



Universidad Politécnica  
de Madrid

**Escuela Técnica Superior de  
Ingenieros Informáticos**



Grado en Grado en Ingeniería Informática

Trabajo Fin de Grado

# **Despliegue de Escritorios Virtuales con GPU**

Autor: Alejandro Moreno Ferreiro  
Tutor: Fernando Pérez Costoya

Madrid, Enero - 2025

Este Trabajo Fin de Grado se ha depositado en la ETSI Informáticos de la Universidad Politécnica de Madrid para su defensa.

*Trabajo Fin de Grado*

*Grado en* Grado en Ingeniería Informática

*Título:* Despliegue de Escritorios Virtuales con GPU

Enero - 2025

*Autor:* Alejandro Moreno Ferreiro

*Tutor:* Fernando Pérez Costoya

Departamento de Arquitectura y Tecnología de Sistemas Informáticos (DATSI)

Escuela Técnica Superior de Ingenieros Informáticos

Universidad Politécnica de Madrid

# Índice general

<b>1. Introducción</b>	<b>4</b>
1.1. Contexto del proyecto . . . . .	4
1.2. Objetivos . . . . .	5
1.2.1. CeSViMa . . . . .	6
1.3. Estructura del documento . . . . .	7
<b>2. Estado del Arte</b>	<b>8</b>
2.1. Historia y evolución de la virtualización de escritorios . . . . .	8
2.2. Virtualización en las universidades . . . . .	10
2.2.1. Beneficios administrativos . . . . .	10
2.2.2. Beneficios organizacionales y para los estudiantes . . . . .	11
2.2.3. Beneficios educativos . . . . .	11
2.2.4. Tendencias actuales y futuras . . . . .	12
2.3. Escritorios virtuales en la UPM . . . . .	12
2.4. GPU en la Virtualización de Escritorios . . . . .	13
2.5. Elección de herramientas . . . . .	16
2.5.1. El papel del software libre y de código abierto . . . . .	16
<b>3. Desarrollo</b>	<b>18</b>
3.1. Diseño de la arquitectura del sistema . . . . .	18
3.1.1. Flujo de operación . . . . .	19
3.2. Instalación y configuración del hipervisor . . . . .	20
3.2.1. Rol . . . . .	20
3.2.2. Herramientas posibles . . . . .	21
3.2.3. Justificación de la elección de Proxmox VE . . . . .	22
3.2.4. Configuración/Implementación . . . . .	23
3.3. NVIDIA vGPU . . . . .	25
3.3.1. Rol . . . . .	25
3.3.2. Herramientas posibles . . . . .	25
3.3.3. Justificación de la elección de NVIDIA vGPU . . . . .	26
3.3.4. Configuración/Implementación . . . . .	27
3.3.4.1. Instalación de Dependencias . . . . .	27
3.3.4.2. Instalación del Software NVIDIA . . . . .	28
3.3.4.3. Configuración de SR-IOV . . . . .	28
3.3.4.4. Deshabilitar el Controlador Nouveau . . . . .	28
3.3.4.5. Dispositivos Mediados en Proxmox . . . . .	29
3.4. Broker/Intermediario de conexiones . . . . .	29
3.4.1. Rol . . . . .	29

3.4.2. Herramientas posibles . . . . .	29
3.4.3. Justificación de la elección de UDS Enterprise . . . . .	30
3.4.4. Configuración de UDS Enterprise . . . . .	31
3.5. Automatización del Despliegue . . . . .	32
3.5.1. Rol . . . . .	32
3.5.2. Herramientas Posibles . . . . .	32
3.5.3. Justificación de la elección de Ansible . . . . .	33
3.5.4. Configuración/Implementación . . . . .	34
3.5.4.1. Rol network . . . . .	34
3.5.4.2. Rol packages . . . . .	34
3.5.4.3. Rol timezone . . . . .	35
3.5.4.4. Rol tuning . . . . .	35
<b>4. Impacto del trabajo</b>	<b>36</b>
4.1. Impacto general . . . . .	36
4.2. Objetivos de Desarrollo Sostenible . . . . .	38
<b>5. Resultados y conclusiones</b>	<b>39</b>
5.1. Resultados . . . . .	39
5.2. Trabajo futuro . . . . .	39
<b>6. Anexo</b>	<b>41</b>
<b>Bibliografía</b>	<b>42</b>

# Índice de Figuras

2.1. Modelos de servidores . . . . .	9
3.1. Arquitectura . . . . .	19
3.2. Arquitectura de un hipervisor . . . . .	21
3.3. Estructura de uso de <i>NVIDIA-GPU</i> (tomada de [41]) . . . . .	26
3.4. Estructura del bróker <i>UDS</i> (tomada de la Documentación de <i>UDS Enterprise</i> ) . . . . .	30
3.5. Arquitectura de <i>Ansible</i> . . . . .	33
4.1. Docencia e I+D+i que involucran procesamiento de imágenes en la UPM	37
5.1. Prueba de estrés para verificar funcionamiento correcto de la GPU . . .	40

# Listado de Códigos

- 3.1. Configuración de VM en *Proxmox* . . . . . 24
- 3.2. Configuración de NVIDIA-GPU . . . . . 28
- 3.3. Habilitación SR-IOV . . . . . 28

*A mi familia,  
por su infinita paciencia y constante aliento durante todos estos largos años.*

*A mis compañeros del CeSViMa,  
por confiar en mí y brindarme apoyo y orientación no solo durante el desarrollo del  
proyecto, sino también durante el tiempo que llevo formando parte de su equipo.*

*A Fernando Costoya,  
por su dedicación y consejos.*

*A mi Isabella y a mi Moka  
por su amor incondicional y por no dejar que me rinda nunca.*

# Resumen

El proyecto busca diseñar y desplegar una arquitectura de virtualización para ofrecer escritorios virtuales con tarjetas gráficas (*GPUs*) buscando una mejora del rendimiento en aplicaciones que requieran aceleración gráfica. Estos escritorios se utilizarán por un alto número de usuarios.

Se utilizará el software *Nvidia vGPU*, una solución de virtualización que permite compartir una *GPU* física entre varias máquinas virtuales, proporcionando a cada una de ellas acceso a las capacidades gráficas y de cómputo de la *GPU*. Con esto se mejora el rendimiento y la compatibilidad de las aplicaciones, al tiempo que optimiza los recursos al compartir la *GPU* entre varias máquinas virtuales (VM), lo que lo convierte en una opción escalable y de coste adecuado.

Para llevar esto a cabo, se utilizará *Proxmox* como plataforma de virtualización y *Ansible* para la automatización de la infraestructura. *Proxmox*, junto con *UDS Enterprise*, un broker/intermediario de conexiones multiplataforma, facilitará la integración de *GPUs* compartidas y la administración de escritorios virtuales. Con *Ansible*, se automatizará el despliegue y la configuración de los entornos, reduciendo el tiempo y los errores manuales en la gestión de la infraestructura.

**Palabras Clave:** Escritorios Virtuales, Virtualización, Nvidia vGPU, Proxmox, Ansible, UDS Enterprise.

# Abstract

The project aims to design and deploy a virtualization architecture to provide virtual desktops with graphics cards (*GPUs*), seeking to improve the performance of applications that require graphical acceleration. These desktops will be used on a reasonable large scale.

The software *Nvidia vGPU* will be used, a virtualization solution that enables sharing a physical *GPU* among multiple virtual machines. This provides each virtual machine access to the *GPU*'s graphical and computational capabilities, enhancing application performance and compatibility while optimizing resources by sharing the *GPU* among several virtual machines (*VMs*). This makes it a scalable and cost-effective option.

To achieve this, *Proxmox* will be used as the virtualization platform, and *Ansible* for infrastructure automation. *Proxmox*, together with *UDS Enterprise*, a cross-platform connection broker, will facilitate the integration of shared *GPUs* and the management of virtual desktops. Using *Ansible*, the deployment and configuration of environments will be automated, reducing time and minimizing manual errors in infrastructure management.

**Keywords:** Virtual Desktops, Virtualization, Nvidia vGPU, Proxmox, Ansible, UDS Enterprise.

# Capítulo 1

## Introducción

Los escritorios virtuales se han propuesto como una forma eficiente de hacer llegar de forma telemática aplicaciones y herramientas docentes y de gestión a la comunidad universitaria. La UPM es pionera (y, probablemente, una de las universidades españolas con servicios más avanzados) en su uso, que, además, se ha visto más demandado y justificado con la pandemia COVID-19 y la optimización en la eficiencia energética de las actividades universitarias.

En los últimos años, la universidad ha experimentado una creciente demanda de aplicaciones con altos requisitos de procesamiento gráfico, impulsada por programas académicos que abarcan áreas como diseño avanzado, simulación, análisis de datos y edición de video. Estos programas requieren equipos de alto rendimiento que pueden ser costosos de adquirir y mantener de manera individualizada. El diseño y la implementación de una infraestructura de escritorios virtuales con soporte de GPU representa una solución para satisfacer esta demanda, proporcionando a estudiantes y docentes acceso a aplicaciones gráficas intensivas desde dispositivos de gama media o baja, reduciendo la necesidad de inversiones en hardware específico. Al emplear escritorios virtuales, se centraliza la administración y seguridad, optimizando el uso de los recursos de GPU y permitiendo un acceso remoto flexible a entornos de trabajo.

### 1.1. Contexto del proyecto

La virtualización se ha consolidado como una herramienta clave en la optimización de infraestructuras tecnológicas, especialmente en entornos educativos que buscan maximizar la eficiencia y seguridad de sus recursos tecnológicos. Sin embargo, para aplicaciones que demandan alto rendimiento gráfico, las soluciones tradicionales de virtualización no alcanzan a cumplir con los estándares necesarios, generando la necesidad de implementar tecnologías específicas como las GPU virtualizadas. Entre las soluciones destacadas, *NVIDIA vGPU* ha emergido como un componente esencial para abordar este desafío, permitiendo que una única GPU física sea compartida por múltiples máquinas virtuales (VMs).

Para este proyecto, el diseño propuesto se implementa con *Proxmox VE* como plataforma de virtualización, *NVIDIA vGPU* para asignación gráfica, *UDS Enterprise* como bróker/intermediario de escritorios virtuales y *Ansible* para automatizar despliegues

y configuraciones. La integración de estas tecnologías, justificada convenientemente en el siguiente capítulo, no es trivial y requiere decisiones de diseño y de adaptación complejas. El resultado permitirá ofrecer una solución robusta y escalable para responder a las necesidades crecientes de rendimiento gráfico en la universidad.

### 1.2. Objetivos

El proyecto tiene como objetivo principal diseñar y proporcionar una infraestructura de escritorios virtuales con soporte para aplicaciones gráficas intensivas, que permita el acceso remoto a estudiantes y profesores en un entorno de alta disponibilidad, seguro y fácil de gestionar. Los objetivos específicos incluyen

- Diseñar la arquitectura de virtualización en *Proxmox VE* e integrar y configurar *NVIDIA vGPU* para asegurar una correcta asignación de recursos gráficos.
- Configurar *UDS Enterprise* para la gestión de escritorios virtuales, facilitando el acceso y control centralizado.
- Automatizar el despliegue y mantenimiento de los escritorios virtuales mediante *Ansible*, reduciendo el tiempo y los errores en la configuración manual.
- Optimizar la infraestructura para que cumpla con las demandas de rendimiento de aplicaciones gráficas, permitiendo la actualización y mantenimiento de los escritorios de manera centralizada.
- Asegurar la escalabilidad de la solución, de modo que pueda ampliarse para responder a futuras necesidades de estudiantes y programas académicos, incorporándola como requisito inicial.



### 1.2.1. CeSViMa

El trabajo se ha desarrollado en los últimos seis meses dentro del CeSViMa de la UPM. El Centro de Supercomputación y Visualización de Madrid, CeSViMa se creó a finales del año 2004, con el objetivo de suplir la carencia de recursos existentes para investigación. Está situado en el Parque Científico y Tecnológico de Montegancedo. Se puede recabar más información del CeSViMa en <https://www.cesvima.upm.es>.

CeSViMa tiene tres objetivos fundamentales:

- Poner a disposición de los usuarios equipos de supercomputación y visualización interactiva, ofreciendo servicios de valor añadido para facilitar la incorporación de estas tecnologías en sus campos de actividad.
- Promocionar la utilización de la computación de altas prestaciones y de técnicas de visualización avanzadas en todos los ámbitos científicos, empresariales y de la administración.
- Realizar actividades de investigación relacionadas con la generación de software especializado para la explotación de la supercomputación y visualización en diversos dominios de la ingeniería, la energía y el medioambiente.

Para alcanzar estos objetivos, el CeSViMa articula sus actuaciones en tres pilares: el almacenamiento masivo de información, computación de altas prestaciones y la visualización interactiva avanzada.

En particular, CeSViMa aloja el supercomputador Magerit-3, un clúster compuesto por 72 nodos que ofrece servicio de supercomputación a la UPM y diversas instituciones y empresas. Este sistema de cómputo científico está destinado a la ejecución de cargas de trabajo científico que requieren cómputo intensivo, pudiéndose acceder a ella desde cualquier lugar y en cualquier momento.

CeSViMa también aloja los servicios de cloud para entornos científicos. Basado en tecnología ownCloud, CeSViMa proporciona la capacidad de disponer de almacenamiento accesible desde cualquier lugar con acceso a Internet y desde cualquier dispositivo, ya sean ordenadores convencionales o dispositivos móviles (tablets o smartphones) con iOS, Android o Windows Phone.

Los grupos de investigación pueden contratar su propia nube premium privada o de gestión delegada empezando con un paquete básico que incluye 500 GB de almacenamiento sin límite de usuarios que se puede incrementar con paquetes de 500 GB.

CeSViMa también gestiona y ofrece la infraestructura para servidores virtuales. La implantación de Servidores Privados Virtuales (VPS - Virtual Private Server) en la UPM permite particionar un equipo físico informático en varios virtuales, logrando que cada uno de ellos se comporte a todos los efectos como una máquina completamente independiente, con su propio sistema operativo, ficheros, servicios, etc. De este modo, el usuario dispone de un sistema VPS en uso exclusivo, con acceso completo a todos los recursos y a su gestión. Se ofrecerán más detalles en el siguiente capítulo.

El prototipo diseñado y desarrollado en el presente trabajo supone una ampliación de los entornos virtuales ofrecidos a la comunidad universitaria por el CeSViMa.

### **1.3. Estructura del documento**

El capítulo 2 presenta diferentes aspectos del contexto del trabajo, repasando el trabajo y los desarrollos ya existentes en los Escritorios Virtuales y su utilización en el ámbito universitario. También describe las infraestructuras virtuales existentes en la UPM, cómo han evolucionado y a dónde pretenden llegar. Se analizan trabajos previos en la virtualización de GPU y su uso en instituciones educativas y de educación superior.

El capítulo 3 incluye las aportaciones principales del trabajo. Tras describir la arquitectura de la solución aportada, se entra en detalle en cada uno de sus elementos. Por cada uno (hipervisor, virtualización gráfica, bróker/intermediario y la automatización del despliegue). Por cada uno de ellos, se presenta el rol que juegan en el prototipo final, las herramientas posibles para implementarlo, se justifica la herramienta elegida y se precisa la configuración y/o implementación realizada para nuestros escritorios virtuales con GPU.

El capítulo 4 se centra en el prototipo desarrollado, las previsiones para su despliegue y los incipientes pasos para su evaluación. También se apuntan el impacto en los Objetivos de Desarrollo Sostenible de la UNESCO.

Finalmente, el capítulo 5 incluye las conclusiones del trabajo y apunta posible trabajos futuros.

## Capítulo 2

# Estado del Arte

### 2.1. Historia y evolución de la virtualización de escritorios

El término virtualización fue acuñado en la década de 1960 para referirse a una máquina virtual (a veces llamada "máquina pseudo"). En el Proyecto M44/44X de IBM, el objetivo era evaluar los conceptos emergentes del sistema de tiempo compartido. La arquitectura se basaba en máquinas virtuales: la máquina principal era una IBM 7044 (M44) y cada máquina virtual era una imagen experimental de la máquina principal (44X). Se desarrollaron varios sistemas de máquinas virtuales basados en IBM, como el CP-40, el CP-67, VM/370 y muchos más. Un componente llamado el monitor de máquinas virtuales (VMM) o hipervisor se ejecutaba directamente sobre el hardware real", y múltiples máquinas virtuales podían ser creadas a través del VMM, permitiendo que cada instancia ejecutara su propio sistema operativo.

En 1974 Popek y Goldberg [1] unificaron los criterios de diseño de los modelos de virtualización. Los conocidos como *Requisitos de Virtualización de Popek y Goldberg* definen las condiciones para que la virtualización pueda implementarse en una arquitectura lógica. Son las siguientes:

- *Equivalencia (o Fidelidad)*: Si un programa se ejecuta en el monitor de máquinas virtuales (VMM) su comportamiento debe ser predecible y esencialmente igual al mostrado cuando se ejecuta sobre el hardware directamente.
- *Control de Recursos (o Seguridad)*: En todo momento debe ser el VMM quien tenga el control total sobre los recursos hardware virtualizados en el sistema operativo invitado.
- *Eficiencia*: Debe de haber un alto número de instrucciones ejecutadas sin la intervención del VMM, esto es por el propio hardware.

Durante varias décadas, la virtualización se utilizó principalmente en entornos de *mainframe*. Sin embargo, con la disponibilidad de servidores más baratos, en la década de 1990, las organizaciones comenzaron a adquirir más servidores, lo que provocó una sobrecarga en la red eléctrica de California debido al aumento en los requisitos de refrigeración y potencia. Investigadores de la Universidad de Stanford descubrieron que este aumento en servidores generaba problemas en la infraestructura energética.

A fines de la década de 1990, Mendel Rosenblum y sus estudiantes [2] en la Universi-

## Estado del Arte

dad de Stanford revivieron el concepto de las máquinas virtuales al resolver complejos problemas técnicos que permitieron que el procesador dominante de la época, el x86 de Intel, pudiera ejecutar instancias de máquinas virtuales, a pesar de que el hardware subyacente no fuera virtualizable. Rosenblum comenzó su investigación para encontrar una mejor solución a este problema, lo que llevó a la creación del software VMware. Descubrió que consolidar varios servidores en uno más grande reducían los requisitos de energía y refrigeración.

La virtualización tiene dos componentes [3]: la virtualización de servidores, que permite consolidar múltiples servidores en un solo servidor, y la virtualización de escritorios, que reduce el tamaño y el soporte del sistema de escritorio.

Una computadora virtual es un escritorio representado solo por software [4, 5]. La virtualización ofrece mayor flexibilidad y aumenta la tasa de utilización del hardware físico subyacente. Las computadoras tradicionales proporcionan una instancia del sistema operativo que aloja varios programas de aplicación. En un entorno virtualizado, se utiliza una sola computadora física (servidor) que ejecuta software para extraer los recursos de la computadora física y compartirlos entre diferentes máquinas virtuales. Aunque se ejecutan varias computadoras virtuales desde un único servidor, cada máquina virtual puede ejecutar un sistema operativo distinto. Si una máquina virtual o un programa específico se bloquea, no afectará a las demás máquinas virtuales que se ejecutan en el mismo servidor [6, 7].

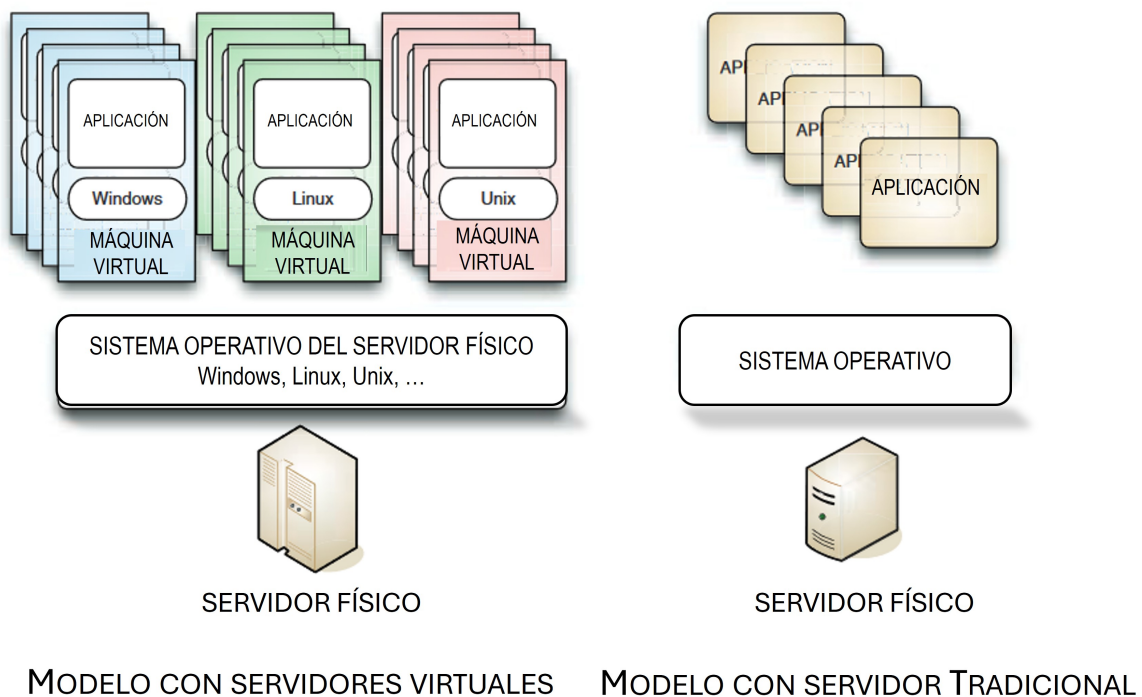


Figura 2.1: Modelos de servidores

La virtualización de escritorios [8] surgió como una solución para simplificar la administración y reducir los costos de hardware en grandes corporaciones, permitiendo la consolidación de múltiples entornos de usuario en servidores centralizados. En sus

primeros años, esta tecnología se limitaba principalmente a entornos empresariales y centros de datos, donde los recursos gráficos no eran críticos.

Sin embargo, a medida que ha aumentado la demanda de aplicaciones gráficas avanzadas, la virtualización de escritorios ha evolucionado, expandiéndose a sectores educativos, de diseño y científicos [9]. Estos sectores han encontrado en la virtualización una forma flexible y económica [10] de acceder a entornos de alto rendimiento, eliminando la necesidad de contar con equipos físicos costosos en cada punto de acceso.

## 2.2. Virtualización en las universidades

La virtualización permite una experiencia informática uniforme para los estudiantes, independientemente de la antigüedad de los dispositivos. Aunque cada computadora pueda tener especificaciones diferentes, la experiencia es homogénea, ya que el entorno de escritorio se encuentra en un servidor virtualizado y no en las computadoras individuales.

Las universidades han comprendido que la virtualización, aplicada tanto en escritorios como en servidores, ofrece una serie de beneficios significativos. Entre ellos se encuentran una mayor eficiencia en la gestión de TI, ahorro de costes y mejoras funcionales. .

### 2.2.1. Beneficios administrativos

- **Reducción de costos:** La virtualización permite reducir el costo total de la computación de escritorio hasta un 50%<sup>1</sup>. Además, la virtualización de escritorios genera ahorros adicionales en licencias de software y costos energéticos [11], permitiendo a las universidades ahorrar entre 300 y 600 euros por puesto eliminado [12].
- **Disminución del consumo energético:** Las organizaciones, incluidas las universidades, utilizan la virtualización para reducir el consumo de energía, las necesidades de refrigeración y el espacio necesario para los centros de procesamiento de datos. La virtualización puede reducir el consumo de energía hasta en un 80% [10, 13]. Además, al haber menos hardware, se disminuyen los costos de refrigeración [14, 15].
- **Homogeneización de la experiencia:** La virtualización proporciona una experiencia uniforme a los estudiantes sin importar la antigüedad de las computadoras. Los estudiantes pueden acceder a su entorno de escritorio desde cualquier dispositivo, incluso sus propios dispositivos móviles, ya que estos solo necesitan funcionar como clientes ligeros. La virtualización también ofrece mayor seguridad, permitiendo que los dispositivos personales se conecten a la red universitaria sin comprometer la seguridad [12].
- **Recuperación ante desastres:** La virtualización facilita la recuperación rápida ante fallos del servidor. Los archivos de servidores virtualizados pueden copiarse fácilmente a otros servidores o computadoras, reduciendo el tiempo de recupe-

---

<sup>1</sup>IDC Research, (2009). "Quantifying the business Value of VMware View", <https://www.yumpu.com/en/document/view/47985073/white-paper-quantifying-the-business-value-of-vmware-view-idc->

ración a minutos en lugar de horas. Esto es crucial en la protección de los datos y aplicaciones de los usuarios.

- **Despliegue de aplicaciones y seguridad de datos:** La implementación de nuevas aplicaciones en un entorno virtualizado es mucho más rápida, reduciendo el tiempo de despliegue de horas a minutos<sup>2</sup>. Además, la centralización de los datos en centros de datos permite mayor seguridad, protegiendo la información sensible como los datos estudiantiles y financieros contra amenazas externas [12].
- **Ahorros en renovación de hardware:** La virtualización permite extender la vida útil de equipos antiguos, que pueden seguir siendo utilizados mediante entornos virtualizados. Esto reduce la necesidad de comprar nuevos equipos costosos, permitiendo a las universidades ahorrar en costos de renovación [12].

### 2.2.2. Beneficios organizacionales y para los estudiantes

Desde el punto de vista organizacional y de los estudiantes, la virtualización ofrece ventajas adicionales [16]:

- **Acceso a recursos educativos en cualquier lugar y dispositivo:** Los estudiantes pueden acceder a aplicaciones y software avanzados desde cualquier dispositivo, incluyendo *Chromebook* y tabletas, sin importar si están en el campus o fuera de él.
- **Virtualización de aulas y laboratorios:** La virtualización permite centralizar las aplicaciones en el centro de datos, facilitando la administración de recursos y proporcionando visibilidad de toda la infraestructura tecnológica. Así se optimiza el uso de los recursos del centro de datos y mejora el espacio disponible en el campus al reducir la cantidad de hardware físico necesario.
- **Expansión de programas en línea y a distancia:** Las universidades pueden ofrecer programas híbridos y en línea a más estudiantes al utilizar infraestructuras virtualizadas. Esto aumenta el acceso a la educación y mejora la capacidad de las universidades para escalar sus servicios sin comprometer el rendimiento [16].

### 2.2.3. Beneficios educativos

Diversos estudios [17, 18, 19] han señalado que la virtualización también ofrece ventajas educativas:

- **Educación tecnológica uniforme:** La virtualización permite que todos los estudiantes trabajen en entornos de escritorio idénticos, eliminando problemas relacionados con las configuraciones individuales de cada computadora. Esto facilita que los estudiantes se concentren en los objetivos de aprendizaje sin distracciones tecnológicas, mejorando la autoeficacia de los estudiantes, especialmente cuando trabajan de forma remota [18].

---

<sup>2</sup>VMware View 4.5 Modernize Desktop and Application Management, <https://www.yumpu.com/en/document/view/29362561/download-vmware-view-brochure>

- **Eliminación de problemas en laboratorios virtuales:** La virtualización de laboratorios elimina muchos de los problemas que surgen al usar diferentes estaciones de trabajo, facilitando una experiencia educativa más fluida.

Sin embargo, algunos estudios [20] sugieren que estos laboratorios virtuales, aunque útiles a nivel individual, llevados al extremo pueden no fomentar la colaboración entre los estudiantes, una de las competencias transversales más demandadas y esenciales.

En la literatura pueden encontrarse diversos trabajos que muestran experiencias de éxito en su uso educativo [21, 18, 22]

### 2.2.4. Tendencias actuales y futuras

La virtualización de escritorios se está adaptando a nuevas tendencias y tecnologías emergentes, como la adopción de soluciones de *Desktop as a Service* (DaaS).

En el ámbito educativo, la virtualización de escritorios con soporte de GPU permite a las instituciones proporcionar acceso remoto a aplicaciones avanzadas de forma escalable y eficiente. Adicionalmente, las arquitecturas de infraestructura hiperconvergente han facilitado la expansión de la virtualización de escritorios, permitiendo que los recursos de GPU se escalen conforme a las demandas académicas, brindando flexibilidad y reduciendo costos a largo plazo.

## 2.3. Escritorios virtuales en la UPM

La Universidad Politécnica de Madrid (UPM) es la universidad politécnica más grande de España y siempre ha procurado estar a la vanguardia de la tecnología en el sistema universitario español. Dispone de dos *Data Centers* (Tier II+) ubicados en Campus de Montegancedo y Campus de Ciudad Universitaria (Rectorado) que alojan todos los servicios TI que la Universidad proporciona a todos sus miembros. En estos DC se aloja toda la infraestructura de servidores, almacenamiento y comunicaciones que soporta los servicios, entre los que se encuentran desde la solución de *Disaster Recovery (active – active)* para los servicios críticos de la UPM, como puede ser *UPMdrive*, hasta el supercomputador Magerit.

Por contrapartida, la UPM presenta una marcada dispersión geográfica, contando con 17 centros educativos dispersos en cuatro campus universitarios, lo que a efectos de soporte informático implica dificultades a la hora de optimizar los recursos técnicos y humanos existentes, pues, además del equipamiento físico, mucho del trabajo se repite en todos los centros con el fin de mantener las mismas aplicaciones.

Por todo ello, se planteó desde hace tiempo el objetivo de proporcionar soporte a más de 40.000 usuarios accediendo con sus propios dispositivos, dando cabida a lo que se denomina *BYOD (Bring Your Own Device)*. Actualmente, la UPM tiene ya una larga experiencia en la utilización de escritorios virtuales en sus actividades docentes y organizativas.

La UPM cuenta con un servicio de escritorios virtuales que permite a estudiantes y personal acceder casi desde cualquier dispositivo a múltiples aplicaciones informáticas para hacer uso de ellas sin necesidad de instalarlas. Entre los programas disponibles figuran los que precisa el alumnado para realizar sus prácticas. El servicio,

denominado *escritorioUPM*<sup>3</sup>, no requiere que el usuario tenga una conexión a internet con mucho ancho de banda ni un dispositivo muy potente, ya sea un ordenador, un móvil o una tableta.

Aunque este servicio estaba previsto con anterioridad, la situación generada por la pandemia COVID-19 aceleró su implantación.

La UPM cuenta con cerca de 39.000 alumnos matriculados, tanto de Grado, Máster y Doctorado, en titulaciones de perfil muy tecnológico y que requieren disponer para la docencia de multitud de herramientas software, generalmente muy pesadas y costosas. Los escritorios virtuales han permitido que ya no sea necesario una actividad presencial en un aula informática de alguna de las escuelas de la UPM y que se extienda a un uso intensivo 7×24.

De todo el personal de la UPM, cerca de 1.000 personas necesitan acceder a servicios de carácter administrativo, ya sea de perfil académico, económico, personal, etc. Una solución basada en escritorios virtuales ha incrementado la seguridad en los procedimientos de la universidad acercándolo a los objetivos del ENS y al mismo tiempo dar solución a un nuevo modelo de trabajo, parcialmente no presencial, permitiendo acceder de manera segura y energéticamente eficiente a este tipo de aplicaciones desde cualquier lugar y dispositivo.

Como ya se ha indicado, la infraestructura de escritorios o servidores virtuales de la UPM se gestiona desde el CeSViMa. Es posible solicitar servidores virtuales privados (SVP) destinados a albergar servicios en red, basados en arquitectura *Intel Xeon Gold 6240R* a 2.40 GHz, con hasta 768 GB de RAM y con el espacio de almacenamiento que se necesite.

El servicio VPS se proporciona de manera delegada, donde el usuario tiene acceso completo (aunque puede existir alguna limitación) como administrador de su equipo. CeSViMa puede custodiar, si así se acuerda, una cuenta de administrador destinada a proporcionar y garantizar el servicio o para recuperación frente a desastres. En todo momento es de aplicación la normativa UPM respecto a recursos informáticos y red de datos de la Universidad.

Los grupos de investigación pueden utilizar servidores VPS adaptado a las necesidades del grupo configurando a la carta el número de vCPUs, memoria RAM, capacidad del disco duro, backup de datos y sistema operativo. El servidor virtual básico proporciona 2 vCPU (*Intel Xeon*), 4 GB de memoria RAM, 100 GB de disco duro y sistema operativo Linux Debian.

## 2.4. GPU en la Virtualización de Escritorios

Con el aumento de la demanda de aplicaciones gráficas avanzadas, la incorporación de GPU en entornos de virtualización se ha convertido en una solución esencial para satisfacer estas necesidades. Inicialmente, la virtualización de GPU permitía la asignación directa de una GPU a una única máquina virtual. Sin embargo, con el tiempo, se han desarrollado tecnologías que permiten particionar una GPU entre varias máquinas virtuales, como es el caso de *NVIDIA vGPU*, optimizando el uso de recursos

---

<sup>3</sup><https://docs.cesvima.upm.es/escritorioupm/>

y reduciendo costos. Esta capacidad de compartir GPUs ha transformado la virtualización de escritorios, permitiendo que aplicaciones de diseño, simulación, procesamiento 3D y análisis de datos funcionen de manera fluida sin requerir hardware dedicado en cada terminal de usuario.

Un componente clave en la mejora del rendimiento de los escritorios virtuales es la aceleración mediante GPU. Una unidad de procesamiento gráfico (GPU) está específicamente diseñada para manejar tareas complejas de procesamiento gráfico y visual de manera eficiente. Tradicionalmente, las GPUs han estado asociadas a videojuegos y aplicaciones intensivas en gráficos, pero su importancia en los entornos de escritorios virtualizados no puede ser subestimada.

La aceleración mediante GPU mejora significativamente la experiencia del usuario al descargar tareas intensivas en gráficos del CPU hacia la GPU, lo que resulta en un rendimiento más fluido y ágil. A continuación, se detallan las razones por las cuales la aceleración mediante GPU es esencial para el rendimiento de los escritorios virtuales:

- **Incremento de la productividad:** Muchas aplicaciones profesionales dependen en gran medida del poder de procesamiento de la GPU, como el software CAD/CAM, herramientas de modelado 3D y visualización de datos complejos. Al aprovechar las capacidades de la GPU, los escritorios virtuales pueden manejar estas tareas intensivas de manera más eficiente, lo que no solo aumenta la productividad, sino que también permite trabajar con grandes conjuntos de datos, simulaciones complejas y renderizados detallados sin experimentar cuellos de botella en el rendimiento.
- **Mejora de la respuesta del usuario:** Las aplicaciones que demandan gráficos suelen sobrecargar la CPU, lo que resulta en tiempos de respuesta más lentos y menor interactividad. Al descargar estas tareas exigentes hacia la GPU, los escritorios virtuales pueden mantener una respuesta fluida incluso durante picos de uso. Esto asegura que los usuarios interactúen con sus escritorios en tiempo real, sin retrasos, mejorando el flujo de trabajo y la experiencia general.
- **Soporte multitarea sin interrupciones:** En un espacio de trabajo moderno, muchos profesionales requieren múltiples pantallas para mejorar su productividad. Las GPUs equipadas con soporte para múltiples monitores permiten a los escritorios virtuales manejar sin esfuerzo la visualización de contenido en varias pantallas. La aceleración mediante GPU optimiza el renderizado de estas pantallas, eliminando parpadeos, latencia o problemas de sincronización, ofreciendo una experiencia fluida al trabajar con múltiples monitores.

Las ventajas de la virtualización y los entornos virtuales se amplifican con el uso de GPUs. Uno de los beneficios principales de usar escritorios virtualizados con GPU es el ahorro de costes. Los profesionales ya no necesitan invertir en hardware físico de alto rendimiento para manejar tareas que requieren un potente rendimiento gráfico.

En el caso de las universidades, esta cuestión toma especial relevancia pues en muchos campos científicos se trabaja con tareas intensivas en gráficos (simulación y predicción, diseño y *autoCAD*, procesamiento de imágenes científicas, representación gráfica en 3D, procesamiento y edición de vídeos, inteligencia artificial y aprendizaje automático, etc.)

La UPM, centrada en la ingeniería y la arquitectura, realiza I+D+i en muchos de estos campos con muchos grupos de investigación, así como asignaturas y docencia enfocada a estos aspectos.

En la literatura pueden encontrarse relativamente pocos trabajos científicos sobre la integración de GPU con escritorios virtuales. Entre los que se basan en *VMware* podemos citar [23, 24]. Otros utilizan *OpenStack* [25]. como [26, 27, 28], de manera que sus soluciones están basadas en código de fuente abierta. Finalmente [29] utiliza *NARI Desktop Cloud*.

Existen varias soluciones comerciales (*Azure Virtual Desktop*<sup>4</sup>, *Supermicro*<sup>5</sup>, *Nebul*<sup>6</sup>, etc.) Todas ellas parecen opciones sólidas, pero, obviamente, tienen una alta inversión en su desarrollo y mantenimiento.

Dentro de su integración en el mundo universitario, hay pocas experiencias. Las universidades americanas de South California, Iowa State, Florida Atlantic, Old Dominion y Michigan State University[30] ofrecen a sus estudiantes (en algunos casos con coste adicional) escritorios virtuales con GPU desarrolladas por VMware<sup>7</sup>. Udacity ha presentado recientemente sus escritorios virtuales con GPU (desarrollados por *Cloudlabs Inc.*<sup>8</sup>

Las universidades de Sacramento State<sup>9</sup> y Appalachian State<sup>10</sup> ofrecen servicios similares basados en *Microsoft Azure*.

[31] describe un *Private Cloud Desktop* que se fusiona con la plataforma de enseñanza y práctica de la Universidad de Tecnología de Zhejiang en China.

La universidad de Colorado ofrece una solución basada en *NVIDIA*, como aparece entre los ejemplos de [16].

En nuestra búsqueda no hemos podido encontrar universidades que ofrezcan a sus estudiantes soluciones para uso de GPU con escritorios virtuales basadas en Software libre desarrolladas dentro de las unidades de tecnologías de la información de la propia institución. Compañeros del CeSViMa conocen experiencias en la Universidad de Murcia (en su servicio de escritorios virtuales EVA) y en la Universidad de Sevilla (dentro de sus escritorios virtuales OpenLabs), aunque no ha sido posible localizar nada en sus webs. Posiblemente alguna universidad adicional haya trabajado en su desarrollo, pero hemos encontrado muy pocas pruebas documentales de ello.

Esto convierte a la UPM, gracias al trabajo aquí presentado, en una de las pioneras en esta tecnología.

---

<sup>4</sup><https://azure.microsoft.com/es-es/products/virtual-desktop>

<sup>5</sup><https://www.supermicro.com/en/accelerators/nvidia/vgpu>

<sup>6</sup><https://nebul.com/powerdesk/>

<sup>7</sup><https://blogs.vmware.com/euc/2016/03/higher-education-3d-software-virtualization-vmware-horizon-nvidia-grid.html>

<sup>8</sup><https://engineering.udacity.com/creating-a-gpu-enhanced-virtual-desktop-for-udacity-497bdd91a505>

<sup>9</sup><https://www.csus.edu/college/engineering-computer-science/computing-services/azure-virtual-desktop.html>

<sup>10</sup><https://confluence.appstate.edu/display/ATKB/Access+Online+Windows+Computer+Lab+with+GPU+from+a+Web+Browser>

### 2.5. Elección de herramientas

En la investigación científica y en la computación aplicada a la ingeniería, no existe una solución única, debido a que las necesidades de los investigadores varían considerablemente según el dominio del problema y el conocimiento y las preferencias del investigador. Por lo tanto, es importante hacer un estudio de las necesidades de los investigadores en el ámbito académico para identificar los requisitos específicos en términos de poder computacional, almacenamiento, programas requeridos y configuraciones del sistema. Una vez que esta información se haya determinado, sería posible ejecutar `scripts` predefinidos, construir la infraestructura requerida e instalar y configurar automáticamente el software necesario. Esto ofrecería una solución completa para los investigadores, reduciendo significativamente el esfuerzo necesario para implementar su nube personalizada.

En la literatura, se pueden encontrar trabajos [32, 33] que han estudiado las ventajas, inconvenientes, beneficios y limitaciones de diversas herramientas comerciales. El capítulo 3 analiza estos artículos en relación con cada una de las necesidades tecnológicas necesarias.

La elección en cada caso concreto se ha basado en tres principios fundamentales:

- Compatibilidad con los servicios previamente ofrecidos en la UPM.
- Utilización de Software Libre, por sus ventajas en cuanto a filosofía, evolución, mantenimiento y coste.
- Eficiencia suficiente para satisfacer las necesidades de la UPM.

#### 2.5.1. El papel del software libre y de código abierto

Los beneficios del software libre combinados con la virtualización forman una combinación destacada.

- *Rentabilidad*: No requiere costosas licencias, lo que lo convierte en una opción ideal para usuarios individuales, pequeñas empresas y organizaciones con presupuestos limitados.
- *Transparencia y Seguridad*: Acceso y modificación del código fuente para mejorar la seguridad y la confianza.
- *Flexibilidad y Personalización*: Personalización de las soluciones para necesidades específicas, permitiendo la integración con otras herramientas y flujos de trabajo de código abierto. Esta flexibilidad fomenta la innovación, ya que las organizaciones pueden experimentar con las últimas tecnologías sin verse restringidas por las limitaciones de los proveedores. Permite una respuesta más ágil a los cambios en los flujos de trabajo o demandas empresariales, lo que es esencial para mantener la competitividad.
- *Comunidad activa*: Beneficio de una gran comunidad de desarrolladores que proporciona soporte, actualizaciones e innovación de manera continua. Este esfuerzo colaborativo mejora la calidad del software y asegura una rápida respuesta a las vulnerabilidades. Además, los usuarios pueden contribuir compartiendo sus historias de éxito o mejoras del código, creando un ciclo de conocimiento y mejora difícil de replicar por las opciones propietarias.

- *Ecosistema diverso*: Elección de herramientas de código abierto según los requisitos específicos.

Al evaluar soluciones de escritorios virtuales de código abierto, se deben considerar métricas de rendimiento como velocidad y eficiencia. Soluciones como *Proxmox VE* aprovechan tecnologías robustas de virtualización para ofrecer tiempos de respuesta consistentes. La velocidad puede verse influenciada por factores como la latencia de red y el hardware subyacente, lo que hace esencial optimizar las configuraciones para obtener los mejores resultados de rendimiento.

Los escritorios virtuales de código abierto ofrecen diversas oportunidades para organizaciones que buscan opciones rentables, flexibles y fáciles de gestionar. Entender sus características y rendimiento es clave para tomar decisiones informadas al adoptar estas tecnologías.

Al considerar soluciones para escritorios virtuales, es esencial distinguir entre opciones propietarias y de código abierto. Las soluciones propietarias, como *VMware Horizon*, *Microsoft Azure* [34] o *Citrix Virtual Apps*, requieren un costo de licencia y soporte específico del proveedor. Estas soluciones pueden ser extremadamente efectivas, pero pueden generar dependencia de ecosistemas específicos y restringir la personalización.

Por otro lado, la opción de código abierto ofrece flexibilidad y soporte impulsado por la comunidad. Aunque las soluciones de software libre puedan carecer del soporte formal de sus contrapartes propietarias, fomentan la innovación y colaboración entre los usuarios.

## Capítulo 3

# Desarrollo

### 3.1. Diseño de la arquitectura del sistema

El diseño de la infraestructura se ha concebido con un enfoque especial en la escalabilidad. Dado que se trata de un prototipo destinado a evaluar la viabilidad de una solución de escritorios virtuales con GPUs, la idea es que, en el futuro, sea posible añadir más nodos para formar un clúster y ofrecer un servicio que satisfaga las necesidades de la UPM. Es importante que el supervisor (*Proxmox VE* en este caso) permita escalar horizontalmente, lo que significa que la ampliación de la infraestructura se logra mediante la incorporación de nuevos nodos al clúster existente. Este modelo de escalabilidad favorece una expansión progresiva, permitiendo distribuir las cargas de trabajo de manera eficiente y mejorar el rendimiento del sistema a medida que crece la demanda.

Además, es fundamental que el hipervisor proporcione una administración centralizada a través de su interfaz gráfica, lo que facilita la integración de recursos adicionales (y es el caso de *Proxmox VE*). En esta línea, se ha optado por implementar un sistema de almacenamiento centralizado basado en *NFS*, que simplifica la distribución de datos entre los nodos del clúster y contribuye a mejorar la escalabilidad general del sistema. Esta configuración asegura un crecimiento sostenible, tanto en términos de capacidad como de rendimiento, y garantiza la flexibilidad necesaria para adaptarse a las demandas futuras.

El bróker debe estar preparado para soportar futuras expansiones de la infraestructura, adaptándose fácilmente al crecimiento en el número de usuarios y servicios requeridos, lo que cumple *UDS Enterprise*. Para facilitar la distribución de las conexiones y optimizar el acceso a los servicios, se implementó un balanceador de carga *HAProxy*. Este sistema, configurado en modo activo/pasivo, gestiona las conexiones tanto de los usuarios como de los administradores al portal de inicio de sesión de *UDS*, así como a los servicios asociados.

La orquestación debe facilitar y potenciar la escalabilidad en una infraestructura virtualizada. Gracias a la automatización de la configuración de los nodos, la integración de estos se vuelve mucho más sencilla y rápida, permitiendo poner en producción nuevos nodos en menor tiempo y con menos intervención manual. Esto no solo reduce el riesgo de errores humanos, sino que también optimiza la eficiencia operativa al permitir un despliegue uniforme y reproducible. Todo ello lo cumple la herramienta

elegida *Ansible*.

Además, *Proxmox VE* ha introducido recientemente una opción de instalación automatizada que simplifica aún más este proceso. Este método permite realizar la instalación de *Proxmox* de forma desatendida, automatizando completamente la configuración inicial sobre hardware *bare-metal*.

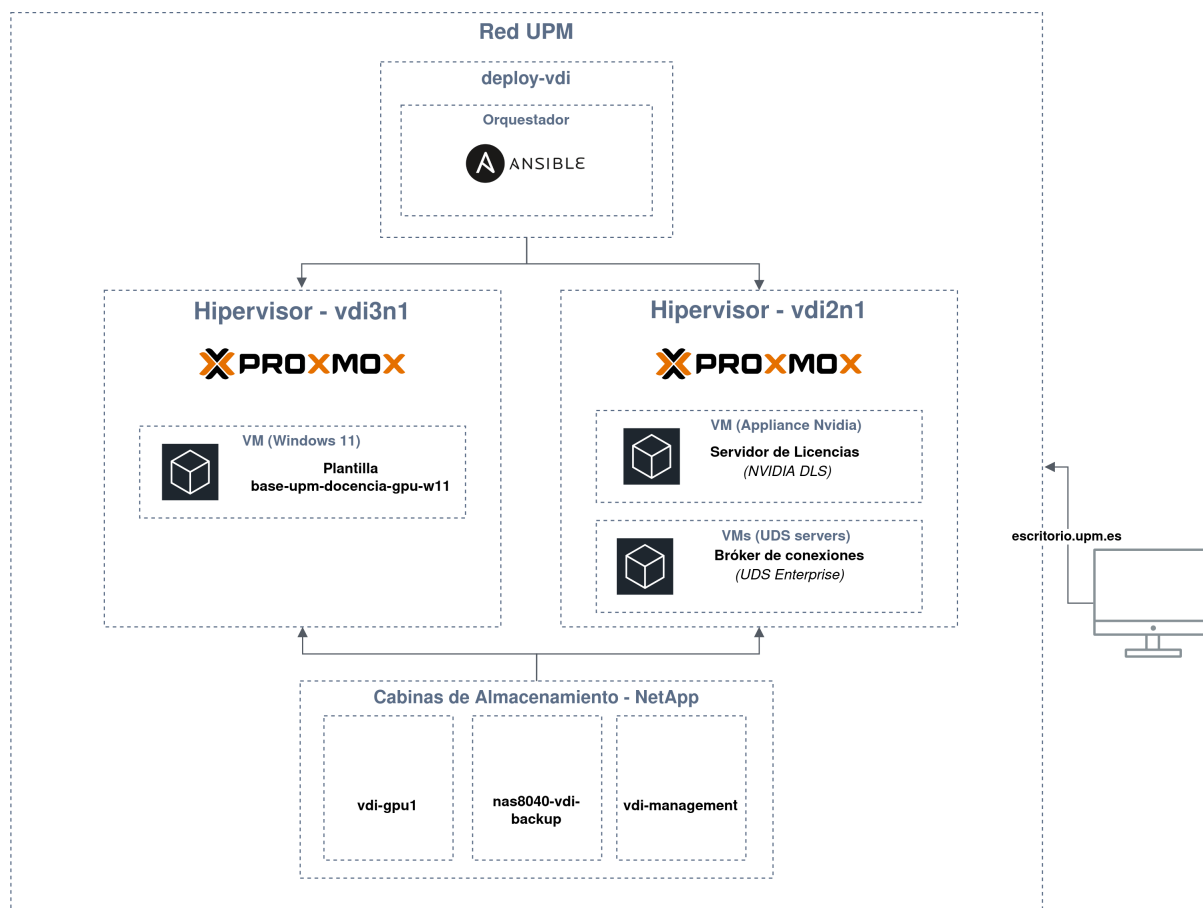


Figura 3.1: Arquitectura

### 3.1.1. Flujo de operación

Los usuarios finales acceden a la infraestructura a través del portal `escritorio.upm.es`. El bróker (servidor UDS) gestiona las solicitudes de conexión y asigna escritorios remotos a los usuarios. Los escritorios remotos son instanciados desde la máquina virtual base. Se dispone de un almacenamiento NFS compartido proporcionado por un clúster de almacenamiento de NetApp. Las tareas de mantenimiento y despliegue de configuración son automatizadas con *Ansible* desde el servidor de despliegue específico para el VDI.

### 3.2. Instalación y configuración del hipervisor

#### 3.2.1. Rol

El hipervisor es uno de los componentes esenciales en un entorno virtualizado. Este software permite alojar múltiples máquinas virtuales en una única máquina física. Existen dos tipos principales de hipervisores: los de tipo 1, también llamados nativos o *bare metal*, que operan directamente sobre el hardware físico del sistema anfitrión, y los de tipo 2, conocidos como *alojados*, que funcionan como una aplicación instalada sobre el sistema operativo anfitrión. Los hipervisores de tipo 1 destacan por su alto rendimiento, eficiencia y aislamiento, ya que acceden directamente a los recursos del hardware. Esto los hace ideales para centros de datos y entornos empresariales con altas exigencias de computación. Por otro lado, los hipervisores de tipo 2 son más fáciles de implementar y administrar, lo que los convierte en una opción adecuada para entornos de desarrollo y uso personal, donde las cargas de trabajo son menos intensivas. En términos generales, el hipervisor abstrae los recursos físicos del sistema anfitrión, dividiéndolos en máquinas virtuales independientes. Esto permite que diferentes sistemas operativos, aplicaciones y configuraciones coexistan en una misma máquina física, garantizando un nivel de aislamiento entre las máquinas virtuales. Además, los hipervisores proporcionan una interfaz uniforme para el hardware, facilitando la interacción entre los sistemas invitados y los recursos físicos subyacentes.

La virtualización basada en hipervisores permite emular completamente otra computadora, lo que hace posible emular otros tipos de dispositivos (por ejemplo, un teléfono móvil), otras arquitecturas de CPU o sistemas operativos diferentes. Esto es útil, por ejemplo, cuando se desarrollan aplicaciones para plataformas móviles: el desarrollador puede probar su aplicación en su sistema de desarrollo sin la necesidad de tener físicamente acceso a un dispositivo concreto. Otro caso común de uso es tener máquinas virtuales con sistemas operativos invitados distintos al sistema operativo anfitrión. Algunos usuarios necesitan software especial que no funciona en su sistema operativo preferido, y la virtualización permite ejecutar casi cualquier entorno y software requerido de manera independiente del sistema anfitrión. El hipervisor permite la ejecución simultánea de instancias completas de sistemas operativos al mismo tiempo. Por ejemplo, un sistema basado en hipervisor puede gestionar simultáneamente un escritorio con Windows y un servidor Linux, y proporcionar instancias de estas máquinas diversas a los usuarios bajo demanda.

Debido a la abstracción del hardware, es más fácil crear, desplegar y mantener imágenes del sistema. En caso de incidentes de hardware, una máquina virtual puede ser trasladada a otro sistema con muy poco esfuerzo y, en el mejor de los casos, incluso sin que el usuario se dé cuenta de que ha ocurrido una migración. La virtualización basada en hipervisores aprovecha las capacidades modernas de los procesadores; Esto permite que la máquina virtual y sus aplicaciones accedan directamente a la CPU en modo no privilegiado, lo que produce mejoras de rendimiento sin sacrificar la seguridad del sistema anfitrión.

La figura 3.2 muestra la arquitectura general de un sistema virtual basado en hipervisor que se ejecuta en una máquina física.

Los hipervisores permiten que numerosos sistemas operativos sigan funcionando en un solo servidor físico. El modelo de computación en la nube es más asequible porque

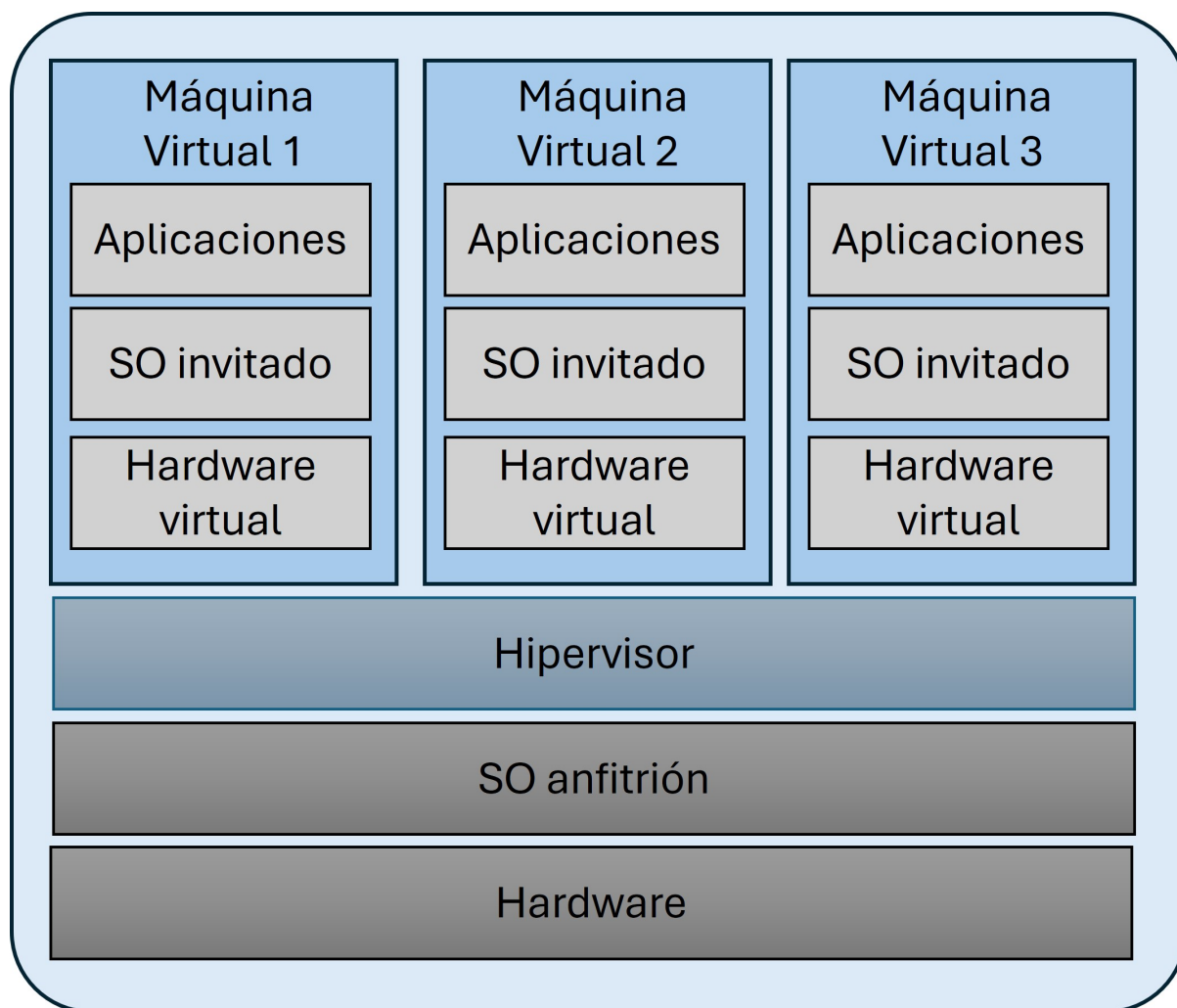


Figura 3.2: Arquitectura de un hipervisor

agiliza la entrega de servicios, proporcionando una plataforma para optimizar los recursos informáticos sofisticados de manera flexible.

### 3.2.2. Herramientas posibles

Entre las plataformas de gestión de la nube (CMP, por sus siglas en inglés) de uso más común pueden mencionarse: *Eucalyptus*, *CloudStack*, *OpenStack* [35], *Nimbus*, *Proxmox* y *Amazon Web Services*.

Todas ellas pueden utilizarse para desplegar y gestionar instancias de máquinas virtuales tanto en recursos físicos locales como en nubes públicas externas. Esto permite la configuración y administración descentralizadas de los recursos de infraestructura en la nube independientemente de la capa de virtualización subyacente (hipervisores como *Xen* o *VMware*).

Algunas de ellas son software propietario, mientras que otras son software de código abierto [36] (o software libre, en lo que sigue estos dos términos significarán lo mismo para este trabajo).

### 3.2.3. Justificación de la elección de Proxmox VE

*Proxmox Virtual Environment* (PVE) [37, 38, 39] es un entorno de virtualización maduro, completo. Es una herramienta de código abierto basada en la distribución Debian GNU/Linux que gestiona contenedores, máquinas virtuales, almacenamiento, redes virtualizadas y clústeres de alta disponibilidad a través de una interfaz web bien diseñada o mediante la interfaz de línea de comandos. Los desarrolladores proporcionaron la primera versión estable de *Proxmox VE* en 2008.

*Proxmox VE* se puede usar para gestionar máquinas virtuales y clústeres utilizando gestión de usuarios y permisos basada en roles para sus objetos. Debido a que Proxmox se basa en Debian, solo puede instalarse en servidores con procesadores de 64 bits. *Proxmox* ofrece virtualización completa (la capacidad de ejecutar cualquier sistema operativo) o paravirtualización (requiere que el sistema operativo invitado sea modificado para ejecutarse sobre el hipervisor).

*Proxmox* tiene la capacidad de migrar máquinas virtuales en caliente a otro servidor *Proxmox* para crear un entorno de alta disponibilidad (HA), donde todas las máquinas virtuales se migran automáticamente a otro nodo de servidor operativo si el servidor principal de *Proxmox* falla.

Un clúster de *Proxmox VE* tiene una característica única en la que no necesita un nodo de gestión. Esto elimina un único punto de fallo, ya que todos los nodos tienen la capacidad de realizar actividades de gestión. *Proxmox* utiliza un sistema de archivos de clúster basado en base de datos para almacenar archivos de configuración, replicados en tiempo real en todos los nodos virtuales.

Otras características son:

- La licencia para *Proxmox VE* es deliberadamente la Licencia Pública General Afiero GNU (V3)<sup>1</sup>. De entre las muchas licencias libres y de código abierto compatibles disponibles, esta es una elección significativa porque está específicamente diseñada para garantizar la cooperación con la comunidad en el caso de software de servidores de red.
- *Proxmox VE* se administra principalmente desde una interfaz web integrada, desde la línea de comandos localmente o a través de SSH. En consecuencia, no es necesario un servidor de gestión separado ni el gasto asociado. De este modo, *Proxmox VE* se diferencia significativamente de las soluciones de virtualización empresarial de proveedores como VMware o Xen.
- Las instancias/nodos de *Proxmox VE* pueden incorporarse a clústeres de PVE y administrarse centralmente desde una interfaz web unificada.
- *Proxmox VE* proporciona migración en vivo: el movimiento de una máquina virtual o contenedor de un nodo de clúster a otro sin interrumpir los servicios.

*Proxmox* usa internamente *KVM* para virtualización completa y *LXC* para contenedores Linux. *KVM* (*Kernel-based Virtual Machine*) es una tecnología de virtualización de código abierto integrada en Linux. Permite gestionar y ejecutar varias máquinas virtuales (VM) de manera simultánea y aislada sobre un único servidor físico. Cada VM opera como un entorno independiente, con su propio sistema operativo y

<sup>1</sup><https://www.gnu.org/licenses/agpl-3.0.html>

recursos virtualizados, como CPU, memoria, almacenamiento y red. También se puede agrupar usando múltiples anfitriones físicos para utilizar características como la migración en vivo de máquinas virtuales invitadas durante el mantenimiento y alta disponibilidad. Utiliza *pmxcfs*, un sistema de ficheros basado en bases de datos que almacena los ficheros de configuración. Están replicados en tiempo real en todos los nodos del clúster mediante *corosync*.

En [40] se analiza la eficiencia de diferentes herramientas para usarse como hipervisor, concluyendo que la eficiencia de *Proxmox* es suficiente y adecuada, especialmente en cuanto al rendimiento de la CPU.

### 3.2.4. Configuración/Implementación

La instalación de *Proxmox VE* se realizó a través de la interfaz gráfica, por ser el proceso más sencillo y rápido. Para ello, se debe iniciar el servidor desde el medio de instalación preparado (un USB) y seleccionar la opción `Install Proxmox VE (Graphical)` en el menú de arranque. El asistente gráfico guía al usuario a través de los pasos necesarios, que incluyen la selección del disco de destino para la instalación, la configuración básica del sistema (ubicación, zona horaria, teclado, contraseña de administrador y correo electrónico) y la configuración de red inicial. En esta pantalla, después de seleccionar la interfaz de red, se deben introducir el nombre del host, la dirección IP del servidor, el Gateway y la dirección del servidor DNS. Tras revisar un resumen de las configuraciones seleccionadas, el instalador comenzará el proceso de instalación. Una vez finalizado, se reinicia el sistema y *Proxmox* estará listo para ser accedido a través de su interfaz web.

Una vez dentro de la interfaz web de *Proxmox*, el siguiente paso es añadir la licencia de *Proxmox Enterprise*. Esto permite acceder a los repositorios *Enterprise*, que ofrecen actualizaciones estables, probadas y optimizadas para entornos de producción. Para añadir la licencia, se debe navegar a la sección `Datacenter > Subscription`, introducir la clave correspondiente y hacer clic en `Register`. Así se habilita automáticamente el acceso a los repositorios *Enterprise*.

Con la licencia activada y el acceso mediante `ssh` habilitado, el siguiente paso es ejecutar el *playbook* de *Ansible*. El *playbook* se ejecuta desde una VM ajena a los nodos de *Proxmox* llamada `deploy-vdi` alojada en un cluster montado sobre `XCP-ng`. El *playbook* de *Ansible* debe ejecutarse dos veces: la primera para que se realicen los cambios necesarios y la segunda para confirmar que todo ha funcionado como se espera (debe mostrar que no se han realizado cambios).

El siguiente paso consiste en configurar el almacenamiento. El servidor dispone de dos discos NVMe configurados en RAID 1 que sirven de almacenamiento para la instalación de *Proxmox*. El almacenamiento que usarán las máquinas virtuales será mediante NFS. Para configurar un almacenamiento por NFS en *Proxmox*, hay dos opciones: hacerlo a través de la interfaz gráfica o modificando el archivo mediante un editor de texto. En este caso, se deben añadir dos almacenamientos NFS, uno que utilizarán las máquinas virtuales y otro que se servirá para las copias de seguridad (*backup*).

Para añadir un almacenamiento por NFS en *Proxmox*, se debe navegar a la sección `Datacenter > Storage` y hacer clic en `Add`. Una vez allí, es necesario rellenar los campos `ID`, `Server`, y `Export` con la información del servidor NFS:

## 3.2. Instalación y configuración del hipervisor

- **ID**: un identificador único que se asigna al almacenamiento NFS dentro de *Proxmox*.
- **Server**: la dirección IP del servidor NFS donde se encuentra compartido el almacenamiento.
- **Export**: la ruta del recurso compartido en el servidor NFS. Define la carpeta o directorio específico que *Proxmox* utilizará para almacenar o acceder a datos en ese servidor.

Se utilizará la plantilla común del CeSViMa como base. Esta plantilla corresponde a un Windows 11 sin programas instalados, pero optimizada para su uso en entornos virtualizados. Para copiarla, necesitaremos su disco y su archivo de configuración. El disco de la plantilla se almacenará en el servidor NFS y el archivo de configuración dentro del almacenamiento interno del servidor. Una vez copiados ambos, se deben realizar modificaciones al archivo de configuración de la plantilla. Esto se debe a que el nodo donde corre la plantilla común del CeSViMa, es nodo con microprocesador Intel. A su vez el almacenamiento de esa plantilla está alojado en otra ruta del servidor NFS, por lo que tenemos que cambiar ambos para que la plantilla funcione correctamente en nuestro entorno *Proxmox*.

El archivo de configuración en *Proxmox* luce así:

Código 3.1: Configuración de VM en *Proxmox*

```
1 root@vdi3n1:~# cat /etc/pve/qemu-server/2010.conf
2 agent: 1
3 balloon: 4096
4 bios: ovmf
5 boot: order=scsi0
6 cores: 6
7 cpu: EPYC-Genoa
8 efidisk0: vdi-gpu1:2010/vm-2010-disk-0.qcow2,efitype=4m,
9 pre-enrolled-keys=1,size=528K
10 machine: pc-q35-9.0
11 memory: 6144
12 meta: creation-qemu=6.1.0,ctime=1639466821
13 name: base-upm-docencia-gpu-windows11
14 net0: virtio=XX:XX:XX:XX:XX:XX,bridge=vbr1595,firewall=1
15 numa: 0
16 ostype: win11
17 scsi0: vdi-gpu1:2010/vm-2010-disk-1.qcow2,cache=writeback,discard=on,size=55G,ssd=1
18 scsihw: virtio-scsi-pci
19 sockets: 1
```

Una vez que la máquina virtual esté en funcionamiento, lo siguiente será configurar el *backup* de la máquina virtual. Para ello, se debe navegar a la sección *Datacenter > Backup* y hacer clic en *Add*. En este menú se selecciona el almacenamiento donde se realizará la copia de seguridad y el horario en el que se realizará la copia de seguridad, que en este caso se ha configurado para todos los días a la 1 de la mañana. También es necesario introducir el correo en *Send email to* para que *Proxmox* mande un aviso por correo en caso de error y en *Selection mode* elegir el modo *Include selected VMs*. En la parte inferior, aparecerá una lista donde se debe seleccionar la máquina de la que se desea hacer la copia de seguridad. Lo más importante es el *Mode*, que debe ser *Snapshot*. Un *snapshot* es una copia puntual del estado de una máquina virtual en un momento específico. Esta instantánea captura todos los datos y el estado de los discos de la VM en ese instante, lo que permite restaurarla a ese

estado si es necesario. El modo `Snapshot` permite realizar una copia de seguridad con un tiempo mínimo de inactividad, ya que la VM sigue operando, reduciendo el impacto en el rendimiento y manteniendo la disponibilidad de los servicios.

### 3.3. NVIDIA vGPU

#### 3.3.1. Rol

La virtualización de GPUs es la tecnología que permite que varios usuarios utilicen de forma eficiente los recursos de una GPU, manteniendo un alto rendimiento gráfico. *NVIDIA vGPU* hace posible que múltiples máquinas virtuales compartan una única GPU física. Esto optimiza el uso de los recursos disponibles y asegura un rendimiento gráfico adecuado para aplicaciones que requieren un alto poder de procesamiento gráfico. Además, esta tecnología permite una gestión centralizada de los recursos, lo que facilita su administración. También tiene la capacidad de escalar según las necesidades de los centros de datos, lo que garantiza flexibilidad y un rendimiento constante en la infraestructura gráfica.

Los perfiles de NVIDIA vGPU son configuraciones predefinidas que controlan cómo se dividen los recursos de una GPU física entre múltiples máquinas virtuales. Cada perfil determina la cantidad de memoria gráfica (vRAM), núcleos de procesamiento y otras capacidades asignadas a cada VM, permitiendo un rendimiento predecible y adaptado a diferentes cargas de trabajo. Tipos de perfiles:

- **Q-series:** Estaciones de trabajo virtuales para profesionales creativos y técnicos que requieren el rendimiento y las funciones de los controladores RTX Enterprise.
- **B-series:** Diseñados para escritorios virtuales de uso empresarial.
- **A-series:** Transmisión de aplicaciones o soluciones basadas en sesiones para usuarios de aplicaciones virtuales.

Para calcular el número de escritorios soportados, se divide la capacidad total de la GPU (en memoria y recursos) entre los requisitos mínimos del perfil asignado, asegurando rendimiento adecuado para cada usuario. Por tanto, con perfiles 2Q y teniendo 4 GPUs L40s de 48 GB, el cálculo sería el siguiente:

$$\text{Memoria total disponible} = 4 \times 48 \text{ GB} = 192 \text{ GB}$$

$$\text{Número de escritorios soportados} = \frac{192 \text{ GB}}{2 \text{ GB por VM}} = 96 \text{ escritorios virtuales}$$

Por lo tanto, con 4 GPUs L40s de 48 GB y utilizando perfiles 2Q, se pueden soportar hasta 96 escritorios virtuales.

#### 3.3.2. Herramientas posibles

Existen diversas herramientas y tecnologías para abordar la virtualización gráfica, aunque en la práctica, *NVIDIA* se destaca como el principal proveedor en este ámbito. Algunas de las alternativas y tecnologías relevantes son:

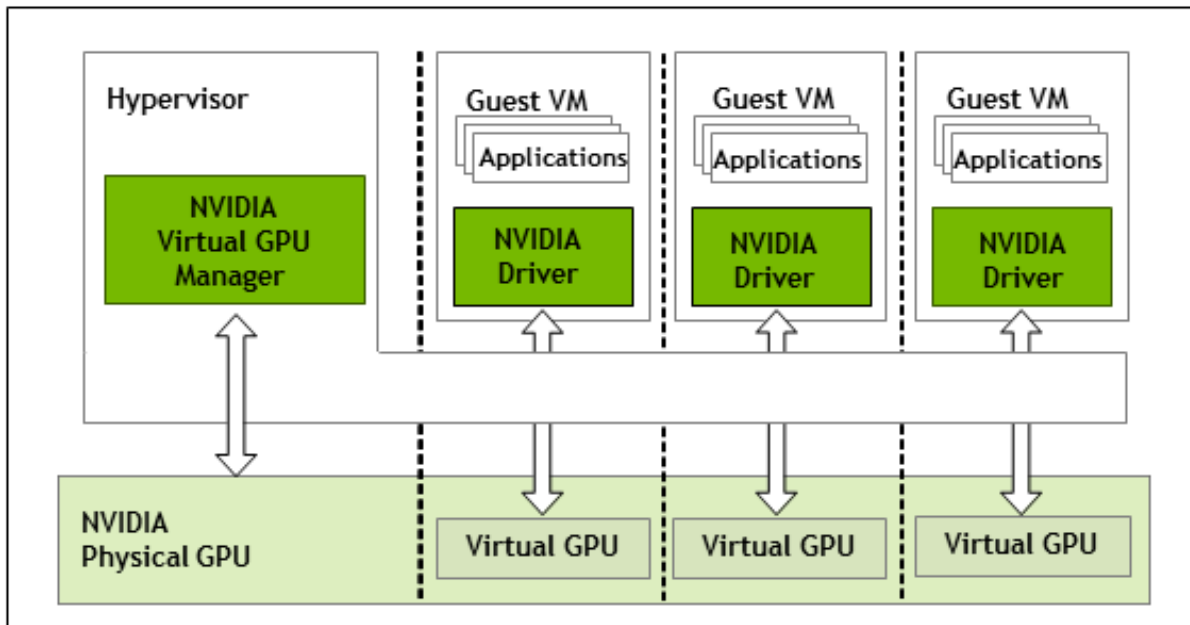


Figura 3.3: Estructura de uso de *NVIDIA-GPU* (tomada de [41])

- **NVIDIA GRID vGPU:** Solución avanzada que permite la virtualización de la GPU directamente en el hardware, compartiendo una GPU física entre múltiples máquinas virtuales sin pérdida de rendimiento significativa. Es la solución de referencia para escenarios de alto rendimiento en gráficos y aplicaciones profesionales.
- **Alternativas a las GPUs de NVIDIA:** En el mercado actual, otras compañías, como **AMD** y **Google** (con sus **TPUs**), también están desarrollando soluciones de procesamiento gráfico e inteligencia artificial, aunque estas no están tan extendidas ni son tan ampliamente utilizadas en el ámbito de la virtualización de escritorios gráficos y aplicaciones profesionales. Las **TPUs** de Google se especializan en tareas de aprendizaje automático y ofrecen eficiencia energética, pero están limitadas a la plataforma Google Cloud.
- **Mercados descentralizados:** Plataformas como **Render Network** o **FluxEdge** están emergiendo como soluciones para investigadores e instituciones que necesitan acceso a GPUs de alto rendimiento mediante el uso de recursos de GPU inactivos, aunque estas alternativas no ofrecen el mismo nivel de rendimiento ni escalabilidad que las soluciones dedicadas como GRID vGPU.

### 3.3.3. Justificación de la elección de NVIDIA vGPU

La elección de *NVIDIA GRID vGPU* está claramente justificada por varias razones clave:

1. **Escalabilidad:** La tecnología *vGPU* es altamente escalable, lo que permite adaptarse a las necesidades de diferentes cargas de trabajo. Puede asignar recursos dedicados de GPU a una sola máquina virtual cuando sea necesario, o bien distribuirlos entre varias máquinas virtuales.

2. **Compatibilidad y fiabilidad:** *GRID vGPU* garantiza la compatibilidad con una amplia gama de aplicaciones profesionales y está probado y validado en entornos de producción.
3. **Gestión centralizada:** La solución *vGPU* de *NVIDIA* permite gestionar todos los recursos de GPU de manera centralizada a través de una consola simplificada. Esto facilita la administración y asegura que los usuarios reciban los recursos gráficos necesarios según sus requerimientos.
4. **Liderazgo en el mercado:** *NVIDIA* ha sido un líder en el mercado de GPUs durante muchos años, y su tecnología *vGPU* es la opción más avanzada y madura en términos de virtualización gráfica. Esto garantiza una solución fiable y de alto rendimiento, respaldada por un soporte extenso y un ecosistema de aplicaciones profesionales.

### 3.3.4. Configuración/Implementación

La configuración de *NVIDIA vGPU* está dividida en 2 partes. El servidor de licencias y el host driver.

El proceso comenzó con la descarga de la imagen virtual preconfigurada (*appliance*) del *Delegated License Service* (DLS) desde el portal de licencias de *NVIDIA*. Esta imagen está diseñada para simplificar la instalación y proporcionar un entorno seguro para la gestión de licencias. En nuestro caso, se utilizó el hipervisor *Proxmox VE* para desplegar esta solución. La imagen se distribuye en formato *QCOW2* (*QEMU Copy-On-Write*) y fue importada a *Proxmox* mediante el comando:

```
qm importdisk 1165 nls-3.4.0-bios.qcow2 vdi-management
```

Aquí, 1165 es el identificador único de la máquina virtual en *Proxmox*, y *vdi-management* corresponde al almacenamiento donde se cargó la imagen. Posteriormente se configuró la máquina virtual utilizando la interfaz de *Proxmox* para asociar el disco importado, asignar memoria, CPU y la MAC para que el servidor DHCP le asignase la IP.

Una vez que se enciende la máquina, se accede a la IP para terminar su configuración, rellenando el nombre del servidor de licencias y registrándolo en el portal de licencias.

A continuación, se detalla el proceso de configuración e instalación de *NVIDIA vGPU* en un entorno virtualizado con *Proxmox VE*, permitiendo la virtualización eficiente de recursos de GPU. Este procedimiento es esencial para habilitar capacidades gráficas avanzadas en máquinas virtuales destinadas a aplicaciones intensivas en procesamiento gráfico.

#### 3.3.4.1. Instalación de Dependencias

Antes de comenzar la instalación, es necesario instalar las herramientas básicas para compilar e integrar los controladores de *NVIDIA*:

```
apt install unzip gcc make
```

Además, se instalan los encabezados del kernel correspondientes a la versión del sistema operativo:

```
apt install pve-headers-$(uname -r)
```

Estas dependencias aseguran la compatibilidad y correcta integración del software con el sistema.

#### 3.3.4.2. Instalación del Software NVIDIA

El software de *NVIDIA vGPU* debe descargarse desde el portal oficial de licencias. Una vez descargado el archivo comprimido, se descomprime y se instala el controlador en el hipervisor.

Código 3.2: Configuración de NVIDIA-GPU

```
1 mkdir NVIDIA_vGPU
2 cd NVIDIA_vGPU
3 scp NVIDIA-GRID-Linux-KVM-550.127.06-550.127.05-553.24.zip root@vdi3n1:/NVIDIA_vGPU
4 unzip NVIDIA-GRID-Linux-KVM-550.127.06-550.127.05-553.24.zip
5 cd Host_Drivers
6 chmod +x NVIDIA-Linux-x86_64-550.127.06-vgpu-kvm.run
7 ./NVIDIA-Linux-x86_64-550.127.06-vgpu-kvm.run --dkms
8 reboot
```

El argumento `-dkms` asegura que los controladores se recompilen automáticamente tras futuras actualizaciones del *kernel*. Finalmente, se reinicia el servidor para aplicar los cambios.

#### 3.3.4.3. Configuración de SR-IOV

Para habilitar *SR-IOV (Single Root I/O Virtualization)*, que permite dividir una GPU física en múltiples dispositivos virtuales, se crea un servicio de `systemd` que habilita SR-IOV al inicio del sistema. El archivo del servicio se crea y edita de la siguiente forma:

```
vim /etc/systemd/system/nvidia-sriov.service
```

El contenido del archivo es:

Código 3.3: Habilitación SR-IOV

```
1 [Unit]
2 Description=Enable NVIDIA SR-IOV
3 After=network.target nvidia-vgpu.service nvidia-vgpu-mgr.service
4 Before=pve-guests.service
5
6 [Service]
7 Type=oneshot
8 ExecStart=/usr/lib/nvidia/sriov-manage -e ALL
9
10 [Install]
11 WantedBy=multi-user.target
```

Tras guardar el archivo, se habilita y ejecuta el servicio con los siguientes comandos:

```
systemctl daemon-reload
systemctl enable --now nvidia-sriov.service
```

#### 3.3.4.4. Deshabilitar el Controlador Nouveau

El controlador *nouveau*, una alternativa de código abierto para GPUs *NVIDIA*, debe deshabilitarse para evitar conflictos. Esto se logra añadiendo una línea al archivo

## Desarrollo

---

```
/etc/modprobe.d/blacklist.conf:  
echo "blacklist nouveau" >> /etc/modprobe.d/blacklist.conf
```

### 3.3.4.5. Dispositivos Mediados en Proxmox

En *Proxmox VE*, los dispositivos mediados (*mediated devices*) permiten crear perfiles de GPU virtuales basados en los diferentes modelos que ofrece *NVIDIA vGPU*. Estos perfiles dividen una GPU física en varias instancias virtuales con configuraciones específicas (como cantidad de memoria y núcleos disponibles), adaptándose a las necesidades de cada máquina virtual.

Para optimizar el consumo de recursos y distribuir la carga de trabajo entre varias GPUs físicas, es necesario reorganizar el archivo de configuración. Este ajuste asegura que las máquinas virtuales consuman los perfiles en orden y no utilicen siempre la primera GPU disponible, logrando una distribución más equilibrada.

- Crear un grupo de dispositivos mediados (*mediated devices*) a nivel datacenter dentro de `Resource Mappings`.
- Reorganizar el archivo `/etc/pve/`, siguiendo las recomendaciones de Proxmox Bugzilla #5216, para garantizar que los perfiles se asignen de manera ordenada.

Una vez se ha configurado todo, se puede sacar del servidor de licencias un *token* para probar la configuración. Este *token* se añade en la plantilla de Windows, y después de instalar los *guest drivers* y reiniciar, podemos comprobar si la máquina consume la licencia correctamente.

## 3.4. Broker/Intermediario de conexiones

### 3.4.1. Rol

Un *Broker* o programa intermediario o mediador es una tecnología encargada de administrar la forma en que los recursos o servicios se asignarán en determinado momento al usuario. Generalmente, estos recursos pueden estar alojados en un centro de datos, ya sean físicos o virtuales.

El Broker es el software que permite la comunicación entre el cliente y el hipervisor, y controla el ciclo de vida de los escritorios virtuales.

A través del broker los administradores pueden gestionar qué recursos virtuales o físicos se ponen a disposición de cada uno de los usuarios o grupos de usuarios, en qué lapso y cuáles son los permisos o privilegios asignados. Algunos brokers incluso permiten configurar sobre qué hipervisor se van a alojar cada escritorio virtual o los servidores de aplicaciones a los que se va a acceder. Los más avanzados ofrecen la posibilidad de elegir entre distintas ubicaciones donde alojar esos servicios o acceder desde aplicaciones de terceros a los recursos del centro de datos.

### 3.4.2. Herramientas posibles

Las alternativas de intermediación incluyen *UDS Enterprise*, *EVA OS*, *VMWARE HORIZON* y *FLEXVDI*. Todos proporcionan alta disponibilidad, permiten múltiples protocolos de conexión de manera simultánea, dan soporte para escritorios virtuales

### 3.4. Broker/Intermediario de conexiones

Windows y Linux, permiten escritorios virtuales basados en plantillas y sus interfaces gráficas son adecuadas y potentes.

*UDS Enterprise* se destaca por su sistema avanzado de caché de dos niveles que permite dejar máquinas en suspensión sin consumir recursos [42]. Su documentación es bastante completa y dispone de funcionalidades, como la capacidad de tener credenciales diferentes para el sistema.

#### 3.4.3. Justificación de la elección de UDS Enterprise

*UDS* se define como una infraestructura de un bróker de conexiones multiplataforma que permite, facilita y gestiona el acceso de usuarios a recursos TI en centros de datos o en la nube. Con *UDS Enterprise* es posible la administración y despliegue de escritorios virtuales tanto en sistemas Windows como Linux, además de la posibilidad de consolidar los servicios de escritorios virtuales. Otra característica importante es el acceso remoto a equipos físicos o virtuales.

*UDS Enterprise* es un bróker de conexiones multiplataforma desarrollado, soportado y comercializado por la empresa VirtualCable.

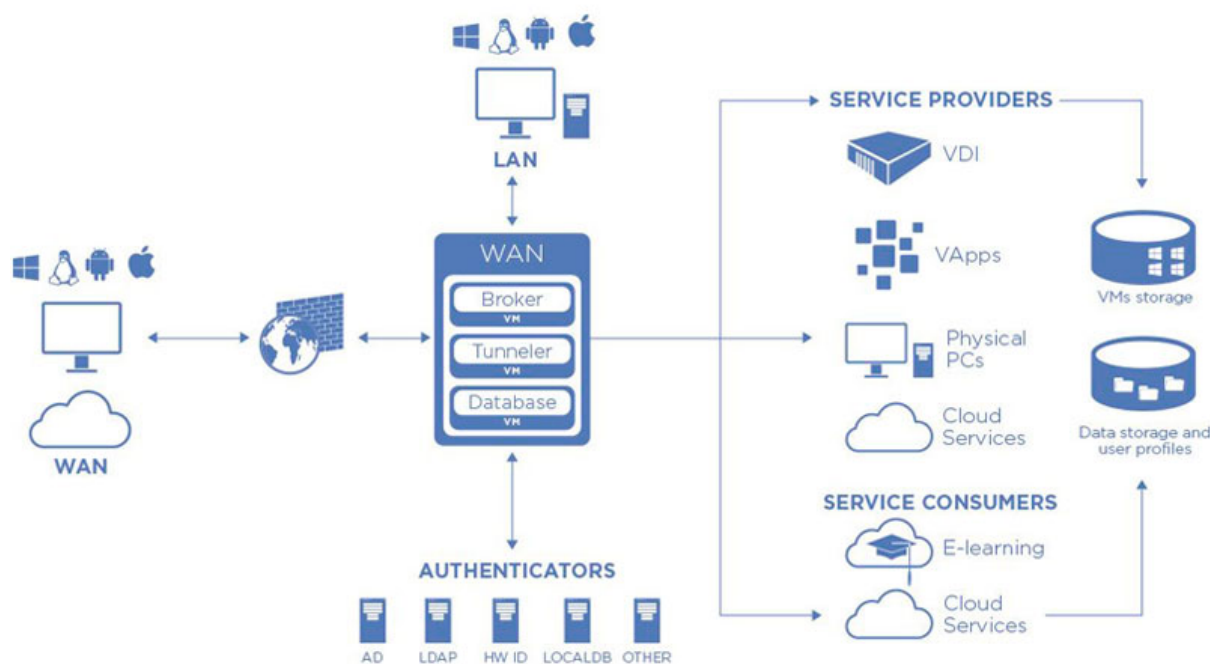


Figura 3.4: Estructura del bróker *UDS* (tomada de la Documentación de *UDS Enterprise*)

Sus características principales son:

- Instalación y administración sencilla.
- Sistema por suscripción, sin necesitar licencias.
- Permite la integración fácil con software de terceros.

La arquitectura de *UDS* se compone de:

1. **Clientes de conexión:** Son los dispositivos desde los cuales el usuario se conecta a los escritorios virtuales.
2. **Servidores UDS:** Son 3: el broker de conexiones que gestiona de principio a fin el ciclo de vida de los servicios de escritorio remoto y la comunicación con los proveedores de servicios, la base de datos *MySQL (Database)* y el servidor túnel (*Tunneler*) permite hacer seguros los accesos desde internet (SSL).
3. **Autenticadores:** Controlan el acceso de usuarios autorizados a los escritorios remotos.

El software *UDS Enterprise* ofrece una interfaz de usuario amigable e intuitiva, de manera que la creación, configuración, despliegue y administración de escritorios virtuales se puede realizar de forma sencilla.

### 3.4.4. Configuración de UDS Enterprise

La configuración de *UDS Enterprise* se realiza completamente a través de la interfaz web. El primer paso consiste en editar el proveedor de servicio, que en este caso es *Proxmox*. Esta configuración establece la conexión con el hipervisor y requiere completar los siguientes campos: Nombre, Servidor, Puerto, Nombre de usuario y Contraseña. Además, en la pestaña Avanzado, se deben especificar varios parámetros adicionales, tales como la concurrencia de creación y eliminación de máquinas virtuales, el rango de direcciones MAC y el *VmID* de inicio.

Una vez que el proveedor de servicio está configurado, el siguiente paso es crear un Servicio. Un servicio en *UDS Enterprise* permite crear y gestionar recursos. Para crear el servicio, primero seleccionamos la opción *Proxmox Linked Clone*, le asignamos un nombre y elegimos un pool de *Proxmox* (en *Proxmox*, los pools son etiquetas que agrupan máquinas virtuales). Deshabilitamos la opción de Alta Disponibilidad (HA), ya que en este caso no es necesaria, y activamos la opción *SOFT Shutdown*.

En la sección de Máquina, seleccionamos la máquina base (que corresponde a la plantilla en *Proxmox*), el almacenamiento previamente configurado en *Proxmox*, asignamos un nombre al servicio y establecemos la longitud de las máquinas virtuales a crear.

Una vez creado el servicio, el siguiente paso es crear el Pool de Servicios. Un pool de servicios actúa como un contenedor lógico que agrupa los recursos de un proveedor de servicios, facilitando la publicación y el acceso a esos recursos. Para configurar el pool, es necesario completar los siguientes campos:

- **Principal:** Nombre, Nombre corto, Servicio base (el servicio creado previamente) y OS Manager.
- **Visualización:** Marcar “Sí”, seleccionar la imagen asociada y el grupo de pool.
- **Avanzado:** Activar la opción de permitir a los usuarios eliminar y mostrar transportes.
- **Disponibilidad:** Completar los campos relacionados con los servicios disponibles, los servicios para mantener en caché y el número máximo de servicios a proveer.

En este caso de prueba, dado que se trata de un piloto, se establecerá que el número de servicios disponibles sea 3, la caché se dejará en 0 y el número máximo de servicios será 96, que corresponde al número máximo de vGPUs disponibles en la infraestructura.

Tras la creación del pool de servicios, es necesario configurar los Grupos, que agrupan a los usuarios con acceso a los escritorios virtuales, y los Transportes, que son los medios de conexión posibles para acceder al escritorio remoto. En los transportes, se deben añadir las opciones `RDP FreeRDP` y `Web` mediante `HTML5 RDP`.

Finalmente, se deben configurar ciertas acciones programadas, como:

- *Eliminar los servicios de usuarios antiguos* después de 12 horas de inactividad.
- *Publicar la plantilla*: La acción de publicar se gestiona mediante un calendario. Para facilitar la delegación de la gestión de la plantilla, se ha configurado de la siguiente manera: el administrador tiene acceso al calendario y, cuando desea publicar una nueva plantilla, asigna una fecha y hora. La acción programada se encarga de publicar la plantilla en ese momento.

## 3.5. Automatización del Despliegue

### 3.5.1. Rol

La automatización del despliegue de software es esencial para los proveedores de servicios en la nube actuales [43]. Esta práctica mejora significativamente la eficiencia operativa al aplicarse a diversas herramientas como la automatización de la gestión y configuración de servidores, la orquestación, la integración continua (CI/CD), las pruebas automáticas y los contenedores.

### 3.5.2. Herramientas Posibles

Diversas herramientas están disponibles para la automatización del despliegue. Algunas de las más conocidas [43] son:

- **Chef**: Herramienta popular para mantener configuraciones consistentes en múltiples servidores, permitiendo la integración con servicios como *Amazon EC2* y *Microsoft Azure*. *Chef* utiliza una arquitectura maestro/cliente y un DSL basado en *Ruby* para la configuración.
- **Ansible**: A diferencia de *Chef*, *Ansible* tiene una arquitectura sin agente (*agentless*), lo que significa que no requiere instalar agentes en los nodos controlados. Usa `SSH` para gestionar los sistemas y permite automatizar tareas repetitivas sin necesidad de un software adicional [44, 45].
- **SaltStack**: Dependiendo de un sistema maestro/minion, *SaltStack* optimiza la velocidad y escalabilidad, permitiendo la gestión de grandes infraestructuras de manera eficiente. Al igual que *Ansible*, también tiene una arquitectura sin agente.
- **Puppet**: Utiliza una arquitectura cliente/servidor y un lenguaje declarativo para definir configuraciones. Permite la automatización de servidores, gestión de configuraciones, cumplimiento y despliegue continuo.

- **Attune:** Solución de automatización para la provisión, parches y procedimientos de construcción automatizados. Admite *Bash*, *CMD* y *Powershell*, y es compatible con plataformas cruzadas como Windows, macOS y Linux.

### 3.5.3. Justificación de la elección de Ansible

*Ansible* es una herramienta de automatización y gestión de configuraciones que ayuda a las organizaciones a mantener y gestionar grandes cantidades de servidores virtuales y privados. *Ansible* se destaca por su arquitectura sin agente, lo que significa que no requiere la instalación de software adicional en los nodos que se gestionan. Así se mejora la seguridad y la estabilidad del sistema. Además, utiliza SSH para comunicarse con los sistemas remotos, lo que simplifica su uso y reduce la sobrecarga de red.

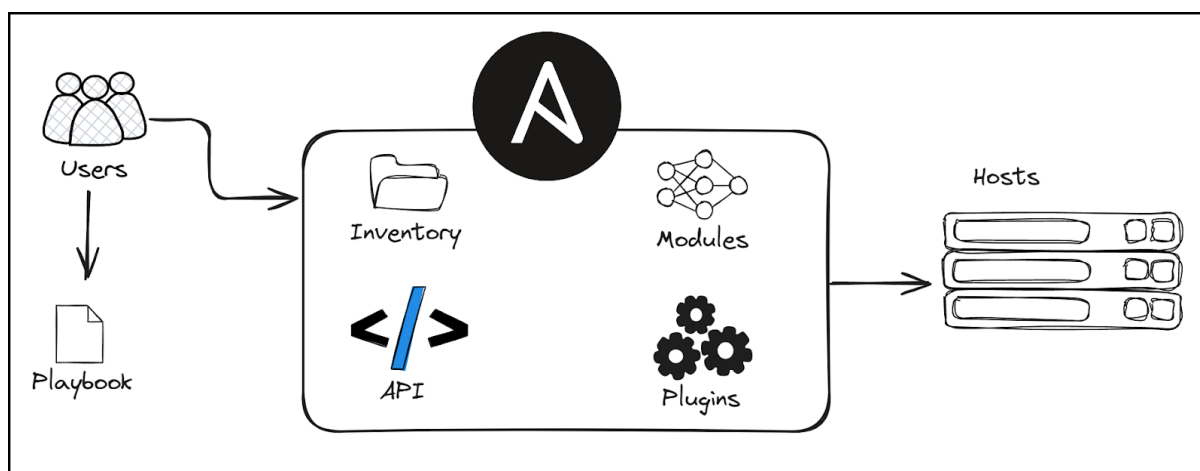


Figura 3.5: Arquitectura de *Ansible*

Los puntos clave de *Ansible* son los siguientes:

- Es una herramienta de automatización de código abierto para gestionar la infraestructura TI.
- Permite la automatización de tareas repetitivas, el despliegue de aplicaciones y la gestión de configuraciones.
- Utiliza la sintaxis *YAML* para describir tareas y flujos de trabajo.
- Se comunica con sistemas remotos utilizando SSH, sin necesidad de instalar ningún software de agente.
- *Ansible* también puede gestionar dispositivos de red si las interfaces API están disponibles.
- *Ansible Galaxy* sirve como un repositorio para roles preconstruidos, módulos y colecciones de *Ansible*.
- *Ansible Automation Platform* (AAP) es la versión empresarial que proporciona características adicionales como la programación de trabajos y controles de acceso.

La elección de *Ansible* para la automatización del despliegue se basa en su facilidad de uso, su arquitectura sin agente y su capacidad para integrarse con otros sistemas y herramientas en la infraestructura.

### 3.5.4. Configuración/Implementación

*Ansible* [46] utiliza *playbooks* escritos en *YAML*, que definen las tareas a ejecutar. En este caso, se han desarrollado varios roles para abordar diferentes aspectos de la configuración, como red, paquetes, zona horaria y optimizaciones. A continuación, se describen los roles y tareas implementados.

#### 3.5.4.1. Rol network

Este rol configura la red de los nodos de *Proxmox VE*, asegurando conectividad, sincronización horaria y la correcta configuración del cliente de correo.

- **Copia del archivo `/etc/hosts`:** Permite la resolución de nombres internos sin depender de DNS externos.
- **Configuración del `hostname`:** Establece el nombre del host utilizando la variable del inventario `inventory_hostname`.
- **Obtención de la IP del nodo:** Utiliza el plugin `dig` para almacenar la IP del nodo como un hecho reutilizable.
- **Creación del archivo de configuración de red:** Genera `/etc/network/interfaces` usando plantillas adaptadas a la versión de *Proxmox*.
- **Optimización de conexiones NFS NetApp:** Ajusta el número máximo de slots en `sunrpc.conf` para mejorar el rendimiento con almacenamiento NFS.
- **Instalación y configuración de Chrony:** Instala el cliente NTP y configura los servidores de tiempo definidos.
- **Configuración del cliente de correo (Postfix):** Define el `relayhost`, establece el `hostname` y configura alias para el usuario `root`.
- **Reinicio del nodo tras cambios críticos:** Reinicia automáticamente si hay cambios en la configuración de red o NFS.

Este rol garantiza la conectividad entre nodos y la correcta sincronización horaria, factores críticos en entornos virtualizados.

#### 3.5.4.2. Rol packages

El rol `packages` gestiona los repositorios y paquetes del sistema, asegurando que los nodos dispongan del software necesario y que el sistema esté actualizado.

- **Eliminación de repositorios antiguos:** Elimina configuraciones obsoletas de `pve-enterprise`, `pve-community` y `Ceph`.
- **Copia de repositorios de Debian:** Configura los repositorios base necesarios para el sistema operativo.
- **Copia de repositorios de Proxmox:** Establece los repositorios de *Proxmox* adaptados al tipo de licencia (*enterprise* o *community*).

- **Instalación de paquetes comunes:** Garantiza que herramientas y dependencias esenciales estén presentes en los nodos.
- **Actualización de todos los paquetes:** Mantiene el sistema actualizado con las últimas versiones disponibles.
- **Actualización completa del sistema:** Realiza una actualización del sistema operativo utilizando `dist-upgrade`, gestionando cambios de dependencias si es necesario.

Este rol asegura una base de software sólida y actualizada para la operación de los nodos.

### 3.5.4.3. Rol `timezone`

El rol `timezone` se encarga de establecer una zona horaria consistente en todos los nodos, garantizando coherencia en registros, cron jobs y sincronización.

- **Establecimiento de la zona horaria:** Crea o actualiza el archivo `/etc/timezone` con la zona horaria definida.
- **Enlace a la zona horaria correcta:** Establece un enlace simbólico entre el archivo de zona horaria y `/etc/localtime`.
- **Reconfiguración de `tzdata`:** Asegura que los cambios se apliquen correctamente al sistema.

Este rol es fundamental para garantizar consistencia temporal en todo el clúster.

### 3.5.4.4. Rol `tunning`

El rol `tunning` realiza ajustes avanzados para optimizar el rendimiento y la estabilidad del sistema.

- **Limpieza automática de estadísticas RRD:** Implementa un script diario para eliminar archivos RRD antiguos y liberar espacio en disco.
- **Configuración de `lvm.conf`:** Copia un archivo de configuración optimizado para LVM, especialmente importante tras actualizar a *Proxmox VE 8*.

Este rol contribuye a mantener un sistema eficiente y limpio, evitando problemas derivados del uso prolongado o actualizaciones.

## Capítulo 4

# Impacto del trabajo

En este capítulo se presenta y discute el impacto del trabajo. En primer lugar, se incluye una sección para describir el impacto del proyecto, especialmente dentro de la UPM. Posteriormente, y según indicaciones del desarrollo de un Trabajo Fin de Grado en la UPM, se enumeran los objetivos del desarrollo sostenible que son abordados por el proyecto.

### 4.1. Impacto general

De manera general, el hecho de que los estudiantes y los investigadores de la UPM puedan disponer de capacidades gráficas avanzadas en sus entornos virtuales supone un importante impacto para su trabajo personal. Hay que tener en cuenta que disponer de un dispositivo con procesamiento gráfico potente no es sencillo: es costoso y los equipos deben de ser de una gama alta para poder instalar una tarjeta gráfica de calidad. Puede que disponga de un dispositivo (por ejemplo, un ordenador de sobremesa) que sí tenga estas capacidades, pero no será así en el resto de sus dispositivos (portátil, tablet, ...)

Para los miembros de la comunidad universitaria que no necesiten estas capacidades frecuentemente (que, en realidad, son la mayoría), se ofrece la posibilidad de que puntualmente (cursar una asignatura específica donde los interfaces gráficos sean imprescindibles, un desarrollo esporádico concreto que utilice vídeos en tiempo real, una colaboración internacional donde se realice diseño avanzado, etc.) las tengan disponibles en sus equipos habituales de trabajo.

Es importante apuntar que, aunque se trata de una mejora sustancial de la infraestructura de entornos virtuales actual de la UPM, no es necesario su disponibilidad masiva para un alto número de usuarios, ya que, en porcentaje de tiempo, las necesidades de capacidades gráficas avanzadas supone una fracción del tiempo total.

No obstante, hoy en día, el impacto de este proyecto en la comunidad universitaria de la UPM es aún limitado, principalmente debido a la naturaleza del prototipo desarrollado. La principal limitación está en la capacidad máxima de escritorios virtuales del desarrollo actual, con un tope de 96 usuarios. Obviamente, este sistema no es suficiente para cubrir las necesidades de los 39,000 usuarios potenciales de la universidad. Pero ese no era el fin último del trabajo ni del prototipo, sino probar que la infraestructura era factible.

## Impacto del trabajo

Sin embargo, las decisiones tomadas durante el diseño y la arquitectura del proyecto están orientadas a hacer que la escalabilidad y el despliegue sean lo más sencillo y rápido posible.

Un número importante de Escuelas y Centros de Investigación de la UPM incluyen asignaturas, docencia o investigación en campos como la simulación y predicción, diseño y AutoCAD, procesamiento de imágenes científicas, representación gráfica en 3D, procesamiento y edición de vídeos, inteligencia artificial y aprendizaje automático, etc. que se verán altamente beneficiados por el uso de los escritorios virtuales aquí presentados. A modo de ejemplo, se incluye una figura con muestras de potenciales necesidades en algunos centros de la UPM con imágenes encontradas en sus respectivas webs 4.1.

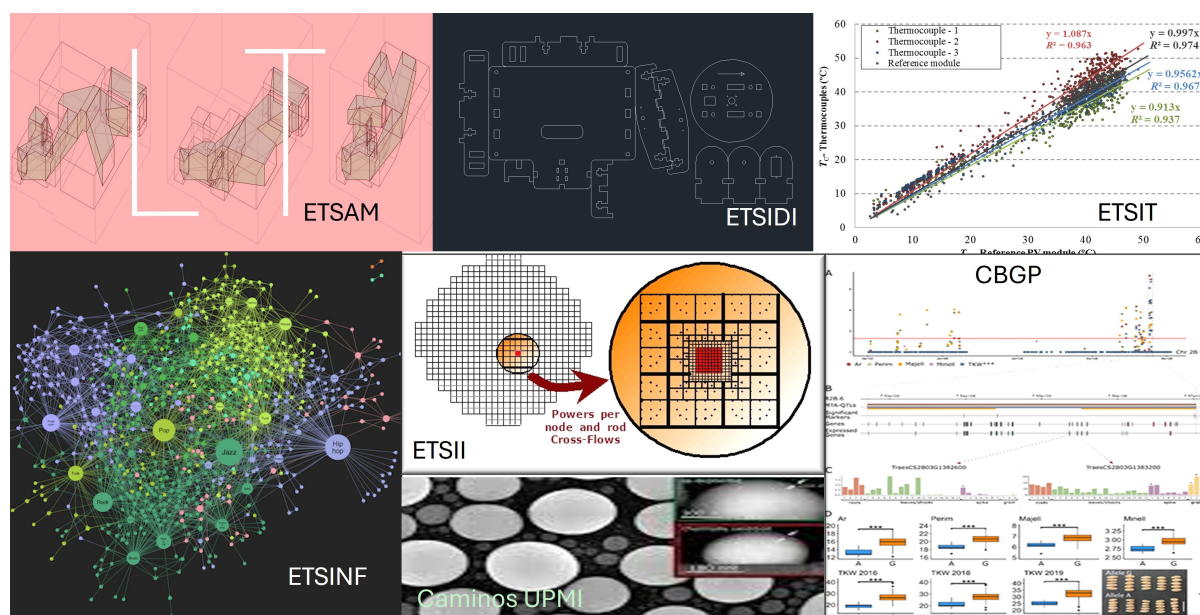


Figura 4.1: Docencia e I+D+i que involucran procesamiento de imágenes en la UPM

Como ya se ha indicado, únicamente hemos encontrado algunas universidades americanas (Sacramento State, State, Colorado, South California, Iowa State, Florida Atlantic, Old Dominion y Michigan State University, Udacity), algunas universidades chinas (Universidad de Tecnología de Zhejiang) y obtenido referencias orales de universidades españolas (Murcia y Sevilla) con oferta y proyectos similares al aquí presentado.

Casi todas ellas, han desarrollado sus proyectos a través de productos comerciales y/o empresas privadas (*Azure, VMware, Cloudlabs Inc., ...*).

Aunque no se ha desarrollado una búsqueda profunda, no hemos sido capaces de encontrar documentos que avalen la disposición de servicios de GPU con escritorios virtuales basadas en Software libre para su comunidad universitaria y que se hayan elaborado por sus propias unidades de desarrollo tecnológico. Es bien posible que otras universidades hayan o estén involucradas en proyectos similares, pero no aparecen trazas públicas de ello.

De esta manera, la UPM dispondrá de una tecnología pionera para su comunidad

científica, docente y estudiantil.

### 4.2. Objetivos de Desarrollo Sostenible

Los Objetivos de Desarrollo Sostenible (ODS), también conocidos como Objetivos Globales, fueron adoptados por las Naciones Unidas en 2015 como plan mundial para resolver problemas consustanciales a la humanidad como la pobreza, el hambre, la protección del planeta, la discriminación contra mujeres y niñas y la garantía de que para el 2030 todas las personas disfruten de paz y prosperidad<sup>1</sup>. Estos ODS se han plasmado en la agenda 2030 del estado español<sup>2</sup>.

Son un total de diecisiete que buscan involucrar a gobiernos, compañías, sociedad civil y personas título individual para conseguir metas globales. Cada Objetivo de Desarrollo Sostenible incluye varias metas específicas con indicadores para evaluar su cumplimiento. Los 17 ODS han de interpretarse en su conjunto, de manera que la acción en un área afectará los resultados en otras áreas y que busca equilibrar la sostenibilidad social, económica y ambiental.

De esta forma, un trabajo fin de grado y el desarrollo de una herramienta para el estudiantado de una universidad pública suponen una aportación, aunque sea modesta, a los ODS.

Se mencionan aquellos de mayor relevancia:

- **ODS 4: EDUCACIÓN DE CALIDAD.** La infraestructura de escritorios virtuales de la UPM, y ahora su extensión con GPU, garantiza que la universidad tenga una alta capacidad para ofrecer una educación de calidad que, además, permite el acceso a los servicios de los campus, incluso ante posibles contratiempos. Así mismo, asegura el acceso universal a los estudios, quitando relevancia al dispositivo concreto que tengan los estudiantes lo que favorece la igualdad de oportunidades.
- **ODS 9: INDUSTRIA, INNOVACIÓN E INFRAESTRUCTURA.** La infraestructura de escritorios virtuales de la UPM permite reducir la huella de carbono de la universidad: disminuye el consumo energético y permite limitar la adquisición de nuevo hardware. Indudablemente es una infraestructura sostenible y resiliente. Es una tecnología llamada a impulsar la sostenibilidad durante los próximos años.
- **ODS 12: PRODUCCIÓN Y CONSUMO RESPONSABLES.** En primer lugar, se asegura una mayor eficiencia energética. Adicionalmente, los escritorios virtuales de la UPM permiten ampliar la vida útil de los equipos informáticos de la universidad. Al añadir la GPU, se favorece la economía circular y se reduce la generación de deshechos tecnológicos.
- **ODS 13: ACCIÓN POR EL CLIMA.** Los escritorios virtuales impulsan el acceso 24x7, desde cualquier lugar y dispositivo, favorecen la educación a distancia y el teletrabajo. Todo ello reduce la movilidad de los estudiantes y de personal, de manera que se limitan las emisiones de CO<sub>2</sub>.

---

<sup>1</sup><https://www.un.org/sustainabledevelopment/es/objetivos-de-desarrollo-sostenible/>

<sup>2</sup>[www.agenda2030.gob.es/objetivos/home.htm](http://www.agenda2030.gob.es/objetivos/home.htm)

## Capítulo 5

# Resultados y conclusiones

### 5.1. Resultados

El proyecto ha logrado cumplir la mayoría de los objetivos planteados inicialmente, alcanzando un diseño de una infraestructura de escritorios virtuales con tarjetas gráficas virtuales para aplicaciones gráficas intensivas. A partir de ese diseño, se implementó una arquitectura de virtualización basada en *Proxmox VE* con integración de *NVIDIA vGPU*, lo que permitió la asignación de recursos gráficos a los escritorios remotos. Se integró *UDS Enterprise* para la gestión centralizada de los escritorios virtuales, facilitando el acceso remoto para estudiantes y profesores y ofreciendo una experiencia de usuario uniforme y segura. Se desarrollaron y probaron *playbooks* de *Ansible* para automatizar la configuración de *Proxmox VE*. Estas automatizaciones redujeron el tiempo de configuración manual y minimizaron los posibles errores humanos.

Por otro lado, el diseño y puesta en acción de la solución ha tenido en cuenta la necesidad de escalabilidad futura como un requisito adicional. La infraestructura permite añadir nuevos nodos al *clúster* de forma sencilla, garantizando la capacidad de soportar un mayor número de usuarios y recursos a medida que crezcan las demandas.

Aunque los objetivos iniciales se han alcanzado satisfactoriamente, el proceso de optimización de la infraestructura aún se encuentra en desarrollo. Se están realizando pruebas de carga con aplicaciones gráficas avanzadas para evaluar el rendimiento de la infraestructura. Un *Plan de pruebas* exhaustivo, detallado y ambicioso está planificado para 2025 por parte del CeSViMa.

Para validar el funcionamiento de la infraestructura se realizó la prueba de consumir uno de los escritorios remotos mediante UDS y comprobar que disponía de la tarjeta gráfica virtual. Se hicieron algunas pruebas de rendimiento para ver qué capacidad gráfica podía soportar el escritorio remoto.

### 5.2. Trabajo futuro

Desde el CeSViMa se acepta que el proyecto aquí presentado ha cumplido con sus objetivos y se inicia ahora una nueva etapa, probablemente llevada a cabo por el mismo autor, de diseñar el *Plan de pruebas* y su ejecución y planificar su despliegue

## 5.2. Trabajo futuro

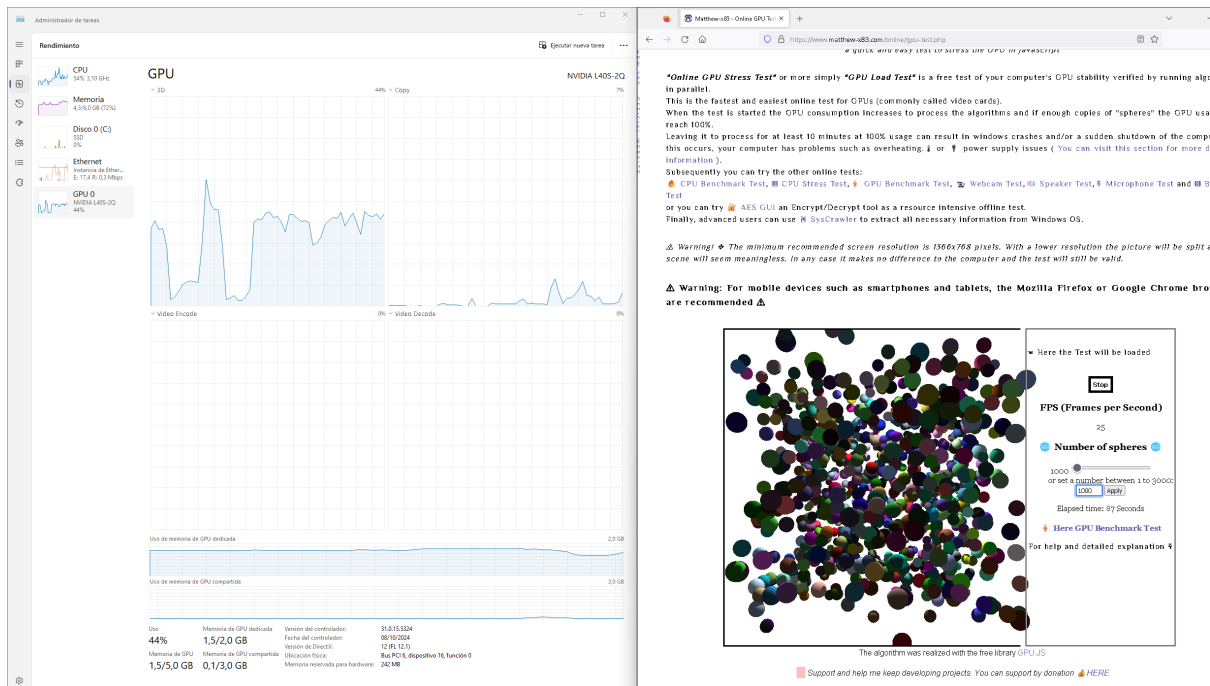


Figura 5.1: Prueba de estrés para verificar funcionamiento correcto de la GPU

general, como se ha realizado en [47, 48, 49, 50, 51]. Además se buscará ampliar las capacidades de los escritorios virtuales permitiendo su uso no sólo en las imágenes Windows sino también en las Linux.

Inicialmente, se pondrá en producción para que algunos usuarios seleccionados prueben sus prestaciones y características con el objetivo de ajustar diferentes parámetros para optimizar el servicio. Posteriormente, durante 2025, se planea incorporar nuevos nodos al sistema, lo cual permitirá aumentar el número de escritorios virtuales disponibles.

Otra medición posible y necesaria está en ver la aceptación del estudiantado y sus beneficios docentes, en la línea de [52, 53, 54]. Se considerará un elemento adicional fundamental en cuanto a la calidad del servicio ofrecido.

También sería interesante medir con detalle los efectos en consumos eléctricos globales y los ahorros globales en equipos, para confirmar las ventajas en costes y sostenibilidad verde de la propuesta de la UPM.

En cuanto a ideas y desarrollos futuros, algunas propuestas sobre nuevas tendencias y avances para la infraestructura de escritorios virtuales pueden encontrarse en [55, 56, 57, 58].

## **Capítulo 6**

### **Anexo**

Repositorio que contiene el código de los playbooks de Ansible: <https://github.com/amferreiro/Ansible>

# Bibliografía

- [1] Gerald J. Popek y Robert P. Goldberg. «Formal requirements for virtualizable third generation architectures». En: *Commun. ACM* 17.7 (jul. de 1974), págs. 412-421. ISSN: 0001-0782. DOI: 10.1145/361011.361073. URL: <https://doi.org/10.1145/361011.361073>.
- [2] Mendel Rosenblum y Tal Garfinkel. «Virtual machine monitors: current technology and future trends». En: *Computer* 38 (2005), págs. 39-47. URL: <https://api.semanticscholar.org/CorpusID:10385623>.
- [3] Tom Wall. *Virtualisation and Thin Client : A Survey of Virtual Desktop environments*. Computer Science, Engineering, 2009. URL: <https://api.semanticscholar.org/CorpusID:57469781>.
- [4] Karissa Miller y Mahmoud Pegah. «Virtualization: virtually at the desktop». En: *Proceedings of the 35th Annual ACM SIGUCCS Fall Conference*. SIGUCCS '07. Orlando, Florida, USA: Association for Computing Machinery, 2007, págs. 255-260. ISBN: 9781595936349. DOI: 10.1145/1294046.1294107. URL: <https://doi.org/10.1145/1294046.1294107>.
- [5] Pranit Patil. «Desktop Virtualization Technologies and Implementation». En: *International organization of Scientific Research* (feb. de 2012). DOI: 10.9790/3021-0202310314.
- [6] Li Yan. «Development and application of desktop virtualization technology». En: *2011 IEEE 3rd International Conference on Communication Software and Networks*. 2011, págs. 326-329. DOI: 10.1109/ICCSN.2011.6013725.
- [7] *Desktop Virtualization: Benefits, Types, Alternatives and How It Works*. <https://www.thesagenext.com/blog/desktop-virtualization>. 2022.
- [8] Kelly S. Hale y Kay M. Stanney. *Handbook of Virtual Environments: Design, Implementation, and Applications*. 2nd. USA: CRC Press, Inc., 2014. ISBN: 1466511842.
- [9] Rupesh Kumar, Arun Yadav y H Verma. «An analysis of Approaches for Desktop Virtualization and Challenges». En: *International Journal of Scientific Research in Science and Technology* (ago. de 2021), págs. 600-612. DOI: 10.32628/CSEIT2174133.
- [10] Abdallah Ibrahim et al. «Virtual Desktop Infrastructures: Architecture, survey and green aspects proof of concept». En: *2016 Seventh International Green and Sustainable Computing Conference (IGSC)*. Nov. de 2016. DOI: 10.1109/IGCC.2016.7892624.
- [11] Michael A. Erskine y János T. Füstös. «Survey of Desktop Virtualization in Higher Education: An Energy- and Cost-Savings Perspective». En: *Americas Conference on Information Systems*. 2013. URL: <https://api.semanticscholar.org/CorpusID:37805810>.


- [12] The Greaves Group. *Virtualization in Education*. <https://www.yumpu.com/en/document/view/6998676/virtualization-in-education-ibm>. 2007.
- [13] Shalabh Agrawal, Rana Biswas y Asoke Nath. «Virtual Desktop Infrastructure in Higher Education Institution: Energy Efficiency as an Application of Green Computing». En: *Proceedings of the 2014 Fourth International Conference on Communication Systems and Network Technologies*. CSNT '14. USA: IEEE Computer Society, 2014, págs. 601-605. ISBN: 9781479930708. DOI: 10.1109/CSNT.2014.250. URL: <https://doi.org/10.1109/CSNT.2014.250>.
- [14] Don-Anthony Dasilva et al. «Enabling Green IT through Building a Virtual Desktop Infrastructure». En: *2012 Eighth International Conference on Semantics, Knowledge and Grids*. 2012, págs. 32-38. DOI: 10.1109/SKG.2012.29.
- [15] Mayssa Youssef, Ruby Krishnaswamy y Adam Ouorou. «Green Virtual Desktop: Design and Operation». En: jun. de 2013. DOI: 10.1109/ICCITechnology.2013.6579533.
- [16] *Empowering Every Student in Education*. <https://resources.nvidia.com/en-us-virtualization-and-edu/vgpu-education-broch>. 2023.
- [17] Jim Denk y Laurie Fox. «The evolution of learning spaces». En: *Conference on User Services*. 2008. URL: <https://api.semanticscholar.org/CorpusID:20737811>.
- [18] C. O'Reilly. *Introducing personal virtual desktop (PVD) technology into a blended learning environment to scaffold technology learning*. Inf. téc. Trinity College Dublin., 2010.
- [19] H. Pandey. «Present scenario analysis of green computing approach in the world of information technology». En: *International Conference On Electrical Engineering and Computer Science*. 2012, págs. 2278-1129.
- [20] Carl K. Chang et al. «REACH platform — Remote access to smart home facility based computer science laboratory». En: *2010 IEEE Frontiers in Education Conference (FIE)*. 2010, F3F-1-F3F-7. DOI: 10.1109/FIE.2010.5673145.
- [21] Roberto C. Portugal, Luis A. Guerrero y David A. Fuller. «DeskTOP, a system based on virtual spaces to support and to promote collaborative learning (poster session)». En: *Proceedings of the Third International Conference on Collaborative Virtual Environments*. CVE '00. San Francisco, California, USA: Association for Computing Machinery, 2000, págs. 199-200. ISBN: 1581133030. DOI: 10.1145/351006.351045. URL: <https://doi.org/10.1145/351006.351045>.
- [22] Pawel Chrobak. «Implementation of Virtual Desktop Infrastructure in academic laboratories». En: *2014 Federated Conference on Computer Science and Information Systems*. 2014, págs. 1139-1146. DOI: 10.15439/2014F213.
- [23] Jheng-Yue Li et al. «The Implementation of a GPU-Accelerated Virtual Desktop Infrastructure Platform». En: *2017 International Conference on Green Informatics (ICGI)*. 2017, págs. 85-92. DOI: 10.1109/ICGI.2017.42.
- [24] Micah Dowty y Jeremy Sugerman. «GPU virtualization on VMware's hosted I/O architecture». En: *SIGOPS Oper. Syst. Rev.* 43.3 (jul. de 2009), págs. 73-82. ISSN: 0163-5980. DOI: 10.1145/1618525.1618534. URL: <https://doi.org/10.1145/1618525.1618534>.
- [25] Zhi-Hui Shang et al. «A project design of virtual cloud desktop system based on OpenStack». En: ene. de 2016. DOI: 10.2991/iccita-16.2016.28.
- [26] Chao-Tung Yang et al. «The Implementation of a Virtual Desktop Infrastructure with GPU Accelerated on OpenStack». En: oct. de 2018, págs. 366-370. DOI: 10.1109/I-SPAN.2018.00069.

- [27] Chih-Hung Chang et al. «On Construction and Performance Evaluation of a Virtual Desktop Infrastructure With GPU Accelerated». En: *IEEE Access* 8 (ene. de 2020), págs. 170162-170173. DOI: 10.1109/ACCESS.2020.3023924.
- [28] Manuel Bentele et al. «Towards a GPU-Accelerated Open Source VDI for OpenStack». En: *Cloud Computing*. Ed. por Mohammad R. Khosravi, Qiang He y Hai-peng Dai. Cham: Springer International Publishing, 2022, págs. 149-164. ISBN: 978-3-030-99191-3.
- [29] Wang Zhao et al. «Performance Analysis of NVIDIA GPU Virtualization in NARI Desktop Cloud». En: *2019 3rd International Conference on Data Science and Business Analytics (ICDSBA)*. 2019, págs. 405-408. DOI: 10.1109/ICDSBA48748.2019.00088.
- [30] Mboundou Mouyabi Seke. «Virtual Desktops in Institutions of Higher Education Using VMware View». En: *International Journal of Communication* 2 (2013). URL: <https://api.semanticscholar.org/CorpusID:59908218>.
- [31] Ting Gui et al. «vGPUPCD: Design and Implementation of Virtual GPU on Private Cloud Desktop». En: *2022 IEEE 2nd International Conference on Educational Technology (ICET)*. 2022, págs. 196-202. DOI: 10.1109/ICET55642.2022.9944412.
- [32] Susanta Nanda y tzi-cker Chiueh. «A Survey on Virtualization Technologies». En: (ene. de 2005).
- [33] Vasuprada Vijayakumar, V. Chitra y D. Priya. «Desktop Virtualization Solutions- A Comprehensive Survey». English. En: *International Journal of Advanced Computer Research* 3.3 (sep. de 2013). Copyright - Copyright International Journal of Advanced Computer Research Sep 2013; Características del documento - ; Illustrations; Diagrams; Última actualización - 2023-11-25, págs. 114-122. URL: <https://www.proquest.com/scholarly-journals/desktop-virtualization-solutions-comprehensive/docview/1464741239/se-2>.
- [34] Arun Sabale y Balu N Ilag. «Design the Azure Virtual Desktop Architecture». En: ene. de 2022, págs. 15-42. ISBN: 978-1-4842-8062-1. DOI: 10.1007/978-1-4842-8063-8\_2.
- [35] Tiago Rosado y Jorge Bernardino. «An overview of openstack architecture». En: *Proceedings of the 18th International Database Engineering & Applications Symposium. IDEAS '14*. Porto, Portugal: Association for Computing Machinery, 2014, págs. 366-367. ISBN: 9781450326278. DOI: 10.1145/2628194.2628195. URL: <https://doi.org/10.1145/2628194.2628195>.
- [36] David Freet et al. «Open source cloud management platforms and hypervisor technologies: A review and comparison». En: *SoutheastCon 2016*. 2016, págs. 1-8. DOI: 10.1109/SECON.2016.7506698.
- [37] Wasim Ahmed. *Mastering Proxmox - Third Edition: Build virtualized environments using the Proxmox VE hypervisor*. 3rd. Packt Publishing, 2017. ISBN: 1788397606.
- [38] Wasim Ahmed. *Proxmox Cookbook*. Packt Publishing, 2015. ISBN: 1783980907.
- [39] Rik Goldman. *Learning Proxmox VE*. Packt Publishing, 2016.
- [40] Sultan Abdullah Algarni et al. «Performance Evaluation of Xen, KVM, and Proxmox Hypervisors». En: *Int. J. Open Source Softw. Process.* 9.2 (abr. de 2018), págs. 39-54. ISSN: 1942-3926. DOI: 10.4018/IJOSSP.2018040103. URL: <https://doi.org/10.4018/IJOSSP.2018040103>.
- [41] Alex Herrera. *NVIDIA GRID vGPU: DELIVERING SCALABLE GRAPHICS-RICH VIRTUAL DESKTOPS*. Inf. téc. NVIDIA, jun. de 2015. URL: <https://images>.

- nvidia.com/content/pdf/grid/whitepaper/NVIDIA-GRID-WHITEPAPER-vGPU-Delivering-Scalable-Graphics-Rich-Virtual-Desktops.pdf.
- [42] W.A. Paucar Castillo B.D. Carrasco Cabezas. «Diseño de una infraestructura de escritorios virtuales, utilizando tecnología VMware y Uds Enterprise para la publicación de servicios VDI sobre Moodle». Tesis de Licenciatura. Universidad del Chimborazo, 2022.
- [43] James Benson, John Prevost y Paul Rad. «Survey of automated software deployment for computational and engineering research». En: *2016 Annual IEEE Systems Conference (SysCon)*. Abr. de 2016, págs. 1-6. DOI: 10.1109/SYSCON.2016.7490666.
- [44] Mohamed Alibi. *Ansible Quick Start Guide: Control and monitor infrastructures of any size, physical or virtual*. Packt Publishing, 2018. ISBN: 1789532930.
- [45] Mohammed Daffalla Elradi. «Ansible: A Reliable Tool for Automation». En: *Electrical and Computer Engineering Studies (2023)*. URL: <https://api.semanticscholar.org/CorpusID:260390273>.
- [46] Lorin Hochstein y Rene Moser. *Ansible: Up and Running Automating Configuration Management and Deployment the Easy Way*. 2nd. O'Reilly Media, Inc., 2017. ISBN: 1491979801.
- [47] Adel Belkhiri y Michel Dagenais. «Analyzing GPU Performance in Virtualized Environments: A Case Study». En: *Future Internet* 16.3 (2024). ISSN: 1999-5903. DOI: 10.3390/fi16030072. URL: <https://www.mdpi.com/1999-5903/16/3/72>.
- [48] Pedram Hossein Nakhai y Nor Badrul Anuar. «Performance evaluation of virtual desktop operating systems in virtual desktop infrastructure». En: *2017 IEEE Conference on Application, Information and Network Security (AINS) (2017)*, págs. 105-110. URL: <https://api.semanticscholar.org/CorpusID:41034738>.
- [49] Alex Berryman et al. «VDBench: A Benchmarking Toolkit for Thin-Client Based Virtual Desktop Environments». En: *2010 IEEE Second International Conference on Cloud Computing Technology and Science*. 2010, págs. 480-487. DOI: 10.1109/CloudCom.2010.106.
- [50] Pedro Casas et al. «Quality of experience in remote virtual desktop services». En: *2013 IFIP/IEEE International Symposium on Integrated Network Management (IM 2013)*. 2013, págs. 1352-1357.
- [51] Arman Sheikholeslami y Kalman Graffi. «A Systematic Quality Analysis of Virtual Desktop Infrastructure Technologies». En: *Euro-Par 2015: Parallel Processing Workshops*. Ed. por Sascha Hunold et al. Cham: Springer International Publishing, 2015, págs. 311-323. ISBN: 978-3-319-27308-2.
- [52] Francisco J. Rodríguez Lera et al. «Measuring Students Acceptance and Usability of a Cloud Virtual Desktop Solution for a Programming Course». En: *Applied Sciences* 11.15 (2021). ISSN: 2076-3417. DOI: 10.3390/app11157157. URL: <https://www.mdpi.com/2076-3417/11/15/7157>.
- [53] Prasad Calyam. «VMLab : Infrastructure to Support Desktop Virtualization Experiments for Research and Education». En: 2012. URL: <https://api.semanticscholar.org/CorpusID:40310692>.
- [54] Elham Alsadoon. «Intentions of Students to Continue Using Virtual Desktop Infrastructure: Expectation Confirmation Model Perspective». En: *IEEE Access* 10 (2022), págs. 49080-49087. DOI: 10.1109/ACCESS.2022.3173299.
- [55] Surjeet Kushwaha, Arun Yadav y H. Verma. «Desktop Virtualization: Benefits, Challenges, and Future Trends». En: *International Journal of Education and*

- Management Engineering* 13 (dic. de 2023), págs. 14-24. DOI: 10.5815/ijeme.2023.06.02.
- [56] J. R. Li, L. P. Khoo y S. B. Tor. «Desktop virtual reality for maintenance training: an object oriented prototype system (V-REALISM)». En: *Comput. Ind.* 52.2 (oct. de 2003), págs. 109-125. ISSN: 0166-3615. DOI: 10.1016/S0166-3615(03)00103-9. URL: [https://doi.org/10.1016/S0166-3615\(03\)00103-9](https://doi.org/10.1016/S0166-3615(03)00103-9).
- [57] Jiewei Wu et al. «SRIDesk: A Streaming based Remote Interactivity architecture for desktop virtualization system». En: *2013 IEEE Symposium on Computers and Communications (ISCC)*. 2013, págs. 000281-000286. DOI: 10.1109/ISCC.2013.6754960.
- [58] Doowon Jeong et al. «Investigation Methodology of a Virtual Desktop Infrastructure for IoT». En: *J. Appl. Math.* 2015 (2015), 689870:1-689870:10. URL: <https://api.semanticscholar.org/CorpusID:11052059>.

Este documento esta firmado por



<b>Firmante</b>	CN=tfgm.fi.upm.es, OU=CCFI, O=ETS Ingenieros Informaticos - UPM, C=ES
<b>Fecha/Hora</b>	Tue Jan 14 21:31:02 CET 2025
<b>Emisor del Certificado</b>	EMAILADDRESS=camanager@etsiinf.upm.es, CN=CA ETS Ingenieros Informaticos, O=ETS Ingenieros Informaticos - UPM, C=ES
<b>Numero de Serie</b>	561
<b>Metodo</b>	urn:adobe.com:Adobe.PPKLite:adbe.pkcs7.sha1 (Adobe Signature)