



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**CARRERA DE TELECOMUNICACIONES**

**INFORME FINAL DEL TRABAJO DE INTEGRACIÓN  
CURRICULAR, PROPUESTAS TECNOLÓGICAS**

**TEMA:**

**“IMPLEMENTACIÓN Y AUTOMATIZACIÓN DE  
INFRAESTRUCTURA BASADA EN CÓDIGO PARA LOS  
LABORATORIOS DEL ÁREA DE REDES DE LA CARRERA DE  
TELECOMUNICACIONES EN LA UNIVERSIDAD TÉCNICA DEL  
NORTE”**

**Trabajo de titulación previo a la obtención del título de Ingeniero en Telecomunicaciones**

**Línea de investigación:** Innovación, Tecnología y de productos

**AUTOR(A):**

Espinoza Guevara Lizbeth Pamela

**DIRECTOR(A):**

Ing. Cuzme Rodríguez Fabián Geovanny, MSc

**Ibarra, 2025**

**UNIVERSIDAD TÉCNICA DEL NORTE  
BIBLIOTECA UNIVERSITARIA**

**IDENTIFICACIÓN DE LA OBRA**

La Universidad Técnica del Norte dentro del proyecto Repositorio Digital Institucional, determinó la necesidad de disponer de textos completos en formato digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la Universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual pongo a disposición la siguiente información:

DATOS DE CONTACTO			
<b>CÉDULA DE IDENTIDAD:</b>	0401861778		
<b>APELLIDOS Y NOMBRES:</b>	Espinoza Guevara Lizbeth Pamela		
<b>DIRECCIÓN:</b>	8 de Diciembre e Ismael Huera		
<b>EMAIL:</b>	<a href="mailto:lpespinzag@utn.edu.ec">lpespinzag@utn.edu.ec</a>		
<b>TELÉFONO FIJO:</b>	062973613	<b>TELF. MOVIL</b>	0994665413

DATOS DE LA OBRA	
<b>TÍTULO:</b>	Implementación y automatización de infraestructura basada en código para los laboratorios del área de redes de la carrera de Telecomunicaciones en la Universidad Técnica del Norte.
<b>AUTOR (ES):</b>	Espinoza Guevara Lizbeth Pamela
<b>FECHA: AAAAMMDD</b>	2025-01-21
SOLO PARA TRABAJOS DE INTEGRACIÓN CURRICULAR	
<b>CARRERA/PROGRAMA:</b>	<input checked="" type="checkbox"/> <b>GRADO</b> <input type="checkbox"/> <b>POSGRADO</b>
<b>TÍTULO POR EL QUE OPTA:</b>	Ingeniero en Telecomunicaciones
<b>DIRECTOR:</b>	Ing. Fabián Geovanny Cuzme Rodriguez, MsC
<b>ASESOR:</b>	Ing. Carlos Alberto Vásquez Ayala, MsC

## AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, Lizbeth Pamela Espinoza Guevara, con cédula de identidad Nro. 0401861778, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de integración curricular descrito anteriormente, hago entrega del ejemplar respectivo en formato digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y uso del archivo digital en la Biblioteca de la Universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión; en concordancia con la Ley de Educación Superior Artículo 144.

Ibarra, a los 21 días del mes de enero de 2025

### EL AUTOR:

Firma.....

Nombre: Espinoza Guevara Lizbeth Pamela

## CONSTANCIAS

El autor manifiesta que la obra objeto de la presente autorización es original y se la desarrolló, sin violar derechos de autor de terceros, por lo tanto, la obra es original y que es el titular de los derechos patrimoniales, por lo que asume la responsabilidad sobre el contenido de la misma y saldrá en defensa de la Universidad en caso de reclamación por parte de terceros.

Ibarra, a los 21 días, del mes de enero de 2025

### EL AUTOR:

Firma.....

Nombre: Espinoza Guevara Lizbeth Pamela

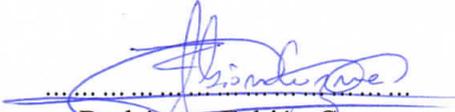
## **CERTIFICACIÓN DEL DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR**

Ibarra, 21 de enero de 2025

ING. CUZME RODRÍGUEZ FABIÁN GEOVANNY. MSC  
DIRECTOR DEL TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICA:

Haber revisado el presente informe final del trabajo de Integración Curricular, el mismo que se ajusta a las normas vigentes de la Universidad Técnica del Norte; en consecuencia, autorizo su presentación para los fines legales pertinentes.

  
Ing. Cuzme Rodríguez Fabián Geovanny. MsC  
C.C.: 1311527012

## APROBACIÓN DEL COMITÉ CALIFICADOR

El Comité Calificado del trabajo de Integración Curricular “IMPLEMENTACIÓN Y AUTOMATIZACIÓN DE INFRAESTRUCTURA BASADA EN CÓDIGO PARA LOS LABORATORIOS DEL ÁREA DE REDES DE LA CARRERA DE TELECOMUNICACIONES EN LA UNIVERSIDAD TÉCNICA DEL NORTE” elaborado por ESPINOZA GUEVARA LIZBETH PAMELA, previo a la obtención del título del INGENIERA EN TELECOMUNICACIONES, aprueba el presente informe de investigación en nombre de la Universidad Técnica del Norte:



.....  
Ing. Cuzme Rodríguez Fabián Geovanny, MsC  
C.C.: 1311527012



.....  
Ing. Carlos Alberto Vásquez Ayala, MsC  
C.C.: 1002424982

## DEDICATORIA

*Quiero dedicar este trabajo de titulación a mi madre, quien, con su amor incondicional y sacrificio incansable, ha sido la base de todos mis logros. Su fortaleza, dedicación y apoyo constante me han dado las herramientas necesarias para alcanzar mis metas. Por ser mi mayor inspiración y estar siempre a mi lado en cada momento, le estaré eternamente agradecida.*

*A mi padre, por su apoyo, sus consejos y por ser un pilar de fortaleza en mi vida. Su confianza en mis capacidades me ha motivado a enfrentar los desafíos con valentía, perseverancia y determinación, recordándome que con esfuerzo y dedicación todo es posible.*

## AGRADECIMIENTO

*En primer lugar, agradezco a Dios por ser mi guía constante, por brindarme la fortaleza y la sabiduría necesarias para superar cada desafío.*

*A mi madre, quien, ha sido la fuerza que me ha impulsado a avanzar y a nunca rendirme. Su ejemplo de perseverancia y dedicación me ha inspirado profundamente a lo largo de este proceso.*

*A mi padre, por su apoyo, su confianza en mí y su presencia en cada etapa de este camino.*

*A mi tía Marthi, por su valioso apoyo en esta etapa de mi vida. Sus palabras de aliento y su confianza en mí han sido un gran impulso para alcanzar este logro.*

*A cada uno de mis tíos, por su apoyo incondicional, sus sabios consejos y por estar siempre dispuestos a brindarme su ayuda cuando más la he necesitado. Su cariño y respaldo han sido un motor importante en este recorrido.*

*A mi tutor de trabajo de titulación, MSc. Fabián Cuzme, y a mi asesor de trabajo de titulación, MSc. Carlos Vásquez, a quienes agradezco profundamente por haber sido una guía fundamental durante todo este trayecto. Su paciencia, dedicación y disposición para compartir su conocimiento y experiencia han sido clave para el desarrollo y culminación de este proyecto.*

*A mis amigos, gracias por su apoyo constante, por estar a mi lado en los momentos de alegría y en los desafíos, y por compartir conmigo esta bonita etapa de la vida. Su compañía y palabras de aliento han hecho de este camino una experiencia inolvidable.*

## RESUMEN EJECUTIVO

Este proyecto se enfoca en la automatización del despliegue de infraestructura basada en código para los laboratorios del área de redes en la carrera de Telecomunicaciones de la Universidad Técnica del Norte. La propuesta busca hacer más eficiente y repetitiva el despliegue de máquinas virtuales de aplicaciones en el entorno de virtualización Proxmox, teniendo en cuenta las características de hardware y software del clúster de la universidad. La solución se basa en la necesidad de mejorar la eficiencia y estandarización de los procesos de despliegue y gestión de infraestructuras virtualizadas mediante la implementación de Infraestructura como Código usando Terraform, Ansible y Proxmox. El objetivo de esta iniciativa es convertir estos procedimientos en esfuerzos automatizados, más eficientes y menos exigentes. El enfoque empleado incluye un inventario preciso de la infraestructura actual para establecer sus requisitos y luego construir scripts y plantillas que ayuden a automatizar tareas clave. Como resultado, se logró una reducción del 65% en las tasas de errores de configuración y la rápida minimización de los tiempos de aprovisionamiento para hasta el 74% de los laboratorios que se configuraron en menos de 2 horas. Además, el 85% de los estudiantes encuestados recomendaría el enfoque a otros cursos. En conclusión, el uso de IaC no solo mejora la efectividad operativa y la calidad de la práctica académica, sino que también cambia la forma en que se adquiere y absorbe el conocimiento de una manera más dinámica y sencilla. Este enfoque enfatiza el concepto de que la innovación tecnológica puede ayudar a obtener nuevas formas de comprensión mientras también maximiza el uso del tiempo y los recursos que se utilizan dentro de los laboratorios.

**Palabras clave:** Infraestructura como Código, Automatización, Terraform, Ansible, Proxmox.

## ABSTRACT

This project focuses on the automation of the deployment of code-based infrastructure for the laboratories of the network area in the Telecommunications department of the Universidad Técnica del Norte. The proposal seeks to make more efficient and repetitive the deployment of application virtual machines in the Proxmox virtualization environment, taking into account the hardware and software characteristics of the university cluster. The solution is based on the need to improve the efficiency and standardization of the deployment and management processes of virtualized infrastructures through the implementation of Infrastructure as Code using Terraform, Ansible and Proxmox. The goal of this initiative is to convert these procedures into automated, more efficient and less demanding efforts. The approach employed includes an accurate inventory of the current infrastructure to establish its requirements and then build scripts and templates to help automate key tasks. As a result, a 65% reduction in configuration error rates and rapid minimization of provisioning times were achieved for up to 74% of labs that were configured in less than 2 hours. In addition, 85% of students surveyed would recommend the approach to other courses. In conclusion, the use of IaC not only improves the operational effectiveness and quality of academic practice, but also changes the way knowledge is acquired and absorbed in a more dynamic and user-friendly way. This approach emphasizes the concept that technological innovation can help gain new forms of understanding while also maximizing the use of time and resources used within laboratories.

**Keywords:** Infrastructure as Code, Automation, Terraform, Ansible, Proxmox.

## **LISTA DE SIGLAS**

**IaC** Infraestructura como código

**LOES** Ley Orgánica de Educación Superior

**ODS** Objetivos de Desarrollo Sostenible

**TI** Tecnología de la Información

**CI/CD** Integración continua/Despliegue continuo

**PC** Computadora personal

**VM** Máquinas virtuales

**API** Interfaz de Programación de Aplicaciones

**LAN** Redes de Área Local

**WAN** Redes de Área Extensa

**UPS** Uninterruptable Power Supply

**HCL** HashiCorp Configuration Language

## ÍNDICE DE CONTENIDOS

### CAPÍTULO I:

<b>CAPÍTULO I: INTRODUCCIÓN</b> .....	19
<b>1.1 Problema de Investigación</b> .....	19
<b>1.2 Justificación</b> .....	20
<b>1.3 Objetivos</b> .....	22
<b>1.3.1 Objetivo General</b> .....	22
<b>1.3.2 Objetivos Específicos</b> .....	22
<b>1.4 Alcance</b> .....	23
<b>CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA</b> .....	27
<b>2.1 Infraestructura como código</b> .....	27
<b>2.1.1 Evolución de la IaC</b> .....	29
<b>2.1.2 Ventajas de implementar IaC</b> .....	30
<b>2.2 Virtualización</b> .....	31
<b>2.2.1 Beneficios de la virtualización</b> .....	32
<b>2.2.2 Herramientas de virtualización</b> .....	33
<b>2.2.3 Clúster de virtualización</b> .....	34
<b>2.2.4 Proxmox</b> .....	35
<b>2.2.5 Proxmox en entornos de aprendizaje</b> .....	35
<b>2.2.6 Automatización en Proxmox</b> .....	37
<b>2.3 Herramientas de Automatización de Infraestructura</b> .....	37

	13
2.3.1 <i>Beneficios de las herramientas de IaC</i> .....	38
2.3.2 <i>Herramientas de IaC</i> .....	38
2.3.3 <i>Herramientas IaC compatibles con Proxmox</i> .....	40
2.4 <b>Desafíos y oportunidades de la automatización de la infraestructura como código</b> .....	42
2.4.1 <i>Oportunidades para mejorar la enseñanza con la IaC en entornos de virtualización</i> .....	43
2.4.2 <i>Casos de éxito en la implementación de virtualización e IaC</i> .....	44
<b>CAPÍTULO III: CASO DE ESTUDIO</b> .....	46
3.1 <b>Análisis actual del clúster de virtualización</b> .....	46
3.1.1 <i>Características de hardware</i> .....	47
3.1.2 <i>Características de software</i> .....	48
3.2 <b>Análisis de asignaturas de redes</b> .....	48
3.2.1 <i>Redes de Área Local</i> .....	48
3.2.2 <i>Redes de Área Extensa</i> .....	49
3.2.3 <i>Sistemas de Comunicación Multimedia</i> .....	49
3.2.4 <i>Seguridad en Redes</i> .....	50
3.2.5 <i>Ingeniería en Tráfico</i> .....	50
3.2.6 <i>Administración de Redes de Telecomunicaciones</i> .....	51
3.3 <b>Análisis de prácticas de laboratorio de las asignaturas</b> .....	51
3.3.1 <i>Análisis de prácticas de laboratorio en Redes de Área Local</i> .....	52
3.3.2 <i>Análisis de prácticas de laboratorio en Redes de Área Extensa</i> .....	53

3.3.3 <i>Análisis de prácticas de laboratorio en Sistemas de Comunicación</i>	
<i>Multimedia</i> .....	54
3.3.4 <i>Análisis de prácticas de laboratorio en Seguridad en Redes</i> .....	55
3.3.5 <i>Análisis de prácticas de laboratorio en Ingeniería en Tráfico</i> .....	57
3.3.6 <i>Análisis de prácticas de laboratorio en Administración de Redes de</i>	
<i>Telecomunicaciones</i> .....	58
3.4 <i>Selección de asignaturas con mayor consumo de recursos</i> .....	60
<b>CAPÍTULO IV: IMPLEMENTACIÓN DE SCRIPTS Y PLANTILLAS DE</b>	
<b>INFRAESTRUCTURA COMO CÓDIGO</b> .....	63
<b>4.1 Integración de Terraform y Ansible</b> .....	63
4.1.1 <i>Instalación de Terraform</i> .....	64
4.1.2 <i>Instalación de Ansible</i> .....	67
<b>4.2 Desarrollo de Scripts y Plantillas</b> .....	69
4.2.1 <i>Creación de scripts con Terraform para Proxmox</i> .....	70
4.2.2 <i>Desarrollo de plantillas de configuración con Ansible</i> .....	75
<b>4.3 Pruebas de Automatización</b> .....	77
4.3.1 <i>Proceso de validación de scripts</i> .....	78
4.3.2 <i>Pruebas de integración continua para garantizar la calidad</i> .....	80
<b>4.4 Escenarios de Laboratorios Automatizados</b> .....	93
4.4.1 <i>Adaptación de guías de laboratorios</i> .....	94
4.4.2 <i>Implementación y despliegue de los laboratorios predefinidos</i> ...	99
4.4.3 <i>Resultados y análisis de la implementación</i> .....	105

<b>Conclusiones y recomendaciones</b> .....	111
<b>Referencias Bibliográficas</b> .....	114
<b>ANEXOS</b> .....	116
<b>Anexo A</b> .....	116
<b>Anexo B</b> .....	119
<b>Anexo C</b> .....	122
<b>Anexo D</b> .....	125
<b>Anexo E</b> .....	129
<b>Anexo F</b> .....	134
<b>Anexo G</b> .....	139
<b>Anexo H</b> .....	148
<b>Anexo I</b> .....	151
<b>Anexo J</b> .....	160

## ÍNDICE DE TABLAS

<b>TABLA 1</b> TABLA COMPARATIVA DE HERRAMIENTAS DE VIRTUALIZACIÓN .....	33
<b>TABLA 2</b> TABLA COMPARATIVA DE LAS HERRAMIENTAS DE IAC.....	39
<b>TABLA 3</b> CARACTERÍSTICAS DE HARDWARE .....	47
<b>TABLA 4</b> PROMEDIO DE UTILIZACIÓN DE LOS COMPONENTES DE INFRAESTRUCTURA DE VIRTUALIZACIÓN .....	48
<b>TABLA 5</b> GUÍAS DE PRÁCTICAS DE LABORATORIO PARA LA ASIGNATURA DE REDES DE ÁREA LOCAL .....	52
<b>TABLA 6</b> GUÍAS DE PRÁCTICAS DE LABORATORIO PARA LA ASIGNATURA DE REDES DE ÁREA EXTENSA.....	54
<b>TABLA 7</b> GUÍAS DE PRÁCTICAS DE LABORATORIO PARA LA ASIGNATURA DE SISTEMAS DE COMUNICACIÓN MULTIMEDIA.....	55
<b>TABLA 8</b> GUÍAS DE PRÁCTICAS DE LABORATORIO PARA LA ASIGNATURA DE SEGURIDAD EN REDES .....	56
<b>TABLA 9</b> GUÍAS DE PRÁCTICAS DE LABORATORIO PARA LA ASIGNATURA DE INGENIERÍA DE TRÁFICO.....	57
<b>TABLA 10</b> GUÍAS DE PRÁCTICAS DE LABORATORIO PARA LA ASIGNATURA DE ADMINISTRACIÓN DE REDES DE TELECOMUNICACIONES .....	59
<b>TABLA 11</b> RECURSOS UTILIZADOS PARA LAS PRÁCTICAS DE LABORATORIO EN LA ASIGNATURA DE INGENIERÍA DE TRÁFICO ....	61
<b>TABLA 12</b> RECURSOS UTILIZADOS PARA LAS PRÁCTICAS DE LABORATORIO EN LA ASIGNATURA DE ADMINISTRACIÓN DE REDES DE TELECOMUNICACIONES .....	62
<b>TABLA 13</b> TABLA DE PRUEBAS DE INTEGRACIÓN CONTINUA DE INFRAESTRUCTURA COMO CÓDIGO .....	81
<b>TABLA 14</b> ESPECIFICACIONES DE LA PRUEBA .....	104
<b>TABLA 15</b> TIEMPO PROMEDIO DE DESPLIEGUE Y CONFIGURACIÓN DE LA MÁQUINA VIRTUAL .....	106
<b>TABLA 16</b> CONSUMO DE RECURSOS SEGÚN LAS MÁQUINAS VIRTUALES ACTIVAS .....	109
<b>TABLA 17</b> TABULACIÓN ENCUESTA PREGUNTA 1.....	151
<b>TABLA 18</b> TABULACIÓN ENCUESTA PREGUNTA 2.....	152
<b>TABLA 19</b> TABULACIÓN ENCUESTA PREGUNTA 3.....	153
<b>TABLA 20</b> TABULACIÓN ENCUESTA PREGUNTA 4.....	154
<b>TABLA 21</b> TABULACIÓN ENCUESTA PREGUNTA 5.....	154
<b>TABLA 22</b> TABULACIÓN ENCUESTA PREGUNTA 6.....	155
<b>TABLA 23</b> TABULACIÓN ENCUESTA PREGUNTA 7.....	156
<b>TABLA 24</b> TABULACIÓN ENCUESTA PREGUNTA 8.....	157
<b>TABLA 25</b> TABULACIÓN ENCUESTA PREGUNTA 9.....	158
<b>TABLA 26</b> TABULACIÓN ENCUESTA PREGUNTA 10.....	159
<b>TABLA 27</b> ESPECIFICACIONES DE LA PRUEBA.....	162

## ÍNDICE DE FIGURAS

<b>FIGURA 1</b> ARQUITECTURA DE AUTOMATIZACIÓN DE INFRAESTRUCTURA COMO CÓDIGO BASADO EN IBM (IBM, 2023).....	26
<b>FIGURA 2</b> INFRAESTRUCTURA COMO CÓDIGO.....	28
<b>FIGURA 3</b> ACTUALIZACIÓN DEL SISTEMA.....	65
<b>FIGURA 4</b> INSTALACIÓN DE DEPENDENCIAS.....	65
<b>FIGURA 5</b> AÑADIR LA CLAVE GPG DE HASHICORP.....	66
<b>FIGURA 6</b> AGREGACIÓN DEL REPOSITORIO DE HASHICORP A LA LISTA DE FUENTES DE APT.....	66
<b>FIGURA 7</b> INSTALACIÓN DE TERRAFORM.....	67
<b>FIGURA 8</b> VERIFICACIÓN DE INSTALACIÓN.....	67
<b>FIGURA 9</b> INSTALACIÓN DE DEPENDENCIAS.....	68
<b>FIGURA 10</b> AGREGACIÓN DEL REPOSITORIO DE ANSIBLE.....	68
<b>FIGURA 11</b> INSTALACIÓN DE ANSIBLE.....	69
<b>FIGURA 12</b> VERIFICACIÓN DE INSTALACIÓN.....	69
<b>FIGURA 13</b> DEFINICIÓN DE VARIABLES DE PROXMOX.....	71
<b>FIGURA 14</b> DEFINICIÓN DE TOKEN API DE PROXMOX.....	71
<b>FIGURA 15</b> IDENTIFICACIÓN DE LA MV QUE SE CREARA.....	72
<b>FIGURA 16</b> TAMAÑO DEL SISTEMA DE ARCHIVOS RAÍZ.....	72
<b>FIGURA 17</b> DEFINICIÓN DE DIRECCIONES IP PARA MV.....	72
<b>FIGURA 18</b> DEFINICIÓN DE CLAVE SSH.....	73
<b>FIGURA 19</b> CONFIGURACIÓN DEL PROVEEDOR PROXMOX.....	73
<b>FIGURA 20</b> CREACIÓN DE RECURSOS DE LAS MÁQUINAS VIRTUALES.....	74
<b>FIGURA 21</b> CONFIGURACIÓN DE MÁQUINA VIRTUAL.....	74
<b>FIGURA 22</b> PROCESO DE PROVISIONAMIENTO.....	75
<b>FIGURA 23</b> CONFIGURACIÓN DE INSTALACIÓN QEMU GUEST AGENT.....	76
<b>FIGURA 24</b> CONFIGURACIÓN DE MÁQUINA VIRTUAL.....	76
<b>FIGURA 25</b> ACTUALIZACIÓN DEL SISTEMA OPERATIVO DE LA MÁQUINA VIRTUAL.....	77
<b>FIGURA 26</b> DIRECTORIO DE SCRIPTS DE INFRAESTRUCTURA COMO CÓDIGO.....	78
<b>FIGURA 27</b> INICIALIZACIÓN DE TERRAFORM.....	79
<b>FIGURA 28</b> VERIFICACIÓN DE LA CONFIGURACIÓN DE TERRAFORM.....	79
<b>FIGURA 29</b> PLAN DE EJECUCIÓN EN TERRAFORM.....	80
<b>FIGURA 30</b> EJECUCIÓN DE TERRAFORM APPLY.....	83
<b>FIGURA 31</b> EJECUCIÓN DEL PLAN DE CREACIÓN DE LA MÁQUINA VIRTUAL.....	83
<b>FIGURA 32</b> CLONACIÓN DE MV.....	84
<b>FIGURA 33</b> REGISTRO DE TAREAS DE AUTOMATIZACIÓN DE INFRAESTRUCTURA.....	84
<b>FIGURA 34</b> EJECUCIÓN DE TERRAFORM APPLY TEST 2.....	85
<b>FIGURA 35</b> PROCESO DE DESTRUCCIÓN Y CREACIÓN DE MÁQUINA VIRTUAL CON TERRAFORM.....	86
<b>FIGURA 36</b> REGISTRO DE ACCIONES DE GESTIÓN DE MÁQUINAS VIRTUALES EN PROXMOX.....	86
<b>FIGURA 37</b> PROCESO DE CREACIÓN Y DESTRUCCIÓN DE MÁQUINAS VIRTUALES EN PROXMOX UTILIZANDO TERRAFORM.....	87
<b>FIGURA 38</b> CREACIÓN DE UN NUEVO WORKSPACE EN TERRAFORM.....	88
<b>FIGURA 39</b> EJECUCIÓN DE TERRAFORM APPLY.....	89
<b>FIGURA 40</b> EJECUCIÓN DEL PLAN DE TERRAFORM PARA AGREGAR UNA MÁQUINA VIRTUAL.....	89
<b>FIGURA 41</b> REGISTRO DE TAREAS DE AUTOMATIZACIÓN DE INFRAESTRUCTURA.....	90
<b>FIGURA 42</b> DECLARACIÓN DE VARIABLES PARA LA CONFIGURACIÓN DE MÁQUINAS VIRTUALES EN TERRAFORM.....	91
<b>FIGURA 43</b> EJECUCIÓN DEL DESPLIEGUE DE MÁQUINAS VIRTUALES CON TERRAFORM.....	91
<b>FIGURA 44</b> PLAN DE EJECUCIÓN DE TERRAFORM PARA EL DESPLIEGUE DE MÁQUINAS VIRTUALES.....	92
<b>FIGURA 45</b> REGISTRO DE ACTIVIDADES DEL DESPLIEGUE DE INFRAESTRUCTURA EN PROXMOX.....	92
<b>FIGURA 46</b> FINALIZACIÓN DEL DESPLIEGUE DE MÁQUINAS VIRTUALES CON TERRAFORM.....	93

<b>FIGURA 47</b> TOPOLOGÍA DE RED MPLS .....	95
<b>FIGURA 48</b> TOPOLOGÍA DE RED IMPLEMENTADA PARA LA MONITORIZACIÓN .....	97
<b>FIGURA 49</b> TOPOLOGÍA DE RED PARA LA MONITORIZACIÓN DE REDES .....	99
<b>FIGURA 50</b> TOPOLOGÍA DE RED DE LA GUÍA DE LABORATORIO 1.....	100
<b>FIGURA 51</b> TOPOLOGÍA DE RED DE LA GUÍA DE LABORATORIO 2.....	101
<b>FIGURA 52</b> TOPOLOGÍA DE RED DE LA GUÍA DE LABORATORIO 3.....	102
<b>FIGURA 53</b> RESUMEN DE RECURSOS EN EL NODO PV5.....	109
<b>FIGURA 54</b> CLAVES SSH .....	126
<b>FIGURA 55</b> VERIFICACIÓN DE LA INSTALACIÓN DEL CLIENTE OPENSASH EN KALI LINUX.....	127
<b>FIGURA 56</b> CONSOLA DE MV EN PROXMOX .....	131
<b>FIGURA 57</b> CONFIGURACIÓN DE IDIOMA Y TECLADO .....	131
<b>FIGURA 58</b> PARTICIÓN DEL DISCO .....	132
<b>FIGURA 59</b> CREACIÓN DE USUARIO .....	132
<b>FIGURA 60</b> INSTALACIÓN .....	133
<b>FIGURA 61</b> MÁQUINA VIRTUAL .....	133
<b>FIGURA 62</b> CÓDIGO DE VARIABLES.....	135
<b>FIGURA 63</b> CÓDIGO PRINCIPAL.....	136
<b>FIGURA 64</b> INICIALIZACIÓN DE TERRAFORM.....	136
<b>FIGURA 65</b> DESPLIEGUE DE INFRAESTRUCTURA.....	137
<b>FIGURA 66</b> INVENTARIO .....	137
<b>FIGURA 67</b> PLAYBOOK PARA ACTUALIZAR MV.....	138
<b>FIGURA 68</b> EJECUCIÓN DE PLAYBOOK .....	138
<b>FIGURA 69</b> GRÁFICO PREGUNTA 1 .....	151
<b>FIGURA 70</b> GRÁFICO PREGUNTA 2.....	152
<b>FIGURA 71</b> GRÁFICO PREGUNTA 3 .....	153
<b>FIGURA 72</b> GRÁFICO PREGUNTA 4.....	154
<b>FIGURA 73</b> GRÁFICO PREGUNTA 5.....	155
<b>FIGURA 74</b> GRÁFICO PREGUNTA 6.....	156
<b>FIGURA 75</b> GRÁFICO PREGUNTA 7 .....	157
<b>FIGURA 76</b> GRÁFICO PREGUNTA 8.....	158
<b>FIGURA 77</b> GRÁFICO PREGUNTA 9.....	158
<b>FIGURA 78</b> GRÁFICO PREGUNTA 10.....	159
<b>FIGURA 79</b> EXPOSICIÓN DE INFRAESTRUCTURA COMO CÓDIGO.....	163
<b>FIGURA 80</b> INTRODUCCIÓN DE LA IMPLEMENTACIÓN Y DESPLIEGUE DE IAC PARA LOS LABORATORIOS DEL ÁREA DE REDES ..	164
<b>FIGURA 81</b> DESARROLLO DE LA PRUEBA DE LOS LABORATORIOS PREDEFINIDOS .....	164

## CAPÍTULO I: INTRODUCCIÓN

El capítulo se presenta como la fase inicial y necesaria para abordar el problema y los objetivos de investigación asociados con la automatización de la infraestructura de los laboratorios en el área de redes en la Universidad Técnica del Norte. La estructura de esta sección ha sido concebida con muchas subsecciones que buscan proporcionar una comprensión completa de la complejidad que es inherente a la gestión moderna de las infraestructuras dentro del campo de las telecomunicaciones. Al mismo tiempo, hace hincapié en la urgente necesidad de adoptar enfoques innovadores, especialmente infraestructura como código (IaC).

### 1.1 Problema de Investigación

La creación y despliegue de máquinas virtuales y contenedores a través de una interfaz gráfica presenta desafíos significativos, especialmente cuando se trata de implementar un gran número de instancias. Navegar a través de múltiples ventanas y configurar manualmente cada instancia no solo consume una cantidad sustancial de tiempo, sino que también aumenta la probabilidad de cometer errores en la configuración; además, en un entorno en constante crecimiento y cambio, la gestión dispersa de la infraestructura se convierte en un desafío difícil de abordar (Unisys, 2018).

Los laboratorios de redes juegan un papel esencial en la formación de futuros profesionales en el campo de las telecomunicaciones. La gestión y configuración eficientes de la infraestructura en estos laboratorios son cruciales para garantizar una experiencia de aprendizaje de alta calidad. De la misma manera la gestión de la infraestructura en estos se vuelve dispersa a medida que crece la cantidad de recursos y

escenarios de laboratorio, por lo que deben estar preparados para satisfacer las necesidades cambiantes del entorno.

La Universidad Técnica del Norte, en la carrera de Telecomunicaciones, cuenta con una infraestructura interna virtualizada basada en Proxmox versión 7.4.3 que consta de 3 nodos donde se encuentran alojados diferentes servicios. Esta plataforma permite la creación de máquinas virtuales y contenedores a través de una interfaz gráfica, esta metodología resulta poco eficiente y lenta, especialmente cuando se requiere el despliegue de entornos completos de ambientes de laboratorios en el área de redes.

Gestionar cada una de las máquinas virtuales alojadas en el sistema individualmente puede ser una tarea tediosa, en la actualidad hay diferentes herramientas disponibles que permiten la gestión y administración del entorno de forma automática. Algunas de las ventajas que proporciona el uso de este tipo de aplicaciones son mayor fiabilidad, reducción de costos, aumento de la productividad y reducción de errores (Errazquin, 2023).

Por lo tanto, automatizar la infraestructura basada en código permite la creación y el despliegue rápido de escenarios de laboratorio predefinidos, reduciendo el tiempo dedicado a configuraciones manuales. En un entorno educativo en constante cambio, la automatización se convierte en una herramienta imprescindible para mantenerse al día con las demandas tecnológicas y asegurar que los estudiantes adquieran las habilidades necesarias para enfrentar los desafíos en el campo de las telecomunicaciones. Así es como la automatización se ha convertido en un elemento clave para optimizar la gestión de recursos en entornos de virtualización en evolución (Sánchez; Romero, 2021a).

## **1.2 Justificación**

En la última década, el ámbito digital ha crecido mucho. En la actualidad, casi todos los procedimientos operativos de las empresas han sido transformados en digitales

y se gestionan y controlan por medio de herramientas tecnológicas. La infraestructura como código o también conocida como IaC que implica la capacidad de un centro de datos y su administración a través del código definido por scripts. Esto permite automatizar las tareas de aprovisionamiento, lo que posibilita crear entornos seguros y escalables en cuestión de minutos (Sánchez; Romero, 2021a).

La provisión y administración de todos los elementos de la infraestructura, desde el hardware hasta los recursos virtuales, plataformas, sistemas de contenedores, servicios y topologías. Esto se logra mediante definiciones declarativas o scripts, en contraposición a la configuración manual o el uso de herramientas tradicionales para configurarlos. La IaC separa las configuraciones, políticas, perfiles, scripts y plantillas para los componentes hardware o software que se implementen en ellos para facilitar su almacenamiento compartido, revisión y aplicación, similar al manejo del código fuente.

Este enfoque, que ha ganado prominencia con la expansión de las infraestructuras en la nube, surge de la filosofía DevOps y aplica a la orquestación de infraestructura el mismo nivel de control de versiones y repetibilidad que los desarrolladores utilizan en el código fuente de aplicaciones. El enfoque IaC apoya la integración continua al generar el mismo entorno de infraestructura en cada implementación (F5, 2023a).

La infraestructura como código viene con algunas ventajas, como la agilidad que permite desplegar rápidamente y replicar eficientemente entornos completos. Proporciona sistemas replicables que son fáciles de desechar, reduciendo así las posibilidades de errores humanos y mejorando la tolerancia a fallos. La automatización resultante reduce costos al liberar personal para actividades estratégicas y prevenir pérdidas debido a interrupciones del servicio, lo que es particularmente relevante en sectores como la microfinanza, donde la continuidad operativa es crítica (Flores, 2023).

Este proyecto desempeña un papel significativo en la formación de profesionales competentes y éticos en el campo de las Telecomunicaciones, con un enfoque específico en redes. La LOES promulgada en 2014 tiene como objetivo principal garantizar una educación superior de calidad a través del acceso universal, la retención, la movilidad y la graduación sin discriminación (LODE & DDR, 2010). Así, la automatización de la infraestructura no solo se alinea con los principios de la LOES, sino que también refuerza la capacitación de profesionales altamente calificados que pueden contribuir al desarrollo sostenible de Ecuador.

Además, el proyecto contribuye directamente a lograr el Objetivo 4 de los Objetivos de Desarrollo Sostenible (ODS) de las Naciones Unidas para la Agenda 2030, que tiene como objetivo garantizar una educación inclusiva, equitativa y de calidad y promover oportunidades de aprendizaje durante toda la vida para todos (Objetivos de Desarrollo Sostenible, 2015). Al proporcionar una infraestructura de laboratorio basada en código, permite a los estudiantes en Telecomunicaciones adquirir competencias especializadas en redes. Esta infraestructura escalable, diseñada modularmente, permite a los estudiantes realizar prácticas de laboratorio que pueden adaptarse a diferentes niveles de complejidad, promoviendo así un aprendizaje integral, continuo y de calidad.

### **1.3 Objetivos**

#### **1.3.1 *Objetivo General***

Automatizar el despliegue de infraestructura basada en código para los laboratorios del área de redes en la carrera de Telecomunicaciones de la Universidad Técnica del Norte.

#### **1.3.2 *Objetivos Específicos***

- Analizar el estado actual del clúster de virtualización de la carrera de Telecomunicaciones y las diferentes herramientas disponibles para la

automatización de infraestructura, cuya solución sea óptima y compatible con la plataforma de virtualización.

- Identificar las necesidades y requisitos específicos de infraestructura en los entornos de laboratorios de redes en la carrera de Telecomunicaciones de la Universidad Técnica del Norte.
- Implementar scripts y plantillas de infraestructura como código que permitan automatizar la configuración, despliegue de máquinas virtuales y contenedores en la plataforma de virtualización Proxmox en los laboratorios en la carrera de Telecomunicaciones de la Universidad Técnica del Norte.

#### **1.4 Alcance.**

El presente proyecto se centra en la automatización del despliegue de infraestructura basada en código para laboratorios en el área de redes en la carrera de Telecomunicaciones de la Universidad Técnica del Norte. Esto implica desplegar máquinas virtuales y contenedores en la plataforma de virtualización Proxmox con su versión 7.4.3, considerando las características de hardware y software del clúster en la carrera de Telecomunicaciones de la Universidad Técnica del Norte.

Para la automatización de infraestructura como código, se utiliza la metodología DevOps que integra el desarrollo de software con la operación del sistema. Esta metodología promueve la colaboración entre los equipos de desarrollo y operaciones, y enfatiza la automatización, la medición y el intercambio de información (Tommaso & Salvarredy, 2015). Estas son las siguientes fases por considerar:

##### ***Fase I: Investigación y análisis***

Para comenzar, se realiza una revisión de fuentes bibliográficas con el fin de recopilar los conceptos básicos de infraestructura como código (IaC). Este proceso

implica buscar y analizar textos académicos, libros y artículos científicos que discuten los principios, definiciones y evolución de IaC. Para entender la esencia de IaC, incluyendo sus beneficios, desafíos y diferentes herramientas y enfoques utilizados en su implementación. Este enfoque metodológico permite establecer un marco teórico que sustenta el desarrollo y la comprensión del papel crucial que desempeña IaC en la gestión moderna de la infraestructura tecnológica (Sánchez; Romero, 2021b).

### ***Fase II: Selección de Herramientas***

La automatización de infraestructura se refiere al uso de sistemas software para crear procedimientos que son repetitivos y reducen la intervención humana en las operaciones de sistemas TI (Red Hat, 2023b). Para lograr esto, se abordará la selección de herramientas como Terraform y Ansible de IaC; estas herramientas permiten definir la infraestructura como código y automatizar su implementación, lo que permite una gestión e implementación de infraestructura eficientes y reproducibles.

### ***Fase III: Caso de estudio***

Posteriormente, se realizará un análisis de las necesidades y requisitos específicos de la infraestructura del laboratorio, con énfasis en asignaturas que ayuden a los estudiantes a adquirir conocimientos sobre redes tales como: Redes de Área Local, Redes de Área Extensa, Sistemas de Comunicación Multimedia, Seguridad en Redes, Ingeniería en Tráfico y Administración de Redes de Telecomunicaciones. Esto implica evaluar los recursos y configuraciones específicos necesarios para optimizar las prácticas educativas en telecomunicaciones. Se prestará especial atención a las materias que requieran un conocimiento previo de redes, asegurando que la infraestructura sea adecuada para el despliegue de escenarios de laboratorio predefinidos.

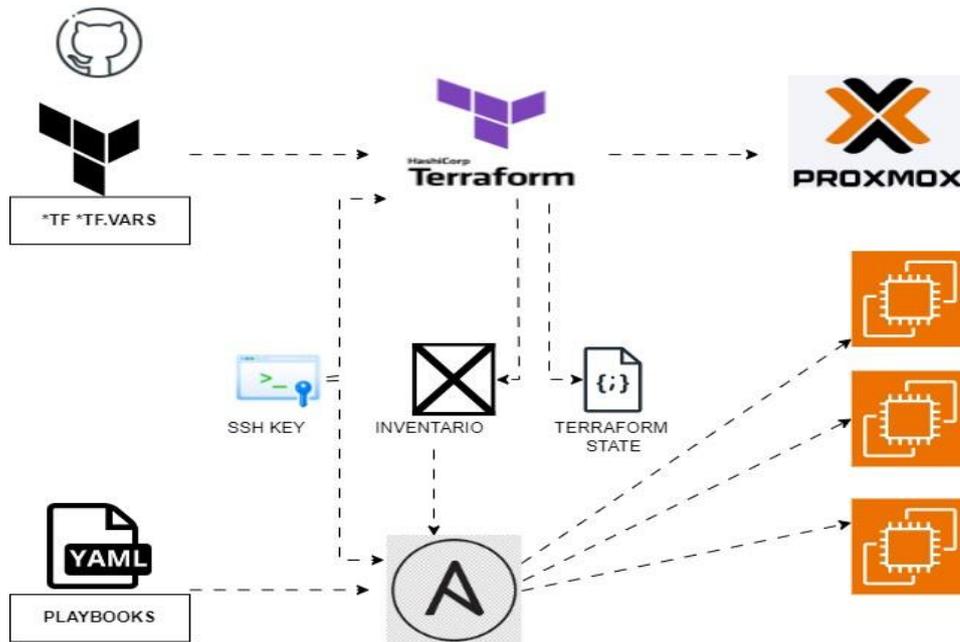
### ***Fase IV: Implementación de scripts y plantillas de infraestructura como código***

En la fase actual, nos embarcamos en el desarrollo de scripts y plantillas para automatizar la infraestructura siguiendo la arquitectura delineada en la Figura 1. En este proceso, se utilizarán dos herramientas complementarias, como son: Terraform para el aprovisionamiento de recursos en la plataforma de virtualización Proxmox y Ansible para la gestión de la configuración. La colaboración de estas herramientas garantiza un enfoque integral, facilitando la automatización de la infraestructura de manera consistente y eficiente según la arquitectura predefinida. Se implementarán pruebas automatizadas para validar la corrección de los scripts y plantillas de infraestructura, así como pruebas de integración continua para asegurar que los cambios introducidos no causen errores. Este enfoque estructurado y centrado en herramientas garantizará consistencia y fiabilidad en el despliegue automatizado de la infraestructura.

Con base en el análisis anterior, se implementará la automatización de infraestructura como código para escenarios de laboratorio predefinidos en el área de redes donde se han adaptado 4 guías de laboratorio junto con herramientas y plantillas para automatizar el despliegue de dichos laboratorios.

**Figura 1**

Arquitectura de automatización de infraestructura como código basado en IBM (IBM, 2023).



*Nota.* La figura muestra cómo se implementará la arquitectura de automatización de infraestructura como código (IaC)

## **CAPÍTULO II: FUNDAMENTACIÓN TEÓRICA**

Este capítulo discute la base teórica a través del análisis bibliográfico que apoyará el tema de investigación, esto nos ayuda a entender conceptos clave relacionados con la infraestructura tecnológica con un enfoque particular en la metodología de Infraestructura como Código (IaC) y sus conexiones con la virtualización.

Debido al crecimiento significativo en el mundo digital en la última década, ha habido una transformación en los procesos operativos donde las herramientas tecnológicas son esenciales para gestionar y administrar tecnologías de información. Esto se debe a que proveer recursos informáticos para alojar aplicaciones ha demostrado ser un desafío debido a numerosas tareas manuales y rutinarias, como la instalación y configuración de servidores. La virtualización ha simplificado este panorama al agilizar estas operaciones, permitiendo así una ejecución más eficiente.

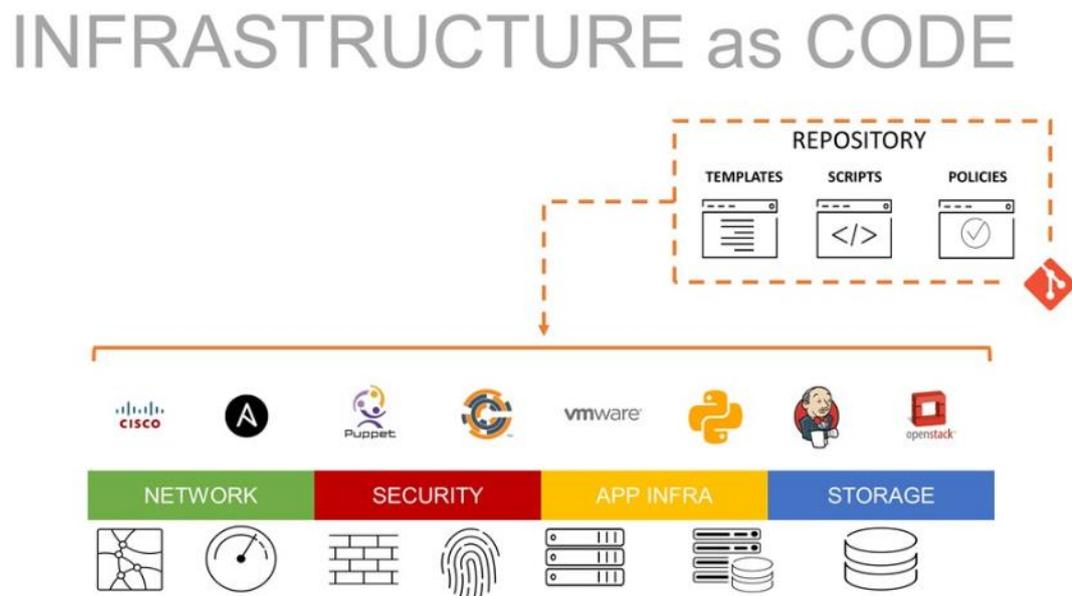
Esto requiere explorar los pros y los contras de IaC, observando sus fundamentos, su integración con la virtualización, el papel que juegan plataformas de virtualización como Proxmox, herramientas impulsadas por la automatización y los desafíos y oportunidades que surgen al implementar esta metodología en entornos específicos, incluyendo entornos educativos.

### **2.1 Infraestructura como código**

Infraestructura como Código (IaC) es una metodología que se ocupa de la gestión y aprovisionamiento de diferentes componentes tecnológicos, tales como hardware, recursos virtuales, plataformas, sistemas de contenedores, servicios y topologías. Este enfoque implica definiciones declarativas o scripts (código), lo que significa que no hay necesidad de configuraciones manuales ni el uso de herramientas convencionales de configuración. En este contexto, las configuraciones, políticas,

perfiles, scripts y plantillas se desvinculan del hardware o software subyacente; esto permite su almacenamiento compartido, revisión y aplicación similar al código fuente *de las aplicaciones (F5, 2023b)*.

**Figura 2**  
*Infraestructura como código.*



*Nota.* El gráfico representa la infraestructura como código, que implica separar los recursos de configuración del hardware o software y permitir que se almacenen, compartan y gestionen de manera similar al código fuente (F5, 2023b).

La Figura 2 muestra un diagrama que detalla la implementación de una metodología IaC, que se utiliza para definir y administrar infraestructuras TI. El código IaC se utiliza para crear, desplegar, actualizar y gestionar diversos componentes de infraestructura TI que van desde redes y servidores hasta almacenamiento y aplicaciones.

La creciente popularidad de esta metodología ha visto un aumento significativo, especialmente en el contexto de la proliferación de infraestructuras en la nube. Además, IaC se alinea estrechamente con la filosofía DevOps al incorporar al aprovisionamiento

de la infraestructura los principios de control de versiones y repetibilidad que son fundamentales en el desarrollo de aplicaciones (Red Hat, 2022)

La IaC respalda prácticas avanzadas como la integración continua, la entrega y la implementación continuas, asegurando la creación coherente del entorno de infraestructura conforme a los principios definidos. Este enfoque no solo simplifica la gestión de configuraciones, sino que también previene cambios no planificados y no documentados, mejorando así la confiabilidad y consistencia del entorno tecnológico.

### ***2.1.1 Evolución de la IaC***

A lo largo del siglo XX y principios del siglo XXI, el desarrollo tecnológico ha sufrido una transformación significativa que ha influido enormemente en nuestra forma de vida, trabajo y relaciones. Desde la introducción de las primeras Mainframes utilizadas por grandes empresas y gobiernos en 1900 hasta la revolución de la inteligencia artificial, blockchain y la tecnología 5G en 2018, esta cronología destaca los hitos clave que han dado forma a tal transformación (Behrend, 2023).

Al mismo tiempo, la evolución de los roles en tecnología y ciberseguridad ha seguido de cerca la progresión de la transformación tecnológica. Inicialmente, se centraba en la administración de sistemas, con Sysadmins responsables del mantenimiento de la infraestructura. Sin embargo, a medida que las aplicaciones se volvieron más complejas y las empresas adoptaron la computación en la nube, surgió la necesidad de una colaboración más estrecha entre desarrolladores y operadores. Esta colaboración impulsó la automatización del ciclo de vida de las aplicaciones a través de tecnologías como Docker, Jenkins y Kubernetes.

Con la migración a la nube, CloudOps surgió como un papel clave en la gestión y optimización de la infraestructura en la nube, utilizando servicios ofrecidos por plataformas como AWS, Azure y Google Cloud. Simultáneamente, la seguridad

adquirió mayor importancia, dando origen a DevSecOps, que integra la seguridad en el desarrollo y operación de software (Behrend, 2023).

En este contexto, la infraestructura como código (IaC) se ha convertido en un componente clave. Esta metodología permite definir y administrar la infraestructura por medio de scripts, lo que facilita una forma eficiente y coherente de generar, configurar y mantener entornos. Por consiguiente, IaC es necesaria para agilidad y flexibilidad en los entornos tecnológicos actuales.

### ***2.1.2 Ventajas de implementar IaC***

La adopción de infraestructura como código tiene ciertos beneficios frente a la forma tradicional de desplegar recursos de tecnología de la información, que es un proceso largo y complicado. El uso de virtualización y desarrollo nativo de la nube resuelve los problemas de la gestión de servidores físicos, permitiendo a los desarrolladores proporcionar servidores virtuales o contenedores según sea necesario.

La Infraestructura como Código es un gran avance en el desarrollo, ya que permite a los programadores desplegar infraestructura completamente descrita y versionada mediante la ejecución de un script. Según (IBM, 2023a), se identifican las siguientes ventajas de la infraestructura como código:

**Tiempo de Producción.** La automatización de IaC acorta de forma drástica el tiempo en el que se puede dotar de infraestructura de TI a nuevos entornos de desarrollo, pruebas o producción e incluso de infraestructura ya existente, eliminando tareas pesadas y posibilitando el aprovisionamiento o desmantelamiento de la infraestructura a voluntad.

**Coherencia Mejorada.** IaC anula el problema de configuración por desvíos al codificar y documentar todo el entorno que ha de ser construido. Eso da lugar a la posibilidad de que no existan problemas en desarrollo, en pruebas o en los despliegues,

que podrían ser problemas posibles, agujeros de cadenas de seguridad y no conformidades.

**Velocidad y Eficiencia.** Al optimizar la provisión mientras se garantiza la consistencia, IaC potencia todas las fases del ciclo de vida de entrega de software. Mejora la provisión rápida de entornos de prueba, permite los procesos de integración continua CI, despliegue continuo CD y pruebas de calidad, lo que hace que el despliegue de código sea más rápido y eficiente cuando pasa las pruebas más fácilmente.

**Seguridad.** IaC asegura que el conocimiento sobre la provisión de infraestructura no salga de la empresa y, por lo tanto, ayuda a reforzar la seguridad de los Sistemas de Información.

**Costos más bajos.** IaC reduce significativamente el tiempo, los esfuerzos y las habilidades avanzadas necesarias para provisionar la infraestructura requerida. Además, permite el gasto de capital con una estructura de pago por uso en la computación en la nube y, al mismo tiempo, quita la carga a los desarrolladores para trabajar en ideas que son fundamentales e innovadoras en el desarrollo de software.

## 2.2 Virtualización

La virtualización emerge como una estrategia crítica para integrar y compartir recursos y es clave en el desarrollo de nuevas tecnologías y en la oferta de servicios novedosos (Fernández Romero & García Pombo, 2011).

Un sistema virtual de software es aquel que es capaz de replicar las características funcionales de algún dispositivo de hardware, particularmente una computadora personal o un servidor. El software de virtualización emula un sistema informático en la medida en que construye un entorno de ejecución que es idéntico en

esencia, aunque no en términos de acceso físico al hardware simulado. Este entorno incluye elementos como CPU, BIOS, placa de video, memoria RAM, placa de red, conexión USB y disco rígido.

La virtualización de software hace posible que múltiples PC o servidores operen simultáneamente con el uso de un solo hardware, mejorando así la utilización de recursos. Aunque el hardware emulado que ejecuta el sistema operativo puede ser más lento debido a la capa de virtualización adicional, sigue funcionando bien para la mayoría de los entornos de producción.

Esta tecnología de virtualización está alterando el panorama de la tecnología de la información (TI) al desacoplar los recursos de hardware, lo que permite al usuario final instanciar varias máquinas virtuales, cada una con una interfaz de hardware compatible con el sistema operativo seleccionado (Toaza, 2011).

### ***2.2.1 Beneficios de la virtualización***

La virtualización es vista como una innovación tecnológica que permite llevar a cabo la consolidación de variedades de sistemas sobrantes en solo un servidor, con lo que se logra reducir de forma considerable el consumo eléctrico, espacio y capacidad de refrigeración. Este sistema también reduce la complejidad del trabajo al reducir la cantidad de servidores y máquinas que se manejan.

En el contexto de virtualización, se solucionan las dificultades asociadas al traslado en vivo. Esta técnica ayuda en la transferencia de un sistema operativo con sus aplicaciones a otro servidor en caso de que la carga se exceda en el hardware disponible.

Esta tecnología también resulta de mayor interés para los programadores. Como el kernel Linux trabaja con un espacio de direcciones único, esto hace que un error global en el kernel o en el controlador provoque el colapso del sistema operativo. La virtualización permite el uso simultáneo de varios sistemas operativos; en caso de que

uno falle, el hipervisor y el resto funcionan. El resultado es la conversión de la resistencia a los errores en muchas instancias en que este es el error probable, a algo muy simple y similar a depurar el kernel a la depuración de aplicaciones en el espacio del usuario (Toaza, 2011).

### 2.2.2 Herramientas de virtualización

La virtualización ha surgido como una tecnología de gran impacto en el ámbito de las Tecnologías de la Información (TI) y ofrece soluciones flexibles y eficientes para la gestión de recursos informáticos. En este sentido, las herramientas de virtualización son componentes clave para construir entornos virtuales que permiten ejecutar múltiples sistemas operativos y aplicaciones en un solo hardware físico.

Estas herramientas hacen posible la creación de máquinas virtuales que representan sistemas operativos y recursos hardware aislados, facilitando así la consolidación del servidor, optimización de recursos y ejecución de aplicaciones en entornos controlados. Además, la virtualización ha demostrado ser esencial en escenarios de desarrollo, pruebas y seguridad al proporcionar espacios aislados para experimentar y evaluar software.

En la Tabla 1 se realizará un análisis detallado sobre diferentes herramientas destacadas para virtualización cada uno con características únicas diseñadas para abordar necesidades específicas.

**Tabla 1**  
*Tabla comparativa de herramientas de virtualización*

	<b>Proxmox VE</b>	<b>VMware vSphere</b>	<b>Windows Hyper-V</b>	<b>Oracle VM VirtualBox</b>	<b>Microsoft Azure Virtual Machines</b>	<b>Amazon Web Services EC2</b>
<b>Sistemas operativos soportados</b>	Windows y Linux (KVM)	Windows, Linux, UNIX	Windows (soporte limitado) e Linux	Windows y Linux	Windows, Linux, macOS,	Windows, Linux, macOS,

			Linux limitado		FreeBSD, Solaris	FreeBSD, Solaris
<b>Open Source</b>	Sí	No	No	Sí	Sí	Sí
<b>Contenedores</b>	Sí	Sí	No	Sí	Sí	Sí
<b>Alta disponibilidad</b>	Sí	Sí	Sí	Sí	Sí	Sí
<b>Copias de seguridad</b>	Sí	Sí	Limitado	Sí	Sí	Sí
<b>Migración</b>	Sí	Sí	Sí	Sí	Sí	Sí
<b>Máxima RAM y CPU por host</b>	160 CPU/2 TB Ram	160 CPU/2 TB Ram	64 CPU / 1TB Ram	64 CPU / 2 TB Ram	256 vCPU / 128 GB de RAM	96 vCPU / 384 GB de RAM
<b>Precio</b>	Gratuito	Desde 1250 USD por año	Desde 1200 USD por año	Gratuito	Desde 0,05 USD por hora	Desde 0,05 USD por hora
<b>Requisitos del sistema</b>	Procesador Intel o AMD de 64 bits					
<b>Soporte técnico</b>	Característica	Gratuito	Gratuito	Gratuito	Gratuito	Gratuito

*Nota.* En la tabla se muestra la comparación de varias plataformas de virtualización y servicios de máquinas virtuales en la nube incluyendo Proxmox VE, VMware vSphere, Windows Hyper-V, Oracle VM VirtualBox, Microsoft Azure Virtual Machines y Amazon Web Services EC2.

### 2.2.3 Clúster de virtualización

Los clústeres se definen como conjuntos de servidores que se gestionan de manera colaborativa, contribuyendo a una gestión eficiente de la carga de trabajo.

Dentro de un clúster, los nodos de aplicación individuales o servidores colaboran en sus operaciones, donde cada nodo generalmente corresponde a un sistema físico con una dirección IP de host única, que ejecuta uno o más servidores de aplicaciones. Estos clústeres tienen la capacidad de organizarse bajo la configuración de una célula, la cual establece de manera lógica la conexión entre diversos servidores y clústeres, adaptándose a diferentes configuraciones y aplicaciones según la decisión del administrador y las necesidades organizativas (IBM, 2023b).

El propósito principal de los clústeres es equilibrar la carga de trabajo entre los servidores. Los servidores que están integrados en un clúster se denominan miembros

del clúster. Cuando se instala una aplicación en un clúster, se instala automáticamente en cada miembro del clúster. Además, es posible configurar un clúster para proporcionar balanceo de carga integrando servicios o utilizando beans impulsados por mensajes en el servidor de aplicaciones.

#### ***2.2.4 Proxmox***

Proxmox VE es una plataforma de código abierto basada en Debian Linux, diseñada para ejecutar máquinas virtuales y contenedores. Diseñada para simplificar la administración, permite la gestión eficiente a través de su interfaz web, ya sea en un solo nodo o en clústeres de múltiples nodos (Proxmox Server Solutions GmbH, 2023b).

La interfaz facilita la gestión de máquinas virtuales, la implementación de características como alta disponibilidad y recuperación ante desastres. La plataforma que se integra con clústeres aborda cargas de trabajo exigentes para aplicaciones de Linux y Windows, permitiendo así la escalabilidad desde un solo nodo hasta un extenso clúster (Proxmox Server Solutions GmbH, 2023a).

Proxmox VE se utiliza para la computación en la nube y la consolidación de recursos de servidor al soportar aplicaciones basadas en máquinas virtuales VM y contenedores. Además, se destaca por facilitar la instalación, gestión y monitoreo de infraestructuras hiper-convergentes, ofreciendo así versatilidad en sus casos de uso. Una característica clave es la migración en vivo, que permite a los administradores mover máquinas virtuales entre nodos de clúster sin ninguna interrupción perceptible (Benalcázar Montesdeoca, 2019).

#### ***2.2.5 Proxmox en entornos de aprendizaje***

La plataforma Proxmox es una herramienta importante en el campo de la tecnología, que facilita la creación y administración eficiente de servidores virtuales. Siendo una herramienta multiplataforma, tiene capacidad para soportar varios sistemas

operativos, lo que proporciona los elementos básicos para un funcionamiento con un equilibrio óptimo entre costos y beneficios, especialmente debido a su licenciamiento gratuito (Castro, 2023).

Proxmox VE basado en la distribución Debian GNU/Linux ofrece a los estudiantes y educadores un entorno robusto para explorar y entender los principios fundamentales de la virtualización y administración de servidores. Las ventajas de esta plataforma en el entorno educativo incluyen:

- ✓ **Práctica Activa.** Esto permite a los estudiantes participar activamente en la creación y gestión de máquinas virtuales y contenedores, consolidando así la comprensión teórica con la experiencia práctica.
- ✓ **Configuración de Clústeres.** La capacidad de gestionar clústeres ofrece a los aprendices la oportunidad de explorar entornos distribuidos, comprendiendo conceptos clave.
- ✓ **Flexibilidad Tecnológica.** Al soportar tanto máquinas virtuales como contenedores, Proxmox permite a los estudiantes experimentar con diferentes tecnologías de virtualización, preparándolos para varios escenarios del mundo real.
- ✓ **Enfoque en Seguridad.** La función de sandboxing facilita la creación de entornos aislados que proporcionan a los estudiantes un espacio para llevar a cabo pruebas de seguridad sin representar ningún riesgo para los sistemas principales.
- ✓ **Gestión Eficiente.** La interfaz web intuitiva simplifica la administración central de recursos, proporcionando a los alumnos una visión práctica de cómo gestionar eficientemente sistemas en entornos virtuales.

- ✓ **Preparación para el mundo laboral.** Con Proxmox presente en entornos empresariales, los estudiantes adquieren habilidades y experiencia práctica que pueden aplicarse directamente en el trabajo, así como al prepararse para certificaciones relacionadas.

### ***2.2.6 Automatización en Proxmox***

Proxmox emerge como una solución altamente eficiente y flexible en el entorno virtual, ofreciendo capacidades integrales para crear, gestionar, configurar y almacenar máquinas virtuales. Proxmox subraya su versatilidad al proporcionar una API robusta que simplifica la invocación de comandos a la plataforma, permitiendo así la automatización y la integración fluida de sus funcionalidades en nuestra infraestructura (Carabas et al., 2023).

La API de Proxmox facilita la invocación de comandos y la realización de acciones directamente desde scripts o herramientas externas. Esta capacidad permite a los administradores automatizar procesos repetitivos como la creación y configuración de máquinas virtuales, la gestión de clústeres, el monitoreo del rendimiento, entre otras actividades.

La automatización en Proxmox no solo contribuye a la eficiencia operativa, sino que también promueve la coherencia en las configuraciones y la aplicación de políticas. El uso de scripts y herramientas de automatización minimiza las posibilidades de errores humanos, estableciendo así un entorno más controlado y predecible.

### **2.3 Herramientas de Automatización de Infraestructura**

La infraestructura como código (IaC) se basa en el uso de herramientas automatizadas que permiten la descripción, configuración y gestión de la infraestructura tecnológica a través del código. Estas herramientas ofrecen una manera eficiente de orquestar y mantener entornos tecnológicos complejos, haciendo que la gestión del

código de software sea lo más simple posible (IONOS Digital Guide, 2023). Cuando proporcionan sus propios lenguajes de configuración, estas herramientas de IaC simplifican la gestión de recursos de múltiples proveedores al eliminar la necesidad de un profundo entendimiento de las API correspondientes.

### ***2.3.1 Beneficios de las herramientas de IaC***

Las herramientas de Infraestructura como Código ofrecen una serie de beneficios significativos, entre los cuales se encuentran:

- ✓ **Automatización.** Las herramientas de IaC permiten una automatización eficiente de la creación, implementación y gestión de infraestructura. Esta automatización ahorra tiempo y esfuerzo sustancial por parte de los administradores del sistema.
- ✓ **Precisión.** Las herramientas de IaC juegan un papel crucial en asegurar la precisión y consistencia en la implementación de infraestructura. Este enfoque ayuda a mejorar la fiabilidad y seguridad de toda la infraestructura.
- ✓ **Transparencia.** Las herramientas de IaC mejoran la transparencia de la infraestructura al permitir que los administradores del sistema tengan una comprensión más profunda de sus operaciones. Esto facilita hacer cambios de manera segura y comprensible.

### ***2.3.2 Herramientas de IaC***

Las herramientas de Infraestructura como Código (IaC) son importantes en la automatización de la gestión y configuración de la infraestructura tecnológica. Algunas de estas herramientas se clasifican en función de su enfoque de la siguiente manera:

- ✓ **Herramientas de IaC multinube.** Estas herramientas permiten gestionar infraestructura en la nube de diferentes proveedores, abarcando entornos como AWS, Azure, GCP, entre otros. Entre las destacadas se encuentran: Terraform, Ansible, Chef, Puppet.

- ✓ **Herramientas de IaC de un solo proveedor.** Estas herramientas están específicamente diseñadas para gestionar infraestructura de un solo proveedor de servicios en la nube. Entre ellas se encuentran: CloudFormation (AWS), ARM (Azure) y Cloud Deployment Manager (GCP).

En la Tabla 2, se presentan algunas de las herramientas de Infraestructura como Código (IaC) junto con sus respectivas características.

**Tabla 2**  
*Tabla comparativa de las herramientas de IaC*

Característica	Terraform	Ansible	Chef	Puppet	AWS CloudFormation	Azure Resource Manager
Lenguaje de programación	HashiCorp (HCL)	YAML	Ruby	Ruby	JSON	JSON
Soporte de proveedores	AWS, Azure, Google Cloud Platform, Proxmox etc.	Nativo para cada proveedor de nube	Nativo para cada proveedor de nube	Nativo para cada proveedor de nube	AWS	Azure
Soporte de recursos	Incluye máquinas virtuales, contenedores, redes, almacenamiento, etc.					
Soporte de automatización	Buena integración con otras herramientas					
Dificultad de aprendizaje	Moderada, requiere un conocimiento básico de programación.	Fácil, requiere un conocimiento básico de YAML.	Moderada, requiere un conocimiento básico de Ruby.	Moderada, requiere un conocimiento básico de Ruby.	Moderada, requiere un conocimiento básico de programación.	Moderada, requiere un conocimiento básico de programación.
Coste	Gratuito	Gratuito	Gratuito	Gratuito	Gratuito con suscripción a AWS	Gratuito con suscripción a Azure

*Nota.* La tabla compara varias herramientas de Infraestructura como Código (IaC) en términos de lenguaje de programación, soporte de proveedores, soporte de recursos, soporte de automatización, dificultad de aprendizaje y costo.

### 2.3.3 Herramientas IaC compatibles con Proxmox

Proxmox VE, un sistema operativo de virtualización de código abierto es conocido por su compatibilidad con varias máquinas virtuales, contenedores y soluciones de almacenamiento. La integración de Proxmox con herramientas de Infraestructura como Código (IaC) en el entorno permite la automatización de procesos críticos asociados con la creación, despliegue y gestión de la infraestructura. Algunas de las herramientas compatibles incluyen:

**Terraform.** HashiCorp Terraform aparece como una herramienta de Infraestructura como Código (IaC) que permite la definición de recursos tanto instalaciones locales como en la nube utilizando archivos de configuración. Terraform utiliza un lenguaje declarativo para permitir una descripción concisa de la infraestructura, simplificando así la creación y el mantenimiento de entornos complejos. Estos archivos de configuración, al ser versionados, reutilizados y compartidos, facilitan la colaboración y la consistencia en el desarrollo y gestión de la infraestructura a lo largo de su ciclo de vida (HashiCorp, 2023).

Además, Terraform simplifica la creación y gestión de recursos en plataformas en la nube y otros servicios a través de interfaces de programación de aplicaciones (API). Sus proveedores ofrecen una amplia compatibilidad con diversas plataformas y servicios que permiten el uso de APIs accesibles para un aprovisionamiento eficiente y una gestión integral de la infraestructura en diversos entornos.

El proceso central del flujo de trabajo de Terraform consta de tres etapas clave, como se detalla en la figura establecida por (HashiCorp, 2023):

- ✓ **Write.** En esta etapa inicial, se definen los recursos como múltiples proveedores y servicios de TI en la nube. Los usuarios, a su vez, describen qué red se requiere

mediante archivos de configuración de manera declarativa, es decir, componentes y sus relaciones.

- ✓ **Plan.** Esto se aplica a Terraform ya que también proporciona y ejecuta un plan en ejecución que especifica los detalles de la infraestructura que se va a crear, actualizar o destruir. Este plan se basa en parámetros específicos en la infraestructura y en lo que ya existe, proporcionando una vista validada de los cambios que se realizarán antes del despliegue.
- ✓ **Apply.** Después de solicitar la aprobación del plan propuesto, Terraform lleva a cabo las operaciones que se han planificado en el orden especificado con respecto a las dependencias de los recursos que se van a utilizar.

Este método estructurado y por etapas permite un control fiable sobre las acciones que se van a realizar en la infraestructura, proporcionando un nivel razonable de control y estabilidad en el ciclo de vida de la gestión de la infraestructura.

**Ansible.** Ansible es una herramienta de automatización de TI de código abierto, permite a los administradores de sistemas automatizar diversas tareas relacionadas con la configuración, despliegue y gestión de sistemas y servicios. Con su modelo de cero-agente, Ansible se diferencia de otras herramientas que requieren agentes en sistemas remotos. En cambio, utiliza SSH para conectarse con sistemas remotos y ejecutar comandos o scripts (Red Hat, 2023a).

Las características clave de Ansible incluyen:

- ✓ **Simplicidad.** Ansible es conocido por ser una herramienta fácil de aprender y utilizar. Esto se debe a que su lenguaje de configuración, YAML, es legible y

comprensible, lo que facilita la creación y comprensión de scripts de automatización.

- ✓ **Versatilidad.** La versatilidad de Ansible es una fortaleza, ya que puede utilizarse para automatizar una amplia gama de tareas de TI. Desde la configuración de sistemas hasta el despliegue de aplicaciones, Ansible ofrece una solución integral para diversas necesidades.
- ✓ **Seguridad.** La seguridad es otro aspecto que distingue a Ansible de otras herramientas de automatización. Utiliza SSH para conectarse con sistemas remotos, proporcionando así una capa adicional de seguridad. Además, su lenguaje de configuración ha sido diseñado de tal manera que minimiza el riesgo de errores humanos, mejorando así la fiabilidad y la consistencia al ejecutar tareas automatizadas.

#### **2.4 Desafíos y oportunidades de la automatización de la infraestructura como código**

En la actualidad, las empresas enfrentan desafíos complejos, como brindar soporte a empleados y socios, ofrecer productos y servicios innovadores debido a la rápida evolución tecnológica. La Tecnología de la Información (TI) juega un papel vital en abordar estas demandas, aunque puede convertirse en una barrera si no logra adaptarse y cumplir con las expectativas.

El crecimiento empresarial trae consigo el problema de mantener entornos estables, uniformes y seguros. Esto se complica por lo difícil que es manejar al mismo tiempo tanto los sistemas y procesos heredados como los nuevos. Además, la velocidad con que aumentan los requisitos y las demandas supera la capacidad de desarrollo de las funciones empresariales y de TI. La magnitud de la tecnología, incluyendo elementos

como virtualización, la nube y contenedores hace prácticamente imposible cualquier gestión manual.

En este contexto cambiante, la automatización se ha vuelto uno de los componentes estratégicos más importantes. Se vuelve fundamental para administrar, modificar o ajustar infraestructuras de TI y operaciones empresariales. La automatización permite a los equipos de TI optimizar procesos complejos, aumentando eficiencia, productividad y flexibilidad al tiempo que disminuyen costes y minimizan errores humanos (Red Hat, 2022, 2023a).

#### ***2.4.1 Oportunidades para mejorar la enseñanza con la IaC en entornos de virtualización***

En el actual panorama educativo, caracterizado por una transición a modalidades remotas, la necesidad de automatizar los exámenes prácticos de laboratorios de software ha sido abordada por (Sorour & Hamdy, 2022) a través de prácticas de DevOps y Infraestructura como Código (IaC). Este enfoque ofrece oportunidades significativas para mejorar la calidad de la enseñanza en entornos virtuales.

La aplicación de IaC se destaca como una solución prometedora a los desafíos identificados, como la diversidad en las configuraciones de computadoras personales de los estudiantes y las limitaciones de potencia de hardware. Una propuesta para una arquitectura de software en la nube escalable y altamente disponible, respaldada por tecnologías modernas de nube y principios de DevOps, forma una base sólida para la automatización eficiente de la entrega de exámenes de laboratorios de software.

IaC permite la creación rápida de instancias preconfiguradas en la nube que incluyen todo el material necesario para el examen y su disponibilidad para su uso en un tiempo excepcionalmente corto. Además, la implementación en un clúster de

Kubernetes proporciona escalabilidad y manejo efectivo de cargas inesperadas a través de sus funciones automáticas de escalado y recuperación.

En este sentido, la adopción de IaC no solo soluciona problemas específicos como la variabilidad del hardware y las limitaciones de potencia, sino que también puede ser una oportunidad para mejorar el aprendizaje en entornos virtuales. La habilidad para crear, administrar y compartir laboratorios automáticamente y de manera eficiente con el apoyo de IaC contribuye enormemente a mejorar la calidad y accesibilidad del aprendizaje práctico en línea.

#### ***2.4.2 Casos de éxito en la implementación de virtualización e IaC***

La exitosa implementación de "Automated Infrastructure Provisioning", tal como lo desarrollaron (Carabas, Mihai, & Brebene, 2023), subraya el impacto transformador de la Infraestructura como Código (IaC) en la administración de sistemas. Este sistema automatizado ha revolucionado el proceso de provisionamiento de infraestructura y tiene beneficios sustanciales en términos de eficiencia, seguridad y gestión.

El enfoque del artículo es el desarrollo de un sistema de provisionamiento de infraestructura automatizado que simplifica la creación y gestión de una red de máquinas virtuales. Utiliza tecnologías de virtualización para configurar rápidamente un grupo de máquinas, las cuales se pueden ajustar fácilmente para satisfacer las necesidades del usuario. Además, el sistema proporciona herramientas para la gestión eficiente de las máquinas y sus correspondientes recursos.

Reconociendo las limitaciones de tiempo y la priorización de actividades, es común que los ingenieros se centren en la creación de la infraestructura, dejando de lado la implementación de soluciones de monitoreo. Para abordar este desafío, utilizan un contenedor Docker que instala y configura una solución de monitoreo completamente

funcional. La elección de Docker garantiza flexibilidad y portabilidad, permitiendo que la solución del monitor funcione en cualquier entorno independientemente del código utilizado para crear y configurar las máquinas virtuales. Este enfoque posibilita proporcionar una solución modular adaptable a diversos escenarios.

El estudio "Adoption, Support, and Challenges of Infrastructure-as-Code: Insights from Industry" realizado por (Guerriero, Garriga, Tamburri, & Palomba, 2019) es otro ejemplo sobresaliente de éxito en la implementación de la Infraestructura como Código (IaC). En el marco de DevOps, esta estrategia es fundamental ya que se enfoca en la gestión y aprovisionamiento de infraestructura a través de archivos de definición legibles por máquina en lugar de depender del hardware físico o herramientas interactivas.

Los profesionales de la industria y académicos han mostrado un considerable interés en este enfoque porque hay relativamente pocos patrones de soporte, mejores prácticas o técnicas de ingeniería de software diseñadas específicamente para mantener y evolucionar IaC de manera efectiva.

## **CAPÍTULO III: CASO DE ESTUDIO**

En este capítulo se analizará la infraestructura de los laboratorios diseñados para apoyar el desarrollo académico en el área de redes. El análisis tiene como objetivo principal identificar las necesidades específicas de recursos y configuraciones que puedan mejorar las prácticas de laboratorio.

A través de este proceso, se realizará una evaluación detallada sobre los recursos hardware y software necesarios para implementar las prácticas de laboratorio. Además, se discutirán configuraciones específicas requeridas para asegurarse de que los laboratorios cumplan con los objetivos educativos establecidos.

Este análisis se centrará en asignaturas clave del plan curricular relacionadas con redes tales como Redes de Área Local, Redes de Área Amplia, Sistemas de Comunicación Multimedia, Seguridad de Redes, Ingeniería de Tráfico y Administración de Redes de Telecomunicaciones. Estas áreas de estudio requieren un entorno de laboratorio adecuadamente equipado y configurado que permita a los estudiantes adquirir habilidades prácticas y aplicar eficazmente los conceptos teóricos.

Al final de este análisis, se espera obtener una comprensión clara de los requisitos y configuraciones específicas de infraestructura necesarias para optimizar las prácticas de laboratorio en redes.

### **3.1 Análisis actual del clúster de virtualización**

Para el análisis del clúster de virtualización, se implementó un cuestionario que incluía preguntas abiertas como se muestra en el Anexo A. Estas preguntas cubrieron información relevante sobre las características de hardware y software del clúster que son necesarias para un análisis integral y detallado.

Considerando los datos recopilados del cuestionario sobre el clúster de virtualización, su principal objetivo es proporcionar a estudiantes y docentes un entorno de pruebas para prácticas de laboratorio e investigación, facilitando así el acceso a un entorno virtualizado. La disponibilidad del clúster depende de la conectividad de la red universitaria, ya que está conectado a una LAN privada y dispone de un UPS que alimenta computadoras y servidores durante aproximadamente una hora.

El clúster de virtualización soporta máquinas virtuales utilizadas para pruebas y producción en asignaturas como Sistemas de Comunicación Multimedia, Seguridad de Redes, Redes de Área Extensa, Gestión de Redes de Telecomunicaciones y Titulación II. Este entorno proporciona un soporte esencial para actividades académicas y de investigación en el área de redes y telecomunicaciones.

### ***3.1.1 Características de hardware***

La infraestructura se compone de tres nodos físicos conectados y un nodo adicional destinado a pruebas. Estos nodos incluyen recursos de hardware como unidad de procesamiento central (CPUs), memoria RAM y almacenamiento, que respaldan las operaciones de virtualización. Los recursos se detallan en la Tabla 3.

**Tabla 3**  
*Características de hardware*

<b>Componente</b>	<b>Cantidad</b>
<b>CPU</b>	48 CPUs
<b>Memoria</b>	93.64 GB
<b>Almacenamiento</b>	2.14 TB

*Nota:* La tabla presenta la cantidad de recursos de hardware del cluster de virtualización, incluyendo la cantidad de CPUs, memoria y almacenamiento disponible.

### 3.1.2 Características de software

El clúster utiliza Proxmox versión 4.7-3 como plataforma de virtualización, que proporciona un rendimiento estable y opera horas del día, los 7 días de la semana para satisfacer las necesidades académicas.

La Tabla 4 a continuación muestra la utilización promedio de los componentes de infraestructura de virtualización:

**Tabla 4**

*Promedio de utilización de los componentes de infraestructura de virtualización*

<b>Componente</b>	<b>Promedio de utilización</b>
<b>CPU</b>	3 %
<b>Memoria RAM</b>	27%
<b>Almacenamiento</b>	44%

*Nota:* La tabla muestra el promedio de utilización de recursos del clúster de virtualización, incluyendo CPU, memoria RAM y almacenamiento.

## 3.2 Análisis de asignaturas de redes

Para llevar a cabo el análisis de las asignaturas de redes y telecomunicaciones utilizando un cuestionario que incluía preguntas abiertas detalladas como se detalla en el Anexo B. Estas preguntas abordaron aspectos clave de las asignaturas, como Redes de Área Local, Redes de Área Extensa, Sistemas de Comunicación Multimedia, Seguridad de Redes, Ingeniería de Tráfico, Administración de Redes de Telecomunicaciones.

### 3.2.1 Redes de Área Local

La asignatura de Redes de Área Local (LAN) se centra en examinar los principios fundamentales que subyacen al diseño y operación de las LAN, considerando

aspectos como protocolos, interfaces, arquitectura, servicios, conmutación, direccionamiento, control de errores, control de flujo, así como los mecanismos y algoritmos clave que subyacen a las funciones esenciales de las redes de comunicación de área local.

A través de esta asignatura teórico-práctica, los estudiantes tienen la oportunidad de profundizar en la estructura y el funcionamiento de las redes de área local, desarrollando así competencias críticas para diseñar, configurar y gestionar entornos de redes a pequeña escala. El análisis detallado de estos conceptos permite a los estudiantes aplicar su conocimiento para resolver problemas reales en telecomunicaciones, mejorando así una comprensión integral de las tecnologías de red que apoyan las infraestructuras de comunicación modernas.

### ***3.2.2 Redes de Área Extensa***

Redes de Área Extensa (WAN) es una asignatura que trata sobre el estudio de redes de datos que cubren áreas geográficas extensas. Se profundiza en varios aspectos clave para entender cómo funcionan y se administran estas redes. Entre estos aspectos se incluyen los protocolos de transporte y los servicios esenciales, que se exploran mediante la configuración práctica de servidores y clientes. Para WAN, esta configuración permite a los estudiantes comprender cómo se ofrecen y reciben diferentes servicios en entornos de redes a gran escala, ya que se implementa un protocolo específico.

### ***3.2.3 Sistemas de Comunicación Multimedia***

La asignatura de Sistemas de Comunicación Multimedia ofrece un estudio detallado de los protocolos de comunicación y los servicios que soportan las aplicaciones multimedia en las redes actuales. Proporcionando una comprensión profunda de los mecanismos y protocolos que sustentan la transmisión de datos

multimedia a través de las redes. Mediante el estudio de protocolos como TCP, UDP, y servicios de capa de aplicación como HTTP, DNS, bases de datos y streaming, VoIP, los estudiantes adquieren conocimientos esenciales para el manejo eficiente de contenido multimedia. Además, se analizan las características de transmisión de audio y video, haciendo hincapié en protocolos en tiempo real y en las arquitecturas de redes que optimizan la distribución de este tipo de contenido. El análisis métrico incluye latencia, jitter y RTT para evaluar y mejorar la calidad del servicio multimedia en redes modernas.

#### ***3.2.4 Seguridad en Redes***

El objetivo principal de la asignatura de seguridad en redes es garantizar que los sistemas y las comunicaciones sean confidenciales e íntegros al proporcionar a los estudiantes conocimientos y habilidades necesarios para proteger eficazmente las redes y sistemas informáticos. Esto incluye una comprensión completa de los principios, herramientas y prácticas críticas para proteger sistemas y comunicaciones en entornos informáticos. Desde el entendimiento de las amenazas hasta la implementación de medidas de seguridad y el cumplimiento de regulaciones legales aplicables, se tratan aspectos cruciales para asegurar la seguridad en el ámbito de las redes informáticas.

#### ***3.2.5 Ingeniería en Tráfico***

La asignatura de Ingeniería de Tráfico se dedica al análisis del tráfico que afecta a las redes de telecomunicaciones con el objetivo de mejorar su eficiencia y rendimiento. En esta asignatura, se exploran varios aspectos relacionados con la gestión del flujo de datos y la optimización en estas redes. Es responsable de analizar, planificar y optimizar el flujo de datos en las redes de telecomunicaciones para mejorar su eficiencia, rendimiento y calidad del servicio. Esta materia prepara a los estudiantes a través del estudio de patrones de tráfico, dimensionamiento de redes, planificación de

recursos y control de congestión para diseñar y gestionar infraestructuras de red que puedan satisfacer de manera efectiva las demandas de conectividad actuales y futuras.

### ***3.2.6 Administración de Redes de Telecomunicaciones***

El objetivo principal de la asignatura de Administración de Redes de Telecomunicaciones es examinar cómo diferentes tecnologías de infraestructura de red se pueden gestionar y monitorear de manera central a través de plataformas de gestión y protocolos de gestión de red. En este sentido, se proporciona a los estudiantes los conocimientos y habilidades necesarios para una administración efectiva de la infraestructura de red y garantizar su rendimiento óptimo. Al comprender cómo gestionar y monitorear redes de manera central, los estudiantes están preparados para enfrentar desafíos en la gestión de redes.

### **3.3 Análisis de prácticas de laboratorio de las asignaturas**

Para el análisis de las asignaturas de redes y telecomunicaciones, se implementó un cuestionario que incluyó preguntas abiertas como se muestra en el Anexo B. Estas preguntas cubrieron información relevante para identificar necesidades específicas para los laboratorios en el área de redes de Telecomunicaciones en la Universidad Técnica del Norte, con miras a mejorar la calidad y eficiencia de las experiencias de aprendizaje práctico para los estudiantes.

El presente proyecto tiene como objetivo abordar varias áreas clave, desde la configuración de equipos de red hasta la gestión y seguridad de redes, incorporando tecnologías de infraestructura como código. Este enfoque promete no solo modernizar la gestión de infraestructura, sino también proporcionar una plataforma flexible y escalable para implementar y gestionar entornos de laboratorio.

### 3.3.1 Análisis de prácticas de laboratorio en Redes de Área Local

En este análisis de las prácticas de laboratorio en Redes de Área Local, se recopiló información sobre las prácticas de laboratorio y los recursos utilizados en esta asignatura específica. Esto es para destacar la importancia de la experiencia práctica en el desarrollo de habilidades necesarias para enfrentar los desafíos inherentes a la implementación y mantenimiento de redes de área local en diversos entornos profesionales.

Las prácticas de laboratorio se realizan cada quince días con una duración de dos horas por sesión. Durante estas prácticas, los estudiantes hacen configuraciones en routers y switches, tanto físicos como virtuales, utilizando equipos hechos por fabricantes reconocidos como Cisco y MikroTik. Para llevar a cabo actividades prácticas, se usan herramientas como Packet Tracer, GNS3 y Wireshark que simulan redes.

A continuación, en la Tabla 5 se detalla el nombre de las guías de laboratorio desarrolladas para la asignatura de Redes de Área Local.

**Tabla 5**  
*Guías de prácticas de laboratorio para la asignatura de Redes de Área Local*

<b>N° de guía de laboratorio</b>	<b>Nombre de la guía de laboratorio</b>
<b>Lab 01</b>	Uso de simuladores
<b>Lab 02</b>	Introducción a equipos SW
<b>Lab 03</b>	HDLC y PPP
<b>Lab 04</b>	Dispositivos interconexión, Capturas de Tramas Ethernet
<b>Lab 05</b>	Tabla MAC - ARP

<b>Lab 06</b>	Selección del hardware de switching (acceso, distribución, core) STP
<b>Lab 07</b>	VLAN, 802.1Q HSRP, Link Aggregation, Routing VLANs

*Nota.* La tabla presenta una lista de guías de laboratorio para la asignatura de Redes de Área Local.

### ***3.3.2 Análisis de prácticas de laboratorio en Redes de Área Extensa***

Con respecto al análisis de las prácticas de laboratorio en Redes de Área Extensa, se recopiló información centrándose en las actividades prácticas y los recursos utilizados en esta materia específica. Estas prácticas son cruciales para proporcionar a los estudiantes una comprensión práctica y aplicada de los conceptos teóricos, lo que les permite desarrollar las habilidades necesarias para enfrentar los desafíos en la implementación y el mantenimiento de redes de área extensa en diferentes entornos.

Las prácticas de laboratorio se realizan cada quince días, con una duración de dos horas cada una. Durante estas prácticas, los estudiantes configuran routers y switch, tanto físicos como virtuales, utilizando equipos de fabricantes reconocidos como Cisco y MikroTik. Esta metodología permite a los estudiantes trabajar con hardware real, así como con herramientas de simulación que proporcionan profundas perspectivas sobre los desafíos y requisitos de las WAN.

En la tabla 6 se detallan las guías de prácticas de laboratorio que se llevarán a cabo en la asignatura de Redes de Área Extensa. Estas prácticas abarcan temas como fundamentos de simulación, protocolos de enrutamiento, tecnologías de conmutación, virtualización y servicios de red.

**Tabla 6**

*Guías de prácticas de laboratorio para la asignatura de Redes de Área Extensa*

<b>N° de guía de laboratorio</b>	<b>Nombre de la guía de laboratorio</b>
<b>Lab 01</b>	Instalación GNS3 – configuraciones básicas: direccionamiento, SSH, y Telnet DHCP/NAT
<b>Lab 02</b>	RIP
<b>Lab 03</b>	OSPF
<b>Lab 04</b>	ATM – FRAME RELAY
<b>Lab 05</b>	MPLS, VPN, BGP
<b>Lab 06</b>	Virtualización Proxmox
<b>Lab 07</b>	DNS, EMAIL, HTTP

*Nota.* La tabla presenta una lista de guías de laboratorio para la asignatura de Redes de Área Extensa.

### ***3.3.3 Análisis de prácticas de laboratorio en Sistemas de Comunicación Multimedia***

Para el análisis de las prácticas de laboratorio de Sistemas de Comunicación Multimedia se recopiló información específica sobre las actividades prácticas y los recursos utilizados en esta asignatura específica. Estas actividades se centran en la configuración de servicios y en el uso de herramientas de análisis de tráfico. Durante las prácticas, se realiza la captura de paquetes que posteriormente se analizan, especialmente en relación con servicios que operan sobre los protocolos TCP y UDP para aplicaciones en tiempo real.

Estas prácticas de laboratorio se llevan a cabo cada 15 días, con un total de 8 laboratorios durante el semestre, distribuidos en 4 prácticas en el primer parcial y 4 prácticas en el segundo parcial, tal como se especifica en la tabla 7.

**Tabla 7**

*Guías de prácticas de laboratorio para la asignatura de Sistemas de Comunicación Multimedia*

<b>N° de guía de laboratorio</b>	<b>Nombre de la guía de laboratorio</b>
<b>Lab 01</b>	Simulación Servicios Multimedia
<b>Lab 02</b>	Aplicación Servicios Multimedia
<b>Lab 03</b>	Simulación /Aplicación Capa Transporte
<b>Lab 04</b>	Propiedades de audio/video
<b>Lab 05</b>	Streaming (VOD)
<b>Lab 06</b>	Transmisión adaptativa (HLS)
<b>Lab 07</b>	Instalación y Configuración PBX virtual RTSP-SIP
<b>Lab 08</b>	Best-Effort

*Nota.* La tabla presenta una lista de guías de laboratorio para la asignatura de Sistemas de Comunicación Multimedia.

Uno de los principales desafíos en estas prácticas es realizar un análisis de rendimiento en la comunicación basada en captura de paquetes. Este ejercicio necesita una comprensión profunda del tráfico de red y las métricas asociadas para evaluar el rendimiento de los servicios de comunicación multimedia.

### ***3.3.4 Análisis de prácticas de laboratorio en Seguridad en Redes***

Las prácticas de laboratorio en la asignatura de Seguridad de Redes se realizan cada quince días, con un total de 8 sesiones durante el semestre, distribuidas en 4 prácticas en el primer parcial y otras 4 en el segundo, como se detalla en la Tabla 8. Durante estas prácticas, se abordan los fundamentos básicos de seguridad, comenzando con el reconocimiento de herramientas y servicios, como sistemas operativos Kali Linux y servidores de prueba. Los estudiantes realizan o reconocen ataques dirigidos, como los de denegación de servicio (DoS), y luego se llevan a cabo prácticas de

mitigación. El objetivo es que los estudiantes apliquen contramedidas basadas en ACLs, reglas de firewall, VPNs, IDS y políticas de seguridad para mitigar los riesgos.

Uno de los desafíos principales durante estas prácticas implica llevar a cabo un ataque exitoso en el ámbito de la seguridad, así como analizar paquetes relacionados con dicho ataque. Este ejercicio permite a los estudiantes comprender en profundidad las vulnerabilidades de seguridad y las medidas necesarias para proteger las redes contra posibles amenazas.

**Tabla 8**

*Guías de prácticas de laboratorio para la asignatura de Seguridad en Redes*

<b>N° de guía de laboratorio</b>	<b>Nombre de la guía de laboratorio</b>
<b>Lab 01</b>	NEXPOSE y evaluación de las vulnerabilidades de un servidor asignado
<b>Lab 02</b>	Generar claves RSA
<b>Lab 03</b>	Herramientas de seguridad de Microsoft MBSA y MSAT
<b>Lab 04</b>	VLAN y Port Security IPv4 – Ipv6
<b>Lab 05</b>	Configuración de ACLs y VPN de acceso remoto en Router Cisco y Mikrotik
<b>Lab 06</b>	Instalación y configuración de firewall y aplicar iptables
<b>Lab 07</b>	Uso de metodología de auditoría y análisis de vulnerabilidades desde una red

<b>Lab 08</b>	Análisis de riesgos sencillo, dado un caso de uso, complementado con algunas cuestiones teóricas
---------------	--

*Nota.* La tabla presenta una lista de guías de laboratorio para la asignatura de Seguridad de Redes.

### ***3.3.5 Análisis de prácticas de laboratorio en Ingeniería en Tráfico***

El análisis de las prácticas de laboratorio en la asignatura de Ingeniería de Tráfico comienza con actividades básicas que incluyen el uso de herramientas y el análisis del tráfico para la transmisión de diversos servicios. El objetivo de estas actividades es lograr la implementación de mecanismos que permitan un tratamiento específico para cada tipo de tráfico, lo que requiere técnicas más avanzadas y especializadas.

Durante estas prácticas, los estudiantes tienen la oportunidad de aplicar los conocimientos teóricos adquiridos en clase y desarrollar habilidades prácticas en el manejo de herramientas de análisis de tráfico. Además, se les presenta el desafío de trabajar con topologías de red convergentes, lo que implica integrar múltiples tipos de tráfico y una gestión eficiente de los recursos de la red.

Las prácticas de laboratorio se llevan a cabo cada quince días, totalizando 8 sesiones por semestre distribuidas en 4 laboratorios en el primer parcial y otros 4 en el segundo, como se muestra en la Tabla 9.

**Tabla 9**

*Guías de prácticas de laboratorio para la asignatura de Ingeniería de Tráfico*

<b>N° de guía de laboratorio</b>	<b>Nombre de la guía de laboratorio</b>
<b>Lab 01</b>	Limitaciones Tecnológicas de la conmutación de paquetes

<b>Lab 02</b>	Iperf
<b>Lab 03</b>	Limitaciones Tecnológicas de la conmutación de paquetes STP
<b>Lab 04</b>	Limitaciones PVSTP y MSTP
<b>Lab 05</b>	MPLS
<b>Lab 06</b>	QoS aplicaciones
<b>Lab 07</b>	MPLS VPN
<b>Lab 08</b>	Tc-netem

*Nota.* La tabla presenta una lista de guías de laboratorio para la asignatura de Ingeniería de Tráfico.

### ***3.3.6 Análisis de prácticas de laboratorio en Administración de Redes de Telecomunicaciones***

El análisis de las prácticas de laboratorio en la asignatura de Administración de Redes de Telecomunicaciones se centra en la gestión de la infraestructura de dispositivos de red utilizando software de gestión específico. Durante las sesiones de laboratorio, los estudiantes trabajan en la configuración de agentes SNMP en diferentes versiones, implementan servidores SNMP para solicitar métricas de los agentes y realizan configuraciones relacionadas con la gestión de registros y la configuración de sistemas de tickets. Además, se exploran herramientas de monitoreo y optimización como Zabbix, Graylog y Elastic.

Se llevan a cabo prácticas de laboratorio semanales, totalizando 13 laboratorios por semestre que cubren una variedad de temas, desde la selección y configuración de administradores de red hasta la implementación de sistemas de monitoreo y gestión como se muestra en la Tabla 10. Para este propósito, se utilizan varias herramientas de software específicas de las prácticas, incluyendo GNS3, VirtualBox, sistemas

operativos de dispositivos de red, así como herramientas de monitoreo y gestión.

También se emplean herramientas de gestión de registros como Graylog y Wazuh.

**Tabla 10**

*Guías de prácticas de laboratorio para la asignatura de Administración de Redes de Telecomunicaciones*

<b>N° de guía de laboratorio</b>	<b>Nombre de la guía de laboratorio</b>
<b>Lab 01</b>	Selección de gestor de red
<b>Lab 02</b>	Configuración de gestores de red
<b>Lab 03</b>	Gestor de red con topología
<b>Lab 04</b>	Configuración SNMP
<b>Lab 05</b>	SNMP con topología
<b>Lab 06</b>	Configuración SNMPv3
<b>Lab 07</b>	SNMPv3 con topología
<b>Lab 08</b>	Configuración Logs
<b>Lab 09</b>	Logs con topología
<b>Lab 10</b>	Configuración RMON
<b>Lab 11</b>	Topología RMON
<b>Lab 12</b>	Configuración GLPI
<b>Lab 13</b>	Topología GLPI

*Nota.* La tabla presenta una lista de guías de laboratorio para la asignatura de Administración de Redes de Telecomunicaciones.

Los aspectos más desafiantes durante las prácticas incluyen las configuraciones de las máquinas virtuales, que pueden consumir mucho tiempo, y asegurarse de que el host anfitrión tenga los recursos suficientes en términos de hardware y software. Estos desafíos resaltan cuán importante es tener una infraestructura adecuada para llevar a cabo las prácticas en los laboratorios.

### 3.4 Selección de asignaturas con mayor consumo de recursos

Para la selección de las asignaturas con mayor consumo de recursos, se han considerado varios factores basados en entrevistas realizadas a docentes de asignaturas de redes y telecomunicaciones y en el análisis de prácticas de laboratorio. Se ha podido determinar que algunas materias requieren un consumo significativo de recursos en términos de hardware, software y tiempo de configuración utilizando este enfoque. También se tiene en cuenta que los cursos de nivel superior requieren conocimientos previos adquiridos de Redes de Área Local, Redes de Área Extensa, Sistemas de Comunicación Multimedia, así como Seguridad de Redes.

A continuación, se presentan las asignaturas identificadas con alto consumo de recursos:

#### *Ingeniería de Tráfico*

Por varios motivos, las prácticas de laboratorio en la asignatura de Ingeniería del Tráfico consumen muchos recursos. El análisis de tráfico y la calidad de servicio (QoS) requieren configuraciones precisas y detalladas, lo que implica un uso intensivo de recursos técnicos y humanos. Además, para que las herramientas de virtualización y análisis usadas en esta asignatura funcionen correctamente, se necesita una cantidad considerable de hardware y software.

La implementación de tecnologías avanzadas como MPLS y VPN requiere una infraestructura robusta y adecuada para soportar las complejidades y exigencias de estos sistemas. Estas razones explican el alto consumo de recursos en la asignatura de Ingeniería del Tráfico como se detalla en la Tabla 11.

**Tabla 11**

*Recursos utilizados para las prácticas de laboratorio en la asignatura de Ingeniería de Tráfico*

<b>Recursos Utilizados</b>	
<b>Hardware:</b>	Routers, switch.
<b>Software:</b>	Iperf, herramientas de análisis de tráfico.
<b>Prácticas:</b>	Análisis y tratamiento específico del tráfico, implementación de QoS, MPLS, VPN.
<b>Frecuencia de las prácticas de laboratorio:</b>	Se lleva a cabo prácticas de laboratorio cada 15 días, teniendo un total de 8 laboratorios por semestre.

*Nota.* La tabla presenta los recursos utilizados en las prácticas de laboratorio. Estos recursos incluyen hardware como routers y switches, software como Iperf y herramientas de análisis de tráfico, y prácticas específicas como análisis y tratamiento del tráfico, implementación de QoS, MPLS y VPN.

La implementación de infraestructura como código podría ser particularmente beneficiosa en estas asignaturas para automatizar y gestionar eficientemente los recursos, asegurando configuraciones consistentes y reproducibles, y optimizando el uso de la infraestructura disponible.

### ***Administración de Redes de Telecomunicaciones***

Tomando en cuenta las prácticas de laboratorio en la asignatura, se considera una serie de factores que contribuyen a un consumo intensivo de recursos. La configuración y gestión de una variedad de dispositivos y herramientas demandan un uso significativo de recursos, ya que cada dispositivo requiere su propio conjunto de configuraciones y mantenimiento. Además, la necesidad de hardware y software adecuados para soportar prácticas complejas y variadas incrementa la demanda de recursos, ya que estos sistemas deben ser lo suficientemente robustos para manejar

múltiples tareas simultáneamente. Por último, la configuración de máquinas virtuales, esenciales para muchas operaciones, implica un uso considerable de recursos en el host anfitrión, ya que cada máquina virtual necesita una porción significativa de la capacidad de procesamiento y memoria disponible.

La Tabla 12 muestra los recursos utilizados para el desarrollo de las prácticas de laboratorio en la asignatura.

**Tabla 12**

*Recursos utilizados para las prácticas de laboratorio en la asignatura de Administración de Redes de Telecomunicaciones*

<b>Recursos Utilizados</b>	
<b>Hardware:</b>	Routers, switch.
<b>Software:</b>	GNS3, VirtualBox, sistemas operativos de dispositivos de red, herramientas de monitoreo y gestión.
<b>Prácticas:</b>	Configuración de agentes SNMP, gestión de logs, configuración de sistemas de gestión de tickets.
<b>Frecuencia de las prácticas de laboratorio:</b>	Se lleva a cabo prácticas de laboratorio semanal, teniendo un total de 13 laboratorios por semestre.

*Nota.* La tabla presenta los recursos utilizados en las prácticas de laboratorio. Estos recursos incluyen hardware como routers y switches, software como GNS3, VirtualBox, sistemas operativos de dispositivos de red y herramientas de monitoreo y gestión. Las prácticas específicas incluyen la configuración de agentes SNMP, la gestión de logs y la configuración de sistemas de gestión de tickets.

## **CAPÍTULO IV: IMPLENTACIÓN DE SCRIPTS Y PLANTILLAS DE INFRAESTRUCTURA COMO CÓDIGO**

En este capítulo, profundizaremos en la fase final del proyecto que implica desarrollar scripts y plantillas para automatizar la infraestructura siguiendo la arquitectura previamente delineada. Esto se realizará utilizando dos herramientas complementarias, como son; Terraform y Ansible.

Terraform se utilizará para aprovisionar recursos en la plataforma de virtualización Proxmox, asegurando que los recursos requeridos se creen y configuren de manera eficiente y repetible. Por otro lado, Ansible gestionará la configuración de estos recursos automatizando la instalación de software, la configuración de servicios, entre otras tareas necesarias para que la infraestructura funcione como se espera.

Para llevar a cabo este proceso, comenzará con la instalación y configuración de las herramientas necesarias, seguida del desarrollo de los scripts y plantillas requeridos para la automatización. Una guía detallada de este procedimiento se incluye en el manual de instalación disponible en el Anexo C. Además, se implementarán pruebas automatizadas para validar la funcionalidad de los scripts, de modo que no aparezcan errores después de introducir cambios, garantizando así una integración continua sin inconvenientes.

### **4.1 Integración de Terraform y Ansible**

Terraform y Ansible son herramientas esenciales en el campo de la infraestructura como código (IaC), en este caso se utilizan como complemento, lo que permiten una automatización integral y eficiente de la infraestructura.

De esta manera, Terraform se utiliza para definir e implementar infraestructura de forma declarativa. Así, es útil para crear y administrar recursos en diferentes proveedores de nube y plataformas de virtualización como Proxmox. Para esto, los archivos de configuración estarán escritos en HCL (HashiCorp Configuration Language) que describe la infraestructura deseada. Después de ser definidos, estos archivos se ejecutan para crear automáticamente y gestionar los recursos garantizando consistencia y repetibilidad.

Por otro lado, Ansible trata sobre el manejo de configuraciones que se usan para controlar las configuraciones del sistema operativo y aplicaciones una vez que los recursos han sido aprovisionados. Utiliza playbooks en formato YAML para automatizar tareas de configuración, despliegue de aplicaciones u orquestación de tareas complejas en varios servidores.

#### ***4.1.1 Instalación de Terraform***

Terraform es una herramienta de código abierto que permite la infraestructura como código. A continuación, se describen los pasos para la instalación de Terraform en el sistema operativo Linux.

**Actualización del sistema.** Para comenzar la instalación del software, se debe actualizar los paquetes del sistema como se muestra en la Figura 3.

**Figura 3**  
Actualización del sistema

```

lizabeth@lizabeth-virtual-machine:~$ sudo apt update
[sudo] password for lizabeth:
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [129 kB]
Hit:2 http://ec.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://ec.archive.ubuntu.com/ubuntu jammy-updates InRelease [128 kB]
Hit:4 http://ec.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:5 http://ec.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1.7
31 kB]
Get:6 http://ec.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [642
kB]
Fetched 2.630 kB in 4s (671 kB/s)
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
430 packages can be upgraded. Run 'apt list --upgradable' to see them.

```

*Nota.* El gráfico ilustra el proceso de actualización del sistema en la máquina virtual, donde se instalarán las herramientas de Infraestructura como Código (IaC).

**Instalación de dependencias.** Para preparar el sistema y manejar repositorios de software adicionales, es fundamental instalar los paquetes necesarios que faciliten la gestión de repositorios HTTPS y aseguren la autenticidad e integridad de los paquetes. Esto garantiza que el sistema pueda agregar nuevos repositorios de manera segura y eficiente. A continuación, se detallan los pasos para la instalación de estos paquetes como se ilustra en la Figura 4.

**Figura 4**  
Instalación de dependencias

```

lizabeth@lizabeth-virtual-machine:~$ sudo apt install -y software-properties-commo
n gnupg
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
gnupg is already the newest version (2.2.27-3ubuntu2.1).
gnupg set to manually installed.
The following additional packages will be installed:
  python3-software-properties software-properties-gtk
The following packages will be upgraded:
  python3-software-properties software-properties-common
  software-properties-gtk
3 upgraded, 0 newly installed, 0 to remove and 427 not upgraded.
Need to get 114 kB of archives.
After this operation, 8,192 B of additional disk space will be used.
Get:1 http://ec.archive.ubuntu.com/ubuntu jammy-updates/main amd64 software-prop
erties-common all 0.99.22.9 [14,1 kB]
Get:2 http://ec.archive.ubuntu.com/ubuntu jammy-updates/main amd64 software-prop
erties-gtk all 0.99.22.9 [71,3 kB]
Get:3 http://ec.archive.ubuntu.com/ubuntu jammy-updates/main amd64 python3-softw
are-properties all 0.99.22.9 [28,8 kB]
Fetched 114 kB in 1s (113 kB/s)
(Reading database ... 90%

```

*Nota.* El gráfico ilustra la instalación de dependencias necesarias que facilitan la gestión de repositorios HTTPS y aseguran la autenticidad e integridad de los paquetes instalados en el sistema.

**Agregación de repositorio de HashiCorp.** Para instalar Terraform, se debe agregar el repositorio oficial de HashiCorp al sistema. Este proceso comienza con la adición de la clave GPG de HashiCorp, que garantiza la autenticidad e integridad de los paquetes descargados desde el repositorio, como se muestra en la Figura 5.

**Figura 5**  
Añadir la clave GPG de HashiCorp

```
lizabeth@lizabeth-virtual-machine:~$ curl -fsSL https://apt.releases.hashicorp.com/gpg | sudo apt-key add -
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see apt-key(8)).
OK
lizabeth@lizabeth-virtual-machine:~$
```

*Nota.* El gráfico ilustra la adición de la clave GPG de HashiCorp al sistema.

A continuación, se añade el repositorio oficial de HashiCorp a la lista de fuentes de APT. Este paso permite que el sistema APT reconozca y descargue los paquetes de Terraform directamente desde el repositorio oficial de HashiCorp, garantizando que se obtengan versiones actualizadas y auténticas del software, como se muestra en la Figura 6.

**Figura 6**  
Agregación del repositorio de HashiCorp a la lista de fuentes de APT.

```
lizabeth@lizabeth-virtual-machine:~$ sudo apt-add-repository "deb [arch=amd64] https://apt.releases.hashicorp.com $(lsb_release -cs) main"
Repository: 'deb [arch=amd64] https://apt.releases.hashicorp.com jammy main'
Description:
Archive for codename: jammy components: main
More info: https://apt.releases.hashicorp.com
Adding repository.
Press [ENTER] to continue or Ctrl-c to cancel.
Adding deb entry to /etc/apt/sources.list.d/archive_uri-https_apt_releases_hashicorp_com-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/archive_uri-https_apt_releases_hashicorp_com-jammy.list
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Get:2 https://apt.releases.hashicorp.com jammy InRelease [12,9 kB]
Hit:3 http://ec.archive.ubuntu.com/ubuntu jammy InRelease
Get:4 https://apt.releases.hashicorp.com jammy/main amd64 Packages [136 kB]
Hit:5 http://ec.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:6 http://ec.archive.ubuntu.com/ubuntu jammy-backports InRelease
Fetched 149 kB in 2s (93,5 kB/s)
Reading package lists... Done
W: https://apt.releases.hashicorp.com/dists/jammy/InRelease: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEPRECATION section in apt-key(8) for details.
lizabeth@lizabeth-virtual-machine:~$
```

*Nota.* El gráfico ilustra la adición del repositorio oficial de HashiCorp a la lista de fuentes de APT.

**Instalación Terraform.** Una vez que se ha agregado el repositorio oficial de HashiCorp y se ha actualizada la lista de paquetes, el siguiente paso es proceder con la instalación de Terraform como se muestra en la Figura 7.

**Figura 7**  
*Instalación de terraform*

```
lizbeth@lizbeth-virtual-machine:~$ sudo apt install terraform -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git git-man liberror-perl
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn
The following NEW packages will be installed:
  git git-man liberror-perl terraform
0 upgraded, 4 newly installed, 0 to remove and 426 not upgraded.
Need to get 31,9 MB of archives.
After this operation, 109 MB of additional disk space will be used.
Get:1 https://apt.releases.hashicorp.com jammy/main amd64 terraform amd64 1.8.5-1 [27,7 MB]
3% [Waiting for headers] [1 terraform 1.376 kB/27,7 MB 5%]
```

*Nota.* El gráfico ilustra la instalación de Terraform junto con las dependencias adicionales necesarias.

**Verificación de instalación.** Después de completar la instalación de Terraform, es esencial verificar que la instalación se haya realizado correctamente y que la herramienta esté lista para su uso. Para ello, se utilizará el siguiente comando, como se muestra en la Figura 8.

**Figura 8**  
*Verificación de instalación*

```
lizbeth@lizbeth-virtual-machine:~$ terraform version
Terraform v1.8.5
on linux_amd64
```

*Nota.* El gráfico ilustra la verificación de que la instalación de Terraform se ha realizado correctamente.

#### 4.1.2 Instalación de Ansible

A continuación, se describen los pasos para la instalación de Ansible en el sistema operativo basado en Linux.

**Actualización del sistema e instalación de dependencias.** Antes de instalar Ansible, es importante asegurarse de que el sistema esté actualizado y que las dependencias necesarias estén instaladas, como se muestra en la Figura 9.

**Figura 9**  
*Instalación de dependencias*

```
lizabeth@lizabeth-virtual-machine:~$ sudo apt install software-properties-common -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
software-properties-common is already the newest version (0.99.22.9).
0 upgraded, 0 newly installed, 0 to remove and 3 not upgraded.
```

*Nota.* El gráfico ilustra la actualización del sistema e instalación de paquetes.

**Agregación del repositorio de Ansible.** El siguiente paso se agrega el repositorio oficial de Ansible a la lista de fuentes de APT en el sistema, como se muestra en la Figura 10.

**Figura 10**  
*Agregación del repositorio de Ansible*

```
lizabeth@lizabeth-virtual-machine:~$ sudo add-apt-repository --yes --update ppa:ansible/ansible
Repository: 'deb https://ppa.launchpadcontent.net/ansible/ansible/ubuntu/ jammy main'
Description:
Ansible is a radically simple IT automation platform that makes your applications and systems easier to deploy. Avoid writing scripts or custom code to deploy and update your applications— automate in a language that approaches plain English, using SSH, with no agents to install on remote systems.

http://ansible.com/

If you face any issues while installing Ansible PPA, file an issue here:
https://github.com/ansible-community/ppa/issues
More info: https://launchpad.net/~ansible/+archive/ubuntu/ansible
Adding repository.
Adding deb entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding disabled deb-src entry to /etc/apt/sources.list.d/ansible-ubuntu-ansible-jammy.list
Adding key to /etc/apt/trusted.gpg.d/ansible-ubuntu-ansible.gpg with fingerprint
6125E2A8C77F2818FB7BD15B93C4A3FD7BB9C367
Hit:1 http://ec.archive.ubuntu.com/ubuntu jammy InRelease
Hit:2 http://ec.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:3 http://security.ubuntu.com/ubuntu jammy-security InRelease
```

*Nota.* El gráfico ilustra la adición del repositorio de Ansible a las fuentes de APT, permitiendo al sistema descargar e instalar Ansible desde su repositorio oficial.

**Instalación de Ansible.** Una vez que el repositorio ha sido agregado, se procede a instalar Ansible, como se muestra en la Figura 11.

**Figura 11**  
Instalación de Ansible

```
lizbeth@lizbeth-virtual-machine:~$ sudo apt install ansible -y
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  ansible-core python-babel-localedata python3-babel python3-jinja2
  python3-jmespath python3-kerberos python3-ntlm-auth python3-packaging
  python3-requests-kerberos python3-requests-ntlm python3-resolveib
  python3-winrm python3-xmltodict sshpass
Suggested packages:
  python-jinja2-doc
The following NEW packages will be installed:
  ansible-core python-babel-localedata python3-babel python3-jinja2
  python3-jmespath python3-kerberos python3-ntlm-auth python3-packaging
  python3-requests-kerberos python3-requests-ntlm python3-resolveib
  python3-winrm python3-xmltodict sshpass
0 upgraded, 15 newly installed, 0 to remove and 3 not upgraded.
Need to get 23,2 MB of archives.
After this operation, 226 MB of additional disk space will be used.
Get:1 http://ec.archive.ubuntu.com/ubuntu jammy/main amd64 python-babel-localedata all 2.8.0+dfsg.1-7 [4.982 kB]
Get:2 https://ppa.launchpadcontent.net/ansible/ansible/ubuntu jammy/main amd64 ansible-core all 2.16.7-1ppa-jammy [1.031 kB]
0% [1 python-babel-localedata 14,3 kB/4.982 kB 0%] [2 ansible-core 16,1 kB/1.031 kB 0%]
```

*Nota.* El gráfico ilustra la instalación de Ansible desde el repositorio agregado anteriormente, asegurando que se obtenga la última versión disponible.

**Verificación de instalación.** Finalmente, es esencial comprobar que Ansible se ha instalado correctamente y esté listo para su usar. Para verificar la instalación, se utiliza el siguiente comando, como se muestra en la Figura 12.

**Figura 12**  
Verificación de instalación

```
lizbeth@lizbeth-virtual-machine:~$ ansible --version
ansible [core 2.16.7]
  config file = /etc/ansible/ansible.cfg
  configured module search path = ['/home/lizbeth/.ansible/plugins/modules', '/usr/share/ansible/plugins/modules']
  ansible python module location = /usr/lib/python3/dist-packages/ansible
  ansible collection location = /home/lizbeth/.ansible/collections:/usr/share/ansible/collections
  executable location = /usr/bin/ansible
  python version = 3.10.12 (main, Nov 20 2023, 15:14:05) [GCC 11.4.0] (/usr/bin/python3)
  jinja version = 3.0.3
  libyaml = True
```

*Nota.* En el gráfico se puede comprobar que Ansible se ha instalado correctamente, mostrando la versión instalada y otros detalles relevantes.

## 4.2 Desarrollo de Scripts y Plantillas

Esta sección discute el desarrollo de scripts y plantillas que son esenciales para automatizar la infraestructura de los laboratorios del área de redes y

telecomunicaciones. La creación de estos scripts y plantillas se basa en dos herramientas, como Terraform y Ansible.

El proceso comenzará creando scripts de Terraform para aprovisionar máquinas virtuales en Proxmox. Estos scripts asegurarán que los recursos estén configurados de manera coherente y repetible, facilitando la gestión y escalabilidad de la infraestructura. Después, se desarrollarán plantillas de configuración utilizando Ansible, centrándose en automatizar tareas como la instalación de paquetes, servicios de red y configuraciones de dispositivos, así como otros componentes vitales.

#### ***4.2.1 Creación de scripts con Terraform para Proxmox***

Esta sección describe cómo crear scripts con Terraform para aprovisionar recursos en la plataforma de virtualización Proxmox. A continuación, se describe cada paso del proceso.

**Código de Variables.** Para comenzar, se crea un archivo que define las variables requeridas para configurar Terraform en la plataforma de virtualización Proxmox.

**Definición de Variables.** Se define las variables que terraform utilizará para la configuración y gestión de los recursos en Proxmox, con la finalidad reutilizar la información específica del host de Proxmox en un entorno de configuración automatizada. En este caso, se define la URL de la API de Proxmox, el usuario y el nodo de destino donde se desplegarán las máquinas virtuales como se observa en la Figura 13.

**Figura 13***Definición de variables de Proxmox*

```
variable "proxmox_host" {
  type = map
  default = {
    pm_api_url = "https://192.168.183.132:8006/api2/json"
    pm_user = "root@pam"
    target_node = "pve"
  }
}
```

*Nota.* En el gráfico se muestra la definición de la variable proxmox\_host, la cual consta de 3 parámetros esenciales como son el url de la API de Proxmox, el usuario que se utiliza para autenticar la conexión con Proxmox y el nodo de destino en el clúster de Proxmox donde se desplegaran las máquinas virtuales.

***Token API de Proxmox.*** Se define el token API para autenticar las solicitudes a Proxmox sin necesidad de ingresar la contraseña repetidamente como se muestra en la Figura 14.

**Figura 14***Definición de Token API de Proxmox*

```
variable "api_token_secret" {
  description = "The secret for the API token"
  type = string
  sensitive = true
}
```

*Nota.* En el gráfico muestra la definición de la variable que se utiliza para almacenar el token secreto de API.

***Identificador de Máquina Virtual.*** En la figura 15 se observa que se establece el identificador inicial para las máquinas virtuales, como el nombre de la máquina virtual que se creara.

**Figura 15***Identificación de la MV que se creara*

```

variable "vmid" {
  default = 500
  description = "Starting ID for the CTs"
}

variable "hostnames" {
  description = "VMs to be created"
  type = list(string)
  default = ["prod-vm1"]
}

```

*Nota.* En el gráfico muestra la definición de dos variables que se utiliza para gestionar identificadores y nombres de máquinas virtuales.

**Tamaño del Sistema de Archivos Raíz.** Se define el tamaño del sistema de archivos raíz para las máquinas virtuales como se muestra en la figura 16.

**Figura 16***Tamaño del sistema de archivos raíz*

```

variable "rootfs_size" {
  default = "2G"
}

```

*Nota.* En el gráfico muestra la definición de la variable que se utiliza para especificar el tamaño del sistema de archivos raíz para una máquina virtual.

**Direcciones IP de las Máquinas Virtuales.** Se establece las direcciones IP asignadas a las máquinas virtuales como se muestra en la figura 17.

**Figura 17***Definición de direcciones IP para MV*

```

variable "ips" {
  description = "IPs of the VMs, respective to the hostname order"
  type = list(string)
  default = ["192.168.183.190"]
}

```

*Nota.* En el gráfico muestra la definición de la variable que se utiliza para especificar una lista de direcciones IP para las máquinas virtuales.

**Clave SSH.** Se define las rutas a las claves pública y privada de SSH utilizadas para la autenticación como se muestra en la Figura 18.

**Figura 18**  
Definición de clave SSH

```
variable "ssh_keys" {
  type = map
  default = {
    pub = "~/.ssh/id_rsa.pub"
    priv = "~/.ssh/id_rsa"
  }
}
```

*Nota.* En el gráfico se muestra la definición de la variable ssh keys que se utiliza para especificar las rutas de las claves SSH públicas y privadas necesarias para la autenticación.

**Configuración de Proxmox en Terraform.** La segunda parte del script es la configuración del proveedor de Proxmox en terraform, con la finalidad de comunicarse con la API de Proxmox para gestionar los recursos.

**Configuración del Proveedor.** Se define un proveedor de recursos llamado "proxmox" como se muestra en la Figura 19. En Terraform, los proveedores son plugins que interactúan con APIs externas o servicios para gestionar recursos.

**Figura 19**  
Configuración del proveedor Proxmox

```
provider "proxmox" {
  pm_api_url = var.proxmox_host["pm_api_url"]
  pm_user    = var.proxmox_host["pm_user"]
  pm_password = var.ssh_password
  pm_tls_insecure = true
}
```

*Nota.* En el gráfico se puede observar la configuración del proveedor donde se especifica los parámetros necesarios para conectar y autenticar Terraform con el servidor Proxmox.

**Creación de Recursos en Proxmox.** En este apartado como se muestra en la Figura 20, se detallará la creación y configuración de las máquinas virtuales utilizando Terraform. Esto incluye definir parámetros como la cantidad de memoria, CPU, almacenamiento, y configuraciones de red.

**Figura 20**

Creación de recursos de las Máquinas Virtuales

```
resource "proxmox_vm_qemu" "prod-vm" {
  count = length(var.hostnames)
  name = var.hostnames[count.index]
  target_node = var.proxmox_host["target_node"]
  vmid = var.vmid + count.index
  full_clone = true
  clone = "SRV-Lizbeth"

  cores = 6
  sockets = 1
  vcpus = 1
  memory = 8192
  balloon = 8192
  boot = "c"
  bootdisk = "virtio0"

  scsihw = "virtio-scsi-single"

  onboot = false
  agent = 1
  cpu = "kvm64"
  numa = true
  hotplug = "network,disk,cpu,memory"

  network {
    bridge = "vbr0"
    model = "virtio"
  }
}
```

*Nota.* En el gráfico muestra la configuración de recursos de la máquina virtual en Proxmox. En el cual se define características y para metros necesarios para crear y gestionar una máquina virtual.

**Configuración de Recursos.** En la Figura 21 se puede observar el siguiente bloque de código que se especifica detalles de red, disco y conexión, así como las credenciales iniciales para la máquina virtual.

**Figura 21**

Configuración de máquina virtual

```
ipconfig0 = "ip=${var.ips[count.index]}/24,gw=${cidrhost(format("%s/24", var.ips[count.index]), 1)}"

disk {
  type = "virtio"
  storage = "local-lvm"
  size = "200G"
}

os_type = "cloud-init"
ciuser = var.user
cipassword = var.ssh_password

connection {
  host = var.ips[count.index]
  user = var.user
  private_key = file(var.ssh_keys["priv"])
  agent = false
  timeout = "3m"
}
```

*Nota.* En el gráfico muestra la configuración de la máquina virtual en la cual definen parámetros de red, disco, credenciales iniciales y la conexión SSH.

**Proceso de provisionamiento en Terraform y Ansible.** En la Figura 22 se puede observar el siguiente bloque que se utiliza aprovisionadores para configurar la máquina

virtual en Proxmox. Los cuales ejecutan comandos tanto de forma remota como local para completar la configuración de la máquina virtual utilizando Ansible.

**Figura 22**

*Proceso de provisionamiento*

```

provisioner "remote-exec" {
  inline = ["echo 'Cool, we are ready for provisioning'"]
}

provisioner "local-exec" {
  working_dir = "../ansible/"
  command = "ansible-playbook -u ${var.user} --key-file ${var.ssh_keys["priv"]} -i ${var.ips[count.index]}, provision.yaml"
}

provisioner "local-exec" {
  working_dir = "../ansible/"
  command = "ansible-playbook -u ${var.user} --key-file ${var.ssh_keys["priv"]} -i ${var.ips[count.index]}, install-qemu-guest-agent.yaml"
}

```

*Nota.* En el gráfico muestra la configuración de provisionadores en Terraform para la automatización de la configuración de máquinas virtuales en Proxmox utilizando Ansible.

Este esquema de variables y configuración del proveedor facilita la gestión automatizada de los recursos en Proxmox, permitiendo un despliegue eficiente y coherente de la infraestructura.

#### **4.2.2 Desarrollo de plantillas de configuración con Ansible**

Ansible permite la configuración, gestión y despliegue de máquinas virtuales. A continuación, se describen los archivos de playbooks de Ansible.

**Instalación del Qemu Guest Agent.** Se utiliza para instalar el agente de invitados de Qemu en las máquinas virtuales como se muestra en la Figura 23. El agente de invitados permite una mejor interacción y gestión de la máquina virtual desde el hipervisor Proxmox.

**Figura 23**  
Configuración de instalación Qemu Guest Agent

```

GNU nano 6.2                                install-qemu-guest-agent.yaml
--
- hosts: all
  gather_facts: false
  become: yes

  tasks:
    - name: update apt cache
      command: apt update

    - name: add universe repo
      command: add-apt-repository universe

    - name: update apt cache
      command: apt update

    - name: install qemu guest agent
      apt:
        name: qemu-guest-agent
        state: present

    - name: upgrade all packages
      command: apt upgrade -y

    - name: reboot host
      reboot:
  
```

*Nota.* En el gráfico muestra la configuración yaml utilizado por Ansible para automatizar la instalación del Qemu Guest Agent en las máquinas virtuales.

**Provisión.** En la Figura 24 se puede observar el playbook que se utiliza para realizar configuraciones iniciales en las máquinas virtuales, como la instalación de paquetes y la configuración de servicios.

**Figura 24**  
Configuración de máquina virtual

```

GNU nano 6.2                                provision.yaml
--
- hosts: all
  become: yes
  tasks:
    - name: Update and upgrade apt packages
      become: yes
      ignore_errors: yes
      failed_when: "'FAILED' in command_result.stderr"
      apt:
        upgrade: yes
        update_cache: yes
        cache_valid_time: 86400 #One day

    - name: Make sure we have a 'wheel' group
      group:
        name: wheel
        state: present

    - name: Allow 'wheel' group to have passwordless sudo
      lineinfile:
        dest: /etc/sudoers
        state: present
        regexp: '%wheel'
        line: '%wheel ALL=(ALL) NOPASSWD: ALL'
        validate: 'visudo -cf %s'

    - name: Add sudoers users to wheel group
      user:
        name: notroot
        groups: wheel
        append: yes
        state: present
        createhome: yes
        shell: /bin/bash

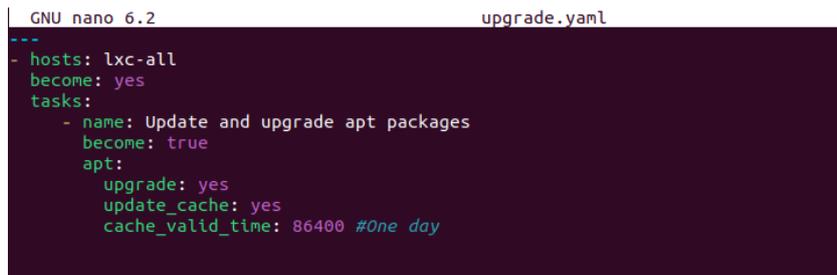
    - name: Create necessary folders
      file:
        path: "{{ item }}"
        recurse: yes
  
```

*Nota.* En el gráfico muestra la configuración yaml utilizado para automatizar las tareas de configuración en máquinas virtuales.

**Actualización.** En la Figura 25 se puede observar el playbook que se utiliza para actualizar el sistema operativo y los paquetes instalados a sus últimas versiones.

**Figura 25**

*Actualización del sistema operativo de la máquina virtual*



```
GNU nano 6.2 upgrade.yaml
--
- hosts: lxc-all
  become: yes
  tasks:
    - name: Update and upgrade apt packages
      become: true
      apt:
        upgrade: yes
        update_cache: yes
        cache_valid_time: 86400 #One day
```

*Nota.* En el gráfico muestra la configuración yaml que permite la automatización de la actualización de las máquinas virtuales.

### 4.3 Pruebas de Automatización

Esta sección ejecutará pruebas de automatización para garantizar una infraestructura robusta y eficiente utilizando herramientas de automatización para minimizar la intervención manual y asegurar una configuración precisa y rápida de los entornos de red. La integración entre Terraform y Ansible proporciona un enfoque integral para la gestión de la infraestructura, mejorando la eficiencia operativa mientras permite una mayor flexibilidad en la administración de recursos.

Los scripts serán validados como parte de estas pruebas a través de un proceso de verificación para identificar y rectificar cualquier posible inconsistencia en la configuración. Una vez validados, se implementarán pruebas automatizadas que garanticen tanto la validez como la funcionalidad de los scripts y plantillas desarrollados, evaluando su rendimiento bajo diferentes escenarios. Este proceso

confirma que los cambios en la infraestructura no introducen errores, asegurando un despliegue fiable y consistente de recursos.

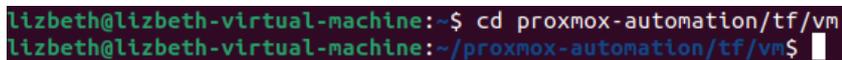
#### ***4.3.1 Proceso de validación de scripts***

Después de que se han desarrollado los scripts de infraestructura, se pueden desplegar para crear y configurar máquinas virtuales. Este despliegue se realiza desde una máquina que tiene herramientas de automatización para IaC como Terraform y Ansible instaladas.

Para desplegar la infraestructura, desde la terminal de la máquina con Terraform y Ansible instalados, se accede al directorio donde se encuentran los scripts. Esto se hace usando el comando **'cd'** para cambiar de directorio como se muestra en la Figura 26.

**Figura 26**

*Directorio de scripts de infraestructura como código*



```
lizbeth@lizbeth-virtual-machine:~$ cd proxmox-automation/tf/vm
lizbeth@lizbeth-virtual-machine:~/proxmox-automation/tf/vm$
```

*Nota.* En el gráfico muestra el acceso al directorio desde el terminal de la máquina, donde se encuentran alojados los scripts de infraestructura como código, necesarios para desplegar y configurar máquinas virtuales usando Terraform y Ansible.

Una vez que se encuentre en el directorio se procede a ejecutar **'terraform init'** para inicializar el entorno de trabajo de Terraform como se muestra en la Figura 27. Este comando descarga los plugins necesarios y configura el entorno.

**Figura 27**  
Inicialización de Terraform

```
(kali@kali)-[~/proxmox-automation/tf/vm]
└─$ terraform init

Initializing the backend ...

Initializing provider plugins ...
- Reusing previous version of telmate/proxmox from the dependency lock file
- Using previously-installed telmate/proxmox v2.9.14

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

*Nota.* En el gráfico muestra la ejecución del comando `terraform init`, indicando que Terraform ha inicializado correctamente el backend y los plugins necesarios. Este paso permite preparar el entorno de Terraform y asegurar que todos los módulos y configuraciones de backend estén listos para su uso.

Después de escribir o modificar los scripts de Terraform, se ejecuta **'terraform validate'** para confirmar que la configuración está lista para ser aplicada como se muestra en la Figura 28. Este proceso de despliegue de infraestructura como código, garantizando que no se propaguen errores al entorno de producción. Dado que revisa los archivos de configuración en el directorio actual y confirma que no haya errores en el código antes de intentar aplicar cambios en la infraestructura.

**Figura 28**  
Verificación de la configuración de Terraform

```
(kali@kali)-[~/proxmox-automation/tf/vm]
└─$ terraform validate
Success! The configuration is valid.
```

*Nota.* En el gráfico muestra la validación de la configuración de Terraform, en el cual se ejecuta el comando `'terraform validate'` y el resultado indica que la configuración a sido validada por lo que no muestra errores de sintaxis en los archivos d configuración.

Una vez validado se procede a ejecutar el comando **'terraform plan'** en Terraform para visualizar los cambios que se van a realizar en la infraestructura antes de aplicarlos. En la Figura 29 se puede observar que al ejecutar `terraform plan`, Terraform

genera un plan de ejecución que detalla las acciones que se tomarán para alcanzar el estado deseado de la infraestructura.

**Figura 29**

*Plan de ejecución en Terraform*

```
(kali㉿kali)-[~/proxmox-automation/tf/vm]
└─$ terraform plan
var.api_token_secret
  The secret for the API token

  Enter a value:

var.ssh_password
  Password for SSH

  Enter a value:

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# proxmox_vm_qemu.prod-vm[0] will be created
+ resource "proxmox_vm_qemu" "prod-vm" {
  + additional_wait      = 5
  + agent                = 1
  + automatic_reboot    = true
  + balloon              = 12288
  + bios                 = "seabios"
  + boot                 = "c"
  + bootdisk             = "virtio0"
  + clone                = "srv-lizbeth"
  + clone_wait           = 10
}
```

*Nota.* En el gráfico detalla las acciones planificadas por Terraform para alcanzar el estado deseado de la infraestructura, en el cual se puede observar el estado actual y el estado esperado de la máquina virtual de esta manera se puede revisar y validar los cambios antes de aplicarlos.

### ***4.3.2 Pruebas de integración continua para garantizar la calidad***

Estas pruebas son fundamentales para asegurar que la infraestructura automatizada se despliega de manera efectiva y sin errores, proporcionando un entorno estable y escalable para las operaciones de red. Para mantener la calidad tanto de la infraestructura como de las configuraciones automatizadas, es imprescindible implementar pruebas de integración continua. En este contexto, se llevarán a cabo cuatro tests específicos como se muestra en la Tabla 13 con el objetivo de validar

continuamente los scripts y configuraciones en diversos escenarios, reduciendo al mínimo la posibilidad de errores y garantizando un despliegue confiable y eficiente.

**Tabla 13**

*Tabla de pruebas de integración continua de infraestructura como código*

<b>Test</b>	<b>Descripción</b>	<b>Resultados</b>
<b>Test 1. Prueba inicial de despliegue de infraestructura como código</b>	Verifica la correcta ejecución y funcionalidad del despliegue inicial de la infraestructura utilizando Terraform y Proxmox.	El despliegue de la infraestructura debe realizarse sin errores, y la máquina virtual debe clonarse y configurarse correctamente, con todos los recursos asignados según los scripts de Terraform.
<b>Test 2. Despliegue de infraestructura con recreación de recursos</b>	Evalúa la consistencia del despliegue tras realizar ajustes y optimizaciones en los scripts.	El despliegue debe destruir la máquina virtual existente y crear una nueva conforme a las configuraciones actualizadas, garantizando un despliegue exitoso y sin errores.
<b>Test 3. Despliegue de infraestructura sin afectar recursos existentes</b>	Asegura que las configuraciones se apliquen correctamente sin modificar ni destruir los recursos actuales.	El despliegue debe llevarse a cabo sin alterar o destruir los recursos existentes, creando una nueva máquina virtual con los cambios aplicados de manera coherente y sin generar nuevos errores.
<b>Test 4. Despliegue simultáneo de</b>	Evalúa la capacidad de los scripts para gestionar	Todas las máquinas virtuales deben desplegarse correctamente con las

---

<b>múltiples máquinas virtuales</b>	el despliegue simultáneo de varias máquinas virtuales, garantizando que las configuraciones se apliquen correctamente y verificando la escalabilidad del proceso.	configuraciones asignadas. El proceso debe ser eficiente y escalable, sin errores durante el despliegue simultáneo.
-------------------------------------	---	---

---

*Nota.* La tabla presenta los diferentes tests que se realizan durante el proceso de integración continua para asegurar la correcta automatización de la infraestructura utilizando Terraform y Proxmox.

### **Test 1.** Prueba inicial de despliegue de infraestructura como código

Este test tiene como objetivo verificar la correcta ejecución y funcionalidad del despliegue de la infraestructura utilizando Terraform y Proxmox. Se realiza un despliegue inicial de una máquina virtual para asegurar que todos los scripts y configuraciones han sido implementados correctamente.

A continuación, se describe el proceso para establecer para aplicar automatización de infraestructura para ello se ejecuta *“terraform apply”* como se puede ver en la Figura 30, el cual genera un plan de ejecución que detalla la creación de un recurso proxmox con diversas configuraciones, incluyendo memoria, CPU, disco, y parámetros específicos de Proxmox. Este proceso asegura que la infraestructura se despliegue de manera coherente y automatizada, siguiendo las definiciones establecidas en los scripts de Terraform.

**Figura 30***Ejecución de terraform apply*

```
(kali@kali)-[~/proxmox-automation/tf/vm]
└─$ terraform apply
var.api_token_secret
The secret for the API token

Enter a value:

var.ssh_password
Password for SSH

Enter a value:

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# proxmox_vm_qemu.prod-vm[0] will be created
+ resource "proxmox_vm_qemu" "prod-vm" {
+   additional_wait = 5
+   agent          = 1
+   automatic_reboot = true
+   balloon        = 12288
+   bios           = "seabios"
+   boot           = "c"
+   bootdisk       = "virtio0"
+   clone          = "SRV-Lizbeth"
}
```

*Nota.* En el gráfico detalla la ejecución del comando terraform apply que se utiliza para aplicar los cambios especificados en los scripts de Terraform, aprovisionando y configurando recursos en Proxmox.

En la Figura 31 se muestra la salida de Terraform cuando se está ejecutando el plan de creación de la máquina virtual en Proxmox. El mensaje de Terraform indica que ha generado un plan de ejecución donde se añadirá un recurso.

**Figura 31***Ejecución del plan de creación de la máquina virtual*

```
Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.

Enter a value: yes

proxmox_vm_qemu.prod-vm[0]: Creating ...
proxmox_vm_qemu.prod-vm[0]: Still creating... [10s elapsed]
proxmox_vm_qemu.prod-vm[0]: Still creating... [20s elapsed]
```

*Nota.* En el gráfico detalla la salida de Terraform cuando se está ejecutando un plan de creación de una máquina virtual en Proxmox.

Una vez aplicadas las configuraciones, se puede visualizar en la interfaz de Proxmox la ejecución de una tarea de clonación de la máquina virtual identificada como VM 121 en el nodo pv5 como se muestra en la Figura 32.

**Figura 32**  
*Clonación de MV*



```
Task viewer: VM 121 - Clonar
Salida Estado
Parar
create full clone of drive ide1 (local-lvm:vm-121-cloudinit)
Logical volume "vm-400-cloudinit" created.
create full clone of drive scsi0 (local-lvm:vm-121-disk-2)
Logical volume "vm-400-disk-0" created.
transferred 0.0 B of 200.0 GiB (0.00%)
transferred 2.0 GiB of 200.0 GiB (1.00%)
transferred 4.0 GiB of 200.0 GiB (2.00%)
transferred 6.0 GiB of 200.0 GiB (3.00%)
transferred 8.0 GiB of 200.0 GiB (4.00%)
transferred 10.0 GiB of 200.0 GiB (5.00%)
transferred 12.0 GiB of 200.0 GiB (6.01%)
transferred 14.0 GiB of 200.0 GiB (7.01%)
transferred 16.0 GiB of 200.0 GiB (8.01%)
transferred 18.0 GiB of 200.0 GiB (9.01%)
transferred 20.0 GiB of 200.0 GiB (10.01%)
```

*Nota.* En el gráfico se puede observar la interfaz de Proxmox que proporciona detalles específicos sobre el proceso de clonación de la máquina virtual.

Finalmente se puede observar en la Figura 33 el registro de tareas para el monitoreo y la gestión de actividades en Proxmox. La cual permite a los administradores verificar que las acciones se han completado exitosamente, o por lo contrario ayuda a la identificación de cualquier problema que pueda surgir durante las operaciones de administración de la infraestructura virtual.

**Figura 33**  
*Registro de tareas de automatización de infraestructura*

Oct 15 14:55:19		pv5	root@pam	VM/CT 400 - Consola	
Oct 15 14:50:21	Oct 15 14:50:22	pv5	root@pam	VM 400 - Configurar	OK
Oct 15 14:44:44	Oct 15 14:44:45	pv5	root@pam	VM 400 - Iniciar	OK
Oct 15 14:44:20	Oct 15 14:44:20	pv5	root@pam	VM 400 - Configurar	OK
Oct 15 14:36:56	Oct 15 14:43:58	pv5	root@pam	VM 121 - Clonar	OK

*Nota.* En el gráfico se muestra la lista de tareas de la automatización de infraestructura en este caso se puede observar que primero se clona la máquina virtual que se utiliza como plantilla, para posteriormente configurar e iniciar la nueva máquina virtual.

## Test 2. Despliegue de infraestructura con recreación de recursos

En este segundo test, se realizará un despliegue adicional de infraestructura con el objetivo de verificar que las configuraciones actualizadas se apliquen de manera correcta, manteniendo la consistencia en todo el proceso. Se procederá a destruir la máquina virtual previamente creada y a reemplazarla con una nueva, basándose en los ajustes y optimizaciones realizados después del primer despliegue.

Para ello se procede a ejecutar el comando **“terraform apply”** para aplicar los cambios y realizar el despliegue, como se ilustra en la Figura 34.

**Figura 34**

*Ejecución de terraform apply test 2*

```
(kali@kali)-[~/proxmox-automation/tf/vm]
└─$ terraform apply
var.api_token_secret
  The secret for the API token

  Enter a value:

var.ssh_password
  Password for SSH

  Enter a value:

proxmox_vm_qemu.prod-vm[0]: Refreshing state ... [id=pv5/qemu/400]

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
-/+ destroy and then create replacement

Terraform will perform the following actions:

# proxmox_vm_qemu.prod-vm[0] is tainted, so must be replaced
-/+ resource "proxmox_vm_qemu" "prod-vm" {
  - args                = "-cpu host" → null
  - default_ipv4_address = "10.0.0.10" → (known after apply)
  - id                  = "pv5/qemu/400" → (known after apply)
    name                 = "prob-vm"
  + nameserver          = (known after apply)
  - qemu_os              = "l26" → null
  - reboot_required     = false → (known after apply)
  + searchdomain        = (known after apply)
}
```

*Nota.* En el gráfico detalla la ejecución del comando terraform apply para realizar el test 2 en este caso se va a destruir la máquina virtual existente y se desplegará una nueva.

En la Figura 35 se observa el proceso en el cual Terraform ejecuta el plan de acción que implica la destrucción de la máquina virtual existente y la creación de una nueva con las configuraciones actualizadas.

**Figura 35**

*Proceso de destrucción y creación de máquina virtual con Terraform*

```
Plan: 1 to add, 0 to change, 1 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

proxmox_vm_qemu.prod-vm[0]: Destroying ... [id=pv5/qemu/400]
proxmox_vm_qemu.prod-vm[0]: Destruction complete after 4s
proxmox_vm_qemu.prod-vm[0]: Creating ...
```

*Nota.* En el gráfico detalla el plan de ejecución que se llevara a cabo en el cual inicia con la destrucción de la máquina virtual, y posteriormente inicia la creación de una nueva máquina permite.

Finalmente, en la Figura 36 se presenta el registro detallado de tareas para el monitoreo y la gestión de actividades en Proxmox. Este registro permite observar el seguimiento de las acciones realizadas sobre la VM 400, comenzando con la destrucción de la máquina virtual anterior. Posteriormente, se procede a la clonación de una máquina virtual utilizada como plantilla. Una vez finalizada la clonación, se crea una nueva instancia de máquina virtual, seguida de la configuración necesaria para su puesta en marcha. Finalmente, se registra el inicio de la VM configurada, completando el ciclo de gestión automatizada de recursos virtuales.

**Figura 36**

*Registro de acciones de gestión de máquinas virtuales en Proxmox*

Hora de inicio ↓	Hora final	Nodo	Nombre de Usuario	Descripción	Estado
Oct 22 16:11:13		pv5	root@pam	VM/CT 400 - Consola	
Oct 22 16:10:27	Oct 22 16:10:28	pv5	root@pam	VM 400 - Iniciar	OK
Oct 22 16:10:02	Oct 22 16:10:02	pv5	root@pam	VM 400 - Configurar	OK
Oct 22 16:02:16	Oct 22 16:09:40	pv5	root@pam	VM 121 - Clonar	OK
Oct 22 16:02:14	Oct 22 16:02:15	pv5	root@pam	VM 400 - Destruir	OK
Oct 22 16:02:12	Oct 22 16:02:13	pv5	root@pam	VM 400 - Parar	OK

*Nota.* El gráfico muestra el registro de tareas utilizadas para el monitoreo y la gestión de la infraestructura en Proxmox, destacando las acciones realizadas sobre las máquinas virtuales de manera detallada y secuencial.

De igual manera en la Figura 37 se presenta el proceso de creación y destrucción de máquinas virtuales en Proxmox, gestionado mediante Terraform. El aprovisionamiento se llevó a cabo utilizando playbooks de Ansible para automatizar la configuración de las máquinas virtuales. Las operaciones fueron completadas de manera exitosa, como se evidencia en el mensaje final que confirma que un recurso fue añadido y otro destruido.

**Figura 37**

*Proceso de creación y destrucción de máquinas virtuales en Proxmox utilizando Terraform*

```

proxmox_vm_qemu.prod-vm[0]: Provisioning with 'local-exec' ...
proxmox_vm_qemu.prod-vm[0] (local-exec): Executing: ["/bin/sh" "-c" "ansible-
playbook -u srv-lizabeth --key-file ~/.ssh/terraform_key -i 10.0.0.160, upgrad
e.yaml"]
proxmox_vm_qemu.prod-vm[0] (local-exec): [WARNING]: Could not match supplied
host pattern, ignoring: lxc-all

proxmox_vm_qemu.prod-vm[0] (local-exec): PLAY [lxc-all] *****
*****
proxmox_vm_qemu.prod-vm[0] (local-exec): skipping: no hosts matched

proxmox_vm_qemu.prod-vm[0] (local-exec): PLAY RECAP *****
*****

proxmox_vm_qemu.prod-vm[0]: Creation complete after 12m54s [id=pv5/qemu/600]
Apply complete! Resources: 1 added, 0 changed, 1 destroyed.

```

*Nota.* El gráfico muestra la exitosa gestión de la infraestructura mediante Terraform, complementada por el aprovisionamiento automatizado con Ansible. Ambas herramientas se integraron eficazmente para realizar las tareas de creación y destrucción de recursos, completándose el proceso sin errores.

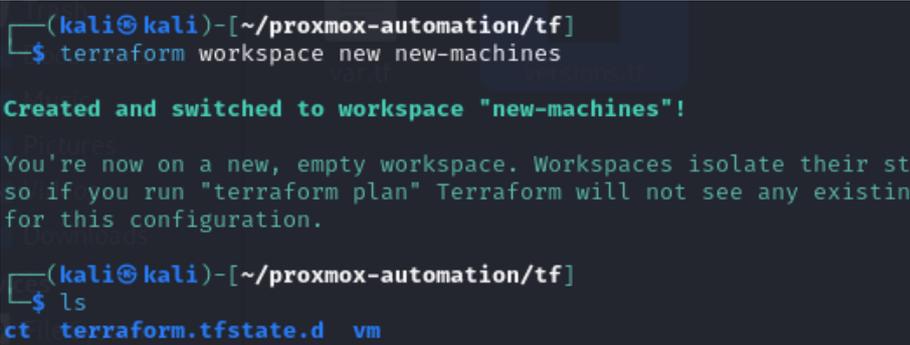
### **Test 3.** Despliegue de infraestructura sin afectar recursos existentes

Este test garantiza que el proceso de creación de una nueva máquina virtual no afecte a los recursos existentes. Para ello, es fundamental verificar que las configuraciones actualizadas no introduzcan nuevos problemas o inconsistencias en el despliegue. El despliegue debe realizarse de manera que no altere ni destruya los recursos ya implementados, asegurando que la nueva máquina virtual se cree con los cambios aplicados de forma coherente y sin generar errores adicionales.

Para lograr esto, se utiliza los workspaces en Terraform, los cuales permiten crear entornos independientes dentro del mismo proyecto. Este enfoque asegura que las nuevas máquinas virtuales puedan gestionarse sin afectar ni destruir los recursos existentes. Para crear un nuevo workspace, se ejecuta el siguiente comando “**terraform workspace new new-machines**”, como se ilustra en la Figura 38.

**Figura 38**

*Creación de un nuevo workspace en Terraform*



```
(kali㉿kali)-[~/proxmox-automation/tf]
└─$ terraform workspace new new-machines

Created and switched to workspace "new-machines"!

You're now on a new, empty workspace. Workspaces isolate their state,
so if you run "terraform plan" Terraform will not see any existing
configuration for this configuration.

(kali㉿kali)-[~/proxmox-automation/tf]
└─$ ls
ct  terraform.tfstate.d  vm
```

*Nota.* El gráfico muestra el proceso de creación y activación de un nuevo workspace en terraform, para el despliegue de infraestructura de manera separada, asegurando que los recursos existentes no se vean afectados.

En el nuevo workspace, se agrega los scripts necesarios para la gestión de la infraestructura en Proxmox. Es importante garantizar que el ID, nombre y dirección IP de la nueva máquina virtual sea diferente a los de las máquinas existentes, evitando conflictos en la red y en la administración de recursos. Una vez realizados estos ajustes, se procede a aplicar los cambios siguiendo el mismo proceso utilizado anteriormente. Esto se logra ejecutando el comando “**terraform apply**”, como se ilustra en la Figura 39.

**Figura 39***Ejecución de terraform apply*

```
(kali@kali)~[~/proxmox-automation/tf/terraform.tfstate.d/new-machines]
└─$ terraform apply
var.api_token_secret
  The secret for the API token

  Enter a value:

var.ssh_password
  Password for SSH

  Enter a value:

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# proxmox_vm_qemu.prod-vm[0] will be created
+ resource "proxmox_vm_qemu" "prod-vm" {
+   additional_wait = 5
+   agent           = 1
+   automatic_reboot = true
+   balloon        = 12288
+   bios           = "seabios"
+   boot           = "c"
}
```

*Nota.* En el gráfico se detalla la ejecución del comando terraform apply, el cual se utiliza para implementar los cambios especificados en los scripts de Terraform, aprovisionando y configurando recursos en Proxmox. De esta manera, se asegura que cualquier ajuste en la configuración se aplique de forma coherente y sin afectar a los recursos que se desea mantener intactos.

En la Figura 40 se puede observar el plan que se llevara a cabo en este caso indica que la acción a realizar es la creación de una nueva máquina virtual. Este plan confirma que no habrá cambios ni destrucciones de otros recursos, y que solo se agregará la nueva máquina de acuerdo con las configuraciones establecidas en los scripts de Terraform.

**Figura 40***Ejecución del plan de Terraform para agregar una máquina virtual*

```
Plan: 1 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

proxmox_vm_qemu.prod-vm[0]: Creating ...
proxmox_vm_qemu.prod-vm[0]: Still creating... [10s elapsed]
```

*Nota.* La figura muestra el resultado de la ejecución de terraform apply en la que se detalla el proceso de creación de una máquina virtual en Proxmox. El plan indica que solo se realizará una acción de adición sin modificar ni destruir otros recursos, asegurando que el entorno actual se mantenga sin interrupciones.

Finalmente, en la Figura 41 se puede observar el registro de tareas correspondientes a las actividades ejecutadas en Proxmox. Este registro permite verificar que el proceso se completó con éxito, mostrando el estado "OK" para cada acción realizada, lo que confirma que la clonación, configuración e inicio de la máquina virtual fueron satisfactorios.

#### **Figura 41**

*Registro de tareas de automatización de infraestructura*

Hora de inicio ↓	Hora final	Nodo	Nombre de Usuario	Descripción	Estado
Oct 15 16:27:49	Oct 15 16:27:50	pv5	root@pam	VM 400 - Iniciar	OK
Oct 15 16:27:26	Oct 15 16:27:26	pv5	root@pam	VM 400 - Configurar	OK
Oct 15 16:20:04	Oct 15 16:27:04	pv5	root@pam	VM 121 - Clonar	OK

*Nota.* En el gráfico se muestra la lista de tareas de la automatización de infraestructura en este caso se puede observar que primero se clona la máquina virtual que se utiliza como plantilla, para posteriormente configurar e iniciar la nueva máquina virtual.

#### **Test 4.** Despliegue simultáneo de múltiples máquinas virtuales

Este test tiene como fin evaluar la capacidad de los scripts y configuraciones para manejar el despliegue simultáneo de varias máquinas virtuales. Se comprueba que todas las máquinas virtuales se despliegan correctamente, con los recursos y configuraciones asignados de acuerdo con lo especificado. Además, se verifica la eficiencia y escalabilidad del proceso de automatización en un entorno de mayor carga.

Para realizar el despliegue, es necesario definir variables para el nombre, ID y direcciones IP que se asignarán a las máquinas virtuales en función del número de instancias que se planean. Estas variables aseguran que cada máquina tenga configuraciones únicas y personalizadas. La Figura 42 muestra cómo se han declarado

estas variables en el código de Terraform, lo que permite un control dinámico y automatizado del despliegue.

**Figura 42**

*Declaración de variables para la configuración de máquinas virtuales en Terraform*

```
variable "vmid" {
  default = 500
  description = "Starting ID for the CTs"
}

variable "hostnames" {
  description = "VMs to be created"
  type = list(string)
  default = ["test1-vm", "test2-vm", "test2-vm"]
}

variable "rootfs_size" {
  default = "2G"
}

variable "ips" {
  description = "IPs of the VMs, respective to the hostname order"
  type = list(string)
  default = ["10.0.0.140", "10.0.0.150", "10.0.0.155"]
}
```

*Nota.* En el gráfico se muestra la configuración de las variables vmid, hostnames y ips, necesarias para el despliegue automatizado de máquinas virtuales en Proxmox mediante Terraform.

Una vez configuradas estas variables, se procede a ejecutar el comando **"terraform apply"** para iniciar el despliegue simultáneo de máquinas virtuales. Este proceso aplica las configuraciones previamente definidas y crea las instancias de acuerdo con los parámetros especificados, como se muestra en la Figura 43.

**Figura 43**

*Ejecución del despliegue de máquinas virtuales con Terraform*

```
(kali@kali)-[~/proxmox-automation/tf/vm]
└─$ terraform apply
var.api_token_secret
  The secret for the API token

  Enter a value:

var.ssh_password
  Password for SSH

  Enter a value:

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# proxmox_vm_qemu.prod-vm[0] will be created
+ resource "proxmox_vm_qemu" "prod-vm" {
  + additional_wait = 5
  + agent          = 1
  + automatic_reboot = true
  + balloon       = 12288
  + bios          = "seabios"
}
```

*Nota.* La figura muestra la ejecución del comando **terraform apply**, que inicia el proceso de despliegue de las máquinas virtuales en Proxmox.

En la Figura 44 se muestra el plan de ejecución, el cual detalla las acciones que se realizarán. En este caso, se contempla la adición de tres nuevas máquinas virtuales, siguiendo las configuraciones definidas previamente en el código de infraestructura.

**Figura 44**

*Plan de ejecución de Terraform para el despliegue de máquinas virtuales*

```
Plan: 3 to add, 0 to change, 0 to destroy.

Do you want to perform these actions?
  Terraform will perform the actions described above.
  Only 'yes' will be accepted to approve.

  Enter a value: yes

proxmox_vm_qemu.prod-vm[2]: Creating ...
proxmox_vm_qemu.prod-vm[0]: Creating ...
proxmox_vm_qemu.prod-vm[1]: Creating ...
```

*Nota.* La figura muestra el plan de ejecución generado por Terraform, que en este caso indica que se adicionan tres máquinas virtuales. El plan resume las variables y configuraciones que se aplicarán una vez que el usuario confirme la ejecución del despliegue.

Finalmente, en la Figura 45 se puede observar el registro de las actividades realizadas durante el proceso de despliegue de infraestructura. Este registro muestra cada paso ejecutado para las máquinas virtuales.

**Figura 45**

*Registro de actividades del despliegue de infraestructura en Proxmox*

Tareas Log del cluster					
Hora de inicio	Hora final	Nodo	Nombre de Usuario	Descripción	Estado
Oct 23 17:28:09	Oct 23 17:28:09	pv5	root@pam	VM 400 - Configurar	OK
Oct 23 17:28:04	Oct 23 17:28:04	pv5	root@pam	VM 402 - Configurar	OK
Oct 23 17:28:04	Oct 23 17:28:05	pv5	root@pam	VM 401 - Configurar	OK
Oct 23 17:12:59	Oct 23 17:27:46	pv5	root@pam	VM 121 - Clonar	OK
Oct 23 17:12:59	Oct 23 17:27:46	pv5	root@pam	VM 121 - Clonar	OK
Oct 23 17:12:59	Oct 23 17:27:46	pv5	root@pam	VM 121 - Clonar	OK

*Nota.* La figura muestra el registro de las actividades realizadas en Proxmox durante el despliegue automatizado de las máquinas virtuales.

En la Figura 46 se puede observar el despliegue de las tres nuevas máquinas virtuales fue finalizado satisfactoriamente sin ningún error. El mensaje final indica que

se añadieron tres recursos nuevos, sin cambios ni destrucciones de los recursos preexistentes.

**Figura 46**

*Finalización del despliegue de máquinas virtuales con Terraform*

```
proxmox_vm_qemu.prod-vm[1]: Still creating ... [15m50s elapsed]
proxmox_vm_qemu.prod-vm[0]: Still creating ... [15m50s elapsed]
proxmox_vm_qemu.prod-vm[2]: Still creating ... [15m50s elapsed]
proxmox_vm_qemu.prod-vm[2]: Creation complete after 15m54s [id=pv5/qemu/402]
proxmox_vm_qemu.prod-vm[1]: Still creating ... [16m0s elapsed]
proxmox_vm_qemu.prod-vm[0]: Still creating ... [16m0s elapsed]
proxmox_vm_qemu.prod-vm[1]: Creation complete after 16m2s [id=pv5/qemu/401]
proxmox_vm_qemu.prod-vm[0]: Creation complete after 16m8s [id=pv5/qemu/400]

Apply complete! Resources: 3 added, 0 changed, 0 destroyed.
```

*Nota.* La figura muestra la confirmación del éxito en la creación de tres máquinas virtuales, cada una con su propio ID.

#### 4.4 Escenarios de Laboratorios Automatizados

Esta sección proporciona tres guías de laboratorio que fueron diseñadas para las asignaturas seleccionados previamente, con el objetivo de implementar su automatización. Esta automatización no solo mejora la eficiencia en la creación y despliegue de entornos de laboratorio, sino que también garantiza una mayor reproducibilidad y escalabilidad. Al automatizar los laboratorios, se permite a los estudiantes acceder a escenarios de red complejos de manera rápida y precisa, reduciendo así el tiempo y los errores asociados con las configuraciones manuales.

La capacidad de replicar automáticamente escenarios de red facilita un aprendizaje práctico y dinámico. Esto asegura que cada estudiante tenga una experiencia uniforme, sin importar la máquina que utilicen. Además, la automatización permite una fácil ampliación de los laboratorios para grupos más grandes sin comprometer la calidad de las prácticas, optimizando así tanto los recursos del laboratorio como el tiempo del docente.

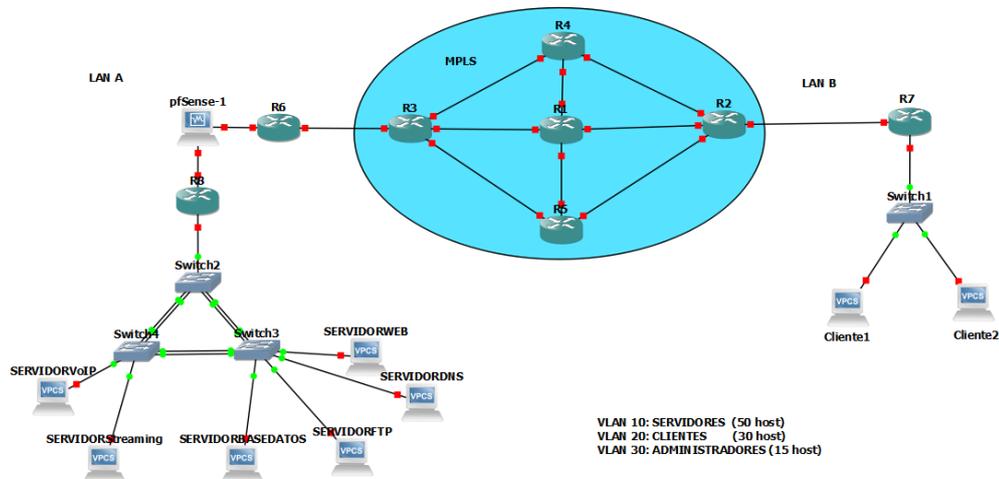
#### **4.4.1 Adaptación de guías de laboratorios**

Se han diseñado y adaptado tres guías de laboratorio en esta sección como se detalla en el Anexo H. Estas guías han sido preparadas en función de los objetivos y requisitos de las asignaturas seleccionados para automatizar su implementación en el entorno de laboratorio. La estructura de las guías se ha dividido de la siguiente manera:

**Guía 1.** La misma que corresponde a la asignatura de Ingeniería de Tráfico y se divide en tres partes, cada una orientada a proporcionar un enfoque práctico y teórico para la gestión y optimización del tráfico de red para un mejor rendimiento. El contenido de cada parte se explica a continuación:

**Parte 1: Configuraciones básicas, seguridad y conectividad.** La primera parte de esta guía se centra en establecer una base sólida para la infraestructura de red a través de configuraciones iniciales y medidas de seguridad. En esta sección, se realizan configuraciones básicas en cada router, incluyendo la asignación de direcciones IP a los dispositivos de red de acuerdo con la topología de red establecida en la guía, como se muestra en la Figura 47.

**Figura 47**  
Topología de red MPLS



*Nota.* En el gráfico se muestra la topología de red establecida para la red MPLS de transporte con servicios.

Además, se implementan configuraciones de seguridad en los puertos para prevenir el acceso no autorizado utilizando técnicas como la Port Security. Por otro lado, EtherChannel se configura para mejorar la redundancia y el rendimiento de la red al combinar varios enlaces físicos en un único enlace lógico; esto aumenta el ancho de banda disponible y garantiza una mayor tolerancia a fallos.

Finalmente, se implementan VLANs para una segmentación eficiente de la red, lo que mejora tanto la seguridad como el rendimiento operativo. Esta separación aísla diferentes secciones de la red, reduciendo el dominio del broadcast y facilitando el control del tráfico de la red de manera más precisa.

**Parte 2: Configuración de MPLS y servicios.** En la segunda parte, se implementa una red basada en MPLS para mejorar la eficiencia del enrutamiento y el tráfico de la red. Esto implica configurar la configuración básica de MPLS en los routers, lo que permite una mejora en el enrutamiento y la eficiencia de los paquetes de datos a través de la red.

Además, se configura un firewall utilizando pfSense que tiene como objetivo proteger la red de posibles amenazas externas. Este firewall asegura la seguridad de la red a través de políticas de filtrado y control de acceso, asegurando que solo el tráfico autorizado pueda ingresar a los recursos internos. Además, se implementan servicios clave que proporcionan funcionalidad adicional a la red, como FTP, bases de datos, DNS, servidores web, VoIP y streaming.

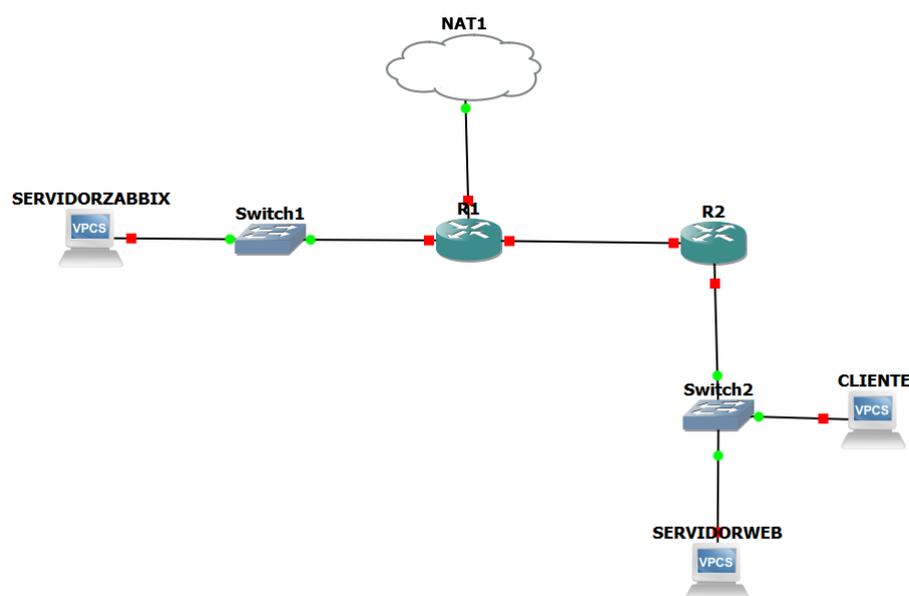
***Parte 3: Configuración de Calidad de Servicio (QoS).*** La parte final de la guía se centra en la Calidad de Servicio (QoS), un componente crucial para garantizar que servicios críticos como VoIP o videoconferencias reciban tratamiento prioritario en la red. La implementación de QoS es clave para gestionar con eficacia el tráfico, especialmente en entornos de alta demanda donde garantizar la eficiencia y calidad del servicio es primordial.

De esta manera, se pretende clasificar y marcar diferentes tipos de tráfico de red, permitiendo una gestión diferenciada basada en la importancia y requisitos de los datos transmitidos. Este proceso asegura que el tráfico más sensible, como voz o video, reciba prioridad sobre el tráfico menos crítico, mejorando así la experiencia del usuario final.

**Guía 2: Monitorización con Zabbix.** Esta guía pertenece a la asignatura de Administración de Redes de Telecomunicaciones y se enfoca en el monitoreo de dispositivos de red utilizando Zabbix, una herramienta de código abierto que monitorea infraestructuras de red. Se divide en dos partes: instalación de la herramienta de monitoreo y configuración de dispositivos de red. A continuación, se presenta el contenido de cada parte:

**Parte 1: Configuraciones iniciales.** Esta sección es fundamental para establecer la base de la infraestructura de red que será monitoreada. Se toma como referencia la topología establecida en la guía, mostrada en la figura 48. Los pasos de esta parte están orientados a preparar los dispositivos de red y garantizar la conectividad entre ellos.

**Figura 48**  
Topología de red implementada para la monitorización



*Nota.* En el gráfico se muestra la topología de red diseñada para llevar a cabo la monitorización con Zabbix. La estructura incluye los routers, switches y servidores necesarios para la correcta implementación del sistema de monitoreo. Cada dispositivo está configurado para garantizar la conectividad y permitir la recopilación de datos de la infraestructura de red en tiempo real.

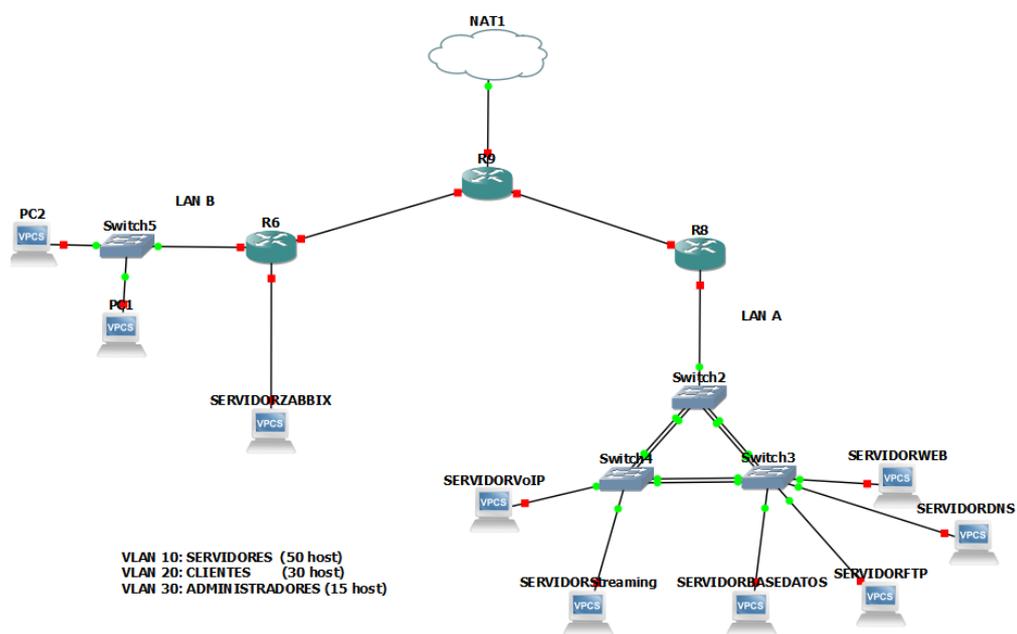
**Parte 2: Configuración e Instalación de Zabbix.** Esta sección está dedicada a la implementación completa de Zabbix como una solución de monitoreo. El proceso implica la configuración del servidor Zabbix y la configuración de agentes en diferentes dispositivos de la red. Estos agentes son responsables de capturar datos críticos sobre el rendimiento y el estado de cada host. Después de que Zabbix esté completamente instalado y comience a recopilar datos, interpreta y analiza la información obtenida. Esto ayuda a identificar posibles problemas con la infraestructura, como cuellos de

botella o fallos de dispositivos, y realizar mejoras para una mejor optimización del rendimiento, así como garantizar la estabilidad operativa de la red.

**Guía 3: Monitorización de Redes con Zabbix.** Esta guía de laboratorio corresponde a la asignatura de Administración de Redes de Telecomunicaciones y está destinada a ser utilizada en el monitoreo de redes utilizando la plataforma Zabbix. El objetivo de esta guía es configurar, integrar y usar Zabbix para el monitoreo eficiente de dispositivos y servicios de red, permitiendo así la detección temprana de problemas y la optimización de su desempeño. La guía se divide en dos secciones principales: configuración inicial de la red e implementación de Zabbix para su monitoreo. A continuación, se detallan los contenidos de cada parte:

***Parte 1: Configuración inicial de la red.*** Esta sección cubre la preparación y configuración básica de dispositivos de red para asegurar su correcto funcionamiento y conectividad antes de implementar Zabbix para monitoreo. Estas actividades incluyen la asignación de direcciones IP, la configuración de VLANs para segmentar el tráfico, y la implementación de mecanismos de seguridad y redundancia que aseguran una infraestructura robusta y eficiente. Tales configuraciones se aplican de acuerdo con la topología establecida en la guía como se muestra en la Figura 49.

**Figura 49**  
 Topología de red para la monitorización de redes



*Nota.* En el gráfico se muestra la topología de red que incluye múltiples servidores para proporcionar una infraestructura completa y diversa que abarque servicios críticos como VoIP, FTP, web y bases de datos. Además, se implementan VLANs para segmentar el tráfico de red y garantizar la eficiencia y seguridad operativa.

**Parte 2: Implementación de Zabbix para el monitoreo.** Esta sección explica la instalación, configuración e integración de Zabbix en la infraestructura de red. Implica configurar Zabbix Server e instalar agentes Zabbix en dispositivos para recopilar datos de rendimiento. Además, se explican características avanzadas de Zabbix, como la creación de gráficos, mapas de red y triggers, que permiten un monitoreo detallado y una gestión proactiva de la red.

#### 4.4.2 Implementación y despliegue de los laboratorios predefinidos

Esta sección explica cómo implementar y desplegar laboratorios predefinidos para redes. Esto se hace comenzando con una máquina virtual que está configurada como una plantilla base, sobre la cual se desarrollan las topologías de red descritas en cada una de las guías de laboratorio. Estas topologías están diseñadas para simular

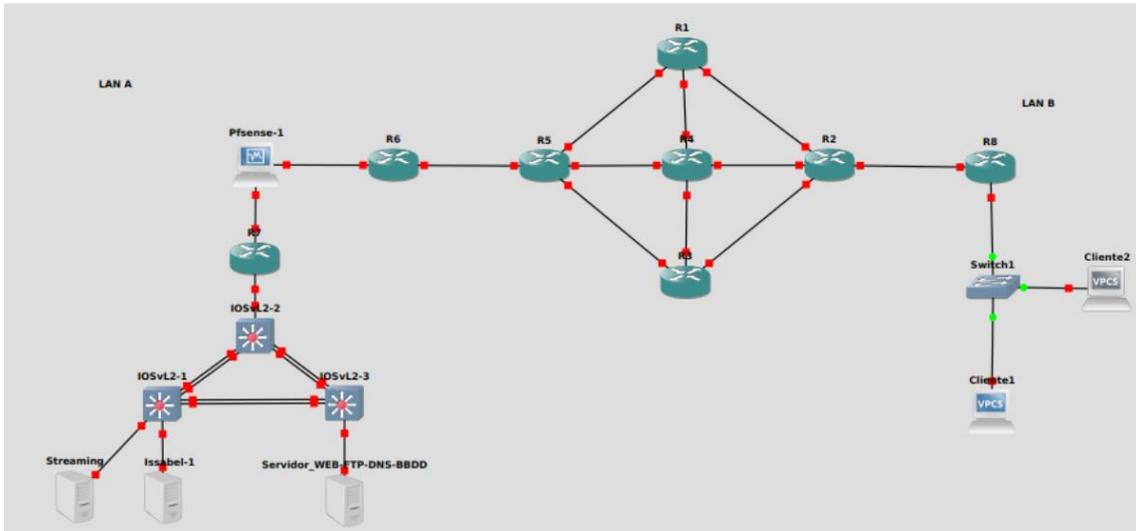
escenarios de la vida real que los estudiantes tendrán que analizar y trabajar durante sus prácticas.

Para este propósito, se utilizó una máquina virtual basada en el sistema operativo Linux, específicamente Ubuntu 22.04, donde se instalaron todas las herramientas necesarias para desplegar laboratorios. La elección de este sistema operativo garantiza estabilidad y compatibilidad con aplicaciones de virtualización y software de diseño de redes.

Se configuraron dos herramientas clave, GNS3 y VirtualBox, en esta máquina virtual. GNS3 se utiliza para diseñar y simular topologías de red, lo que permite crear entornos de red complejos que replican escenarios reales. Por otro lado, VirtualBox se utiliza para crear y gestionar máquinas virtuales adicionales en las que se alojarán los servidores necesarios para cada laboratorio. Estos servidores sirven como nodos en las topologías de red, permitiendo a los estudiantes tener experiencia práctica en un entorno virtual controlado y seguro.

Así, una vez que se instalan las herramientas necesarias, se crean máquinas virtuales en VirtualBox antes de instalar y configurar los servidores requeridos para construir la topología en GNS3 según las especificaciones de cada guía de laboratorio. Tales configuraciones aseguran que cada servidor esté correctamente colocado y cumpla con los requisitos de conectividad establecidos. En la Figura 50 se presenta la topología correspondiente a la guía 1, mientras que la Figura 51 muestra la topología de la guía 2 y la Figura 52 muestra la topología correspondiente a la guía 3.

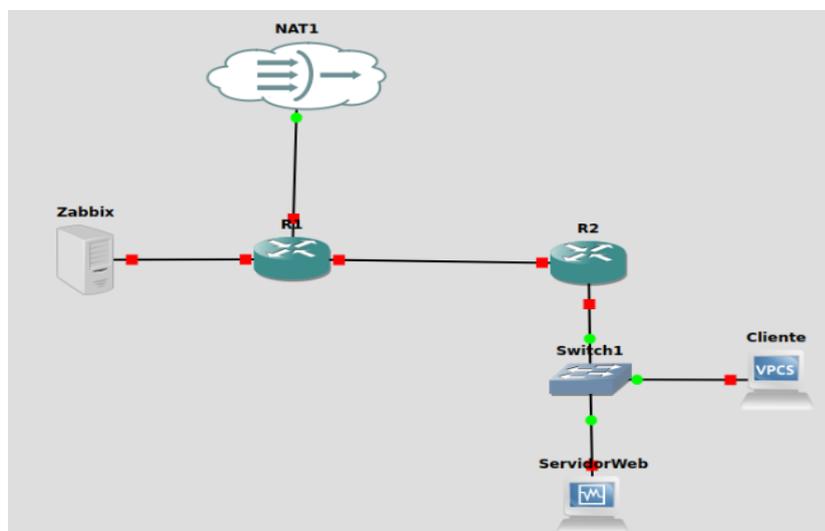
**Figura 50**  
*Topología de red de la guía de laboratorio 1*



*Nota.* En el gráfico se muestra la topología correspondiente a la guía de laboratorio. Esta topología ilustra la estructura de red configurada para el laboratorio, incluyendo los elementos clave de interconexión y distribución de servicios entre las redes LAN A y LAN B.

Esta topología incluye dos redes LAN (LAN A y LAN B) interconectadas por una nube MPLS para simular la integración de redes en un entorno real. LAN A tiene un servidor de streaming, un servidor VoIP y otras máquinas que alojan servidores web, FTP, DNS y de bases de datos. Además, cuenta con un firewall para la gestión y control del tráfico. En LAN B hay dos clientes conectados a través de un switch que representan un entorno de usuario final.

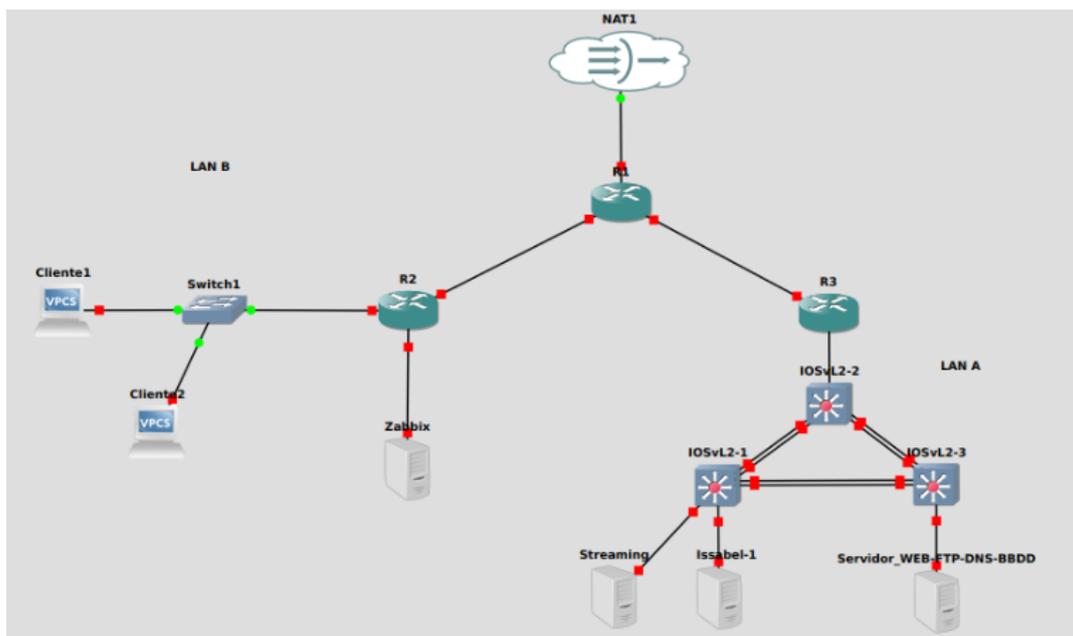
**Figura 51**  
*Topología de red de la guía de laboratorio 2*



*Nota.* En el gráfico se muestra la topología correspondiente a la guía de laboratorio 2, que ilustra la simulación de un entorno de monitoreo de red utilizando *Zabbix*. La configuración refleja un escenario básico de red empresarial, integrando componentes de monitoreo y gestión de tráfico.

Esta topología incluye un servidor de monitoreo *Zabbix*, vinculado a un router principal que se interconecta con un dispositivo NAT para la salida a la red externa. La red interna consiste en un router R2 conectado a un switch que distribuye la conexión a un servidor web y un dispositivo cliente. Esta configuración representa un entorno básico de red donde los servicios de monitoreo y el acceso a internet están integrados, simulando un escenario típico de redes corporativas.

**Figura 52**  
*Topología de red de la guía de laboratorio 3*



*Nota.* En el gráfico se muestra la topología correspondiente a la guía de laboratorio 3. Esta configuración simula un entorno de red dividido en dos segmentos principales, LAN A y LAN B, interconectados mediante los routers R2 y R3. La topología incluye un servidor de monitoreo *Zabbix* y un dispositivo NAT para salida a redes externas, proporcionando una estructura completa de monitoreo y servicios.

Esta topología simula un entorno de red segmentado en dos áreas principales, LAN A y LAN B, interconectadas a través de los routers R2 y R3. Varios servidores se encuentran en LAN A; estos incluyen un servidor de streaming, un servidor VoIP y un

servidor de propósito múltiple que proporciona servicios web, servicios FTP, servicios DNS así como servicios de base de datos. Estos servidores están conectados a través de tres switches que gestionan el tráfico interno. Por otro lado, en LAN B hay dos clientes conectados a un switch principal que a su vez se conecta al resto de la red a través del router R2; esta topología incluye un servidor de monitoreo Zabbix. Esta estructura presenta un escenario de red avanzado que involucra múltiples nodos de servicio y nodos de monitoreo que simulan un entorno empresarial complejo.

Una vez que las topologías han sido configuradas, el siguiente paso es automatizar su despliegue, utilizando herramientas como Terraform y Ansible para crear los entornos de laboratorio de forma rápida y eficiente. Este proceso asegura que los entornos de laboratorio se creen rápidamente y sin problemas, sin intervención manual en cada despliegue.

La automatización garantiza que todos los recursos sean replicados exactamente tal como están definidos, permitiendo a los estudiantes acceder a laboratorios uniformes y consistentes. Este enfoque optimiza tanto el tiempo de preparación como la calidad del entorno, asegurando que cada estudiante trabaje en un laboratorio idéntico, confiable y adecuado para el aprendizaje práctico de redes de telecomunicaciones.

Para esto se llevó a cabo una prueba con estudiantes para evaluar los beneficios de implementar Infraestructura como Código (IaC). La actividad consistió en el despliegue de 8 máquinas virtuales asignadas a diferentes grupos de estudiantes, diseñada para explorar tanto el proceso manual como el automatizado de configuración infraestructural.

La prueba se dividió en dos etapas fundamentales, cada una destinada a simular situaciones prácticas en un entorno de laboratorio:

1. **Configuración Manual:** En esta etapa inicial, se pidió a los estudiantes que desplegaran la infraestructura manualmente. Este proceso implicó instalar y configurar cada componente individual necesario, replicando las tareas típicas de un entorno tradicional. Este método permitió afrontar desafíos comunes como errores humanos, tiempos prolongados y variabilidad en la configuración entre diferentes grupos.
2. **Configuración Automatizada:** En la segunda etapa, se utilizó Infraestructura como Código (IaC) a través de herramientas como Terraform y Ansible para automatizar el proceso. Los estudiantes trabajaron con máquinas virtuales previamente desplegadas y se centraron en ejecutar la configuración de dispositivos de red y elementos de topología de manera estandarizada y eficiente. Este paso subrayó la capacidad de IaC para reducir el tiempo mientras aseguraba configuraciones uniformes.

Este enfoque comparativo ha permitido identificar y analizar las diferencias clave entre ambos métodos, tomando en cuenta aspectos tales como el tiempo de ejecución, la consistencia de los entornos generados y la precisión de la configuración. De esta manera, se obtuvo una visión más completa y detallada sobre la eficacia de la automatización en contextos educativos. Las especificaciones de la prueba realizada se detallan en la Tabla 14.

**Tabla 14**  
*Especificaciones de la prueba*

Aspecto	Detalle
---------	---------

<b>Número de estudiantes</b>	32
<b>Número de máquinas virtuales</b>	8
<b>Herramientas utilizadas</b>	Terraform, Ansible
<b>Métodos de configuración</b>	Manual y automatizado
<b>Tiempo promedio (manual)</b>	2-3 horas
<b>Tiempo promedio (automatizado)</b>	1-2 horas

*Nota.* La tabla presenta las especificaciones empleadas para comparar los métodos manual y automatizado en el despliegue de infraestructuras virtuales.

La tabla destaca las ventajas de los métodos automáticos en el despliegue de infraestructura dentro de un contexto educativo. En esta prueba, participaron treinta y dos estudiantes, utilizando ocho máquinas virtuales alojadas en un servidor Proxmox. Los resultados evidenciaron que utilizar herramientas de Infraestructura como Código (IaC) como Terraform y Ansible redujo significativamente el tiempo promedio dedicado a las configuraciones. Además, la automatización garantiza entornos generados más consistentes y precisos, optimizando así los recursos disponibles y minimizando los tiempos de inactividad.

#### ***4.4.3 Resultados y análisis de la implementación***

En esta sección se presentan los resultados de la creación y configuración de máquinas virtuales mediante la implementación de entornos de laboratorio basados en Infraestructura como Código (IaC). Este análisis abarca aspectos técnicos y educativos, evaluando la eficiencia de herramientas como Terraform y Ansible utilizadas, además de la percepción de los estudiantes sobre su impacto en los laboratorios de redes. Se pone énfasis en medir los tiempos de implementación, la uniformidad de los entornos

generados y la utilidad en el aprendizaje práctico, mostrando cómo IaC puede superar desafíos operativos, optimizar recursos y mejorar la experiencia educativa.

La evaluación técnica considera el tiempo requerido para desplegar y configurar cada VM como un indicador clave de la efectividad del enfoque automatizado. A su vez, se complementa con los hallazgos de una encuesta a los estudiantes participantes que exploran la facilidad de configuración, la reducción de errores, la utilidad educativa, así como los tiempos de implementación.

**Tiempo de despliegue y configuración.** El tiempo de despliegue automatizado se ha medido desde el momento en que se ejecuta el comando “**terraform apply**” hasta que la máquina virtual está completamente operativa. Este proceso involucra dos fases principales, como crear una máquina virtual utilizando Terraform y configurar la máquina con playbooks de Ansible. Así, los tiempos promedio obtenidos en cada fase, mostrados en la Tabla 15, permiten evaluar la eficiencia de las herramientas utilizadas y asegurar que los entornos de laboratorio estén listos lo antes posible.

**Tabla 15**  
*Tiempo promedio de despliegue y configuración de la máquina virtual*

<b>Fase del Proceso</b>	<b>Tiempo Promedio (minutos)</b>
<b>Creación de la máquina virtual</b>	8 min
<b>Configuración de la máquina virtual</b>	5 min
<b>Tiempo total del despliegue</b>	13 min
<b>Automatizado</b>	

*Nota.* En la tabla se muestra los tiempos de despliegue y configuración de una máquina virtual en un despliegue ideal, realizado en condiciones óptimas donde la máquina virtual a clonar se encuentra apagada y se dispone del espacio de almacenamiento necesario.

En un caso ideal en el que hay suficiente espacio de almacenamiento para el despliegue y la máquina a clonar está apagada, esta tabla muestra los tiempos específicos de cada fase del despliegue. Estos tiempos son una representación del funcionamiento perfecto de las herramientas automatizadas, y garantizan que la creación y configuración de una máquina virtual se haga rápidamente y eficientemente, ahorrando recursos y permitiendo a los estudiantes acceder a laboratorios.

### **Análisis de la implementación.**

Para evaluar el impacto de la Infraestructura como Código (IaC) en el desarrollo de laboratorios de redes, se realizó una encuesta dirigida a 32 estudiantes que participaron en la implementación de prueba de la automatización de la infraestructura. La encuesta, presentada en el Anexo G, constó de preguntas cerradas centradas en áreas clave como la facilidad de configuración, consistencia, reducción de errores, utilidad educativa y tiempos de despliegue. Los resultados analizados en el Anexo H proporcionan una perspectiva integral sobre cómo los estudiantes perciben la implementación de IaC dentro de un contexto educativo.

No obstante, los desafíos enfrentados durante la práctica, como la falta de recursos en el servidor, la calidad variable de la red y cortes de energía, la práctica culminó con éxito. Esto demuestra que el enfoque automatizado es lo suficientemente resiliente como para superar las dificultades operativas, asegurando la continuidad y efectividad de las actividades. Además de evaluar los beneficios técnicos y educativos de IaC a través de esta experiencia, también mostró su aplicabilidad en escenarios de la vida real, reforzando así su importancia como una herramienta clave para modernizar y optimizar los laboratorios de redes.

Los datos obtenidos revelan que el 56% de los estudiantes considera que IaC facilita en gran medida la configuración inicial de los laboratorios, lo que disminuye la complejidad operativa y acelera el inicio de las prácticas. Además, el 65% afirmó que la automatización garantiza entornos consistentes y uniformes, un aspecto crucial para el aprendizaje efectivo y práctico. Sin embargo, solo el 44% opinó que IaC contribuye a reducir errores, lo que implica una necesidad de reforzar capacitación y estrategias de implementación para maximizar este beneficio.

Además, en relación con la educación, el 78% de los estudiantes respondió “útil” o “muy útil” cuando se les preguntó sobre la aplicación del IaC en sus clases. Igualmente, este mismo número también admitió que usar IaC reduce los tiempos de despliegue mejorando así eficiencia y minimizando tiempos muertos lo cual permite una mayor utilización de las sesiones prácticas.

Al final, el 85% de los estudiantes recomendaría que IaC se implementara en otros cursos del programa debido a su alto nivel de aceptación. Esta opinión está respaldada por los tiempos promedio de configuración, que indican una disminución significativa en comparación con los métodos manuales, confirmando así la eficiencia y efectividad del enfoque automatizado en entornos educativos.

El análisis del rendimiento del nodo pv5 se muestra en la Figura 53 del nodo Proxmox en el que se puede notar que la carga promedio de CPU fue del 11.96%, lo que muestra eficiencia en los recursos computacionales. Sin embargo, el nodo operó utilizando el 98.18% de la memoria RAM, mostrando una alta demanda de recursos atribuida a la complejidad de los laboratorios predefinidos en máquinas virtuales previamente implementadas y además el uso de memoria SWAP alcanzó el 91.23%, lo que indica una insuficiencia de RAM para manejar las operaciones en curso; por lo tanto, se debería considerar evaluar la expansión de la memoria física para que no afecte

negativamente el rendimiento de los escenarios de laboratorio. En cuanto al almacenamiento, se utilizó el 25% del espacio en disco, asegurando así una capacidad suficiente que debe ser monitoreada a medida que cambian los escenarios de laboratorio.

**Figura 53**  
*Resumen de recursos en el nodo pv5*



*Nota.* En el gráfico se puede observar el análisis detallado del consumo de los recursos del nodo pv5.

Por otro lado, considerando la prueba realizada según la guía de laboratorio sobre la red MPLS con transporte de servicios, se identificó un alto consumo de memoria RAM relacionado con el número de máquinas virtuales activas. Según los datos presentados en la Tabla 16, se concluye que el número óptimo de máquinas virtuales para ejecutar este tipo de prácticas que requieren de un consumo significativo de recursos sin incidencias es de cuatro. El uso de más de esta cantidad genera una sobrecarga significativa en la memoria SWAP, comprometiendo la estabilidad del sistema.

**Tabla 16**  
*Consumo de recursos según las máquinas virtuales activas*

N° de máquinas virtuales	% Memoria RAM utilizada	% Memoria SWAP utilizada
1 MV	45 %	0%
2 MV	63,74%	0%
3 MV	87,57%	0%

<b>4 MV</b>	98,46%	62,29%
<b>5 MV</b>	98,58%	64,21%
<b>6 MV</b>	98,27%	77,18%
<b>7 MV</b>	84,74%	82,22%
<b>8 MV</b>	98,18%	91,23%

*Nota.* La tabla detalla el porcentaje de consumo de memoria RAM y memoria SWAP en función del número de máquinas virtuales utilizadas en el entorno de laboratorio. Se evidencia que con más de cuatro máquinas virtuales el consumo de memoria SWAP incrementa significativamente, afectando la estabilidad del sistema.

Con base en estos resultados, se enfatiza la necesidad de optimizar el uso de recursos asignados y planificar futuras expansiones para asegurar la eficiencia y el éxito de los laboratorios sin comprometer la estabilidad del entorno.

## Conclusiones y recomendaciones

### Conclusiones

La adopción de Infraestructura como Código (IaC) ha surgido como una herramienta útil que ayuda a trazar una mejor forma de administrar la infraestructura tecnológica. El uso de herramientas como Terraform y Ansible ha ayudado a automatizar procesos tediosos y repetitivos, logrando así un nivel operativo del 70%, según indican los datos obtenidos de la encuesta, mediante una gran reducción del tiempo de implementación y la disminución de la cantidad de posibles errores humanos.

Esta estandarización también ha sido beneficiosa para permitir una experiencia enriquecedora donde los instructores y estudiantes pueden centrarse en elementos que son más pertinentes para la práctica a desarrollar en lugar de configuraciones iniciales extensas, como señalaron el 75% de los que participaron en la investigación. Con esta combinación, la extensión y la integración más fácil de Terraform con otras herramientas como Ansible para aprovisionar y configurar la infraestructura ha resultado en un mayor éxito al superar un aumento del 81% en la estandarización, lo que asegura que hay uniformidad y consistencia en los entornos de laboratorios predefinidos.

Las pruebas y la información muestran que el uso de IaC permite que los laboratorios que anteriormente requerían más de 3 horas para configurarse, ahora están en funcionamiento en un tiempo promedio de 1.5 horas. Esto representa una mejora del 50% en la velocidad de implementación, adaptándose de forma ágil a los requerimientos académicos.

Tanto con los logros alcanzados, resultó necesario identificar la importancia de fortalecer la infraestructura tecnológica del servidor Proxmox. Será esencial aumentar las capacidades de memoria interna, espacio en disco y potencia de los procesadores para garantizar un soporte adecuado a laboratorios de gran complejidad, como los de MPLS, y conseguir una experiencia educativa más fluida y enriquecedora.

Unos de los puntos más fuertes de IaC en Proxmox es la minimización de la dependencia del equipo físico, enfatizado por el 87% de los encuestados. Esto no solo asegura un entorno ecológico y eficiente, sino que también deja espacio que está listo para los futuros desafíos tecnológicos.

En general, la adopción de la tecnología de Infraestructura como Código (IaC) no es solo un cambio en la gestión técnica; también reestructura cómo los estudiantes practican y aprenden entornos al participar en dinámicas profesionales del mundo real, que son pocas veces disponibles. Esto permite un proceso educativo que es más novedoso, más asequible, más ecológico y así sucesivamente, como lo esclarecen las respuestas positivas del 85% de los participantes, quienes promoverían esta metodología hacia otros cursos.

## Recomendaciones

Al trabajar en la plataforma de virtualización Proxmox, es esencial asegurarse de que todos los scripts y configuraciones cumplan con los requisitos del entorno objetivo. Esto implica probar los scripts en entornos de laboratorio controlados antes de usarlos en configuraciones más complejas.

Para gestionar correctamente los scripts y configuraciones creados con Terraform y Ansible en Proxmox, es importante mantener una documentación organizada y estructurada de cada plantilla y playbook de manera detallada. Una documentación adecuada no solo mejora el flujo de trabajo, sino que también garantiza que otros usuarios encuentren fácil entender y reutilizar los scripts.

Para garantizar una funcionalidad óptima, particularmente en laboratorios con topologías complejas, se recomienda aumentar los recursos de hardware del servidor Proxmox, incluyendo RAM, espacio en disco, entre otros detalles. Esto permitirá un soporte más efectivo de las máquinas virtuales desplegadas.

Dado que los scripts de configuración fueron creados en un entorno Linux, es importante mantener siempre este tipo de entorno para beneficiarse plenamente de la compatibilidad y las capacidades de las herramientas utilizadas. Esto también facilita el trabajo con Proxmox, que está optimizado para sistemas que ejecutan Linux.

Es recomendable establecer una estructura clara y ordenada de los directorios donde se almacenen los scripts y configuraciones. Esto no sólo facilita el acceso o el control de dichas configuraciones, sino que también ayuda a prevenir errores a la hora de ejecutar determinadas tareas.

## Referencias Bibliográficas

- Behrend, J. J. (2023, October 23). *Evolución de la infraestructura y los roles: Desde Sysadmin, DevOps, DevSecOps hasta Platform Engineering*.  
<https://www.linkedin.com/pulse/evolucion-infraestructura-y-los-roles-desde-hasta-juan-jose-behrend/?originalSubdomain=es>.
- Benalcázar Montesdeoca, G. M. (2019). *Implementación de una infraestructura de IT virtual para el data center de la facultad de ingeniería en Ciencias Aplicadas, en la Universidad Técnica del Norte*. <https://repositorio.utn.edu.ec/handle/123456789/9560>
- Carabas, M., Mihai, D., & Brebene, A. (2023). *Automated Infrastructure Provisioning*.
- Castro, J. G. A. (2023). Herramienta PROXMOX como plataforma tecnológica para la virtualización de servidores en la organización. *Aula Virtual*.
- Errazquin, M. A. (2023). *Configuración y puesta en marcha de un sistema de virtualización en alta disponibilidad vía Proxmox*.
- F5. (2023a). *¿Qué es la infraestructura como código? | F5*. F5.  
[https://www.f5.com/es\\_es/glossary/infrastructure-as-code-iac](https://www.f5.com/es_es/glossary/infrastructure-as-code-iac)
- F5. (2023b). *¿Qué es la infraestructura como código? | F5*. F5.  
[https://www.f5.com/es\\_es/glossary/infrastructure-as-code-iac](https://www.f5.com/es_es/glossary/infrastructure-as-code-iac)
- Fernández Romero, Y., & García Pombo, K. (2011). Virtualización. *Revista Telem@tica*.  
<http://biblioteca.udgvirtual.udg.mx>
- Flores, F. (2023). *Infraestructura como Código: Qué es y herramientas | OpenWebinars*. OpenWebinars S.L. <https://openwebinars.net/blog/infraestructura-como-codigo-que-es-y-herramientas/>
- Guerriero, M., Garriga, M., Tamburri, D. A., & Palomba, F. (2019). Adoption, Support, and Challenges of Infrastructure-as-Code: Insights from Industry. *Proceedings - 2019 IEEE International Conference on Software Maintenance and Evolution, ICSME 2019*, 580–589.  
<https://doi.org/10.1109/ICSME.2019.00092>
- HashiCorp. (2023). *Terraform | Desarrollador de HashiCorp*.  
[https://developer.hashicorp.com/terraform?product\\_intent=terraform](https://developer.hashicorp.com/terraform?product_intent=terraform)
- IBM. (2023a). *Infraestructura como código | IBM*. <https://www.ibm.com/es-es/topics/infrastructure-as-code>
- IBM. (2023b). *Introducción: clústeres - Documentación de IBM*.  
<https://www.ibm.com/docs/es/was-zos/9.0.5?topic=servers-introduction-clusters>
- IONOS Digital Guide. (2023). *Infraestructura as code (IaC): Función y ejemplos - IONOS*.  
<https://www.ionos.com/es-us/digitalguide/servidores/know-how/infrastructure-as-code/>
- LODE, S., & DDR, O. (2010). Ley orgánica de educación superior. Quito, *Pichincha: Nacional*.  
 Obtenido de [Http://Es.Slideshare. Net/Fortizvizuet/Loes-2010](http://Es.Slideshare.Net/Fortizvizuet/Loes-2010).

- Objetivos de Desarrollo Sostenible. (2015). Objetivos de desarrollo sostenible. *Obtenido de:* <Http://Www.Undp.Org/Content/Undp/Es/Home/Sustainable-Development-Goals.Html>.
- Proxmox Server Solutions GmbH. (2023a). *Entorno virtual Proxmox: plataforma de virtualización de servidores de código abierto*. <https://www.proxmox.com/en/proxmox-virtual-environment/overview>
- Proxmox Server Solutions GmbH. (2023b, November 23). *Proxmox VE Administration Guide*. <https://pve.proxmox.com/pve-docs/pve-admin-guide.html>
- Red Hat. (2022, May 11). *¿Qué es la infraestructura como código - Infrastructure as Code?* <https://www.redhat.com/es/topics/automation/what-is-infrastructure-as-code-iac>
- Red Hat. (2023a). *Automatización de infraestructura con Red Hat Ansible Automation Platform*. <https://www.redhat.com/en/technologies/management/ansible/infrastructure>
- Red Hat. (2023b, September 20). *Automatización: ¿qué es y qué ventajas ofrece?* <https://www.redhat.com/es/topics/automation>
- Sánchez; Romero, D. (2021a). *Infraestructura como código*. *España de Creative Commons*.
- Sánchez; Romero, D. (2021b). *Infraestructura como código*. *España de Creative Commons*.
- Sorour, A., & Hamdy, A. (2022). DevOps and IaC to Automate the Delivery of Hands-On Software Lab Exams. *Proceedings - 2022 6th International Conference on Computer, Software and Modeling, ICCSM 2022*, 28–35. <https://doi.org/10.1109/ICCSM57214.2022.00012>
- Toaza, V. (2011). *SISTEMAS DE VIRTUALIZACIÓN PARA LA OPTIMIZACIÓN DE GASTOS EN LA INFRAESTRUCTURA DE TECNOLOGÍAS DE INFORMACIÓN EN LAS EMPRESAS PÚBLICAS DE LA PROVINCIA DE TUNGURAHUA*.
- Tommaso, L. Di, & Salvarredy, J. (2015). *Diseño de un proceso de gestión de infraestructura tecnológica y propuesta de un sistema que lo implemente*.
- Unisys. (2018). *AUTOMATIZACIÓN DE LA INFRAESTRUCTURA DE NUBE Y DE TI*. [https://www.app5.unisys.com/library/cmsmail/Automate/BR\\_180470\\_AutomatingCloudandITInfrastructure\\_ES\\_AA.pdf](https://www.app5.unisys.com/library/cmsmail/Automate/BR_180470_AutomatingCloudandITInfrastructure_ES_AA.pdf)

## ANEXOS

### Anexo A



## UNIVERSIDAD TÉCNICA DEL NORTE

### FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS

#### Carrera de Ingeniería en Telecomunicaciones

Proyecto: Implementación y automatización de infraestructura basada en código para los laboratorios del área de redes de la carrera de Telecomunicaciones en la Universidad Técnica del Norte.

Objetivo: Este cuestionario ha sido diseñado con el propósito de investigar el clúster de virtualización de la carrera de Telecomunicaciones en la Universidad Técnica del Norte, que será de importancia para comprender la implementación y automatización de dicha infraestructura.

#### Cuestionario sobre el Clúster de Virtualización

##### 1. Información General

- Nombre del entrevistado:

- Cargo:

- Fecha de la entrevista:

##### 2. Descripción del Clúster

a) ¿Cuál es el propósito principal del clúster de virtualización en su infraestructura?

b) ¿Qué tecnología de virtualización se utiliza en el clúster?

c) ¿Cuántos nodos físicos conforman el clúster actualmente?

d) ¿Cuál es la capacidad aproximada de recursos (CPU, RAM, almacenamiento) del clúster?

### 3. Estado Actual

a) ¿Cómo describiría la estabilidad del clúster en términos de rendimiento y disponibilidad?

b) ¿Se han experimentado recientemente problemas de rendimiento o interrupciones en el clúster? En caso afirmativo, ¿cuáles fueron las causas conocidas?

c) ¿Cuál es la utilización promedio de recursos del clúster? (Por ejemplo, CPU, RAM, almacenamiento)

d) ¿Cuál es la carga de trabajo principal que soporta el clúster? (Ej. máquinas virtuales de producción, desarrollo, pruebas, etc.)

e) ¿Cuál es el tiempo promedio de operación del clúster?

f) ¿Se utiliza el clúster para alguna asignatura en específico? En caso afirmativo, describir las actividades de la asignatura.

### 4. Gestión y Mantenimiento

a) ¿Qué herramientas o software se utilizan para la gestión del clúster?

b) ¿Con qué frecuencia se aplican actualizaciones y parches de seguridad en el clúster?

c) ¿Cómo se realiza la monitorización del clúster? ¿Hay algún sistema de alertas configurado?

d) ¿Cuál es el plan de respaldo y recuperación en caso de falla del clúster?

Elaborado por:

Lizbeth Espinoza

Aprobado por director:

  
Ing. Fabián Cuzme, MSc.

**Anexo B****UNIVERSIDAD TÉCNICA DEL NORTE****FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS****Carrera de Ingeniería en Telecomunicaciones**

Proyecto: Implementación y automatización de infraestructura basada en código para los laboratorios del área de redes de la carrera de Telecomunicaciones en la Universidad Técnica del Norte.

Objetivo: El propósito de este cuestionario es conocer sobre las asignaturas del área de redes y telecomunicaciones, con énfasis en identificar las necesidades específicas de infraestructura de laboratorios. Este análisis es fundamental para optimizar las prácticas educativas relacionadas con Redes de Área Local, Redes de Área Extensa, Sistemas de Comunicación Multimedia, Seguridad en Redes, Ingeniería de Tráfico y Administración de Redes.

**Cuestionario sobre las asignaturas de Redes y Telecomunicaciones****1. Información General**

- Nombre:

- Asignatura:

- Fecha de la entrevista:

**2. Asignaturas de Redes y Telecomunicaciones:**

- ¿Qué asignaturas relacionadas con redes y telecomunicaciones imparte actualmente?

- ¿Cuál es el enfoque principal de esta asignatura? (por ejemplo, conceptos teóricos, configuración y administración de redes, protocolos de comunicación, seguridad en redes, etc.)

### 3. Prácticas de Laboratorio:

- ¿Se realizan prácticas de laboratorio en estas asignaturas? Si es así, ¿con qué frecuencia y duración?

- ¿Qué tipo de actividades prácticas se llevan a cabo en el laboratorio? (configuración de equipos de red, simulaciones de redes, análisis de tráfico, pruebas de seguridad, etc.)

### 4. Infraestructura y Equipamiento:

- ¿Qué tipo de equipamiento de red está disponible en el laboratorio? (routers, switches, firewalls, herramientas de monitorización, etc.)

- ¿Se utilizan herramientas de software específicas para las prácticas de redes y telecomunicaciones? (GNS3, Wireshark, Packet Tracer, etc.)

### 5. Experiencia de Aprendizaje:

- ¿Qué aspectos ha encontrado más desafiantes durante las prácticas?

- ¿Cómo cree que las prácticas de laboratorio han contribuido a la comprensión de los conceptos teóricos en redes y telecomunicaciones?

### 6. Infraestructura como código:

- ¿Está familiarizado con el concepto de infraestructura como código?

- ¿Podría enumerar algunos beneficios que ofrece la infraestructura como código en comparación con métodos tradicionales de gestión de infraestructura?

- ¿Estaría dispuesto a implementar infraestructura como código para facilitar el despliegue de laboratorios?

Elaborado por:

Lizbeth Espinoza

Aprobado por director:

Ing. Fabián Cuzme, MSc.



## Anexo C

### *Manual de instalación de las herramientas de Infraestructura como Código*

#### 1. Introducción

Este manual proporciona una guía detallada para la instalación y configuración de las herramientas de Infraestructura como Código (IaC), esenciales para la gestión y automatización de infraestructura en el ámbito de TI.

#### 2. Requisitos previos

Antes de proceder con la instalación, es importante cumplir con los siguientes requisitos:

- Un sistema operativo basado en Linux (en este caso, se ha utilizado Kali Linux; los comandos aquí descritos son aplicables a sistemas similares como Ubuntu y Debian).
- Conexión a internet para la descargar de paquetes y dependencias.
- Privilegios de superusuario.

#### 3. Instalación de Terraform

##### 3.1. Actualización del Sistema

Es recomendable actualizar el sistema antes de instalar nuevos paquetes. Para ello, se ejecute los siguientes comandos *“sudo apt-get update”* y *“sudo apt-get upgrade”*.

Esto comandos aseguran que todos los paquetes instalados se actualizan y que se eliminen las versiones antiguas.

##### 3.2. Descargar e instalar Terraform

###### 3.2.1. Añadir el Repositorio de HashiCorp

Terraform no está disponible en los repositorios predeterminados de Kali Linux. Por lo tanto, es necesario agregar el repositorio oficial de HashiCorp utilizando los siguientes comandos:

```
“sudo apt-get install -y gnupg software-properties-common”
```

```
“wget -O- https://apt.releases.hashicorp.com/gpg | gpg --dearmor | sudo tee /usr/share/keyrings/hashicorp-archive-keyring.gpg”.
```

### 3.2.2. Configuración del Repositorio

Agregue el repositorio oficial de HashiCorp a su lista de fuentes de APT con el siguiente comando:

```
“echo “deb [signed-by=/usr/share/keyrings/hashicorp-archive-keyring.gpg] https://apt.releases.hashicorp.com $(lsb_release -cs) main” | sudo tee /etc/apt/sources.list.d/hashicorp.list”.
```

### 3.2.3. Instalación de Terraform

Actualice la lista de repositorios e instale Terraform ejecutando los siguientes comandos:

```
“sudo apt-get update”
```

```
“sudo apt-get install terraform”
```

### 3.2.4. Verificación de la instalación

Verificar que Terraform se haya instalado correctamente ejecutando el siguiente comando

```
“terraform -versión”.
```

## 4. Instalación de Ansible

### 4.1. Actualización del Sistema

Al igual que con Terraform, asegúrese de que el sistema esté completamente actualizado antes de instalar Ansible. Ejecute los siguientes comandos:

***“sudo apt-get update && sudo apt-get upgrade”***

#### **4.2. Añadir el repositorio de Ansible**

Para obtener la última versión de Ansible, es necesario añadir el PPA (Personal Package Archive) oficial. Utilice el siguiente comando para ello:

***“sudo apt-add-repository --yes --update ppa:ansible/ansible”***

#### **3.3. Instalación Ansible**

Una vez añadido el repositorio, proceda a instalar Ansible con el siguiente comando:

***“sudo apt-get install ansible”***

Finalmente, verifique la instalación de Ansible:

***“ansible --version”***

## Anexo D

### *Manual de creación de clave SSH*

#### 1. Introducción

Este manual detalla el proceso de creación de una clave SSH, una herramienta fundamental para establecer conexiones seguras entre sistemas mediante el protocolo SSH. Estas conexiones son esenciales para la administración remota y la configuración automatizada de máquinas, especialmente en entornos de despliegue continuo y seguro, como servidores, contenedores y máquinas virtuales. Las claves SSH permiten autenticación segura y eliminan la necesidad de ingresar contraseñas de manera constante y mejora la seguridad.

#### 2. Requisitos Previos

Antes de iniciar el proceso de creación de la clave SSH, asegúrate de cumplir con los siguientes requisitos:

- Sistema operativo Linux, en este caso el manual utiliza Kali Linux ya que es compatible con SSH y tiene las herramientas de IaC previamente instaladas.
- Acceso a la terminal en el sistema donde se generará la clave SSH.
- Software OpenSSH: El cliente OpenSSH debe estar instalado.

#### 3. Creación de clave SSH

El proceso de creación de una clave SSH implica la generación de un par de claves, una clave pública y una clave privada. La clave privada debe mantenerse segura, mientras que la clave pública se comparte con el servidor o dispositivo al que deseas acceder. Para ello se debe seguir los siguientes pasos:

- 3.1 Generar el par de claves SSH: Para ello desde el terminal se ejecuta el siguiente comando “`ssh-keygen -t rsa -b 4096 -C correo@example.com`”.

Donde:

- -t rsa: Especifica que se utiliza el algoritmo RSA para la generación de la clave.
- -b 4096: Indica que la clave tendrá un tamaño de 4096 bits.
- -C correo@example.com: Proporciona un comentario asociado con la clave, generalmente tu correo electrónico.

3.2 Ubicación del archivo de clave: Después de ejecutar el comando, se te pedirá que elijas una ubicación para guardar la clave generada.

3.3 Configuración de una contraseña para la clave: Después de definir la ubicación del archivo, se te preguntará si se desea agregar una contraseña para la clave privada. Esto proporciona una capa adicional de seguridad.

3.4 Verificación de las claves generadas: Una vez generadas, se verifica que las claves se encuentran en el directorio `~/.ssh` con el siguiente comando “`ls ~/.ssh`”, como se muestra en la siguiente figura.

**Figura 54**  
Claves SSH

```
lizbeth@lizbeth-virtual-machine:~$ ls ~/.ssh
id_rsa  id_rsa.pub
```

*Nota:* La figura se observa la creación de la clave pública y privada.

#### 4. Instalación del Cliente OpenSSH

El cliente OpenSSH debe estar instalado en el sistema local o en la máquina desde la cual se establecerán las conexiones SSH. Esto es fundamental para que el cliente SSH envíe solicitudes de conexión a las máquinas virtuales o servidores que se desean administrar.

Para instalar el cliente OpenSSH en sistemas basados en Debian (como Kali Linux), utiliza el siguiente comando en la terminal:

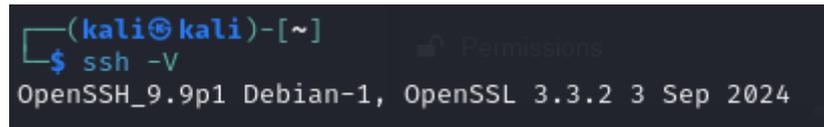
### “sudo apt-get install openssh-client”

Una vez completada la instalación, puedes verificar que el cliente OpenSSH se haya instalado correctamente ejecutando como se muestra en la figura.

### “ssh -V”

**Figura 55**

*Verificación de la Instalación del Cliente OpenSSH en Kali Linux*



```
(kali@kali)-[~]
└─$ ssh -V
OpenSSH_9.9p1 Debian-1, OpenSSL 3.3.2 3 Sep 2024
```

*Nota:* La figura muestra la versión de OpenSSH y OpenSSL instalados en el sistema, confirmando que el cliente SSH está correctamente configurado y listo para su uso.

## 5. Instalación de servidor OpenSSH

El servidor OpenSSH permite que una máquina acepte conexiones SSH, lo cual es fundamental para la administración remota de sistemas, especialmente en entornos donde se gestionan servidores, contenedores o máquinas virtuales de manera centralizada.

Para instalar el servidor OpenSSH en la máquina de gestión, se ejecuta el siguiente comando en la terminal:

### “sudo apt-get install openssh-server”

Una vez instalada la herramienta, verifica que el servidor SSH esté en funcionamiento ejecutando el comando:

### "sudo systemctl status ssh"

### *Configuración del servidor SSH*

Después de la instalación, se personaliza la configuración del servidor SSH. Esto incluye definir el puerto de conexión y habilitar la autenticación mediante clave pública.

Para configurar estos ajustes, abre el archivo de configuración con:

**“sudo nano /etc/ssh/sshd\_config”**

Después de realizar los cambios y guarda el archivo, se reinicia el servidor SSH para que los cambios surtan efecto:

**“sudo systemctl restart ssh”**

Finalmente, para verificar que el servidor SSH está funcionando correctamente, intenta conectarte desde otra máquina utilizando el siguiente comando:

**“ssh usuario@ip\_del\_servidor”**

## **6. Agregar la clave pública en la máquina de destino**

Para habilitar el acceso sin contraseña, la clave pública generada en la máquina local debe estar presente en el archivo “**~/.ssh/authorized\_keys**” de cada máquina de destino (es decir, la máquina que se gestionará).

Para copiar la clave pública a la máquina de destino, se ejecuta el siguiente comando:

**“ssh-copy-id usuario@ip\_del\_servidor”**

Este comando añade la clave pública automáticamente al archivo “**~/.ssh/authorized\_keys**” de la máquina de destino, lo que permite la autenticación sin contraseña.

## **Anexo E**

### ***Manual de creación de máquina virtual plantilla en Proxmox***

#### **Introducción**

Este manual describe los pasos para crear una máquina virtual (VM) en Proxmox, que puede ser utilizada para desplegar rápidamente nuevas máquinas virtuales con configuraciones predefinidas.

#### **Requisitos Previos**

- Acceso al servidor Proxmox a través de la interfaz web.
- Imagen ISO del sistema operativo deseado cargada en el almacenamiento del servidor.

#### **Pasos para la creación de la Máquina Virtual**

##### **1. Crear una Nueva Máquina Virtual**

Para comenzar, sigue estos pasos:

- Ingresa a la interfaz web de Proxmox.
- Haz clic en el botón Create VM, y completa los parámetros iniciales:
  - Node: Selecciona el nodo donde deseas crear la VM.
  - VM ID: Puedes usar el ID sugerido o asignar uno personalizado.
  - Name: Ingresa un nombre descriptivo para identificar la VM.
- Haz clic en Next para continuar con el proceso.

##### **2. Seleccionar el Medio de Instalación**

- En la sección OS:

- Elige el almacenamiento que contiene la imagen ISO.
- Selecciona la ISO del sistema operativo que deseas instalar.

### **3. Configurar el Hardware de la VM**

Configura los recursos y componentes de la máquina virtual según tus necesidades:

#### 1. System:

- Configura el tipo de BIOS.
- Habilita la opción "Qemu Agent" para una mejor integración con Proxmox.

#### 2. Disco Duro:

- Define el tipo de almacenamiento y el tamaño del disco.

#### 3. CPU:

- Asigna la cantidad de núcleos necesarios para la VM.

#### 4. Memory:

- Configura la cantidad de memoria RAM.

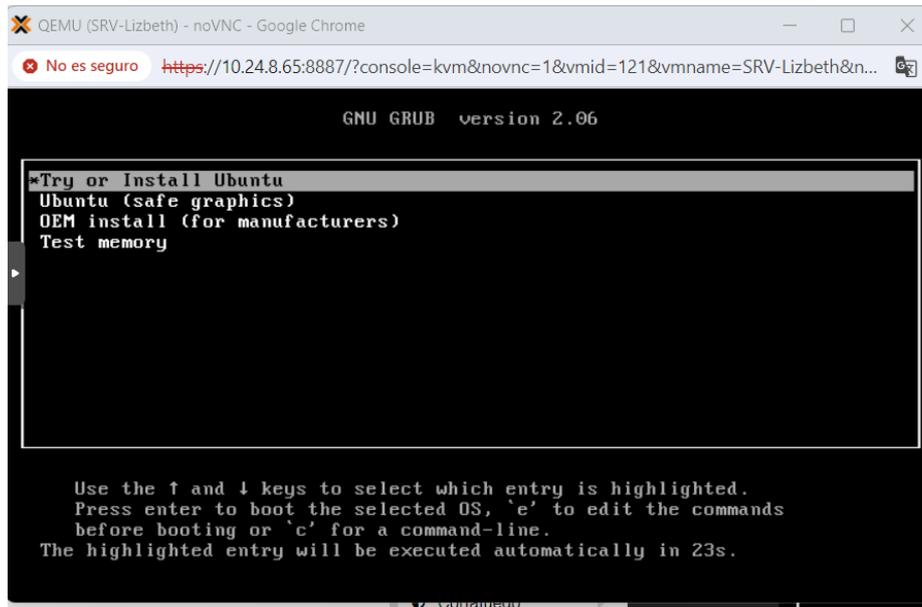
#### 5. Network:

- Selecciona el adaptador de red (VirtIO para un mejor rendimiento).

### **4. Instalar el Sistema Operativo**

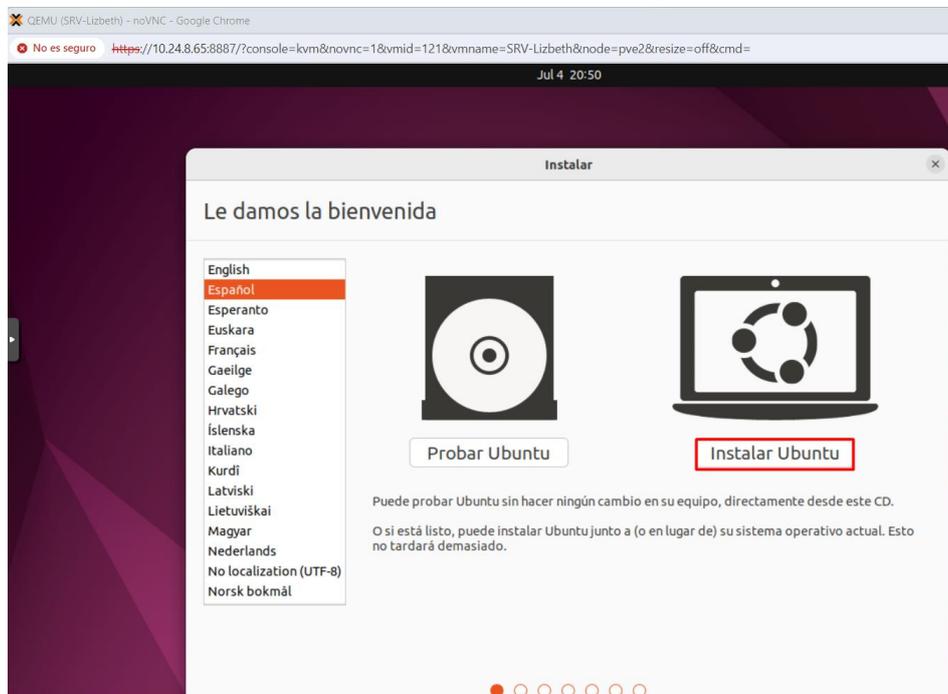
- Inicia la máquina virtual desde la interfaz de Proxmox y accede a su consola.

**Figura 56**  
Consola de MV en Proxmox



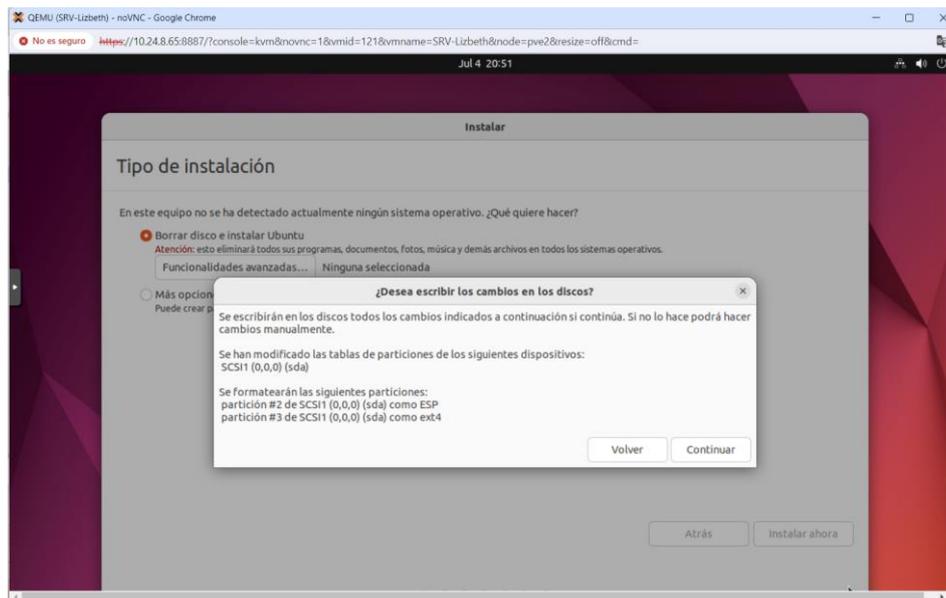
- Se procede a instalar el sistema operativo en este caso se utilizó Ubuntu 22.04, para ello selecciona el idioma, la zona horaria y la distribución del teclado.

**Figura 57**  
Configuración de idioma y teclado



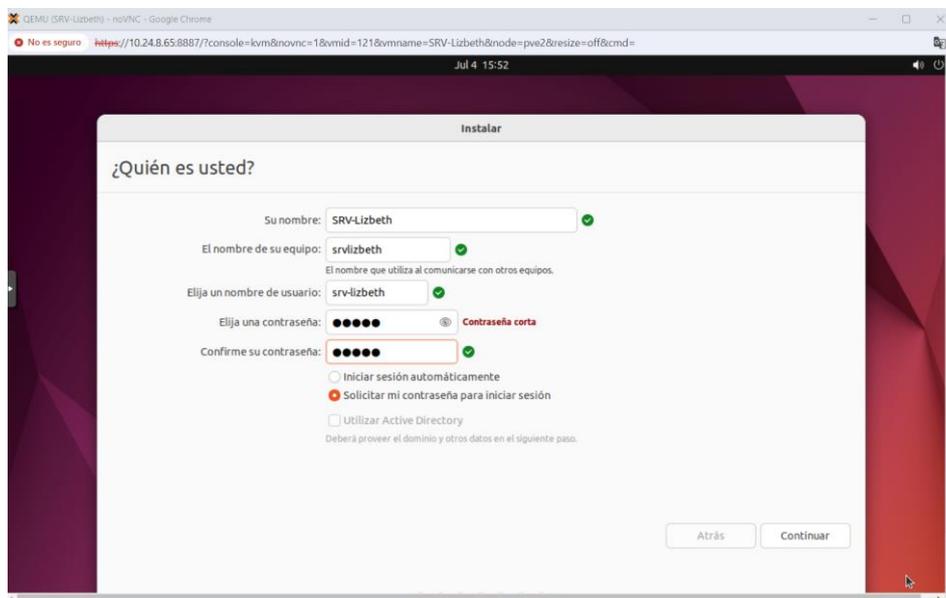
- Configura la partición del disco

**Figura 58**  
Partición del disco



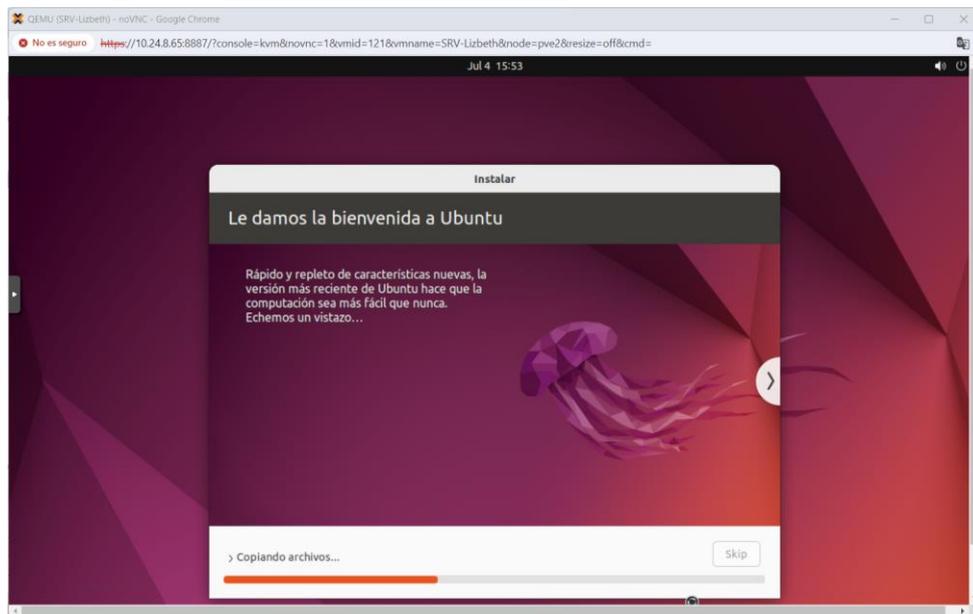
- Crea un usuario administrador e ingresa las credenciales necesarias.

**Figura 59**  
Creación de usuario



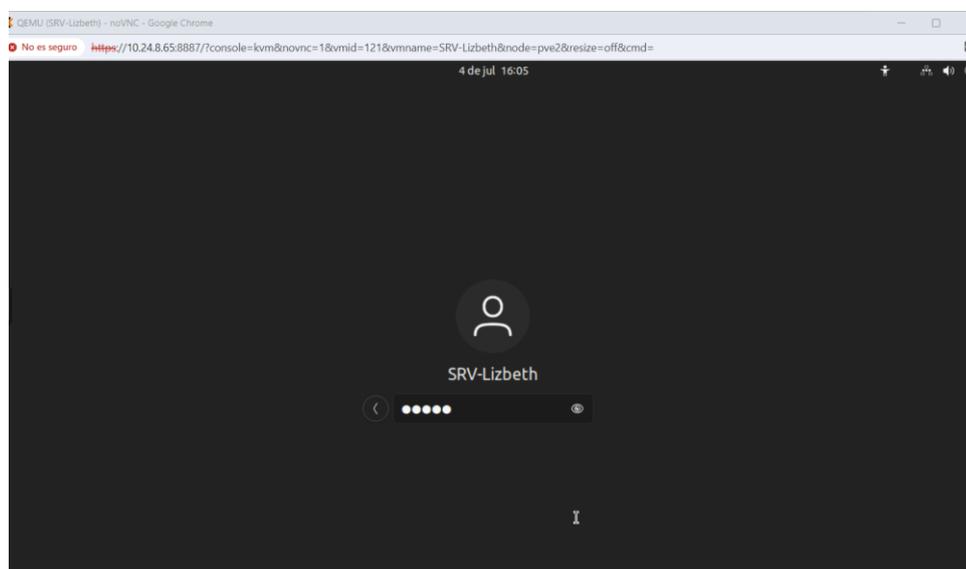
- Confirma la instalación y espera a que el proceso finalice.

**Figura 60**  
*Instalación*



- Una vez instalada, reinicia la máquina virtual y verifica que todo funcione correctamente.

**Figura 61**  
*Máquina virtual*



## **Anexo F**

### ***Manual de Despliegue de Infraestructura en Proxmox utilizando Terraform y Ansible***

#### **Introducción**

Este manual detalla el proceso para desplegar infraestructura en Proxmox utilizando herramientas de Infraestructura como Código (IaC), específicamente Terraform para la provisión de máquinas virtuales y Ansible para su configuración y aprovisionamiento.

#### **Requisitos Previos**

- Acceso a Proxmox con permisos suficientes para crear y administrar máquinas virtuales.
- Conocimientos básicos de Terraform y Ansible.
- Sistema operativo Linux o compatible con las herramientas de automatización.
- Claves SSH generadas para el acceso remoto a las máquinas virtuales.

#### **Pasos para el despliegue de infraestructura**

##### **1. Configuración Inicial**

Se inicia con la preparación del entorno:

- Instala Terraform y Ansible en tu máquina de administración como se muestra en el Anexo C.
- Asegúrate de que Proxmox VE esté configurado y accesible.
- Verifica que tengas acceso SSH a los nodos y a las plantillas necesarias.

Crear un usuario API en Proxmox:

- Crea un usuario con privilegios adecuados y un rol con permisos API.
- Genera un token de API para la autenticación.

Configurar la máquina plantilla en Proxmox:

- Prepara una máquina virtual base como Ubuntu 22.04 u otro sistema operativo, según tus necesidades.
- Instala paquetes básicos necesarios, habilita SSH y configura una red básica.

## 2. Configuración de Terraform

Instalar el proveedor de Proxmox:

- Asegúrate de tener el archivo de configuración del proveedor terraform-provider-proxmox.

Definir variables:

- Crea un archivo variables.tf para almacenar configuraciones reutilizables, como:

**Figura 62**  
*Código de variables*

```
GNU nano 8.2      var.tf
variable "proxmox_host" {
  type = map
  default = {
    pm_api_url = "https://10.0.0.128:8006/api2/json"
    pm_user = "root@pam"
    target_node = "pv5"
  }
}

variable "api_token_secret" {
  description = "The secret for the API token"
  type = string
  sensitive = true
}

variable "vmid" {
  default = 503
  description = "Starting ID for the CTs"
}
```

Configurar el archivo main.tf:

- Escribe la configuración básica para crear una máquina virtual:

**Figura 63**

*Código principal*

```
GNU nano 8.2                               main.tf
provider "proxmox" {
  pm_api_url = var.proxmox_host["pm_api_url"]
  pm_user    = var.proxmox_host["pm_user"]
  pm_password = var.ssh_password
  pm_tls_insecure = true
}

resource "proxmox_vm_qemu" "prod-vm" {
  count = length(var.hostnames)
  name = var.hostnames[count.index]
  target_node = var.proxmox_host["target_node"]
  vmid = var.vmid + count.index
  full_clone = true
  clone = "SRV-Lizbeth"

  cores = 3
  sockets = 2
  vcpus = 6
  memory = 12288
  balloon = 12288
  boot = "c"
  bootdisk = "virtio0"
}
```

Inicializar y ejecutar:

- Inicializa Terraform:

**Figura 64**

*Inicialización de Terraform*

```
(kali@kali)-[~/proxmox-automation/tf/nw.d.terraform.tfstate.d/new-mv]
└─$ terraform init

Initializing the backend ...

Initializing provider plugins ...
- Reusing previous version of telmate/proxmox from the dependency lock file
- Using previously-installed telmate/proxmox v2.9.14

Terraform has been successfully initialized!

You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.

If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

- Aplica los cambios para desplegar la infraestructura:

**Figura 65**  
Despliegue de infraestructura

```
(kali@kali)-[~/proxmox-automation/terraform.tfstate.d/new_vm_workspace]
└─$ terraform apply
var.api_token_secret
  The secret for the API token

  Enter a value:

var.ssh_password
  Password for SSH

  Enter a value:

Terraform used the selected providers to generate the following execution
plan. Resource actions are indicated with the following symbols:
+ create

Terraform will perform the following actions:

# proxmox_vm_qemu.prod-vm[0] will be created
+ resource "proxmox_vm_qemu" "prod-vm" {
  + additional_wait = 5
  + agent           = 1
```

### 3. Configuración con Ansible

Definir el inventario:

- Utiliza un script para generar un inventario basado en las VMs creadas por Terraform.

**Figura 66**  
Inventario

```
GNU nano 8.2 /etc/ansible/hosts
[servers]
10.0.0.140
10.0.0.145
10.0.0.150
10.0.0.155
10.0.0.160
10.0.0.165
10.0.0.170
10.0.0.175

[servers:vars]
ansible_user=srv-lizbeth
ansible_ssh_private_key_file=~/.ssh/terraform_key
```

Crear de archivo playbook.yml:

- Define las tareas para configurar las máquinas creadas:

**Figura 67**  
Playbook para actualizar mv

```
GNU nano 8.2          upgrade.yaml
- hosts: lxc-all
  become: yes
  tasks:
    - name: Update and upgrade apt packages
      apt:
        upgrade: dist
        update_cache: yes
        cache_valid_time: 86400 # One day
    # Tarea para limpiar paquetes innecesarios
    - name: Remove unnecessary packages
      apt:
        autoremove: yes
```

## 2. Configurar Ansible para ejecutar el playbook:

- Llama al inventario y aplica el playbook:

**Figura 68**  
Ejecución de playbook

```
(kali@kali)-[~/proxmox-automation/ansible]
└─$ ansible-playbook -i /etc/ansible/hosts upgrade.yaml --key-file ~/.ssh/terraform_key -u srv-lizbeth

[WARNING]: Could not match supplied host pattern, ignoring: lxc-all

PLAY [lxc-all] *****
***
skipping: no hosts matched

PLAY RECAP *****
***
```

## Anexo G

### *Guía de Laboratorio 1: Red MPLS de transporte con servicios*

#### INGENIERÍA EN TELECOMUNICACIONES GUÍA DE LABORATORIO DE INGENIERIA DE TRAFICO

**ESTUDIANTE:**

**PERIODO ACADÉMICO:**

Nombre de la práctica:

MPLS

Objetivo(s) de la práctica:

- Adquirir conocimientos sobre los principios básicos de Calidad de Servicio (QoS), incluyendo su importancia en la gestión del tráfico de red para garantizar el rendimiento óptimo de los servicios críticos.
- Configurar y aplicar políticas de QoS en diversos dispositivos de red.
- Clasificar y marcar diferentes tipos de tráfico de red, lo que permite una gestión diferenciada del tráfico basada en la importancia y los requisitos de los datos transmitidos.
- Configurar la gestión de colas y el control de tráfico.
- Realizar pruebas de tráfico y análisis de desempeño

Materiales y equipos

- Computados personal o de laboratorio.
  - Software de simulación GNS3.

Procedimiento experimental

Parte 1:

#### 1. Configuraciones Básicas

- Configurar el nombre en cada router según el diagrama de red proporcionado.
- Configurar las direcciones IP en las interfaces de cada router según el direccionamiento asignado.
- Comprobar las configuraciones en cada router para asegurar que sean correctas.

#### 2. Seguridad y Conectividad

- Configurar la seguridad en los puertos.
- Configurar la seguridad en los equipos, incluyendo las clases de consola y las líneas VTY.
- Configurar EtherChannel para mejorar la redundancia y el rendimiento.

- Configurar VLANs para segmentar la red y mejorar la seguridad y eficiencia.

## Parte 2: Configuración de MPLS

### 1. Configuración de MPLS

- Configurar MPLS básico en los routers.
- Configurar los clientes A y B.
- Probar el correcto funcionamiento de la nube MPLS.
- Analizar las tablas LFIB, LIB y FIB para comprender el enrutamiento MPLS.

### 2. Configuraciones Avanzadas

- Configurar un firewall utilizando pfSense o iptables para proteger la red.
- Configurar servicios adicionales como FTP, base de datos, DNS, web, VoIP y streaming para proporcionar funcionalidad adicional a la red.

## Parte 3: Configuración de QoS en Routers

### 1. Habilitación de QoS

- Acceder a la configuración de los routers y habilitar QoS en las interfaces correspondientes.

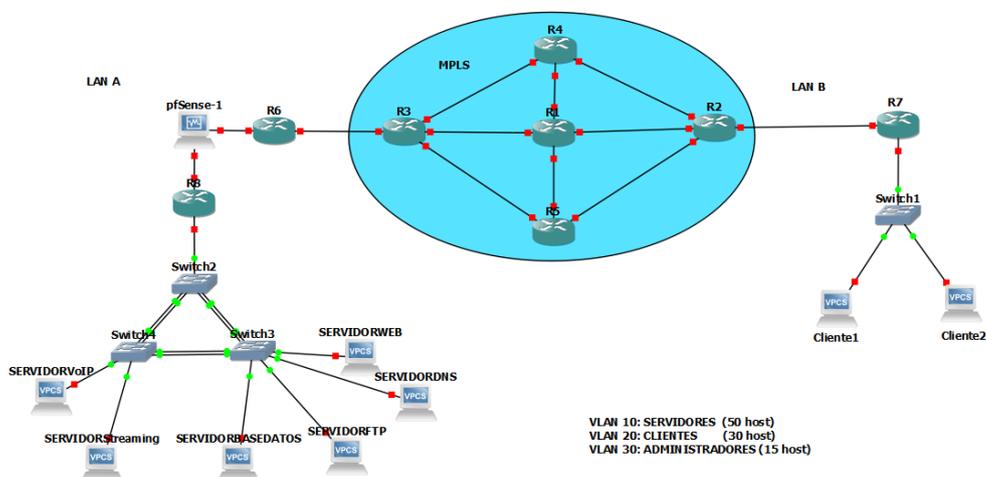
### 2. Clasificación y Marcación de Tráfico

- Configurar las clases de tráfico en el router para clasificar diferentes tipos de tráfico.
- Crear políticas de marcación para marcar los paquetes de acuerdo con las clases de tráfico.

### 3. Gestión de Tráfico

- Configurar colas de salida para gestionar el tráfico de manera eficiente y asegurar la calidad del servicio.

## TOPOLOGIA:



## Resultados

Realizar un informe técnico.

El informe técnico debe contener los siguientes elementos:

### 1. Objetivos de la Práctica:

- Definir claramente los propósitos de la práctica.

### 2. Marco Teórico:

- Proporcionar una base conceptual sólida sobre los temas tratados en la práctica, como MPLS, QoS, seguridad en red, y otros conceptos relevantes.
- Incluir definiciones, principios fundamentales, y una revisión de la literatura pertinente.

### 3. Procedimiento:

- Detallar minuciosamente cada paso realizado para alcanzar los objetivos de la práctica.
- Incluir capturas de pantalla, configuraciones de dispositivos, comandos utilizados y cualquier otra información visual que facilite la comprensión del proceso.
- Asegurar que cada procedimiento esté bien documentado para que pueda ser replicado fácilmente por otros.

### 4. Resultados Obtenidos:

- Presentar de manera clara y concisa los resultados alcanzados durante la práctica.
- Incluir análisis de rendimiento, tablas de enrutamiento, etiquetas de tráfico.

### 5. Conclusiones:

- Realizar un análisis crítico de los resultados obtenidos.

### 6. Recomendaciones:

- Proponer mejoras o ajustes que podrían optimizar los procedimientos o resultados futuros.

### 7. Bibliografía

## ***Guía de Laboratorio 2: Monitoreo de Red con Zabbix***

### **INGENIERÍA EN TELECOMUNICACIONES GUÍA DE LABORATORIO DE ADMINISTRACIÓN DE REDES DE TELECOMUNICACIONES**

**ESTUDIANTE:**

**PERIODO ACADÉMICO:**

Nombre de la práctica:

Monitorización de dispositivos de red con Zabbix

Objetivo(s) de la práctica:

1. Zabbix para la monitorización de la red.
2. Implementar una topología de red con los dispositivos especificados.
3. Configurar y asegurar la conectividad entre todos los dispositivos de la red.
4. Monitorizar el rendimiento y el estado de los dispositivos de red utilizando Zabbix.
5. Analizar los datos de monitorización y generar informes de estado y rendimiento.

Preparatorio

- Un servidor con un sistema operativo compatible (Rocky Linux).
- Acceso de administrador (root o con privilegios sudo) en el servidor.
- Conexión a internet.

Materiales y equipos

- Computador personal o de laboratorio.
  - Virtualizado de máquinas.

Procedimiento experimental

Parte 1: Configuraciones Iniciales

#### *Configuraciones básicas*

- Asignar nombres a los routers y switches según la topología.
- Configurar las direcciones IP en las interfaces de cada dispositivo de acuerdo con el direccionamiento asignado.
- Verificar la conectividad entre los dispositivos.

#### *Configuración de NAT y Enrutamiento*

- Configurar NAT en el router conectado a Internet.
- Configurar enrutamiento para asegurar la conectividad entre subredes.
- Realizar ping y traceroute entre los dispositivos para asegurar que la configuración es correcta.

## Parte 2: Configuración e Instalación de Zabbix

### Instalación del Servidor Zabbix en el Servidor Zabbix

- Instalar Rocky Linux en una máquina virtual para el Servidor Zabbix.
- Instalar PostgreSQL y configurar la base de datos para Zabbix.
- Instalar Zabbix Server, Frontend y Agent.

### Configuración del Servidor Zabbix

- Configurar la base de datos Zabbix.
- Editar los archivos de configuración de Zabbix Server y Frontend.

### Acceso a la Interfaz Web de Zabbix

- Acceder a la interfaz web de Zabbix desde un navegador.
- Completar la configuración inicial a través del asistente web.

### Instalación del Agente Zabbix en los Servidores

- Instalar Zabbix Agent en los servidores de la red.

### Configuración del Monitoreo en la Interfaz Web de Zabbix

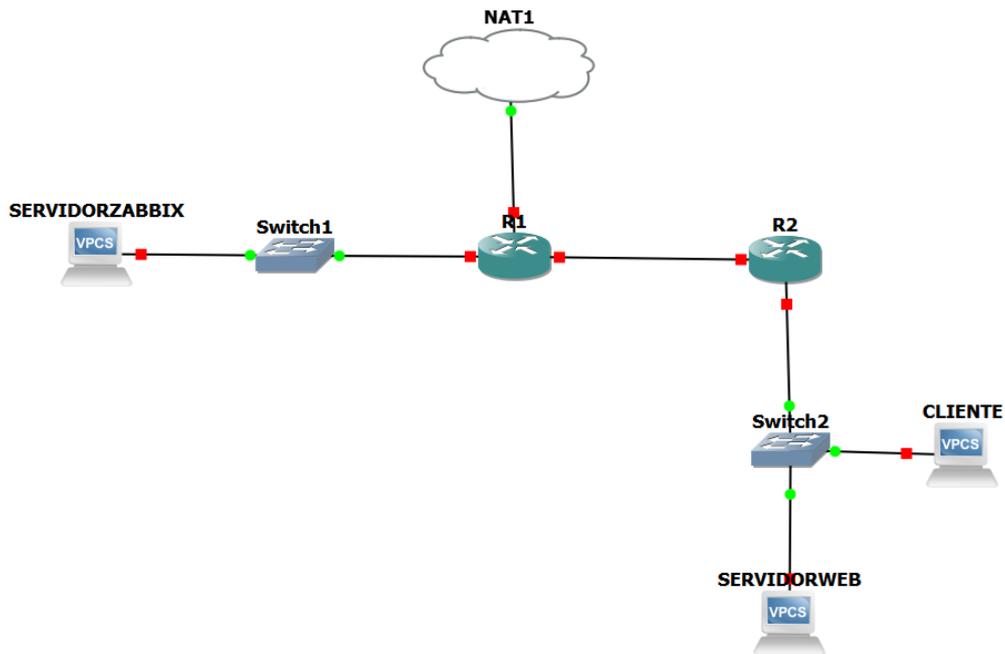
- Agregar los dispositivos de red como hosts en Zabbix.
- Configurar los templates y elementos de monitoreo necesarios para cada dispositivo.

### Análisis de Datos

- Analizar los datos de monitorización para identificar posibles problemas y áreas de mejora.

### Topología de red

La topología de red a implementar es la siguiente:



## Resultados

- Detallar los objetivos logrados durante el laboratorio.
- Explicar los conceptos de monitorización de red y la importancia de Zabbix.
- Describir paso a paso los procedimientos realizados, incluyendo configuraciones y comandos utilizados.
- Documentar los resultados obtenidos, incluyendo capturas de pantalla de configuraciones y verificaciones.
- Analizar los resultados y reflexionar sobre el funcionamiento de la red monitorizada.
- Proponer mejoras o ajustes para optimizar la configuración.
- Referenciar todas las fuentes y materiales utilizados.

## Referencias

Documentación oficial de Zabbix: <https://www.zabbix.com/documentation/current/en/>

### *Guía de Laboratorio 3: Gestión de Redes con Zabbix*

## INGENIERÍA EN TELECOMUNICACIONES GUÍA DE LABORATORIO DE ADMINISTRACIÓN DE REDES DE TELECOMUNICACIONES

**ESTUDIANTE:**

**PERIODO ACADÉMICO:**

Nombre de la práctica:

Monitorización de redes con Zabbix

Objetivo(s) de la práctica:

- Configurar el servidor Zabbix y los agentes en los dispositivos de red.
- Integrar Zabbix con los dispositivos de red para la monitorización.
- Configurar los elementos de datos y los triggers en Zabbix.
- Implementar gráficos y mapas de red en Zabbix.
- Realizar pruebas de tráfico y análisis de desempeño para evaluar la efectividad de la monitorización.

Materiales y equipos

- Computador personal o de laboratorio.
  - Virtualizado de máquinas.

Procedimiento experimental

### **Parte 1: Configuraciones Básicas**

#### **Configuración de básicas**

- Configurar el nombre en cada router según el diagrama de red proporcionado.
- Configurar las direcciones IP en las interfaces de cada router según el direccionamiento asignado.
- Comprobar las configuraciones en cada router para asegurar que sean correctas.

#### **Configuración de la seguridad en los puertos**

- Configurar la seguridad en los equipos, incluyendo las clases de consola y las líneas VTY.
- Configurar EtherChannel para mejorar la redundancia y el rendimiento.
- Configurar VLANs para segmentar la red y mejorar la seguridad y eficiencia.

#### **Configuración de NAT y Enrutamiento**

- Configurar enrutamiento para asegurar la conectividad entre subredes.

- Realizar ping y traceroute entre los dispositivos para asegurar que la configuración es correcta.

### **Configuración de servidores**

- Configurar servicios adicionales como FTP, base de datos, DNS, web, VoIP y streaming para proporcionar funcionalidad adicional a la red.

### Parte 2: Configuración de Zabbix

#### **Configuración del agente Zabbix**

- Iniciar y habilitar el servidor Zabbix Agent
- Integración y configuración en Zabbix
- Agregar dispositivos de red a Zabbix
- Configurar elementos de datos y triggers

#### **Implementación de gráficos y mapas de red**

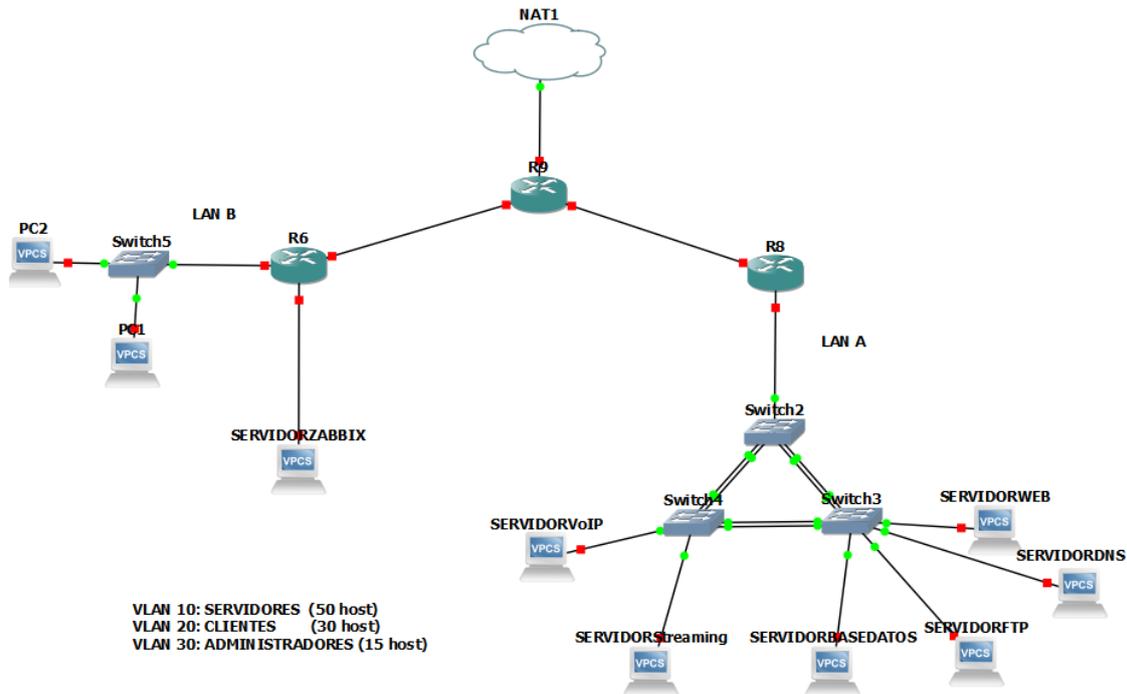
- Crear gráficos en Zabbix
- Crear mapas de red en Zabbix

#### **Pruebas de monitorización y análisis de resultados**

- Realizar pruebas de monitorización
- Analizar los resultados en Zabbix

### Topología de red

La topología de red a implementar es la siguiente:



## Resultados

- El servidor Zabbix debe estar correctamente configurado y operando.
- Los dispositivos de red deben estar integrados en Zabbix y enviando datos de monitorización.
- Los elementos de datos y triggers deben estar configurados y funcionando correctamente.
- Los gráficos y mapas de red deben mostrar información en tiempo real sobre el estado de la red.
- Las pruebas de monitorización deben confirmar que Zabbix está recolectando y presentando los datos adecuadamente.

## Referencias

Documentación oficial de Zabbix:

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

**Anexo H****UNIVERSIDAD TÉCNICA DEL NORTE****FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS****Carrera de Ingeniería en Telecomunicaciones**

**Proyecto:** Implementación y automatización de infraestructura basada en código para los laboratorios del área de redes de la carrera de Telecomunicaciones en la Universidad Técnica del Norte.

**Objetivo:** Esta encuesta tiene como objetivo evaluar el impacto de la Infraestructura Basada en Código (IaC) en el desarrollo, configuración y aprovechamiento de los laboratorios del área de redes en la carrera de Telecomunicaciones. La finalidad es determinar si la implementación de IaC facilita el aprendizaje práctico de los estudiantes, mejora la disponibilidad y accesibilidad de los recursos de laboratorio, y optimiza tanto el tiempo como la precisión en la configuración de los entornos educativos. Además, se busca conocer la percepción de los estudiantes sobre cómo IaC contribuye a una experiencia de aprendizaje más eficiente y efectiva.

**Encuesta sobre el Impacto de la Implementación y Automatización de  
Infraestructura Basada en Código (IaC) en los Laboratorios de Redes**

**Instrucciones:** Por favor, marque la respuesta que mejor refleje su experiencia.

1. ¿Considera que la automatización de infraestructura mediante IaC facilita el proceso de configuración inicial de los laboratorios de redes?

- Muy de acuerdo
- De acuerdo
- Neutral
- En desacuerdo
- Muy en desacuerdo

2. ¿La implementación de IaC asegura que todos los entornos de laboratorio se configuren de manera uniforme y consistente?

- Muy de acuerdo

- De acuerdo
- Neutral
- En desacuerdo
- Muy en desacuerdo

3. ¿Cree que el uso de IaC reduce significativamente los errores en la configuración de los laboratorios de redes?

- Muy de acuerdo
- De acuerdo
- Neutral
- En desacuerdo
- Muy en desacuerdo

4. ¿Qué tan útil considera la aplicación de IaC en el contexto educativo de los laboratorios de redes?

- Muy útil
- Útil
- Neutral
- Poco útil
- Nada útil

5. ¿El uso de IaC permite que los laboratorios estén listos y disponibles en menor tiempo para los estudiantes?

- Muy de acuerdo
- De acuerdo
- Neutral
- En desacuerdo
- Muy en desacuerdo

6. ¿Considera que la automatización mediante IaC contribuye a una experiencia de aprendizaje más efectiva para los estudiantes en los laboratorios de redes?

- Muy de acuerdo
- De acuerdo
- Neutral
- En desacuerdo
- Muy en desacuerdo

7. ¿La disponibilidad de laboratorios automatizados mediante IaC permite a los estudiantes concentrarse más en el contenido práctico en lugar de la configuración inicial o despliegue de la topología?

- Muy de acuerdo
- De acuerdo
- Neutral
- En desacuerdo
- Muy en desacuerdo

8. ¿Recomendaría la implementación de IaC en otros cursos de laboratorio de la carrera de Telecomunicaciones?

- Definitivamente sí
- Probablemente sí
- No estoy seguro
- Probablemente no
- Definitivamente no

9. ¿Cuánto tiempo promedio se tarda en realizar un laboratorio configurado desde cero (sin IaC)?

- Menos de 1 hora
- 1 a 2 hora
- 2 a 3 horas
- Más de 3 horas

10. ¿Cuánto tiempo promedio se tarda en realizar un laboratorio automatizado mediante IaC?

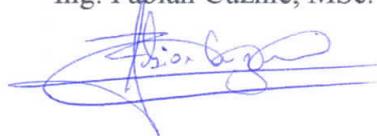
- Menos de 30 minutos
- 30 minutos a 1 hora
- 1 a 2 horas
- Más de 2 horas

Elaborado por:

Lizbeth Espinoza

Aprobado por director:

Ing. Fabián Cuzme, MSc.



## Anexo I

### *Tabulación de la Encuesta sobre el Impacto de IaC en los Laboratorios de Redes*

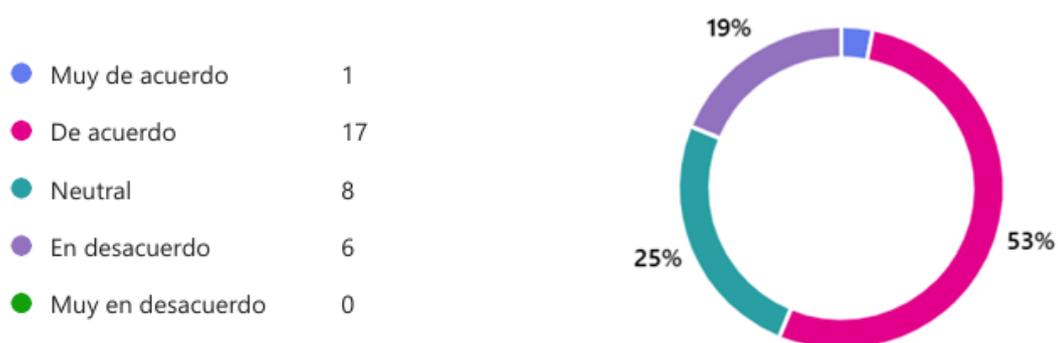
Se presenta la tabulación de las respuestas obtenidas en la encuesta realizada.

1. ¿Considera que la automatización de infraestructura mediante IaC facilita el proceso de configuración inicial de los laboratorios de redes?

**Tabla 17**  
*Tabulación encuesta pregunta 1*

<b>Respuesta</b>	<b>Frecuencia</b>	<b>%</b>
<b>Muy de acuerdo</b>	1	3%
<b>De acuerdo</b>	17	53%
<b>Neutral</b>	8	25%
<b>En desacuerdo</b>	6	19%
<b>Muy en desacuerdo</b>	0	0%
<b>Total</b>	32	100%

**Figura 69**  
*Gráfico pregunta 1*



Los datos de la tabla 16 y la figura 68 muestran que el 56% de los encuestados perciben que IaC facilita significativamente la configuración inicial de los laboratorios, destacando su impacto positivo en el ámbito educativo al mejorar los procesos y optimizar tiempos. Aunque un 19% manifestó cierto desacuerdo, esto podría atribuirse a

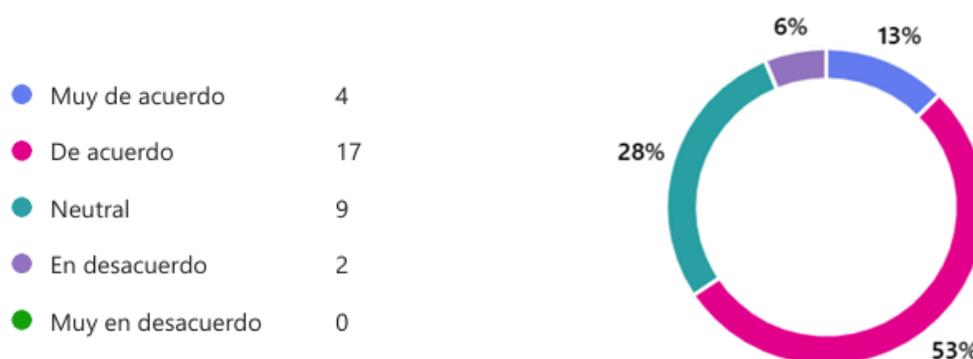
desafíos técnicos, como la falta de recursos del servidor, cortes de energía o problemas de conectividad. En general, la percepción es favorable, consolidando a IaC como una herramienta valiosa para aportar eficiencia y consistencia en los laboratorios académicos.

2. ¿La implementación de IaC asegura que todos los entornos de laboratorio se configuren de manera uniforme y consistente?

**Tabla 18**  
*Tabulación encuesta pregunta 2*

<b>Respuesta</b>	<b>Frecuencia</b>	<b>%</b>
<b>Muy de acuerdo</b>	<b>4</b>	<b>12%</b>
<b>De acuerdo</b>	<b>17</b>	<b>53%</b>
<b>Neutral</b>	<b>9</b>	<b>28%</b>
<b>En desacuerdo</b>	<b>2</b>	<b>6%</b>
<b>Muy en desacuerdo</b>	<b>0</b>	<b>0%</b>
<b>Total</b>	<b>32</b>	<b>100%</b>

**Figura 70**  
*Gráfico pregunta 2*



Los datos presentados en la tabla 17 y la figura 69 indica que el 66% de los encuestados perciben que IaC garantiza una configuración uniforme, validando su utilidad para mantener consistencia en entornos de laboratorio. Este resultado valida el papel de IaC

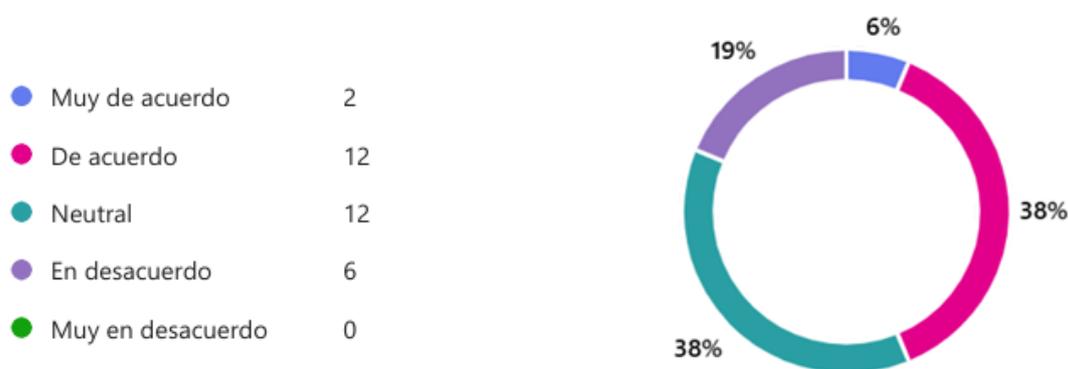
como una herramienta eficaz para asegurar la estandarización en el despliegue de infraestructura.

3. ¿Cree que el uso de IaC reduce significativamente los errores en la configuración de los laboratorios de redes?

**Tabla 19**  
Tabulación encuesta pregunta 3

Respuesta	Frecuencia	%
Muy de acuerdo	2	6%
De acuerdo	12	37%
Neutral	12	37%
En desacuerdo	6	19%
Muy en desacuerdo	0	0%
<b>Total</b>	<b>32</b>	<b>100%</b>

**Figura 71**  
Gráfico pregunta 3



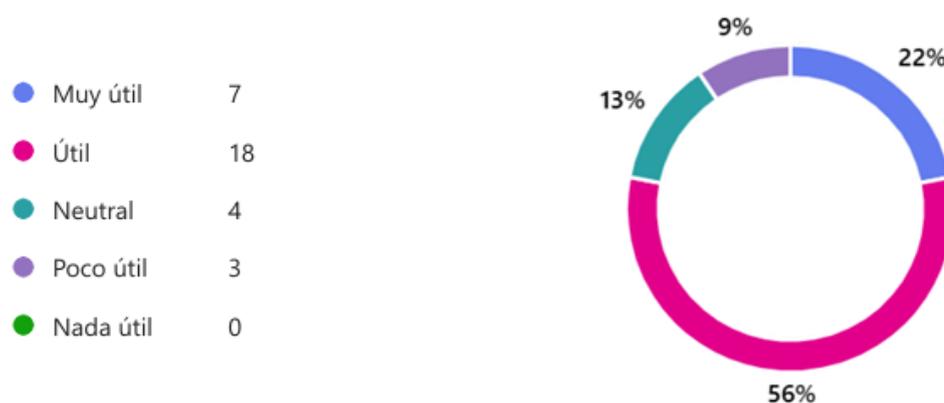
Los datos presentados en la tabla 18 y la figura 70 muestran que el 43% de los encuestados considera que IaC ayuda a reducir los errores en la configuración. Sin embargo, un 37% se mostró neutral, lo que podría indicar falta de conocimiento o experiencia directa. A pesar de esto, el resultado general sugiere que IaC es percibido como un recurso prometedor para minimizar errores en los laboratorios.

4. ¿Qué tan útil considera la aplicación de IaC en el contexto educativo de los laboratorios de redes?

**Tabla 20**  
Tabulación encuesta pregunta 4

Respuesta	Frecuencia	%
Muy útil	7	22%
Útil	18	56%
Neutral	4	12%
Poco útil	3	9%
Nada útil	0	0%
<b>Total</b>	<b>32</b>	<b>100%</b>

**Figura 72**  
Gráfico pregunta 4



Los datos presentados en la tabla 19 y la figura 71 muestran que el 78% de los encuestados percibe IaC como útil o muy útil en el contexto educativo.

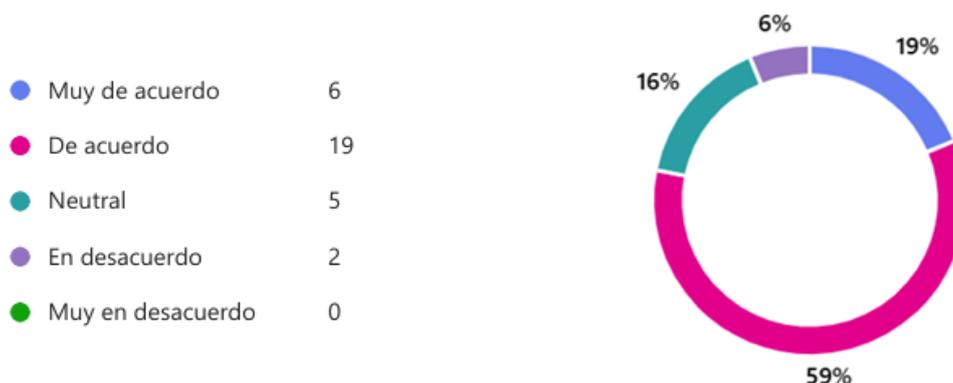
5. ¿El uso de IaC permite que los laboratorios estén listos y disponibles en menor tiempo para los estudiantes?

**Tabla 21**  
Tabulación encuesta pregunta 5

Respuesta	Frecuencia	%
Muy de acuerdo	6	19%

<b>De acuerdo</b>	19	59%
<b>Neutral</b>	5	16%
<b>En desacuerdo</b>	2	6%
<b>Muy en desacuerdo</b>	0	0%
<b>Total</b>	32	100%

**Figura 73**  
Gráfico pregunta 5



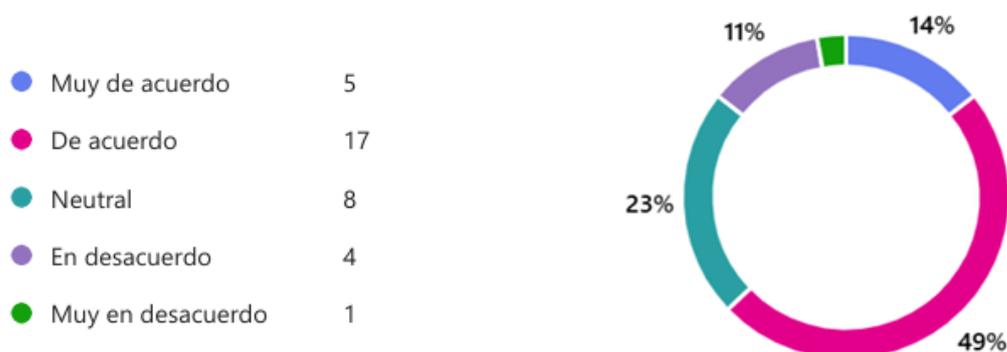
Los datos que se muestran en la tabla 20 y figura 72 muestran que el 78% de los encuestados considera que IaC acelera la disponibilidad de los laboratorios, destacando su efectividad para optimizar tiempos y recursos.

6. ¿Considera que la automatización mediante IaC contribuye a una experiencia de aprendizaje más efectiva para los estudiantes en los laboratorios de redes?

**Tabla 22**  
Tabulación encuesta pregunta 6

<b>Respuesta</b>	<b>Frecuencia</b>	<b>%</b>
<b>Muy de acuerdo</b>	5	16%
<b>De acuerdo</b>	17	53%
<b>Neutral</b>	8	25%
<b>En desacuerdo</b>	4	12%
<b>Muy en desacuerdo</b>	1	3%
<b>Total</b>	32	100%

**Figura 74**  
Gráfico pregunta 6



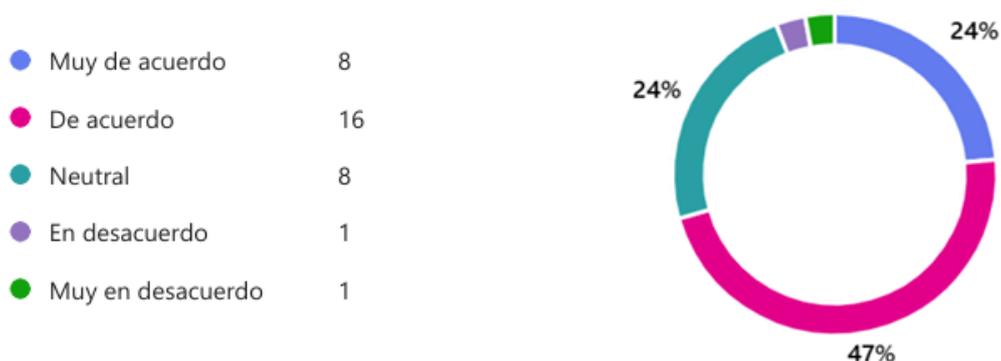
Los datos presentados en la tabla 21 y la figura 73 indican que un 69% de los encuestados considera que IaC contribuye a una experiencia de aprendizaje más efectiva. Sin embargo, un 15% expresó desacuerdo, lo que podría reflejar áreas de mejora en su implementación.

7. ¿La disponibilidad de laboratorios automatizados mediante IaC permite a los estudiantes concentrarse más en el contenido práctico en lugar de la configuración inicial o despliegue de la topología?

**Tabla 23**  
Tabulación encuesta pregunta 7

Respuesta	Frecuencia	%
<b>Muy de acuerdo</b>	8	25%
<b>De acuerdo</b>	16	50%
<b>Neutral</b>	8	25%
<b>En desacuerdo</b>	1	3%
<b>Muy en desacuerdo</b>	1	3%
<b>Total</b>	32	100%

**Figura 75**  
Gráfico pregunta 7



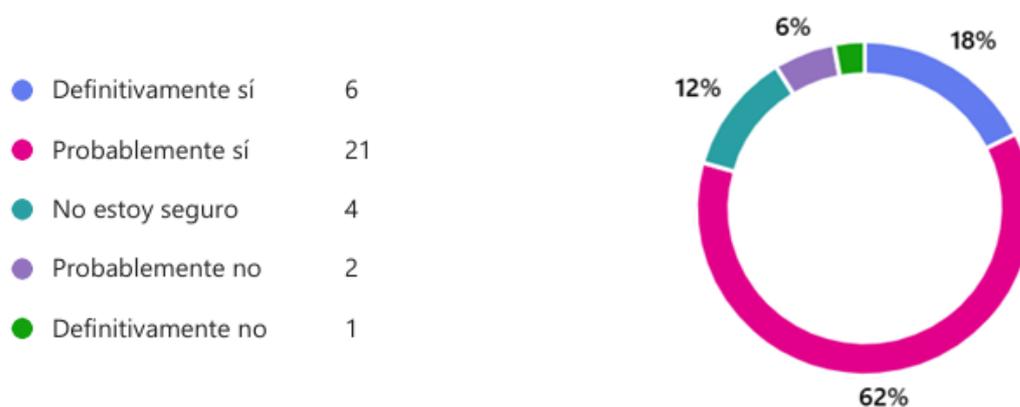
Los datos presentados en la tabla 22 y la figura 74 muestran que el 75% de los encuestados cree que IaC permite a los estudiantes centrarse más en la práctica, destacando su valor educativo al reducir el tiempo que se tarda en realizar la configuración inicial.

8. ¿Recomendaría la implementación de IaC en otros cursos de laboratorio de la carrera de Telecomunicaciones?

**Tabla 24**  
Tabulación encuesta pregunta 8

<b>Respuesta</b>	<b>Frecuencia</b>	<b>%</b>
<b>Definitivamente sí</b>	6	19%
<b>Probablemente sí</b>	21	66%
<b>No estoy seguro</b>	4	13%
<b>Probablemente no</b>	2	6%
<b>Definitivamente no</b>	1	3%
<b>Total</b>	32	100%

**Figura 76**  
Gráfico pregunta 8



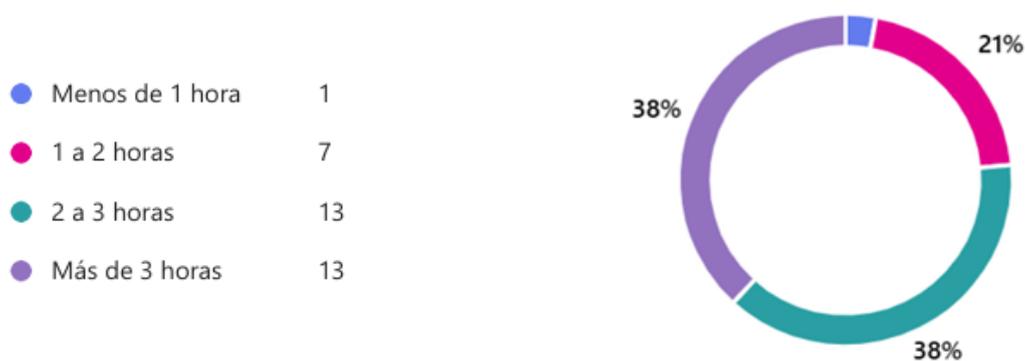
Los datos reflejados en la tabla 23 y la figura 75 destacan que el 85% de los encuestados recomendaría IaC para otros cursos de laboratorio, lo que refleja su aceptación.

9. ¿Cuánto tiempo promedio se tarda en realizar un laboratorio configurado desde cero (sin IaC)?

**Tabla 25**  
Tabulación encuesta pregunta 9

Respuesta	Frecuencia	%
Menos de 1 hora	1	3%
1 a 2 horas	7	22%
2 a 3 horas	13	41%
Más de 3 horas	13	41%
<b>Total</b>	<b>32</b>	<b>100%</b>

**Figura 77**  
Gráfico pregunta 9



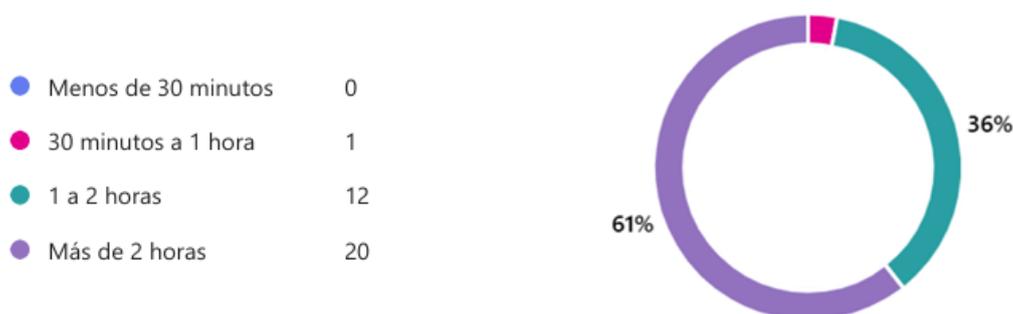
Los datos presentados en la tabla 24 y la figura 76 muestran que el 82% de los encuestados tarda más de dos horas en configurar un laboratorio sin IaC, lo que evidencia la necesidad de optimizar tiempos mediante la automatización.

10. ¿Cuánto tiempo promedio se tarda en realizar un laboratorio automatizado mediante IaC?

**Tabla 26**  
Tabulación encuesta pregunta 10

Respuesta	Frecuencia	%
<b>Menos de 30 minutos</b>	0	0%
<b>30 minutos a 1 hora</b>	1	3%
<b>1 a 2 horas</b>	12	38%
<b>Más de 2 horas</b>	20	59%
<b>Total</b>	32	100%

**Figura 78**  
Gráfico pregunta 10



Los datos presentados en la tabla 25 y la figura 77 evidencian que, aunque IaC logra reducir los tiempos en comparación con los métodos manuales, un 59% de los encuestados aún requiere más de dos horas para completar un laboratorio automatizado. Esto indica que, aunque IaC es una herramienta efectiva, todavía existen áreas de mejora para optimizar su implementación y abordar las dificultades que puedan surgir durante su uso.

## **Anexo J**

### ***Informe de la implementación y despliegue de los laboratorios predefinidos***

#### **Introducción**

En desarrollo del despliegue de laboratorios en el área de redes mediante Infraestructura como Código (IaC), se implementó una prueba con el propósito de evaluar los beneficios de la Infraestructura como Código (IaC). Esta prueba se centró en el despliegue de laboratorios predefinidos mediante el uso de herramientas como Terraform y Ansible, lo que permitió analizar y comparar el desempeño de métodos de configuración manual y automatizada.

#### **Objetivo**

Evaluar la eficiencia, consistencia y precisión en el despliegue de entornos de laboratorio mediante el uso de Infraestructura como Código (IaC). Además, se buscó analizar los beneficios y limitaciones de los métodos tradicionales en comparación con IaC.

#### **Metodología**

##### ***Participantes***

Se seleccionaron 32 estudiantes de la asignatura de Ingeniería de Tráfico de la carrera de Telecomunicaciones, organizados en 8 grupos.

##### ***Proceso de la Prueba***

1. Despliegue Inicial: Se utilizó Terraform y Ansible para configurar 8 máquinas virtuales con topologías predefinidas, asignadas a cada grupo de trabajo.

2. **Ejercicio Práctico:** Cada grupo trabajó con la máquina virtual asignada para realizar tareas de configuración de red, monitoreo y evaluación del rendimiento. De igual manera cada grupo realizara el procedimiento de manera tradicional armando la topología desde cero.
3. **Recopilación de Datos:** Al finalizar la actividad, los estudiantes completaron una encuesta estructurada para evaluar diversos aspectos de la implementación de IaC.

### **Desarrollo de la Prueba**

La actividad se llevó a cabo con la participación de 32 estudiantes, organizados en grupos, a quienes se les asignaron 8 máquinas virtuales previamente configuradas. La prueba fue diseñada para explorar dos enfoques de configuración:

#### ***Configuración de los Laboratorios***

##### **1. Configuración Manual:**

- Durante esta etapa, los estudiantes llevaron a cabo el proceso de instalación y configuración de cada componente de forma manual, replicando las tareas comunes que se realizan en un entorno tradicional.
- Este enfoque permitió identificar una serie de desafíos inherentes al método manual, tales como errores humanos, variabilidad en las configuraciones entre los diferentes grupos de trabajo y un tiempo considerablemente prolongado para completar las tareas.

##### **2. Configuración Automatizada:**

- En esta etapa, se hizo uso de herramientas de Infraestructura como Código (IaC), como Terraform y Ansible, para automatizar el despliegue de los entornos de laboratorio.
- Los estudiantes accedieron a las máquinas virtuales previamente asignadas, que contenían laboratorios predefinidos. Su tarea consistió en realizar configuraciones rápidas y precisas de acuerdo con lo establecido en la guía. Este enfoque automatizado no solo simplificó las tareas, sino que también optimizó el tiempo necesario para completarlas.

## Resultados

### *Especificaciones de la Prueba*

**Tabla 27**  
*Especificaciones de la prueba*

<b>Aspecto</b>	<b>Detalle</b>
<b>Número de estudiantes</b>	32
<b>Número de máquinas virtuales</b>	8
<b>Herramientas utilizadas</b>	Terraform, Ansible
<b>Métodos de configuración</b>	Manual y automatizado
<b>Tiempo promedio (manual)</b>	2-3 horas
<b>Tiempo promedio (automatizado)</b>	1-2 horas

### *Ventajas del Método Automatizado*

1. **Reducción de Tiempos:** El método automatizado permitió una reducción significativa del tiempo de configuración, con una disminución del 40% en comparación con el método manual.
2. **Consistencia y Precisión:** IaC garantizó la replicación exacta de los entornos, eliminando discrepancias entre los grupos.

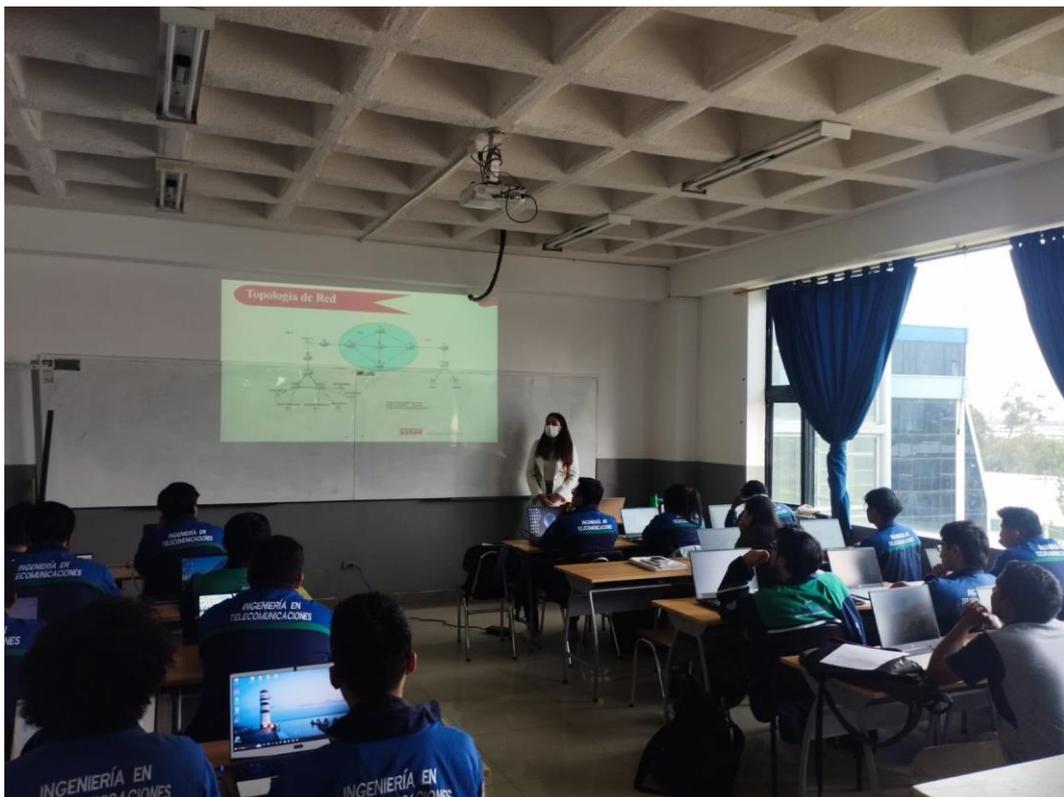
3. **Aprendizaje:** Los estudiantes pudieron enfocarse en el análisis y desarrollo práctico, dejando de lado tareas repetitivas y propensas a errores.
4. **Percepción Positiva:** La mayoría de los estudiantes destacó la simplicidad y efectividad del enfoque automatizado, considerándolo altamente beneficioso para el desarrollo de actividades académicas.

## Evidencia

El desarrollo de la práctica fue documentado con imágenes y un informe elaborado por los estudiantes. A continuación, se presenta las evidencias de la prueba realizada:

**Figura 79**

*Exposición de Infraestructura como código*



**Figura 80**

*Introducción de la implementación y despliegue de IaC para los laboratorios del área de redes*

**Figura 81**

*Desarrollo de la prueba de los laboratorios predefinidos*

