



**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL**

**FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA INGENIERÍA EN TELECOMUNICACIONES**

TEMA:

**Diseño y análisis de un control de congestión para servidores en un
entorno virtualizado utilizando Zabbix y Python como herramientas
Open-Source**

AUTOR:

Monar Idrovo, Jair Cristopher

**Trabajo de titulación previo a la obtención del título de
INGENIERO EN TELECOMUNICACIONES**

TUTOR:

Ing. Medina Moreira, Washington Adolfo, PhD. D.

Guayaquil, Ecuador

02 de septiembre del 2024



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

CERTIFICACIÓN

Certificamos que el presente trabajo de titulación, fue realizado en su totalidad por **Monar Idrovo Jair Christopher**, como requerimiento para la obtención del título de **Ingeniero en Telecomunicaciones**.

TUTOR

f. _____

Ing. Medina Moreira, Washington Adolfo, Ph. D.

DIRECTOR DE LA CARRERA

f. _____

Ing. Bohórquez Escobar, Celso Bayardo, Ph. D.

Guayaquil, a los 02 del mes de septiembre del año 2024



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

DECLARACIÓN DE RESPONSABILIDAD

Yo, **Monar Idrovo, Jair Cristopher**

DECLARO QUE:

El Trabajo de Titulación, **Diseño y análisis de un control de congestión para servidores en un entorno virtualizado utilizando Zabbix y Python como herramientas Open-Source** previo a la obtención del título de **Ingeniero en Telecomunicaciones**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Titulación referido.

Guayaquil, a los 02 del mes de septiembre del año 2024

EL AUTOR

f. _____

Monar Idrovo, Jair Cristopher



UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

AUTORIZACIÓN

Yo, **Monar Idrovo Jair Cristopher**

Autorizo a la Universidad Católica de Santiago de Guayaquil a la **publicación** en la biblioteca de la institución del Trabajo de Titulación, **Diseño y análisis de un control de congestión para servidores en un entorno virtualizado utilizando Zabbix y Python como herramientas Open-Source**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, a los 02 del mes de septiembre del año 2024

EL AUTOR:

f.

Monar Idrovo, Jair Cristopher



UNIVERSIDAD CATÓLICA

DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA EN TELECOMUNICACIONES

TRABAJO DE INTEGRACIÓN CURRICULAR

INFORME SOFTWARE ANTIPLAGIO

REPORTE DE COMPILATIO

CERTIFICADO DE ANÁLISIS
registro

TrabajoFinal5- Jair Monar

4% Textos sospechosos

< 1% Similitudes
0% similitudes entre familias
= 1% entre las fuentes mencionadas

0% Idiomias no reconocidos

3% Textos potencialmente generados por la IA

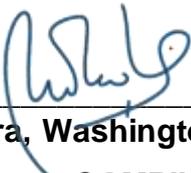
Nombre del documento: TrabajoFinal5- jair Monar.pdf
ID del documento: a640d8f9145eae3748d1d7e00cc411d81561d1e19e3
Tamaño del documento original: 4.22 MB
Autores: []

Depositante: Washington Adolfo Medina Moreira
Fecha de depósito: 21/01/2024
Tipo de carga: interface
fecha de fin de análisis: 22/01/2024

Número de palabras: 12.805
Número de caracteres: 89.134

Ubicación de las similitudes en el documento:

Reporte de Compilatio del trabajo de titulación de la Carrera Telecomunicaciones denominado: **“Diseño y análisis de un control de congestión para servidores en un entorno virtualizado utilizando Zabbix y Python como herramientas Open-Source”**, del estudiante **Monar Idrovo, Jair Cristopher** se encuentra al 4% de coincidencias.

f. 
Ing. Medina Moreira, Washington Adolfo, PhD. D.
Revisor - COMPILATIO

AGRADECIMIENTO

Primeramente, agradezco a Dios y a mi familia por haberme guiado a aconsejado en esta etapa de mi vida muy importante. Agradezco todo el apoyo que he recibido por parte de mi familia en especial de mis padres quienes han sido un pilar y una guía fundamental en mi vida, también agradezco a todos los profesores que me han brindado su sabiduría mediante la experiencia que han tenido en su trayecto como profesionales. Este logro no lo hubiera hecho sin el apoyo de todos ustedes, mi más sincero agradecimiento a todos.

DEDICATORIA

Dedicado a mis padres, Lenin y María quienes me han brindado su apoyo incondicional en todo este largo trayecto para poder llegar a ser un profesional. También agradezco todo el apoyo que mi familia y amigos me han brindado para poder culminar esta etapa de mi vida, la cual ha sido muy importante para mí debido a todas las buenas experiencias compartidas con todos ellos. Todo el esfuerzo realizado va dedicado a todos ellos, lo que hicieron posible que llegara a donde estoy ahora, muchas gracias.

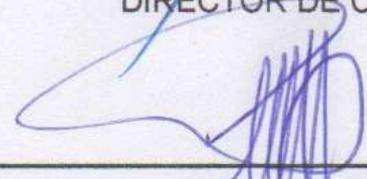


**UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA EN TELECOMUNICACIONES**

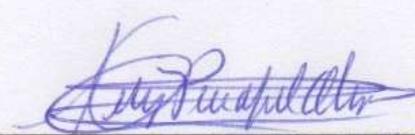
TRIBUNAL DE SUSTENTACIÓN

f. 

ING. BOHÓRQUEZ ESCOBAR, CELSO BAYARDO, Ph. D.
DIRECTOR DE CARRERA

f. 

Ms. C. UBILLA GONZALEZ, RICARDO XAVIER
COORDINADOR DEL ÁREA

f. 

ING. PEÑAFIEL OLIVO, KETY JENNY, MBA.
OPONENTE

ÍNDICE GENERAL

RESUMEN.....	XIII
ABSTRACT.....	XIV
CAPÍTULO 1: CONSIDERACIONES GENERALES.....	2
1.1 INTRODUCCIÓN	2
1.2 ANTECEDENTES	3
1.3 JUSTIFICACIÓN	3
1.4 PLANTEAMIENTO DEL PROBLEMA.....	4
1.5 OBJETIVOS.....	5
1.5.1 Objetivo General	5
1.5.2 Objetivos específicos	5
1.6 HIPÓTESIS	6
1.7 METODOLOGÍA DE INVESTIGACIÓN	6
CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA.....	7
2.1 INTRODUCCIÓN A ZABBIX.....	7
2.1.1 Características de Zabbix	9
2.1.2 Arquitectura de Zabbix.....	11
2.1.3 Monitores de Zabbix.....	12
2.1.4 Iniciadores de Zabbix	14
2.1.5 Comandos de Zabbix	15
2.1.6 Acciones de iniciadores de Zabbix	16
2.2 INTRODUCCIÓN A SERVIDORES VIRTUALES	17
2.2.1 Requerimientos Generales	18
2.3 INTRODUCCIÓN A UBUNTU SERVER	20
2.3.1 Comandos en Linux	21
2.4 INTRODUCCIÓN A PYTHON.....	22
2.4.1 Librería OS	22
2.4.2 Librería psutil	23
2.4.3 Librería shutil.....	23
2.4.4 Librería subprocess.....	23
CAPÍTULO 3: APORTES DE LA INVESTIGACIÓN.....	24
3.1 CREACIÓN DEL AMBIENTE.....	25
3.2 INSTALACIÓN Y CONFIGURACIÓN DE HERRAMIENTAS	26

3.2.1	Chrony	26
3.2.2	Dependencias de Zabbix	27
3.2.3	Servicio de base de datos para Zabbix	28
3.2.4	Configuración de php	30
3.2.5	Configuración de Zabbix	33
3.3	DESARROLLO DE SCRIPTS	35
3.3.1	Script para sobrecarga de CPU.....	36
3.3.2	Script para sobrecarga de RAM	37
3.3.3	Script para bajo espacio en disco.....	39
3.3.4	Creación de monitores para control de parámetros	41
3.3.5	Creación de iniciadores para control de parámetros	45
3.3.6	Creación de comandos.....	49
3.3.7	Creación de acciones con iniciadores	51
3.4	EVALUACIÓN Y PRUEBAS	53
CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES.....		57
4.1	CONCLUSIONES	57
4.2	RECOMENDACIONES	61
BIBLIOGRAFÍAS		62
ANEXOS		65

ÍNDICE DE FIGURAS

FIGURA 1 PANTALLA DE INICIO DE ZABBIX	8
FIGURA 2 INFORME DEL ESTADO ACTUAL DEL SISTEMA	9
FIGURA 3 OPCIONES DE VISUALIZACIÓN DE INICIADORES	10
FIGURA 4 RED SIMPLIFICADA HACIENDO USO DE ZABBIX.....	12
FIGURA 5 VISTA DE LA VENTANA DE CREACIÓN DE ÍTEMS DE ZABBIX.....	13
FIGURA 6 VISTA DE LA VENTANA DE CREACIÓN DE TRIGGERS DE ZABBIX.....	14
FIGURA 7 VISTA DE LA VENTANA DE CREACIÓN DE COMANDOS DE ZABBIX.....	16
FIGURA 8 VISTA DE LA VENTANA DE CREACIÓN DE ACCIONES DE INICIADORES DE ZABBIX	16
FIGURA 9 VISTA DE LA VENTANA PARA CONFIGURAR OPERACIONES DE ACCIONES DE ZABBIX.....	17
FIGURA 10 REPRESENTACIÓN DE UN MODELO DE MÁQUINA VIRTUAL.....	18
FIGURA 11 DIAGRAMA DE UNA HYPER-V SIN ANIDAMIENTO.....	19
FIGURA 12 INTERFAZ DE COMANDO DE UBUNTU SERVER	21
FIGURA 13 DIAGRAMA DE BLOQUES DEL PROCESO	24
FIGURA 14 VISTA DE LA CONSOLA DE COMANDOS DE UBUNTU SERVER	26
FIGURA 15 VISTA POR CONSOLA DEL ESTADO DEL SERVICIO DE CHRONY	27
FIGURA 16 VISTA DE LA INSTALACIÓN DE LAS DEPENDENCIAS DE ZABBIX.....	28
FIGURA 17 ESTADO DEL SERVICIO DE BASE DE DATOS EN EL SERVIDOR UBUNTU.....	29
FIGURA 18 COMANDOS PARA CONFIGURACIÓN DE BASE DE DATOS PARA ZABBIX.....	30
FIGURA 19 CONTENIDO DEL ARCHIVO PHP.INI POR CONSOLA.....	30
FIGURA 20 CONFIGURACIONES DEL ARCHIVO PHP.INI	31
FIGURA 21 CONTENIDO DEL ARCHIVO WWW.CONF POR CONSOLA.....	32
FIGURA 22 CONFIGURACIONES DEL ARCHIVO WWW.CONF	33
FIGURA 23 ESTADO DEL SERVICIO DE PHP 8.3 FPM EN EL SERVIDOR UBUNTU.....	33
FIGURA 24 COMANDOS PARA DESCARGAR E INSTALAR ZABBIX.....	34
FIGURA 25 CONTENIDO DEL ARCHIVO ZABBIX_SERVER.CONF POR CONSOLA	34
FIGURA 26 CONFIGURACIONES DEL ARCHIVO ZABBIX_SERVER.CONF.....	35
FIGURA 27 CONFIGURACIONES DE FIREWALL DEL SERVIDOR UBUNTU	35
FIGURA 28 DIAGRAMA DE FLUJO DE SCRIPT PARA SOBRECARGA DE CPU.....	36
FIGURA 29 DIAGRAMA DE FLUJO DE SCRIPT PARA SOBRECARGA DE RAM.....	38
FIGURA 30 DIAGRAMA DE FLUJO DE SCRIPT PARA ESPACIO EN DISCO LIMITADO	40
FIGURA 31 CONFIGURACIÓN DE ÍTEM PARA CPU.....	42
FIGURA 32 CONFIGURACIÓN DE ÍTEM PARA RAM	43
FIGURA 33 CONFIGURACIÓN DE ÍTEM PARA ESPACIO EN DISCO	44

FIGURA 34 LISTA DE MONITORES	45
FIGURA 35 CONFIGURACIÓN DE INICIADOR PARA CPU	46
FIGURA 36 CONFIGURACIÓN DE INICIADOR PARA RAM.....	47
FIGURA 37 CONFIGURACIÓN DE INICIADOR PARA ESPACIO EN DISCO	48
FIGURA 38 LISTA DE INICIADORES	48
FIGURA 39 COMANDO PARA CPU	49
FIGURA 40 COMANDO PARA RAM.....	50
FIGURA 41 COMANDO PARA ESPACIO EN DISCO.....	50
FIGURA 42 ACCIÓN PARA EJECUCIÓN DE COMANDO PARA CPU.....	51
FIGURA 43 ACCIÓN PARA EJECUCIÓN DE COMANDO PARA RAM	52
FIGURA 44 ACCIÓN PARA EJECUCIÓN DE COMANDO PARA ESPACIO EN DISCO	52
FIGURA 45 EJECUCIÓN DEL COMANDO STRESS-NG PARA SOBRECARGAR LA MEMORIA RAM.....	53
FIGURA 46 REGISTRO DE ACCIÓN EJECUTADA PARA CONTROLAR LA BAJA MEMORIA RAM DISPONIBLE	54
FIGURA 47 CORREO RECIBIDO POR ALERTA DE MEMORIA RAM INSUFICIENTE.....	54
FIGURA 48 EJECUCIÓN DEL COMANDO STRESS-NG PARA SOBRECARGAR LA CPU	54
FIGURA 49 REGISTRO DE ACCIÓN EJECUTADA PARA CONTROLAR EL ALTO CONSUMO DE CPU.....	55
FIGURA 50 CORREO RECIBIDO POR ALERTA DE ALTO CONSUMO DE CPU.....	55
FIGURA 51 REGISTRO DE ACCIÓN EJECUTADA PARA CONTROLAR EL ESPACIO EN DISCO INSUFICIENTE	56
FIGURA 52 CORREO RECIBIDO POR ALERTA DE ESPACIO EN DISCO INSUFICIENTE	56
FIGURA 53 GRAFICA DEL USO DE CPU CUANDO ESTÁ HABILITADA LA ACCIÓN DE CONTROL	58
FIGURA 54 GRAFICA DEL USO DE CPU CUANDO ESTA DESHABILITADA LA ACCIÓN DE CONTROL	58
FIGURA 55 GRAFICA DEL PORCENTAJE DE MEMORIA RAM DISPONIBLE CUANDO ESTÁ HABILITADA LA ACCIÓN DE CONTROL	59
FIGURA 56 GRAFICA DEL PORCENTAJE DE MEMORIA RAM DISPONIBLE CUANDO ESTA DESHABILITADA LA ACCIÓN DE CONTROL.....	59
FIGURA 57 GRAFICA DEL PORCENTAJE DE ESPACIO DISPONIBLE EN DISCO CUANDO ESTÁ HABILITADA LA ACCIÓN DE CONTROL	60
FIGURA 58 GRAFICA DEL PORCENTAJE DE ESPACIO DISPONIBLE EN DISCO CUANDO ESTA DESHABILITADA LA ACCIÓN DE CONTROL.....	61

RESUMEN

Este trabajo de integración curricular propone la creación de un diseño de un control de congestión para servidores mediante el uso de herramientas open-source para poder monitorear constantemente los componentes del sistema y tomar medidas cuando estos no estén en los rangos necesarios para un buen rendimiento del servidor. El diseño está enfocado en la creación de scripts mediante código en Python para poder gestionar los recursos de un servidor virtualizado de una forma automatizada mediante la integración con triggers de Zabbix los cuales nos permiten realizar acciones cuando un parámetro del sistema no se encuentre en los rangos establecidos, de esta manera podremos tomar las medidas necesarias accionando comandos que ejecuten tareas automatizadas que permitan gestionar los procesos que están causando una sobrecarga en el sistema. El enfoque se centra en destacar las diferentes funcionalidades que posee la herramienta de Zabbix, así como la configuración recomendada para sacar provecho del programa que junto con la integración con scripts de Python pueda ser una solución que brinde una ayuda para evitar problemas a futuro.

Palabras Claves: Zabbix, Diseño, Servidor Ubuntu, Códigos, Python, Servidores virtualizados, Automatización

ABSTRACT

This curricular integration work proposes the creation of a congestion control design for servers through the use of open-source tools to be able to constantly monitor the system components and take measures when they are not within the ranges necessary for good server performance. The design is focused on the creation of scripts using Python code to be able to manage the resources of a virtualized server in an automated way through integration with Zabbix triggers which allow us to perform actions when a system parameter is not within the established ranges, in this way we can take the necessary measures by activating commands that execute automated tasks that allow us to manage the processes that are causing an overload in the system. The focus is on highlighting the different functionalities that the Zabbix tool has, as well as the recommended configuration to take advantage of the program that together with the integration with Python scripts can be a solution that provides great help to avoid future problems.

Keywords: Zabbix, Design, Ubuntu Server, Codes, Python, Virtualized Servers, Automation

CAPÍTULO 1: CONSIDERACIONES GENERALES

1.1 Introducción

En la era actual, la creciente demanda de servicios y aplicaciones en la nube ha creado un interés sin precedentes en la gestión eficaz de los recursos de servidores virtualizados. La capacidad de gestión que tengan los sistemas para manejar grandes cantidades de datos sin perder su eficiencia se ha convertido en un requisito básico para satisfacer las necesidades de una sociedad cada vez más digital. En este contexto, la implementación de herramientas de monitoreo como Zabbix han sido de mucha utilidad para llevar una administración y control del sistema que permita que los servicios desplegados puedan operar con eficiencia y sin interrupciones.

La virtualización ha cambiado la forma en que las organizaciones administran los recursos de TI, brindando mayor flexibilidad y eficiencia (Arias Chaves, 2008). Pero esta flexibilidad también puede provocar congestión y problemas de congestión si no se gestiona adecuadamente. Este desafío exige soluciones innovadoras que puedan monitorear, detectar y mitigar automáticamente los problemas de congestión.

El enfoque de este proyecto está centrado en la gestión eficaz de los diversos parámetros que intervienen para que exista un buen rendimiento en el servidor, lo que nos permite ofrecer un servicio de calidad para los usuarios que hagan uso de los diferentes servicios que estén operando para lo cual mediante el uso de herramientas open-source como Zabbix y Python se puede obtener un monitoreo en tiempo real de los recursos que nos permite poder establecer medidas para mejorar la eficiencia en el instante que sea requerido. Este enfoque nos permite tener un control absoluto a nivel de configuración para poder establecer de acuerdo al criterio de los administradores de TI los diferentes parámetros tanto a nivel de monitoreo estableciendo límites en lo que es el consumo de recursos y en las medidas correctivas a nivel de código de Python (Zabbix, 2024).

1.2 Antecedentes

La llegada de las tecnologías de monitorización y virtualización de servidores ha cambiado drásticamente la gestión de recursos en los entornos de TI. La migración de datos y aplicaciones a la nube y el uso de herramientas de código abierto como Zabbix y Python han mejorado la flexibilidad y eficiencia en la gestión de servidores virtualizados.

En los últimos años, la virtualización de servidores ha permitido a las organizaciones trasladar las operaciones diarias de servidores físicos a servidores virtuales y aprovechar la escalabilidad, flexibilidad y gestión simplificada que brindan las soluciones de virtualización (VMblog, 2023). Además, la adopción de la virtualización ha aumentado las preocupaciones de seguridad en respuesta a un aumento significativo de los ciberataques y la necesidad de proteger los datos en un entorno distribuido (CheckPoint Research, 2022).

La automatización de procesos ha permitido que los sistemas crezcan a un ritmo exponencial lo que ha provocado que los grandes centros de procesamiento de datos tengan que adaptarse a toda la carga de trabajo que se requiere en la actualidad para lo cual Zabbix ha sido de gran utilidad para brindar un sistema de monitoreo capaz de proveer en tiempo real alertas preprogramadas que permiten a los administradores dar solución en caso que el sistema no esté en una buena condición (ServerWatch, 2023). Además, la integración con lenguajes de programación como Python ha permitido tener una valiosa herramienta para poder realizar tareas automatizadas mediante código para diversos sistemas mediante su gran variedad de librerías.

1.3 Justificación

En la actualidad los avances tecnológicos han permitido el crecimiento de los sistemas informáticos a la par que el número de personas que hacen uso de estos servicios por lo cual en algunos casos los recursos que poseen los servidores para poder procesar grandes cantidades de información pueden llegar a colapsar debido a la alta operatividad dependiendo del servicio que se brinde, por lo tanto es de suma importancia mantener un

monitoreo constante de todos los recursos del servidor para poder prevenir cualquier lentitud que pueda provocar algún colapso en el servicio que se está brindando.

Un sistema de monitoreo es de gran importancia a nivel de administración de TI por varias razones entre las cuales están el poder tener en tiempo real el estado o condición en la cual se encuentra nuestro sistema para tener reportes que nos permitan realizar análisis para aplicar mantenimientos preventivos en el sistema de ser requerido y aún más importante si es que la situación lo requiera aplicar alguna medida correctiva para de esa manera mejorar el rendimiento del mismo, para lo cual el uso de herramientas de código abierto como Zabbix y Python proporciona no sólo una solución rentable, sino también soluciones flexibles y adaptables a diferentes necesidades empresariales.

Los sistemas corporativos siempre pueden estar expuestos a cualquier tipo de baja por diversos factores por lo que la disponibilidad y eficiencia del servicio que proporcionen siempre debe ser de excelente calidad para lo cual la implementación de sistemas de monitoreo es muy esencial. Estos sistemas permiten supervisar y controlar que los servidores. Esto no sólo permite una gestión más eficiente de los recursos del servidor sino también da inicio a que se puedan aplicar diversos análisis con los datos recopilados para determinar alcances adicionales que permitan expandir la infraestructura cuando el sistema lo requiera. Las herramientas a utilizar en esta solución son gratuitas y poseen actualizaciones frecuentes que incorporan nuevas funcionalidades para lo cual son de gran utilidad para poder hacer uso de las mismas y realizar un sistema autónomo de control de congestión en servidores.

1.4 Planteamiento del problema

En un entorno virtualizado, la sobrecarga de recursos puede degradar significativamente el rendimiento del servidor, afectando así la disponibilidad y calidad de los servicios prestados. Este problema se agrava si no existen sistemas eficaces de seguimiento y mitigación. El desafío era crear un sistema que no sólo pudiera detectar rápidamente la sobrecarga de recursos, sino

también tomar acciones automatizadas para mitigar la sobrecarga y así optimizar el uso de los recursos del servidor.

Además, la ausencia de un sistema que proporcione registros de la actividad a nivel de recursos del equipo puede llegar a que el sistema tenga problemas de rendimiento sin tener conocimiento del mismo por la falta de alertas que nos permitan tomar medidas para corregir el inconveniente que se esté presentando en el menor tiempo posible y que este no genere que se tenga que dejar inactivado el servicio por la gran sobrecarga de recursos. Las herramientas de monitoreo tradicionales a menudo son inadecuadas para abordar la complejidad y la dinámica del entorno virtualizado actual.

Por lo tanto, existe una necesidad urgente de desarrollar una solución que integre capacidades avanzadas de monitoreo y automatización que puedan responder de manera proactiva a los indicadores de congestión y ajustar los recursos en consecuencia para garantizar la continuidad y eficiencia de los servicios de TI.

1.5 Objetivos

1.5.1 Objetivo General

Diseñar y analizar un sistema de control de congestión para servidores en un entorno virtualizado.

1.5.2 Objetivos específicos

- Configurar y desplegar Zabbix para el monitoreo de servidores virtualizados.
- Desarrollar scripts en Python para la detección y mitigación de congestión.
- Integrar Zabbix y Python para automatizar la respuesta a incidentes de congestión.
- Realizar pruebas de estrés y evaluar la eficacia del sistema de control de congestión.

1.6 Hipótesis

El sistema de control de congestión propuesto aumentará la comprensión práctica de la gestión y optimización de recursos de servidores virtuales, proporcionará habilidades técnicas avanzadas y mejorará la eficiencia operativa del servidor.

1.7 Metodología de Investigación

En el presente trabajo de investigación se va a elaborar un sistema de control para la gestión de recursos de un servidor virtualizado para lo cual se utilizare una máquina virtual en la que se levantara un servidor Ubuntu para la implementación y configuración de las herramientas requeridas, entre las cuales esta Zabbix la cual nos proporcionara el monitoreo de la actividad de nuestro servidor para poder tener registros que nos permitirán obtener un análisis detallado de los diversos parámetros relevantes para que el servicio que se esté brindando no se vea interrumpido.

Para el control de congestión del servidor se hará uso de scripts en Python que mediante la integración con Zabbix sean capaces de detectar y mitigar la congestión en tiempo real para liberar recursos que permitan que el sistema tenga un mejor rendimiento. Estos scripts se ejecutarán en un ambiente controlado para pruebas que permitan validar su eficacia frente a diversos escenarios, por lo cual también se tendrán alertas que permitan al administrador tener conocimiento de las acciones realizadas para mitigar la sobrecarga.

Para obtener los resultados de las pruebas se hará uso de la interfaz gráfica de Zabbix la cual nos permitirá recopilar y analizar todos los registros de manera ordenada y detallada para su presentación mediante comparaciones teniendo como ejemplo el estado del sistema sin ejecutar los scripts y con ejecución para validar la eficacia de los mismos en un entorno virtualizado.

CAPÍTULO 2: FUNDAMENTACIÓN TEÓRICA

2.1 Introducción a Zabbix

Zabbix es un software de código abierto que tiene como finalidad el monitoreo de numerosos parámetros de servidores, máquinas virtuales, aplicaciones, servicios y más. Además de ser uno de los softwares más utilizados para la implementación de sistemas de monitoreo de recursos en servidores o dispositivos de red debido a su compatibilidad con diversas plataformas lo que permite tener un mayor alcance para su uso en diferentes equipos y también posee la capacidad de generar informes estadísticos basados en los diferentes parámetros que se necesiten evaluar (Olups, 2016).

La expansión de los sistemas en la actualidad han abierto la necesidad de implementar herramientas de monitoreo como Zabbix en sus sistemas debido a que soporta múltiples plataformas de ejecución para su monitoreo lo que permite a los administradores de TI implementarlo en una amplia variedad de servidores además de proveer una escalabilidad al usuario para poder ir agregando más equipos al servicio para tener un sistemas centralizado donde poder realizar diversas acciones que posee esta herramienta como la recolección de datos para informes, despliegues por comando, compatibilidad con varios protocolos como SNMP, IMPI, entre otros (Mardiyono, Sholihah, & Hakim, 2020).

La última versión oficial de esta herramienta es la 7.0 tanto para el servidor como agente Zabbix esta versión trae consigo muchas funcionalidades como el preprocesamiento para pruebas con un valor real que permite que antes de guardar un dato en la base de datos de Zabbix este mismo pueda ser transformado de acuerdo a las necesidades del usuario (Zabbix, 2024).

En la Figura 1 se presenta una vista del entorno de inicio de la interfaz web de Zabbix donde posee información los hosts conectados además del estado del sistema.

Figura 1

Pantalla de Inicio de Zabbix



Fuente: Tomada de pantalla de inicio de Zabbix.

La herramienta de monitoreo ofrece la capacidad de definir alertas las cuales nos sirven para poder tomar acciones de acuerdo con el problema presentado para lo cual por defecto existen unos triggers o iniciadores que nos proveen de alertas cuando una expresión definida anteriormente es evaluada en verdadero.

Esto permite a los equipos de TI establecer iniciadores que emitan alertas de acuerdo a las necesidades, para poder evitar problemas potenciales antes de que se conviertan en interrupciones graves en el servicio alojado en el servidor.

Las alertas que proporciona esta herramienta se pueden configurar de diversas formas ya que es compatible con protocolos como SNMP (Simple Network Management Protocol) lo que le permite realizar envíos por correo electrónico y también posee integraciones con APIs en lenguajes como Python para interconectarse y emitir alguna alerta. Aparte de que proporciona diversos datos que recompilan los ítems de monitoreo de la herramienta para poder realizar diversos tipos de análisis de acuerdo a las necesidades del administrador (Dalle Vacche, 2015).

2.1.1 Características de Zabbix

Zabbix nos brinda una gran cantidad de integraciones por medio de APIs las cuales nos brinda la posibilidad de tener un control total de las diferentes acciones que requiera realizar el administrador del sistema. Una de las características más notables de Zabbix es su capacidad para monitorear varios indicadores de rendimiento de sistemas, redes, aplicaciones y servicios en tiempo real lo que permite realizar la configuración de tareas automáticas de acuerdo con el indicador que se requiera. Esto incluye monitorear la disponibilidad y el rendimiento de la CPU, la memoria, el espacio en disco, el tráfico de red y aplicaciones específicas (Chuvakin, 2013).

Además, Zabbix posee una base de datos que almacena todos los registros de los censados que realiza por medio del monitoreo facilitando el análisis de tendencias y el diagnóstico de problemas. Zabbix admite varios métodos de monitoreo, incluidos agentes de software, consultas SNMP, ping ICMP, monitoreo a través de protocolos HTTP y SSH, y más. Esto permite integrarlo con diferentes dispositivos y servicios en diferentes entornos de TI (Tadger, 2019). En la Figura 2 se muestra el apartado de información del sistema que nos proporciona Zabbix para ver su estado.

Figura 2

Informe del estado actual del sistema



The screenshot displays the Zabbix web interface, specifically the 'Información del sistema' (System Information) page. The interface is dark-themed and includes a sidebar on the left with navigation options like 'Inicio', 'Monitorización', 'Servicios', 'Inventario', and 'Integración'. The main content area shows a table of system metrics. The table has three columns: 'Parámetro' (Parameter), 'Valor' (Value), and 'Estado' (Status). The data is as follows:

Parámetro	Valor	Estado
El servidor Zabbix se está ejecutando	0	localhost1061
Versión del servidor Zabbix	7.8.0rc3	¡Nueva actualización disponible!
Versión de interfaz de Zabbix	7.8.0rc3	¡Nueva actualización disponible!
Actualización de software (comparada por última vez)	2024-07-11	
Última versión	7.8.0	Ir a la página de versión
Número de nodos (substituido/eliminados)	1	1/0
Número de plantillas	312	
Número de métricas (substituido/eliminados/reparadas)	162	112/49/1
Número de acciones (substituido/eliminados/programadas)	38	38/0/0/0/0
Número de usuarios (en línea)	2	1
Rendimiento de servidor requerido, nuevos valores por segundo	4.74	
Tamaño del disco en el servidor Zabbix	Desconocido	
Código de alta disponibilidad	Desconocido	

Fuente: Tomada de pantalla de información del sistema de Zabbix.

Otra de las virtudes de la herramienta de monitoreo es su capacidad para generar alertas y notificaciones basadas en parámetros definidos por el usuario como por ejemplo porcentajes de consumo de recursos, estado de servicios en ejecución entre otros.

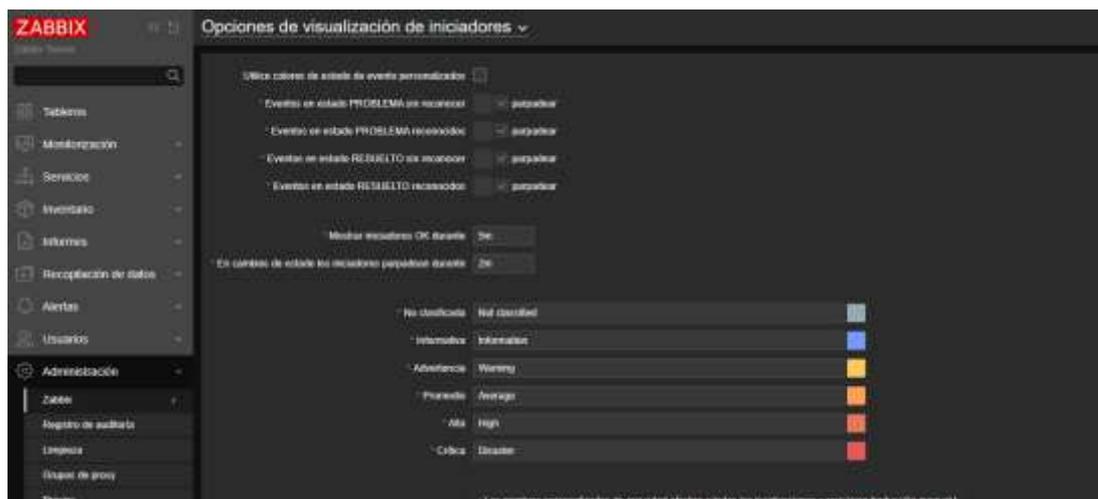
Zabbix provee a los administradores una interfaz amigable donde pueden realizar las configuraciones específicas de acuerdo con los requerimientos definidos para realizar diversas acciones desde activar alertas hasta realizar acciones a tomar en respuesta a esos eventos, como enviar correos electrónicos, mensajes de texto o ejecutar scripts personalizados.

La interfaz web de la herramienta también le permite crear plantillas personalizadas y gráficos detallados para visualizar el estado y el rendimiento de los sistemas monitoreados de una manera intuitiva y fácilmente accesible (Salah, 2018).

En la Figura 3 se muestran las opciones para definir las opciones de visualización de los triggers o iniciadores de acuerdo al nivel de alerta que el usuario defina según el proceso a monitorear.

Figura 3

Opciones de visualización de iniciadores



Fuente: Tomada de pantalla de opción de visualización de iniciadores de Zabbix.

2.1.2 Arquitectura de Zabbix

La arquitectura de Zabbix permite que el sistema donde se implemente esta herramienta pueda llegar a crecer con el tiempo ya que está separada por componentes. El componente principal es el Zabbix Server debido a que este es el núcleo principal que se encarga del procesamiento de los datos monitorizados para su posterior guardado en la base de datos que también posee.

Para esta base de datos se puede incorporar cualquiera de estos motores como lo son MySQL, MariaDB, PostgreSQL que son compatibles con la herramienta para una mejor gestión de los datos de acuerdo al motor de base de datos de preferencia del usuario.

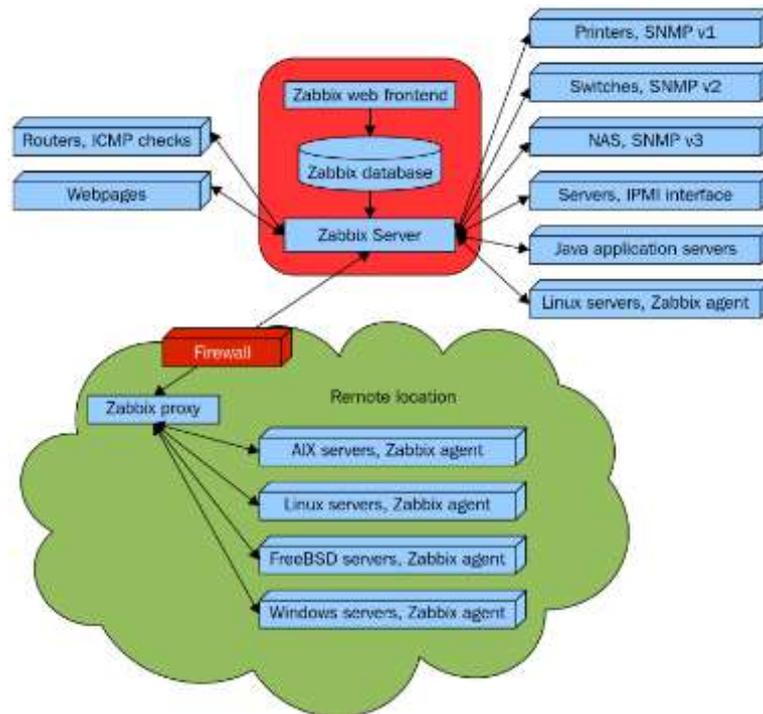
Los servidores Zabbix almacenan datos históricos en una base de datos para facilitar el análisis y la generación de informes detallados sobre el rendimiento y la disponibilidad de los recursos monitoreados a lo largo de un gran tiempo el cual se puede configurar en los ítems de monitoreo de la herramienta.

Otro componente importante es el del agente Zabbix que se refiere al programa instalado en el servidor que se requiera monitorear para la recopilación de datos y envié al servidor Zabbix. Estos agentes brindan un monitoreo detallado en tiempo real del estado de los diversos parámetros de los equipos conectados (Zabbix, 2024).

Zabbix también posee una interfaz web denominada Zabbix Frontend con la cual se puede gestionar mediante usuarios la administración del sistema de monitoreo. Por medio de esta se puede llegar a configurar todas las acciones e iniciadores necesarios para tener un registro de los parámetros de interés que los usuarios requieran para tener un histórico del rendimiento del sistema además de poder generar informes que permitan la toma de decisiones al momento de que se requiera aplicar algún mantenimiento o mejora en el equipo. (Liefiting & van Baekel, 2024).

Figura 4

Red simplificada haciendo uso de Zabbix



Fuente: (Olups, 2016).

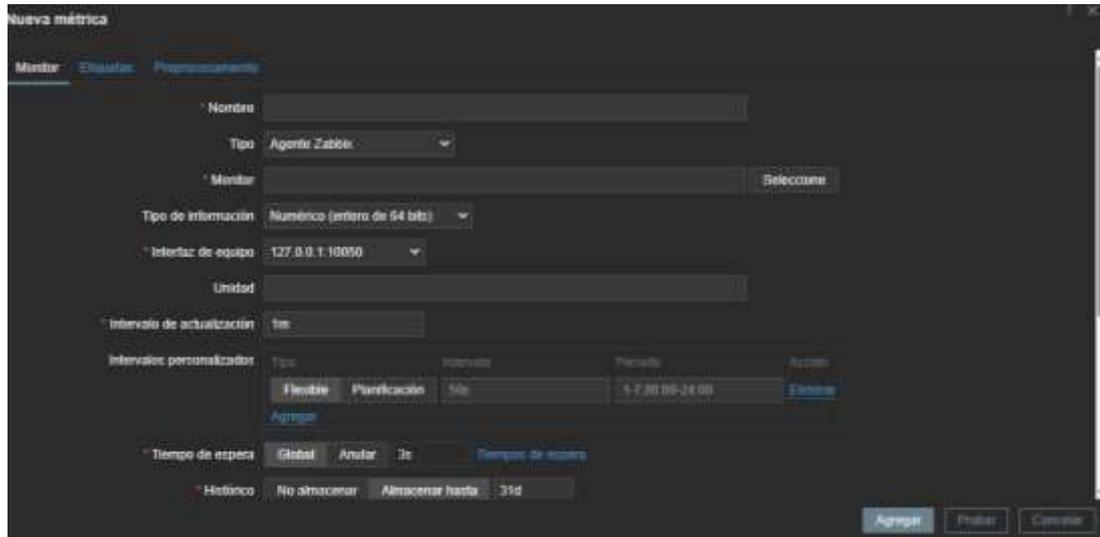
2.1.3 Monitores de Zabbix

Los monitores o también denominados ítems son elementos que cumplen la función de recopilar los datos obtenidos de acuerdo a una configuración preestablecida lo cual nos permite obtener diversos parámetros que nos permiten detectar el estado de diferentes componentes como por ejemplo el consumo de la CPU, el espacio disponible en disco entre otros, con los cuales podemos llegar a tener en tiempo real el estado del elemento que queramos monitorear (Alves, 2015).

La herramienta de monitoreo también provee una gran cantidad de ítems por defecto al instalar el programa que pueden servir como guía para nuevas implementaciones. En la Figura 5 se puede visualizar los diversos parámetros que se pueden configurar para crear un ítem en Zabbix.

Figura 5

Vista de la ventana de creación de ítems de Zabbix



Fuente: Tomada de pantalla de creación de ítems de Zabbix.

Para la creación de ítems es necesario definir claramente el nombre con el cual se denominará al mismo debido a que se pueden crear una gran cantidad variaciones de ítems de una función en específico por ejemplo si se requiere determinar el espacio disponible en disco de un servidor Ubuntu se puede usar la siguiente key: `vfs.fs.size[fs, < mode >]`

La key presentada acepta 2 parámetros el primero es el directorio donde queremos obtener la información del tamaño que sería en este caso el directorio raíz para obtener el total de espacio disponible y el segundo parámetro sería el modo que corresponde `free` debido a que este nos proporciona el porcentaje disponible del directorio parametrizado pero a su vez podemos tener otro ítem similar solo que enfocado en determinar el espacio que ocupa un directorio por lo cual el nombre que los identifique debe ser claro y conciso para que todo usuario que acceda a los registros puedan identificar a quien le pertenece los datos monitoreados.

También es importante definir el tipo de ítem de acuerdo al requerimiento del usuario generalmente se utiliza directamente el agente Zabbix para poder interactuar con los equipos que estén destinados al monitoreo, el tipo de información se refiere al tipo de valor ya sea este carácter o numérico que va de acuerdo a la key elegida previamente, la interfaz de

equipo nos permite seleccionar la Ip del equipo a monitorear junto al puerto, la unidad va de acuerdo al valor obtenido y al tipo de dato proveniente de la key.

Otro de los parámetros requeridos es el intervalo de actualización el cual nos permite seleccionar el tiempo de espera entre un monitoreo y otro. También podemos configurar el histórico y las tendencias para definir los tiempos en el cual podremos tener la información para llevar un registro más detallado o general (Barros, 2022).

2.1.4 Iniciadores de Zabbix

Los iniciadores o también denominados triggers son funciones lógicas que evalúan los datos obtenidos por los ítems previamente creados que tienen como finalidad su activación cuando la expresión lógica que se defina se evalúe en verdadero, estos permiten que se activen ciertas alertas cuando algún parámetro del equipo no se encuentra en los rangos específicos definidos en la expresión o cuando se necesita saber algún cambio en algún elemento del sistema (Zabbix, 2024). En la Figura 6 se puede visualizar los diversos parámetros que se pueden configurar para crear un trigger en Zabbix.

Figura 6

Vista de la ventana de creación de triggers de Zabbix

Fuente: Tomada de pantalla de creación de triggers de Zabbix.

Para la creación de triggers es muy importante establecer un nombre que tenga relación con la acción que provoca su activación para que el usuario pueda dar solución a la alerta notificada, por ejemplo si es requerido realizar un trigger que se active cuando el porcentaje de espacio en disco del sistema sea inferior al 10% podríamos definirlo con la siguiente expresión `last(/Zabbix server / vfs.fs.size[/,pfree] < 10)` la cual nos permite establecer de acuerdo al ítem que nos proporciona el porcentaje de espacio libre de disco una validación para que se active cuando el ultimo valor obtenido del porcentaje sea inferior al 10 %.

También se pueden clasificar las gravedades de acuerdo a las necesidades de los usuarios, entre las cuales tenemos por defecto: no clasificada, informativa, advertencia, promedio, alta y crítica. Se puede también configurar expresiones para que el evento que se genera cuando se activa el trigger que tiene por defecto OK cuando la expresión está en falso y PROBLEMA cuando la expresión nos da verdadero, por lo tanto si se requiere que la expresión regrese a su estado por defecto que es OK se puede generar un expresión de recuperación que permite que una vez este solventado el problema vuelva a su estado por defecto, estos elementos son de mucha importancia al momento de definir los parámetros a monitorear ya que dependiendo de estos podemos llegar a definir las expresiones de acuerdo al criterio del administrador de TI.

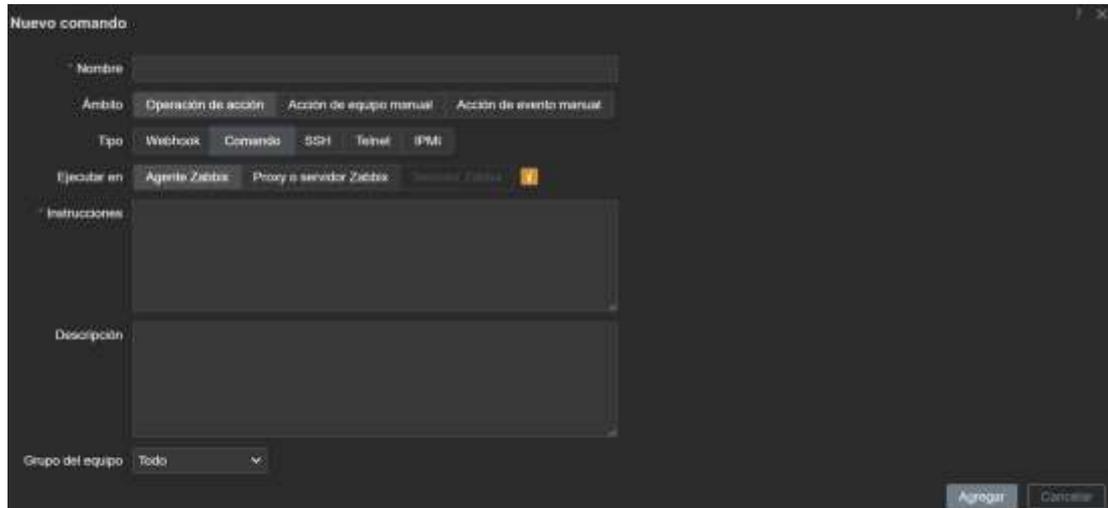
2.1.5 Comandos de Zabbix

Los comandos en Zabbix son un elemento que nos permite realizar diferentes tipos de acciones al equipo requerido entre las cuales esta realizar llamadas HTTP por medio de webhooks personalizados de acuerdo a las preferencias del usuario, también por comando de consola, SSH, Telnet, IPMI.

En el apartado de comando de consola se puede escribir la instrucción a ejecutar y se puede elegir en donde se procederá a ejecutar y en la parte inferior nos permite elegir el grupo de equipos que aplicará para la ejecución de ese comando (Zabbix, 2024). En la Figura 7 se puede visualizar los diversos parámetros mencionados que se pueden configurar para crear un comando en Zabbix.

Figura 7

Vista de la ventana de creación de comandos de Zabbix



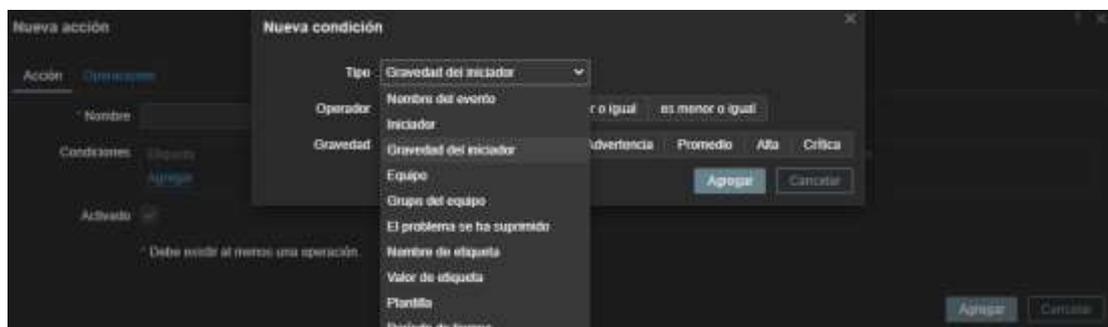
Fuente: Tomada de pantalla de creación de comandos de Zabbix.

2.1.6 Acciones de iniciadores de Zabbix

Las acciones de iniciadores en Zabbix nos permiten realizar operaciones automáticas cuando ocurre un evento en específico lo que nos permite tomar las medidas necesarias dependiendo del estado de iniciadores previamente definidos por el usuario. En la Figura 8 se presenta la pantalla para creación de acciones de acuerdo a condiciones previamente establecidas para lo cual la herramienta nos provee algunos tipos de configuraciones de acuerdo a diferentes estados.

Figura 8

Vista de la ventana de creación de acciones de iniciadores de Zabbix

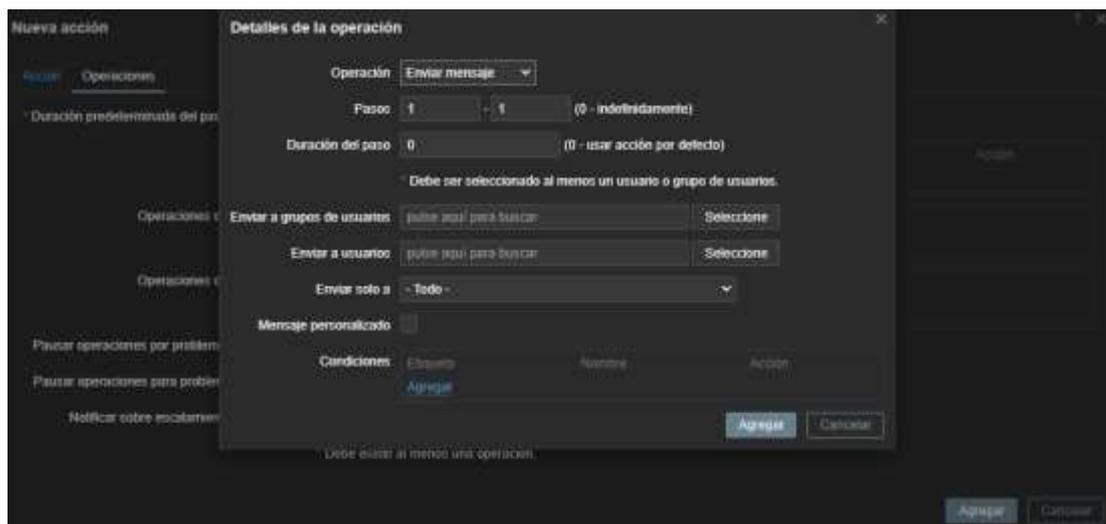


Fuente: Tomada de pantalla de creación de acciones de iniciador de Zabbix.

En la Figura 9 se presenta las diferentes configuraciones disponibles para establecer operaciones cuando la condición definida en la acción sea evaluada como resultado verdad, se puede llegar a personalizar la acción de acuerdo al grupo seleccionado de equipos en donde aplica realizar dicha acción lo que permite al usuario personalizar la acción a sus necesidades.

Figura 9

Vista de la ventana para configurar operaciones de acciones de Zabbix



Fuente: Tomada de pantalla para configurar operaciones de acciones en Zabbix.

2.2 Introducción a servidores virtuales

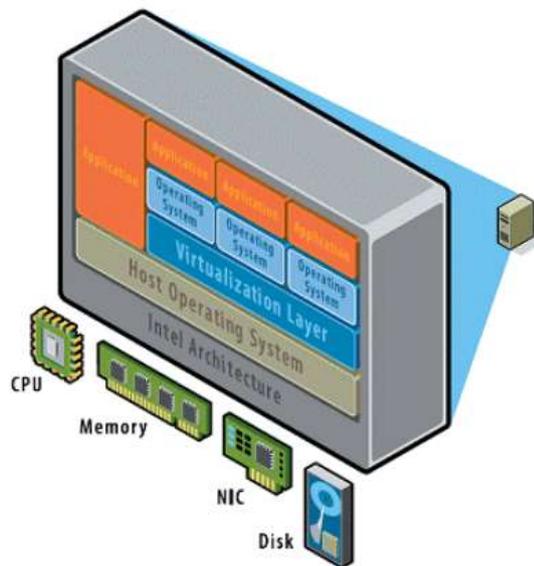
Los servidores virtuales son una de las tecnologías de gestión de sistemas y ordenadores más revolucionarias. Al crear múltiples entornos virtuales en una única máquina física, la virtualización cambia la forma en que las organizaciones implementan, administran y optimizan sus recursos de TI. Esta tecnología ofrece varias ventajas, como costos reducidos, mejor utilización de recursos y mayor flexibilidad y escalabilidad en la gestión del sistema (Barham et al., 2003).

La virtualización de servidores no sólo facilita la consolidación de servidores, sino que también simplifica el proceso de desarrollo y prueba de software al crear un entorno de prueba que replica fielmente el entorno de producción. Además, proporciona una infraestructura sólida que puede soportar fallas de hardware y facilita la recuperación ante desastres mediante

migración en vivo y replicación de máquinas virtuales (Rosenblum & Garfinkel, 2005). En la Figura 10 se presenta un modelo de máquina virtual con cada uno de sus componentes para su correcto funcionamiento al momento de operarlo.

Figura 10

Representación de un modelo de máquina virtual



Fuente: (Doña et al., 2010, p. 1)

2.2.1 Requerimientos Generales

La implementación de un servidor virtual requiere un análisis detallado de los requisitos técnicos y operativos. La capacidad del hardware es un aspecto importante que considerar porque la virtualización implica compartir recursos físicos entre múltiples máquinas virtuales.

La CPU, la memoria y el almacenamiento deben evaluarse cuidadosamente para garantizar que el servidor físico pueda manejar las demandas de la máquina virtual sin comprometer el rendimiento (Barham et al., 2003).

La correcta elección para un hipervisor es de mucha importancia a la hora de querer implementar un servidor virtual debido a que estos son los

encargados de crear y ejecutar las máquinas virtuales, estos también se pueden utilizar para ejecutar una o varias máquinas virtuales a la vez.

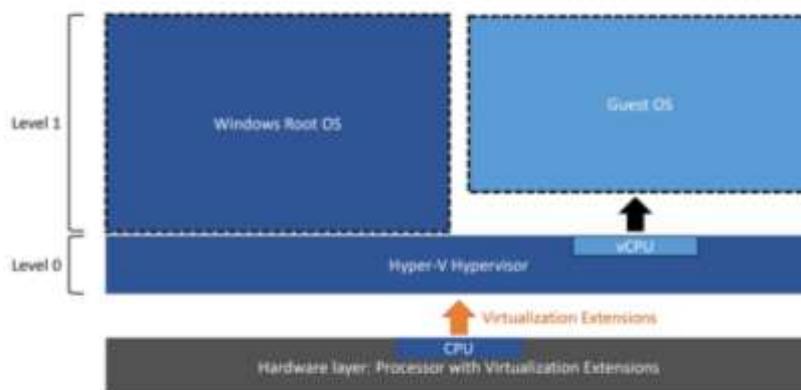
Entre los hipervisores más conocidos están VMware, Microsoft Hyper-V entre otros. Es necesario tener bien definido los recursos destinados para el funcionamiento de estos sistemas ya que al depender del hardware del equipo donde se está en ejecución, puede llegar a consumir una gran parte de los recursos si es que no se lo administra correctamente (Sugerman, Venkitachalam, & Lim, 2001).

La configuración de la red y el almacenamiento también es muy importante para garantizar el rendimiento y la disponibilidad del servidor virtual. Esto incluye la implementación de VLAN, conmutadores virtuales y soluciones de almacenamiento compartido para tener una alta disponibilidad en el servicio que se esté brindando.

La capacidad de administrar y monitorear de manera efectiva estos recursos es fundamental para mantener la estabilidad y el rendimiento de su infraestructura virtual. La configuración adecuada de red y almacenamiento garantiza que las máquinas virtuales puedan comunicarse y acceder a los recursos que necesitan sin interrupción. (Clark et al., 2005). En la Figura 11 se puede visualizar un diagrama de Hyper-V sin anidamiento.

Figura 11

Diagrama de una Hyper-V sin anidamiento



Fuente: (Microsoft, 2023).

2.3 Introducción a Ubuntu Server

La distribución de Ubuntu Server es un sistema operativo basado en GNU/Linux diseñada específicamente para su uso en servidores. Esta distribución tiene muchas ventajas entre las cuales están los bajos recursos que se requieren para su ejecución lo que ha permitido que una gran cantidad de personas la utilicen debido a su estabilidad, seguridad y amplia compatibilidad con diversos entornos de hardware y software.

Ubuntu Server nos permite realizar una amplia gama de configuraciones a nivel de sistema y al ser también Open-Source cuenta con una gran comunidad que brinda su ayuda para realizar nuevas configuraciones que permiten realizar mejoras en el sistema lo cual es de mucha utilidad debido a su flexibilidad y adaptación a diferentes tecnologías (Gonçalves, 2019).

Una de las ventajas de usar Ubuntu Server es su facilidad de uso para integrar una gran variedad de herramientas y configurar servicios que nos permitan tener un control total de nuestro sistema. Además de que posee una gran comunidad que siempre ofrece su conocimiento para la creación de nuevo contenido para las nuevas versiones del sistema, este servidor permite y facilita la implementación rápida y eficiente de servicios web y de bases de dato.

Además, Ubuntu Server también admite tecnologías de virtualización y contenedorización como KVM y Docker, lo que permite a las organizaciones utilizar plenamente los recursos de hardware y optimizar el rendimiento de las aplicaciones (Hill,2018).

Figura 12

Interfaz de comando de Ubuntu Server

```
ubuntu 24.04 LTS localhost tty1
localhost login: jair
Password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-36-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

System information as of Tue 11 Jul 2024 14:52:09 -05

System load: 0.81          Processes:              323
Usage of /:  45.3% of 13.67GB Users logged in:        0
Memory usage: 13%         IPv4 address for ens33: 192.168.163.128
Swap usage:  0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.

   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado
Se pueden aplicar 42 actualizaciones de forma inmediata.
Para ver estas actualizaciones adicionales, ejecute: apt list --upgradable

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute #sudo pro status

jair@localhost:~$ _
```

Elaborado por: Autor.

La distribución de Ubuntu Server cuenta con soporte a largo plazo (LTS) lo que permite que las actualizaciones que contengan parches para seguridad y otros elementos del sistema se encuentren siempre en el mejor estado para que el rendimiento del servidor sea siempre el más óptimo.

Además, la gran comunidad de usuarios que utilizan Ubuntu proporciona valiosos conocimientos y recursos de soporte a través de foros, documentos en línea y proyectos colaborativos, lo que facilita la resolución de problemas y el intercambio de mejores prácticas (Kissling, 2020).

2.3.1 Comandos en Linux

La distribución de Ubuntu Server por defecto solo incluye una consola de comandos con respecto a la versión de escritorio de Ubuntu la cual, si posee interfaz gráfica, por lo cual es de mucha importancia conocer la sintaxis general de los comandos más requeridos y utilizados por administradores de sistemas, entre las cuales están los comandos para instalar y actualizar los paquetes del sistema como `sudo apt update` que sirve para actualizar la lista de paquetes disponibles y sus versiones.

Si se requiere se pueden llegar actualizar los paquetes instalados en el equipo haciendo uso de `sudo apt upgrade` el cual actualiza los paquetes instalados en el sistema a su última versión disponible. Para instalar nuevos paquetes se puede hacer uso de comandos como `sudo apt install <paquete>` el cual nos permite instalar nuevos paquetes desde los repositorios configurados en el sistema.

2.4 Introducción a Python

Python es un lenguaje de programación de alto nivel que permite al programador realizar instrucciones mediante código para que posteriormente en su ejecución dicho código de transforme en un lenguaje que el computador pueda interpretar para su ejecución en el sistema.

En la actualidad Python es muy utilizado en diferentes áreas de las ciencias entre las que más se destaca en el análisis de datos y el desarrollo de redes neuronales debido a los grandes avances que se han realizado en los últimos años, el auge de la inteligencia artificial ha llevado a lenguajes como Python a ser muy utilizados en la comunidad por su fácil comprensión para su estudio e implementación en algunos sistemas de hoy en día (Buriticá & Guerrero, 2021, p. X).

2.4.1 Librería OS

La librería OS nos permite realizar operaciones como crear, editar, eliminar archivos tal cual como si estuviéramos interactuando directamente con la consola del sistema aparte de que tiene muchas más funcionalidades útiles que permiten al usuario interactuar con el sistema mediante código de Python, una de las ventajas que posee esta librería es que podemos hacer uso de la misma tanto en sistemas operativos Windows como Linux. Para importarla en nuestro código de Python podemos hacer uso de `import os` (Rey Vela, 2018).

2.4.2 Librería psutil

La librería psutil provee la información sobre los procesos que se encuentran en ejecución en el sistema y también provee funciones para controlar el comportamiento de estos procesos para una mejor administración del sistema mediante código de Python, esta librería esta funcional a partir de la versión 2.6 de Python en adelante y también ofrece las mismas ventajas que la librería OS que soporta múltiples plataformas (Solís, Trujillo-Romero, & Capulín, 2023)

2.4.3 Librería shutil

La librería shutil nos permite manipular los directorios y archivos del sistema donde se ejecute la instrucción con un buen manejo de errores para los escenarios donde algún proceso de transferencia de archivos de un directorio a otro tengo algún problema y no se complete el traslado. Esta librería destaca también por su funcionalidad de poder consultar el porcentaje de ocupación del disco cuando el usuario lo requiera para alguna aplicar alguna funcionalidad por medio de código de Python (Rey, 2023).

2.4.4 Librería subprocess

La librería subprocess es de mucha utilidad para realizar automatización de tareas que requieran la ejecución de comandos por consola y para la creación y manejo de procesos que requieran que se ejecuten en paralelo o asíncronamente. Por lo cual esta librería es muy esencial para la administración de procesos en el entorno donde se requiera de su uso. También permite la integración con programas externos manejando a nivel de ejecución y procesos la información (José Ortega, 2022).

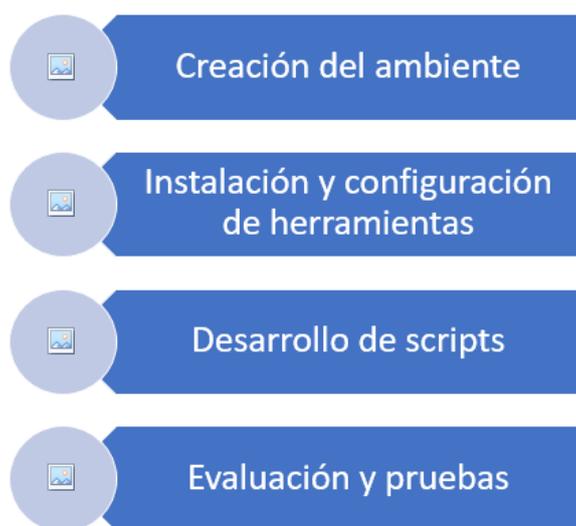
CAPÍTULO 3: APORTES DE LA INVESTIGACIÓN

En este capítulo, se aborda el diseño y la implementación de un sistema de control de gestión para servidores mediante códigos en Python para poder gestionar los procesos además de integrarla junto con Zabbix el cual es el software encargado del monitoreo de la actividad en tiempo real de los diferentes parámetros a analizar entre los cuales tenemos el consumo a nivel de procesamiento del CPU, memoria RAM y el almacenamiento interno que el equipo gestiona.

En la Figura 13 se presenta el diagrama de bloques del proceso a realizar para obtener una solución que mediante una configuración correcta pueda mejorar el rendimiento de los servidores.

Figura 13

Diagrama de bloques del proceso



Elaborado por: Autor.

El enfoque de esta investigación es de mostrar las configuraciones necesarias para poder implementar una solución a los problemas que surgen en los servidores debido a los diversos componentes del hardware del equipo los cuales al estar en un entorno generalmente cerrado, su gestión y monitoreo es de mucha importancia para que todos los servicios que se estén ejecutando funcionen con normalidad y no haya ninguna afectación que pueda ocasionar graves problemas en las comunicaciones que este tenga con los

diversos equipos que hagan uso de los servicios que estén levantados en el servidor.

3.1 Creación del ambiente

Para la virtualización del servidor se ha utilizado el software VMware Workstation Pro 17 por su alta compatibilidad y soporte que ofrece para una gran variedad de sistemas operativos, además de que gestiona eficientemente todos los recursos del ordenador para poder obtener el mejor rendimiento al momento de la virtualización y que además posee una licencia gratis para uso personal.

Como servidor virtualizado se ha tomado la decisión de usar un servidor Ubuntu 24.04 LTS debido a sus principales características entre las que destaca es su alta estabilidad y fiabilidad lo que nos permite tener un entorno de pruebas robusto para que los resultados obtenidos sean lo más aproximado a la realidad, además de que nos permite hacer uso del sistema con una cantidad mínima recursos para su ejecución, para la implementación se ha hecho uso de un equipo con las siguientes características:

- Laptop Asus
- Procesador AMD Ryzen 9 8945HS
- Memoria RAM de 16GB
- Almacenamiento de 1TB
- Sistema Operativo Windows 11

Para descargar el archivo ISO que contiene el sistema operativo del servidor de Ubuntu se puede ingresar a la página oficial de Ubuntu y en la sección de descargas se encuentre el apartado de servidor del cual se ha obtenido al archivo para su instalación por medio de VMware Workstation Pro, una vez descargado el programa procederemos con su instalación para lo cual se ha establecido los siguientes recursos para poner a prueba el rendimiento del servidor en ejecución.

Una vez culminado el proceso de instalación del servidor Ubuntu e iniciada la sesión en la consola de comandos del servidor debería poder

visualizarse lo que se muestra en la Figura 14 para poder comenzar a instalar todos los paquetes necesarios para instalar y configurar Zabbix.

Figura 14

Vista de la consola de comandos de Ubuntu Server

```
Ubuntu 24.04 LTS localhost tty1
localhost login: jair
Password:
Welcome to Ubuntu 24.04 LTS (GNU/Linux 6.8.0-39-generic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/pro

System information as of dom 28 Jul 2024 15:58:05 -05

System load:  0.0                Processes:    298
Usage of /:   51.5% of 13.67GB   Users logged in:  1
Memory usage: 15%              IPv4 address for ens33: 192.168.198.128
Swap usage:  0%

 * Strictly confined Kubernetes makes edge and IoT secure. Learn how MicroK8s
   just raised the bar for easy, resilient and secure K8s cluster deployment.
   https://ubuntu.com/engage/secure-kubernetes-at-the-edge

El mantenimiento de seguridad expandido para Applications está desactivado
Se pueden aplicar 0 actualizaciones de forma inmediata.

Active ESM Apps para recibir futuras actualizaciones de seguridad adicionales.
Vea https://ubuntu.com/esm o ejecute «sudo pro status»

jair@localhost:~$
```

Elaborado por: Autor.

3.2 Instalación y configuración de herramientas

Para poder establecer el sistema de monitoreo mediante el uso de la herramienta Zabbix es necesario tener incorporado en el sistema algunos paquetes requeridos por el programa que nos permitirá que la herramienta de monitoreo funcione correctamente para poder hacer uso de todas sus funcionalidades que nos permitirán obtener las alertas que se requieran cuando algún parámetro del equipo este saturado.

3.2.1 Chrony

Para obtener todos los registros del monitoreo por parte de Zabbix es necesario que el servidor posea un Daemon o servicio que le permita estar sincronizado con la fecha y hora exacta para su buen desempeño, para lo cual se sugiere el uso del paquete chrony debido a su compatibilidad con Linux para lo cual se utiliza el comando `sudo apt install` para realizar la instalación del paquete.

Posterior a la ejecución del comando de instalación, realizamos el uso del comando `sudo systemctl enable` para habilitar `chrony` y puedo iniciar automáticamente al arrancar el sistema.

Para validar el estado de `chrony` hacemos uso del comando `sudo systemctl status` el cual nos detalla el estado actual del servicio de `chrony`. Adicionalmente para poder establecer la zona horaria requerida por el usuario se puede hacer uso del comando `sudo timedatectl set-timezone`.

Figura 15

Vista por consola del estado del servicio de Chrony

```
jair@localhost:~$ sudo apt install chrony
[sudo] password for jair:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
chrony ya está en su versión más reciente (4.5-1ubuntu4.1).
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
 libdaxctl1 libdaxctl6 libdaxml
Utilice «sudo apt autoremove» para eliminarlos.
0 actualizados, 0 nuevos se instalarán, 0 para eliminar y 0 no actualizados.
jair@localhost:~$ sudo systemctl enable chrony
Synchronizing state of chrony.service with SysV service script with /usr/lib/systemd/systemd-sysv-install.
Executing: /usr/lib/systemd/systemd-sysv-install enable chrony
jair@localhost:~$ sudo systemctl status chrony
* chrony.service - chrony, an NTP client/server
   Loaded: loaded (/usr/lib/systemd/system/chrony.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-07-31 21:44:10 -05; 30min ago
     Docs: man:chrony.conf(8)
           man:chronyc(1)
           man:chrony.conf(5)
   Main PID: 979 (chronyd)
     Tasks: 2 (Limit: 4556)
    Memory: 2.2M (peak: 3.0M)
         CPU: 346ms
   CGroup: /system.slice/chrony.service
           └─979 /usr/sbin/chronyd -F 1
             └─987 /usr/sbin/chronyd -F 1

Jul 31 21:44:10 localhost chronyd[979]: Using right/UTC timezone to obtain leap second data
Jul 31 21:44:10 localhost chronyd[979]: Loaded seccomp filter (level 1)
Jul 31 21:44:10 localhost systemd[1]: Started chrony.service - chrony, an NTP client/server.
Jul 31 21:44:19 localhost chronyd[979]: Selected source 185.125.190.57 (ntp.ubuntu.com)
Jul 31 21:44:19 localhost chronyd[979]: System clock wrong by -1.824163 seconds
Jul 31 21:44:18 localhost chronyd[979]: System clock was stepped by -1.824163 seconds
Jul 31 21:44:18 localhost chronyd[979]: System clock TAI offset set to 37 seconds
Jul 31 21:44:19 localhost chronyd[979]: Selected source 185.125.190.56 (ntp.ubuntu.com)
Jul 31 21:45:22 localhost chronyd[979]: Selected source 185.125.190.57 (ntp.ubuntu.com)
Jul 31 21:49:41 localhost chronyd[979]: Selected source 179.60.247.252 (0.ubuntu.pool.ntp.org)
jair@localhost:~$ sudo timedatectl set-timezone America/Guayaquil
```

Elaborado por: Autor.

3.2.2 Dependencias de Zabbix

Para instalar y configurar un entorno necesario para ejecutar Zabbix es necesario la instalación de los siguientes paquetes mediante el comando `sudo apt install mariadb-server php php-cli php-common php-fpm php-curl php-mysql apache2 curl`, el cual instala varios paquetes esenciales para el correcto funcionamiento de la herramienta de monitoreo.

Esto incluye MariaDB, que es un sistema de gestión de bases de datos necesario para almacenar datos de monitoreo de Zabbix, varios módulos de

PHP, que se utilizan para ejecutar scripts del lado del servidor de Zabbix y Apache que actúa como el servidor web para la interfaz de usuario de Zabbix.

Además, se instala curl, una herramienta útil para realizar transferencias de datos desde la línea de comandos, a menudo utilizada en scripts de administración de sistemas. En la figura 16 se muestra la instalación por consola de comandos de Ubuntu.

Figura 16

Vista de la instalación de las dependencias de Zabbix

```
jair@localhost:~$ sudo apt install mariadb-server php php-cli php-common php-fpm php-curl php-mysql apache2 curl
[sudo] password for jair:
Leyendo lista de paquetes... Hecho
Creando árbol de dependencias... Hecho
Leyendo la información de estado... Hecho
mariadb-server ya está en su versión más reciente (1:10.11.8-0ubuntu0.24.04.1).
php ya está en su versión más reciente (2:8.3+93ubuntu2).
php-cli ya está en su versión más reciente (2:8.3+93ubuntu2).
php-common ya está en su versión más reciente (2:93ubuntu2).
php-fpm ya está en su versión más reciente (2:8.3+93ubuntu2).
php-curl ya está en su versión más reciente (2:8.3+93ubuntu2).
php-mysql ya está en su versión más reciente (2:8.3+93ubuntu2).
apache2 ya está en su versión más reciente (2.4.58-1ubuntu0.4).
curl ya está en su versión más reciente (8.5.0-2ubuntu10.1).
Los paquetes indicados a continuación se instalaron de forma automática y ya no son necesarios.
 libdaxctl1 libndctl6 libpmem1
Utilice «sudo apt autoremove» para eliminarlos.
@ actualizados, @ nuevos se instalarán, @ para eliminar y @ no actualizados.
```

Elaborado por: Autor.

3.2.3 Servicio de base de datos para Zabbix

Para la configuración e instalación de MariaDB como servicio de base de datos para Zabbix es requerido realizar algunas configuraciones para establecer los enlaces que necesita Zabbix para conectar correctamente y poder llevar un registro de todos los monitoreos y alertas detectados por el servidor Zabbix.

Primero tenemos que establecer que el servicio se ejecute automáticamente una vez el sistema arranque para lo cual ejecutamos sudo systemctl enable MariaDB y siguiente a la ejecución validamos que el servicio se encuentre activo y en ejecución por medio del sudo systemctl status mariadb tal y como se muestra en la Figura 17.

Figura 17

Estado del servicio de base de datos en el servidor Ubuntu

```
Jair@localhost:~$ sudo systemctl status mariadb
* mariadb.service - MariaDB 10.11.8 database server
   Loaded: loaded (/usr/lib/systemd/system/mariadb.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-07-31 21:44:16 -05; 2h 15min ago
     Docs: man:mariadb(8)
           https://mariadb.com/kb/en/library/systemd/
  Main PID: 1077 (mariabdd)
    Status: "Taking your SQL requests now..."
     Tasks: 50 (limit: 30074)
    Memory: 179.1M (peak: 179.6M)
       CPU: 1min 3.563s
    CGroup: /system.slice/mariadb.service
            └─1077 /usr/sbin/mariabdd
```

Elaborado por: Autor.

Después de validar que el servicio se encuentre ejecutando correctamente ejecutamos `sudo mysql_secure_installation` el cual es un comando que mejora la seguridad de MariaDB estableciendo una contraseña para el usuario administrador, eliminando usuarios anónimos, deshabilitando los inicios de sesión remotos de administrador, eliminando la base de datos de prueba y recargando la tabla de permisos para aplicar los cambios.

Estas medidas son esenciales para proteger las bases de datos de Zabbix del acceso no autorizado y mantener la integridad de los datos monitoreados. Luego de iniciar el servicio de base de datos, procedemos a configurar la base de datos Zabbix, para lo cual usamos el comando `sudo mysql -u root -p`. Al ejecutar este comando, se accederá a la consola MariaDB como usuario root y se requerirá la contraseña de root para autenticar la sesión.

El comando `crear base de datos zabbix charset utf8mb4 collate utf8mb4_bin` para crear una nueva base de datos llamada zabbix es `utf8mb4` y su intercalación es `utf8mb4_bin` es correcto. Luego procedemos a otorgar todos los permisos al usuario de Zabbix y le damos permisos para que pueda crear funciones personalizadas requeridas por Zabbix y refrescamos los privilegios y salimos para que los cambios se apliquen. Los comandos utilizados se presentan en la Figura 18.

Figura 18

Comandos para configuración de base de datos para Zabbix

```
sudo mysql -u root -p
create database zabbix character set utf8mb4 collate utf8mb4_bin;
create user zabbix@localhost identified by 'contraseña';
grant all privileges on zabbix.* to zabbix@localhost;
set global log_bin_trust_function_creators = 1;
flush privileges;
exit
```

Elaborado por: Autor.

3.2.4 Configuración de php

Una vez configurada la base de datos debemos habilitar el servicio web mediante el comando `sudo systemctl enable` usado anteriormente para habilitar el servicio de base de datos cada vez que arranque el servidor.

Después procedemos con la edición del archivo `php.ini` utilizando el comando `sudo nano /etc/php/8.3/cli/php.ini`, en cual se realizarán varios cambios importantes en la configuración de PHP, específicamente para la línea de comandos (CLI). En la Figura 19 se muestra el contenido inicial del archivo.

Figura 19

Contenido del archivo `php.ini` por consola

```
PHP
[PHP]
;;;;;;;;;;;;;;;;;;;;;;;;;
; About php.ini
;;;;;;;;;;;;;;;;;;;;;;;;;
; PHP's initialization file, generally called php.ini, is responsible for
; configuring many of the aspects of PHP's behavior.

; PHP attempts to find and load this configuration from a number of locations.
; The following is a summary of its search order:
; 1. BPF module specific location.
; 2. The PHPRC environment variable.
; 3. A number of predefined registry keys on Windows.
; 4. Current working directory (except CLI)
; 5. The web server's directory (for BPF modules), or directory of PHP
; (otherwise in Windows)
; 6. The directory from the --with-config-file-path compile time option, or the
; Windows directory (usually C:\windows)
; See the PHP docs for more specific information.
;

; The syntax of the file is extremely simple. Whitespace and lines
; beginning with a semicolon are silently ignored (as you probably guessed).
; Section headers (e.g. [foo]) are also silently ignored, even though
; they might mean something in the future.

; Directives following the section heading [PATH=/www/mySite] only
; apply to PHP files in the /www/mySite directory. Directives
; following the section heading [HOST=/www.example.com] only apply to
; PHP files served from www.example.com. Directives set in these
; special sections cannot be overridden by user-defined INI files or
; at runtime. Currently, [PATH=] and [HOST=] sections only work under
; CGI/FastCGI.

; Directives are specified using the following syntax:
; directive = value
; Directive names are *case sensitive* - fooBar is different from F00bar.
; Directives are variables used to configure PHP or PHP extensions.
; There is no name validation. If PHP can't find an expected
; directive because it is not set or is mistyped, a default value will be used.

; The value can be a string, a number, a PHP constant (e.g. E_ALL or M_PI), one
; of the INI constants (On, Off, True, False, Yes, No and None) or an expression
; (e.g. E_ALL & ~E_NOTICE), a quoted string ("bar"), or a reference to a
```

Elaborado por: Autor.

Estos ajustes afectan el rendimiento y el comportamiento de PHP en el sistema. Tal y como se muestran en las siguientes figuras donde se muestra el archivo y las configuraciones correspondientes para un mejor desempeño de la plataforma.

Las configuraciones correspondientes a la figura nos permiten establecer límites para la memoria que un script de PHP puede usar, también el tiempo máximo de ejecución que puede tener de espera un script de PHP antes de que se detenga el proceso, también la configuración de la zona horaria, y por último desactiva la corrección automática de la ruta de archivo script en entornos CGI para evitar que se ejecuten scripts no deseados.

En la Figura 20 se muestra la configuración mencionada la cual es necesaria para poder utilizar la herramienta de monitoreo debido a que la interfaz web de Zabbix esta creada con el lenguaje de programación de PHP para lo cual al realizar estas configuraciones nos permitirá aprovechar la herramienta de una mejor manera.

Figura 20

Configuraciones del archivo php.ini

```
memory_limit = 2G
max_execution_time = 360
date.timezone = America/Guayaquil
cgi.fix_pathinfo=0
```

Elaborado por: Autor.

Luego debemos configurar el archivo `/etc/php/8.3/fpm/pool.d/www.conf` el cual pertenece a la configuración de PHP-FPM (FastCGI Process Manager). Este archivo nos permite acceder a los ajustes para realizar una gestión eficiente y segura de los procesos de PHP para un entorno de servidor web. En la Figura 21 se muestra el contenido del archivo.

Figura 21

Contenido del archivo www.conf por consola

```
GNU nano 2.2 /etc/php/8.3/fpm/pool.d/www.conf
; Start a new pool named 'www'.
; the variable $pool can be used in any directive and will be replaced by the
; pool name ('www' here)
[www]

; Per pool prefix
; It only applies on the following directives:
; - 'access.log'
; - 'slowlog'
; - 'listen' (unixsocket)
; - 'chroot'
; - 'chdir'
; - 'php_values'
; - 'php_admin_values'
; When not set, the global prefix (on/usr) applies instead.
; Note: This directive can also be relative to the global prefix.
; Default Value: none
;prefix = /path/to/pools/$pool

; Unix user/group of the child processes. This can be used only if the master
; process running user is root. It is set after the child process is created.
; The user and group can be specified either by their name or by their numeric
; IDs.
; Note: If the user is root, the executable needs to be started with
; --allow-to-run-as-root option to work.
; Default Values: The user is set to master process running user by default.
;                  If the group is not set, the user's group is used.
user = www-data
group = www-data

; The address on which to accept FastCGI requests.
; Valid syntaxes are:
;   'ip.add.re.ss:port'    - to listen on a TCP socket to a specific IPv4 address on
;                           a specific port;
;   '[ip:6:addr:ess]:port' - to listen on a TCP socket to a specific IPv6 address on
;                           a specific port;
;   'port'                - to listen on a TCP socket to all addresses
;                           (IPv6 and IPv4-mapped) on a specific port;
;   '/path/to/unix/socket' - to listen on a unix socket.
; Note: This value is mandatory.
listen = /run/php/php8.3-fpm.sock

; Set listen(2) backlog.
; Default Value: 511 (-1 on Linux, FreeBSD and OpenBSD)
;listen.backlog = 511
```

Elaborado por: Autor.

Primero procedemos a configurar el PHP-FPM para que solo escuche solicitudes del socket especificado en la figura para un mejor rendimiento, después especificamos que los procesos se ejecuten bajo un usuario y grupo típico para servidores en Linux.

Luego permitimos que cualquier cliente se conecte al socket y definimos al mismo usuario y grupo anteriormente configurado para asegurar que solo el servidor web pueda interactuar con PHP-FPM, después configuramos que el gestor de procesos maneje dinámicamente los procesos ejecutados. En la Figura 22 se muestran todas las configuraciones mencionadas por consola.

Figura 22

Configuraciones del archivo www.conf

```
listen = /run/php/php8.3-fpm.sock
user = www-data
group = www-data
listen.allowed_clients = 0.0.0.0
listen.owner = www-data
listen.group = www-data
pm = dynamic
```

Elaborado por: Autor.

Una vez configurado el PHP 8.3 FPM (FastCGI Process Manager) el cual es una implementación alternativa de PHP que está diseñada para gestionar de manera eficiente y segura la ejecución de scripts PHP en entornos de servidor web. FPM mejora el rendimiento y la capacidad de respuesta del servidor web al manejar múltiples procesos PHP simultáneamente, lo cual es especialmente útil para sitios web con alta carga de tráfico, procedemos a validar el estado del servicio después de activarlo tal y como se muestra en la Figura 23.

Figura 23

Estado del servicio de PHP 8.3 FPM en el servidor Ubuntu

```
jair@localhost:~$ sudo systemctl status php8.3-fpm
[sudo] password for jair:
● php8.3-fpm.service - The PHP 8.3 FastCGI Process Manager
   Loaded: loaded (/usr/lib/systemd/system/php8.3-fpm.service; enabled; preset: enabled)
   Active: active (running) since Wed 2024-07-31 21:44:11 -05; 2h 40min ago
     Docs: man:php-fpm8.3(8)
    Main PID: 919 (php-fpm8.3)
   Status: "Processes active: 0, idle: 2, Requests: 0, slow: 0, Traffic: 0req/sec"
     Tasks: 3 (limit: 4556)
  Memory: 27.9M (peak: 28.6M)
     CPU: 1.488s
    CGroup: /system.slice/php8.3-fpm.service
            └─ 919 "php-fpm: master process (/etc/php/8.3/fpm/php-fpm.conf)"
               └─ 1087 "php-fpm: pool www"
                  └─ 1088 "php-fpm: pool www"
```

Elaborado por: Autor.

3.2.5 Configuración de Zabbix

Una vez configurado todas las dependencias de Zabbix procedemos a ejecutar los comandos presentados en la Figura 24 para la descarga e instalación de los paquetes de la herramienta y posteriormente inicializamos la base de datos de Zabbix para poder comenzar a hacer uso de la herramienta y poder ingresar a la interfaz web.

Figura 24

Comandos para descargar e instalar Zabbix

```
wget -O https://repo.zabbix.com/zabbix/6.2/ubuntu/pool/main/z/zabbix-release/zabbix-release_latest+ubuntu24.04_all.deb
sudo dpkg -i zabbix-release_latest+ubuntu24.04_all.deb
sudo apt update
sudo apt install zabbix-server-mysql zabbix-frontend-php zabbix-apache-conf zabbix-sql-scripts zabbix-agent
cpnet /usr/share/zabbix-sql-scripts/mysql/server.sql.gz | mysql --default-character-set=utf8mb4 -uzabbix -p zabbix
sudo mysql -u root -p
set global log_bin_trust_function_creators = 0;
exit
```

Elaborado por: Autor.

Luego de instalar Zabbix procedemos a configurar el archivo `/etc/zabbix/zabbix_server.conf` el cual contiene las configuraciones de la herramienta de monitoreo para lo cual procedemos a configurar especificamos la ubicación donde estará la base de datos en este caso estamos en un ambiente local para pruebas por lo que se colocaría localhost después introducimos el nombre de la base de datos y el usuario que accederá a la misma junto con su contraseña. El contenido del archivo de configuración se puede ver en la Figura 25 y las configuraciones realizadas en la Figura 26.

Figura 25

Contenido del archivo `zabbix_server.conf` por consola

```
GNU nano 7.2 /etc/zabbix/zabbix_server.conf
# This is a configuration file for Zabbix server daemon
# To get more information about Zabbix, visit https://www.zabbix.com

##### GENERAL PARAMETERS #####

### Option: ListenPort
# Listen port for trapper.
#
# Mandatory: no
# Range: 1024-32767
# Default:
# ListenPort=10051

### Option: SourceIP
# Source IP address for outgoing connections.
#
# Mandatory: no
# Default:
# SourceIP=

### Option: LogType
# Specifies where log messages are written to:
# system - syslog
# file - file specified with LogFile parameter
# console - standard output
#
# Mandatory: no
# Default:
# LogType=file

### Option: LogFile
# Log file name for LogType 'file' parameter.
#
# Mandatory: yes, if LogType is set to file, otherwise no
# Default:
# LogFile=

LogFile=/var/log/zabbix/zabbix_server.log
```

Elaborado por: Autor.

Figura 26

Configuraciones del archivo zabbix_server.conf

```
DBHOST=localhost
DBName=zabbix
DBUser=zabbix
DBPassword=password
```

Elaborado por: Autor.

Por último, podemos configurar el firewall del servidor para que permita el tráfico HTTP por el puerto 80 y el HTTPS por el puerto 443, también configuramos los puertos TCP 10050 y 10051 los cuales son utilizados para la comunicación por Zabbix y permitimos el tráfico SSH el cual pertenece al puerto 22 y activamos el firewall con las reglas configuradas para filtrar el tráfico de la red y no permitir accesos no autorizados. En la Figura 27 se muestra las configuraciones mencionadas por comando.

Figura 27

Configuraciones de Firewall del servidor Ubuntu

```
sudo ufw allow 80/tcp
sudo ufw allow 443/tcp
sudo ufw allow 10050:10051/tcp
sudo ufw allow 22/tcp
sudo ufw enable
```

Elaborado por: Autor.

3.3 Desarrollo de scripts

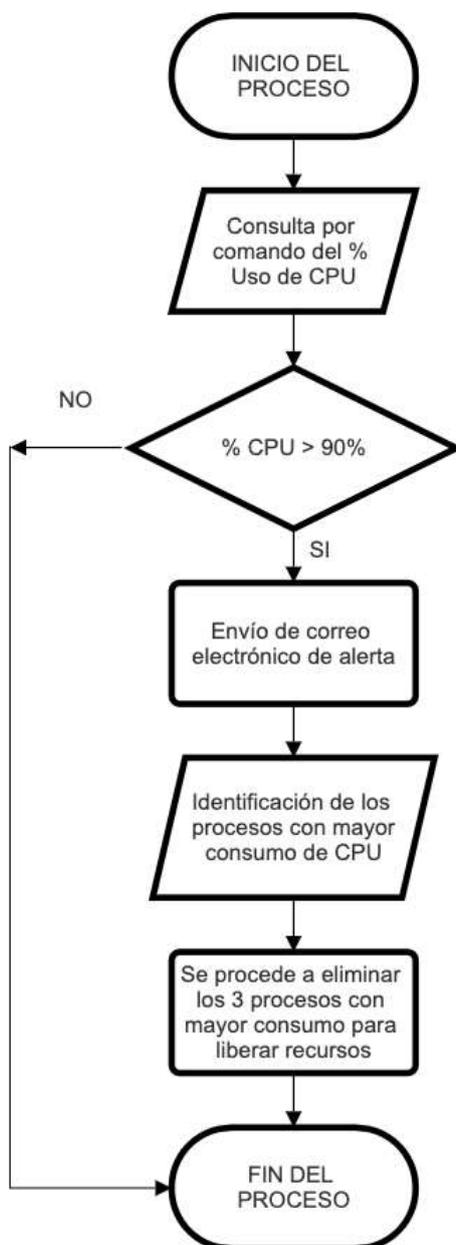
Para poder ejecutar scripts de Python en el servidor Ubuntu debemos comprobar que el servidor tenga instalado los paquetes necesarios por lo que primero verificamos que tengamos instalado Python usando el comando `python3 --version` que nos permite ver la versión que tenemos instalada en el equipo. Para la ejecución de estos scripts se ha utilizado la versión 3.12.3 de Python debido a que esta es una de las últimas versiones estables actualmente. A continuación, se presentará la estructura de los códigos que se utilizó para el sistema de control de congestión para lo cual se definió los siguientes componentes del hardware como lo son el CPU, RAM y espacio en disco como elementos importantes para que el sistema funcione en óptimas condiciones.

3.3.1 Script para sobrecarga de CPU

Para la creación del script que mitigue la sobrecarga por el alto uso de CPU por parte del servidor se detalló por medio de un diagrama de flujo secuencial los pasos a seguir para controlar las congestiones provocadas por un consumo excesivo por parte del procesador. En la Figura 28 se presenta un diagrama de flujo que contiene la estructura del código de Python.

Figura 28

Diagrama de flujo de script para sobrecarga de CPU



Elaborado por: Autor.

El script primero consulta por medio de la función `cpu_percent` de la librería `psutil` el porcentaje de CPU utilizado en ese momento para poder validar que efectivamente el equipo se encuentra con el CPU saturado, si la condición establecida por defecto que es cuando el porcentaje de CPU actual sea mayor al 90% de verdadera procederá a enviar un correo electrónico por medio de la librería `EmailMessage` de Python para alertar a los usuarios requeridos que el servidor está teniendo un consumo superior al establecido con anterioridad que fue 90%, este parámetro se puede editar en la cabecera de ser necesario otro tipo de límite para el control del sistema.

Una vez enviado el correo de alerta a los usuarios pertinentes se procederá a consultar los procesos que más estén utilizando el CPU para filtrar los 3 primeros que posteriormente se procederán a iterar cada uno de los procesos para que por medio del identificador único del proceso o también denominado PID procedamos a eliminarlo usando la instrucción `os.kill` que recibe dos parámetros lo cuales son el PID y el segundo es la señal que le vamos a dar a ese proceso para su eliminación para lo cual utilizamos el 9 el cual hace que el proceso termine de inmediato su ejecución.

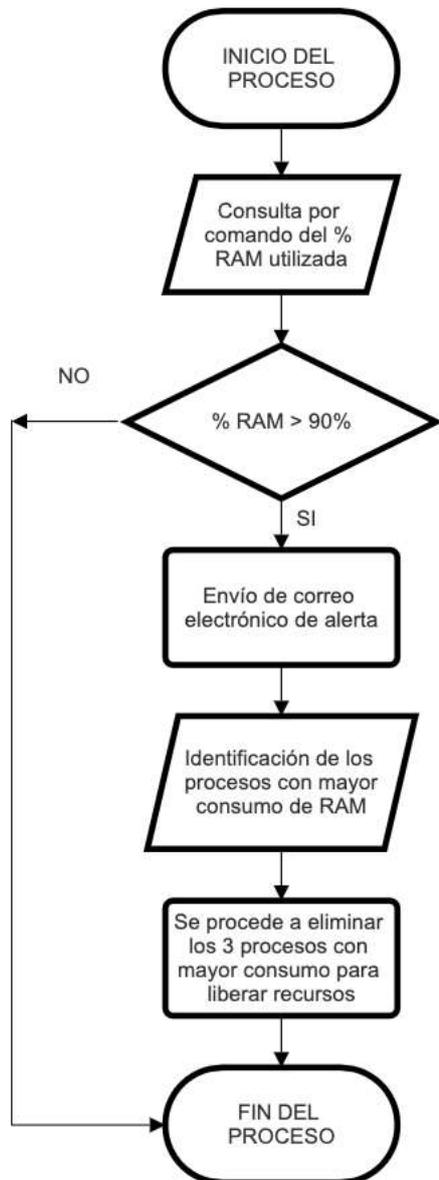
Para este script se ha definido por defecto los parámetros del límite máximo permitido para el consumo de CPU y el número de procesos a eliminar por motivo de pruebas y evaluación del script en un entorno de prueba. Estos parámetros pueden ser variables de acuerdo con el juicio que tenga el administrador del servidor.

3.3.2 Script para sobrecarga de RAM

Para la creación del script que mitigue la sobrecarga por el alto uso de memoria RAM por parte del servidor se realizó por medio de un diagrama de flujo secuencial los pasos a seguir para controlar que el equipo no satura al máximo este recurso del sistema. En la Figura 29 se presenta un diagrama de flujo que contiene la estructura del código de Python.

Figura 29

Diagrama de flujo de script para sobrecarga de RAM



Elaborado por: Autor.

El script primero consulta por medio de la función `virtual_memory` de la librería `psutil` el porcentaje de RAM utilizado en ese momento para poder validar que efectivamente el equipo se encuentra con la RAM saturada, si la condición establecida por defecto que es cuando el porcentaje de RAM actual sea mayor al 90% de verdadera procederá a enviar un correo electrónico por medio de la librería `EmailMessage` de Python para alertar a los usuarios requeridos que el servidor está teniendo un consumo superior al establecido

con anterioridad que fue 90%, este parámetro se puede editar en la cabecera de ser necesario otro tipo de límite para el control del sistema.

Una vez enviado el correo de alerta a los usuarios pertinentes se procederá a consultar los procesos que más estén consumiendo memoria RAM para filtrar los 3 primeros los cuales posteriormente ir ciclando para que por medio de la instrucción `os.kill` ir eliminando cada uno usando el PID del proceso y el parámetro 9 que notifica que el proceso termine de inmediato su ejecución.

3.3.3 Script para bajo espacio en disco

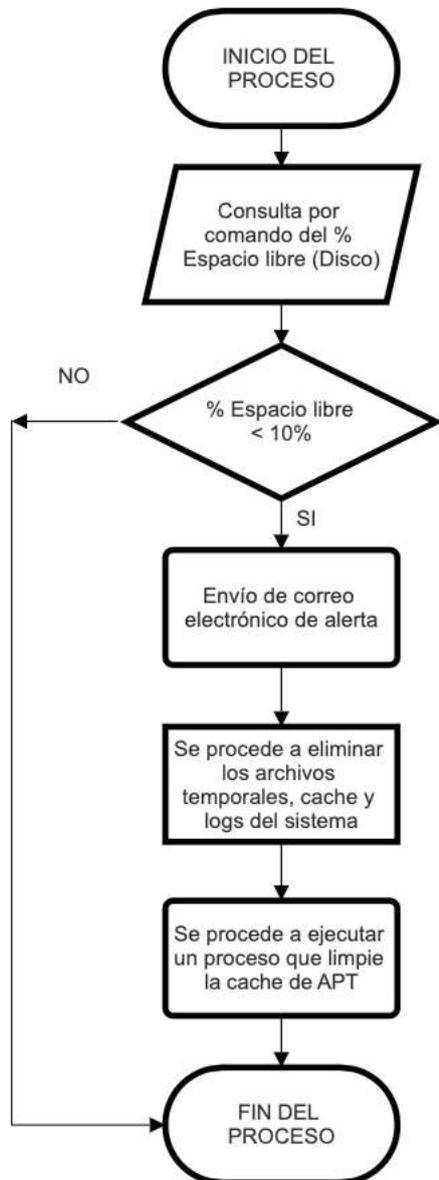
Para la creación del script que permita liberar espacio en disco del servidor es necesario primero determinar y analizar todas las rutas posibles de archivos temporales o cache en el servidor de Ubuntu para lo cual se recomienda también el poder agregar aparte de las aquí analizadas también directorios que contengan logs de sistemas web que puedan llegar a consumir una cantidad considerable de memoria en disco debido a que estas con el tiempo van creciendo a un ritmo considerable de acuerdo al tamaño del sistema que estén alojando y a la programación del mismo para el guardado de logs informativos o de errores.

En sistemas operativos Linux como lo es Ubuntu los archivos temporales del sistema se encuentran en el directorio `/tmp` y `/var/tmp` además también existen los archivos caches y logs propios del sistema los cuales también pueden ser eliminados para dar un poco de espacio libre al disco.

Algunas distribuciones también poseen una cache de APT la cual almacena localmente información sobre los paquetes de softwares disponibles por lo que contiene datos como listas de paquetes disponible e información detallada de los mismos. En la Figura 30 se presentará un diagrama de flujo el cual muestra secuencialmente los pasos seguidos de acuerdo a lo anteriormente planteado para el barrido de datos que consuman memoria cuando el espacio en disco está por debajo del límite establecido.

Figura 30

Diagrama de flujo de script para espacio en disco limitado



Elaborado por: Autor.

El script primero obtiene el espacio en disco disponible por medio de la función `disk_usage` de la librería `shutil` a la cual pasamos como parámetro la raíz del sistema de archivo para obtener el espacio total y disponible del sistema con el cual podemos determinar el porcentaje de espacio libre del equipo para poder establecer la condición que determine que si el % Espacio disponible de disco es menor al 10% proceda a limpiar los archivos temporales del sistema para liberar espacio, junto con la cache y logs del sistema.

3.3.4 Creación de monitores para control de parámetros

Los monitores de Zabbix nos van a permitir establecer parámetros para monitorear algunos componentes del sistema para lo cual es de mucha importancia su correcta creación para poder depender de los mismos aparte de que nos provee información a tiempo real de estado de algunos componentes esenciales para un buen funcionamiento del sistema.

Primero procedemos a crear un monitor o ítem para supervisar el estado del consumo de CPU para lo cual primero nos tenemos que ir al apartado de monitores del equipo y darle al botón en la parte superior izquierda que dice Crear monitor y nos debería presentar una pantalla similar a la de la Figura 31 en donde procederemos a ingresar un ítem con la siguiente key llamada `system.cpu.util[all, user, avg1]` el cual realizar una consulta del consumo de todos los núcleos del procesador, el segundo parámetro ponemos por defecto el nombre de user y el tercero corresponde al promedio del tiempo en minutos que tomara en cuenta para proporcionar el valor por lo que para tener un valor de alguna saturación de recursos en el procesador cuando apenas haya pasado 1 minuto de haber estado en constante procesamiento para su alerta.

El tipo de información a colocar en el monitor es de tipo numérico (coma flotante) debido a que en ese formato nos lo proporciona la key previamente configurada, también es muy importante validar que la interfaz del equipo sea la correcta en este entorno de prueba estoy apuntando a mi localhost a su vez hay que determinar la unidad utilizada de acuerdo con el dato recibido y el intervalo de actualización que en este caso será de 1 minuto.

La herramienta también nos permite establecer los parámetros de histórico y tendencias los cuales nos sirven para tener un registro de los datos monitoreados por el ítem para poder realizar informes estadísticos de acuerdo al tiempo requerido por el usuario y finalmente colocamos una breve descripción de la funcionalidad que realiza el ítem. En la figura 31 se presenta una captura de la configuración realizada.

Figura 31

Configuración de ítem para CPU

The screenshot shows the Zabbix Monitor configuration page for a CPU item. The main configuration fields are as follows:

- Métricas principales:** Control de congestión
- Nombre:** % Uso de CPU
- Tipo:** Agente Zabbix
- Monitor:** system.cpu.util[all,user,avg1]
- Tipo de información:** Numérico (coma flotante)
- Interfaz de equipo:** 127.0.0.1:10050
- Unidad:** %
- Intervalo de actualización:** 1m

Intervalos personalizados:

Tipo	Planificación	Intervalo	Periodo	Acción
Flexible	Planificación	50s	1-7,00:00-24:00	Eliminar

[Agregar](#)

Tempo de espera: Global Anular 3s [Tiempos de espera](#)

Histórico: No almacenar Almacenar hasta

Tendencias: No almacenar Almacenar hasta

Asignación de valores: [Selecciones](#)

Relena el campo de inventario del equipo: -Ninguno-

Descripción: Proporciona el porcentaje del uso del CPU tomando como referencia el promedio del consumo de 1 minuto del procesador

Elaborado por: Autor.

Una vez terminado el ítem para el CPU procedemos a realizar un ítem para el consumo de memoria RAM del equipo para lo cual requerimos del uso de la siguiente key llamada `vm.memory.size[pavailable]` la cual nos permite consultar el porcentaje de espacio de memoria RAM disponible en el servidor virtualizado el cual realizar una consulta del consumo de todos los núcleos del procesador.

El segundo parámetro ponemos por defecto el nombre de user y el tercero corresponde al promedio del tiempo en minutos que tomara en cuenta para proporcionar el valor por lo que para tener un valor de alguna saturación de recursos en el procesador cuando apenas haya pasado 1 minuto de haber estado en constante procesamiento para su alerta.

Al ítem de la memoria RAM también le definimos la unidad en porcentaje debido a que el tipo de información obtenida por la key es el porcentaje de memoria disponible el intervalo de actualización será de un minuto para poder realizar las pruebas correspondientes al sistema de control de congestión. La Figura 32 presente la configuración mencionada del ítem para el monitoreo de RAM.

Figura 32

Configuración de ítem para RAM

The screenshot shows the configuration interface for a monitor item in Nagios XI. The configuration is as follows:

- Métricas principales:** Control de congestión
- Nombre:** % Memoria RAM disponible
- Tipo:** Agente Zabbix
- Monitor:** vm.memory.size[pavailable]
- Tipo de información:** Numérico (coma flotante)
- Interfaz de equipo:** 127.0.0.1:10050
- Unidad:** %
- Intervalo de actualización:** 1m
- Intervalos personalizados:**

Tipo	Intervalo	Periodo	Acción	
Flexible	Planificación	50s	1-7,00:00-24:00	Eliminar
- Tiempo de espera:** Global, Actual: 3s, [Tiempos de espera](#)
- Histórico:** No almacenar, Almacenar hasta: 7d
- Tendencias:** No almacenar, Almacenar hasta: 365d
- Asignación de valores:** [Selección](#)
- Relena el campo de inventario del equipo:** -Ninguno-
- Descripción:** Proporciona el porcentaje de memoria RAM disponible en el servidor Ubuntu con una tasa de actualización de 1 minuto

Elaborado por: Autor.

Por último, procedemos a crear el ítem para obtener el porcentaje de espacio disponible en disco del servidor para lo cual necesito la siguiente key: `vfs.fs.size[/,pfree]` la cual nos provee el porcentaje de espacio disponible en disco de todo el sistema debido al parámetro `pfree` y también colocando la raíz del sistema de archivos para que me proporcione el porcentaje libre tomando de referencia todo el almacenamiento del sistema.

También es requerido colocar el tipo de información correspondiente a numérico (coma flotante) debido a que recibimos en ese formato el valor y como unidad el porcentaje, el intervalo de actualización en este escenario puede ser de acuerdo al criterio del administrador debido a que el espacio de almacenamiento generalmente tiene un menor porcentaje de cambio en el tiempo que con respecto a la CPU y a la RAM debido a que estas tienen fluctuaciones mayores por su uso constante por lo cual se la dejó en 1 minuto para pruebas en el servidor virtualizado. En la Figura 33 se presenta la configuración mencionada del ítem para el espacio en disco.

Figura 33

Configuración de ítem para Espacio en disco

The screenshot shows the configuration page for a monitor item in Nagios Core. The configuration is as follows:

- Métricas principales:** Control de congestión
- Nombre:** % Espacio disponible en disco
- Tipo:** Agente Zabbix
- Monitor:** vfs.fs.size[/,pfree]
- Tipo de información:** Numérico (coma flotante)
- Interfaz de equipo:** 127.0.0.1:10050
- Unidad:** %
- Intervalo de actualización:** 1m
- Intervalos personalizados:**

Tipo	Intervalo	Periodo	Acción
Flexible	Planificación	50s	1-7:00:00-24:00
- Tiempo de espera:** Global, Anular, 3s
- Histórico:** No almacenar, Almacenar hasta 7d
- Tendencias:** No almacenar, Almacenar hasta 365d
- Asignación de valores:** (Empty)
- Relena el campo de inventario del equipo:** -Ninguno-
- Descripción:** Proporciona el porcentaje disponible en disco del servidor Ubuntu con una tasa de actualización de 1 minuto por censado.

Elaborado por: Autor.

En el apartado de monitores del sistema debería poder visualizar lo que se presenta en la Figura 34 donde se muestran todos los monitores activos y en ejecución.

Figura 34

Lista de monitores



Nombre	Estado	Último	Problema	Severidad	Tipo	Estado	Acciones
Monitor de temperatura y uso de CPU	Activado	100%	7d	Alto	Agente Zabbix	Activado	Enviar correo electrónico
Monitor de temperatura y uso de memoria RAM	Activado	100%	7d	Alto	Agente Zabbix	Activado	Enviar correo electrónico
Monitor de temperatura y uso de espacio en disco	Activado	100%	7d	Alto	Agente Zabbix	Activado	Enviar correo electrónico

Elaborado por: Autor.

3.3.5 Creación de iniciadores para control de parámetros

Los iniciadores de Zabbix nos van a permitir establecer expresiones que de acuerdo a su evaluación pueden verdaderas o falsas las cuales nos sirven para realizar alertas o ejecutar acciones en el escenario donde la expresión realizada sea verdadera. Estos iniciadores son de mucha importancia debido a que por medio de ellos podemos establecer acciones posteriormente cuando se ejecuten.

Para realizar un iniciador que se active cuando el porcentaje de uso del CPU este superior al 90% es necesario utilizar los valores obtenidos por el ítem de CPU creado anteriormente para lo cual establecemos primero la expresión del problema que requerimos para activar el iniciador el cual será declarado de la siguiente forma `avg(/Zabbix server/system.cpu.util[all, user, avg], 1m) > 90` la cual se traduce a que tomamos el promedio de los valores generados del ítem que creamos para obtener el porcentaje de uso del CPU por el tiempo de 1 minuto, este valor puede ser parametrizado directamente en esta configuración de acuerdo a lo que requiera el sistema.

La expresión evalúa que cuando se cumpla la condición establecida se generara un evento que alertara del problema en el sistema de acuerdo con la clasificación de gravedad que le asignemos en este escenario le colocamos una gravedad Alta debido a que puede ser peligroso que el sistema este mucho tiempo con el CPU sobrecargado.

Luego procedemos a realizar la expresión de recuperación para que la alerta se desactive cuando la expresión sea contraria a la establecida inicialmente por lo que usamos el mismo formato solo que con la expresión

invertida para que se active cuando el CPU se encuentre en los límites establecidos. En la Figura 35 se muestra la configuración realizada.

Figura 35

Configuración de iniciador para CPU

The screenshot shows the configuration page for a Zabbix trigger named 'Alto uso de CPU'. The interface includes the following fields and options:

- Nombre:** Alto uso de CPU
- Nombre del evento:** Alto uso de CPU
- Datos operativos:** (Empty field)
- Gravedad:** Radio buttons for 'No clasificada', 'Informativa', 'Advertencia', 'Promedio', 'Alta' (selected), and 'Crítica'.
- Expresión de problema:** `avg(/Zabbix_server/system.cpu.util[all,user,avg],1m) > 90` with an 'Agregar' button.
- Generación del evento OK:** Radio buttons for 'Expresión', 'Expresión de recuperación', and 'Ninguno'.
- Expresión de recuperación:** `avg(/Zabbix_server/system.cpu.util[all,user,avg],1m) < 90` with an 'Agregar' button.
- Modo de generador de eventos de PROBLEMA:** Radio buttons for 'Único' and 'Múltiple'.
- Cierre del evento OK:** Radio buttons for 'Todos los problemas' and 'Todos los problemas si los valores coinciden'.
- Permitir cierre manual:** A checked checkbox.

Elaborado por: Autor.

Luego procedemos a realizar el iniciador para cuando la memoria RAM este saturada para lo cual utilizamos la siguiente expresión `avg(/Zabbix_server/vm.memory.size[pavailable], 1m) > 90` el cual tiene un funcionamiento parecido al del iniciador del CPU debido a que utilizamos el mismo formato de expresión solo que colocamos el ítem que nos permite obtener el porcentaje de memoria RAM disponible para tomar el promedio de los valores en 1 minuto y colocamos como valor que activa el iniciador el 90 que corresponde al porcentaje limite definido el cual si se supera se procede activar la alerta.

A su vez también realizamos la expresión de recuperación para que el estado de alerta se desactive cuando la expresión sea evaluada en verdadero para que así regrese a un estado de OK, establecemos la gravedad como alta

para tener estandarizado todos los iniciadores del sistema de monitoreo. En la Figura 36 se presenta la configuración realizada.

Figura 36

Configuración de iniciador para RAM

The screenshot shows the configuration page for a Nagios XI trigger named 'Baja memoria RAM disponible'. The interface includes the following fields and options:

- Nombre:** Baja memoria RAM disponible
- Nombre del evento:** Baja memoria RAM disponible
- Datos operativos:** (Empty text area)
- Gravedad:** No clasificada, Informativa, Advertencia, Promedio, **Alta** (selected), Crítica
- Expresión de problema:** `avg(/Zabbix:server/vm.memory.size[pavailable],1m) < 10` (with an 'Agregar' button)
- Constructor de expresiones:** (Link to the expression builder)
- Generación del evento OK:** Expresión, Expresión de recuperación, Ninguno
- Expresión de recuperación:** `avg(/Zabbix:server/vm.memory.size[pavailable],1m) > 10` (with an 'Agregar' button)
- Constructor de expresiones:** (Link to the expression builder)
- Modo de generador de eventos de PROBLEMA:** Único, **Múltiple** (selected)
- Cierre del evento OK:** Todos los problemas, Todos los problemas si los valores coinciden
- Permitir cierre manual:**

Elaborado por: Autor.

Para finalizar la configuración de los iniciadores realizamos una última expresión para validar el porcentaje de espacio en disco disponible usando el ítem creado para obtener el porcentaje de espacio disponible del equipo lo que nos daría como resultado la siguiente expresión `last(/Zabbix:server/vfs.fs.size[/,pfree]) < 10` la cual activara la alerta cuando el ultimo valor obtenido por el ítem de como resultado un porcentaje de espacio libre inferior al 10%, también configuramos la expresión de recuperación cuando el estado del porcentaje de disco vuelva a estar dentro del límite permitido que es superior al 10% del porcentaje de espacio en disco del equipo. La Figura 37 presenta la configuración establecida para el espacio en disco.

Figura 37

Configuración de iniciador para Espacio en disco

The screenshot shows the configuration page for a trigger named 'Bajo espacio en disco'. The interface includes the following fields and options:

- Nombre:** Bajo espacio en disco
- Nombre del evento:** Bajo espacio en disco
- Datos operativos:** (Empty field)
- Gravedad:** No clasificada, Informativa, Advertencia, Promedio, **Alta**, Critica
- Expresión de problema:** `last(/Zabbix_server/vfs.fs.size(/,pfree)) < 10` (with an 'Agregar' button)
- Expresión de recuperación:** `last(/Zabbix_server/vfs.fs.size(/,pfree)) > 10` (with an 'Agregar' button)
- Generación del evento OK:** Expresión, Expresión de recuperación, Ninguno
- Modo de generador de eventos de PROBLEMA:** Único, Múltiple
- Cierre del evento OK:** Todos los problemas, Todos los problemas si los valores coinciden
- Permitir cierre manual:**

Elaborado por: Autor.

En el apartado de iniciadores del sistema debería poder visualizar lo que se presenta en la Figura 38 donde se muestran todos los iniciadores configurados junto con la información de las expresiones que permiten que activen las alertas.

Figura 38

Lista de iniciadores

The screenshot shows a table listing configured triggers. The table has columns for 'Estado', 'Nombre', 'Descripción', 'Expresión', 'Gravedad', 'Operaciones', and 'Acciones'. Three triggers are listed:

Estado	Nombre	Descripción	Expresión	Gravedad	Operaciones	Acciones
OK	Alto uso de CPU		Problema: <code>avg(CpuUsr[server]/100) > 90</code> Recuperar: <code>avg(CpuUsr[server]/100) < 90</code>	Alta		Enviar mensaje
OK	Alto consumo de memoria		Problema: <code>avg(CpuMem[server]/100) > 90</code> Recuperar: <code>avg(CpuMem[server]/100) < 90</code>	Alta		Enviar mensaje
OK	Bajo espacio en disco		Problema: <code>last(/Zabbix_server/vfs.fs.size(/,pfree)) < 10</code> Recuperar: <code>last(/Zabbix_server/vfs.fs.size(/,pfree)) > 10</code>	Alta		Enviar mensaje

Elaborado por: Autor.

3.3.6 Creación de comandos

Para poder establecer que Zabbix pueda realizar una acción de ejecutar comando en nuestro servidor Ubuntu es necesario definir las instrucciones a ejecutar por consola para lo cual hemos establecido directorios en el sistema del servidor que contendrán archivos con extensión .sh debido a que contendrán la secuencia de comandos que ejecutaran los scripts de Python para liberar al equipo de la sobrecarga que posea en ese momento.

Para la creación de estos comandos seguiremos la misma estructura seleccionando el tipo como comando y como instrucción colocaremos la ruta del sistema donde se encuentra nuestro archivo que contiene la secuencia de comandos requerida de acuerdo con el tipo de alerta en las siguientes figuras se mostrara las configuraciones realizadas de acuerdo al tipo operación que se requiera accionar.

Figura 39

Comando para CPU

The screenshot shows the Zabbix configuration interface for creating a command. The form is titled "Comando" and includes the following fields and options:

- Nombre:** Script para CPU
- Ámbito:** Operación de acción (selected), Acción de equipo manual, Acción de evento manual
- Tipo:** Webhook, Comando (selected), SSH, Telnet, IPMI
- Ejecutar en:** Agente Zabbix, Proxy o servidor Zabbix, Servidor Zabbix (with a warning icon)
- Instrucciones:** /usr/local/bin/zabbix-scripts/control_cpu.sh
- Descripción:** Script en Python para gestionar los procesos que consumen una gran cantidad de recursos en el servidor Ubuntu
- Grupo del equipo:** Todo (dropdown menu)

Elaborado por: Autor.

Figura 40

Comando para RAM

The screenshot shows the 'Comando' configuration page in Zabbix. The 'Nombre' field contains 'Script para RAM'. The 'Ámbito' is set to 'Operación de acción'. The 'Tipo' is 'Comando'. The 'Ejecutar en' is 'Agente Zabbix'. The 'Instrucciones' field contains the command: `/usr/local/bin/zabbix-scripts/control_ram.sh`. The 'Descripción' field contains: 'Script de Python para gestionar todas las operaciones que consumen un gran porcentaje de memoria RAM en el servidor Ubuntu'. The 'Grupo del equipo' is set to 'Todo'.

Elaborado por: Autor.

Figura 41

Comando para espacio en disco

The screenshot shows the 'Comando' configuration page in Zabbix. The 'Nombre' field contains 'Script para espacio en disco'. The 'Ámbito' is set to 'Operación de acción'. The 'Tipo' is 'Comando'. The 'Ejecutar en' is 'Agente Zabbix'. The 'Instrucciones' field contains the command: `/usr/local/bin/zabbix-scripts/control_disco.sh`. The 'Descripción' field contains: 'Script de Python para limpiar cache y archivos temporales que puedan liberar espacio en disco del servidor Ubuntu'. The 'Grupo del equipo' is set to 'Todo'.

Elaborado por: Autor.

3.3.7 Creación de acciones con iniciadores

Una vez ya configurado los monitores, iniciadores y comandos podemos llegar a la última configuración requerida para la automatización del sistema de control de congestión del servidor virtualizado la cual consiste en unir los iniciadores junto con los comandos realizados en Zabbix para lo cual existe un apartado que nos permite crear acciones a partir de iniciadores de acuerdo a la condición que establezcamos inicialmente para lo cual de acuerdo a la expresión de cada iniciador podemos definir que cuando el iniciador este activo y el problema que tiene no se suprime lo cual nos da a comprender que aún no se ha solventado la sobrecarga del mismo se procede a ejecutar el comando definido en el apartado de operaciones.

Realizamos el proceso mencionado para crear las 3 acciones de acuerdo a los iniciadores correspondientes asociando los comandos con sus respectivos iniciadores. Las siguientes mostraran las condiciones establecidas para su ejecución.

Figura 42

Acción para ejecución de comando para CPU

The screenshot shows the 'Acción' configuration page in Zabbix. The title is 'Acción' and there are two tabs: 'Acción' and 'Operaciones 1'. The main configuration area includes:

- Nombre:** Controlar alto uso de CPU
- Tipo de cálculo:** Y (dropdown menu), A and B
- Condiciones:** A table with two columns: 'Etiqueta' and 'Nombre'.

Etiqueta	Nombre
A	Iniciador igual a <i>Zabbix server: Alto uso de CPU</i>
B	El problema no se suprime

Below the table is a link labeled 'Agregar'.
- Activado:**

At the bottom, there is a note: '* Debe existir al menos una operación.'

Elaborado por: Autor.

Figura 43

Acción para ejecución de comando para RAM

The screenshot shows a configuration form for an action named 'Controlar baja memoria RAM'. The form includes a 'Nombre' field with the value 'Controlar baja memoria RAM', a 'Tipo de cálculo' dropdown set to 'Y' with 'A and B' as a secondary label, and a 'Condiciones' table. The table has two columns: 'Etiqueta' and 'Nombre'. It contains two rows: 'A' with the name 'Iniciador igual a Zabbix server: Baja memoria RAM disponible' and 'B' with the name 'El problema no se suprime'. There is an 'Agregar' link below the table. The 'Activado' checkbox is checked. A note at the bottom states '* Debe existir al menos una operación.'

Etiqueta	Nombre
A	Iniciador igual a Zabbix server: Baja memoria RAM disponible
B	El problema no se suprime

Elaborado por: Autor.

Figura 44

Acción para ejecución de comando para espacio en disco

The screenshot shows a configuration form for an action named 'Controlar bajo espacio en disco'. The form includes a 'Nombre' field with the value 'Controlar bajo espacio en disco', a 'Tipo de cálculo' dropdown set to 'Y' with 'A and B' as a secondary label, and a 'Condiciones' table. The table has two columns: 'Etiqueta' and 'Nombre'. It contains two rows: 'A' with the name 'Iniciador igual a Zabbix server: Bajo espacio en disco' and 'B' with the name 'El problema no se suprime'. There is an 'Agregar' link below the table. The 'Activado' checkbox is checked. A note at the bottom states '* Debe existir al menos una operación.'

Etiqueta	Nombre
A	Iniciador igual a Zabbix server: Bajo espacio en disco
B	El problema no se suprime

Elaborado por: Autor.

3.4 Evaluación y pruebas

Para la evaluación del sistema de control de congestión se planteó el uso de herramientas como stress-ng que nos permite realizar pruebas de estrés en el sistema para poder validar que todos los comandos se inicien correctamente y realicen sus funciones respectivas para poder aplicar las contramedidas establecidas por medio del código de Python y obtener las alertas preestablecidas por la plataforma de Zabbix. Stress-ng nos permite evaluar cómo responde el hardware y el software del equipo en situaciones donde exista un alto uso de recursos.

Para la evaluación de la memoria RAM se empleó el comando presentado en la Figura 45 el cual nos permite configurar la carga de trabajo que tendrá nuestro sistema para lo cual establecimos que la herramienta genere 3 procesos que consuman la memoria RAM y a su vez cada proceso intentará utilizar el 100% de la memoria disponible del sistema para saturar la misma y colocaremos al final un tiempo de prueba de unos 5 minutos para validar el rendimiento del equipo.

Figura 45

Ejecución del comando stress-ng para sobrecargar la memoria RAM

```
jair@localhost:~$ stress-ng --vm 3 --vm-bytes 100% --timeout 300s
stress-ng: info: [6916] setting to a 5 mins, 0 secs run per stressor
stress-ng: info: [6916] dispatching hogs: 3 vm
```

Elaborado por: Autor.

Posterior a la ejecución del comando se pudo validar desde la plataforma de Zabbix la ejecución del script para contrarrestar la sobrecarga, en la pantalla de registro de acciones se pudo comprobar que la instrucción para la ejecución del comando se realizó exitosamente, en la Figura 46 se presenta la acción realizada por Zabbix dado el evento de sobrecarga generado por la memoria RAM. El script de Python envió un correo para notificar el evento que activo la acción para contrarrestar la sobrecarga del recurso pertinente. En la Figura 47 se muestra el correo que se recibió de la alerta personalizada.

Figura 46

Registro de acción ejecutada para controlar la baja memoria RAM disponible



Elaborado por: Autor.

Figura 47

Correo recibido por alerta de memoria RAM insuficiente

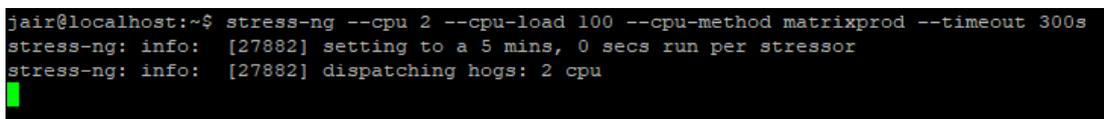


Elaborado por: Autor.

El comando `stress-ng` también nos permite poner a prueba el procesamiento del equipo mediante cargas de procesos a los núcleos del CPU, en la Figura 48 se muestra cómo se establecieron los parámetros requeridos para generar una carga para lo cual utilizamos todos los núcleos del CPU los cuales en su totalidad son 2 para lograr sobrecargar todo el procesador y se estableció que ambos tengan una carga del 100 % mediante el método de cálculo de productos de matrices durante un tiempo de 5 minutos para probar el rendimiento de sistema.

Figura 48

Ejecución del comando `stress-ng` para sobrecargar la CPU

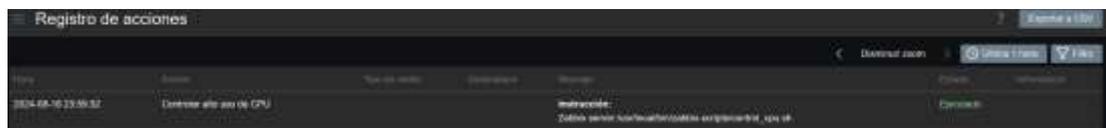


Elaborado por: Autor.

Después de la ejecución del comando para poner a prueba el rendimiento del CPU nos mostrara el registro de la acción realizada para contrarrestar el alto consumo del CPU, en la Figura 49 se presenta el registro de la acción en un estado de ejecutado. El script de Python envió un correo para notificar el evento que activo la acción para contrarrestar la sobrecarga del recurso pertinente. En la Figura 50 se muestra el correo que se recibió de la alerta personalizada de la acción ejecutada.

Figura 49

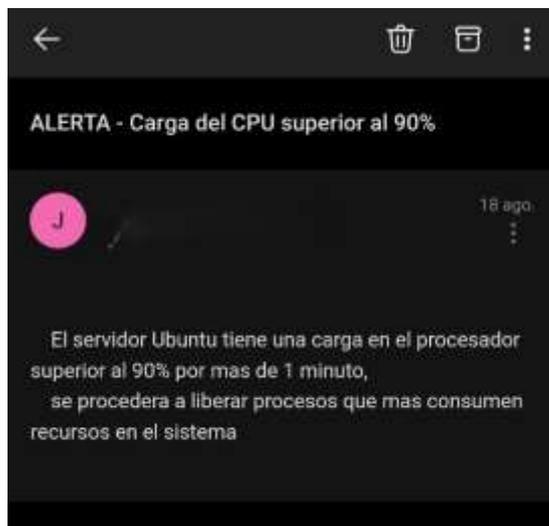
Registro de acción ejecutada para controlar el alto consumo de CPU



Elaborado por: Autor.

Figura 50

Correo recibido por alerta de alto consumo de CPU



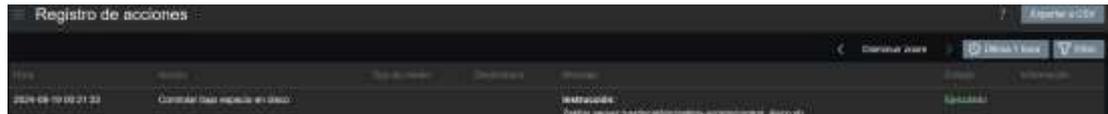
Elaborado por: Autor.

Para evaluar el escenario donde el sistema tuviera el espacio en disco saturado se realizó por comandos inserciones de logs para simular la operatividad de los sistemas, los cuales después de un tiempo la cantidad logs informativos que realizan algunos sistemas web y algunas APIs instanciadas en el servidor provoca que este con el tiempo tenga menos almacenamiento disponible. Para lo cual el sistema de control de congestión procedió a ejecutar

automáticamente la acción por medio de Zabbix tal y como se muestra en la Figura 51 donde se logra visualizar que el comando se ejecutó correctamente. También el script de Python realizó el envío de correo respectivo por la alerta correspondiente, en la Figura 52 se presenta el correo recibido.

Figura 51

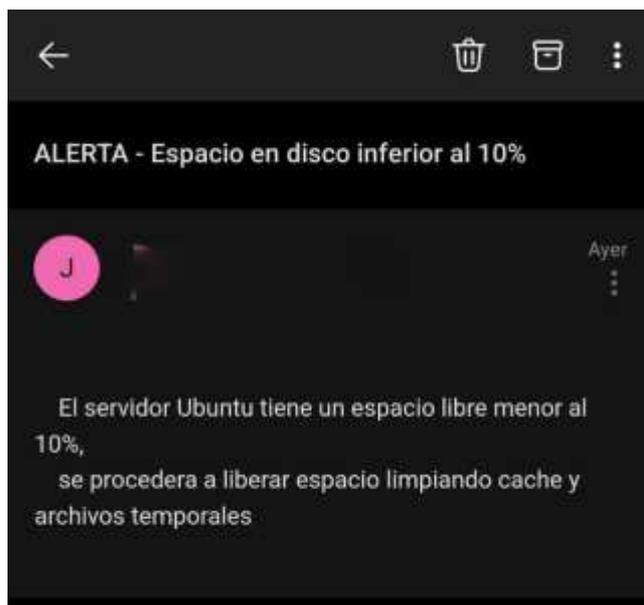
Registro de acción ejecutada para controlar el espacio en disco insuficiente



Elaborado por: Autor.

Figura 52

Correo recibido por alerta de espacio en disco insuficiente



Elaborado por: Autor.

CAPÍTULO 4: CONCLUSIONES Y RECOMENDACIONES

4.1 Conclusiones

El uso de la herramienta de Zabbix para el monitoreo del estado del servidor ha sido de mucha utilidad para poder aplicar diversas reglas para establecer parámetros que nos permitan realizar acciones predefinidas para solventar varios escenarios posibles de acuerdo a las necesidades del administrador, por medio de esta herramienta se ha logrado aplicar acciones de contra medida para las sobrecargas de CPU, memoria RAM y espacio en disco.

El sistema propuesto permite al usuario definir y personalizar los diferentes parámetros establecidos para el rango de operación aceptado debido a que este se divide en diferentes secciones con una funcionalidad específica cada uno desde el monitoreo que los realiza por medios de los ítems hasta las alertas las cuales son activadas por los triggers.

Los resultados obtenidos han podido validar la eficacia de la integración de código en Python con la herramienta Zabbix para lo cual se presentará a continuación las gráficas obtenidas por la herramienta para su análisis respectivo.

La Figura 53 corresponde al estado del uso del CPU cuando el iniciador para el control de congestión esta activado por lo cual se logra evidenciar una bajada en la carga del uso del CPU muy notable debido a que el sistema solo estaba usando casi toda su capacidad en solo un proceso el cual lo realizamos con la prueba de estrés.

Figura 53

Grafica del uso de CPU cuando está habilitada la acción de control

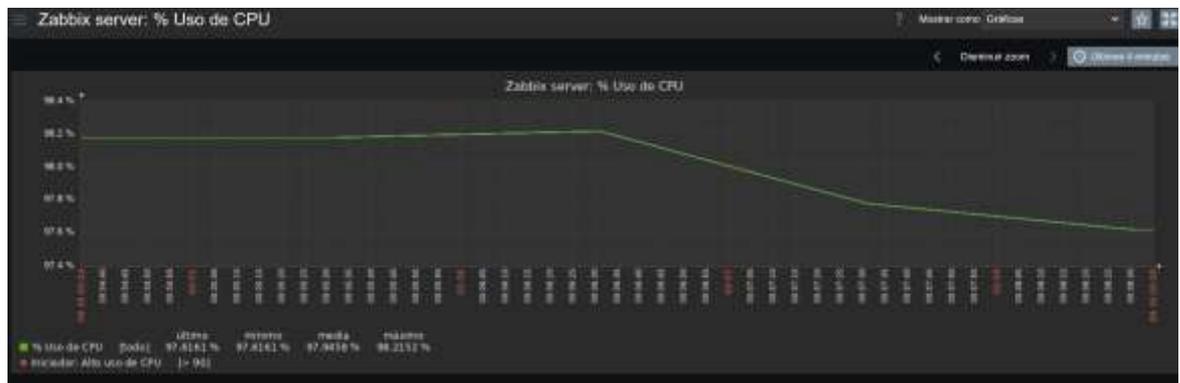


Elaborado por: Autor.

La Figura 54 corresponde al estado del uso del CPU cuando el iniciador estaba desactivado para lo cual se pudo validar que el equipo se mantuvo con un uso del CPU por un tiempo considerable lo que podemos interpretar como una baja eficiencia en tareas que el servidor tenga que realizar aparte de las que generan un alto consumo.

Figura 54

Grafica del uso de CPU cuando esta deshabilitada la acción de control



Elaborado por: Autor.

La Figura 55 corresponde al estado del porcentaje de memoria RAM disponible cuando el iniciador para el control de congestión esta activado lo que se visualiza al inicio es el poco espacio porcentaje de memoria RAM disponible para lo cual el sistema tiene que usar espacio en disco como memoria RAM de respaldo lo cual es muy ineficiente debido a que este es más lento en velocidad de transmisión de datos que la RAM.

Figura 55

Grafica del porcentaje de memoria RAM disponible cuando está habilitada la acción de control



Elaborado por: Autor.

La Figura 56 corresponde al estado del porcentaje de memoria RAM disponible cuando la acción de control esta desactivada por lo que se logra ver varios picos en donde en un breve espacio de tiempo hay un porcentaje considerable de memoria RAM, pero en gran parte existe una saturación de la misma lo que puede provocar que algunos programas trabajen de una manera no tan eficiente debido a la rapidez que provee cargar los archivos en la RAM que con respecto al disco.

Figura 56

Grafica del porcentaje de memoria RAM disponible cuando esta deshabilitada la acción de control



Elaborado por: Autor.

La Figura 57 corresponde al estado del porcentaje de espacio en disco disponible cuando el iniciador para el control de congestión esta activado, al principio podemos notar que el espacio disponible era inferior al 10% debido

a la cantidad de logs generados por consola para lo cual después del barrido de los mismos junto a archivos temporales se logró recuperar un gran porcentaje del espacio disponible, como recomendación se menciona la posibilidad de agregar nuevas rutas donde se encuentren logs de varios sistemas web establecidos en el servidor además de directorios donde se ubiquen backups de base de datos que tengan un buen tiempo de haberse respaldado para un barrido de memoria.

Figura 57

Grafica del porcentaje de espacio disponible en disco cuando está habilitada la acción de control



Elaborado por: Autor.

La Figura 58 corresponde al estado del porcentaje de espacio en disco cuando no está habilitada la acción de control esta desactivada por lo que se logra observar una constante en lo que es el espacio limitado del disco lo que puede provocar a la larga que no tenga más espacio para almacenar los datos de los registros que posea el equipo, lo que iniciará algunos errores en algunos programas que requieran guardado en el sistema.

Figura 58

Grafica del porcentaje de espacio disponible en disco cuando esta deshabilitada la acción de control



Elaborado por: Autor.

4.2 Recomendaciones

La elección de herramientas open-source como Zabbix y Python ha sido de mucha utilidad para el desarrollo del sistema para lo cual se recomienda revisar toda la documentación que posee Zabbix en su página web oficial en la cual van sacando nuevas actualizaciones con mejoras que son de mucha utilidad para mantener nuestros servidores monitoreados y de esa manera poder validar que el servicio que brindemos sea de calidad.

En el caso del script para mitigar el bajo espacio en disco se recomienda eliminar también rutas de archivos que contengan datos únicamente informativos debido a que almacenar tantos datos, aunque no sean de mucho tamaño pueden llegar a la larga a ocupar un espacio significativo en el sistema. También se recomienda manejar listas de exclusiones en el código de Python para mantener procesos que sean de mucha importancia para el sistema y que solo los mismos administradores pueden realizar su eliminación manualmente.

La configuración realizada por comando de la herramienta de Zabbix puede ser automatizada para lo cual se recomienda el uso de archivos bash los cuales pueden automatizar gran parte del proceso de instalación y configuración de la herramienta para futuras implementaciones en sistemas que lo requieran.

BIBLIOGRAFÍAS

- Alves, L. (2015). Zabbix Performance Tuning. Packt Publishing Ltd.
- Arias Chaves, M. (2008). Percepción general de la virtualización de los recursos informáticos. *InterSedes: Revista de las Sedes Regionales*, IX (17), 147-171. Recuperado de <https://www.redalyc.org/pdf/666/66615066011.pdf>
- Barham, P., Dragovic, B., Fraser, K., Hand, S., Harris, T., Ho, A., ... & Warfield, A. (2003). Xen and the art of virtualization. *ACM SIGOPS Operating Systems Review*, 37(5), 164-177. <https://doi.org/10.1145/1165389.945462>
- Barros, F. (2022). Advanced Oracle monitoring agent for Zabbix (Master's thesis, Universidade do Porto (Portugal)).
- Buriticá, O. I., & Guerrero, L. E. M. (2021). Introducción a la programación con Python. Ra-Ma Editorial.
- CheckPoint Research. (2022). Cyber Attack Trends: 2022 Mid-Year Report. Recuperado de <https://blog.checkpoint.com/2022/07/20/cyber-attack-trends-2022-mid-year-report/>
- Chrony Project. (n.d.). Chrony - A versatile implementation of the Network Time Protocol (NTP). Chrony Project. <https://chrony-project.org/>
- Clark, C., Fraser, K., Hand, S., Hansen, J. G., Jul, E., Limpach, C., Pratt, I., & Warfield, A. (2005). Live Migration of Virtual Machines. *Proceedings of the 2nd Symposium on Networked Systems Design & Implementation*, 273-286.
- Dalle Vacche, A. (2015). Mastering Zabbix. Packt Publishing Ltd.
- Doña, J., García, J. E., López, J., Pascual, F., & Pascual, R. (2010). Virtualización de servidores—Una solución de Futuro. Campus Universitario de Teatinos. Málaga. España.

- Giampaolo Rodola. (n.d.). psutil 5.9.5 documentation. <https://psutil.readthedocs.io/en/latest/>
- Gonçalves, J. (2019). Mastering Ubuntu Server: Expert techniques to build, manage, and secure Ubuntu Server (2nd ed.). Packt Publishing.
- Hill, B. (2018). Ubuntu Server Administration. Apress.
- José Ortega, A. (2022). Aplicación de registro de PC's para eReuse. org (Bachelor's thesis, Universitat Politècnica de Catalunya).
- Kili, A. How to Impose High CPU Load and Stress Test on Linux Using 'Stress-ng' Tool. URL <https://www.tecmint.com/linux-cpu-load-stress-test-with-stress-ng-tool/>[accessed 06 Aug, 2024].
- Kissling, M. (2020). Ubuntu Server Essentials. Packt Publishing.
- Liefting, N., & van Baekel, B. (2024). Zabbix 7 IT Infrastructure Monitoring Cookbook: Explore the new features of Zabbix 7 for designing, building, and maintaining your Zabbix setup. Packt Publishing Ltd.
- Mardiyono, A., Sholihah, W., & Hakim, F. (2020). Mobile-based Network Monitoring System Using Zabbix and Telegram. Mobile-based Network Monitoring System Using Zabbix and Telegram, 3, 1,2. <https://doi.org/10.1109/IC2IE50715.2020.9274582>
- Microsoft. (s. f.). Virtualización anidada. Microsoft Learn. Recuperado de <https://learn.microsoft.com/es-es/virtualization/hyper-v-on-windows/user-guide/nested-virtualization>
- Olups, R. (2016). Zabbix Network Monitoring. Packt Publishing Ltd.
- Olups, R., Dalle Vacche, A., & Uytterhoeven, P. (2017). Zabbix: Enterprise Network Monitoring Made Easy. Packt Publishing Ltd.
- Python Software Foundation. (n.d.). os — Interfaces para funcionalidades específicas del sistema operativo. En Python 3.10. Documentación en español. <https://docs.python.org/es/3.10/library/os.html>

- Python Software Foundation. (n.d.). shutil — Operaciones de archivos de alto nivel. En Python 3.10. Documentación en español. <https://docs.python.org/es/3/library/shutil.html>
- Rey Vela, N. E. (2018). Implementación del algoritmo pulso en Python.
- Rey, C. F. (2023). Sistema de control de asistencia empresarial con inteligencia artificial por reconocimiento facial y análisis de sentimientos.
- ServerWatch. (2023). Top 10 Server Technology Trends for the New Decade. Recuperado de <https://www.serverwatch.com/trends/top-10-server-technology-trends-for-the-new-decade/>
- Solís, G. R., Trujillo-Romero, F., & Capulín, C. H. G (2023). Comparativa de frameworks para cómputo evolutivo implementados en Python.
- Sugerman, J., Venkitachalam, G., & Lim, B.-H. (2001). Virtualizing I/O Devices on VMware Workstation's Hosted Virtual Machine Monitor. USENIX Annual Technical Conference, 1-14
- VMblog. (2023). Top 5 Virtualization Trends That IT Pros Should Be Aware Of. Recuperado de <https://vmblog.com/archive/2023/05/12/top-5-virtualization-trends-that-it-pros-should-be-aware-of.aspx>
- Zabbix. (2024). Documentación Zabbix. Obtenido de Documentación Zabbix: <https://www.zabbix.com/documentation/current/es/manual/introduction/about>
- Zabbix. (2024). Enterprise Monitoring. Recuperado de https://www.zabbix.com/la/enterprise_monitoring

ANEXOS

Figura A1: Código de Python utilizado para mitigar la sobrecarga del CPU, parte 1

```
import psutil
import os
from email.message import EmailMessage
import ssl
import smtplib

# Definir el umbral de uso de CPU
CPU_limite = 90
# Número de procesos a eliminar (los de mayor consumo de CPU)
NUM_PROCESOS_ELIMINAR = 3

"""
Verifica el uso de CPU y elimina los procesos que están utilizando más CPU
si se supera el umbral definido.
"""
uso_cpu = psutil.cpu_percent(interval=1)
if uso_cpu > CPU_limite:
    """Notifica por correo cuando la memoria RAM disponible sea inferior al 10%."""
    email_sender = "user@gmail.com"
    password = "abcd abcd abcd abcd"
    email_reciver = "user@hotmail.com"
    subject = "ALERTA - Carga del CPU superior al 90%"
    body = """
El servidor Ubuntu tiene una carga en el procesador superior al 90% por mas de 1 minuto,
se procedera a liberar procesos que mas consumen recursos en el sistema
"""
    em = EmailMessage()
    em["From"] = email_sender
    em["To"] = email_reciver
    em["Subject"] = subject
    em.set_content(body)
    context = ssl.create_default_context()
    with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as smtp:
        smtp.login(email_sender, password)
        smtp.sendmail(email_sender, email_reciver, em.as_string())
```

Figura A2: Código de Python utilizado para mitigar la sobrecarga del CPU, parte 2

```
print(f"Alerta: Uso de CPU alto - {uso_cpu}%")
"""
Obtiene una lista de procesos que están utilizando más CPU.
"""
procesos = [(proc.pid, proc.info['name'], proc.info['cpu_percent'])
            for proc in psutil.process_iter(['name', 'cpu_percent'])]
# Ordenar los procesos por uso de CPU en orden descendente
procesos.sort(key=lambda x: x[2], reverse=True)
procesos_alto_consumo = procesos[:NUM_PROCESOS_ELIMINAR]
for pid, name, cpu in procesos_alto_consumo:
    try:
        os.kill(pid, 9) # Eliminar proceso
        print(f"Proceso eliminado: PID={pid}, Nombre={name}, CPU={cpu}%")
    except Exception as e:
        print(f"No se pudo eliminar el proceso {pid}: {e}")
```

Figura A3: Archivo bash usado para la activación del entorno virtual y el código en Python de la CPU

```
DIR: /usr/local/bin/zabbix-scripts/control_cpu.sh

#!/bin/bash
source /home/jair/zabbix-env/bin/activate
/home/jair/zabbix-env/bin/python /home/jair/scripts_control/cpu.py
```

Figura A4: Código de Python usado para mitigar la sobrecarga de la RAM, parte 1

```
import psutil
import os
from email.message import EmailMessage
import ssl
import smtplib

# Definir el umbral de uso de memoria
MEMORIA_LIMITE = 90
# Número de procesos a eliminar (los de mayor consumo de memoria)
NUM_PROCESOS_ELIMINAR = 3

uso_memoria = psutil.virtual_memory().percent
"""
    Verifica el uso de memoria y elimina los procesos que están utilizando más memoria
    si se supera el umbral definido.
"""
if uso_memoria > MEMORIA_LIMITE:
    """Notifica por correo cuando la memoria RAM disponible sea inferior al 10%. """
    email_sender = "user@gmail.com"
    password = "abcd abcd abcd abcd"
    email_reciver = "user@hotmail.com"
    subject = "ALERTA - Memoria RAM insuficiente"
    body = """
    El servidor Ubuntu tiene la memoria RAM a mas del 90%, de su capacidad por mas de 1 minuto,
    se procedera a liberar recursos de los procesos que mas consumen recursos
    """
    em = EmailMessage()
    em["From"] = email_sender
    em["To"] = email_reciver
    em["Subject"] = subject
    em.set_content(body)
    context = ssl.create_default_context()
    with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as smtp:
        smtp.login(email_sender, password)
        smtp.sendmail(email_sender, email_reciver, em.as_string())
```

Figura A5: Código de Python usado para mitigar la sobrecarga de la RAM, parte 2

```

print(f"Alerta: Uso de memoria alto - {uso_memoria}%")
"""
Obtiene una lista de procesos que están utilizando más memoria.
"""
procesos = [(proc.pid, proc.info['name'], proc.info['memory_percent'])
             for proc in psutil.process_iter(['name', 'memory_percent'])]
# Ordenar los procesos por uso de memoria en orden descendente
procesos.sort(key=lambda x: x[2], reverse=True)
proceso_alto_consumo = procesos[:NUM_PROCESOS_ELIMINAR]
for pid, name, memory in proceso_alto_consumo:
    try:
        os.kill(pid, 9) # Eliminar proceso
        print(f"Proceso eliminado: PID={pid}, Nombre={name}, Memoria={memory}%")
    except Exception as e:
        print(f"No se pudo eliminar el proceso {pid}: {e}")

```

Figura A6: Archivo bash para la activación del entorno virtual y el código en Python de la RAM

```

DIR: /usr/local/bin/zabbix-scripts/control_ram.sh

#!/bin/bash
source /home/jair/zabbix-env/bin/activate
/home/jair/zabbix-env/bin/python /home/jair/scripts_control/ram.py

```

Figura A7: Código de Python utilizado para mitigar el espacio en disco limitado, parte 1

```

import os
import shutil
import subprocess
from email.message import EmailMessage
import ssl
import smtplib

"""Verifica si el porcentaje de espacio libre en disco es menor que el umbral especificado."""
estado = shutil.disk_usage("/")
porcentajeEspacioLibre = (estado.free / estado.total) * 100
if porcentajeEspacioLibre < 10:
    """Notifica por correo cuando el espacio en disco sea inferior al 10%. """
    email_sender = "jairmonar@gmail.com"
    password = "lxpf v1wq uenv qzbu"
    email_reciver = "jair.m.i@hotmail.com"
    subject = "ALERTA - Espacio en disco inferior al 10%"
    body = """
    El servidor Ubuntu tiene un espacio libre menor al 10%,
    se procederá a liberar espacio limpiando cache y archivos temporales
    """
    em = EmailMessage()
    em["From"] = email_sender
    em["To"] = email_reciver
    em["Subject"] = subject
    em.set_content(body)
    context = ssl.create_default_context()
    with smtplib.SMTP_SSL("smtp.gmail.com", 465, context=context) as smtp:
        smtp.login(email_sender, password)
        smtp.sendmail(email_sender, email_reciver, em.as_string())

    """Limpia archivos en /tmp y /var/tmp."""
    tmp_dir = '/tmp'
    for root, dirs, files in os.walk(tmp_dir):
        for name in files:
            file_path = os.path.join(root, name)
            try:
                os.remove(file_path)
                print(f"Archivo eliminado: {file_path}")
            except Exception as e:
                print(f"No se pudo eliminar el archivo {file_path}: {e}")

```

Figura A8: Código de Python para utilizado mitigar el espacio en disco limitado, parte 2

```
var_tmp_dir = '/var/tmp'
for root, dirs, files in os.walk(var_tmp_dir):
    for name in files:
        file_path = os.path.join(root, name)
        try:
            os.remove(file_path)
            print(f"Archivo eliminado: {file_path}")
        except Exception as e:
            print(f"No se pudo eliminar el archivo {file_path}: {e}")

"""Limpia archivos de caché en /var/cache y logs en /var/log."""
cache_dir = '/var/cache'
for root, dirs, files in os.walk(cache_dir):
    for name in files:
        file_path = os.path.join(root, name)
        try:
            os.remove(file_path)
            print(f"Archivo eliminado: {file_path}")
        except Exception as e:
            print(f"No se pudo eliminar el archivo {file_path}: {e}")

log_dir = '/var/log'
for root, dirs, files in os.walk(log_dir):
    for name in files:
        file_path = os.path.join(root, name)
        try:
            if file_path.endswith('.log'):
                os.remove(file_path)
                print(f"Archivo de log eliminado: {file_path}")
        except Exception as e:
            print(f"No se pudo eliminar el archivo de log {file_path}: {e}")

"""Limpia la caché de APT."""
try:
    subprocess.run(['apt-get', 'clean'], check=True)
    print("Caché de APT limpiada")
except subprocess.CalledProcessError as e:
    print(f"Error al limpiar caché de APT: {e}")
```

Figura A9: Archivo bash para la activación del entorno virtual y el código en Python del espacio en disco

```
DIR: /usr/local/bin/zabbix-scripts/control_disco.sh

#!/bin/bash
source /home/jair/zabbix-env/bin/activate
/home/jair/zabbix-env/bin/python /home/jair/scripts_control/disco.py
```

DECLARACIÓN Y AUTORIZACIÓN

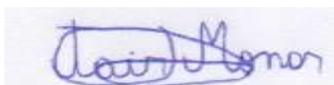
Yo, **Monar Idrovo Jair Cristopher**, con C.C: **0932122823** autor/a del trabajo de titulación: **Diseño y análisis de un control de congestión para servidores en un entorno virtualizado utilizando Zabbix y Python como herramientas Open-Source** previo a la obtención del título de **INGENIERO EN TELECOMUNICACIONES** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, **02 de septiembre** de **2024**

f. _____



Monar Idrovo Jair Cristopher

C.C: 0932122823



REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA

FICHA DE REGISTRO DE TESIS/TRABAJO DE TITULACIÓN

TÍTULO Y SUBTÍTULO:	Diseño y análisis de un control de congestión para servidores en un entorno virtualizado utilizando Zabbix y Python como herramientas Open-Source.		
AUTOR(ES)	Jair Cristopher Monar Idrovo		
REVISOR(ES)/TUTOR(ES)	Ing. Medina Moreira Washington Adolfo, PhD. D		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería en Telecomunicaciones		
TÍTULO OBTENIDO:	Ingeniero en Telecomunicaciones		
FECHA DE PUBLICACIÓN:	02 de Septiembre de 2024	No. PÁGINAS:	DE 67 p.
ÁREAS TEMÁTICAS:	Servidores, monitoreo, Programación		
PALABRAS CLAVES/ KEYWORDS:	Zabbix, Diseño, Servidor Ubuntu, Códigos, Python, Servidores virtualizados, Automatización.		
RESUMEN:	<p>Este trabajo de integración curricular propone la creación de un diseño de un control de congestión para servidores mediante el uso de herramientas open-source para poder monitorear constantemente los componentes del sistema y tomar medidas cuando estos no estén en los rangos necesarios para un buen rendimiento del servidor. El diseño está enfocado en la creación de scripts mediante código en Python para poder gestionar los recursos de un servidor virtualizado de una forma automatizada mediante la integración con triggers de Zabbix los cuales nos permiten realizar acciones cuando un parámetro del sistema no se encuentre en los rangos establecidos, de esta manera podremos tomar las medidas necesarias accionando comandos que ejecuten tareas automatizadas que permitan gestionar los procesos que están causando una sobrecarga en el sistema. El enfoque se centra en destacar las diferentes funcionalidades que posee la herramienta de Zabbix, así como la configuración recomendada para sacar provecho del programa que junto con la integración con scripts de Python pueda ser una solución que brinde una ayuda para evitar problemas a futuro.</p>		
ADJUNTO PDF:	<input checked="" type="checkbox"/> SI	<input type="checkbox"/> NO	
CONTACTO CON AUTOR:	Teléfono: +593-997402928	E-mail: jairmonar@gmail.com	
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO UTE	Nombre: Ubilla González, Ricardo Xavier		
	Teléfono: +593-999528515		
	E-mail: ricardo.ubilla@cu.ucsg.edu.ec		
SECCIÓN PARA USO DE BIBLIOTECA			
Nº. DE REGISTRO (en base a datos):			
Nº. DE CLASIFICACIÓN:			
DIRECCIÓN URL (tesis en la web):			