



# EMUHUB

---

## TRABAJO DE FINAL DE GRADO

Borja Garcia Amengual Bartomeu Cifre Ramis



CIFP Francesc de Borja Moll

ASIX Presencial 2º año

Daniel Pueyo Soler

## <> ÍNDICE

<> ÍNDICE.....	1
<> CRONOLOGÍA DE TFG.....	2
<> ABSTRACT.....	7
<> SÍNTESIS.....	7
<> PRESENTACIÓN.....	8
<> HARDWARE y RECURSOS.....	10
<> DESCRIPCIÓN DE LA ARQUITECTURA.....	14
<> VIRTUALIZACIÓN PROXMOX - PQ ESTE HYPERVISOR.....	16
<> APACHE GUACAMOLE - CENTRALIZACIÓN.....	21
<> TAILS SCALE - A PROBLEMAS, SOLUCIONES.....	22
<> DOMINIO, CLOUDFLARE, MIEDO AL EXTERIOR.....	24
<> SERVIDOR LAMP ; UN TRIDENTE CLÁSICO.....	27
<> PÁGINA WEB, EMPEZAR DE CERO ES PEOR QUE EMPEZAR DE NUEVO.....	29
<> KUMA, PROMETHEUS, GRAFANA: EL OJO TODO LO VE.....	33
<> COPIAS DE SEGURIDAD Y ALMACENAMIENTO - IMPRESCINDIBLES.....	36
<> TESIS DE LA DEFENSA.....	37

## <> CRONOLOGÍA DE TFG

### I. Infraestructura de Virtualización y Recursos (PROXMOX)

#### Fase Inicial: Selección y Preparación del Hardware

- Selección del hardware recomendado para el host físico del proyecto.
- Especificaciones técnicas del servidor anfitrión (PC server host).
- Detección de limitaciones en el rendimiento al utilizar máquinas virtuales exclusivamente; evaluación de la potencia disponible.
- Instalación y configuración de almacenamiento compartido mediante carpeta de red (utilizando discos físicos).
- Configuración de recursos: asignación de CPU, RAM y almacenamiento a cada contenedor (LXC) y máquina virtual (VM).
- Pruebas de rendimiento y métricas para evaluar la eficiencia general del entorno.

#### Implementación de Proxmox VE

- Instalación de Proxmox VE (PVE) como entorno de virtualización.
- Virtualización de sistemas operativos:
  - Se utiliza Ubuntu 22.04 LTS Jammy Jellyfish exclusivamente desde su repositorio oficial.
- Virtualización de contenedores (LXC) y máquinas virtuales (KVM):
  - Duplicati (LXC-PVE): backups en carpeta compartida (HDD externo).
  - Apache Guacamole (LXC-101): acceso remoto vía navegador con protocolo VNC/RDP.
  - Kuma (LXC-103): herramienta de monitorización del estado del sistema.

- Grafana (LXC-104): visualización de datos y métricas del sistema.
- Prometheus (LXC-105): recopilación de logs y métricas para su análisis.

#### Máquinas Virtuales Ubuntu (GUI + Emuladores)

- Instalación de VNCServer y entorno de escritorio XFCE.
- Configuración de archivos xstartup y servicios VNC para automatizar sesiones gráficas.
- Justificación técnica: XFCE liviano, bajo consumo de recursos, ideal para ejecución de emuladores.
- Instalación y configuración de emuladores personalizados.
- Restricción del sistema operativo para limitar su uso exclusivo al emulador.
- Implementación de scripts automáticos para:
  - Arrancar el emulador.
  - Reiniciar el sistema si el emulador falla.
  - Ejecutar el entorno gráfico.
- Transferencia de archivos de juegos y clonado de VMs para replicar el entorno.

#### II. Entorno Web y Acceso Remoto (Guacamole + Web + Servidor LAMP)

##### Apache Guacamole

- Instalación de LXC Guacamole utilizando scripts auxiliares para PVE.
- Configuración avanzada: privilegios, recursos, contraseñas.
- Desde el puerto 8080 (redireccionado a 8443) se accede a la interfaz de administración.
- Creación de conexiones seguras entre Guacamole y las VM con emuladores (protocolo VNC sobre HTTPS).

- Configuración de usuarios individuales por VM, con credenciales personalizadas.
- Acceso mediante URL pública a cada entorno emulado.

#### Tailscale: Red Privada Virtual

- Creación de cuenta y conexión de todos los nodos del proyecto a la red privada Tailscale.
- Ejecución de tailscaled y tailscale up en cada nodo.
- Resultados:
  - Conectividad segura desde cualquier lugar.
  - Acceso remoto al entorno PVE y servicios internos desde la red TOMEU.

#### Dominio y Cloudflare

- Compra del dominio emuhub.org directamente desde Cloudflare.
- Configuración completa:
  - Email routing
  - Registros DNS
  - Apertura y redirección de puertos (443 → 8443)
  - Configuración de UFW en Guacamole (LXC 101)
  - Configuración de Apache como proxy reverso
  - Protección mediante .htpasswd
  - Blindaje de servicios y pruebas de seguridad (inyección SQL, pruebas de DDoS pendientes).
  - Acceso seguro mediante certificados SSH originales.

#### Servidor LAMP + PHPMYADMIN

- Instalación de Apache2, MariaDB y PhpMyAdmin en el contenedor Guacamole.
- El servidor LAMP centraliza conexiones RDP y la web del proyecto.

- Configuración de virtualhosts para alojar múltiples servicios.
- Migración del contenido /emubase a /espacio, montado como disco compartido en red.
- Acceso a PhpMyAdmin desde la red TOMEU gracias a la integración con Tailscale.

## Desarrollo Web

- Reutilización de proyecto OOP (mysocialweb.iaw).
- Reestructuración completa del HTML y rediseño CSS desde cero.
- Expansión de funcionalidades PHP:
  - Registro con selección de avatar.
  - Nuevas columnas en la base de datos (profile\_img, etc.).
  - Mensajería instantánea (tabla all\_mssg).
  - Timeline, perfil de amigos, consolas virtuales conectadas.
  - Cambios de usuario, email, contraseña y eliminación de cuenta.
- Integración de JavaScript con animaciones y mejoras visuales.

## III. Seguridad, Monitorización y Backup

### Monitorización: Prometheus + Grafana + Kuma

- Sistema centralizado para supervisión de servicios e infraestructura.
- Recopilación de métricas: accesos, errores, logs, tiempos de respuesta.
- Transformación de datos en dashboards legibles (Grafana).
- Validación del estado general del entorno virtualizado y detección temprana de fallos.

### Copias de Seguridad y Almacenamiento

- Configuración de Duplicati para respaldos periódicos:

- Destino: unidad de red (/mnt/pbemuhub) conectada físicamente a otro host.
- Backups automáticos desde PVE y también desde dispositivos Windows.
- Estrategia híbrida: copias en la nube, copias locales y backups manuales del PVE.

## <> ABSTRACT

EMUHUB is a remote proxmox virtualization environment, specifically tailored for the deployment and management of emulator systems. Developed as a final degree project, this initiative serves both educational and particular purposes, aligning with open-source principles and the self-hosted community ethos.

EMUHUB is rather a technical synthesis of the academic content covered during the degree than an open-source contribution to the emulation and virtualization communities, but has kept the aim of acknowledging the technical evolution of video games while preserving their heritage.

The architecture integrates a PVE node hosting virtual machines and four lightweight containers, focusing on performance, accessibility, and availability, supported with a fully functional web platform that enables user interaction, profile management, and emulator safe and easy access.

## <> SÍNTESIS

EMUHUB es un entorno de virtualización proxmox remoto para emuladores. Esta arquitectura se plantea tanto como un trabajo de fin de curso como un proyecto orientado a la comunidad self-hosted. Se espera representar una síntesis fiel, y lo más completa posible a los materiales y clases impartidas en el grado, haciendo uso de herramientas y tecnologías habituales durante la formación.

Asimismo, deseamos poder ofrecer este esfuerzo como configuración de código abierto en un repositorio público GitHub. La construcción y optimización de una consola total nos pareció una idea apasionante, además de un desafío técnico para un primer contacto con tecnologías remotas y accesibles desde redes externas.

## <> PRESENTACIÓN

*El entorno de virtualización proxmox para emulador es un proyecto de fin de curso destinado a fines educativos o particulares, no pretende ni busca incentivar la distribución de productos de terceros. Todo el entorno ha sido diseñado sin ánimo de lucro y desarrollado con fines estrictamente académicos,*

*Todo su contenido está dedicado a la investigación y análisis de las tecnologías involucradas, con el objetivo de preservar y recordar la historia y evolución de los videojuegos tanto a nivel técnico como a escala personal.*

*El uso ilegal de software protegido por derechos de autor no está respaldado bajo ninguna circunstancia. Los emuladores aquí presentados tratan de respetar en todo momento la legislación vigente en materia de propiedad intelectual.*

El entorno se ajusta a las necesidades más cruciales para cualquier contexto de emulación: **Optimización, Accesibilidad y Disponibilidad.**

### **OPTIMIZACIÓN:**

Proxmox aloja en su nodo PVE una red de MÁQUINAS VIRTUALES y CONTENEDORES ultraligeros para favorecer el rendimiento de los sistemas usando las dependencias imprescindibles y paquetes habilitados por el repositorio oficial de Ubuntu.

- > 6 VM Ubuntu 22.04 LTS JAMMY JELLYFISH
- > 4 CONTENEDORES LXC

## **ACCESIBILIDAD:**

El SERVIDOR LAMP cuenta con el tridente clásico: APACHE2, PHPMYADMIN y MARIADB. El desarrollo web ha consistido en un auténtico ejercicio de restauración, estilización y debugging del proyecto de Programación Orientada a Objetos. Ofrece a los usuarios una interfaz agradable y coherente con su ejecución, donde poder registrarse, publicar contenido, chatear con otros usuarios, ajustar sus perfiles, ayudarse del asistente y por supuesto, jugar.

## **DISPONIBILIDAD:**

Con Cloudflare emuhub.org ya es una parcela adquirida en internet. El servicio cubre las conexiones al PVE, certificados oficiales, protocolos cifrados con 443 y redirigidos a puertos escogidos manualmente. Apache Guacamole administra las conexiones entrantes desde las VM y centraliza todas en un mismo HOST fortalecido con UFW y soluciones de seguridad y monitorización. Las VM también incorporan scripts automatizados para asegurar una conexión estable y fluida, más lograda aún en su respuesta ante interferencias.

## <> HARDWARE y RECURSOS

Inicialmente, hubo varios prototipos para familiarizarse con las sensibilidades de proxmox, se instalaron pruebas en máquinas virtuales Virtualbox, y en otros equipos físicos. Finalmente, optamos por descartar las máquinas previas y destinar el ordenador con mayor rendimiento a la instalación de Proxmox.

### PROTOTIPOS:

{ <i>En la empresa me prestaron un ODROID para realizar pruebas de administración con Proxmox y familiarizarme con el PVE }</i>	MEMORIA	16 GB
	PROCESADOR	Intel Processor N97
	ALMACENAMIENTO	1 TB
	ADAPTADOR DE RED	BRIDGED vmbr0
Buen rendimiento general, es un producto industrial, destinado a cargas de trabajo muy pesadas ; no ofrece la máxima velocidad, pero sí una performance sólida y estable. Fue muy útil para determinar la configuración y paquetes necesarios para construir todo el entorno		

{ <i>Usamos un disco Kingston 240 SSD limpio para una instalación de pruebas, especialmente para hardening }</i>	MEMORIA	32 GB
	PROCESADOR	Intel(R) Core(TM) i7-4770S 3.10GHz
	ALMACENAMIENTO	SSD KINGSTON 240GB
	TARJETA GRÁFICA	NVIDIA GeForce GTX 1050
Rendimiento insuficiente. El procesador sufría mucho con solo 3 VM emuladores simultáneos, además la conexión era por Wifi y no cabe, la estabilidad y la velocidad eran otro inconveniente. Decidimos conservar el PVE para realizar pruebas de mantenimiento y comprobar la funcionalidad de los scripts con los emuladores.		

HOST PROXMOX BORJA	MEMORIA	32 GB
	PROCESADOR	AMD Ryzen 5 5600X 6Nc
	ALMACENAMIENTO	SSD 1TB
	TARJETA GRÁFICA	NVIDIA GeForce RTX 4060
El rendimiento de este equipo ha sido excelente. Muy superior en cuanto a los recursos solicitados, pudiendo correr todo el PVE simultáneo mientras recibe conexiones en cada una de sus máquinas. El rendimiento es estable, la latencia tampoco es problema gracias a la conexión por fibra 1GB simétrica que dispone el HOST PROXMOX.		

## CONSIDERACIONES

La idea inicial era virtualizar completamente el entorno, incluyendo el PVE, y anidar la virtualización de los emuladores Ubuntu LTS con escritorios XFCE accesibles por VNC. Sin embargo, tras las primeras pruebas, detectamos importantes limitaciones de rendimiento:

Latencia elevada en la ejecución de emuladores dentro de máquinas virtuales. El aislamiento de la tarjeta gráfica entre el HOST y una VM anidada al PROXMOX virtualizado supuso atascos en el desarrollo. Debido a la dificultad de aplicar passthrough o aceleración 3D completa en entornos virtualizados sin configuración avanzada, y experimentar cuellos de botella en el acceso a memoria y CPU durante la ejecución simultánea de máquinas, optamos por descartar esta primera idea.

A pesar de todo, las observaciones nos ayudaron en la toma de decisiones estratégicas para distribuir mejor la carga de trabajo y optimizar el rendimiento general del sistema.

## **ALMACENAMIENTO**

Además del SSD interno destinado al sistema base y las máquinas virtuales, utilizamos un segundo disco externo HDD conectado a otro equipo fuera de la red proxmox, como almacenamiento compartido de red: un medio físico de respaldo aislado. Esto nos permitió separar datos menos críticos o con menor frecuencia de acceso, y evitar saturar el almacenamiento principal del host, se puede añadir espacio en caso de necesidad a la máquina o contenedor que lo requiera.

La conexión al recurso compartido se realizó mediante las utilidades CEPH/CEFS. Para ello se tienen que discriminar como privilegiados a los equipos donde se quiera tener acceso al recurso. Se creó un punto de montaje en el host Proxmox (/mount/pbe) y se accedió al recurso remoto a través de una red privada establecida con Tailscale, lo cual permitió una conexión segura y sin necesidad de exponer puertos a Internet. Hay que configurar los archivos.

De nuevo, con este sencillo recurso, logramos extender el almacenamiento, asegurar copias de respaldo, transferencia privada de archivos entre nodos, y almacenamiento auxiliar para tareas que no requieren demasiados recursos.

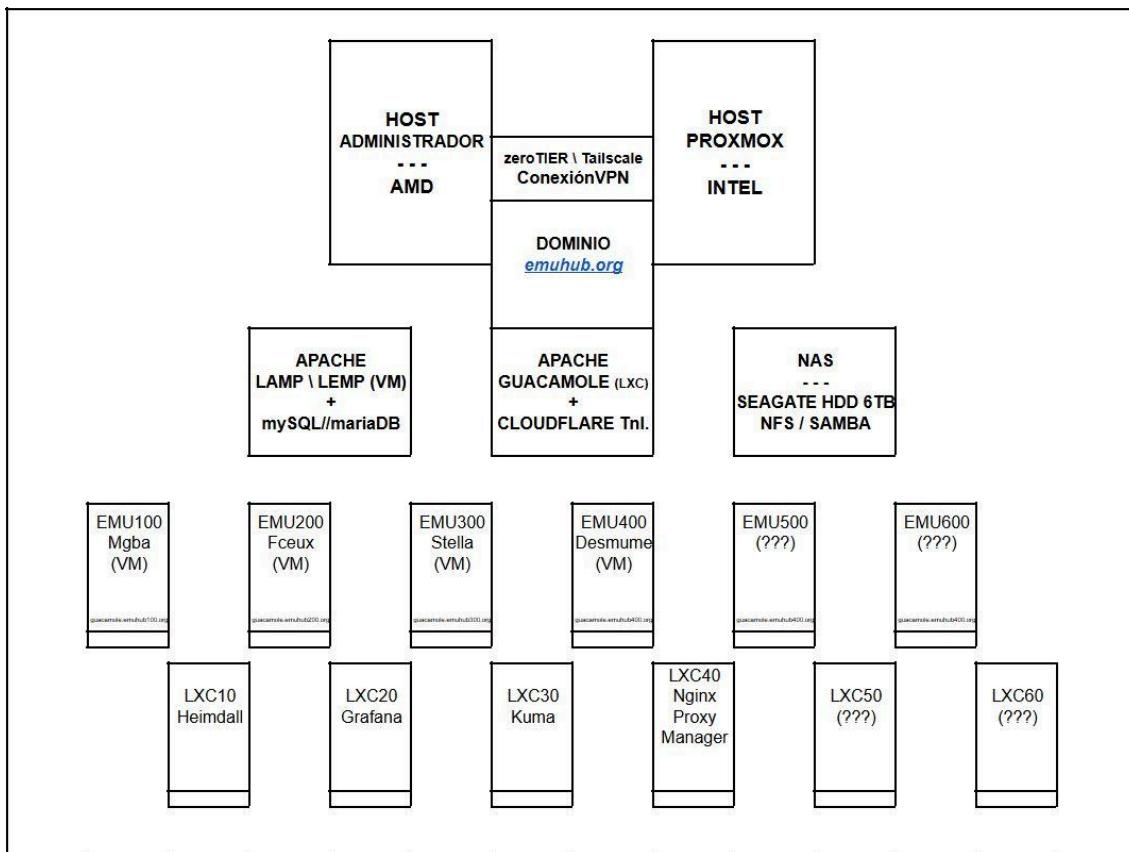
## **ANÁLISIS DEL RENDIMIENTO**

Aunque el desglose detallado de cada máquina virtual se encuentra en su sección correspondiente hemos priorizado áreas según su relevancia en la ejecución de los emuladores y el tráfico concurrente de conexiones.

La asignación de CPU, RAM y disco entre contenedores y máquinas ha sido medida con testeos y prototipos para no saturar el host. Los datos recogidos mediante Prometheus y visualizados en Grafana confirman que el sistema se mantiene estable en momentos donde el volumen de trabajo es elevado. No hay sobrecargas ni en CPU ni en RAM, y las gráficas de uso mostraban un comportamiento ligero, moderado en momentos puntuales, pero nunca insuficiente.

La monitorización constante permite realizar ajustes finos en tiempo real y garantiza la fiabilidad del entorno durante toda la fase de pruebas y desarrollo.

## <> DESCRIPCIÓN DE LA ARQUITECTURA



- Proxmox maneja las conexiones de red seguras a través de Cloudflare Tunnel. Conexión entre los servidores locales y los servicios externos, los que sean inaccesibles alojados en la red tailscale.
- Guacamole Apache, solución para CENTRALIZAR las VM. Un contenedor que administra las conexiones VNC salientes de cada emulador.
- La Exposición a través del proxy de Cloudflare, una protección incluida en sus dominios. Permite acceso remoto seguro a los emuladores desde el navegador. De este modo, solo se necesita

exponer 1 servicio al exterior, el Guacamole central, quien gestiona y proyecta las sesiones iniciadas en las VM.

- Configurar entorno LAMP o LEMP dedicado a la web: Apache + PHP + MariaDB (LAMP) o Nginx + PHP-FPM + MariaDB (LEMP).
- Emuladores (GBA, N64, etc.): Cada VM de emulador contiene un entorno Ubuntu con VNC y TigerVNC para ofrecer el acceso remoto.
- Conexión al almacenamiento compartido a través de NFS/Samba.
  - HDD montado en el host de Proxmox o en una VM NAS
  - Las VMs emuladoras acceden al recurso /mnt/pbemuhub
- Contenedores LXC (Heimdall, Grafana, Prometheus, Kuma). Permiten centralizar los servicios en el nodo PVE, su instalación se lleva a cabo mediante un script en el shell, gracias a la tecnología KVM, el sistema es capaz de crear una distribución ultra ligera para desplegar un único servicio enlazado al proxmox.
- Cloudflare Tunnel: DOMINIO emuhub.org, PUERTO 443; Usado para exponer todos los servicios a internet de manera segura. HTTPS INTERNO, redirección 443 por 8443. Protección adicional mediante firewall UFW y certificados y soluciones de Cloudflare.
- Aislamiento del usuario final: Simplificar XFCE. Permisos limitados: el sistema no permite escritura sin root, solo lectura. Usuario auto-login. Solo una ventana abierta (el emulador). Sin acceso a terminal, menú, etc. Bloquear KEYBINDS, HOTKEYS sensibles. Eliminar acceso al panel, escritorio, navegador, etc.

# <> VIRTUALIZACIÓN PROXMOX - PQ ESTE HYPERVISOR

## VIRTUALIZACIÓN CON PROXMOX VE

El proyecto está basado en Proxmox Virtual Environment (PVE), un hipervisor y solución open-source para virtualización bare-metal, que permite administrar máquinas virtuales (KVM) y contenedores (LXC) desde una interfaz web intuitiva y eficiente. Optamos por proxmox ya que al ser open-source enlaza con nuestros ideales y es la mejor opción de mercado sin necesidad de pagar.

## ENTORNO BASE

Proxmox se instaló directamente en un disco SSD crucial reservado para él. Para determinar la eficiencia del disco contamos con herramientas como HDDSCAN o DiskGenius. Mediante la ISO oficial de Proxmox VE, asegurando un entorno estable y optimizado. La instalación la llevamos a cabo mediante una unidad booteable Ventoy<sup>1</sup>. La elección del sistema operativo base para las máquinas virtuales y contenedores fue Ubuntu 22.04 LTS (Jammy JellyFish), el sistema operativo con el que más familiarizados estamos los autores, sin entrar en Windows.

---

<sup>1</sup> Ventoy es una herramienta de código abierto que transforma una unidad USB en un dispositivo de arranque múltiple de manera sencilla y versátil. A diferencia de los métodos tradicionales, que requieren formatear la unidad y "quemar" cada sistema operativo por separado, Ventoy permite copiar múltiples archivos ISO directamente al USB y luego seleccionarlos desde un menú de arranque al encender el equipo. Es una solución sumamente práctica para técnicos, administradores de sistemas, y cualquier persona que necesite llevar consigo una suite de herramientas portátiles.

## SERVICIOS DESPLEGADOS

ID	SERVICIO	FUNCIÓN
101	APACHE GUACAMOLE	Cliente VNC/RDP accesible vía navegador
102 (PVE)	DUPPLICATI	Solución código abierto para backups -- recurso de red compartido HDD
103	UPTIME KUMA	Estado de salud y alertas sobre los servicios y equipos del PVE
104	GRAFANA	Monitorización y análisis de métricas, registros y accesos
105	PROMETHEUS	Recopilación de métricas para conversión legible

## VM EMULADOR DESPLEGADAS

ID	CONSOLA	EMULADOR
100	GAMEBOY ADVANCE, 2001	MGBA-QT
110	NINTENDO DS, 2004	DESMUME
120	PLAYSTATION ONE, 1994	PCSXR
130	SEGA SATURN, 1994	YABause

140	NINTENDO 64, 1996	MUPEN64
150	NES, 1983	FCEUX

Red de 6 máquinas virtuales (VMs) en Proxmox VE, todas basadas en Ubuntu 22.04 LTS. Distribución estable, tiene buen soporte de paquetes y nos permite depender únicamente del repositorio oficial. (jammy).

No usamos ningún PPA ni fuentes externas, lo que también refuerza la seguridad. Las VMs están diseñadas para una única tarea: emular consolas de forma encapsulada, sin posibilidad de que el usuario modifique el sistema.

Las Máquinas Virtuales son idénticas en recursos, optimización y automatización. Usamos una VM Clon para desplegar todas las demás y ajustar la configuración modularmente. Ubuntu 22.04 LTS no proporciona una GUI para el usuario, pero más que suponernos un problema, lo minimiza. Usar SO gráficos tradicionales conlleva un volumen muy exagerado en cuanto a carga de trabajo se refiere.

VM100-150: EMUHUB100 - EMUHUB150	MEMORIA	4 GB
	PROCESADOR	2 CPU
	BIOS	SeaBIOS
	MÁQUINA	i440fx
	CONTROLADORES	VirtIO SCSI Single
	ALMACENAMIENTO	LOCAL-LVM {20G}
	ADAPTADOR DE RED	BRIDGED vmbr0

The screenshots illustrate the Proxmox Virtual Environment 8.4.1 interface. The top screenshot shows the 'Nodes' view, listing various hosts (lxc, qemu, storage) with their names, types, descriptions, disk usage, memory usage, CPU usage, uptime, host CPU usage, host memory usage, and tags. The bottom screenshot shows the 'Summary' view, displaying real-time system metrics for CPU usage, load average, RAM usage, and HD space. It also includes a detailed hardware configuration section and a graph showing CPU usage over time.

Inicialmente, instalamos XFCE, un escritorio más liviano compatible con el sistema. Para poder acceder a la interfaz, disponemos de VNC SERVER, una herramienta de control remoto de escritorio, se utiliza en aplicaciones como TeamViewer. Nos permite controlar un sistema informático desde cualquier lugar de forma gratuita o para uso doméstico, además de editar limitaciones y asignar permisos a usuarios

según la actividad a realizar. Finalmente, proyectamos este escritorio mediante Apache Guacamole desplegado como un LXC. Este último equipo se encarga de centralizar todas las sesiones VNC activas y sacarlas a internet bajo el dominio emuhub.org. Esta cadena de producción es cíclica:

- a. CONFIGURAMOS UN SERVICIO `systemd vncservice` QUE ARRANCA UNA INSTANCIA EN EL DISPLAY 1.
- b. CONFIGURAMOS EL `xstartup` PARA ARRANCAR EL EMULADOR (`Xstartup` se ejecuta cuando hay una sesión VNC activa)
- c. CONFIGURAMOS SCRIPTS AUTOMATIZADOS: `launch-(emulador).sh` y `monitor_(emulador).sh` MANEJADOS POR CRON. Si detectan alguna irregularidad en el emulador, la VM se reinicia, así como sus servicios. Asegurando la disponibilidad a las VM incluso si experimentan alguna interferencia o interrupción.

Otras medidas de seguridad implementadas en todas las VM son: todos los directorios del sistema están protegidos contra escritura. No existe forma de abrir un terminal (desinstaladas). No se puede instalar nada. Cerrar el emulador conlleva el reinicio inmediato del sistema. Las ROMs ya están cargadas en cada máquina, dentro de la carpeta `/home/emuhub100/Descargas`. No hay sincronización entre máquinas ni transferencia automática.

LXC 101: LAMP + APACHE GUACAMOLE	MEMORIA RAM	2 GB
	MEMORIA SWAP	512 MB
	PROCESADOR	1 CPU
	ALMACENAMIENTO	LOCAL-PVE {4G}
	ADAPTADOR DE RED	BRIDGED vmbr0

LXC 103: UPTIME KUMA	MEMORIA RAM	1 GB
	MEMORIA SWAP	512 MB
	PROCESADOR	1 CPU
	ALMACENAMIENTO	LOCAL-PVE {4G}
	ADAPTADOR DE RED	BRIDGED vmbr0

LXC 104: GRAFANA	MEMORIA RAM	512 MB
	MEMORIA SWAP	512 MB
	PROCESADOR	1 CPU
	ALMACENAMIENTO	LOCAL-PVE {2G}
	ADAPTADOR DE RED	BRIDGED vmbr0

LXC 104: PROMETHEUS	MEMORIA RAM	2 GB
	MEMORIA SWAP	512 MB
	PROCESADOR	1 CPU
	ALMACENAMIENTO	LOCAL-PVE {4G}
	ADAPTADOR DE RED	BRIDGED vmbr0

## <> APACHE GUACAMOLE - CENTRALIZACIÓN

Apache Guacamole<sup>2</sup> es una herramienta de acceso remoto que permite conectarse a escritorios remotos directamente desde un navegador web, sin necesidad de instalar software adicional en el equipo cliente. Esta plataforma es compatible con los protocolos estándar más utilizados: VNC, RDP, SSH, etc. lo que facilita la conexión a varios sistemas.

---

<sup>2</sup> [GUÍA DE INSTALACIÓN ÚTIL](#)

Guacamole puede ser desplegado dentro de LXC y utilizarlo como destino de conexiones remotas y gestionado como pasarela central de acceso remoto. Esta implementación es una práctica mucho más eficiente, segura y escalable, casi obligada; instalar el cliente guacd en cada VM es un gasto innecesario de recursos y espacios.

Apache Guacamole puede configurarse tanto desde sus archivos de configuración como desde una interfaz, similar a proxmox, accesible por el navegador. Agregamos usuarios específicos por máquina y conexión ; aunque tuvimos que limitar los accesos a las máquinas a uno concurrente por cada una.

## <> TAILSCALE - A PROBLEMAS, SOLUCIONES

### TAILSCALE

Durante el desarrollo del proyecto se hizo imprescindible contar con una red privada entre los distintos dispositivos participantes, sin necesidad de abrir puertos adicionales ni depender de direcciones IP públicas. Para resolver esto, se optó por utilizar Tailscale, una solución de red VPN basada en WireGuard, que permite conectar equipos de forma segura y directa, incluso si están detrás de NAT o firewalls.

El primer paso fue la creación de una cuenta gratuita en Tailscale, vinculada al correo del proyecto [tybemuhub@gmail.com](mailto:tybemuhub@gmail.com). A partir de ahí, se fueron registrando todos los dispositivos<sup>3</sup> implicados en la VPN.

#### HOST TOMEU

*IP Tailscale: 100.98.198.121*

*Sistema operativo: Windows 10 22H2*

#### SERVIDOR PVE

*IP Tailscale: 100.88.35.66*

*Kernel: Linux 6.8.12-10-pve*

Este proceso inicia el servicio Tailscale en segundo plano y lanza el asistente para iniciar sesión con la cuenta del proyecto, tras lo cual se autoriza el dispositivo y queda enlazado a la red privada. Finalmente, descartamos la mayoría de dispositivos unidos a la red VPN, para aislarlos y no gastar recursos de red innecesariamente.

---

<sup>3</sup> La unión de cada máquina a la red se realiza de forma sencilla mediante los comando:`tailscaled & tailscale up`

Tailscale permite acceder al entorno Proxmox desde cualquier lugar a través de la IP proporcionada en la consola de administración, sin necesidad de exponerlo directamente a Internet. Esto facilita la gestión del clúster y el mantenimiento de las máquinas virtuales desde fuera de la red local.

The screenshot shows the Tailscale web interface with the following details:

- Header:** tybemuhub@gmail.com, Download, Support, Docs, X icon.
- Navigation:** Machines (selected), Apps, Services, Users, Access controls, Logs, DNS, Settings, Get started.
- Machines Section:**
  - Search bar: Search by name, owner, tag, version...
  - Filter button: Filters ▾
  - Download button: ⬇
  - Table:

MACHINE	ADDRESSES ⓘ	VERSION	LAST SEEN	...
pc tybemuhub@gmail.com	100.98.198.121 ⓘ	1.82.5 Windows 10 22H2	Connected	...
pve tybemuhub@gmail.com	100.88.35.66 ⓘ	1.84.0 Linux 6.8.12-10-pve	Connected	...
  - Add device button: Add device ▾
- Add devices to your network section:**
  - Operating systems: Linux (checked), Windows, Mac, See more.
  - Cloud providers: Add AWS VM, Microsoft Azure, Google Cloud Platform, See more.
  - Containers: Docker, Kubernetes.

Además, al combinar esto con la redirección VPN desde otro equipo (por ejemplo, el portátil de Tomeu), se consigue un entorno totalmente portátil y seguro para administración remota, sin comprometer la seguridad del servidor doméstico.

## <> DOMINIO, CLOUDFLARE, MIEDO AL EXTERIOR

### CLOUDFLARE

Una parte fundamental del proyecto fue la publicación del entorno web de forma segura en Internet. Para ello, se optó por la compra de un dominio propio y su integración con los servicios de protección y redirección que ofrece Cloudflare.

### CONFIGURACIÓN BÁSICA

Se adquirió el dominio emuhub.org directamente a través del registrador oficial de Cloudflare, lo que permitió evitar intermediarios y obtener un precio fijo y transparente. El uso de Cloudflare también ofrecía ventajas adicionales en términos de seguridad, cifrado SSL gratuito y gestión avanzada de DNS.

El dominio se enlazó directamente con la IP pública del host principal (88.20.77.79), habilitando el modo proxy de Cloudflare (ícono naranja), lo cual permite ocultar la IP real, mitigar ataques directos y aplicar políticas de protección en capa 7 (HTTP/HTTPS).

### REDIRECCIÓN DE PUERTOS

Dado que el entorno del proyecto se aloja en un servidor doméstico, fue necesario realizar una apertura de puertos en el router. El objetivo era redirigir las peticiones HTTPS (puerto 443) al puerto 8443 interno, que era el utilizado por Apache Guacamole para su servicio web seguro.

La redirección se configuró así:

Puerto externo 443 → Puerto interno 8443

IP interna del servidor Guacamole: 192.168.1.142

Esta decisión fue arbitraria y podría haberse hecho con otro puerto, pero se eligió 8443 por disponibilidad y para evitar conflictos con otros servicios.

## **SEGURIDAD: FIREWALL y AUTENTICACIÓN**

Para reforzar la seguridad del entorno expuesto, se aplicaron las siguientes medidas:

UFW (Uncomplicated Firewall): Se configuró en el servidor Guacamole para permitir únicamente las conexiones necesarias, principalmente 8443/tcp<sup>4</sup> desde la red de Cloudflare.

Apache con proxy inverso: Se habilitó un proxy sencillo en Apache para enviar las peticiones del dominio hacia los servicios internos (la web y Guacamole).

```
<VirtualHost *:443>
    ServerName emuhub.org

    SSLEngine on
    SSLCertificateFile /etc/ssl/cert.pem
    SSLCertificateKeyFile /etc/ssl/key.pem

    ProxyPreserveHost On
    ProxyPass / http://localhost:8080/
    ProxyPassReverse / http://localhost:8080/
</VirtualHost>
```

---

<sup>4</sup> ufw allow 8443/tcp  
ufw enable, ufw reload

Autenticación HTTP básica: Se incorporó un archivo .htpasswd para proteger accesos sensibles como phpMyAdmin, añadiendo una capa extra de validación antes de la sesión web.

Los servicios publicados cuentan con protección mediante control de sesión en PHP (session\_start()), lo que evita el acceso directo a páginas sin haber iniciado sesión correctamente. Cualquier intento de acceder a rutas internas sin una sesión válida resulta en una redirección o expulsión inmediata.

## <> SERVIDOR LAMP ; UN TRIDENTE CLÁSICO

### SERVIDOR LAMP

Para centralizar las conexiones y alojar la web junto con el cliente VNCserver, configuramos un contenedor LXC dedicado en Proxmox llamado Guacamole, donde instalamos el stack LAMP con las siguientes versiones:

- **Apache 2.4.62 (Debian)**
- **MariaDB 15.1 Distrib 10.11.11**
- **phpMyAdmin 5.2.1deb1+deb12u1**

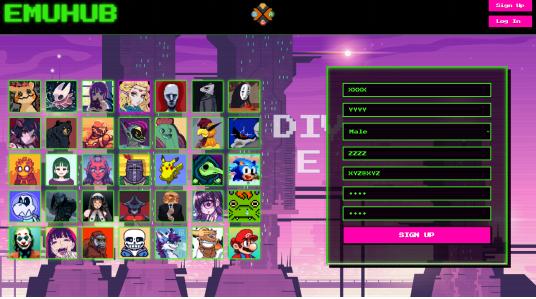
### CONFIGURACIÓN ARCHIVO .conf DE VIRTUALHOST

La instalación se realizó usando los repositorios oficiales de Debian y Ubuntu, principalmente con apt install. VirtualHost y configuración Apache; el archivo de configuración principal del sitio, emuhub.conf está pensado para:

- Redirigir todo el tráfico HTTP (puerto 80) a HTTPS, asegurando seguridad en las comunicaciones.
- Configurar el servidor para responder en HTTPS (puerto 443) con certificados SSL ubicados en /espacio/emubase/ssl/.
- Definir /espacio/emubase como la carpeta raíz de la web, donde está alojado todo el contenido estático y dinámico.
- Permitir overrides con .htaccess para mayor flexibilidad.

- Proxy inverso para redirigir todas las peticiones que lleguen a /guacamole/ hacia el cliente VNCserver local corriendo en el puerto 8080, incluyendo soporte para WebSocket, imprescindible para la funcionalidad de Guacamole
- Registro de logs para errores y accesos específicos del sitio.
- Este VirtualHost permite integrar la web y el acceso remoto VNCserver en un único punto centralizado, facilitando la gestión y el acceso desde internet.
- Dentro de /espacio/ (directorio de expansión destinado a añadir capacidad al contenedor LXC que gestiona la página web), carpeta /emubase, contiene toda la página web y los certificados SSL necesarios para HTTPS.
- Aunque no se configuraron reglas específicas de firewall para phpMyAdmin, el acceso a la herramienta está limitado al contenedor Guacamole y a la red interna Tailscale, restringiendo exposición externa directa. Lo protegimos con nuestra contraseña más férrea y un archivo .htpasswd delante.

# <> PÁGINA WEB, EMPEZAR DE CERO ES PEOR QUE EMPEZAR DE NUEVO

	
disclaimer	tomeu_login
	
tomeu_signup	profile_page
	
display_posts	profile_page_B

timeline	about
settings	settings_B

La página web de emuhub ha sido rescatada del proyecto de programación orientada a objetos visto en Implementación de aplicaciones Web:

- > CSS: TOTALMENTE NUEVO
- > HTML: REESTRUCTURADO y EXTENDIDO
- > PHP: AÑADIDAS NUEVAS FUNCIONALIDADES
- > JAVASCRIPT: INCORPORACIÓN AL PROYECTO

## FUNCIONALIDADES

A continuación se describen las funcionalidades testeadas y operativas permitidas en la página web de [emuhub.org](http://emuhub.org):

- Puedes registrar tantos usuarios como deseas.
- Puedes iniciar sesión y disponer de un perfil personal donde

- ajustar tus preferencias y publicar lo que deseas compartir.
- c) Puedes cambiar la contraseña tantas veces como quieras (las contraseñas se cifran inmediatamente al ser introducidas).
  - d) Puedes publicar cualquier contenido que deseas en texto plano.
  - e) Verás cómo tus amigos se agregan a tu lista cuando se registran en la plataforma.
  - f) Puedes cerrar sesión y no es posible acceder al sitio web sin volver a iniciarla previamente. Tus datos están protegidos.
  - g) El panel de administración es una característica que se encuentra deshabilitada, y solo puede ser activada desde el menú de phpmyadmin o editando los datos insertados en la propia base de datos.
  - h) Como administrador, se autoriza acceso a un panel de control donde podrás ver datos descriptivos de los usuarios, registros y también otorgar o revocar privilegios de administrador.
  - i) Puedes visitar los perfiles de otros usuarios desde tu lista de amigos. Solo necesitas hacer clic en el nombre o imagen de alguno de ellos para ser redirigidos a su timeline. Para hacer las conexiones al perfil de otro usuario independientes de la sesión actual se usa el userid de los miembros registrados.
  - j) Puedes elegir una foto de perfil de una piscina predeterminada y cambiarla en los ajustes. También es posible cambiar el nombre de usuario, el correo electrónico, incluso borrar la cuenta.
  - k) Se ha implementado un asistente de diálogo, para guiar a los usuarios durante su primer login e indicarles qué posibilidades existen en nuestra página web.
  - l) La navegación se ha implementado de la forma más accesible y visual posible, con menús desplegables, animaciones y unos estilos coherentes con los protagonistas del proyecto: las consolas retro.

- m) Puedes chatear con todos los usuarios. El emuchat proporciona una interfaz de mensajería en diferido para comunicarse de forma privada con otros usuarios. No contamos con un servicio de mensajería instantánea al uso, pero nos pareció la solución más eficiente en cuanto a optimización de los recursos restantes.
- n) Añadido un disclaimer con información sensible sobre los permisos legales del proyecto para advertir a los navegantes.

## **<> KUMA, PROMETHEUS, GRAFANA: EL OJO TODO LO VE**

### **CONTENEDORES LXC**

Para minimizar los recursos invertidos en los servicios, optamos por utilizar contenedores LXC en lugar de máquinas virtuales completas siempre que fue posible. Los contenedores son más ligeros y permiten un uso más eficiente del hardware.

Las plantillas LXC se descargaron y configuraron mediante línea de comandos usando scripts oficiales disponibles en GitHub, facilitando la automatización y control de la infraestructura.

Para asegurar el correcto funcionamiento y optimización de los recursos del sistema, se implementó un sistema de monitorización basado en Prometheus y Grafana, dos herramientas open-source ampliamente usadas en entornos profesionales.

### **PROMETHEUS**

Prometheus es un sistema de monitorización y recopilación de métricas que funciona recogiendo datos de diferentes fuentes mediante “exporters” o agentes que exponen información sobre el estado de sistemas y aplicaciones. En nuestro caso, Prometheus recopila métricas del propio nodo Proxmox, los contenedores LXC y otros servicios, alimentándose en una base de datos de series temporales.

Estas métricas incluyen uso de CPU, memoria, disco, red, estado de servicios, y cualquier otro dato relevante para conocer la salud del sistema en tiempo real.

## **GRAFANA**

Grafana es una plataforma para visualización y análisis de datos, que se conecta a bases de datos como la de Prometheus para mostrar paneles gráficos personalizados (dashboards). Con Grafana, se crean vistas interactivas que facilitan entender rápidamente el rendimiento y estado del sistema, identificar cuellos de botella o posibles fallos.

En el proyecto, Prometheus actúa como la fuente de datos, recogiendo y almacenando métricas periódicamente, mientras que Grafana consume esos datos para crear dashboards accesibles desde el navegador. Esta integración permite: Monitorizar en tiempo real el rendimiento de las máquinas y contenedores. Visualizar históricos para análisis y optimización. Detectar anomalías o fallos de forma proactiva.

Ambos servicios corren en contenedores LXC gestionados por Proxmox y están configurados para arrancar automáticamente con el sistema mediante systemd.

## **CONSIDERACIONES: NODE-EXPORTER.js<sup>5</sup>**

Para monitorizar el estado y rendimiento de cada máquina virtual Ubuntu 22.04, se instaló el agente oficial Prometheus node\_exporter en cada VM. Este agente recopila métricas del sistema operativo como uso de CPU, memoria, disco y red, y las expone para que Prometheus pueda recogerlas y analizarlas. La instalación se realizó con los comandos estándar de Ubuntu, utilizando el paquete oficial para mayor facilidad y compatibilidad:

---

<sup>5</sup> sudo apt install prometheus-node-exporter

Este paquete configura automáticamente el servicio node\_exporter, que se ejecuta en segundo plano mediante systemd y expone las métricas en el puerto 9100.

Para que Prometheus recoja las métricas de los node\_exporters instalados en las máquinas virtuales, se añade una sección scrape\_configs en su archivo de configuración prometheus.yml como esta:

```
scrape_configs:  
  - job_name: 'node_exporter_vms'  
    static_configs:  
      - targets:  
        - '192.168.1.101:9100'  
        - '192.168.1.102:9100'  
        - '192.168.1.103:9100'
```

## **UPTIME KUMA**

Su instalación y puesta en marcha en un LXC permite tener un panel adicional para supervisar la salud general del sistema de virtualización, aportando una capa extra de control junto con Prometheus y Grafana

Kuma se despliega en un contenedor LXC dedicado dentro de Proxmox (por ejemplo, LXC 103) y ofrece una interfaz sencilla para consultar la disponibilidad y el estado de los nodos, servicios y recursos esenciales del clúster.

Entre sus funcionalidades destacan: Visualización rápida del estado de los nodos PVE, Alertas básicas sobre fallos o indisponibilidades. Monitoreo simplificado sin necesidad de configuraciones complejas.

## **<> COPIAS DE SEGURIDAD Y ALMACENAMIENTO - IMPRESCINDIBLES**

El disco HDD sirve como recurso de red compartido. Ofrece capacidad (ampliable) para almacenar las copias de seguridad y backups del entorno PVE.

Compartimos el recurso mediante las utilidades CIF / CEPH, permiten a las máquinas proxmox acceder al recurso compartido desde Windows si se disponen de las credenciales del usuario.

Los backups se realizan bien por Duplicati , software de código abierto y multiplataforma, diseñado para realizar copias de seguridad cifradas, incrementales y automatizadas. Dispone de una gran variedad de protocolos para transferir los backups entre equipos y nos ha permitido proteger la conexión para recibir las transferencias con el archivo de certificados SSL de cloudflare.

Además de eso, Proxmox cuenta con una solución de backups integrada en su interfaz ; aunque más limitada en parámetros, permite hacer las copias de forma segura, comunicándose exclusivamente con el disco duro.

La transferencias de backups es posible, de nuevo, gracias a la implementación de la red Tailscale, que conecta ambos equipos estando en redes separadas (Borja - Proxmox / Tomeu - HDD). Tuvimos dudas sobre la velocidad de transmisión pero no ha acabado siendo un inconveniente gracias a su programación horaria.

## <> TESIS DE LA DEFENSA

> LA FILOSOFÍA SELF-HOSTED

> UNA SÍNTESIS DE ASIX

> PRUEBA y ERROR

> HAZLO CON PROPÓSITO

