI se despliega de forma independiente en una máquina virtual o contenedor, evitando la concentración de múltiples servicios críticos en un solo sistema, lo que reduce riesgos ante fallos y facilita la administración. La estructura general del laboratorio está pensada para integrarse con herramientas de monitoreo y automatización, permitiendo una gestión proactiva del entorno y reduciendo la necesidad de intervención manual.

El diseño también permite que el laboratorio sea escalable, ya que se pueden agregar nuevos servicios o aumentar recursos sin afectar la estructura general del entorno.

#### **4.3.5. Componentes de la arquitectura**

La arquitectura del laboratorio virtual está compuesta por los siguientes elementos principales:

- Proxmox VE como plataforma de virtualización y administración centralizada.

- Máquinas virtuales destinadas a servicios críticos de la infraestructura.
- Contenedores utilizados para servicios ligeros o de apoyo.
- Una red virtual que permite la comunicación entre todos los componentes.
- Herramientas de monitoreo y automatización integradas al entorno.

Estos componentes trabajan de forma conjunta para garantizar el correcto funcionamiento del laboratorio y facilitar su administración.

#### 4.3.6. Diseño de máquinas virtuales y contenedores

El laboratorio virtual utiliza una combinación de máquinas virtuales VMs y contenedores CTs con el fin de optimizar el uso de los recursos del servidor físico y facilitar la administración de los servicios. Las máquinas virtuales se emplean para alojar servicios que manejan información crítica o que requieren mayor aislamiento, mientras que los contenedores se utilizan para servicios más ligeros que no necesitan un sistema operativo completo.

Esta combinación permite que el entorno sea flexible, eficiente y fácil de escalar, lo que sea alinea con el enfoque de monitoreo y automatización planteado en este proyecto.

De forma general, la distribución de servicios dentro del laboratorio es la siguiente:

- **VM para Zentyal**, encargada de la gestión de red, administración de usuarios y servicios básicos como DNS, DHCP y correo electrónico. Se implementa como máquina virtual debido a su rol central dentro del laboratorio.
- **VM para la gestión de TI (Help Desk)**, destinada al registro y seguimiento de incidentes, solicitudes y soporte técnico, permitiendo una mejor administración de los servicios del laboratorio y control interno de la PYME.
- **VM para TrueNAS**, utilizada como solución de almacenamiento compartido para los usuarios y para las máquinas virtuales o contenedores. Este servicio permite centralizar la información y facilita procesos como la replicación y el respaldo de datos.
- **CT con Zabbix**, junto con el uso de Zabbix Agent en los servicios monitoreados, encargado de supervisar el estado de las máquinas virtuales y contenedores, así como de recolectar métricas que serán utilizadas para la automatización y visualización en una página web.
- **CT con n8n**, utilizado para la automatización de procesos, gestión de alertas y ejecución de acciones automáticas sobre las máquinas virtuales o contenedores, en respuesta a eventos detectados por el sistema de monitoreo.

- **CT con base de datos**, destinado al almacenamiento de la información generada por los distintos servicios del laboratorio, como el sistema de monitoreo, la aplicación web y el sistema de gestión de TI.
- **CT con servicio web**, utilizado para la visualización del estado del laboratorio, métricas de monitoreo y resultados de los procesos de automatización.
- **CT con Nginx Proxy Manager**, actuando como balanceador de carga y gestor de accesos, permitiendo la replicación de servicios web sin afectar la experiencia del usuario final y facilitando la escalabilidad del laboratorio.

#### **4.3.7. Gestión de la red del laboratorio virtual**

El diseño de la red del laboratorio virtual tiene como objetivo garantizar la correcta comunicación entre las máquinas virtuales y contenedores, asegurar el acceso controlado a los servicios y facilitar el monitoreo y la automatización del entorno. La red se estructura de manera simple, pero funcional, permitiendo un entorno estable y fácil de administrar.

##### **a) Gestión de red y usuarios**

La máquina virtual de Zentyal actúa como el servidor principal de red, encargándose de servicios como la administración de usuarios, DNS y DHCP. Esto permite que las demás máquinas virtuales y contenedores obtengan configuración de red de forma automática y centralizada.

Además, la gestión de usuarios desde un único punto simplifica el acceso a los distintos servicios del laboratorio y mejora el control administrativo del entorno.

##### **b) Monitoreo y automatización en la red**

El sistema de monitoreo, implementado con Zabbix, se comunica con las máquinas virtuales y contenedores a través de la red del laboratorio, recolectando métricas sobre disponibilidad, uso de recursos y estado de los servicios.

Estas métricas son utilizadas por la herramienta de automatización n8n, la cual recibe alertas y eventos para ejecutar acciones automáticas, como notificaciones o tareas de gestión, contribuyendo a un entorno autónomo y controlado.

En conjunto, el diseño de la red del laboratorio virtual permite una comunicación eficiente entre los servicios, facilita el monitoreo continuo y soporta los procesos de automatización definidos en el proyecto.

### 4.3.8. Diagrama general de la arquitectura



*Ilustración 1 Diagrama general de la arquitectura*

## 4.4. Implementación del laboratorio virtual

### 4.4.1. Infraestructura física del laboratorio

El laboratorio fue implementado en un mini pc, específicamente un Intel NUC con modelo NUC8i7HVK, el cual es la base para la ejecución de las máquinas virtuales y contenedores definidos en la arquitectura del proyecto. Este equipo es de tamaño compacto, ofrece buen rendimiento y bajo consumo energético, esto lo hace adecuado para entornos de prueba y empresas pequeñas. Sin embargo, este puede ser reemplazado o ajustado según las necesidades y presupuestos de cada caso.

El servidor físico es el encargado de alojar la plataforma de virtualización Proxmox VE, sobre la cual se despliegan los distintos servicios del laboratorio virtual, permitiendo simular un entorno real de infraestructura de TI.

En la siguiente tabla se presentan las características técnicas de los componentes del servidor físico.

*Tabla 4 Características del servidor*

Componente	Cantidad	Especificación
Equipo	1	NUC8i7HVK
Procesador	1	Intel® Core™ i7-8809G
Memoria RAM	2	8 GB – 2400mhz
Almacenamiento total	2	500 GB + 2TB
Interfaz de red	2	Ethernet Gigabit

*Tabla 5 Características del procesador*

Característica	Detalle
Modelo	Intel® Core™ i7-8809G

Núcleos	4 núcleos físicos
Hilos	8 hilos
Frecuencia base	3.10 GHz
Arquitectura	64 bits

Esta configuración permite ejecutar varias máquinas virtuales y contenedores de forma simultánea, distribuyendo la carga de trabajo entre los distintos núcleos del procesador.

**Tabla 6 Características de la memoria RAM**

Característica	Detalle
Capacidad total	16 GB
Configuración	2 x 8 GB
Tipo	DDR4
Modo de operación	Doble canal

La memoria RAM disponible es suficiente para la ejecución concurrente de los servicios definidos en el laboratorio, permitiendo un desempeño estable durante las tareas de monitoreo y automatización.

**Tabla 7 Configuración de almacenamiento**

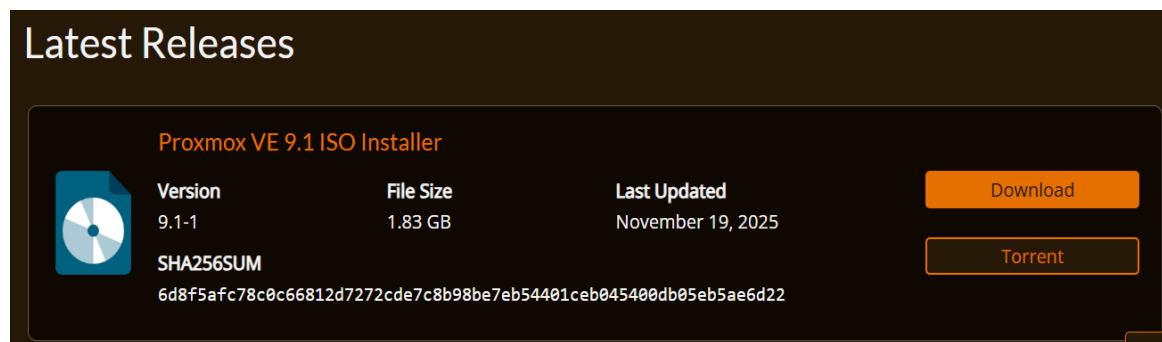
Unidad	Capacidad	Uso
Disco 1	500 GB	Sistema de virtualización, templates, isos, etc.
Disco 2	2 TB	Discos de VMs y CTs

Esta distribución permite separar el sistema operativo de los datos, facilitando la administración y mejorando la organización del entorno virtualizado. El almacenamiento será asignado a la VM de TrueNAS la cual será la encargada de proporcionar el almacenamiento a los demás elementos del laboratorio virtual.

#### 4.4.2. Instalación hipervisor

##### a) Descarga de la ISO

En el sitio oficial de proxmox <https://www.proxmox.com/en/downloads> podremos descargar la ISO correspondiente.

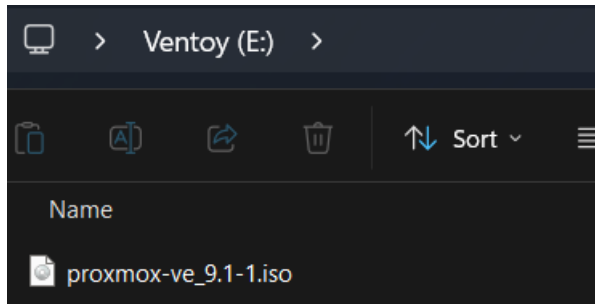


**Figura 1 ISO ProxmoxVE**

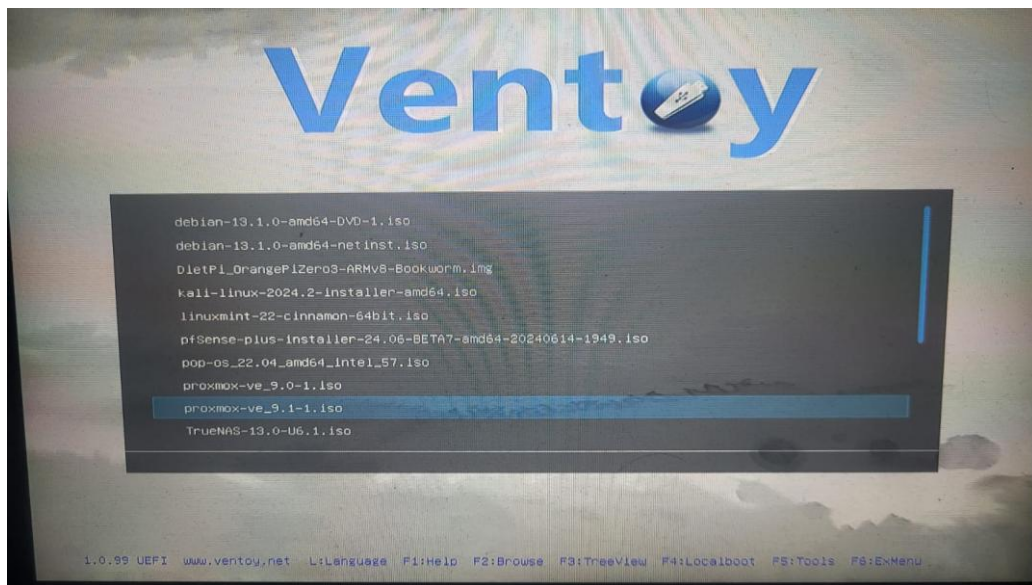
## b) Configurar como medio booteable

Para este laboratorio se utilizará la herramienta ventoy, la cual permite contener varias ISO en una memoria USB. <https://www.ventoy.net/en/index.html>

Se configura una memoria usb con el programa y después cargan las ISO en la carpeta de nombre ventoy.



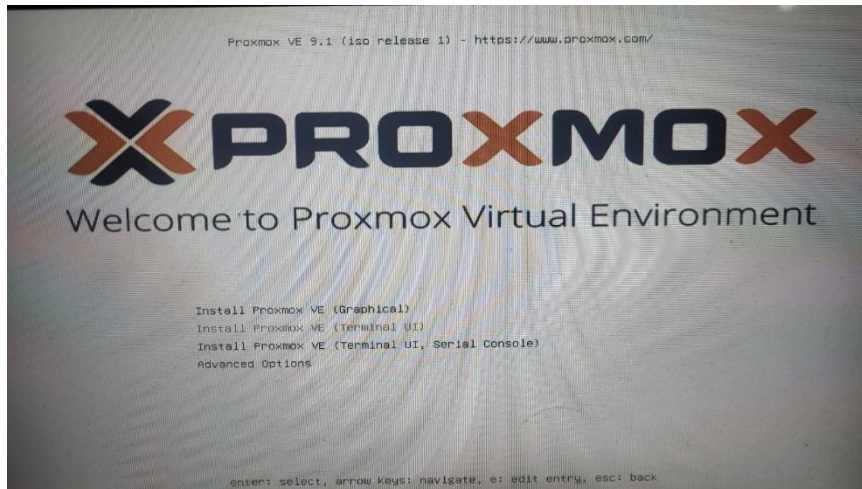
Después ingresamos al menú de boot de nuestro servidor, se abrirá ventoy con las ISO de nuestra usb, seleccionamos Ventoy y seguimos con el proceso de instalación.



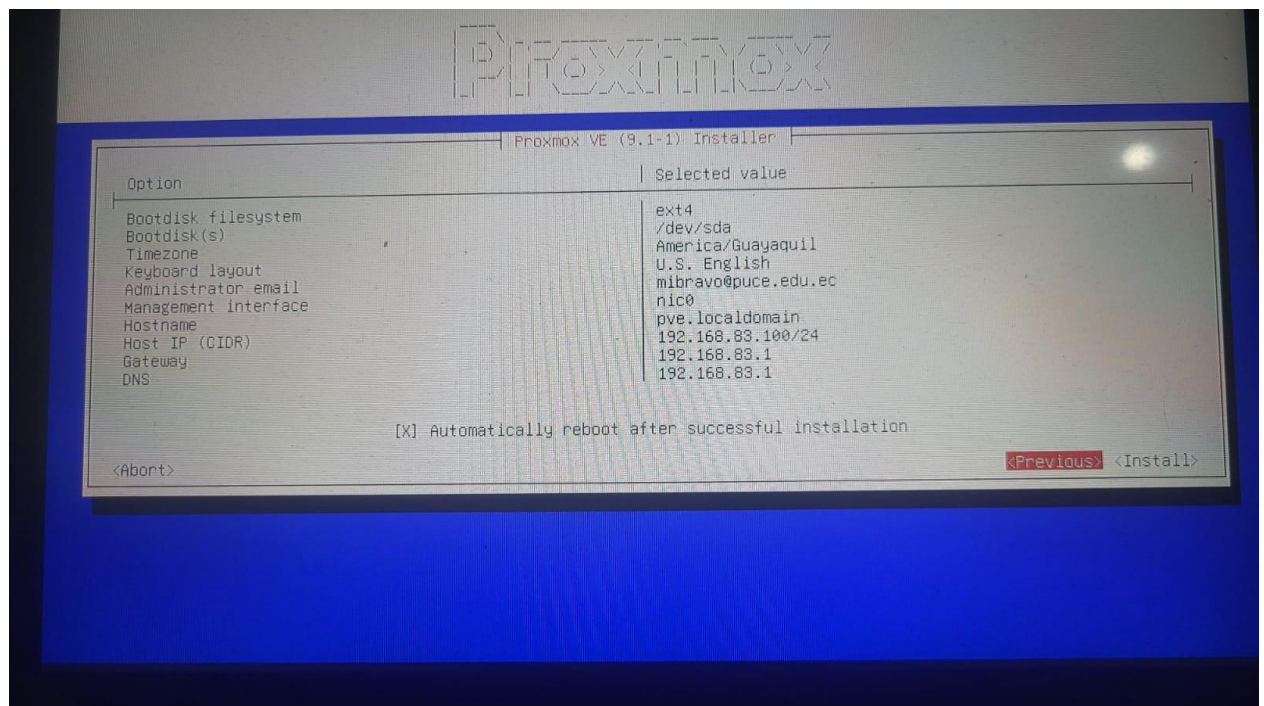
*Ilustración 2 Ventoy*

## c) Instalación del hipervisor

El proceso de instalación es similar a cualquier sistema operativo de Linux, simplemente se deberá llenar la información solicitada y avanzar.



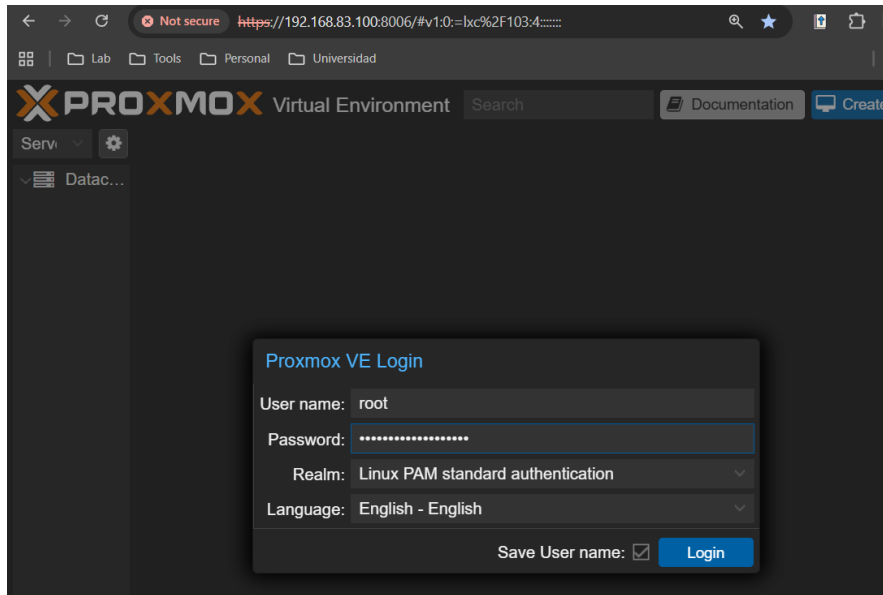
Una vez completa toda la información podremos ver un menú con el resumen de la información ingresada.



*Ilustración 3 Instalación proxmox*

#### **d) Ingreso**

Nos dirigimos a la IP que nos indica la terminal de proxmox después de la instalación e ingresamos nuestras credenciales.

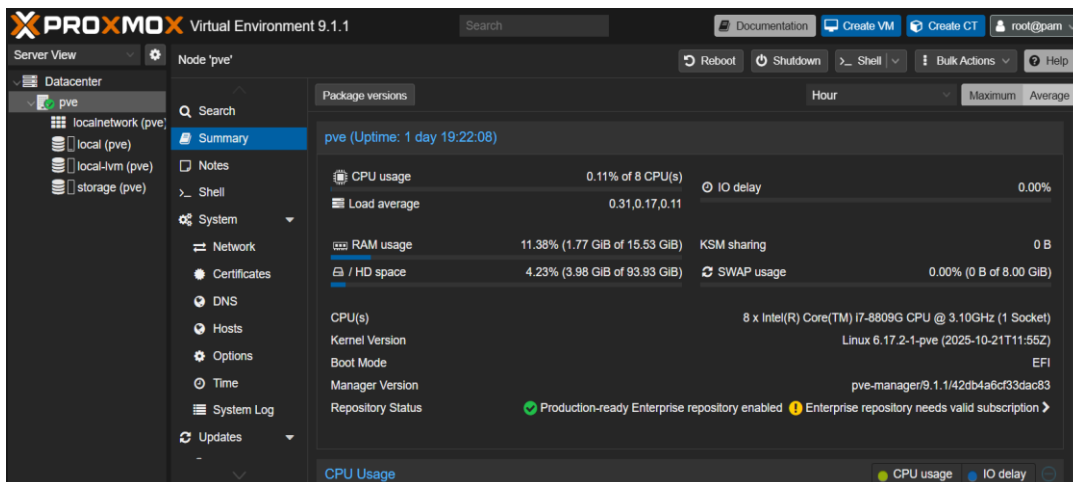


*Ilustración 4 Primer ingreso Proxmox*

#### e) Revisión de componentes

Después de la instalación es recomendable revisar que todos los componentes sean detectados por el hipervisor.

Para esto se verificaron los componentes a través de la interfaz de proxmox, en este caso tiene el nombre pve, después a la opción summary y veremos nuestros componentes detectados por el hipervisor.



*Ilustración 5 Revisión componentes*

### 4.4.3. Configuración inicial proxmox

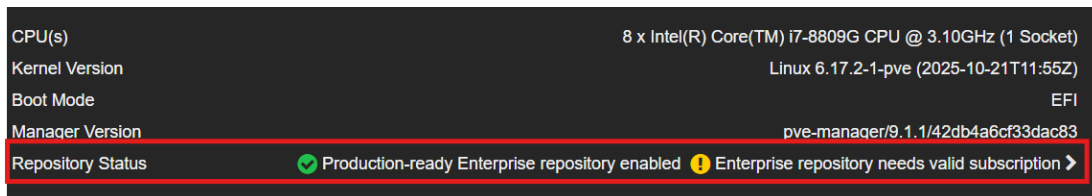
Después de instalar el hipervisor podremos visualizar que se presentan varios errores al tratar de actualizar los repositorios. Para esto utilizaremos un script de la comunidad de proxmox utilizado después de la instalación que hará lo siguiente “*Este script proporciona opciones para administrar los repositorios de Proxmox VE, incluida la desactivación del repositorio Enterprise, la adición o corrección de fuentes PVE, la activación del repositorio sin suscripción, la adición del repositorio de prueba, la desactivación del aviso de suscripción, la actualización de Proxmox VE y el reinicio del sistema.*” (Bram Suurd & Community Scripts, 2025).

Esto nos ayudará con los repositorios, cambiar el ambiente de producción, pasar a modo sin suscripción, etc. Con eso no tendremos problemas en el futuro en actualizaciones. El script está en el siguiente enlace: <https://community-scripts.github.io/ProxmoxVE/scripts?id=post-pve-install>



*Ilustración 6 Script Post install*

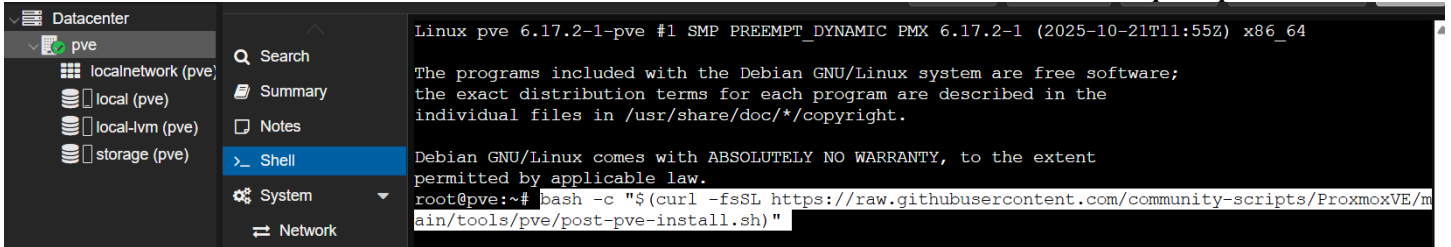
El estado lo podremos revisar en la consola de proxmox donde veremos el mensaje indicando que se están utilizando los repositorios Enterprise, que no utilizaremos por lo que se deberán cambiar con el script mencionado.



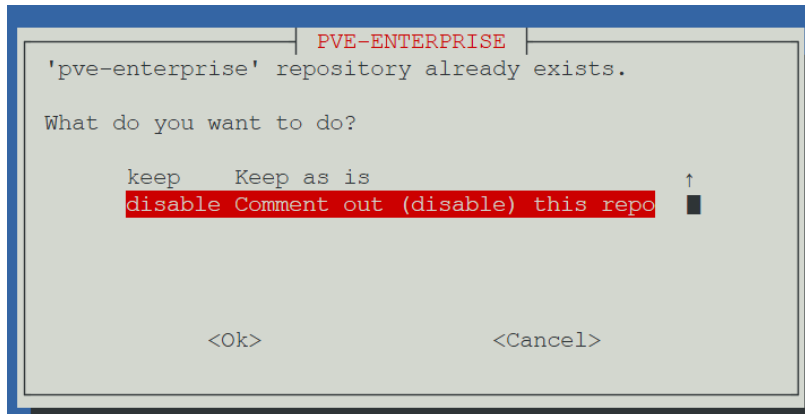
*Ilustración 7 Repositorio enterprise proxmox*

Para reemplazarlos por los repositorios de prueba que serán más que suficientes se ejecutó el siguiente comando en la consola de comandos del nodo de proxmox.

*Ilustración 8 Script en proxmox*

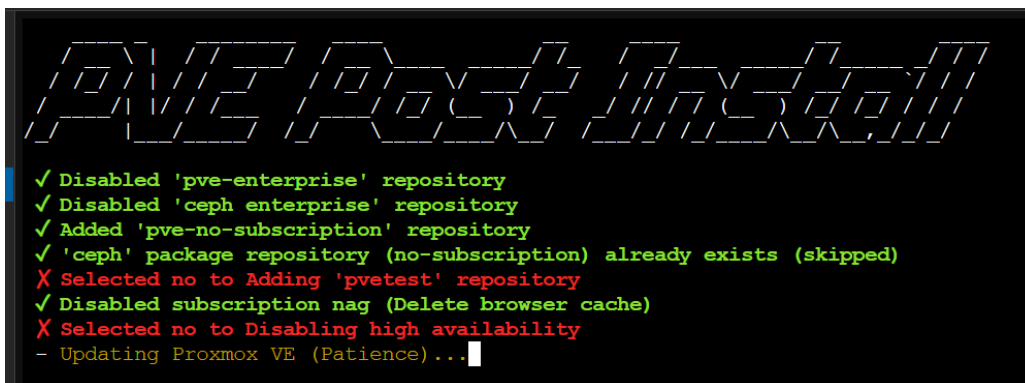


Durante la script se nos solicitará confirmar algunas cosas como las mencionadas anteriormente, deshabilitar repositorios Enterprise, etc.



*Ilustración 9 Deshabilitar repositorio enterprise*

Después se aplicarán nuestros cambios y veremos un resumen de nuestra configuración.



*Ilustración 10  
Resumen Post  
Install*

#### 4.4.4. Creación de máquinas virtuales

La creación de las VMs, dentro de la implementación del laboratorio virtual es una de las fases más importantes, ya que en ellas se alojan los servicios críticos de la infraestructura

de TI. Las máquinas virtuales permiten ejecutar sistemas operativos completos de forma aislada, brindando mayor seguridad, estabilidad y control de recursos.

En este proyecto se decidió utilizar máquinas virtuales para aquellos servicios que cumplen un rol central dentro del laboratorio y que manejan información sensible o requieren una mayor separación del resto de servicios.

Las máquinas virtuales fueron utilizadas considerando los siguientes criterios:

- Servicios que requieren un sistema operativo completo.
- Servicios con alto nivel de dependencia dentro del laboratorio.
- Necesidad de mayor aislamiento y control de recursos.
- Facilidad para realizar respaldos y restauraciones mediante snapshots.

Bajo estos criterios, se implementaron máquinas virtuales para los servicios de red, almacenamiento y gestión de soporte de TI.

En la siguiente tabla se detallan las características de máquina virtual.

**Tabla 8 Detalle VMs**

VM	Servicio	OS	vCPU	RAM	ALMACENAMIENTO
VM-Zentyal	Gestión de red y usuarios	Ubuntu Server22.04 - zentyal	2	4GB	80GB
VM - TrueNAS	Almacenamiento VMs - CTs	TrueNAS OS	2	4GB	1.2TB
VM-NextCloud	Almacenamiento Para usuarios	Ubuntu Server	2	4GB	500GB
VM-HelpDesk	Mesa de ayuda y soporte de TI	Debian	2	4GB	60GB

Los recursos fueron asignados considerando el hardware disponible y el consumo estimado de cada servicio, priorizando la estabilidad del sistema y el uso eficiente de los recursos del servidor físico. Sin embargo, uno de los beneficios de tener los servicios virtualizados es que si vemos que el consumo de CPU o RAM es menor o mayor podemos aumentar estos recursos asignados a cada VM.

Los recursos que utilizaremos son las ISOs de cada VM que después las subiremos a proxmox.

Para la creación de las VMs el proceso es similar por lo que se utilizará como ejemplo la creación de la VM de Zentyal.

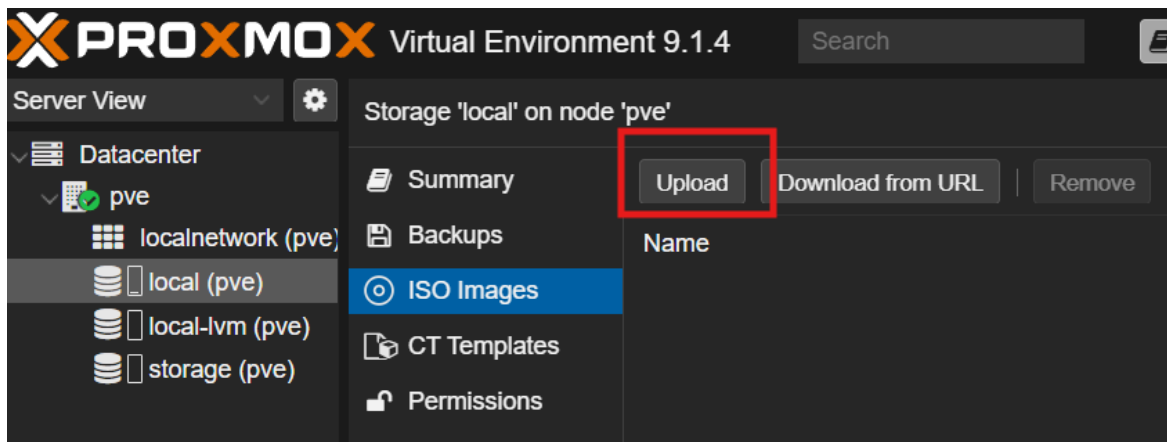
#### **4.4.4.1. Creación VM Zentyal**

Previamente, se descargó la imagen ISO desde el sitio oficial de zentyal, la edición de desarrollo o community. <https://www.zentyal.com/community/>



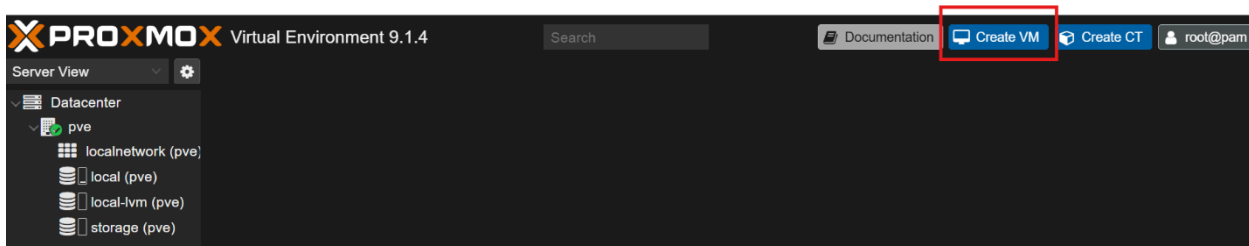
*Ilustración 11 ISO zentyal*

Una vez descargada debemos subir la ISO a proxmox. Para esto nos dirigimos a nuestro disco de proxmox, a la opción ISO images y al presionar el botón upload seleccionamos nuestro ISO.



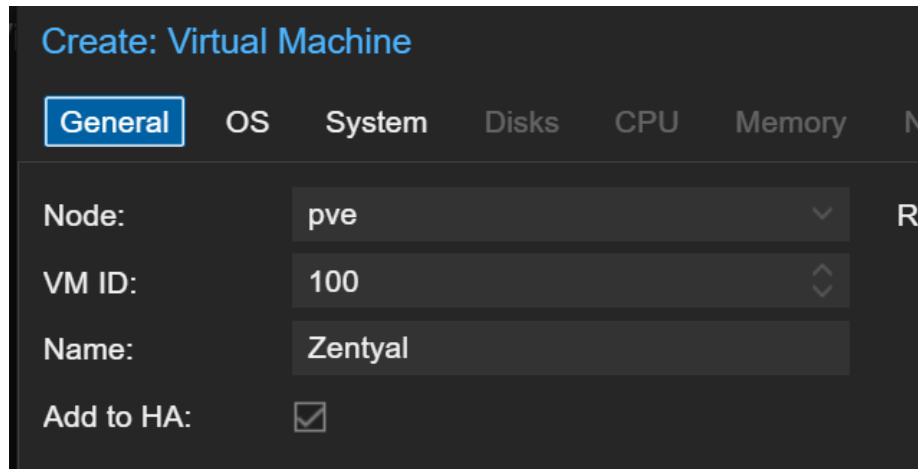
*Ilustración 12 Carga de ISOs en proxmox*

Con la ISO cargada en proxmox procedemos a la creación de la VM de Zentyal, en proxmox damos click en créate VM. Y llenamos la información solicitada.



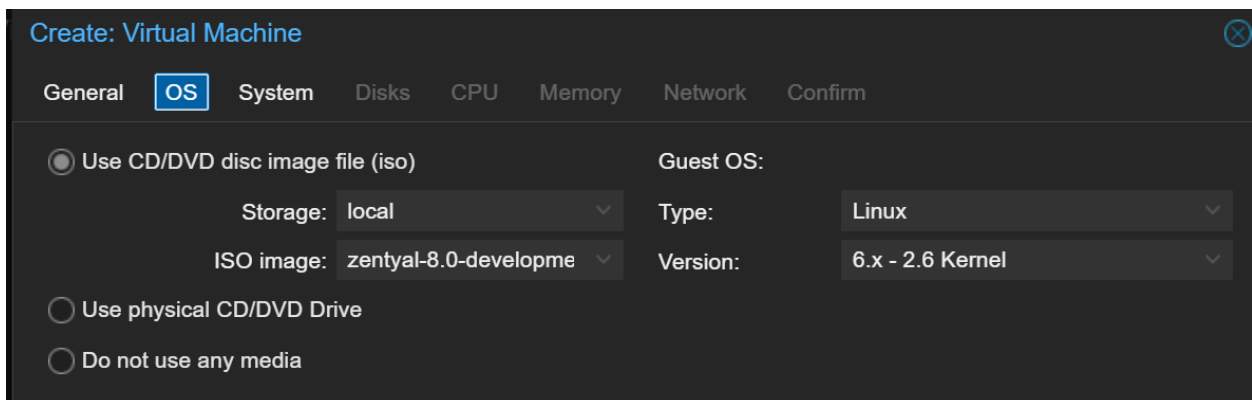
*Ilustración 13 Creación de VM*

Le damos un nombre y lo podemos agregar a HA que es la alta disponibilidad de proxmox.



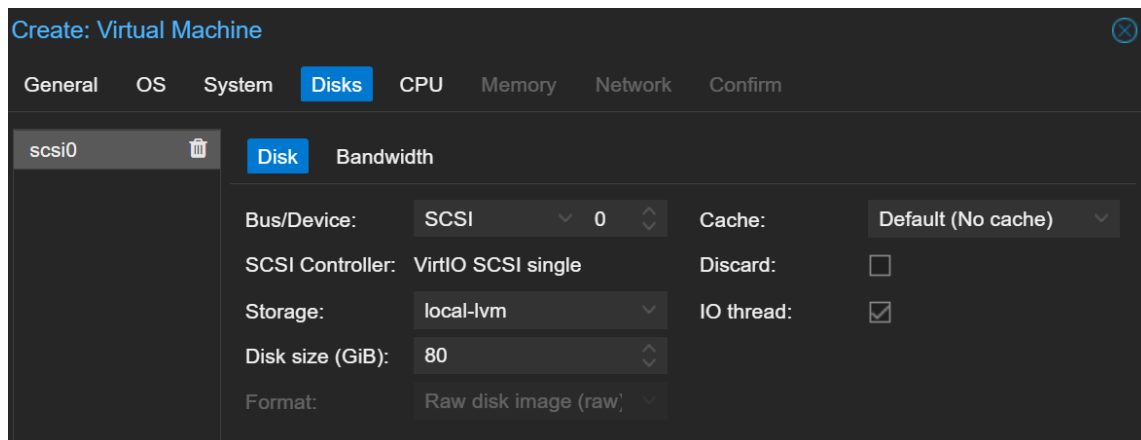
*Ilustración 14 Asignar nombre a VM*

Seleccionamos la ISO de zentyal que cargamos previamente.



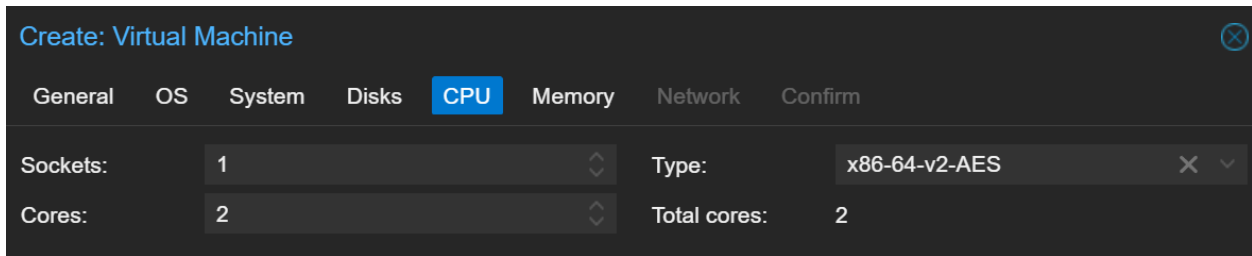
*Ilustración 15 Selección ISO para VM*

Asignamos la cantidad de GB de memoria de almacenamiento.



### Ilustración 16 Asignación Almacenamiento

Asignamos los vCPUs.



Create: Virtual Machine

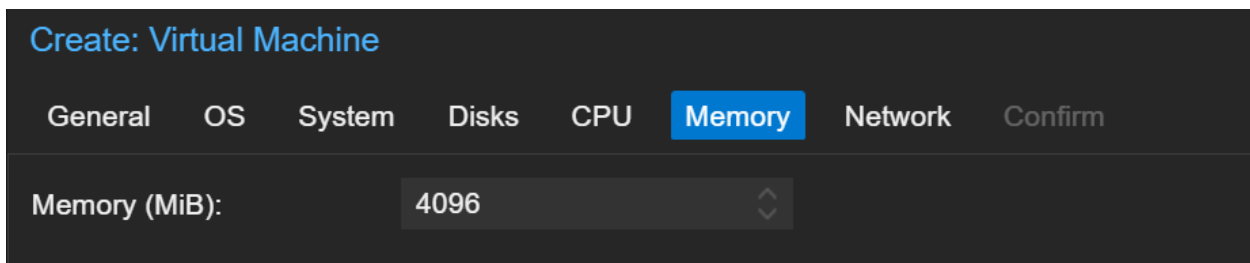
General OS System Disks **CPU** Memory Network Confirm

Sockets: 1 Type: x86-64-v2-AES

Cores: 2 Total cores: 2

Ilustración 17 Asignación vCPUs

Asignamos la memoria RAM.



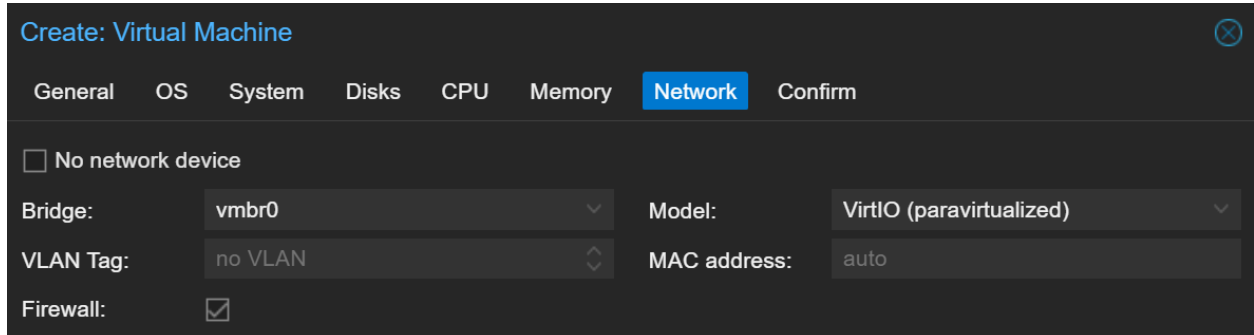
Create: Virtual Machine

General OS System Disks CPU **Memory** Network Confirm

Memory (MiB): 4096

Ilustración 18 Asignación RAM

Asignamos la red.



Create: Virtual Machine

General OS System Disks CPU Memory **Network** Confirm

No network device

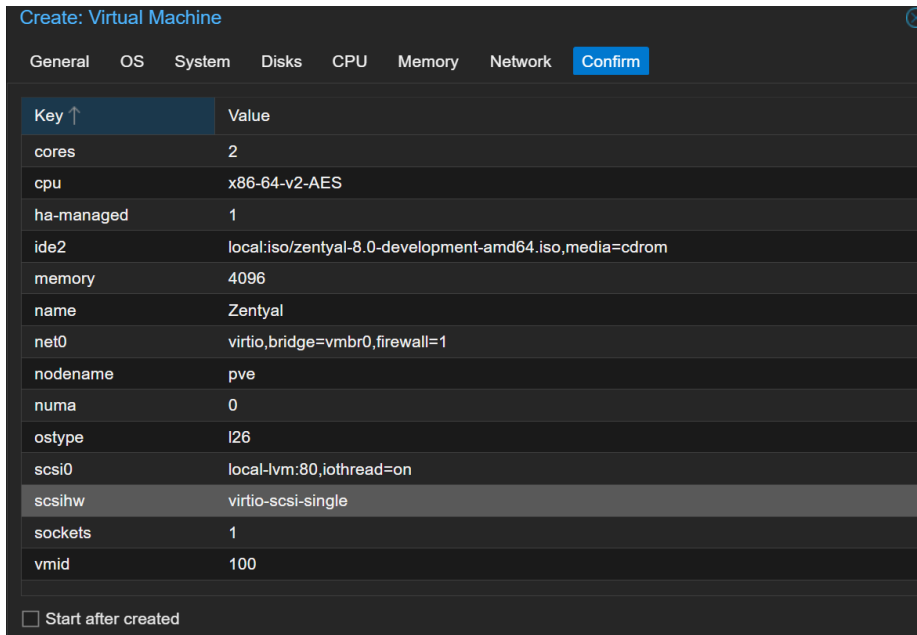
Bridge: vibr0 Model: VirtIO (paravirtualized)

VLAN Tag: no VLAN MAC address: auto

Firewall:

Ilustración 19 Selección Interfaz de Red

Al finalizar podremos ver un resumen de la configuración seleccionada y confirmamos la creación.

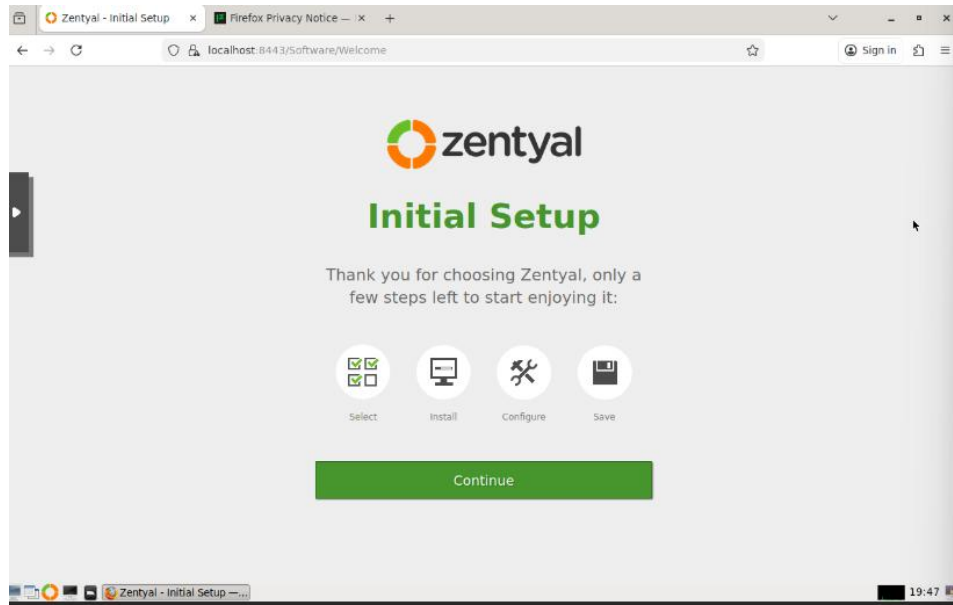


*Ilustración 20 Resumen de VM Zentyal*

#### 4.4.4.2. Implementación Zentyal

Una vez iniciada la VM el proceso de instalación es similar al de cualquier sistema operativo Linux, en específico es similar a Ubuntu server ya que Zentyal usa como base este mismo.

Después de la instalación podremos ver la GUI de Zentyal.



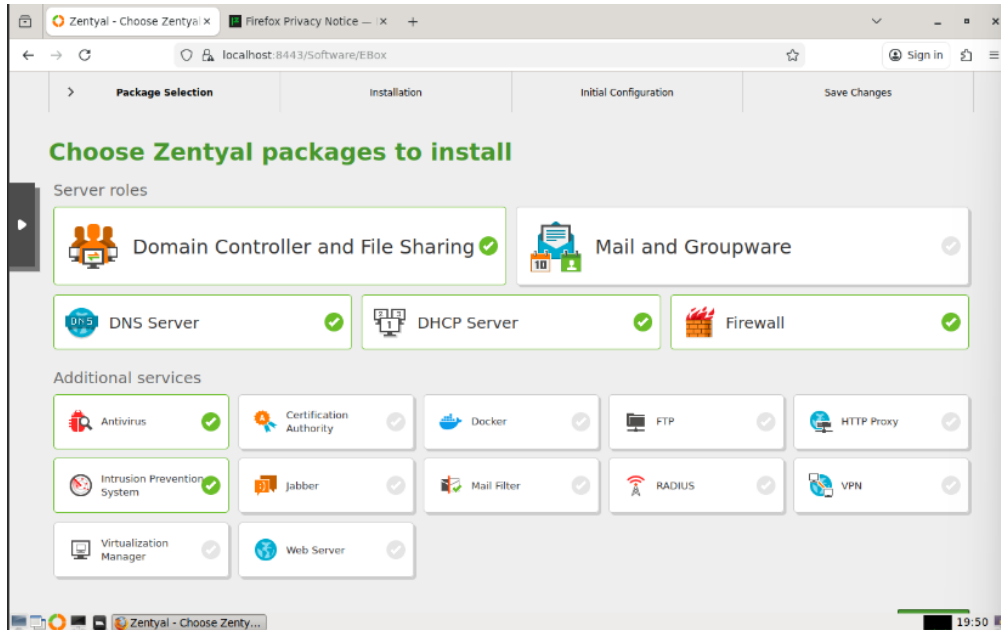
*Ilustración 21 Zentyal Setup Inicial*

Como se mencionó en la tabla comparativa para la selección de herramientas, se escogió Zentyal al ser un entorno integrado en el que se tiene varios servicios en uno facilitando la administración centralizada, sin embargo, se prefirió separar ciertos servicios para poder tener un mayor control sobre cierto servicio sin tener que afectar a otros como es el caso del web server, loadbalancer y el FTP.

Para este proyecto estos fueron los servicios seleccionados que se utilizarán mediante Zentyal.

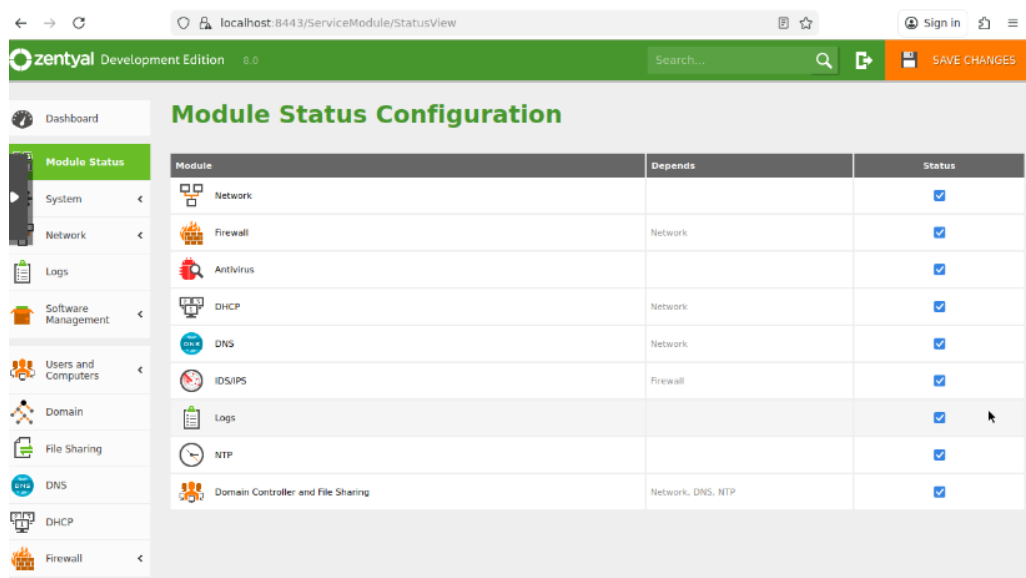
- Controlador de dominio
- DNS
- DHCP
- FIREWALL
- Antivirus
- Sistema de prevención de intrusos.

Se debe mencionar que no son los únicos, pero son los que se consideran apropiados para este prototipo.



*Ilustración 22 Servicios Zentyal*

Al finalizar nos podemos dirigir al dashboard de zentyal, ingresar con las credenciales que son las mismas de la VM y verificamos que los servicios se encuentren activos, si no lo están deberemos activarlos manualmente.



*Ilustración 23 Dashboard Zentyal*

Module Status		
Network	Running	
Firewall	Running	
Antivirus	Running	<input type="button" value="Restart"/>
DHCP	Running	<input type="button" value="Restart"/>
DNS	Running	<input type="button" value="Restart"/>
IDS/IPS	Running	<input type="button" value="Restart"/>
Logs	Running	<input type="button" value="Restart"/>
NTP	Running	<input type="button" value="Restart"/>
Domain Controller and File Sharing	Running	

*Ilustración 24 Revisión servicios Zentyal*

Para que el servicio de DHCP funcione es necesario que la VM tenga una IP estática, esto se lo debe hacer desde el router y desde la consola de zentyal. Se seleccionó primero que la VM obtenga un IP por DHCP para obtener una IP durante la instalación y después que configurará esa misma IP como estática para zentyal.

The screenshot shows the Zentyal web interface for configuring network interfaces. On the left, a sidebar contains navigation options like Dashboard, Module Status, System, Network, Interfaces, Gateways, DNS, Objects, Services, Static Routes, Tools, Logs, and Software Management. The main area is titled 'Network Interfaces' and shows the configuration for the 'eth0' interface. The 'Name' field is 'eth0', the 'Method' is set to 'Static', and the 'IP address' is '192.168.83.228' with a 'Netmask' of '255.255.255.0'. A 'CHANGE' button is visible at the bottom of the configuration form.

Overlaid on the right is a terminal window titled 'zentyal-admin@zentyal-server: ~'. It shows the output of the 'ip a' command, displaying details for the loopback interface 'lo' (127.0.0.1) and the physical interface 'eth0' (192.168.83.228).

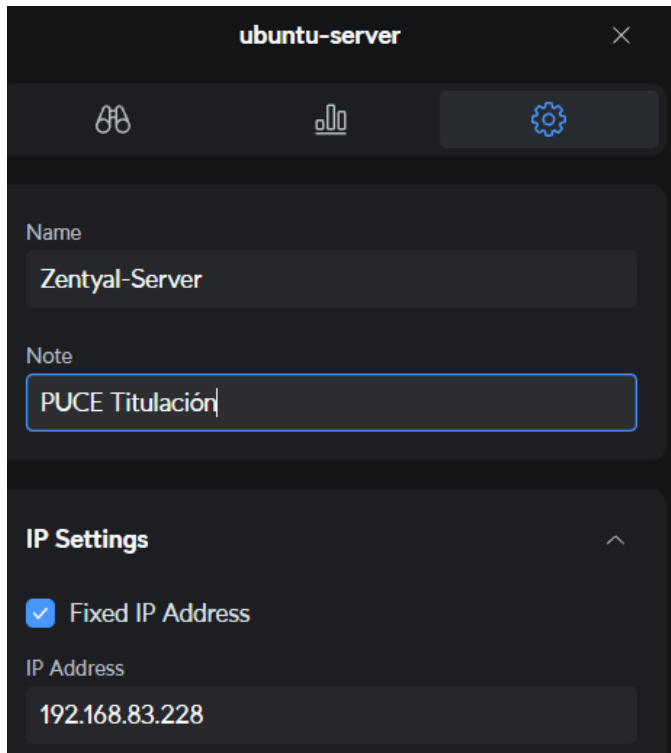
```

zentyal-admin@zentyal-server: ~
Archivo Editar Pestañas Ayuda
To run a command as administrator (user "root"), use "sudo <command>".
See "man sudo_root" for details.

zentyal-admin@zentyal-server: $ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
        valid_lft forever preferred_lft forever
    inet 127.0.1.1/8 scope host secondary lo
        valid_lft forever preferred_lft forever
2: eth0: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether bc:24:11:23:22:bb brd ff:ff:ff:ff:ff:ff
    altname enp0s18
    altname ens18
    inet 192.168.83.228/24 brd 192.168.83.255 scope global dynamic noprefixroute eth0
        valid_lft 85952sec preferred_lft 85952sec
zentyal-admin@zentyal-server: $
  
```

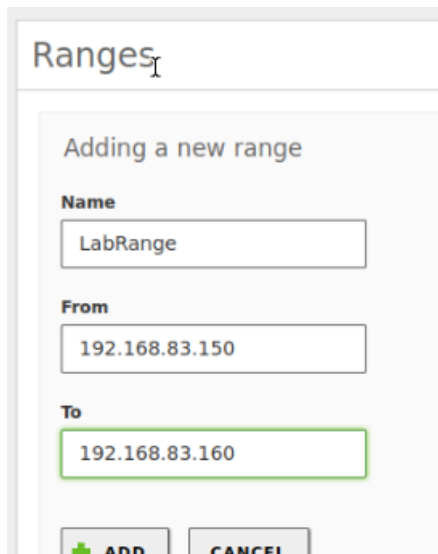
*Ilustración 25 IP estática Zentyal*

En este caso a través del router de unifi se configurará una IP estática.



*Ilustración 26 IP estática en Router*

Adicionalmente se necesita configurar un rango de IPs en zentyal, en este caso se seleccionó el siguiente que previamente se verificó no tenía ningún conflicto con otro equipo en la red.



*Ilustración 27 Rango IPs Zentyal DHCP*

Así mismo se configuraron el resto de los servicios como el de dominio y DNS que servirán para la configuración de las otras VM o CTs.



*Ilustración 28 Dominio Zentyal*

#### 4.4.4.3. Creación VM TrueNAS

TrueNAS será la herramienta que facilite crear un almacenamiento NFS que permita la replicación y orquestación de los contenedores.

Esta VM requiere de su propio OS por lo que se debe seguir un proceso similar al de Zentyal, dirigirse a la página oficial de la herramienta, descargar la ISO y subirla a proxmox. <https://www.truenas.com/download-truenas-community-edition/>

Una vez subida la ISO configuramos la VM con todo lo solicitado y revisaremos el resumen general.

Create: Virtual Machine

General OS System Disks CPU Memory Network Confirm

Key ↑	Value
cores	4
cpu	x86-64-v2-AES
ha-managed	1
ide2	local:iso/TrueNAS-SCALE-25.10.1.iso,media=cdrom
memory	4096
name	TrueNAS
net0	virtio,bridge=vibr0,firewall=1
nodename	pve
numa	0
ostype	l26
scsi0	storage:931,iotread=on
scsihw	virtio-scsi-single
sockets	1
vmid	108

*Ilustración 29 Resumen VM TrueNAS*

Se debe mencionar que en la creación de la vm se debe colocar un disco pequeño para el boot de aproximadamente 32 - 64 GB. Después se agregará otro disco en el que se tendrá el storage de las VMs.

#### 4.4.4.4. Implementación TrueNAS

Una vez iniciada la VM el proceso de instalación es similar a los otros, sin embargo, de aquí en adelante se colocará la información del servidor configurado de Zentyal para DNS, DHCP, dominio, etc.

Finalizada la instalación escogeremos la opción de configuración de red.

```

Console setup
-----
The web user interface is at:
http://192.168.83.140
https://192.168.83.140

1) Configure network interfaces
2) Configure network settings
3) Configure static routes
4) Set up local administrator
5) Reset configuration to defaults
6) Open TrueNAS CLI Shell
7) Open Linux Shell
8) Reboot
9) Shutdown

Enter an option from 1-9:

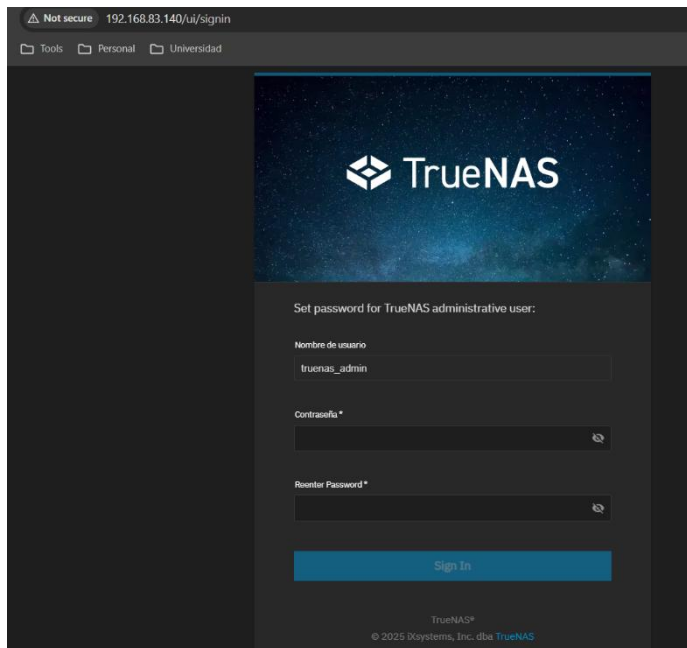
```

Aquí configuramos la información de Zentyal.

```
Network Configuration
hostname: truenas
domain: project-domain
ipv4gateway: 192.168.83.228
ipv6gateway: <empty string>
nameserver1: 192.168.83.228
nameserver2: <empty string>
nameserver3: <empty string>
```

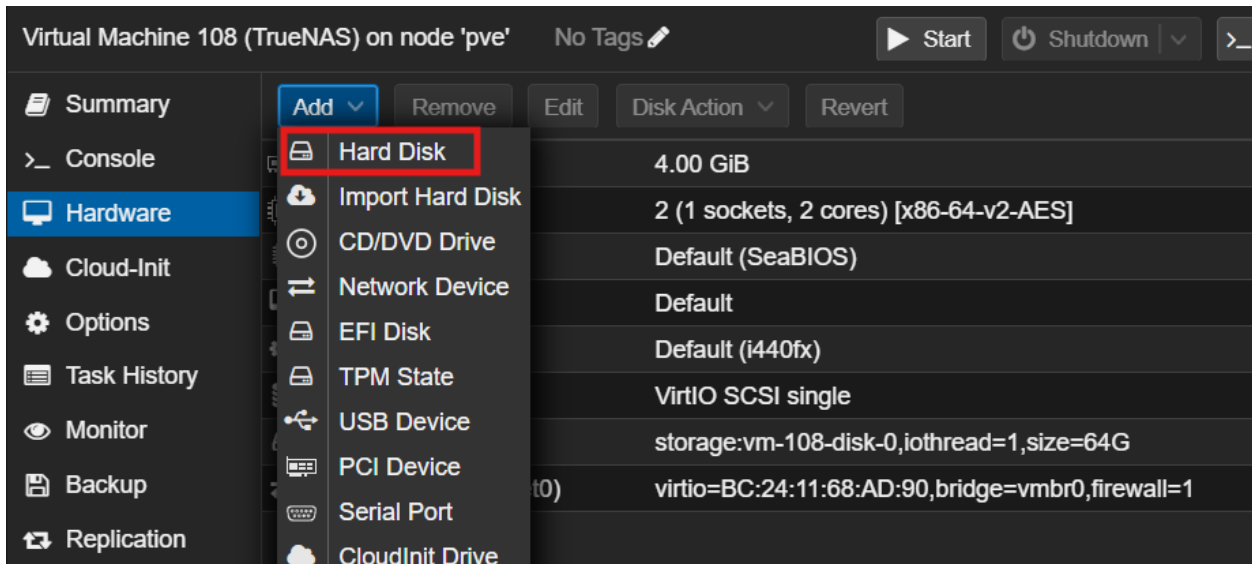
Después del primer inicio y configuración debemos apagar la máquina virtual para poder agregar el nuevo disco desde proxmox.

Aquí se puede observar que la instalación fue correcta.



*Ilustración 30 Login TrueNAS*

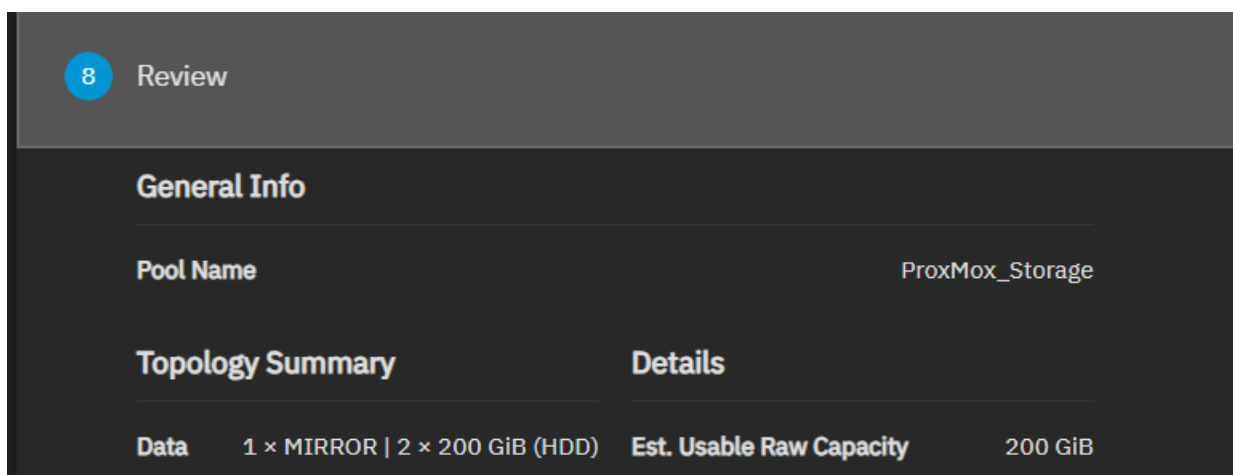
Después nos dirigimos al menú hardware de nuestra VM para agregar el disco mencionado antes para el almacenamiento de las VMs y CTs.



*Ilustración 31 Creación disco para VMs y CTs*

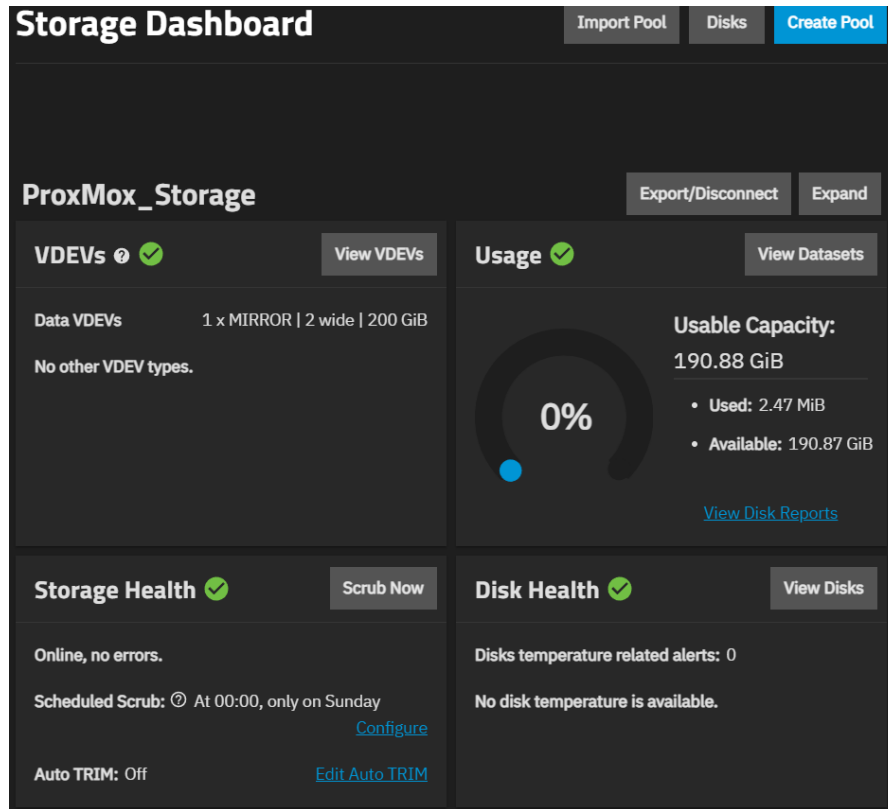
Después ingresaremos a la GUI de TrueNAS para agregar el disco y crear un pool que pueda ser utilizado por los demás servicios.

En este caso se crearán dos discos para tener uno como backup, se configurarán como mirror, es decir se replicará la información de un disco virtual en otro.



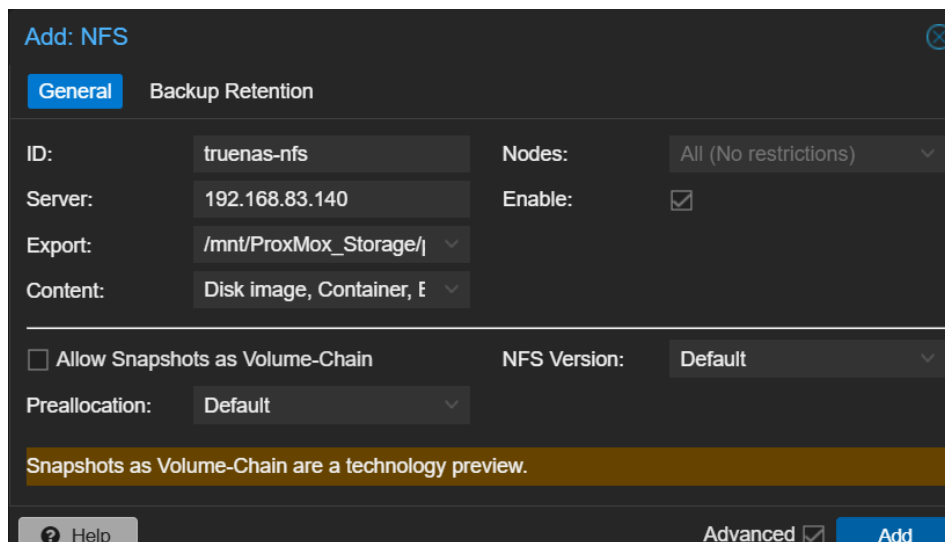
*Ilustración 32 Configuración Discos*

Una vez configurado el storage podremos verificarlo y crear un dataset con ese pool.



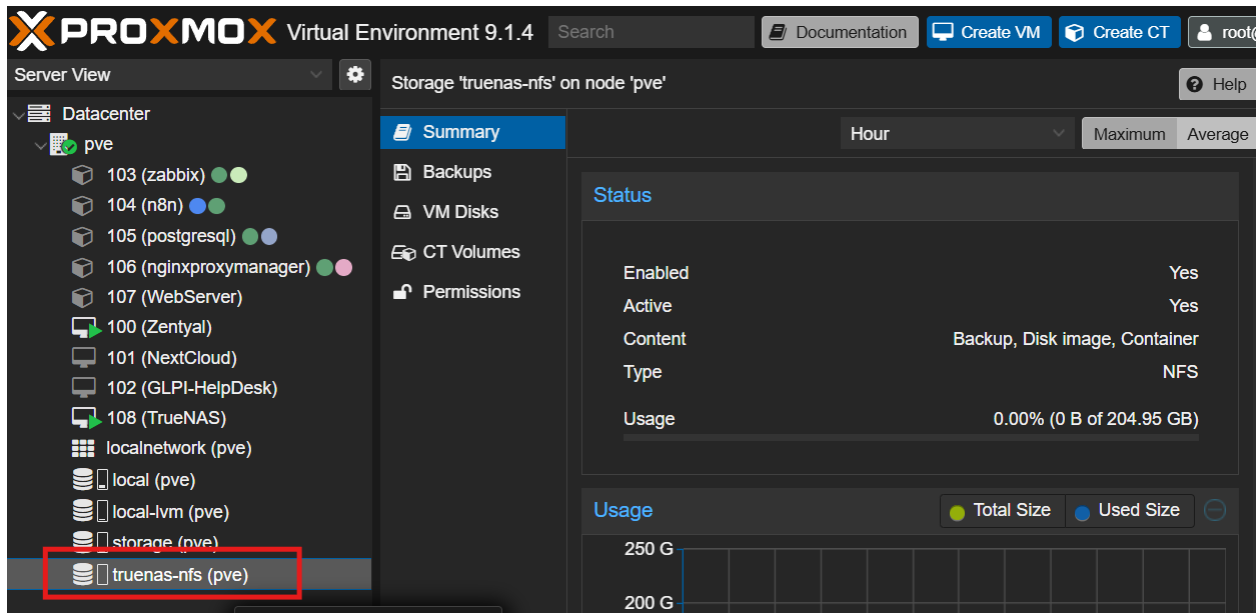
*Ilustración 33 Creación Pool TrueNAS*

El siguiente paso es agregar este dataset como almacenamiento NFS a proxmox, así las demás VMs podrán acceder a este storage lo que permite la replicación y orquestación de varios servicios. Colocamos la IP e información de la ubicación del almacenamiento recién creado.



*Ilustración 34 Agregar almacenamiento NFS en proxmox*

Y podremos verificar en proxmox que este el disco agregado.



*Ilustración 35 Disco TrueNAS en ProxMox*

#### 4.4.4.5. Creación VM NextCloud

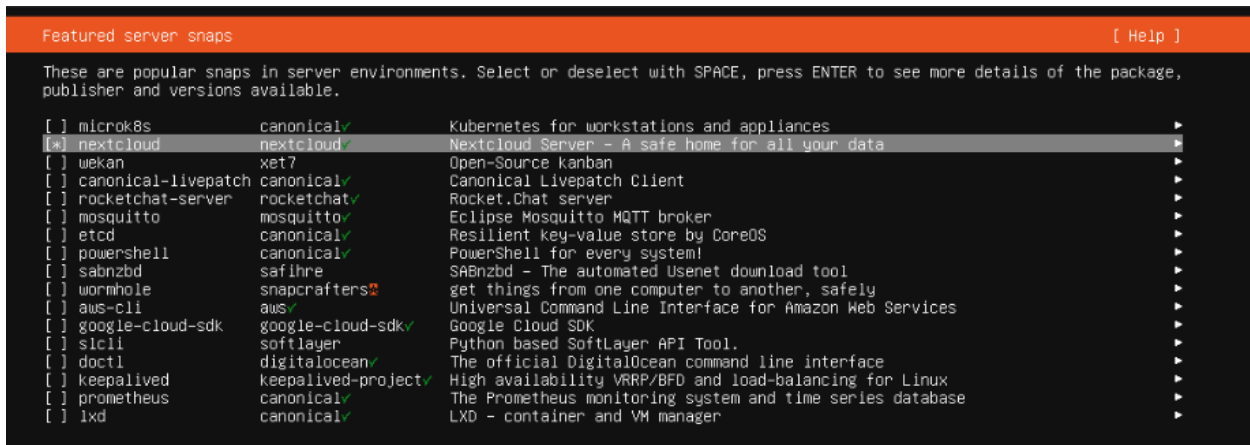
Para NextCloud utilizaremos como OS base Debian 12. Por lo que debemos seguir los pasos anteriores para descargar la ISO y subirla a proxmox. <https://www.debian.org/distrib/>

El siguiente es el resumen de la VM para NextCloud.

Create: Virtual Machine	
Key ↑	Value
cores	2
cpu	x86-64-v2-AES
ha-managed	1
ide2	local:iso/debian-13.2.0-amd64-DVD-1.iso,media=cdrom
memory	4096
name	NextCloud
net0	virtio,bridge=vibr0,firewall=1
nodename	pve
numa	0
ostype	l26
scsi0	local-lvm:476.83,iosthread=on
scsihw	virtio-scsi-single
sockets	1
vmid	101

*Ilustración 36 Resumen VM NextCloud*

Si instalamos el sistema operativo Ubuntu como OS base podremos escoger durante la instalación el servicio de nextcloud.

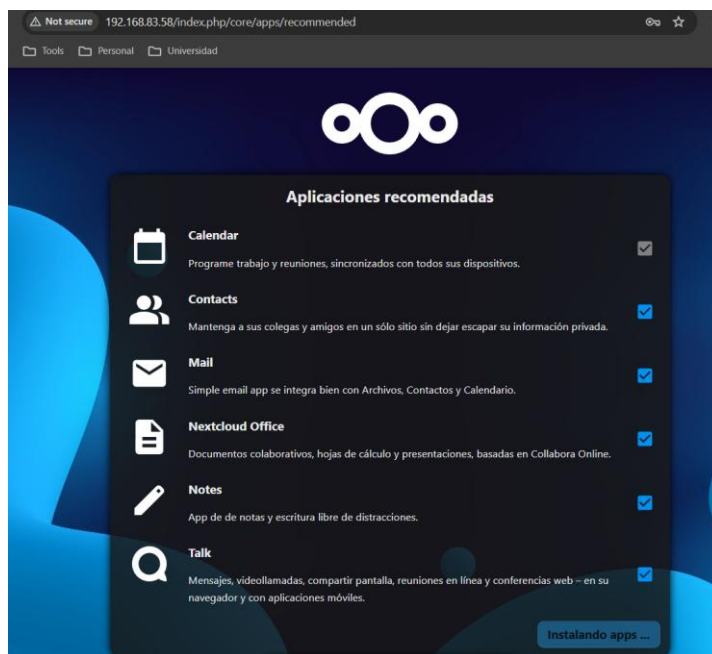


*Ilustración 37 Selección NextCloud*

#### 4.4.4.6. Implementación VM NextCloud

Nextcloud fue implementado como una plataforma de almacenamiento y colaboración orientada a los usuarios del laboratorio virtual. Su función principal es permitir el acceso, sincronización y compartición de archivos dentro de la red, facilitando el trabajo colaborativo y el manejo de información de forma centralizada.

Durante la configuración seleccionaremos las herramientas que sea útiles para la PYME, en este caso se seleccionaron todas.



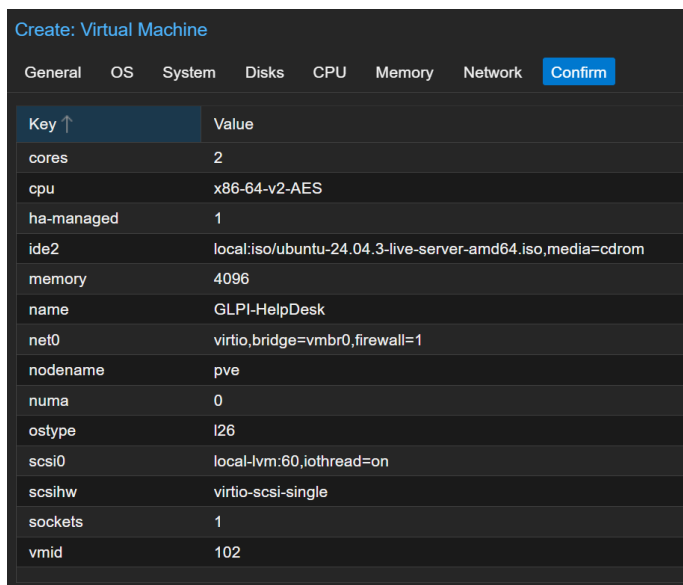
*Ilustración 38 Aplicaciones NextCloud*

#### 4.4.4.7. Creación VM HelpDesk GLPI

Para la herramienta de helpdesk utilizaremos GLPI debido a que es una solución de código abierto ampliamente utilizada para la gestión de incidencias y soporte técnico en pequeñas y medianas empresas.

Utilizaremos Ubuntu 24.04 LTS como OS base para esta VM, ya que es la recomendada por GLPI.

Resumen de la VM para GLPI.



Key ↑	Value
cores	2
cpu	x86-64-v2-AES
ha-managed	1
ide2	local:iso/ubuntu-24.04.3-live-server-amd64.iso,media=cdrom
memory	4096
name	GLPI-HelpDesk
net0	virtio,bridge=vbr0,firewall=1
nodename	pve
numa	0
ostype	l26
scsi0	local-lvm:60,iosthread=on
scsihw	virtio-scsi-single
sockets	1
vmid	102

*Ilustración 39 Resumen VM GLPI*

#### 4.4.4.8. Implementación VM HelpDesk

Para esta VM utilizaremos como OS base Ubuntu server y los siguientes comandos.

Comando	Qué hace
apt install apache2	Instala servidor web Apache
apt install php libapache2-mod-php	Instala PHP para GLPI
apt install php-gd php-curl php-zip php-xml php-mbstring php-intl php-bcmath php-ldap php-pgsql	Instala dependencias de PHP
Wgethttps://github.com/glpi-project/glpi/releases/download/11.0.4/glpi-11.0.4.tgz	Descarga GLPI
a2ensite glpi.conf	Activa el sitio GLPI
a2enmod rewrite	Habilita URLs amigables

*Tabla 9 Comandos VM GLPI-HelpDesk*

#### 4.4.5. Creación de contenedores

Los contenedores Linux (LXC) se utilizaron en el laboratorio virtual como una alternativa eficiente para el despliegue de servicios que no requieren un sistema operativo completo. Este enfoque permite optimizar el uso de los recursos del servidor físico y simplifica la administración de los servicios.

En este proyecto se optó por el uso de contenedores LXC para alojar servicios específicos como monitoreo, automatización, bases de datos y visualización web, los cuales cumplen funciones de apoyo dentro de la infraestructura de TI y pueden ejecutarse de manera independiente.

El uso de contenedores se definió considerando los siguientes aspectos:

- Menor consumo de recursos en comparación con las máquinas virtuales.
- Rapidez en la creación y puesta en marcha de los servicios.
- Facilidad para la administración y mantenimiento.
- Posibilidad de escalar recursos según las necesidades del laboratorio.
- Integración directa con el entorno de virtualización Proxmox VE.

En la siguiente tabla se presentan las características de los contenedores implementados.

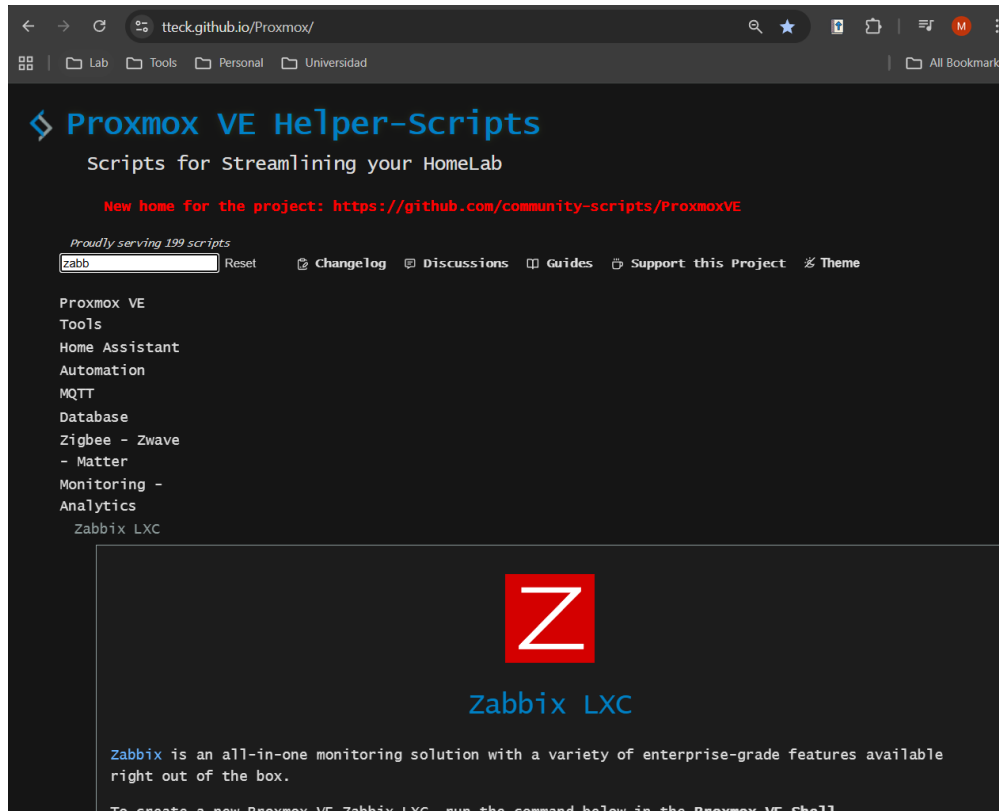
Contenedor	Servicio	vCPU	RAM	Almacenamiento
CT-Zabbix	Monitoreo	2	4GB	6GB
CT-n8n	Automatización	2	2GB	10GB
CT-Bdd	Base de datos	1	2GB	40GB
CT-Web	Web	1	1GB	20GB
CT-Nginx	Proxy-LoadBalancer	1	1GB	8GB

**Tabla 10 Detalle Contenedores**

Para la creación de los contenedores se utilizó la herramienta de Proxmox Helper Scripts, la misma que fue utilizada anteriormente para la configuración inicial después de la instalación por lo que el proceso será el mismo para todos los contenedores, a continuación, se muestra un ejemplo de la creación del contenedor de Zabbix la herramienta de monitoreo.

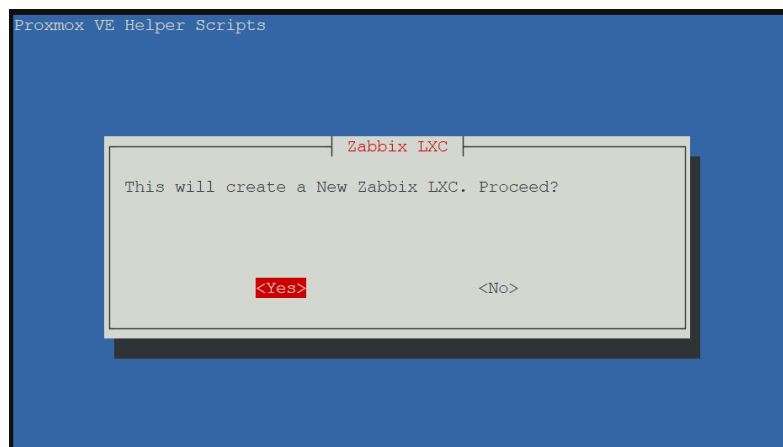
### 4.4.5.1. Creación contenedor Zabbix

Primero nos dirigimos a la página web <https://community-scripts.github.io/ProxmoxVE/scripts?id=zabbix> aquí buscaremos el nombre de nuestro contenedor en este zabbix.



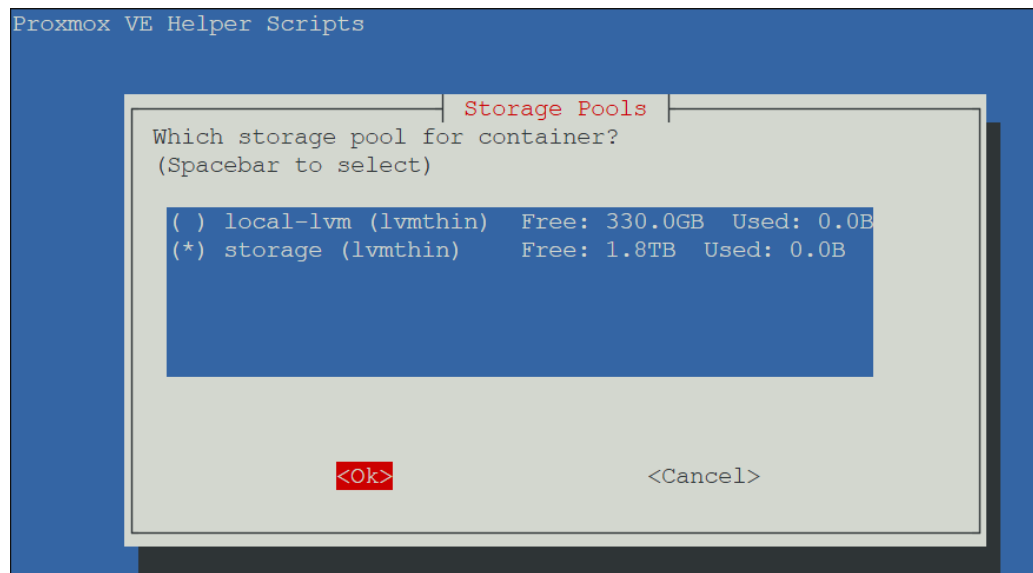
*Ilustración 40 Zabbix Helper Scripts*

Copiaremos el comando que nos indica la guía en nuestra consola de proxmox para crear el contenedor.



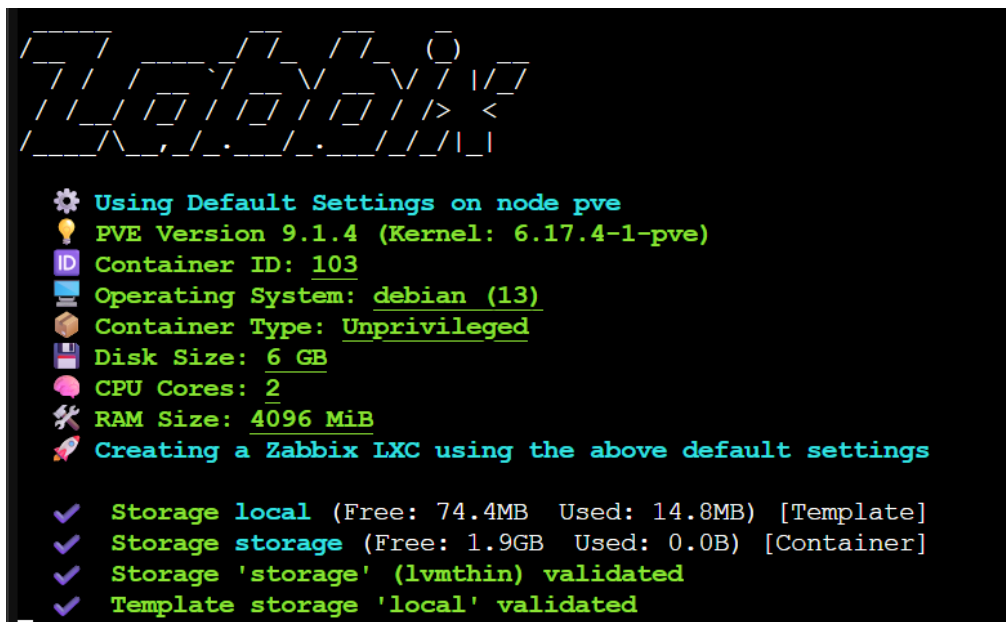
*Ilustración 41 Creación contenedor Zabbix*

Seleccionamos nuestro almacenamiento.



*Ilustración 42 Selección almacenamiento CT*

Al finalizar podremos ver este resumen con la configuración.

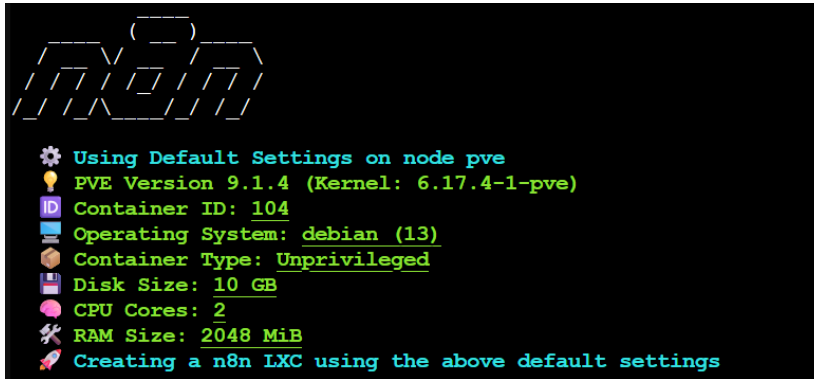


*Ilustración 43 Resumen CT Zabbix*

### 4.4.5.2. Creación CT n8n

Para este contenedor utilizaremos de igual forma los helper scripts para su instalación. <https://community-scripts.github.io/ProxmoxVE/scripts?id=n8n>

La siguiente imagen indica las características del contenedor.



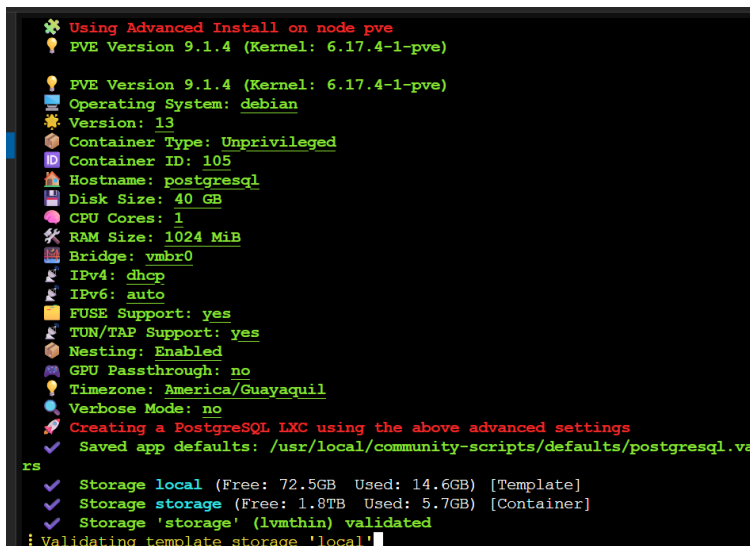
*Ilustración 44 Resumen CT n8n*

La implementación a detalle de este contenedor se realizará en su propio apartado ya que esta herramienta es la encargada de la automatización el cual es uno de los objetivos principales de este proyecto de titulación.

### 4.4.5.3. Creación contenedor BDD

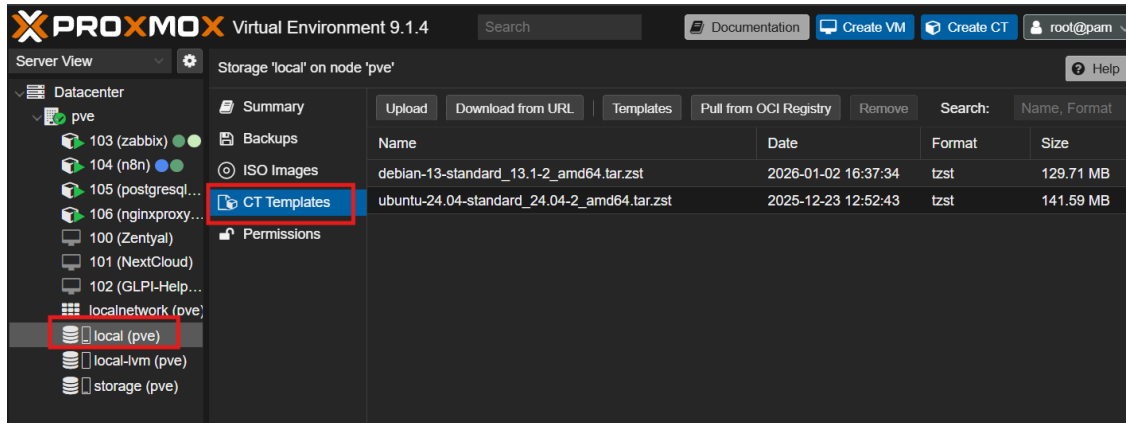
Para este contenedor utilizaremos el script de <https://community-scripts.github.io/ProxmoxVE/scripts?id=postgresql>

En la siguiente imagen se pueden ver las características del contenedor.



*Ilustración 45 Resumen CT Base de datos*

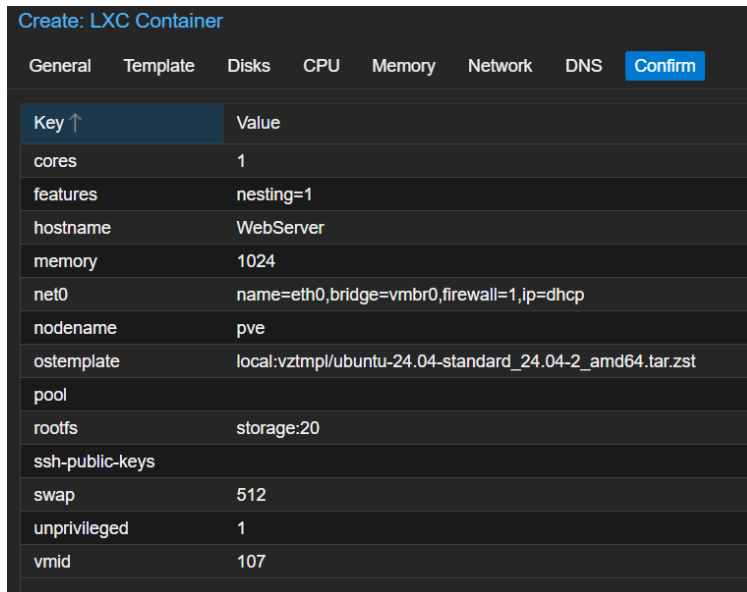




*Ilustración 47 Ubicación template CTs*

Daremos click en templates y seleccionaremos el que nos interese en este caso Ubuntu 24.04.

Una vez descargado daremos click en Create CT y completaremos la información necesaria al igual que lo hicimos con la creación de las VM anteriores, al finalizar podremos ver un resumen similar.



*Ilustración 48 Creación CT PáginaWeb*

#### 4.4.5.2. Implementación CT Web Server

Para la implementación de este servicio utilizaremos node y una página web básica hecha en html, css y js para visualización de ciertos parámetros de red como velocidad, equipos conectados, puertos abiertos, etc. Con la herramienta nmap y otras. Se mostrará información útil con gráficas sencillas para facilidad la interpretación del usuario.

Se requiere la herramienta curl para descargar el repositorio de node para esto utilizaremos los siguientes comandos.

Comando	Qué hace
sudo apt install curl -y	Instalar la herramienta curl para manejar descargar
curl -fsSL https://deb.nodesource.com/setup_lts.x   sudo -E bash -	Agregar el repositorio oficial de Node.js (versión LTS)
sudo apt install nodejs -y	Instalar Node.js y npm
node -v npm -v	Verificar instalación

*Tabla 11 Comandos Instalación node.js*

```
root@WebServer:~# node -v
v24.12.0
11.6.2
```

*Ilustración 49 Resumen instalación node.js*

Una vez instalada la herramienta se procede a crear una carpeta con la estructura de nuestra web, en este caso será sencilla con solo html, css y js. Para esto se recomienda que la ruta este en el directorio /var

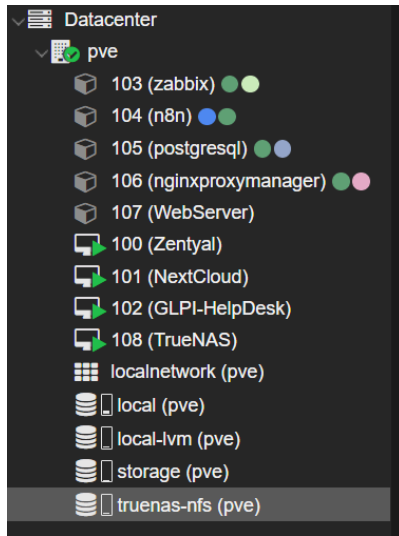
Comando	Qué hace
Mkdir /var/www Mkdir /var/www/web_metricas Mkdir /var/www/web_metricas/css Mkdir /var/www/web_metricas/js Mkdir /var/www/web_metricas/data	Crear la estructura del proyecto FrontEnd
mkdir /var/www/api_metricas	Crear la estructura del proyecto BackEnd
npm init -y	Inicializar proyecto
npm install express axios dotenv	Instalar dependencias
nano .env	Configurar variables de entorno

*Tabla 12 Comandos creación estructura página web*

Con estos comandos se puede crear la página web para la visualización de la red y equipos, sin embargo, el detalle se explicará en el capítulo 4.4. Implementación del monitoreo, con las llamadas al api de Zabbix para obtener las métricas de los servicios.

#### 4.4.6. Resumen Creación e Implementación VMs y CTs

Una vez levantados todos los servicios tendremos lo siguiente:



*Ilustración 50 Resumen Servicios levantados*

Se realizó la siguiente tabla para asociar el ID de cada VM y CT, esto será útil más adelante para identificar cada servicio.

Nombre	Tipo	ID Proxmox
n8n	CT	104
postgresql	CT	105
nginxproxymanager	CT	106
WebServer	CT	107
Zentyal	VM	100
NextCloud	VM	101
GLPI-HelpDesk	VM	102
Zabbix	VM	103
TrueNAS	VM	108

*Tabla 13 IDs de servicios*

También podemos verificar los discos de los CTs alojados en el almacenamiento proporcionado por truenas.

Storage 'truenas-nfs' on node 'pve'				
		Search:		Help
		Name	Date	Format
Summary	Remove			
Backups				
VM Disks		vm-103-disk-0.raw	2026-01-04 17:35:00	raw 6.44 GB
CT Volumes		vm-104-disk-0.raw	2026-01-04 17:37:19	raw 10.74 GB
Permissions		vm-105-disk-0.raw	2026-01-04 17:37:57	raw 42.95 GB
		vm-106-disk-0.raw	2026-01-04 17:39:14	raw 8.59 GB
		vm-107-disk-0.raw	2026-01-04 17:40:00	raw 21.47 GB

*Ilustración 51 Discos en TrueNAS*

Dado que las PYMEs pueden ser muy variadas y su objetivo de negocio puede llegar a ser distinto es posible que los servicios difieran. Como se mencionó antes se seleccionaron servicios que son de uso general y muy posiblemente sean utilizadas por varias empresas, tal vez con distintas herramientas, pero con los mismos objetivos, redes, almacenamiento, etc.

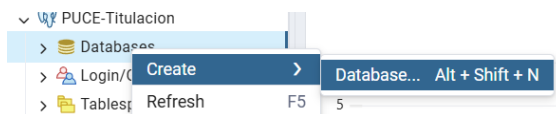
En el capítulo anterior solo se crearon las VMs y CTs necesarias sin embargo, en el siguiente capítulo y ya que este proyecto se centra en el monitoreo y automatización se procederá con la configuración a detalle de ambos servicios.

## 4.5. Implementación del monitoreo

### 4.5.1. Instalación Zabbix

Dado que este es uno de los servicios principales de este proyecto se enfatizará y se ahondará más en este, la herramienta seleccionada es zabbix la cual según sus creadores es *“Un software que monitorea numerosos parámetros de una red y la salud y la integridad de servidores, máquinas virtuales, aplicaciones, servicios, bases de datos, sitios web, la nube y más. Zabbix utiliza un mecanismo de notificación flexible que permite a los usuarios configurar alertas basadas en correo electrónico para prácticamente cualquier evento. Esto permite una reacción rápida ante problemas del servidor. Zabbix ofrece excelentes informes y funciones de visualización de datos basadas en los datos almacenados.”* (2 *Qué Es Zabbix, 2025*).

Durante la instalación utilizaremos la base de datos postgresql que tenemos levantada, simplemente crearemos una base de datos como nos indica la herramienta, para esto utilizaremos la herramienta pgadmin para conectarnos al servidor, crear la base y un usuario.

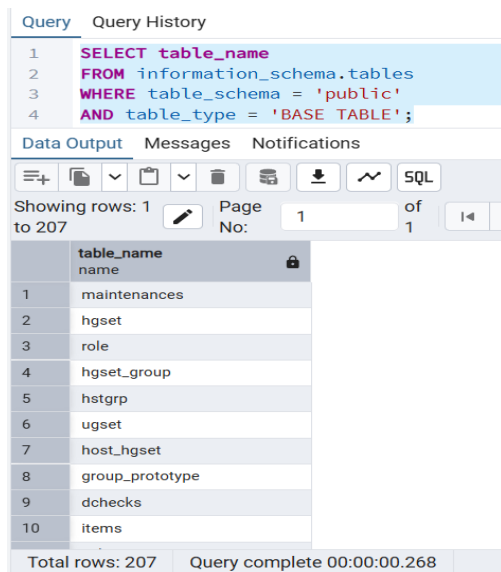


*Ilustración 52 Creación Base de datos para zabbix*

Dentro del contenedor de zabbix utilizaremos el siguiente comando para utilizar el script que configuré la base de datos.

```
zcat /usr/share/zabbix/sql-scripts/postgresql/server.sql.gz | \ psql -U zabbix_admin -h 192.168.83.57 zabbix.
```

Este script creará tablas, relaciones y demás elementos que necesita zabbix para su funcionamiento, después podremos verificarlo en pgadmin.



The screenshot shows a PostgreSQL query tool interface. At the top, there are tabs for 'Query' and 'Query History'. The 'Query' tab is active, displaying a SQL query:

```
1 SELECT table_name
2 FROM information_schema.tables
3 WHERE table_schema = 'public'
4 AND table_type = 'BASE TABLE';
```

Below the query, there are tabs for 'Data Output', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table of results. The table has a header row with 'table\_name' and 'name'. The results are as follows:

	table_name	name
1	maintenances	
2	hgset	
3	role	
4	hgset_group	
5	hstgrp	
6	ugset	
7	host_hgset	
8	group_prototype	
9	dchecks	
10	items	

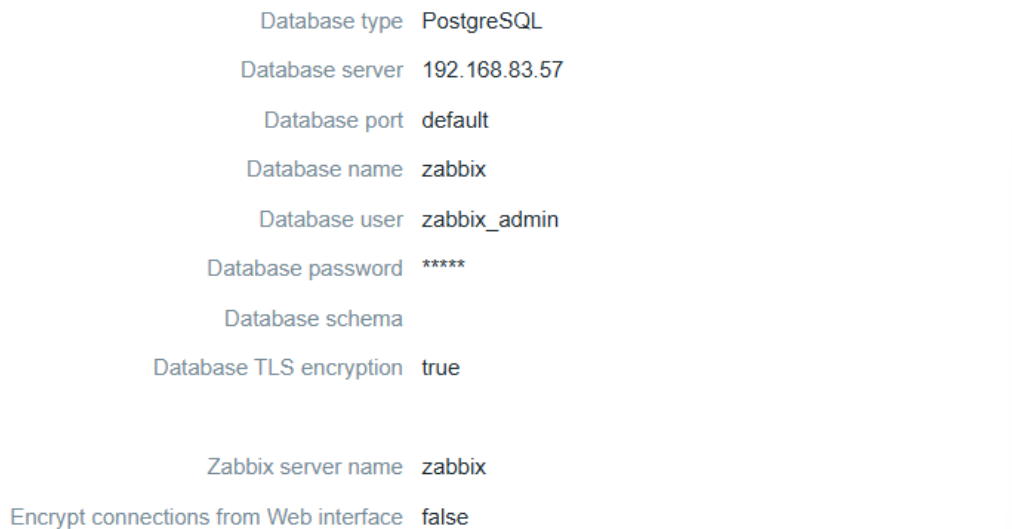
At the bottom of the interface, it shows 'Total rows: 207' and 'Query complete 00:00:00.268'.

*Ilustración 53 Creación estructura base de datos para zabbix*

Al finalizar veremos un resumen de toda la instalación

## Pre-installation summary

Please check configuration parameters. If all is correct, press "Next step" button, or "Back" button to change configuration parameters.



The screenshot shows the 'Pre-installation summary' screen. It displays the following configuration parameters:

- Database type: PostgreSQL
- Database server: 192.168.83.57
- Database port: default
- Database name: zabbix
- Database user: zabbix\_admin
- Database password: \*\*\*\*\*
- Database schema:
- Database TLS encryption: true
- Zabbix server name: zabbix
- Encrypt connections from Web interface: false

*Ilustración 54 Resumen instalación zabbix*

## 4.5.2. Instalaciones agentes Zabbix

Para la instalación de agentes se utilizarán dos métodos distintos uno para los CTs y otro para los VMs, sin embargo, esto se podría hacer manualmente de máquina en máquina.

### 4.5.2.1. Agentes en CTs

Para la instalación de cada agente seguiremos la guía de zabbix en la cual seleccionaremos nuestro OS y servicio a monitorear, nos dará unos comandos que debemos seguir para instalar el agente. En este caso haremos la instalación para el contenedor de base de datos.

ZABBIX VERSION	OS DISTRIBUTION	OS VERSION	ZABBIX COMPONENT
7.4	Alma Linux	24.04 Noble (amd64, arm64)	Server, Frontend, Agent
7.2	Amazon Linux	22.04 Jammy (amd64, arm64)	Server, Frontend, Agent 2
7.0 LTS	CentOS	20.04 Focal (amd64, arm64)	Proxy
6.0 LTS	Debian	18.04 Bionic (amd64, i386)	Agent
8.0 PRE-RELEASE	OpenSUSE Leap	16.04 Xenial (amd64, i386)	Agent 2
	Oracle Linux		Java Gateway
	Raspberry Pi OS		Web Service
	Red Hat Enterprise Linux		
	Rocky Linux		
	SUSE Linux Enterprise Server		
	Ubuntu		

*Ilustración 55 Agente para CT BDD*

Después de seleccionar nuestro sistema operativo, seguiremos los siguientes comandos que nos da la herramienta.

**2** Install and configure Zabbix for your platform

**a. Become root user**  
Start new shell session with root privileges.

```
$ sudo -s
```

**b. Install Zabbix repository** [Documentation](#)

```
wget https://repo.zabbix.com/zabbix/7.4/release/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest_7.4ubuntu24.04_all.deb
dpkg -i zabbix-release_latest_7.4ubuntu24.04_all.deb
apt update
```

**c. Install Zabbix agent**

```
# apt install zabbix-agent
```

**d. Start Zabbix agent process**  
Start Zabbix agent process and make it start at system boot.

```
# systemctl restart zabbix-agent
# systemctl enable zabbix-agent
```

*Ilustración 56 Comandos instalación agente*

Una vez instalado debemos agregar al archivo de configuración la IP del servidor de zabbix, esto está en la ruta /etc/zabbix/zabbix.agentd.conf.

```
GNU nano 8.4 /etc/zabbix/zabbix_agentd.conf *
##### Passive checks related

### Option: Server
# List of comma delimited IP addresses, optionally in CIDR notation, or DNS name
# Incoming connections will be accepted only from the hosts listed here.
# If IPv6 support is enabled then '127.0.0.1', ':::127.0.0.1', '::ffff:127.0.0.1'
# and ':::/0' will allow any IPv4 or IPv6 address.
# '0.0.0.0/0' can be used to allow any IPv4 address.
# Example: Server=127.0.0.1,192.168.1.0/24,:::1,2001:db8::/32,zabbix.example.com
#
# Mandatory: yes, if StartAgents is not explicitly set to 0
# Default:
# Server=

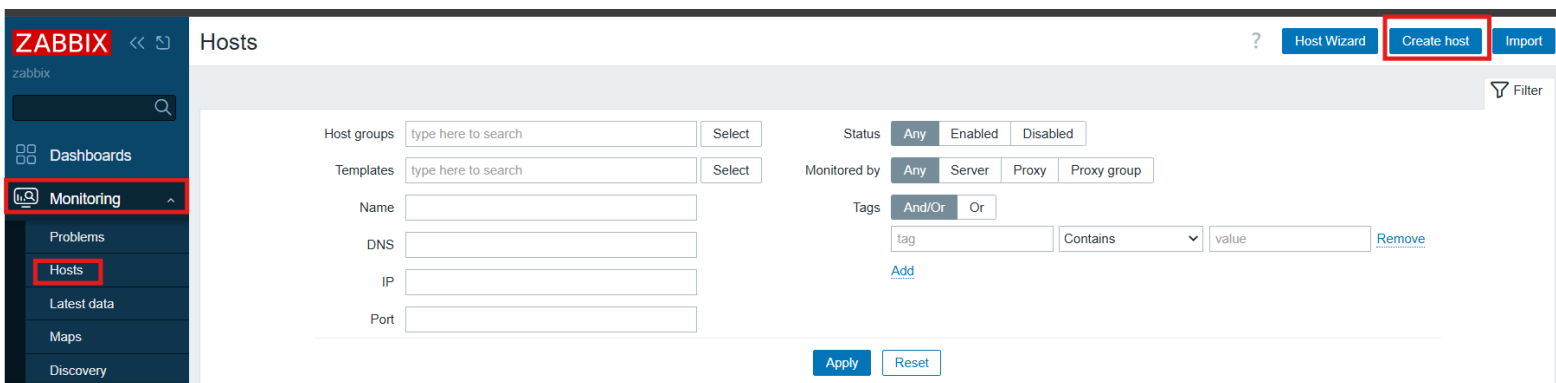
Server=192.168.83.245

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

*Ilustración 57 Agregar servidor en agente*

Después debemos reiniciar el servicio y podremos agregarlo al servidor.

En el dashboard de zabbix nos dirigimos a monitoring -> hosts -> create host



*Ilustración 58 Agregar host en Zabbix*

Agregamos la información del Contenedor. También agregamos los templates y grupos que correspondan en este caso de Sistema Operativo Linux, base de datos PostgreSQL y VMs.

### New host

Host IPMI Tags Macros Inventory Encryption Value mapping

\* Host name

Visible name

Templates    
type here to search

\* Host groups    
type here to search

Interfaces	Type	IP address	DNS name	Connect to	Port	Default
Agent		<input type="text" value="192.168.83.57"/>	<input type="text"/>	<input checked="" type="radio"/> IP <input type="radio"/> DNS	<input type="text" value="10050"/>	<input checked="" type="radio"/> <a href="#">Remove</a>

**Ilustración 59 Información Host en Zabbix**

Este proceso se repetirá para todos las máquinas virtuales y contenedores del laboratorio, de esta forma podremos monitorear el consumo de todos los servicios y proceder con la automatización y control de los elementos.

**Top hosts by CPU utilization** ⚙️ ⋮

Host name	Utilization	1m avg	5m avg	15m avg	Processes
<a href="#">CT-BDD</a>	5.09 %	0.24	0.24	0.30	74
<a href="#">CT-WebServer</a>	3.64 %	0.32	0.25	0.31	32
<a href="#">CT-n8n</a>	3.51 %	0.23	0.25	0.30	30
<a href="#">VM-NextCloud</a>	1.38 %	0.01	0.02	0.00	152
<a href="#">Zabbix server</a>	0.84 %	0.10	0.07	0.02	187
<a href="#">VM-GLPI</a>	0.08 %	0.00	0.00	0.00	129
<a href="#">CT-Nginx</a>	0.00 %	0.27	0.24	0.30	33

**Ilustración 60 Servicios registrados en Zabbix**

### 4.5.3. Página monitoreo personalizada

Otro aspecto fundamental de este proyecto es poder tener centralizada la información de los servicios y poder interactuar con ellas para esto utilizaremos la API de Zabbix y Proxmox. También utilizaremos una VM con herramientas de red que permitan escanear y controlar los equipos de red.

#### 4.5.3.1. Zabbix API

El primer paso es crear un token para poder acceder a la información que provee Zabbix. Desde el menú usuarios -> API Tokens -> Create API Token.

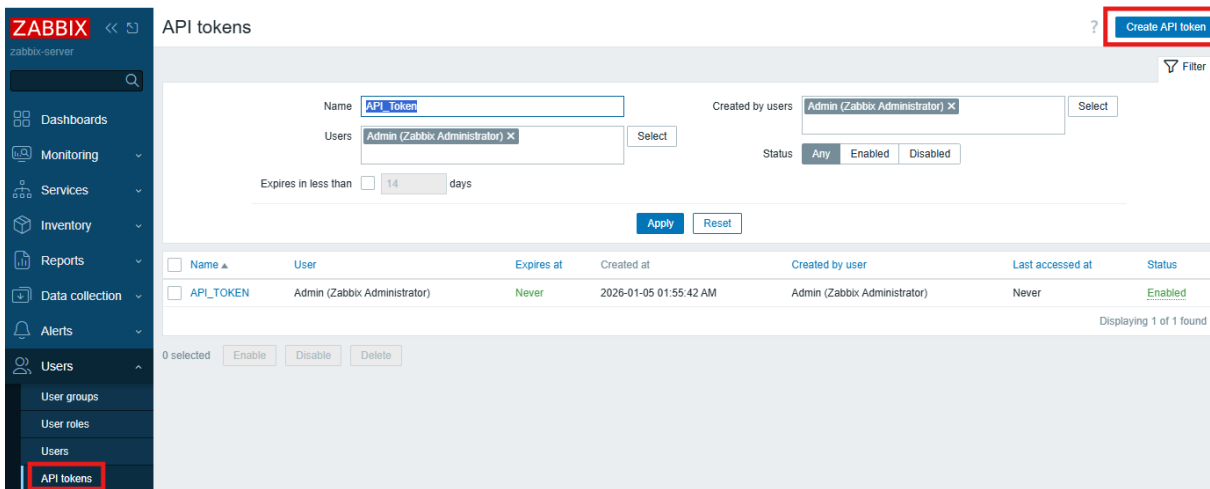


Ilustración 61 Zabbix API Token

Se pueden hacer pruebas por postman utilizando mandando el token como bearer y los requests que nos entrega zabbix, por ejemplo:

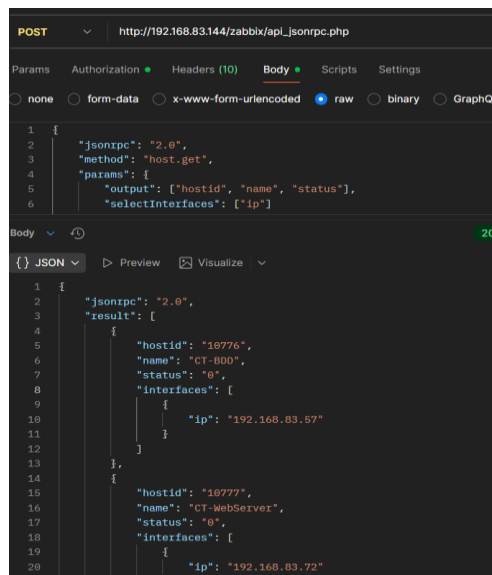


Ilustración 62 Pruebas API Zabbix

### 4.5.3.2. Página web

Como uno de los objetivos es el monitoreo se realizó una página web propia como interfaz gráfica para visualización sobre el estado de las VMs y CTs, para esto se utilizó el API que zabbix nos ofrece para recopilar métricas sobre nuestros servicios.

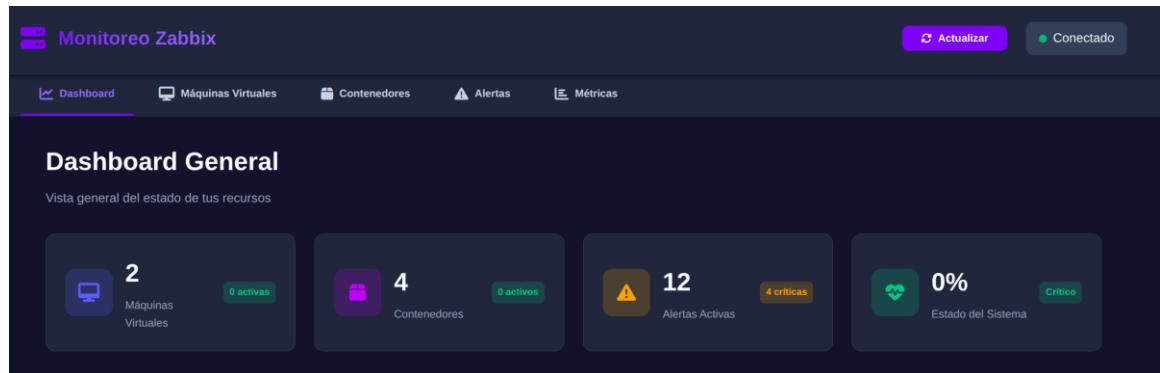
En la siguiente tabla se detalla las llamadas a las APIs utilizadas.

Componente	Contenido	Comentario
URL	http://192.168.83.144/zabbix/api_jsonrpc.php	La URL es la misma para todo, lo que cambia es el method en el body.
Body-Login	<pre>{   "jsonrpc": "2.0",   "method": "user.login",   "params": {     "username": "Admin",     "password": "zabbix"   },   "id": 1 }</pre>	Body para realizar un login y obtener un token que será utilizado en las siguientes llamadas.
Body-métricas	<pre>{   "jsonrpc": "2.0",   "method": "item.get",   "params": {     "output": ["itemid", "name", "lastvalue",     "units"],     "hostids": "10779",     "search": {       "name": ["CPU utilization", "Memory utilization"]     },     "searchByAny": true   },   "id": 1 }</pre>	Con este request podemos obtener métricas de CPU y memoria RAM de un cierto host.

*Tabla 14 Comandos API Zabbix*

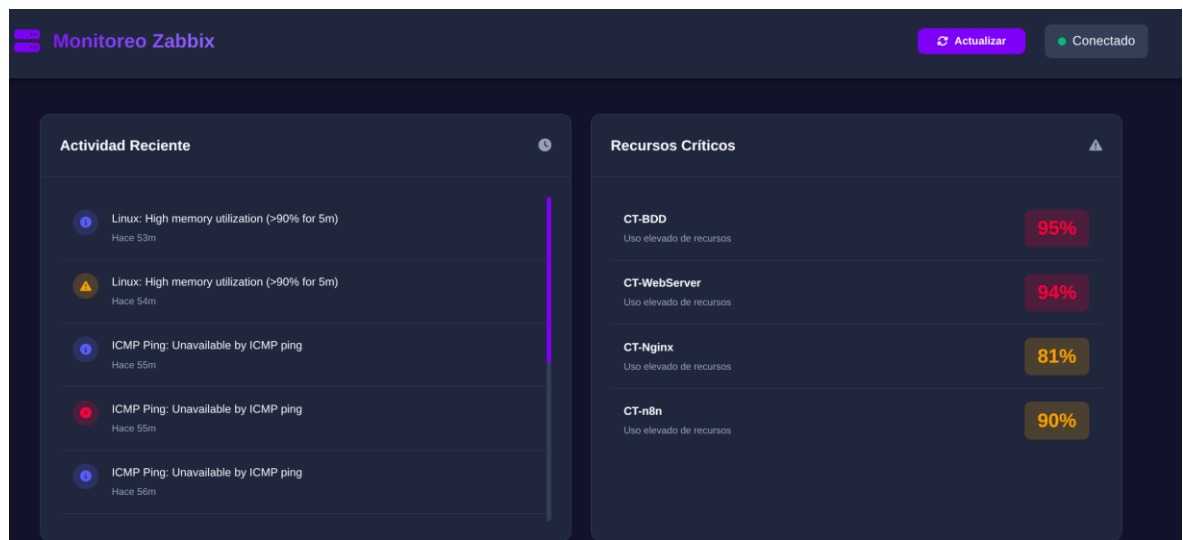
Con los comandos anteriores podemos armar nuestra página web para que utilice dichas métricas y se realicen gráficas con esa data.

Ya en nuestra página web lo primero que tenemos es un dashboard que nos ayudará con la navegación.



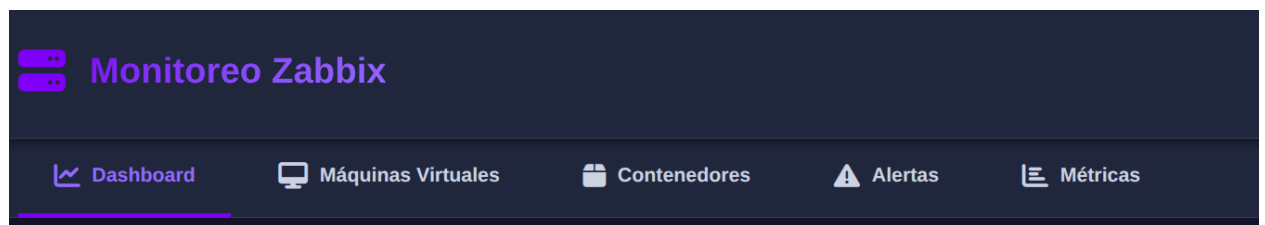
*Ilustración 63 Dashboard página web*

Debajo se muestra información importante sobre nuestros servicios como la actividad reciente, aquí se incluye información de los triggers configurados en zabbix, además de servicios que necesiten atención inmediata.



*Ilustración 64 Dashboard - Información importante*

Para facilidad del usuario se incluyó un menú de navegación en el se podría revisar más a detalle alguno de los componentes.

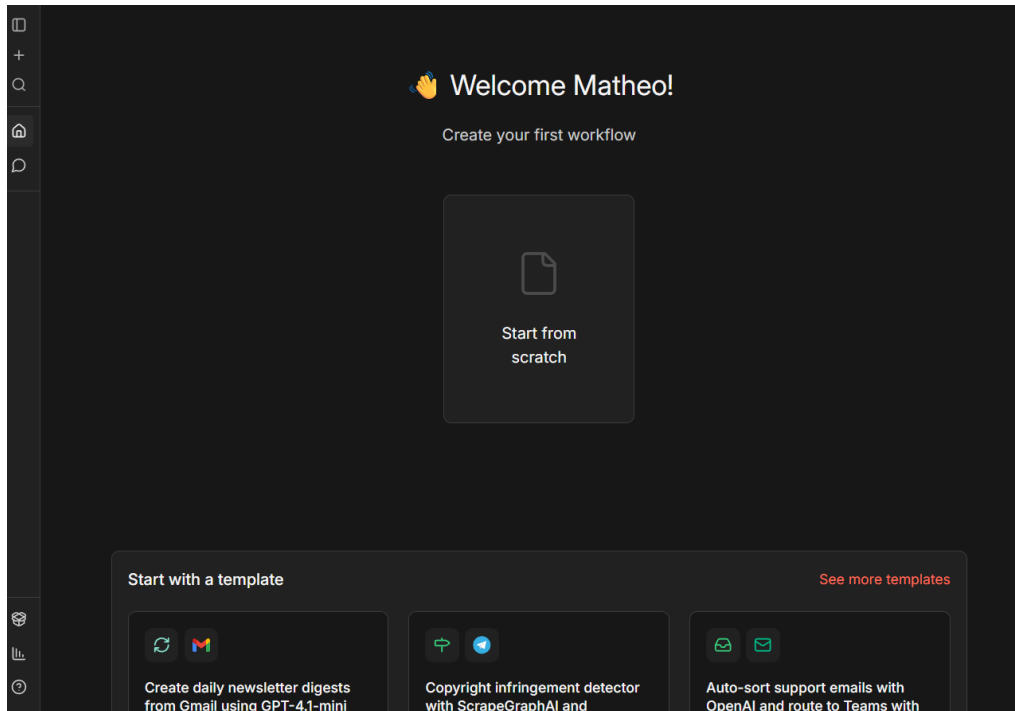


*Ilustración 65 Menú de navegación*

## 4.6. Implementación de la automatización

Este capítulo se centra en la automatización de las VMs y CTs mediante n8n la herramienta que permite generar flujos para automatizar acciones sobre los demás servicios. Utilizaremos la información que provee la herramienta de monitoreo zabbix para poder determinar acciones sobre los elementos del laboratorio.

En nuestro primer ingreso debemos crear una cuenta y después podremos visualizar el dashboard.



*Ilustración 66 Dashboard n8n*

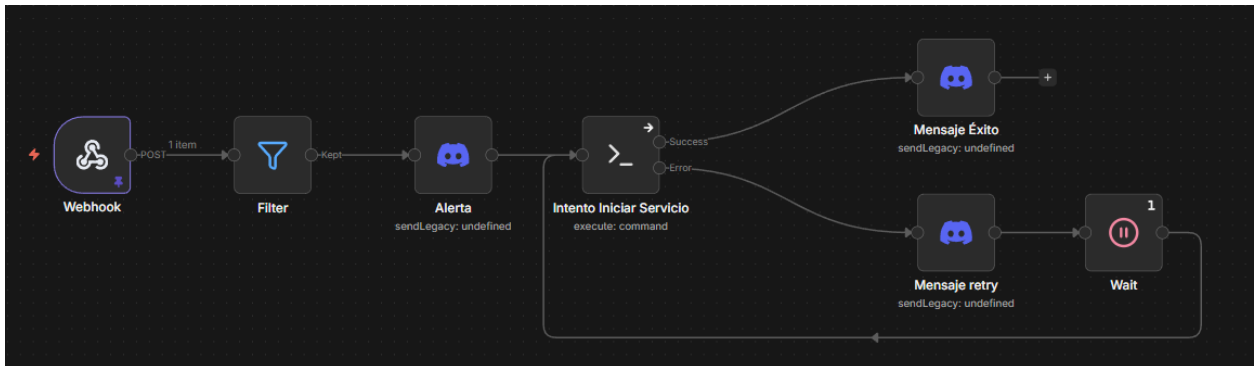
### Flujos n8n

Dentro de n8n podremos establecer varios flujos para distintas acciones que planeamos automatizar, estás puedes ser controlar el estado de los servicios, aumentar o disminuir instancias o recursos de cierto servicio, etc.

#### 4.6.1. Flujo activación de servicios

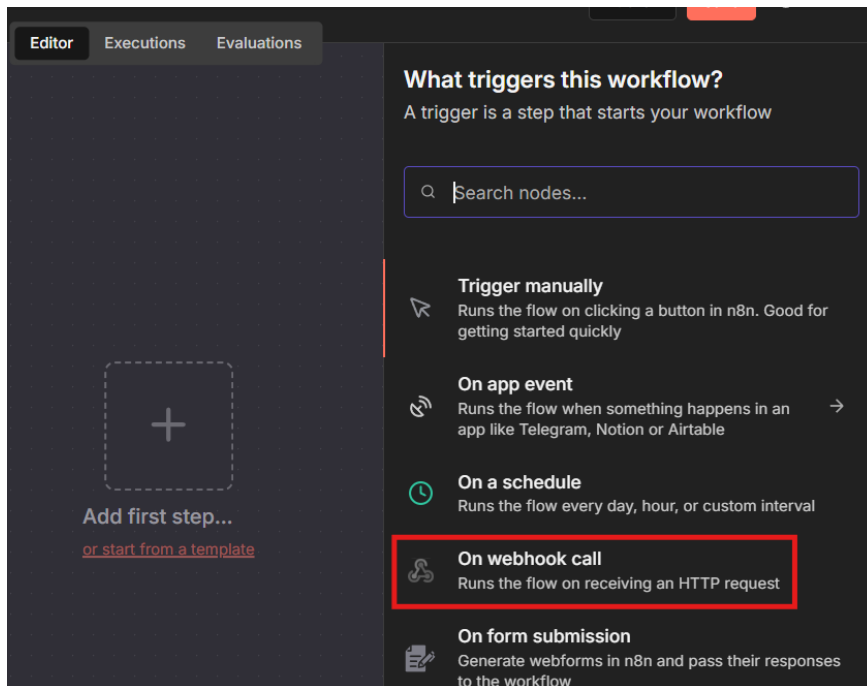
El primer flujo consultará si un servicio este activo haciendo ping a cada servicio y si este no responde se debe iniciar o reiniciar una VM o CT.

En esta imagen se puede visualizar un resumen del flujo completo y después se detallará cada uno de los nodos.



*Ilustración 67 Flujo reactivación de servicios*

El primer paso es seleccionar una acción que inicié el flujo, esto podrían ser varias acciones dependiendo del tipo de flujo que se quiere automatizar, en este caso el primer paso del flujo es utilizar el webhook para recibir una alerta de zabbix que nos indique el estado del servicio.



*Ilustración 68 Acción Inicio Flujo*

Después se configura un endpoint al que enviaremos la alerta de webhook.

The screenshot shows the 'Settings' tab for a Zabbix trigger. At the top, there are tabs for 'Parameters' and 'Settings', and a red button labeled 'Listen for test event'. Below this, there are two buttons: 'Test URL' and 'Production URL'. The 'Production URL' is selected, showing a POST request to 'http://192.168.83.46:5678/webhook/service\_state'. The 'HTTP Method' is set to 'POST'. The 'Path' is 'service\_state'. Under 'Authentication', it is set to 'None', with a tooltip showing 'Parameter: "authentication"'. The 'Respond' dropdown is set to 'Immediately'.

*Ilustración 69 Configuración primer Trigger*

Después nos dirigimos a zabbix para establecer la alerta, en el menú alerts y media types.

The screenshot shows the 'Media types' configuration page in Zabbix. The left sidebar has 'Media types' highlighted. The main area has a search bar, a 'Create media type' button, and filters for Name, Status (Any, Enabled, Disabled), and Display actions (All, All available, Specific). Below the filters is a table of media types:

Name	Type	Status	Used in actions	Details
Brevis one	Webhook	Disabled	4	Report not supported items, Report not supported low level discovery rules, Report problems to Zabbix administrators, Report unknown triggers
Discord	Webhook	Disabled	4	Report not supported items, Report not supported low level discovery rules, Report problems to Zabbix administrators, Report unknown triggers
Email	Email	Disabled	4	Report not supported items, Report not supported low level discovery rules, Report problems to Zabbix administrators, Report unknown triggers SMTP server: "mail.example.com", SMTP helo: "example.com", email: "zabbix@example.com"
Email (HTML)	Email	Disabled	4	Report not supported items, Report not supported low level discovery rules, Report problems to Zabbix administrators, Report unknown triggers SMTP server: "mail.example.com", SMTP helo: "example.com", email: "zabbix@example.com"
Event-Driven Ansible	Webhook	Disabled	4	Report not supported items, Report not supported low level discovery rules, Report problems to Zabbix administrators, Report unknown triggers

*Ilustración 70 Crear media type Zabbix*

Después configuramos la alerta con los parámetros que queremos enviar.

### New media type

? >

Media type Message templates Options

\* Name

Type

Name	Value	
URL	<input type="text" value="http://192.168.83.46:5678/webhoc"/>	<a href="#">Remove</a>
Subject	<input type="text" value="{TRIGGER.STATUS}: {TRIGGER."/>	<a href="#">Remove</a>
Message	<input type="text" value="{TRIGGER.ID}"/>	<a href="#">Remove</a>
EventID	<input type="text" value="{EVENT.ID}"/>	<a href="#">Remove</a>
Hostname	<input type="text" value="{HOST.NAME}"/>	<a href="#">Remove</a>
IP	<input type="text" value="{HOST.IP}"/>	<a href="#">Remove</a>

[Add](#)

\* Script

\* Timeout

*Ilustración 71 Parámetros alerta*

Después debemos crear una acción con información sobre nuestra alerta.

### New action

⌂ ✕

Action Operations 1

\* Default operation step duration

Operations

Steps	Details	Start in	Duration	Actions
1	<b>Send message to users:</b> Admin (Zabbix Administrator) via n8n Webhook vm state <b>Send message to user groups:</b> Zabbix administrators via n8n Webhook vm state	Immediately	Default	<a href="#">Edit</a> <a href="#">Remove</a>

[Add](#)

*Ilustración 72 Creación accion en Zabbix*

Esta acción la debemos asignar a todos nuestros equipos seleccionando todos y con la opción mass update.

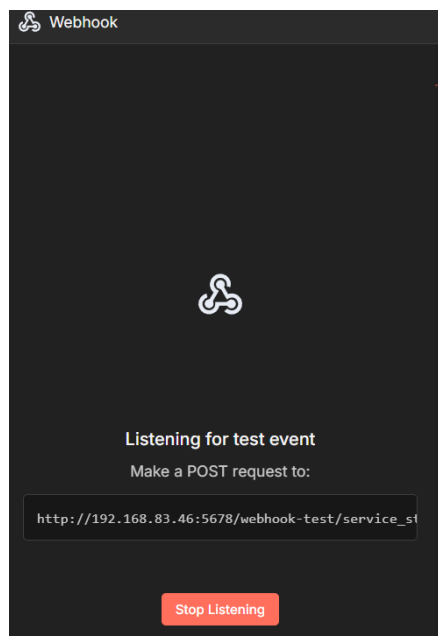
<input checked="" type="checkbox"/>	Name ▲	Items	Triggers	Graphs	Discovery	Web	Interface
<input checked="" type="checkbox"/>	CT-BDD	Items 102	Triggers 38	Graphs 19	Discovery 4	Web	192.168.83.57:10050
<input checked="" type="checkbox"/>	CT-n8n	Items 59	Triggers 24	Graphs 11	Discovery 3	Web	192.168.83.46:10050
<input checked="" type="checkbox"/>	CT-Nginx	Items 82	Triggers 33	Graphs 15	Discovery 4	Web	192.168.83.99:10050
<input checked="" type="checkbox"/>	CT-WebServer	Items 59	Triggers 24	Graphs 11	Discovery 3	Web	192.168.83.72:10050
<input checked="" type="checkbox"/>	VM-GLPI	Items 75	Triggers 30	Graphs 16	Discovery 3	Web	192.168.83.191:10050
<input checked="" type="checkbox"/>	VM-NextCloud	Items 107	Triggers 42	Graphs 20	Discovery 4	Web	192.168.83.58:10050
<input checked="" type="checkbox"/>	Zabbix server	Items 153	Triggers 83	Graphs 16	Discovery 6	Web	127.0.0.1:10050

7 selected

*Ilustración 73 Asignación a todos los hosts*

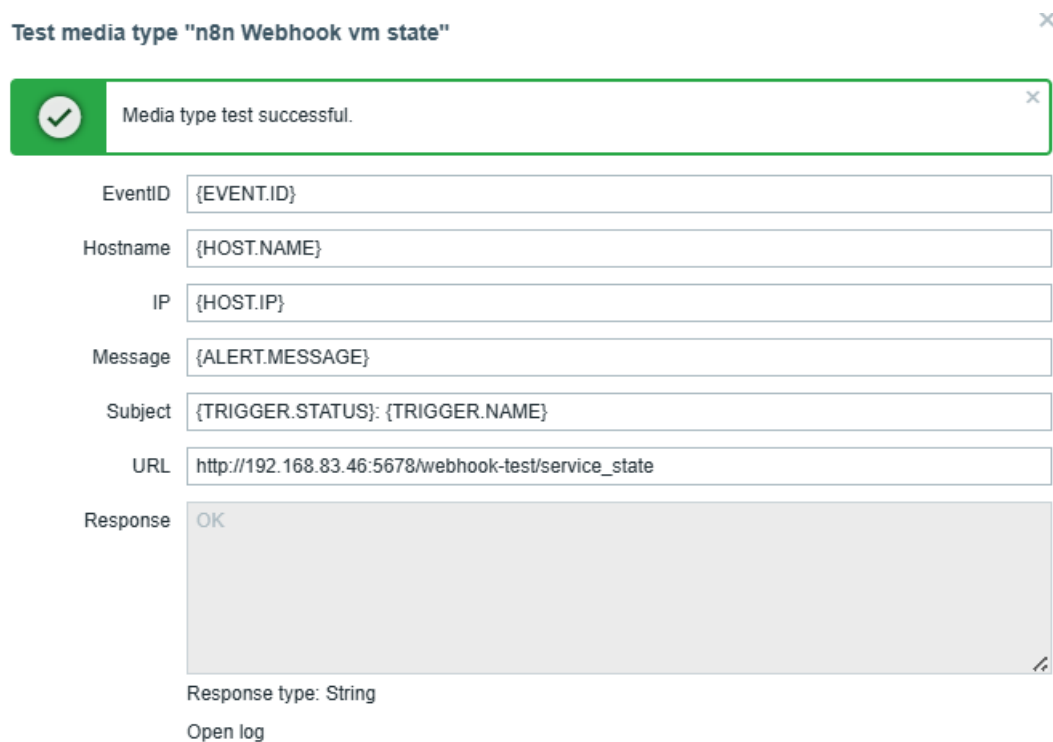
Con este paso Podemos ya probar si funciona el webhook de la siguiente forma.

En n8n en el trigger del webhook colocamos Test y empezará a funcionar esperando que zabbix envíe al webhook la alerta.



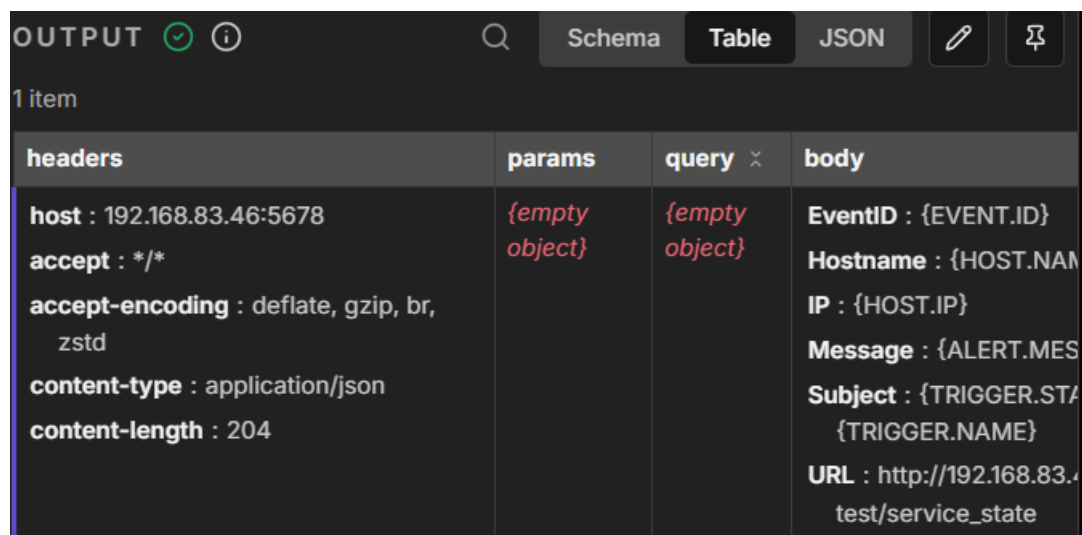
*Ilustración 74 Webhook test n8n*

Desde en zabbix podremos hacer una prueba con el trigger que creamos anteriormente.



*Ilustración 75 Prueba Webhook*

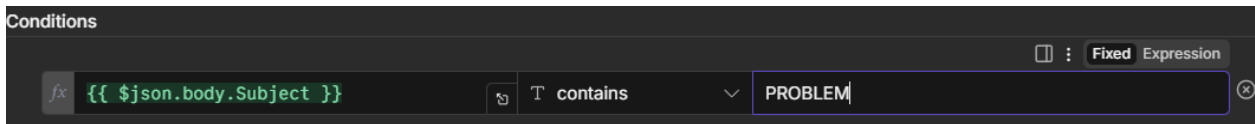
Y desde n8n podremos verificar que se recibió esta información.



*Ilustración 76 Revisión werbhook test en n8n*

El siguiente paso es definir un nodo filtro después del webhook para definir dos caminos, si lo que recibimos es un problema se toman ciertas acciones y si es un mensaje de recuperación o solo informativo se deberán tomar otras acciones.

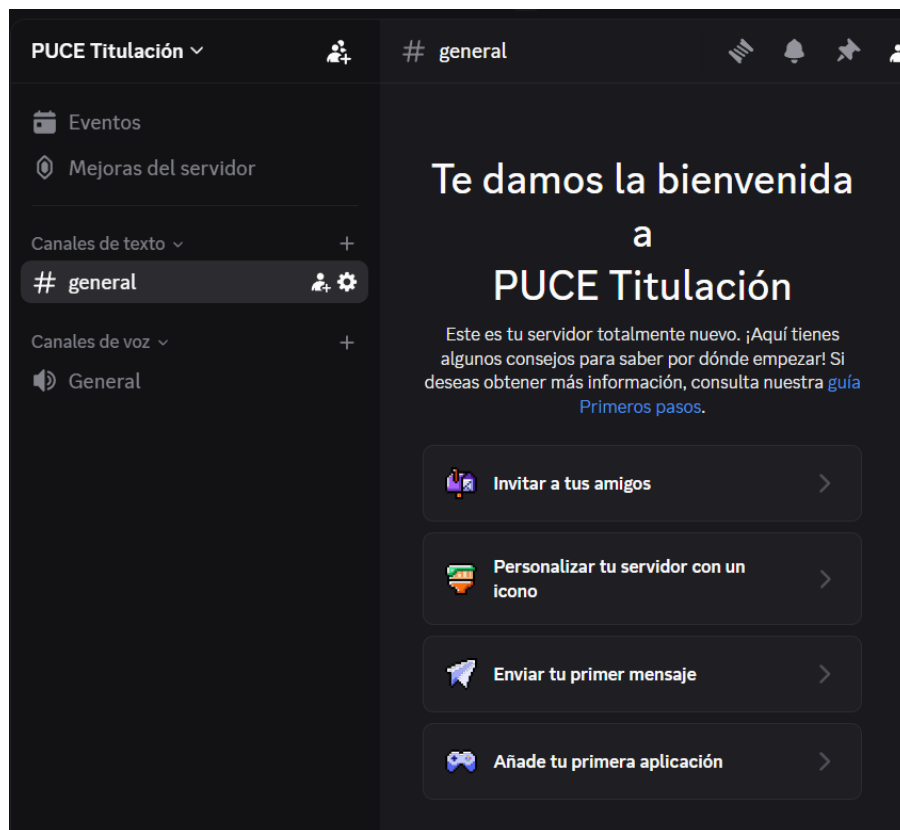
En este nodo definiremos una condición en este caso si el body de nuestro webhook contiene la palabra problema haremos que envíe una notificación mediante una aplicación de mensajería y además intente iniciar el nodo.



*Ilustración 77 Contenido webhook*

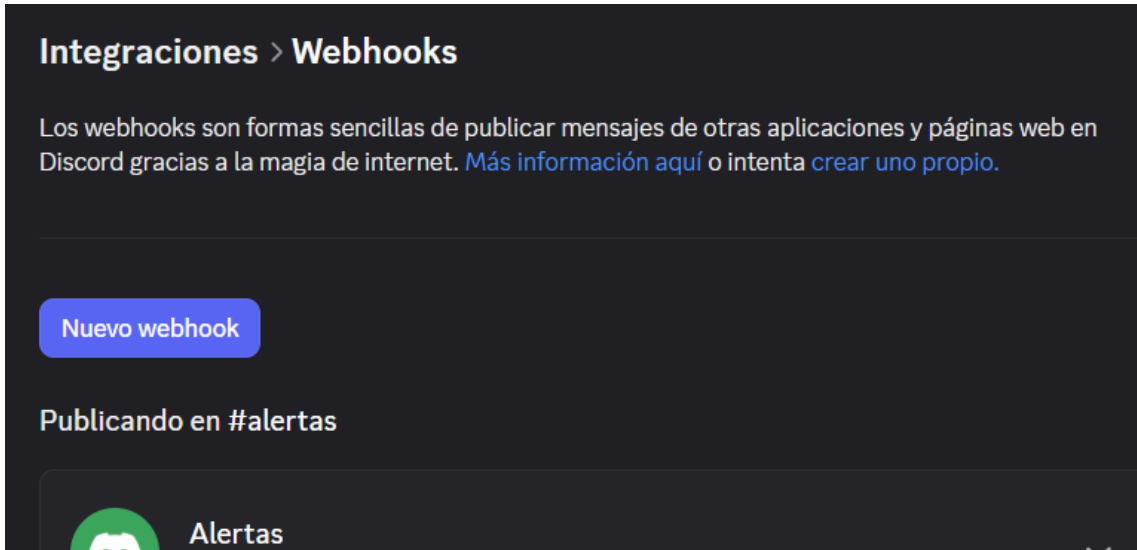
El siguiente paso en este flujo es enviar una notificación a la herramienta de comunicación que se desee, n8n tiene integraciones con whatsapp, telegram, Gmail y demás herramientas de mensajería. Para este proyecto de titulación se escogió la herramienta de Discord por su versatilidad y uso común en varias empresas que personalmente se ha observado.

Inicialmente, se creó un servidor en Discord que será el lugar en el que se enviarán las notificaciones de los servicios y se establecerán canales de texto para otras acciones.



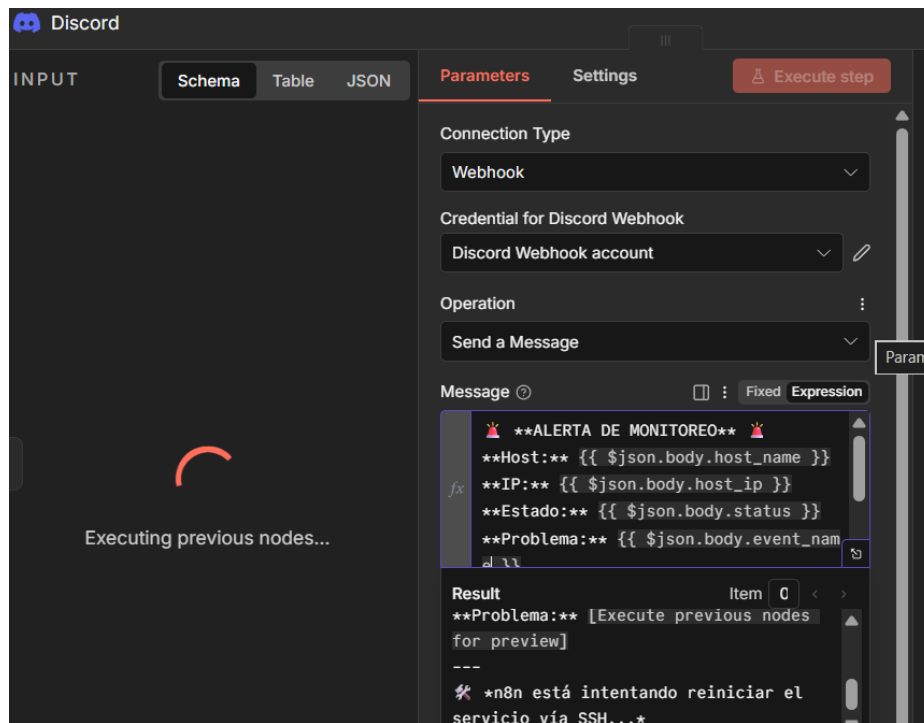
*Ilustración 78 Servidor Discord*

Aquí crearemos nuestro webhook al cual enviaremos la información en el apartado de integraciones.



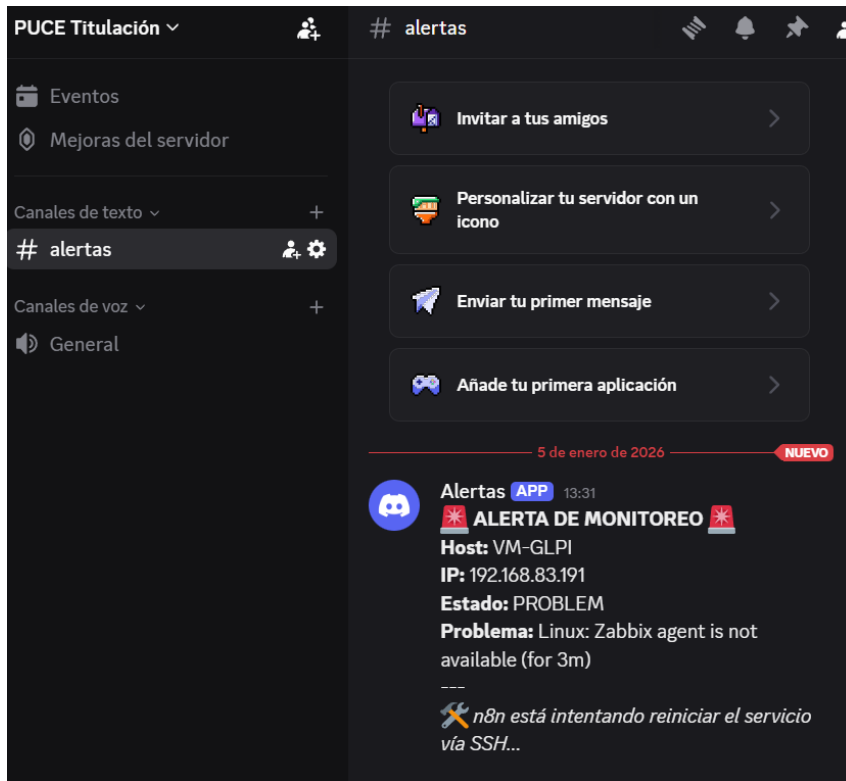
*Ilustración 79 WebHook Discord*

Copiaremos esta url del webhook que nos da discord y la colocaremos en la configuración de nuestro nodo de discord en n8n.



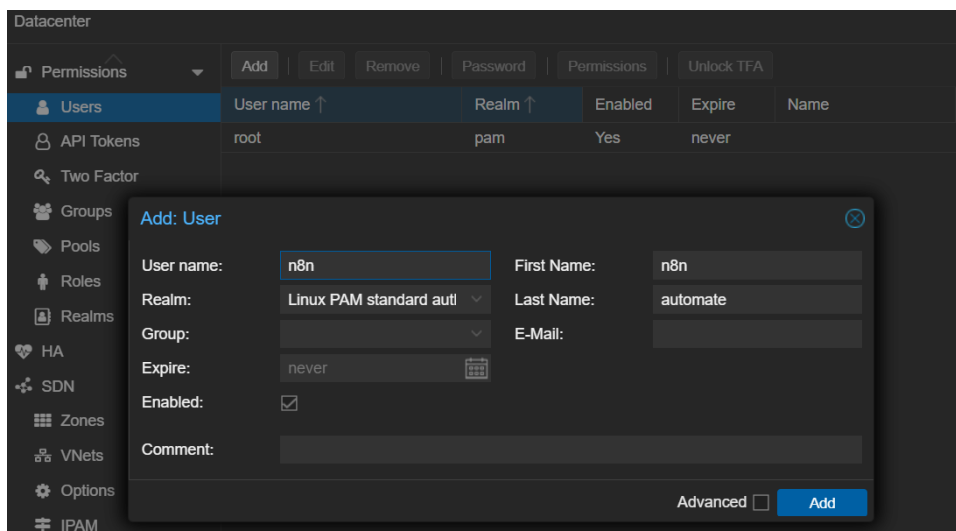
*Ilustración 80 Configuración nodo discord*

Una vez configurados los 3 nodos y simulando la caída de uno de los servicios en nuestra aplicación de mensajería recibiremos el mensaje de nuestro bot de discord.



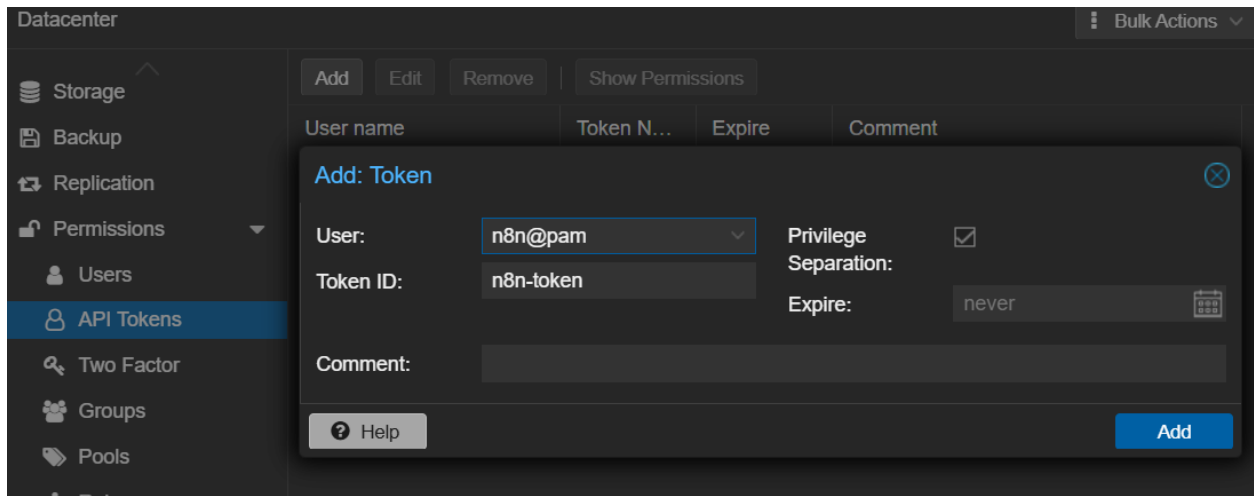
*Ilustración 81 Mensaje de alerta*

Una vez recibida la notificación y para automatizar el proceso debemos acceder al hipervisor mediante ssh e iniciar la VM o CT que falló. El primer paso será crear un usuario en proxmox para después agregarlo en el nodo de n8n.



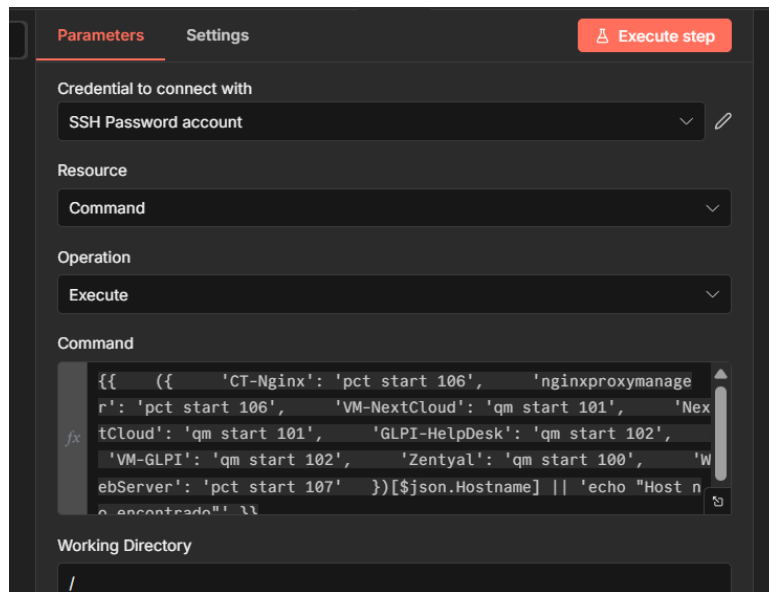
*Ilustración 82 Creación usuario proxmox*

Para poder darle acceso a n8n al hipervisor crearemos un API token.



*Ilustración 83 Proxmox API Token*

Después agregaremos el nodo ssh y lo configuramos con esas credenciales.



*Ilustración 84 Nodo SSH*

Utilizaremos el siguiente comando que lo que hace es usar `$json.Hostname`, n8n toma el nombre de la máquina que envió Zabbix en esa alerta específica y lo busca en la "lista" que creamos. Si falla el ID 106, n8n solo enviará el comando para el 106.

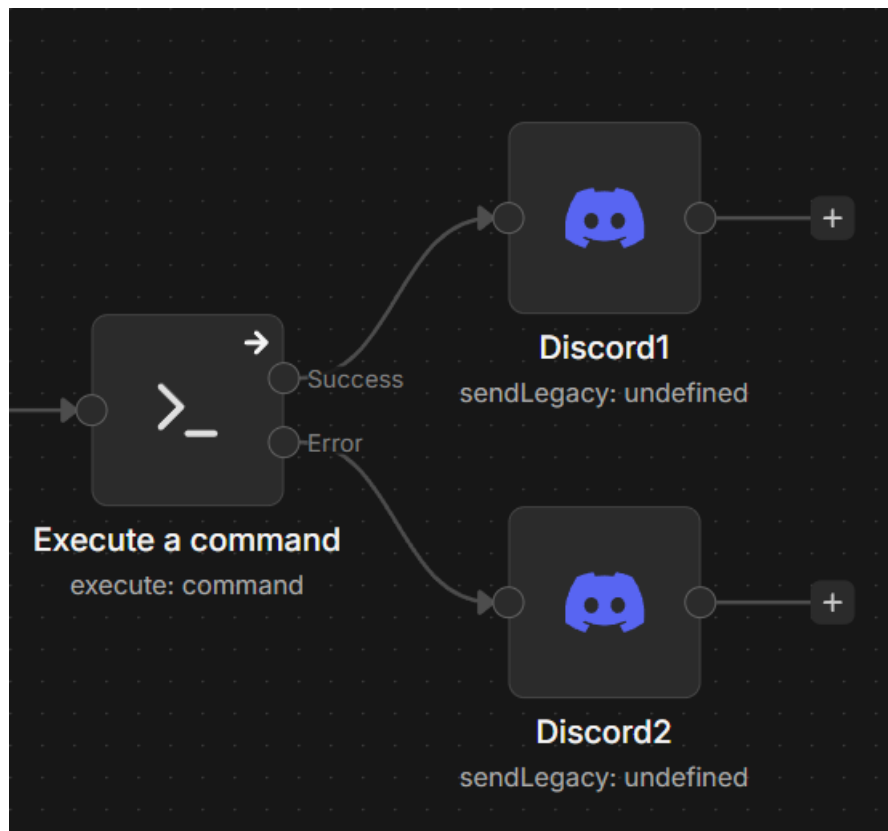
```

1  {{
2  ({
3    'CT-Nginx': 'pct start 106',
4    'nginxproxymanager': 'pct start 106',
5    'NextCloud': 'qm start 101',
6    'GLPI-HelpDesk': 'qm start 102',
7    'Zentyal': 'qm start 100',
8    'WebServer': 'pct start 107',
9    'postgresql': 'pct start 105',
10   'Zabbix': 'qm start 103',
11   'TrueNAS': 'qm start 108'
12  })[$json.Hostname] || 'echo "Host no configurado en la lista"'
13  }}

```

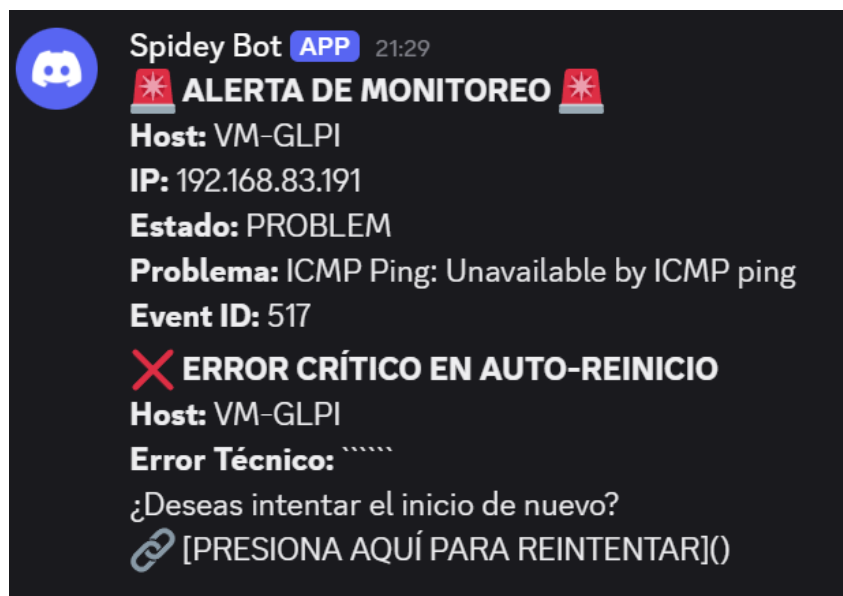
*Ilustración 85 Script para iniciar servicios*

En nuestros nodos podemos definir también varios flujos dependiendo del resultado de nuestro nodo actual, en este caso definiremos dos caminos distintos para dos notificaciones diferentes dependiendo del resultado de tratar de encender la máquina virtual.



*Ilustración 86 Varios resultados para un nodo*

Para asegurarnos que el flujo sea seguro si después de intentar iniciar falla por algún motivo se esperará 30 segundos antes de volver a intentar iniciar el servicio y después enviará otro mensaje con un enlace con el que el usuario podría volver a intentar iniciar el equipo.



*Ilustración 87 Mensaje reintentado*

#### 4.6.2. Flujo auto escalado vertical

El siguiente flujo hará que nuestros servicios puedan escalar según la demanda y recursos que se estén utilizando. El flujo empieza cuando Zabbix detecta que un servicio está con cierto recurso estresado por cierta cantidad de tiempo (3 minutos) En ese momento, Zabbix envía a n8n un mensaje mediante el webhook con los datos de la máquina que necesita asistencia.

Primero debemos establecer un límite de la cantidad de recursos que se puede aumentar para evitar problemas en cuanto al consumo general de los servicios.

Tipo	Servicio	vCPU Inicial	vCPU Máximo	RAM Inicial	RAM Máxima
VM	VM-Zentyal	2	4	4 GB	6 GB
VM	VM-NextCloud	2	4	4 GB	6 GB
VM	VM-HelpDesk	2	4	4 GB	6 GB
CT	CT-bdd	1	2	2 GB	3 GB
CT	CT-Web	1	2	1 GB	3 GB
CT	CT-Nginx	1	2	1 GB	3 GB

*Tabla 15 Límites escalado de servicios*

Una vez definidos los límites en la **Tabla 15 Límites escalado de servicios**, el proceso de automatización se ejecuta mediante una comunicación constante entre el sistema de monitoreo y el orquestador de flujos.

- a) **Disparador de evento (Zabbix Trigger):** Para evitar falsos positivos por picos de consumo momentáneos, se configuró un disparador en Zabbix que se activa

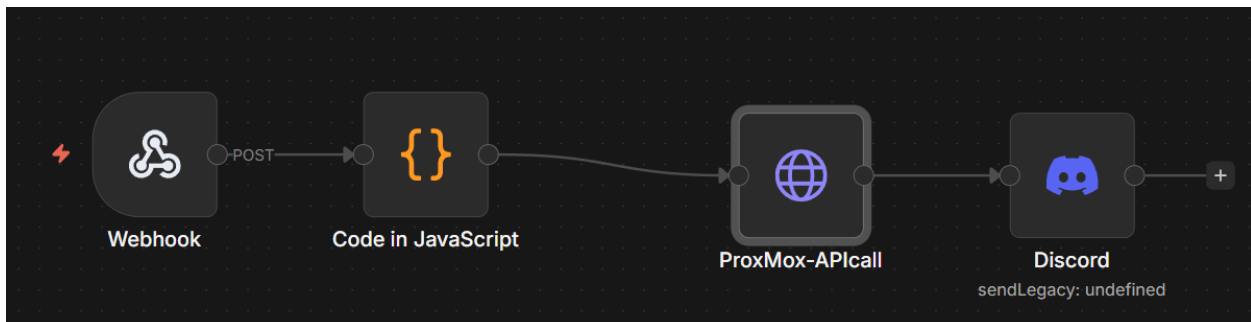
únicamente cuando el consumo de CPU o RAM supera el 85% durante un periodo sostenido de 3 minutos. Al cumplirse esta condición, Zabbix ejecuta un "Action" que envía una solicitud HTTP POST al Webhook de n8n.

- b) **Procesamiento y toma de decisiones (n8n Logic):** Al recibir los datos (Nombre del servicio, Recurso afectado y Valor actual), n8n realiza las siguientes validaciones:
- Consulta de límites: Verifica en su base de datos interna si el servicio afectado aún tiene margen de crecimiento según la **Tabla 15 Límites escalado de servicios**
  - Ejecución de escalado: Si hay margen, n8n se conecta a la API de Proxmox y envía el comando para modificar el parámetro de la VM o CT (por ejemplo, subir de 2GB a 4GB de RAM) sin necesidad de reiniciar el servicio (Hot-plug).
  - Notificación: Envía un mensaje automático vía discord informando que el servicio "X" ha sido escalado para evitar una caída.

#### 4.6.2.1. Flujo

El primer paso es crear nuestro media type que es configurar la data que se enviará al webhook de n8n.

En la siguiente imagen se podrá visualizar el flujo del autoescalado vertical y después se detallará su configuración.



*Ilustración 88 Flujo autoescalado vertical*

Media type **Message templates** Options

\* Name

Type

Parameters	Name	Value	
	URL	:f3-597f-465e-b3f9-c78844b29b13	<a href="#">Remove</a>
	vm_id	{HOST.HOST}	<a href="#">Remove</a>
	resource	{ITEM.KEY}	<a href="#">Remove</a>
	event_value	{ITEM.LASTVALUE}	<a href="#">Remove</a>
	<a href="#">Add</a>		

*Ilustración 89 Configuración media type flujo 2*

Al igual que el anterior la URL será la del webhook que nos da n8n.

El siguiente paso es configurar el trigger aquí seleccionaremos las acciones que encenderán este trigger y enviarán la data al webhook en este caso son los triggers de la utilización de CPU y RAM de todos los servicios.

New action

Action **Operations 1**

\* Name

Type of calculation  A or B or C or D or E or F or G or H or I or J or K

Conditions	Label	Name
	A	Trigger equals CT-n8n: Linux: High CPU utilization
	B	Trigger equals CT-Nginx: Linux: High CPU utilization
	C	Trigger equals CT-Nginx: Linux: High memory utilization
	D	Trigger equals VM-NextCloud: Linux: High CPU utilization
	E	Trigger equals VM-NextCloud: Linux: High memory utilization
	F	Trigger equals CT-BDD: Linux: High CPU utilization
	G	Trigger equals CT-BDD: Linux: High memory utilization
	H	Trigger equals CT-WebServer: Linux: High CPU utilization
	I	Trigger equals CT-WebServer: Linux: High memory utilization
	J	Trigger equals VM-GLPI: Linux: High CPU utilization
	K	Trigger equals VM-GLPI: Linux: High memory utilization
	<a href="#">Add</a>	

*Ilustración 90 Trigger Auto Escalado*

El siguiente paso es configurar los nodos en n8n, en este caso como ya conocemos los IDs de cada uno de nuestros servicios y si son VMs o CTs gracias a la tabla realizada **Tabla 13 IDs de servicios** utilizaremos esta información para distinguir la cantidad de recursos y el comando que se debe utilizar, ya que, proxmox utiliza distintos comandos para interactuar si es una VM o un CT.

Para este filtro utilizaremos el nodo code en el que podemos colocar código de programación para realizar la lógica de filtrado. El código es:

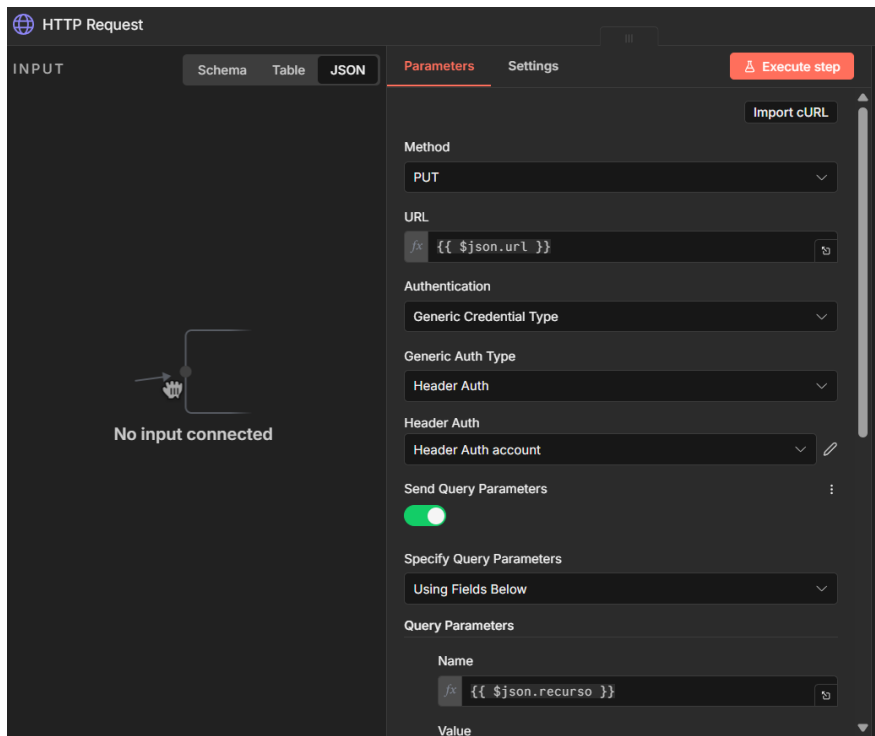
```

1  const body = $node["webhook"].json.body;
2  const vm_id = body.vm_id;
3  const isProblem = parseFloat(body.event_value) > 85;
4  const isRAM = body.resource.includes("memory");
5
6  // Definición de los servicios con sus IDs
7  const inventario = {
8    "100": { tipo: "qemu", nombre: "Zentyal" },
9    "101": { tipo: "qemu", nombre: "NextCloud" },
10   "102": { tipo: "qemu", nombre: "GLPI-HelpDesk" },
11   "103": { tipo: "qemu", nombre: "Zabbix" },
12   "104": { tipo: "lxc", nombre: "n8n" },
13   "105": { tipo: "lxc", nombre: "postgresql" },
14   "106": { tipo: "lxc", nombre: "nginx" },
15   "107": { tipo: "lxc", nombre: "WebServer" },
16   "108": { tipo: "qemu", nombre: "TrueNAS" }
17 };
18
19 const maquina = inventario[vm_id];
20
21 let config = {
22   tipoRecurso: isRAM ? "memory" : "cores",
23   valorNuevo: 0
24 };
25
26 // Aplicar límites
27 if (maquina.tipo === "qemu") {
28   // VM: RAM 4GB -> 6GB | CPU 2 -> 4 cores
29   config.valorNuevo = isRAM ? (isProblem ? 6144 : 4096) : (isProblem ? 4 : 2);
30 } else {
31   // CT: RAM 2GB -> 3GB | CPU 1 -> 2 cores
32   config.valorNuevo = isRAM ? (isProblem ? 3072 : 2048) : (isProblem ? 2 : 1);
33 }
34
35 return {
36   url: "https://192.168.83.100:8006/api2/json/nodes/pve/${maquina.tipo}/${vm_id}/config",
37   recurso: config.tipoRecurso,
38   valor: config.valorNuevo,
39   nombre: maquina.nombre,
40   accion: isProblem ? "ESCALADO" : "RESTABLECIMIENTO"
41 };

```

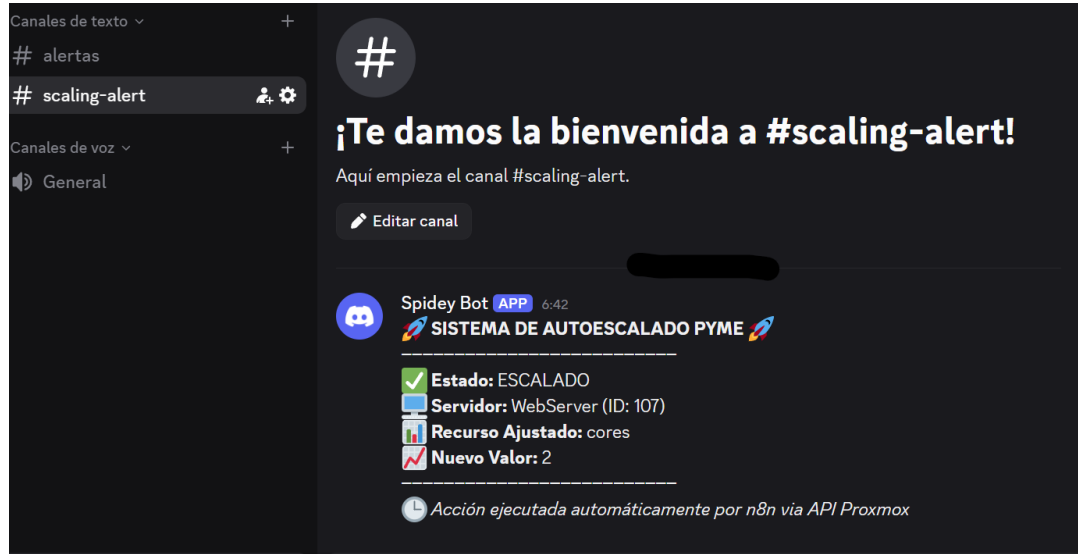
*Ilustración 91 Nodo Code*

El siguiente nodo será el HTTP request este será el encargado de realizar la instrucción en el API de Proxmox, dado que en el nodo anterior Code ya generamos la URL y los parámetros necesarios en este simplemente debemos pasarlos para que se envíen.



*Ilustración 92 Nodo HTTP Request*

Para informar al equipo del autoescalado se enviará una notificación a discord a otro canal de texto para tener un orden y los mensajes de alertas no se mezclen entre sí.



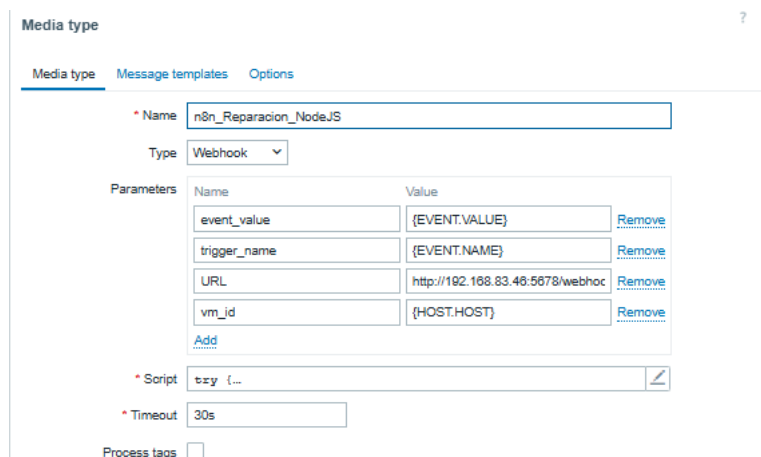
*Ilustración 93 Mensaje discord flujo 2*

### 4.6.3. Flujo auto reinicio servicios

Este flujo se enfocará en los servicios que corre cada VM o CT, principalmente se enfocará en los servicios de base de datos (postgresql), página web (pm2) y load balancer (nginx).

#### 4.6.3.1. Auto inicio de página web

Para este flujo utilizaremos igual una alerta de zabbix y el flujo de n8n, así quedarían los elementos.



*Ilustración 94 Alerta Zabbix página web*

Y en n8n este será el flujo completo.

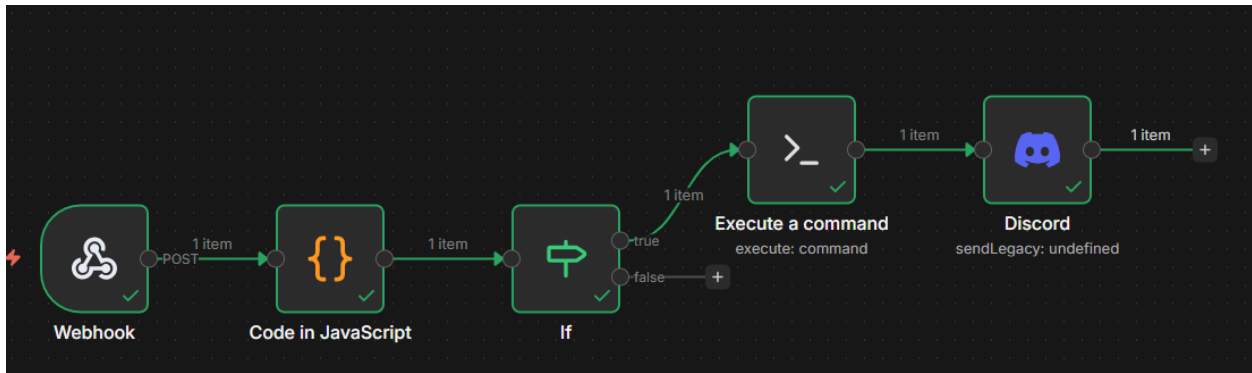


Ilustración 95 Flujo autoinicio node.js

## 4.7. Pruebas

### 4.7.1. Pruebas Flujo 1

Para probar este flujo realizaremos un caso un poco extremo que es apagar manualmente un servicio, lo haremos con el fin de activar el trigger de zabbix y que empiece el flujo de restauración de servicios.

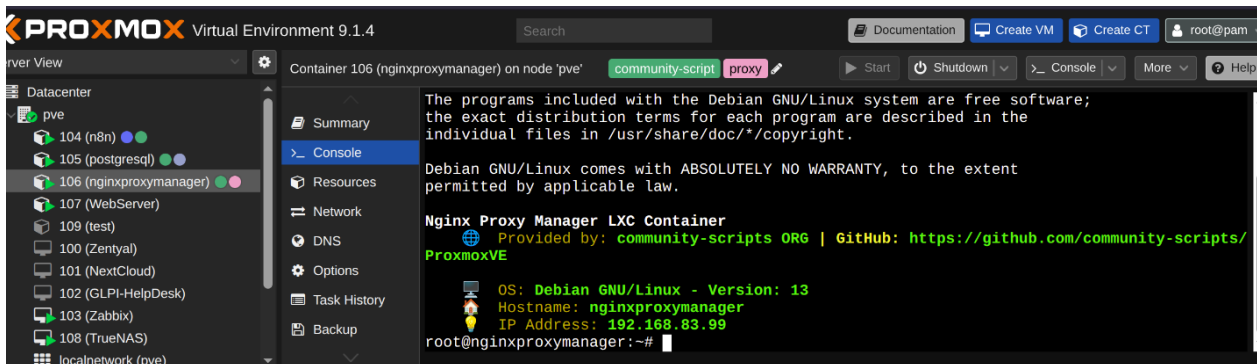


Ilustración 96 Caída de un servicio

Después de apagar el servicio podremos verificar la notificación de zabbix en su dashboard.

Time	Info	Host	Problem • Severity	Duration	Update	Actions	Tags
12:40:22 PM	CT-WebSe	WebSe	ICMP Ping: Unavailable by ICMP ping	5s	Update	1	class: network component: health component: network

Ilustración 97 Revisión en Zabbix

Según el flujo lo primero que debe hacer es notificar al canal de discord correspondiente para después tratar de iniciar el servicio nuevamente. En la imagen anterior podemos verificar la hora en que se ejecutó la acción y la hora en el sistema notifico al usuario y realizo el auto encendido, se puede observar que todo el flujo sea realizo en segundos.



*Ilustración 98 Alerta flujo 1 en Discord*

Al finalizar el servicio estará nuevamente disponible y funcional.

#### 4.7.1.1. Resumen Flujo 1

En esta prueba se simuló la caída completa de un servicio apagándolo por completo, en cuanto el servicio se detuvo, Zabbix activó la alerta, notificó a n8n y empezó el flujo de autoinicio, es interesante que el sistema no solo notificó al instante por Discord, sino que también se encargó de conectarse a Proxmox para encender la máquina de nuevo por su cuenta. Al final, el servicio volvió a estar operativo en pocos segundos, demostrando que la automatización funciona y que puede mantener los servicios funcionando sin necesidad de esperar a que alguien note la falla y la arregle “manualmente”.

### 4.7.2. Pruebas Flujo 2

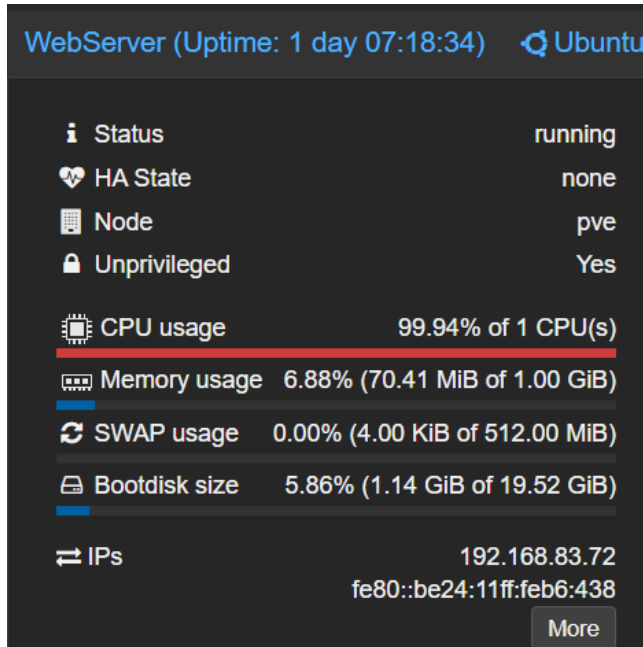
#### 4.7.2.1. Escalado Vertical (vCPUs)

Para probar este flujo utilizaremos dos comandos de Linux que nos sirven para estresar el CPU o RAM por cierta cantidad de tiempo los comandos son:

Comando	Qué hace
stress --cpu 1 --timeout 200s	Somete al servidor a una carga sintética de CPU superior al 90%
stress --vm 1 --vm-bytes 1.8G --timeout 200s	Somete al servidor a una carga sintética de

*Tabla 16 Comandos para estresas CPU y RAM*

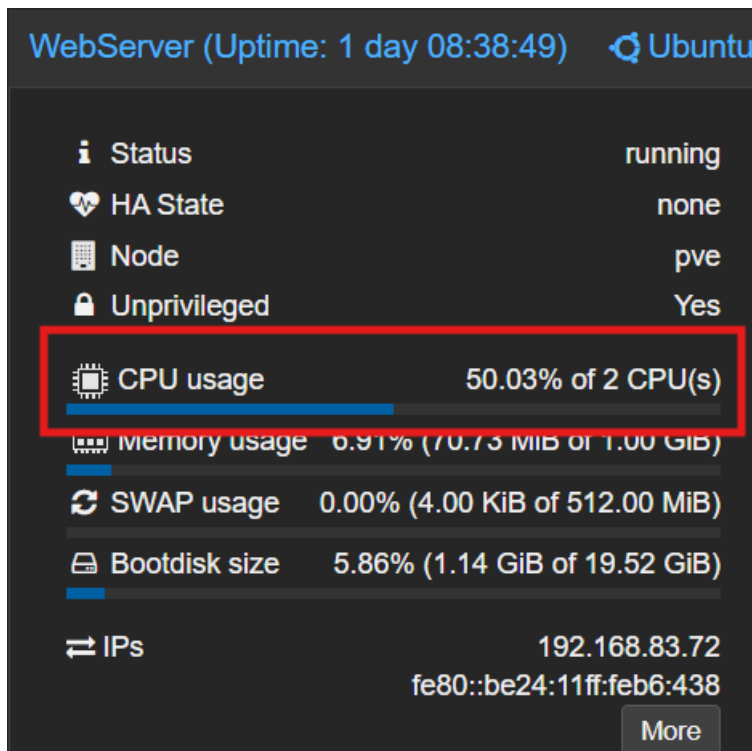
Con dichos comandos podremos verificar directamente en proxmox el porcentaje de CPU que se está utilizando.



*Ilustración 99 CPU estresado*

Como podemos ver el 1 core del CPU esta estresado después del flujo lo que podemos verificar es que realmente funcionó la actualización de recursos en este caso de cores.

Revisamos nuevamente y podemos ver que en efecto se aumentó la cantidad de cores de dicho servicio.



### Ilustración 100 Después del flujo - aumento cores

También recibiremos la notificación elaborada en discord, indicando el servidor, el recurso y el nuevo valor.

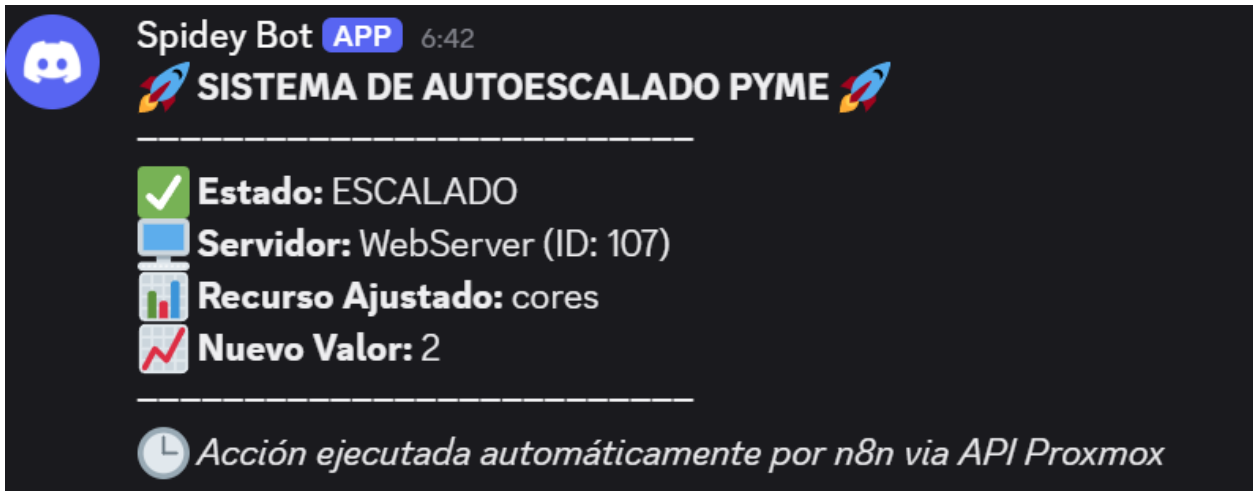


Ilustración 101 Mensaje discord autoescalado

#### 4.7.2.2. Escalado Vertical (RAM)

Para realizar estas pruebas se estresará con el comando mencionado en **Tabla 16 Comandos para estresas CPU y RAM** y podremos verificar un flujo similar al anterior pero el escalado será de memoria RAM.

Primero verificamos el valor original de la RAM en Proxmox.

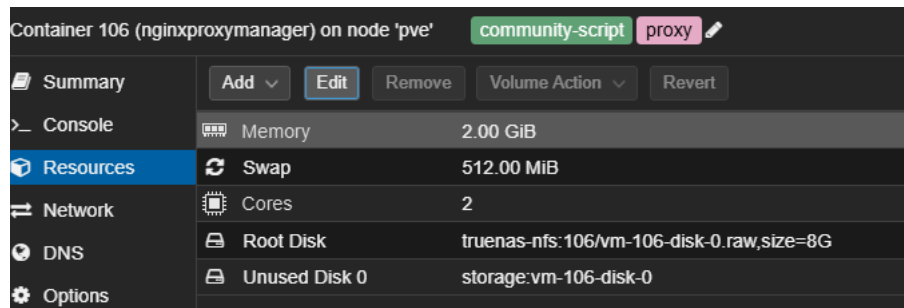
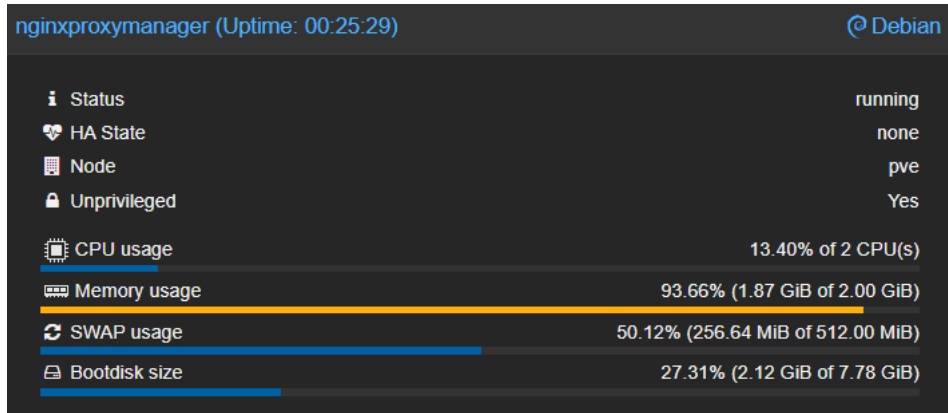


Ilustración 102 Estado CT previo comando estrés RAM

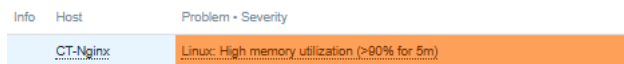
Después corremos el comando para estresar la RAM.

```
root@nginxproxymanager:~# stress --vm 1 --vm-bytes 1.8G --timeout 360s
stress: info: [571] dispatching hogs: 0 cpu, 0 io, 1 vm, 0 hdd
```



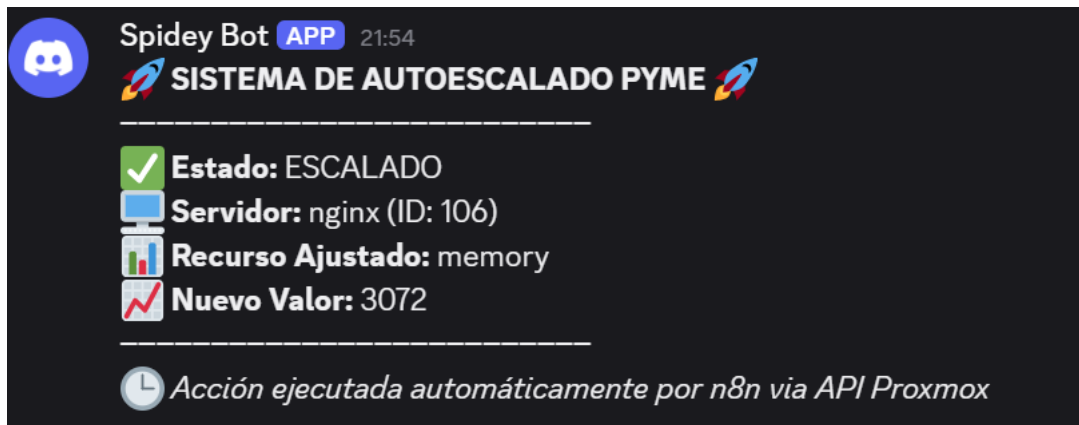
**Ilustración 103 Comando para estresar RAM**

En zabbix podremos ver al agente detectando un alto uso de la RAM.



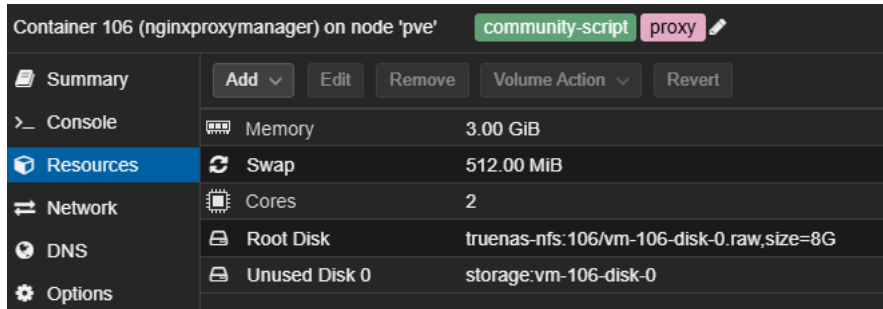
**Ilustración 104 Alto uso de RAM**

Inmediatamente después empezará el flujo enviando el mensaje a discord y el autoescalado de RAM.



**Ilustración 105 Notificación en discord flujo 2**

Y podremos hacer una verificación adicional en ProxMox.



*Ilustración 106 Verificación flujo 2 en Proxmox*

### 4.7.2.3. Resumen flujo autoescalado

Para probar que el sistema funciona, forzamos al servidor a trabajar al máximo usando un comando de estrés. Cuando el CPU pasó del 90%, Zabbix monitorea el evento y espera 3 minutos para confirmar que no era un pico pasajero. En ese momento, envía una notificación al webhook de n8n.

El flujo en n8n reconoció qué servidor era, entró a Proxmox usando su API y le aumentó los núcleos de 1 a 2 de forma automática. Al terminar, el sistema notificó por Discord que se realizó cierta acción sobre uno de los servicios. Todo esto se realizó en segundos y sin que se requiera la intervención de algún técnico o personal de TI.

## 4.7.3. Pruebas flujo 3

### 4.7.3.1. Flujo página web

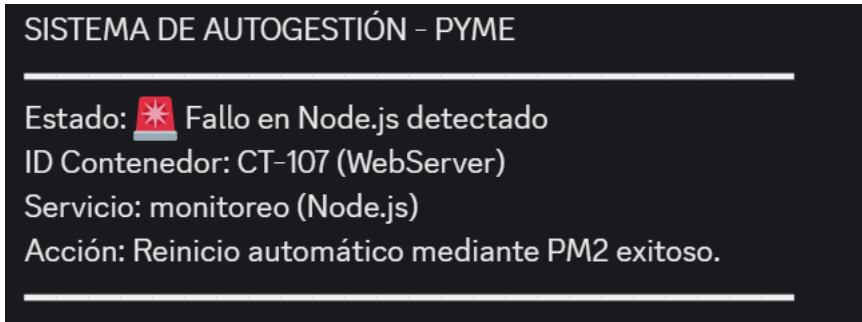
Para este flujo primero pararemos el servicio de nuestra página web con pm2.

```
remote@WebServer:~/monitoreo/server$ pm2 stop index.js --name "monitoreo"
[PM2] Applying action stopProcessId on app [index.js](ids: [ 0 ])
[PM2] [monitoreo](0) ✓
```

id	name	namespace	version	mode	pid	uptime	0	status	cpu	mem	user	watching
0	monitoreo	default	1.0.0	fork	0	0	0	stopped	0%	0b	remote	disabled

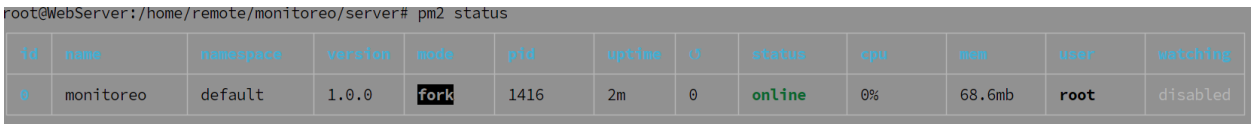
*Ilustración 107 Parar servicio web*

Una vez se detecte que el servicio se dejó de funcionar zabbix enviará la notificación a n8n y empezará el flujo, en este caso iniciará el servicio y enviará una notificación a discord.



*Ilustración 108 Mensaje en discord flujo 3.1*

Y podremos utilizar el comando pm2 status para verificar que si se inicio el servicio.



*Ilustración 109 Verificación estado servicio web flujo 3.1*

**4.8. Resumen pruebas**

Caso (flujo)	Condición / Umbral (Zabbix)	Acción automática ejecutada	Tiempo de confirmación (Zabbix)	Tiempo de ejecución	Resultado observado	Evidencia (ilustraciones)
<b>Flujo 1 – Reactivación</b> (caída total de servicio/VM)	Contenedor o Máquina virtual apagad (alerta Zabbix)	Notifica a n8n y ejecuta flujo de encendido en Proxmox + aviso por Discord	<i>Aproximadamente 10s</i>	Menos de 5 segundos	Servicio vuelve a estar operativo	Alerta/Discord + verificación final: <b>Ilustración 76</b> y resumen del flujo
<b>Flujo 2 – Autoescalado vCPU</b>	CPU > (85% o 90%) durante 3 min	Incrementa núcleos vía API de Proxmox + aviso Discord	<b>3 min</b>	Menos de 5 segundos	Se confirma aumento de cores + mensaje de autoescalado	CPU estresado / después del flujo / Discord: <b>Ilustraciones 77–79</b>
<b>Flujo 2 – Autoescalado RAM</b>	RAM > (85% o 90%) sostenido 3 min	Incrementa RAM vía API + aviso Discord	<b>3 min</b> (misma política del trigger)	Menos de 5 segundos	Se confirma aumento de RAM en Proxmox	Alto uso RAM / Discord / verificación Proxmox: <b>Ilustraciones 82–84</b>
<b>Flujo 3 – Reinicio servicio web (pm2)</b>	Servicio web detenido (alerta Zabbix)	Inicia servicio (pm2) + notifica Discord	<i>Aproximadamente 10s</i>	<i>Menos de 10 segundos</i>	Servicio web vuelve a “ejecutarse” (pm2 status)	Paro servicio / Discord / verificación: <b>Ilustraciones 85–87</b>

*Tabla 17 Resumen pruebas flujos n8n*

**5. Conclusiones y Recomendaciones**

En este capítulo se darán conclusiones sobre todo el trabajo realizado, además se darán recomendaciones para asegurar un entorno autónomo que pueda ser implementando con diferentes tecnologías y características extras que se podrían utilizar en las herramientas de virtualización, monitoreo y automatización.

## 5.1. Conclusiones

Según el desarrollo de este proyecto de titulación, se puede concluir que es totalmente viable diseñar e implementar un entorno virtualizado que permita el monitoreo y la gestión automatizada de servicios de TI utilizando herramientas de código abierto y hardware con recursos moderados. El prototipo desarrollado demuestra que no es necesario contar con infraestructuras costosas ni con servicios de pago para lograr un entorno estable, escalable y funcional, especialmente enfocado en pequeñas y medianas empresas.

El uso de Proxmox VE como plataforma de virtualización permitió administrar de forma centralizada tanto máquinas virtuales como contenedores, logrando una mejor distribución de los recursos del servidor físico. Esta combinación facilitó la separación de servicios críticos y ligeros, optimizando el uso de CPU, memoria RAM y almacenamiento, lo que resulta clave para entornos donde los recursos son limitados.

El sistema de monitoreo implementado con Zabbix cumplió correctamente su función al supervisar el estado, disponibilidad y consumo de recursos de los distintos servicios del laboratorio. Gracias a esto, fue posible detectar fallas, sobrecargas y eventos críticos en tiempo real, evitando una gestión reactiva y permitiendo una respuesta más rápida ante incidentes. Además, el uso de la API de Zabbix permitió integrar la información de monitoreo en una interfaz web personalizada, facilitando la visualización del estado general del entorno.

Uno de los principales aportes del proyecto fue la integración del monitoreo con la automatización mediante la herramienta n8n. Los flujos implementados demostraron que es posible crear un entorno autónomo capaz de ejecutar acciones correctivas sin intervención humana, como el reinicio de servicios, el encendido de máquinas virtuales y el escalado automático de recursos ante picos de consumo. Esto reduce significativamente los tiempos de inactividad y mejora la disponibilidad de los servicios.

Las pruebas realizadas confirmaron que el prototipo responde de manera efectiva ante escenarios reales de falla y sobrecarga, cumpliendo con los objetivos planteados al inicio del proyecto. La automatización permitió disminuir el tiempo de respuesta frente a incidentes y mantener los servicios operativos, lo que representa una ventaja importante para las PYMEs que no cuentan con personal de TI dedicado de forma permanente.

Adicionalmente, el prototipo desarrollado permite evidenciar mejoras en indicadores de disponibilidad propuestos por el marco de referencia ITIL, específicamente el MTBF (Tiempo Medio Entre Fallas) y el MTRS (Tiempo Medio para Restaurar el Servicio). Gracias al monitoreo continuo y a la integración con procesos de automatización, el sistema es capaz de detectar problemas de manera temprana, lo que contribuye a disminuir el tiempo entre fallas de los servicios. De igual forma, el tiempo de restauración se reduce considerablemente, ya que acciones como el reinicio de servicios, encendido de máquinas virtuales o escalado de recursos se ejecutan de forma automática y en pocos segundos.

Es importante mencionar que no se presentan valores numéricos específicos para los indicadores MTBF y MTRS, debido a que el entorno implementado corresponde a un laboratorio de pruebas y no a un sistema en producción con datos históricos prolongados. Adicionalmente, en el contexto de las pequeñas y medianas empresas, no siempre se cuenta con un equipo de TI dedicado que permita realizar una comparación directa entre la atención manual de incidentes y la gestión automatizada. Factores como la disponibilidad del personal, la carga de trabajo, la atención a múltiples tareas y los horarios laborales influyen directamente en el tiempo de respuesta ante una alerta, lo que dificulta una medición objetiva y constante. En este sentido, la automatización propuesta permite garantizar una disponibilidad continua 24/7, reducir costos operativos asociados al soporte permanente y minimizar el impacto del error humano, mejorando de forma significativa la estabilidad y capacidad de recuperación de los servicios.

Finalmente, se concluye que este prototipo puede servir como una base sólida para la implementación de infraestructuras de TI autogestionadas, no solo en PYMEs, sino también en entornos educativos, laboratorios de pruebas y equipos de desarrollo. El proyecto demuestra que la combinación de virtualización, monitoreo y automatización es una solución práctica, accesible y alineada con las necesidades actuales de las organizaciones.

## **5.2. Recomendaciones**

Se recomienda aprovechar al máximo las herramientas que ofrecen los servicios seleccionados, ProxMox permite realizar varias cosas de forma automática que son esenciales para empresas y son sencillas de configurar, éstas pueden ser realizar respaldos periódicos tanto para las máquinas virtuales como para los contenedores, utilizando las herramientas de snapshots y backups.

Se recomienda también ampliar los flujos de automatización desarrollados en n8n, incorporando acciones adicionales como la creación automática de tickets en el sistema de mesa de ayuda cuando ocurra un incidente crítico, o la ejecución de tareas de mantenimiento programado, como limpieza de logs y verificación de servicios, lo que contribuiría a una gestión aún más autónoma del entorno.

Otra recomendación importante es fortalecer la seguridad del laboratorio virtual, integrando herramientas adicionales como firewalls más avanzados, sistemas de detección de intrusos o autenticación multifactor para el acceso a los servicios críticos. Esto ayudaría a proteger la infraestructura ante accesos no autorizados y posibles ataques.

A nivel de monitoreo, se sugiere configurar más métricas y umbrales personalizados según el comportamiento real de los servicios, así como incluir monitoreo de disponibilidad desde ubicaciones externas, lo que permitiría detectar problemas de conectividad o acceso desde el punto de vista del usuario final.

Otra de las principales recomendaciones para asegurar la alta disponibilidad es incluir los nodos que sean posibles a la infraestructura, una de las razones por las que se eligió proxmox como hipervisor es la facilidad con la que se puede realizar un clúster, es decir, incluir varios servidores físicos que puedan manejar más servicios y en caso de que alguno falle este pueda ser replicado en otro nodo. Esta recomendación es especialmente útil para empresas que planean crecer o que requieren mayor tolerancia a fallos.

Finalmente, se recomienda documentar y capacitar al personal encargado de la administración del entorno, incluso si no cuenta con un perfil técnico avanzado. Una correcta documentación y capacitación facilitará el mantenimiento del sistema, permitirá su replicación en otros escenarios y asegurará que el entorno autónomo pueda ser adaptado a diferentes tecnologías y necesidades futuras.

## 6. Referencias Bibliográficas

¿Qué es la virtualización? - Explicación de la virtualización de la computación en la nube - AWS. (2022). Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/virtualization/>

¿Qué es Nextcloud? La solución en la nube. (2025, November 12). IONOS Digital Guide. <https://www.ionos.com/es-us/digitalguide/servidores/herramientas/que-es-nextcloud/>

¿Qué es un webhook y para qué sirve? | Automatización integral empresarial de la mano de Red Hat. (2024). Redhat.com. <https://www.redhat.com/es/topics/automation/what-is-a-webhook>

¿Qué es una máquina virtual? Usos y ventajas de las máquinas virtuales | Google Cloud. (2025). Google Cloud. <https://cloud.google.com/learn/what-is-a-virtual-machine?hl=es>

¿Qué son los contenedores Linux o LXC (Linux Containers)? - Qindel: Consultoría IT. (2023, June 26). Qindel: Consultoría IT. <https://www.qindel.com/que-son-los-contenedores-linux-o-lxc-linux-containers/>

2 Qué es Zabbix. (2025). Zabbix.com. <https://www.zabbix.com/documentation/current/es/manual/introduction/about>

Amazon Web Services. (2026, January 12). ¿Qué es una interfaz de programación de aplicaciones (API)? Amazon Web Services, Inc. <https://aws.amazon.com/es/what-is/api/>

Ansible: ¿qué es y cómo funciona? (2024). Redhat.com. <https://www.redhat.com/es/topics/automation/learning-ansible-tutorial>

Aranda, M. (2019, June 26). ¿Qué es ITIL® 4? Aranda Software. <https://arandasoft.com/blog/itil-4/>

Auxilion. (2024). Managed It Services For Small Businesses: Boost Your Business with Expert IT Support Services! Auxilion.com. <https://www.auxilion.com/insights/do-small-businesses-need-managed-it-services>

Boada, D. (2025, March 4). ¿Qué es n8n? Introducción a una herramienta de automatización del flujo de trabajo. Hostinger Tutoriales; Tutoriales Hostinger. <https://www.hostinger.com/es/tutoriales/que-es-n8n>

Bram Suurd, & Community Scripts. (2025). Proxmox VE Helper-Scripts. Proxmox ve Helper-Scripts. <https://community-scripts.github.io/ProxmoxVE/scripts?id=post-pve-install>

Bram Suurd, & Community Scripts. (2026). Proxmox VE Helper-Scripts. Proxmox ve Helper-Scripts. <https://community-scripts.github.io/ProxmoxVE/scripts?id=zabbix>

Cassia, A. M. (2024, May 4). ITIL 4 Foundation Guide (Spanish PDF) - Carlos Rivas Istacuy. Studocu. <https://www.studocu.com/pe/document/universidad-tecnologica-del-peru/inteligencia-de-negocios/itil-foundation-itil-4-edition-spanish-pdf-carlos-rivas-istacuy-flip-pdf-en-linea-flip-html-5-1-44/92940930>

Content Studio. (2025, October). Hipervisor tipo 1 frente a tipo 2: ¿Cuál es la diferencia? Purestorage.com; Pure Storage. <https://www.purestorage.com/la/knowledge/type-1-vs-type-2-hypervisor.html#:~:text=%C2%BFQu%C3%A9%20es%20un%20hipervisor%20tipo,gama%20de%20sistemas%20operativos%20invitados.&text=Los%20hipervisores%20tipo%201%20ofrecen,de%20plataformas%20de%20virtualizaci%C3%B3n%20dedicadas>

de, E. (2024, December 17). Nagios: Qué es, cómo funciona y beneficios para la monitorización empresarial. GoDaddy Resources - Spain. <https://www.godaddy.com/resources/es/crearweb/nagios-que-es>

Departamento de Consultoría. (2020, November 12). ¿Qué es ITIL y para que sirve? GlobalSuite Solutions. <https://www.globalsuitesolutions.com/es/que-es-itil-y-para-que-sirve/>

dknappettsft. (2025, August 13). Hyper-V virtualización en Windows Server y Windows. Microsoft.com. <https://learn.microsoft.com/es-es/windows-server/virtualization/hyper-v/overview>

Features. (2019). Proxmox. <https://www.proxmox.com/en/products/proxmox-virtual-environment/features>

<https://es.scribd.com/document/769736358/ITIL-Foundation-ITIL-4-Edition-Spanish>  
<https://worldaedit.com.mx/wp-content/uploads/2019/09/ITIL-4-Foundation-Material-Participante.pdf>

IBM. (2023, June 28). Supervisión de infraestructura. [Ibm.com](http://ibm.com/). <https://www.ibm.com/mx-es/think/topics/infrastructure-monitoring>

intel-hades-canyon-nuc. (2018, March 29). Tom's Hardware. <https://www.tomshardware.com/reviews/intel-hades-canyon-nuc-vr,5536.html>

La automatización: qué es y sus ventajas | Red Hat. (2022). Redhat.com. <https://www.redhat.com/es/topics/automation>

Network Configuration - Proxmox VE. (2025). Proxmox.com. [https://pve.proxmox.com/wiki/Network\\_Configuratio](https://pve.proxmox.com/wiki/Network_Configuratio)

Nginx Proxy Manager. (2016). Nginxproxymanager.com. <https://nginxproxymanager.com/guide/>

Noé González, & PrevenControl. (2019, April 23). AHP: un método para fortalecer la toma de decisiones en SST - PrevenControl. PrevenControl. <https://prevencontrol.com/prevenblog/ahp-un-metodo-para-fortalecer-la-toma-de-decisiones-en-sst/>

Pepe Lepu. (2025). ITIL Foundation ITIL 4 Edition (Spanish). Scribd. <https://es.scribd.com/document/769736358/ITIL-Foundation-ITIL-4-Edition-Spanish>

Peterka, P. (2024, April 15). SixSigma.us. <https://www.6sigma.us/six-sigma-in-focus/analytic-hierarchy-process-ahp/>

Procesador Intel® Core™ i7-8809G con gráficos Radeon™ RX Vega M GH (caché de 8 M, hasta 4,20 GHz). (2023). Intel. <https://www.intel.la/content/www/xl/es/products/sku/130409/intel-core-i78809g-processor-with-radeon-rx-vega-m-gh-graphics-8m-cache-up-to-4-20-ghz/specifications.html>

Quest Software. (2026). Quest.com. <https://www.quest.com/mx-es/solutions/active-directory/what-is-active-directory.aspx>

Santander Universidades. (2024, January 11). Herramientas de monitorización | Santander Open Academy. Santander Open Academy. <https://www.santanderopenacademy.com/es/blog/herramientas-de-monitorizacion.html>

Schneider, S. (2019, March 14). DHCP and DNS: Introduction, functions and options. Univention. [https://www.univention.com/blog-en/2019/03/brief-introduction-dhcp-dns/#What\\_is\\_DNS](https://www.univention.com/blog-en/2019/03/brief-introduction-dhcp-dns/#What_is_DNS)

Self-hosted cloud collaboration platform for home users - Nextcloud. (2025, November 3). Nextcloud. <https://nextcloud.com/home-users/>

Servicios de Mesa de Ayuda Informatica y Help Desk Informático en Ecuador. (2019, September 30). E-Open Solutions Soluciones IT. <https://www.eos.com.ec/help-desk-mesa-ayuda-informatica-ecuador/>

Susnjara, S., & Smalley, I. (2024, October 30). Hipervisores. [Ibm.com](http://ibm.com/). <https://www.ibm.com/mx-es/think/topics/hypervisors>

Todo lo que necesita saber sobre ESXi: ventajas y aplicaciones. (2022, April 12). IONOS Digital Guide. <https://www.ionos.com/es-us/digitalguide/servidores/know-how/esxi/>

TrueNAS CORE. (2026). GetApp. <https://www.getapp.es/software/2048361/truenas-core>

WebSphere Application Server. (2025, July). [Ibm.com](http://ibm.com/). <https://www.ibm.com/docs/es/was/9.0.5?topic=services-web>

What Is a Database? (2020, November 24). Oracle.com; Oracle.  
<https://www.oracle.com/latam/database/what-is-database/>

What Is Samba? (2023). Alibaba Cloud Community.  
[https://www.alibabacloud.com/blog/what-is-samba\\_599981](https://www.alibabacloud.com/blog/what-is-samba_599981)

Zabbix LLC. (s.f.). Disparadores. Documentación de Zabbix.  
<https://www.zabbix.com/documentation/current/es/manual/config/triggers>

Zabbix Manual. (2025). Zabbix.com.  
<https://www.zabbix.com/documentation/current/en/manual>

Zabbix LLC. (s.f.). Tipos de medios. Documentación de Zabbix.  
<https://www.zabbix.com/documentation/current/es/manual/config/notifications/media>

Zentyal 8.0 Official Documentation — Zentyal 8.0 Documentation. (2024). Zentyal.org.  
<https://doc.zentyal.org/en/>

Zentyal S.L. (s. f.). Zentyal, el servidor Linux para PYMEs: la alternativa en software libre a Windows Small Business Server. Recuperado de [[https://proactinfo.com/final\\_website/images/PDF/Zentyal\\_espanol.pdf](https://proactinfo.com/final_website/images/PDF/Zentyal_espanol.pdf)]([https://proactinfo.com/final\\_website/images/PDF/Zentyal\\_espanol.pdf?utm\\_source=chatgpt.com](https://proactinfo.com/final_website/images/PDF/Zentyal_espanol.pdf?utm_source=chatgpt.com))