UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Solución de infraestructura de nube privada y almacenamiento para entornos multimedia

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: Margossian, Nicolas Alejandro

Tutor/a: Serrat Olmos, Manuel David

CURSO ACADÉMICO: 2025/2026

Dedicatoria

Dedico este trabajo a mis padres, **Gabriela** y **Alejandro**, por entregar su vida y su esfuerzo para que este logro sea hoy una realidad. A mi hermano **Santiago**, mi compañero de vida, por la complicidad de siempre y creer en mí desde el primer día. A mi novia **Ampi**, por su amor, paciencia y apoyo constante. Gracias por haber estado siempre a mi lado, demostrándome que un año y medio separados por la distancia y las fronteras nunca fue una barrera para sostenernos. Finalmente, rindo un sentido homenaje a la memoria de mi mentor y profesor, **Rodrigo Pablo Balsells**, quien a lo largo de más de diez años me inculcó no solo la vocación por la informática, sino también los pilares de la disciplina, la ética y la cultura del trabajo.

Agradecimientos

Deseo expresar mi más sincero agradecimiento a las personas e instituciones que han hecho posible la realización de este Trabajo de Fin de Grado:

- A mi tutor, **Manuel David Serrat Olmos**, a quien primero debo agradecer por brindarme una de las asignaturas más enriquecedoras de la titulación, y posteriormente, por su compromiso en la dirección y tiempo brindado en cada revisión de este trabajo.
- Al **tribunal y comité académico**, por la atenta consideración de este escrito, y la profesionalidad aplicada durante el proceso de calificación que permite culminar esta etapa.
- A la **Universitat Politècnica de València (UPV)** y a la **Escuela Técnica Superior de Ingeniería Informática (ETSINF)**, por haberme abierto sus puertas y aceptado para cursar este programa de doble titulación en Ingeniería Informática durante tres enriquecedores semestres.
- A la **Universidad de Belgrano**, por haberme brindado una sólida formación académica a lo largo de cuatro años y por otorgarme la invaluable oportunidad de participar en este programa de doble titulación internacional.
- A mi primo, **Gaston Avigliano**, por su apoyo incondicional y por haberme acompañado y guiado en mi crecimiento tanto personal como profesional.

Resum

El present Treball Final de Grau exposa el disseny, la implementació i la validació d'una infraestructura centralitzada orientada a la gestió d'actius audiovisuals de gran volum. El projecte sorgix davant la necessitat de resoldre les deficiències organitzatives i les limitacions tècniques que presenten els serveis comercials d'allotjament a través d'internet, com ara els alts costos recurrents i les restriccions de velocitat en la transferència de fitxers de grans dimensions.

Per a solucionar aquestes problemàtiques, s'ha construït una arquitectura basada primordialment en tecnologies de codi obert, garantint així la privacitat de les dades i la independència enfront de proveïdors externs. El sistema es fonamenta en un entorn de virtualització híbrida que orquestra tant màquines virtuals completes com contenidors aïllats sobre la base de l'hipervisor Proxmox VE.

La capa d'aplicacions proporciona als usuaris un accés ubic i multiplataforma mitjançant una eina de col·laboració autoallotjada i un servei de transmissió audiovisual contínua. La connectivitat remota s'estableix a través de xarxes privades virtuals amb xifratge criptogràfic, eliminant l'exposició directa dels serveis interns. Tot el trànsit cap a l'exterior es troba estrictament regulat per un tallafoc perimetral, complementat amb sistemes de prevenció d'intrusions i intel·ligència col·laborativa per a neutralitzar amenaces en temps real. Així mateix, s'ha integrat un ecosistema d'observabilitat proactiva sustentat en la recopilació de mètriques i panells de visualització interactius, a més d'una rigorosa política de còpies de seguretat i replicació per a assegurar la recuperació del sistema davant de possibles errades mecàniques o lògiques.

Els resultats confirmen una notable optimització en els fluxos de treball mitjançant l'aprofitament de xarxes locals d'alta velocitat i l'establiment d'un entorn segur, privat i escalable per a un entorn professional.

Paraules clau: Servidor, Emmagatzematge, NAS, Virtualització, Multimèdia, Xarxes, Ciberseguretat, IDS/IPS, Proxmox VE, ZFS, VPN, Docker, Monitoratge, Núvol Privat

Resumen

El presente Trabajo Final de Grado expone el diseño, la implementación y la validación de una infraestructura centralizada orientada a la gestión de activos audiovisuales de gran volumen. El proyecto surge ante la necesidad de resolver las deficiencias organizativas y las limitaciones técnicas que presentan los servicios comerciales de alojamiento a través de internet, tales como los altos costes recurrentes y las restricciones de velocidad en la transferencia de ficheros de gran tamaño.

Para subsanar estas problemáticas, se ha construido una arquitectura basada primordialmente en tecnologías de código abierto, garantizando así la privacidad de los datos y la independencia frente a proveedores externos. El sistema se fundamenta en un entorno de virtualización híbrida que orquesta tanto máquinas virtuales completas como contenedores aislados sobre la base del hipervisor Proxmox VE.

La capa de aplicaciones proporciona a los usuarios un acceso ubicuo y multiplataforma mediante una herramienta de colaboración autoalojada y un servicio de transmisión audiovisual continua. La conectividad remota se establece a través de redes privadas virtuales con cifrado criptográfico, eliminando la exposición directa de los servicios internos. Todo el tráfico hacia el exterior se encuentra estrictamente regulado por un cortafuegos perimetral, complementado con sistemas de prevención de intrusiones e inteligencia colaborativa para neutralizar amenazas en tiempo real. Asimismo, se ha integrado un

ecosistema de observabilidad proactiva sustentado en la recolección de métricas y paneles de visualización interactivos, además de una rigurosa política de copias de seguridad y replicación para asegurar la recuperación del sistema ante posibles fallos mecánicos o lógicos.

Los resultados confirman una notable optimización en los flujos de trabajo mediante el aprovechamiento de redes locales de alta velocidad y el establecimiento de un entorno seguro, privado y escalable para un entorno profesional.

Palabras clave: Servidor, Almacenamiento, NAS, Virtualización, Multimedia, Redes, Ciberseguridad, IDS/IPS, Proxmox VE, ZFS, VPN, Docker, Monitorización, Nube Privada

Abstract

This Bachelor's Thesis presents the design, implementation, and validation of a centralized infrastructure aimed at managing large volume audiovisual assets. The project arises from the need to solve the organizational deficiencies and technical limitations presented by commercial cloud hosting services, such as high recurring costs and speed restrictions in the transfer of large files.

To overcome these issues, an architecture based primarily on open-source technologies has been built, thus ensuring data privacy and independence from external providers. The system is founded on a hybrid virtualization environment that orchestrates both full virtual machines and isolated containers based on the Proxmox VE hypervisor.

The application layer provides users with ubiquitous, cross-platform access through a self-hosted collaboration tool and a continuous audiovisual streaming service. Remote connectivity is established through virtual private networks with cryptographic encryption, eliminating the direct exposure of internal services. All external traffic is strictly regulated by a perimeter firewall, complemented by intrusion prevention systems and collaborative intelligence to neutralize threats in real time. Furthermore, a proactive observability ecosystem based on metrics collection and interactive visualization dashboards has been integrated, alongside a rigorous backup and replication policy to ensure system recovery in the event of potential mechanical or logical failures.

The results confirm a notable optimization in workflows through the use of high-speed local networks and the establishment of a secure, private, and scalable environment for professional use.

Key words: Server, Storage, NAS, Virtualization, Multimedia, Networks, Cybersecurity, IDS/IPS, Proxmox VE, ZFS, VPN, Docker, Monitoring, Private Cloud

Índice general

Índice de figuras

Índice de tablas

Glosario de Acrónimos

A continuación se definen los principales términos técnicos y acrónimos utilizados a lo largo de la presente memoria, con el fin de facilitar la comprensión de la arquitectura a lectores que puedan provenir de áreas ajenas a la administración de sistemas y redes:

CAPEX (*Capital Expenditure*): Gasto de capital o inversión inicial en infraestructura física y componentes.

OPEX (*Operating Expense*): Gasto operativo. Hace referencia a los costes recurrentes o permanentes para mantener un servicio funcionando (ej. suscripciones mensuales en la nube).

DMZ (*Demilitarized Zone*): Zona desmilitarizada. Subred aislada que se ubica entre la red interna segura y una red externa insegura (Internet) para exponer servicios de forma controlada.

DNAT (*Destination Network Address Translation*): Traducción de direcciones de red de destino. Técnica utilizada en cortafuegos para redirigir el tráfico entrante hacia servidores internos.

IDS (*Intrusion Detection System*): Sistema de Detección de Intrusos. Herramienta de seguridad que monitoriza el tráfico de red en busca de comportamientos anómalos o firmas maliciosas.

IPS (*Intrusion Prevention System*): Sistema de Prevención de Intrusos. Herramienta similar al IDS, pero con capacidad activa para bloquear o rechazar el tráfico malicioso detectado.

LAN (*Local Area Network*): Red de área local que interconecta ordenadores y dispositivos dentro de un entorno físico limitado (como un domicilio u oficina).

NAS (*Network Attached Storage*): Dispositivo o servidor de almacenamiento conectado en red que centraliza los datos y permite su acceso a diferentes clientes autorizados.

Proxmox VE (*Virtual Environment*): Plataforma de gestión de virtualización de servidores de código abierto, basada en Debian, capaz de orquestar tanto máquinas virtuales como contenedores Linux.

SaaS (*Software as a Service*): Modelo de distribución de software donde las aplicaciones y los datos se alojan en los servidores de un proveedor externo (ej. Google Drive, Dropbox).

TCP/IP (*Transmission Control Protocol / Internet Protocol*): Pila de protocolos de comunicaciones en los que se basa Internet y el enrutamiento de red moderno.

VPN (*Virtual Private Network*): Red privada virtual que permite crear una conexión segura y cifrada sobre una red pública o menos segura.

ZFS (*Zettabyte File System*): Sistema de archivos y gestor de volúmenes lógicos que destaca por su alta protección contra la corrupción de datos y soporte para grandes capacidades.

ARC (*Adaptive Replacement Cache*): Mecanismo de caché avanzado utilizado por ZFS en la memoria RAM para agilizar las operaciones de lectura y escritura.

CoW (*Copy-on-Write*): Paradigma de copia en escritura donde los bloques de datos no se sobrescriben, sino que los cambios se graban en bloques nuevos, facilitando la creación de instantáneas inmutables.

DDNS (*Dynamic Domain Name System*): Sistema de Nombres de Dominio Dinámico. Servicio que actualiza automáticamente la dirección IP de un nombre de dominio en tiempo real.

DPI (*Deep Packet Inspection*): Inspección profunda de paquetes. Técnica de seguridad que analiza el contenido de los datos en tránsito, y no solo sus cabeceras, para detectar amenazas.

ISP (*Internet Service Provider*): Proveedor de Servicios de Internet. Empresa que proporciona la conexión y el acceso a la red externa (Internet).

KVM (*Kernel-based Virtual Machine*): Tecnología de virtualización integrada en el núcleo de Linux que permite al sistema operativo actuar como un hipervisor de tipo 1 para ejecutar máquinas virtuales completas.

NIC (*Network Interface Card*): Tarjeta de interfaz de red. Componente de hardware que conecta un dispositivo a una red informática, determinando la velocidad máxima de transferencia física.

ROI (*Return on Investment*): Retorno de la inversión. Métrica financiera que compara el beneficio o ahorro obtenido en relación con la inversión de capital realizada.

SSH (*Secure Shell*): Protocolo de administración remota que permite controlar servidores a través de Internet mediante un canal de comunicación cifrado.

TSDB (*Time Series Database*): Base de Datos de Series Temporales. Sistema de almacenamiento optimizado para registrar secuencias de valores numéricos asociados a marcas de tiempo, utilizado en la monitorización de sistemas.

UPS / SAI (*Uninterruptible Power Supply* / Sistema de Alimentación Ininterrumpida): Dispositivo que gracias a sus baterías puede proporcionar energía eléctrica tras un apagón, permitiendo un apagado controlado del servidor.

CAPÍTULO 1
Introducción

1.1 Contexto y motivación

En la última década, la producción de contenido audiovisual ha generado un crecimiento exponencial en el volumen de datos que manejan no solo las grandes productoras, sino también los creadores independientes y los entornos domésticos profesionales. La gestión eficiente de estos activos digitales se ha convertido en un desafío crítico de ingeniería.

La motivación principal de este Trabajo Final de Grado nace de una necesidad real en mi entorno familiar. Tanto mi padre, como mi hermano y yo, desarrollamos actividades profesionales vinculadas a la creación y edición de contenido multimedia. Este escenario implica la generación constante de archivos de gran peso (vídeo en alta resolución, proyectos de edición, bibliotecas de fotografía RAW), cuyo almacenamiento y transmisión requieren infraestructuras que superan las capacidades domésticas convencionales.

Históricamente, la infraestructura tecnológica de la que disponíamos se basaba en el almacenamiento local distribuido (discos duros internos en cada estación de trabajo) y el uso de unidades externas para el intercambio de información. Sin embargo, a medida que la calidad y el volumen del material aumentaban, esta metodología se volvió insostenible. La dependencia de servicios de nube pública (como Google Drive o Dropbox) resultaba inviable técnica y económicamente debido a los grandes volúmenes de datos (superiores a los 10 TB) y a las limitaciones de ancho de banda de subida y bajada típicas de las conexiones residenciales para mover archivos de tal magnitud.

Este proyecto surge, por tanto, de la voluntad de aplicar los conocimientos de ingeniería informática adquiridos para diseñar una solución de Nube Privada robusta. El objetivo no es solo almacenar datos, sino transformar el flujo de trabajo familiar mediante la centralización, garantizando la soberanía de los datos, eliminando costes recurrentes de suscripciones y asegurando un rendimiento de red local capaz de soportar edición y streaming en tiempo real.

1.2 Definición del problema

Antes de la implementación de la solución propuesta en este TFG, el flujo de trabajo presentaba ineficiencias críticas que afectaban tanto a la productividad como a la seguridad de la información. La problemática no se limitaba a una cuestión de orden, sino que se veía agravada por una barrera geográfica y limitaciones tecnológicas en los métodos de transferencia. Se pueden identificar cuatro vectores principales de conflicto:

1.2.1. Fragmentación Geográfica y Silos de Información

Uno de los desafíos más importantes era gestionar el entorno familiar que abarca dos domicilios físicos distintos (mi residencia y la de mi padre). A menudo, los activos digitales (brutos de cámara y contenido de tarjetas SD recién volcadas) se encontraban físicamente en una ubicación, mientras que la necesidad de edición o consulta surgía en la otra. Esto generaba desorganización en el acceso a la información:

- Imposibilidad de acceso a los archivos sin desplazarse físicamente.

1.2.2. Limitaciones de la Nube Pública (SaaS)

Se intentó mitigar la dispersión utilizando servicios de almacenamiento en la nube comercial (*Google Drive*). Sin embargo, esta solución presentó cuellos de botella insalvables para nuestro caso de uso:

- **Flujo de trabajo ineficiente:** Al tratarse de un modelo SaaS, cada archivo debía subirse primero a través de internet y luego descargarse nuevamente en el otro domicilio para su edición o consulta. Dado el volumen de material audiovisual (RAW/4K), este proceso implicaba tiempos de espera prolongados tanto en la carga como en la descarga, introduciendo una ineficiencia operativa que penalizaba el trabajo colaborativo.
- **Costes recurrentes y límites de almacenamiento:** La necesidad de compartir entregables pesados con terceros provocaba que el almacenamiento gratuito o los planes básicos se agotaran rápidamente, obligando a gestionar suscripciones mensuales que escalaban en precio según el espacio necesitado.

1.2.3. Riesgos del transporte físico

Ante la lentitud de la nube, el recurso habitual era el transporte físico de datos mediante tarjetas SD, pendrives y discos duros externos. Esta práctica introdujo riesgos de seguridad:

- **Extravío de soportes:** En el tránsito constante entre domicilios, se experimentaron pérdidas de unidades flash, lo que implicaba no solo la pérdida del activo digital (si no había copia), sino una brecha de privacidad.
- **Desgaste y fallo:** La manipulación constante aumentaba la tasa de fallos mecánicos en los discos duros portátiles.
- **Desalineación física entre equipo y datos:** En múltiples ocasiones, el equipo fotográfico, junto con la tarjeta SD, debía permanecer en un domicilio, mientras que la necesidad de acceso a la información surgía en el otro. Esto obligaba a desplazamientos adicionales, incluso dentro del mismo día, para copiar el contenido y posteriormente devolver la unidad flash junto al equipo.

1.2.4. Inexistencia de Acceso Móvil

El consumo de los contenidos estaba restringido estrictamente a los ordenadores de escritorio. No existía una infraestructura que permitiera visualizar fotos, vídeos o documentos desde dispositivos móviles (smartphones o tablets) de forma fluida. Si un familiar quería mostrar un vídeo o foto reciente desde su teléfono, no podía hacerlo a menos

que el archivo hubiera sido copiado previamente a la memoria interna del dispositivo, eliminando cualquier posibilidad de acceso bajo demanda o en directo.

1.3 Objetivos del proyecto

1.3.1. Objetivo general

El objetivo principal de este trabajo ha sido el diseño, implementación y puesta en marcha de una infraestructura de nube privada de alto rendimiento, basada en software de código abierto, capaz de centralizar el almacenamiento, la gestión y la distribución de contenido multimedia, en un entorno pequeño distribuido geográficamente, garantizando el control absoluto de los datos y la independencia de proveedores de nube pública.

1.3.2. Objetivos específicos

Para alcanzar el objetivo general, se han definido los siguientes hitos técnicos:

- **Implementación de almacenamiento en red (NAS):** Desplegar un servidor NAS (Network Attached Storage) como repositorio centralizado de archivos, accesible a través de la red local mediante protocolos de almacenamiento en red como SMB/CIFS. Esto permite compartir y acceder a los activos digitales desde cualquier dispositivo autorizado dentro de la red (ordenadores, portátiles, estaciones de edición, móviles, entre otros).

- **Acceso audiovisual bajo demanda:** Garantizar la disponibilidad del contenido audiovisual (fotografías y vídeos) desde dispositivos móviles en cualquier momento, permitiendo la consulta, visualización y revisión del material sin necesidad de trasladar físicamente soportes de almacenamiento.

- **Implementación de estrategia de respaldo duplicado local:** Definir la información crítica que se almacena en el NAS y se replica a un HDD local, así como a un ordenador personal en el hogar. Esta estrategia garantizará la redundancia dentro del entorno físico principal, aunque no cumple con el principio 3-2-1 de respaldo debido a la ausencia de una copia fuera del sitio.

- **Optimización del rendimiento de red:** Configurar la infraestructura de red local para aprovechar el ancho de banda disponible (2.5 Gbps), minimizando los tiempos de transferencia.

- **Aseguramiento del acceso remoto:** Implementar una arquitectura de seguridad que combine conexión segura mediante VPN para administración, gestión y acceso a recursos internos (como SMB/CIFS), junto con la exposición controlada de ciertos servicios a Internet (por ejemplo, Nextcloud), utilizando medidas de seguridad adicionales para minimizar riesgos. Esto permitirá el acceso remoto funcional y seguro, sin depender exclusivamente de la VPN, adaptándose a las necesidades de uso real de los dispositivos, hogares y clientes.

- **Gestión de intercambio externo:** Habilitar mecanismos para compartir entregables finales de gran tamaño con terceros a través de enlaces públicos seguros y temporales, eliminando la dependencia de servicios de transferencia de archivos comerciales.

- **Monitorización y observabilidad:** Implementar un sistema de vigilancia del estado del servidor (CPU, RAM, Temperaturas, Estado de los discos), asegurando una respuesta proactiva ante posibles fallos de hardware.

1.4 Impacto esperado

La implementación de esta infraestructura de nube privada no solo busca resolver problemas técnicos inmediatos, sino generar un cambio estructural en la forma en que se gestionan los activos digitales del entorno familiar. Se proyectan los siguientes impactos clave:

1.4.1. Soberanía de Datos e Independencia Tecnológica

El proyecto devuelve el control absoluto de la información a sus propietarios. Se elimina la dependencia de los Términos de Servicio (ToS) cambiantes de las grandes tecnológicas y se evita la cautividad del proveedor.

1.4.2. Mejora sustancial en el Rendimiento (Velocidad)

Se espera una reducción significativa en los tiempos de acceso y transferencia a nivel local. Al pasar de intercambios manuales mediante pendrives o discos externos, procesos más lentos, propensos a errores y operativamente engorrosos, a una red local (LAN) operando a 2.5 Gbps, el acceso y la transferencia de archivos pesados se vuelven considerablemente más rápidos y eficientes.

1.4.3. Optimización de costes

Económicamente, el proyecto plantea una transición de un modelo de costes operativos (OPEX), basado en suscripciones mensuales por almacenamiento en la nube que aumentan linealmente con el volumen de datos, a un modelo de gasto de capital (CAPEX). Aunque la inversión inicial en hardware es mayor, el coste por Terabyte a largo plazo es significativamente inferior, amortizándose la inversión en un periodo estimado de 18 a 24 meses frente a servicios equivalentes de nube pública para 16TB de datos.

1.4.4. Privacidad y Seguridad

Al mantener los datos dentro del perímetro físico y lógico controlado, se mitiga la exposición a escaneos de contenido automatizados por parte de proveedores de nube pública para fines de perfilado comercial o entrenamiento de IA. La privacidad se refuerza mediante el cifrado en tránsito (VPN) y el control granular de accesos.

1.4.5. Alineación con los Objetivos de Desarrollo Sostenible (ODS)

Este proyecto genera un impacto positivo directo en los Objetivos de Desarrollo Sostenible de la Agenda 2030 de Naciones Unidas. Específicamente, la solución contribuye al **ODS 7 (Energía asequible y no contaminante)** mediante la consolidación de recursos por virtualización, lo que optimiza el consumo eléctrico, al **ODS 8 (Trabajo decente y crecimiento económico)** por la redirección de la inversión tecnológica hacia el comercio

local y la contratación de mano de obra cualificada regional, al **ODS 9 (Industria, innovación e infraestructura)** mediante la construcción de una infraestructura de red resiliente, y al **ODS 12 (Producción y consumo responsables)** al fundamentar el montaje en hardware reutilizado que fomenta la economía circular. El desarrollo completo de esta matriz de impacto se detalla en el anexo A al final de esta memoria.

1.5 Metodología y Plan de Trabajo

Para el desarrollo de este Trabajo Final de Grado se ha seguido una metodología de ingeniería iterativa e incremental, dividida en cinco fases diferenciadas que abarcan desde la concepción teórica hasta la validación funcional.

1.5.1. Fase 1: Análisis y Definición de Requisitos

En esta etapa inicial se realizó el modelado de las necesidades del entorno familiar (volumen de datos actual y crecimiento proyectado a 3 años). Se evaluaron las limitaciones de la red existente y se definieron los requisitos no funcionales (disponibilidad, seguridad, consumo energético, velocidad y fiabilidad).

1.5.2. Fase 2: Diseño de la Arquitectura y Selección de Hardware

Se procedió al diseño de la topología de red y la arquitectura de almacenamiento. Esta fase incluyó:

- Análisis comparativo de soluciones de hardware y redes que optimicen la relación coste-beneficio para el proyecto. Esto incluye la evaluación de unidades de almacenamiento tipo NAS, tarjetas de red 2.5GbE para hosts, puntos de acceso (AP) y routers con puertos 2.5Gbps que permitan conectar tanto dispositivos cableados como inalámbricos a altas velocidades.
- Diseño lógico de la gestión del almacenamiento y sus sistemas de archivos.
- Planificación del esquema de virtualización (separación de roles entre LXC, Docker y VM).
- Estudio de los distintos servicios y alternativas para el cumplimiento de los objetivos.

1.5.3. Fase 3: Aprovisionamiento y Montaje Físico

Adquisición de los componentes y ensamblaje del servidor, considerando su funcionamiento continuo 24 horas.

1.5.4. Fase 4: Implementación y Configuración del Software

Esta fue la fase más extensa y se subdividió en capas a los efectos de facilitar el manejo de la problemática y la implementación de la solución final:

- **Capa Base:** Instalación de Proxmox VE y configuración de interfaces de red (*bridges*).

- **Capa de Almacenamiento:** Creación de los conjunto de datos ZFS y ajuste de parámetros (*recordsize*, compresión, *atime*).
- **Capa de Servicios:** Despliegue de los contenedores Docker (Nextcloud, Emby) y configuración de la VPN (Tailscale/Wireguard/OpenVPN).
- **Capa de Seguridad:** Configuración del firewall, reglas de acceso y sistema de detección y prevención de intrusos.

1.5.5. Fase 5: Pruebas, Optimización y Documentación

Finalmente, se realizaron pruebas de carga real (transferencia de archivos grandes, streaming simultáneo) para detectar cuellos de botella. Se ajustaron configuraciones para optimizar el rendimiento y se procedió a la redacción de la documentación técnica y de usuario final que conforma la presente memoria.

1.6 Estructura de la memoria

El presente documento se ha organizado siguiendo una estructura lógica que abarca desde la fundamentación teórica hasta la implementación práctica y su posterior validación. A continuación, se detalla brevemente el contenido de los capítulos restantes:

Capítulo 2: Estado del Arte y Marco Teórico. Se realiza una revisión bibliográfica y técnica de las tecnologías clave utilizadas en el proyecto. Se analizan los paradigmas de virtualización (Hipervisores Tipo 1 vs Tipo 2), la contenerización (Docker vs LXC), los sistemas de archivos avanzados (ZFS) y las tendencias actuales en arquitecturas de nube privada y seguridad perimetral.

Capítulo 3: Análisis del Problema y Requisitos. Se formalizan las necesidades del entorno, estableciendo los requisitos funcionales (qué debe hacer el sistema) y no funcionales (rendimiento, seguridad, disponibilidad) que guiarán el diseño de la solución.

Capítulo 4: Diseño de la Solución (Arquitectura). Se describe la arquitectura propuesta a nivel de hardware, topología de red y estructura de almacenamiento. Se justifican las decisiones de diseño tomadas para cumplir con los requisitos planteados anteriormente.

Capítulo 5: Desarrollo e Implementación de la Infraestructura Base. Detalla el proceso técnico de despliegue del hipervisor Proxmox VE, la configuración de ZFS y la preparación del entorno de ejecución para contenedores (LXC y Docker).

Capítulo 6: Despliegue de Servicios y Aplicaciones. Describe la instalación y configuración de los servicios de usuario final, centrándose en la orquestación de Nextcloud para la colaboración y Emby para el contenido multimedia, así como la integración de la conectividad remota mediante servicios VPN.

Capítulo 7: Seguridad Avanzada. Aborda las medidas de seguridad implementadas para proteger la infraestructura, incluyendo la configuración de cortafuegos (OPNsense), sistema de detección y prevención de intrusos (IDS/IPS) y gestión de certificados SSL/TLS.

Capítulo 8: Observabilidad y Alertas. Explica el ecosistema de monitorización basado en Prometheus y Grafana para la visualización de métricas en tiempo real y la gestión automatizada de alertas críticas.

Capítulo 9: Copias de Seguridad. Define la estrategia de continuidad del negocio y protección de datos, detallando el uso de instantáneas ZFS con Sanoid, replicación con Rsync y políticas de respaldo del hipervisor.

Capítulo 10: Estudio Económico y Retorno de Inversión. Presenta un análisis detallado de los costes de capital (CAPEX) y operativos (OPEX), comparándolos con soluciones comerciales para calcular el tiempo de amortización del proyecto.

Capítulo 11: Divulgación Tecnológica y Validación Comunitaria. Expone la validación del proyecto mediante su presentación en un congreso internacional (evento Nerdearla), analizando la retroalimentación recibida por la comunidad técnica.

Capítulo 12: Conclusiones y Líneas Futuras. Recoge las conclusiones finales, el grado de cumplimiento de los objetivos, la relación del trabajo con las competencias de la titulación y las propuestas de mejora para futuras iteraciones.

CAPÍTULO 2
Estado del Arte y Marco Teórico

2.1 Tecnologías de Virtualización

La consolidación de servidores mediante virtualización ha transformado la arquitectura de los centros de datos modernos, permitiendo abstraer el hardware físico para ejecutar múltiples sistemas operativos aislados sobre un mismo host. Para el diseño de la solución objeto de este Trabajo Final de Grado, es importante comprender las diferencias entre las distintas soluciones de mercado disponibles.

2.1.1. Comparativa de Hipervisores de Tipo 1

Un hipervisor de Tipo 1 o *Bare-Metal* se ejecuta directamente sobre el hardware del servidor, gestionando los recursos de CPU, memoria y E/S sin la intermediación de un sistema operativo de propósito general. En el mercado actual para entornos domésticos y PYMES, los dos exponentes principales son VMware ESXi y Proxmox Virtual Environment (VE).

VMware ESXi

Históricamente el estándar de la industria, ESXi utiliza un kernel propietario altamente optimizado basado en GNU/Linux.

- **Ventajas:** Gran estabilidad y ecosistema corporativo maduro.
- **Desventajas:** Su modelo de licenciamiento se ha vuelto restrictivo tras la adquisición por Broadcom, eliminando la versión gratuita (ESXi Free Hypervisor), lo que encarece esta opción enormemente. Además, presenta una estricta lista de compatibilidad de hardware, lo que dificulta su uso en hardware de consumo.

Proxmox VE (KVM + LXC)

Proxmox VE es una plataforma de código abierto basada en la distribución Debian GNU/Linux. No utiliza un kernel propietario, sino el kernel estándar de Linux con soporte KVM (*Kernel-based Virtual Machine*).

- **Arquitectura KVM:** Convierte el kernel de Linux en un hipervisor, permitiendo que el host ejecute máquinas virtuales aprovechando las extensiones de virtualización del procesador como Intel VT-x.

- **Soporte de Hardware:** Al basarse en GNU/Linux, hereda la compatibilidad con una inmensa gama de hardware, incluyendo interfaces de red de 2.5 y 10Gbps, cruciales para este proyecto.
- **Almacenamiento:** Integra soporte nativo para ZFS, permitiendo gestionar un sistema de ficheros con integridad de datos e instantáneas directamente desde el hipervisor, eliminando la necesidad de tarjetas RAID hardware.
- **Contenedores LXC (*Linux Containers*):** Implementa una arquitectura de virtualización a nivel de sistema operativo, posibilitando la ejecución simultánea de múltiples entornos Linux aislados que comparten un único núcleo del sistema anfitrión. Esta técnica suprime la necesidad de emular componentes de soporte físico, lo que reduce drásticamente la sobrecarga computacional en comparación con la virtualización completa.

2.2 Almacenamiento Definido por Software

El almacenamiento en una infraestructura de nube privada moderna trasciende la mera gestión de dispositivos de bloque físicos. El concepto de Almacenamiento Definido por Software (SDS) abstrae la capa de hardware, permitiendo que el software gestione la redundancia, la integridad y el aprovisionamiento. En este proyecto, la elección del sistema de archivos es crítica, ya que determina no solo el rendimiento de lectura/escritura, sino la supervivencia de los datos a largo plazo frente a fenómenos como la degradación silenciosa de datos.

2.2.1. Sistemas de Archivos

Análisis de ZFS

ZFS no es simplemente un sistema de archivos, sino un gestor de volúmenes lógicos y un sistema de archivos combinados. Su arquitectura se basa en el principio de copia en escritura (Copy-on-Write, 'CoW').

- **Integridad de Datos:** A diferencia de los sistemas tradicionales, ZFS utiliza un árbol de Merkle donde cada bloque de datos tiene una suma de control (checksum) almacenado en el puntero de su bloque padre. Esto permite detectar y reparar automáticamente (si hay redundancia) la corrupción silenciosa de datos durante la lectura.
- **Instantáneas y Clones:** Gracias al modelo copia en escritura, las instantáneas (snapshots) son operaciones de coste casi nulo. El mecanismo CoW implica que los bloques de datos no se sobrescriben directamente; cuando un archivo se modifica, el sistema escribe los cambios en nuevos bloques y conserva los originales intactos. De este modo, las instantáneas no copian físicamente la información, sino que "congelan" las referencias a los bloques existentes en un momento determinado. Esto permite crear puntos de restauración instantáneos antes de realizar cambios críticos en el sistema, como actualizaciones.
- **Consumo de Recursos (Memoria RAM):** El sistema ZFS requiere una asignación significativa de memoria principal para operar con una eficiencia óptima. Esto se debe, principalmente, a la gestión de la **Caché de Reemplazo Adaptativo** (*Adaptive Replacement Cache* o ARC), un mecanismo avanzado que utiliza la memoria RAM

como nivel de almacenamiento intermedio para agilizar las operaciones de lectura y escritura. Como regla general, se recomienda disponer de al menos 1 GB de RAM por cada TB de almacenamiento, lo que puede suponer una limitación en entornos con hardware restringido.

- **Limitaciones de Portabilidad y Dependencia de Terceros:** A diferencia de sistemas como EXT4 o XFS, ZFS no está integrado de forma nativa en el kernel de Linux debido a incompatibilidades de licencias. Esto implica que su uso depende de la implementación *ZFS on Linux* (ZoL). Ante un fallo crítico del sistema operativo anfitrión, la recuperación de los datos no es inmediata: requiere la instalación previa de módulos específicos y la importación manual del sistema de ficheros. Además, la falta de soporte nativo en sistemas Windows o macOS dificulta enormemente la migración de datos o el acceso a los mismos desde entornos externos.

Análisis de EXT4

EXT4 es la evolución del sistema de archivos estándar de Linux. Es un sistema de archivos con registro por diario (journaling).

- **Ventajas:** Es extremadamente maduro, robusto y universal. Requiere muy poca sobrecarga de CPU y RAM en comparación con ZFS. Es ideal para situaciones donde la recuperación de datos mediante herramientas estándar es prioritaria o donde los recursos de hardware son limitados.

- **Limitaciones:** Carece de sumas de control para los datos (solo para metadatos), por lo que no puede detectar corrupción silenciosa en el contenido de los archivos. Las instantáneas no son nativas y dependen de capas externas como LVM (Logical Volume Manager), siendo menos eficientes que en ZFS.

Análisis de Btrfs (B-Tree Filesystem)

Considerado a menudo como la alternativa moderna a ZFS en el ecosistema puro de Linux [19].

- **Similitudes con ZFS:** También opera bajo el modelo CoW, soporta instantáneas atómicas, sumas de control y gestión de volúmenes (RAID software).

- **Diferencias clave:** Aunque ofrece mayor flexibilidad para reconvertir arrays RAID en caliente, su implementación de RAID 5/6 ha sufrido históricamente de problemas de estabilidad. En el contexto de Proxmox VE, ZFS tiene un enfoque principal, mientras que Btrfs tiene un soporte secundario.

2.3 Plataformas de Nube Privada

La elección del software de orquestación de la nube privada es determinante para la experiencia de usuario final. Este componente actúa como el frontend que unifica el acceso a los archivos y la sincronización entre dispositivos.

2.3.1. Análisis de Nextcloud frente a Owncloud

Históricamente, Owncloud fue la solución pionera en el mercado de nubes privadas de código abierto. Sin embargo, en 2016 se produjo una bifurcación (fork) del proyecto original dando lugar a Nextcloud. Para este TFG se ha evaluado ambas opciones:

Owncloud

Mantiene un modelo de negocio conocido como Open Core. Esto significa que, aunque el núcleo es de código abierto, ciertas funcionalidades empresariales críticas (como auditorías avanzadas, soporte para almacenamiento de objetos o ciertas integraciones de seguridad) están restringidas a la versión Enterprise bajo licencia comercial. Su desarrollo comunitario ha perdido velocidad en favor de la versión reescrita en Go (Owncloud Infinite Scale), que aún presenta inmadurez para entornos de producción críticos.

Nextcloud

Surgió como respuesta a las limitaciones del modelo Open Core.

- **Modelo 100 % de código abierto:** Todas las funcionalidades, incluidas las de seguridad (protección contra fuerza bruta, autenticación de dos factores, entre otros) y gestión de archivos (bloqueo de archivos, control de versiones, entre otros), están disponibles en la versión comunitaria sin coste de licencia.
- **Ecosistema de Aplicaciones:** Dispone de una arquitectura modular que permite instalar complementos desarrollados por la comunidad para agregar funcionalidad a la plataforma.
- **Seguridad:** Implementa una serie de medidas de aseguramiento, como cabeceras de seguridad HTTP estrictas y protección contra ransomware mediante la detección de anomalías en la escritura de archivos.

2.3.2. Análisis frente a Soluciones Propietarias (SaaS)

Comparado con servicios como Google Drive o Dropbox, Owncloud o Nextcloud ofrecen:

- **Privacidad:** Los datos no son analizados para perfilado publicitario ni entrenamiento de modelos de IA.
- **Rendimiento Local:** La transferencia de archivos grandes en la red local (LAN) no consume ancho de banda de internet y opera a velocidad 2.5 Gbps, eliminando la latencia de la nube pública.
- **Coste:** Eliminación de cuotas mensuales recurrentes por almacenamiento adicional.

2.4 Plataformas de Transmisión Multimedia

Para la gestión integral y el consumo de la biblioteca audiovisual, que abarca tanto producciones de vídeo de gran volumen como extensas galerías fotográficas, se requiere una solución centralizada capaz de catalogar el contenido, respetar la jerarquía de los proyectos y realizar transcodificación en tiempo real para adaptar el formato al dispositivo cliente y al ancho de banda disponible.

2.4.1. Análisis de Alternativas: Plex, Jellyfin, Immich y Emby

El ecosistema actual ofrece diversas herramientas para la orquestación multimedia. Se han evaluado cuatro de las soluciones más relevantes del mercado, abarcando opciones de código abierto y privativas, para determinar cuál resuelve de manera más eficiente los flujos de trabajo requeridos, especialmente en dispositivos móviles (iOS).

Jellyfin

Nacido como una bifurcación (fork) de Emby cuando este adoptó un modelo de código cerrado, Jellyfin es la alternativa libre más reconocida.

- **Ventajas:** Es software libre (licencia GPL), carece de muros de pago y garantiza privacidad total al no requerir autenticación contra servidores externos.
- **Desventajas:** Su interfaz de usuario y la experiencia general (UI/UX) carecen del nivel de pulido de sus competidores comerciales. Esta carencia es especialmente crítica en su aplicación nativa para el ecosistema iOS, lo que genera fricción y dificulta la navegación fluida para usuarios no técnicos.

Plex

Históricamente la plataforma líder y la más extendida en el ámbito doméstico, operando bajo un modelo Freemium.

- **Ventajas:** Amplia compatibilidad de clientes nativos en el mercado de Smart TVs y un motor de transcodificación altamente maduro.
- **Desventajas:** Aunque su núcleo es propietario y de pago, su mayor deficiencia actual reside en la gestión de imágenes. Tras el lanzamiento de su nueva aplicación dedicada, Plex Photos, la plataforma ha sufrido una regresión funcional significativa, careciendo de múltiples herramientas básicas de organización que sí están presentes en la competencia.

Immich

Es una solución de código abierto de muy rápido desarrollo, diseñada específicamente como una alternativa autoalojada y de alto rendimiento a Google Photos.

- **Ventajas:** Interfaz moderna, capacidades excepcionales de reconocimiento facial mediante inteligencia artificial y carga ultrarrápida de copias de seguridad desde terminales móviles.
- **Desventajas:** Su estructura de base de datos impone una organización plana. Carece de soporte para álbumes anidados o jerarquías complejas. Para el caso de uso de este proyecto, donde el material audiovisual se organiza en múltiples subcarpetas de proyectos, esta limitación arquitectónica hace que la gestión del contenido sea inmanejable.

Emby

Solución a partir de la cual se originó Jellyfin, operando actualmente bajo un modelo de código cerrado y licencia comercial (Emby Premiere) para habilitar todas sus capacidades.

- **Ventajas:** Ofrece un rendimiento superior, con tiempos de respuesta y carga de bibliotecas notablemente más rápidos que Plex. Su interfaz de usuario, particularmente en dispositivos iOS, está altamente refinada. Adicionalmente, gestiona con solvencia tanto bibliotecas de vídeo como estructuras complejas de directorios fotográficos.
- **Desventajas:** Requiere el pago de una suscripción mensual o licencia vitalicia para desbloquear funciones clave como la transcodificación por hardware en servidores Linux.

2.5 Arquitectura de Seguridad

La exposición de servicios a internet conlleva riesgos inherentes. El modelo de seguridad perimetral tradicional basado únicamente en el redireccionamiento de puertos (Port Forwarding) es insuficiente ante el panorama actual de amenazas automatizadas.

2.5.1. Estrategia de Defensa en Profundidad

La Defensa en Profundidad es un modelo estratégico de seguridad informática que consiste en la implementación de múltiples capas de controles defensivos de manera redundante. Su premisa fundamental es que ningún mecanismo de protección es infalible por sí solo; por ello, se estructuran diversas barreras como: físicas, lógicas y administrativas para que, ante el compromiso de una de ellas, las capas subsecuentes logren detectar, retardar o neutralizar la amenaza.

En el contexto de una infraestructura virtualizada, este enfoque se fundamenta en tres pilares teóricos:

- **Seguridad Multicapa:** Disposición de barreras secuenciales donde el **cortafuegos** y los **sistemas de prevención de intrusos (IPS)** constituyen la primera línea de defensa perimetral. Su función es filtrar el tráfico y neutralizar amenazas antes de que alcancen los niveles internos; si esta capa fuera superada, el atacante se enfrentaría inmediatamente a controles adicionales, evitando que un fallo único comprometa la integridad de los datos.
- **Mínimo Privilegio:** Restricción de facultades y accesos al nivel estrictamente necesario para la operación de cada servicio o usuario, minimizando así la superficie de exposición.
- **Segmentación y Aislamiento:** Fragmentación de la arquitectura en zonas independientes para contener posibles brechas de seguridad y evitar el movimiento lateral de agentes maliciosos dentro del sistema.

Cortafuegos de Nueva Generación (NGFW)

En las arquitecturas de red modernas, los Cortafuegos de Nueva Generación (NGFW, por sus siglas en inglés) representan la evolución natural de los cortafuegos tradicionales de filtrado de paquetes. Mientras que un sistema clásico se limita a inspeccionar las cabeceras de red y transporte (direcciones IP y puertos), un NGFW integra capacidades de inspección profunda de paquetes (*Deep Packet Inspection* o DPI). Esta evolución proporciona un ecosistema de seguridad modular que unifica el filtrado de estado, la prevención de intrusiones y la inteligencia frente a amenazas.

Dentro del paradigma del software de código abierto, existen diversas plataformas de enrutamiento y seguridad perimetral, destacando históricamente dos soluciones basadas en sistemas operativos de la familia BSD: **pfSense** y **OPNsense**.

Análisis Comparativo: pfSense y OPNsense

Ambas plataformas comparten un origen común, siendo OPNsense una bifurcación directa de pfSense [27], pero han evolucionado con filosofías de diseño y arquitecturas divergentes:

- **pfSense:**
  - **Ventajas:** Cuenta con una trayectoria consolidada, lo que se traduce en una base de usuarios inmensa, documentación exhaustiva y un alto nivel de estabilidad probada en entornos empresariales durante años.
  - **Desventajas:** Su código base y arquitectura subyacente son más veteranos. El ciclo de actualizaciones es menos predecible y la integración de ciertas herramientas de terceros puede resultar menos orgánica frente a soluciones de diseño más reciente.
- **OPNsense:**
  - **Ventajas:** Adopta una arquitectura de software moderna basada en el patrón **MVC** (*Modelo-Vista-Controlador*), lo que facilita la auditoría del código y el desarrollo de interfaces modulares. Además, se fundamenta en **HardenedBSD**, una variante del sistema operativo centrada en la seguridad que incorpora protecciones a nivel de núcleo, como ASLR (*Address Space Layout Randomization*), para mitigar técnicas de explotación de memoria. Posee un ciclo de actualizaciones estructurado y predecible.
  - **Desventajas:** Al ser un producto más reciente, su comunidad de usuarios y el volumen de documentación generada por terceros, aunque en constante crecimiento, es numéricamente menor que la de su predecesor.

Mecanismos de Seguridad Avanzada: IDS/IPS e Inteligencia Colectiva

La eficacia de un NGFW radica en su capacidad para integrar motores de seguridad dinámicos que superen el simple bloqueo por puertos o direcciones estáticas. Entre las tecnologías más destacadas en este ámbito se encuentran:

- **Sistemas de Detección y Prevención de Intrusiones (IDS/IPS):** A diferencia de un cortafuegos tradicional que fundamenta sus decisiones de bloqueo en reglas estáticas basadas en puertos y direcciones IP, los sistemas IDS/IPS implementan técnicas de Inspección Profunda de Paquetes (*Deep Packet Inspection* o DPI). Estos motores analizan el flujo continuo de datos y la carga útil (*payload*) de cada paquete en tiempo real. Mediante la comparación con bases de firmas actualizadas y el análisis heurístico, permiten identificar patrones anómalos, tráfico de programas maliciosos (*malware*) y tentativas de vulneración de la red. Mientras que un IDS opera de forma pasiva, limitándose a registrar y alertar sobre anomalías, un IPS actúa en línea (*inline*) con la capacidad de bloquear activamente y descartar el tráfico malicioso.

  Dentro del ecosistema del software de código abierto, la industria se apoya principalmente en dos motores de análisis:

  - **Snort:** Constituye el estándar histórico y el motor más veterano en la detección de intrusiones. Su extensa trayectoria le otorga una base de datos de firmas inmensa y un respaldo documental masivo. No obstante, su arquitectura clásica se diseñó bajo un modelo monohilo (*single-threaded*), lo que tradicionalmente ha limitado su capacidad para escalar y aprovechar el procesamiento paralelo en arquitecturas de hardware modernas, generando posibles cuellos de botella en redes de alta velocidad.
  - **Suricata:** Representa la evolución hacia los motores de nueva generación. Su principal ventaja arquitectónica frente a Snort radica en su diseño nativamente multihilo. Esta característica le permite distribuir la carga computacional del análisis de paquetes a través de múltiples núcleos del procesador de forma simultánea. En consecuencia, Suricata ofrece un rendimiento significativamente superior y menor latencia en infraestructuras virtualizadas y enlaces de gran ancho de banda, sumado a una integración nativa de funciones avanzadas como la extracción automática de archivos en tránsito para su posterior análisis.

- **Inteligencia Colectiva:** Complementando el análisis basado en firmas, plataformas como CrowdSec aportan un modelo de seguridad colaborativa. Este sistema analiza los registros de actividad locales para identificar comportamientos hostiles (como escaneos de puertos automatizados). La innovación reside en su red comunitaria: cuando un nodo detecta una IP maliciosa, esta información se anonimiza y se comparte con una base de datos global, permitiendo que el resto de las infraestructuras conectadas bloqueen preventivamente esa dirección, generando un efecto de inmunidad a escala global.

2.5.2. Tecnologías de Conectividad Segura y Redes Privadas Virtuales

En el diseño de infraestructuras de red modernas, la implementación de Redes Privadas Virtuales (VPN) constituye un pilar fundamental para garantizar la confidencialidad, integridad y disponibilidad del acceso remoto. Las arquitecturas de alta resiliencia no dependen de un único protocolo, sino que adoptan estrategias de conectividad híbrida y redundante. Este enfoque asegura que, ante la degradación de un servicio, bloqueos de red o fallos de software, existan vías alternativas para mantener la comunicación.

Dentro del estado del arte de las comunicaciones seguras, destacan cuatro protocolos y arquitecturas principales, cada uno con características criptográficas y operativas diferenciadas:

- **L2TP/IPsec (Protocolo de Túnel de Capa 2):** Constituye un estándar clásico y robusto en redes corporativas. L2TP se encarga de encapsular las tramas a nivel de enlace de datos, pero, al carecer de mecanismos de cifrado propios, se implementa de forma conjunta con **IPsec** (*Internet Protocol Security*) para proveer autenticación y encriptación. Su ventaja más destacada es la integración nativa, la inmensa mayoría de los sistemas operativos clientes incluyen soporte por defecto, eliminando la necesidad de desplegar software de terceros. Como contrapartida, su diseño basado en una doble encapsulación genera una mayor sobrecarga de red, lo que penaliza el rendimiento general.

- **OpenVPN:** Es uno de los protocolos más maduros, auditados y extendidos de la industria, fundamentado en los estándares de seguridad SSL/TLS. Su principal fortaleza reside en la alta granularidad de su configuración criptográfica, lo que permite a los administradores seleccionar algoritmos de cifrado e intercambio de claves

específicos según las normativas exigidas. Además, destaca por su robusta integración con infraestructuras de autenticación corporativas (como directorios LDAP o servidores RADIUS). No obstante, su desventaja inherente frente a alternativas de nueva generación, como WireGuard, es la significativa sobrecarga computacional derivada de su extensa base de código, lo que penaliza el rendimiento máximo de transferencia e incrementa la latencia.

- **WireGuard:** Representa el estándar contemporáneo en la creación de túneles criptográficos. Su diseño minimalista y su capacidad para ejecutarse directamente en el espacio del núcleo (*kernel space*) le otorgan un rendimiento superior y una latencia mínima. Al prescindir de negociaciones de estado complejas y utilizar sistemas criptográficos de última generación (como ChaCha20 y Poly1305), ofrece una conexión extremadamente rápida y un consumo de recursos notablemente inferior al de sus predecesores como OpenVPN.

- **Arquitecturas de Red en Malla (*Mesh VPN*):** Evolucionan el modelo de VPN centralizada tradicional hacia una topología descentralizada de nodo a nodo (*Peer-to-Peer* o P2P). Soluciones modernas en esta categoría (como Tailscale o ZeroTier) abstraen la complejidad criptográfica y utilizan técnicas avanzadas de penetración de NAT (*NAT Traversal*) apoyadas en protocolos como STUN y TURN. Esto permite establecer túneles cifrados directos entre nodos sin requerir la apertura explícita de puertos en el cortafuegos perimetral, facilitando un aprovisionamiento ágil y completamente desacoplado del hardware de enrutamiento.

2.5.3. Observabilidad y Monitorización de Sistemas

La observabilidad y la monitorización constituyen pilares fundamentales en la administración proactiva de infraestructuras informáticas. Mientras que la monitorización tradicional se centra en alertar sobre el estado de un componente (determinar *cuándo* ocurre una anomalía), la observabilidad proporciona una comprensión profunda de las operaciones internas mediante la recolección y el análisis continuo de métricas, eventos y registros, permitiendo diagnosticar *por qué* ocurre un fallo.

Dentro del estado del arte de las plataformas de recolección de métricas, existen diferentes paradigmas arquitectónicos. Dos de las soluciones más representativas en el ámbito del código abierto, que ilustran la evolución de estos sistemas, son **Zabbix** y el ecosistema modular encabezado por **Prometheus**.

Evolución Arquitectónica: Modelos Centralizados frente a Modulares

Ambas tecnologías abordan la gestión de la información desde filosofías de diseño distintas:

- **Arquitectura Monolítica y Estática (Zabbix):** Representa el enfoque tradicional y consolidado. Utiliza un motor centralizado que apoya su almacenamiento en bases de datos relacionales clásicas (SQL). Su recolección se basa, en gran medida, en el despliegue de agentes locales y en el uso de protocolos estándar de gestión de red como SNMP. Es una arquitectura diseñada para ofrecer máxima fiabilidad en infraestructuras de red estáticas y servidores físicos persistentes.

- **Arquitectura Distribuida (Ecosistema Prometheus):** Surge como respuesta a las necesidades de los entornos de contenedores y microservicios. En lugar de un modelo relacional, implementa una **Base de Datos de Series Temporales** (TSDB), optimizada específicamente para registrar altos volúmenes de valores numéricos in-

dexados por marcas de tiempo y etiquetas multidimensionales. Su diseño es inherentemente modular: disocia la recolección de datos, la gestión de alertas (mediante componentes como **Alertmanager**) y la representación gráfica (habitualmente delegada a plataformas de visualización avanzada como **Grafana**).

Análisis Comparativo de Componentes Clave

Para comprender el alcance de cada solución, resulta necesario evaluar cómo abordan los procesos fundamentales del ciclo de vida de las métricas:

Característica	**Modelo Tradicional (ej. Zabbix)**	**Arquitectura Distribuida (ej. Prometheus + Grafana)**
Modelo de almacenamiento	Bases de datos relacionales (SQL). Almacenamiento estructurado en tablas.	Base de datos de series temporales (TSDB) con categorización por etiquetas.
Paradigma de recolección	Modelo híbrido con predominancia de inserción (*Push*) desde agentes locales y sondeo SNMP.	Modelo estricto de extracción (*Pull*) mediante interrogación periódica a exportadores HTTP.
Representación visual	Consola web integrada en el núcleo de la aplicación, unificando gestión y gráficos.	Arquitectura desacoplada; delega la visualización a herramientas especializadas (Grafana).
Gestión de la configuración	Interfaz gráfica centralizada; la configuración reside en la base de datos principal.	Basada en archivos de texto declarativos (YAML), facilitando la *monitorización como código*.
Escenario de uso óptimo	Topologías de red predecibles, hardware físico e infraestructuras monolíticas persistentes.	Arquitecturas altamente dinámicas, contenedores y servicios que escalan de forma automatizada.

Tabla 2.1: Comparativa arquitectónica entre sistemas de monitorización tradicionales y nativos en nube.

CAPÍTULO 3
Análisis del Problema y Requisitos

Tras definir el contexto, la problemática existente y evaluar las tecnologías disponibles en el estado del arte, en este capítulo se formalizan las necesidades del proyecto. La transición de un flujo de trabajo fragmentado a una infraestructura centralizada requiere la definición precisa de los requisitos del sistema. Estos se dividen en funcionales y no funcionales, y servirán como base para el diseño arquitectónico de la solución.

3.1 Especificación de Requisitos Funcionales

Los requisitos funcionales describen las capacidades y los servicios específicos que la infraestructura debe proveer a los usuarios finales (el entorno familiar, los editores y los clientes externos).

- **RF1 - Repositorio Histórico Centralizado:** El sistema debe proveer un volumen de almacenamiento masivo unificado que actúe como repositorio histórico para las fotografías y vídeos familiares de los últimos 20 años. Este volumen debe ser la fuente única para el archivo a largo plazo.

- **RF2 - Acceso Ubicuo y Multiplataforma:** El sistema debe permitir visualizar todo el contenido histórico, así como a los editores acceder a los materiales audiovisuales nuevos, directamente desde sus teléfonos móviles (iOS y Android) y ordenadores.

- **RF3 - Espacios de Trabajo Personales Aislados:** El servidor debe aprovisionar carpetas de red privadas para cada usuario, ubicadas en una unidad de almacenamiento independiente al histórico. Estos espacios se utilizarán para alojar proyectos en curso y archivos personales, permitiendo trabajar fuera del almacenamiento local de cada ordenador.

- **RF4 - Gestión de Identidades y Control de Acceso Estricto:** Se debe implementar un sistema de permisos basado en usuarios. El requisito crítico es garantizar la privacidad: ningún usuario debe tener visibilidad, permisos de lectura o de escritura sobre las carpetas personales y los proyectos en curso de los demás usuarios del sistema.

- **RF5 - Intercambio Externo y Colaboración:** El sistema debe incorporar una herramienta que permita generar enlaces públicos de descarga o subida para compartir el trabajo final (entregables) con terceros y clientes, eliminando la dependencia de servicios comerciales y evitando los límites de tamaño en los adjuntos.

- **RF6 - Acceso Remoto Seguro (VPN):** Los usuarios autorizados deben poder acceder al repositorio fotográfico y a sus carpetas de trabajo desde cualquier ubicación externa a la red local. Este acceso debe realizarse de forma cifrada mediante una Red Privada Virtual, garantizando que el tráfico no pueda ser interceptado.

- **RF7 - Estación de Trabajo Virtualizada para la gestión de los activos almacenados:** El sistema debe proveer una máquina virtual equipada con un entorno de escritorio completo. Esta estación tiene como finalidad permitir a los usuarios, una vez conectados desde el exterior mediante la Red Privada Virtual (VPN), operar directamente sobre el almacenamiento centralizado como si se encontraran físicamente en la red local. Esto garantiza que las tareas de organización, traslado y administración masiva del archivo histórico y audiovisual se ejecuten con latencia mínima dentro de la propia infraestructura. Este entorno estará dedicado de manera estricta a la gestión de ficheros, excluyendo de forma explícita la ejecución de cargas de trabajo computacionalmente intensivas, tales como la edición o el renderizado multimedia.

3.2 Especificación de Requisitos No Funcionales

Los requisitos no funcionales definen los atributos de calidad, rendimiento y restricciones tecnológicas que la infraestructura debe cumplir para que la experiencia de usuario sea óptima.

- **RNF1 - Rendimiento de Red Local:** Para soportar el flujo de trabajo de edición de vídeo, la infraestructura de red debe operar a una velocidad mínima de 2.5 Gbps. Esto es necesario para evitar cuellos de botella en la transferencia de archivos de gran tamaño entre los ordenadores de edición y el almacenamiento del NAS.

- **RNF2 - Seguridad Perimetral y Mitigación de Riesgos:** La infraestructura no debe exponer interfaces ni puertos de administración a la Internet pública. Todo flujo de red entrante (limitado de forma exclusiva a los enlaces de intercambio de archivos con terceros y al acceso remoto a la plataforma de transmisión multimedia) debe ser canalizado e inspeccionado a través de los sistemas de defensa perimetral.

- **RNF3 - Escalabilidad del Almacenamiento:** La arquitectura de almacenamiento definida por software debe ser lo suficientemente flexible como para permitir la adición futura de nuevos discos duros.

- **RNF4 - Observabilidad y Alertas Proactivas:** El sistema debe contar con un mecanismo de monitorización continua del estado de salud del hardware y los recursos lógicos. Se requiere la configuración y envío automatizado de alertas tempranas ante eventos críticos que puedan comprometer la integridad o el rendimiento del servidor. Esto engloba la detección de umbrales anómalos de temperatura (en la CPU, unidades de almacenamiento y tarjetas de red), la identificación de fallos de hardware (como errores S.M.A.R.T. o degradación del sistema de ficheros ZFS) y advertencias preventivas por escasez de espacio de almacenamiento disponible.

- **RNF5 - Eficiencia Económica y Reducción de Costes:** El sistema debe minimizar los gastos operativos recurrentes mediante la adopción prioritaria de tecnologías de código abierto e infraestructuras locales. Un pilar fundamental de este ahorro es la eliminación de las cuotas mensuales por almacenamiento en proveedores externos, evitando el sobrecoste injustificado de alquilar capacidad externa cuando

ya se dispone de los recursos físicos en la propia red. Aunque se favorece fuertemente la elección de software libre de licenciamiento para prescindir del resto de suscripciones costosas, este criterio no posee un carácter estrictamente excluyente. Se contempla la adopción puntual de software privativo o soluciones con costes de licencia reducidos, siempre y cuando aporten un valor técnico o de usabilidad superior que justifique la inversión.

- **RNF6 - Integridad de Datos en Entornos Concurrentes:** La infraestructura debe soportar el flujo de trabajo simultáneo de varios usuarios. Los protocolos de red y los sistemas implementados deben gestionar la concurrencia, previniendo conflictos de acceso y garantizando que la edición simultánea en los volúmenes compartidos no resulte en la sobrescritura accidental ni en la pérdida del trabajo realizado por otros miembros.
- **RNF7 - Privacidad de Datos:** El proyecto debe asegurar el control absoluto sobre los activos digitales, donde estos deben residir de forma exclusiva en los soportes de almacenamiento físico dentro de la infraestructura local. Queda estrictamente excluida cualquier sincronización, volcado o delegación del alojamiento hacia servidores de terceros o proveedores de nube pública, garantizando así la privacidad total y evitando su exposición y entrenamiento de modelos de inteligencia artificial.
- **RNF8 - Operatividad Continua:** La infraestructura debe ser diseñada para funcionar 24 horas, proveyendo acceso a los servicios. Sin embargo, este enfoque debe buscar una mejora de la disponibilidad a un coste reducido. Por consiguiente, ante averías de los componentes físicos o cortes eléctricos, se debe asumir la posible interrupción temporal del sistema.

CAPÍTULO 4
Diseño de la Solución (Arquitectura)

En este capítulo se detalla la arquitectura técnica de la solución implementada. Se describe la selección y dimensionamiento del hardware, el software utilizado para cumplir con los distintos requisitos funcionales, la topología de la red, la estructura lógica del almacenamiento y la estrategia de seguridad en profundidad que protege la infraestructura.

4.1 Arquitectura de Hardware

El dimensionamiento del hardware del servidor se abordó bajo una premisa de eficiencia económica y sostenibilidad tecnológica. Aprovechando la experiencia profesional previa en soporte técnico, gran parte de los componentes base se reciclaron de equipos anteriores (reaprovechamiento de activos), minimizando el gasto de capital (CAPEX). Las inversiones estratégicas se limitaron a aquellos componentes estrictamente necesarios para cumplir con los Requisitos No Funcionales (RNF) de almacenamiento y ancho de banda.

4.1.1. Procesamiento, Memoria y Almacenamiento del servidor

El núcleo computacional está compuesto por los siguientes elementos:

- **Procesador:** Intel Core i5-9400F (6 núcleos físicos, 6 hilos, sin gráfica integrada). Este procesador de novena generación ofrece un equilibrio excelente entre bajo consumo energético en reposo y potencia de cálculo.
- **Memoria RAM:** 40 GB DDR4. Una capacidad elevada y poco habitual, fruto de la combinación de módulos preexistentes de 8GB y 32GB de la marca Corsair. Esta cantidad resulta especialmente adecuada para un servidor ya que permite asignar recursos de forma holgada a contenedores y máquinas virtuales sin incurrir en paginación.
- **Almacenamiento:** El servidor dispone de 4 medios de almacenamiento (2 SSD y 2 HDD), organizados según su función para optimizar rendimiento y fiabilidad:
    - **SSD 500 GB (Western Digital Blue):** Unidad de arranque. Aloja el sistema principal y destaca por su buena calidad y fiabilidad, lo que lo hace adecuado para garantizar estabilidad en el sistema anfitrión.

- **SSD 1 TB (Kingston A400):** Unidad con buena velocidad destinada a la ejecución de máquinas virtuales y contenedores, aprovechando la baja latencia y rapidez de acceso propias de los SSD.
- **HDD 12 TB y 3 TB (Western Digital Red):** Unidades de almacenamiento de mayor capacidad. Al ser discos orientados a entornos NAS, está optimizado para funcionamiento continuo, alta durabilidad y gestión eficiente de grandes volúmenes de datos.

4.1.2. Conectividad de Red (NIC)

Para cumplir con el RNF1 (Rendimiento de red a 2.5 Gbps), el puerto Gigabit Ethernet integrado en la placa base resultaba un cuello de botella inaceptable. Por ello, se adquirió e instaló una tarjeta de red dedicada **TP-Link TX401 (10 GbE)**. La justificación de esta compra es doble: permite utilizar por completo la red local actual de 2.5 Gbps durante la transferencia de material audiovisual pesado, y deja margen de mejora para una eventual actualización del conmutador a 10 Gbps sin necesidad de intervenir el servidor.

4.1.3. Alimentación, Chasis y Gestión Térmica

A diferencia de un equipo de escritorio convencional, un servidor requiere un tratamiento riguroso de la energía, la temperatura y el polvo debido a su funcionamiento ininterrumpido.

- **Fuente de Alimentación (PSU):** EVGA SuperNOVA 1000 G5 (1000W, certificación 80 PLUS Gold). Aunque su capacidad máxima excede el consumo real del equipo, la reutilización de este componente de alta gama aporta beneficios críticos a la infraestructura:
  - **Operación Pasiva:** Al trabajar con una carga inferior al 30 %, la fuente activa su *ECO Mode*. El ventilador permanece apagado, eliminando el ruido y evitando la succión de polvo.
  - **Fiabilidad y Flujo de Aire:** Integra protecciones eléctricas de hardware completas para salvaguardar los distintos componentes y, por ende, los datos:
    - **Protección contra sobretensión (OVP):** Evita que el voltaje supere niveles seguros, protegiendo los componentes frente a picos eléctricos.
    - **Protección contra subtensión (UVP):** Apaga o bloquea el sistema cuando el voltaje cae por debajo de un umbral seguro, evitando fallos de funcionamiento.
    - **Protección contra sobrecorriente (OCP):** Limita la corriente en cada línea para prevenir daños en los circuitos por exceso de intensidad.
    - **Protección contra sobrepotencia (OPP):** Desconecta la fuente si la potencia total excede su capacidad nominal.
    - **Protección contra cortocircuitos (SCP):** Detecta cortocircuitos y corta la alimentación de inmediato para evitar daños graves.
    - **Protección contra sobretemperatura (OTP):** Supervisa la temperatura interna y apaga el sistema si se alcanzan los niveles máximos.

  Además, su diseño totalmente modular permite instalar solo los cables necesarios, eliminando obstáculos físicos y favoreciendo la presión estática del chasis.

- **Sistema de Alimentación Ininterrumpida (UPS):** Para mitigar los riesgos asociados a la inestabilidad de la red eléctrica, se ha integrado un equipo Lyonn CTB de 800VA con estabilizador de tensión incorporado. Su función principal es el aprovisionamiento de energía al servidor frente a fallos de suministro:
  - **Apagado Controlado (Graceful Shutdown):** El hipervisor monitoriza de forma continua el estado del UPS. Ante un corte eléctrico, se detecta el cambio al modo de batería y si este estado de alerta se mantiene durante un umbral definido de X segundos sin recuperar la energía, el hipervisor inicia un procedimiento de apagado de forma controlada. Esto garantiza que las transacciones en memoria se escriban en los discos y no queden datos u operaciones inconsistentes.
  - **Recuperación Automatizada:** Al restablecerse el suministro de la red, el UPS reanuda la alimentación. Gracias a la directiva *Restaurar tras pérdida de energía de CA* configurada en la UEFI de la placa base, el servidor arranca automáticamente sin intervención manual, iniciando todos los servicios de forma desatendida.
- **Chasis:** Thermaltake Chaser MK1. Se adquirió este gabinete de formato gran torre por su amplia capacidad de expansión (6 bahías para discos duros) y sus opciones avanzadas de ventilación, factores que suelen ser limitantes en gabinetes estándar.

Estrategia de Refrigeración: Presión Positiva

Se ha implementado un diseño térmico poco convencional centrado en la **prevención del ingreso de polvo** por encima de la disipación térmica. Al no disponer de una tarjeta gráfica dedicada (gran generador de calor) y contar con un disipador de CPU de alto rendimiento (Cooler Master Hyper 212 EVO de doble flujo con dos ventiladores), el calor interno generado es moderado.

La configuración del flujo de aire del chasis utiliza **alta presión positiva** [17]:

1. Se han instalado dos ventiladores de entrada de gran caudal: un ventilador de 200 mm en la parte superior y otro en el panel frontal, ambos forzando el ingreso de aire fresco hacia los discos duros, la placa base y el disipador del procesador.

2. Ambos ventiladores de ingreso cuentan con filtros antipolvo de malla fina. Este tejido actúa como una barrera física que captura partículas de polvo antes de que entren al chasis. Al mantener los componentes internos limpios, se previene la acumulación de calor y se extiende la vida útil del servidor.

3. Al contar con dos ventiladores de gran tamaño dedicados exclusivamente a la inyección de aire y omitir por completo el uso de ventiladores de extracción activa, se introduce un volumen de aire que no tiene una vía de escape mecánica. Esto genera una presión estática interna mayor que la presión atmosférica del entorno. Como resultado, el aire se ve forzado a escapar de forma pasiva por todas las rendijas y paneles perforados del gabinete, impidiendo físicamente que el polvo ingrese por estas aberturas carentes de filtro.

Aunque la entrada de aire por la parte superior va en contra de la tendencia natural del calor, que tiende a subir, la presión estática generada por el ventilador de 200 mm vence la convección térmica, asegurando un entorno libre de polvo que prolonga significativamente la vida útil de los componentes.

4.2 Estrategia de Virtualización y Orquestación

Para la implementación de este proyecto, se ha adoptado una arquitectura de virtualización híbrida apoyada en el hipervisor **Proxmox VE**. Esta elección de código abierto se fundamenta en las siguientes razones técnicas y filosóficas:

1. **Independencia tecnológica:** Evita la dependencia de un proveedor específico y elimina los costes de licencia recurrentes asociados a soluciones privativas como VMware ESXi.
2. **Flexibilidad híbrida:** Es la única solución del mercado que permite orquestar Máquinas Virtuales (KVM) y Contenedores Linux (LXC) desde una misma interfaz unificada, facilitando la gestión integral del nodo.
3. **Madurez y compatibilidad de hardware:** Al estar basado en la distribución **Debian**, hereda una estabilidad excepcional y un soporte nativo para el sistema de ficheros **ZFS**. Esta base sólida, junto con la inclusión de una amplia gama de controladores, garantiza una compatibilidad total con hardware de consumo, permitiendo un despliegue eficiente sin depender exclusivamente de componentes de servidor especializados.

Para maximizar la eficiencia, la arquitectura despliega los servicios en tres niveles de aislamiento distintos, asegurando que el sistema operativo anfitrión se mantenga lo más limpio posible:

- **Máquinas Virtuales (KVM):** Se reservan para sistemas que requieren un aislamiento total del kernel anfitrión. En este proyecto, su caso de uso principal es la ejecución de OPNsense (NGFW), garantizando que el enrutamiento y el cortafuegos operen en un entorno controlado, y la ejecución de un Windows 11 completo.
- **Contenedores de Sistema (LXC):** Se utilizan para la virtualización de la inmensa mayoría de los servicios de red del proyecto. Al compartir el núcleo del sistema anfitrión, ofrecen un entorno que se comporta de manera análoga a una máquina virtual completa y persistente, pero con una eficiencia superior. Esto permite una arquitectura completamente **modular**, garantizando el aislamiento de procesos y del sistema operativo sin la sobrecarga computacional que supone la emulación de hardware.
- **Contenedores de Aplicación Docker:** Para el despliegue de las aplicaciones finales, se utiliza Docker ejecutándose dentro de los contenedores LXC. Docker provee entornos efímeros, inmutables y altamente portátiles bajo el estándar OCI (*Open Container Initiative*). Esta técnica de anidamiento permite aislar de forma estricta las dependencias de cada aplicación, facilitando actualizaciones atómicas y reversiones de estado (*rollbacks*) rápidas, sin comprometer ni alterar el sistema operativo base del contenedor LXC [3].

4.3 Selección y Justificación de la Capa de Aplicaciones

Sobre la infraestructura de Virtualización descrita, el ecosistema de servicios se ha cimentado sobre aplicaciones cuidadosamente seleccionadas para cumplir con los Requisitos Funcionales y No Funcionales del proyecto, priorizando el rendimiento, la experiencia de usuario y el control de costes.

4.3.1. Servicio de Archivos en Red: Samba (SMB)

Para la compartición centralizada de datos y el aprovisionamiento de los espacios de trabajo definidos en los requisitos del proyecto, se ha seleccionado el protocolo **SMB/CIFS** mediante la implementación del software de código abierto **Samba**. La elección de este protocolo frente a alternativas como NFS (*Network File System*) o el discontinuado AFP (*Apple Filing Protocol*) se fundamenta en los siguientes criterios técnicos y operativos:

- **Interoperabilidad Universal y Transparencia:** SMB/CIFS constituye el estándar de facto en la industria. Ofrece soporte nativo e integrado en la práctica totalidad de sistemas operativos de escritorio (Windows, macOS y distribuciones Linux) así como en los gestores de archivos de dispositivos móviles. Esto permite a los usuarios montar las unidades de red directamente en sus exploradores nativos (como *Finder* o el *Explorador de Windows*) como si fuesen discos locales. Al eliminar la necesidad de instalar software cliente de terceros, se reduce drásticamente la curva de aprendizaje para los usuarios no técnicos.
- **Gestión Granular de Permisos:** Samba permite una traducción bidireccional entre los permisos de red y los permisos del sistema de archivos local, soportando además Listas de Control de Acceso (ACL). Esta capacidad de segmentación garantiza el aislamiento lógico de las carpetas personales de cada usuario frente al resto de integrantes de la red. Con la implementación de estas características se da cumplimiento al cuarto requisito funcional enfocado en la gestión de identidades y el control de acceso estricto.
- **Seguridad:** La implementación permite forzar la negociación exclusiva de las versiones modernas del protocolo (SMBv3). Esto asegura el cifrado de las credenciales en tránsito y previene vectores de ataque de movimiento lateral asociados a versiones heredadas e inseguras (SMB/CIFS).

4.3.2. Distribución de contenido y Colaboración: Nextcloud

Para la sincronización de archivos personales y la compartición de trabajos finalizados con terceros, se ha implementado **Nextcloud**. La selección de esta plataforma frente a su predecesor directo, **Owncloud** [26], resulta fundamental para cumplir con los requisitos de eficiencia económica y privacidad de datos establecidos en el proyecto.

Aunque Owncloud fue la solución pionera en el mercado, actualmente mantiene un modelo de negocio de núcleo abierto. Bajo este esquema, si bien el código base es público, funcionalidades críticas para entornos productivos (como auditorías avanzadas, integraciones de seguridad, ciertas capacidades de almacenamiento, etc) se encuentran restringidas a una versión empresarial sujeta a licencia comercial. Adicionalmente, su desarrollo comunitario ha perdido fuerza frente a nuevas iteraciones de la plataforma.

En contraposición, la elección de **Nextcloud**, proyecto surgido como una bifurcación directa para superar dichas limitaciones operativas y comerciales, se justifica de manera técnica por los siguientes motivos:

- **Modelo 100 % de Código Abierto:** La totalidad de las funcionalidades avanzadas se encuentra disponible de forma gratuita en su versión comunitaria. Esto incluye capacidades críticas de seguridad (autenticación de doble factor, mitigación de ataques de fuerza bruta, etc) y herramientas de gestión documental (bloqueo concurrente de archivos, control de versiones temporal, entre otros), alineándose de forma estricta con el objetivo de reducción de costes operativos.

- **Arquitectura Modular y Escalable:** Dispone de una gran cantidad de complementos dentro de su ecosistema desarrollados por la comunidad, lo que permite extender la funcionalidad base del servidor.
- **Reforzamiento de Seguridad:** Al tratarse de un servicio expuesto hacia redes externas, Nextcloud aporta una capa de defensa fundamental mediante la implementación nativa de cabeceras de seguridad HTTP estrictas y algoritmos de detección de anomalías en la escritura de archivos, diseñados para detectar y mitigar ataques de secuestro de datos.

4.3.3. Visualización Multimedia: Emby

Tras someter a prueba en entornos controlados distintas alternativas del mercado (Emby, Plex, Jellyfin e Immich), se ha seleccionado **Emby** [4] como la plataforma definitiva para la gestión multimedia. Esta decisión se fundamenta en tres pilares:

1. **Rendimiento y Usabilidad (UI/UX):** Emby ha demostrado ser operativamente más ágil que Plex en el hardware disponible. Además, su aplicación nativa para iOS proporciona una experiencia de usuario superior a la de Jellyfin y Plex, garantizando que los usuarios no técnicos puedan consumir el contenido sin curvas de aprendizaje ni fallos de interfaz.
2. **Flexibilidad Estructural:** A diferencia de la rigidez de plataformas como Immich (que no soporta álbumes anidados de forma nativa) y las recientes limitaciones introducidas en Plex Photos, Emby respeta e integra a la perfección la organización jerárquica de carpetas ya existente en el almacenamiento masivo del servidor.
3. **Viabilidad Económica:** Aunque Emby incluye componentes de código cerrado y requiere una suscripción mensual (aprox. 4,99 USD) [18] para desbloquear la transcodificación por hardware y funcionalidades completas, este gasto operativo (OPEX) está justificado. Al compararlo con el coste de mantener el material histórico en la nube pública (donde un plan de Google One de 10 TB asciende a 49,99 USD mensuales), la inversión inicial en la infraestructura NAS local (CAPEX) combinada con la reducida cuota de Emby supone una disminución drástica y sostenible de los costes recurrentes.

4.3.4. Observabilidad y Monitorización: Ecosistema Prometheus

Para la recolección de métricas, el análisis de rendimiento y la generación de alertas, se evaluó inicialmente la adopción de soluciones como Zabbix. Si bien esta herramienta resulta excelente para infraestructuras corporativas tradicionales y la monitorización de dispositivos de red físicos mediante el protocolo SNMP, su arquitectura monolítica y su dependencia de bases de datos relacionales no se alineaban de forma óptima con la naturaleza virtualizada de este proyecto. En su lugar, se seleccionó el ecosistema modular conformado por **Prometheus** [11], **Grafana** [5] y **Alertmanager** [12] .

La justificación de esta elección se fundamenta en los siguientes criterios técnicos, directamente vinculados con las tecnologías analizadas en el marco teórico y el diseño híbrido de la infraestructura (Proxmox, LXC y Docker):

- **Alineación con Entornos Dinámicos (Modelo de Extracción):** A diferencia de Zabbix, que fue concebido para topologías de red estáticas y servidores físicos persistentes, Prometheus es una solución diseñada nativamente para arquitecturas de

contenedores. Su paradigma estricto de extracción (*Pull*), fundamentado en la interrogación periódica a exportadores HTTP (*exporters*), resulta excepcionalmente eficiente. Esto permite extraer métricas de múltiples contenedores LXC y nodos de Proxmox sin la sobrecarga computacional que implicaría desplegar y mantener agentes de monitorización pesados en cada entorno aislado.

- **Desacoplamiento y Visualización Avanzada:** Al adoptar un diseño modular, la plataforma disocia la recolección de datos de su representación visual, delegando esta última tarea íntegramente a **Grafana**. Esta separación de responsabilidades permite explotar una inmensa biblioteca comunitaria de paneles de control (*dashboards*), logrando una observabilidad profesional, altamente personalizable y estéticamente superior a las consolas web integradas propias de los sistemas monolíticos.

- **Monitorización como Código y Gestión de Incidentes:** La configuración del ecosistema Prometheus se realiza mediante archivos de texto declarativos (YAML), lo que facilita el versionado, las auditorías y las copias de seguridad de la propia infraestructura de monitorización. Adicionalmente, la gestión de notificaciones se centraliza en **Alertmanager**, un componente que permite agrupar alarmas e integrarse con canales de comunicación modernos (como telegram), mitigando de forma efectiva el riesgo de "fatiga de alertas"para el administrador del sistema.

4.3.5. Conectividad Segura y Acceso Remoto: Estrategia de Triple Vía

Para garantizar la disponibilidad de la infraestructura desde el exterior, se ha descartado la dependencia de un único protocolo de red privada virtual *(VPN)*. En su lugar, se ha implementado una arquitectura de conectividad híbrida y redundante denominada *estrategia de triple vía*. Esta solución integra tres tecnologías complementarias seleccionadas por sus características criptográficas y operativas específicas, asegurando que, ante bloqueos de red externos o fallos de encaminamiento, siempre exista una ruta alternativa de acceso. Con la implementación de esta arquitectura se da cumplimiento al sexto requisito funcional enfocado en el acceso remoto seguro:

- **WireGuard (Vía Principal de Alto Rendimiento):** Se ha establecido como el túnel criptográfico primario por defecto. Su justificación técnica radica en su diseño minimalista y su ejecución nativa en el espacio del núcleo (*kernel space*) del sistema operativo. Al utilizar algoritmos criptográficos modernos y eficientes (como ChaCha20 y Poly1305), proporciona la latencia más baja y el mayor rendimiento de transferencia de datos.

- **Tailscale (Vía Secundaria y Resiliencia ante NAT):** Se ha integrado esta solución basada en una arquitectura de red en malla (*Mesh VPN*) como primer nivel de respaldo. Su elección se fundamenta en su capacidad para establecer túneles directos de nodo a nodo (*Peer-to-Peer*) mediante técnicas avanzadas de superación de traductores de direcciones de red (*NAT Traversal*). Esto permite acceder al servidor incluso si la dirección IP pública dinámica cambia abruptamente, o si el servicio de DNS dinámico (DDNS) falla.

- **OpenVPN (Vía de Respaldo y Compatibilidad Máxima):** Constituye la última capa de redundancia dentro de la estrategia de conectividad. Aunque presenta una mayor sobrecarga computacional frente a WireGuard, su madurez, estabilidad y amplio recorrido en entornos productivos lo convierten en una solución altamente fiable.

Su principal valor en este proyecto reside en su elevada compatibilidad multiplataforma, lo que lo hace especialmente adecuado como alternativa universal en escenarios donde otros protocolos puedan presentar limitaciones operativas.

4.4 Diseño de la Topología de Red

La infraestructura de red de este proyecto se ha diseñado bajo un modelo híbrido que combina una topología física de alta velocidad (2.5 Gbps / Wi-Fi 7) para el acceso de los dispositivos cliente, con una topología lógica virtualizada y segmentada (gestionada por OPNsense) para aislar las cargas de trabajo del servidor.

4.4.1. Conectividad a Internet (ISP)

El acceso a la red externa se provee a través de una conexión de fibra óptica simétrica proporcionada por el operador Movistar. Actualmente, el enlace dispone de un ancho de banda contratado de 1 Gbps, existiendo la viabilidad técnica de escalar a 2 Gbps en fases posteriores del proyecto para satisfacer incrementos en la demanda de tráfico. Para maximizar el control y el rendimiento de la red local, el equipo terminal de red óptica (ONT) suministrado por el operador ha sido configurado en modo *puente*. Esta configuración delega de forma exclusiva las funciones de enrutamiento, gestión de la traducción de direcciones de red (NAT) y emisión de red inalámbrica al enrutador principal de la red, evitando los cuellos de botella y las limitaciones inherentes al hardware comercial básico de los proveedores de servicios de internet.

4.4.2. Topología Física y Capa de Acceso (LAN)

La red local física está diseñada para eliminar los cuellos de botella en la transferencia de archivos de gran tamaño. En la figura 4.1 se puede observar un diagrama de la red física principal.

- **Enrutamiento principal y capa inalámbrica:** Se utiliza un sistema Mesh TP-Link Deco BE63 (Wi-Fi 7 / IEEE 802.11be). El sistema consta de dos nodos:
  - **Nodo Principal (Router):** Actúa como puerta de enlace, gestionando el NAT, el servidor DHCP de la red física y el cortafuegos externo. Dispone de 4 puertos Base-T a 2.5 Gbps.
  - **Nodo Secundario (Punto de Acceso Mesh):** Extiende la cobertura inalámbrica.
- **Conexión Cableada:** Ambos nodos están enlazados mediante un cable UTP Categoría 6 de 18 metros de longitud. Al ser inferior a 55 metros, este cableado garantiza soporte técnico para velocidades de hasta 10 Gbps, operando actualmente a 2.5 Gbps. Esto elimina la dependencia del espectro inalámbrico para la comunicación entre nodos, reduciendo drásticamente la latencia.
- **Conmutación (Switching) y Nodos Físicos:** Al nodo principal se conectan directamente un ordenador portátil de edición audiovisual (MacBook Pro) mediante un adaptador de 2.5 GbE y un conmutador de 2.5 Gbps no gestionable. A través de este conmutador se interconectan los equipos con mayor exigencia de ancho de banda: el servidor Proxmox VE (utilizando la interfaz de 10 GbE descrita en la sección anterior) y un segundo ordenador de escritorio también destinado a la edición (equipado con una NIC de 2.5 GbE).

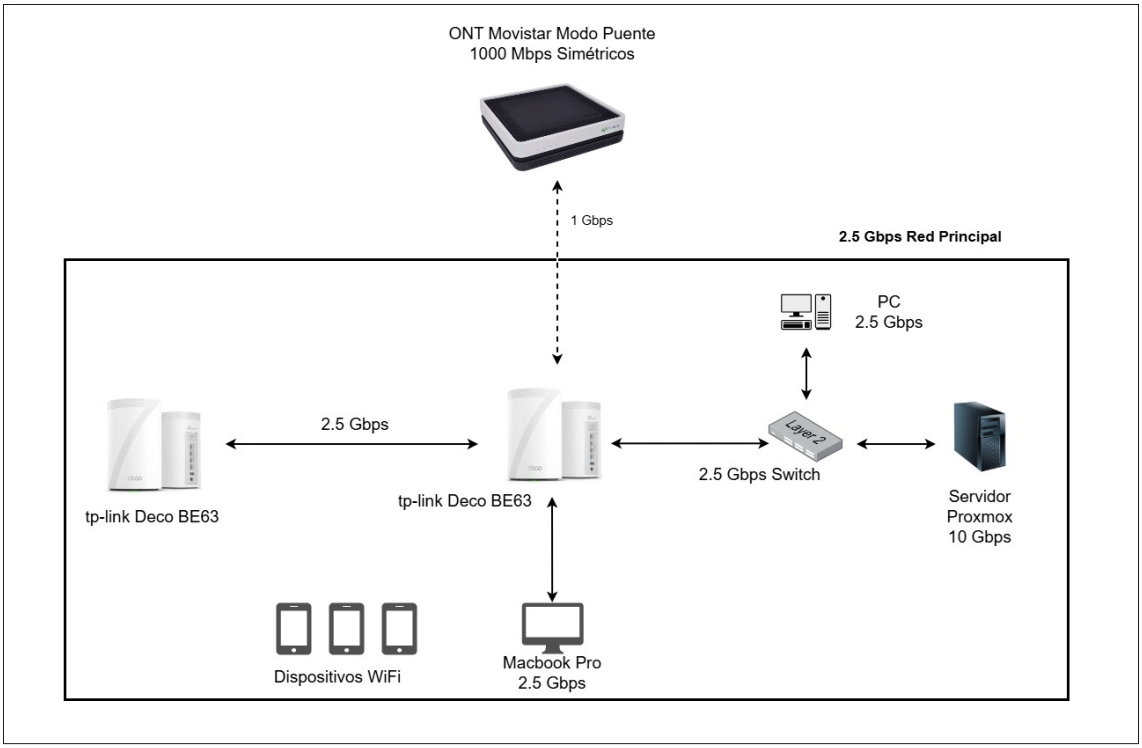

Figura 4.1: Diagrama de Red Local Física

4.4.3. Topología Lógica, Segmentación de Red y Seguridad

Para garantizar la seguridad y aplicar el principio de defensa en profundidad (*Defense in Depth*), la infraestructura establece múltiples capas de protección mediante la segmentación estricta de los servicios lógicos. Mientras que el enrutador físico TP-Link gestiona la red principal, el servidor Proxmox aloja una máquina virtual dedicada a **OPNsense**. Este sistema actúa como enrutador y cortafuegos perimetral interno, incorporando capacidades de Sistema de Prevención de Intrusiones (IPS) y un sistema de Reputación IP para gestionar y filtrar el tráfico entre tres distintas subredes.

Para ilustrar esta topología, la infraestructura se ha documentado mediante un diagrama lógico detallando la segmentación virtual dentro del hipervisor (figura 4.2).

Red de Área Local Física (192.168.68.0/24 - LAN Principal)

Es la red de confianza gestionada por el enrutador físico TP-Link.

- **Propósito y Dispositivos:** Destinada al acceso de usuarios finales. Aloja los teléfonos móviles, portátiles, ordenadores de edición y la propia interfaz de administración anfitriona del servidor Proxmox VE.

- **Integración Virtual:** Los servicios de consumo puramente interno que no requieren exposición a Internet ni aislamiento estricto operan en esta misma red. Se conectan directamente a la tarjeta de red del hipervisor mediante interfaces virtuales en modo puente (*Bridge*). Entre ellos se encuentran el contenedor LXC de **SMB** (gestión de archivos locales compartidos), el contenedor LXC de **Tailscale** (nodo de túnel VPN) y una máquina virtual ejecutando **Windows 11**.

Zona Desmilitarizada (DMZ Virtual - 10.0.1.0/24)

Subred gestionada de forma exclusiva por el cortafuegos virtual OPNsense.

- **Propósito y Servicios:** Aislar los servicios que requieren interacción con el exterior (Internet) o con redes de terceros. En este segmento operan el contenedor del Proxy Inverso (que gestiona los certificados SSL y filtra el tráfico web), la plataforma Nextcloud y el servidor multimedia Emby.

- **Reglas de Tráfico y Contención:** Los equipos de la LAN principal (192.168.68.0/24) pueden iniciar conexiones hacia la DMZ para consumir los servicios. Sin embargo, por defecto, los contenedores de la DMZ tienen denegado el inicio de conexiones hacia la LAN principal. Bajo esta directiva de contención, si un servicio expuesto fuera comprometido, el atacante quedaría confinado en este segmento sin visibilidad de los ordenadores personales ni de la interfaz de gestión de Proxmox.

Red de Seguridad y Observabilidad (10.0.3.0/24)

Segunda subred virtualizada gestionada por OPNsense, de carácter estrictamente aislado.

- **Propósito y Servicios:** Alojar la infraestructura crítica que no debe ser accesible por los usuarios finales ni exponerse a Internet bajo ninguna circunstancia. Contiene los sistemas de monitorización y alertas (Prometheus, Grafana, Alertmaneger, exportadores de métricas) y el servicio de actualización de IP dinámica (Cloudflare DDNS) [20].

- **Reglas de Tráfico:** Tráfico altamente restringido. Solo el administrador, a través de una dirección IP específica de la LAN Base y canalizado mediante un proxy inverso interno, tiene permisos de enrutamiento hacia esta subred para consultar los paneles de estado del servidor.

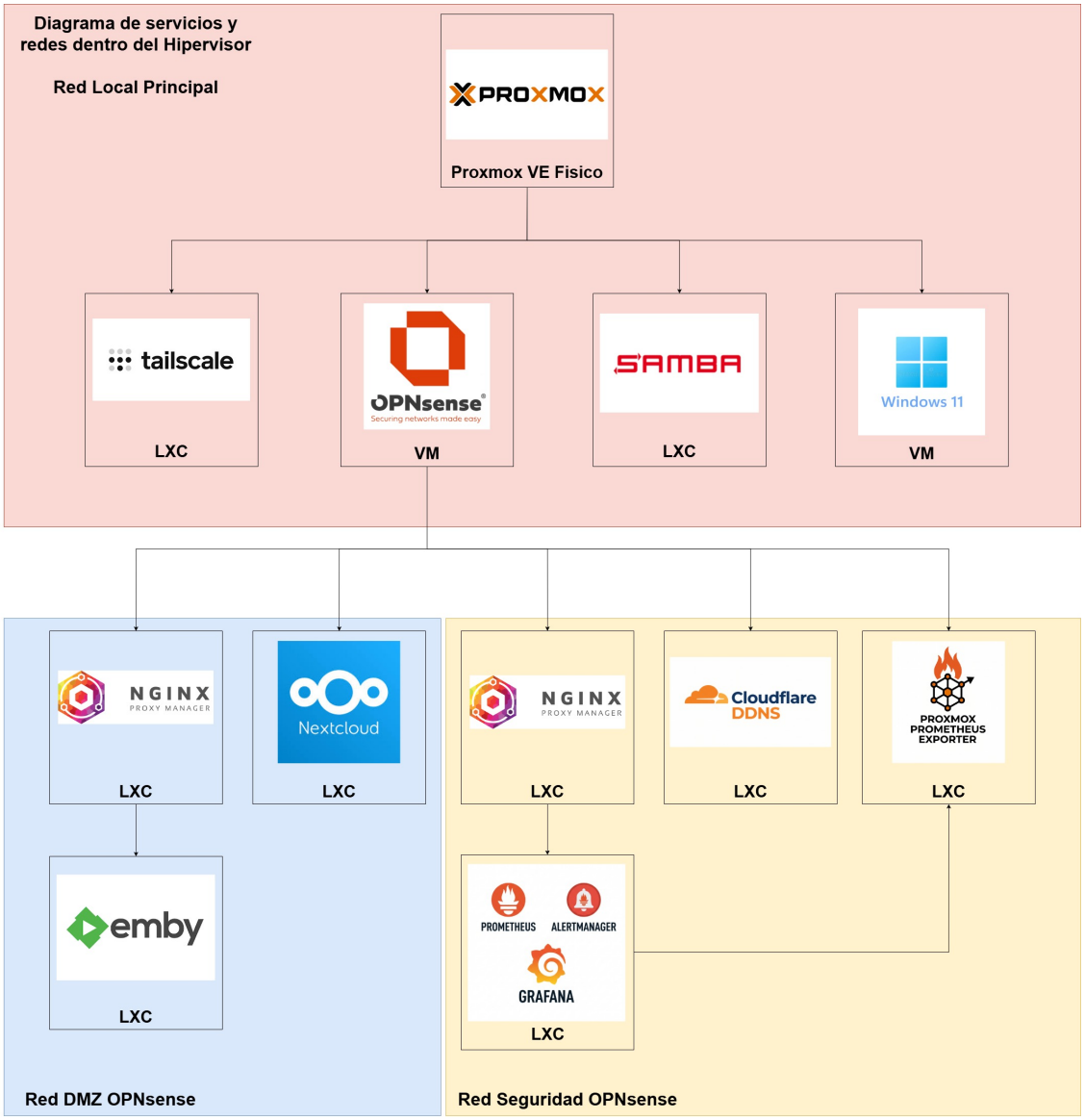

Figura 4.2: Diagrama de Segmentación Lógica en Proxmox VE

4.4.4. Flujo de Tráfico y Aislamiento

El diseño segmentado garantiza un control absoluto sobre el flujo de datos. Todo el tráfico entrante desde Internet hacia la DMZ es sometido inicialmente a una inspección profunda y proactiva antes de ser procesado por las reglas de filtrado del cortafuegos. En esta primera línea de defensa perimetral interviene el motor de Prevención de Intrusiones (IPS) Suricata, el cual evalúa los paquetes en tiempo real para descartar de forma inmediata aquellos que coincidan con firmas de ataques conocidos. De forma paralela, actúa CrowdSec, aportando una capa de seguridad dinámica que bloquea conexiones entrantes basándose en el análisis de comportamiento y en listas de reputación de direcciones IP alimentadas por inteligencia colaborativa global.

Una vez superado este primer escrutinio de amenazas, el flujo de red depurado es evaluado por las políticas del cortafuegos de OPNsense. En esta etapa se aplica una restricción estricta basada en geolocalización, la cual descarta por defecto cualquier intento de conexión que no provenga de la red local principal, o de los bloques de direcciones IP asignados a Argentina y España. Esta medida reduce drásticamente la superficie de ex-

posición al neutralizar de forma automática los escaneos masivos originados en regiones donde no existen usuarios legítimos.

La directiva principal de este diseño es la contención de brechas: si un atacante lograra evadir estos controles y vulnerar un servicio expuesto en la DMZ (por ejemplo, explotando una vulnerabilidad en Nextcloud o Emby), quedaría confinado dentro de esa subred. OPNsense bloquea por defecto cualquier intento de conexión originado en la DMZ hacia la Red Principal (protegiendo los archivos personales en el Contenedor Linux de compartición de archivos y los equipos de los usuarios) o hacia la Red de Seguridad (protegiendo la infraestructura de monitorización).

4.5 Diseño del Almacenamiento Lógico

El diseño del almacenamiento se fundamenta en una estrategia que asigna distintos sistemas de archivos y configuraciones lógicas, dependiendo del perfil de entrada/salida (E/S) requerido por los datos. Se prescinde de arreglos RAID de hardware tradicionales en favor de sistemas de archivos avanzados.

A modo de resumen estructural, la distribución física y lógica de las cuatro unidades de almacenamiento se organiza de la siguiente manera:

- **SSD 500 GB (Western Digital Blue):** Unidad de arranque. Aloja el hipervisor anfitrión (Proxmox VE), las imágenes ISO y las plantillas base LXC.
- **SSD 1 TB (Kingston A400):** Unidad de ejecución de aplicaciones. Configurado como sistema de ficheros ZFS para la ejecución de Máquinas Virtuales, contenedores LXC, volúmenes persistentes docker y bases de datos.
- **HDD 12 TB (Western Digital Red):** Unidad de almacenamiento masivo (formato EXT4). Dedicado en exclusiva al archivo histórico familiar de 20 años y copias de seguridad de las otras unidades.
- **HDD 3 TB (Western Digital Red):** Unidad de almacenamiento de trabajo (formato EXT4). Reservado para carpetas personales aisladas y proyectos audiovisuales en curso.

4.5.1. Almacenamiento del Hipervisor y Discos de Alta Capacidad

La capa de almacenamiento que opera fuera del ecosistema ZFS está compuesta por la unidad de arranque del hipervisor y los discos mecánicos destinados a los datos de gran volumen:

- **Unidad del Hipervisor (SSD 500 GB):** Utiliza el particionado estándar del instalador de Proxmox VE. El volumen raíz aloja exclusivamente el sistema operativo base. Además, este disco funciona como repositorio local para las imágenes ISO y las plantillas de los contenedores LXC. Es fundamental aclarar que las máquinas virtuales y los contenedores en ejecución no residen en esta unidad, sino que se despliegan en la otra unidad SSD con el sistema de archivos ZFS. Esta segregación garantiza que el aprovisionamiento de las máquinas sea ágil y que las operaciones del sistema anfitrión no interfieran con las cargas de trabajo de las aplicaciones.
- **Discos de Alta Capacidad (HDD 12 TB y 3 TB):** Las unidades mecánicas operan bajo el sistema de archivos EXT4 y alojan el archivo histórico de 20 años y las car-

petas personales. Se ha descartado explícitamente el uso de ZFS para estos discos por los siguientes tres criterios:

1. **Eficiencia de recursos:** EXT4 presenta un consumo de memoria RAM casi nulo, a diferencia de la alta demanda que impone la caché ARC de ZFS, liberando memoria para las máquinas virtuales.
2. **Simplicidad operativa:** Su gestión y mantenimiento diario resultan considerablemente más sencillos.
3. **Recuperación ante desastres:** En caso de un fallo catastrófico del hardware del servidor, el formato EXT4 garantiza una portabilidad universal. Los discos pueden ser extraídos y conectados físicamente a cualquier ordenador con una distribución Linux estándar para acceder a los datos de forma inmediata y nativa, sin necesidad de importar ZFS.

4.5.2. Estructura Lógica del Sistema de Archivos ZFS

El almacenamiento destinado a las aplicaciones y a las operaciones de alto rendimiento recae sobre el disco de estado sólido Kingston de 1 TB. A diferencia de los sistemas de particionado tradicionales, ZFS permite la creación de conjuntos de datos (*Datasets*) dinámicos que actúan como sistemas de archivos independientes compartiendo un mismo espacio físico.

En lugar de gestionar el disco como una unidad de almacenamiento única e indivisible, la reserva de almacenamiento se ha segmentado de forma jerárquica. Se estableció una separación primaria entre la infraestructura de virtualización (`/lxc`) y los datos persistentes de los contenedores (`/docker`). A su vez, dentro del entorno de aplicaciones, cada servicio posee su propio conjunto de datos independiente. Esta estrategia de diseño a nivel de arquitectura responde a tres objetivos operativos fundamentales:

- **Gestión de Instantáneas Aisladas:** Dado que ZFS aplica las instantáneas a nivel de conjunto de datos, esta segregación permite capturar el estado de una aplicación específica y revertirla ante un fallo catastrófico (por ejemplo, una actualización fallida) sin alterar los datos ni interrumpir el servicio del resto de aplicaciones alojadas en el mismo disco.
- **Afinamiento Granular:** El diseño fragmentado prepara la estructura para aplicar políticas de rendimiento individualizadas. Como se detallará en la fase de implementación, esto permite asignar distintos algoritmos de compresión o tamaños de bloque.

4.5.3. Políticas de Redundancia y Riesgo Asumido

Es necesario señalar que a nivel de hardware, el SSD con el sistema de archivos ZFS carece de tolerancia a fallos por redundancia física (no hay espejado).

Esto implica que el sistema puede detectar la corrupción silenciosa de datos gracias a las sumas de verificación (`checksum=on`), pero no puede autorrepararla. Esta limitación arquitectónica (derivada de la optimización del CAPEX) se asume formalmente en el diseño, delegando la disponibilidad y recuperación ante desastres a la estrategia de respaldos automatizados (Backups) que se abordará en detalle en el Capítulo 9.

CAPÍTULO 5
Desarrollo e Implementación de la Infraestructura Base

En este capítulo se detalla el proceso técnico de despliegue de la infraestructura base. Se documenta la instalación del hipervisor Proxmox VE [14], la configuración de las políticas de seguridad a nivel de anfitrión, la optimización del sistema de archivos ZFS [9] y la puesta en marcha del contenedor principal encargado de gestionar el servicio de archivos en red (SMBv3) [15].

5.1 Despliegue y configuración del Hipervisor Proxmox VE

La base de la infraestructura virtualizada opera sobre Proxmox Virtual Environment. El proceso de instalación y configuración inicial requirió adaptaciones específicas para el hardware seleccionado.

5.1.1. Instalación Base y Adaptación de Hardware

Para la creación del medio de instalación se utilizó la herramienta BalenaEtcher, garantizando la correcta escritura de la imagen en un pendrive. Durante el primer arranque, se detectaron conflictos en la inicialización de los controladores gráficos del kernel de Linux. Para solventar este problema, se editó la secuencia de arranque de GRUB, añadiendo el parámetro `nomodeset`. Esta directiva instruye al kernel para que no intente cargar controladores de vídeo de alta resolución, permitiendo completar la instalación en modo texto básico.

5.1.2. Configuración de Red y Repositorios

Tras el arranque definitivo, se configuró el hipervisor para integrarse en la topología de red:

- **Gestión de Repositorios:** Se deshabilitó el repositorio *Enterprise* (destinado a entornos con licencias comerciales) y se habilitó el repositorio *No-Subscription*, garantizando el acceso a las actualizaciones de paquetes del sistema operativo base.

- **Puentes de Red (*Linux Bridge*):** La topología virtual se estructuró mediante el aprovisionamiento de dos puentes diferenciados para separar el tráfico:

- **Puente Principal (`vmbr0`):** Se vinculó la interfaz física de 10 Gbps a este puente predeterminado, permitiendo que las máquinas virtuales y los contenedores compartan el enlace físico de alto rendimiento hacia la red local.
- **Puente Aislado (`vmbr1`):** Se creó un segundo puente virtual destinado de forma exclusiva a la Zona Desmilitarizada (DMZ). Este conmutador virtual opera de forma estrictamente interna y aislada; carece de vinculación con cualquier tarjeta de red física y no posee dirección IP asignada en el sistema anfitrión (Proxmox). Esto garantiza que, a nivel de capa de enlace de datos, el tráfico de los servicios expuestos no tenga visibilidad ni capacidad de enrutamiento directo hacia el hipervisor ni hacia la red local.

- **Redes Definidas por Software (SDN):** Se creó una zona SDN, bajo la cual se aprovisionó la red virtual `vnetSec`. Esta configuración es el cimiento para aislar la red de seguridad y observabilidad descrita en el diseño de arquitectura.

5.1.3. Configuración de Autenticación Segura mediante Claves SSH

Con el objetivo de reforzar la seguridad en el acceso remoto al sistema anfitrión, se implementó un mecanismo de autenticación basado exclusivamente en criptografía asimétrica, eliminando la dependencia de contraseñas.

Generación del Par de Claves Criptográficas

Desde un entorno cliente basado en Windows, se procedió a la generación de un par de claves pública y privada utilizando el algoritmo `ed25519`, mediante la herramienta `ssh-keygen`. La elección de este algoritmo se fundamenta en sus ventajas frente a alternativas tradicionales como RSA:

- **Mayor seguridad criptográfica:** ed25519 ofrece resistencia robusta frente a ataques criptográficos conocidos con claves de menor tamaño.
- **Mejor rendimiento:** presenta tiempos de generación y autenticación más rápidos.
- **Claves más compactas:** reduce la sobrecarga en almacenamiento y transmisión.

El proceso generó dos archivos:

- **Clave privada:** almacenada de forma segura en el equipo cliente y nunca compartida.
- **Clave pública:** destinada a ser instalada en el servidor para permitir la autenticación.

Despliegue de la Clave Pública en el Servidor

La clave pública generada se copió al sistema anfitrión Proxmox, integrándose en el archivo `/.ssh/authorized_keys` del usuario administrador. Este fichero actúa como lista de control de acceso, autorizando únicamente a los clientes que posean la clave privada correspondiente.

Securización del Servicio SSH

Como medida adicional de seguridad, se modificó la configuración del servicio SSH en el archivo `/etc/ssh/sshd_config` para deshabilitar completamente la autenticación mediante contraseña. Se ajustaron los siguientes parámetros:

- `PasswordAuthentication no`
- `ChallengeResponseAuthentication no`

Tras aplicar estos cambios, se reinició el servicio SSH para hacer efectiva la nueva política de autenticación.

Esta estrategia elimina por completo el vector de ataque basado en intentos de fuerza bruta sobre contraseñas, obligando a que todo acceso remoto se realice mediante claves criptográficas válidas. De esta forma, se establece un modelo de acceso remoto altamente robusto y alineado con las buenas prácticas de seguridad en sistemas Linux.

5.1.4. Políticas de Cortafuegos del Centro de Datos

Se adoptó un modelo de seguridad basado en el principio de *Zero Trust* (Confianza Cero) a nivel de Centro de Datos en Proxmox. Para garantizar este paradigma, las políticas globales del cortafuegos para el tráfico entrante (INPUT) y saliente (OUTPUT) se establecieron en modo de descarte (`DROP`). Todo tráfico que no se encuentre explícitamente autorizado es bloqueado por defecto.

Antes de activar el cortafuegos general, se configuraron reglas de entrada críticas para evitar la pérdida de conectividad y asegurar la administración del servidor:

- **Acceso a la Interfaz de Gestión Web (Puerto 8006 TCP):** Se autorizó el tráfico de entrada de forma exclusiva para una única dirección IP administrativa (`192.168.68.3/32`). Solo el dispositivo que tenga asignada esta dirección IP específica posee acceso al panel web del hipervisor, minimizando de manera drástica la superficie de exposición ante intentos de acceso no autorizados.
- **Acceso por Consola Segura (SSH - Puerto 22 TCP):** A diferencia de la restricción aplicada al panel web, la conexión remota mediante terminal se habilitó de manera global para toda la red local (`192.168.68.0/24`). Esta decisión asegura una vía de administración de respaldo, facilitando la resolución de incidencias y tareas de mantenimiento desde cualquier equipo conectado a la red física de confianza en caso de experimentar problemas con la interfaz gráfica.

Para la gestión del tráfico saliente (OUTPUT), se diseñó e implementó un grupo de reglas lógicas denominado `DefaultOut`, estructurado con el siguiente orden de prioridad:

- **Aislamiento Preventivo de la Red Local:** La regla de mayor jerarquía bloquea explícitamente cualquier intento de conexión iniciado desde el propio sistema anfitrión (Proxmox) hacia la red local (`192.168.68.0/24`). A nivel arquitectónico, el hipervisor no tiene justificación operativa para establecer comunicaciones de salida hacia los dispositivos personales de los usuarios, previniendo así el movimiento lateral de amenazas en el caso hipotético de que el nodo principal fuera comprometido.

- **Excepciones para Servicios Esenciales:** Una vez aislado el tráfico hacia la red local, se habilitaron reglas de aceptación (`ACCEPT`) limitadas estrictamente a los protocolos indispensables para la operatividad y mantenimiento del sistema operativo base: NTP (sincronización de tiempo), DNS (resolución de nombres de dominio), HTTP y HTTPS (descarga de actualizaciones de paquetes) e ICMP (Ping, para diagnósticos de conectividad). Cualquier otro tráfico saliente no contemplado en estas excepciones es descartado por la última política de descarte final.

5.2 Implementación del Almacenamiento ZFS y EXT4

El subsistema de almacenamiento requirió un nivel de personalización avanzado para diferenciar las cargas de trabajo secuenciales de las transaccionales.

5.2.1. Montaje de Discos EXT4

Los discos mecánicos de 12 TB y 3 TB, formateados en EXT4, se montaron directamente sobre el sistema anfitrión. Para garantizar su disponibilidad en cada reinicio, se editaron las entradas correspondientes en el archivo `/etc/fstab` utilizando sus identificadores únicos (UUID). Se añadieron los parámetros `noatime` (para evitar la escritura de metadatos de acceso temporal, mejorando el rendimiento) y `nofail` (para asegurar que Proxmox inicie correctamente incluso si uno de los discos experimenta un fallo de hardware durante la secuencia de arranque).

5.2.2. Creación de grupos de almacenamiento y Optimizaciones ZFS (Cargas Transaccionales)

Sobre el SSD de 1 TB se creó el grupo de almacenamiento `datapoolSSD1TB`, destinado al alojamiento de contenedores LXC y volúmenes de Docker. Se establecieron de forma explícita propiedades de afinamiento a nivel de sistema de archivos:

Ajustes de Rendimiento y Atributos

Aprovechando la flexibilidad de ZFS, se han aplicado los siguientes parámetros no predeterminados a los Datasets para maximizar el rendimiento de las bases de datos y prolongar la vida útil del SSD:

- **Compresión (`compression=lz4`):** El algoritmo lz4 comprime los datos al vuelo antes de escribirlos en el disco. Dado que el procesador i5-9400F posee ciclos de reloj suficientes, este proceso es imperceptible en latencia y reduce drásticamente el desgaste por escritura del SSD.
- **Tiempos de Acceso (`atime=off`):** Por defecto, se escriben metadatos cada vez que se lee un archivo, generando una amplificación de escritura innecesaria. Desactivar `atime` elimina esta penalización, algo vital para contenedores que leen miles de archivos pequeños constantemente.
- **Atributos Extendidos (`xattr=sa`):** La configuración en modo *Atributos del sistema* obliga a ZFS a almacenar los atributos extendidos de Linux (como los permisos ACL) dentro del mismo inodo del archivo en lugar de en bloques separados, acelerando significativamente las operaciones de búsqueda.

- **Mantenimiento y Desgaste:** Se activó el parámetro `autotrim=on` para notificar al controlador del SSD sobre los bloques liberados, manteniendo la velocidad de escritura a largo plazo. Se desactivó el registro de acceso (`atime=off`).
- **Segmentación por Datasets:** Se crearon conjuntos de datos individuales para separar el sistema de archivos de los contenedores (`/lxc`) de los volúmenes de datos de las aplicaciones (`/docker`).

Optimización del Tamaño de Bloque (`recordsize`)

El `tamaño de bloque` en ZFS define el tamaño máximo del bloque de datos. La arquitectura de este proyecto demuestra una adaptación a la carga de trabajo:

- **Propósito general (128K):** Datasets como `nextcloudAIO2_data` operan con el valor predeterminado de 128 KB, ideal para el almacenamiento de archivos estándar y documentos.
- **Cargas transaccionales (32K):** El Dataset `obs_data` (observabilidad, Grafana, Prometheus, etc) ha sido explícitamente afinado con un `recordsize=32K`. Disminuir el tamaño de bloque beneficia a bases de datos o aplicaciones que realizan escrituras aleatorias pequeñas, evitando el fenómeno de amplificación de lectura/escritura que ocurriría si se modificaran bloques de 128 KB para cambiar solo unos pocos kilobytes de datos.

Finalmente, el dataset `/lxc` se integró en la interfaz de gestión de almacenamiento de Proxmox. Se desactivó la directiva `discard` nativa de los discos virtuales de los contenedores, ya que esta generaba conflictos con el motor de instantáneas (Snapshots) de ZFS.

5.3 Implementación del Entorno de Ejecución de Contenedores

Para mantener la seguridad y el aislamiento, el servicio de gestión de archivos y el motor de Docker se implementaron utilizando Contenedores de Sistema Linux (LXC) en modo **sin privilegios** (*Unprivileged*).

5.3.1. Mapeo de Usuarios (User Namespaces) y Puntos de Montaje

En un contenedor LXC sin privilegios, el usuario *root* interno no tiene permisos de *root* sobre el hipervisor anfitrión, operando por defecto bajo el identificador (UID) 100000. Para que los contenedores LXC puedan leer y escribir en los volúmenes EXT4 y en el conjunto de datos de Docker, se aplicaron dos configuraciones críticas:

1. Se modificó la propiedad del directorio de datos en el anfitrión mediante el comando `chown -R 100000:100000`, transfiriendo el control de los archivos al UID virtual.
2. Se editaron los archivos de configuración de los contenedores (`/etc/pve/lxc/100.conf`), añadiendo directivas de montaje (Bind Mounts) para inyectar los directorios `/mnt/3tb` y `/mnt/12tb` directamente en el sistema de archivos del contenedor, junto con reglas de mapeo de identidades para alinear los permisos de lectura y escritura entre el host y el contenedor.

5.4 Implementación de Servicios de Archivos en Red (SMB/-CIFS)

El servidor de archivos se materializó mediante el despliegue de un contenedor LXC basado en la plantilla de Debian 13, asignándole 3 núcleos de procesamiento, 1 GB de memoria RAM y una dirección IP estática (192.168.68.4) en la LAN principal.

5.4.1. Gestión de Usuarios, Grupos y Permisos Locales

Se diseñó un esquema de permisos estrictos para la administración de los directorios de los activos digitales.

- **Identidades:** Se crearon cuentas de usuario del sistema independientes y se definieron grupos de control lógicos (`familia` para el acceso general e histórico, y grupos privados para proyectos personales). Esta separación lógica garantiza que ningún usuario tenga visibilidad sobre los archivos ajenos, dando cumplimiento al cuarto requisito funcional enfocado en la privacidad y el control de acceso estricto.
- **Herencia de Permisos:** Para garantizar que cualquier nuevo archivo creado en el servidor mantuviera los permisos correctos, se aplicó el bit de identificación de grupo (`chmod g+s`) a los directorios de red. Esto fuerza a que todos los ficheros y subdirectorios hereden el grupo propietario del directorio padre, resolviendo los problemas comunes de archivos inaccesibles al ser subidos por diferentes usuarios. Esta herencia automática consolida el aislamiento de los espacios de trabajo y afianza el cumplimiento del cuarto requisito funcional.

5.4.2. Configuración del demonio Samba (`smb.conf`)

El servicio de compartición de archivos se configuró utilizando Samba, endureciendo sus parámetros de red:

- **Protocolo Cifrado:** Se forzó el parámetro `client max protocol = SMB3`, deshabilitando versiones heredadas e inseguras como SMBv1, previniendo vulnerabilidades conocidas.
- **Enlaces Simbólicos:** Se habilitó el soporte para enlaces simbólicos. Esto permitió crear accesos directos lógicos (Soft Links) dentro del disco de 3 TB que apuntan de forma transparente hacia el disco histórico de 12 TB, unificando la experiencia de navegación para el usuario final sin mezclar los datos físicamente.
- **Aislamiento de Recursos Compartidos (Shares):** Mediante la directiva `valid users`, se restringió el acceso a nivel de protocolo, denegando la entrada a invitados (`guest ok = no`) y forzando una máscara de creación de archivos `0770`, asegurando que solo el propietario y los miembros del grupo asignado tengan visibilidad y privilegios de modificación.

5.5 Organización Lógica y Control de Acceso a Nivel de Ficheros

La arquitectura de permisos y la topología de directorios se han diseñado aplicando el principio de mínimo privilegio de forma granular. La distribución lógica de los

datos y sus correspondientes controles de acceso se segmentan según el volumen de almacenamiento físico subyacente. Esta estructura garantiza la consolidación unificada de la información, la provisión de entornos de trabajo independientes y la protección de la privacidad mediante el uso de identidades. La implementación conjunta de estas medidas da cumplimiento al primer requisito funcional enfocado en el repositorio histórico centralizado, al tercer requisito referente a los espacios de trabajo personales aislados y al cuarto requisito sobre el control de acceso estricto.

5.5.1. Volumen Histórico (Unidad de 12 TB)

Este disco centraliza la totalidad del archivo familiar de las últimas dos décadas bajo un modelo de permisos orientado a la colaboración controlada. Esta capacidad de unificación y preservación da cumplimiento al primer requisito funcional enfocado en proveer un repositorio histórico centralizado. Su organización interna es la siguiente:

- **Archivo Cronológico:** El contenido multimedia principal está organizado jerárquicamente por años. A nivel de sistema de archivos, el usuario propietario de estos directorios es el administrador del sistema, mientras que el grupo propietario asignado es `familia`. Todos los miembros autenticados que pertenecen a este grupo heredan los permisos de acceso correspondientes, permitiendo la visualización y gestión conjunta de este material histórico.
- **Directorio de Dispositivos Móviles:** Este espacio está destinado a preservar las fotografías provenientes de teléfonos móviles antiguos de los distintos integrantes. Se implementó un modelo de permisos mixto donde el directorio padre posee permisos de lectura y descubrimiento para que todo el grupo `familia` pueda acceder al listado de carpetas. En su interior, cada subdirectorio aplica un control de acceso discrecional estricto. La propiedad de usuario y de grupo de cada subdirectorio está asignada de forma exclusiva a su respectivo dueño, impidiendo que el resto de los usuarios puedan ingresar, visualizar o modificar el contenido ajeno. Este nivel de aislamiento da cumplimiento al cuarto requisito funcional enfocado en la privacidad y el control de acceso estricto.

5.5.2. Volumen de Trabajo y Espacios Personales (Unidad de 3 TB)

El disco secundario aborda específicamente la creación de entornos aislados para la operatividad diaria. Se divide en dos áreas funcionales delimitadas. La creación de estos entornos da cumplimiento al tercer requisito funcional enfocado en proveer espacios de trabajo personales aislados:

- **Directorios Privados de Usuario:** Se aprovisionó una carpeta de red aislada para cada integrante del sistema. Estas carpetas operan bajo un esquema de privacidad estricto. Tanto el usuario propietario como el grupo propietario coinciden exclusivamente con la identidad del individuo, revocando cualquier tipo de permiso (lectura, escritura o ejecución) para el resto de las cuentas del sistema. Esto garantiza espacios de trabajo completamente seguros para documentos personales y material audiovisual en fase de edición, lo cual da cumplimiento al cuarto requisito funcional enfocado en la privacidad y el control de acceso estricto.
- **Directorio de Intercambio Temporal:** Para facilitar la colaboración interna y evitar la duplicación innecesaria de datos en las carpetas privadas, se configuró un directorio de tránsito común. Este espacio posee permisos de lectura y escritura

habilitados para todos los usuarios autenticados en el servidor, funcionando como una zona de intercambio rápido para compartir ficheros entre distintos equipos y miembros de la red sin alterar las políticas de seguridad de los repositorios principales.

5.6 Estado Final de la Infraestructura Base

Tras la ejecución de los procesos descritos en este capítulo, la infraestructura alcanza un estado operativo funcional que constituye una base sólida para futuras ampliaciones.

Por un lado, el hipervisor Proxmox Virtual Environment queda completamente desplegado, securizado y segmentado a nivel de red. La configuración de almacenamiento, tanto en sistemas de archivos EXT4 como en ZFS, ha sido optimizada para soportar diferentes tipos de carga de trabajo, garantizando rendimiento y fiabilidad.

Por otro lado, se ha implementado el entorno de ejecución de contenedores mediante LXC en modo sin privilegios, asegurando aislamiento y control granular de permisos. Sobre esta base, el contenedor destinado a servicios de archivos en red (SMB) se encuentra plenamente operativo.

El servicio Samba dispone de:

- Un modelo estructurado de usuarios y grupos.
- Políticas de permisos alineadas con los requisitos de privacidad y colaboración.
- Integración funcional con los volúmenes de almacenamiento definidos.

Como resultado, los usuarios pueden conectarse de forma remota dentro de la red local y acceder a los recursos compartidos del servidor de manera segura y controlada.

Este estado inicial no solo satisface los requisitos funcionales básicos, sino que establece el punto de partida para la siguiente fase del proyecto, centrada en el despliegue progresivo de servicios y aplicaciones adicionales sobre la infraestructura virtualizada.

CAPÍTULO 6
Despliegue de Servicios y Aplicaciones

6.1 Nube Privada y Colaboración: Nextcloud

El núcleo de la infraestructura orientada al usuario final recae sobre la plataforma de nube privada. Para satisfacer los requisitos funcionales de acceso ubicuo, intercambio de archivos y sincronización de clientes, se ha optado por el despliegue de **Nextcloud** [6] [7].

6.1.1. Justificación Tecnológica: Nextcloud All-in-One (AIO) sobre Docker

Históricamente, el despliegue de una instancia productiva de Nextcloud requería la configuración manual y el acoplamiento de múltiples componentes independientes: un servidor web (Apache o Nginx), un motor de base de datos (MariaDB o PostgreSQL), un servidor de caché en memoria (Redis) y un entorno de ejecución PHP (PHP-FPM). Este enfoque tradicional generaba un alto coste operativo en el mantenimiento y elevaba el riesgo de fallos durante las actualizaciones de versión.

Para mitigar esta complejidad, se seleccionó la solución oficial **Nextcloud All-in-One (AIO)**. Esta variante encapsula todos los servicios necesarios en contenedores Docker preconfigurados y orquestados por un contenedor maestro (*Mastercontainer*). Las principales ventajas que justifican esta decisión son:

- **Gestión del ciclo de vida:** El contenedor maestro se encarga de aprovisionar, actualizar y respaldar automáticamente el resto de los contenedores subyacentes (Base de datos, Redis, Apache), garantizando que operen de la forma en que los desarrolladores originales de la plataforma lo han diseñado.

- **Integración nativa con Proxies Inversos:** Nextcloud AIO está diseñado para integrarse fluidamente con proxies inversos modernos como **Caddy** [28], evitando los comunes conflictos de cabeceras HTTP y redirecciones que suelen ocurrir en despliegues manuales.

- **Aislamiento mediante Docker:** Al ejecutar la plataforma sobre el motor de contenedores Docker, se asegura que las dependencias de software de Nextcloud no contaminen ni alteren el sistema operativo base del entorno virtualizado.

6.1.2. Preparación del Entorno y Sistema de Archivos

El despliegue no se realizó de forma monolítica, sino respetando la arquitectura segmentada definida en capítulos anteriores. El proceso comenzó por la capa de almacenamiento:

1. **Aprovisionamiento ZFS:** Se creó un conjunto de datos específico en el grupo de almacenamiento ZFS destinado exclusivamente a alojar los volúmenes persistentes de los contenedores Docker de Nextcloud. A este conjunto de datos se le aplicaron las optimizaciones de rendimiento detalladas previamente.

2. **Entorno de Ejecución (LXC):** Se desplegó un Contenedor Linux (LXC) basado en la distribución Debian 13 [31]. A nivel de red, la interfaz virtual de este contenedor se asignó a la Zona Desmilitarizada (DMZ) gestionada por OPNsense.

3. **Políticas de Cortafuegos:** Se aplicó el principio de mínimo privilegio en el cortafuegos del anfitrión. Se habilitaron reglas de entrada explícitas únicamente para los puertos TCP/UDP 443 (tráfico HTTPS), bloqueando por defecto cualquier otra solicitud entrante. Para el tráfico de salida, se aplicó la política `DefaultOut`, permitiendo exclusivamente los protocolos necesarios para la operación y diagnóstico del sistema (HTTP, HTTPS, DNS, NTP e ICMP).

6.1.3. Despliegue del Servicio

Con el entorno preparado, se procedió a la actualización de los repositorios del sistema operativo Debian y a la instalación del motor de Docker siguiendo la documentación oficial. Posteriormente, se redactó el archivo de orquestación `docker-compose.yml` basado en la plantilla oficial de Nextcloud AIO.

Para adaptar la plantilla a los requisitos de la infraestructura local, se modificaron y establecieron variables de entorno críticas:

- `NEXTCLOUD_DATADIR=/mnt/ncdata`: Se redirigió el directorio de datos principal hacia el punto de montaje del almacenamiento de ZFS.

- `NEXTCLOUD_MEMORY_LIMIT=2048M`: Se aumentó el límite de memoria de PHP a 2 GB para garantizar la fluidez de la interfaz ante galerías de imágenes de gran tamaño.

- `NEXTCLOUD_UPLOAD_LIMIT=700G`: Se expandió drásticamente el límite de subida para permitir la transferencia ininterrumpida de entregables audiovisuales pesados (archivos de vídeo RAW/4K o ficheros comprimidos).

Tras la ejecución del archivo compose, el contenedor maestro aprovisionó exitosamente la base de datos, el servicio de caché de alto rendimiento (Redis) y el servidor web, dejando la plataforma operativa.

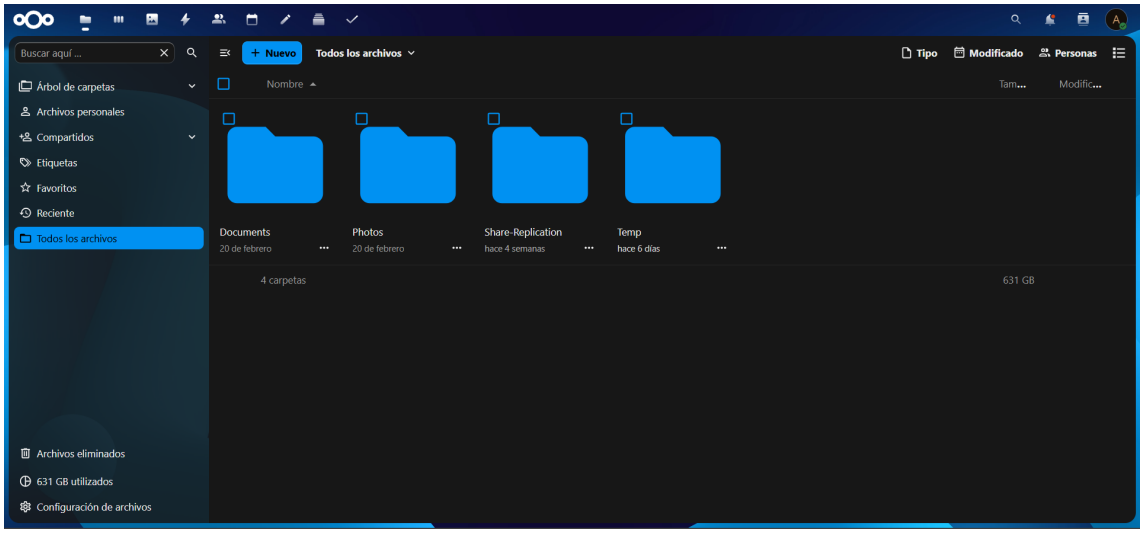

Figura 6.1: Interfaz principal de archivos de Nextcloud.

6.1.4. Enrutamiento Perimetral y Doble DNAT

Dado que el contenedor de Nextcloud reside en una red aislada (DMZ en la subred 10.0.1.0/24) gestionada por OPNsense, y el acceso físico a Internet está gobernado por el router TP-Link (en la subred 192.168.68.0/24), fue necesario implementar una topología de **Doble DNAT** y reenvío de puertos en cascada.

El flujo de tráfico entrante se resolvió de la siguiente manera:

1. **Primer salto (Router Físico):** En el enrutador TP-Link se configuró una regla de *reenvío de puertos* que redirige el tráfico HTTPS (puerto 443) proveniente de Internet hacia la dirección IP de la interfaz WAN externa *(192.168.68.20)* del cortafuegos virtual OPNsense.
2. **Segundo salto (OPNsense):** El cortafuegos recibe la petición, la procesa mediante sus motores de seguridad (IDS/IPS) y realiza una segunda redirección de puertos hacia la IP interna del contenedor LXC de Nextcloud ubicado en la DMZ *10.0.1.2*.

Esta técnica de doble traducción de direcciones, tanto de entrada como de salida, es necesaria en esta arquitectura dado que el equipo comercial TP-Link carece de capacidades avanzadas de traducción de direcciones *(NAT)* de redes que no administra, como la red desmilitarizada que gestiona el OPNsense.

6.1.5. Configuración General y Aseguramiento (Hardening)

Una vez establecida la conectividad, se procedió a la configuración desde la interfaz web de administración de Nextcloud, aplicando medidas de fortalecimiento de seguridad:

- **Autenticación de Dos Factores (2FA):** Se forzó el uso de 2FA mediante aplicaciones de contraseñas de un solo uso basadas en el tiempo *(TOTP)* para la cuenta del administrador, mitigando riesgos de acceso no autorizado.
- **Protección contra Software Malicioso:** Se activó y configuró la aplicación *Antivirus para archivos* (integrada con el motor de código abierto ClamAV). Este servicio escanea automáticamente cada archivo subido al servidor en busca de firmas maliciosas.

- **Control de Acceso:** Se inhabilitó el registro público de usuarios. La política de seguridad dicta que el aprovisionamiento de nuevas cuentas es una tarea reservada de forma exclusiva y manual al administrador del sistema.
- **Identidad Visual:** Se personalizó la interfaz gráfica adaptando los colores, el logotipo y la pantalla de inicio predeterminada, para generar un entorno de trabajo unificado.

6.1.6. Flujos de Trabajo y Experiencia del Usuario Final

La puesta en producción de Nextcloud transformó radicalmente las metodologías de trabajo definidas en la problemática inicial, cumpliendo varios requisitos funcionales críticos:

- **Intercambio Externo:** Los usuarios ahora pueden generar enlaces públicos seguros, con fecha de caducidad y protección por contraseña, para compartir entregables finales (carpetas completas o archivos comprimidos) con clientes externos. Los clientes pueden descargar el material audiovisual directamente desde el servidor local sin necesidad de registrarse en la plataforma, eliminando el uso de servicios comerciales de alojamiento de terceros. Esta capacidad de generar enlaces de descarga públicos da cumplimiento al quinto requisito funcional enfocado en el intercambio externo y la colaboración.
- **Sincronización en Tiempo Real:** Se desplegó el cliente de escritorio oficial de Nextcloud en las estaciones de edición de vídeo y en portátiles personales. Esta aplicación permite seleccionar carpetas de trabajo locales que se replican y sincronizan de forma transparente en el servidor. Cualquier modificación realizada en un archivo del escritorio se refleja en tiempo real en el almacenamiento del servidor.
- **Acceso Móvil Ubicuo:** Mediante la aplicación móvil para iOS y Android, los usuarios pueden consultar, compartir o almacenar documentos privados de uso frecuente bajo demanda, sin ocupar el almacenamiento interno de sus teléfonos móviles.

6.2 Centro Multimedia: Emby y Gestión de Proxy Inverso

Se procedió al despliegue del servidor multimedia Emby. Al tratarse de un servicio que requiere exposición hacia el exterior para su consumo remoto, su arquitectura de implementación se dividió en dos componentes alojados dentro de la zona desmilitarizada: el núcleo de la aplicación (Emby) y el gestor de tráfico cifrado proxy inverso (Nginx Proxy Manager). La implementación de esta arquitectura garantiza la visualización audiovisual bajo demanda y la disponibilidad del sistema en diferentes dispositivos, dando cumplimiento al segundo requisito funcional enfocado en el acceso ubicuo y multiplataforma.

6.2.1. Despliegue y Optimización del Contenedor Emby

El servidor multimedia se implementó sobre un Contenedor Linux (LXC) basado en la distribución Debian 13. A diferencia de otros servicios del proyecto, se descartó explícitamente el uso de Docker para esta aplicación. La instalación se realizó de forma nativa en el sistema de archivos del contenedor (mediante la instalación directa del paquete .deb). Esta decisión arquitectónica de la no utilización de Docker, elimina cualquier capa de abstracción adicional maximizando el rendimiento computacional y reduciendo la

latencia de las operaciones de lectura, factores críticos para la transcodificación de vídeo en tiempo real.

Para garantizar un funcionamiento fluido en la generación de miniaturas y el procesamiento de vídeo, se le asignaron recursos holgados: 6 núcleos de procesamiento y 6 GB de memoria RAM.

Puntos de Montaje de Solo Lectura y Permisos

Cumpliendo con el principio de seguridad y protección de datos, el acceso de Emby al almacenamiento de material histórico se configuró de forma estrictamente unidireccional. Se establecieron puntos de montaje desde el hipervisor anfitrión hacia el contenedor con el parámetro de solo lectura (`ro=1`).

Posteriormente, dentro del contenedor, se crearon los grupos correspondientes a la estructura de archivos y se añadió el usuario de ejecución del servicio (`emby`) a dichos grupos. Esta configuración impide a nivel de sistema de archivos que una vulnerabilidad en Emby pueda resultar en la modificación, borrado o secuestro *(ransomware)* del archivo histórico familiar.

Configuración y Activación de la Plataforma

Una vez instalado, se accedió a la interfaz de administración mediante el puerto local 8096. Tras corroborar la correcta indexación de las bibliotecas y la generación de las miniaturas del histórico de 2 decadas, se procedió a la activación de la licencia comercial (Emby Premiere) por un coste recurrente de 4,99 USD mensuales. Esta acción habilitó el uso completo de las capacidades de transcodificación por hardware y el acceso nativo a las aplicaciones móviles, justificando la inversión frente a los altos costes del almacenamiento en la nube. Finalmente, se crearon los perfiles de usuario con acceso segmentado a las distintas bibliotecas.

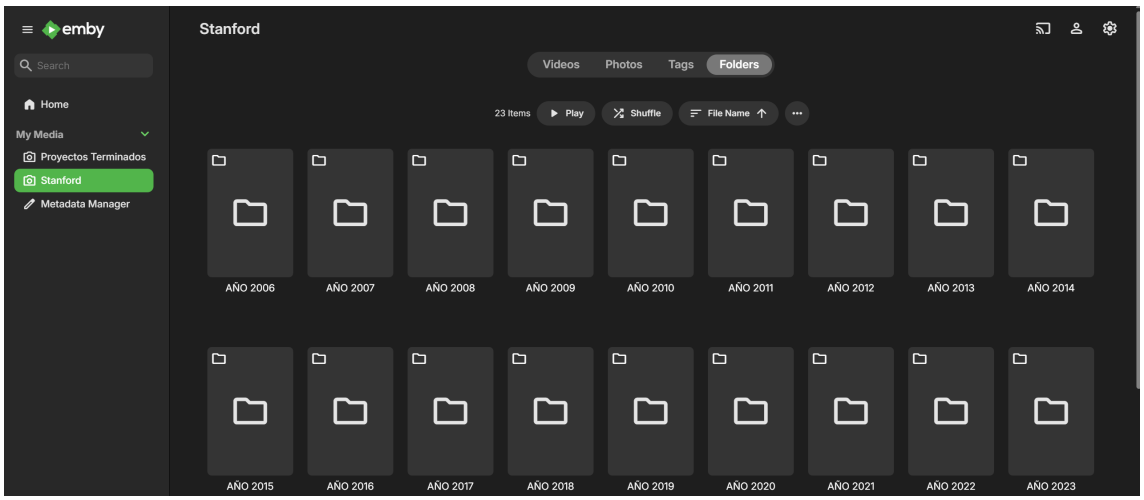

Figura 6.2: Interfaz principal del servidor multimedia Emby.

6.2.2. Exposición Segura: Nginx Proxy Manager (NPM)

Para no exponer directamente el puerto HTTP (8096) de Emby a Internet y garantizar el cifrado de las comunicaciones en tránsito, se desplegó un segundo contenedor LXC en

la DMZ, aprovisionado con Debian 13, 2 núcleos y 1 GB de RAM. Este nodo actúa como un Proxy Inverso mediante la herramienta **Nginx Proxy Manager** (NPM) [8].

Un proxy inverso actúa como un punto de entrada único entre la red pública (Internet) y los servicios alojados en la red interna. Su función principal es recibir todas las peticiones externas dirigidas al servidor y enrutarlas hacia el contenedor o servicio correspondiente basándose en el nombre de dominio solicitado. Esto reduce drásticamente la superficie de ataque, ya que evita la necesidad de abrir múltiples puertos en el cortafuegos perimetral.

En este contexto, la elección de NPM resulta fundamental por su capacidad para automatizar la seguridad criptográfica: la herramienta integra de forma nativa la emisión, configuración y renovación automática de certificados SSL/TLS gratuitos a través de la autoridad certificadora *Let's Encrypt* [29]. Gracias a esto, todo el tráfico intercambiado entre los dispositivos móviles y el servidor NPM queda encapsulado en un canal seguro, previniendo la interceptación de credenciales de acceso y mitigando los ataques de intermediario (*Man-in-the-Middle*).

Dado que NPM requiere de una base de datos relacional para gestionar sus configuraciones y certificados, su instalación se realizó mediante Docker, facilitando el empaquetado del ecosistema. El archivo `docker-compose.yml` se estructuró con un enfoque de seguridad de red interna, aislando la base de datos MariaDB y exponiendo únicamente los puertos necesarios del proxy:

```
services:
  db:
    image: mariadb:10.11
    container_name: npm-db
    restart: unless-stopped
    volumes:
      - db_data:/var/lib/mysql
    networks:
      - npm_internal        # Red interna aislada
  app:
    image: jc21/nginx-proxy-manager:v2.13.7
    container_name: nginx-proxy-manager
    restart: unless-stopped
    ports:
      - "46519:443"         # Puerto público
      - "81:81"             # Panel de administración local
    volumes:
      - npm_data:/data
      - npm_letsencrypt:/etc/letsencrypt
    networks:
      - proxy               # Red de cara al exterior
      - npm_internal        # Acceso a la base de datos
```

6.2.3. Flujo de Red, Enrutamiento y Políticas de Cortafuegos

La publicación del servicio hacia el exterior se articuló a través de una cadena de redirección de puertos diseñada para atravesar las distintas capas de la infraestructura:

1. **Capa Física:** El enrutador perimetral (TP-Link) redirige el tráfico TCP/UDP entrante por el puerto no convencional `46519` hacia la dirección IP de la interfaz WAN del cortafuegos virtual OPNsense *(192.168.68.20)*.
2. **Capa de Seguridad Lógica (OPNsense):** El cortafuegos recibe la petición, la somete a la inspección del IPS Suricata y Crowdsec, y tras validarla, aplica una regla D-NAT *Destination NAT* que reenvía el tráfico del puerto `46519` hacia la dirección IP privada del contenedor Nginx Proxy Manager en la DMZ *(10.0.1.3)*.
3. **Capa de Proxy y Cifrado:** NPM recibe la conexión externa cifrada (HTTPS), gestiona el túnel criptográfico utilizando certificados SSL/TLS emitidos por *Let's Encrypt*, y redirige el tráfico en texto plano (HTTP) internamente hacia el puerto `8096` del contenedor Emby *(10.0.1.4)*.

Fortalecimiento de Reglas Internas

Para mantener la integridad del modelo de confianza cero dentro de Proxmox, se aplicaron reglas de cortafuegos restrictivas a nivel de contenedor:

- **NPM LXC:** Se habilitó el tráfico entrante únicamente para el puerto `46519` (TCP) destinado al tráfico de clientes externos, y el puerto `81` (TCP) restringido exclusivamente para la red local de administración. Se aplicó la política de descarte saliente (`DefaultOut`).
- **Emby LXC:** Se configuró una regla estricta que permite el tráfico entrante al puerto `8096` (TCP/UDP) cuyo origen provenga de manera exclusiva de la dirección IP del proxy inverso (NPM). Cualquier intento de conexión directa desde otro nodo de la DMZ u origen no autorizado es descartado, asegurando que todo el tráfico transite de forma obligatoria por los sistemas de inspección y cifrado.

6.3 Virtualización de Escritorio: Estación de Gestión

Se desplegó una máquina virtual KVM equipada con el sistema operativo Windows 11. A diferencia de los servicios empaquetados en contenedores LXC, este entorno requiere una virtualización completa de los recursos físicos para ejecutar un sistema operativo de escritorio de forma aislada. La provisión de este entorno dedicado para la organización y administración de los activos digitales da cumplimiento al séptimo requisito funcional enfocado en la estación de trabajo virtualizada.

6.3.1. Aprovisionamiento y Optimización del Sistema Operativo

El proceso de instalación se realizó a partir de la imagen ISO oficial descargada desde el sitio web de Microsoft, y luego cargada en el almacenamiento local del hipervisor. Para garantizar un rendimiento ágil, se le asignaron recursos considerables: 6 núcleos de procesamiento y 8 GB de memoria RAM.

Dado que la interacción con esta máquina se realiza de forma estrictamente remota, se aplicó un perfil de optimización centrado en la fluidez de la transmisión gráfica. Se deshabilitaron los programas de inicio automático en segundo plano, así como la totalidad de los efectos visuales, animaciones de ventanas y transparencias del entorno de escritorio. Esta reducción de la carga del sistema garantiza una experiencia de uso ágil y sin interrupciones. Finalmente, se habilitó el servicio de escritorio remoto nativo de Windows para permitir las conexiones entrantes desde la red local o la VPN.

6.3.2. Integración con el Almacenamiento Centralizado

Con el sistema operativo optimizado, se procedió a integrar la estación de trabajo con el servidor de archivos Samba aprovisionado previamente en la infraestructura. Se mapearon como unidades de red persistentes las carpetas de uso general: el archivo histórico completo alojado en el disco de 12 TB y los directorios de proyectos colaborativos del disco de 3 TB.

En estricto cumplimiento del cuarto requisito funcional, políticas de control de acceso y privacidad, los espacios de trabajo personales de los distintos usuarios no se montan de forma automática. Si un integrante necesita operar sobre sus archivos privados desde esta máquina virtual, debe mapear su unidad de red introduciendo sus credenciales específicas de Samba, asegurando que el acceso sea temporal y restringido a su sesión activa.

6.3.3. Delegación de Privilegios y Gestión del Ciclo de Vida

Para evitar que la máquina virtual consuma recursos del servidor de forma permanente cuando no está en uso, se implementó un modelo de encendido bajo demanda mediante la delegación de privilegios en Proxmox.

Se creó un usuario genérico dentro del dominio de autenticación interno del hipervisor. A esta cuenta se le asignó un rol altamente restrictivo, configurado exclusivamente para otorgar permisos de gestión de energía (iniciar, reiniciar y apagar) sobre esta máquina virtual. De esta manera, cualquier miembro autorizado puede iniciar sesión en la interfaz web de Proxmox con este usuario restringido, encender la máquina virtual sin tener visibilidad ni acceso administrativo sobre el resto del centro de datos y, una vez operativa, conectarse a ella mediante su cliente de Escritorio Remoto habitual. De esta manera, se logra replicar a un modelo de gestión bajo demanda muy similar al que se utiliza en plataformas comerciales de nube pública, como Microsoft Azure o Amazon Web Services (AWS).

6.4 Acceso Remoto

Para llevar a cabo la estrategia de conectividad de triple vía diseñada en la arquitectura, se procedió a la implementación y configuración de los protocolos de red privada virtual y los servicios de resolución de nombres necesarios. La configuración de estos componentes da cumplimiento al sexto requisito funcional enfocado en el acceso remoto seguro.

6.4.1. Despliegue de WireGuard y OpenVPN en la Capa de Enrutamiento

La primera línea de conectividad remota se delegó al enrutador físico principal de la infraestructura (TP-Link). Desde su panel de administración, se configuraron y habilitaron los servicios nativos de **WireGuard** [22] (como túnel principal de alto rendimiento) y **OpenVPN** [23] (como túnel de respaldo para maximizar la compatibilidad).

La decisión de alojar estos dos servicios directamente en el equipo de frontera de red presenta una ventaja arquitectónica clave: descarga al servidor Proxmox del procesamiento criptográfico necesario para cifrar y descifrar el tráfico de red, y asegura que la conectividad remota siga operativa incluso si el hipervisor o alguna de sus máquinas virtuales requiere ser reiniciado o apagado.

6.4.2. Resolución de IP Dinámica: Cloudflare DDNS

El despliegue de servidores VPN en una red doméstica presenta un desafío técnico inherente: los proveedores de servicios de internet (ISP) asignan direcciones IP públicas dinámicas que cambian periódicamente. Si la dirección IP pública cambia, los clientes VPN configurados en los dispositivos móviles perderán la conexión al intentar apuntar a una dirección obsoleta.

Para solucionar esta problemática y garantizar el acceso ininterrumpido a WireGuard, OpenVPN, Nextcloud y Emby, se adquirió un nombre de dominio y se implementó un servicio de **DNS Dinámico (DDNS)** apoyado en la infraestructura de Cloudflare.

La implementación se realizó desplegando un Contenedor Linux (LXC) sin privilegios en Proxmox, ejecutando Debian 13. Dado su carácter crítico pero de bajo consumo, se le asignaron recursos mínimos (1 núcleo de CPU, 300 MB de RAM y 3 GB de almacenamiento) y se ubicó en la red de seguridad aislada.

Para su configuración se siguieron estrictos principios de seguridad:

- **Autenticación Restringida:** Se generó una clave de acceso(*Token*) en Cloudflare con permisos limitados exclusivamente a la edición de los registros DNS de la zona del dominio adquirido. Esto evita que, en caso de compromiso, la clave de acceso pueda utilizarse para alterar configuraciones de la cuenta o de otros dominios.
- **Configuración de Red:** Los registros en Cloudflare se configuraron en modo *Solo DNS* (sin el proxy de seguridad web activado) y con el enrutamiento IPv6 deshabilitado, forzando la resolución directa a la dirección IPv4 pública.
- **Políticas de Cortafuegos:** Al encontrarse en la subred de seguridad, se aplicó una política de descarte total para el tráfico entrante. Para el tráfico saliente, se aplicó la política de grupo *DefaultOut*. Esta medida garantiza que este contenedor no tiene visibilidad ni capacidad de comunicación con el resto de los servicios del servidor ni de las redes locales.

Gracias a este servicio, el contenedor comprueba periódicamente la IP pública del enrutador y actualiza automáticamente los registros en Cloudflare. Los clientes VPN se configuran para apuntar al nombre de dominio en lugar de a una IP numérica, asegurando que la conexión siempre se establezca correctamente sin importar los cambios del ISP.

6.4.3. Red en Malla Alternativa: Tailscale

Como complemento a las VPN centralizadas del enrutador, se implementó **Tailscale** [16] dentro de un contenedor LXC basado en Debian 13 y conectado a la red local principal (`192.168.68.0/24`). Este nodo actúa como una vía de acceso independiente capaz de sortear bloqueos de red externos o fallos en el servicio DDNS mediante la perforación de NAT.

Para garantizar su operatividad, a nivel del núcleo de Linux, se habilitó el reenvío de paquetes IP (*IP forwarding*), un requisito técnico indispensable para que la máquina pueda actuar como enrutador dentro de la red privada virtual.

La autenticación del nodo se realizó con la cuenta de administrador principal, y se configuró bajo dos roles operativos fundamentales:

- **Anuncio de Subredes (*Advertise Subnets*):** Esta funcionalidad permite que el contenedor actúe como una pasarela o puente de red. Al anunciar la subred local

(`192.168.68.0/24`) a la red de Tailscale, los dispositivos remotos autorizados pueden acceder a todos los equipos de la red (como el servidor de archivos) de forma transparente, sin necesidad de instalar la aplicación de Tailscale en cada uno de los dispositivos de destino.

- **Nodo de Salida (*Exit Node*):** Al habilitar esta característica, se permite a los usuarios remotos enrutar la totalidad de su tráfico de internet a través de este contenedor. Esto resulta especialmente útil al conectarse desde redes Wi-Fi públicas o de poca confianza (como aeropuertos, hoteles, entre otros).

6.4.4. Conclusión de Conectividad y Acceso de Usuarios

Con la implementación de la actualización dinámica de dominios (Cloudflare DDNS) y el despliegue de los servicios VPN en los dispositivos cliente (ordenadores, portátiles y teléfonos móviles), la infraestructura alcanza su objetivo de ubiquidad.

Actualmente, cualquier miembro autorizado puede establecer una conexión cifrada hacia la red doméstica utilizando el nombre de dominio personalizado. Una vez establecido el túnel criptográfico, el dispositivo remoto opera lógicamente como si se encontrara físicamente conectado al enrutador del hogar. Esto permite a los usuarios montar sus carpetas de red de almacenamiento masivo (Samba), acceder a la interfaz de administración web de Proxmox para encender y apagar la máquina virtual de gestión de archivos, conectarse a dicha estación mediante escritorio remoto (RDP), o realizar tareas de mantenimiento del servidor mediante consola segura (SSH), todo ello manteniendo los puertos internos completamente ocultos hacia internet.

CAPÍTULO 7
Seguridad Avanzada

La exposición de servicios a Internet, como Nextcloud y Emby, requiere una estrategia de mitigación de amenazas robusta. En este capítulo se detalla la implementación de la seguridad perimetral y la prevención de intrusiones, consolidando la arquitectura de "Defensa en Profundidad"[2].

7.1 Seguridad Perimetral: Implementación de OPNsense

Para gestionar el enrutamiento y el filtrado de paquetes entre la red principal, la DMZ y la red de seguridad, se ha implementado **OPNsense** [10] como solución principal para el control y la segmentación del tráfico.

OPNsense es una plataforma de enrutamiento y cortafuegos de código abierto basada en HardenedBSD (una bifurcación de FreeBSD centrada en la seguridad). Sus principales ventajas frente a enrutadores comerciales radican en su motor de filtrado de paquetes de alto rendimiento, soporte nativo para VLANs, actualizaciones frecuentes de seguridad y un ecosistema de complementos *(plugins)* de grado corporativo que permiten integrar capacidades de Sistema de Detección y Prevención de Intrusiones (IDS/IPS).

Dada la carga computacional que exige la inspección profunda de paquetes (DPI), OPNsense se desplegó como una Máquina Virtual en Proxmox con recursos dedicados: **3 núcleos de CPU y 6 GB de memoria RAM**.

7.1.1. Topología Lógica y Segmentación de Interfaces

La máquina virtual cuenta con tres interfaces de red virtuales, segmentadas lógicamente para aislar los flujos de tráfico:

- **WAN (**`192.168.68.20/24`**):** Interfaz externa principal, con puerta de enlace en el router TP-Link `192.168.68.1`. Esta dirección IP se utiliza, entre otras cosas, para el acceso a los servicios externos como Nextcloud o el proxy inverso que redirige el tráfico hacia Emby.
- **WAN - IP Virtual (**`192.168.68.21/24`**):** Se configuró un alias de IP (Virtual IP) en la interfaz WAN. Esta técnica permite recibir tráfico en los mismos puertos estándar (ej. 443), pero derivarlo de forma independiente hacia el segundo Proxy Inverso, ubicado en la red de Seguridad.
- **DMZ / OPT1 (**`10.0.1.1/24`**):** Puerta de enlace para los servicios expuestos a Internet (Nextcloud, Emby y su Proxy Inverso Nginx).

- **Security / LAN (`10.0.3.1/24`):** Puerta de enlace para la red de máxima restricción (Monitorización, Exporters, DDNS y Proxy Inverso interno).

7.1.2. Traducción de Direcciones de Red de Destino (DNAT) y Enrutamiento Entrante

En esta fase de la arquitectura, el objetivo principal es gestionar cómo el tráfico proveniente de la red física local y del exterior alcanza los servicios alojados en las subredes virtuales aisladas. Dado que los equipos externos no tienen visibilidad directa ni capacidad de enrutamiento hacia las redes DMZ (`10.0.1.0/24`) y Seguridad (`10.0.3.0/24`), se implementó un esquema de Traducción de Direcciones de Red de Destino (DNAT), comúnmente conocido como redireccionamiento de puertos [1].

El funcionamiento de este esquema en OPNsense se basa en interceptar el tráfico que llega a la interfaz WAN y reescribir la cabecera IP de destino del paquete, enviándolo hacia la dirección privada correcta del contenedor correspondiente.

Es importante destacar la estrecha integración entre el NAT y el cortafuegos en esta plataforma. OPNsense opera bajo una política restrictiva por defecto (*Default Deny*), lo que significa que todo el tráfico entrante es bloqueado. Sin embargo, al crear una regla de redirección NAT, el sistema genera de forma automática y vinculada la regla de permiso correspondiente en el cortafuegos para esa interfaz WAN. Esto garantiza que el paquete llegue a su destino (el análisis de las políticas de salida y las reglas avanzadas del cortafuegos se abordará en profundidad en las siguientes secciones).

Como se observa en la tabla 7.1, la distribución del tráfico se optimizó haciendo uso de la IP principal y la IP Virtual asignadas a la interfaz WAN:

- **Enrutamiento hacia la DMZ (`192.168.68.20`):** El tráfico dirigido a la IP principal de la WAN se traduce hacia la Zona Desmilitarizada. Por ejemplo, el tráfico HTTPS estándar (puerto 443) se dirige directamente a Nextcloud, mientras que el puerto personalizado 46519 se redirige al Proxy Inverso de Emby.
- **Enrutamiento hacia la red de Seguridad (`192.168.68.21`):** La utilización de la IP Virtual resulta fundamental para evitar colisiones de puertos. Al recibir tráfico en esta segunda dirección, el sistema permite reutilizar puertos idénticos (como el 443 para conexiones web seguras o el 81 para paneles de administración) y derivarlos de forma independiente hacia el segundo Proxy Inverso (NPM) ubicado en la red de Seguridad, manteniendo la separación lógica entre ambos segmentos.

Interfaz	Protocolo	Destino Original	Puerto	IP Redirigida (Backend)	Descripción
WAN	TCP/UDP	192.168.68.20	443	10.0.1.2 (Nextcloud)	Nextcloud
WAN	TCP	192.168.68.20	46519	10.0.1.3 (Nginx Emby)	Proxy Inverso DMZ (Emby)
WAN	TCP	192.168.68.21	443	10.0.3.12 (Nginx PM)	Proxy Inverso Seguridad (HTTPS)
WAN	TCP	192.168.68.20	81	10.0.1.3 (Nginx PM)	Panel de Gestión Proxy DMZ
WAN	TCP	192.168.68.21	81	10.0.3.12 (Nginx PM)	Panel de Gestión Proxy Seguridad

Tabla 7.1: Tabla de reglas principales de redirección de puertos (NAT).

7.2 Sistemas de Prevención de Intrusiones (IDS/IPS)

El tráfico que fluye a través de las reglas de NAT es sometido a un análisis de seguridad profundo mediante la integración conjunta de Suricata [25] y Crowdsec [21].

7.2.1. Inspección Profunda con Suricata

Suricata es un motor de análisis de amenazas de red de alto rendimiento. En este proyecto, se ha instalado como complemento de OPNsense y se ha configurado de la siguiente forma:

- **Modo de Operación:** Configurado en modo IPS utilizando **Netmap** para el acceso directo a la memoria de la tarjeta de red. Al operar en esta capa, Suricata captura los paquetes inmediatamente después de su recepción por el hardware, permitiendo una inspección profunda y el descarte de tráfico malicioso antes de que el núcleo del sistema operativo asuma el procesamiento de los mismos.
- **Modo Promiscuo:** Configurado en la interfaz WAN en modo promiscuo, analizando todo el tráfico entrante.
- **Motor de Expresiones:** Se ha seleccionado **Hyperscan**, una biblioteca de coincidencia de patrones de alto rendimiento desarrollada por Intel. A diferencia de los motores de expresiones regulares convencionales, Hyperscan permite el escaneo simultáneo de miles de firmas en un solo paso de datos, aprovechando el paralelismo a nivel de hardware del procesador. Esta capacidad es fundamental para mantener la integridad de la red de 2.5 Gbps, garantizando que el análisis exhaustivo de reglas complejas no degrade el rendimiento del tráfico ni genere cuellos de botella en el sistema.

Conjuntos de Reglas (Rulesets) y Firmas Aplicadas

Un motor de Prevención de Intrusiones (IPS) como Suricata basa su capacidad de detección en el uso de firmas y conjuntos de reglas (*Rulesets*). Estas reglas actúan como una base de datos de conocimiento dinámico que describe los patrones, anomalías de red y secuencias de bytes características de las ciberamenazas conocidas. Si el IPS detecta una coincidencia, el sistema ejecuta la acción predefinida, en este caso, descartar los paquetes.

Sin embargo, habilitar la totalidad de las reglas disponibles en los repositorios provistos por la comunidad resulta una mala práctica arquitectónica. La activación indiscriminada de firmas provoca un consumo excesivo y constante de recursos computacionales (CPU y memoria RAM), incrementa significativamente la latencia en las conexiones y puede generar una alta tasa de falsos positivos, interrumpiendo el tráfico legítimo de los usuarios.

Por consiguiente, para mantener un equilibrio óptimo entre seguridad perimetral y el rendimiento de la red a 2.5 Gbps, se ha implementado un perfilado selectivo. Se ha habilitado un conjunto de reglas centradas exclusivamente en mitigar los vectores de ataque que amenazan a los servicios que este proyecto realmente expone hacia el exterior (el servidor web de Nextcloud y el proxy inverso Nginx), priorizando los siguientes conjuntos de reglas:

- **Abuse.ch (Feodo Tracker y ThreatFox):** Protege contra conexiones hacia y desde servidores de Comando y Control (C2) de botnets activas y dominios de distribución de malware. La integración de estos feeds proporciona Indicadores de Compromiso (IoCs) orientados a mitigar amenazas avanzadas, como troyanos bancarios y *ransomware*, bloqueando la comunicación maliciosa antes de que pueda producirse la exfiltración de datos o el movimiento lateral.
- **Emerging Threats (ET Open):** Se han activado las categorías *botcc*, *compromised*, *drop* y *dshield* (listas negras de reputación de IP dinámicas). Al operar a nivel de red

y transporte (Capas 3 y 4), estas reglas resultan altamente efectivas incluso frente a tráfico cifrado (TLS). Permiten denegar de forma preventiva el acceso desde bloques de red controlados por cibercriminales, nodos legítimos comprometidos y direcciones IP que se encuentran realizando escaneos masivos o ataques activos en internet.

La efectividad de esta implementación queda demostrada en los registros de auditoría del sistema, donde se observan bloqueos automáticos hacia la IP de la WAN provenientes de redes categorizadas como maliciosas por listas de reputación global:

```
2026-03-29T12:41:02  blocked  WAN  46.151.178.133  -> 192.168.68.20:443
[ET CINS Active Threat Intelligence Poor Reputation IP group 75]

2026-03-29T12:34:42  blocked  WAN  79.124.62.126   -> 192.168.68.20:443
[ET DROP Spamhaus DROP Listed Traffic Inbound group 10]
```

7.2.2. Inteligencia Colaborativa con CrowdSec

Para complementar el enfoque basado en firmas estáticas de Suricata, se ha integrado la versión comunitaria del motor de seguridad **CrowdSec**.

Mientras Suricata inspecciona el interior de los paquetes de red, CrowdSec analiza el comportamiento de las conexiones a través del análisis de registros (logs). Si detecta un comportamiento anómalo (como escaneos de puertos agresivos), bloquea la IP agresora. Además, CrowdSec es alimentado de forma continua por una red de inteligencia colaborativa global (CTI - *Cyber Threat Intelligence*), descargando listas negras de atacantes en tiempo real reportados por miles de servidores en todo el mundo, bloqueándolos preventivamente antes de que interactúen con la infraestructura local.

El funcionamiento de CrowdSec se basa en una arquitectura modular compuesta por un motor de análisis y agentes de remediación denominados *Bouncers*. El motor analiza los registros generados por servicios críticos como OPNsense, procesándolos con escenarios definidos en ficheros YAML. Cuando un patrón de tráfico coincide con un escenario de ataque, como un intento de fuerza bruta o un escaneo de vulnerabilidades web, el motor genera una alerta y el *Bouncer* correspondiente aplica de inmediato una regla de filtrado en la capa de red, y por ende, el descarte de estos paquetes.

Este ecosistema se retroalimenta mediante un modelo de reciprocidad: al tiempo que la infraestructura local se beneficia de la base de datos global de reputación, los intentos de intrusión detectados y validados localmente se comparten de forma anónima con la red de CrowdSec, fortaleciendo la inmunidad colectiva de todos los nodos participantes.

7.2.3. Mitigación de Superficie: Bloqueo Geográfico (GeoBlocking)

Como tercera línea de defensa, se implementó un filtrado de tráfico por geolocalización utilizando la base de datos **MaxMind GeoLite2-Country**.

A través de la configuración *GeoIP* de OPNsense, el firewall descarga periódicamente la base de datos actualizada en formato CSV, la cual asocia bloques de direcciones IP con identificadores de países, compilándolas en tablas ultrarrápidas en memoria.

Se creó el alias `Geo_Arg_Esp_RFC1918`, que agrupa exclusivamente las direcciones IP de España, Argentina y los rangos de red local privada (RFC 1918) [32]. Posteriormente, se configuró una de las reglas de cortafuegos de mayor prioridad en la interfaz WAN:

```
Acción: BLOCK
Interfaz: WAN
Origen: ! Geo_Arg_Esp_RFC1918 (Todo lo que NO coincida con el alias)
```

Esta regla de *descarte* elimina cualquier intento de conexión proveniente de países no autorizados, descartando ataques automatizados (escaneos, botnets) desde países de alto riesgo. Los países con mayor índice de ataque en base a los bloqueos por todas las medidas de prevención en OPNsense son: Estados Unidos, Países Bajos, Alemania, China y Rusia.

7.3 Gestión de Cifrado

La gestión de la capa de seguridad de cifrado de la información en tránsito de los distintos servicios dentro de la infraestructura se centraliza mediante el uso de proxies inversos. Estos actúan como puntos de terminación TLS (*TLS termination proxy*), lo que permite gestionar los certificados de manera unificada, delegar la carga criptográfica y forzar políticas de transporte seguro (como HSTS) antes de que el tráfico alcance los servicios subyacentes.

7.3.1. Nginx Proxy Manager (DMZ) - Servidor Emby

Para aislar y proteger el servidor multimedia Emby, alojado dentro de la zona desmilitarizada (DMZ), se implementó Nginx Proxy Manager (NPM).

Esta arquitectura está diseñada de tal manera que cualquier cliente, independientemente de si origina la petición desde la red LAN principal o desde el exterior a través de Internet, establece una conexión inicial completamente cifrada hacia el proxy inverso. El proxy se encarga de negociar y asegurar este canal de comunicación público.

NPM asume el rol de punto de terminación criptográfica (*TLS termination*). El flujo de datos opera bidireccionalmente de la siguiente manera: cuando el cliente realiza una petición, esta viaja de forma segura y cifrada a través de Internet hasta NPM, luego el proxy se encarga de descifrarla y reenviarla en texto plano (HTTP) al servidor Emby. En sentido inverso, Emby transmite el flujo de video y los metadatos sin cifrar hacia el proxy, siendo NPM quien asume la tarea de cifrar toda la información utilizando el certificado TLS antes de enviarla al exterior. Este proceso aísla de manera efectiva la capa de exposición de la capa de aplicación y, además, libera al servidor multimedia de la carga computacional que exige el cifrado continuo.

Configuración de Enrutamiento y Certificado

- **Certificado TLS:** Se generó un certificado público gratuito a través de Let's Encrypt, asegurando que toda la comunicación entre el cliente y el proxy esté cifrada.
- **Destino Interno:** El tráfico entrante por NPM se redirige al LXC de Emby mediante el esquema `http` hacia la dirección IP interna `10.0.1.4` a través del puerto `8096`.

Configuración General del Proxy

- **Cacheo de Activos (Desactivado):** Se deshabilitó el almacenamiento en caché de los recursos estáticos en el proxy. Para un servidor de transmisión de vídeo como

Emby, donde el contenido es dinámico y los archivos son pesados, cachear activos en el proxy es innecesario y puede consumir recursos de almacenamiento excesivos.

- **Soporte a Websockets (Activado):** Habilitado para permitir la comunicación bidireccional en tiempo real. Esto es crucial para que Emby actualice el estado de reproducción, sincronice dispositivos y envíe notificaciones a la interfaz web sin necesidad de que el cliente recargue la página.
- **Bloqueo de Exploits Comunes (Activado):** Esta directiva añade una capa de seguridad en la configuración de Nginx. Al activarla, el proxy bloquea automáticamente solicitudes maliciosas conocidas, como intentos de inyección SQL, ataques de *Cross-Site Scripting* (XSS), accesos a archivos ocultos (como `.git` o `.env`) y peticiones provenientes de *User-Agents* sospechosos o bots automatizados maliciosos.

Gestión de Cifrado y Protocolos SSL/TLS

Para maximizar la seguridad del enlace y el rendimiento, se aplicaron las siguientes políticas:

- **Force SSL:** Fuerza la redirección automática de cualquier intento de conexión por HTTP (puerto 80) hacia la versión segura HTTPS (puerto 443), garantizando que nunca se transmitan datos en texto plano.
- **HTTP/2 Support:** Habilita el protocolo HTTP/2, el cual permite la multiplexación. Esto significa que el cliente puede solicitar múltiples recursos (imágenes, metadatos, fragmentos de video) a través de una única conexión TCP de forma simultánea, reduciendo drásticamente la latencia y los tiempos de carga de la interfaz.
- **HSTS (HTTP Strict Transport Security):** Esta cabecera de seguridad instruye a los navegadores web para que, en futuras visitas, se comuniquen con el dominio **únicamente** a través de HTTPS durante un período de tiempo definido. Esto mitiga los ataques de degradación de protocolo e intercepciones *Man-in-the-Middle*.
- **HSTS Subdomains:** Extiende la política estricta de HSTS aplicada en el dominio principal a todos sus subdominios, asegurando que cualquier servicio derivado herede esta obligación de cifrado.

Optimizaciones Avanzadas de Transmisión (Nginx)

Dado que el objetivo principal del servidor es la transmisión de medios pesados de manera fluida, se inyectaron las siguientes configuraciones personalizadas en Nginx:

- `proxy_buffering off;` y `proxy_request_buffering off;`
  Deshabilitar el *almacenamiento en búfer* evita que Nginx intente descargar y almacenar el archivo de video en la memoria RAM o en el disco del servidor proxy antes de enviarlo al cliente. Esto reduce drásticamente el consumo de memoria en el proxy, disminuye el tiempo de la primera respuesta y permite que la transmisión comience casi de inmediato.
- `client_max_body_size 0;`
  Establece el tamaño máximo del cuerpo de la petición del cliente a 'ilimitado' (0). Esto es necesario para permitir la carga de archivos pesados, sincronización de medios o envío de grandes volúmenes de datos hacia Emby sin que el proxy corte la conexión lanzando un error 413 (*Cuerpo demasiado grande*).

- `proxy_read_timeout 3600s;`, `proxy_send_timeout 3600s;` y `send_timeout 3600s;` Se amplían los tiempos de espera a 1 hora (3600 segundos). En un entorno de transmisión, un cliente puede pausar un vídeo durante un largo periodo o tener fluctuaciones de red. Estos valores evitan que Nginx cierre prematuramente la conexión TCP por inactividad durante la visualización de medios de larga duración.

7.3.2. Nginx Proxy Manager - Red de Seguridad

Para la administración y monitorización interna de la infraestructura, se desplegó una segunda instancia de Nginx Proxy Manager (NPM) ubicada en la **Red de Seguridad**. A diferencia del proxy de la DMZ, este entorno opera bajo un modelo de acceso estrictamente interno.

El acceso a estos servicios está restringido exclusivamente a los clientes conectados desde la red LAN física principal. Esta instancia de NPM no tiene salida ni visibilidad desde Internet, garantizando su aislamiento mediante la ausencia de redireccionamientos de puertos en el enrutador principal (TP-Link).

Dentro de la Red de Seguridad, este contenedor LXC actúa como el único punto de entrada y exposición, limitando su superficie de ataque a la escucha en los puertos `443` (tráfico HTTPS) y `81` (interfaz de administración del propio proxy).

Enrutamiento y Terminación TLS

Siguiendo la misma filosofía de terminación TLS descrita en la DMZ, NPM es el intermediario criptográfico. El administrador accede a los servicios a través de conexiones seguras validadas por el proxy, y este se encarga de resolver la petición y comunicarse con los servicios.

El enrutamiento del tráfico se realiza dinámicamente mediante la evaluación del subdominio solicitado por el cliente, redirigiendo las peticiones a los siguientes servicios críticos:

- **Proxmox VE:** Interfaz de usuario (UI) para la gestión del hipervisor.
- **OPNsense:** Firewall y enrutador de la red virtualizada.
- **Grafana:** Paneles de visualización de métricas y telemetría de la infraestructura.
- **Prometheus:** Motor de recolección de métricas y monitorización de sistemas.
- **Alertmanager:** Servicio encargado de gestionar, agrupar y enrutar las alertas del sistema (configurado para el envío de notificaciones vía Telegram).
- **NPM UI:** La propia interfaz de administración del proxy inverso.

Gestión de Certificados (Wildcard)

Para simplificar la administración de certificados en la red interna, se optó por la implementación de un **certificado comodín (*Wildcard Certificate*)**.

En lugar de emitir y gestionar un certificado individual para cada servicio, el certificado *wildcard* (ej. `*.dominio.com`) cubre automáticamente todos los subdominios gestionados por NPM. Esto asegura que el administrador en la LAN, siempre reciba una conexión cifrada y validada por el navegador al acceder a cualquier herramienta de monitorización, sin alertas de seguridad por certificados autofirmados, mientras NPM gestiona la comunicación interna con cada contenedor o máquina virtual (figura: 7.1).

Optimización de Tráfico y Caché

A diferencia de la política aplicada en el servidor Emby (DMZ), donde se deshabilitó el caché debido a la naturaleza dinámica y pesada de la reproducción de vídeo, en la Red de Seguridad la estrategia es opuesta:

- **Caché de Activos (Activado):** Dado que el tráfico hacia servicios como Grafana, Proxmox o NPM consiste principalmente en interfaces web estáticas (HTML, CSS, JavaScript, fuentes e iconos), se habilitó la función de almacenamiento en caché en el proxy. NPM almacena temporalmente estos objetos estáticos y los sirve directamente al cliente en peticiones subsecuentes. Esto reduce drásticamente las peticiones a los servicios, minimiza el uso de ancho de banda interno y acelera los tiempos de carga de los paneles de administración.

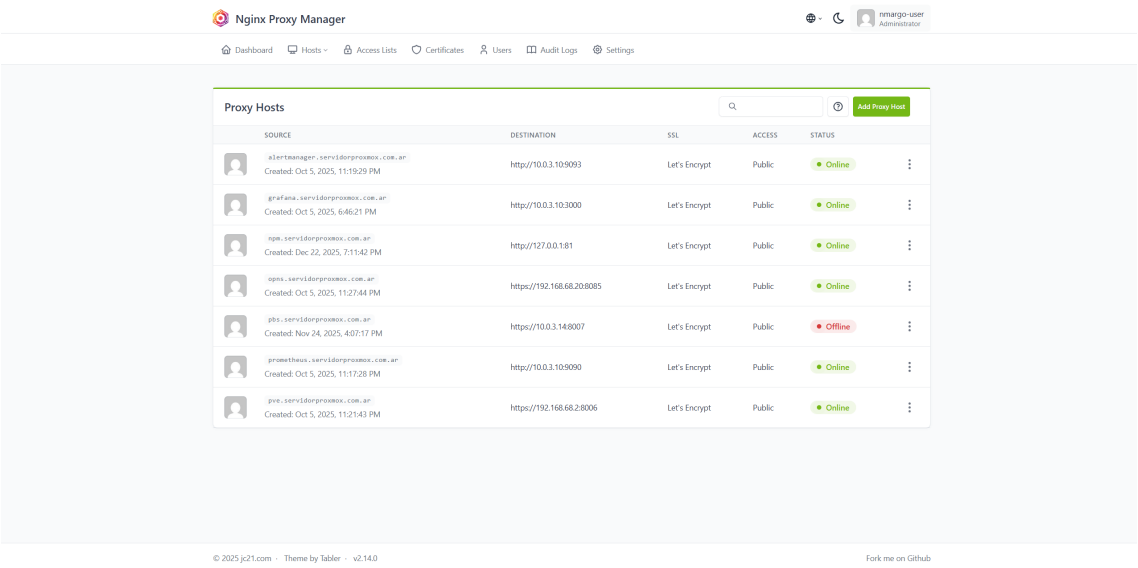

Figura 7.1: Panel de administración de subdominios de NPM

7.3.3. Nextcloud Todo-en-Uno (AIO) y Proxy Caddy

Para la gestión de archivos y colaboración en la nube, se ha implementado la solución *Nextcloud Todo-en-Uno* (AIO). A diferencia de las instancias previas gestionadas con Nginx Proxy Manager, esta sección utiliza una arquitectura de contenedores interconectados donde el servidor web **Caddy** juega un papel fundamental en la seguridad y el transporte de datos.

Orquestación mediante contenedor maestro

El pilar central de esta implementación es el contenedor maestro de Nextcloud AIO. Este contenedor se despliega con acceso directo al *socket* de Docker (`/var/run/docker.sock`), lo que le otorga la capacidad de actuar como un orquestador local.

El contenedor maestro es el encargado de descargar, crear y configurar de forma automática el resto de los componentes del ecosistema (PostgreSQL, Caddy, Nextcloud, etc). Esta metodología garantiza que todos los contenedores se desplieguen en versiones compatibles y con configuraciones validadas directamente por los desarrolladores de Nextcloud, eliminando errores de configuración manual y asegurando un entorno óptimo para producción.

Caddy y Apache: Coexistencia y Configuración de Fábrica

En esta arquitectura, el servidor **Caddy** no se despliega como una entidad aislada, sino que reside dentro del mismo contenedor que el servidor **Apache** (`nextcloud-aio-apache`).

- **Caddyfile de Referencia:** El archivo de configuración `Caddyfile` utilizado es el proporcionado y mantenido por los desarrolladores de Nextcloud AIO. Esto asegura que todas las directivas de seguridad, optimizaciones de rendimiento y reglas de reescritura necesarias para el correcto funcionamiento de Nextcloud estén presentes sin intervención del administrador.
- **Cifrado TLS Automatizado:** Caddy se encarga de la negociación y renovación de los certificados TLS, abstrayendo la complejidad criptográfica y asegurando que la conexión sea siempre segura desde el primer momento.

Seguridad por Defecto

Gracias al uso de la configuración oficial de Caddy, se aplican automáticamente medidas de securización que protegen la instancia de Nextcloud:

- **HSTS (Strict-Transport-Security):** Se implementa una cabecera estricta con un max-age de un año, obligando a los clientes a utilizar exclusivamente HTTPS.
- **Ofuscación de Cabeceras:** El `Caddyfile` está configurado para eliminar las cabeceras `Server` y `X-Powered-By`, reduciendo la información expuesta sobre la infraestructura ante posibles escaneos de vulnerabilidades.
- **Optimización de Empuje (Client Push):** El proxy está preparado para trabajar en conjunto con el contenedor de *Client Push*, permitiendo actualizaciones de la interfaz en tiempo real de manera eficiente.

La adopción de este modelo consolidado permite que la infraestructura herede automáticamente las mejores prácticas de seguridad web sin la fricción de una configuración manual extendida. Al unificar el servidor de aplicaciones y el proxy en un flujo de trabajo orquestado por el propio sistema, se garantiza una capa de transporte segura y optimizada, donde la consistencia de los certificados TLS y la correcta aplicación de las cabeceras de seguridad se mantienen vigentes de forma transparente ante futuras actualizaciones o cambios en el entorno de ejecución.

CAPÍTULO 8
Observabilidad y Alertas

Una infraestructura de estas características requiere supervisión constante y proactiva. En este capítulo se detalla el diseño y despliegue de las herramientas de Observabilidad, un ecosistema centralizado encargado de recopilar, almacenar y visualizar métricas de rendimiento en tiempo real, así como de emitir alertas ante comportamientos anómalos.

Para mantener el principio de aislamiento estricto, todo el ecosistema de monitorización opera dentro de un único Contenedor Linux (LXC) ubicado en la red de Seguridad, utilizando Docker.

8.1 Sistema de Monitorización: Arquitectura y Conceptos Base

Antes de abordar el despliegue técnico de Prometheus, es necesario comprender la arquitectura lógica que permite a un contenedor aislado supervisar un hipervisor completo y sus capas de almacenamiento. El sistema se basa en un modelo de extracción, donde un servidor centralizado consulta periódicamente a diferentes agentes distribuidos.

8.1.1. El Flujo de Observabilidad

La topología de monitorización se compone de tres pilares fundamentales:

1. **Agentes de Exportación (Exporters):** Son pequeños servicios instalados directamente en el hipervisor. Su única función es traducir los datos de hardware (temperatura, uso de CPU, estado de los discos, entre otros) a un formato de texto plano comprensible por el sistema de monitorización, exponiéndolos a través de un puerto y el protocolo HTTP.

2. **Motor de Recopilación y Almacenamiento (Prometheus y su TSDB):** Es el núcleo del sistema, desplegado en Docker dentro del contenedor LXC. Prometheus se conecta periódicamente a los puertos HTTP de los exportadores para extraer el estado actual del hardware y los servicios. En lugar de volcar estos datos en un motor relacional tradicional, Prometheus los importa directamente en su propia Base de Datos de Series Temporales (TSDB - *Time Series Database*). Una TSDB es un modelo de almacenamiento diseñado específicamente para manejar el registro masivo, constante y secuencial de eventos a lo largo del tiempo. Cada métrica recolectada se almacena como un punto de datos inmutable compuesto por tres elementos: un valor numérico, una marca de tiempo (*timestamp*), y un conjunto de etiquetas multidimensionales que otorgan contexto (por ejemplo, `hdd='sda', mode='idle'`). Esta

arquitectura opera mediante una lógica de 'solo adición', agrupando y comprimiendo la información en memoria antes de volcarla a disco en bloques compactos. Esto permite registrar miles de métricas por segundo con un consumo de E/S muy bajo, justificando plenamente el ajuste previo del `recordsize` a 32K en el conjunto de datos ZFS para evitar la amplificación de escritura.

3. **Visualización (Grafana):** Plataforma conectada a la base de datos de Prometheus, encargada de traducir las series temporales en paneles gráficos (*Dashboards*) interpretables por el administrador.

8.1.2. Preparación del Entorno: Almacenamiento y Permisos (LXC Unprivileged)

El alojamiento de bases de datos de series temporales (como la de Prometheus) genera un volumen constante de pequeñas operaciones de escritura. Para optimizar este comportamiento, se aprovisionó un conjunto de datos de ZFS dedicado (`datapoolSSD1TB/docker/obs_data`) con un **tamaño de bloque (`recordsize`) ajustado a 32K**. Esta reducción desde los 128K predeterminados es una optimización crítica que mitiga la amplificación de escritura en operaciones transaccionales.

Adicionalmente, el contenedor fue instanciado en modo **sin privilegios (*Unprivileged*)** por motivos de seguridad. El usuario administrador interno del LXC (UID 0) no tiene equivalencia de *root* en el hipervisor, sino que opera bajo el UID 100000. Para que Docker, ejecutándose dentro del LXC, pudiera escribir datos persistentes en el disco físico, se aplicaron dos configuraciones de mapeo:

- **Propiedad del conjunto de datos:** Se asignó la propiedad del directorio ZFS en el hipervisor al usuario virtual (`chown 100000:100000`).
- **Mapeo de Identidades (*ID Mapping*):** Se agregó al archivo de configuración del contenedor el mapeo de los identificadores de usuario y grupo (UID/GID) y se montó directamente el conjunto de datos ZFS como un volumen en la ruta `/var/lib/docker`.

De este modo, se obtiene un entorno Docker completamente funcional con almacenamiento persistente de alto rendimiento, sin comprometer el aislamiento de seguridad del anfitrión.

8.2 Despliegue de Agentes de Recopilación (Exporters)

Dado que Prometheus está confinado en un contenedor, es ciego al hardware subyacente. Para solucionar esto, se instalaron dos agentes directamente en el sistema operativo base de Proxmox.

8.2.1. Prometheus Node Exporter

Se instaló mediante los repositorios oficiales de Debian para exponer las métricas del sistema operativo y del hardware general (uso de CPU, RAM, red). Adicionalmente, se ejecutó la utilidad `sensors-detect` en el anfitrión para cargar los módulos del kernel correspondientes, permitiendo a Node Exporter leer y transmitir la temperatura del procesador y las revoluciones de los ventiladores del chasis en tiempo real a través del puerto `9100/tcp`.

8.2.2. ZFS Exporter

Dado que Node Exporter no posee visibilidad profunda sobre sistemas de archivos avanzados, se compiló e instaló un exportador dedicado (`zfs_exporter`). Este agente expone métricas detalladas sobre la salud de las unidades de almacenamiento con ZFS a través del puerto `9134/tcp`.

8.2.3. Políticas de Cortafuegos Interno

Se crearon reglas específicas en el cortafuegos de Proxmox que permiten exclusivamente que la dirección IP de origen del contenedor de observabilidad pueda establecer conexiones TCP hacia los puertos 9100 y 9134 del hipervisor físico (192.168.68.2). Todo intento de consulta desde cualquier otra IP de la red de seguridad es descartado.

8.3 Implementación del Servidor Prometheus

Con los agentes funcionales y el almacenamiento ZFS correctamente mapeado, se procedió a la instalación de Docker sobre una plantilla de Debian 13 en el contenedor LXC.

El despliegue de los servicios se estructuró mediante un archivo `docker-compose.yml`, garantizando la reproducibilidad y el control de versiones de la infraestructura como código.

8.3.1. Orquestación del Contenedor Prometheus

El servicio de Prometheus se aprovisionó como el núcleo de las herramientas de monitorización dentro de una red virtual interna de Docker, exponiendo únicamente su interfaz web a través del puerto 9090. Se evitó el uso de configuraciones predeterminadas, inyectando los parámetros operativos a través de directivas de inicio y volúmenes enlazados (*Bind Mounts*).

Gestión de Retención y Almacenamiento Persistente

Para salvaguardar el rendimiento del almacenamiento y evitar la saturación del disco a largo plazo, se estableció de la siguiente forma la política de ciclo de vida de los datos:

Retención en la TSDB: Mediante argumentos de ejecución en el orquestador, se limitó explícitamente el tiempo de retención de la base de datos de series temporales a **200 horas** (poco más de 8 días). Esto garantiza un volumen de datos suficiente para el análisis de tendencias semanales sin penalizar la capacidad del SSD.

8.3.2. Estructura de Recolección y Trabajos Configurados

El comportamiento de Prometheus está gobernado por su archivo de configuración principal (`prometheus.yml`). Se ha diseñado bajo un intervalo de recolección global y de evaluación de reglas de 15 segundos, asegurando una resolución de métricas cercana a tiempo real.

La topología de extracción se dividió lógicamente en cinco trabajos específicos para auditar todas las capas de la infraestructura:

1. **Infraestructura Física:** Dos trabajos dedicados a consultar directamente la IP del nodo anfitrión (192.168.68.2). El primero (`proxmox-node`) captura las métricas de hardware general en el puerto 9100, y el segundo (`zfs`) extrae la telemetría del sistema de archivos avanzado en el puerto 9134.
2. **Capa de Virtualización (PVE Exporter):** Se implementó una configuración de recolección para auditar la API nativa de Proxmox. Dado que el hipervisor no exporta métricas de Prometheus por defecto, se utiliza un contenedor LXC intermediario ejecutando la herramienta Prometheus PVE Exporter en la red de seguridad (`10.0.3.11:9221`). Mediante técnicas de etiquetado dinámico, Prometheus se comunica con este LXC y le ordena auditar a la IP anfitriona, modificando los atributos en tiempo real para que las métricas parezcan provenir directamente del clúster de virtualización. Para aligerar la carga de la API, este trabajo en particular tiene su propio intervalo de sondeo ampliado a 45 segundos.
3. **Auditoría Interna del Stack:** Los dos últimos trabajos (`grafana` y `prometheus`) se encargan de monitorizar la propia salud y rendimiento de los contenedores de observabilidad utilizando la resolución DNS interna de Docker.

8.3.3. Motor de Reglas y Enrutamiento de Alertas

Finalmente, el archivo principal enlaza con un directorio de reglas estáticas (`/etc/prometheus/rules/alertas.yml`). Este archivo contiene expresiones lógicas y umbrales predefinidos (como uso excesivo de CPU, caída de discos o indisponibilidad de contenedores). Cada 15 segundos, Prometheus evalúa la base de datos contra estas reglas. Si una condición de alerta se cumple, el evento se delega inmediatamente al servicio de **Alertmanager** (configurado estáticamente en el puerto 9093 dentro de la misma red virtual) para su posterior gestión y notificación al administrador.

8.4 Visualización de Métricas

Tras la recolección y el almacenamiento de los datos en la base de datos de series temporales, es indispensable contar con una interfaz que permita interpretar esta información de manera ágil. Para cumplir con este propósito se ha implementado Grafana, una plataforma de código abierto líder en la industria para la visualización y el análisis de datos operativos.

Grafana funciona como la capa de presentación del ecosistema de observabilidad. Su propósito principal es conectarse a múltiples fuentes de datos, consultar la información bruta y traducirla en representaciones gráficas comprensibles para el administrador del sistema. En esta infraestructura, Grafana se integra de forma nativa con Prometheus. Mediante el uso del lenguaje de consultas propio de Prometheus, conocido como PromQL, Grafana solicita los valores numéricos almacenados y los representa visualmente en gráficos en tiempo real. Esta separación entre la base de datos y la capa de visualización aporta una ventaja arquitectónica fundamental, ya que permite escalar ambos servicios de forma independiente y evita sobrecargar el motor de recolección con tareas de renderizado visual.

8.4.1. Paneles de Visualización y Comunidad

El núcleo funcional de Grafana son los cuadros de mando o paneles de visualización. Estos paneles son espacios de trabajo configurables compuestos por diversos elementos

interactivos, como gráficos de líneas, medidores de aguja, tablas de estado y mapas de calor.

Existen dos métodos para construir estas interfaces. El primero consiste en la creación manual, donde el administrador diseña cada gráfico desde cero escribiendo las fórmulas matemáticas exactas para extraer la métrica deseada. El segundo método, que representa una de las mayores ventajas operativas de Grafana, es la utilización de plantillas preconfiguradas provistas por la comunidad de usuarios. Cualquier usuario puede exportar su cuadro de mando como un archivo de texto estructurado en formato JSON y compartirlo de manera pública. Esto permite importar configuraciones altamente complejas en cuestión de segundos, acelerando drásticamente la puesta en marcha del sistema.

8.4.2. Cuadros de Mando Implementados

Para auditar cada capa de la infraestructura, se han importado y adaptado tres paneles de visualización principales creados por la comunidad, los cuales cubren todos los aspectos críticos del servidor de manera unificada. Al convivir en el mismo entorno de contenedores, Grafana extrae la información exclusivamente de la base de datos central de Prometheus, consultando las métricas obtenidas mediante los diferentes trabajos de recolección configurados:

- **Panel Completo del Exportador de Nodo:** Alimentado por las métricas del trabajo de recolección denominado `proxmox-node`, este cuadro de mando proporciona una visión exhaustiva del estado de los componentes físicos del sistema operativo anfitrión. Permite observar el porcentaje de uso de cada uno de los seis núcleos del procesador, el consumo de la memoria principal, el ancho de banda saturado en la tarjeta de red de 10 gigabits, las operaciones de lectura y escritura por segundo de los discos y, de vital importancia para la estrategia de refrigeración por presión positiva diseñada previamente, las temperaturas exactas de los sensores térmicos de la placa base, el procesador, la placa de red y de las unidades de almacenamiento (véase la Figura 8.1).

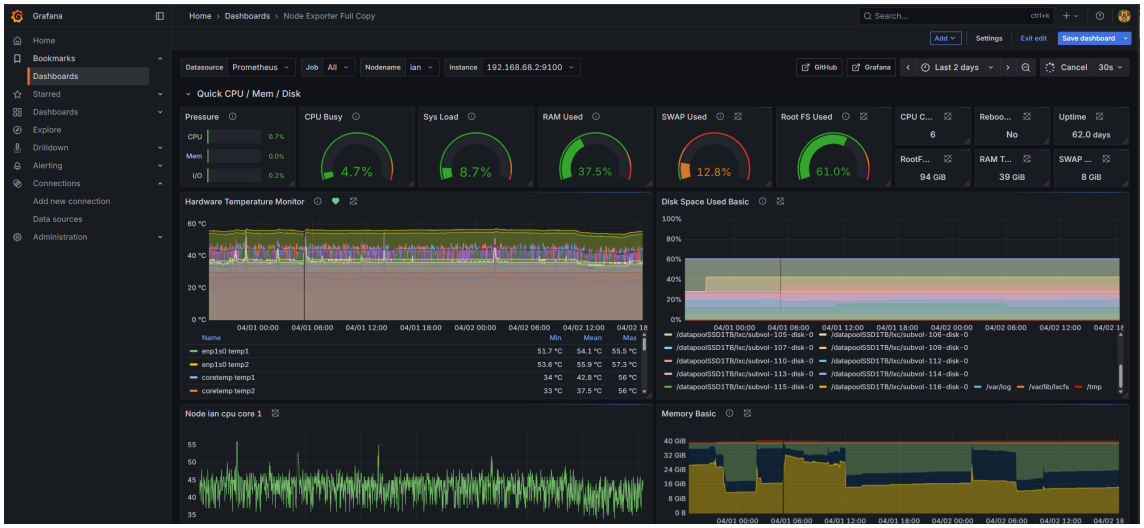

Figura 8.1: Panel de visualización del exportador de nodo

- **Panel de Proxmox mediante Prometheus:** Este panel consolida los datos capturados por el trabajo `pve`, el cual interroga a la API de Proxmox a través del contenedor intermediario detallado anteriormente. Ofrece una perspectiva centrada en la virtualización, permitiendo visualizar el estado de ejecución, el tiempo de actividad

ininterrumpida y el consumo individual de recursos de cada máquina virtual y contenedor Linux. Es una herramienta fundamental para detectar si un servicio expuesto en la red perimetral, como la plataforma de archivos o el servidor multimedia, está consumiendo más recursos de los estimados. Además, brinda información detallada del uso de recursos del procesador, la memoria, el almacenamiento, entre otros (véase la Figura 8.2).

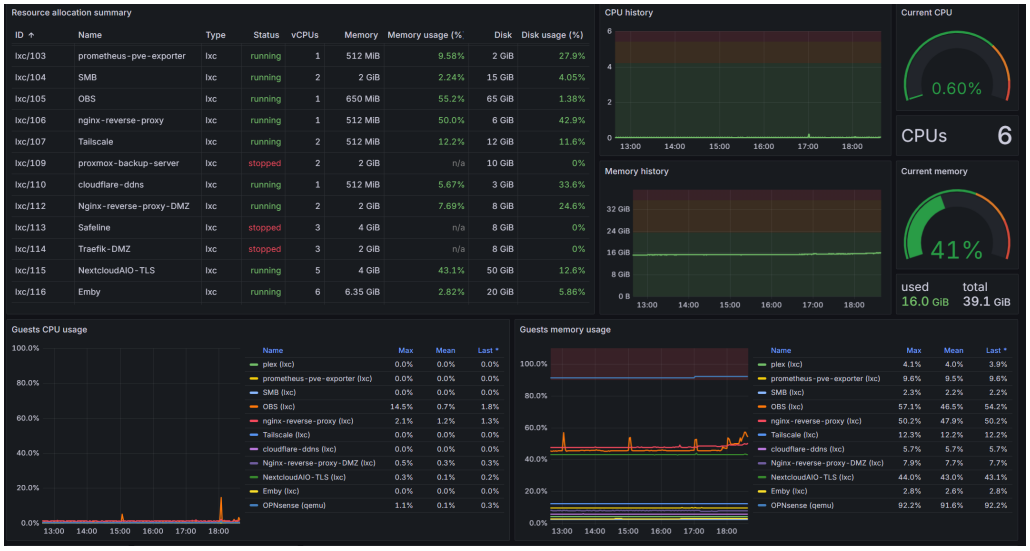

Figura 8.2: Panel de visualización del exportador de Proxmox mediante Prometheus

- **Panel del Exportador de ZFS:** Sustentado por la telemetría del trabajo `zfs`, está dedicado exclusivamente a monitorizar la salud del sistema de archivos en el servidor físico. Este panel traduce las métricas del almacenamiento en indicadores claros sobre la capacidad disponible de los distintos conjuntos de datos y el nivel de fragmentación de los bloques de datos (véase la Figura 8.3).

Figura 8.3: Panel de visualización del exportador de ZFS

8.5 Gestión de Alertas

La recolección pasiva de métricas resulta insuficiente en un entorno de producción si no va acompañada de un sistema proactivo capaz de notificar al administrador ante

eventos críticos. Para cubrir esta necesidad, la arquitectura de observabilidad incorpora el gestor de alertas, conocido técnicamente como *Alertmanager*, el cual se encarga de canalizar las advertencias hacia dispositivos móviles mediante la plataforma de mensajería Telegram.

8.5.1. Arquitectura y Flujo de Comunicación

El gestor de alertas actúa como un intermediario inteligente. Su función principal no es evaluar los datos, sino recibir las notificaciones de los motores de evaluación, agruparlas y enrutarlas hacia el canal de comunicación adecuado.

El flujo de comunicación se origina principalmente en el servidor Prometheus. Como se definió en su archivo de configuración, Prometheus evalúa de forma ininterrumpida las expresiones lógicas del archivo de reglas cada 15 segundos. Cuando una métrica supera el umbral establecido, Prometheus transfiere la advertencia al puerto 9093, donde reside el gestor de alertas. Adicionalmente, el ecosistema permite que la plataforma de visualización Grafana actúe como un motor de evaluación secundario. Grafana puede generar notificaciones basadas en los límites gráficos de sus paneles y enviarlas al mismo gestor de alertas, logrando una estrategia de aviso unificada y redundante.

8.5.2. Configuración de Enrutamiento y Tiempos

Para evitar la saturación de mensajes y garantizar la eficacia de las advertencias, se implementaron directivas estrictas de tiempo en el archivo de configuración del servicio:

- **Intervalo de Agrupación:** Configurado en un minuto, este parámetro define el tiempo que el servicio espera para agrupar múltiples alertas nuevas del mismo tipo en un solo bloque de texto, evitando enviar decenas de notificaciones individuales simultáneas ante un fallo en cascada.
- **Intervalo de Repetición:** Establecido en treinta minutos, determina la frecuencia con la que el sistema reenviará una notificación si la incidencia no ha sido resuelta por el administrador, asegurando que los fallos críticos no pasen desapercibidos.
- **Canal de Recepción (Telegram):** Se configuró un receptor específico apuntando a la interfaz de programación de aplicaciones de Telegram. Mediante el uso de un código de autorización y un identificador de canal privado, un agente automatizado envía los mensajes directamente al teléfono móvil del administrador, garantizando una recepción inmediata sin depender del correo electrónico.

8.5.3. Reglas Críticas Implementadas

El conjunto de reglas diseñado tiene como prioridad salvaguardar la integridad de los datos y prevenir el desgaste acelerado de los componentes físicos. Se han configurado alertas operativas específicas para los siguientes escenarios de riesgo:

- **Disponibilidad de los agentes:** Se genera una advertencia inmediata si Prometheus detecta que algún exportador ha dejado de responder, indicando una posible caída del hipervisor o de un contenedor crítico.
- **Temperatura del procesador central:** Notificación si el procesador supera los 55°C, indicando una carga de trabajo anómala o una deficiencia en la presión positiva del flujo de aire.

- **Temperatura de la tarjeta de red:** Advertencia si el controlador de 10 GbE, propenso a generar calor bajo altas tasas de transferencia, supera la barrera de los 65°C.
- **Temperatura de las unidades mecánicas:** Alerta configurada a los 40°C para los discos de alta capacidad de 12 y 3 terabytes, previniendo la dilatación térmica de los platos magnéticos.
- **Temperatura de las unidades de estado sólido:** Advertencia establecida en 35°C o superior, protegiendo las celdas de memoria frente al calor residual del chasis.
- **Saturación de almacenamiento:** Notificación prioritaria si cualquier unidad, ya sea el sistema de archivos avanzado ZFS o las particiones ext4, alcanza o supera el 80 % de su capacidad lógica.
- **Diagnóstico de integridad:** Alerta de máxima severidad si la tecnología de supervisión y reporte automático de cualquier disco reporta atributos de fallo inminente durante sus pruebas de salud, permitiendo la sustitución preventiva de la unidad antes de una pérdida de datos.

La implementación de este gestor de alertas transforma la monitorización pasiva en una estrategia de mantenimiento preventivo activo. Al integrar notificaciones instantáneas mediante dispositivos móviles y establecer umbrales rigurosos, el administrador adquiere la capacidad de anticiparse a fallos mecánicos, incrementos anómalos de temperatura y saturaciones lógicas antes de que impacten en la disponibilidad de las aplicaciones. Esta capa de aviso temprano consolida la robustez de toda la infraestructura, garantizando la preservación continua de los datos y estableciendo el escenario ideal para abordar el último mecanismo de resiliencia del proyecto, el cual se enfoca en las políticas de retención y las copias de seguridad.

CAPÍTULO 9
Copias de Seguridad

Este capítulo detalla la arquitectura de respaldo diseñada para asegurar que ninguna falla en el *hardware* o *software* resulte en la destrucción definitiva de la información, garantizando la capacidad de reconstruir el sistema completo mediante copias de seguridad consistentes.

9.1 Estrategia de Continuidad: Conceptos y Clasificación de Activos

El primer paso para garantizar la continuidad operativa es distinguir entre la redundancia del sistema y la retención segura de los datos. Mientras una busca evitar que el servicio se caiga, la otra asegura que la información no se pierda de forma permanente. Por lo tanto, su mecánica de trabajo es diferente:

- **Copia de Seguridad (*Backup*):** Representa un estado histórico de la información capturado en un punto exacto del tiempo (*Point-in-Time*) y almacenado en un medio físico independiente. Su valor fundamental reside en la **regresión temporal**, permitiendo reconstruir estados del sistema o recuperar archivos que han sido modificados o eliminados en el origen. A diferencia de un simple volcado de datos, una *copia de seguridad* suele ser un archivo comprimido y empaquetado (como los archivos `.vzdump` en Proxmox), lo que facilita su transporte y almacenamiento a largo plazo. Las copias de seguridad son ideales ante desastres físicos totales o corrupciones lógicas, ya que el historial de datos permanece completamente aislado de cualquier fallo eléctrico, mecánico o de software que afecte a la unidad de producción principal.

- **Replicación de Datos:** Consiste en la sincronización constante o periódica de contenidos entre dos o más nodos o unidades. Su objetivo primordial es la alta disponibilidad: si la unidad A queda fuera de servicio, la unidad B posee una copia idéntica y lista para asumir la carga de trabajo de manera casi inmediata. No obstante, es necesario entender que la replicación no constituye un sistema de copias de seguridad. Debido a su naturaleza de 'espejo', cualquier error humano, borrado accidental o infección por *ransomware* en el origen se propaga poco tiempo después al destino en tiempo real o en el próximo ciclo de sincronización. Por lo tanto, la replicación protege la continuidad del servicio ante fallos de hardware, pero no protege la integridad de los datos ante eventos lógicos maliciosos, fallos en el *software*/procesos o errores humanos.

- **Instantánea (*Snapshot*):** Es un registro lógico de los metadatos y punteros del sistema de archivos en un momento preciso. Su ejecución es prácticamente instantánea y no consume espacio adicional de almacenamiento en su creación inicial, ya que solo comienza a ocupar bloques de disco a medida que los datos originales cambian. Una característica fundamental en entornos ZFS es que estas instantáneas son inherentemente **inmutables** (de solo lectura). Esto significa que su contenido no puede ser alterado, sobreescrito ni cifrado por procesos externos, ofreciendo una capa de defensa contra ataques de *ransomware* o eliminaciones accidentales. Se pueden distinguir dos escenarios operativos:

  - **Instantánea Local:** Reside en el mismo conjunto de discos que los datos activos. Es la herramienta ideal para la **recuperación operativa rápida** (por ejemplo, para revertir el sistema tras una actualización de *software* fallida), pero carece de autonomía física, si el disco original falla, la instantánea se pierde.
  - **Instantánea Replicada como Copias de Seguridad:** Mediante mecanismos de transporte de bloques (como `zfs send/receive`), una instantánea puede ser enviada a un medio físico independiente. En este caso, la instantánea adquiere autonomía y se comporta como una `copia de seguridad` incremental: si la unidad original sufre un fallo total, los datos pueden ser reconstruidos íntegramente en un nuevo medio a partir de la cadena de instantáneas almacenadas en la unidad secundaria.

Esta distinción permite implementar una estrategia en capas, donde las *instantáneas* locales ofrecen agilidad para la corrección de fallos inmediatos, mientras que las *copias de seguridad* garantizan la supervivencia de la información a largo plazo.

Para optimizar los recursos del servidor, se ha realizado una identificación de activos críticos de los cuales es necesario tener algún tipo de respaldo:

1. **Virtualización:** Comprende las máquinas virtuales y los contenedores Linux (LXC), los cuales contienen el sistema operativo, las aplicaciones, las propias configuraciones y los servicios en ejecución.

2. **Volúmenes Docker:** Abarca los conjuntos de datos ZFS que contienen las bases de datos de Docker, los volúmenes de observabilidad y el contenido de datos de nextcloudAIO.

3. **Configuración del Hipervisor:** Incluye el estado del nodo físico Proxmox VE, abarcando las definiciones de almacenamiento, los esquemas de redes virtuales (*Linux Bridges*), las reglas de cortafuegos, los archivos de configuración, entre otros. Su respaldo es vital para reconstruir la lógica de la infraestructura en caso de un fallo total del disco de sistema.

4. **Datos Operativos y Documentos (HDD 3TB):** Engloba la información de uso frecuente y personal, como documentos, proyectos activos y carpetas privadas de los usuarios.

5. **Contenido Histórico (HDD 12TB):** Representa el repositorio de larga duración que contiene más de 20 años de material digital acumulado.

La Figura 9.1 sintetiza de forma visual la arquitectura de almacenamiento y la estrategia de copias de seguridad descrita en este capítulo. En ella se representa la segregación funcional de las cuatro unidades físicas del servidor Proxmox VE, así como los flujos de respaldo y replicación entre ellas y el nodo secundario.

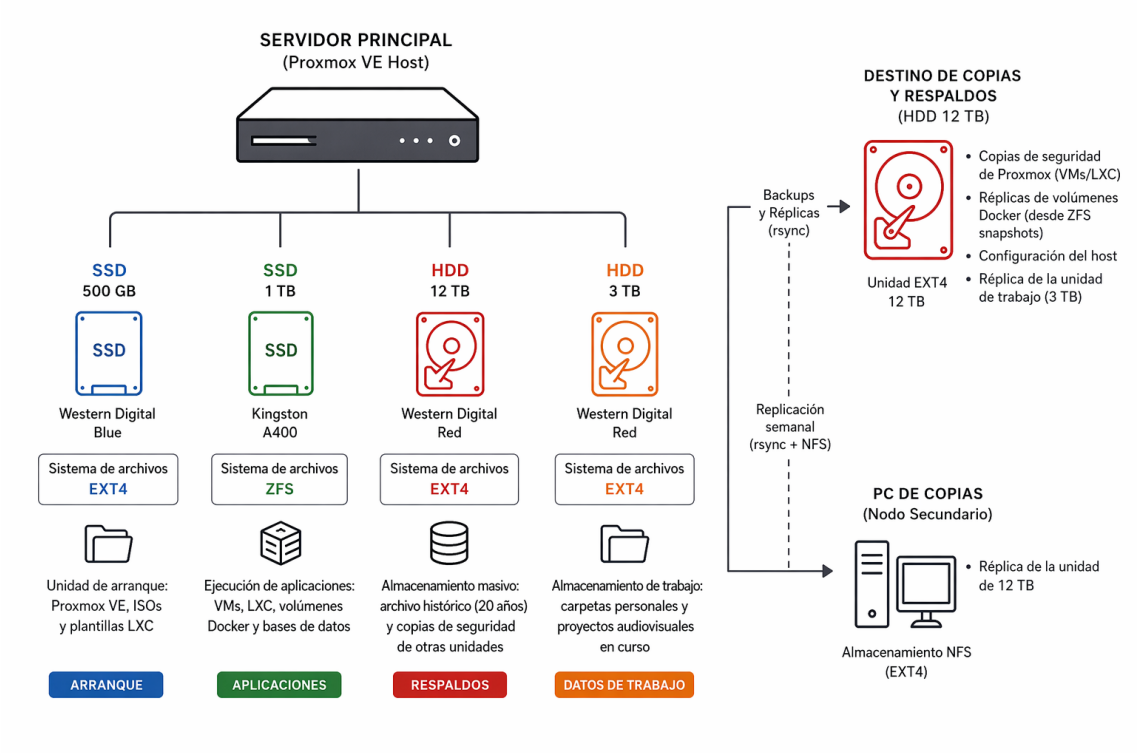

Figura 9.1: Diagrama de contenido y copias de seguridad de las unidades de almacenamiento.

9.2 Modalidades de Respaldo en Proxmox VE

Proxmox VE ofrece tres estrategias nativas para la ejecución de copias de seguridad a través de la herramienta `vzdump`. La elección de cada una impacta directamente en el RTO (*Recovery Time Objective*, o Tiempo Objetivo de Recuperación) y en la integridad transaccional de los datos.

Cabe precisar que estas modalidades operan de forma específica sobre la capa de virtualización de la infraestructura, protegiendo exclusivamente las Máquinas Virtuales y los Contenedores Linux (LXC) alojados en el hipervisor. Para respaldar estos activos, se puede optar por una de las siguientes opciones:

- **Modo Detenido:** Es el método más robusto en términos de consistencia. El sistema apaga completamente el contenedor o VM, realiza la copia de los datos y vuelve a iniciar el servicio.
  - **Ventajas:** Garantiza una consistencia total del sistema de archivos y de las bases de datos, ya que no hay archivos abiertos ni transacciones en memoria durante el proceso. Es la opción recomendada para aplicaciones críticas donde la integridad del dato es prioritaria sobre la disponibilidad.
  - **Desventajas:** Provoca un tiempo de inactividad proporcional al tamaño del disco y a la velocidad de escritura en el destino.
- **Modo Suspendido:** Esta modalidad pausa temporalmente la ejecución de la instancia y realiza una copia del estado actual. En el caso de las VMs, Proxmox intenta congelar el estado de la memoria RAM.

- **Ventajas:** Reduce significativamente el tiempo de inactividad en comparación con el modo *detenido*, ya que no requiere un arranque completo del sistema operativo tras la copia.
- **Desventajas:** Existe un riesgo residual de inconsistencia en sistemas de archivos complejos si las escrituras no se descargan correctamente del caché a la persistencia. Además, el tiempo de pausa puede ser notable en instancias con gran asignación de memoria RAM.

- **Modo Instantánea:** Es la modalidad 'en caliente' que aprovecha las capacidades nativas de ZFS y QEMU. Realiza la copia mientras la instancia sigue operativa, utilizando el mecanismo transaccional de *copia-en-escritura* (CoW) propio de ZFS, el cual preserva los bloques de datos estáticos en el estado exacto de la instantánea, mientras que las operaciones nuevas se escriben de forma transparente en bloques libres.

  - **Ventajas: Disponibilidad total del servicio**. No hay interrupción visible para el usuario final. Al trabajar sobre ZFS, la creación de la instantánea es prácticamente inmediata, minimizando el impacto en el rendimiento del *host* y en la disponibilidad del servicio.
  - **Desventajas:** Al realizar la copia con el sistema encendido, existe el riesgo de que la información no sea totalmente coherente. Esto ocurre porque, en el momento exacto de la instantánea, puede haber datos en la memoria RAM que aún no se han guardado en la unidad de almacenamiento.
    - **Integridad tras fallo eléctrico:** Si se restaura una copia de este tipo, el sistema operativo se comportará como si hubiera sufrido un apagón repentino. Al arrancar, es probable que deba realizar una comprobación automática de errores en el disco para asegurarse de que no hay archivos corruptos.
    - **Necesidad de coordinación:** En las máquinas virtuales, para que la copia sea perfecta, se necesita un agente interno (llamado *Guest Agent*). Este programa avisa al sistema operativo justo antes de la copia para que escriba rápidamente todo lo que tenga pendiente en la memoria hacia el disco y detenga las escrituras por un instante.

9.3 Políticas de Copias de Seguridad de Proxmox VE

Para proteger los contenedores y máquinas virtuales alojados en el disco de estado sólido, se ha diseñado un cronograma automatizado de copias de seguridad que balancea la disponibilidad del servicio con la consistencia de los datos. El destino de almacenamiento de todas estas copias de seguridad se ha configurado de forma estricta hacia el disco mecánico de 12 terabytes. Esta decisión arquitectónica cumple un doble propósito: evita el desgaste por escritura en las celdas de memoria del disco de estado sólido y garantiza que la copia de seguridad resida en un soporte magnético físicamente distinto, cumpliendo con la premisa básica de la protección contra fallos de hardware de la unidad de almacenamiento.

Las políticas de retención se gestionan íntegramente desde el planificador del centro de datos de Proxmox VE, asegurando que las copias obsoletas se depuren automáticamente sin intervención manual, evitando la saturación del disco mecánico. El esquema de retención es el siguiente:

- **Retención Diaria (Modo Instantánea):** Se ejecuta una copia en caliente todos los días para cada máquina y contenedor en uso. Se retienen únicamente los últimos dos días. Su objetivo es la recuperación inmediata frente a errores humanos, actualizaciones fallidas o errores de configuración recientes.

- **Retención Semanal (Modo Detenido):** Se ejecuta una vez por semana apagando temporalmente los servicios para garantizar la consistencia transaccional absoluta de las bases de datos internas. Se retienen tres copias semanales.

- **Retención Mensual (Modo Detenido):** Diseñado para recuperación de configuraciones antiguas. Se almacenan dos copias mensuales.

9.4 Protección de Datos Transaccionales con Sanoid

Los datos persistentes de los volúmenes de Docker, alojados en el conjunto de datos `datapoolSSD1TB/docker`, presentan un alto grado de volatilidad y constantes operaciones transaccionales. Para garantizar la integridad de esta información frente a errores operativos, actualizaciones fallidas o corrupciones lógicas, se requiere un mecanismo de protección ágil.

Para este cometido se ha implementado **Sanoid** [24], una herramienta de código abierto desarrollada en el lenguaje Perl, diseñada específicamente para automatizar la creación y depuración de instantáneas en sistemas de archivos ZFS basándose en políticas de retención declarativas. Sanoid se integra directamente con el motor nativo de ZFS, instruyendo al sistema para que congele el estado de los bloques de datos a nivel de núcleo del sistema operativo de manera casi instantánea y sin penalizar el rendimiento de entrada y salida de las bases de datos de Docker.

La configuración aplicada en el archivo principal de Sanoid refleja una política agresiva de recuperación ante errores lógicos para todas las aplicaciones en contenedor:

- **Configuración Aplicada:** Se estableció un esquema de retención totalmente automatizado, delegando en la herramienta tanto la creación programada de nuevas instantáneas como la depuración y eliminación de las obsoletas. Esta configuración protege de forma recursiva todo el árbol de directorios de Docker, conservando cuatro instantáneas diarias, tres semanales y dos mensuales.

- **Justificación Arquitectónica:** Es necesario reiterar que estas instantáneas de Sanoid no constituyen una copia de seguridad debido a que se almacenan en la propia unidad. Si el disco de estado sólido donde reside el conjunto de datos sufre una falla de hardware, todas las instantáneas se perderán simultáneamente con los datos de origen.

- **Casos de Uso Previstos:** La utilidad real de Sanoid radica en la resiliencia operativa. Si una actualización de la plataforma Nextcloud corrompe la base de datos o si el administrador elimina por error un panel de visualización crítico en Grafana, este puede utilizar el comando nativo de reversión de ZFS para retroceder el contenedor exacto a su estado de hace 24 horas, restaurando el servicio muy rápidamente sin necesidad de extraer o transferir gigabytes de información desde el disco mecánico de copias de seguridad.

9.5 Replicación y Persistencia con Rsync

Para los activos que no requieren una regresión en el tiempo de los mismos, se ha implementado una estrategia de replicación basada en **rsync** [30].

A diferencia de las herramientas de empaquetado, `rsync` realiza una sincronización a nivel de archivo bit a bit, utilizando algoritmos de transferencia para enviar únicamente los bloques modificados. En esta infraestructura, se utiliza para cuatro propósitos críticos: el guardado de la configuración del hipervisor, el espejo de datos del disco de 3TB al de 12TB, la extracción consistente de datos desde instantáneas ZFS y el espejado del disco de 12TB hacia otro ordenador.

9.5.1. Respaldo de Configuración y Entorno del Hipervisor

La configuración de la infraestructura, Proxmox VE, requiere que su configuración lógica sea respaldada más allá de las máquinas y contenedores Linux. Para ello, se ha automatizado la replicación de directorios críticos del sistema raíz hacia la unidad de almacenamiento de 12TB:

- **Configuración de Clúster (`/etc/pve`):** Es el directorio más crítico, ya que contiene la configuración de Proxmox VE, incluyendo los archivos `.conf` de cada LXC/VM, los permisos de usuario, la definición de los almacenamientos, las reglas del cortafuegos, entre otros.
- **Entorno de Administración (`/root`):** Se replica el directorio del superusuario para preservar los scripts de automatización personalizados y archivos de administración, incluyendo el fichero de historial de comandos (`.bash_history`) con el fin de mantener un seguimiento del mismo.
- **Servicios de Acceso y Políticas (`/etc/ssh/ssh_config` y `/etc/sanoid`):** Se aseguran las configuraciones de acceso remoto via SSH y las definiciones de las políticas de retención de instantáneas.
- **Optimización de Hardware (`/etc/hdparm.conf`):** Se respalda la configuración de rendimiento y gestión de energía de los discos mecánicos para garantizar que, tras una restauración, el hardware opere bajo los mismos parámetros de eficiencia.
- **Gestión de Energía y UPS (`/etc/nut`):** Contiene la configuración integral del Sistema de Alimentación Ininterrumpida. Su respaldo es vital para preservar los parámetros de comunicación y las políticas de apagado automático, asegurando que el hipervisor y los servicios se detengan de forma segura y controlada al detectar que el equipo ha pasado a operar en modo batería tras un corte de suministro eléctrico.
- **Automatización de Tareas (`/var/spool/cron/crontabs/root`):** Almacena la configuración del planificador *cron* del superusuario. Su respaldo es indispensable para preservar la programación y ejecución desatendida de los *ficheros ejecutables* de replicación personalizados (como los espejos de `rsync`). Restaurar este archivo garantiza que las políticas de retención y sincronización vuelvan a operar de forma autónoma inmediatamente después de reconstruir el sistema, sin necesidad de reconfigurar los intervalos de ejecución manualmente.
- **Puntos de Montaje (`/etc/fstab`):** Define la estructura de montaje persistente de los sistemas de archivos durante el proceso de arranque del hipervisor. Su preservación es crítica para asegurar que las unidades de almacenamiento secundario conserven

sus rutas exactas en `/mnt` y sus parámetros de montaje personalizados. Restaurar este archivo previene que los *ficheros ejecutables* de automatización fallen por no encontrar los destinos esperados tras una reconstrucción del sistema.

9.5.2. Replicación de Grandes Volúmenes (HDD 3TB a 12TB)

Para los datos operativos y documentos personales alojados en la unidad de 3TB, se ejecuta un fichero de replicación (`rsync-mirror-3tbTo12tb.sh`) que genera un espejo exacto (1:1) en la unidad de 12TB.

El fichero utiliza el parámetro `-delete` para garantizar que el destino sea un reflejo fiel del origen, eliminando archivos que ya no existan en la unidad de 3TB. Para asegurar la integridad de la ejecución, el proceso cuenta con un sistema de bloqueo que evita ejecuciones solapadas y verificaciones de punto de montaje que impiden que el *script* escriba accidentalmente en el disco de sistema si las unidades mecánicas no están disponibles.

9.5.3. Sincronización Consistente de conjunto de datos ZFS a Ext4

Una de las implementaciones más avanzadas de la estrategia es la secuencia de comandos
`rsync-Datasets-1tbTo12tb.sh`. Este proceso resuelve el desafío de copiar los volúmenes de Docker con bases de datos y archivos en constante cambio sin detener los servicios.

- **Lectura desde Instantáneas:** En lugar de copiar los datos directamente desde el sistema de archivos activo (donde los archivos podrían cambiar durante la copia), la secuencia de comandos identifica la instantánea más reciente generada por Sanoid.

- **Preservación de Metadatos:** Se utilizan las banderas `-aHAX` para asegurar que se mantengan los permisos, enlaces duros, atributos extendidos y contextos de seguridad, garantizando que el espejo en la unidad de 12TB sea funcionalmente idéntico al original del SSD.

9.5.4. Replicación de Nivel 2 y Preservación (12TB a nodo secundario)

Para garantizar la supervivencia de la información ante un evento que comprometa el servidor principal, se sincroniza la unidad de 12TB una vez por semana hacia un segundo ordenador físico ubicado en la misma red local. Recordemos que esta unidad incluye el archivo histórico, las copias de seguridad de Proxmox y las réplicas de los conjuntos de datos ZFS.

Arquitectura de la Transferencia (NFS y Rsync)

El equipo de destino opera bajo una distribución GNU/Linux y expone un recurso compartido a través del protocolo NFS. El hipervisor Proxmox monta este recurso de red localmente y utiliza la herramienta `rsync` para escanear y transferir únicamente los bloques de datos que han sufrido modificaciones desde la última ejecución semanal, optimizando enormemente el tiempo de sincronización frente a una copia tradicional.

Lógica de Ejecución y Eficiencia Energética

Dado que el ordenador de respaldo no requiere estar operativo de forma ininterrumpida, el proceso de replicación cuenta con un script de control condicional diseñado para maximizar el ahorro energético y aislar los datos de la red cuando no están en uso:

1. **Validación Inicial:** El sistema comprueba mediante un ping de red si el equipo de destino se encuentra en línea y responde. Si la conexión es exitosa, se procede a iniciar la replicación.
2. **Encendido Remoto:** En caso de no obtener respuesta, el hipervisor envía un 'Paquete Mágico' utilizando la tecnología de encendido remoto por red (*Wake-on-LAN*) para despertar físicamente la placa base del equipo destino, y por ende encenderlo.
3. **Ventana de Inicialización:** El script entra en espera 3 minutos, permitiendo que el sistema operativo secundario inicie por completo y levante sus servicios de red.
4. **Reintento y Ejecución:** Tras la pausa, se realiza una segunda validación. Una vez que el equipo responde, el recurso NFS se monta correctamente y `rsync` comienza la transferencia.

El Rol Crítico de la Red a 2.5 Gbps

En este escenario de respaldo masivo, la infraestructura de red a 2.5 Gbps deja de ser una simple mejora para convertirse en un pilar fundamental de la viabilidad del proyecto. La sincronización semanal sobre un volumen de 12 terabytes implica la evaluación y potencial transferencia de cientos de gigabytes en una sola sesión.

Mientras que un enlace tradicional de 1 Gbps (125 MB/s teóricos) generaría un cuello de botella que limitaría el rendimiento, la red de 2.5 Gbps (capaz de alcanzar hasta 312 MB/s) elimina esta restricción. En la infraestructura actual, los discos mecánicos instalados poseen un límite físico de lectura y escritura secuencial cercano a los 190 MB/s debido a su generación. Al utilizar el enlace de alta velocidad, se garantiza que las unidades operen a su capacidad máxima, reduciendo significativamente los tiempos de sincronización en comparación con una red Gigabit.

9.6 Cuadro Resumen de la Estrategia de Continuidad

La siguiente tabla sintetiza la arquitectura de protección implementada para cada activo de la infraestructura:

Activo	Herramienta	Frecuencia	Destino
VMs / LXC	Proxmox VZDump	Diaria/Semanal/Mensual	HDD 12TB
Volúmenes Docker	Sanoid	Diaria/Semanal/Mensual	SSD
Volúmenes Docker	Rsync	Diario	HDD 12TB
Configuración Host	Rsync	Diaria	HDD 12TB
Toda la unidad (3TB)	Rsync	Diaria	HDD 12TB
Toda la unidad (12TB)	Rsync	Semanal	Nodo secundario

Tabla 9.1: Matriz de copias de seguridad y redundancia.

CAPÍTULO 10
Estudio Económico y Retorno de Inversión

La arquitectura de la infraestructura de servicios no es independiente de su arquitectura financiera. En este sentido, la evaluación comparativa entre la inversión en equipos propios, la adquisición de soluciones comerciales preconfiguradas y la externalización mediante servicios *cloud* constituye un pilar crítico en la fase de planificación. En este capítulo se detalla el análisis económico del proyecto, evaluando los gastos de capital (*Capital Expenditure, CAPEX*) y los gastos operativos (*Operational Expenditure, OPEX*).

Para aportar el mayor rigor posible al análisis, se contrastan cuatro escenarios de despliegue tecnológico: la construcción de un servidor físico a medida (*On-premise*), la adquisición de un NAS comercial especializado de la empresa Synology, el alquiler de infraestructura en la nube (IaaS) y el uso de un servicio de almacenamiento gestionado (SaaS) representado por el plan de 10 Terabytes de Google Drive.

10.1 Análisis de Gastos de Capital e Implementación (CAPEX)

En términos financieros, los Gastos de Capital (*Capital Expenditure, CAPEX*) no solo contemplan la adquisición de los activos físicos, sino también todos los costes técnicos y humanos necesarios para la puesta en marcha de la solución (horas de implementación inicial).

En esta sección se desglosa el CAPEX de las cuatro alternativas de despliegue evaluadas: la implementación en un servidor físico local, la adquisición de un producto NAS synology preconfigurado, el uso de infraestructura en la nube (IaaS) y la contratación de un servicio gestionado (SaaS).

10.1.1. Escenario 1: Servidor Físico Local (On-premise)

Este modelo implica la propiedad total del hardware y la instalación manual del sistema desde sus cimientos. Por tanto, el CAPEX se compone de la adquisición de componentes y del tiempo de ingeniería.

Para la adquisición del hardware, la construcción del servidor se ha fundamentado en una política de reaprovechamiento de activos, lo que reduce drásticamente la inversión. Se definen dos subescenarios para valorar este coste físico:

- **Escenario A (Inversión Real):** Contempla únicamente el desembolso económico de los componentes adquiridos específicamente para este proyecto.

- **Escenario B (Valoración de Mercado):** Contempla el coste total de la máquina si se hubiese tenido que adquirir desde cero, valorando las piezas reutilizadas según su precio medio actual en el mercado de segunda mano.

En la Tabla 10.1 se detalla el inventario completo, clasificando los elementos según su procedencia y valor económico.

Componente	**Estado**	**Coste / Valor estimado**
Componentes Adquiridos (Inversión Real)		
HDD 12TB Western Digital Red Plus NAS	Nuevo	204,00 €
Tarjeta de red TP-Link TX401 10 GbE	Nuevo	70,00 €
Conmutador (*Switch*) TP-Link TL-SG105-M2	Nuevo	45,00 €
Chasis Thermaltake Chaser MK-1	Usado	60,00 €
SAI (*UPS*) Lyonn CTB 800VA	Nuevo	60,00 €
Subtotal Inversión Real (Escenario A)		**439,00 €**
Componentes Reutilizados (Valoración Segunda Mano)		
CPU Intel Core i5-9400F	Usado	30,00 €
SSD 1TB Kingston A400	Usado	80,00 €
SSD 1TB Western Digital Blue	Usado	80,00 €
Placa Base Gigabyte H310M H 2.0	Usado	50,00 €
Memoria RAM Corsair Vengeance DDR4 8GB	Usado	40,00 €
Memoria RAM Corsair Vengeance DDR4 32GB	Usado	130,00 €
Fuente de Alimentación EVGA 1000W Supernova	Usado	110,00 €
Disipador Cooler Master 212 EVO	Usado	15,00 €
HDD 3TB Western Digital Red	Usado	115,00 €
Subtotal Valor Reutilizado		**650,00 €**
Valor Total de la Infraestructura (Escenario B)		**1.089,00 €**

Tabla 10.1: Desglose de costes de hardware (nuevos y valoración de usados).

Al coste del hardware debe sumarse el **tiempo de implementación (Mano de obra)**: Se estima una dedicación de 80 horas técnicas para el montaje físico de los componentes, configuración de red e instalación del hipervisor y servicios. A un precio de mercado de 40 €/h:

- Coste de personal: 80 h $\times$ 40 €/h = **3.200,00 €**.

En conclusión, el CAPEX total para el Escenario 1 varía según la política de hardware:

- **CAPEX Local (Escenario A - Inversión Real):** 439,00 € + 3.200,00 € = **3.639,00 €**.
- **CAPEX Local (Escenario B - Adquisición Total):** 1.089,00 € + 3.200,00 € = **4.289,00€**.

10.1.2. Escenario 2: Infraestructura como Servicio (IaaS)

En este modelo, al alquilar los recursos de computación a un proveedor de nube (ej. AWS, Google Cloud), no existe un desembolso inicial por la compra de bienes físicos (Hardware = 0 €).

Sin embargo, la configuración de almacenamiento y la instalación del software requieren intervención técnica. Al no requerir montaje físico ni instalación del hipervisor, se estima un ahorro de 6 horas respecto al servidor local, resultando en 74 horas de trabajo técnico.

- **Implementación (Personal):** 74 h $\times$ 40 €/h = **2.960,00 €**.

Por lo tanto, el **CAPEX total para el Escenario IaaS es de 2.960,00 €**, correspondiente íntegramente a costes de mano de obra en puesta en marcha.

10.1.3. Escenario 3: Software como Servicio (SaaS)

El modelo SaaS (ej. Google Drive) es una solución preempaquetada lista para su uso. El proveedor gestiona la infraestructura subyacente y la plataforma de software, por lo que no requiere adquisición de equipos físicos ni horas de ingeniería complejas para su despliegue inicial. El tiempo de creación de cuenta y estructuración básica de datos se considera despreciable desde el punto de vista financiero.

- **Coste de Hardware:** 0 €.
- **Implementación (Personal):** 0 €.

En consecuencia, el **CAPEX total para el Escenario SaaS es de 0 €**, convirtiéndose en la opción sin barrera de entrada financiera.

10.1.4. Escenario 4: Solución Comercial Empaquetada (NAS Synology)

Como alternativa de hardware local, se presupuesta la adquisición de una solución preconfigurada (NAS) que cumpla con los mismos requisitos de almacenamiento y conectividad de red que el servidor armado. Se ha seleccionado la solución del fabricante Synology que satisface estas demandas técnicas.

En la Tabla 10.2 se detalla el inventario de componentes comerciales necesarios para igualar las capacidades operativas del servidor local, junto con sus precios actuales de mercado utilizando componentes oficiales de la marca para asegurar total compatibilidad y soporte.

Componente Comercial	**Coste**
Synology 5-Bahías DiskStation DS1525+	680,76 €
Expansión RAM Synology D4ES03-16G DDR4 ECC SODIMM	1.341,89 €
Tarjeta de red Synology E10G22-T1-Mini 10 GbE	131,89 €
HDD 12TB Synology HAT3310-12T Plus	531,19 €
HDD 4TB Synology HAT3300-4T Plus	239,58 €
SSD 800GB Synology SNV5420-800G NVMe M.2	961,95 €
Subtotal Inversión Hardware Synology	**3.887,26 €**

Tabla 10.2: Desglose de costes de hardware para el NAS comercial (Synology).

A diferencia del servidor montado desde cero, en este modelo el sistema operativo propietario (*DiskStation Manager*) viene preinstalado y su ecosistema está estandarizado,

por lo que el tiempo de configuración de volúmenes, red y servicios se reduce de forma considerable. Se estima una dedicación de 40 horas técnicas para su puesta en marcha integral.

- **Implementación (Personal):** 40 h $\times$ 40 €/h = **1.600,00 €**.

De este modo, sumando el valor de adquisición de los componentes y la mano de obra de configuración, el **CAPEX total para el Escenario 4 (NAS Synology) asciende a 5.487,26 €**.

10.2 Análisis de Gastos Operativos (OPEX)

El Gasto Operativo (*Operational Expenditure, OPEX*) agrupa todos los costes recurrentes y periódicos necesarios para mantener el sistema en funcionamiento tras su puesta en marcha inicial. A diferencia del CAPEX, que representa una inversión puntual, el OPEX es un flujo de caja continuo que determina la sostenibilidad financiera del proyecto a largo plazo.

A continuación, se detallan los gastos operativos anualizados para cada uno de los cuatro escenarios planteados. Cabe destacar que, superada la fase de despliegue, se asume que las tareas de administración y mantenimiento habituales demandan un esfuerzo equivalente en todas las plataformas, por lo que no suponen un factor diferencial en el coste.

10.2.1. Escenario 1: Servidor Físico Local (On-premise)

El gasto operativo anual de la infraestructura física propia se compone de dos elementos principales: el consumo eléctrico ininterrumpido y el licenciamiento de software.

1. **Consumo Eléctrico:** Tras realizar mediciones sobre el equipo, se ha determinado que el servidor operando con una carga de trabajo habitual tiene un consumo real de 52,3 vatios (W). Tomando como referencia el coste de la energía en la Ciudad Autónoma de Buenos Aires (CABA, Argentina), estimado en 0,085 €/kWh, el cálculo es el siguiente:
   - **Consumo diario:** 52,3 W $\times$ 24 h = 1.255,2 Wh $\approx$ 1,255 kWh/día
   - **Consumo anual:** 1,255 kWh/día $\times$ 365 días = 458,15 kWh/año
   - **Coste eléctrico anual:** 458,15 kWh/año $\times$ 0,085 €/kWh = **38,94 €/año**
2. **Licencias de Software:** Para la gestión y transmisión del contenido multimedia de forma eficiente, se requiere una suscripción a la plataforma Emby Premiere, cuyo coste es de 4,26 €/mes.
   - **Coste de software anual:** 4,26 €/mes $\times$ 12 meses = **51,12 €/año**

Sumando ambos conceptos, el **OPEX total del servidor local asciende a 90,06 € anuales**.

10.2.2. Escenario 2: Infraestructura como Servicio (IaaS)

En este modelo, los recursos de hardware son alquilados a un proveedor de nube. Debido a los altos requisitos de capacidad de procesamiento y almacenamiento del sistema, este escenario exige mantener en ejecución continua 4 máquinas virtuales junto con sus volúmenes de datos.

- **Recursos de Computación y Almacenamiento:** El coste de las instancias virtuales y el almacenamiento asciende a 380 €/mes.
  - **Coste de infraestructura anual:** 380 €/mes × 12 meses = **4.560,00 €/año**
- **Licencia Emby Premiere:** Al tratarse de un despliegue sobre infraestructura no gestionada a nivel de aplicación, se sigue requiriendo la suscripción al software multimedia.
  - **Coste de software anual: 51,12 €/año**

Sumando la infraestructura y el software, el **OPEX total IaaS asciende a 4.611,12 € anuales**.

10.2.3. Escenario 3: Software como Servicio (SaaS)

Este modelo (ej. Google Drive) empaqueta tanto la infraestructura como el software en una única cuota de servicio, eliminando la necesidad de pagar licencias de terceros o consumos eléctricos.

- **Suscripción al servicio:** Tomando como referencia el plan de almacenamiento de 10 TB, el coste es de 42,47 €/mes.
  - **Coste de suscripción anual:** 42,47 €/mes × 12 meses = **509,64 €/año**
- **Licencia de Software:** No es necesaria, la funcionalidad y visualización están incluidas de forma nativa en la plataforma. (**0 €**)

Por lo tanto, el **OPEX total SaaS asciende a 509,64 € anuales**.

10.2.4. Escenario 4: Solución Comercial (NAS Synology)

El gasto operativo anual de la solución NAS comercial es similar al del servidor físico local en cuanto a licencias, pero presenta una ventaja en eficiencia energética, ya que este tipo de hardware especializado está optimizado para ofrecer un menor consumo eléctrico, a coste de un menor rendimiento que el servidor armado.

1. **Consumo Eléctrico:** Según las especificaciones operativas, se estima un consumo promedio de 30 vatios (W) para esta unidad bajo carga de trabajo. Tomando como referencia el mismo coste de la energía (0,085 €/kWh):
   - **Consumo diario:** 30 W × 24 h = 720 Wh = 0,72 kWh/día
   - **Consumo anual:** 0,72 kWh/día × 365 días = 262,80 kWh/año
   - **Coste eléctrico anual:** 262,80 kWh/año × 0,085 €/kWh = **22,34 €/año**

2. **Licencias de Software:** Al igual que en la solución montada a medida, se requiere la suscripción a Emby Premiere para la decodificación y transmisión eficiente del contenido multimedia.

    - **Coste de software anual: 51,12 €/año**

Sumando ambos conceptos, el **OPEX total del NAS Synology asciende a 73,46 € anuales**.

10.2.5. Resumen de Gastos Operativos

En la Tabla 10.3 se consolida el gasto operativo anualizado para cada uno de los modelos evaluados. Esta comparativa evidencia cómo la nube pública (IaaS) penaliza financieramente las cargas de trabajo continuas con altos requerimientos de almacenamiento, mientras que las infraestructuras locales minimizan el gasto recurrente una vez superada la barrera inicial del CAPEX.

Escenario de Despliegue	OPEX Total (Anual)
1. Servidor Físico Local (On-premise)	90,06 €
2. Infraestructura como Servicio (IaaS)	4.611,12 €
3. Software como Servicio (SaaS)	509,64 €
4. Solución Comercial (NAS Synology)	73,46 €

Tabla 10.3: Comparativa de Gastos Operativos (OPEX) anuales según el modelo.

10.3 Coste Total de Propiedad (TCO)

El Coste Total de Propiedad (*Total Cost of Ownership, TCO*) es una métrica financiera diseñada para evaluar de manera integral tanto el coste de adquisición y puesta en marcha (CAPEX) como todos los gastos operativos (OPEX) asociados a una solución tecnológica durante su ciclo de vida.

Para este proyecto, se analizan cuatro escenarios comparativos: la implementación en un servidor físico local, la adquisición de una solución comercial (NAS Synology), el uso de infraestructura en la nube (IaaS) y la contratación de un servicio gestionado (SaaS).

La fórmula fundamental para calcular el TCO a lo largo de un horizonte temporal determinado (t, expresado en años) se define en la ecuación 10.1:

$$TCO = CAPEX + (OPEX \times t) \tag{10.1}$$

Para evaluar la viabilidad a medio y largo plazo de la infraestructura, se ha establecido un horizonte de proyección de 5 años, que se corresponde con la vida útil garantizada a corto plazo del hardware.

10.3.1. Consolidación de Costes y TCO a 5 años

Tomando los datos calculados en las secciones anteriores (donde el CAPEX incluye tanto la adquisición de componentes como las horas de ingeniería necesarias para la implementación), la Tabla 10.4 muestra la proyección financiera total para cada alternativa.

Para el Servidor Físico Local se desglosan los dos escenarios de inversión (A y B) planteados previamente.

Escenario de Despliegue	CAPEX Total	OPEX (Anual)	TCO (5 años)
1. Local (Escenario A - Inv. Real)	3.639,00 €	90,06 €	4.089,30 €
1. Local (Escenario B)	4.289,00 €	90,06 €	4.739,30 €
2. Solución Comercial (NAS Synology)	5.487,26 €	73,46 €	5.854,56 €
3. Infraestructura (IaaS)	2.960,00 €	4.611,12 €	26.015,60 €
4. Software como Servicio (SaaS)	0,00 €	509,64 €	2.548,20 €

Tabla 10.4: Comparativa del Coste Total de Propiedad (TCO) proyectado a 5 años.

10.3.2. Análisis del TCO y Viabilidad Técnica

A partir de los datos expuestos en la Tabla 10.4, se extraen las siguientes conclusiones fundamentales para la dirección del proyecto:

- **Descarte definitivo de la opción IaaS:** El modelo de Infraestructura como Servicio presenta un TCO desorbitado de más de 26.000 € a 5 años. Esto se debe a la necesidad de mantener 4 máquinas virtuales y almacenamiento en ejecución continua. Esta alternativa es financieramente inviable para este caso de uso y queda descartada.
- **El Factor Técnico y el descarte de la Nube (SaaS):** Un análisis estrictamente financiero sugiere que Google Drive (SaaS) es la opción ganadora al no tener un CAPEX inicial y presentar el TCO teórico más bajo a 5 años (2.548,20 €). Sin embargo, esta alternativa queda **descartada por inviabilidad técnica**. El ecosistema multimedia profesional exige trabajar de forma colaborativa entre múltiples dispositivos de la red local, accediendo a carpetas y proyectos que alcanzan los 500 GB por sesión. Intentar editar, sincronizar o transferir estos volúmenes a través de una conexión a Internet genera una ineficiencia operativa que hace que no sea posible realizar los procesos de edición.
- **Comparativa de las Soluciones Locales Viables:** Al descartar los modelos *Cloud* por limitaciones técnicas y financieras, la decisión arquitectónica real se reduce a elegir entre las dos opciones *On-premise* capaces de soportar la carga de trabajo en red local a alta velocidad: el Servidor Físico Armado a medida y el NAS Comercial Synology. El TCO a 5 años del NAS Synology (5.854,56 €) es notablemente superior al del Servidor Armado en su Escenario A de inversión real (4.089,30 €).

Esta consolidación técnica y financiera sienta las bases para el siguiente apartado. Al haber descartado la externalización a la nube, se evaluará el Retorno de Inversión (ROI) y la amortización comparando directamente las dos alternativas físicas viables.

10.4 Retorno de la Inversión (ROI)

Para determinar la viabilidad definitiva del proyecto y justificar la elección del despliegue del servidor físico a medida frente a la adquisición de una solución comercial empaquetada (Synology), es indispensable analizar la rentabilidad financiera de la decisión.

Como se concluyó en la sección anterior, las alternativas en la nube (SaaS e IaaS) quedan descartadas por limitaciones técnicas y financieras. Por tanto, el análisis tradicional de Retorno de la Inversión (ROI) basado en la recuperación del CAPEX a través del ahorro en suscripciones mensuales pierde su aplicabilidad. En este contexto, ambas soluciones viables son físicas y presentan un Gasto Operativo (OPEX) sumamente bajo y casi idéntico (90,06 € del servidor armado frente a 73,46 € anuales del Synology).

La decisión financiera recae entonces en la optimización del flujo de caja inicial. Se define a la solución comercial Synology como la alternativa predeterminada del mercado para resolver la necesidad técnica, frente a la cual se medirá el ahorro generado por la decisión de construir el servidor armado a medida.

Para mantener un rigor contable, el cálculo del Retorno de Inversión integra el CAPEX total definido en secciones anteriores. El ROI a 5 años se calcula según la ecuación 10.2:

$$ROI = \left(\frac{\text{Ahorro Neto Acumulado (5 años)}}{\text{CAPEX Total}} \right) \times 100 \tag{10.2}$$

10.4.1. Cálculo para el Escenario A (Inversión Real en Hardware)

En este escenario realista se ha aplicado una política de reaprovechamiento de componentes, resultando en un CAPEX total de **3.639,00 €** (439,00 € de hardware + 3.200,00 € de implementación).

- **Ahorro neto acumulado a 5 años:** 5.854,56 € (TCO Synology) − 4.089,30 € (TCO Escenario A) = **1.765,26 €**.
- **ROI a 5 años:** $(1{,}765{,}26/3{,}639{,}00) \times 100 =$ **48,51 %**.

Interpretación: El ROI del 48,51 % indica que la decisión de ensamblar el servidor a medida y reaprovechar componentes genera un rendimiento positivo sustancial. En 5 años, el ahorro conseguido equivale a casi la mitad de lo que costó poner en marcha el servidor completo (incluyendo el valor de las horas de trabajo).

10.4.2. Cálculo para el Escenario B (Adquisición Total de Hardware)

En este caso hipotético, se asume la compra desde cero de todos los componentes a valor de mercado actual, resultando en un CAPEX total de **4.289,00 €** (1.089,00 € de hardware + 3.200,00 € de implementación).

- **Ahorro neto acumulado a 5 años:** 5.854,56 € (TCO Synology) − 4.739,30 € (TCO Escenario B) = **1.115,26 €**.
- **ROI a 5 años:** $(1{,}115{,}26/4{,}289{,}00) \times 100 =$ **26,00 %**.

Interpretación: Incluso asumiendo la compra de la máquina completamente y sumando el esfuerzo de instalación técnica, la decisión de utilizar hardware estándar frente a un ecosistema propietario cerrado genera una rentabilidad del 26 % a los 5 años.

10.5 Conclusiones Económicas y Justificación del Modelo Local

El análisis financiero integral desarrollado a lo largo de este capítulo demuestra que el diseño de infraestructuras tecnológicas críticas no puede regirse exclusivamente por la búsqueda del Coste Total de Propiedad (TCO) más bajo. Si la evaluación se limitara al plano puramente contable, el modelo SaaS (Google Drive) parecería la opción más atractiva a corto y medio plazo debido a la ausencia de inversión inicial en hardware.

Sin embargo, la arquitectura de sistemas exige una alineación entre la viabilidad financiera y la viabilidad técnica. En este sentido, la decisión de construir e implementar un Servidor Físico Local (*On-premise*) ensamblado a medida se fundamenta en dos grandes pilares: primero, la necesidad técnica de descartar la nube pública, y segundo, la superioridad financiera y operativa del hardware estándar frente a las soluciones comerciales cerradas.

A continuación, se detallan los factores concluyentes que justifican esta elección:

- **Inviabilidad del Modelo Cloud y Necesidad Operativa Local:** El entorno de producción multimedia profesional presenta exigencias de rendimiento que la nube pública no puede satisfacer. Trabajar con proyectos audiovisuales implica el manejo concurrente de archivos individuales que pueden pesar 40 GB y sesiones completas que alcanzan fácilmente los 500 GB. Intentar transferir o editar este volumen de información a través de una conexión a Internet convencional seria agobiante e insostenible de realizar.

  La infraestructura local resuelve este problema de raíz operando sobre una red LAN de **2,5 Gbps**, lo que reduce las transferencias de horas a minutos y permite la edición directa sobre el servidor. A esto se suma la soberanía total sobre los datos, garantizando la privacidad frente a cambios en las políticas de terceros, y la continuidad operativa, ya que el trabajo interno no se detiene ante posibles caídas en el servicio del proveedor de Internet.

- **Eficiencia del Flujo de Caja y Evasión del Ecosistema Propietario:** Una vez establecida la necesidad ineludible de contar con hardware físico local, la disyuntiva recae entre adquirir un servidor comercial (Synology) o ensamblar un servidor a medida.

  Las soluciones comerciales como Synology imponen un ecosistema cerrado que obliga a adquirir componentes certificados por la propia marca (discos duros, memoria RAM, tarjetas de red) con sobreprecios considerables frente a sus equivalentes en el mercado abierto. Esto eleva la barrera de entrada a más de 5.400 €. Por el contrario, el servidor armado fomenta la economía circular mediante el reaprovechamiento de activos y el uso de componentes de hardware estándar, logrando superiores capacidades operativas, debido a mejor procesador.

- **Versatilidad de Procesamiento y Escalabilidad Abierta:** Al adquirir un equipo comercial, la inversión queda atada a las limitaciones impuestas por el fabricante tanto en hardware (placas base con procesadores de bajo consumo soldados) como en software (sistemas operativos propietarios). Si en el futuro las necesidades del proyecto cambian, la única vía de actualización suele ser la compra de un equipo de gama superior.

  El servidor local ensamblado a medida representa un activo tecnológico de propósito general. Al contar con una arquitectura x86 estándar, capacidad de procesa-

miento superior y una gestión de recursos abierta, se convierte en una plataforma de despliegue de mayor flexibilidad.

La construcción de la infraestructura basada en componentes de propósito general logra el equilibrio óptimo para el proyecto. Transforma flujos de trabajo lentos en operaciones en tiempo real, asegura el control absoluto de la información y evita los elevados costes iniciales de las marcas propietarias. En entornos de alta demanda, la inversión de tiempo técnico en un servidor a medida se traduce en un ahorro económico y en la obtención de un sistema robusto, flexible y completamente adaptado a las necesidades operativas actuales y futuras.

CAPÍTULO 11

Divulgación Tecnológica y Validación Comunitaria

La culminación de este proyecto no radica únicamente en el despliegue exitoso de los servicios y su viabilidad económica inmediata, sino en su capacidad para resolver problemas concretos fuera del ámbito del desarrollo inicial. La validación por pares garantiza que la arquitectura diseñada no solo resuelve un problema individual, sino que posee la escalabilidad y robustez necesarias para ser adoptada como un modelo de referencia. En este capítulo se documenta la presentación pública de la infraestructura en un congreso tecnológico internacional, analizando su impacto y la recepción por parte de la comunidad de especialistas.

11.1 Contexto de la Validación

La exposición del proyecto tuvo lugar en el marco de la conferencia internacional Nerdearla España 2025, un evento de divulgación tecnológica de acceso libre y gratuito de gran prestigio en la comunidad hispanohablante. La presentación se llevó a cabo de manera presencial el viernes 14 de noviembre de 2025 en la instalación de La Nave, ubicada en la ciudad de Madrid.

El formato de participación consistió en una ponencia principal de veinticinco minutos de duración, seguida de un espacio de cinco minutos dedicado a preguntas del público. Para maximizar el alcance de la divulgación, la sesión fue transmitida en vivo a través de la plataforma virtual del evento y posteriormente archivada en su repositorio público de vídeo y YouTube para su consulta asíncrona por la comunidad global.

11.2 Exposición del Caso de Estudio

La ponencia fue titulada de manera oficial como "Mi nube, mis reglas, libre de suscripciones: Proxmox, Docker, VPN y Nextcloud". El hilo conductor de la presentación fue la demostración teórica y práctica de cómo es posible construir una infraestructura de almacenamiento integral, capaz de sustituir por completo a los proveedores de nube pública comerciales, garantizando la privacidad absoluta de los datos mediante herramientas de código abierto.

La agenda de la charla se estructuró en cinco bloques fundamentales que replicaron la arquitectura del presente Trabajo Final de Grado:

1. **Plataforma de Aplicación (Nextcloud):** Se expuso como la capa visible para el usuario, destacando su capacidad para reemplazar servicios de almacenamiento, permitiendo el acceso seguro desde cualquier dispositivo.

2. **Aislamiento Lógico (Docker):** Se justificó el uso de contenedores para ejecutar la plataforma de manera modular, garantizando la portabilidad de los servicios.

3. **Capa de Virtualización (Proxmox VE):** Se presentó el hipervisor de tipo uno, detallando las ventajas operativas de combinar máquinas virtuales completas con contenedores Linux ligeros para consolidar múltiples servidores, servicios y estaciones de trabajo en un único equipo físico.

4. **Acceso Seguro (Tailscale):** Se explicó la implementación de una red privada virtual moderna basada en conexiones directas entre nodos, eliminando la necesidad de abrir puertos perimetrales en el enrutador y reduciendo drásticamente la superficie de ataque cibernético.

5. **Integridad de Datos (ZFS):** El bloque final abordó el sistema de archivos avanzado, explicando mecanismos críticos como las sumas de verificación, la autoreparación frente a la corrupción silenciosa de datos y la eficiencia de las instantáneas inmutables para la recuperación ante desastres.

11.3 Análisis de Resultados y Retroalimentación

Durante la ponencia, se desglosó el esquema de ingeniería ante la audiencia. En lugar de limitarse a mostrar interfaces gráficas, se profundizó en las decisiones de diseño arquitectónico. Se explicó de manera detallada cómo la infraestructura delega la gestión de recursos físicos al hipervisor, mientras que las aplicaciones críticas se confinan en entornos sin privilegios.

El impacto de la ponencia superó las expectativas iniciales. La sala asignada contó con gran asistencia, evidenciando un interés generalizado en las soluciones de autoalojamiento y privacidad digital. Durante el bloque de preguntas y respuestas oficiales, la audiencia planteó interrogantes de alto nivel técnico relacionados con la gestión de memoria del sistema de archivos y las políticas de retención de copias de seguridad.

La validación definitiva del proyecto se produjo una vez finalizada la charla. En los espacios de intercambio de contactos posteriores a la presentación, una gran cantidad de asistentes se acercó para debatir sobre la infraestructura. Muchos de estos profesionales confesaron poseer laboratorios domésticos o servidores, pero manifestaron que sus implementaciones carecían de la robustez, la seguridad perimetral y la gestión avanzada de almacenamiento que se demostró en la ponencia.

El interés explícito de la comunidad por estudiar la topología presentada, con el objetivo de elevar sus propios proyectos hacia un estándar de grado corporativo, confirma de manera rotunda que este Trabajo Final de Grado trasciende el ámbito académico. Se ha validado empíricamente que la solución desarrollada resuelve un problema contemporáneo real, estableciendo un modelo arquitectónico de alta calidad, replicable y altamente valorado por la industria tecnológica (figura 11.1).

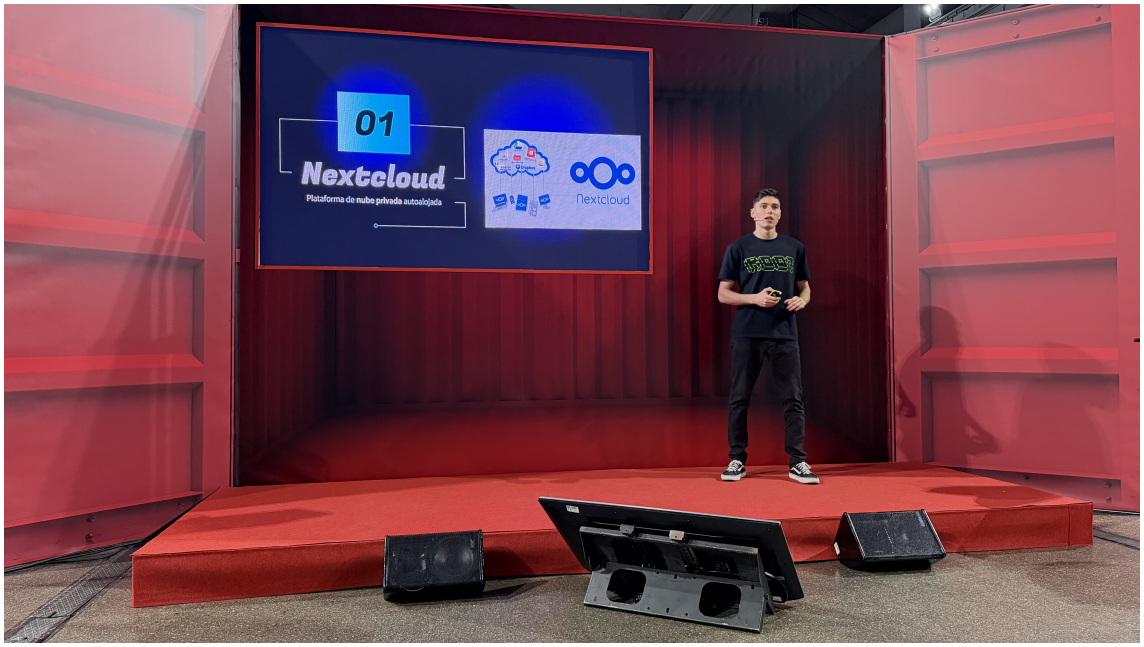

Figura 11.1: Ponencia de Nicolas Alejandro Margossian en el congreso internacional Nerdearla.

CAPÍTULO 12
Conclusiones y Líneas Futuras

Tras el diseño, implementación y validación de la infraestructura del servidor autoalojado, este capítulo presenta el cierre formal del Trabajo Final de Grado. Se expone el grado de cumplimiento de los objetivos iniciales, se establece la correlación directa entre el proyecto y los estudios universitarios cursados, se reflexiona sobre las competencias profesionales adquiridas y se proponen vías de expansión para la arquitectura desarrollada.

12.1 Conclusiones del Proyecto

El presente Trabajo Final de Grado ha finalizado con el cumplimiento de todos los objetivos planteados en su concepción. Se ha demostrado de manera teórica y práctica la viabilidad técnica de desplegar una infraestructura de servicios autoalojados integral, segura e independiente de los grandes proveedores comerciales corporativos, utilizando componentes físicos reacondicionados y priorizando el uso de herramientas de código abierto, complementadas de forma puntual con soluciones propietarias de bajo coste para cubrir necesidades avanzadas de transmisión multimedia.

La arquitectura resultante ha logrado sustituir con éxito a los proveedores de nube pública, garantizando la privacidad absoluta de los datos. La implementación del hipervisor Proxmox VE ha permitido una consolidación eficiente de los recursos, mientras que el diseño de almacenamiento basado en el sistema de archivos ZFS ha proporcionado un nivel de integridad de datos propio de entornos corporativos. Asimismo, la segmentación estricta de la red mediante un cortafuegos virtualizado y la integración de sistemas de prevención de intrusiones han establecido un perímetro de seguridad robusto.

12.2 Relación del Trabajo Desarrollado con los Estudios Cursados

Este proyecto constituye la aplicación práctica e integrada de los conocimientos adquiridos a lo largo de la titulación. La construcción de esta infraestructura ha requerido de múltiples disciplinas de la ingeniería informática, destacando la vinculación directa con las siguientes asignaturas:

- **Redes Corporativas:** La asignatura reforzó de manera práctica los conceptos avanzados de la pila de protocolos TCP/IP. Estos conocimientos fueron fundamentales para el diseño de redes virtuales dentro del hipervisor, el enrutamiento interno de

paquetes y el despliegue de redes privadas virtuales de última generación para el acceso remoto seguro.

- **Sistemas y Servicios en Red:** Resultó de gran importancia para comprender la fluidez de las comunicaciones cliente y servidor. Permitió aplicar la teoría del protocolo HTTP en la configuración de los proxys inversos, analizar el establecimiento de conexiones seguras mediante el saludo a tres vías del protocolo TCP y optimizar las tasas de transferencia en megabits por segundo para garantizar una transmisión ininterrumpida de contenido multimedia.
- **Seguridad en Redes y Sistemas Informáticos:** Todo el diseño perimetral se fundamenta en el temario de esta disciplina. Se aplicaron directamente los conceptos de segmentación de subredes, despliegue de zonas desmilitarizadas, configuración estricta de reglas de cortafuegos y la puesta en marcha de sistemas avanzados de detección y prevención de intrusiones.
- **Administración de Sistemas:** La gestión del entorno operativo exigió un dominio profundo de los sistemas basados en GNU/Linux. Se aplicaron de manera intensiva las políticas de seguridad a nivel de sistema de archivos, configurando esquemas de control de acceso discrecional y estableciendo reglas estrictas de herencia de permisos jerárquicos para usuarios y grupos.
- **Diseño, Configuración y Evaluación de los Sistemas Informáticos:** Esta asignatura proporcionó la base conceptual necesaria para estructurar la capa de observabilidad. Aunque las herramientas específicas varían con el tiempo, los fundamentos teóricos permitieron diseñar un ecosistema coherente para la monitorización de métricas, el análisis del consumo de recursos físicos, la estructuración del almacenamiento por niveles y la aplicación de políticas de crecimiento y rotación automatizada de los registros de actividad del sistema.

12.3 Lecciones Aprendidas y Competencias Adquiridas

A nivel técnico, la integración de sistemas heterogéneos ha supuesto el mayor reto y, en consecuencia, el mayor aprendizaje del proyecto. Lograr que el sistema de archivos ZFS, los contenedores sin privilegios, las redes virtuales y el cortafuegos perimetral operen de manera armónica ha requerido superar la teoría académica para adentrarse en la resolución de conflictos técnicos prácticos. Se ha adquirido una profunda competencia en el diagnóstico de errores, la lectura analítica de registros del sistema y la comprensión de la documentación técnica oficial.

A nivel de gestión, el proyecto ha demandado una estricta planificación temporal y procedimental. La necesidad de orquestar el despliegue siguiendo un orden lógico, desde el ensamble de los componentes físicos hasta la configuración de la última regla de alerta, ha fortalecido las habilidades de gestión de proyectos, la toma de decisiones basada en el análisis de riesgos y la capacidad para documentar de manera profesional una arquitectura compleja.

12.4 Trabajos Futuros y Líneas de Investigación

Aunque la infraestructura actual opera con un muy buen rendimiento y seguridad, el diseño modular permite proponer diversas líneas de mejora para futuras iteraciones del proyecto:

- **Actualización Integral de la Capa de Red:** Si bien el servidor principal cuenta con un enlace de 10 GbE, la topología perimetral opera actualmente a 2.5 GbE. Un trabajo futuro inmediato consistiría en la sustitución de los conmutadores y puntos de acceso para unificar toda la red troncal bajo el estándar de 10 GbE, eliminando cualquier posible cuello de botella en la transferencia de archivos.
- **Implementación de una Interfaz de Gestión Visual mediante Arcane:** Con el objetivo de simplificar la administración y el despliegue de los servicios virtualizados, se proyecta la integración de Arcane como panel de control centralizado para la orquestación de contenedores Docker. Esta herramienta permitirá la supervisión en tiempo real del estado de los servicios, la gestión del ciclo de vida de los contenedores y el monitoreo del consumo de recursos mediante una interfaz gráfica intuitiva. La adopción de esta solución no solo reducirá la dependencia de la interfaz de línea de comandos para tareas rutinarias, sino que facilitará el diagnóstico de incidencias y agilizará la escalabilidad de las aplicaciones.
- **Modernización y Redundancia del Almacenamiento Mecánico:** Actualmente, el almacenamiento reside en discos independientes. Como línea de evolución tecnológica, se propone la adquisición de múltiples unidades magnéticas de nueva generación, diseñadas específicamente para servidores, para agruparlas en matrices redundantes bajo el ecosistema ZFS. Esta modificación no solo aportaría tolerancia a fallos físicos en la capa de almacenamiento, sino que aumentaría las velocidades de lectura y escritura mediante el paralelismo de los discos.
- **Centralización de Logs con Grafana Loki:** Como mejora en la observabilidad del sistema, se plantea la implementación de Grafana Loki para la agregación y consulta de registros. Esta integración permitirá visualizar los logs de todas las aplicaciones y contenedores de forma centralizada en los paneles de Grafana, facilitando la correlación de eventos, la detección temprana de anomalías y la agilización de las tareas de depuración.
- **Automatización del Despliegue y Potencial de Comercialización:** Como evolución natural hacia la escalabilidad corporativa, se plantea la adopción de metodologías de Infraestructura como Código mediante la implementación de herramientas de automatización de la industria, como Ansible. El objetivo de esta línea de trabajo es codificar todos los parámetros y ajustes realizados en este proyecto para empaquetar la arquitectura completa. Esta estandarización técnica abre una vía directa hacia la viabilidad comercial, permitiendo ofrecer el sistema como una solución integral y preconfigurada para pequeñas y medianas empresas que requieran privaciad y velocidad de acceso a los datos. En consecuencia, el proyecto trasciende el ámbito de laboratorio/personal para adquirir una fuerte aplicabilidad en el mercado.

Bibliografía

[1] Centro Criptológico Nacional (CCN-CERT). *CCN-STIC-1453: Procedimiento de Empleo Seguro Cortafuegos OPNsense*. Consultado en https://www.ccn-cert.cni.es/es/pdf/guias/series-ccn-stic/1000-procedimientos-de-empleo-seguro/7083-ccn-stic-1453-procedimiento-de-empleo-seguro-cortafuegos-opnsense/file.html.

[2] Mike Chapple, David Seidl. *Guía de estudio CompTIA Security+: Examen SY0-701, Novena edición*. Sybex, 2023.

[3] Docker, Inc. *Documentación de Docker*. Consultado en https://docs.docker.com/.

[4] Emby LLC. *Documentación de soporte de Emby*. Consultado en https://emby.media/support/articles/Home.html.

[5] Grafana Labs. *Documentación de Grafana*. Consultado en https://grafana.com/docs/grafana/latest/.

[6] Nextcloud GmbH. *Documentación de Nextcloud*. Consultado en https://docs.nextcloud.com/.

[7] Nextcloud GmbH. *Nextcloud Todo en Uno (All-in-One)*. Repositorio de GitHub. Consultado en https://github.com/nextcloud/all-in-one.

[8] Nginx Proxy Manager. *Guía de Nginx Proxy Manager*. Consultado en https://nginxproxymanager.com/guide/.

[9] Proyecto OpenZFS. *Documentación de OpenZFS*. Consultado en https://openzfs.github.io/openzfs-docs/index.html.

[10] OPNsense / Deciso B.V. *Documentación de OPNsense*. Consultado en https://docs.opnsense.org/index.html.

[11] Prometheus. *Documentación de Prometheus*. Consultado en https://prometheus.io/docs/prometheus/latest.

[12] Prometheus. *Documentación de Alertmanager*. Consultado en https://prometheus.io/docs/alerting/latest/alertmanager/.

[13] Prometheus. *Código fuente de Alertmanager*. Repositorio de GitHub. Consultado en https://github.com/prometheus/alertmanager.

[14] Proxmox Server Solutions GmbH. *Documentación de Proxmox VE*. Consultado en https://pve.proxmox.com/pve-docs/.

[15] Equipo de Samba (Samba Team). *Documentación de Samba*. Consultado en https://www.samba.org/samba/docs/.

[16] Tailscale Inc. *Documentación de Tailscale*. Consultado en https://tailscale.com/docs.

[17] Profesional Review. *Qué es la presión positiva y negativa de aire en el PC*. Consultado en https://www.profesionalreview.com/2017/10/29/que-es-presion-positiva-y-negativa-de-aire-pc/.

[18] Google LLC. *Planes y precios de Google One*. Consultado en https://one.google.com/about/plans?hl=es&g1_landing_page=0.

[19] Proyecto Btrfs. *Documentación de Btrfs*. Consultado en https://btrfs.readthedocs.io/en/latest/.

[20] Cloudflare, Inc. *Gestión de direcciones IP dinámicas (DDNS)*. Consultado en https://developers.cloudflare.com/dns/manage-dns-records/how-to/managing-dynamic-ip-addresses/.

[21] CrowdSec. *Documentación de CrowdSec*. Consultado en https://docs.crowdsec.net/.

[22] Jason A. Donenfeld / Proyecto WireGuard. *Guía de inicio rápido de WireGuard*. Consultado en https://www.wireguard.com/quickstart/.

[23] OpenVPN Inc. *Documentación comunitaria de OpenVPN*. Consultado en https://openvpn.net/community-docs/?lang=en.

[24] Jim Salter. *Sanoid*. Repositorio de GitHub. Consultado en https://github.com/jimsalterjrs/sanoid.

[25] Open Information Security Foundation (OISF). *Documentación de Suricata*. Consultado en https://docs.suricata.io/en/suricata-8.0.4/.

[26] ownCloud GmbH. *Documentación de ownCloud*. Consultado en https://doc.owncloud.com/.

[27] Rubicon Communications, LLC (Netgate). *Portal oficial de pfSense*. Consultado en https://www.pfsense.org/.

[28] Caddy. *Documentación de Caddy*. Consultado en https://caddyserver.com/docs/.

[29] Let's Encrypt. *Documentación de Let's Encrypt*. Consultado en https://letsencrypt.org/docs/.

[30] ArchWiki. *Rsync (Español)*. Consultado en https://wiki.archlinux.org/title/Rsync.

[31] Proyecto Debian. *Documentación de Debian*. Consultado en https://www.debian.org/doc/.

[32] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot, E. Lear. *RFC 1918: Asignación de direcciones en Internets Privadas*. Consultado en https://www.rfc-es.org/rfc/rfc1918-es.txt.

APÉNDICE A
Relación del Trabajo con los Objetivos de Desarrollo Sostenible (ODS)

La ingeniería informática moderna asume una responsabilidad ineludible frente a los desafíos globales. Este anexo presenta una reflexión estructurada sobre el impacto ético, medioambiental y tecnológico del Trabajo Final de Grado en relación con las directrices estipuladas por la Agenda 2030 de las Naciones Unidas. A través del análisis de las decisiones de diseño adoptadas, se demuestra que la implementación de infraestructuras autoalojadas puede superar los requisitos puramente técnicos para convertirse en un agente activo de sostenibilidad y consumo responsable.

A.1 Grado de relación con los ODS

A continuación se detalla el nivel de alineación del proyecto con cada uno de los Objetivos de Desarrollo Sostenible:

ODS	**Denominación**	**Alto**	**Medio**	**Bajo**	**No Procede**
ODS 1	Fin de la pobreza				X
ODS 2	Hambre cero				X
ODS 3	Salud y bienestar				X
ODS 4	Educación de calidad				X
ODS 5	Igualdad de género				X
ODS 6	Agua limpia y saneamiento				X
ODS 7	Energía asequible y no contaminante		X		
ODS 8	Trabajo decente y crecimiento económico		X		
ODS 9	Industria, innovación e infraestructura	X			
ODS 10	Reducción de las desigualdades				X
ODS 11	Ciudades y comunidades sostenibles				X
ODS 12	Producción y consumo responsables	X			
ODS 13	Acción por el clima				X
ODS 14	Vida submarina				X
ODS 15	Vida de ecosistemas terrestres				X
ODS 16	Paz, justicia e instituciones sólidas				X
ODS 17	Alianzas para lograr los objetivos				X

Tabla A.1: Matriz de relación con los Objetivos de Desarrollo Sostenible.

A.2 Justificación del impacto

El presente Trabajo Final de Grado contribuye de manera significativa a los siguientes objetivos:

- **ODS 7 (Energía asequible y no contaminante):** La consolidación de servicios mediante el hipervisor Proxmox permite optimizar el consumo energético de forma drástica y medible. En la arquitectura de redes tradicional, una infraestructura de esta magnitud requeriría múltiples dispositivos físicos encendidos de forma ininterrumpida, como un enrutador perimetral dedicado, un servidor de almacenamiento en red independiente y un equipo de cómputo adicional para las aplicaciones. La virtualización revierte esta ineficiencia al aprovechar al máximo la capacidad de procesamiento de un único servidor anfitrión, reduciendo el consumo eléctrico en estado de reposo de tres equipos independientes a solo uno. El procesador utilizado cuenta con estados de energía avanzados que reducen el voltaje y la frecuencia de los núcleos cuando los contenedores no exigen cálculos complejos, manteniendo un perfil de consumo mínimo durante la mayor parte del día.

  Adicionalmente, el diseño térmico y eléctrico del servidor juega un papel fundamental en la sostenibilidad energética del proyecto. La selección de una fuente de alimentación con certificación de eficiencia máxima garantiza que la conversión de corriente alterna a corriente continua se realice con una pérdida térmica muy pequeña. Al operar en modo pasivo durante cargas de trabajo estándar, la fuente elimina el consumo eléctrico de su propio ventilador. Esta eficiencia eléctrica se complementa de forma sinérgica con la estrategia de refrigeración por presión estática positiva. La prevención sistemática del ingreso de polvo no solo cumple una función de limpieza, sino que evita que las partículas actúen como aislantes térmicos sobre los disipadores. En sistemas convencionales, la acumulación de suciedad obliga a los ventiladores a incrementar sus revoluciones para compensar la pérdida de transferencia de calor, elevando el consumo eléctrico de manera progresiva. Al mantener los componentes mecánicos y electrónicos libres de obstrucciones, la infraestructura asegura que cada vatio consumido se traduzca en rendimiento real, manteniendo una demanda eléctrica plana y predecible a lo largo de los años.

- **ODS 8 (Trabajo decente y crecimiento económico):** La implementación de infraestructuras autoalojadas ejerce un impacto económico significativo al promover la descentralización del capital tecnológico. En el modelo de consumo predominante, los usuarios e instituciones destinan presupuestos recurrentes al pago de suscripciones en plataformas centralizadas, operadas por conglomerados internacionales. Esta dinámica genera una fuga constante de capitales hacia el exterior, limitando el desarrollo económico regional. Al redirigir esta inversión hacia la adquisición de componentes físicos en el mercado local, el proyecto estimula directamente el comercio de proximidad. La compra de placas base, procesador, unidades de almacenamiento, entre otros, a proveedores regionales fomenta la creación y el mantenimiento de puestos de trabajo locales, apoyando a un sector especializado que basa su modelo de negocio en la venta de equipamiento tecnológico.

  Adicionalmente, la adopción de este modelo arquitectónico por parte de pequeñas y medianas empresas transforma la demanda laboral del sector informático. Mientras que la dependencia de servicios externos reduce la necesidad de personal técnico local, la transición hacia infraestructuras privadas requiere la contratación de profesionales altamente cualificados. El diseño, despliegue y mantenimiento de redes virtuales, sistemas de archivos avanzados y contenedores de aplicaciones demanda ingenieros y administradores de sistemas con un conocimiento profundo de

las herramientas. De este modo, se impulsa la creación de empleo de calidad y bien remunerado en la región, en lugar de delegar la administración técnica a centros de datos ubicados en el extranjero.

- **ODS 9 (Industria, innovación e infraestructura):** Este objetivo constituye el pilar filosófico y estructural del trabajo. El despliegue de una infraestructura de red y almacenamiento resiliente fomenta la innovación tecnológica al demostrar que los paradigmas corporativos son escalables y adaptables a entornos de menores recursos. Al utilizar plataformas de código abierto y virtualización avanzada, se democratiza el acceso a capacidades operativas de profesionales, tales como la protección de datos mediante sistemas de archivos tolerantes a fallos, la orquestación ágil de contenedores, la inspección profunda de paquetes y el enrutamiento seguro mediante redes privadas virtuales de conexión directa.

  Esta arquitectura establece un modelo de referencia que desafía el monopolio de las corporaciones tecnológicas internacionales. Demuestra de manera irrefutable que los usuarios individuales, los investigadores y las pequeñas organizaciones pueden construir ecosistemas digitales independientes, seguros y altamente disponibles, sin depender financieramente de los modelos de suscripción recurrentes dictados por los grandes conglomerados de la industria. La innovación de este proyecto no reside en la invención de nuevos protocolos, sino en la integración de herramientas existentes para resolver problemas contemporáneos de privacidad, latencia y velocidad de acceso a los datos.

- **ODS 12 (Producción y consumo responsables):** El proyecto ejerce un impacto directo, profundo y cuantificable en la mitigación de la contaminación tecnológica, al fundamentar la construcción del servidor casi en su totalidad en componentes físicos reutilizados. El paradigma de la obsolescencia programada y los ciclos de actualización comercial acelerados han generado una crisis global de basura electrónica, donde equipos con una inmensa capacidad de cómputo son desechados de manera prematura. Esta tesis aplica con rigor el paradigma de la economía circular, interceptando el modelo lineal de fabricación y desecho.

  La decisión arquitectónica de otorgar una segunda vida a un chasis de formato torre, una placa, un procesador y una fuente de alimentación evita que estos elementos engrosen las alarmantes estadísticas de residuos electrónicos. La fabricación de circuitos integrados es uno de los procesos industriales con mayor impacto medioambiental del planeta, requiriendo un volumen masivo de agua purificada, un gasto energético intensivo y la minería de tierras raras a menudo procedentes de zonas de conflicto geopolítico. Al rescatar este equipamiento plenamente funcional, el proyecto neutraliza en gran parte la huella de carbono y el impacto ecológico asociados a la fase de manufactura de un servidor nuevo.

  Esta estrategia de recuperación y máxima optimización de recursos se completa con la integración de dos módulos de memoria RAM, dos unidades de almacenamiento de estado sólido y un disco duro mecánico. Al diseñar el entorno operativo alrededor de contenedores Linux, los cuales demandan muchos menos recursos en comparación con los sistemas tradicionales, se demuestra que los componentes de generaciones anteriores son perfectamente viables para ejecutar infraestructuras con altas capacidades. Este enfoque responsable no solo evita la acumulación de desechos por líquidos tóxicos que se generan cuando el agua (lluvia o humedad) percola a través de los componentes, arrastrando contaminantes en vertederos, sino que asume una postura ética a favor del consumo sostenible, probando que la administración de sistemas no requiere la adquisición constante de dispositivos de última generación.