

UACM

Universidad Autónoma
de la Ciudad de México

Nada humano me es ajeno

COLEGIO DE CIENCIA Y TECNOLOGÍA

LICENCIATURA EN INGENIERÍA EN SISTEMAS ELECTRÓNICOS Y
DE TELECOMUNICACIONES

Implementación y Administración de un Entorno de Virtualización con Proxmox VE

TRABAJO DE TESIS

QUE PARA OBTENER EL TÍTULO DE LICENCIADO EN INGENIERÍA
EN SISTEMAS ELECTRÓNICOS Y DE TELECOMUNICACIONES

PRESENTA

Eladio García García

DIRECTOR DE TRABAJO

M. en I. Oscar René Valdez Casillas

Ciudad de México, noviembre del 2025

INTEGRACIÓN DEL JURADO:

Presidente:

- Dr. Miguel Ángel Borja Benítez, Casa Libertad

Secretario:

- Mtro. David Estrada Espinosa, Casa Libertad

Vocal:

- M.I. Oscar Rene Valdez Casillas, Casa Libertad

Plantel de adscripción: Casa Libertad

DIRECTOR DE TESIS.

M. en I. Oscar Rene Valdez Casillas

UACM Plantel Casa Libertad.

AGRADECIMIENTO



A Dios, A mi Ángel guardián por llegar a estos momentos tan importante y especial en mi camino profesional.

A mis Padres, que me han dado apoyo en todo momento. Muchas Gracias

A la Universidad Autónoma de la Ciudad de México (UACM), por brindarme la oportunidad de continuar con mis estudios y que me permitió continuar trabajando.

A cada profesor de la Academia de Ciencia y Tecnología del Plantel Casa Libertad, por su valioso apoyo y enseñanza. De manera especial, agradezco a mis tres lectores: la Dra. Tejinder Kaur, el M. en I. David Estrada Espinosa y el Dr. Miguel Ángel Borja Benítez, por su orientación y comentarios que fueron fundamentales para terminar este trabajo.

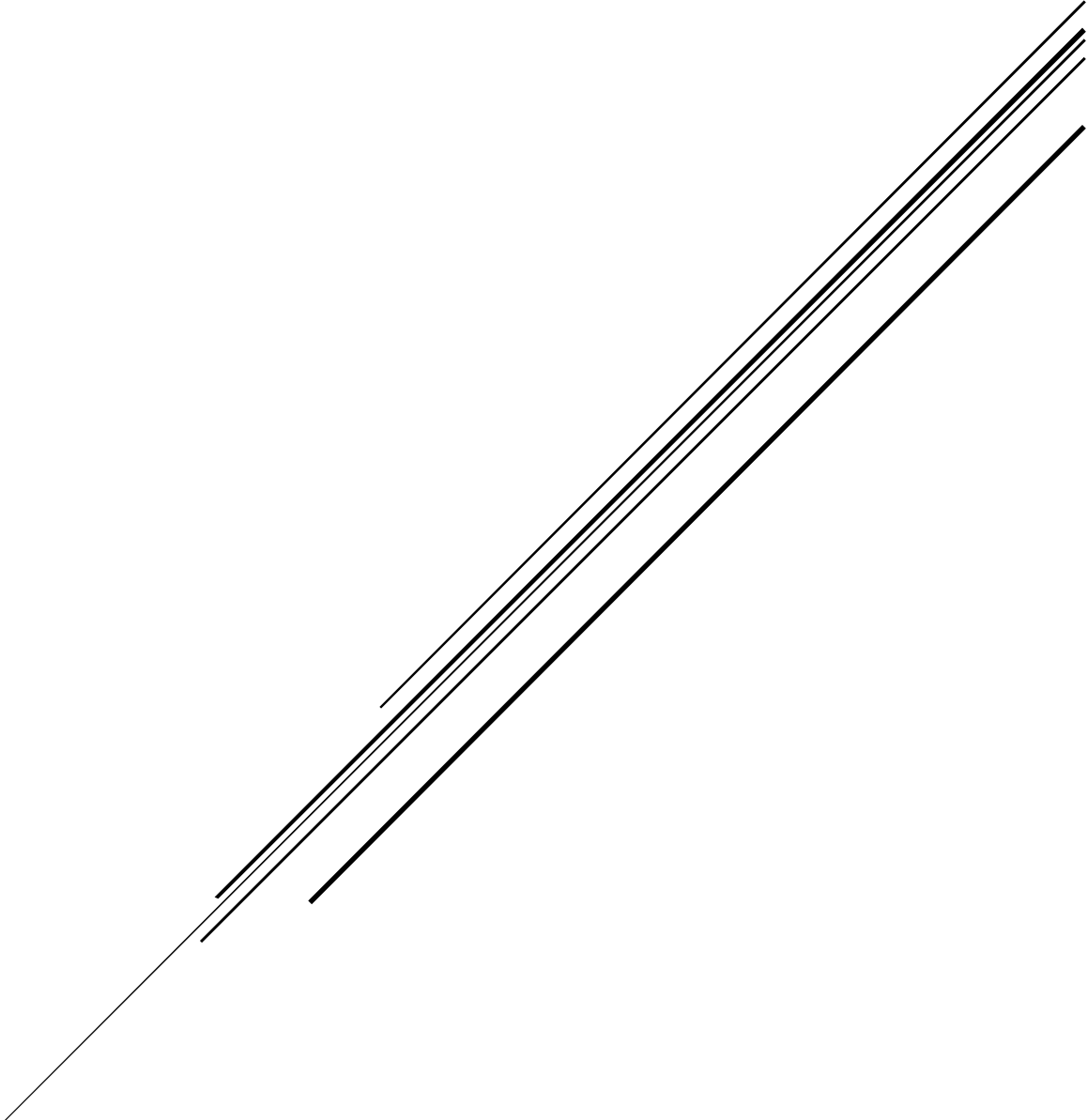
Al profesor Oscar Rene Valdez Casillas por sus enseñanzas, apoyo y asesoramiento para terminar este trabajo. Muchas gracias.

A Iván y mis demás compañeros por ser mis amigos en el transcurso de mis estudios. Muchas gracias.

A Guadalupe por ser como mi segundo padre y por entenderme en cada momento difícil y confiar en mí. Muchas gracias.

A Rebeca al ser una compañera en motivarme a terminar este trabajo. Muchas gracias.

CONTENIDO



AGRADECIMIENTO	III	
CONTENIDO	V	
ÍNDICE DE FIGURAS Y TABLAS	VIII	
OBJETIVOS DE LA INVESTIGACIÓN.....	XIV	
CAPÍTULO 1		XVI
INTRODUCCIÓN.....	XVI	
1.1 Contextualización del problema	XX	
1.2 Planteamiento del problema.....	XXI	
1.3 Justificación del estudio	XXI	
1.4 Alcance y limitaciones.....	XXII	
CAPÍTULO 2		1
MARCO TEÓRICO	1	
2.1 Sistemas operativos.....	2	
2.2 Historia y evolución de la virtualización	5	
2.3 Tipos de virtualización: completa y paravirtualización	7	
2.3.1 Definición de virtualización.....	7	
2.3.2 Modelos y tipos de virtualización.....	9	
2.4 Hipervisores	14	
2.4.1 Tipos de hipervisores	16	
2.3.1 Hipervisor de tipo 1	17	
2.3.2 Hipervisor de tipo 2	18	
2.4 Beneficios y retos de la virtualización.....	20	
2.5 Comparativa de plataformas: Proxmox VE, VMware ESXi, Hyper-V, VirtualBox	23	
CAPÍTULO 3		27
ARQUITECTURA Y CARACTERÍSTICAS DE PROXMOX VE	27	
3.1 Introducción a Proxmox VE	28	
3.1.1 Tecnologías base.....	32	
3.1.2 Sistemas de archivos	36	
3.2 Interfaz gráfica <i>web</i> y e interfaz de línea de comandos.....	37	
3.3 Gestión de usuarios, roles y permisos.....	39	
3.4 Seguridad básica del sistema.....	41	
3.4.1 Autenticación y control de acceso.....	41	
3.4.2 Copias de seguridad mediante la opción de <i>backup</i>	42	

CAPÍTULO 4	43
DISEÑO E IMPLEMENTACIÓN DEL ENTORNO VIRTUALIZADO.....	43
4.1 Análisis y selección del <i>hardware</i> disponible.....	46
4.2 Dimensionamiento de recursos para máquinas virtuales.....	48
4.3 Creación de máquinas virtuales	52
4.4 Instalación de servicios	53
4.4.1 Utilidad y aplicaciones de los servidores virtualizados	53
4.4.2 Servidores instalados.....	54
4.5 Criterios para evaluación del rendimiento y métricas de desempeño	61
4.6 Análisis de los servidores.....	68
4.6.1 Velocidad de transferencia cliente-servidor	68
4.6.2 Captura de tráfico.....	74
4.6.3 Simulación de peticiones en la máquina virtual y la máquina física.....	80
4.7 Análisis de métricas del servidor Físico y el Servidor Virtual	85
CAPÍTULO 5	97
CONCLUSIONES	97
REFERENCIAS	102
SIGLAS Y ACRÓNIMOS	108
APÉNDICES.....	110
Apéndice A.....	111
Apéndice B.....	116
Apéndice C	119
Apéndice D	127

ÍNDICE DE FIGURAS Y TABLAS



Índice de figuras

CAPÍTULO 2. MARCO TEÓRICO.

Figura 2.1: Componentes de una computadora.	5
Figura 2.2: Modelos de sistema. (a) Máquina no virtual. (b) Máquina virtual.	6
Figura 2.3: Hyper-V en Windows Server 2008 ejecutando varias máquinas virtuales.	13
Figura 2.4: Monitor de Máquina Virtual básico	16
Figura 2.5: Hipervisor de tipo 1	17
Figura 2.6: Ejemplo de un Hipervisor tipo 2.	19
Figura 2.7: Hipervisor Oracle Virtual Box tipo 2.	20
Figura 2.8: Ejemplo de uso de CPU en la Máquina Virtual.	21

CAPÍTULO 3. ARQUITECTURA Y CARACTERÍSTICAS DE PROXMOX VE

Figura 3.1: Diagrama de Flujo de datos de como la MV accede a la red.	29
Figura 3.2: Diagrama integración entre <i>software</i> hipervisor y <i>hardware</i> .	34
Figura 3.3: Diagrama de Relación entre KVM y QEMU en entornos virtualizados.	35
Figura 3.4: Relación entre KVM y QEMU en entornos virtualizados.	36
Figura 3.5: Pantalla Principal de Proxmox VE.	38
Figura 3.6: Opciones del Menú de Proxmox VE.	38
Figura 3.7: Sección de Salud de Proxmox VE.	39
Figura 3.8: Diagrama de procesos de usuarios.	40

CAPÍTULO 4: DISEÑO E IMPLEMENTACIÓN DEL ENTORNO VIRTUALIZADO

Figura 4.1: Topología de la Red física.	44
Figura 4.2: Topología de la Red virtual.	45
Figura 4.3: Servidor Web.	55
Figura 4.4: Diagrama de Bloques Servidor Web.	55
Figura 4.5: Servidor FTP.	56
Figura 4.6: Servidor SSH.	57
Figura 4.7: Servidor cliente NFS.	58
Figura 4.8: Servidor Samba instalado en Ubuntu.	59
Figura 4.9: Diagrama Servidor VoIP.	60
Figura 4.10: Ejemplo de llamadas usando un servidor VoIP.	60
Figura 4.11: Tasa de Transferencia cliente y Servidor virtual.	72
Figura 4.12: Tasa de Transferencia cliente y Servidor Físico.	73
Figura 4.13: Grafica de tráfico Servidor del Servidor virtual.	75
Figura 4.14: Grafica de tráfico de Servidor físico.	76
Figura 4.15: Grafica de tráfico del servidor Web virtual y servidor web físico.	79
Figura 4.16: Resultados de la Máquina física y la máquina virtual.	84
Figura 4.17: Grafica de resultados de la CPU de la Máquina física y la Máquina virtual.	90
Figura 4.18: Grafica de resultados de la Memoria RAM de la Máquina física y la Máquina virtual.	91
Figura 4.19: Grafica de resultados de la Memoria <i>Cache</i> de la Máquina física y la Máquina virtual.	92
Figura 4.20: Grafica de resultados de memoria de intercambio(SWAP) de la Máquina virtual.	93

APÉNDICE A

Figura 5.1: Pantalla de Inicio antes de Instalar de Proxmox VE.	112
Figura 5.2: Consola de Proxmox VE.	113
Figura 5.3: Subir la primera Imagen ISO a Proxmox.	113
Figura 5.4: Ventana de usuarios en Proxmox VE.	114
Figura 5.5: Ventana Administración para visualizar MV.	115
Figura 5.6: Ventana de Google Authenticator QR.	115
Figura 5.7: Ventana de <i>Backup</i> de la MV.	116

APÉNDICE B

Figura 6.1: Ventana de consola <code>#pvcreate /dev/sda(id)</code> .	117
Figura 6.2: Ventana de consola crear volumen.	117
Figura 6.3: Ventana de Almacenamiento.	118
Figura 6.4: Ventana de consola creación de zpool.	118
Figura 6.5: Ventana de directorio.	118

APÉNDICE C

Figura 7.1: Ventana para crear la Máquina virtual.	120
Figura 7.2: Ventana de red.	120
Figura 7.3: Ventana de resumen de recursos.	120
Figura 7.4: Diagrama de la red del Laboratorio de Telecomunicaciones.	122
Figura 7.5: Ventana para crear una red	125
Figura 7.6: Ventana de Firewall.	126

APÉNDICE D

Figura 8.1: Diagrama API con Python.	128
Figura 8.2: Diagrama API con Python crear una MV.	128
Figura 8.3: Diagrama de un clúster de Proxmox VE básico.	130
Figura 8.4: Menú de administración de nodos.	132
Figura 8.5: Equipos del Laboratorio de Telecomunicaciones.	135

Índice de tablas.

CAPÍTULO 2. MARCO TEÓRICO

Tabla 2.1 Comparativa de Hipervisores.	25
--	----

CAPÍTULO 4. DISEÑO E IMPLEMENTACIÓN DEL ENTORNO VIRTUALIZADO

Tabla 4.1: Comparación de configuración recomendada de Proxmox VE.	47
Tabla 4.2: Dimensionamiento de recursos para cada sistema operativo	49
Tabla 4.3: Comparación de los equipos de Laboratorio de Telecomunicaciones.	51
Tabla 4.4: Comparación de recursos en las Máquinas virtuales.	52
Tabla 4.5: Definiciones y Métricas.	66
Tabla 4.6: Datos de la prueba de Red utilizando iperf3.	70
Tabla 4.7: Ecuaciones de las pruebas realizadas.	71
Tabla 4.8: Comentarios de los resultados obtenidos.	71
Tabla 4.9: Datos de las pruebas realizadas.	74
Tabla 4.10: Datos de la gráfica de tráfico servidor físico y servidor virtual (Asterisk).	77
Tabla 4.11: Pruebas obtenidos en el servidor web.	82
Tabla 4.12: Porcentaje de recursos de la CPU de la máquina física.	86
Tabla 4.13: Porcentaje de recursos de la CPU de la máquina virtual.	87
Tabla 4.14: Porcentaje de recursos de Memoria RAM de la máquina física.	87
Tabla 4.15: Porcentaje de recursos de Memoria RAM de la máquina virtual.	87
Tabla 4.16: Porcentaje de recursos de Memoria <i>Cache</i> de la máquina física.	88
Tabla 4.17: Porcentaje de recursos de Memoria <i>Cache</i> de la máquina virtual.	88
Tabla 4.18: Porcentaje de recursos de Memoria compartida de la máquina virtual.	88
Tabla 4.19: Comentarios de las Métricas de la Máquina física y la Máquina virtual.	95

OBJETIVOS DE LA INVESTIGACIÓN



Objetivo general

- Implementar y administrar un entorno de virtualización utilizando Proxmox VE (Entorno Virtual) versión 8.3 como *software* principal en los equipos disponibles en el Laboratorio de Telecomunicaciones de la Universidad Autónoma de la Ciudad de México (UACM), Plantel Casa Libertad para extender su vida útil y mejorar su rendimiento.

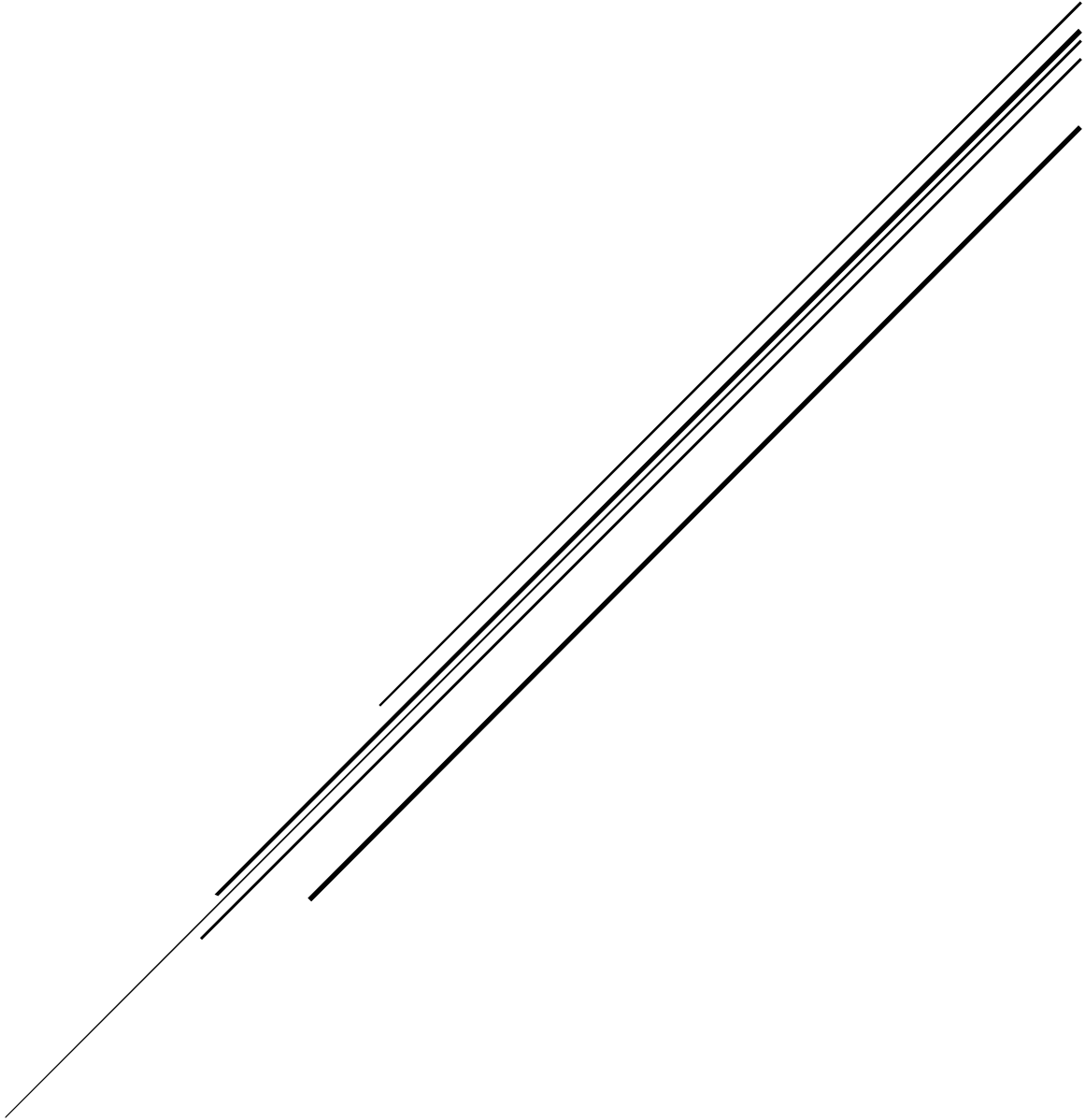
Objetivos específicos

- Documentar la instalación y operación del hipervisor Proxmox VE (Entorno Virtual) versión 8.3.
- Configurar y administrar máquinas virtuales basados en el *núcleo* (KVM¹).
- Implementar mecanismos de respaldo, *snapshots* y almacenamiento compartido.

¹ Las máquinas virtuales basadas en el *núcleo* (KVM) son una tecnología de virtualización open source para los sistemas operativos Linux, los cuales pueden funcionar como hipervisores que ejecutan varias máquinas virtuales aisladas. [87]

CAPÍTULO 1

INTRODUCCIÓN



En la actualidad, la búsqueda de información es una tarea que se realiza de forma muy rápida debido al incremento y la evolución de la tecnología. Portnoy menciona que “La era de la información está en plenitud y nos da acceso a cantidades asombrosas de datos tan pronto como se encuentre disponible” [1]. De este modo, existe una tendencia irrefrenable en la tecnología a ofrecer al usuario la mayor cantidad de información en el menor tiempo posible.

La necesidad de trabajar con un volumen cada vez mayor de datos y procesar de forma más rápida ha llevado al desarrollo de soluciones como la virtualización. Se entiende por ésta a la práctica de emplear los recursos reservados de una computadora física dividiéndolos en bloques/contenedores separados para alojar sistemas virtuales más pequeños e independientes [2].

Una forma de evaluar la capacidad de las computadoras es a través de la clasificación en gamas. Tanto Intel² como AMD³ tienen en el mercado una gama de procesadores que van de los dos a los seis *núcleos*. Dentro de la unidad central de procesamiento (CPU), hay dos componentes importantes: el *overclocking*⁴, y la memoria *cache*⁵.

² Intel ofrece procesadores Core i3 con dos *núcleos*, i5 con dos y cuatro *núcleos*, e i7 con cuatro y seis *núcleos* [3].

³ “AMD ofrece procesadores Phenom II X2, X3, X4 y X6 con dos, tres, cuatro y seis *núcleos* respectivamente [3].

⁴ *overclocking* Técnica que consiste en aumentar la frecuencia de trabajo el componente.

⁵ La memoria ultrarrápida (*caché*) está categorizada en tres niveles: L1, que presenta una capacidad baja con una velocidad muy alta; L2, que presenta una capacidad media y una velocidad alta; e L3, que presenta una capacidad alta con una velocidad media. La *caché* L3 es la más lenta. La mayoría los procesadores actuales tiene, al menos, dos niveles de memoria *caché*. También, hay algunos que incluyen un tercer nivel. [3].

Las gamas de los procesadores son:

- Gama baja⁶: “Los procesadores de gama baja son los más económicos del mercado, tienen menor frecuencia, a veces carecen de caché L3, y las cachés L1 y L2 están recortadas en su tamaño, no soportan un *overclocking* y no suelen tener más de dos *núcleos*” [3]. Dichos equipos son más accesibles de adquirir. Las tareas que realiza son ideales para “navegar por internet, enviar correos, procesar documentos de textos y contenido multimedia” [4].
- Gama media⁷: “Los procesadores de gama media permiten un *overclocking* medianamente decente, tienen *caché* L1 y L2 un poco más de espacio, dependiendo del modelo y tienen una caché L3 acotada, y cuenta con entre tres y cuatro *núcleos*” [3]. Las tareas que se pueden realizar son “edición de fotos y videos, juegos de nivel medio, programación y multitarea intensivo” [4].
- Gama alta⁸: entre los procesadores de gama alta se encuentran “dispositivos con seis u ocho *núcleos*, cachés L1, L2 y L3 amplias, capaces de aceptar *overclocking* extremo (siempre teniendo cuidado con la refrigeración tanto del procesador como de las memorias RAM) y, en general, son los más rápidos del mercado” [3]. Son ideales para realizar “edición de video profesional, diseño gráfico avanzado, simulaciones 3D y juegos de alta gama” [4].

⁶ características principales, cuentan con “Procesadores Intel Celeron, Pentium o AMD Athlon, Memoria RAM limitada (generalmente entre 2GB y 4GB), Almacenamiento puede ser Disco duro mecánico o Unidad de estado sólido (SSD) (generalmente entre 128GB y 500GB) y una batería de 65W [79]

⁷ características principales, cuentan con “Procesadores Intel Core i3, i5, i7 o AMD Ryzen 5, almacenamiento unidad de estado sólido (SSD) de entre 256 GB a 512 GB, memoria RAM generalmente de entre 8GB y 16GB; y una batería con un consumo de 95 W [79].

⁸ características principales, cuentan con “Procesadores Intel Core i7, i9, AMD Ryzen 7 o Ryzen 9, Almacenamiento SSD de 512 GB a 1 TB o más, Memoria RAM de 16 GB o más (generalmente entre 16GB y 32GB, o más), gráficos dedicados de alta gama, y una batería de 130 W [79].

Proxmox VE, versión 8.3, es un *software* conocido por ser un hipervisor tipo 1, debido a que no necesita de un sistema operativo para que sea instalado, ya que se instala como sistema anfitrión, a diferencia del hipervisor tipo 2, que sí lo requiere. Con dicho *software* es posible administrar varias máquinas virtuales que funcionen como una suerte de computadoras independientes dentro de un mismo equipo físico.

La virtualización permite superar algunas barreras impuestas por el *hardware* para poder procesar una mayor cantidad de datos en menor tiempo. Robinson menciona que. “Las únicas limitaciones son la cantidad de recursos en el equipo host y los requisitos de recursos de los equipos virtualizados (máquinas virtuales, VM)” [2]. De este modo, la virtualización se revela como una práctica atractiva para mejorar el rendimiento de los equipos.

La virtualización es la base de la computación en la nube. Consiste en trabajar con máquinas virtuales, que son equipos de cómputo que cuentan con toda lo necesario para administrar los recursos de la computadora anfitriona, para mantener la instalación de los servidores que se usan en diversas materias, tales como informática en las telecomunicaciones, sistemas telefónicos, estándares en área local y amplia TCP/IP, y topologías y diseño de redes. Estas máquinas utilizan una tecnología de virtualización de código abierto que se encuentra integrado en el *núcleo* de Linux en el que está basadas y se instalan como sistema anfitrión, utilizando un emulador llamada emulación rápido (QEMU)⁹ para una virtualización completa de acuerdo con los sistemas operativos donde se implementen.

⁹ QEMU es una herramienta de emulación y virtualización de código abierto que permite ejecutar sistemas operativos y aplicaciones diseñados para una arquitectura de *hardware* en otra. Esto permite ejecutar máquinas virtuales con un rendimiento cercano al nativo cuando el *hardware* del host es compatible con la arquitectura del invitado [31].

1.1 Contextualización del problema

Este trabajo expone los resultados de la implementación de un hipervisor para la administración de los equipos de bajos recursos que se encuentran en el Laboratorio de Telecomunicaciones de la Universidad Autónoma de la Ciudad de México, plantel Casa Libertad. Se trata de un espacio dirigido a los estudiantes de Ingeniería en Sistemas Electrónicos y de Telecomunicaciones, con el fin de que éstos puedan ejercitarse en actividades relacionadas con las asignaturas de carrera, como el uso de una base de datos, servidores WEB¹⁰ SSH¹¹, FTP¹², NFS¹³, Samba¹⁴ para el cliente de Windows y Servidor VoIP¹⁵ (Asterisk¹⁶).

Dichos servidores permiten compartir archivos entre computadoras a través de distintos puertos y realizar llamadas a una conexión de Internet, para mantener la instalación de los servidores que se usan en diversas materias, tales como informática en las telecomunicaciones, sistemas telefónicos, estándares en área local y amplia TCP/IP, y topologías y diseño de redes.

¹⁰ La página web se almacena en el servidor. Cada vez que llega una solicitud de un cliente, se le envía el documento correspondiente. Un servidor también puede ser más eficiente mediante multihilo o multiprocesamiento. En este caso, un servidor puede responder a más de una solicitud a la vez [83]. Algunos servidores web populares son Apache y Microsoft Internet Information Server.

¹¹ shell seguro (SSH) Un programa cliente-servidor que proporciona seguridad [83].

¹² El Protocolo de Transferencia de Archivos (FTP) es el mecanismo estándar proporcionado por TCP/IP para copiar archivos de un host a otro [83].

¹³ Sistema de archivos de red (NFS). Protocolo de aplicación TCP/IP que permite al usuario acceder y manipular sistemas de archivos remotos como si fueran locales. Utiliza los servicios del Protocolo de Llamada a Procedimiento Remoto [83].

¹⁴ Samba puede facilitar la coexistencia de computadoras Windows y Linux en la misma red [84].

¹⁵ (Voz sobre Protocolo de Internet), "se refiere a la capacidad de realizar llamadas de voz a través de Internet en lugar de utilizar las tradicionales líneas fijas de cobre. Es decir convierte la señal de voz en paquetes de datos que se transmiten a través de la red de Internet hasta llegar al destinatario" [85].

¹⁶ Asterisk es un programa de *software* libre (bajo licencia GPL) que proporciona funcionalidades de un conmutador IP [82].

1.2 Planteamiento del problema

Los equipos del Laboratorio de Telecomunicaciones se encuentran en estado de obsolescencia o están próximos a volverse obsoletos. La adquisición de nuevos equipos de cómputo siempre es un esfuerzo complejo para las instituciones educativas como la Universidad Autónoma de la ciudad de México (UACM), debido a precio y a las limitaciones de almacenamiento físico en las instalaciones. De ahí la necesidad de buscar soluciones que permitan extender el tiempo de vida de los equipos para aprovecharlos al máximo.

Este trabajo plantea el diseño e implementación de un entorno de virtualización explorando las oportunidades de implementación del *software* hipervisor tipo 1 Proxmox VE (Entorno Virtual) versión 8.3 como sistema anfitrión. El propósito es mejorar el rendimiento de los servidores físicos creando varios servidores (máquinas virtuales) dentro de cada uno con ayuda de dicho *software* y, de paso, demostrar su funcionalidad, facilidad de administración y beneficios técnicos y económicos.

1.3 Justificación del estudio

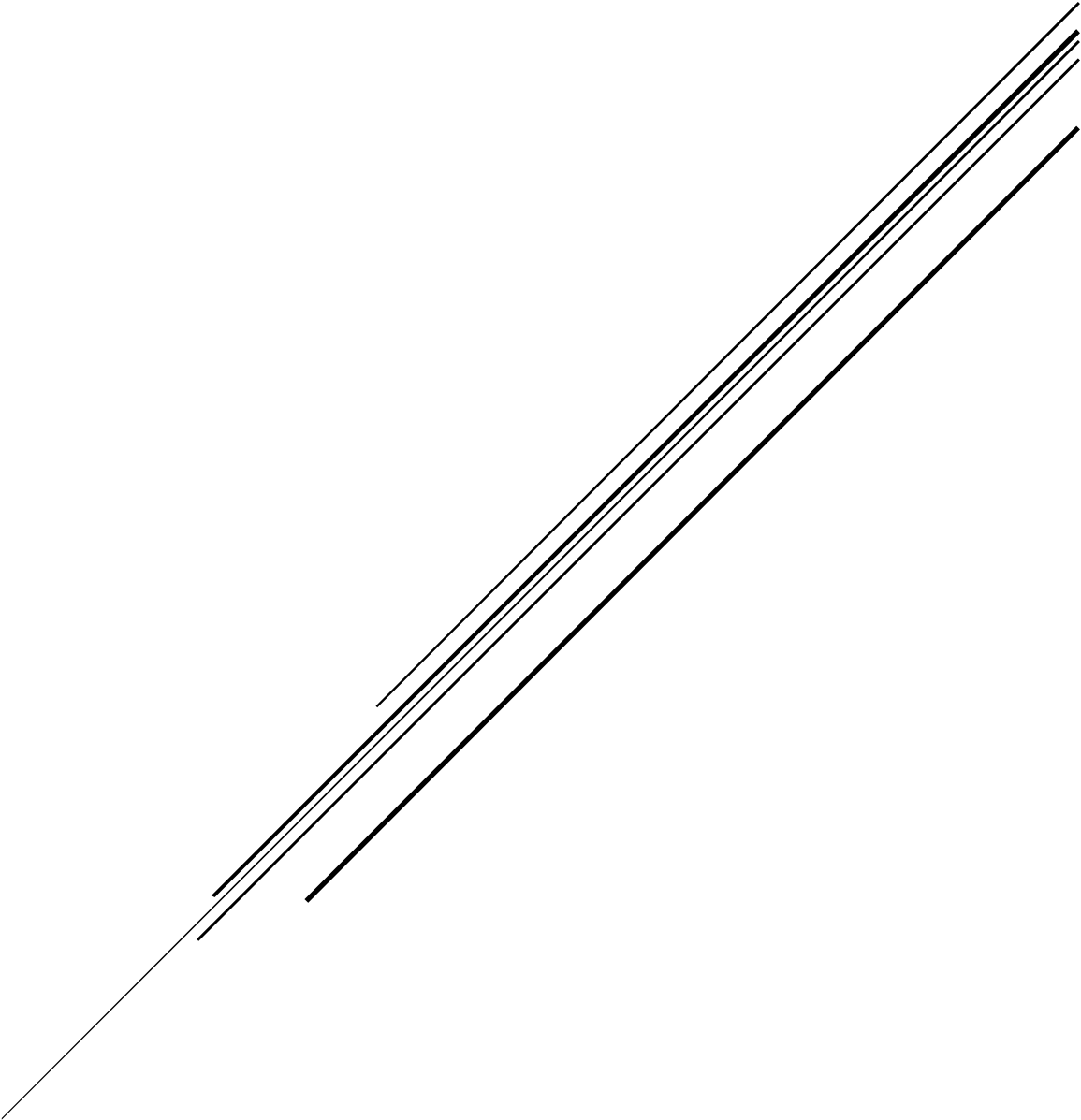
Los hipervisores se ofrecen bajo licencia y los equipos en que se instalen deben cumplir ciertos requisitos, pero el *software* elegido es de uso libre —aunque también está disponible una licencia recomendada para entornos productivos— y no impone restricciones mayores, lo que lo convierte en una solución particularmente atractiva. Esto evita la dependencia hacia algún proveedor. De este modo, la implementación del *software* constituye una oportunidad para la optimización de recursos, el ahorro de energía y espacio, y la mejora en las capacidades de servicio del Laboratorio, derivando en beneficios para los estudiantes.

1.4 Alcance y limitaciones

El trabajo contempla en los equipos del Laboratorio de Telecomunicaciones, el análisis de su operar, la realización de diversas tareas de configuración y administración de máquinas virtuales. Se busca determinar los beneficios en rendimiento alcanzables para los equipos y estimar la prolongación de su vida útil y el ahorro en inversión de recursos.

CAPÍTULO 2

MARCO TEÓRICO



Se presenta el marco teórico de este trabajo, donde se abordan distintos conceptos clave cuya comprensión es indispensable como parte de los requisitos para ejecutar Proxmox VE versión 8.3. En un primer momento, se explica qué son los sistemas operativos y su importancia para el almacenamiento y la transmisión de información. Luego, se expone una breve historia de la virtualización, con los hitos que marcaron su desarrollo. Se presentan los diferentes tipos de virtualización —como emulación, paravirtualización, virtualización de servidor, virtualización de sistema operativo (también conocida como contenedores) y virtualización de aplicaciones—, de acuerdo con la Universidad de Massachusetts. Por último, se realiza un análisis comparativo para detectar las diferencias entre los tipos de hipervisores, contrastando las características de Proxmox VE (Entorno Virtual) contra las de (VMware ESXi)¹⁷, (Hyper-V)¹⁸ y (VirtualBox)¹⁹.

2.1 Sistemas operativos

Un sistema operativo (SO) “es un programa que tiene encomendadas una serie de funciones diferentes cuyo objetivo es simplificar el manejo y la utilización de la computadora, haciéndolo seguro y eficiente” [5]. En su estructura, normalmente se recurre a dos tipos de soluciones: sistemas por capas y sistemas cliente-servidor.

¹⁷ VMware ESXi es un hipervisor tipo 1 o "bare metal" enfocado en el centro de datos, que reemplaza al sistema operativo principal que interactuaría con los componentes físicos de una computadora. ESXi sucedió a ESX, un hipervisor más grande que utilizaba más recursos del equipo host [86].

¹⁸ Hyper-V es un sistema de virtualización para Windows que permite a un único servidor físico alojar varias máquinas virtuales que ejecutan su propio sistema operativo [86].

¹⁹ VirtualBox es el *software* de virtualización multiplataforma, gratuito y de código abierto más popular del mundo. Se basa en la vibrante participación de la comunidad, combinada con el desarrollo y soporte de primera clase de Oracle [26].

- **Sistemas por capa:** en un sistema por capas, el sistema operativo se organiza como una jerarquía de capas, donde cada capa ofrece una interfaz clara y bien definida a la capa superior y solamente utiliza los servicios que le ofrece la capa inferior. La principal ventaja que ofrece este tipo de estructuras es el modularidad y la ocultación de la información. Una capa no necesita conocer como se ha implementado la capa sobre la que se construye, sino que únicamente necesita conocer la interfaz que ofrece. Esto facilita enormemente la depuración y verificación del sistema, puesto que las capas se pueden ir construyendo y depurando por separado. Este enfoque fue utilizado por primera vez en el "THE, diseñado por Dijkstra en 1968, un sistema operativo sencillo que estaba formado por seis capas" [5].
- **Modelo cliente-servidor:** en este tipo de modelo, el enfoque consiste en implementar la mayor parte de los servicios y funciones del sistema operativo en procesos de usuario, dejando solo una pequeña parte del sistema operativo ejecutando en modo *núcleo*. A esta parte se le denomina "*micronúcleo*" y a los procesos que ejecutan el resto de funciones se les denomina "servidores".

Un sistema operativo está formado por tres capas principales:

- La capa *m*: se denomina "*núcleo*" (*kernel*) y es la que gestiona los recursos *hardware* del sistema y la que suministra otra la funcionalidad básica del sistema operativo. De acuerdo con Tanenbaum, "El *núcleo* es la parte del sistema operativo que interacciona directamente con el *hardware* de la máquina" [6]; es decir, actúa como un intermediario para que sus funciones sean administradas como los recursos de la computadora el procesador y funciones básicas para manejar la memoria entre en *hardware* y las aplicaciones. Esta capa ha de ejecutar en nivel *núcleo*, mientras que las otras pueden ejecutar en niveles menos permisivos.

- La capa de servicios o llamadas al sistema: ofrece a los programas unos servicios en forma de una interfaz de programación o API (interfaz de programación de aplicaciones). Desde el punto de vista de los programas, esta capa extiende la funcionalidad de la computadora, por lo que se suele decir que el sistema operativo ofrece una máquina virtual extendida a los programas. De esta forma se facilita la elaboración de éstos, puesto que se apoyan en las funciones que le suministra el sistema operativo. Un ejemplo son los sistemas de archivos.
- La capa de intérprete de comandos o *shell*: suministra una interfaz a través de la cual el usuario puede dialogar de forma interactiva con la computadora. El *shell* recibe los mandatos u órdenes del usuario, los interpreta y, si puede, los ejecuta. También se le conoce como interfaz de usuario.

Una computadora debe contar con un sistema operativo para que tenga su funcionamiento básico y la interacción entre el *hardware*, los programas y el usuario. Conforme a Nair, “Los programas informáticos se compilan, distribuyen y almacenan como archivos binarios ejecutables que se ajustan a una interfaz binaria de aplicación (ABI) específica, que incluye características tanto del conjunto de instrucciones del hardware como del sistema operativo” [7]. Por lo general, es muy común ejecutar un sistema operativo Windows o Linux.

En la Figura 2.1 se pueden apreciar todos los elementos que conforman una computadora: una placa principal, un procesador, un disco duro y una serie de dispositivos periféricos (teclado, ratón, vídeo, etc.).

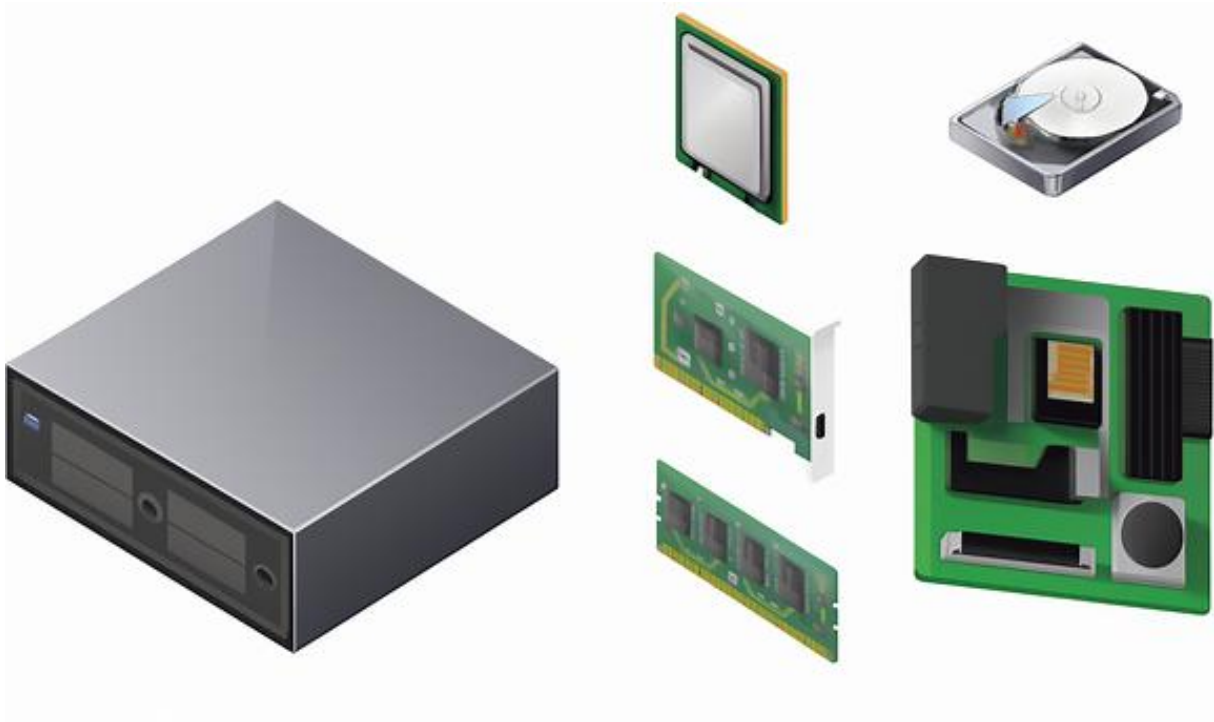


Figura 2.1: Componentes de una computadora [Fuente: E. T. Gómez, Proxmox. Curso Práctico, Birmingham, UK]: Alfa omega – RAMA, 2024.]

2.2 Historia y evolución de la virtualización

Marshall señala que “la historia de la virtualización comenzó mucho antes de lo que se imagina, siendo discutida por primera vez a finales de la década de 1950” [8]. Antes de la virtualización, la implementación de servidores físicos era costosa y lenta, y se subutilizaban recursos como la Unidad Central de Procesamiento (CPU), la Memoria de Acceso Aleatorio (RAM) y los discos. Pero en los años 90, “Sun Microsystems introdujo Java, permitiendo el paradigma 'escribe una vez, ejecuta en cualquier lugar' mediante la virtualización del nivel de proceso en el entorno de ejecución de Java” [9]. Por su parte, “VMware, fundada en 1998, desarrolló técnicas avanzadas de virtualización de *hardware* y patentó sus tecnologías en 2002, consolidándose en el mercado como la primera plataforma comercial de virtualización en arquitectura x86” [8].

Abraham menciona que “Las máquinas virtuales se utilizaron comercialmente por primera vez en 1972 en los mainframes de Máquina de Negocios Internacionales (IBM) a través del sistema operativo de las máquinas virtuales” [10]. En otras palabras, una vez creado, se puede ejecutar cualquier sistema operativo o paquete de *software* disponible en la máquina subyacente.

De acuerdo con Portnoy, “Las computadoras personales, la digitalización del escritorio físico y la tecnología cliente/servidor encabezaron las décadas de los ochenta y los noventa” [1]. Mientras, “los primeros monitores de máquinas virtuales se usaron para el desarrollo y depuración de sistemas operativos, ofreciendo un entorno seguro para pruebas repetidas” [1]. Pronto se añadió la capacidad de ejecutar varios entornos simultáneamente, lo que permitió dividir los recursos del hardware en servidores virtuales con su propio sistema operativo. Para comprender la diferencia entre las máquinas anfitrión y máquinas virtuales, puede recurrirse a la Figura 2.2. El inciso (a) corresponde a una computadora única con tres capas: el *hardware*, el *kernel* y los procesos, como se explicó anteriormente. Mientras, en el inciso (b) se puede ver una computadora que está mandando a generar tres máquinas virtuales para realizar varios procesos, con las capas que los conforman.

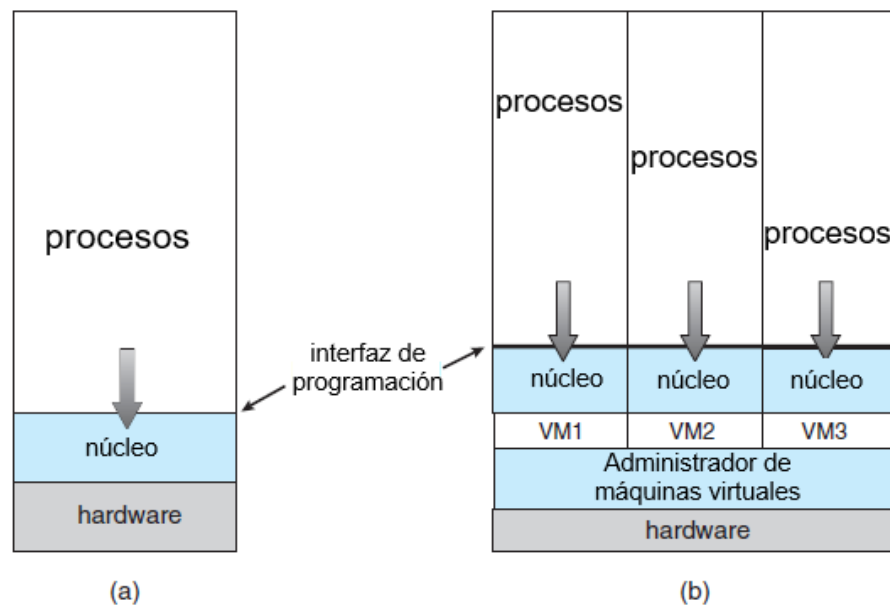


Figura 2.2: Modelos de sistema. (a) Máquina no virtual. (b) Máquina virtual. [Fuente: P. B. Abraham Silberschatz, *Conceptos del sistema operativo*, Hoboken, Nueva Jersey, Estados Unidos: John Wiley & Sons, Inc., 2008.]

El *software* es indispensable para que las computadoras puedan ejecutar sus tareas. “Sin su *software*, la computadora es básicamente un montón de metal inútil” [11]. Cuando se implementa, el *software* logra la ejecución de tareas como almacenar, procesar y recuperar información, exhibir documentos multimedia y realizar búsquedas en la Internet.

2.3 Tipos de virtualización: completa y paravirtualización

2.3.1 Definición de virtualización

La virtualización es una tecnología que divide una computadora en múltiples máquinas independientes que pueden soportar la ejecución simultánea de diferentes sistemas operativos. Se le llama virtualizar a la acción de emplear un *software* que permite realizar las labores de dos o más equipos de cómputo de manera simultánea utilizando los recursos de uno solo. Es como tener varios sistemas operativos trabajando al mismo tiempo a través del *hardware* de una sola computadora. Así, la virtualización proporciona una abstracción sobre los recursos reales, optimizando su rendimiento. Algunas definiciones relevantes de virtualización son:

- “La virtualización es una tecnología que permite crear una versión virtual de un recurso informático, como un servidor, un sistema operativo, una red o un dispositivo de almacenamiento” [12].
- “La virtualización es el concepto de tomar una computadora física y dividir sus recursos en fragmentos o contenedores separados para alojar sistemas virtuales más pequeños e independientes utilizando esos recursos reservados” [2].
- “La virtualización es una tecnología inusual que cambia el estado en cómo se debe manejar las computadoras físicas, para brindar servicios y asignar presupuestos” [13].

- “La virtualización es una tecnología de *software* que permite ejecutar varias máquinas virtuales en una sola máquina física, compartiendo los recursos de ese único ordenador entre varios entornos” [8].

Haciendo una lectura conjunta de estas definiciones, se llega a la siguiente definición operativa de virtualización: la virtualización es una tecnología de *software* que abstrae y divide los recursos de un *hardware* físico para crear versiones virtuales e independientes de componentes informáticos —servidores, sistemas operativos, redes o almacenamiento—, de modo que múltiples máquinas o entornos virtuales se ejecuten simultáneamente en un mismo equipo, compartiendo y asignando dinámicamente dichos recursos. Éste es el concepto que se empleará de aquí en adelante dentro de este trabajo.

La virtualización ha ganado importancia en la computación actual porque permite ejecutar más servicios con menos *hardware* físico, facilitando la creación de entornos de prueba para desarrolladores y profesionales de Tecnologías de la Información (TI), donde el *software* de virtualización admite instantáneas y permite restaurar a un estado anterior en cuestión de minutos. En tal sentido, “La virtualización es conceptualmente muy similar a la emulación. En la emulación, un sistema pretende ser otro sistema. En la virtualización, un sistema pretende ser dos o más sistemas del mismo sistema” [14].

“Los sistemas de virtualización para los sistemas operativos modernos suelen contener un sistema simplificados cada proceso en ejecución puede actuar como si fuera el único en ejecución” [14]. Si un proceso intenta utilizar toda la Unidad Central de Procesamiento (CPU), el sistema se bloqueará para permitir que otros procesos utilicen su parte. Sobre este aspecto, hay que tomar en cuenta que las aplicaciones no son las únicas que propensa errores: los sistemas operativos también suelen tener errores y esto puede provocar que una aplicación comprometa su aislamiento habitual.

La virtualización puede llevarse a cabo en distintos niveles. El nivel en el que se aplica da lugar a diferentes técnicas de virtualización. A un nivel superior, hay dos técnicas principales: la virtualización basada en máquinas virtuales (VM) y la basada en contenedores. En el primero de estos casos, “Las máquinas virtuales son computadoras independientes que pueden alojarse en una computadora física, limitadas sólo por los recursos disponibles en el *host* y las necesidades de las VM” [2]. Nair Menciona que “Desde la perspectiva de un proceso una máquina virtual es una combinación de sistema operativo y hardware subyacente (hardware físico real del sistema anfitrión) que permite ejecutar procesos mediante memoria lógica, registros e instrucciones a nivel usuario que permiten la ejecución del código perteneciente al proceso” [7]. “Al igual que una máquina física es una máquina física antes de la instalación de un sistema operativo, una VM [máquina virtual] es una VM antes de la instalación de un sistema operativo invitado (el término ‘sistema operativo invitado’ se utiliza para referirse a una instancia del sistema operativo instalada en una VM)” [7]. En síntesis, Scarfone define que “La máquina virtual es un entorno simulado creado mediante virtualización” [15].

2.3.2 Modelos y tipos de virtualización

La virtualización se divide en los siguientes tipos, de acuerdo a “la Universidad de Massachusetts Amherst” [8]:

- Emulación: “es la imitación de un programa informático correcto en otra plataforma. De esta manera, es posible visualizar archivos o ejecutar programas que no están diseñados para esta operación (Koninklijke Bibliotheek, 2009)” [8]. Según la descripción de la Universidad de Massachusetts Amherst, “la máquina virtual (guest) se emula exactamente como en una máquina física, es decir, simula completamente el *hardware* de la máquina física (*host*) y puede ejecutar un sistema operativo inmutable de otra computadora” [8]. Algunos ejemplos son QEMU, Android Emulator.

- Virtualización a nivel de sistema operativo: “la virtualización en el ámbito del sistema operativo proporciona un alto nivel de seguridad para el funcionamiento tanto de servidores virtuales como físicos, utilizando un sistema operativo común, pero de manera aislada, lo que evita que uno afecte el desempeño de las aplicaciones del otro” [8]. Este modelo comparte el recurso del sistema anfitrión, así como también el entorno se ejecuta de forma propia y aislada. Algunas definiciones relevantes son: “es un entorno de ejecución autónomo que comparte el *núcleo* del sistema anfitrión y puede estar aislado” [16], “es la instancia de tiempo de ejecución de una imagen²⁰” [17], “se centra típicamente en una sola aplicación, pila de aplicaciones o entorno” [18]. “Pueden acceder directamente a los recursos del sistema servidor conectado en una red (host)” [19], “consta de varios bloques de construcción, siendo los dos más importantes los espacios de nombres y los (grupos de control)” [20]. “El sistema facilita la gestión de contenedores Linux (LXC) de forma sencilla” [21]. “Kubernetes²¹ utiliza esta tecnologías en Linux para aislar aplicaciones en ejecución y gestionar recursos de manera eficiente” [22].
- Paravirtualización: de acuerdo a Christine, “A diferencia de la virtualización completa, [en la paravirtualización] el sistema operativo invitado sabe que está siendo virtualizado y puede comunicarse directamente con el hipervisor en lugar de intentar engañar a un sistema operativo invitado para que crea que tiene un sistema propio” [18]. Es decir, la paravirtualización le presenta al invitado un sistema que es similar, pero no idéntico, al sistema preferido del invitado.

²⁰ “Una imagen es un paquete binario que encapsula todos los archivos necesarios para ejecutar una aplicación dentro de un contenedor del sistema operativo”. [67]

²¹ Kubernetes es una plataforma de orquestación de contenedores de código abierto. [69] y “ha sido accesible a millones de usuarios gracias a los lenguajes modernos y las bibliotecas de código abierto” [71].

Miguel menciona que esto “difiere de la virtualización nativa porque el *hardware* de la máquina física no se simula en el sistema operativo de la máquina virtual” [8]. En efecto, se utiliza una interfaz de programación incorporada, permitiendo que las aplicaciones interactúen con los requisitos del sistema operativo modificado de la máquina virtual. Algunos ejemplos son Xen²², (VirtIO Drivers)²³, (IBM z/VM)²⁴.

- Virtualización de aplicaciones: de acuerdo con Taboada “Este tipo de virtualización permite que las aplicaciones que se ejecuten en un entorno virtualizado independiente de la infraestructura subyacente” [12]; es decir, se puede ejecutar las aplicaciones sin la necesidad de que sea instalado en el sistema anfitrión. Según Kusnetzky, “la virtualización de aplicaciones implica el uso de tecnología de *software* que permite ejecutar aplicaciones en diversos sistemas operativos y plataformas de *hardware*” [8]. Así, las aplicaciones pueden ser diseñadas para aprovechar marco de trabajos (*frameworks*)²⁵ específicos; sin embargo, aquellas que no los utilizan no obtendrán los beneficios de la virtualización de aplicaciones. De acuerdo a Miguel, esta capa de virtualización permite:
 - Reiniciar la aplicación en caso de un error. [8]
 - Iniciar una nueva instancia de la aplicación que no cumpla sus objetivos en el nivel de servicio. [8]

²² XenServer es una plataforma de virtualización de servidores que ofrece un rendimiento de virtualización para sistemas operativos de clientes y servidores virtualizados que casi iguala el rendimiento de los servidores bare metal. [86]

²³ Interfaces paravirtualizadas para mejorar rendimiento de disco, red, etc., en KVM/QEMU.

²⁴ Corporación Internacional de Máquinas Comerciales (IBM) z/VM es un sistema operativo con tecnología de virtualización e hipervisor escalable y de alta seguridad, diseñado para ejecutar servidores invitados como máquinas virtuales Linux, z/OS y z/TPF [86].

²⁵ Un framework, o un marco de trabajo, es una plantilla para desarrollar softwares de manera más rápida y eficiente. [88]

- Atender interrupciones, ya sean programadas o imprevistas.
- Facilitar la distribución de carga entre varias aplicaciones para alcanzar altos niveles de escalabilidad.
- Algunos ejemplos de tecnologías de virtualización de aplicaciones son mediante tecnologías como empresa de *software* americana de virtualización (Citrix)²⁶ y aplicación virtualizada (XenApp)²⁷. [12]

- Virtualización de servidores: la virtualización de servidor permite que la máquina virtual reproduzca el *hardware* necesario para que un sistema operativo sin modificaciones se ejecute de manera aislada, utilizando la misma unidad central de procesamiento (CPU) del servidor físico. “Ejemplos de la virtualización de servidores se encuentran en aplicaciones como *software* de máquinas virtuales (VMware Workstation), máquina de negocio internacional en máquinas virtuales (IBM) VM y *software* de virtualización que permite ejecutarse en sistemas MAC (Parallels)²⁸”. [23] Como muestra, en la Figura. 2.3 se puede ver un servidor físico ejecutando Windows Server 2008; es decir, crea la máquina virtual (MV) sobre el *hardware* físico usando un hipervisor, por lo que hay distintas versiones de sistemas operativos ejecutándose al mismo tiempo.

²⁶ Los escritorios virtuales basados en agentes VDA de Citrix XenDesktop 7.6 siempre se ejecutan en un host de hipervisor. Estos escritorios virtuales también suelen aprovisionarse automáticamente; según el diseño del entorno de trabajo, se requieren hasta dos discos virtuales por escritorio. [89]

²⁷ Funcionalidad de XenApp: Las aplicaciones individuales se ponen a disposición de los usuarios a través de un servidor Microsoft Windows; el escritorio y las aplicaciones publicadas pueden asignarse al mismo servidor. Las aplicaciones se ejecutan mediante procedimientos idénticos en modo multiusuario. [89]

²⁸ Parallels es un *software* de virtualización que permite a los usuarios de Mac utilizar aplicaciones de Windows en sus dispositivos y ejecutar múltiples sistemas operativos en una sola computadora. [90]

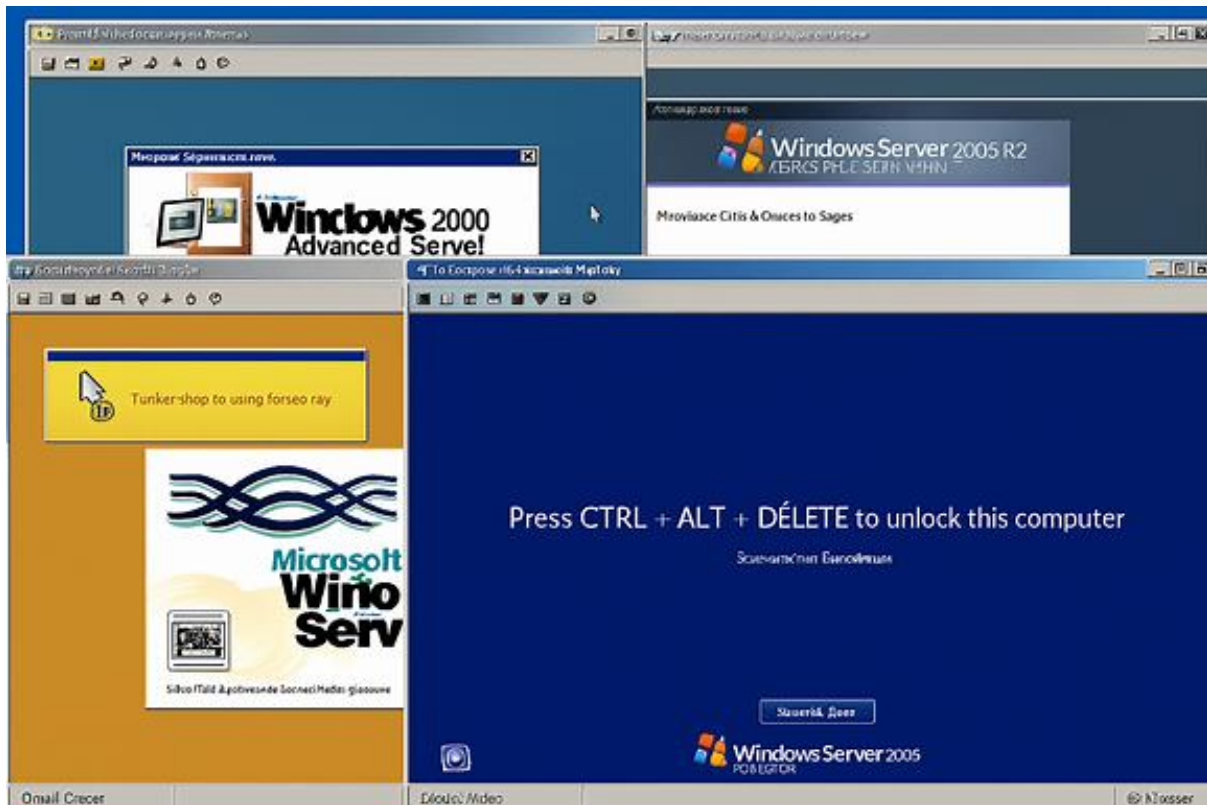


Figura 2.3: Hyper-V en Windows Server 2008 ejecutando varias máquinas virtuales. [Fuente: M. Minasi y D. Gibson, * Dominación de Servidor de Windows 2008 R2*, 1st ed., Wiley Publishing, Inc., Indianapolis, Indiana, EE. UU., 2010.]

Por otro lado, la virtualización puede manifestarse en diferentes capas: [8]

- Acceso: permite que distintos dispositivos puedan acceder a cualquier aplicación sin necesidad de técnicas del entorno. De acuerdo a Río, “las funciones como servicios de terminal (Servicios de Terminal de Microsoft, por ejemplo) y gestores de presentaciones se encuadran en la capa” [8].
- Aplicación: permite la ejecución de la aplicación en el servidor anfitrión para no tener la necesidad de ser instalado en el sistema y tenga su propio entorno virtual así cada computadora puede ejecutar programas sabiendo que se encuentra en otro lado.

- Procesamiento: permite que la computadora física se pueda dividir en múltiples entornos que son ocultados en el sistema operativo para que el *software* interactúa en el *hardware* de forma virtual.
- Almacenamiento: permite que distintos servidores o aplicaciones puedan ser compartido en un mismo recurso de almacenamiento y con esto se pueda realizar distintas actividades como copias de seguridad archivos entre otros.
- Red: permite que distintas redes tengan una combinación de recurso de tecnología de *software* y *hardware* en una única unidad.

2.4 Hipervisores

Un hipervisor, también conocido como monitor de máquina virtual (VMM), es una capa de *software* que se encuentra entre el *hardware* y el sistema operativo, y es el responsable de proporcionar la abstracción de la máquina virtual al sistema operativo que accede. El hipervisor controla el acceso del sistema operativo a los dispositivos de *hardware*. En otras palabras, no se ejecuta en modo usuario, porque debe ejecutar o emular las instrucciones privilegiadas necesarias para acceder al sistema operativo (MATTOS, 2008).

Ruest señala que “Los hipervisores son capas de *software* que pueden ser ejecutados directamente en el *hardware*, eliminando la elevación de otro sistema operativo” [8]. Por su parte, Christine menciona que “la herramienta de *software* principal que se utiliza para crear y administrar máquinas virtuales es un hipervisor, que históricamente se ha denominado monitor de máquina virtual o administrador de máquina virtual (VMM)” [18].

Waters (2009) define al hipervisor “como el componente de virtualización más básico” [8]. Vale la pena decir que es un *software* que separa el sistema operativo y las aplicaciones de sus recursos físicos y tiene su propio *núcleo* que se encuentra entre el *hardware* y el sistema operativo. Según IBM “los hipervisores permiten que diferentes sistemas operativos (o instancias de uno único) sean ejecutados en el mismo *hardware* a la vez” [8]. Cabe observar que el rendimiento de los hipervisores no tiene una relación directa con el sistema operativo que se ejecuta en la máquina virtual. Su uso garantiza un mejor rendimiento porque se puede acceder y utilizar todos los recursos de *hardware* de forma instantánea y se abstraen directamente al hipervisor a través de una serie de interfaces.

Así, para efectos de este trabajo, y considerando todas las definiciones hasta aquí expuestas, se entenderá como hipervisor o “monitor de máquina virtual” a una capa de *software* de bajo nivel que se sitúa entre el *hardware* y los sistemas operativos para crear, ejecutar y administrar máquinas virtuales. Un hipervisor abstrae y reparte los recursos físicos, controla el acceso a los dispositivos y ejecuta/emula instrucciones privilegiadas, lo que permite que múltiples sistemas operativos (o instancias) funcionen de forma aislada y simultánea en el mismo equipo, desacoplando los sistemas operativos y aplicaciones de la infraestructura física y exponiendo las interfaces para un uso eficiente de los recursos.

La Figura 2.4 muestra el diagrama de un monitor de máquina virtual. En él se pueden ver las capas que lo conforman. Primero, se encuentra el servidor físico; después, se presenta el hipervisor; luego, está el sistema operativo invitado; finalmente, la última capa es la conformada por las máquinas virtuales.

Así pues, cuando se tiene diferentes máquinas virtuales, éstas se pueden ejecutar en distintos sistemas operativos y múltiples aplicaciones en la misma computadora física. Al igual que un servidor físico, una máquina virtual tiene una placa base, un procesador, memoria, un disco duro y una interfaz de red; sin embargo, estos recursos están disponibles y limitados por los servidores físicos que administran.

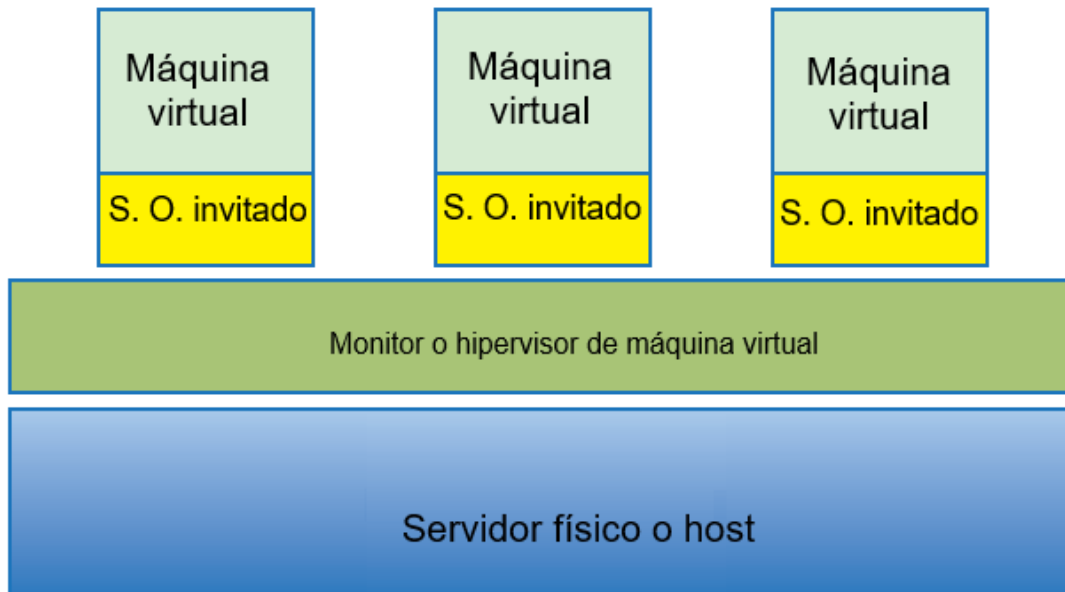


Figura 2.4: Monitor de Máquina Virtual básico. [Fuente: M. Portnoy, Fundamentos de la virtualización, Hoboken, Nueva Jersey, Estados Unidos: Sybex, 2023.]

2.4,1 Tipos de hipervisores

Aunque no existe una taxonomía clara o estandarizada para los hipervisores, se clasifican principalmente en tipo 1 o tipo 2, dependiendo de su posición en el sistema y si hay un sistema operativo subyacente. [24] Si se ejecuta directamente en el *hardware*, generalmente se considera un hipervisor tipo 1. Cuando hay un sistema operativo presente y el hipervisor se ejecuta como una capa separada, generalmente se considera un hipervisor tipo 2.

2.3.1 Hipervisor de tipo 1

De acuerdo con Delap, se “definen como hipervisores tipo 1 de sistemas nativos o ‘bare-metal’, aquellos que son instalados directamente en el *hardware* (como VMware ESX Server, Citrix XenServer) y sistemas operativos invitados (*guests*) a aquellos que son ejecutados sobre estos (hosts)” [8]. Los hipervisores de tipo 1 pueden aprovechar el concepto de paravirtualización mediante un conjunto de dispositivos de interfaces virtuales estándar (*virtio*) en las máquinas virtuales basadas en el *núcleo* (KVM), técnica de presentación de una interfaz de *software* similar al *hardware* oculto. Además, las configuraciones de arranque son administradas por el sistema básico de entrada/salida (BIOS) que se ejecuta al encender un equipo y se encarga de iniciar el *hardware* y arrancar el sistema operativo. Blawat menciona que “Las máquinas virtuales de generación 1 sólo pueden arrancar desde un disco conectado al controlador IDE o realizar un arranque por red desde un adaptador de red heredado ” [25]. En la Figura 2.5 se pueden ver las capas que conforma el hipervisor de tipo 1: primero, está el servidor físico; después, se encuentra el hipervisor. Al final, se encuentra cada sistema operativo que conforma la máquina virtual, la cual puede operar con base en Linux o en Windows.

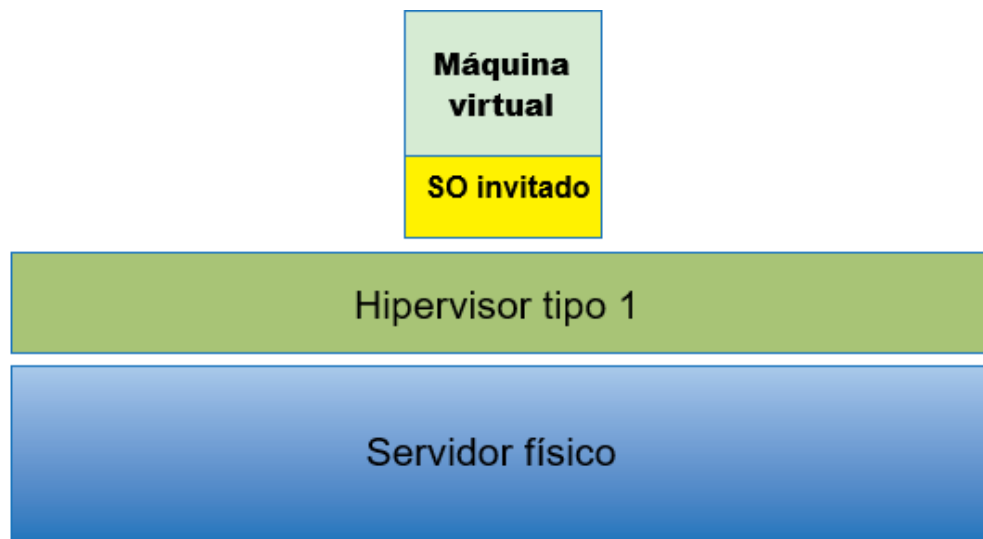


Figura 2.5: Hipervisor de tipo 1 [Fuente: M. Portnoy, Fundamentos de virtualización, Hoboken, Nueva Jersey, Estados Unidos: Sybex, 2023.]

Algunas ventajas de los hipervisores de tipo 1 son que son fáciles de instalar y configurar; son pequeños en tamaño, optimizados para dar la mayoría de los recursos físicos a las máquinas virtuales alojadas (invitados); generan menos sobrecarga, ya que sólo incluye las aplicaciones necesarias para ejecutar máquinas virtuales; y son más seguros, porque los problemas en un sistema invitado no afectan a los otros sistemas invitados que se ejecutan en el hipervisor. Además, un hipervisor de tipo 1 no requeriría el costo de un sistema operativo *host*, aunque desde un punto de vista práctico el análisis sería mucho más complejo e involucraría a todos los componentes que comprenden un cálculo del costo total de propiedad.

2.3.2 Hipervisor de tipo 2

De acuerdo con Delap, “Los hipervisores de tipo 2 a todos aquellos que, para ser ejecutados, dependen de un sistema operativo existente” [8]. De esa forma, los sistemas operativos virtuales se ejecutan en un tercer nivel por encima del *hardware* con recursos agregados y disponibles para el hipervisor. Ejemplos son Oracle VirtualBox y VMware Workstation Player. Blawat menciona que “Las máquinas virtuales de generación 2 se basan en Interfaz de firmware extensible unificado (UEFI), lo que nos ofrece características como arranque seguro” [25]. En suma, estas máquinas virtuales permiten arrancar desde el disco a la Interfaz de un Sistema de Computadora Pequeña (SCSI). Es un estándar para conectar y transferir datos entre computadoras eliminando la necesidad de utilizar métodos de arranque basados en controladores de Electrónica de Unidad Integrada (IDE). Por otro lado, ofrecen soporte para el arranque por red a través del adaptador de red sintético.

La Figura 2.6 muestra un ejemplo esquemático de un hipervisor de tipo 2. Se puede ver en la primera capa el servidor físico. En segundo lugar, se encuentra el sistema operativo que presenta sistema anfitrión. Después se presenta el hipervisor. Como última capa, figuran los sistemas operativos que se incorporan en cada máquina virtual, por ejemplo, las distintas versiones Linux o Windows.



Figura 2.6: Ejemplo de un Hipervisor tipo 2 [Fuente: M. Portnoy, Fundamentos de la virtualización, Hoboken, Nueva Jersey, Estados Unidos: Sybex, 2023.]

El hipervisor de tipo 2 actúa como una aplicación de *software* típica, debido a que interactúa con el sistema operativo del *host*. Sin embargo, su distinción radica en el hecho de que proporciona uno o más entornos virtualizados de máquinas virtuales. Cada una de estas máquinas virtuales tiene su propio sistema operativo (invitado) y pueden tener varias aplicaciones ejecutándose en ellas.

La Figura 2.7 muestra Oracle VirtualBox con dos sistemas de servidor Linux instalados. Se pueden apreciar las distintas opciones que presenta la interfaz gráfica. En la parte izquierda se presentan las máquinas virtuales que ya se encuentran ejecutadas y están disponibles para que sea utilizado. Se puede ver un claro ejemplo del hipervisor tipo 2 que se encuentra como capa después de un sistema operativo existente.

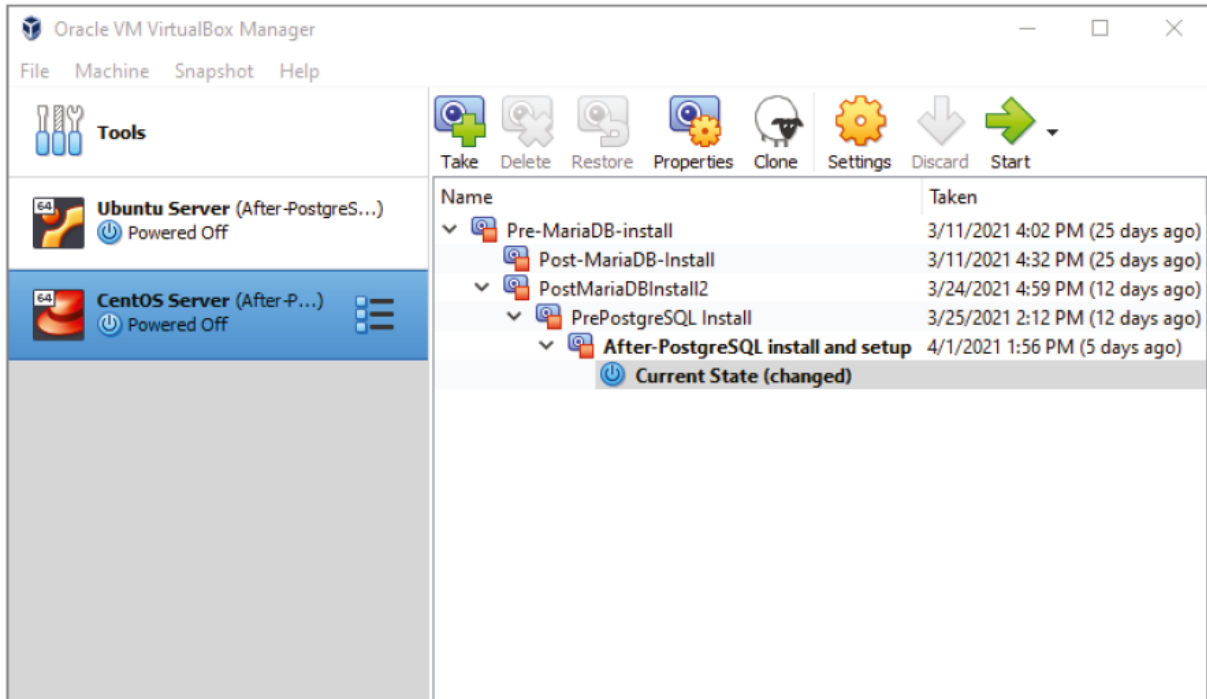


Figura 2.7: Hipervisor Oracle VirtualBox tipo 2 [Fuente: R. B. Christine Bresnahan, Dominando la administración de sistemas Linux, Indianápolis, Indiana, Estados Unidos: John Wiley & Sons, Inc., 2021.] [18]

2.4 Beneficios y retos de la virtualización

El principal beneficio de las máquinas virtuales es el aislamiento total que proporcionan; es decir, cada máquina virtual ejecuta su propio *núcleo*. Dicho de otra manera, las máquinas virtuales son una opción viable cuando se tiene una cantidad limitada de recursos de *hardware* y se desea aislar unos pocos procesos.

En la Figura 2.8 se puede ver un ejemplo de cómo mediante una computadora físico encuentran divididos en distintas etapas aquí las máquinas virtuales se encuentran divididas para cada uno y tiene su propio recurso asignado.

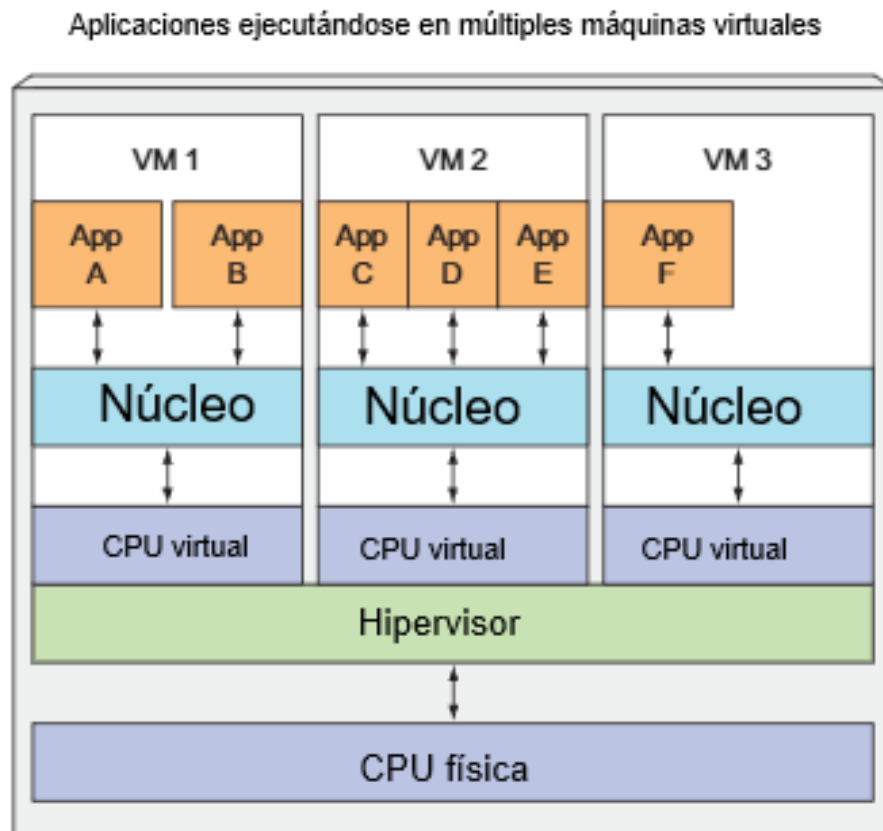


Figura 2.8: Ejemplo de uso de CPU en la Máquina Virtual. [Fuente: M. Luksa, *Kubernetes en acción*, Shelter Island, Nueva York, Estados Unidos: Manning Publications, 2017.]

Aunado a lo anterior, las máquinas virtuales proporcionan otros beneficios que son comercialmente significativos:

- **Gestión de *software*:** las máquinas virtuales proporcionan una abstracción que puede ejecutar la pila de *software* completa, incluso incluyendo sistemas operativos antiguos. Una implementación típica podría ser algunas máquinas virtuales que ejecutan sistemas operativos heredados, muchas que ejecutan la versión estable actual del sistema operativo y algunas que prueban la próxima versión del sistema operativo.

- Administración del *hardware*: una razón para tener varios servidores es que cada aplicación se ejecute con la versión compatible del sistema operativo en computadoras separadas. Las máquinas virtuales permiten que estas pilas de *software* separadas se ejecuten de manera independiente, pero compartan *hardware*, consolidando así la cantidad de servidores.
- Ejecutar diferentes sistemas operativos al mismo tiempo: esto permite que el sistema operativo ejecute *software* escrito en otro sistema operativo, por ejemplo: *software* de Windows en Linux sin necesidad de reiniciar o apagar el equipo.
- Instalación rápida: los proveedores de *software* pueden utilizar máquinas virtuales para ofrecer configuraciones de *software* completas. Esto permite agrupar arquitecturas complejas, incluidos múltiples sistemas operativos y aplicaciones en un solo dispositivo.
- Pruebas de aplicaciones: esta herramienta sirve para que los desarrolladores creen y prueben aplicaciones. Una vez instaladas, las máquinas virtuales se pueden congelar, reiniciar, copiar, realizar copias de seguridad y transferir entre *hosts* como si fueran dispositivos.
- Consolidar recursos: de acuerdo con Simón, “si las computadoras sólo se utilizan parcialmente y cada una tiene su propio sistema operativo” [26]. Es decir, permiten consolidar múltiples máquinas virtuales en un solo sistema de escritorio o portátil sin tener que ejecutar múltiples tareas físicas.

2.5 Comparativa de plataformas: Proxmox VE, VMware ESXi, Hyper-V, VirtualBox

Para elegir una plataforma de virtualización, es importante considerar el entorno y las necesidades específicas, debido a que cada una ofrece tanto ventajas como limitaciones:

VMware ESXi: predomina como un hipervisor tipo 1 en el ámbito empresarial. Aunque la versión básica es gratuita, para utilizar funciones más avanzadas se debe ocupar una licencia [27].

- Proxmox VE: es una plataforma de clúster de hipervisores de servidores, donde por “clúster” se entiende a “un ‘grupo’ de múltiples nodos de servidor físico” [28]. Es una alternativa al contar con una licencia gratuita y al ser de código abierto que trabaja con virtualización de Máquina Virtual Basada en Kernel (KVM) utilizando un emulador rápido (QEMU) para una virtualización completa y contenedores de Linux (LXC), y que se integra mediante una interfaz *web* donde puede ingresar en cualquier navegador. Es recomendable para laboratorios y empresas pequeñas debido a que su sistema es básico para su uso en computadoras actuales para no adquirir nuevo equipo. Además, hay la opción de adquirir una suscripción de paga en caso de requerir soporte técnico profesional.
- Microsoft Hyper-V: es la tecnología de hipervisor de nivel empresarial de Microsoft. Está integrado en Windows Server y Windows. Ofrece virtualización para entornos de Microsoft. Se debe contar con una licencia para su uso, así como considerar el rendimiento con los sistemas operativos, debido a que no puede ser inferior a lo recomendado por parte de Microsoft. Sus mayores ventajas son la administración a través de System Center y la compatibilidad con cargas de trabajo en Windows [29].

- Virtual Box: desarrollado por Oracle y con virtualización de tipo 2, está orientada a usuarios domésticos. Destaca por la facilidad que tiene de usarse mediante múltiples sistemas. No obstante, su rendimiento es lento dependiendo de los valores que tiene, especialmente en computadoras con pocos recursos. Esto hace que no sea tan adecuado para el uso en entornos empresariales [30].

Mientras ESXi y Hyper-V son para entornos empresariales, utilizar Proxmox VE es una alternativa al tener toda la funcionalidad sin ninguna restricción y se pueda ocupar con equipos de bajos recursos. VirtualBox es otra opción para pruebas y desarrollo en equipos personales.

La Tabla 2.1 presenta un análisis comparativo de las diferencias que tiene cada hipervisor de forma resumido entre ello radica sus características más importantes. Este análisis comparativo de Proxmox VE con VMware ESXi, Hyper-V y VirtualBox se concentra en ciertas cualidades que presenta Proxmox VE versión 8.3, como el tipo de hipervisores —si son de tipo 1 o tipo 2—, el sistema anfitrión, la licencia, etcétera. Se muestran sus características como el número de clúster recomendado, si cuentan con alta disponibilidad, los tipos de almacenamiento que se puede ocupar, el sistema operativo que se puede instalar, la capacidad máxima que se le puede asignar a una máquina virtual de memoria RAM, la unidad central de procesamiento virtual y el disco virtual máximo.

Característica	Proxmox VE	VMware ESXi	Hyper-V	VirtualBox
Tipo de Hipervisor	Tipo 1 (Bare-metal) con (KVM/LXC)	Tipo 1 (Bare-metal)	Tipo 1 (Bare-metal)	Tipo 2 (Alojado)
Licencia	Open Source (AGPLv3)	Gratis (limitado) / Pago (vSphere)	Incluido en Windows Server	Open Source (PUEL para enterprise)
Sistemas Anfitriones	Linux (Debian-based)	Bare-metal (<i>hardware</i> certificado)	Windows Server / Windows 10/11 Pro+	Windows, Linux, macOS, Solaris

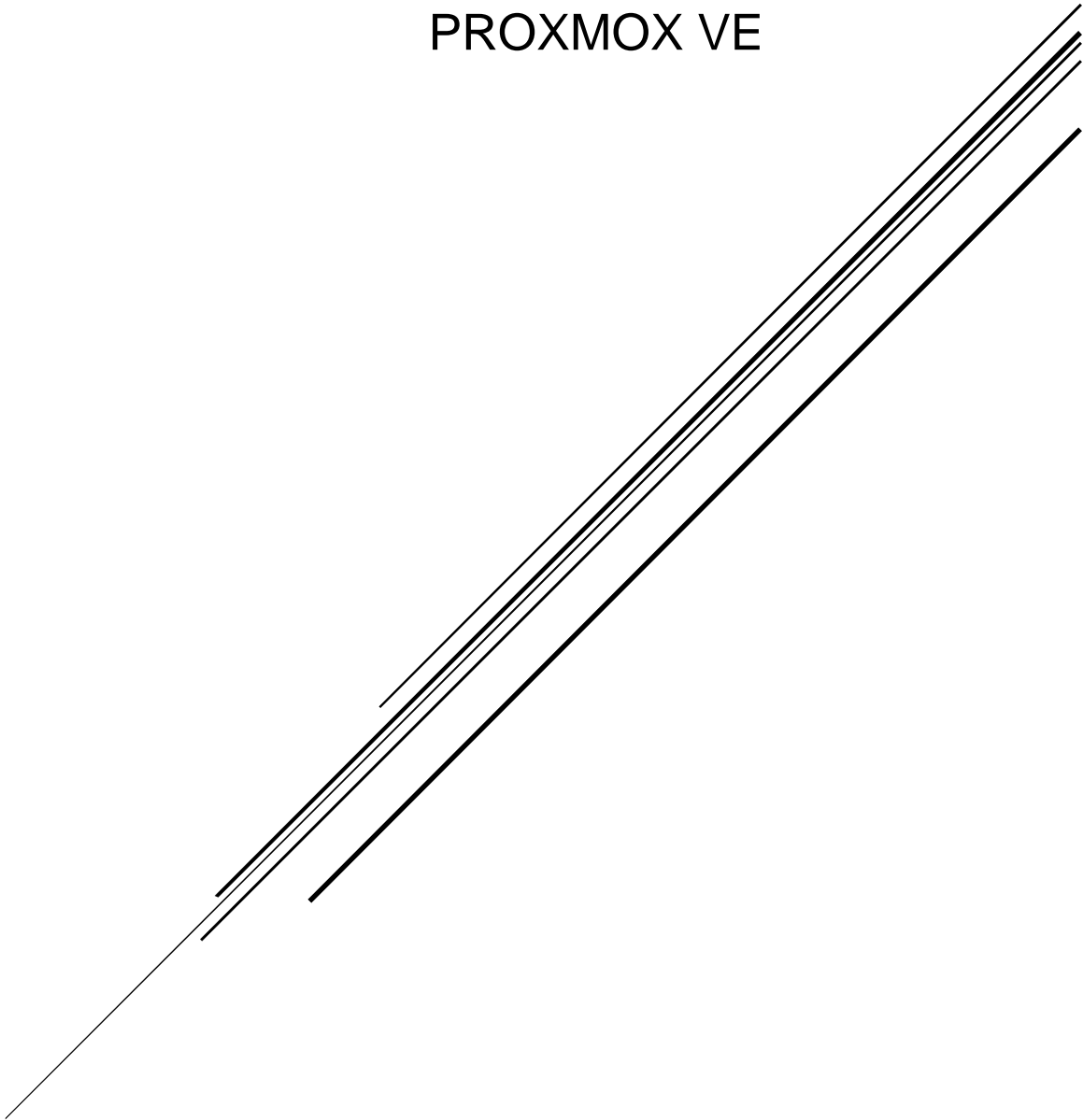
Característica	Proxmox VE	VMware ESXi	Hyper-V	VirtualBox
Soporte de SO Invitados	Linux, Windows, BSD basado en UNIX	Windows, Linux, BSD basado en UNIX	Windows, Linux	Windows, Linux, macOS, Solaris
Clúster de hosts	Sí (en clúster) recomendado hasta 32-64 (según recursos) [31]	Sí (vMotion) Sin límite duro, Hasta 96 hosts por [31]	Sí (Quick Migration) Hasta 64 nodos por cluster [31]	No cuenta con la opción de incluir clúster.
Almacenamiento	ZFS, Ceph, LVM, NFS	VMFS, NFS, vSAN	VHDX, NTFS, CSV	VDI, VMDK, VHD
API/Gestión	REST API, Web GUI	vSphere API (REST/SDK)	PowerShell, WMI	CLI, SDK Python
vCPU por VM	Hasta 255 (según kernel/QEMU) [31]	Hasta 768 [31]	hasta 240 [31]	Depende de la capacidad del sistema que sea instalado
RAM máxima por VM	6 TiB (recomendado, ampliable) [31]	24 TiB [31]	12 TiB [31]	Depende de la capacidad del sistema que sea instalado
Nº máximo de discos por VM	32 (virtio/IDE/SCSI) [31]	256 vNVMe [31]	256 VHDX [31]	Depende de la capacidad del sistema que sea instalado
Versión <i>hardware</i> VM más reciente	QEMU/KVM 7.2/8.x, VirtIO latest VM y en la actualidad 9	<i>hardware</i> v21 (ESXi 8.0 U2)	versión 10.0 (Windows Server 2022)	Versión 7.0.12 es el más actual
Alta Disponibilidad (HA)	Sí (HA en clúster)	Sí (vSphere HA)	Sí (Failover Cluster)	No cuenta con la opción de incluir.
Contenedores	Sí (LXC)	No	Sí (Windows Containers)	No cuenta con la opción de incluir

Tabla 2.1 Comparativa de hipervisores. [Fuente: R. B. Christine Bresnahan, Dominando la administración de sistemas Linux, Indianapolis, Indiana, Estados Unidos: John Wiley & Sons, Inc., 2021. D. G. Mark Minasi, Dominando Windows Server 2008 R2, Indianapolis, Indiana, Estados Unidos: Wiley Publishing, Inc., 2010. P. B. Abraham Silberschatz, Conceptos del sistema operativo, Hoboken, Nueva Jersey, Estados Unidos: John Wiley & Sons, Inc., 2008. P. Matthew, Fundamentos de la virtualización, Indianapolis, Indiana: John Wiley & Sons, Inc., 2012. H. D. Chiramal, Mastering KVM Virtualización, Birmingham, Reino Unido: Packt Publishing Ltd., 2016.] [32]

Finalmente, para desarrollar este trabajo, se utilizó el sistema operativo con distribuciones de Linux Ubuntu en su versión 20.04. La elección de este sistema operativo obedeció a que pertenece a la familia Linux, en la cual está basado Proxmox VE —en su caso, Debian Linux— y, aunque la virtualización bien puede aplicarse en otros entornos, Linux tiende a ser el sistema operativo más recurrente para la instalación de máquinas virtuales. Asimismo, se empleó una virtualización completa a nivel de servidor con KVM y QEMU como emulación soportada nativamente por Proxmox VE. Si bien a lo largo de este capítulo se observó que existen otras modalidades como paravirtualización, virtualización de aplicaciones entre otros, cabe aclarar que éstas fueron explicadas sólo para dar contexto, pero se decidió que no formaran parte del alcance del trabajo. Como hipervisor principal, se utilizó Proxmox VE en su versión 8.3 debido a que es un *software* libre y de código abierto, lo cual brinda una ventaja económica y de autonomía institucional. Además, permite combinar máquinas virtuales en un mismo entorno, lo cual es una ventaja importante. Además su interfaz de administración web es amigable y posee soporte para *snapshots*, respaldo y gestión de clústeres, lo que lo hace una herramienta atractiva, y aunque en este mismo capítulo se han descrito otras plataformas (VMware, Hyper-V, VirtualBox, etc.), se decidió no trabajarlas porque son de paga e imponen mayores requisitos de *hardware*, razón por la cual no se ajustan del todo al objetivo del trabajo, que es aprovechar los equipos de bajos recursos en el Laboratorio de Telecomunicaciones. Así, de aquí en adelante, el análisis a desarrollar se acotará exclusivamente al sistema operativo Linux Ubuntu versión. 20.04, la virtualización KVM con QEMU y el sistema de hipervisor Proxmox VE versión. 8.3.

CAPÍTULO 3

ARQUITECTURA Y CARACTERÍSTICAS DE PROXMOX VE



Se presenta una breve descripción del sistema de Proxmox VE (Entorno virtual) 8.3 donde se explica por qué se le prefirió frente a la competencia, abordando además sus principales características para conocer el funcionamiento del sistema. Se aborda el uso de aplicación de autenticación de dos factores para ingresar, ya sea como administrador o usuario. Por último, se detalla que hay distintas opciones para mantener la información en caso de realizar cambios y evitar estropear el entorno virtual.

3.1 Introducción a Proxmox VE

Proxmox VE es una plataforma de entorno virtual para administrar clústeres de hipervisores. Se trata de un *software* de sistema de virtualización de servidores de código abierto basado en Debian GNU/Linux. Este proyecto empezó con Dietmar y Martin, dos desarrolladores de Linux con nombres, quienes se percataron que en su momento “OpenVZ²⁹ no contaba con ninguna interfaz gráfica que fuera administrada por el usuario y que realizara copias de Seguridad” [33].

“La primera versión de Proxmox VE fue lanzada el 15 de abril del 2008” [33]. Con ella se buscaba combinar servidores, donde el uso se basará en contenedores (usando contenedores Linux LXC) y la virtualización de máquinas virtuales basadas en el *núcleo* (usando KVM) en Emulador rápido (QEMU) para tener una virtualización completa. Con este *software*, la interfaz web se puede administrar utilizando cualquier navegador. Cuenta con una interfaz fácil de usar y no presenta ninguna restricción. También tiene la opción de ocupar la línea de comandos para realizar tareas más avanzadas.

²⁹ OpenVZ, se centra en la virtualización del sistema, que le permite ejecutar el sistema operativo completo dentro de un contenedor, donde puede iniciar sesión SSH, agregar usuarios, ejecutar servidores Apache, etc. [28]

El uso de Proxmox VE en un clúster presenta como ventajas: “Arquitectura de clúster multimaestro: todo el clúster se puede administrar desde cualquier nodo, así como el sistema de archivos de configuración distribuida (pmxcfs) guarda los archivos de configuración con un sistema de archivos basado en bases de datos y garantiza que la información de configuración de todos los nodos se sincronice en tiempo real a través de sincronización de recursos del clúster (corosync)” [28]. Además, las máquinas virtuales se pueden migrar entre nodos y presenta un *firewall* basado en clústeres y servicios de alta disponibilidad; tanto es así que permite crear una infraestructura virtual de clase empresarial de forma gratuita sin adquirir una licencia de suscripción.

Algunos puntos clave para comprender este *software* son:

- Proporciona un entorno de clúster multinodo para la virtualización.
- Proporciona alta disponibilidad (HA) de máquinas virtuales.
- Se administra en cualquier navegador *web* y proporciona permisos basada en roles.
- Admite los principales tipos de almacenamiento.
- No depende de ningún proveedor y se puede usar sin la necesidad de una suscripción.

En la Figura. 3.1 se puede ver el diagrama básico de cómo interactúan los componentes cuando una máquina virtual de Proxmox VE accede a la red.

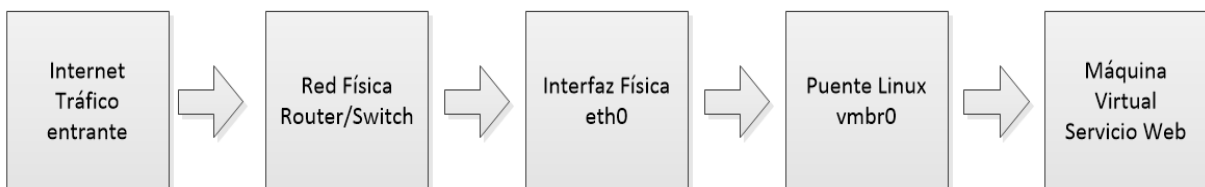


Figura 3.1: Diagrama de Flujo de datos de como la MV accede a la red. [Elaboración Propia]

La elección del sistema con entorno virtual (VE) del *software* Proxmox VE como hipervisor tipo 1 obedeció a la consideración de sus múltiples ventajas en términos de costos, flexibilidad, compatibilidades, requerimientos previos, opciones de gestión, capacidades, oportunidades y alternativas. Algunas de las razones que más incidieron en la decisión y que marcan las ventajas de Proxmox VE frente a su competencia fueron:

- **Bajo costo:** es un *software* libre (licencia AGPLv3)³⁰. No hay necesidad de pagar licencias pesadas por uso del hipervisor, lo que facilita que el proyecto sea viable para la institución dado su limitado presupuesto.
- **Flexibilidad y compatibilidad:** el *software* es flexible para la virtualización completa y a nivel sistema operativo; es decir, es compatible con distintas aplicaciones de código abierto que basadas en desarrollos de Linux. Además, cuenta con la opción de soporte comercial para producción. [28]
- **Características clave:** permite trabajar mediante una interfaz gráfica o bien sobre su consola. Cuenta con ciertas herramientas para alta disponibilidad, clúster, interfaz de programación de aplicación (API), soporte en distintas versiones de almacenamiento, respaldos y una política de seguridad.
- **Uso en producción:** comparado con VirtualBox para empresas pequeñas y micro empresas, Proxmox VE permite un mejor aprovechamiento de los recursos, al ser un hipervisor tipo 1 que se instala directamente en el *hardware*. En contraparte, Virtual Box, al ser un hipervisor de tipo 2, debe ser instalado dentro de un sistema operativo ya existente y en sus nuevas versiones demanda requisitos mínimos muy específicos,

³⁰Es una licencia Pública General GNU Affero, versión 3 (AGPLv3) esto Permite: Usar, modificar y distribuir el *software* libremente y permite acceder al código fuente completo. [28]

como contar con un “Procesador Intel VT-x o AMD-V habilitado. Al menos 512 MiB de RAM. Al menos 15 GiB de espacio libre en el disco duro para la instalación de los sistemas operativos virtuales” [34]. Esto hace a Proxmox VE más flexible que otros hipervisores.

- Almacenamiento: la compatibilidad con distintos tipos de almacenamiento resulta similar al que ofrece ESXi en su versión Premium. Esto demuestra que, aunque se trata de una herramienta de libre acceso, posee características que lo vuelven competitivo frente a las de paga. [27].
- Oportunidad de adquisición de conocimientos: el manejo de Proxmox VE con las muchas características de su consola obliga al administrador a familiarizarse con el trabajo en entornos virtualizados, lo que le brinda un mayor conocimiento sobre el uso de éste y otros hipervisores, contribuyendo a su formación.

Por supuesto, Proxmox VE también presenta algunas desventajas frente a opciones competidoras que fueron tomadas en cuenta al momento de efectuar la decisión de utilizarlo como hipervisor para la creación de máquinas virtuales en este trabajo. Las principales fueron:

- Soporte empresarial: otros competidores como ESXi y Hyperf-V gozan de un mejor ecosistema, mucho más potente, utilizados en empresas potenciales en el mercado. Estas características son especialmente útiles para proyectos de gran escala, aunque se estimó que en realidad no son ventajas definatorias en el caso de este trabajo dadas sus dimensiones.
- Compatibilidad de *hardware*: ESXi presenta una lista de *hardware* compatible (HCL) certificado que hace más sencilla la instalación, mientras que en Proxmox VE se requiere contar con conocimiento en Linux para realizar tareas más avanzadas.

Esto puede ser un problema para los iniciados, aunque en el caso del presente trabajo se cuenta con la experiencia suficiente para trabajar sin la ayuda de dicha guía.

- Uso de clústeres: si un equipo cuenta con varios nodos, todos ellos se deben estar encendidos para un funcionamiento óptimo. Ésta es una desventaja importante si sólo se desea trabajar un nodo a la vez, sobre todo en términos de ahorro de energía; sin embargo, dado que la aplicación se realizaría en un laboratorio de cómputo, no se estimó como un problema mayor pues es habitual que estos espacios operen teniendo todos sus nodos encendidos al mismo tiempo.

De este modo, si bien a pesar de sus deficiencias, se estimó que Proxmox VE cumplía con muchas características ventajosas que orillaron a decantarse por él para la realización del presente trabajo, al tiempo que sus desventajas no serían lo suficientemente importantes como para comprometerlo.

3.1.1 Tecnologías base

Proxmox VE tiene como base las tecnologías de contenedores de Linux (LXC) y de máquinas virtuales basadas en *kernel* (KVM):

- Contenedores de Linux (LXC): de acuerdo con Taboada, “LXC es una tecnología donde se ejecutan las aplicaciones que corren en los contenedores Linux (LXC) de forma protegida, mediante la aplicación de los procesos como AppArmor³¹ y cgroups³²” [12].

³¹AppArmor es un módulo de seguridad del *kernel* de Linux que controla los accesos de los programas a determinados recursos del sistema, como archivos, directorios y sockets de red. Funciona definiendo perfiles de seguridad para cada aplicación, especificando a qué recursos pueden acceder y en qué condiciones [12].

³² Los grupos de control, son una característica de Linux que permite limitar, priorizar y controlar el uso de recursos de un conjunto de procesos [12].

Esto permite ejecutar aplicaciones sobre el propio *núcleo* del hipervisor; sin embargo, presenta limitaciones en el consumo de recursos y permisos que se pueda acceder dentro del *núcleo*, por lo cual sólo es posible ejecutarse en aplicaciones de Linux.

- Máquinas virtuales basadas en *kernel* (KVM): son un módulo del *núcleo* añadido a Linux para una virtualización completa. Permiten crear máquinas virtuales aisladas y completamente independientes del sistema operativo *host*, pero requieren que la función de virtualización esté habilitada en el BIOS. Según reza en la página de Debian “(KVM) es una solución de virtualización completa para Linux en *hardware* x86 (incluyendo *hardware* de 64-bits) que contienen las extensiones de virtualización Intel VT (Tecnología de virtualización) o AMD-V (Virtualización AMD)” [12]. La idea central consiste en un módulo de *kernel* cargable que proporciona la infraestructura de virtualización básica y un módulo específico del procesador. “KVM es compatible con diversos sistemas operativos de máquinas virtuales, incluyendo Linux y Windows” [35].

Para ejecutar las (Máquina virtual basada en *núcleo*) KVM, se necesita un emulador rápido, en este caso (QEMU). Eduardo menciona que “QEMU es un *software* de emulación de código abierto que permite la ejecución de máquinas virtuales” [14]. Vale la pena decir que puede emular una gran variedad de arquitecturas de CPU y dispositivos para aprovechar las capacidades de KVM para la virtualización de *hardware*. La instalación del sistema operativo y las aplicaciones se realiza sobre esta emulación. Al respecto, hay que tomar en cuenta que se proporciona una capa de abstracción suficiente como para permitir la ejecución de cualquier sistema operativo de virtualización y que los contenedores Linux (LXC) no puede realizar. Cabe añadir que su rendimiento no es compatible al ejecutar una máquina virtual con el mismo conjunto de instrucciones que el *hardware* que ejecuta el hipervisor, debido a que la conversión del conjunto de instrucciones del procesador se debe realizar durante la emulación.

Según Ahmed “Emulador rápido (QEMU) proporciona una interfaz de emulación y virtualización que puede ser programada o controlada de otro modo por un usuario” [35]. La Figura. 3.3 muestra un diagrama de las relaciones entre KVM y QEMU, dando una idea de cómo trabajan juntos en los entornos de virtualización. Hay procesos normales como `/bin/bash` y `/usr/bin/xcalc` que coexisten con las máquinas virtuales sobre el mismo sistema operativo anfitrión, compartiendo los mismos recursos físicos. Por otro lado, se encuentra los sistemas en el modo virtualizado, que son las máquinas virtuales invitadas (*guest mode*) que usan dispositivos virtuales (QEMU I/O). El *kernel* de Linux y los controladores KVM trabajan para acelerar las instrucciones de las máquinas virtuales al tiempo que QEMU utiliza KVM para delegar intrusiones críticas para el *hardware* físico. El CPU/RAM del *hardware* ejecuta la carga de trabajo. Por último, la red y el almacenamiento son los recursos que permiten que las máquinas virtuales se comuniquen al exterior y almacenen información. Luego, en la Figura. 3.4., se puede ver claro un ejemplo del diagrama anterior entre las relaciones de KVM y QEMU con sus diferentes etapas que se mencionaron anteriormente. Hay que hacer mención de que todos deben colaborar para que se pueda trabajar en conjunto, ya que cada proceso es el resultado que hace posible la virtualización mediante el *software* Proxmox VE [Figura 3.2].

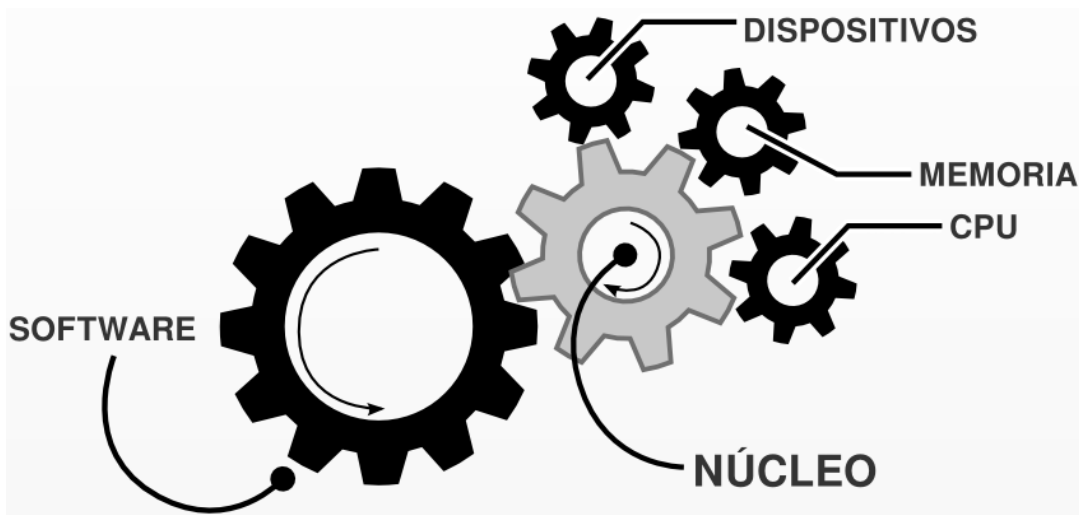


Figura 3.2: Diagrama integración entre *software* hipervisor y *hardware*. [Fuente: W. Commons, «Wikipedia,» Wikimedia Foundation, [En línea]. Available: https://upload.wikimedia.org/wikipedia/commons/e/ec/Infografia_NucleoLinux.png. [Último acceso: 1 Octubre 2025 [36]]

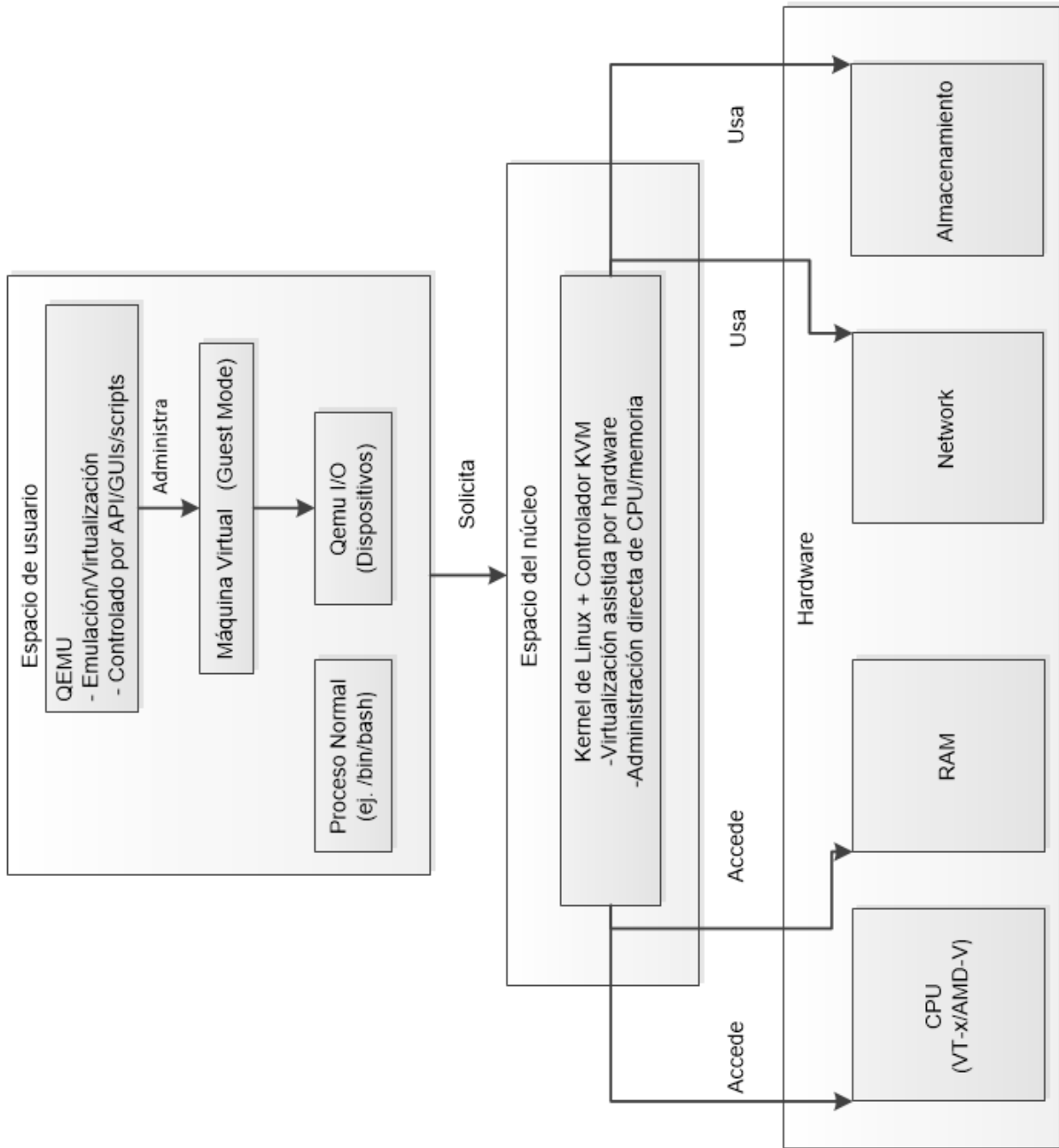


Figura 3.3: Diagrama de Relación entre KVM y QEMU en entornos virtualizados. [Elaboración propia]

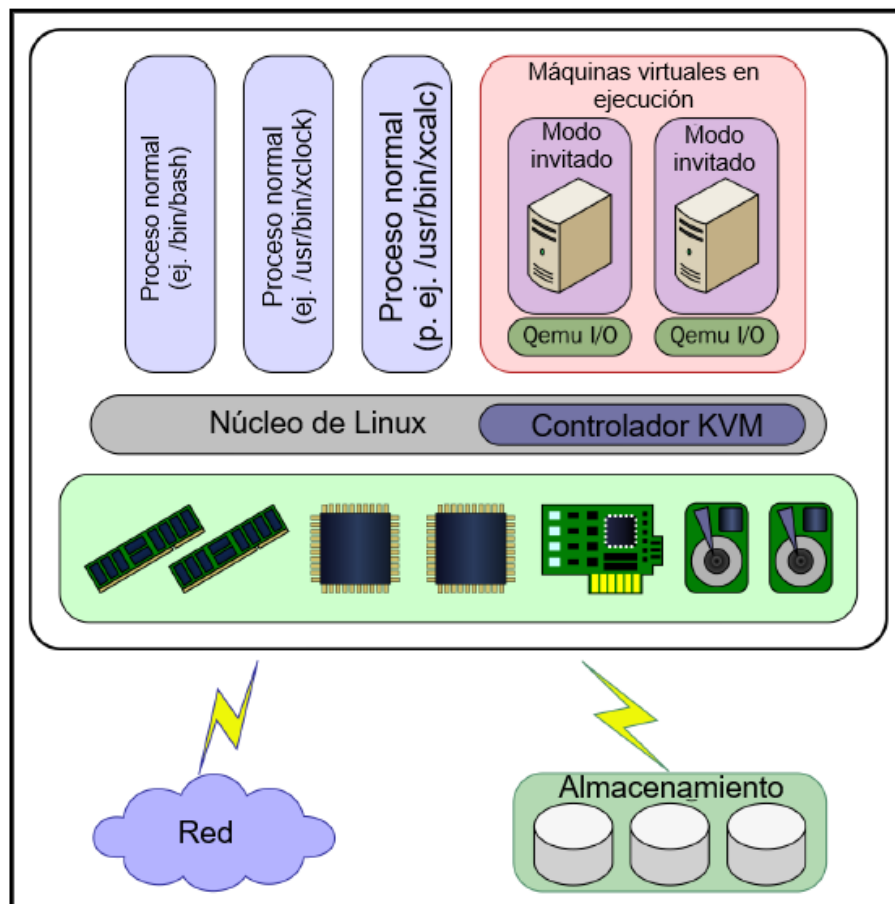


Figura 3.4: Relación entre KVM y QEMU en entornos virtualizados. [Fuente: R. Goldman, *Learning Proxmox VE*, Birmingham, Reino Unido: Packt Publishing Ltd., 2016.]

3.1.2 Sistemas de archivos

Algunos archivos que ocupa Proxmox VE son:

- Cuarto sistema de archivos extendido (Ext4): “El Ext4, es un sistema de archivos de diario para Linux, desarrollado como sucesor de Ext3, introducido en 2008” [37]. Puede decirse que administra y admite archivos más grandes a diferencia de Ext3, permitiendo almacenar más información. Vinchin destaca que “es una opción ideal para cargas de trabajo generales, visto que soporta una capacidad de sistema de 1 exabyte(EB) y archivos de hasta 16 terabyte (TB)” [38].

- Sistema de archivos de 64 bits (XFS): “desarrollado originalmente por Silicon Graphics a principios de los años 90, es un sistema de archivos de alto rendimiento diseñado para escalabilidad” [31]. Es decir, su reconocimiento se basa en la administración de grandes cantidades de archivos y sistemas de almacenamiento con gran capacidad. iBoysoft menciona que “una de sus ventajas es que soporta archivos y sistemas de archivos grandes (hasta 8 exabytes), lo que lo hace ideal para aplicaciones que requieren grandes cantidades de almacenamiento” [32].

3.2 Interfaz gráfica *web* y e interfaz de línea de comandos

El entorno virtual de Proxmox VE cuenta con una interfaz gráfica que permite acceder a cualquier navegador *web*, permitiendo la administración de clústeres. Su sistema en línea facilita tareas relacionadas con el control de entornos virtualizados. Para acceder a la interfaz de Proxmox VE, primero hay que ingresar a la URL —en este caso, se empleó la dirección IP 192.168.0.127—. Se mostrará la pantalla de inicio de sesión, donde se deben ingresar las credenciales. La autenticación se realiza mediante Linux PAM (módulos de autenticación conectables). Este módulo “es un sistema de autenticación centralizado basado en Linux que utiliza una arquitectura conectable y modular, proporcionando un esquema de autenticación común para todo el sistema operativo” [39]. Por el contrario, la autenticación de Proxmox VE (PVE) almacena las contraseñas en `/etc/pve/priv/shadow.cfg`. Ésta es la autenticación preferida al crear usuarios generales, además de los usuarios administrativos. De esta manera, el usuario solo accede a Proxmox y no a otras partes del sistema operativo [Figura. 3.5]. Para iniciar la sesión, se debe ingresar como usuario *root* con la contraseña que se creó.

La parte superior de la interfaz gráfica de usuario presenta accesos para la administración del sistema. De acuerdo a la Figura. 3.6, se pueden observar opciones como “documentación”, “crear VM (Máquina Virtual)”, “crear CT (Contenedor)” y un campo de búsqueda para navegar dentro del entorno. La información del usuario autenticado aparece en la esquina superior derecha, —en este caso se muestra el usuario root@pam— y se puede ver opciones como “idioma (*language*)”, “contraseña (*password*)”, “autenticación en dos pasos (*two-factor authentication*)”, “configuración del perfil (*user settings*)” y “cerrar sesión (*logout*)”. El acceso cuenta con privilegios administrativos. Además, se comparte la versión del sistema —en este caso, Entorno Virtual 8.3.0.—.

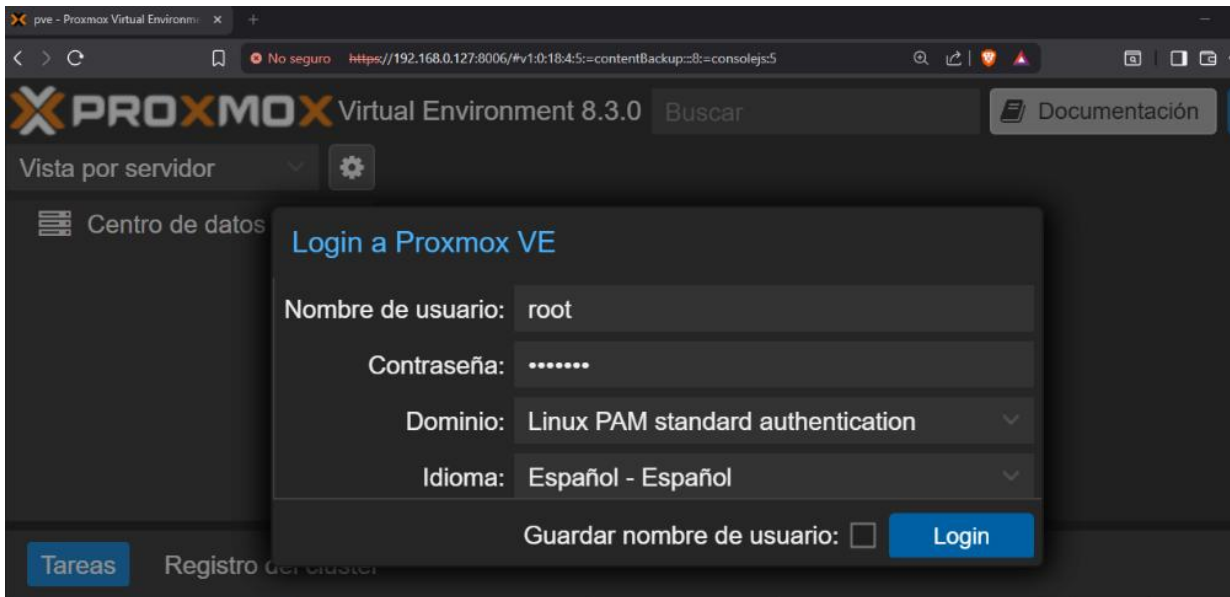


Figura 3.5: Pantalla Principal de Proxmox VE. [Captura Propia]

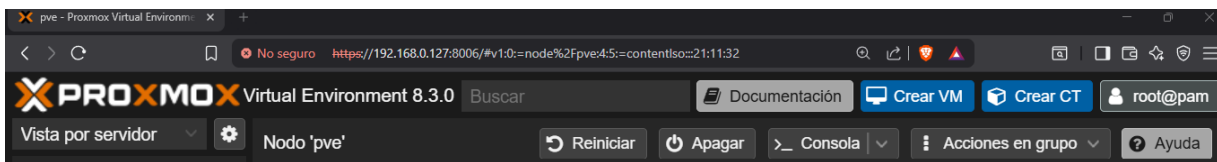


Figura 3.6: Opciones del Menú de Proxmox VE. [Captura Propia]

El Centro de Datos (*Datacenter*) en Proxmox VE es la sección donde se administran los recursos y configuraciones del clúster. El menú lateral proporciona accesos como “cluster”, “ceph”, “almacenamiento”, “respaldo”, “usuarios”, “permisos”, y más. La sección de Salud muestra el estado del clúster. Se aprecia que la comunicación entre los nodos es estable. Además, se presentan detalles sobre el estado de los nodos —en este caso, 2 están en línea y ninguno está desconectado [Figura. 3.7]. La sección de Guests muestra información sobre las máquinas virtuales y contenedores LXC dentro del clúster —en este caso, hay 2 máquinas virtuales activas y 2 detenidas, no hay ningún contenedor que se encuentre en ejecución y uno se encuentra detenido [Figura. 3.7]—.



Figura 3.7: Sección de Salud de Proxmox VE. [Captura Propia]

3.3 Gestión de usuarios, roles y permisos

- **Usuarios:** El menú de usuarios permite crear, editar, eliminar o cambiar las contraseñas de usuarios en un clúster de Proxmox VE. Con respecto de la asignación de grupos, es necesario que el grupo sea creado desde el menú grupos antes de asociar usuarios. En caso de ser administrador otorgar los permisos al usuario que se va crear, para crear un nuevo usuario se debe hacer clic en “Agregar” y completar la información requerida. [Figura. 3.8].

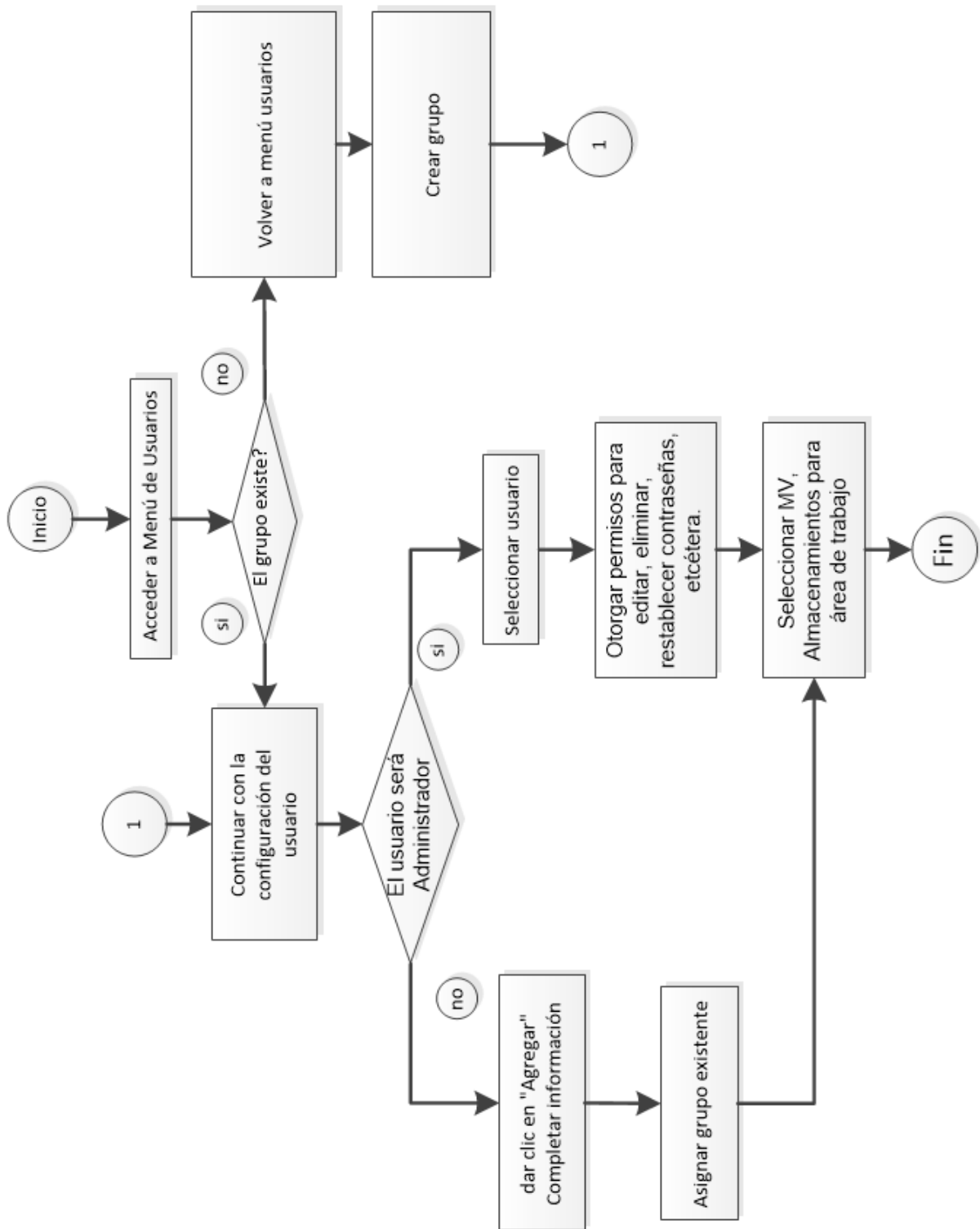


Figura 3.8: Diagrama de procesos de usuarios. [Diseño Propio]

- Roles: El menú Roles presenta todos los roles predeterminados de Proxmox VE junto con sus privilegios asociados. Estos roles no pueden ser eliminados ni editados. En relación con los privilegios, cada rol tiene distintos niveles de acceso aplicables a entidades dentro de un clúster de Proxmox VE.
- Permisos: El menú Permisos se utiliza para configurar los niveles de acceso de usuarios en el clúster. Donde se muestra una lista completa de permisos ya asignados que pueden configurarse desde el menú de permisos —“Máquinas virtuales y almacenamiento”—. Si los permisos se configuran desde el menú “Centro de datos”, es necesario introducir la ruta de la entidad correspondiente. Es importante comentar que el sistema permite la administración tanto a nivel de grupos como de usuarios individuales.

3.4 Seguridad básica del sistema

3.4.1 Autenticación y control de acceso

Una de las medidas más importantes es la autenticación en dos pasos mediante Google Authenticator. El acceso se realiza mediante una prueba cerrando sesión y volviendo a ingresar. Para terminar el proceso de autenticación, el sistema debe solicitar el código generado. Esto, debido a que únicamente el acceso del usuario está autorizado. Estos perfiles se crearon de forma secundaria para poder realizar procesos de automatización de tareas. “A partir de Proxmox VE 3.4, se admite la autenticación de dos factores (TFA)” [39]. Se debe tomar en cuenta que no hay un cuadro de entrada de contraseña en la ventana de creación de usuarios cuando se selecciona la autenticación estándar de Linux PAM. “El cuadro de entrada de contraseña está disponible en la ventana de creación de usuarios al crear usuarios en el servidor de autenticación de Proxmox VE”. [39]

3.4.2 Copias de seguridad mediante la opción de *backup*

Para realizar la configuración del *backup* en Proxmox VE, se debe acceder al apartado de “Respaldo” y seleccionar la opción “Respaldar ahora”. Se define el almacenamiento eligiendo destino donde se guardará la copia de seguridad: “disco”, “local”, “NFS”, etcétera. Esto determinará dónde se almacenará la copia de seguridad. Luego, se debe elegir el tipo de *backup* adecuado entre las opciones “parar” (detiene la MV antes de hacer el respaldo), “suspender” (pausa la MV temporalmente) y “snapshot” (realiza una copia sin detener la MV). Hay que considerar que cada uno ofrece distintos niveles de fiabilidad y rapidez.

Visto que el espacio de almacenamiento, se habilita la compresión para reducir el tamaño del archivo generado. Se puede optar por “LZO” (para velocidad),³³ “ZSTD” (para balance)³⁴ y “GZIP” (para una mayor reducción de espacio)³⁵. Se configuran las notificaciones por correo electrónico con una alerta sobre el estado del proceso de respaldo.

³³ Los archivos LZO pertenecen a “lzop de Markus F.X.J. Oberhumer. LZO es un formato de archivo que consiste en archivos comprimidos mediante el algoritmo Lempel-Ziv-Oberhumer (LZO)” [81].

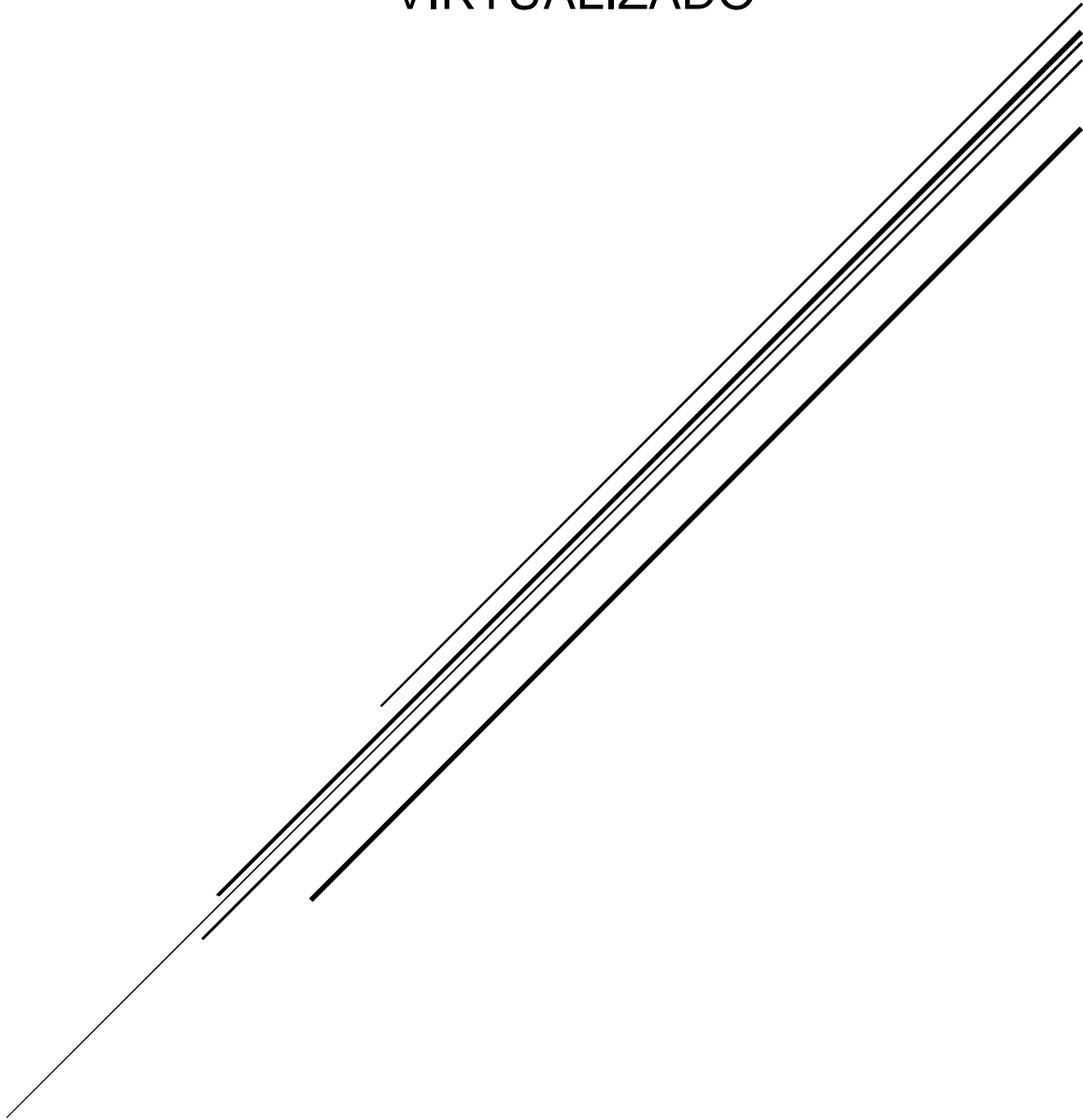
³⁴ Zstandard, también conocido como zstd, “es un algoritmo de compresión (tipo de compresión de datos que conserva toda la información original sin perder calidad.) lossless rápido que proporciona altas relaciones de compresión, creado por Yann Collet, un desarrollador de Facebook, en 2015” [80]. Es recomendable el uso debido a que “recibe un apoyo total por parte de distribuciones Linux, que cambió la compresión de sus paquetes XZ a Zstandard” [81].

³⁵ Los archivos GZIP pertenecen principalmente a “GNU Zip de Jean-loup Gailly and Mark Adler” [81]. Estos son archivos comprimidos creados mediante el algoritmo de compresión Gnu GZIP y se utiliza en sistemas operativos basados en Unix y Unix, como Linux y Mac OS.

CAPÍTULO 4

DISEÑO E IMPLEMENTACIÓN DEL ENTORNO

VIRTUALIZADO



Este capítulo presenta la implementación del entorno virtualizado para las máquinas virtuales y contenedores llevado a cabo en este trabajo. Los distintos servidores que serán ejecutados en las máquinas virtuales son servidor web, base de datos, servidor FTP, servidor SSH, servidor NFS, Samba para el cliente Windows y servidor VoIP. En este apartado se muestran distintas tablas como el uso recomendado de equipos para la implementación del hipervisor Proxmox VE, las capacidades recomendables de cada sistema operativo y los resultados que se tiene al trabajar con las máquinas virtuales con un servidor físico.

La Figura. 4.1 presenta una Topología de red local, debido a que, “de tamaño superior, ya que se extiende hasta algunos centenares de metros máximo 100 metros, a esto se le conoce como una red de área local” [40]. Conecta entre sí ordenadores, servidores, generalmente se utiliza para compartir recursos comunes, como periféricos, datos o aplicaciones, con una estructura física de estrella.

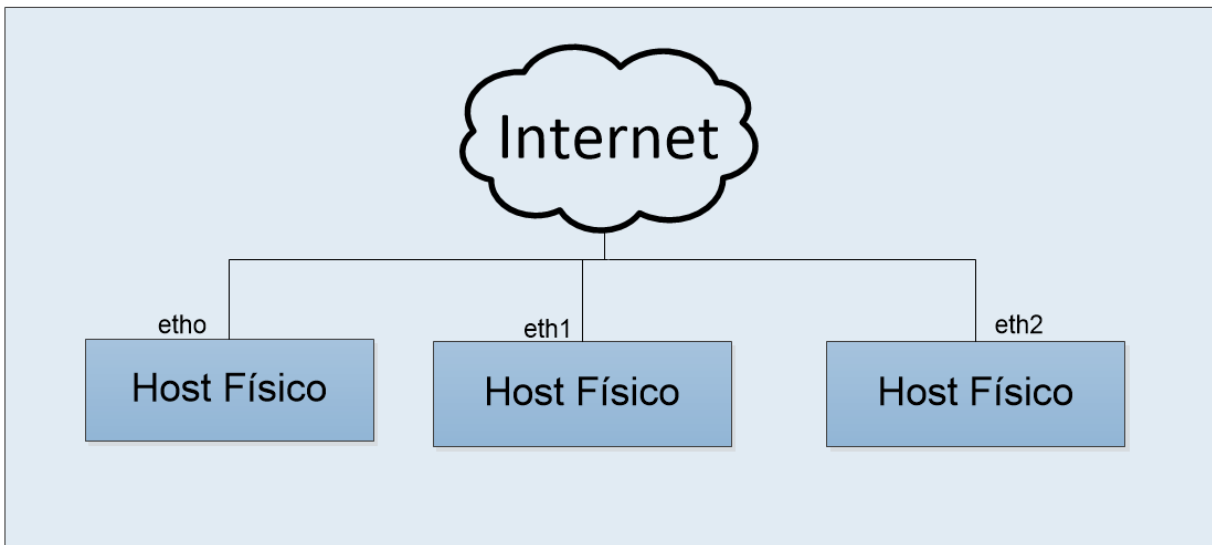


Figura 4.1: Topología de la Red física. [Fuente: elaboración Propia]

La Figura. 4.2 presenta una topología virtual donde presenta a los servidores como SSH, FTP, NFS, SAMBA, etcétera, que se implementaron en cada máquina virtual. Los primeros 3 sistemas virtualizados presentan un sistema operativo Ubuntu versión 20.2, mientras que el último presenta un sistema operativo Windows 10.

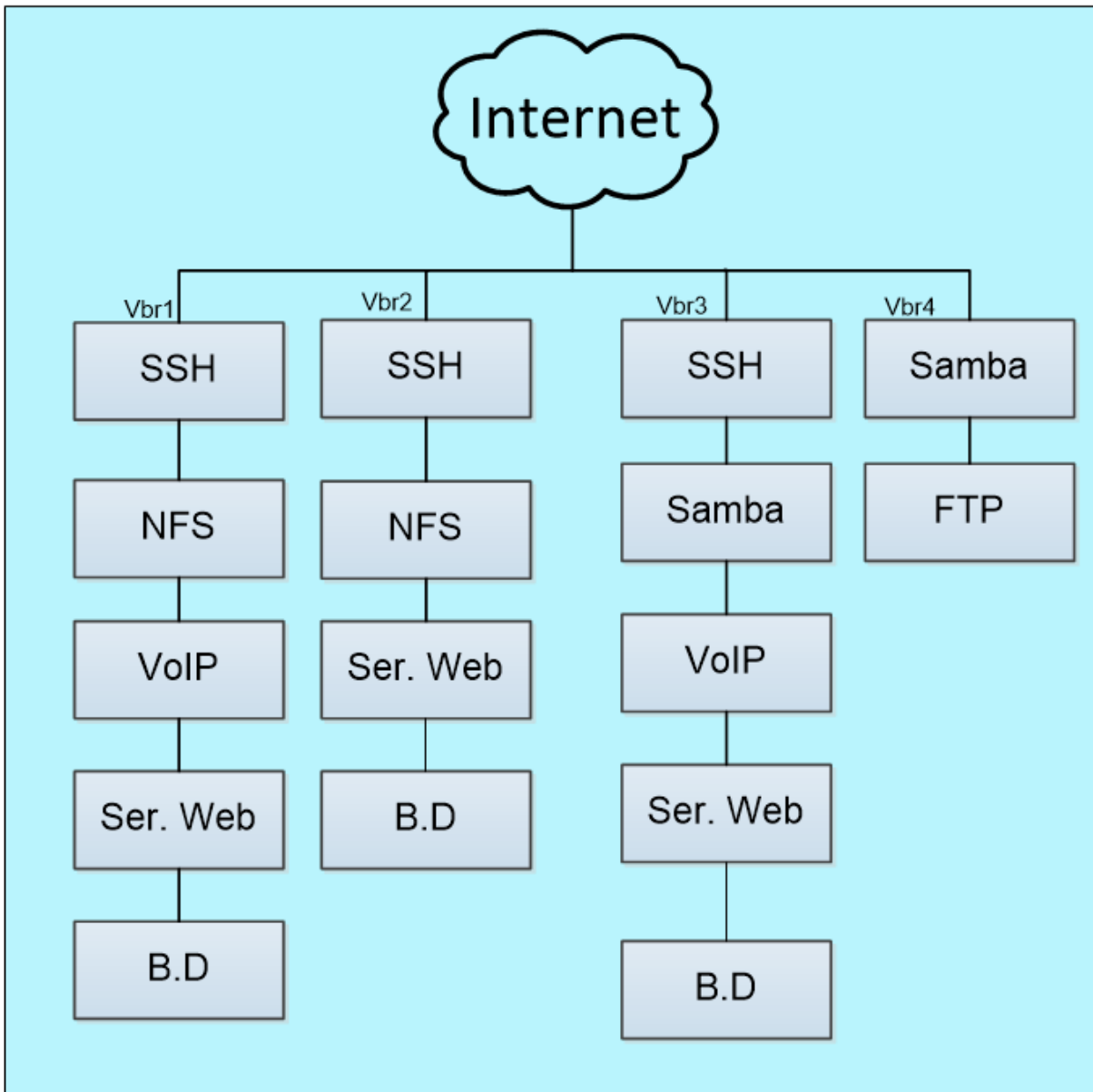


Figura 4.2: Topología de la Red Virtual. [Fuente: elaboración Propia]

4.1 Análisis y selección del *hardware* disponible

En el Apéndice A se muestra la documentación del proceso de instalación del *software* de hipervisor tipo 1 Proxmox VE, con los pasos a seguir para presenta la administración y operación del *software*. Además, los Apéndices B, C y D. incluye cómo se encuentra el laboratorio y cómo está administrada la red. Se muestran las IP de cada máquina virtual. Este proceso es importante para llevar un control al momento de administrar y verificar alguna falla que se pueda presentar. También se presentan los distintos tipos de almacenamientos que se pueden ocupar en el hipervisor. En lo concerniente a la administración, se incluyen clústeres, alta disponibilidad (HA), copias de seguridad, recuperación de datos y el almacenamiento disponible.

Antes de iniciar la instalación, hubo que tomar en cuenta que el *hardware* que se iba a ocupar cumpliera con los requisitos mínimos. Existe literatura previa que establece algunas especificaciones para pruebas básicas, como ejecutar una máquina virtual o un contenedor mínimo. En la Tabla 4.1 se pueden observar algunas recomendaciones para la implementación en Proxmox VE.

Componente	Uso básico (pruebas, laboratorio)	Uso más avanzado más carga en servidores.	Producción Configuración avanzada (máximo rendimiento)	Observaciones
CPU	Procesador: Intel o AMD de 64 bits	Un procesador Intel o AMD de 64 bits	Intel Xeon E5-2620 v3 (6 núcleos, 2.40 GHz) [39]	Se recomienda soporte de virtualización (VT-x / AMD-V).

Componente	Uso básico (pruebas, laboratorio)	Uso más avanzado más carga en servidores.	Producción Configuración avanzada (máximo rendimiento)	Observaciones
Placa base	Compatible con Intel VT/AMD-V	Compatible con Intel VT/AMD-V	Una placa base con CPU dual o cuádruple compatible con Intel VT o AMD-V	La compatibilidad con virtualización completa KVM es clave.
RAM	Mínimo 1 GB	16 GB mínimo, más es mejor	Crucial 32 GiB DDR4-2133 LRDIMM	La RAM depende de la cantidad de VMs y contenedores en ejecución.
Almacenamiento	1 disco HDD	RAID de <i>hardware</i> con caché protegida (BBU)	1 TB VelociRaptor WD1000DHTZ	RAID y almacenamiento SSD mejoran la velocidad y confiabilidad.
Interfaz de red	Una sola NIC (tarjeta de red)	Dos NICs Gigabit (para enlace)	Tarjetas adicionales según el clúster	Interfaces redundantes garantizan estabilidad en producción.
Clúster	No necesario	Recomendado para alta disponibilidad	Imprescindible en infraestructura avanzada. [35]	Permite administración centralizada y migración sin tiempo de inactividad.
Backup	Manual	Programado, almacenamiento externo	Programado con compresión optimizada. [41]	Los respaldos aseguran recuperación en caso de fallo.
Virtualización	Opcional	PCI passthrough activado	Virtualización avanzada con GPU asignada	Es útil para máquinas virtuales con alto procesamiento gráfico.

Tabla 4.1: Comparación de configuración recomendada de Proxmox VE. [W. Ahmed, Mastering Proxmox, Birmingham, Reino Unido: Packt Publishing, 2014.]

4.2 Dimensionamiento de recursos para máquinas virtuales

La cantidad de máquinas virtuales que se pueden instalar en un equipo depende del espacio de almacenamiento que esté disponible y de los recursos con los que cuenta el sistema. Entre los recursos más importantes se encuentra la CPU, el almacenamiento y la memoria RAM, que influye en la carga de trabajo del equipo. Vale la pena decir que la capacidad de almacenamiento varía mucho debido a que las máquinas virtuales requieren de cierto espacio. El disco puede ser HDD o SSD, y la memoria RAM puede ser dividida en particiones para cada máquina virtual, puesto que a cada equipo virtualizado se le asignará la parte del espacio de almacenamiento disponible en el equipo físico que sea necesaria.

De acuerdo con la literatura técnica sobre Proxmox VE, para ejecutar una máquina virtual o un contenedor básico se requiere como mínimo “un procesador de 64 bits con soporte para virtualización (Intel VT-x o AMD-V), al menos 1 GB de memoria RAM y un disco duro estándar de 10 a 25 GB de almacenamiento, dependiendo del sistema operativo invitado”. [41] Estos parámetros permiten iniciar pruebas de laboratorio en un entorno controlado, aunque se reconoce que constituyen el umbral más bajo y limitan considerablemente el número y tipo de servicios que podrían alojarse. En este trabajo se utilizaron dos equipos: una computadora personal de escritorio con procesador Intel Core i7 de tercera generación y una computadora portátil con procesador Intel Core i5 de décima generación. Ambos equipos contaban con cuatro *núcleos* físicos y memoria RAM de 8 GiB. Cada uno de los equipos de cómputo empleados en este trabajo contaba con 4 *núcleos*. [Tabla 4.3]. Con esta configuración, se determinó que era posible implementar de forma estable hasta dos máquinas virtuales por equipo, siempre y cuando cada una se configurase con al menos dos *núcleos* virtuales, 4 GiB de RAM y 25 a 30 GB de almacenamiento asignado.

Bajo estas condiciones, se podrían ejecutar servicios básicos como servidor web, servidor FTP o servidor SSH, manteniendo un rendimiento aceptable para pruebas académicas. En la [Tabla 4.2] se detallan los datos mínimos para la instalación del sistema de acuerdo con la información de las páginas oficiales de cada sistema operativo. Dicha información se tomó como referencia para asignar los recursos a cada máquina virtual. Puede observarse que Linux requiere una demanda menor en comparación de Windows.

Sistema Operativo	Procesador	RAM	Almacenamiento
Ubuntu de oficina	2 GHz doble núcleo	4 GB (físico) 2 GB (virtual)	25 GB (8.6 GiB mínimo)
Ubuntu (instalación mínima)	1 GHz doble núcleo	1 GB / 2 GB	10 GB / 20 GiB
Debian	Doble núcleo 2 GHz	24-64 MB (128 MiB servidor)	450 MB - 1 GB (4 GB servidor)
Fedora	Doble núcleo 2 GHz	2 GB mínimo 4 GiB recomendado	20 GB mínimo
Windows 10	1 GHz o más	1 GB (32-bit) / 2 GB (64-bit)	16 GB (32-bit) / 20 GB (64-bit)

Tabla 4.2: Dimensionamiento de recursos para cada sistema operativo. [Fuente: [1] Canonical Ltd., " Sistema Requerimientos- Ubuntu Server documentación," Ubuntu, 29 de abril de 2025. [2] Canonical Ltd., "Instalación/Sistema Requerimientos - Community Help Wiki," Ubuntu, 2025. [3] Debian Project, "Requisitos del sistema para Debian," Debian, 2025. [3] Microsoft Corporación, "Requisitos del sistema de Windows 10," Microsoft, 2025. [5] "Requisitos para instalar Linux," 2025.]

En la Tabla 4.3 se presentan los datos de los equipos que fueron utilizados como servidores para llevar a cabo las pruebas de este trabajo. Nótese que la memoria RAM de ambos equipos es de 8 GB, y que cada uno cuenta con 4 *núcleos*, por lo que se trata de equipos con características muy similares. La primera computadora sobre la cual se intervino tiene instalado como sistema operativo Ubuntu 24.04, y es importante notar, de entrada, de que se trata de un sistema operativo próximo a la obsolescencia. Esta computadora cuenta con 500 MB de almacenamiento, 8 GB de memoria RAM y un procesador con 4 *núcleos*. Por su parte, la segunda computadora tiene instalado el *software* de Proxmox VE versión 8.3 como sistema anfitrión y cuenta con máquinas virtuales con sistema operativo Ubuntu 20.04, y tiene instalados los mismos servidores que las que se implementaran en el servidor físico.

Característica	Nodo 1	Nodo 2	Observaciones
Sistema Operativo	Ubuntu 24.04 Proxmox VE	Proxmox VE	Ambos equipos cuentan con el Proxmox VE como sistema anfitrión.
Fabricante	HP	Acer	Ambos son de fabricantes en tecnología de la información con marcas reconocidas en el mercado.
Modelo	Compaq elite 8300 sff 3 ^o generación	aspire 10 ^o generación	El HP está diseñado para escritorio, mientras que la laptop Acer es portátil.
Memoria RAM	8 GB	8 Gib (7.78 GB utilizable)	La RAM es similar en ambos equipos.
Almacenamiento	HDD externo 1 TB, interno 500 GB	HDD interno 1 TB	La laptop tiene almacenamiento interno, mientras que el equipo de escritorio cuenta con un almacenamiento externo para instalar Proxmox VE.
Procesador	Intel Core i7 tercera generación.	Intel Core i5-1035g1 decima generación.	Ambos son de gama media.

Característica	Nodo 1	Nodo 2	Observaciones
Núcleos del CPU	4 <i>núcleos</i> , 8 hilos	4 <i>núcleos</i> , 8 hilos	Ambos tienen la misma cantidad de <i>núcleos</i> e hilos.
Uso como nodo	nodo principal	nodo secundario	El equipo de escritorio marca HP es el nodo principal y el equipo portátil Acer es el nodo secundario.
Interfaz de red	1 nic gigabit	1 nic gigabit	Los dos se conectan con cable UTP configurados mediante el protocolo IPV4.
Servidor en MV	1. Ubuntu 20.04. 2. Ubuntu 20.04.	1. Windows 10 2. Ubuntu 20.04	Se realiza la carga de diferentes servidores como servidor web, base de datos, servidor FTP, servidor SSH, servidor NFS, servidor samba para el cliente Windows y Servidor VoIP.

Tabla 4.3: Comparación de los equipos del Laboratorio de Telecomunicaciones. [Fuente: Elaboración Propia]

En contraste con lo anterior, en una configuración avanzada pensada para producción —y no para la realización de pruebas, que es el propósito del Laboratorio de Telecomunicaciones— se recomendarían procesadores de gama servidor, “como *Hardware* Procesador/unidad central de procesamiento (CPU) Intel Xeon E5-2630 v3 2.4 GHz, Placa base Intel S2600CP2 Dual LGA2011, Memoria Kingston 16 GB DDR3 ECC registrada, Fuente de alimentación Redundante”. [39] Esta infraestructura permitiría soportar múltiples máquinas virtuales en paralelo —entre 6 y 10 instancias por servidor físico— y servicios de mayor exigencia, como bases de datos de gran tamaño o entornos de VoIP. De este modo, se establecen ciertos límites cuantitativos: con *hardware* básico se pueden realizar únicamente prácticas y prototipos, mientras que con *hardware* de nivel servidor se alcanzaría la máxima capacidad de despliegue prevista para entornos profesionales.

4.3 Creación de máquinas virtuales

En el caso del Laboratorio de Telecomunicaciones, se trabajó con un equipo de escritorio con procesador Core i7 de tercera generación y uno portátil con procesador Core i5 de décima generación. Con la implementación del *software* Proxmox VE estas 2 computadoras se le dio la utilizada de instalar tres máquinas virtuales con sistema operativo Ubuntu versión 20.2 y una máquina virtual con sistema operativo Windows 10 con el objetivo de levantar servidores como FTP, Servidor Web con base de datos, Servidor Asterisk para enlazar llamadas de forma local, Servidor SSH para comunicarse a la terminal de las máquinas virtuales de forma remota sin necesidad de ingresar directamente a la máquina virtual, también se implementó la instalación del Servidor Samba para la comunicación con otros sistemas operativos como Windows, servidor FTP, estos servidores son importantes para la importación de almacenamiento además de que dentro del sistema de Proxmox VE se puede realizar distintos tipos de almacenamientos compartidos. En la Tabla 4.4, se observan los datos del estado inicial de las 2 máquinas virtuales del primer nodo (HP) y los datos del estado las 2 máquinas virtuales del segundo nodo (Acer).

Parámetro	Máquina virtual 101	Máquina virtual 102	Máquina virtual 103	Máquina virtual 104
Sistema operativo	Ubuntu 20.04	Ubuntu 20.04	Windows 10	Ubuntu 20.04
Uso de CPU	2 Core	2 Core	2 Core	2 Core
Uso de RAM	4.00 GB	4.00 GB	4.00 GB	4.00 GB
Disco Asignado	32.00 GB	32.00 GB	32.00 GB	32.00 GB
Modelo de CPU	Intel Core i7-3770	Intel Core i7-3770	Intel Core i5-1035g1	Intel Core i5-1035g1
Frecuencia CPU	3.4 GHz - 3.9 GHz	3.4 GHz - 3.9 GHz	3.4 GHz - 3.9 GHz	3.4 GHz - 3.9 GHz

Tabla 4.4: Comparación de recursos en las Máquinas Virtuales. [Fuente: Elaboración Propia]

4.4 Instalación de servicios

4.4.1 Utilidad y aplicaciones de los servidores virtualizados

El trabajo realizado sobre servidores web no consistió únicamente en “levantar un servidor”, sino en procurar espacios de alojamiento para proyectos académicos. Estos espacios estuvieron pensados para fungir como lugares de práctica para que en ellos los estudiantes pudieran configurar y administrar servicios web sin que dependieran de una infraestructura institucional centralizada. En tal sentido, se buscó que el Laboratorio de Telecomunicaciones se convirtiera en un espacio para proyectos académicos de desarrollo web o de aplicaciones útiles por sus capacidades, trayendo así consigo beneficios para la comunidad estudiantil. En general, estos servidores tenían como fin el ofrecer beneficios como:

- Extender la vida útil de *hardware* a punto de la obsolescencia: al correr varios servicios en paralelo dentro de las máquinas virtuales, les brindan una segunda oportunidad a los equipos.
- Ofrecer un espacio para la realización de pruebas: un lugar donde se pueda instalar, fallar, reinstalar y experimentar con libertad sin comprometer equipos que pudieran estar destinados a la producción, lo que hace que las pruebas sean mucho más realistas.
- Fomentar el desarrollo de proyectos aplicados: en este tipo de equipos, se puede trabajar proyectos que involucren al área de telecomunicaciones, pero también a seguridad, Internet de las cosas (IoT), desarrollo web entre otros.

4.4.2 Servidores instalados

Algunos servidores que se instalaron para este trabajo fueron:

- Servidor web y base de datos: para este trabajo se implementó la instalación de MySQL debido a que se trata de un sistema de administración de base de datos de código abierto utilizado por organizaciones importantes. Como ejemplo, en la dirección IP 192.168.0.224 se realiza un *login* de accesos roles y permiso. Este sistema funciona para controlar quien accede mediante usuario y contraseña [Figura 4.3].
- De acuerdo a la Figura 4.4, el usuario accede desde el navegador; el navegador solicita la página al servidor Apache³⁶ para el puerto 80. “En este caso, Apache ejecuta un PHP que se genera desde el servidor web HTML,³⁷ el PHP³⁸ consulta MySQL³⁹ para obtener datos (usuarios, permisos); la página web con HTML, CSS⁴⁰ y JavaScript⁴¹ se envía al navegador; y el navegador muestra la página web. Este esquema permite comprender como interactúa el servidor.

³⁶ APACHE es un *Software* de código abierto para alojar más de un sitio web y no es necesario instalar un nuevo servidor para cada sitio web que se desee ejecutar [68].

³⁷ HTML es un lenguaje de marcado de hipertexto para estructura el contenido de la página, dentro de cada archivo generado.

³⁸ PHP es un lenguaje de programación que interactúa con el servidor apache, además de ser diseñado para crear sitios web dinámicos.

³⁹ “SQL (pronunciado como las letras S-Q-L o como sequel) lenguaje diseñado específicamente para comunicarse con bases de datos” [76]

⁴⁰ CSS lenguaje que se incorpora una hoja de estilos conocido como para darle una mejor presentación al sitio web

⁴¹ JavaScript, “un lenguaje de programación para incorporar acciones que se activan al pulsar botones y ventanas con mensajes de aviso al Usuario” [91].

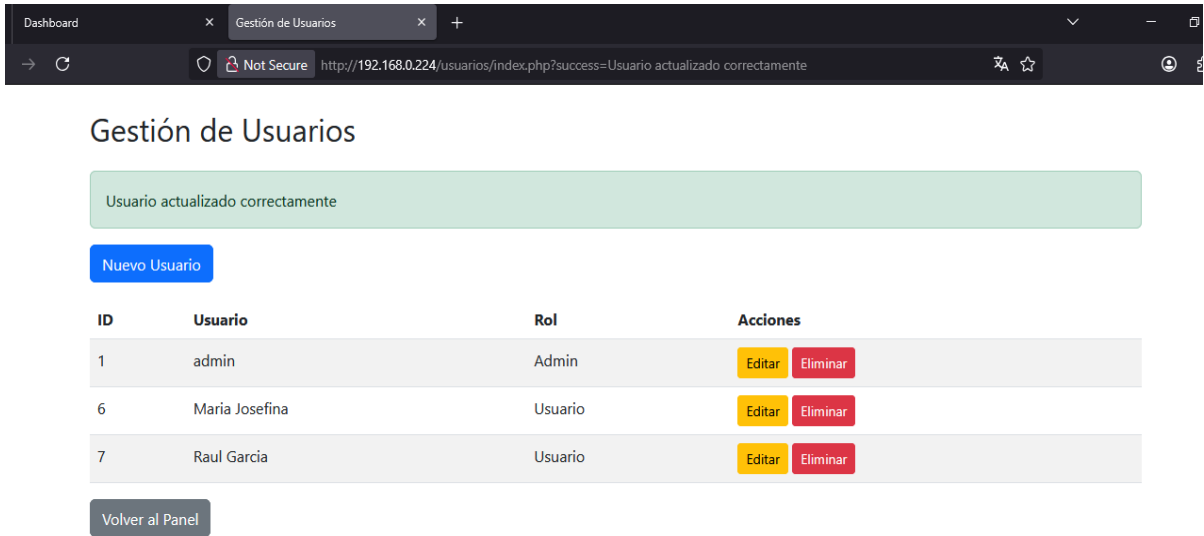


Figura 4.3: Servidor Web. [Captura Propia]

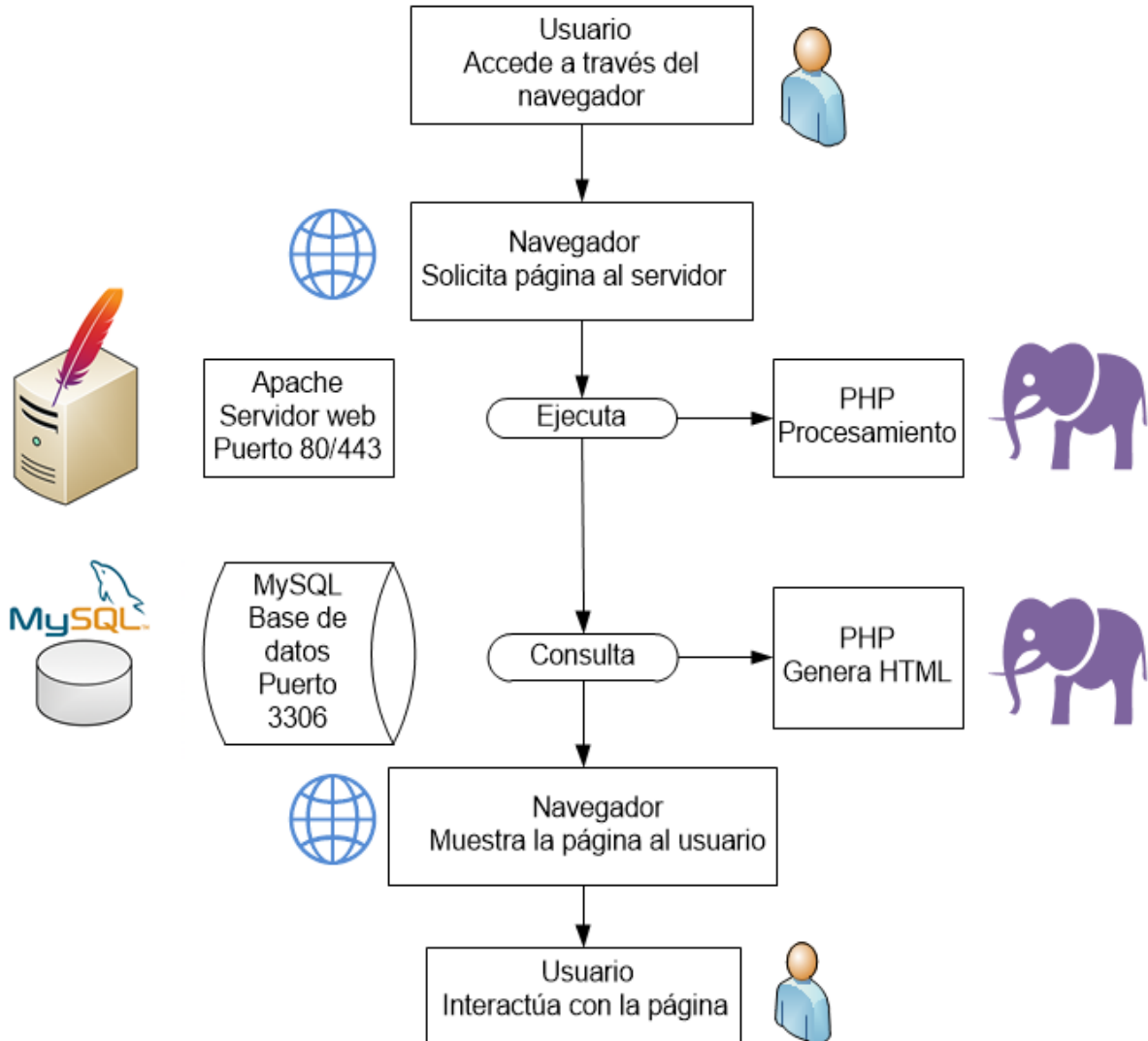


Figura 4.4: Diagrama de Bloques Servidor Web. [Elaboración Propia]

Este tipo de servidor coincide con el que habitualmente se solicita a los alumnos que desplieguen en asignaturas como informática en las telecomunicaciones, de modo que su creación puede servir a los alumnos que cursen por estas materias para tomarlo como un referente en la creación de sus propios proyectos.

- Servidor FTP: se implementó un servidor de Protocolo de Transferencia de Archivos (FTP) desde la consola de Ubuntu con la dirección IP 192.168.0.179 para que cualquier otro equipo se pudiera conectar y permitiera realizar distintas tareas como la transferencia archivos entre computadoras a través de una red entre cliente y servidor remoto. La Figura 4.5 muestra cómo se ingresa al servidor FTP con el usuario y contraseña del equipo de Ubuntu con dirección IP 192.168.0.179, mediante un equipo de Windows utilizando el puerto 21, donde se pueden crear, eliminar y editar archivos.



Figura 4.5: Servidor FTP. [Elaboración Propia]

Este servidor se desplegó para proveer almacenamiento compartido en red dentro del laboratorio. La idea es que ello permita facilitar la realización de prácticas de gestión de archivos, usuarios y permisos en entornos reales. También puede servir para que los estudiantes experimenten con protocolos de transferencia en condiciones controladas.

- Servidor SSH: en cada máquina virtual se implementó la instalación del servidor SSH utilizando el puerto 22. Esto permite el acceso remoto seguro desde otro equipo utilizando la terminal o alguna aplicación y, así, poder administrar la máquina virtual. En este ejemplo, se utilizó una computadora principal para conectarse a la máquina virtual por medio de SSH desde la consola para realizar distintas tareas [Figura 4.6].

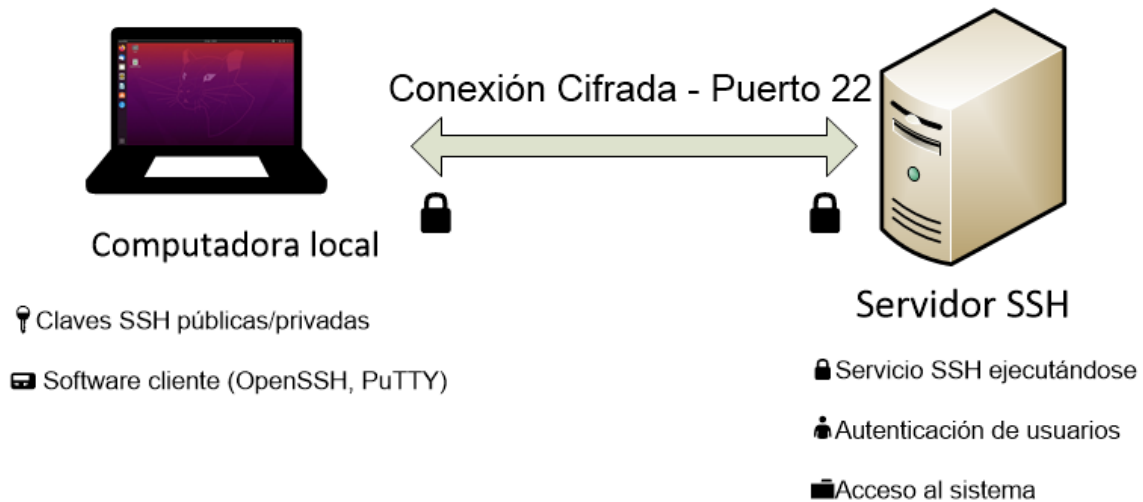


Figura 4.6: Servidor SSH. [Elaboración Propia]

La razón de ser de este tipo de servidor en el trabajo es que, a través de él, los estudiantes aprendan a conectarse de forma remota a servidores y puedan administrarlos vía terminal, una habilidad que es clave en la ingeniería de Telecomunicaciones para un trabajo optimizado. Con este tipo de servidor instalado es posible simular entornos de acceso remoto como pudieran ser empresas o centros de datos, contribuyendo así a mejorar las oportunidades formativas de los alumnos.

- Cliente y servidor NFS: se instaló el servidor NFS para compartir archivos y directorios entre computadoras. El servidor NFS comparte carpetas a las cuales el cliente accede como si fuera su propio disco. Como ejemplo, se realizó la carga del servidor NFS en la máquina virtual Ubuntu con la IP 192.168.0.179 para conectarse como cliente desde

otra máquina virtual con la IP 192.168.0.224. En la siguiente figura se puede ver el acceso como cliente al servidor NFS [Figura 4.7].

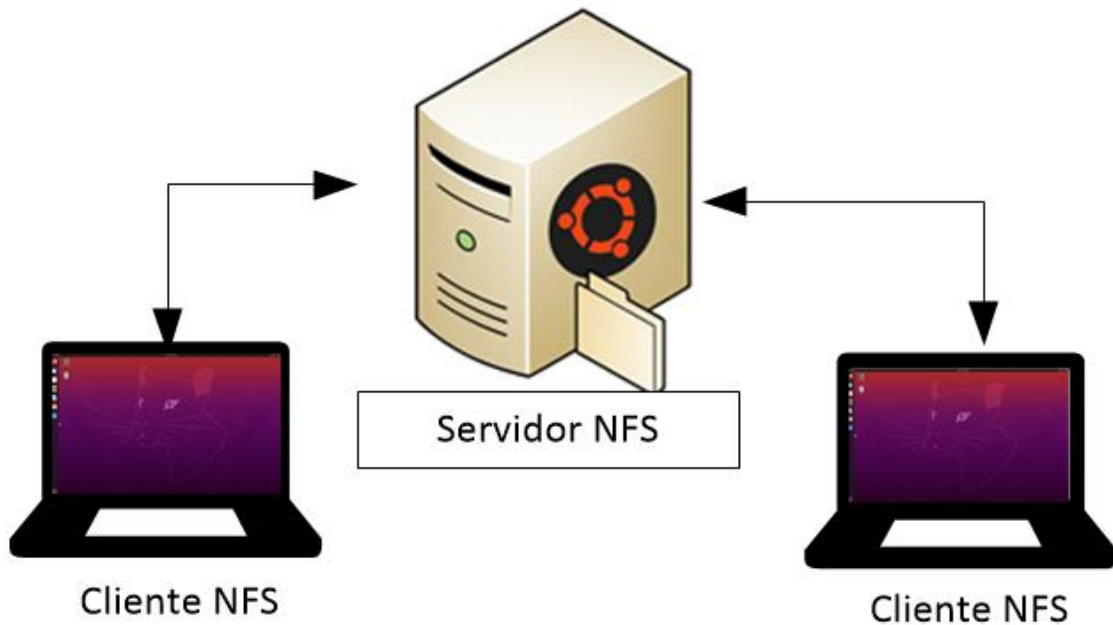


Figura 4.7: Servidor cliente NFS. [Elaboración Propia]

Este tipo de servidor proporciona un tipo de almacenamiento diferente que puede ser compartido entre máquinas virtuales en Ubuntu, aunque su instalación es un tanto más compleja que la del resto debido a que se debe realizar primero la configuración del cliente y después del servidor.

Servidor Samba y cliente Windows: se instaló el servidor Samba para convertir una máquina virtual Linux en un servidor que permitiera compartir archivos e impresoras entre los sistemas de Linux y Windows. Se realizó la carga del servidor Samba mediante la dirección IP 192.168.0.64 desde la terminal de Ubuntu [Figura 4.8]. Este tipo de servidor se instaló para compartir información como documentos, imágenes, etcétera entre Windows y Linux.

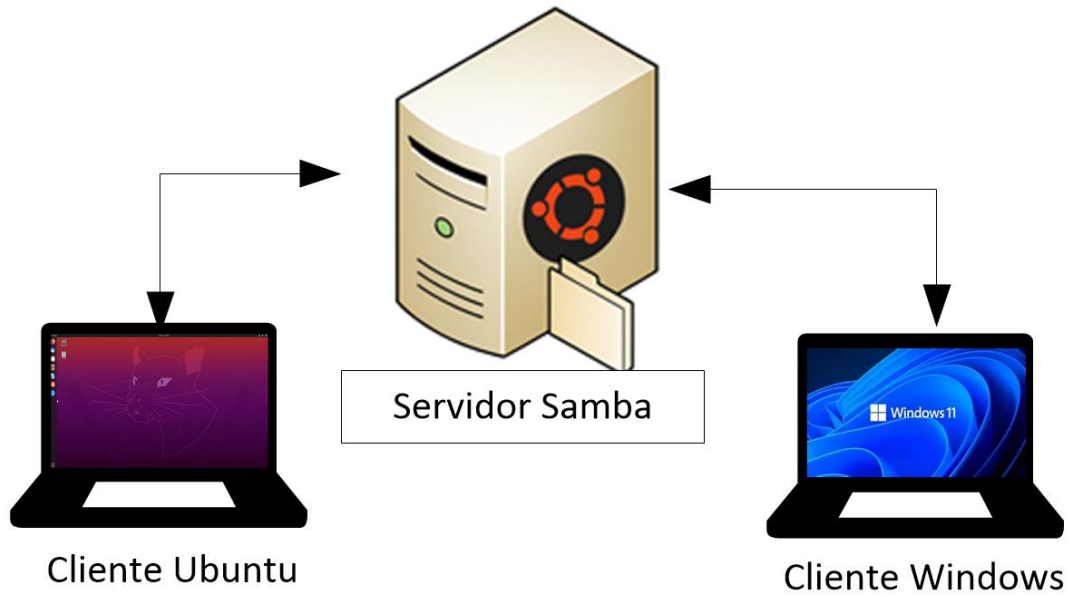


Figura 4.8: Servidor Samba instalado en Ubuntu. [Elaboración Propia]

- Servidor VoIP (voz sobre protocolo de internet): se diseñó e implementó una central telefónica PBX con un *software* de licencia libre —servidor Asterisk— que permitió realizar las funciones de una central telefónica y establecer configuraciones básicas para llamadas. Asterisk es un *software* de código abierto que puede ejecutarse en Linux o en otro tipo de sistemas operativos. Se creó originalmente como el motor de un sistema PBX con los componentes necesarios para construir un sistema telefónico empresarial. Asterisk es independiente, por lo que se puede conectar al mundo exterior mediante VoIP o tecnologías de telefonía tradicional. Se utiliza Zoiper como aplicación para realizar las llamadas debido a que se puede ejecutar en varias plataformas diferentes sin importar si se está utilizando macOS, Linux, Windows, iOS, Android o un navegador. La Figura 4.9 muestra un ejemplo de comunicación entre dos llamadas: el cliente 101 marca; Asterisk recibe la solicitud SIP en el puerto 5060, consulta el archivo de extensiones y conecta la llamada con el destino 102 [Figura 4.10].

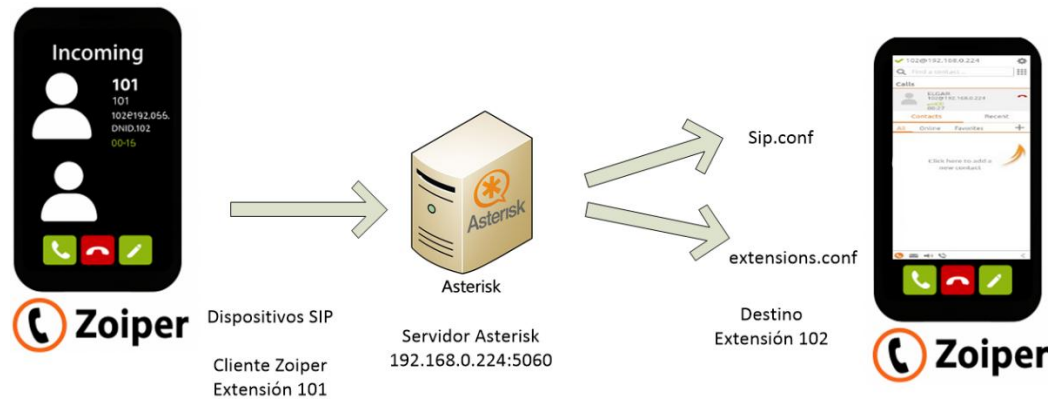


Figura 4.9: Diagrama Servidor VoIP. [Elaboración Propia]

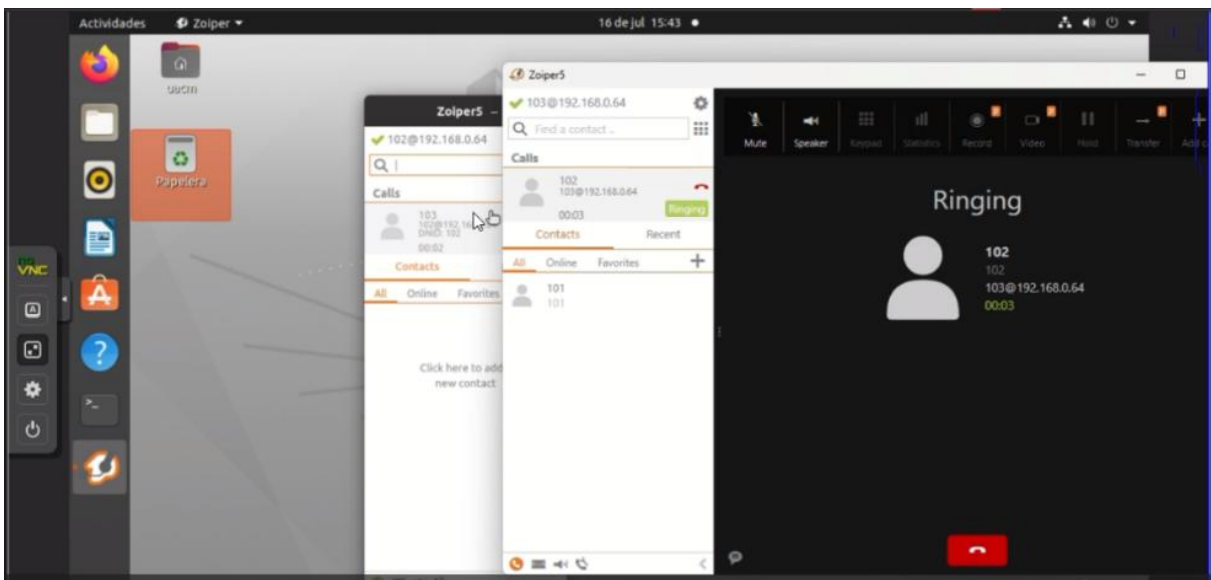


Figura 4.10: Ejemplo de llamadas usando un servidor VoIP. [Elaboración Propia]

Cabe aclarar que el despliegue de este servidor en el contexto del presente trabajo no sólo obedeció a la necesidad de tener telefonía IP, sino a conformar un espacio para la práctica con configuraciones por medio de la tecnología de voz sobre IP (VOIP) relevante en el mundo de las telecomunicaciones. Asimismo, con este esfuerzo primigenio se puede encaminar la creación de un laboratorio para la realización de proyectos de integración de servicios de voz y datos. Como puede advertirse a partir de las explicaciones precedentes, son varios los servicios para el Laboratorio de Telecomunicaciones de la instalación de los servidores virtualizados.

4.5 Criterios para evaluación del rendimiento y métricas de desempeño

De acuerdo con varios fascículos (Publicaciones periódicas) y web tecnológicos (IBERTEL), los principales criterios para determinar el rendimiento adecuado de máquinas virtuales son “el tiempo de respuesta, la latencia, el uso del CPU, Memoria RAM y el Almacenamiento” [42]. Realizar esta evaluación permite identificar posibles cuellos de botella, así como buscar una forma de mejorar el sistema para un mejor entorno, ya sea un laboratorio, una institución o entornos de producción y pruebas.

Las métricas esenciales que deben monitorearse al momento de evaluar el rendimiento son:

- Unidad Central de Procesamiento (CPU): “En el corazón de cada computadora hay un chip conocido como procesador, que determina hasta dónde llega la capacidad de trabajo del equipo. También conocido como CPU (unidad central de proceso) o microprocesador. La función del procesador es ejecutar instrucciones, realizar cálculos y coordinar el funcionamiento del resto de los dispositivos” [43]. Una forma de explicar es pensar en un sistema el sistema son todos los componentes, para que funcione se necesita de un líder, quien hace que todos presenten el mismo objetivo y lo realiza por medio de instrucciones. “Los principales fabricantes de procesadores en la actualidad son Intel, Motorola, VIA y AMD” [43]. Al realizar la evaluación de se debe ver la capacidad de procesamiento de la máquina virtual. Esto obliga a analizar la forma de utilizar la CPU, debido a que es clave para el rendimiento. Si la máquina virtual tiene poca demanda de CPU virtual asignada o si se encuentra sobrecargada, esto puede conducir a una disminución al rendimiento.

- Memoria RAM: la memoria RAM (memoria de acceso aleatorio) “es uno de los tres soportes de una plataforma informática. Cumple una función primordial, ya que se ocupa de almacenar la información de las instrucciones que el procesador debe gestionar como los resultados de dichas tareas y cualquier otro dato que fuera necesario” [3]. Donde realiza la función de acceso rápido y almacenamiento temporal de datos en las máquinas virtuales al momento de ejecutar diversas tareas. En las máquinas virtuales, “La memoria RAM base instalada en muchas computadoras es de 4 GB o 8 GB, si bien pueden ser suficientes para el sistema operativo y algunas aplicaciones” [44]. Sin embargo, si el Administrador de tareas muestra un uso elevado de la memoria RAM” puede basarse a que se encuentre abiertos demasiados programas ejecutándose en segundo plano. La regla general es que, entre más memoria, mejor. Cuando menos, duplicar la memoria dará un mejor entorno para trabajar. “Esto es cierto sobre todo en las aplicaciones actuales que requieren mucha memoria, como programas de oficina, paquetes de edición multimedia y juegos con uso intensivo de gráficos” [45]. Como ejemplo, “utilizando el sistema Windows 10, donde en los gráficos se presenta “el uso de memoria RAM es superior al 70% y sólo está ejecutando tareas sencillas o no está haciendo nada en absoluto, es posible que necesite más RAM para evitar un rendimiento bajo” [46]. Así, en promedio, un uso de RAM del 30% al 50% cuando está inactivo es normal en el Administrador de Tareas. También hay que considerar las pestañas abiertas: Si la máquina virtual presenta un uso excesivo de RAM cuando está inactiva, es posible que haya algún problema como un virus o aplicaciones en segundo plano que consumen recursos. Sin embargo, cuando la Memoria RAM alcanza el punto máximo, se implementa la memoria Swap, también conocida como espacio de intercambio, en la que el disco duro se utiliza como una extensión de la memoria RAM. Esto ayuda a que los datos menos utilizados se trasladen a dicha memoria y es cuando se la memoria RAM alcanza su límite.

- Unidad de Almacenamiento: en la actualidad hay Dos tipos de almacenamiento el primero es el disco duro Mecánico, “se compone de uno o varios platos metálicos giratorios, donde un cabezal magnético lee o escribe información a medida que utilizamos la computadora” [3]. El segundo es el disco de estado sólido (SSD), estas usan chips de memoria flash (NAND) para almacenar datos y su objetivo de los dos es hacer que la computadora tenga una unidad de almacenamiento donde se instale el sistema operativo y almacene toda la información necesaria.

Para realizar la evaluación de las máquinas virtuales hay técnicas. La primera es el monitoreo del rendimiento. Otras pueden ser las pruebas de cargas, que son técnicas que se deben realizar en un entorno para poder dar una evaluación. “Las pruebas de carga permiten identificar posibles cuellos de botella y evaluar el rendimiento en condiciones de alta demanda” [42]. En el caso concreto de este trabajo, para utilizar Ubuntu 20.04 como máquina virtual en la instalación se necesita de una implementación de “Procesador de doble *núcleo* de 2 GHz, 4 GiB de RAM (memoria del sistema) 25 GB de espacio en el disco duro” [47]. En lo que respecta al *hardware* y *software*, MySQL exige como requisitos mínimos: “Procesador (CPU): arquitectura x86-64, mín. 1 GHz (Dual-Core) 1 *núcleo* mínimo, Memoria (RAM): mín. 1 GB, Sistema operativo: Ubuntu 20.04, Firewall: puerto 3306 de MySQL abierto, Espacio en disco duro: mínimo de 500 MiB” [48].

En la Tabla 4.5 se muestran los rangos esperados e interpretaciones en los cuales se basará la evaluación de los rendimientos de las máquinas virtuales a desarrollar en el Laboratorio de Telecomunicaciones.

Métricas	Definición	Rango esperado	Interpretación
De la red			
Velocidad de transferencia cliente-servidor	“Se refieren a la velocidad a la que los datos se transmiten entre dispositivos o a través de una red. Medido en unidades como los bits por segundo (bps), bytes por segundo (Bps)” [49].	720-960 millones de bits	Esta métrica es de suma importancia cuando se quiere conocer la velocidad a la que se envían los datos a través de un ancho de banda al momento en que se comunican las máquinas virtuales, ya sea entre en mismo nodo o cuando salen del nodo. Esta velocidad depende del ancho de banda y de la cantidad de datos que se intenta transmitir por una red que excede la capacidad disponible. Esta métrica ayuda a identificar problemas de red, aplicaciones críticas para evitar congestión y cuellos de botella.
Velocidad de transferencia fuera del ambiente virtual	Es la velocidad con la que la máquina virtual puede enviar o recibir información a través de la red física, como si fuera una computadora real. Su unidad de medida es Mbps.	70-95 Mbps	El entorno de Proxmox VE trabaja con 1000 Mbps en su entorno de red.
Uso del ancho de banda (%)	Es el porcentaje de ocupación del ancho de banda (<i>ethtool</i>).	7.5%-9.5% para 1000 Mbps	Bajo uso a la capacidad de la red.
Velocidad de transferencia fuera del servidor físico	Es cuando la máquina física puede enviar o recibir información a través otro dispositivo físico. Su unidad es MB/s.	70-95 Mbps	El entorno de Proxmox VE trabaja con 1000 Mbps.
Captura de tráfico			
Captura de tráfico	Es el proceso para ver y analizar el registro de la red que se desea analizar. No hay una unidad específica para su medición; se utilizan las unidades estándares.	Depende el tipo de Servidor y protocolo si es TCP o UDP	En este entorno se pueden filtrar y monitorear el tráfico de casa servidor instalado.
Pérdida de paquetes	Es el “Porcentaje de paquetes que nunca llegan al destino” [50].	Excelente < 1% Aceptable < 5% Pobre > 5%	Es uno de los principales factores que afecta la calidad de la llamada.
Variación del retardo (<i>jitter</i>)	Es la diferencia en el tiempo de llegada entre paquetes consecutivos.	Excelente < 30 ms Aceptable < 60 ms Pobre > 60 ms	Cuando la variación es pobre, el audio llega desordenado o con intervalos irregulares, generando una voz robotizada o entrecortada.

Métricas	Definición	Rango esperado	Interpretación
Retardo	“Es el tiempo que tarda un paquete de voz en viajar desde el emisor hasta el receptor” [50].	Excelente < 50 ms Aceptable < 80 ms Pobre > 150 ms	Si se está por debajo de los 150 ms, la conversación se vuelve “cortada” o con pausas incómodas, similar a una llamada satelital.
De la PC			
Uso de CPU (%)	<p>Es el porcentaje utilizado por parte de la computadora física al realizar distintas actividades. Indica la eficiencia en el uso de los <i>núcleos</i> disponibles [51].</p> <p>Se presenta los siguientes conceptos que fueron adquiridos para realizar las métricas, estas son (us) es el porcentaje de tiempo que el CPU dedica a ejecutar procesos, (sy) es el porcentaje de tiempo que el CPU dedica a procesos del <i>núcleo</i> (kernel), (wa) (es el porcentaje de CPU esperando operaciones de disco/red y (ld) es el porcentaje de tiempo que el CPU está inactivo, sin ejecutar tareas.</p>	adecuado: < 80% Aceptable: > 80%, < 90% No adecuado: ≥ 90% [51]	Valores altos sugieren la necesidad de contar con más CPU; valores bajos, exceso de porcentaje asignado. Ante valores con promedio superior al 80% o incluso al 90% durante las operaciones normales se recomienda aumentar la cantidad de vCPU asignadas [52]. “top es una especie de versión en tiempo real de “ps” que ofrece un resumen interactivo y actualizado periódicamente de los procesos y su uso de recursos” [53]. También nos gusta htop, “un visualizador de procesos interactivo, multiplataforma y de código abierto que ofrece más funciones y una interfaz más atractiva que la de top”. [53].
Uso de vCPU (%)	Porcentaje de utilización de vCPU asignada.	adecuado: < 80% Aceptable: > 80%, < 90% No adecuado: ≥ 90%	Indica la eficiencia en el uso de los <i>núcleos</i> disponibles. Valores altos sugieren necesidad de contar con más CPU; valores bajos, exceso de porcentaje asignado [51].
Uso de RAM (%)	Porción de memoria utilizada sobre el total asignado.	adecuado: < 80% Aceptable: 80–90% No adecuado: ≥ 90%	“Una pestaña de Chrome con poco contenido utiliza alrededor del 1% al 2% de la RAM. Entonces, si tienes 10 pestañas, utiliza entre el 10% y el 20% de la RAM disponible” [45]. “Para un PC con 4 GiB: un uso de RAM del 80% o más cuando está inactivo es normal. 8 GiB: un uso del 50% o menos. 16 GiB: un uso del 30% o menos. 32 GiB: un uso del 5% o menos” [45].

Métricas	Definición	Rango esperado	Interpretación
Almacenamiento (%)	Ocupación del disco. En el caso de las máquinas virtuales, puede ser un Virtual Hard Disk (VHD), archivo que simula un disco duro dentro de una máquina virtual.	Recomendable: $\leq 90\%$	Un uso igual o superior al 90% puede generar cuellos de botella y evita la instalación de más aplicaciones o el guardado de más información.
Uso de memoria compartida (SWAP)	Porcentaje del uso de memoria compartida. Cuando la Memoria RAM queda limitada, se utiliza memoria compartida, que es extraída de la unidad de almacenamiento.	adecuado: < 80% Aceptable: 80–90% No adecuado: $\geq 90\%$	“Se asigna de forma eficazmente tanta memoria como los procesos soliciten al aumentar la RAM real con espacio de intercambio” [54].

Tabla 4.5: Definiciones y Métricas. [Elaboración Propia]

Por otro lado, también es importante considerar algunos valores estándar que pueden ser útiles para medir el desempeño en la realización de tareas con las máquinas virtuales. Éstas son:

- Llamadas telefónicas: si en un segundo hay 8,000 muestras y cada muestra tiene 8 bits, el número total de bits en un segundo está dado por $8,000 \times 8 = 64,000$ bits (64 kbit/s). [55] Por lo tanto, la velocidad estándar de la línea telefónica digital es 64 kbit por segundo (kbit/s o kbps), que es la velocidad básica de un circuito digital. Por esta razón, desde siempre, los circuitos digitales se miden en múltiplos de 64 kbit/s. El circuito telefónico de 2 Mbit/s es un circuito de 30 canales de 64 kbit/s.
- Acceso a bases de datos: para velocidades menores a 2 Mbps se utilizan textos; entre 2 y 10 Mbps se manejan gráficos; y con velocidades superiores a 10 Mbps se accede a rayos X y ficheros CAD [56].
- Transferencia de ficheros: a menos de 2 Mbps se transfieren textos; entre 2 y 10 Mbps se manejan gráficos y comunicaciones host; y por encima de 10 Mbps se transfieren datos meteorológicos y de superordenadores. [56]

- Interconexión LAN: con menos de 2 Mbps se emplea Token Ring a 4 Mbps; entre 2 y 10 Mbps se utiliza Ethernet y Token Ring a 16 Mbps; y con más de 10 Mbps se implementa FDDI [56].
- Transferencia de imágenes: a velocidades inferiores a 2 Mbps se transmiten imágenes monocromo; entre 2 y 10 Mbps se transfieren imágenes a color; y por encima de 10 Mbps se manejan imágenes de alta resolución CAD/CAM [56].
- Videoconferencias PC: con menos de 2 Mbps se obtiene baja definición; entre 2 y 10 Mbps se logra alta calidad en pantalla pequeña; y con más de 10 Mbps se alcanza alta calidad en pantalla grande [56].
- Redes multimedia: a cualquier velocidad —ya sea menor a 2 Mbps, entre 2 y 10 Mbps, o superior a 10 Mbps— se soporta contenido multimedia [56].

También se tomaron en cuenta los criterios de (Instituto de Ingenieros Eléctricos y Electrónicos) IEEE 802.3, que concretan diferentes tipos de red —denominadas genéricamente redes Ethernet—, las cuales tienen en común la utilización del mismo protocolo de acceso al medio MAC (CSMA/CD), con velocidad de 10 a 10.000 Mbit/s [55] y 802.10 Redes LAN virtuales y seguridad [57]. Gigabit Ethernet es una extensión de las normas Ethernet de 10 y 100 Mbit/s que ofrece en modo semidúplex o dúplex un ancho de banda de 1 a 10 Gbit/s, asegurando la compatibilidad con la base instalada de Ethernet y Fast Ethernet. Es necesario precisar que, utilizando el script ethtool eno1, el equipo sobre el cual se trabajó en este trabajo incluye de fábrica el adaptador Intel 82579LM Gigabit Ethernet, el cual está limitado por diseño a 1 Gbps. Por consiguiente, aunque la red física sea de 10 Gbps, el sistema no podrá negociar más allá de lo que le permite el hardware.

4.6 Análisis de los servidores

4.6.1 Velocidad de transferencia cliente-servidor

Para conocer la velocidad de transferencia fuera del ambiente virtual del servidor con la "IP 192.168.0.64" se utilizó iPerf3, una herramienta de código abierto diseñada para realizar mediciones como el ancho de banda, la pérdida de paquetes, etcétera, conectados a través una red entre dispositivos. Primero se conectó el servidor desde la terminal de Ubuntu con el comando "iperf3 -s -p 8000". Luego, como cliente en Windows, se ejecutó el comando ".\iperf3.exe -c 192.168.0.64 -p 8000". Los resultados de la transferencia se pueden ver en la Figura 4.11, en intervalos de 10 segundos que indican el ancho de banda promedio de 95.5 Mbits/seg. y la cantidad de datos transferidos, de 114 MBytes. Estos valores obtenidos se pueden considerar adecuados debido a que la red donde está conectado se encuentra diseñada para operar a una velocidad de 1000Mb/s.

Por otro lado, se realizaron los mismos cálculos, pero con un servidor real que tiene las mismas aplicaciones. El servidor web físico con la "IP 192.168.0.127" se conectó el servidor desde la terminal de Ubuntu versión 24.204 con el comando "iperf3 -s -p 8000". Luego, como cliente en el equipo físico Windows, se ejecutó el comando ".\iperf3.exe -c 192.168.0.127 -p 8000". Los resultados de la transferencia se pueden ver en la [Figura 4.12] en intervalos de 10 segundos que indican el ancho de banda promedio de 76.2 Mbps. y la cantidad de datos transferidos, de 90.9 MBytes. Los valores obtenidos se pueden considerar adecuados debido a que la red donde está conectado se encuentra diseñada para operar a 1000Mb/s.

La Tabla 4.6 presenta los valores obtenidos del servidor virtual realizando una prueba de transferencia cliente-servidor, donde la transferencia total se encuentra definido mediante la ecuación (1.1).

$$\text{Trans. (Mbits)} = MB * 8 \quad (1.1)$$

Sustituyendo, se tiene 956,301,312 bits. Después, se calcula la tasa y se obtiene como resultado 95,05 Mbps, mediante la ecuación (1.2).

$$\text{Tasa}_1 = 95,63 \text{ Mbps}; \text{Tasa}_2 = 95,05 \text{ Mbps.} \quad (1.2)$$

Para finalizar, se calculó el porcentaje de uso con la ecuación (1.3) y (1.4), y se obtiene como resultado 9.5% para ambos casos.

$$\text{uso}(\%)_1 = 9.5\% \quad (1.3)$$

$$\text{uso}(\%)_2 = 9.5\% \quad (1.4)$$

Por otro lado, también se presentaron los valores obtenidos del servidor físico. La transferencia total está dada por la ecuación (1.5).

$$\text{Trans. (Mbits)} = MB * 8 \quad (1.5)$$

Sustituyendo, se tienen 762,524,467,2 bits. Se calculó la tasa mediante la ecuación (1.6) y (1.7), con un valor de 76,17 Mbps y 76,02 Mbps.

$$\text{Tasa}_1 = 76,17 \text{ Mbps} \quad (1.6)$$

$$\text{Tasa}_2 = 76,02 \text{ Mbps} \quad (1.7)$$

Después, se calculó el porcentaje de uso, donde se tuvieron los valores de 7,6% para ambos casos, mediante la ecuación (1.8) y (1.9).

$$\text{uso}(\%)_1 = 7.6\% \quad (1.8)$$

$$\text{uso}(\%)_2 = 7.6\% \quad (1.9)$$

Todas estas operaciones se encuentran expresadas en la Tabla 4.7, mientras que en la Tabla 4.8 se elaboran algunos comentarios en torno a la comparación de estos resultados.

Parámetros	Cliente (Windows)	Servidor virtual (Linux)	Cliente (Windows)	Servidor físico (Linux)
IP	192.168.0.128	192.168.0.64	192.168.0.128	192.168.0.127
Puerto utilizado	8000	8000	8000	8000
Ancho de Banda	95.5 Mbps	95.0 Mbps	76.2 Mbps	76.0 Mbps
Transferencia total	114 MB	114 MB	90.9 MB	90.9 MB
Duración de prueba	10.00 segundos	10.06 segundos	10.01 segundos	10.03 segundos
Ancho de Banda Máximo	99.2 Mbps	100 Mbps	82.4 Mbps	92.1 Mbps
Ancho de Banda Mínimo	89.6 Mbit/s	86.2 Mbit/s	70.9 Mbps	70.2 Mbps
Estabilidad	89.6 – 99.2 Mbps	86.2 – 100 Mbps	70.9-82.4 Mbps	70.2 -92.1 Mbps
Porcentaje Utilizado	9.12% de 1 Gbps	9.12% de 1 Gbps	7.2 % de 1 Gbps	7.2% de 1 Gbps

Tabla 4.6: Datos de la prueba de Red utilizando iperf3. [Captura Propia]

concepto		Ecuación	Resultado
#1	Trans. (bits)	$Trans. (bits) = 114 * 1024^2 = 119,537,664 \text{ Bytes}$ $Trans. (bits) = 119,537,664 \text{ Bytes} * 8$	956,301,312 bits
	Tasa	$Tasa = \frac{\text{Cantidad de datos transferidos}}{\text{Tiempo empleado}}$ $Tasa_1 = \frac{956,301,312 \text{ bits}}{10}$ $Tasa_2 = \frac{956,301,312 \text{ bits}}{10.06}$	$Tasa_1 = 95,63 \text{ Mbps}$ $Tasa_2 = 95,05 \text{ Mbps}$
	uso(%)	$uso(\%) = \left(\frac{\text{Tasa Real}}{\text{Ancho de Banda Sistema}} \right) * 100\%$ $uso(\%)_1 = \left(\frac{95,63 \text{ Mbps}}{1000 \text{ Mbps}} \right) * 100\%$ $uso(\%)_2 = \left(\frac{95,05 \text{ Mbps}}{1000 \text{ Mbps}} \right) * 100\%$	$uso(\%)_1 = 9.5\%$ $uso(\%)_2 = 9.5\%$
#2	Trans. (bits)	$Trans. (bits) = 90.9 * 1024^2 = 95,315,558.4 \text{ Bytes}$ $Trans. (bits) = 95,315,558.4 \text{ Bytes} * 8$	762,524,467.2 bits

concepto		Ecuación	Resultado
	Tasa	$Tasa = \frac{\text{Cantidad de datos transferidos}}{\text{Tiempo empleado}}$ $Tasa_1 = \frac{762,524,467.2 \text{ bits}}{10.01}$ $Tasa_2 = \frac{762,524,467.2 \text{ bits}}{10.03}$	$Tasa_1 = 76,17 \text{ Mbps}$ $Tasa_2 = 76,02 \text{ Mbps}$
	uso(%)	$uso(\%) = \left(\frac{\text{Tasa Real}}{\text{Ancho de Banda Sistema}} \right) * 100\%$ $uso(\%)_1 = \left(\frac{76,17 \text{ Mbps}}{1000 \text{ Mbps}} \right) * 100\%$ $uso(\%)_2 = \left(\frac{76,02 \text{ Mbps}}{1000 \text{ Mbps}} \right) * 100\%$	$uso(\%)_1 = 7.6\%$ $uso(\%)_2 = 7.6\%$

Tabla 4.7: Ecuaciones de las pruebas realizadas. [Captura Propia]

Comentarios
<ul style="list-style-type: none"> • Caso 1#: se tiene una red de 1000 Mbps y sólo se está utilizando una tasa del 94,5 Mbps (es decir, está utilizando 9.5% de su capacidad total). • Caso 2#: se tiene una red de 1000 Mbps y sólo se está utilizando una tasa del 76 Mbps (está utilizando 7.6% de su capacidad total).

Tabla 4.8: Comentarios de los resultados obtenidos. [Captura Propia]

De este modo, comparando el desempeño de la máquina virtual contra el de la real en cuanto a la velocidad de transferencia cliente-servidor, la máquina virtual fue capaz de ejecutar del mismo modo que una máquina física, con métricas muy similares. De hecho, la transferencia total fue ligeramente superior en la máquina virtual frente a la física. Cabe aclarar que, aunque la red sea de 10,000 Mbps, se utilizaron capacidades mucho menores debido a que la tarjeta de red del equipo en que se trabajó tiene un ancho de banda de 1 Gbit/s, por lo que limita la

velocidad, tanto para la máquina física como para la virtual, y se tendría que cambiar por otra de mayor capacidad si se desea aprovechar más de la red. Sin embargo, se puede concluir, en lo que a esta métrica atañe, que lo que hace una máquina virtual es lo mismo que lo que puede hacer una física.

```

uacm@uacm-Standard-PC-i440FX-PIIX-1996:~$ iperf3 -s -p 8000
-----
Server listening on 8000
-----
Accepted connection from 192.168.0.128, port 63272
[ 5] local 192.168.0.64 port 8000 connected to 192.168.0.128 port 63273
[ ID] Interval          Transfer       Bitrate
[ 5]  0.00-1.00    sec  10.4 MBytes  87.1 Mbits/sec
[ 5]  1.00-2.00    sec  11.3 MBytes  95.0 Mbits/sec
[ 5]  2.00-3.00    sec  10.7 MBytes  89.5 Mbits/sec
[ 5]  3.00-4.00    sec  11.5 MBytes  96.4 Mbits/sec
[ 5]  4.00-5.00    sec  11.2 MBytes  93.7 Mbits/sec
[ 5]  5.00-6.00    sec  11.5 MBytes  96.1 Mbits/sec
[ 5]  6.00-7.00    sec  11.5 MBytes  96.7 Mbits/sec
[ 5]  7.00-8.00    sec  11.9 MBytes  100 Mbits/sec
[ 5]  8.00-9.00    sec  11.6 MBytes  97.2 Mbits/sec
[ 5]  9.00-10.00   sec  11.8 MBytes  98.7 Mbits/sec
[ 5] 10.00-10.06   sec    593 KBytes  86.2 Mbits/sec
-----
[ ID] Interval          Transfer       Bitrate
[ 5]  0.00-10.06   sec   114 MBytes  95.0 Mbits/sec
-----
Server listening on 8000
-----
PS C:\Users\ELGAR\Downloads\descarga\iperf3.1.1_64> .\iperf3.exe -c 192.168.0.64 -p 8000
Connecting to host 192.168.0.64, port 8000
[ 4] local 192.168.0.128 port 63273 connected to 192.168.0.64 port 8000
[ ID] Interval          Transfer       Bandwidth
[ 4]  0.00-1.00    sec  10.9 MBytes  91.0 Mbits/sec
[ 4]  1.00-2.01    sec  11.5 MBytes  95.5 Mbits/sec
[ 4]  2.01-3.01    sec  10.6 MBytes  89.6 Mbits/sec
[ 4]  3.01-4.01    sec  11.5 MBytes  96.1 Mbits/sec
[ 4]  4.01-5.00    sec  11.1 MBytes  94.0 Mbits/sec
[ 4]  5.00-6.01    sec  11.5 MBytes  95.9 Mbits/sec
[ 4]  6.01-7.01    sec  11.5 MBytes  96.5 Mbits/sec
[ 4]  7.01-8.00    sec  11.8 MBytes  99.2 Mbits/sec
[ 4]  8.00-9.01    sec  11.9 MBytes  99.0 Mbits/sec
[ 4]  9.01-10.00   sec  11.6 MBytes  98.1 Mbits/sec
-----
[ ID] Interval          Transfer       Bandwidth
[ 4]  0.00-10.00   sec   114 MBytes  95.5 Mbits/sec
[ 4]  0.00-10.00   sec   114 MBytes  95.5 Mbits/sec
-----
iperf Done.
PS C:\Users\ELGAR\Downloads\descarga\iperf3.1.1_64>

```

Figura 4.11: Tasa de Transferencia cliente y Servidor virtual. [Captura Propia iperf [58]]

```

Server listening on 8000 (test #1)
-----
Accepted connection from 192.168.0.128, port 55660
[ 5] local 192.168.0.127 port 8000 connected to 192.168.0.128 port 55661
[ ID] Interval          Transfer      Bitrate
[ 5]  0.00-1.00      sec  8.75 MBytes  73.3 Mbits/sec
[ 5]  1.00-2.00      sec  8.88 MBytes  74.4 Mbits/sec
[ 5]  2.00-3.00      sec  8.62 MBytes  72.4 Mbits/sec
[ 5]  3.00-4.00      sec  8.50 MBytes  71.3 Mbits/sec
[ 5]  4.00-5.00      sec  9.88 MBytes  82.8 Mbits/sec
[ 5]  5.00-6.00      sec  9.50 MBytes  79.7 Mbits/sec
[ 5]  6.00-7.00      sec  8.38 MBytes  70.2 Mbits/sec
[ 5]  7.00-8.00      sec  9.88 MBytes  82.8 Mbits/sec
[ 5]  8.00-9.00      sec  9.00 MBytes  75.5 Mbits/sec
[ 5]  9.00-10.00     sec  9.12 MBytes  76.5 Mbits/sec
[ 5] 10.00-10.03     sec   384 KBytes 92.1 Mbits/sec
-----
[ ID] Interval          Transfer      Bitrate
[ 5]  0.00-10.03     sec  90.9 MBytes  76.0 Mbits/sec
-----
Server listening on 8000 (test #2)
-----
PS C:\Users\ELGAR\Downloads\descarga\iperf3.1.1_64>
.\iperf3.exe -c 192.168.0.127 -p 8000
Connecting to host 192.168.0.127, port 8000
[ 4] local 192.168.0.128 port 55661 connected to 19
2.168.0.127 port 8000
[ ID] Interval          Transfer      Bandwidth
[ 4]  0.00-1.00      sec  9.00 MBytes  75.1 Mbits/sec
[ 4]  1.00-2.01      sec  8.88 MBytes  73.8 Mbits/sec
[ 4]  2.01-3.01      sec  8.62 MBytes  72.7 Mbits/sec
[ 4]  3.01-4.00      sec  8.62 MBytes  72.7 Mbits/sec
[ 4]  4.00-5.00      sec  9.75 MBytes  82.0 Mbits/sec
[ 4]  5.00-6.01      sec  9.50 MBytes  79.1 Mbits/sec
[ 4]  6.01-7.00      sec  8.38 MBytes  70.9 Mbits/sec
[ 4]  7.00-8.01      sec  9.88 MBytes  82.4 Mbits/sec
[ 4]  8.01-9.01      sec  8.88 MBytes  74.1 Mbits/sec
[ 4]  9.01-10.01     sec  9.38 MBytes  79.0 Mbits/sec
-----
[ ID] Interval          Transfer      Bandwidth
[ 4]  0.00-10.01     sec  90.9 MBytes  76.2 Mbits/sec
sender
[ 4]  0.00-10.01     sec  90.9 MBytes  76.2 Mbits/sec
receiver

iperf Done.
PS C:\Users\ELGAR\Downloads\descarga\iperf3.1.1_64>

```

Figura 4.12: Tasa de Transferencia cliente y Servidor Físico. [Captura Propia iperf [58]]

4.6.2 Captura de tráfico

Para analizar el tráfico de red del servidor físico y el servidor virtual que administra a los diferentes servidores instalados, se utilizó el programa Wireshark, el cual permite observar los picos, filtrar el tráfico, visualizar el volumen de tráfico en paquetes, etcétera. La Tabla 4.9 muestran los resultados de las solicitudes de lo que se envía y se recibe en una llamada de prueba para analizar el flujo de la voz y los datos, tanto en el servidor virtual como el físico. El tráfico de voz de la máquina virtual se presentó en un rango de entre aproximadamente de 170,000 Bit/s y picos máximos de aproximadamente 290,000 bit/s hasta llegar a 0, donde finalizó la llamada en un intervalo de tiempo de prueba de [0-800] segundo. Se obtuvo un flujo estable. Por otro lado, en el servidor físico, se observó un flujo de datos bajo, de aproximadamente 0-30,000, con un tráfico de voz de un rango de aproximadamente de 170,000 Bit/s, en un intervalo de [0 - 2000] segundo. [Figura 4.13 y Figura 4.14]. Se puede observar que tanto el servidor físico como el servidor virtual presentan las mismas características.

Servidor	Flujo	Color	Rango (Bit/s)	Picos máx. aprox. (Bit/s)
Virtual (ens18)	Voz	Verde	~170,000	~220,000
	Datos	Azul	~0	~10,000
Físico (eno1)	Voz	Verde	~170,000	~290,000
	Datos	Azul	~0	~30,000

Tabla 4.9: Datos de las pruebas realizadas del servidor Asterisk. [Captura Propia]

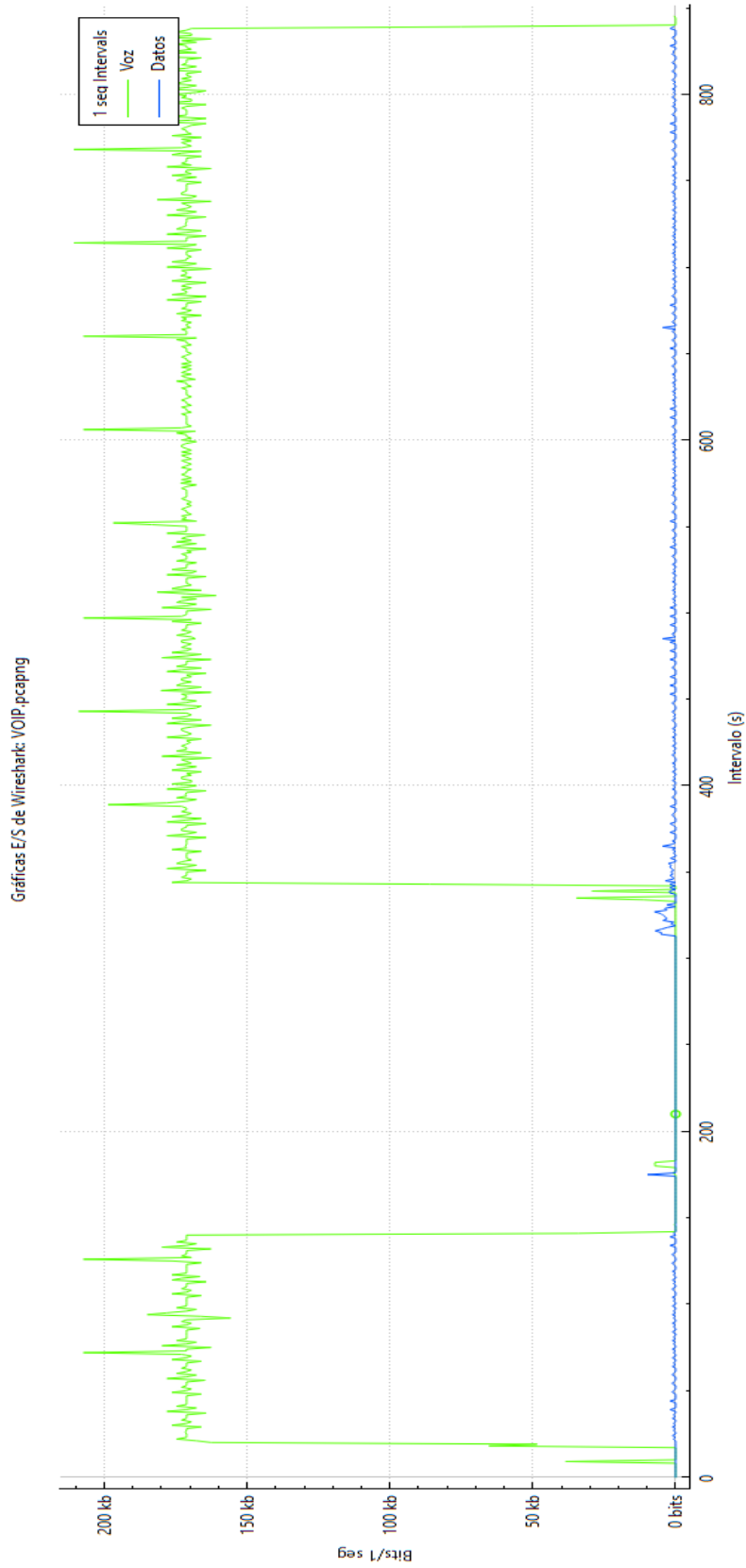


Figura 4.13: Grafica de tráfico Servidor del Servidor virtual. [Captura Propia [59]]

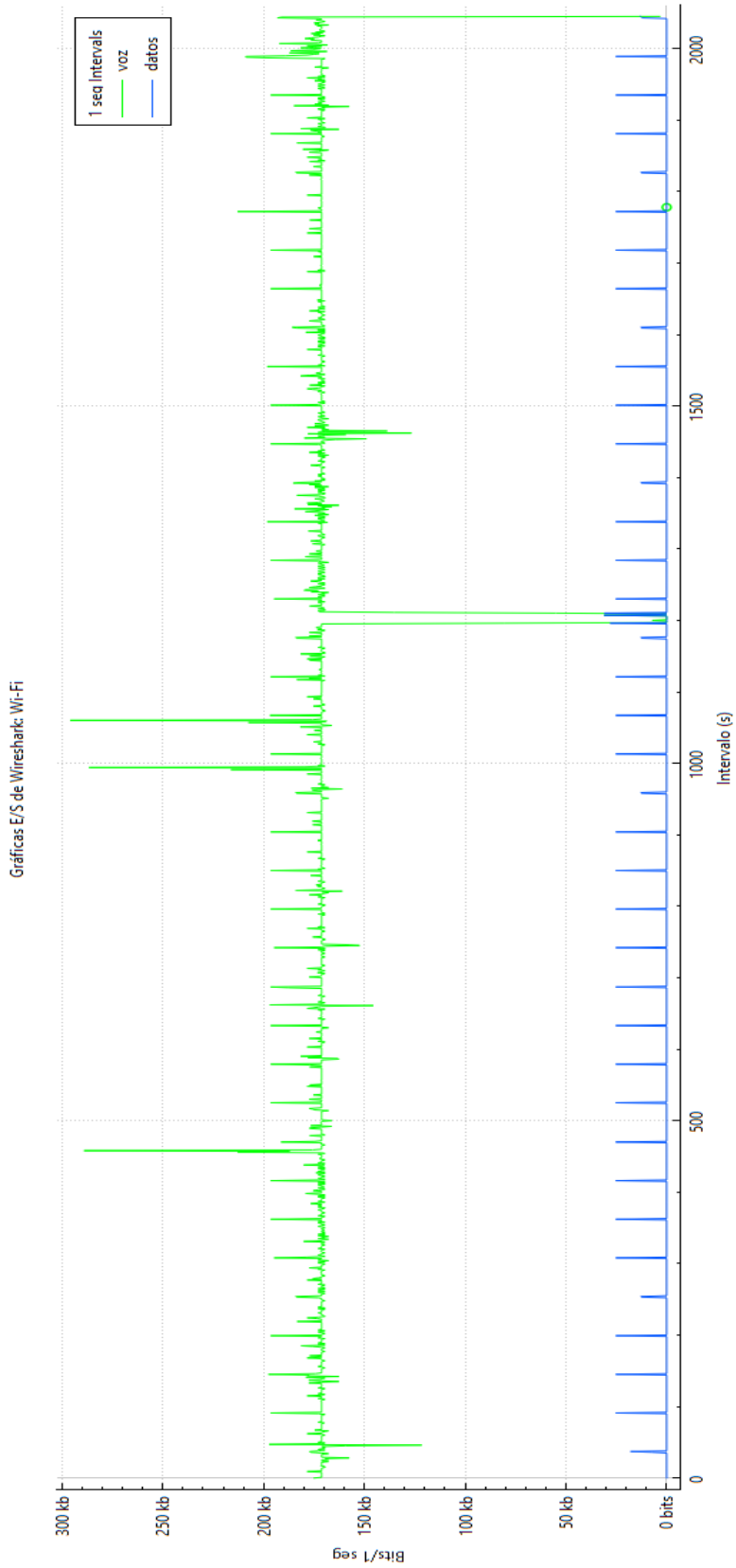


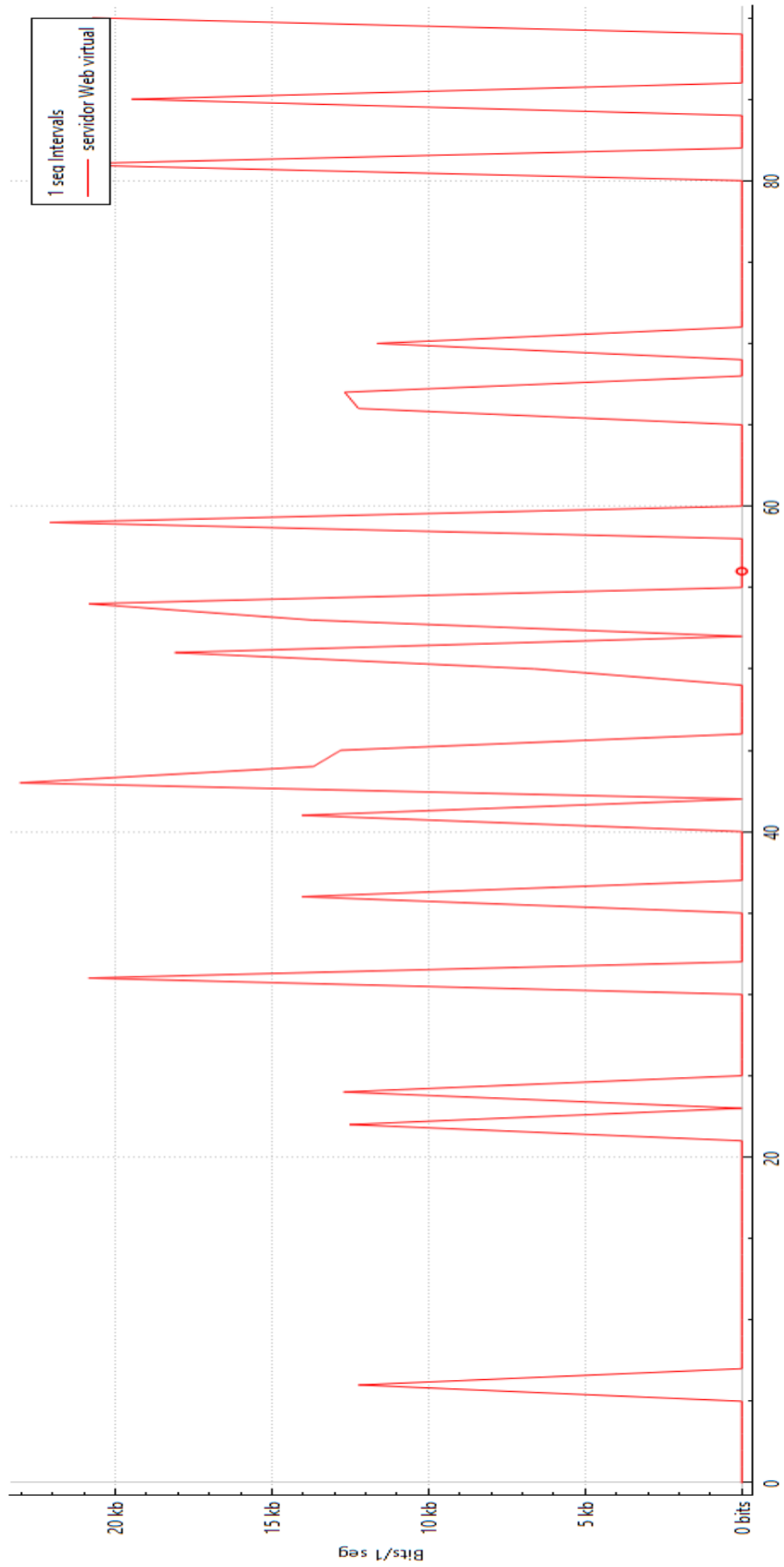
Figura 4.14: Grafica de tráfico de Servidor físico [Captura Propia [59]]

En lo que respecta al análisis de tráfico del servidor web, se obtuvieron picos máximos de aproximadamente 42,000 Bit/s para el servidor físico, y de aproximadamente 23,000 Bit/s para el servidor virtual. Con esto se puede ver los puntos donde hay tráfico en la red de la máquina física y de la máquina virtual para poder monitorearlos [Figura 4.15]. Esta gráfica es importante debido a que con eso se puede ver que las páginas se encuentran trabajando en el sistema. Para el servidor físico se cuenta con la dirección: “http://192.168.0.127/index.php” y para el servidor virtual con la dirección: “http://192.168.0.224/index.php”. [Tabla 4.10].

Servidor	Color	Servidor web	Picos máx. aprox. (Bit/s)	Picos mínimos aprox. (Bit/s)
Virtual	Rojo	192.168.0.224	~23,000 Bit/s	~10,000 Bit/s
Físico	Morado	192.168.0.127	~42,000 Bit/s	~8,000 Bit/s

Tabla 4.10: Datos de las pruebas realizadas del servidor web. [Captura Propia]

Gráficas E/S de Wireshark: Wi-Fi



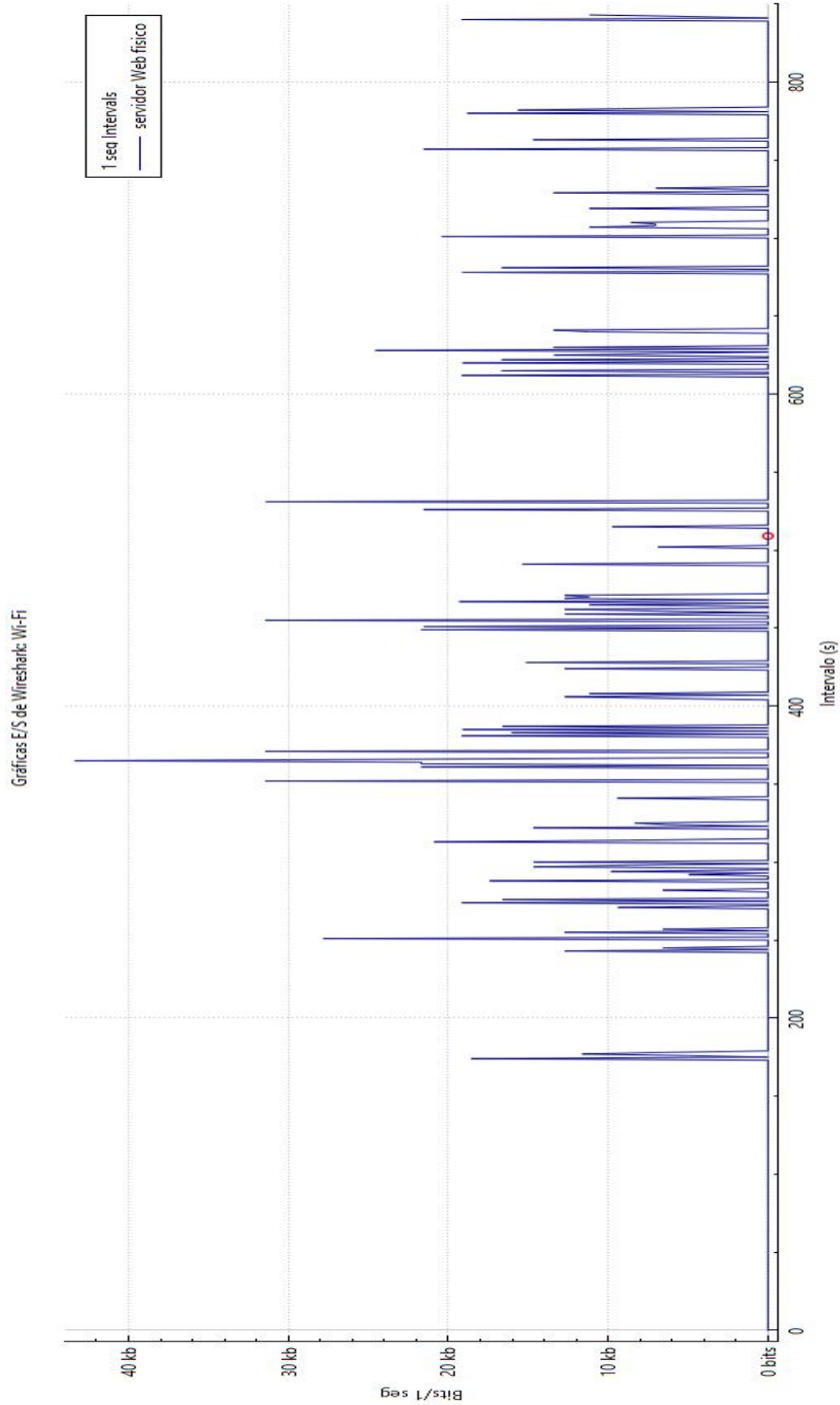


Figura 4.15: Grafica de tráfico del servidor Web virtual y servidor web físico. [Captura Propia [59]]

4.6.3 Simulación de peticiones en la máquina virtual y la máquina física.

Se utilizó la herramienta Siege para simular peticiones al servidor y medir la capacidad básica del servidor para manejar 10 solicitudes concurrentes y conocer el límite de carga. Antes de la degradación del rendimiento se utilizaron los siguientes *scripts*: “siege -c 50 -t 30s http://192.168.0.224/dashboard.php”, “siege -c 50 -t 1M http://192.168.0.224/usuarios/index.php” y “siege -c 50 -t 30s http://192.168.0.127/index.php”, “siege -c 50 -t 1M http://192.168.0.127/index.php” [Figura 4.16]. Utilizando la ecuación de transacciones por segundo, se empleó la fórmula de la ecuación (1.10).

$$tasa = \frac{\text{cantidad de transacciones}}{\text{tiempo total}} \quad (1.10)$$

Donde 14,450 transacciones (*hits*), 133,436 (*hits*), 19,979 transacciones (*hits*), 42,559 (*hits*), representan el número total de peticiones y 29.03 segundos, 59.89 segundos, 29.26 segundos 59.92 segundos representan el tiempo total de la prueba. Sustituyendo la ecuación con los valores obtenidos, se obtuvo 497.76 hits/s mediante la ecuación (1.11). Mientras que para la segunda prueba se tuvo 2,260.86 hits/s mediante la ecuación (1.12). Éstos son los resultados del servidor web de la máquina virtual. Para la tercera y cuarta prueba del servidor web de la máquina física, se tuvo 682.7 hits/s y 710.2 hits/s mediante la ecuación (1.13, 1.14).

$$Tasa_1 = \frac{14,450 \text{ hits}}{29.03 \text{ seg.}} = 497.76 \frac{\text{hits}}{s} \quad (1.11)$$

$$Tasa_2 = \frac{133,436 \text{ hits}}{59.02 \text{ seg.}} = 2,260.86 \frac{\text{hits}}{s} \quad (1.12)$$

$$Tasa_3 = \frac{19,978 \text{ hits}}{29.26 \text{ seg.}} = 682.7 \frac{\text{hits}}{s} \quad (1.13)$$

$$Tasa_4 = \frac{42,559 \text{ hits}}{59.92 \text{ seg.}} = 710.2 \frac{\text{hits}}{s} \quad (1.14)$$

En el primer caso, el resultado de la prueba se considera apropiado debido a que el servidor puede responder a más de 400 peticiones por segundo. En el segundo caso sigue siendo apropiado, al tener más de 2,000 respuestas por segundo. En el tercer caso, el resultado de

la prueba se considera apropiado debido a que el servidor puede responder a más de 600 peticiones por segundo. En el cuarto caso sigue siendo apropiado, al tener más de 700 respuestas por segundo.

Para realizar la carga por usuario, se obtuvieron 289 transacciones por usuario, mediante la ecuación (1.15). Para la segunda carga se obtuvo 2668.72 transacciones por usuario, mediante la ecuación (1.16). Para la tercera carga, se obtuvieron 399.56 transacciones por usuario, mediante la ecuación (1.17). Para la cuarta carga, se obtuvieron 851.18 transacciones por usuario, mediante la ecuación (1.18). En todos estos casos, el servidor respondió adecuadamente.

$$Carga_1 = \frac{14,450 \text{ hits}}{50 \text{ usuario}} = 289 \text{ transacciones por usuario} \quad (1.15)$$

$$Carga_2 = \frac{133,436 \text{ hits}}{50 \text{ usuario}} = 2668.72 \text{ transacciones por usuario} \quad (1.16)$$

$$Carga_3 = \frac{19,978 \text{ hits}}{50 \text{ usuario}} = 399.56 \text{ transacciones por usuario} \quad (1.17)$$

$$Carga_4 = \frac{42,559 \text{ hits}}{50 \text{ usuario}} = 851.18 \text{ transacciones por usuario} \quad (1.18)$$

Para terminar, se obtuvo el rendimiento de datos mediante la fórmula de la ecuación (1.19).

$$rendimiento_1 = \frac{\text{Datos transferidos (MB)}}{\text{Tiempo total (segundos)}} \quad (1.19)$$

Sustituyendo los valores, se obtuvo, en el primer caso, 4.91 MB/s, mediante la ecuación (1.20); en el segundo caso, 0.29 MB/s, mediante la ecuación (1.21); en el tercer caso, 5.28 MB/s, mediante la ecuación (1.22); y en el cuarto caso, 5.51 MB/s, mediante la ecuación (2.23).

$$ren_{.1} = \frac{142.72 \text{ MB}}{29.03 \text{ seg.}} = 4.91 \frac{\text{MB}}{\text{s}} \quad (1.20)$$

$$ren_{.2} = \frac{17.5 \text{ MB}}{59.02 \text{ seg.}} = 0.29 \frac{\text{MB}}{\text{s}} \quad (1.21)$$

$$ren_{.3} = \frac{154.55 \text{ MB}}{29.26 \text{ seg.}} = 5.28 \frac{\text{MB}}{\text{s}} \quad (1.22)$$

$$ren_{.4} = \frac{330.23 \text{ MB}}{59.92 \text{ seg.}} = 5.51 \frac{\text{MB}}{\text{s}} \quad (1.23)$$

	Métrica	Página PHP servidor de la Máquina virtual	Página PHP servidor de la Máquina virtual	Página PHP servidor físico	Página PHP servidor físico
1	Transacciones	14,450 hits	133,436 hits	19,978 hits	42,559 hits
2	Tiempo total	29.03 seg.	59.02 seg.	29.26 seg.	59.92 seg.
3	Datos transferidos	142.72 MB	17.5 MB	154.55 MB	330.23 MB
4	Tiempo de respuesta	0.1 seg	0.02 seg	0.07 seg.	0.07 seg.
5	Transacciones por segundo	497.76	2,260.86	682.78	710.26
6	Throughput rendimiento	4.92 MB/ seg	0.3 MB/ seg	5.28 MB/ seg.	5.51 MB/ seg.
7	Concurrencia	49.5	49.84	48.19	48.7
8	Transacción más larga	0.52 seg.	0.34 seg.	1.3 seg.	0.71 seg.
9	Transacción más corta	0.0 seg.	0.0 seg.	0.0 seg.	0.0 seg.

Tabla 4.11: Pruebas obtenidos en el servidor web. [Captura Propia]

```
uacm@uacm-Standard-PC-i440FX-PIIX-1996:~$ siege -c 5
0 -t 30s http://192.168.0.224/dashboard.php
** SIEGE 4.0.4
** Preparing 50 concurrent users for battle.
The server is now under siege...
Lifting the server siege...
Transactions:          14450 hits
Availability:          100.00 %
Elapsed time:          29.03 secs
Data transferred:     142.72 MB
Response time:         0.10 secs
Transaction rate:      497.76 trans/sec
Throughput:            4.92 MB/sec
Concurrency:           49.50
Successful transactions: 14454
Failed transactions:   0
Longest transaction:   0.52
Shortest transaction:  0.00

uacm@uacm-Standard-PC-i440FX-PIIX-1996:~$ siege -c 5
0 -t 1M http://192.168.0.224/usuarios/index.php
** SIEGE 4.0.4
** Preparing 50 concurrent users for battle.
The server is now under siege...
Lifting the server siege...
Transactions:          133436 hits
Availability:          100.00 %
Elapsed time:          59.02 secs
Data transferred:     17.50 MB
Response time:         0.02 secs
Transaction rate:      2260.86 trans/sec
Throughput:            0.30 MB/sec
Concurrency:           49.84
Successful transactions: 66751
Failed transactions:   0
Longest transaction:   0.34
Shortest transaction:  0.00
```

```
eladio@eladio-HP-Compaq-Elite-8300-SFF:~$ siege -c 50 -t 30s
http://192.168.0.127/index.php

{
  "transactions":          19978,
  "availability":         100.00,
  "elapsed_time":         29.26,
  "data_transferred":     154.55,
  "response_time":        0.07,
  "transaction_rate":     682.78,
  "throughput":           5.28,
  "concurrency":          48.19,
  "successful_transactions": 19978,
  "failed_transactions":   0,
  "longest_transaction":   1.30,
  "shortest_transaction":  0.00
}
eladio@eladio-HP-Compaq-Elite-8300-SFF:~$ siege -c 50 -t 1M
http://192.168.0.127/index.php

{
  "transactions":          42559,
  "availability":         100.00,
  "elapsed_time":         59.92,
  "data_transferred":     330.23,
  "response_time":        0.07,
  "transaction_rate":     710.26,
  "throughput":           5.51,
  "concurrency":          48.70,
  "successful_transactions": 42565,
  "failed_transactions":   0,
  "longest_transaction":   0.71,
  "shortest_transaction":  0.00
}
```

Figura 4.16: Resultados de la Máquina física y la máquina virtual. [Captura Propia]

4.7 Análisis de métricas del servidor físico y el servidor virtual

Para monitorear el rendimiento del sistema, se implementaron los comandos “top” y “htop” desde la consola de Proxmox VE. Estos comandos arrojaron el porcentaje ocupado del CPU, memoria RAM, memoria *caché* y memoria compartida del servidor Proxmox VE. Para monitorear el rendimiento del equipo utilizando las métricas de la CPU, se tomaron los porcentajes de CPU ocupado e inactivo. El uso total de la CPU se consideró con la fórmula de la ecuación (1.24).

$$User(\%) + System(\%) + Wa(\%) + Idle(\%) = 100\% \quad (1.24)$$

Por su parte, el uso total de la memoria se calculó con la ecuación (1.25).

$$Uso\ Memoria\ RAM(\%) = \left(\frac{M.RAM\ usado(MiB)}{M.RAM\ Total(MB)} \right) * 100\% \quad (1.25)$$

Donde su unidad son los Mebibytes (MiB); y el uso de la memoria *caché* se calculó con la ecuación (1.26).

$$Uso\ Memoria\ Caché(\%) = \left(\frac{M.usado(MiB)}{M.Total(MB)} \right) * 100\% \quad (1.26)$$

Primero, se realizaron pruebas de la máquina física sin virtualización y después se realizaron las mismas pruebas con el equipo físico utilizando máquinas virtuales. La Tabla 4.12 presenta las métricas obtenidas de la CPU física; la Tabla 4.13, las métricas obtenidas de la CPU de la máquina virtual; la Tabla 4.14, las métricas obtenidas de la memoria RAM física; la Tabla 4.15, las métricas obtenidas de la memoria RAM de la máquina virtual; la Tabla 4.16, las métricas obtenidas de la memoria *caché* física; la Tabla 4.17, las métricas obtenidas de la memoria *caché* de la máquina virtual; y la tabla 4.18, las métricas obtenidas de la memoria compartida (SWAP) de la máquina virtual. Con respecto de éstas últimas, el primer proceso (#1) es el de encendido; el segundo proceso (#2) implementa los servidores Asterisk, Web —con su base de datos— y Samba; y el tercer proceso (#3) ejecuta otras tareas sobre el navegador, como videos.

Por otro lado, para el caso de la máquina virtual, el primer proceso (#1) fue el del equipo encendido; el segundo proceso (#2), el levantamiento de una máquina virtual; el tercer proceso (#3), el levantamiento de dos máquinas virtuales; y el cuarto proceso (#4), la implementación en las máquinas virtuales de los servidores Asterisk, Web —con su base de datos— y Samba.

En la Figura 4.17, el color verde es el porcentaje de tiempo que el usuario (*user*) dedica a ejecutar procesos; el naranja, el porcentaje de tiempo que el sistema (*system*) dedica a la CPU; el morado, el porcentaje de operaciones de disco/red (*wait*); y el gris, el porcentaje de tiempo en que el CPU se encuentra inactivo (*idle*). Asimismo, en la Figura 4.18, el color verde representa el porcentaje de uso de la memoria RAM en cada proceso con un adecuado porcentaje de trabajo; el amarillo, su porcentaje de incremento; y el naranja, la memoria con más del 90% del uso. En la Figura 4.19, el amarillo representa el porcentaje de uso de la memoria *caché* para cada proceso. Y en la Figura 4.20, el naranja representa el porcentaje de uso de la memoria compartida. Finalmente, en la Tabla 4.19 se interpretan los resultados obtenidos. Para todos los casos, el color gris representa la memoria y espacio disponible.

	User (%)	System (%)	I/O Wait(%)	Idle (id%)	Uso de CPU	Uso total de CPU
#1	1.20%	2.50%	1,0%	95.30%	1.20%+2.50%+1,0%=4.70%	4.70%+95.30%=100%
#2	6.00%	10.70%	0,0%	83.30%	6.00%+10.70%=16.70%	16.70%+83.30%=100%
#3	6.00%	7.10%	0,0%	86.90%	6.00%+7.10%=13.10%	13.10%+86.90%=100%

Tabla 4.12: Porcentaje de recursos de la CPU de la máquina física. [Elaboración Propia]

	User (%)	System (%)	I/O Wait(%)	Idle (%)	Uso de CPU	Uso total de CPU
#1	0,1%	0,0%	0,0%	99,9%	0,1%+0,0%+0,0%=0.10%	99,9%+0,10%=100%
#2	2,9%	0,5%	13,3%	81,0%	2,9%+0,5%+13,3%=16.70%	81,0%+16,70%=97.70%
#3	7,4%	0,0%	11,1%	81,5%	7,4%+0,0%+11,1%=18.50%	81,5%+18.50%=100%
#4	4,7%	0,6%	11,8%	79,4%	4,7%+0,6%+11,8%=17.10%	17,10%+79,10%=96.5%

Tabla 4.13: Porcentaje de recursos de la CPU de la máquina virtual. [Elaboración Propia]

	Libre (MiB)	Usado (MiB)	Disponible (MiB)	Uso de Memoria RAM	Uso total de RAM (%)
#1	4830.6	1895.5	5919.9	$RAM(\%) = \left(\frac{1895.5 \text{ MiB}}{7815.4 \text{ MiB}} \right) * 100\%$	24.3%
#2	4712.6	1976.9	5838.5	$RAM(\%) = \left(\frac{1895.5 \text{ MiB}}{7815.4 \text{ MiB}} \right) * 100\%$	25.3%
#3	3087.3	2709.2	5106.2	$RAM(\%) = \left(\frac{1895.5 \text{ MiB}}{7815.4 \text{ MiB}} \right) * 100\%$	34.7%

Tabla 4.14: Porcentaje de recursos de Memoria RAM de la máquina física. [Elaboración Propia]

	Libre (MiB)	Usado (MiB)	Disponible (MiB)	Ecuaciones	Uso total de RAM (%)
#1	5891,1	1504,8	6313,4	$RAM(\%) = \left(\frac{1504,8 \text{ MiB}}{7818,1 \text{ MiB}} \right) * 100\%$	19.2%
#2	3259.9	4414,2	3403,9	$RAM(\%) = \left(\frac{4414,2 \text{ MiB}}{7818,1 \text{ MiB}} \right) * 100\%$	56.4%
#3	155.4	7624.2	193.9	$RAM(\%) = \left(\frac{7624.2 \text{ MiB}}{7818,1 \text{ MiB}} \right) * 100\%$	97.5%
#4	152,2	7708,5	109,7	$RAM(\%) = \left(\frac{7708,5 \text{ MiB}}{7818,1 \text{ MiB}} \right) * 100\%$	98.5%

Tabla 4.15: Porcentaje de recursos de Memoria RAM de la máquina virtual. [Elaboración Propia]

	Memoria Total (MiB)	Caché (MiB)	Ecuaciones	Uso total de Cache (%)
#1	7815.4	1517.9	$Cache(\%) = \left(\frac{1517.9 \text{ MiB}}{7815.4 \text{ MiB}}\right) * 100\%$	19.4 %
#2	7815.4	1571.1	$Cache(\%) = \left(\frac{1571.1 \text{ MiB}}{7815.4 \text{ MiB}}\right) * 100\%$	20.1 %
#3	7815.4	2538.1	$Cache(\%) = \left(\frac{2538.1 \text{ MiB}}{7815.4 \text{ MiB}}\right) * 100\%$	32.5 %

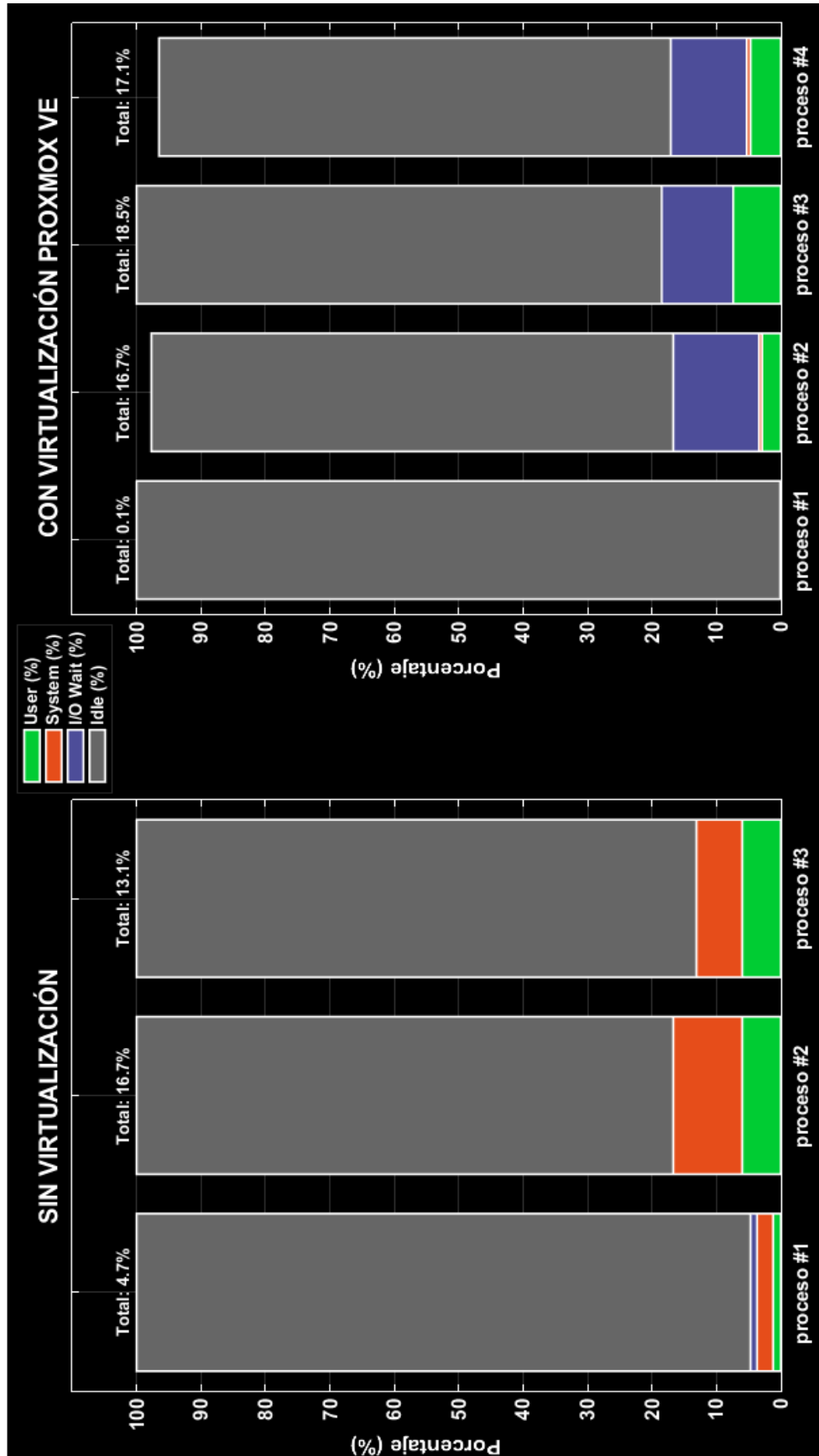
Tabla 4.16: Porcentaje de recursos de Memoria Cache de la máquina física. [Elaboración Propia]

	Memoria Total (MiB)	Caché (MiB)	Ecuaciones	Uso total de Cache (%)
#1	7818.1	703.8	$Cache(\%) = \left(\frac{703.8 \text{ MiB}}{7818.1 \text{ MiB}}\right) * 100\%$	9.53%
#2	7818.1	427.2	$Cache(\%) = \left(\frac{427.2 \text{ MiB}}{7818.1 \text{ MiB}}\right) * 100\%$	5.61%
#3	7818.1	321.1	$Cache(\%) = \left(\frac{321.1 \text{ MiB}}{7818.1 \text{ MiB}}\right) * 100\%$	4.86%
#4	7818.1	207.6	$Cache(\%) = \left(\frac{207.6 \text{ MiB}}{7818.1 \text{ MiB}}\right) * 100\%$	2.63%

Tabla 4.17: Porcentaje de recursos de Memoria Cache de la máquina virtual. [Elaboración Propia]

	Libre (MiB)	Usado (MiB)	Ecuaciones	SWAP
#1	7816	0	$SWAP(\%) = \left(\frac{0 \text{ MiB}}{7816 \text{ MiB}}\right) * 100\%$	0%
#2	7816	0	$RAM(\%) = \left(\frac{0 \text{ MiB}}{7816 \text{ MiB}}\right) * 100\%$	0%
#3	7815.7	0.3	$RAM(\%) = \left(\frac{0.3 \text{ MiB}}{7816 \text{ MiB}}\right) * 100\%$	0.0038%
#4	7189.4	626.6	$SWAP(\%) = \left(\frac{626.6 \text{ MiB}}{7816 \text{ MiB}}\right) * 100\%$	8.02%

Tabla 4.18: Porcentaje de recursos de Memoria compartida de la máquina virtual. [Elaboración Propia]



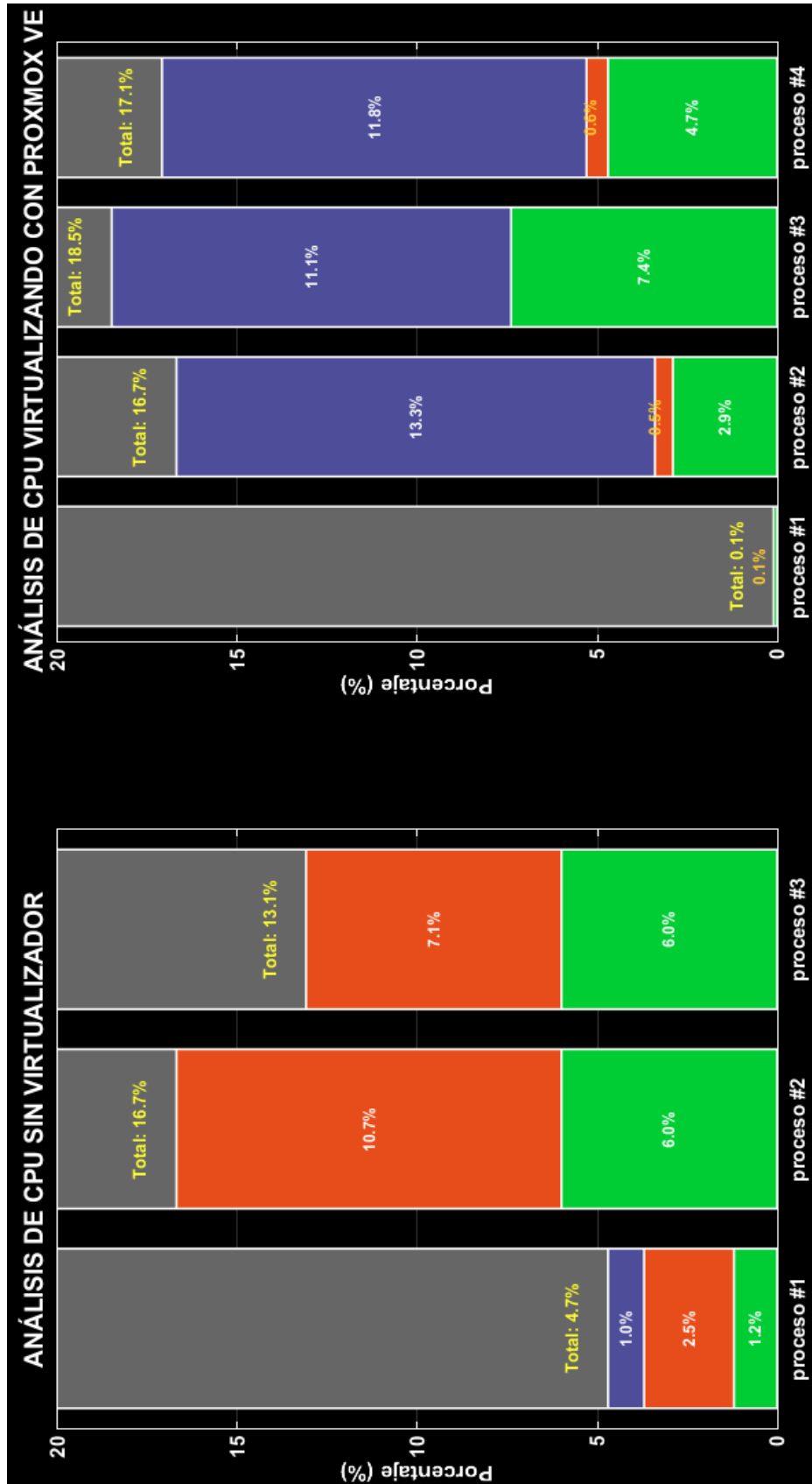


Figura 4.17: Grafica de resultados de la CPU de la Máquina física y la Máquina virtual. [Captura Propia]

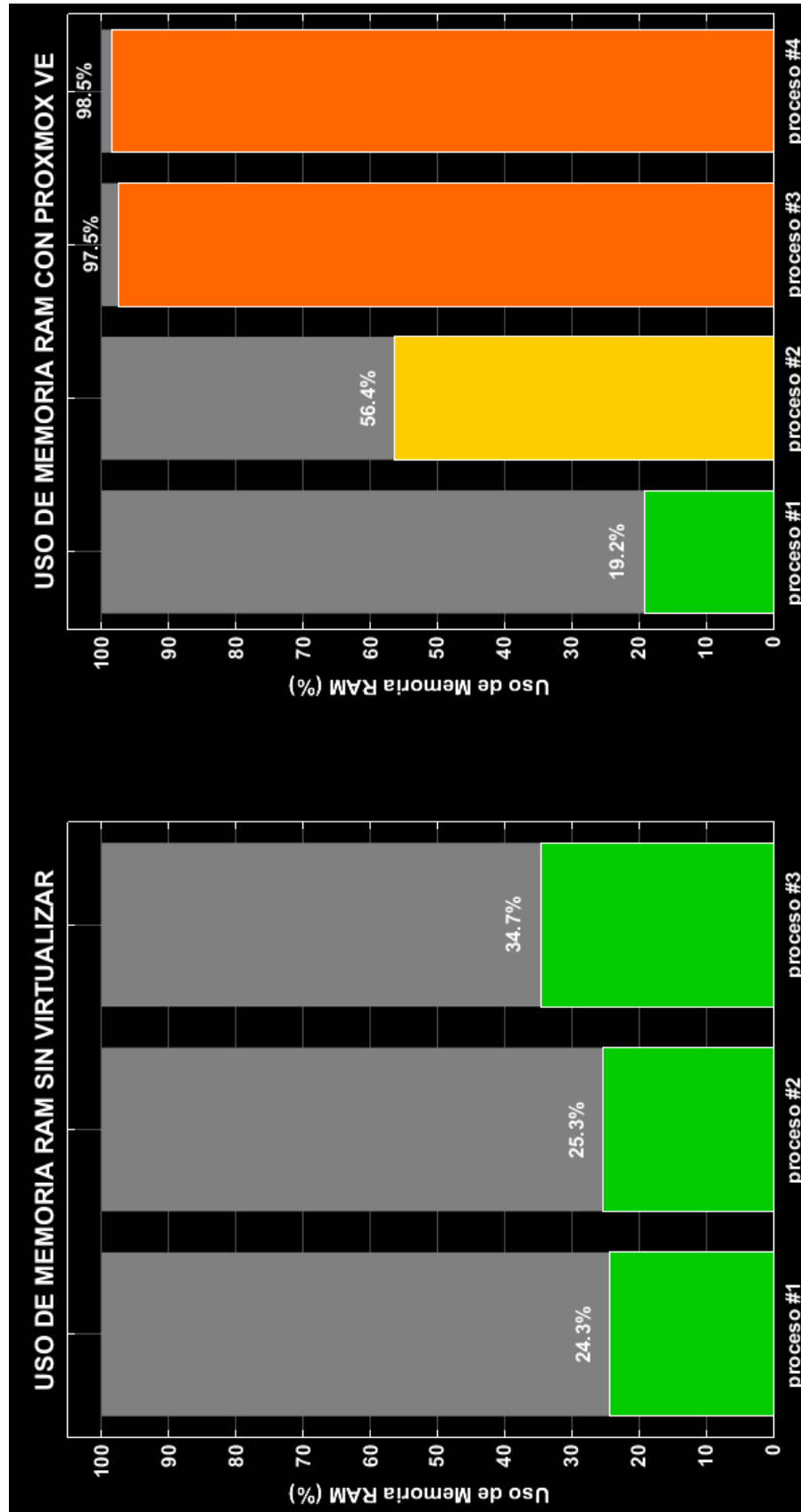


Figura 4.18: Grafica de resultados de la Memoria RAM de la Máquina física y la Máquina virtual. [Captura Propia]

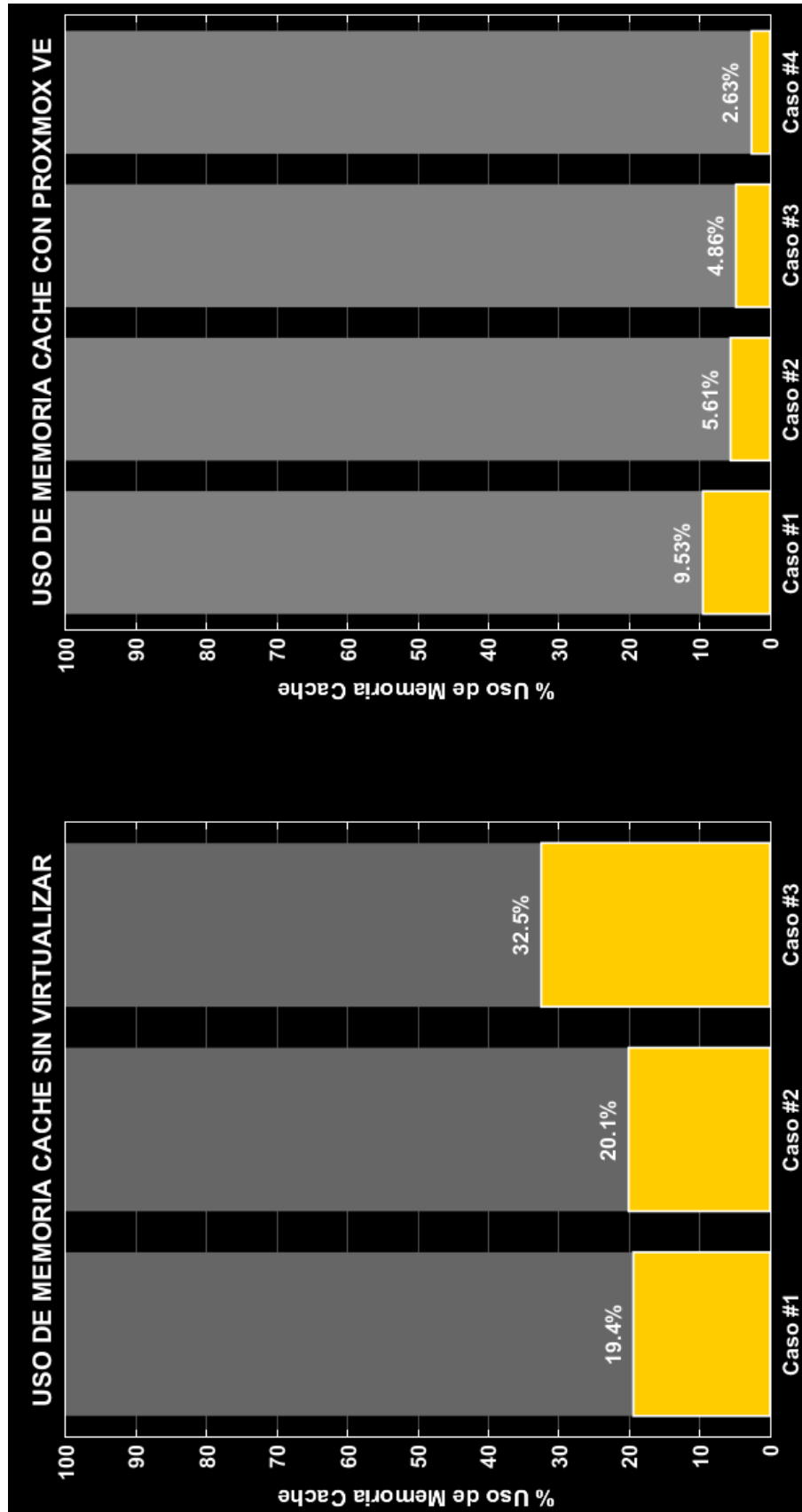


Figura 4.19: Grafica de resultados de la Memoria Cache de la Máquina física y la Máquina virtual. [Captura Propia]

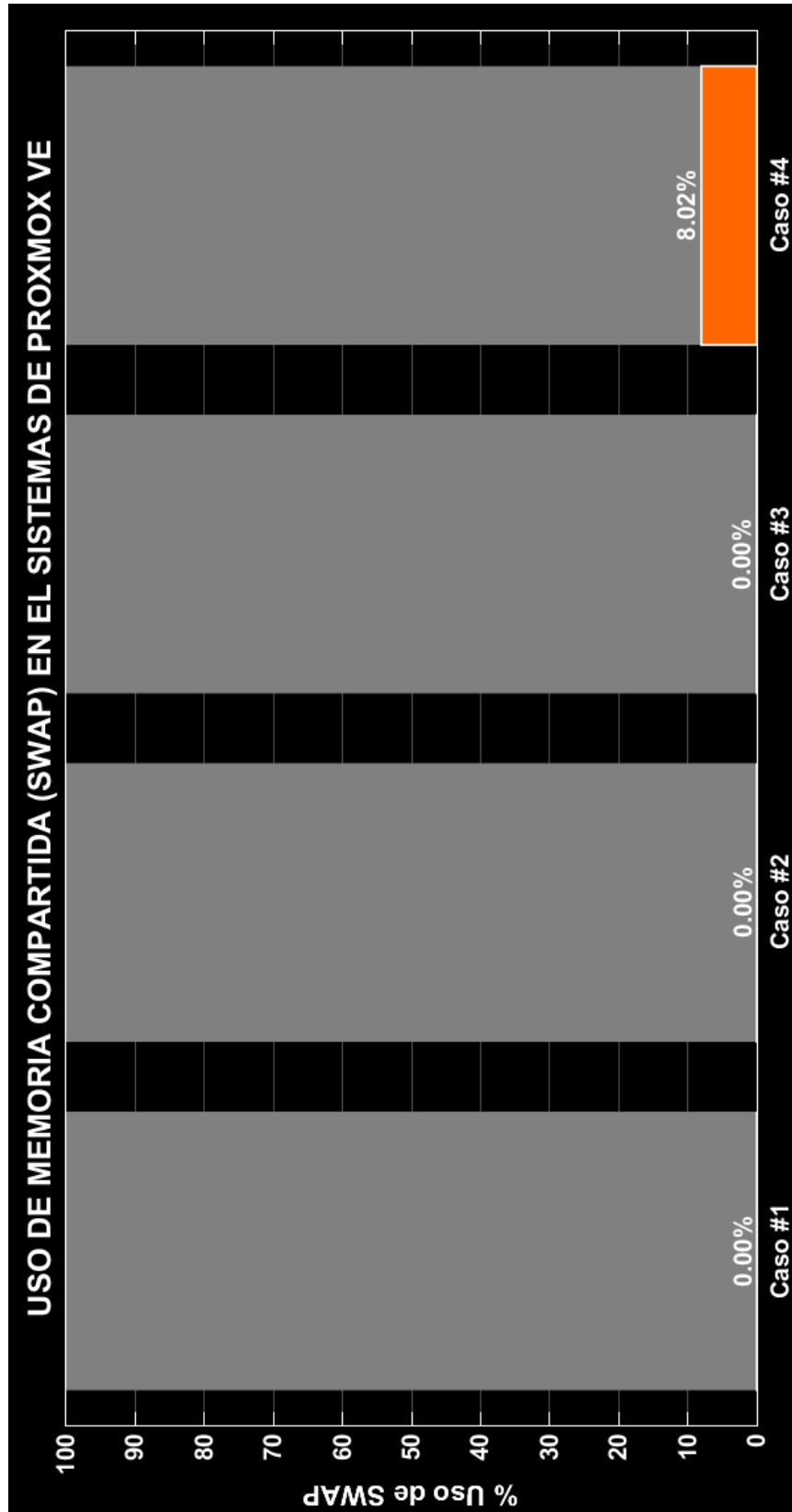


Figura 4.20: Grafica de resultados de memoria de intercambio(SWAP) de la Máquina virtual. [Captura Propia]

Comentarios

- #1: debido a que se tiene un uso del 4.70% y un espacio libre de 95.30% en CPU, se considera que el sistema es adecuado, ya que cumple con la primera parte Uso total < 80% [Figura 4.17]. Además, debido a que se tiene un uso del 24.3% y un espacio libre de 75.7% en RAM, se considera que el sistema es adecuado, ya que cumple con la primera parte Uso total < 80% [Figura 4.18].
- #1: debido a que se tiene un uso del 0.10% y un espacio libre de 99.9% en CPU, se considera que el sistema es adecuado, ya que cumple con la primera parte (uso total < 80%), tomando en cuenta que el *software* Proxmox VE está encendido, pero sin virtualizar ninguna máquina [Figura 4.17]. Asimismo, debido a que se tiene un uso del 19.2% y un espacio libre de 80.7% en RAM, se considera que el sistema es adecuado, ya que cumple con la primera parte (uso total < 80%) [Figura 4.18].
- #2: debido a que se tiene un uso del 16.70% y un espacio libre de 83.30% en CPU, se considera que el sistema es adecuado, ya que cumple con la primera parte Uso total < 80% [Figura 4.17]. Además, debido a que se tiene un uso del 25.3% y un espacio libre de 74.7% en RAM, se considera que el sistema es adecuado, ya que cumple con la primera parte Uso total < 80% [Figura 4.18].
- #2: debido a que se tiene un uso del 16.70% y un espacio libre de 81.0% en CPU, se considera que el sistema es adecuado, ya que cumple con la primera parte (uso total < 80%). Aquí, la primera máquina virtual ya se encuentra encendida [Figura 4.17]. Asimismo, debido a que se tiene un uso del 56.4% y un espacio libre de 43.5% en RAM, se considera que el sistema es adecuado, ya que cumple con la primera parte (uso total < 80%) [Figura 4.18].

Comentarios

- #3: debido a que se tiene un uso del 13.10% y un espacio libre de un 86.90% en CPU, se considera que el sistema es adecuado, ya que cumple con la primera parte Uso total < 80% [Figura 4.17]. Además, debido a que se tiene un uso del 34.7% y un espacio libre de un 65.3% en RAM, se considera que el sistema es adecuado, ya que cumple con la primera parte Uso total < 80% [Figura 4.18].
- #3: debido a que se tiene un uso del 18.50% y un espacio libre de un 81.5% en CPU, se considera que el sistema es adecuado, ya que cumple con la primera parte (uso total < 80%). En este momento, las dos máquinas virtuales se encuentran ya encendidas [Figura 4.17]. Por otro lado, debido a que se tiene un uso del 97.5% y un espacio libre de un 2.4%, pudiera considerarse que el sistema no es adecuado, ya que cumple con el tercer caso ($\geq 90\%$); sin embargo, se utiliza la memoria de intercambio para que el sistema evite bloqueos y mantenga los procesos activos con un porcentaje del 0.0038% [Figura 4.18].
- #4: debido a que se tiene un uso del 17.10% y un espacio libre de un 86.90% en CPU, se considera que el sistema es adecuado, ya que cumple con la primera parte (uso total < 80%). En este momento, la máquina virtual se encuentra encendida y con carga de trabajo, como ingresar a la base de datos, realizar una llamada y conectarse con SSH [Figura 4.17]. Por otro lado, debido a que se tiene un uso del 98.5% y un espacio libre de un 1.4%, pudiera considerarse que el sistema no es adecuado, ya que cumple con el tercer caso ($\geq 90\%$); sin embargo, se utiliza la memoria de intercambio para que el sistema evite bloqueos y mantenga los procesos activos con un porcentaje del 8.2% [Figura 4.20].

Tabla 4.19: Comentarios de las Métricas de la Máquina física y la Máquina virtual. [Elaboración Propia]

En suma, la conclusión del análisis comparativo del rendimiento de la computadora antes y después de levantar las máquinas virtuales revela que, de acuerdo con el uso de la CPU, antes de la intervención el equipo sin virtualizar presentaba una baja demanda como máximo (13.1%) [Figura 4.17], porcentaje que se elevó con una diferencia mínima (17.1%) [Figura 4.17] tras virtualizar. De acuerdo con el uso del *idle*, que da cuenta de la inactividad del equipo, antes de la implementación se presentó un alto nivel de inactividad (86.9%), mientras después de ésta se dio un bajo nivel de inactividad (82.9%). [Figura 4.17], debido a que el equipo se empezó a utilizar de una manera más rápida con los servidores disponibles, lo cual es adecuado ya que el procesador tiene aún mucho más que dar. De hecho, antes de la implementación de máquinas virtuales, la memoria *caché* presentaba un nivel de uso de 32.5%, mientras después se logró un bajo nivel de inactividad (2.63%). Y de acuerdo con el uso de la memoria RAM, antes se presentaba una demanda moderada de 34.7%, pero después de virtualizar se presentó un alto porcentaje de memoria usada (98.5%) y un bajo porcentaje de memoria libre (1.5%), lo cual revela que con la virtualización la memoria RAM trabaja al tope y, por lo tanto, para realizar más tareas se necesitaría más memoria. En tal sentido, un incremento de memoria RAM sería beneficioso para trabajar mejor; sin embargo, el uso de memoria compartida presentó un bajo porcentaje: 8.02% [Figura 4.20], lo cual es muy beneficioso para que el equipo no se vea afectado en los servidores que se implementaron y el sistema no solicite más memoria. Es por tal motivo que nunca se reflejó en el sistema alguna anomalía como lentitud.

CAPÍTULO 5

CONCLUSIONES



En la actualidad, las innovaciones en tecnología avanzan a toda velocidad. Sistemas operativos reconocidos como Windows pronto dejan de ser compatibles para los equipos a unos 4 a 7 años de haber salido al mercado, los cuales tienen una vida útil sólo en el breve período en que son tendencia tras su lanzamiento. Esta obsolescencia programada de las computadoras representa un problema para las universidades y centros de estudios, pues deben realizar esfuerzos presupuestarios para mantenerse actualizadas renovando constantemente sus equipos. Es por ello que en este trabajo se implementó y administró un entorno virtual utilizando como un *software* de hipervisor tipo 1 Proxmox VE en el Laboratorio de Telecomunicaciones, utilizando los equipos disponibles, con el fin de mejorar su rendimiento y ampliar su vida útil. Al implementar las máquinas virtuales en equipos disponibles en un entorno virtual, se pudo prolongar su vida útil contribuyendo a la optimización de los recursos tecnológicos de esta Universidad.

De este modo, se cumplió con el objetivo general del trabajo de “Implementar y administrar un entorno de virtualización utilizando Proxmox VE (Entorno Virtual) versión 8.3 como *software* principal en los equipos disponibles en el Laboratorio de Telecomunicaciones de la Universidad Autónoma de la Ciudad de México (UACM), Plantel Casa Libertad para extender su vida útil y mejorar su rendimiento”. A través de los resultados obtenidos, se demostró que la virtualización permitió hacer todo aquello que se puede hacer en una computadora física, con los mismos sistemas operativos y servidores, pero generando dos máquinas virtuales por cada equipo físico, lo que equivale a duplicar la cantidad de espacios para que los estudiantes realicen pruebas y, así, a maximizar el empleo de los recursos tecnológicos de la universidad.

Las máquinas trabajaron con adecuado rendimiento, sin alteraciones ni lentitud, y frente al uso de una máquina física, la virtual sólo incrementó el uso de la CPU en 5%, al tiempo que redujo la inactividad en 4% y bajó el uso de la memoria *caché* en 29.8%. El único indicador no

favorable fue un incremento del uso de la memoria RAM al 98.5%; sin embargo, la memoria compartida actuó para salvar al sistema de la saturación con un uso del 8.2%, por lo que nunca se experimentó lentitud de los equipos en la práctica. Estos resultados dejan ver que la máquina virtual permite hacer más trabajo sin saturación al tiempo que se ejerce menor presión sobre la memoria *caché*, entregando así más trabajo efectivo con la misma fluidez y controlando el riesgo de saturación de la memoria RAM.

En cuanto a la velocidad de transferencia, el ancho de banda promedio fue de 95.5 Mbps. (0.9 Gbit/s) y la cantidad de datos transferidos, de 114 MBytes, valores que se consideraron adecuados debido a que la red está diseñada para operar a una velocidad de 1000 Mb/s (1 Gbit/s) y que son superiores a los resultados de la máquina física, que tuvo un ancho de banda promedio de 76.2 Mbps. (0.7 Gbit/s) y una cantidad de datos transferidos, de 90.9 MBytes (0.9 Gbit/s). Esto indica que la máquina virtual aprovecha de manera más eficiente la capacidad de la red, logrando transmitir más información en menos tiempo y con mayor estabilidad, lo que refleja una mejor optimización del entorno virtual para operaciones intensivas en datos.

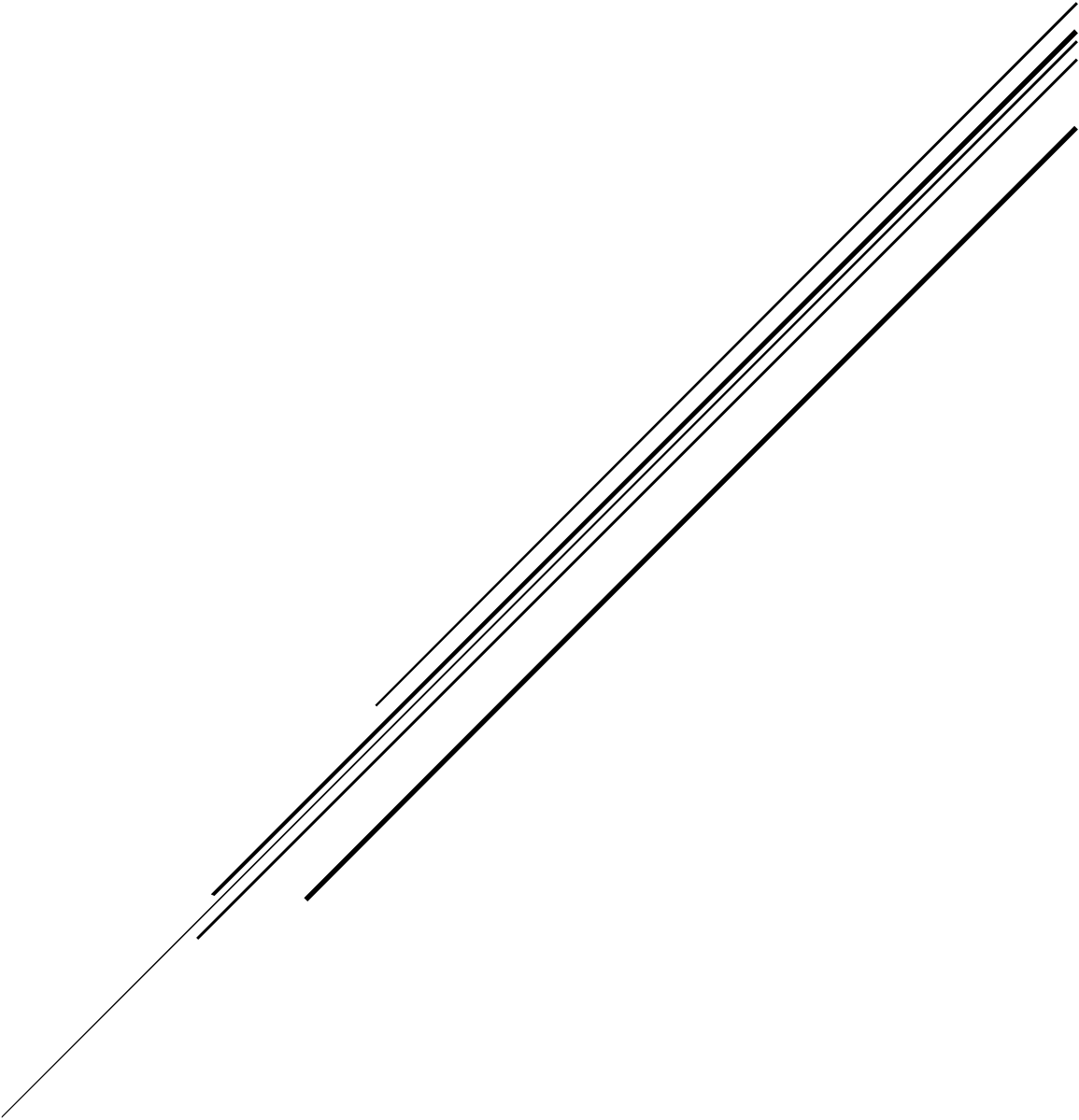
El tráfico de voz, en la máquina física se presentó en un rango de ~170,000 Bit/s y picos máximos de ~290,000 bit/s hasta llegar a 0, con un flujo de datos de ~30,000 Bit/s y mejor que el del servidor de la máquina virtual, que tuvo un flujo de datos de ~10,000 Bit/s y picos de hasta ~220,000 Bit/s. En el servidor web, se obtuvieron picos máximos de aproximadamente ~42,000 Bit/s para el servidor físico y ~23,000 Bit/s para el virtual. Esto demuestra una mejor gestión de la carga de red y menor fluctuación en la transmisión de paquetes. Aunque los picos del servidor web fueron menores, hay una administración más equilibrada del tráfico, evitando saturaciones y manteniendo un rendimiento más uniforme.

En la simulación de peticiones para acceso a páginas web a ambas máquinas, la máquina virtual fue superior en eficiencia de manejo de solicitudes, ya que tuvo más transacciones por segundo (1,379.31 en promedio frente a 696.52 del servidor físico), tiempo de respuesta promedio menor (0.06 segundos contra 0.07 segundos del servidor físico) y una transacción más larga significativamente más corta (0.43 segundos contra 1.01 segundos del servidor físico); sin embargo, la máquina física destacó por su capacidad de entrada/salida transferencia de datos (*rendimiento* promedio de 5.40 MB/s vs. 2.61 MB/s de la máquina virtual) y mayor volumen de éstos (242.39 MB vs. 80.11 MB de la máquina virtual). Por consiguiente, la máquina virtual se recomendaría para atender muchas solicitudes pequeñas con baja latencia, mientras que la física rinde de mejor forma y con mayor consistencia a comportamientos que requieran respuestas pesadas o mover más datos por segundo.

Cabe resaltar que, para la instalación del sistema, se llevaron a la práctica conocimientos previos adquiridos a lo largo de la formación profesional entre los cuales se cuenta la administración del sistema Linux. Este *software* se empleó en el desarrollo de este trabajo instalándolo en las máquinas virtuales adecuadas y utilizando los recursos disponibles. Al tratarse de un sistema operativo más ligero, fue muy útil para correr los programas y tener compatibilidad como el equipo real. En cumplimiento de los objetivos particulares definidos al inicio del trabajo, a lo largo de este proceso, se documentó la instalación y operación del hipervisor Proxmox VE versión 8.3; se configuraron y administraron las máquinas virtuales y se implementaron mecanismos de respaldo, *snapshots* y almacenamiento compartido. Todo esto ayuda a crear un entorno de visualización y que es posible utilizar a su mayor capacidad.

Para continuar con el trabajo a futuro, se propone realizar copias de seguridad mínimo una vez al mes; incrementar el número de nodos mínimo una vez al año —si así se requiriera— anexar credenciales primarias para los profesores y secundarias para los estudiantes y administrarlas; e implementar la automatización de procesos. También se sugiere mantener observación sobre la saturación de la memoria RAM en los equipos y, si se llegara a presentar alguna anomalía, considerar su expansión de 8 GB a 16 GB (8 GB por máquina), especialmente si se deseara instalar más de dos máquinas por equipo. Por último, se estima que sería posible implementar esta estrategia en otras instituciones fortaleciendo la infraestructura de sus entornos virtuales, para lo cual se deben analizar las particularidades de la infraestructura y los contextos de trabajo de cada lugar.

REFERENCIAS



- [1] P. Matthew, *Virtualization Essentials*, Indianapolis, Indiana: John Wiley & Sons, Inc., 2012.
- [2] T. Robinson, *Building Virtual Machine Labs: A Hands-On Guide*, San Francisco, California, Estados Unidos: Independently published, 2017.
- [3] F. Andina, *Profesional de PC: Claves de Mantenimiento y Reparación*, Buenos Aires, Argentina: Editorial Users, 2013.
- [4] ServerNet, «ServerNet,» [En línea]. Available: <https://servernet.com.ar/tipos-de-gamas-de-computadoras/>. [Último acceso: 23 Septiembre 2025].
- [5] F. G. C. P. M. A. F. P. C. Jesús Carretero Pérez, *Sistemas operativos: una visión aplicada*, Madrid: McGraw-Hill, 2001.
- [6] G. Q. Vieyra, *Sistemas operativos: Una visión aplicada*, Madrid: McGraw-Hill/Interamericana de España, S.A.U., 2001.
- [7] J. E. S. y. R. Nair, *Virtual Machines: Versatile Platforms for Systems and Processes*, San Francisco, CA: Morgan Kaufmann Publishers, 2005.
- [8] M. D. G. Río, *Tecnologías de Virtualización*, Bilbao, España: Ediciones ENI, 2016.
- [9] S. M. Jain, *Linux Containers and Virtualization: A Kernel Perspective*, Nueva York, Estados Unidos: Apress, 2020.
- [10] P. B. Abraham Silberschatz, *Operating System Concepts*, Hoboken, Nueva Jersey, Estados Unidos: John Wiley & Sons, Inc., 2008.
- [11] A. S. Tanenbaum, *Sistemas Operativos Modernos*, Madrid, España: Pearson Educación, 2009.
- [12] E. T. Gómez, *Proxmox. Curso Práctico*, Birmingham, UK: Alfaomega – RAMA, 2024.
- [13] M. Portnoy, *Virtualization Essentials*, Hoboken, Nueva Jersey, Estados Unidos: Sybex, 2023.
- [14] D. Chisnall, *The Definitive Guide to the Xen Hypervisor*, Upper Saddle River, Nueva Jersey, Estados Unidos: Prentice Hall, 2007.
- [15] K. Scarfone, «Guide to Security for Full Virtualization Technologies,» 13 Enero 2011. [En línea]. Available: <https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-125.pdf>. [Último acceso: 12 Marzo 2025].
- [16] K. M. y. S. P. Kane, *Docker: Up and Running*, Sebastopol, California, Estados Unidos: O'Reilly Media, Inc., 2015.
- [17] N. Poulton, *Docker Deep Dive*, Reino Unido: Leanpub, 2017.
- [18] R. B. Christine Bresnahan, *Mastering Linux System Administration*, Indianapolis, Indiana, Estados Unidos: John Wiley & Sons, Inc., 2021.
- [19] U. Brandes, *Proxmox VE Homelab*, s.l.: Selbstverlag, 2023.
- [20] H. Saito, *DevOps with Kubernetes*, Birmingham, Reino Unido: Packt Publishing, 2017.
- [21] S. G. Russ McKendrick, *Mastering Docker. Second Edition*, Birmingham: Packt Publishing Ltd., 2017.
- [22] M. Luksa, *Kubernetes in Action*, Shelter Island, Nueva York, Estados Unidos: Manning Publications, 2017 .
- [23] D. G. Mark Minasi, *Mastering Windows Server 2008 R2*, Indianapolis, Indiana, Estados Unidos: Wiley Publishing, Inc., 2010.

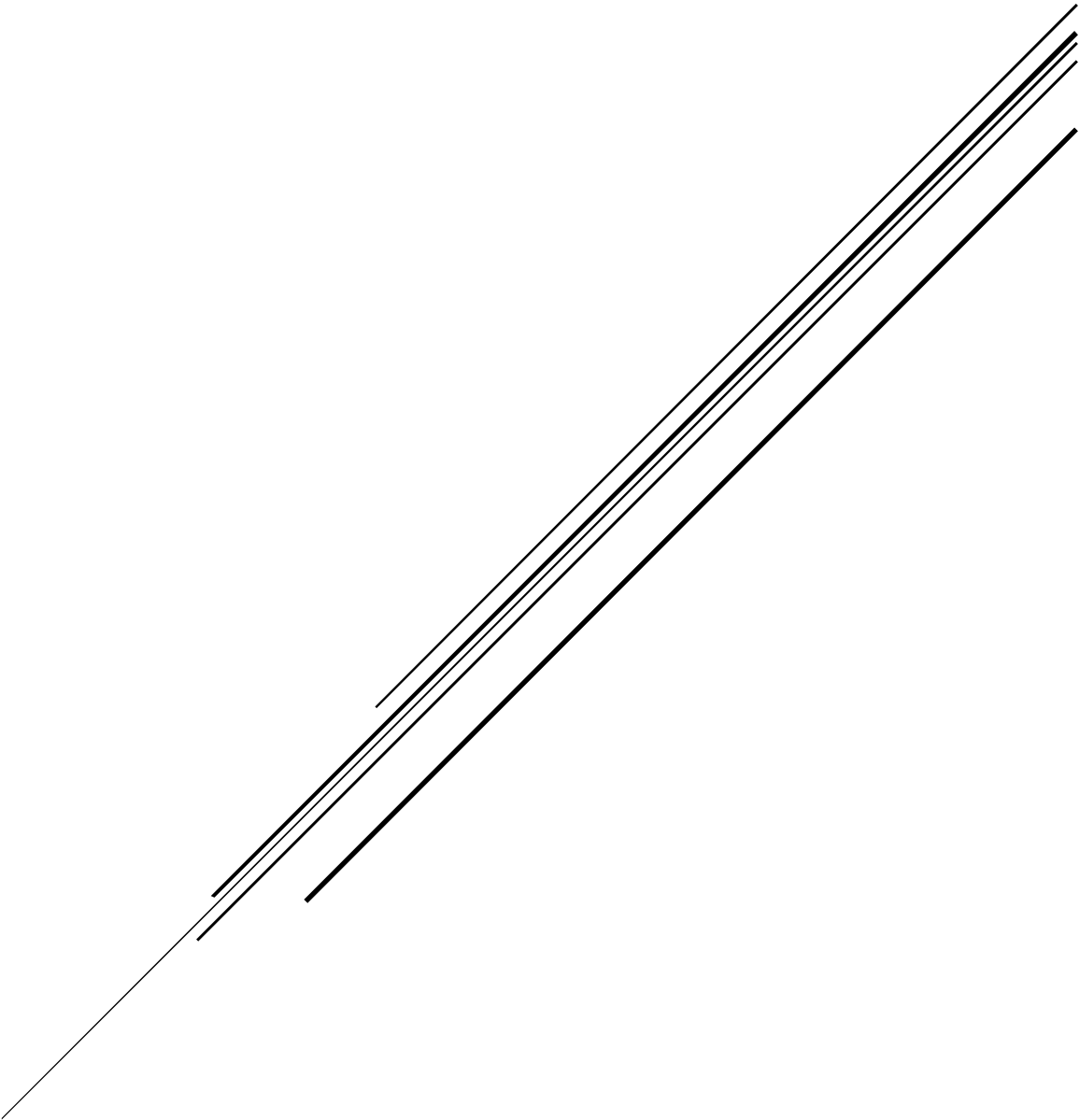
- [24] H. D. Chiramal, *Mastering KVM Virtualization*, Birmingham, Reino Unido: Packt Publishing Ltd., 2016.
- [25] B. J. Blawat, *Mastering Windows PowerShell Scripting*, Birmingham, Reino Unido: Packt Publishing Ltd., 2014.
- [26] S. C. y. S. Hayler, *VirtualBox For Dummies*, Hoboken, Nueva Jersey: John Wiley & Sons, Inc., 2021.
- [27] S. L. F. G. M. L. J. A. Nick Marshall, *Mastering VMware vSphere® 5.5*, Indianapolis, Indiana: John Wiley & Sons, Inc., 2013.
- [28] Bingsin, «Documentación de Proxmox VE en chino,» 2022. [En línea]. Available: https://pve-doc-cn.readthedocs.io/zh-cn/latest/chapter_installation/require.html. [Último acceso: 19 septiembre 2025].
- [29] M. Corporation, «Microsoft Learn,» Microsoft, 5 Agosto 2025. [En línea]. Available: <https://learn.microsoft.com/en-us/windows-server/virtualization/hyper-v/overview>. [Último acceso: 23 Septiembre 2025].
- [30] D. A. P. y. J. L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface*, Cambridge, Massachusetts, Estados Unidos: Morgan Kaufmann, 2017.
- [31] Color Vivo Internet, SLU, «Administración de Sistemas,» Administraciondesistemas.com, 9 Junio 2025. [En línea]. Available: <https://administraciondesistemas.com/evolucion-de-la-version-de-hardware-de-maquina-virtual-en-vmware-comparativa-con-proxmox-ve-y-hyper-v/>. [Último acceso: 24 Septiembre 2025].
- [32] R. S. Boldrito, *Virtualización e hipervisores*, Barcelona, España: Fundació Universitat Oberta de Catalunya, 2022.
- [33] PatoJAD, «¿Qué es Proxmox?,» PatoJAD.com.ar, Noviembre 2019. [En línea]. Available: <https://patojad.com.ar/post/2019/11/que-es-proxmox/>. [Último acceso: 28 Julio 2025].
- [34] «Requisitos Necesarios,» RequisitosNecesarios.com, [En línea]. Available: <https://www.requisitosnecesarios.com/requisitos-para-instalar-virtualbox/>. [Último acceso: 21 Septiembre 2025].
- [35] W. Ahmed, *Mastering Proxmox*, Birmingham, Reino Unido: Packt Publishing, 2014.
- [36] W. Commons, «Wikipedia,» Wikimedia Foundation, [En línea]. Available: https://upload.wikimedia.org/wikipedia/commons/e/ec/Infografia_NucleoLinux.png. [Último acceso: 1 Octubre 2025].
- [37] phoenixNAP, «phoenixNAP IT Glossary,» phoenixNAP, 24 Abril 2024. [En línea]. Available: <https://phoenixnap.mx/glosario/ext4>. [Último acceso: 14 Mayo 2025].
- [38] Vinchin, *EXT4 vs XFS: ¿Cuál es el mejor sistema de archivos para Proxmox?*, US: <https://www.vinchin.com/es/tech-tips/proxmox-ext4-vs-xfs.html>, 2023.
- [39] W. Ahmed, *Proxmox Cookbook*, Birmingham, UK: Packt Publishing Ltd., 2015.
- [40] J. Dordogne, *Redes informáticas: Nociones fundamentales (Protocolos, Arquitecturas, Redes inalámbricas, Virtualización, Seguridad, IP v6 ...)*, París: ENI Ediciones, 2023.
- [41] R. Goldman, *Learning Proxmox VE*, Birmingham, UK: Packt Publishing Ltd., 2016.
- [42] I. NETWORKS, «Máquinas Virtuales,» maquinasvirtuales.net, [En línea]. Available: <https://maquinasvirtuales.net/evaluacion-de-rendimiento-y-ajuste-en-maquinas-virtuales/>. [Último acceso: 21 Septiembre 2025].

- [43] Editorial Users, Microprocesadores (fascículo 07 de la colección Curso Visual y Práctico Técnico PC: Mantenimiento y Reparación), Buenos Aires, Argentina: Editorial Users, 2013.
- [44] K. Technology, «Kingston Technology Latinoamérica,» Kingston Technology Company, Inc., Febrero 2024. [En línea]. Available: <https://www.kingston.com/latam/blog/pc-performance/memory-assessor>. [Último acceso: 21 Septiembre 2025].
- [45] «ServerNet,» ServerNet Argentina, [En línea]. Available: <https://servernet.com.ar/porque-mi-pc-ocupa-toda-la-memoria-ram>. [Último acceso: 20 Septiembre 2025].
- [46] Crucial, «Crucial.es,» Micron Technology, Inc., Octubre 2023. [En línea]. Available: <https://www.crucial.es/articulos/about-memory/how-to-check-ram-windows-10>. [Último acceso: 20 Septiembre 2025].
- [47] K. Brown, «LinuxConfig.org,» 15 Mayo 2020. [En línea]. Available: <https://linuxconfig.org/ubuntu-20-04-system-requirements>. [Último acceso: 20 Septiembre 2025].
- [48] E. e. d. IONOS, «Digital Guide,» IONOS México, 13 Octubre 2023. [En línea]. Available: <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/instalar-mysql-en-ubuntu-2004/>. [Último acceso: 20 Septiembre 2025].
- [49] phoenixNAP, «phoenixNAP Global IT Services,» phoenixNAP LLC, 6 octubre 2021. [En línea]. Available: <https://phoenixnap.mx/glosario/tasa-de-transferencia>. [Último acceso: 29 septiembre 2025].
- [50] lavariega.com, «lavariega.com,» Andrés Lavariega, 17 abril 2025. [En línea]. Available: <https://lavariega.com/qos-en-voip-como-optimizar-la-calidad-de-tus-llamadas/>. [Último acceso: 28 septiembre 2025].
- [51] A. v. Lieshout, «Leaseweb Blog,» 10 enero 2023. [En línea]. Available: <https://blog.leaseweb.com/2023/01/10/performance-metrics-in-vmware>. [Último acceso: 20 Septiembre 2025].
- [52] I. Ros, «MuyComputer,» TPNNet (Total Publishing Network), 25 Enero 2024. [En línea]. Available: <https://www.muycomputer.com/2024/01/25/procesadores-intel-core-diferencias-elegir/>. [Último acceso: 22 Septiembre 2025].
- [53] G. S. Evi Nemeth, UNIX and Linux System Administration Handbook, Upper Saddle River, Nueva Jersey, Estados Unidos: Pearson Education, 2017.
- [54] G. S. T. R. H. Evi Nemeth, Linux Administration Handbook, Upper Saddle River, New Jersey: Prentice Hall PTR, 2006 (segunda edición).
- [55] J. M. H. Moya, Manual de Telecomunicaciones, México, D.F.: S/N, 2004.
- [56] J. M. Caballero, Redes de banda ancha, Barcelona: Marcombo Boixareu Editores, 1997.
- [57] D. J. W. Andrew S. Tanenbaum, Redes de computadoras (quinta edición), México, D.F.: Pearson Educación, 2012.
- [58] iPerf, «iPerf - The ultimate speed test tool for TCP, UDP and SCTP,» 28 Mayo 2025. [En línea]. Available: <https://iperf.fr/>. [Último acceso: 28 Mayo 2025].
- [59] W. Foundation, «Wireshark,» Wireshark Foundation, 28 Mayo 2025. [En línea]. Available: <https://www.wireshark.org/>. [Último acceso: 28 Mayo 2025].

- [60] P. VE, «Proxmox Virtual Environment Documentation,» Proxmox Server Solutions GmbH, 21 Mayo 2025. [En línea]. Available: https://pve.proxmox.com/pve-docs/chapter-pvesdn.html#pvesdn_config_dhcp. [Último acceso: 21 Mayo 2025].
- [61] G. N. Arribas, Sistemas de cortafuegos, Barcelona : FUOC (Fundació per a la Universitat Oberta de Catalunya), S/N.
- [62] W. Ahmed, Mastering Proxmox, Birmingham, Reino Unido: Packt Publishing Ltd., 2017.
- [63] S. d. Luz, «RedesZone.net,» ADSLZone Group, 8 Enero 2025. [En línea]. Available: <https://www.redeszone.net/tutoriales/seguridad/nftables-firewall-linux-configuracion>. [Último acceso: 24 Agosto 2025].
- [64] G. S. Evi Nemeth, Linux Administration Handbook 5nd ed, NJ: Addison-Wesley Professional., 2006.
- [65] J. L. H. y. D. A. Patterson, Computer Architecture: A Quantitative Approach, Waltham, Massachusetts, EE. UU.: Morgan Kaufmann Publishers, 2012.
- [66] G. d. T. d. Contenedores, Libro blanco sobre tecnología de contenedores y sus aplicaciones [V1.0], Beijing, China: Alianza de Nube de Código Abierto de China, 2016.
- [67] J. B. K. H. Brendan Burns, Kubernetes: Up & Running: Dive into the Future of Infrastructure, California, Estados Unidos: O'Reilly Media, Inc., 2019.
- [68] S. v. Vugt, Red Hat RHCSA/RHCE 7 Cert Guide: Red Hat Enterprise Linux 7 (EX200 and EX300), Indianapolis, Indiana, Estados Unidos: Pearson IT Certification, 2016.
- [69] B. B. J. B. Kelsey Hightower, Kubernetes: Up and Running: Dive into the Future of Infrastructure, United States of America.: Angela Rufino, 2017.
- [70] N. Poulton, The Kubernetes Book, Birmingham, Reino Unido: Packt Publishing, 2021 .
- [71] A. Hohn, The Book of Kubernetes: A Complete Guide to Container Orchestration, San Francisco, California, Estados Unidos: No Starch Press, 2022.
- [72] Vinchin, «EXT4 vs XFS: ¿Cuál es el mejor sistema de archivos para Proxmox?,» Vinchin, 1 Noviembre 2024. [En línea]. Available: <https://www.vinchin.com/es/tech-tips/proxmox-ext4-vs-xfs.html>. [Último acceso: 30 Marzo 2025].
- [73] iBoysoft, «Sistema de archivos XFS: Definición, características, XFS vs. EXT4 vs. BTRFS,» iBoysoft, 3 Diciembre 2024. [En línea]. Available: <https://iboysoft.com/es/wiki/sistema-de-archivos-xfs.html>. [Último acceso: 14 Mayo 2025].
- [74] C. Melantuche, «Virtualisierung – Proxmox: Lokaler Speicher,» NoSoloHacking , 10 Agosto 2022. [En línea]. Available: <https://www.nosolohacking.info/proxmox-almacenamiento-local/>. [Último acceso: 21 Mayo 2025].
- [75] IEEE, «IEEE Standard for Ethernet,» 2018. [En línea]. Available: https://standards.ieee.org/standard/802_3-2018.html. [Último acceso: 07 08 2025].
- [76] B. Forta, MySQL® Crash Course, Indianapolis, IN: sams Publishing, 2006.
- [77] Microsoft, «Soporte técnico de Microsoft,» 14 Octubre 2025. [En línea]. Available: <https://support.microsoft.com/es-es/windows/windows-10-soporte-t%C3%A9cnico-finaliza-el-14-de-octubre-de-2025-2ca8b313-1946-43d3-b55c-2b95b107f281>. [Último acceso: 19 Septiembre 2025].

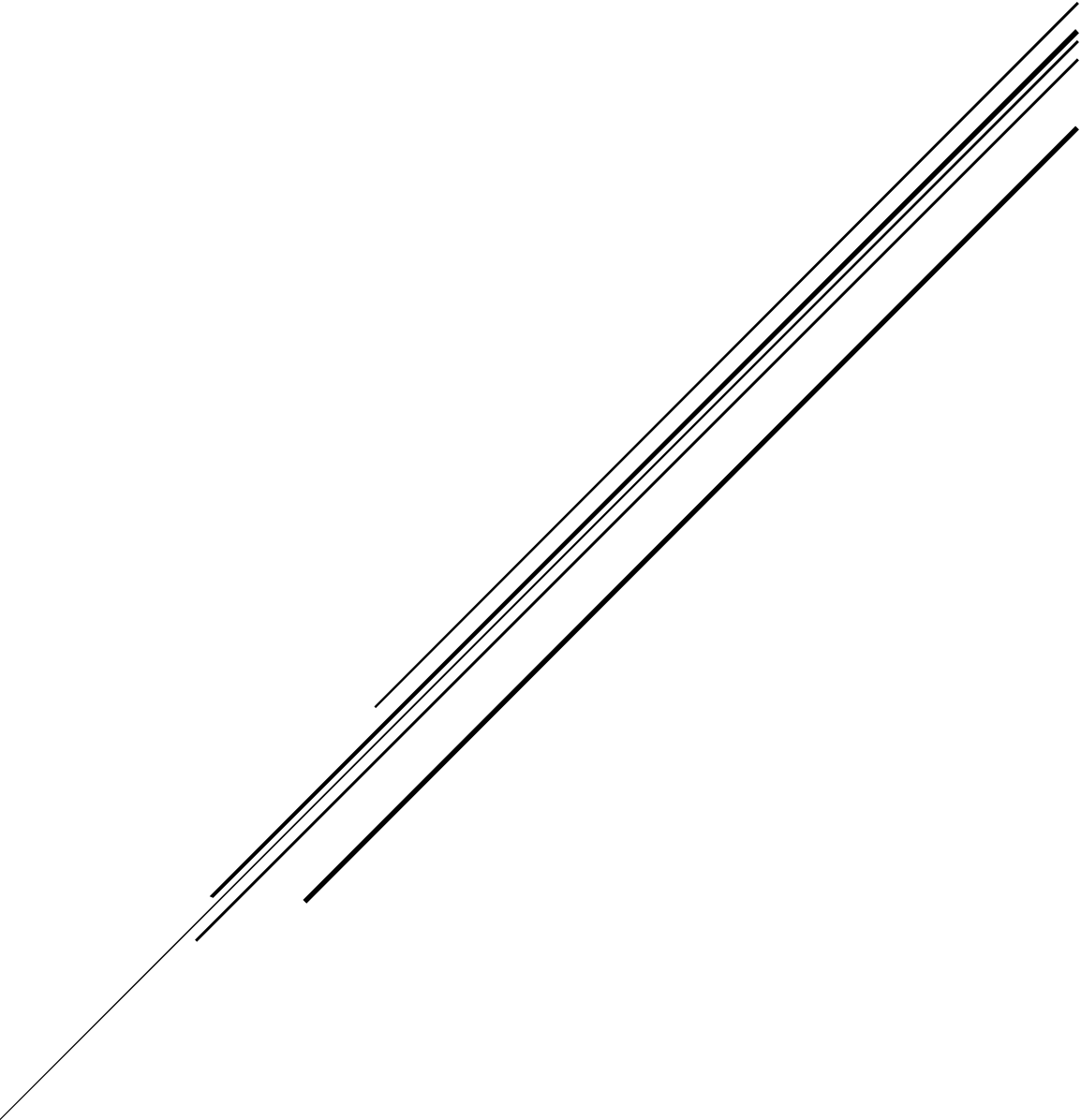
- [78] Isaac, «Guía Hardware,» 12 Agosto 2025. [En línea]. Available: <https://www.guiahardware.es/todas-las-formulas-y-metodos-para-calcular-el-rendimiento-de-una-cpu/>. [Último acceso: 20 Septiembre 2025].
- [79] S. Completo, «Soporte Completo en Informática,» [En línea]. Available: <https://soportecompleto.com/index.php/blog/consejos-informaticos/que-son-las-gamas-de-computadoras-y-cual-elegir>. [Último acceso: 23 Septiembre 2025].
- [80] J. Izquierdo, «Weblinus,» Weblinus.com, 4 Octubre 2022. [En línea]. Available: <https://weblinus.com/comprimir-ficheros-con-zstandard-en-linux/>. [Último acceso: 24 Septiembre 2025].
- [81] Copyright 2000-2025 FILExt, «FILExt,» FILExt.com, 2023. [En línea]. Available: <https://filext.com/es/extension-de-archivo/LZO>. [Último acceso: 24 Septiembre 2025].
- [82] Telradio, «Telradio,» 2013. [En línea]. Available: <https://www.telradio.com.mx/preguntas/que-es-asterisk>. [Último acceso: 3 Octubre 2025].
- [83] B. A. Forouzan, Conjunto de protocolos TCP/IP (cuarta edición), Nueva York, NY: McGraw-Hill, 2010.
- [84] J. T. G. C. Robert Eckstein, Using Samba (3rd Edition), Sebastopol, California: O'Reilly Media, 2007.
- [85] contenido institucional de VoIPstudio, «VoIPstudio,» VoIPstudio Ltd., 2022. [En línea]. Available: <https://voipstudio.mx/blog/que-es-voip-la-guia-completa/>. [Último acceso: 3 Octubre 2025].
- [86] IBM Cloud Docs, «IBM Cloud Docs,» Corporación Internacional de Máquinas de Negocios (IBM), 19 Abril 2024. [En línea]. Available: <https://cloud.ibm.com/docs/virtualization?topic=virtualization-what-is-citrix-xenserver&locale=es>. [Último acceso: 3 Octubre 2025].
- [87] R. Hat, «Red Hat,» Red Hat, Inc., 11 Mayo 2022. [En línea]. Available: <https://www.redhat.com/es/topics/virtualization/what-is-KVM>. [Último acceso: 3 Octubre 2025].
- [88] E. Sandoval, «EBAC México,» EBAC (Escuela Británica de Artes Creativas y Tecnología), 21 Septiembre 2023. [En línea]. Available: <https://ebac.mx/blog/frameworks>. [Último acceso: 8 Octubre 2025].
- [89] R. Bigler, XenDesktop & XenApp 7.6: Citrix Planung–Implementierung–Betrieb, Norderstedt, Alemania: Books on Demand GmbH, 2015.
- [90] R. PcHardwarePro, «PcHardwarePro,» PcHardwarePro, 5 Octubre 2023. [En línea]. Available: <https://www.pchardwarepro.com/que-es-parallels-y-para-que-sirve/>. [Último acceso: 8 Octubre 2025].
- [91] J. E. Pérez, Introducción a JavaScript, Sevilla: S/N, 2008.

SIGLAS Y ACRÓNIMOS



BIOS	Sistema Básico de Entrada/Salida (Basic Input/Output System)
CPU	Unidad Central de Procesamiento (Central Processing Unit)
ESX	Elastic Sky X (Hipervisor de VMware)
Hyper-V	Hipervisor de Microsoft
VLAN	Red de Área Local Virtual (Virtual Local Area Network)
VM	Máquina Virtual (Virtual Machine)
Xen	Hipervisor de código abierto
ZFS	Sistema de Archivos Zettabyte (Zettabyte File System)
OS	Sistema Operativo (Operating System)
HW	<i>Hardware</i>
SW	<i>Software</i>
ISO	Imagen de disco estándar
GUI	Interfaz Gráfica de Usuario (Graphical User Interface)
CLI	Interfaz de Línea de Comandos (Command Line Interface)
E/S	Entrada/Salida
KVM	Máquina Virtual basada en Kernel (Kernel based Virtual Machine)
QEMU	Emulador rápido: proporciona API para KVM
API	Interfaz de programación de aplicación
LXC	Linux Containers
VMM	Monitor de Máquina Virtual (Virtual Machine Monitor)
OpenVZ	Tecnología de virtualización de contenedores para sistemas Linux
VMware	Empresa líder en virtualización
Citrix Xen	Hipervisor basado en Xen
Proxmox VE	Entorno de Virtualización Proxmox Virtual Environment (VE).
vzdump	Herramienta de respaldo de Proxmox
PAM	(Módulos de autenticación conectables) Servidor DHCP para una zona que asigna automáticamente
DHCP	direcciones IP. parte del sistema operativo que interacciona directamente con el
<i>Núcleo</i>	<i>hardware.</i>
top	Resumen interactivo y actualizado periódicamente de los procesos
htop	un visualizador de procesos interactivo
JSON	formato de texto para almacenar e intercambiar datos
REST	Transferencia de Estado Representacional

APÉNDICES



Apéndice A

Descarga e instalación de Proxmox VE: para empezar con la instalación de Proxmox VE es fundamental asegurarse de que la opción de Máquina Virtual esté habilitada en el BIOS del equipo para evitar problemas posteriores. Para ingresar al BIOS, se presionó “F10”, esto depende mucho de la marca cada computadora, ahora se debe navegar a: “seguridad”, después “seguridad de dispositivos”, y al final “Tecnología de Virtualización”, para habilitar la opción correspondiente. Ahora se requiere de una memoria “USB” y un programa, se puede utilizar “Rufus”, así como la imagen “ISO “.Después desde su página oficial de Proxmox VE, se debe presionar Instalar, el sistema procederá a asignar las unidades y extraer los archivos necesarios, una vez dentro se verifico las versiones disponibles que presenta su página oficial “(<https://proxmox.com/en/downloads>)”, “formato ISO”. Hacer clic en el botón “Download” para iniciar la descarga, el sistema debe arrancar desde el medio de instalación (USB). Esto implica seleccionar el dispositivo de arranque en las opciones iniciales del equipo. Una vez que el instalador de Proxmox VE, se debe seleccionar “Instalar Proxmox VE”. [Figura 5.1]. El sistema cargará la interfaz gráfica y mostrará el CLUF (Contrato de Licencia de Usuario Final). Hacer clic en "Aceptar" y se seleccionó el disco de instalación con el botón "Opciones “Posteriormente aparece “Contrato de licencia” en este punto, se hizo clic en “aceptar los términos de acuerdo a la licencia”, luego se ajustaron parámetros para personalizar el almacenamiento, las que salen disponibles son “Swapsize”, “maxroot”, “minfree” y “maxvz”, si su *hardware* lo permite, se recomienda configurar sistemas con “ZFS” o “BTRFS”, en este apartado se debe especificar “país”, “zona horaria” y “teclado”. Esta información es recomendable para la sincronización del clúster. A continuación, se ingresa el usuario para este caso “root” y su contraseña luego se agrega un correo electrónico para que el administrador reciba notificaciones. Después se configura las “direcciones IP”, “puerta de enlace” y “DNS” para conectarse a internet. Y luego, al terminal todo lo configurado se inició el proceso para presentar una introducción breve sobre

Proxmox VE, el sistema se reiniciar hay que tomar en cuenta la dirección IP para conectarse desde el cualquier navegador web. [Figura 5.2], una vez finalizado el proceso anterior lo que se debe hacer para acceder primero se presenta un mensaje confirmando la instalación exitosa. Se debe iniciar el sistema y abrir el navegador web “<https://192.168.0.127:8006>”, el enlace permite acceder a la interfaz *web* de Proxmox VE

Carga de imagen ISO: para realizar la carga de imágenes de distintos sistemas operativos primero se debe hacer clic en “Cargar ISO” luego se seleccionó el archivo desde el equipo local se buscó y se subió, después se seleccionó el nodo correspondiente desde el panel izquierdo (“pve”) , después se accedió al “almacenamiento local” y luego se dirigió a la pestaña “Imágenes ISO”, se hizo clic en “Subir” y, en la ventana emergente, se eligió la imagen que se utilizó, la ISO aparece en la lista de archivos disponibles dentro del almacenamiento. [Figura 5.3].

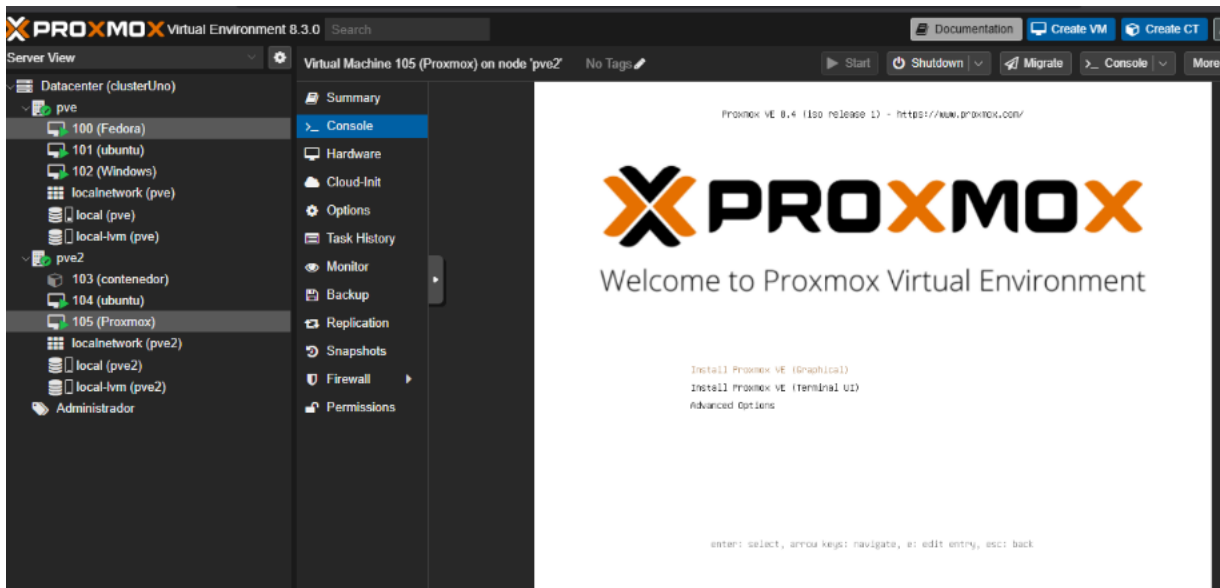


Figura 5.1: Pantalla de Inicio antes de Instalar de Proxmox VE. [Captura Propia]

```

login as: root
root@192.168.0.127's password:
Linux pve 6.8.12-4-pve #1 SMP PREEMPT_DYNAMIC PMX 6.8.12-4 (2024-11-06T15:04Z) x
86_64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Aug 19 18:28:21 2025
root@pve:~# █

```

Figura 5.2: Consola de Proxmox VE. [Captura Propia]

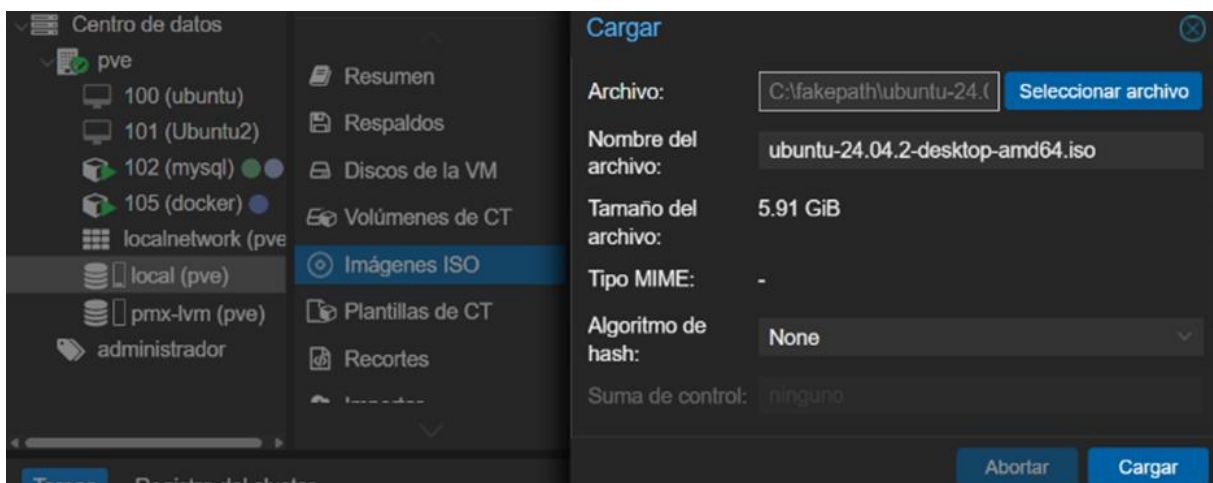


Figura 5.3: Subir la primera Imagen ISO a Proxmox. [Captura Propia]

Para crear un usuario en la interfaz de Proxmox VE se anexa el campo Nombre de usuario mientras que en el dominio de autenticación se selecciona Proxmox VE autenticador, debido a que permite administrar credenciales dentro del sistema. Además, se establece una contraseña Finalmente, se debe presionar el botón “Agregar” para que el sistema registre al usuario correspondiente.

Para crea un grupo se debe ingresar dentro del sistema —“Administradores”—. Luego, en el cuadro de diálogo de configuración, se introduce el nombre del grupo y se deja el campo de comentario vacío. Al presionar el botón “Crear”, el sistema registra el grupo dentro de la plataforma para la asignación de usuarios y permisos, después viene la creación de conjuntos (*pools*), a través de los cuales se define qué equipos puede visualizar el usuario y con qué

nivel de acceso interactuará. A continuación, se procede a la creación de un conjunto — “administrador”—, La Figura 5.4, se observa que cada usuario (“root”, “uacm1”) cada uno asociado a un dominio (“pam”, “pve”) y se señala si está activado, su caducidad y si se requiere sin autenticación en dos pasos Proxmox VE permite administrar la autenticación en Linux y Unix con diferentes métodos como “contraseñas”, “biometría” y “tarjetas inteligentes”.

Nombre de usuario ↑	Dominio ↑	Activado	Caducar	Nombre	TFA	Grupos
root	pam	Sí	nunca		No	administradores
uacm1	pve	Sí	nunca	admin	Si	
uacm2	pve	Sí	nunca	Admin	No	administradores

Figura 5.4: Ventana de usuarios. en Proxmox VE. [Fuente: Captura Propia]

Para realiza un grupo se debe hacer clic en crear el grupo “Administradores” para el usuario “uacm1”, para que se encuentre en el *pool* de administrador, primero se visualizó la máquina virtual correspondiente. Luego, se accedió al apartado de –“Administración”, luego se ingresó a la sección “Miembros” y, después, se agregó la máquina virtual que se deseó asignar, en este caso, se ha seleccionado la máquina “ID 100”, ubicada en el nodo “pve”, se configuraron los permisos para que todos los usuarios pertenecientes a este grupo puedan acceder. No obstante, también es posible asignar permisos específicos a determinados usuarios. Para la visualización de recursos, al momento de realizar esta asignación, sólo se podrá visualizar la máquina –“ID 100”, con restricciones que le impedirán administrar el entorno de Proxmox VE en su totalidad. Finalmente, el cuadro de diálogo para agregaron permisos de usuario. Se seleccionó –“uacm1@pve” y se le asignó el rol de Administrador, para que se pueda otorgar los permisos definidos para la administración del sistema. [Figura 5.5].

Agregar: Máquina virtual ✕

<input type="checkbox"/>	ID ↑	Nodo	Conjunto a...	Estado	Nombre	Tipo
<input checked="" type="checkbox"/>	100	pve		parado	Fedora	qemu
<input type="checkbox"/>	101	pve		parado	ubuntu	qemu
<input type="checkbox"/>	102	pve		parado	Windows	qemu
<input checked="" type="checkbox"/>	103	pve2		parado	contenedor	lxc
<input type="checkbox"/>	104	pve2		ejecutándose	ubuntu	qemu

Figura 5.5: Ventana Administración para visualizar MV. [Fuente: Captura Propia]

Para realizar la autenticación de 2 factores se debe acceder a la configuración del usuario – “root”, donde se habilita la opción –“TFA (Two Factor Authentication)”. Luego, se debe generar un código –“QR” que debe escanearse con la aplicación de autenticación, esta aplicación se puede descargar desde la tienda de –“Play Store” con el nombre de –“Google Authenticator”. Después de escanear el código, la aplicación proporciona un código de verificación que se debe ingresar en Proxmox VE para completar la configuración. Además, es recomendable renombrar cada instancia de autenticación para identificar correctamente los accesos en un clúster con múltiples nodos. En la Figura 5.6 muestra como ejemplo para habilitar la autenticación en dos pasos donde se adjunta un código QR generado por la aplicación.

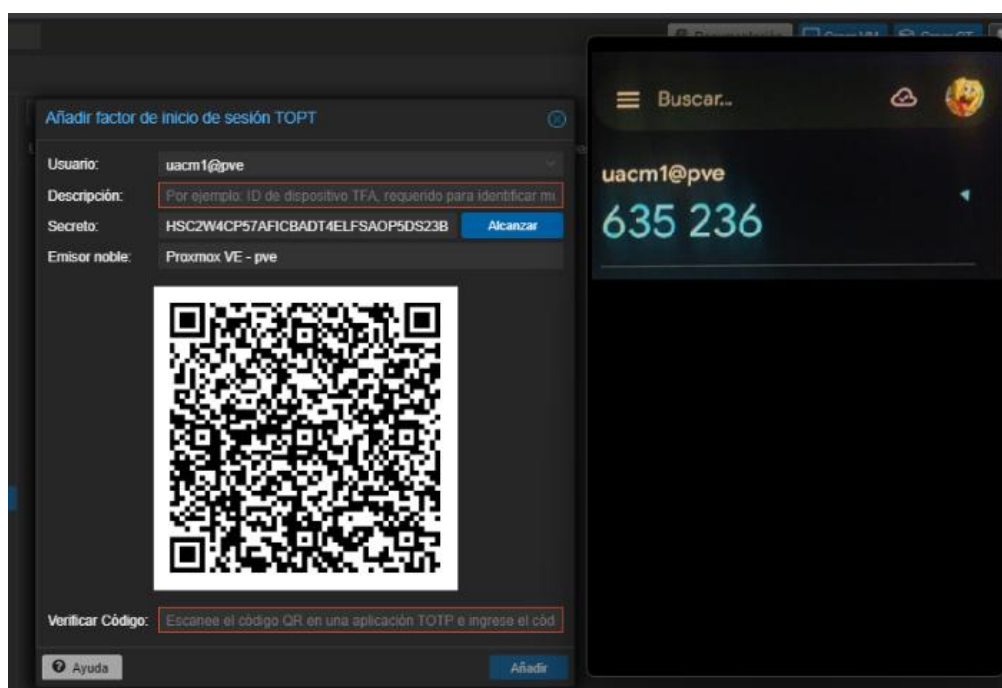


Figura 5.6: Ventana de Google Authenticator QR. [Captura Propia]

Como ejemplo para realizar la ejecución del *backup*, se debe hacer clic en –“Iniciar”, para monitorear el progreso desde la pestaña de tareas. [Figura 5.7], Para respaldar los contenedores, se debe seguir el mismo proceso que con las MV.

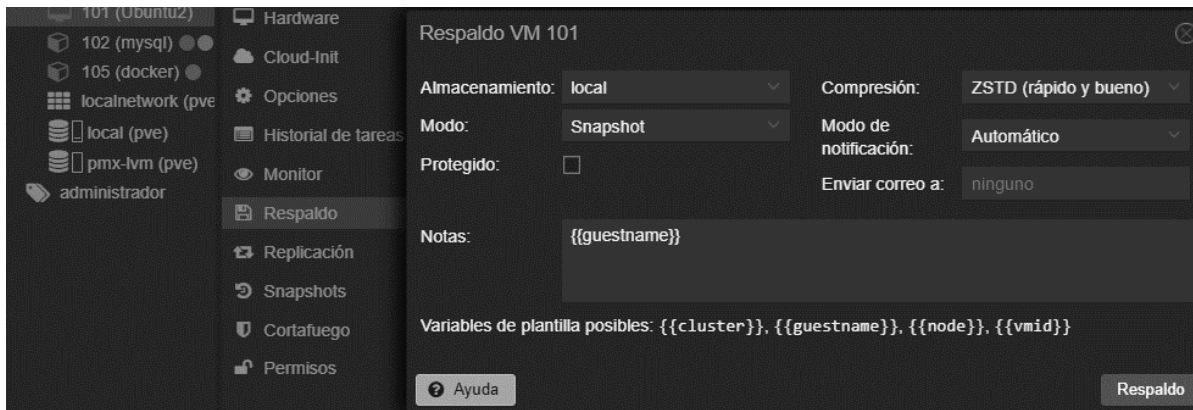


Figura 5.7: Ventana de *Backup* de la MV. [Fuente: Captura Propia]

Apéndice B

Administración del almacenamiento: El modelo que se ocupa en Proxmox VE para almacenamiento es muy dócil. Las imágenes de las máquinas virtuales se pueden almacenar en distintos almacenamientos locales o en almacenamiento como NFS o iSCSI (NAS, SAN). No hay límites y se pueden configurar tantos grupos de almacenamiento como se desee. Se puede usar todas las tecnologías de almacenamiento disponibles para Linux Debian. Al respecto, “Una de las principales ventajas de almacenar máquinas virtuales en almacenamiento compartido es la capacidad de migración en vivo de máquinas en ejecución sin ningún tiempo de inactividad” [60]; es decir que para realizar una migración en vivo no se requiere una copia de información debido a que los nodos dentro del clúster tienen acceso directo a la imagen de la MV. Cuando el sistema inició, hubo que configurar las particiones, ya que por defecto se proporciona un cierto espacio, pero, para tener un mejor uso al momento de almacenar la información en las máquinas virtuales, hubo que borrar la partición y crear uno o distintos almacenamientos. Esto, a fin de tener un mejor control al momento que administrar las cargas dentro del espacio de trabajo.

Algunas configuraciones relevantes del almacenamiento en Proxmox VE son:

- Almacenamiento local: Primero se debe acceder al centro de datos y hacer clic en almacenamiento, luego seleccione *local-lvm* y elimínelo; abra la consola y ejecute el comando `lvremove /dev/pve/data`. presione `[y]` para confirmar la eliminación. A continuación, ejecute el comando `lvresize -l +100%free /dev/pve/root`. Para cambiar el tamaño del disco, seguido de `resize2fs /dev/mapper/pve-root` para cambiar el tamaño del archivo. por último, hacer clic en “editar” en el almacenamiento local para asignar todo el contenido.
- Almacenamiento LVM con dispositivos locales: A través de SSH, Ahora revisar y crear una partición, ejecutar el comando `# cfdisk /dev/sda` y después guarde los cambios. [Figura 6.1]. Ahora se ejecuta el comando `# pvcreate /dev/sda(id)` para crear un nuevo volumen y crear un grupo de volúmenes con el nombre `pmx-lvmvg` o cualquier otro deseado, ejecutar `# vgcreate pmx-lvmvg /dev/sda(id)` [Figura 6.2]. Para incluirlo se debe acceder al menú “Almacenamiento”, hacer clic en 'Agregar' y seleccionar 'LVM', Introducir “`pmx-lvm`” como nombre del almacenamiento y llenar los demás campos

```
Label: gpt, identifier: D43768B3-8653-4D7D-8AA1-3CFDF9E05384
```

Device	Start	End	Sectors	Size	Type
/dev/sda1	2048	175781887	175779840	83.8G	Linux filesystem
/dev/sda2	175781888	207032319	31250432	14.9G	Linux swap
/dev/sda3	207032320	246093823	39061504	18.6G	Linux filesystem
/dev/sda4	246093824	832030719	585936896	279.4G	Linux filesystem
Free space	832030720	976773134	144742415	69G	

Figura 6.1: Ventana de consola `#pvcreate /dev/sda(id)`. [Captura Propia]

```
root@pve:~# pvcreate /dev/sda5
Physical volume "/dev/sda5" successfully created.
root@pve:~# vgcreate pmx-lvmvg /dev/sda5
Volume group "pmx-lvmvg" successfully created
root@pve:~# █
```

Figura 6.2: Ventana de consola crear volumen. [Captura Propia]

- Almacenamiento ZFS: Primero se debe acceder a la consola del nodo e inserte disco externo al nodo que se va ocupar, para este ejemplo será 'sdb1'. [Figura 6.3]. Insertar el comando `'lsblk'` y `'fdisk -l'` en la consola para consultar la partición. Crear el pool ZFS

utilizando el comando: `# zpool create -f zfspool1 /dev/sdb1` y verificar la creación del pool con el comando: `# zpool list` [Figura 6.4]. El comando `zpool list` mostrará los detalles del almacenamiento ZFS disponible. Finalmente, dirigirse a la sección de 'Almacenamiento' y seleccionar la opción 'Agregar ZFS'. Luego, completar todos los campos y hacer clic en "Agregar".

Dispositivo	Tipo	Almacenamiento - ...	Tamaño	GPT	Modelo
/dev/sda	Hard Disk	partitions	1.00 TB	Sí	WDC_WD10SPZX-21
/dev/sda1	partition	BIOS boot	1.03 MB	Sí	
/dev/sda2	partition	EFI	1.07 GB	Sí	
/dev/sda3	partition	LVM	999.13 GB	Sí	
/dev/sdb	unknown	partitions	2.00 GB	No	JD FireFly
/dev/sdb1	partition	No	2.00 GB	No	

Figura 6.3: Ventana de Almacenamiento. [Captura Propia]

```

root@pve2:~# zpool create -f zfspool1 /dev/sdb1
root@pve2:~# zpool list
NAME      SIZE  ALLOC   FREE CKPOINT  EXPANDSZ   FRAG    CAP  DEDUP  HEALTH  ALTROOT
zfspool1 1.75G  108K   1.75G      -         -         0%    0%  1.00x  ONLINE  -
root@pve2:~#
    
```

Figura 6.4: Ventana de consola creación de zpool. [Captura Propia]

- Almacenamiento de Directorios: Se debe acceder al “Centro de Datos”, dirigirse a la sección del nodo, hacer clic en “Directorio” y seleccionar la opción “Crear: Directorio”. En suma, completar los campos correspondientes. [Figura 6.5].

Crear: Directorio

Disco: /dev/sda

Sistema de archivos: ext4

Nombre: Directorio1

Agregar almacenamiento:

Ayuda Crear

Figura 6.5: Ventana de directorio. [Captura Propia]

- Almacenamiento de Descarga y Gestión de Plantillas: Ahmed nos menciona que “la administración en Proxmox VE permite administrar el almacenamiento de plantillas en

distintos nodos” [39], Se debe ingresar al 'centro de datos', 'clusterUno' y luego al nodo 'pve2' Finalmente, seleccionar la opción Plantillas CT y hacer clic en 'Plantillas' para descargar. Para elegir la descarga de cualquier plantilla hay que considerar que cada nodo almacena las descargas de forma independiente, esto significa que una plantilla descargada en un nodo no se encuentra automáticamente disponible en los demás nodos. La administración del clúster de cada nodo administra sus propias plantillas y las usa para la creación de contenedores.

Apéndice C

Para configura la interfaz de red, estableciendo su nombre y tipo, se puede asignar una dirección IP estática o definir un modo “DHCP”, al final se establece manualmente un “servidor DNS” público o privado, para permitir una resolución de nombres de dominio dentro del contenedor, se revisa toda la configuración antes de proceder con la creación. Al confirmar estos datos, se inicia la generación del contenedor para verificar la conectividad del clúster mediante un envío de Ping entre los nodos dentro de la misma LAN para que se integre correctamente a la infraestructura virtual. Para crear una máquina virtual como ejemplo, se debe acceder a la interfaz web y hacer clic en la opción Crear VM. y después llenar los campos de la MV. [Figura 7.1]. Se elige el tipo de OS que se instalará. Se puede seleccionar “Windows”, “Linux”, u otro dependiendo del requerimiento. Además, se define la versión del “kernel” si se trata de una distribución de *Linux*. Se debe cargar una imagen ISO previamente al almacenamiento. Luego, en la configuración de la máquina virtual, se selecciona la imagen para proceder con la instalación, se recomienda seleccionar un tamaño acorde a las necesidades del sistema, puede ser de “50 GiB”, “100 GiB” o más, después e configura la cantidad de *núcleos* para la máquina virtual, se debe asignar suficiente procesamiento para sus aplicaciones. se recomienda definir un número adecuado dependiendo de la carga de trabajo como ejemplo se adjuntó 2 *núcleos*. se seleccionó la cantidad mínima y máxima que

podrá utilizar la máquina virtual. Se recomienda definir un mínimo de 1 GiB y un máximo de 2 GiB. A continuación, se define la interfaz de red y el tipo de dirección IP. Se puede configurar una IP estática o utilizar “DHCP” para asignar automáticamente. [Figura 7.2]. Antes de finalizar el proceso, se presenta un resumen de todas las configuraciones establecidas, si todo está correcto hacer clic en “confirmar”.



Figura 7.1: Ventana para crear la Máquina virtual. [Fuente: Captura Propia]

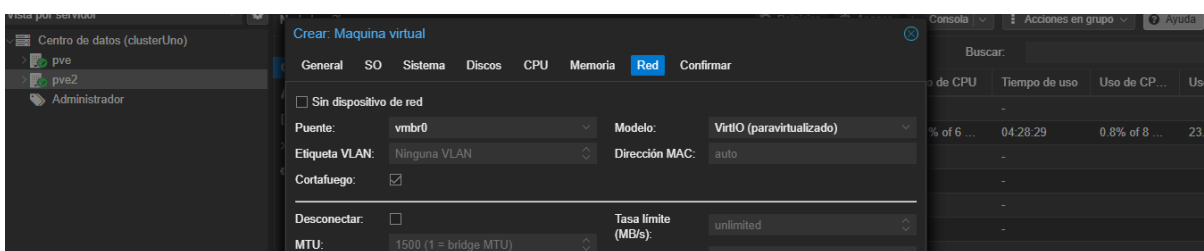


Figura 7.2: Ventana de red. [Fuente: Captura Propia]

Para ingresar al centro de datos Primero debe ir en la barra lateral izquierda, hacer clic en la opción “Centro de datos”, aquí se presentan los indicadores de uso de “CPU (1% de 16 cores)”, “Memoria (32%: 4.92 GiB de 15.17 GiB)” y “Almacenamiento (4%: 74.49 GiB de 1.74 TiB)”. “Pve” tiene un consumo de CPU del 1% y memoria del 47%, por otro lado, la “pve2” mantiene un uso de CPU del 0% y memoria del 18%. [Figura 7.3].

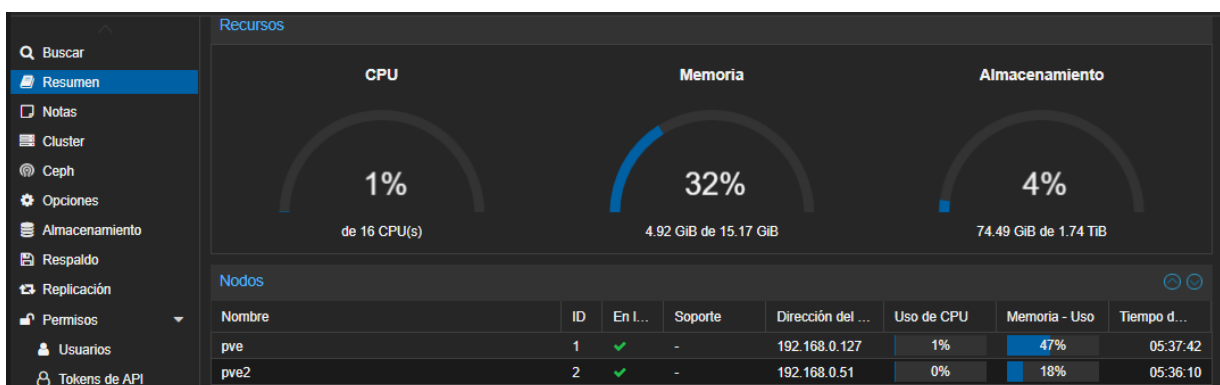


Figura 7.3: Ventana de Resumen de recursos. [Fuente: Captura Propia]

“La configuración de red en Proxmox VE se puede realizar mediante la interfaz gráfica de usuario (GUI) o la línea de comandos (CLI)” [39]. En lo que concierne a la activación de los

cambios, las modificaciones en la GUI requieren un reinicio, mientras que los cambios en la CLI se aplican en tiempo real.

En la Figura 7.4 se presenta un diagrama de la estructura de la red que conecta a las máquinas virtuales consistente; una computadora principal o Administrador con dirección IP 192.168.0.128, que es la estación de trabajo para conectarse y administrar la interfaz *web* de Proxmox VE; un servidor 1 con nombre “Proxmox pve” y dirección IP 192.168.0.127, que es el nodo principal y cuenta con dos máquinas virtuales, la primera con dirección IP 192.168.0.224 y la segunda con dirección IP 192.168.0.179, ambas con sistema operativo Ubuntu 20.04; y un servidor 2 con el nombre “Proxmox pve2” con dirección IP 192.168.0.51, que es el segundo nodo de virtualización, con las direcciones IP 192.168.0.246 con sistema operativo Windows 10 y 192.168.0.64 con sistema operativo Ubuntu 20.04.

La configuración se realiza principalmente en el archivo `/etc/network/interfaces` o mediante la interfaz *web* de Proxmox VE. “El complemento DHCP en Proxmox VE SDN (redes definidas por *software*) se puede utilizar para implementar automáticamente un servidor DHCP para una zona” [60]. En otras palabras, proporciona el DHCP a todas las subredes de una zona que tienen un alcance DHCP configurado. Para este trabajo se realiza la implementación de una red de área local (LAN) con topología estrella. A cada equipo y máquina virtual se le asigna una IP privada de forma automática por medio de DHCP, que es distribuido por el módem.

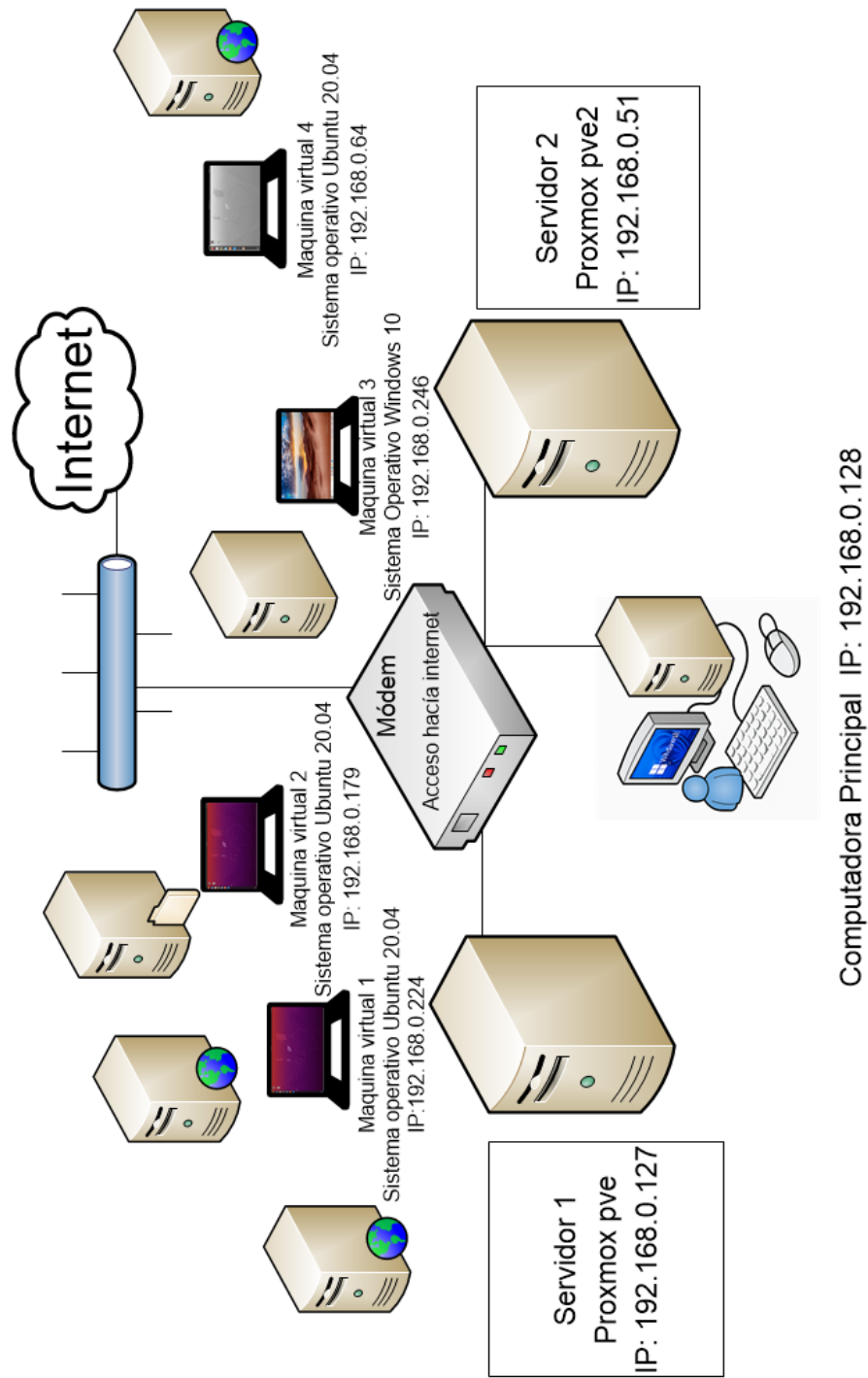


Figura 7.4 Diagrama de la red del Laboratorio de Telecomunicaciones. [Elaboración Propia]

Proxmox VE cuenta con su propio cortafuegos o *firewall*. “Un cortafuego es un sistema de red encargado de separar redes informáticas, efectuando un control del tráfico que transcurre entre ellas. Este control consiste, en última instancia, en permitir o denegar el paso de la comunicación de una red a otra mediante el control de los protocolos de red” [61]. De manera específica, “El *firewall* de Proxmox VE proporciona una manera fácil de proteger su TI infraestructura. Puede configurar reglas de firewall para todos los hosts dentro de un clúster, o definir reglas para máquinas virtuales y contenedores. Características como macros de *firewall*, grupos de seguridad, conjuntos de IP y los alias ayudan a facilitar esa tarea” [60]. Este cortafuego “se basa en la consolidada tecnología Netfilter basada en Linux. Netfilter se basa en un marco de filtrado de paquetes, donde los paquetes de datos de red se permiten o deniegan según un conjunto de reglas definidas. Todas estas reglas se definen como estructuras de tablas en iptables” [39] En la actualidad, las empresas siempre suelen contar con algún *firewall* corporativo. El modelo de la red de Proxmox VE está basado en Linux Bridgers, que actúa como un *switch* virtual. Ahmed menciona que “El firewall Proxmox VE es una función de seguridad que permite proteger de forma sencilla y eficaz un entorno virtual, tanto para el tráfico de red interno como externo. Al aprovechar este firewall, podemos proteger máquinas virtuales, nodos host o todo el clúster mediante la creación de reglas de firewall.” [62]; es decir, se crean reglas para las máquinas virtuales, nodos, *host* y todo el clúster del sistema. De acuerdo con el autor “Todas las copias de seguridad deben almacenarse en un nodo separado, configurado exclusivamente para respaldos” [39]. Esto, ya que si las copias de seguridad se guardan en el mismo nodo de Proxmox VE y ocurre una falla en el *hardware* del equipo, la restauración de las máquinas virtuales se vería impedida. El de Proxmox VE es un *firewall* con estado; es decir, no es sólo un filtro de paquetes de datos, sino que también mantiene un seguimiento constante del estado de las conexiones de red activas, como los protocolos TCP o UDP. También se lo conoce como “filtrado dinámico de paquetes”, que combina las reglas del *firewall* con la naturaleza de las conexiones activas, lo que proporciona

una mejor protección que simplemente filtrar paquetes. Para conectar a los invitados a la red física donde se requiere de una interfaz de puente Linux (comúnmente llamada *vmbrx*), funciona como un conmutador virtual al que se conectan los invitados y las interfaces físicas. “La configuración se puede realizar ya sea a través de la GUI, o editando manualmente el archivo */etc/network/interfaces*, que contiene toda la configuración de la red” [60]. El sistema ofrece dos opciones: *backup* y *snapshot*, cada una con características específicas para la recuperación de datos en caso de fallos o actualizaciones. Las reglas del *firewall* y las distintas formas que se distribuye en cada nodo del clúster. Es ideal para protocolos IPV4 e IPV6. Se distribuye con servicios *iptables*, herramienta que permite configurar las reglas del *firewall* como el tráfico de red que se autoriza, bloquea o redirige. En la zona de agrupación, primero se presenta el tráfico desde/hacia el *host*, tráfico reenviado y reglas del clúster o *host*. Después, está la zona de la máquina virtual, con el tráfico desde/hacia la máquina virtual y reglas individuales. Y luego se presenta la zona de red, con el tráfico entre invitados o *host*/invitado. Sólo tráfico reenviado (*forward*) requiere un *firewall* para *nftables*, que consiste en un “proyecto de netfilter que proporciona filtrado de paquetes y clasificación de paquetes en Linux” [63], y reglas a nivel de red virtual. Algunas configuraciones relevantes de la Red en Proxmox VE son:

- Configuración de Red mediante la GUI: Primero se debe iniciar sesión como root en la GUI para editar las configuraciones de red. Hacer clic en el nodo donde se realizará la configuración para visualizar las opciones de red. Para crear una nueva interfaz, hacer clic en 'Crear' para desplegar el menú de configuración. En este caso, se nombra como *vmbr10* y la red 192.168.2.11/24. [Figura 7.5] Antes de aplicar los cambios, aparece el mensaje Cambios Pendientes indicando que se requiere un reinicio. Para revertir los cambios antes del reinicio, hacer clic en 'Revertir' para restaurar la configuración previa. Si todo está correcto, hacer clic en 'Aplicar configuración'. Una vez finalizado, la conectividad se puede verificar desde la consola enviando un ping con el comando: `ping c 4 192.168.2.11`

Nombre ↑	Tipo	Activo	Inicio a...	Consci...	Puertos/Es...	Modo de B...	CIDR	Puerta de enlace
enp1s0	Dispositivo de ...	Sí	No	No				
vmbr0	Linux Bridge	Sí	Sí	No	enp1s0		192.168.0.51/24	192.168.0.1
wlp2s0	Desconocido	No	No	No				

Figura 7.5: Ventana para crear una red. [Captura Propia]

- Configuración de una Red VLAN: Ahmed menciona que “Una LAN virtual (VLAN) es un estándar de red basado en 802.1q que permite crear una partición lógica en la misma interfaz de red para aislar el tráfico de múltiples redes” [39]. La configuración inicial, se debe acceder al nodo 'pve', y verificar las redes disponibles. Para empezar, se crea una nueva red que servirá como LAN, utilizando 'vmbr1'. Hacer clic en aplicar la configuración para guardar los cambios. Se descarga la imagen iso de *PFsense* desde su página oficial, para administrar el tráfico entre máquinas virtuales y contenedores. En cuanto a la carga de la imagen ISO, primero se accede al almacenamiento local, luego se carga la imagen. Para configurar la consola de *PFsense* después de ser instalado para inicia el sistema y se selecciona la opción 'No' cuando se pregunta por ajustes iniciales, se accede al menú de 'configuración de red' y se define la 'bethernet0' como la interfaz WAN y la 'btnet1' como LAN. La WAN obtiene automáticamente una dirección IP por DHCP, mientras que la LAN se configura manualmente, se selecciona la opción 2 del menú de interfaces, ingresando la IP estática 192.168.10.2/24 y dejando el Gateway vacío, debido a que la WAN manejará la conexión principal. Se establece el rango DHCP para clientes, permitiendo asignaciones dinámicas desde 192.168.10.2 hasta 192.168.10.10.
- Agregar la red a una MV: Para agregar la red a una MV, se debe acceder a la configuración y seleccionar 'Hardware'. Dentro de esta sección, se elige la opción 'Network Device' hace clic en 'agregar' para agregar una nueva interfaz. se selecciona 'vmbr1'. Una vez configurada, se guarda la configuración y la máquina virtual puede utilizar la nueva red asignada. Se abre el CMD de la máquina virtual (Windows 10) y se consulta la nueva red.

- Configuraciones de firewall: Para realizar la configuración del Firewall primero se debe activar el Firewall del 'Centro de Datos', accediendo al cortafuego, hacer clic en agregar y habilitarlo. [Figura 7.6] Luego, se procede con la activación del Firewall del 'nodo', seleccionando Firewall y habilitarlo. Sobre las configuraciones de la MV se ingresa al 'Cortafuegos', se selecciona 'Agregar' y, posteriormente, se realizan pruebas de conectividad en la consola enviando un ping a puertos específicos. De esta manera, se verifican los cambios realizados. [Figura 7.9] Para habilitar la opción, hacer clic en 'Habilitar'. Se dispone de una lista, y si el puerto que se desea bloquear no está incluido, se puede hacer clic en 'Nueva regla', luego seleccionar el protocolo correspondiente.

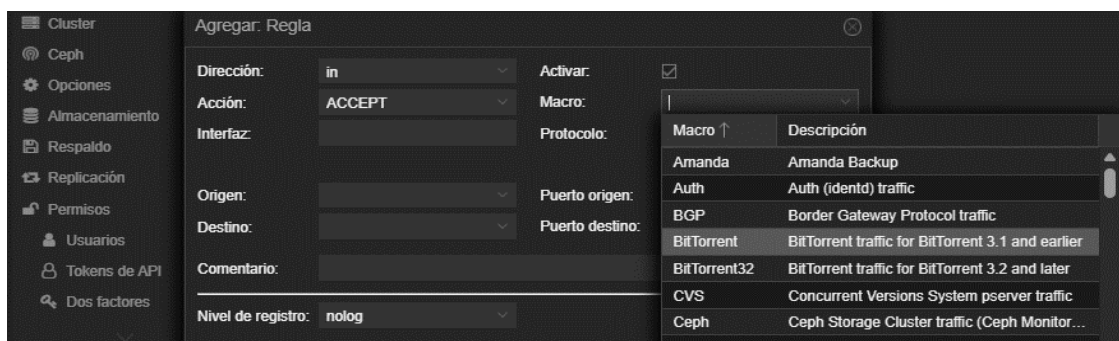


Figura 7.6: Ventana de Firewall. [Captura Propia]

- Copias de seguridad en Proxmox VE: La opción de backup, permite crear un archivo con una copia de seguridad completa de la máquina virtual, antes de comenzar hay que definir un destino de almacenamiento donde se guarda la información. Por otro lado, para la opción de snapshot, se guarda una imagen del sistema, esto permite recuperar el sistema, funcionando de manera similar a una pausa en el estado de ejecución. Para ejecutar un backup se debe seleccionar dónde se almacenará la copia. Cabe observar que, si no hay discos adicionales, el respaldo se guarda en el almacenamiento local, el sistema debe solicitar la confirmación antes de iniciar el proceso de respaldo.

Apéndice D

Automatización de tareas y mantenimiento: Proxmox VE cuenta con una Interfaz de Programación de aplicaciones (API) que utiliza tecnología similar a la Transferencia de Estado Representacional (REST). Esto permite la integración con otras herramientas y sistemas para administrar sus componentes como máquinas virtuales, contenedores y clústeres. Esta operación se realiza mediante solicitudes HTTP. “La herramienta de gestión Proxmox VE (PVESH) permite invocar directamente la función API, sin utilizar servidores REST / HTTPS” [60]. El formato principal se adecua a la estructura del esquema JSON, herramientas de administración de terceros para una incorporación rápida al realizar automatizaciones y *scripts* propios.

La Figura 8.1 muestra cómo se realiza el proceso de instalación para la automatización API utilizando Python para comunicarse al servidor de Proxmox VE. Se pueden ver sus distintas etapas. A propósito de esto, Python “Es el lenguaje de referencia actual, tanto para la administración de sistemas como para el scripting de propósito general. Se integra fluidamente con las API REST; y cuenta con bibliotecas bien desarrolladas para aprendizaje automático, análisis de datos y computación numérica” [64]. Mientras, la Figura 8.2 presenta el proceso al momento de automatizar una tarea para crear una máquina virtual.

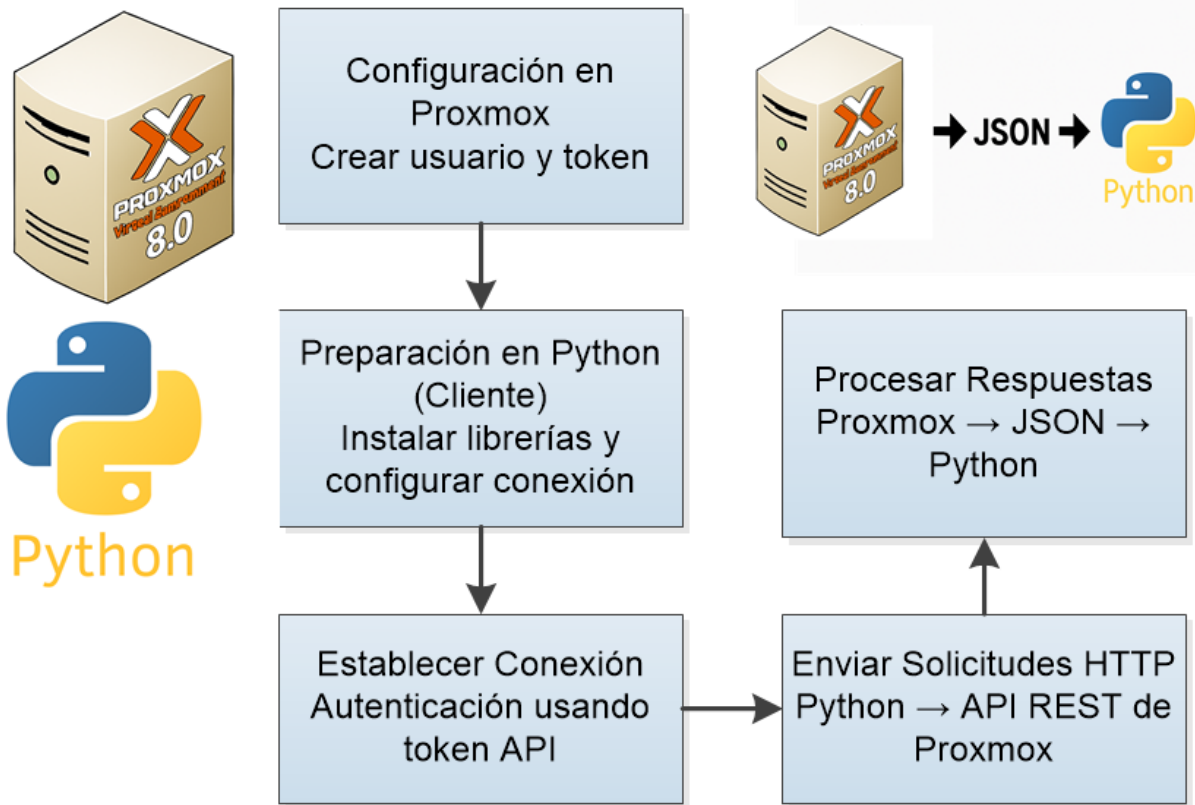


Figura 8.1: Diagrama API con Python. [Elaboración Propia]

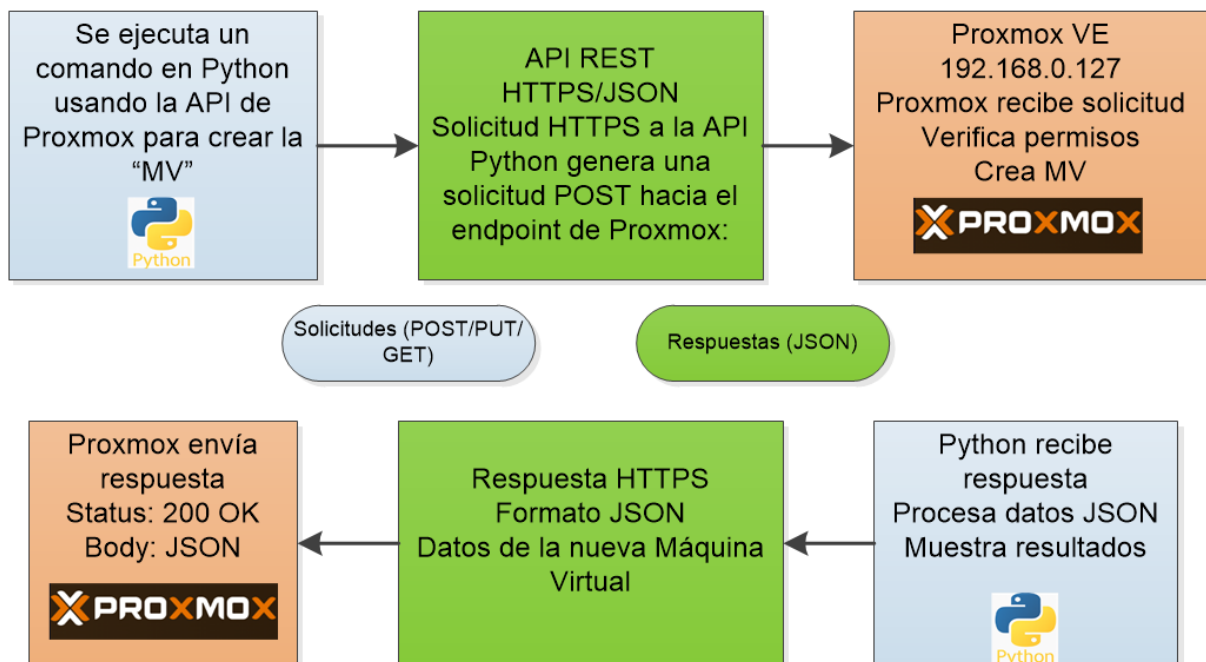


Figura 8.2: Diagrama API con Python crear una MV. [Elaboración Propia]

“Los entornos de virtualización como Proxmox VE facilitan enormemente la alta disponibilidad, ya que eliminan la dependencia del *hardware*. Además, permiten la configuración y el uso de dispositivos de red y almacenamiento redundantes, de modo que, si un *host* falla, puede iniciar esos servicios en otro *host* del clúster” [60]. Matemáticamente, la disponibilidad se define como la relación entre el tiempo total que un servicio puede utilizarse durante un intervalo (A) y la duración de dicho intervalo (B). Normalmente, se expresa como un porcentaje del tiempo de actividad en un año:

Los Requisitos para lograr una alta disponibilidad con Proxmox VE son “Al menos tres nodos de clúster (para obtener un quórum confiable)”, “Almacenamiento compartido para máquinas virtuales y contenedores”, “Redundancia de *hardware* (en todas partes)” y “Utilizar componentes de ‘servidor’ confiables” [60]. Al respecto, “Un clúster es una herramienta para crear un grupo de servidores físicos mediante el *pvecm* (Administrador de clústeres de Proxmox VE) del gestor de clústeres” [60]; es decir, un clúster consiste en dos o más nodos de computadoras conectados en la misma red utilizando Proxmox VE como sistema anfitrión. Esto facilita la migración de máquinas virtuales entre nodos, asegurando un tiempo de actividad máximo y una administración centralizada. Esto también permite redundancia en caso de que un nodo falle por cualquier razón. Los clústeres no se ofrecen de forma predeterminada, sino que se requiere que se configuren previamente para automatizarse. Los clústeres dependen de ciertos puertos específicos, los cuales deben permanecer accesibles y libres de restricciones por parte del *firewall*.

Cabe observar que cada nodo contiene máquinas virtuales y contenedores conectados mediante puentes virtuales (*vmbr0*); es decir, las máquinas virtuales y los contenedores se conectan a la red del servidor vinculándose a la interfaz física de cada nodo como si fueran dispositivos anfitriones.

Así pues, una vez conectados los clústeres a los servidores, todos deben estar encendido para que el clúster permita trabajar. La Figura 8.3 muestra un diagrama que representa la simplificación de cómo se conectan los componentes dentro de un clúster. Se puede ver que primero se encuentra la Internet, que es ingresada al *firewall* de Proxmox VE. Después, se presenta el *switch*, el cual distribuye la conexión a la Internet. Después, se presentan los dos servidores de Proxmox VE, y dentro de cada servidor puede haber máquinas virtuales, contenedores. A un lado se presenta el servidor de almacenamiento, donde puede haber espacios compartidos, respaldos o archivos.

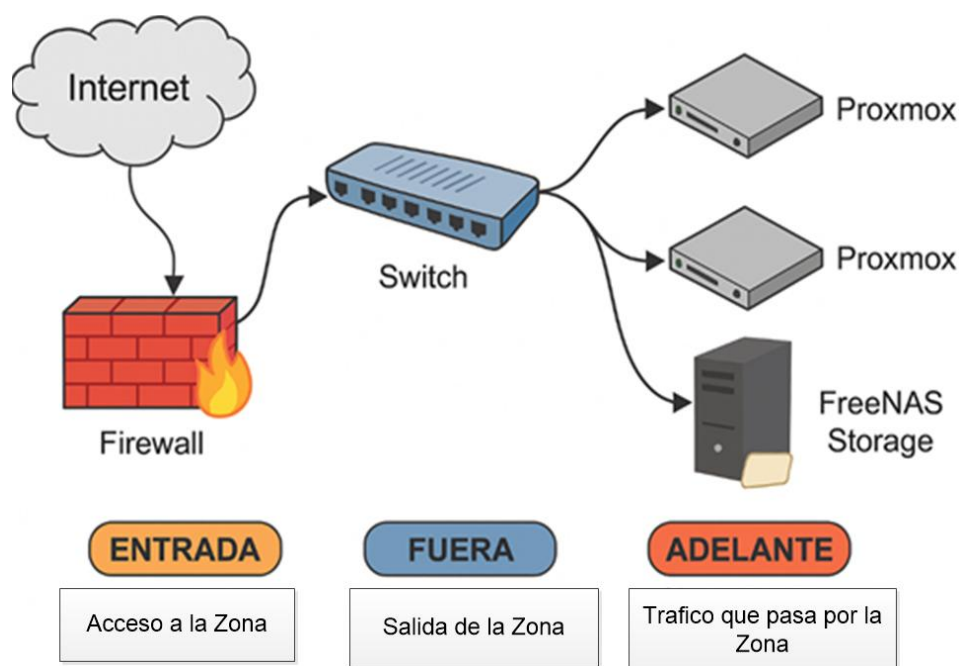


Figura 8.3: Diagrama de un clúster de Proxmox VE básico. [Fuente: W. Ahmed, Mastering Proxmox, Birmingham, Reino Unido: Packt Publishing Ltd., 2017.]

Para empezar con la creación de un clúster, se pueden utilizar dos o más nodos. Cada nodo debe contar con una IP única. En este trabajo, se ocuparon dos nodos para el clúster: el nodo principal (IP 192.168.0.127) y un nodo secundario (IP 192.168.0.51). El nodo principal fue el encargado de ser el administrador principal para que así se puedan realizar nodos secundarios, mientras que el nodo secundario se une al clúster principal. Cabe añadir que se pueden incluir más nodos dependiendo de la cantidad de servidores que se requiera conectar.

Para iniciar con la configuración del clúster, se debe acceder a la interfaz del centro de datos. Una vez dentro, hay que seleccionar la opción “Clúster” y hacer clic en “crear Clúster”. Se debe asignar un nombre al clúster que no exceda 15 caracteres. Para este trabajo, el clúster para el nodo principal se nombró como “clusterUno”. Al finalizar la creación del clúster, se generó la información de unión (*join information*), que incluye una clave digital en sha-2. Esta información fue utilizada para autenticar las conexiones entre los nodos. La dirección IP del nodo principal se quedó registrada dentro del sistema. Hay que guardar la clave, debido a que se ocupará para la conexión entre nodos.

Para realizar cambios a futuro y cuando se requiera modificar la IP de un nodo, se podría requerir la reconfiguración completa del clúster. La configuración inicial está directamente vinculada con las direcciones IP establecidas. Este aspecto se debe tomar en cuenta para evitar problemas en la gestión del clúster. Para iniciar con el proceso de unión de un nodo a un clúster principal, se utilizó la dirección IP 192.168.0.51 del nodo secundario, haciendo clic en “Unirse al clúster”. Cabe señalar que la IP se asignó por medio del protocolo DHCP. Una vez dentro del clúster, se introdujo la clave de información codificada y se hizo clic en “Unión” para completar el proceso. A continuación, se debe llenar la información en los campos requeridos. Para terminar, se debe hacer clic en unir “ClusterUno”. Se observó que la dirección IP del nodo principal generase un *fingerprint* único y se confirmó la autenticidad de la conexión; es decir, se generó una conexión segura de forma legítima para evitar intercambios y algún ataque a la conexión. Una vez unidos, los nodos pudieron ser administrados desde el nodo principal [Figura 8.4]. En Proxmox VE también se pudieron eliminar nodos; sin embargo, para dicha tarea se requiere asegurar que las máquinas virtuales no estén en ejecución y éstas se deben mover a un nodo diferente siguiendo los pasos para migrar una máquina virtual.

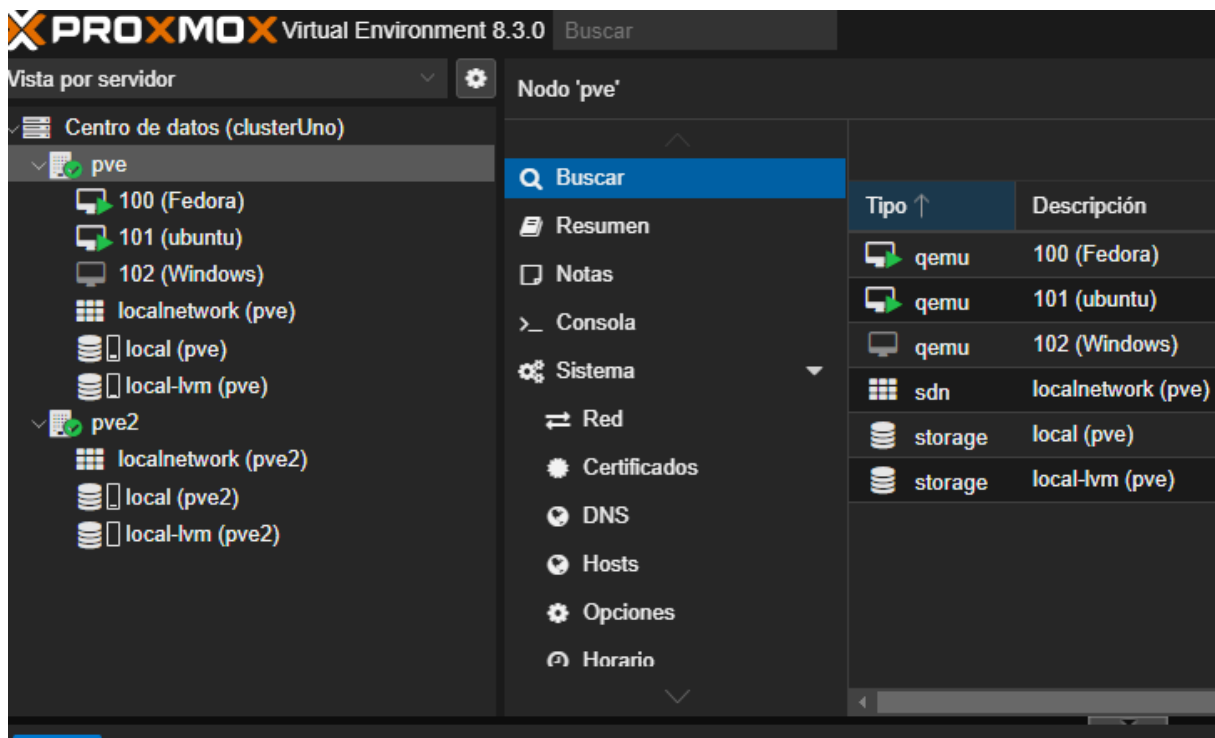


Figura 8.4: Menú de administración de nodos. [Captura Propia]

“Las copias de seguridad son esenciales para cualquier implementación de TI sensata, y Proxmox VE ofrece una solución totalmente integrada que aprovecha las capacidades de cada tipo de almacenamiento y sistema invitado. Esto permite al administrador del sistema ajustar con precisión, mediante la opción “de modo”, la consistencia de las copias de seguridad y el tiempo de inactividad del sistema invitado” [60].

El proceso de clonación permite duplicar máquinas virtuales existentes para su uso en otros nodos del clúster. Para este trabajo, cada nodo contaba con dos máquinas virtuales. Para clonar una máquina virtual, se hizo clic en “clonar” dentro de la interfaz. Una vez terminado, se pudo iniciar la máquina virtual para ocuparlo.

Para empezar con el proceso de migración y trasladar recursos entre nodos dentro de un clúster, se debe hacer clic en la opción “migrar”. Este proceso transfiere completamente el disco del contenedor o máquina virtual. El tiempo dependerá del tamaño del disco y la velocidad de la red.

Al respecto, hay dos tipos de migración: con almacenamiento compartido, que puede ser como “NFS”, “iSCSI”, etcétera, y con almacenamiento local, que puede tomar un mayor tiempo debido a que se envía a un nodo diferente. Para migrar la máquina virtual se hizo clic en “migrar” para iniciar el proceso y verificar la transferencia hacia el nodo destino. Al completar la migración, se inició sesión en la máquina virtual y se verificó su funcionalidad.

En el proceso de administración y operación del entorno virtual con Proxmox VE, se observaron algunos aspectos relevantes de sus funcionalidades que merecen mención. En primer lugar, la generación de usuarios y la asignación de roles y permisos es una función que permite un procedimiento más ordenado y facilita la administración del proceso. Como administrador, uno puede decidir qué permisos otorgar a cada uno de los usuarios generados. Ejecutar estas funciones, ya sea por medio de la interfaz web o la consola de Proxmox VE, no es en realidad una tarea complicada de entender. De hecho, una vez que se comienza, con el paso del tiempo se hace cada vez más fácil utilizarla.

Asimismo, la elaboración de clústeres es una función relevante que brinda Proxmox VE y que puede ser altamente aprovechada. Al momento de realizar algún respaldo, la estructura en clústeres hace posible copiar la máquina virtual completa y migrarla a nodos secundarios. Con esto se torna más fácil tener una paquetería instalada o un desarrollo de *software* en un nodo principal y que estos elementos se puedan reflejar en los nodos secundarios.

Otro de los puntos que llamó la atención fue la opción de uso de almacenamiento compartido que ofrece la herramienta, debido a que permite conservar el sistema original que presenta el equipo y utilizar un almacenamiento externo. Esto es de ayuda en caso de que el equipo original tenga información importante y no se quiera eliminarla. Al instalar un disco externo, puede utilizarse como un sistema anfitrión para Proxmox VE.

Una función interesante más es la de automatización de tarea. Como usuario, permite realizar distintos *scripts* y ejecutarlos para varias actividades. Esto facilita mucho el tiempo debido a que con cada *script* que se realice se estarán llevando a cabo ciertos procesos. En este caso, se implementó Python para realizar este tipo de movimientos.

Uno de los mayores retos fueron que al momento de administrar y operar el entorno virtual por medio de Proxmox VE fue entender cómo funciona cada *script*. Afortunadamente, la información consignada en la página oficial de Proxmox VE y algunos libros que ofrecen guías del funcionamiento de cada *script* me ayudaron a entender mejor el proceso. En ellas pude encontrar la documentación del uso e implementación de la interfaz de programación de aplicaciones (API). Al realizar distintos *scripts*, obtuve un código, usuario y contraseña para tender las conexiones mediante Python. Esto fue muy útil para automatizar los procesos.

Gracias a que Proxmox VE cuenta con su propia interfaz de monitoreo con resultados similares a los que ofrecen los *scripts* “htop” y “top” que se ocuparon, pude observar de manera más ágil cómo estaban trabajando los equipos. Con ello, me di cuenta de que a estos equipos aún se les puede sacar provecho. Además, Proxmox VE brinda respaldos de seguridad en caso de que se presente alguna falla. De este modo, se puede restablecer el sistema.

Para cerrar este trabajo, una de las claves más importantes para que se pueda proteger el servidor es la implementación del *firewall* que se tiene dentro y fuera. Éste permite controlar quien puede entrar y salir cerrando los puertos que no deban usarse y dejando abiertos sólo los necesarios para que el sistema de Proxmox VE funcione. Se trata de una herramienta integrada con el fin de proteger a las máquinas virtuales, lo que permite realizar establecer restricciones, ya sea por puertos o por direcciones IP.

Gracias a esta medida de seguridad se puede controlar el tráfico que entra y sale en cada servidor. Utilizada de forma adecuada, contribuye a la seguridad del sistema. Otra forma de proteger la seguridad es la implementación del autenticado de dos pasos permite que sólo se tenga acceso con el consentimiento de la persona responsable del usuario. Finalmente, cabe observar que Proxmox VE es una herramienta que aún está en desarrollo y cada versión nueva implementa nuevos procesos que permiten apoyar aún más a los administradores. Es preciso estar atentos a las nuevas actualizaciones y adoptarlas para aprovechar las nuevas funcionalidades que se le vayan incorporando.

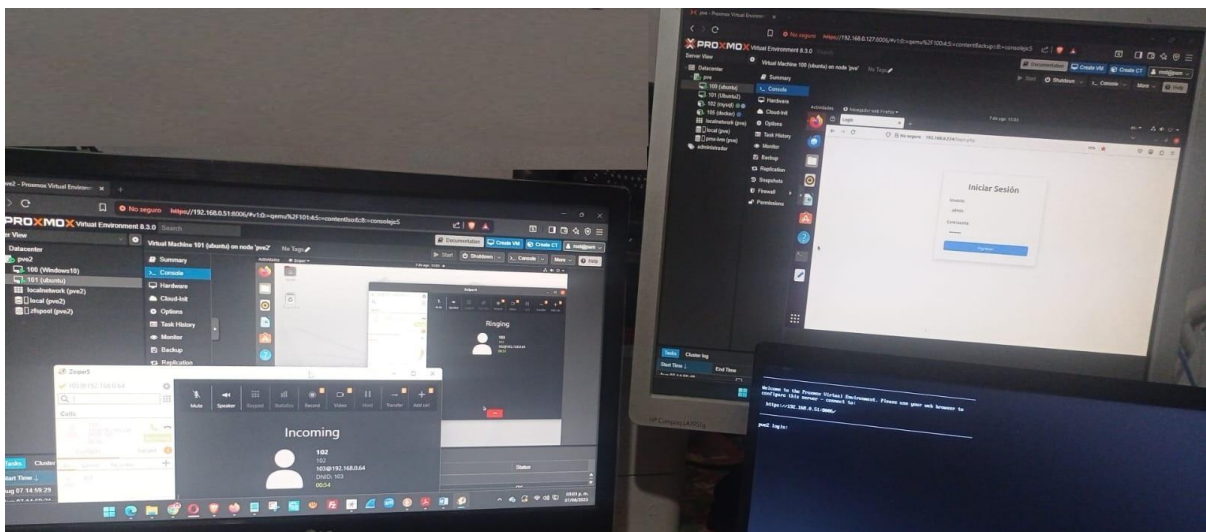


Figura 8.5: Equipos del Laboratorio de Telecomunicaciones. [Captura Propia]