



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



Escola Tècnica  
Superior d'Enginyeria  
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica  
Universitat Politècnica de València

# Servicio de escritorio remoto en cluster virtualizado usando tecnología Xen Server y Windows 2012

Trabajo Fin de Grado

**Grado en Ingeniería Informática**

**Autor:** Joaquín Ruiz Gil

**Tutor:** Juan Carlos Cano Escribá

**Tutor:** Juan Luís Posadas Yagüe

Curso académico 2015/2016



# Resumen

---

En este trabajo se diseña y configura un sistema basado en virtualización que permite a los usuarios trabajar de forma remota en un sistema *Windows* o *Linux* de manera completamente transparente, permitiendo reanudar una sesión de trabajo sin la pérdida de tiempo que ocasionaría el cambiar de un equipo a otro. La solución propuesta se caracteriza por poseer: (1) alta escalabilidad, al mantener la eficiencia el aumento del número de servidores; (2) alta disponibilidad, gracias a ser un sistema por definición redundante; (3) balanceo de carga, al repartir la carga según reglas prefijadas o dinámicas; (4) mantenimiento eficiente, que permite una rápida recuperación frente a fallos hardware o software, en la mayor parte de los casos dejando al usuario ajeno a estos problemas. Tres servidores *XenServer* en configuración *Pool* hospedan servidores virtuales de *Windows Server* y *CentOS*, los cuales ofrecen servicios de escritorio remoto. Estos tres servidores albergan las máquinas virtuales que, bajo determinadas circunstancias de servicio, y gracias a contar con almacenamiento centralizado, pueden ser migradas, apagadas, arrancadas o creadas. El sistema inicial se ha evaluado mediante diferentes test de carga de trabajo con resultados satisfactorios.

**Palabras clave:** Virtualización, *XenServer*, *Windows Server*, *CentOS*, Servicios de Escritorio Remoto.



# Abstract

---

The objective of this work has been to design and set up a system that allows users to work remotely, either on a *Windows* or *Linux* system. It is completely transparent and allows the facility to restart a working session without the need to move from a computer desktop to another one. The proposed solution is characterized by: (1) high scalability and an efficient maintenance even with an increasing number of servers, (2) high availability due to a redundant system; (3) load balancing by distributing workload according to present rules or dynamic rules; (4) efficient maintenance that allows quick recovery from hardware or software failures, being these problems completely transparent for end users. Three *XenServer* servers with virtual *host* systems *Windows Server* and *CentOS*, which offer Remote Desktop Services. These three servers *hosting* virtual machines that under certain conditions of service, and by having centralized storage can be migrated, shut down, started or created. The initial system has been evaluated by different workload test with satisfactory results.

**Keywords:** Virtualization, *XenServer*, *Windows Server*, *CentOS*, Remote desktop services.

# Índice de contenidos

---

|   |           |
|---|-----------|
| <b>1. Introducción .....</b>  | <b>13</b> |
| 1.1. Motivación.....  | 13        |
| 1.2. Objetivos del trabajo .....                                    | 14        |
| 1.3. Estructura del trabajo.....                                    | 14        |
| <b>2. Tecnologías de Virtualización .....</b>                       | <b>17</b> |
| 2.1. Virtualización .....   | 17        |
| 2.2. Tipos de virtualización.....                                   | 18        |
| 2.2.1. Hipervisor tipo 1 .....                                      | 19        |
| 2.2.2. Hipervisor tipo 2.....                                       | 19        |
| 2.2.3. Emulación, paravirtualización y virtualización completa..... | 19        |
| 2.3. Plataformas de virtualización .....                            | 20        |
| 2.3.1. vSphere.....   | 20        |
| 2.3.2. Hyper-V.....   | 20        |
| 2.3.3. KVM.....   | 21        |
| 2.3.4. Xen Project .....  | 21        |
| 2.3.5. Características de las plataformas .....                     | 21        |
| 2.4. XenServer .....  | 25        |
| 2.4.1. Alta escalabilidad.....                                      | 26        |
| 2.4.2. Alta disponibilidad .....                                    | 26        |
| 2.4.3. Mantenimiento y recuperación frente a fallos .....           | 27        |
| 2.4.4. Balanceo de carga .....                                      | 28        |
| 2.4.5. Administración.....  | 28        |
| <b>3. Servicio de Escritorio Remoto en Pool .....</b>               | <b>31</b> |
| 3.1. Centro de Almacenamiento.....                                  | 32        |
| 3.2. Centro de Proceso.....   | 32        |
| 3.3. Bolsa de Servicios.....  | 32        |
| 3.4. Agente.....  | 33        |



|           |   |           |
|-----------|---|-----------|
| 3.5.      | <i>Equipos de SERP</i> .....  | 33        |
| <b>4.</b> | <b>Centro de Almacenamiento</b> .....                                       | <b>35</b> |
| 4.1.      | <i>NAS - Open Media Vault</i> .....   | 35        |
| 4.2.      | <i>Instalación</i> .....  | 36        |
| 4.3.      | <i>Configuración RAIDo</i> .....  | 36        |
| 4.4.      | <i>NFS y recursos exportados</i> .....                                      | 37        |
| 4.5.      | <i>Otros servicios y configuraciones</i> .....                              | 38        |
| <b>5.</b> | <b>Centro de Proceso</b> .....  | <b>41</b> |
| 5.1.      | <i>Instalación XenServer</i> .....  | 42        |
| 5.2.      | <i>Configuración sistema XenServer</i> .....                                | 46        |
| 5.2.1.    | <i>Añadir servidores a la consola</i> .....                                 | 53        |
| 5.2.2.    | <i>Configurar la red</i> .....  | 54        |
| 5.2.3.    | <i>Crear el Resource Pool</i> .....   | 55        |
| 5.2.4.    | <i>Configurar los repositorios</i> .....                                    | 55        |
| 5.2.5.    | <i>Encendido remoto</i> .....   | 58        |
| 5.2.6.    | <i>Máquinas Virtuales</i> .....   | 59        |
| 5.2.7.    | <i>Xs-tools</i> .....   | 60        |
| 5.2.8.    | <i>Instantáneas y Plantillas</i> .....                                      | 61        |
| 5.2.9.    | <i>Visualización del rendimiento</i> .....                                  | 63        |
| 5.2.10.   | <i>Alta disponibilidad</i> .....  | 65        |
| <b>6.</b> | <b>Bolsa de Servicios</b> .....   | <b>69</b> |
| 6.1.      | <i>Configuración de los servicios de escritorio remoto en Windows</i> ..... | 70        |
| 6.2.      | <i>Configuración de los servicios de escritorio remoto en Linux</i> .....   | 78        |
| 6.2.1.    | <i>Configuración kerberos</i> .....   | 79        |
| 6.2.2.    | <i>Configuración PAM</i> .....  | 82        |
| 6.2.3.    | <i>Configuración NIS</i> .....  | 83        |
| 6.2.4.    | <i>Configuración Thinlinc</i> .....   | 85        |
| <b>7.</b> | <b>Agente</b> .....   | <b>89</b> |

|   |            |
|---|------------|
| 7.1. HAProxy .....                            | 90         |
| <b>8. Pruebas de rendimiento .....</b>        | <b>93</b>  |
| 8.1. Carga de sesiones .....                  | 93         |
| 8.2. Prueba de balanceo de carga .....        | 95         |
| 8.3. Prueba de la alta disponibilidad .....   | 95         |
| <b>9. Conclusiones .....</b>                  | <b>99</b>  |
| 9.1. Propuestas de mejora .....               | 100        |
| <b>Bibliografía .....</b>                     | <b>103</b> |
| <b>ANEXO 1: Caso concreto: Hyper-V .....</b>  | <b>109</b> |
| <b>ANEXO 2: Características vSphere .....</b> | <b>111</b> |



# Índice de ilustraciones

---

|  |    |
|--|----|
| ILUSTRACIÓN 1 - HIPERVISOR TIPO 1.....                                       | 18 |
| ILUSTRACIÓN 2 - HIPERVISOR TIPO 2 .....                                      | 19 |
| ILUSTRACIÓN 3 - LOGOTIPOS PLATAFORMAS VIRTUALIZACIÓN.....                    | 20 |
| ILUSTRACIÓN 4 - MODELO SERP.....   | 31 |
| ILUSTRACIÓN 5 - SISTEMA IMPLEMENTADO.....                                    | 33 |
| ILUSTRACIÓN 6 –SERP ( CENTRO DE ALMACENAMIENTO).....                         | 35 |
| ILUSTRACIÓN 7 - APLICACIÓN WEB DE CONFIGURACIÓN DE OMV .....                 | 36 |
| ILUSTRACIÓN 8 - DISCOS DEL NAS .....   | 37 |
| ILUSTRACIÓN 9 - RAID 0 (STRIPE).....   | 38 |
| ILUSTRACIÓN 10 - RECURSOS EXPORTADOS EN /EXPORT/ .....                       | 38 |
| ILUSTRACIÓN 11 - CAPTURA DE CONSOLA SSH CON VARIOS COMANDOS EJECUTADOS.....  | 39 |
| ILUSTRACIÓN 12 – SERP (CENTRO DE PROCESO).....                               | 41 |
| ILUSTRACIÓN 13 - PANTALLA DEL ARRANQUE DE LA INSTALACIÓN DE XENSERVER .....  | 42 |
| ILUSTRACIÓN 14 - ARRANQUE DE XENSERVER .....                                 | 45 |
| ILUSTRACIÓN 15 - PANTALLA INICIAL XS-SERVER-0.....                           | 46 |
| ILUSTRACIÓN 16 - DESCARGA DE XENCENTER DESDE NUESTRO SERVIDOR XENSERVER..... | 47 |
| ILUSTRACIÓN 17 - APARIENCIA DE XENCENTER .....                               | 48 |
| ILUSTRACIÓN 18 - AÑADIR SERVIDOR.....  | 53 |
| ILUSTRACIÓN 19 - CONFIGURACIÓN DE RED CON XENCENTER.....                     | 54 |
| ILUSTRACIÓN 20 - AÑADIR SR (1/3) .....                                       | 56 |
| ILUSTRACIÓN 21 - AÑADIR SR (2/3) .....                                       | 57 |
| ILUSTRACIÓN 22 -AÑADIR SR (3/3).....   | 57 |
| ILUSTRACIÓN 23 - ACTIVACIÓN DE WAKE-ON-LAN.....                              | 58 |
| ILUSTRACIÓN 24 - ASISTENTE CREACIÓN MV .....                                 | 59 |
| ILUSTRACIÓN 25 - GESTIÓN DE INSTANTÁNEAS .....                               | 62 |
| ILUSTRACIÓN 26 - VISUALIZACIÓN DEL POOL.....                                 | 63 |
| ILUSTRACIÓN 27 - VISUALIZACIÓN DEL HOST.....                                 | 64 |



|  |    |
|--|----|
| ILUSTRACIÓN 28 - VISUALIZACIÓN DE LA MÁQUINA W2K12-LAB .....               | 64 |
| ILUSTRACIÓN 29 - ASISTENTE HA (1/4).....                                   | 65 |
| ILUSTRACIÓN 30 - ASISTENTE HA (2/4) .....                                  | 66 |
| ILUSTRACIÓN 31 - ASISTENTE HA (3/4).....                                   | 67 |
| ILUSTRACIÓN 32 - ASISTENTE HA (4/4) .....                                  | 67 |
| ILUSTRACIÓN 33 - SERP (BOLSA DE SERVICIOS) .....                           | 69 |
| ILUSTRACIÓN 34 - PANEL DEL SERVIDOR W2K12-LAB .....                        | 70 |
| ILUSTRACIÓN 35 - SELECCIÓN BASADA EN ROLES .....                           | 71 |
| ILUSTRACIÓN 36 - SELECCIÓN DEL SERVIDOR .....                              | 72 |
| ILUSTRACIÓN 37 - SELECCIÓN DE LOS SERVICIOS DE ESCRITORIO REMOTO.....      | 72 |
| ILUSTRACIÓN 38 - ELECCIÓN DE CARACTERÍSTICAS ADICIONALES.....              | 73 |
| ILUSTRACIÓN 39 - SERVICIOS EXISTENTES EN EL ROL DE ESCRITORIO REMOTO ..... | 73 |
| ILUSTRACIÓN 40 - CONFIRMACIÓN DE REINICIO .....                            | 74 |
| ILUSTRACIÓN 41 - INFORMACIÓN DE RESULTADOS DE INSTALACIÓN .....            | 75 |
| ILUSTRACIÓN 42 – POLÍTICAS DE GRUPO (LÍNEA DE COMANDO).....                | 76 |
| ILUSTRACIÓN 43 - UBICACIÓN EN LAS POLÍTICAS LOCALES .....                  | 76 |
| ILUSTRACIÓN 44 - INFORMACIÓN DEL SERVIDOR DE LICENCIAS.....                | 77 |
| ILUSTRACIÓN 45 - CONFIGURACIÓN DEL TIPO DE LICENCIA .....                  | 77 |
| ILUSTRACIÓN 46 - ACTIVACIÓN Y SELECCIÓN DE USUARIOS .....                  | 78 |
| ILUSTRACIÓN 47 - ASISTENTE AUTHCONFIG (1/3) .....                          | 84 |
| ILUSTRACIÓN 48 - ASISTENTE AUTHCONFIG (2/3) .....                          | 84 |
| ILUSTRACIÓN 49 - ASISTENTE AUTHCONFIG (3/3).....                           | 84 |
| ILUSTRACIÓN 50 - INSTALADOR THINLINC .....                                 | 86 |
| ILUSTRACIÓN 51 - CLIENTE THINLINC. DIALOGO INICIO SESIÓN.....              | 86 |
| ILUSTRACIÓN 52 - SESIÓN INICIADA CON THINLINC .....                        | 87 |
| ILUSTRACIÓN 53 - SERP (AGENTE) .....                                       | 89 |
| ILUSTRACIÓN 54 - ESTADÍSTICAS HAPROXY .....                                | 92 |
| ILUSTRACIÓN 55 - CONSUMO DE SESIONES (WINDOWS/LINUX) .....                 | 94 |



|  |    |
|--|----|
| ILUSTRACIÓN 56 - ESCENARIO INICIAL HA (1/2)..... | 95 |
| ILUSTRACIÓN 57 - ESCENARIO INICIAL HA (2/2)..... | 96 |

# Índice de Tablas

---

|   |    |
|---|----|
| TABLA 1 – COMPARATIVA PLATAFORMAS (1/4).....        | 23 |
| TABLA 2 - COMPARATIVA PLATAFORMAS (2/4).....        | 23 |
| TABLA 3 - COMPARATIVA PLATAFORMAS (3/4).....        | 24 |
| TABLA 4 - COMPARATIVA PLATAFORMAS (4/4).....        | 24 |
| TABLA 5 - CARACTERÍSTICAS XS-SERVER-[0..2].....     | 34 |
| TABLA 6 - CARACTERÍSTICAS XS-NAS.....               | 34 |
| TABLA 7 - INSTALACIÓN XENSERVER (VIÑETAS 1-8).....  | 43 |
| TABLA 8 - INSTALACIÓN XENSERVER (VIÑETAS 9-14)..... | 44 |
| TABLA 9 - CLASES Y EJEMPLOS EN LA CLI.....          | 53 |
| TABLA 10 - CÁLCULO DE SESIONES.....                 | 94 |
| TABLA 11 – EVENTOS EN LA PRUEBA HA.....             | 97 |





# 1. Introducción

---

## 1.1. Motivación

Siempre es deseable trabajar dónde y a la hora que queramos. Para hacer esto posible, desde el punto de vista de una informática tradicional, deberíamos llevarnos el equipo a todas horas y a todas partes. El problema es que esta solución puede ser insuficiente si además deseamos trabajar con programas que hacen uso de licencia (que no poseemos) o de máquinas con unas características distintas de las que disponemos para nuestro uso personal (capacidad de cómputo elevado, sistema operativo distinto al de nuestro PC, conexión restringida a algunas redes, etc.).

Aún si descartamos estos problemas y nos concentrásemos en supuestos empresariales en los que los empleados que lo necesitan disponen de un PC sin la problemática expuesta en el párrafo anterior (supuesto ideal sin contar el lastre económico de dotar un equipo portátil a cada miembro de la organización y que habrá que renovar con una frecuencia razonable), no estaríamos exentos de problemas relativos a actualizaciones de software o fallos de hardware, y en el caso de no contar con un equipo de sustitución estaríamos perdiendo un tiempo adicional.

Trasladando lo anterior a un ámbito docente como la universidad, en el que nos encontramos con un número elevado de alumnos (usuarios) que necesitan para desarrollar su actividad un ordenador que cuente con una serie de programas, no podemos ofrecer una solución que consista en dotar a cada alumno con un equipo: en lugar de esto dotamos a las escuelas o departamentos con aulas en las que los alumnos pueden acceder en un horario determinado. En estas salas o laboratorios, los alumnos pueden disponer de equipos con la configuración y las licencias correspondientes para usar el software que deseen, pero no pueden acceder siempre que quieran. Por otra parte, están obligados a trabajar en un lugar concreto, y esto es una restricción que solo alberga incomodidades y ofrece pocas ventajas.

En este trabajo presentamos una arquitectura que permite acceder en cualquier momento a estos recursos software de forma virtual y completamente transparente al usuario. Este servicio pretende ser una solución que elimine la dependencia temporal y de localidad que impone este tipo de salas, permitiendo una disponibilidad permanente



con un coste reducido, y que solo obliga al usuario a disponer de un dispositivo con acceso internet.

La solución propuesta posee: alta escalabilidad, al mantener la eficiencia con aumento del número de servidores; alta disponibilidad, gracias a ser un sistema por definición redundante; balanceo de carga, al repartir el flujo de trabajo según reglas prefijadas o dinámicas; y permite un mantenimiento transparente, que además ofrece una rápida recuperación frente a fallos hardware o software y que en la mayor parte de los casos deja al usuario ajeno a estos problemas.

## **1.2. Objetivos del trabajo**

El objetivo global de este trabajo es implementar una solución que permita a un grupo de usuarios trabajar de forma remota en ordenadores con unas características dadas y consiguiendo una estructura dotada de alta escalabilidad, alta disponibilidad, balanceo de carga y que posibilite un mantenimiento eficiente y una rápida recuperación frente a fallos hardware o software. Para satisfacer este objetivo se fijan como objetivos parciales:

- Diseñar un sistema mediante *XenServer*, utilizando servidores configurados en un *Pool* con un número de servidores virtuales miembro apropiados a las circunstancias, y que éstos ofrezcan los servicios de escritorio remoto.
- Configurar la plataforma para dotarla de autogestión. El sistema deberá tener en cuenta niveles de carga o necesidades de mantenimiento y decidir cuál es el número de máquinas virtuales usadas y cuál es el servidor *XenServer* adecuado para ejecutarse.
- Evaluar la funcionalidad y las prestaciones del sistema diseñado. Nos centraremos en evaluar el tiempo de recuperación frente a fallos hardware y la capacidad de carga en número de sesiones.
- Realizar el proyecto haciendo uso de software libre, o al menos gratuito.

## **1.3. Estructura del trabajo**

El resto de este trabajo está organizado como sigue:

En el capítulo dos se introduce la virtualización y se mencionan las características de las diferentes soluciones que han desarrollado las principales empresas. Se expone con un poco más de detalle la arquitectura *XenServer*, la solución seleccionada para este TFG.

El capítulo tres presenta el modelo del trabajo: SERP (Servicio de Escritorio Remoto en Pool). Se presentan sus cuatro componentes fundamentales: Centro de Almacenamiento, Centro de Proceso, Bolsa de Servicios y Agente. Cada uno de los cuatro se abordará sucesivamente y en ese orden en los capítulos tres, cuatro, cinco y seis.

El capítulo cuatro se desarrolla el Centro de Almacenamiento. Presenta su configuración y la red de almacenamiento, explicando con detalle como ofrece su servicio.

El capítulo cinco se dedica al Centro de Proceso. Se muestran las utilidades de configuración del sistema y el trabajo realizado con ellas. En él se integra el conjunto y la funcionalidad de SERP.

El capítulo seis se presenta la Bolsa de Servicios y cómo han sido configurados las máquinas virtuales (servicios *Windows* y *Linux*) a las que los usuarios pueden conectarse remotamente.

En el capítulo siete se presenta el Agente. Se explica su configuración y el servicio que presta.

En el capítulo ocho se presentan las pruebas realizadas en SERP para comprobar el desempeño de la solución propuesta.

Finalmente, el capítulo nueve presenta las conclusiones de este trabajo e indica algunas líneas de trabajo futuras.







## 2. Tecnologías de Virtualización

---

La virtualización es un término que desde hace mucho tiempo nos acompaña en la informática en diferentes ámbitos, aunque la RAE no lo reconozca en el diccionario [1]. En el caso que nos ocupa, nos referiremos a ella como a la creación o migración de sistemas, dispositivos, servidores o recursos informáticos a entornos virtuales. En este capítulo se introduce el término de hipervisor y después se presentan brevemente los principales productos del mercado, soluciones tanto gratuitas como de pago. Posteriormente se presenta *XenServer*, la plataforma elegida para el desarrollo de este TFG.

### 2.1. Virtualización

El término virtualización en sí, hace referencia a que algo se comporta como, o parece, algo que no es. Decir que dos cosas son virtualmente iguales es decir que dos cosas son distintas, pero que, en alguna característica o uso, la diferencia no es apreciable o ni siquiera existente. En [2] se hace una definición perfecta para este caso: *Virtualization in computing often refers to the abstraction of some physical component into a logical object*. Este trabajo consiste en abstraer la idea de laboratorio (referido a las máquinas físicas) y convertirlo en un servicio (lógico).

El componente fundamental de la virtualización son las máquinas virtuales. Son, siguiendo las definiciones anteriores, algo que se comporta como una máquina real, pero que no lo es. Se podría definir como un software que simula a un ordenador y que ejecuta aplicaciones como si fuera un ordenador real. La definición que podría ser más correcta y que introduce el término fundamental en este capítulo sería: Una máquina virtual es el contenedor de un sistema que se ejecuta con un hipervisor sobre una máquina.

El término hipervisor tampoco está reconocido en el diccionario, pero se puede definir como software, *firmware* o hardware que gestiona las máquinas virtuales.

El concepto de hipervisor evolucionó desde *VMM (Virtual Machine Monitor)*. A finales de los sesenta, *IBM* utilizaba un software para desarrollar y depurar sistemas operativos en sus mainframes. Este software, con estilo *sandbox*<sup>1</sup> y precursor de las *VM (Virtual Machine)*, permitía a los programadores realizar un desarrollo rápido sin hacer uso de

---

<sup>1</sup> Ejecutar programas en entornos separados asegurando que la estabilidad está protegida



todos los recursos hardware y accediendo a zonas de memoria que no eran normalmente accesibles para los programadores. Este software fue evolucionando y permitiendo ejecutar más de una instancia concurrentemente, algo muy necesario dado el coste del hardware en aquella época. Debido a que al sistema operativo se le llamaba supervisor por su labor con los recursos del sistema y a que este software era superior en jerarquía se acuñó el término hipervisor.

Los hipervisores actuales realizan una abstracción a distintos niveles incluyendo el conjunto de instrucciones, la capa de hardware (*HAL, Hardware Abstraction Layer*), los niveles de llamada al sistema operativo, el espacio de usuario y el nivel de aplicación. La forma de resolver esta abstracción determinará el tipo de hipervisor y de virtualización utilizada.

## 2.2. Tipos de virtualización

Como se indica en [3], desde el punto de vista de las aplicaciones o los usuarios, una *VM* tiene los atributos y características de un sistema físico, pero es estrictamente un software que lo emula. El programa que a bajo nivel provee a las *VM* de acceso a los recursos del sistema es el hipervisor. Actualmente se clasifican en dos tipos en función de la relación entre el hipervisor y el sistema operativo, dividiendo a los hipervisores en tipo 1 y tipo 2 (Ilustraciones 1 y 2). Existe definido un tercer tipo, tipo 0, del que no hay implementación [4].

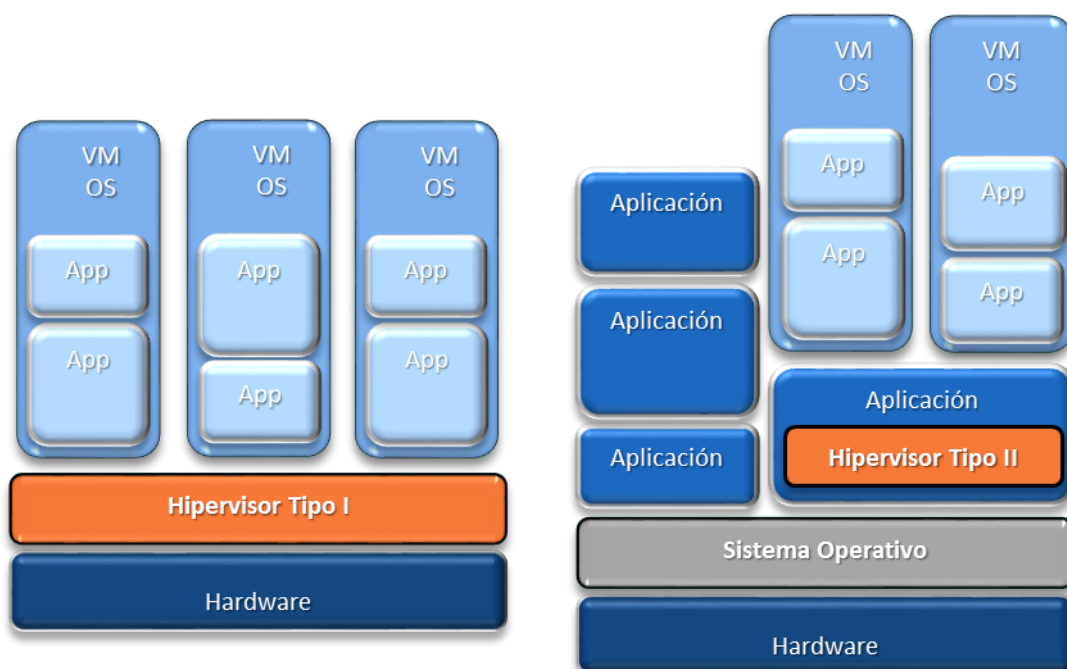


ILUSTRACIÓN 1 - HIPERVISOR TIPO 1

## ILUSTRACIÓN 2 - HIPERVISOR TIPO 2

### 2.2.1. Hipervisor tipo 1

Los hipervisores de tipo 1 son también llamados *bare-metal*, nativos o *unhosted*. En este modelo el hipervisor descansa directamente sobre el hardware de la máquina, teniendo acceso a todos los recursos (el hipervisor se carga antes que ninguno de los *OS* invitados, y él gestiona todos los accesos directos al hardware).

### 2.2.2. Hipervisor tipo 2

Los de tipo 2 son análogamente llamados hospedados (*hosted*). En este modelo el servidor tiene instalado un sistema operativo sobre el cual se ejecuta el hipervisor. De esta forma, el hipervisor pasa a ser un proceso del sistema operativo, y las *VM* trabajan en un nivel exterior.

### 2.2.3. Emulación, paravirtualización y virtualización completa

Además de la clasificación anterior existe otra determinada por la interacción con el hipervisor y el hardware o la dependencia del sistema operativo instalado en las *VM*.

Se habla de emulación cuando la máquina virtual simula el hardware y no tiene acceso directo a el hardware real. Esta configuración tiene como ventaja que el sistema operativo instalado en la *VM* no necesita ser modificado, pero en contrapartida el rendimiento quedará mermado respecto a otros modelos que sí puedan interactuar directamente con el hardware sin tener que usar una capa de traducción para sus llamadas.

En cambio, en la técnica conocida como paravirtualización se provee de una interfaz que permitirá el acceso al hardware. Este tipo de configuración requiere que el *OS* de la *VM* sea modificado.

El último de los métodos, virtualización completa, solo está disponible para hipervisores del tipo 1, y permite el acceso de las *VM* directamente al hardware del sistema. Esto es posible gracias al soporte hardware que los procesadores de los últimos años han incorporado, aumentando el conjunto de instrucciones para proveer de un acceso rápido tanto a la memoria como a la E/S. Tanto *Intel* como *AMD* (*Advanced Micro Devices*) han



desarrollado diferentes instrucciones de soporte, siendo las primeras y las más conocidas las denominadas *Vanderpool* y *Pacifica*<sup>2</sup>.

## 2.3. Plataformas de virtualización

En este punto se hace una breve introducción a las plataformas de virtualización más utilizadas: *VMWare vSphere*, *Hyper-V*, *KVM* y *XenServer*. Tras una breve reseña de cada una de ellas se comentan sus características de forma global y se presentan en una tabla comparativa. Todas las soluciones aquí expuestas trabajan como hipervisores de tipo 1.



ILUSTRACIÓN 3 - LOGOTIPOS PLATAFORMAS VIRTUALIZACIÓN

### 2.3.1. vSphere

Actualmente *VMWare*, con su producto *vSphere*, es el líder del mercado de la virtualización; no en vano es la empresa que tiene la trayectoria más larga como empresa dentro del negocio de la virtualización de sistemas. Ofrece varias versiones/paquetes empresariales y necesita de *VMWare vCenter* para su funcionamiento (*vCenter* es el centro de control donde se despliegan las herramientas soportadas por *vSphere*). Es quien ofrece un mayor número de características en una infraestructura que pretende dar servicio a clientes de cualquier envergadura. Según su propio *CEO* [5], en marzo de este año anunciaba tener un 80% del mercado (en entornos de acceso privado).

### 2.3.2. Hyper-V

*Hyper-V* es la plataforma de virtualización presentada e incluida por *Microsoft* en *Windows server 2008* y versiones sucesivas (antes de *Hyper-V*, *Microsoft* solo había aparecido vagamente en el horizonte de la virtualización con su *VirtualPc* y *VirtualServer*). Su última versión está disponible en las versiones de *Windows server 2012 R2 standard* y *Windows server 2012 R2 Datacenter* (en lo que respecta a *Hyper-*

---

<sup>2</sup>“Vandepool” o *Intel-VT* (*Intel virtualization technology*) y en *AMD* “*Pacifica*” o *AMD-V* (*AMD Virtualization*).

V es el mismo producto, solo cambia el número de sistemas que soporta por licencia). Actualmente es el segundo en volumen de mercado y también el que más se acerca a *VMWare* en prestaciones.

### 2.3.3.KVM

Las siglas de *KVM* vienen de *kernel-based Virtual Machine*, y como su nombre indica es una infraestructura basada en el *kernel* de *Linux*. *KVM* forma parte del núcleo de *Linux* (desde 2007) y se podría decir que es la virtualización nativa de *Linux*. Gracias a su integración en el núcleo es capaz de acceder al hardware sin necesidad de ningún *driver*. Es la solución más joven de las cuatro y está impulsada por *Red Hat*.

### 2.3.4.Xen Project

*Xen Project* es el líder en plataformas de virtualización *open source*: *Amazon Web Services*, *Aliyun*, *Rackspace Public Cloud* y *Verizon Cloud* entre otras, han confiado su servicio al hipervisor *Xen*. *Citrix* compró *XenSource Inc.* en 2007, y desde entonces da soporte a *Xen Project* y usa el hipervisor en sus productos comerciales, y en el caso de *XenServer* lo ofrece de forma gratuita (sin soporte).

### 2.3.5.Características de las plataformas

En este punto se repasan las características más relevantes de las diferentes plataformas.

Los requisitos de las distintas plataformas son similares, y todas se benefician de procesadores de 64 bits y de su soporte de virtualización. Particularmente, *Xen* no necesita el soporte de instrucciones cuando usa la paravirtualización, y *vSphere* necesita que se emplee un segundo servidor dedicado para *vCenter*, y que la máquina (virtual o no) disponga al menos de 2 procesadores o un doble núcleo a 2GHz, 2GB de *RAM* y red con interfaz Gigabit.

*vSphere*, *Hyper-V* y *XenCenter* pueden trabajar funcionalmente a nivel de clúster (en *XenServer* se utiliza el término de *Resource Pool*). Esto, como se verá en el punto 2.4, proporciona grandes ventajas. El número de *hosts* y *VM* dentro de una agrupación también estará restringido a unos máximos que se pueden consultar en la tabla 1.

Respecto al número de *CPU* soportadas hay que aclarar que *vSphere* es la única que tiene soporte a nivel de *socket*, no de hilo lógico. Esto quiere decir, por ejemplo, que un equipo con 4 procesadores que contengan 6 núcleos y soporten *HyperThreading*<sup>3</sup> o alguna

---

<sup>3</sup> Tecnología implementada en procesadores *Intel* que permite dos hilos de procesamiento en cada núcleo del procesador.



tecnología similar tendría un peso de 4 (*sockets*) para *vSphere* y de 48 (procesadores lógicos) para el resto. La tabla 2 muestra la gran diferencia que existe entre *vSphere* y el resto de las plataformas.

*vSphere*, *Hyper-V* y *XenServer* disponen de interfaz gráfica (todas bajo *Windows*), respectivamente *vCenter Server*, *System Center 2012R2* y *XenCenter*. *KVM* no dispone de herramienta propia, pero se puede conectar a sus hipervisores mediante *libvirt* (que también tiene soporte para todos los anteriores). *vSphere 6.0 vSphere Web Client Enhancements* permite usar un cliente web para conectarse a *vCenter*.

Otra característica importante es la de poder generar *snapshots* y *templates*. Todas las plataformas permiten realizar estas operaciones online (a excepción de las *templates* para *KVM*) y *vSphere* y *Hyper-V* son capaces de realizar un *deploy*<sup>4</sup> de sistemas *Windows*.

En cuanto a los mecanismos de alta disponibilidad o de recuperación de desastres, todos excepto *KVM* tienen desarrollada en parte o totalmente las características, aunque sea con dependencia de productos de terceros.

Otra característica importante es la migración de *VM* activas (con o sin su almacenamiento). Todas las plataformas permiten la migración de la *VM* en marcha sin pérdida del servicio, y *vSphere* y *Hyper-V* permiten además migrar varias al mismo tiempo. Además, todas excepto *KVM* permiten migrar el almacenamiento con ellas.

Tanto *KVM* como *XenServer* son gratuitas.

En el caso de querer usar *Hyper-V*, la licencia de *Windows Server 2012 R2 Datacenter* tiene un precio aproximado de 4500€ (para cada dos procesadores) y *System Center* 3500€. Cada acceso necesitará además una *CAL*<sup>5</sup> (*Client Access License*).

*vSphere* funciona conjuntamente con *vCenter*. El precio de su licencia es, desde su página, de aproximadamente 6600 €. Las licencias se venden por máquina (hasta dos *sockets*), y hay distintos paquetes. El coste de la licencia estándar está sobre 1200 € para servidores con hasta dos procesadores. En las páginas siguientes se pueden observar las tablas que resumen de los datos expuestos. [6] [7] [8]

---

<sup>4</sup> Nos referimos como tal a la utilización y configuración mediante la herramienta *sysprep* para preparar una correcta clonación del sistema.

<sup>5</sup> Cada usuario que acceda al servicio deberá de tener su propia licencia (o el servidor por cada cliente que se conecte).





|   | <i>CPU/RAM (requisitos)</i>   | <i>CLUSTER/nº de VM</i> | <i>HOSTS<br/>VM/vCPU</i>                        |
|---|---|-------------------------|---|
|  | Instr. Virt., 64bits<br>Servidor adicional<br><i>Dual-Core con 2 GB</i> | 64 Máquinas<br>8000 VM  | 1024 VM<br>2048 vCPU                            |
|  | Instr. Virt., 64bits  | 64 Máquinas<br>8000 VM  | 1024 VM<br>2048 vCPU                            |
|  | Instr. Virt., 64 bits   | NO                      | NO  |
|  | Instr. Virt. y 64 bits<br>para usar virtualización<br>completa          | 16 Máquinas<br>4096 VM  | 500 VM ( <i>win</i> )/<br>650 VM ( <i>lin</i> ) |

TABLA 1 – COMPARATIVA PLATAFORMAS (1/4)





|   | <i>HOSTS<br/>CPU/RAM</i> | <i>VM<br/>vCPU/RAM</i>   |
|---|--------------------------|--|
|  | 480 CPU / 12TB RAM       | 128 vCPU / 4 TB  |
|  | 320 hilos / 4 TB RAM     | 64 vCPU / 1 TB   |
|  | 256 hilos / 1TB RAM      | 16 vCPU / 256 GB   |
|  | 160 hilos / 1 TB RAM     | 16 vCPU / 192 GB ( <i>win</i> )<br>32 vCPU / 192 GB ( <i>lin</i> ) |

TABLA 2 - COMPARATIVA PLATAFORMAS (2/4)





|   | Interfaz gráfico         | vHD        | Snapshots/ Templates |
|---|--------------------------|------------|----------------------|
|  | Propio / cliente web     | 62 TBytes  | Sí / online y deploy |
|  | Propio                   | 256 TBytes | Sí / online y deploy |
|  | Necesita Libvirt u otros | 2 TBytes   | Sí / offline         |
|  | Propio                   | 2 TBytes   | Sí / online          |

TABLA 3 - COMPARATIVA PLATAFORMAS (3/4)





|   | Alta disponibilidad | Movimiento activa | Movimiento de almacenamiento |
|---|---------------------|-------------------|------------------------------|
|  | SÍ                  | SÍ, varias        | SÍ                           |
|  | SÍ                  | SÍ, varias        | SÍ                           |
|  | NO                  | SÍ                | NO                           |
|  | SÍ                  | SÍ                | SÍ                           |

TABLA 4 - COMPARATIVA PLATAFORMAS (4/4)



## 2.4. XenServer

*Citrix XenServer* es, desde la versión 6.2, una plataforma de virtualización de nube, servidores y escritorios, *Open Source* y gratuita, desarrollada en conjunto por una amplia comunidad (proyecto *Xen*) y por *Citrix*, que presenta todas las funcionalidades que anteriormente solo estaban presentes en las versiones empresariales (de pago), y solo unas pocas características adicionales quedan reservadas para las versiones licenciadas con soporte, poniendo a nuestra disposición de forma gratuita una solución potente, que no dista tanto de *vSphere* y *Hyper-V*.

El proyecto *Xen* desarrolla el *Hypervisor Xen*, *open source*. Es un proyecto maduro con 12 años de desarrollo:

- Es parte de la *Linux Foundation* y tiene licencia *GPLv2*.
- Utiliza el proyecto *Open-source Hypervisor Xen*.
- Desarrollo y gestión a cargo de *Linux Foundation* (este hipervisor es el utilizado por *XenServer* como el núcleo de su arquitectura).
- *Citrix* añade otros componentes propios como *XenCenter*, la consola de administración.

El soporte empresarial para esta versión (6.5) de *Citrix XenServer*, así como el que las soluciones de los demás competidores orientados a empresas, se vende por *socket*.

La arquitectura de *XenServer* se compone, de forma similar a los demás entornos de virtualización, de tres partes:

- Servidores miembro (llamados habitualmente *hosts*) con el hipervisor instalado (*XenServer*), que son los que ofrecen, al menos, sus recursos de cómputo a las máquinas virtuales
- Interfaz de administración (*XenCenter* y *CLI*<sup>6</sup>, pero compatible con distintas aplicaciones de terceros).
- Recursos de almacenamiento local o remoto (*SAN*<sup>7</sup>, *NAS*<sup>8</sup>, ...)⁹ en el que se encuentran hospedadas las máquinas virtuales.

---

<sup>6</sup> *Command-Line Interface*

<sup>7</sup> *storage area network*

<sup>8</sup> *network area storage*

<sup>9</sup> *SAN* necesita de una red dedicada. Trabaja a más bajo nivel y su rendimiento y precio es muy superior a las soluciones *NAS*

El hipervisor *XenServer*, crea una capa en el hardware físico de los *hosts* y presenta a las máquinas virtuales un hardware adaptado. Al estar basado en *Xen*, realiza esta abstracción con una máquina virtual en estado privilegiado y asegurado denominado *domo* (también llamado *Control Domain* o *domain0*) que ejecuta *XAPI*, el conjunto de herramientas que gestionan *XenServer*. *XAPI* se encarga de manejar los *drivers* de los dispositivos físicos hardware.

El objetivo de este TFG es diseñar una estructura dotada de alta escalabilidad, alta disponibilidad, balanceo de carga y que posibilite un mantenimiento eficiente y una rápida recuperación frente a fallos hardware o software, y en esta sección se va a mostrar como ayuda la tecnología *XenServer* a cumplir estos objetivos.

#### 2.4.1. Alta escalabilidad

Nuestros servidores van a trabajar de forma coordinada en un *Resource Pool*. Un *Resource Pool* es un agrupamiento lógico de servidores que permite su administración conjunta. Un *Pool* lo integran servidores de similares características y nivel de actualización<sup>10</sup>. Como se puede observar en la tabla 3, *XenServer* tiene capacidad para hasta 16 *hosts*, por lo que las posibilidades de crecer son amplias, y se podría redimensionar el sistema simplemente añadiendo servidores (y la electrónica necesaria). Si con 16 *hosts* todavía se tuviesen mayores necesidades se podrían crear más *Resource Pool* con otros tantos servidores miembro, y gracias a la administración centralizada no sería un problema para la gestión.

El agrupamiento de interfaces de red, es otra posibilidad disponible con *XenServer* que también proporciona un mayor ancho de banda para atender demandas exigentes.

#### 2.4.2. Alta disponibilidad

*XenMotion*<sup>11</sup> permite mover las máquinas virtuales en ejecución entre distintos *hosts* de un *Resource Pool* sin que las aplicaciones se vean afectadas. Para mover una máquina virtual basta, por ejemplo, con hacer un par de *clicks* con *XenCenter* seleccionado la opción de migrar y el *host* de destino. Esta opción estará disponible siempre que el *host* no haya sufrido en error inesperado de hardware o de red y quede fuera de línea. Los *Resource Pool* presentan una arquitectura replicada con modelo maestro – esclavo. El *master* es el *host* a través del cual se administra el *pool*. La información de la

---

<sup>10</sup> Para *XenServer* existen actualizaciones al igual que en los sistemas operativos. Todos los miembros de un *Resource Pool* deben ejecutar la misma versión actualizada.

<sup>11</sup> Denominación que otorga *XenServer* a la característica

configuración del *pool* está replicada, y este hecho permite que, en caso de necesidad, otro *host* cualquiera pueda adoptar el papel de *master* sin que con ello concurra pérdida de funcionalidad alguna (o pérdida de datos). Cuando en un *pool* existe configurado un espacio de almacenamiento compartido entre los *hosts* del *pool* y allí se ubican las máquinas virtuales, estas podrán ser iniciadas en cualquiera de los *hosts* del *pool*, y podrán moverse en “caliente” entre los *hosts*. En este caso, si las características de alta disponibilidad están configuradas y habilitadas, las máquinas virtuales que se encontraban en ejecución en el *host* caído serán iniciadas automáticamente en otro *host*. *HA*<sup>12</sup> se encargará, en caso de que nuestro *host* sea el *master*, de promover a otro *host* a *master*.

### 2.4.3. Mantenimiento y recuperación frente a fallos

Como se ha visto en el punto anterior, *XenMotion* permite mover las máquinas virtuales en ejecución entre distintos *hosts* de un *Resource Pool* sin que las aplicaciones se vean afectadas. Dado que esto resulta transparente al usuario, es posible realizar tareas de mantenimiento sin que afecte al desempeño (el proceso necesita copiar los estados de la máquina y puede ser un proceso de unos 15 segundos, pero solo en el momento de cambiar el anfitrión se provocará necesariamente un paro, que tiene una duración del orden de décimas de segundo y normalmente pasará inadvertido). En *XenServer* existe también una opción llamada modo de mantenimiento. Esta opción provocará que todas las *VM* que se estén ejecutando en ese servidor migren a otros servidores, dejando el *host* deshabitado para poder realizar cambios de configuración sin que afecten a las máquinas en ejecución. El reinicio automático de máquinas virtuales permite tenerlas siempre disponibles ante un fallo a nivel de *host*.

Otra característica que podríamos asociar tanto a la alta disponibilidad como al mantenimiento y la recuperación frente a fallos es un plan de recuperación (*Recovery disaster*). En caso de que el almacenamiento en red falle en un *Resource Pool* existe la posibilidad de que este use un respaldo, permitiendo nuevamente iniciar las máquinas caídas. Las máquinas virtuales son, físicamente, ficheros almacenados en un recurso de almacenamiento. En la terminología de *XenServer* son denominados *Storage Repository (SR)*. Pueden ser locales (discos propios del servidor), o remotos.

Si se necesita cambiar la ubicación de la *VM* se puede usar *Xen StorageMotion*, que es la extensión de *XenMotion* que elimina la restricción de que las máquinas virtuales deban moverse entre *hosts* de un mismo *Resource Pool*. Permite mover los discos de las

---

<sup>12</sup> *HA (High availability)*



máquinas virtuales en funcionamiento (junto con su entorno de ejecución) entre distintos *Resource Pool*, *SR* o incluso a un almacenamiento local.

También destacar que para detectar fallos y realizar diagnósticos de rendimiento *XenServer* integra una herramienta que proporciona métricas del rendimiento de *CPU*, memoria, disco, red, y almacenamiento entre otras.

#### 2.4.4. Balanceo de carga

*XenServer* cuenta con una característica para el balance de carga que está solo disponible para la versión empresarial, y que no podremos usar (se satisfará este objetivo por medio de otra herramienta). En cualquier caso, tiene otras cualidades que permiten de forma natural repartir la carga y los recursos. Al igual que en las demás plataformas de virtualización, para obtener el mayor partido es necesario instalar un software en las máquinas virtuales una vez que hayan sido instaladas y se encuentren en ejecución (de la misma forma que se instalaría una aplicación cualquiera). Este software, además, proporciona información adicional que podremos consultar a través *XenCenter*. Este software se denomina *XenServer Tools* y proporciona los *drivers* necesarios para el correcto funcionamiento del sistema operativo invitado y el hipervisor. Este paquete está disponible para varias plataformas (entre las que se encuentran las que hemos utilizado).

Cuando *XenServer Tools* está instalado se puede compartir la memoria de un *host* sin utilizar entre las máquinas virtuales, optimizando el rendimiento. Si configuramos la cantidad de memoria *RAM* de las máquinas virtuales con un intervalo en lugar de un valor concreto, estaremos diciéndole al *host* que, en caso de necesidad, puede disponer de la *RAM* no esencial (de la cantidad asignada por encima de su mínimo) según entienda conveniente. Esto facilita, entre otras cosas, la migración de máquinas al menor número de *hosts* posibles apagando los que no se encuentran en uso.

#### 2.4.5. Administración

Además de las características anteriores *XenServer* posee otras importantes para la gestión, como por ejemplo el manejo de *Snapshots* (instantáneas) y *Templates* (plantillas).

En el transcurso del ciclo de vida de nuestras máquinas virtuales existen muchos motivos por los que podemos desear salvaguardar su estado: realizar una actualización, instalar un programa, hacer cambios de configuración importante. En estos casos, si por alguna razón el resultado no resulta satisfactorio podríamos desear volver al estado anterior. El uso de los *snapshots* nos ofrece esta posibilidad (y mucho más). Podemos crear instantáneas de nuestra máquina y volver atrás. Además, de la forma que está

implementado, solo se hace una copia diferencial de la máquina objetivo, por lo que el espacio en disco se minimiza y no es un problema realizar la cantidad de instantáneas que queramos (incluso es posible copiar el estado exactamente en el momento de ejecución). Un *snapshot* guarda el almacenamiento, la configuración y la información de red de la máquina virtual, por lo que es muy útil para hacer un *backup* de la misma.

Para crear VM se usa un asistente (en *XenCenter*, pero también podemos crearlas en línea de comandos). En el proceso determinamos el sistema operativo y le facilitamos el soporte de instalación del mismo (que podemos tener disponible en un *SR*). Este método nos permite crear una máquina limpia, en la que será necesario instalar todo el software que necesitemos (empezando por la instalación del sistema operativo), pero es muy posible que tengamos una idea del servicio que le vamos a dar y sería de utilidad disponer de una serie de aplicaciones o, simplemente, tener el sistema instalado con los parches y no de origen. Para todo esto podemos hacer uso de las plantillas. Las plantillas nos replicarán una máquina tipo ahorrándonos la mayoría del tiempo que de otra forma hubiéramos invertido. Estas plantillas las obtendremos generándolas a partir de nuestras máquinas o de nuestros *snapshots*, y solo nos quedará hacer los pequeños ajustes de red, nombre o pequeños servicios que debamos modificar ahorrando mucho tiempo. En las máquinas con sistema operativo *Windows*, en la mayoría de los casos, deberemos ejecutar previamente la utilidad *sysprep* antes de convertir la máquina en plantilla.

*XenServer* se puede integrar con varios directorios y asignar permisos a usuarios. Así, se obtiene un acceso de seguridad por capas a los distintos objetos, como VM. Esto permite mejorar la seguridad y delegar el uso y gestión de *XenServer*, al proporcionar una seguridad basada en niveles de permisos.

Por último, además de la consola de administración *XenCenter*, que como ya se ha explicado, permite una gestión centralizada a todos los niveles, no se puede olvidar que la interfaz de línea de comandos, con sus comandos *xe* disponibles en la interfaz local de cada servidor (o por *ssh*), dan acceso a todas las tareas que puedan hacerse desde *XenCenter* y más. Otras opciones de administración son la consola del servidor físico, que cuenta con un menú que permite realizar tareas relativas a *backups*, *resource pools*, *networking* y otras, y el *Pssnapin* de *PowerShell*, contenido en el *XenServer SDK*.

Como se ha visto, *XenServer* es una herramienta muy completa y será parte fundamental del modelo presentado en el siguiente capítulo.





### 3. Servicio de Escritorio Remoto en Pool

En este capítulo se presenta SERP (Servicio de Escritorio Remoto en Pool). SERP es el modelo propuesto que definimos, y conceptualmente se basa en cuatro componentes que podemos ver en la ilustración 4.

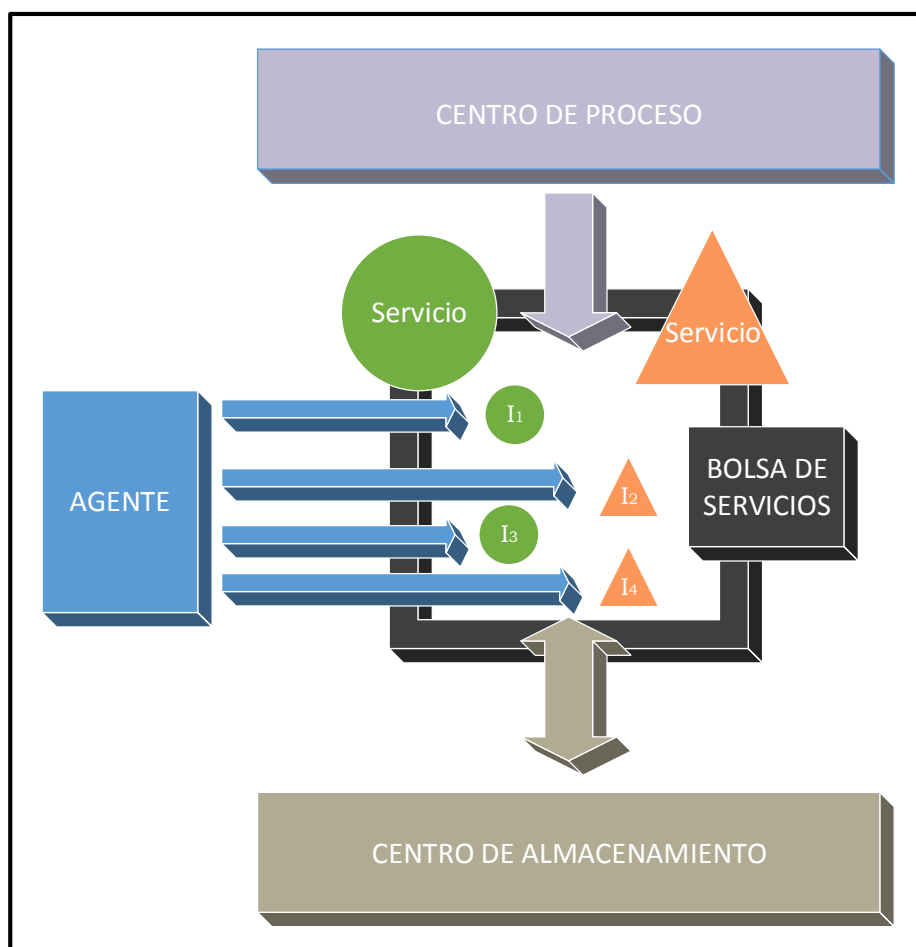


ILUSTRACIÓN 4 - MODELO SERP

- El Agente, el gestor de la conexión. Es el encargado de establecer la conexión entre el usuario y el servicio. Se encarga de elegir la instancia apropiada.
- La Bolsa de Servicios, en nuestro caso servicios de escritorio remoto. Configura las instancias de los servicios ofertados.
- El Centro de Proceso, es el centro de virtualización. Ejecuta las instancias y las dota de recursos. Centraliza la gestión de SERP.
- El Centro de Almacenamiento, ofrece el espacio de almacenamiento al sistema.

A lo largo de este capítulo introduciremos los conceptos de los componentes del modelo, para en capítulos sucesivos entrar en la implementación técnica realizada. Presentaremos los elementos en el siguiente orden: Centro de Almacenamiento (3.1), Bolsa de Servicios (3.2), Centro de Proceso (3.3) y el Agente (3.4). Por último, en 3.5 mostramos un esquema de los equipos reales usados en SERP.

### **3.1. Centro de Almacenamiento**

Uno de los pilares de la arquitectura de *XenServer* es el almacenamiento en red, ya que es básico para poder dotar al sistema de las cualidades de alta disponibilidad y de un mantenimiento eficiente. Su función es sencilla y no requiere de muchos recursos, pero a la vez fundamental: ser el soporte en el que serán almacenadas las *VM*. En el capítulo 4 se aborda con detenimiento.

### **3.2. Centro de Proceso**

El Centro de Proceso centraliza la gestión de SERP. Está fuertemente conectado con todo el sistema y es el responsable de alcanzar y mantener la mayoría de los objetivos. En el capítulo 5 se muestra como se configura para que desarrolle las funcionalidades que se describen en la sección 2.4.

### **3.3. Bolsa de Servicios**

Como se ha visto en la ilustración 4, en la parte superior de la figura que representa la Bolsa de Servicios, aparecen dos figuras geométricas etiquetadas con el nombre "Servicio" y en el centro de la ilustración otras más pequeñas etiquetadas con una "I" y un número. Las dos figuras grandes representan a las dos plataformas que van a estar disponibles para los usuarios (los servicios de escritorio remoto en *Windows* y en *Linux*) y las pequeñas son las instancias a los que los usuarios se conectarán. Si profundizamos más, y lo relacionamos con el capítulo 2 de este trabajo, es posible identificar esos elementos con las plantillas y con las máquinas virtuales que se generan a raíz de ellas. La preparación de los servicios de escritorio remoto para conexiones sobre sistemas *Windows* (con *Windows server 2012 R2*) y en sistemas *Linux* (con *CentOS 7*) se explica en el capítulo 6.



### 3.4. Agente

El Agente es el componente que, aplicando unos criterios adecuados, nos permite balancear la carga entre las máquinas virtuales. En el capítulo 7 se muestra una implementación del agente con un programa balanceador.

### 3.5. Equipos de SERP

La ilustración 5 nos muestra los equipos empleados. Vemos el Centro de Proceso formado por los servidores *XenServer* (equipos *XS-server-0*, *XS-server-1* y *XS-server-2*) que se encuentran conectados a internet (red pub) a través de la red de la UPV y cómo se conectan con el centro de almacenamiento (*XS-nas*) mediante una red privada. También aparecen integrados el Agente, la Bolsa de Servicios e Instancias que, aunque no formen parte del sistema físico se han colocado para hacer el sistema completo más comprensible.

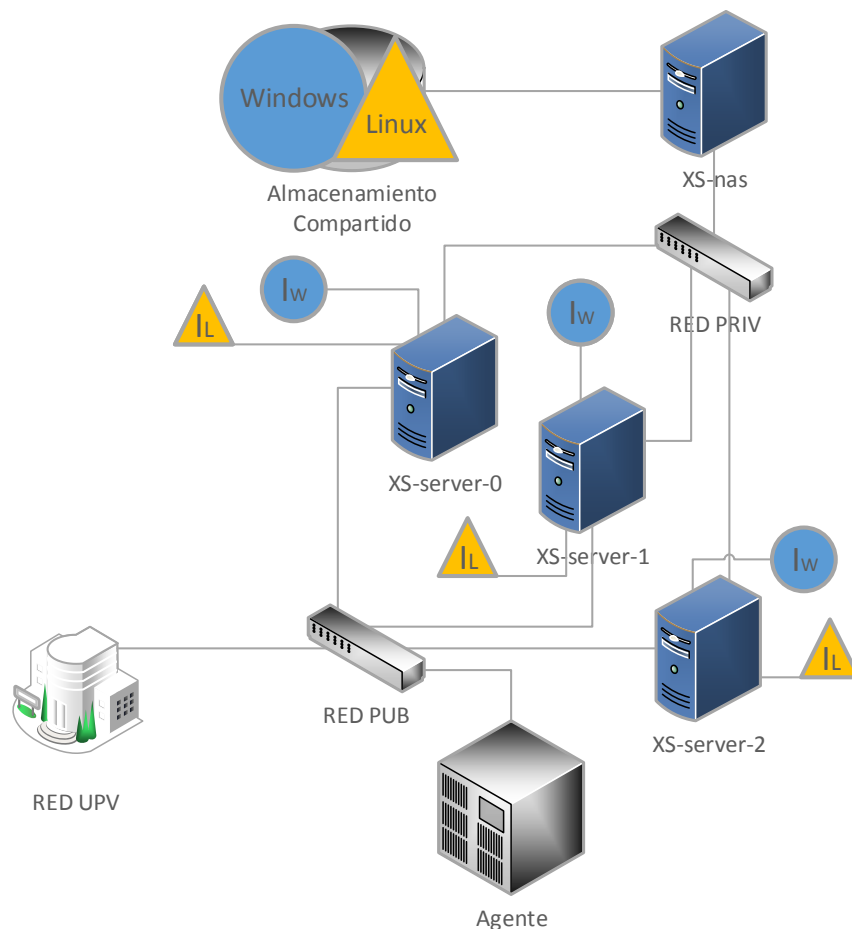


ILUSTRACIÓN 5 - SISTEMA IMPLEMENTADO

En la tabla 5 se muestra la configuración hardware de los equipos *XS-server-0*, *XS-server-1* y *XS-server-2* (Servidores *XenServer*):

|                       |                           |   |
|-----------------------|---------------------------|---|
| <b>CPU</b>            | <i>AMD Phenom x 3 705</i> | <i>Tri-Core 2500 Mhz</i>                                    |
| <b>RAM</b>            | 8 GB                      | 4 x 2 GB <i>DDR2 800MHz</i>                                 |
| <b>ALMACENAMIENTO</b> | 250 GB                    | <i>Seagate SATA2</i>  |
| <b>TARJETAS RED</b>   | Integrada<br>PCIe         | <i>MCP77 nVidia Gigabit</i><br><i>82541PI Intel Gigabit</i> |

TABLA 5 - CARACTERÍSTICAS XS-SERVER-[0..2]

Y en la tabla 6 a *XS-nas* (el servidor NAS):

|                       |                           |   |
|-----------------------|---------------------------|---|
| <b>CPU</b>            | <i>AMD Phenom x 3 705</i> | <i>Tri-Core 2500 Mhz</i>                                    |
| <b>RAM</b>            | 8 GB                      | 4 x 2 GB <i>DDR2 800MHz</i>                                 |
| <b>ALMACENAMIENTO</b> | 250 GB<br>2 x 1TB         | <i>Seagate SATA2</i><br>2 x <i>WD SATA2</i>                 |
| <b>TARJETAS RED</b>   | Integrada<br>PCIe         | <i>MCP77 nVidia Gigabit</i><br><i>82541PI Intel Gigabit</i> |

TABLA 6 - CARACTERÍSTICAS XS-NAS

En este capítulo hemos mostrado de una forma general cómo está implementado SERP; tanto de forma conceptual, repasando los cuatro componentes fundamentales, como en su forma física, mostrando los equipos que lo integran y cómo están interconectados. En los capítulos 4 al 7 se desgranarán los diferentes componentes mostrando como están integrados.

## 4. Centro de Almacenamiento

Para poder implementar SERP es fundamental disponer de almacenamiento en red. Una manera sencilla de implementarlo es mediante otro servidor que ofrezca su almacenamiento a los servidores miembro. Siguiendo con la filosofía del proyecto que intenta usar software libre se ha optado por usar *Open Media Vault (OMV)* [9].

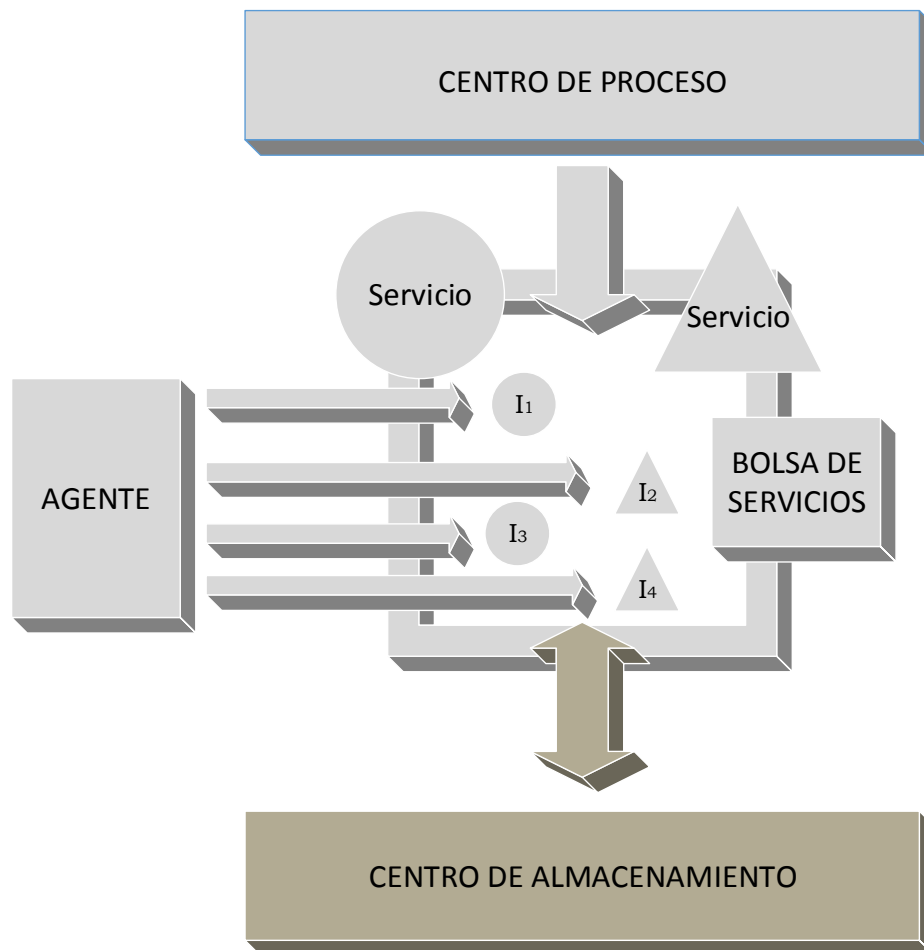


ILUSTRACIÓN 6 –SERP (CENTRO DE ALMACENAMIENTO)

### 4.1. NAS - Open Media Vault

*OMV* es una distribución basada en *Debian* que implementa funcionalidades *NAS* y tiene una herramienta gráfica de gestión basada en web (aunque también tiene disponible *CLI* para poder administrarlo). Existen multitud de herramientas gratuitas muy similares y más completas, pero al hacer un uso básico del mismo no era determinante su elección. Se ha elegido por seguir con las directrices *GPL*, poseer un entorno amigable para su configuración y estar basado en una distribución conocida (*Debian*). En la ilustración 7

podemos observar el aspecto de la herramienta. En esta vista, gracias a las gráficas y estadísticas de rendimiento, se observa fácilmente el estado del sistema.

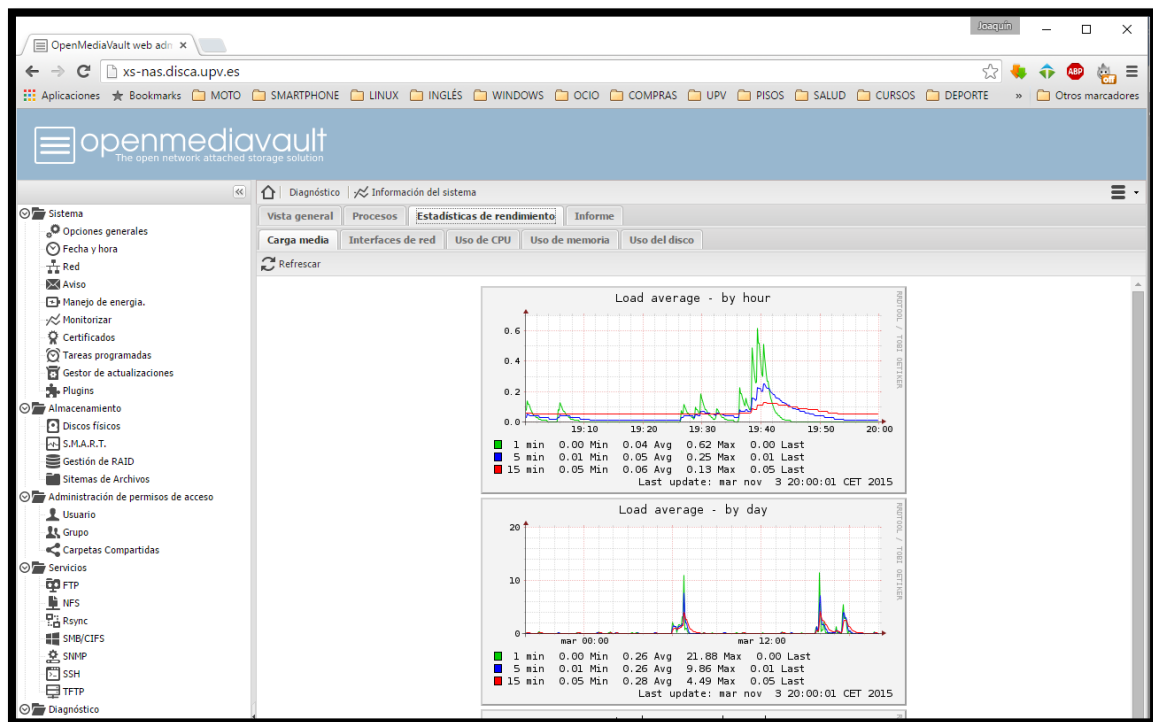


ILUSTRACIÓN 7 - APLICACIÓN WEB DE CONFIGURACIÓN DE OMV

## 4.2. Instalación

El proceso de instalación no supone ninguna complicación. El sistema es una distribución *Linux* que se puede instalar arrancando el servidor con un *CD* o *USB*, y el único paso determinante en el que hay que prestar atención es el que se determina el disco donde instalar el sistema, puesto que se instalará donde se le indique y no se podrá cambiar posteriormente. En el sitio web [9] tenemos disponible tanto el software como manuales para ahondar más en alguna de sus funcionalidades. Una vez instalado *OMV*, nos conectaremos con un navegador e introduciremos la contraseña de la cuenta por defecto.

## 4.3. Configuración RAID 0

El primer paso que debemos seguir para exportar su almacenamiento es crearlo. Como vimos en la tabla 6 del punto 3.5, la máquina contiene (ilustración 8) tres discos duros: uno de ellos se dedica íntegramente al sistema, y en los otros dos formaremos en *RAID 0* (*Redundant Array of Independent Disks*) para asegurar el máximo rendimiento posible. En un sistema real, esta configuración, al aumentar el riesgo de pérdida de datos,

no es adecuada, pero dado que el hardware que se dispone para SERP es bastante limitado, se decidió que era una opción a considerar. En este *RAID 0* exportaremos dos carpetas: *MVs*, en la que almacenaremos las *VM*, y *ISOs*, donde colocaremos las imágenes de disco y software que deseemos que esté disponible para el SERP.

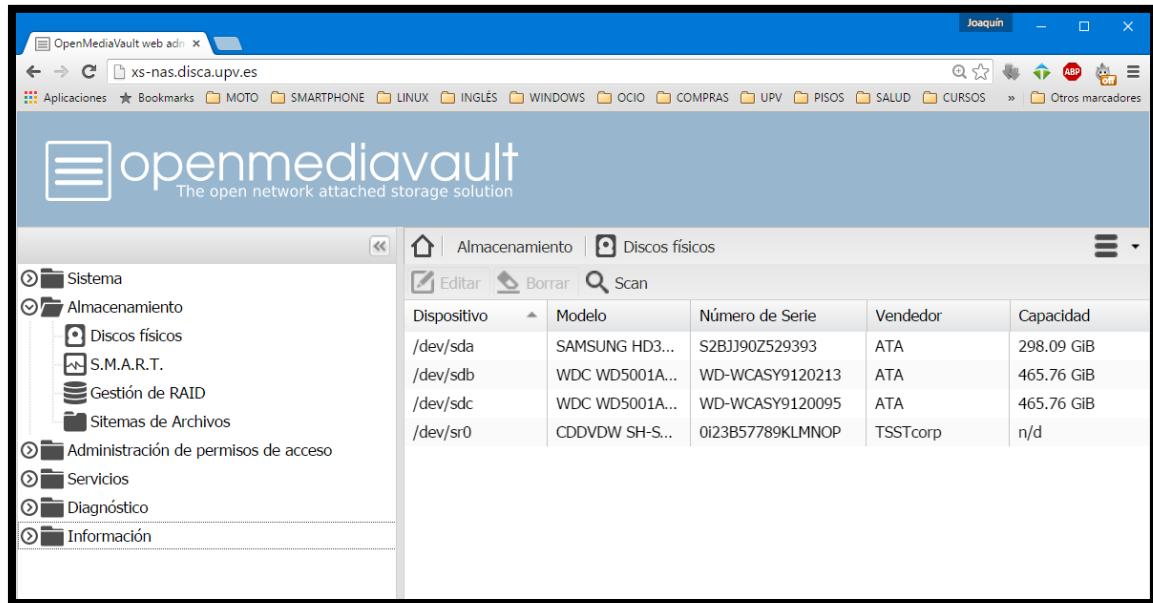


ILUSTRACIÓN 8 - DISCOS DEL NAS

A través de la herramienta web (Almacenamiento -> Gestión Raid) se ha configurado usando los dos volúmenes libres y seleccionando RAID 0 (se reconoce en la ilustración 9 con el nombre de *stripe*). Una vez construido se ha seleccionado la carpeta */export* como punto de montaje.

#### 4.4. NFS y recursos exportados

OMV nos ofrece la posibilidad de exportar su espacio a través de diferentes servicios. En nuestro caso se ha elegido el protocolo *NFSv4*, aunque también podríamos haber optado por *SMB*. Como se puede comprobar en el punto 3.5, en las tablas 5 y 6, todos los equipos integran tarjetas ethernet Gigabit, por lo que el desempeño de la red no debería limitar el funcionamiento del sistema.

Para activar el servicio recurrimos a la herramienta web (Servicios -> NFS). Como se observa en la ilustración 10, se han creado los dos recursos que estarán accesibles para SERP. Estos recursos son */ISOs* y */VMs*. Se han creado ambos con permisos de lectura/escritura para *root*. Estos recursos se montan en */export*

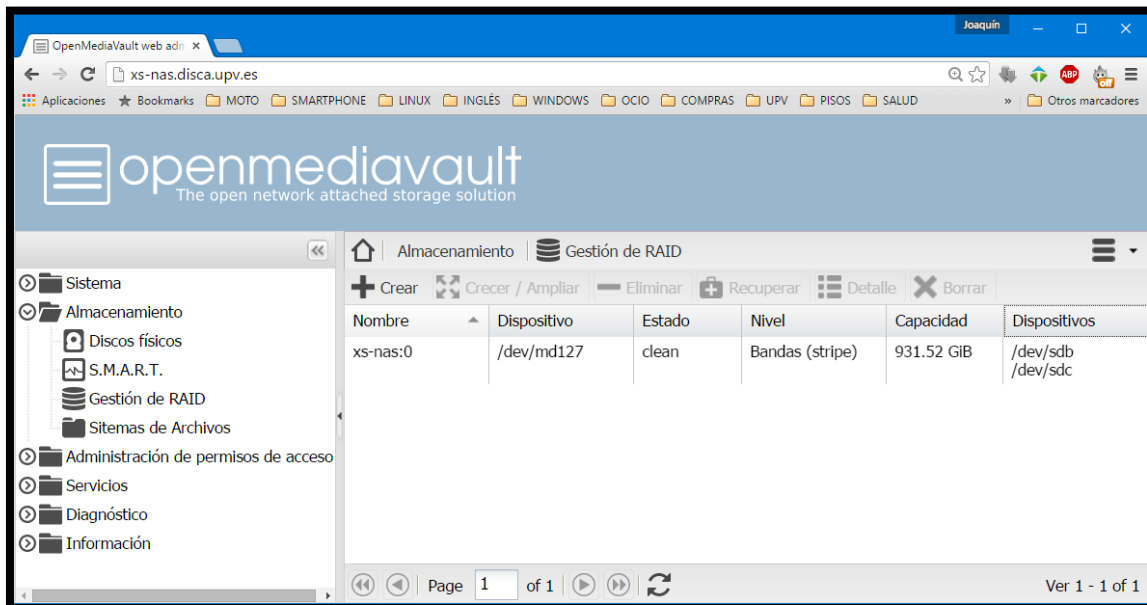


ILUSTRACIÓN 9 - RAID 0 (STRIPE)

## 4.5. Otros servicios y configuraciones

Además de las acciones descritas en los puntos anteriores, se ha activado el servicio `ssh` (Servicios ->SSH).

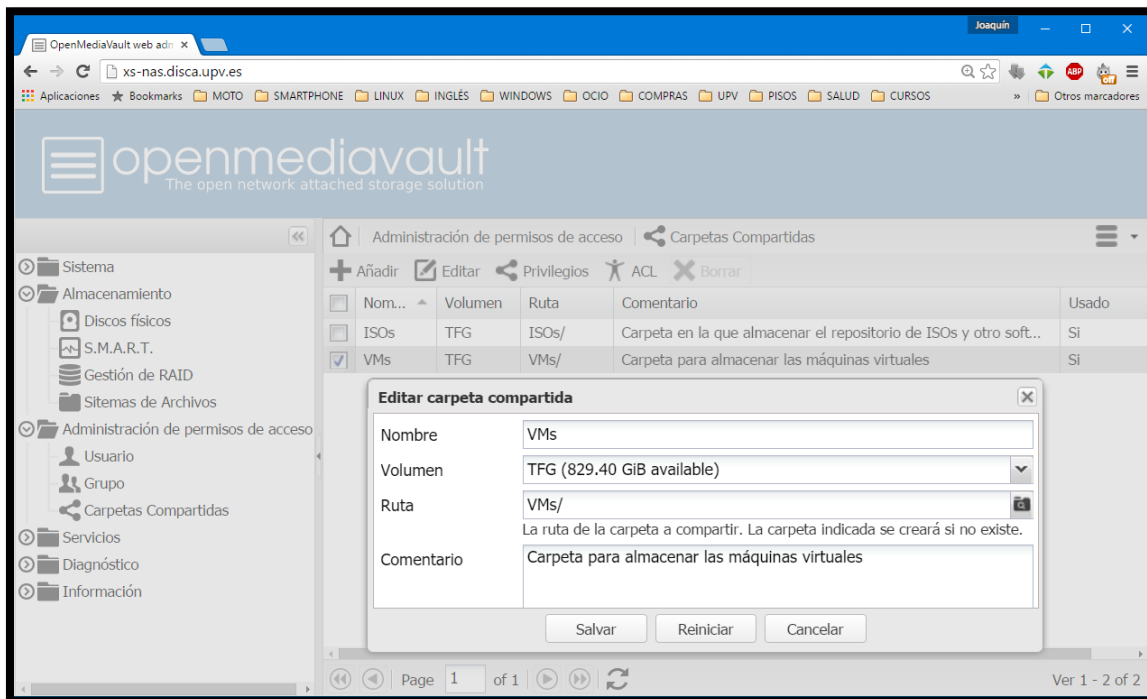


ILUSTRACIÓN 10 - RECURSOS EXPORTADOS EN /EXPORT/

```

      * MobaXterm 8.2 *
      (SSH client, X-server and networking tools)
  > SSH session to root@xs-nas.disca.upv.es
    * SSH compression : x (disabled or not supported by server)
    * SFTP Browser      : x
    * X11 forwarding    : x (disabled or not supported by server)
    * DISPLAY           : 192.168.1.104:0.0
  > For more info, ctrl+click on help or visit our website

Linux xs-nas 3.2.0-4-amd64 #1 SMP Debian 3.2.68-1+deb7u2 x86_64

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

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Nov 3 17:07:42 2015 from astel2.disca.upv.es
root@xs-nas:~# ls
root@xs-nas:~# cat /etc/exports
# /etc/exports: the access control list for filesystems which may be exported
# to NFS clients.  See exports(5).
/export/VMs (rw,subtree_check,secure)
/export/ISOs (rw,subtree_check,secure)

# NFSv4 - pseudo filesystem root
/export (ro,fsid=0,root_squash,no_subtree_check,hide)
root@xs-nas:~# lsblk
NAME        MAJ:MIN RM  SIZE RO TYPE MOUNTPOINT
├─sda       8:0    0 298.1G  0 disk
├─┌sda1     8:1    0  286G  0 part /
├─└sda2     8:2    0    1K  0 part
├─sda5     8:5    0  12.1G  0 part [SWAP]
├─sdb       8:16   0  465.8G  0 disk
├─┌md127    9:127  0  931.5G  0 raid0 /media/653ca60e-20c1-44a2-9cbf-6d705510a0a6
├─└sdc      8:32   0  465.8G  0 disk
├─┌md127    9:127  0  931.5G  0 raid0 /media/653ca60e-20c1-44a2-9cbf-6d705510a0a6
└─sr0      11:0    1  1024M  0 rom
root@xs-nas:~# mount
sysfs on /sys type sysfs (rw,nosuid,nodev,noexec,relatime)
proc on /proc type proc (rw,nosuid,nodev,noexec,relatime)
udev on /dev type devtmpfs (rw,relatime,size=10240k,nr_inodes=982139,mode=755)
devpts on /dev/pts type devpts (rw,nosuid,noexec,relatime,gid=5,mode=620,ptmxmode=000)
tmpfs on /run type tmpfs (rw,nosuid,noexec,relatime,size=794024k,mode=755)
/dev/disk/by-uuid/05b0b51-92b2-40ac-ae57-985daebc1b92 on / type ext4 (rw,relatime,errors=remount-ro,user_xattr,barrier=1,data=ordered)
tmpfs on /run/lock type tmpfs (rw,nosuid,nodev,noexec,relatime,size=5120k)
tmpfs on /run/shm type tmpfs (rw,nosuid,nodev,noexec,relatime,size=4124680k)
/dev/md127 on /media/653ca60e-20c1-44a2-9cbf-6d705510a0a6 type ext4 (rw,noexec,relatime,user_xattr,barrier=1,stripe=256,data=ordered,jqfmt=vfsv0,usrjquota=aquo
/dev/md127 on /export/VMs type ext4 (rw,noexec,relatime,user_xattr,barrier=1,stripe=256,data=ordered,jqfmt=vfsv0,usrjquota=aquota.user,grpquota=aquota.group)
rpc_pipefs on /var/lib/nfs/rpc_pipefs type rpc_pipefs (rw,relatime)
nfsd on /proc/fs/nfsd type nfsd (rw,relatime)
root@xs-nas:~#

```

ILUSTRACIÓN 11 - CAPTURA DE CONSOLA SSH CON VARIOS COMANDOS EJECUTADOS

En la ilustración 11 se muestra la consola de una conexión *ssh* a la máquina *xs-nas*. En la imagen se puede observar el resultado de varios comandos ejecutados en los que se muestra información relativa a los dispositivos físicos de almacenamiento (*/dev/sdb* y */dev/sdc*, son los discos usados en */dev/md127*, el *RAID 0*), su punto de montaje (*/export/VMs* y */export/ISOs*) y la configuración de los recursos exportados mediante *NFS* (*rw, subtree\_check, secure*).

El Centro de Almacenamiento (*xs-nas.disca.upv.es*) implementando un servidor *NAS OMV* ofrece el espacio diferenciando dos puntos de conexión (*ISOs* y *MVs*) en ellos, el Centro de Proceso ubicará los *SR* correspondientes: uno para almacenar las imágenes de instalación de los sistemas operativos y otro para almacenar las máquinas virtuales. Nuestro servidor *OMV* ya está ofreciendo servicio y espera que el Centro de Proceso use su espacio.









## 5.1. Instalación XenServer

*XenServer* es Open Source, y desde su web en [10] está disponible todo el software de la plataforma, así como la documentación relacionada. Para la realización de este trabajo se ha seguido, fundamentalmente, el manual disponible allí [11].

Para instalarlo (en todos los servidores *xs-server*) deberemos descargarnos la imagen *ISO* disponible para después grabarla en un *CD*. Arrancando la máquina con el *CD* en su interior y con la configuración correcta del *BIOS* nos aparece la pantalla que muestra la ilustración 13.



ILUSTRACIÓN 13 - PANTALLA DEL ARRANQUE DE LA INSTALACIÓN DE XENSERVER

Para realizar la instalación seguiremos las instrucciones del asistente. Las siguientes tablas (8 y 9) contienen 14 viñetas en las que se observa el proceso de instalación, que comienza con elementos básicos como la selección del teclado, aceptación de la licencia, y con la opción de instalar o actualizar (por si se tuviera una versión anterior a la que se instala). Posteriormente se elige el disco de destino, indicando después el tipo de medio (local, *http* o *FTP*, o *NFS*), ya que al parecer no es consciente de que estamos instalando de un medio u otro.

|   |  |
|---|--|
| <p>Welcome to XenServer - Version 6.5.0 (#90233c)<br/>Copyright (c) 2014 Citrix Systems, Inc.</p> <p style="text-align: center;"><b>Select Keymap</b></p> <p>Please select the keymap you would like to use:</p> <pre>[qwert] es [qwert] es-cp850 [qwert] et [qwert] et-nodeadkeys [qwert] fi [qwert] fi-latin1 [qwert] fi-latin9 [qwert] fi-old</pre> <p style="text-align: center;"><b>Ok</b></p> <p>&lt;Tab&gt;&lt;Alt-Tab&gt; between elements     &lt;F1&gt; Help screen</p>   | <p>Welcome to XenServer - Version 6.5.0 (#90233c)<br/>Copyright (c) 2014 Citrix Systems, Inc.</p> <p style="text-align: center;"><b>Welcome to XenServer Setup</b></p> <p>This setup tool can be used to install or upgrade XenServer on your system or restore your server from backup. Installing XenServer will erase all data on the disks selected for use.</p> <p>Please make sure you have backed up any data you wish to preserve before proceeding.</p> <p>To load a device driver press &lt;F9&gt;.</p> <p style="text-align: center;"><b>Ok</b>                      <b>Reboot</b></p> <p>&lt;Tab&gt;&lt;Alt-Tab&gt; between elements   &lt;F9&gt; load driver   &lt;F1&gt; Help screen</p>   |
| <p>Welcome to XenServer - Version 6.5.0 (#90233c)<br/>Copyright (c) 2014 Citrix Systems, Inc.</p> <p style="text-align: center;"><b>End User License Agreement</b></p> <p>Please view the accompanying source code (including LICENSE, COPYING, and README files, if any) for the applicable license agreements. To the maximum extent permitted by applicable law, the accompanying software is provided "as is" WITHOUT WARRANTIES OF ANY KIND, including the implied warranties of merchantability, fitness for a particular purpose, and non-infringement.</p> <p style="text-align: center;"><b>Accept EULA</b>                      <b>Back</b></p> <p>&lt;Tab&gt;&lt;Alt-Tab&gt; between elements     &lt;F1&gt; Help screen</p> | <p>Welcome to XenServer - Version 6.5.0 (#90233c)<br/>Copyright (c) 2014 Citrix Systems, Inc.</p> <p style="text-align: center;"><b>Virtual Machine Storage</b></p> <p>Which disks would you like to use for Virtual Machine storage? One storage repository will be created that spans the selected disks. You can choose not to prepare any storage if you wish to create an advanced configuration after installation.</p> <p style="text-align: center;"><b>[*] sda - 19 GB [VBOX HARDDISK]</b></p> <p style="text-align: center;"><b>[*] Enable thin provisioning (Optimized storage for XenDesktop)</b></p> <p style="text-align: center;"><b>Ok</b>                      <b>Back</b></p> <p>&lt;Tab&gt;&lt;Alt-Tab&gt; between elements   &lt;F5&gt; more info   &lt;F1&gt; Help screen</p> |
| <p>Welcome to XenServer - Version 6.5.0 (#90233c)<br/>Copyright (c) 2014 Citrix Systems, Inc.</p> <p style="text-align: center;"><b>Select Installation Source</b></p> <p>Please select the type of source you would like to use for this installation</p> <pre>Local media HTTP or FTP NFS</pre> <p style="text-align: center;"><b>Ok</b>                      <b>Back</b></p> <p>&lt;Tab&gt;&lt;Alt-Tab&gt; between elements     &lt;F1&gt; Help screen</p>   | <p>Welcome to XenServer - Version 6.5.0 (#90233c)<br/>Copyright (c) 2014 Citrix Systems, Inc.</p> <p style="text-align: center;"><b>Supplemental Packs</b></p> <p>Would you like to install any Supplemental Packs?</p> <p style="text-align: center;"><b>Yes</b>                      <b>No</b>                      <b>Back</b></p> <p>&lt;Tab&gt;&lt;Alt-Tab&gt; between elements     &lt;F1&gt; Help screen</p>  |
| <p>Welcome to XenServer - Version 6.5.0 (#90233c)<br/>Copyright (c) 2014 Citrix Systems, Inc.</p> <p style="text-align: center;"><b>Verify Installation Source</b></p> <p>Would you like to test your media?</p> <pre>Skip verification Verify installation source</pre> <p style="text-align: center;"><b>Ok</b>                      <b>Back</b></p> <p>&lt;Tab&gt;&lt;Alt-Tab&gt; between elements     &lt;F1&gt; Help screen</p>  | <p>Welcome to XenServer - Version 6.5.0 (#90233c)<br/>Copyright (c) 2014 Citrix Systems, Inc.</p> <p style="text-align: center;"><b>Networking</b></p> <p>Which network interface would you like to use for connecting to the management server on your host?</p> <pre>eth0 (08:00:27:62:61:17) eth1 (08:00:27:47:de:75)</pre> <p style="text-align: center;"><b>Ok</b>                      <b>Back</b></p> <p>&lt;Tab&gt;&lt;Alt-Tab&gt; between elements   &lt;F5&gt; more info   &lt;F1&gt; Help screen</p>  |

TABLA 7 - INSTALACIÓN XENSERVER (VIÑETAS 1-8)<sup>13</sup>

<sup>13</sup> Las viñetas se encuentran ordenadas de arriba a abajo y de izquierda a derecha

Posteriormente, el instalador pregunta por *Supplemental Packs*<sup>14</sup> y por la verificación del medio.

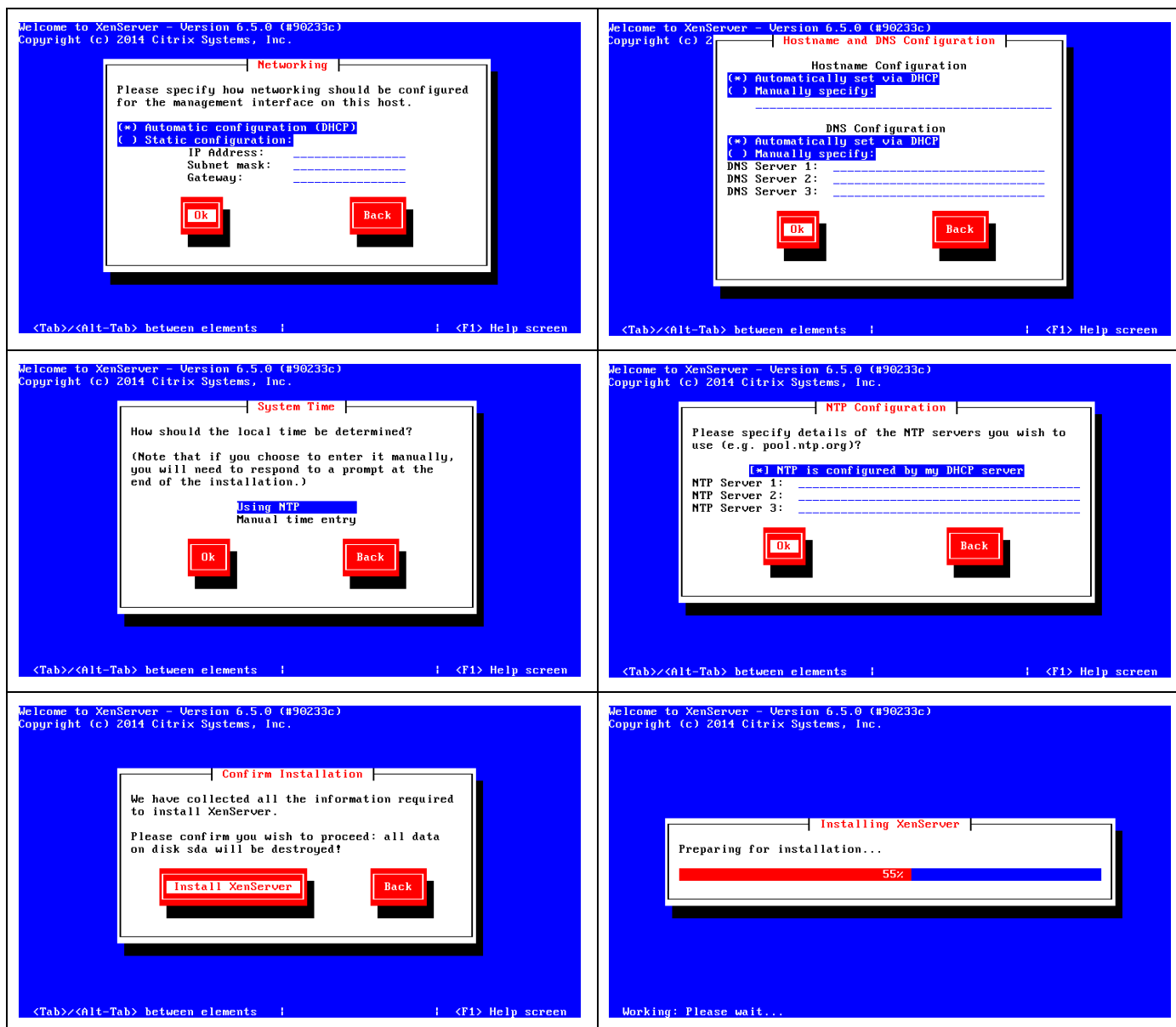


Tabla 8 - INSTALACIÓN XENSERVER (VIÑETAS 9-14)<sup>15</sup>

Si como en nuestro caso, se cuenta con más de una tarjeta de red, habrá que seleccionar cual se usa para la gestión y configurarla, bien de forma automática por *DHCP* o bien definiendo de manera estática sus parámetros. Uno de los primeros pasos que realizaremos en la sección 5.2.2 cuando acabe la instalación, es configurar la otra interfaz

<sup>14</sup> Software específico para *XenServer*.

<sup>15</sup> Las viñetas se encuentran ordenadas de arriba a abajo y de izquierda a derecha

de red, la que utilizaremos para la red de almacenamiento y que conecta en exclusiva al Centro de Proceso con el Centro de Almacenamiento.

*XenServer* necesita configurar su reloj y en el paso siguiente propone elegir entre la hora del sistema o la elección de un servidor *NTP (Net Time Protocol)*. Una vez seleccionado, se procede a la instalación del sistema, que ya no tendrá ninguna interrupción.

Una vez acaba de copiar y configurarse con las opciones que se han seleccionado se reiniciará y arrancará por primera vez (este proceso se repetirá para servidor todos los *xs-server*). La primera pantalla que nos muestra es la que se ve en la ilustración 14.



ILUSTRACIÓN 14 - ARRANQUE DE XENSERVER

Cuando termine el proceso de arranque nos mostrará, en el caso concreto de *XS-server-0*, la pantalla que aparece en la ilustración 15.

Desde la consola mostrada tendremos toda la funcionalidad de *XenServer*, pero la interacción no es tan agradable o intuitiva como en *XenCenter*, que veremos en la sección 5.2. En cualquier caso solo tendremos que recurrir a ella para realizar acciones muy concretas (básicamente de copia y restauración de algunos elementos).

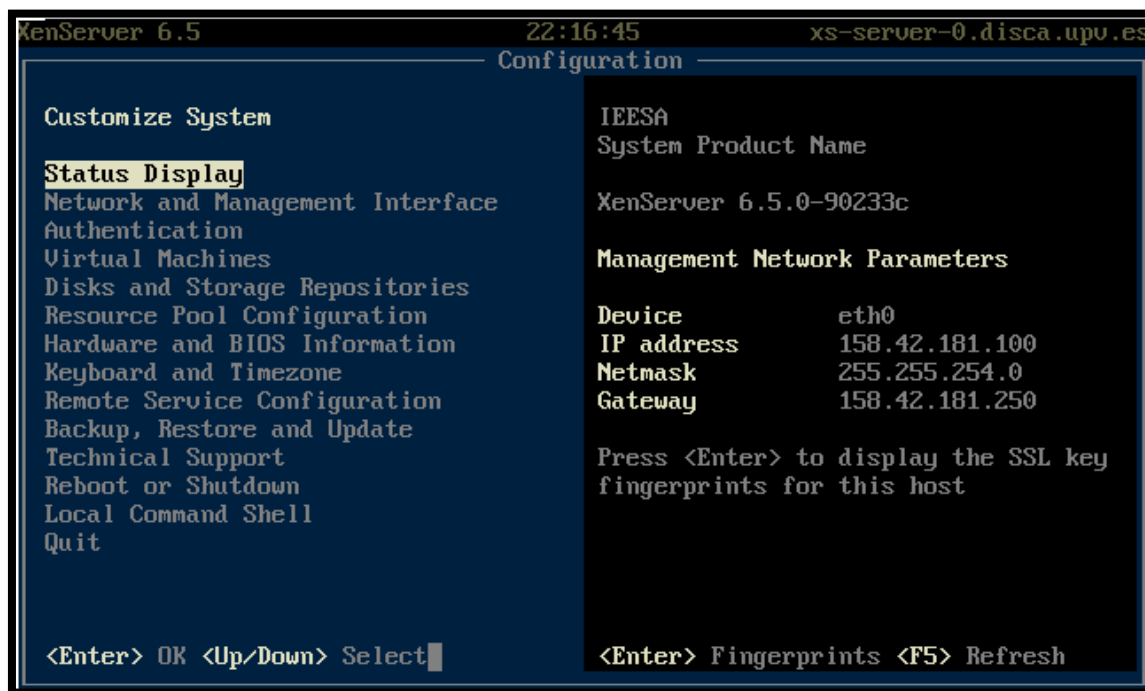


ILUSTRACIÓN 15 - PANTALLA INICIAL XS-SERVER-0

En la imagen se observará un menú en la parte izquierda, con el que podremos desplazarnos verticalmente y visualizar en la parte derecha la información referente a la opción actual. En el caso de la ilustración 15 se encuentra seleccionado el elemento *Status Display*, que muestra en el panel derecho información de la interfaz de la red administrativa.

## 5.2. Configuración sistema XenServer

Una vez instalados los servidores llega el momento de configurarlos. Para llevar a cabo esta labor se puede hacer uso de *XenCenter*, *CLI* local o por *SSH*, *addon* para *PowerShell* con las extensiones *XenServer*, *Xen Orchestra* u otras alternativas. En el desarrollo de este proyecto se ha usado de forma habitual la herramienta gráfica *XenCenter* y la *CLI*.

*XenCenter* está disponible en la misma página web que *XenServer*, pero si no lo hemos descargado, una forma de obtenerlo es conectándonos a cualquier servidor *XenServer* a través de un navegador. Como podemos observar en la ilustración 16, existe un servicio activo que nos muestra una sencilla página cuya funcionalidad es descargar *XenCenter*.

El proceso de instalación de *XenCenter* no tiene nada importante que reseñar, salvo que solo se encuentra disponible para plataformas *Windows*, y si no disponemos de un equipo en esa plataforma no lo podremos usar (aunque desde este momento contamos

con la posibilidad de crear una *VM Windows*, e instalando *XenCenter* administrarlo desde dentro).

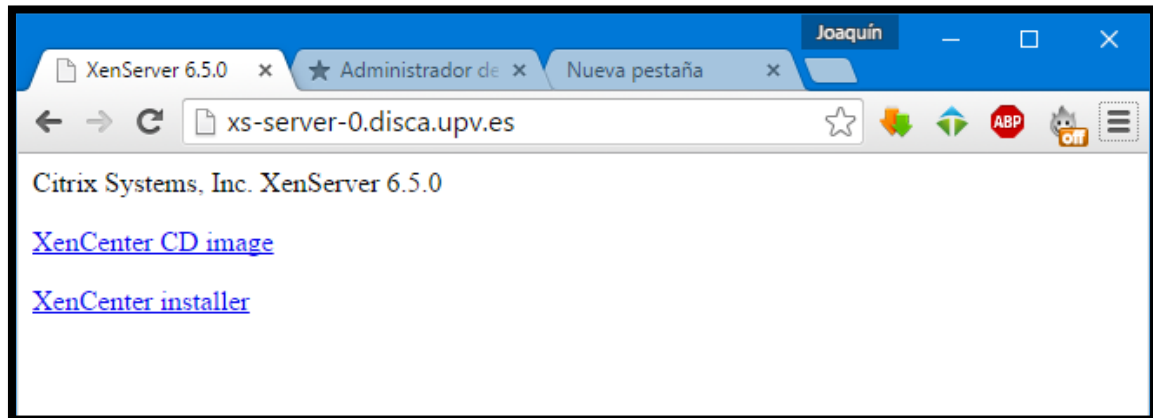


ILUSTRACIÓN 16 - DESCARGA DE XENCENTER DESDE NUESTRO SERVIDOR XENSERVER

En la ilustración 17 encontramos una captura de la aplicación. Este no es el aspecto que encontraremos al abrir la aplicación por primera vez. La consola que se muestra ya tiene objetos de varios servidores *XenServer* (la primera vez aparecería vacía), pero es más adecuado explicar de qué partes está compuesta y cuál es su funcionalidad si presenta contenido. A continuación, realizamos un recorrido visual de la aplicación *XenCenter*.

En la parte superior (1) se puede ver el menú de la aplicación, y justo debajo la barra de herramientas. El recuadro inferior izquierdo (2) muestra 5 ítems que cambian la representación del panel superior (3): los dos primeros permiten cambiar entre una representación jerárquica o agrupada por objetos de la misma clase; los dos siguientes muestran consultas y el último alertas, errores y actualizaciones. A su vez, la selección del panel izquierdo cambia la representación del derecho (4), que es el principal. En este último marco se detalla la información del objeto seleccionado, y si tiene más de una vista, en las etiquetas de la parte superior (5) se seleccionará el panel correspondiente.

En el ejemplo de la ilustración 17 podemos ver que está seleccionada la vista *Infraestructure* (2), que da la visión jerárquica del marco superior, y en este está seleccionado el objeto *XenCenter* (3), mostrándolo en el panel principal. El panel principal (4) muestra la información de la pestaña *Search* (5), que permite ver la información general de todos los *hosts* y *VM*.

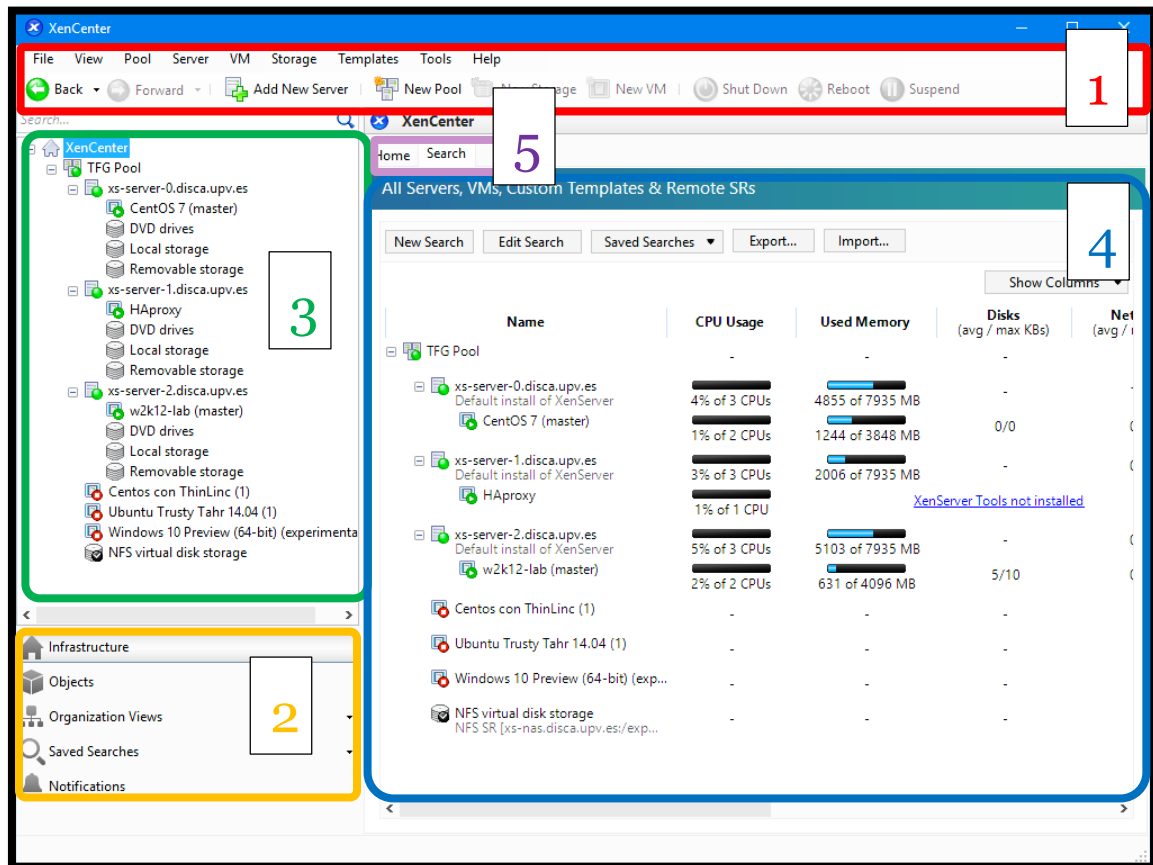


ILUSTRACIÓN 17 - APARIENCIA DE XENCENTER

La aplicación *XenCenter* admite la interacción habitual de las aplicaciones visuales y nos permite efectuar acciones muy rápidamente, con apenas un par de *clicks*, o incluso en algunas como, por ejemplo, migrar una máquina virtual de un *host* a otro, simplemente arrastrando el objeto que representa la *VM* y soltarlo encima del servidor de destino. De la misma forma, las vías para realizar una acción son amplias, pudiendo usar atajos de teclado, menú principal, botones de la barra de herramienta, menú contextual, o como hemos mencionado, movimientos de ratón o una combinación de varios de ellos. Dado que su interacción es tan intuitiva centraremos el resto de la sección en la *CLI*, por ser esta más potente y mucho más compleja.

La *CLI* basa su funcionamiento en el comando *xe*. Si instalamos *XenCenter* (estamos hablando de un equipo *Windows*) en la carpeta de instalación del programa veremos que tenemos disponibles el ejecutable *xe.exe*:

```
C:\Program Files (x86)\Citrix\XenCenter>xe
Error: No hostname was specified.
Usage:
  xe -s <server> -u <username> -pw <password> [-p <port>] <command>
  <arguments>
For help, use xe -s <server> -u <user> -pw <password> [-p <port>] help
C:\Program Files (x86)\Citrix\XenCenter>
```



La gran diferencia que existe entre hacer esto desde nuestro *Windows* o desde el servidor (bien en local o mediante una conexión *ssh*) es que deberemos introducir en cada instrucción el servidor, la cuenta de usuario y la contraseña para acceder, algo que puede resultar muy molesto. Todas las operaciones que se mostrarán a continuación se han hecho desde el servidor mediante una conexión *ssh*, por lo que no era necesario definir estos parámetros en cada llamada.

A continuación, mostraremos como usar *xe*

La CLI tiene la sintaxis básica del modo:

```
xe nombre-comando argumento1=valor1 argumento2=valor2...
```

Es decir, utilizaremos *xe* seguido de uno de los comandos, y después todos los pares *argumento=valor* que necesite la instrucción. En el caso de *Windows* debería ser

```
xe nombre-comando argumento1=valor1 argumento2=valor2...-s host -u usuario -pw contraseña
```

(admite las opciones tanto al principio como al final). Existe, además, un concepto de clase (no en todos los comandos). Esto significa que, como clase, todas aceptarán una serie de parámetros comunes que manipulan objetos como *hosts*, máquinas virtuales, plantillas, instantáneas, etc.

*<clase>-list*

*<clase>-param-get*

*<clase>-param-set*

*<clase>-param-list*

*<clase>-param-add*

*<clase>-param-remove*

*<clase>-param-clear*

...

Como se puede intuir, son operaciones genéricas y habituales que se invocan de la misma forma. Viendo un ejemplo queda más claro:

Veamos las clases máquinas virtuales (*vm*) y servidores *XenServer* (*host*).

Según lo explicado, existe un comando *vm-list* y *host-list*:

```
root@xs-server-0 ~]# xe vm-list
```



```

uuid ( RO)          : 7166c92a-1131-4eaa-b762-a921dcddb692
  name-label ( RW): Control domain on host: xs-server-2.disca.upv.es
  power-state ( RO): running

uuid ( RO)          : e55075f9-22a8-b7c8-ac3a-9c2eca623e7e
  name-label ( RW): Centos PM-KRB-NIS
  power-state ( RO): running

uuid ( RO)          : a0fb16b7-42f2-4810-be08-e3e0d9639499
  name-label ( RW): Control domain on host: xs-server-1.disca.upv.es
  power-state ( RO): running

uuid ( RO)          : 7adc44d3-df01-d308-d527-9e00ab413b17
  name-label ( RW): Ubuntu Trusty Tahr 14.04 (1)
  power-state ( RO): halted

uuid ( RO)          : fb90c579-e89b-1b83-a50d-211596dd7e24
  name-label ( RW): w2k12-lab (master)
  power-state ( RO): running

uuid ( RO)          : b31d4ccd-8fdd-74d3-249b-894c1bb61058
  name-label ( RW): HAProxy
  power-state ( RO): running

uuid ( RO)          : 606ae760-8168-1660-dc4b-8d487f833f1a
  name-label ( RW): CentOS 7 (master)
  power-state ( RO): running

uuid ( RO)          : 8cde6c93-247d-4107-96b0-0b9faa6d9371
  name-label ( RW): Control domain on host: xs-server-0.disca.upv.es
  power-state ( RO): running

```

Cada tripleta (*uuid*, *name-label*, *power-state*) nos muestra una máquina virtual (las que aparecen como *Control domain on host*: son los propios hipervisores). Nos lista todas las máquinas virtuales.

Si probamos con *host-list* obtenemos un resultado similar.

```

[root@xs-server-0 ~]# xe host-list
uuid ( RO)          : 9f772f02-c379-4631-a0e2-1dfa1617ab58
  name-label ( RW): xs-server-0.disca.upv.es
  name-description ( RW): Default install of XenServer

uuid ( RO)          : e2fb3289-dabb-4c2f-9df1-44de2173d404
  name-label ( RW): xs-server-2.disca.upv.es
  name-description ( RW): Default install of XenServer

uuid ( RO)          : 4bd9ee78-ed5f-4562-b73a-f9f3281ce7b9
  name-label ( RW): xs-server-1.disca.upv.es
  name-description ( RW): Default install of XenServer

```

No todos los comandos que se pueden utilizar en una clase admiten todas estas variaciones, pero son muchas las clases a las que se puede acceder de esta manera (*bond*, *console*, *host*, *host-crashdump*, *host-cpu*, *network*, *patch*, *pbd*, *pif*, *pool*, *sm*, *sr*, *task*, *template*, *vbd*, *vdi*, *vif*, *vlan* y *vm*).

Hay un concepto importante que comentar y que se puede observar en la respuesta de los comandos; el atributo *uuid*. El *uuid* (*universally unique identifier*)<sup>16</sup> identifica inequívocamente a cualquier elemento del sistema. En determinadas operaciones, como quitar un *host* de un *resource pool*, debe usarse este parámetro, no siendo aceptado otro identificador (como *name-label*). El *uuid* se puede usar siempre que se solicite un identificador de un objeto. Por ejemplo, otra operación común que podemos hacer con las máquinas virtuales es encenderlas (*xe vm-start*). Si escribimos el comando en consola y presionamos la tecla tabuladora<sup>17</sup>, en la línea inferior aparecerán impresas las opciones disponibles, no siendo ninguna de ellas el *uuid*:

```
[root@xs-server-0 ~]# xe vm-start
force=  on=  paused=  vm=
```

Si completamos el comando nuevamente con el parámetro *vm=* y presionamos nuevamente la tecla tabuladora nos aparece una lista de las máquinas por su atributo *name-label*:

```
[root@xs-server-0 ~]# xe vm-start vm=
CentOS\ 7\ \ (master\          Control\ domain\ on\
host:\ xs-server-2.disca.upv.es
Centos\ PM-KRB-NIS           HAProxy
Control\ domain\ on\ host:\ xs-server-0.disca.upv.es  Ubuntu\ Trusty\ Tahr\
14.04\ \ (1\
Control\ domain\ on\ host:\ xs-server-1.disca.upv.es  w2k12-lab\ \ (master\)
```

En la pantalla se imprimen (aunque no se ve claramente) el nombre de las máquinas.

Todo parece indicar que es ese el único parámetro válido, pero si queremos utilizar el *uuid* de la máquina virtual *Ubuntu Trusty Tahr 14.04 (1)*...

```
[root@xs-server-0 ~]# xe vm-start vm=7adc44d3-df01-d308-d527-9e00ab413b17
[root@xs-server-0 ~]#
```

Al cabo de unos segundos aparecerá de nuevo el *prompt* en la línea inferior y nuestra máquina estará encendida. Volveremos a apagarla con el comando siguiente:

```
[root@xs-server-0 ~]# xe vm-shutdown vm=Ubuntu\ Trusty\ Tahr\ 14.04\ \ (1\
[root@xs-server-0 ~]#
```

Por último, cabe citar que la *CLI* ofrece más posibilidades de configuración, ya que nos permite automatizar funciones por medio de scripts y en determinados casos, realizar acciones que no son posibles con *XenCenter* (como algunos procesos de recuperación, o

<sup>16</sup> Es un identificador de 128 bits, ampliamente usado, que normalmente se representa con 32 caracteres hexadecimales agrupados y separados por guiones en formato 8-4-4-4-12.

<sup>17</sup> En una consola Windows no se soporta la opción de sugerir o completar con el tabulador.



por ejemplo, dotar de más de 16 *vCPU* a las *VM Linux*) pero tiene por contrapartida un uso más complejo.

Resumiendo, en esta introducción debemos tener presente, primero, que la interfaz CLI está disponible para Windows y Linux, pero es más eficiente usarla desde Linux. También cabe resaltar que, para ejecutar alguna instrucción que tenga como objetivo a un objeto concreto, siempre podremos referenciarlo con su *uuid*, que es un identificador inequívoco.

En la tabla 9 se presentan contextos y algunos comandos de ejemplo de las clases implicadas.

| Área  | Ejemplo                         |
|---|---------------------------------|
| <b>Appliances de máquinas virtuales (vApps)</b>       | <i>xe appliance-create ...</i>  |
| <b>Audit de ficheros de acceso (RBAC) del pool</b>    | <i>xe audit-log-get ...</i>     |
| <b>Agrupamientos de interfaces (bonds)</b>            | <i>xe bond-create ...</i>       |
| <b>Unidades de CD/DVD de <i>hosts</i> físicos</b>     | <i>xe cd-list</i>               |
| <b>Consolas</b>                                       | <i>xe console-list</i>          |
| <b>Disaster Recovery</b>                              | <i>xe appliance-recover...</i>  |
| <b>Eventos</b>  | <i>xe event-wait...</i>         |
| <b>GPUs físicas, virtuales y agrupadas</b>            | <i>xe vgpu-create...</i>        |
| <b>Host físicos</b>                                   | <i>xe host-disable...</i>       |
| <b>Logs</b>   | <i>xe log-set-output...</i>     |
| <b>Mensajes: notificaciones alertas y eventos</b>     | <i>xe message-create...</i>     |
| <b>Network</b>  | <i>xe network-create...</i>     |
| <b>Patch: updates para los <i>hosts XenServer</i></b> | <i>xe patch-apply...</i>        |
| <b>PBD: Physical Block Devices</b>                    | <i>xe pbd-create...</i>         |
| <b>PIF: objetos de las interfaces de red</b>          | <i>xe pif-reconfigure-ip...</i> |
| <b>Resource pools</b>                                 | <i>xe pool-ha-enable...</i>     |
| <b>Storage Manager</b>                                | <i>xe sm-list</i>               |
| <b>Storage repositories</b>                           | <i>xe sr-list</i>               |
| <b>Task: tareas largas asíncronas</b>                 | <i>xe task-list</i>             |
| <b>Template</b>                                       | <i>xe template-list...</i>      |

|  |                                   |
|--|-----------------------------------|
| <b>Updates de XenServer</b>                  | <i>xe update-upload...</i>        |
| <b>User: cambio de contraseña de usuario</b> | <i>xe user-password-change...</i> |
| <b>VBD: Virtual Block Devices</b>            | <i>xe vbd-list</i>                |
| <b>VDI: Virtual Disk Images</b>              | <i>xe vdi-list</i>                |
| <b>VIF: Virtual Network Interfaces</b>       | <i>xe vif-list</i>                |
| <b>VLAN</b>                                  | <i>vlan-create...</i>             |
| <b>VM: máquinas virtuales</b>                | <i>xe vm-list...</i>              |

TABLA 9 - CLASES Y EJEMPLOS EN LA CLI

En las secciones 5.2.1 a 5.2.10 se explica cómo se realizan las acciones de configuración descritas en el índice, bien con *XenCenter* o bien con la *CLI*.

### 5.2.1. Añadir servidores a la consola

Nuestra consola *XenCenter* estará vacía la primera vez que la ejecutemos, y no podremos realizar ninguna acción hasta que añadamos algún servidor. Hay muchas formas de hacerlo (por ejemplo, desde la barra de herramientas, con el botón *Add New Server*), pero en cualquier caso llegaremos a una ventana de dialogo como la que muestra la ilustración 18.

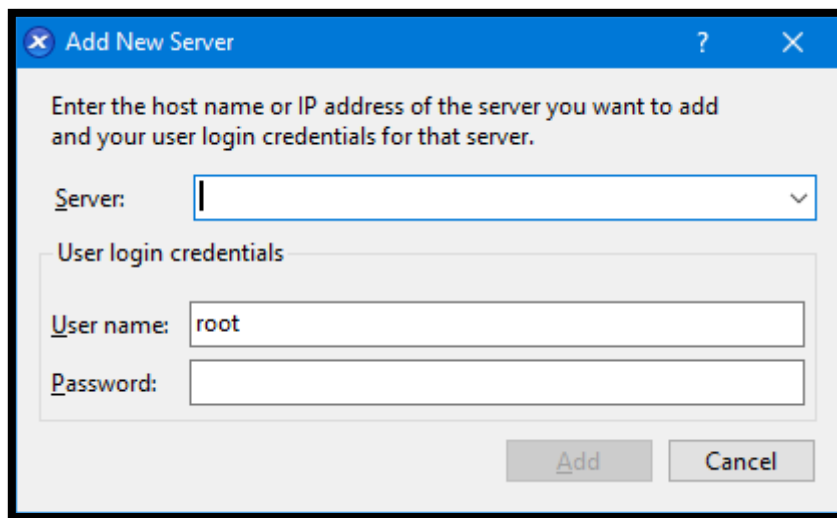


ILUSTRACIÓN 18 - AÑADIR SERVIDOR

En este dialogo introduciremos la dirección del servidor y la contraseña del usuario con permisos, que normalmente será root. Añadiremos todos los servidores con los que deseemos administrar repitiendo sucesivamente la misma operación. Cada vez que añadamos un servidor lo podremos ver en la vista de la izquierda.

### 5.2.2. Configurar la red

Durante la instalación de *XenServer* hemos definido la red administrativa (*management*) y quedará pendiente configurar la segunda interfaz, que usaremos para conectar al Centro de Almacenamiento. Para hacer esto podemos seleccionar el servidor en el marco de la derecha, seleccionar la pestaña *networking*, seleccionar la interfaz de red *Network 1* en la ventana principal (la interfaz de *management* es siempre la *Network 0*), seleccionar el botón que aparece en la parte inferior, etiquetado como *configure*, y finalmente seleccionando la segunda interfaz, que en nuestro caso ya la habíamos etiquetado como *storage* y confirmando los cambios que efectuemos presionando el botón *OK*. Todo esto es lo que muestra la ilustración 19.

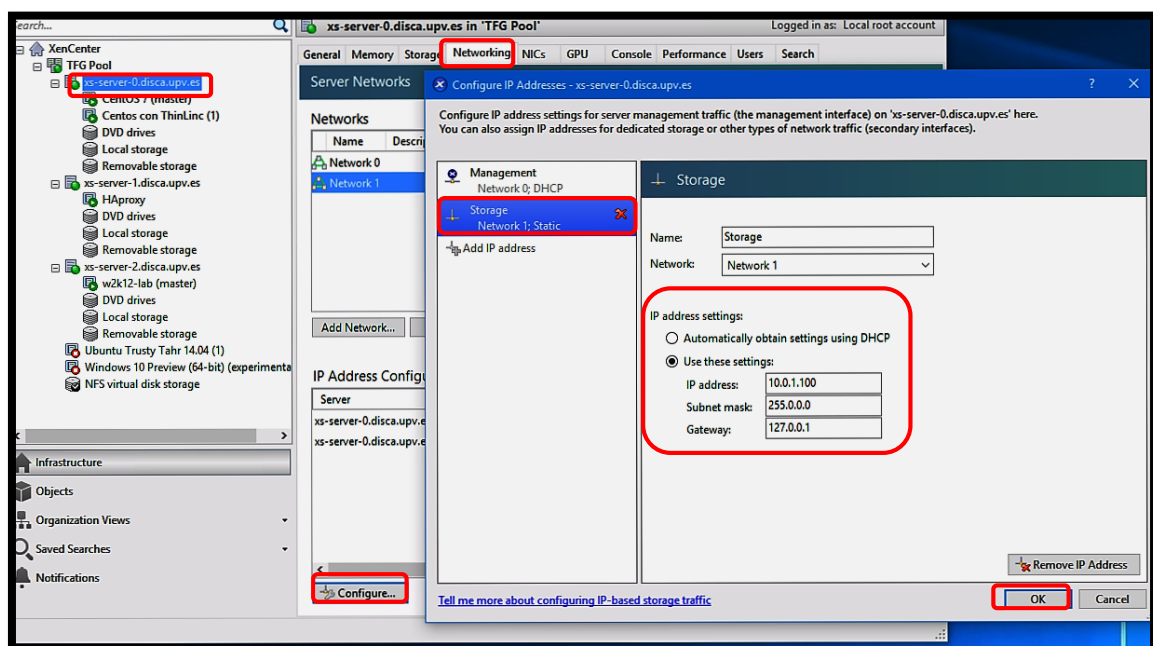


ILUSTRACIÓN 19 - CONFIGURACIÓN DE RED CON XENCENTER

Como decimos, la red ya estaba configurada y se había etiquetado la *Network 1* con el nombre *Storage*. En cualquier caso, tanto para crear o reconfigurar solo hay que completar los datos que aparecen reflejados en la ventana. Tal y como está configurado el sistema (con dos tarjetas de red por servidor) se ha destinado una para administración y servicio (conexión con la consola *XenCenter* y comunicación entre servidores para acciones coordinadas del *resource pool*) y otra para la red de almacenamiento. En el caso de disponer de una tercera interfaz de red se habrían separado las redes de administración y servicio.

### 5.2.3. Crear el Resource Pool

El *resource pool* es la entidad que distribuye los recursos de nuestro Centro de Proceso. Es vital que los servidores trabajen juntos de forma coordinada para lograr la alta disponibilidad y un balanceo de carga transparente. Para crear un *RP* desde la línea de comando, después de haberlos introducido en el punto 5.2.3, deberemos iniciar sesión en la consola de cada *host* que deseemos integrar (excepto desde el que vayamos a designar como *master*). En el caso de que nuestros servidores *xs-server* no estuvieran configurados en un *RP*, podríamos añadirlos haciendo uso de las siguientes instrucciones:

```
[root@xs-server-1 ~]# xe pool-join master-address=xs-server-0.disca.upv.es
master-username=root master-password=tfq2015
[root@xs-server-2 ~]# xe pool-join master-address=xs-server-0.disca.upv.es
master-username=root master-password=tfq2015
```

Como vemos, los datos solicitados corresponden con los mismos para añadir un servidor, solo que la acción se realiza desde el servidor e introduciendo la dirección, el usuario y la contraseña del servidor maestro del *Pool*.

Si lo hacemos nos dará el siguiente error:

```
You attempted an operation that was not allowed.
reason: Host is already part of a pool
```

Como vemos en el ejemplo, un servidor solo puede pertenecer a un *RP*.

Lo que sí podemos hacer es cambiarle o darle un nombre. En el siguiente ejemplo se le cambia el nombre a “TFG Pool actualizado”. Necesitaremos el *uuid* del *RP*, que podremos obtener con *pool-list*

```
[root@xs-server-0 ~]# xe pool-list
uuid ( RO) : 900825ed-9b51-59cf-7e93-fef7fd0913c2
  name-label ( RW): TFG Pool
  name-description ( RW):
    master ( RO): 9f772f02-c379-4631-a0e2-1dfa1617ab58
    default-SR ( RW): a4533971-dfec-b80c-1fbc-2b6f9564bfff
[root@xs-server-1 ~]# xe pool-param-set name-label="TFG Pool Actualizado"
uuid=900825ed-9b51-59cf-7e93-fef7fd0913c2
```

### 5.2.4. Configurar los repositorios

Tanto las *VM* como el software para instalar sistemas operativos están disponibles a través de un recurso de red, que en *XenServer* se denominan *Software Repository*. Nos uniremos al Centro de Almacenamiento configurando los *SR*, que añadiremos de forma muy similar (con el mismo asistente). A continuación, se muestran los pasos para añadir un *SR* que contenga las imágenes de los sistemas a instalar.



Desde la consola *XenCenter*, en la barra de herramientas seleccionamos *New Storage* y elegimos las opciones adecuadas (en este caso *NFS ISO* y el servidor *10.0.1.1:/export/ISOs*). Lo podemos ver en las ilustraciones 20-22.

La diferencia para añadir un recurso para el uso de *VM* o como repositorio de software para instalar sistemas radica únicamente en la selección del tipo, que en el primer caso corresponde *NFS VHD*<sup>18</sup> y el segundo a *NFS ISO*.

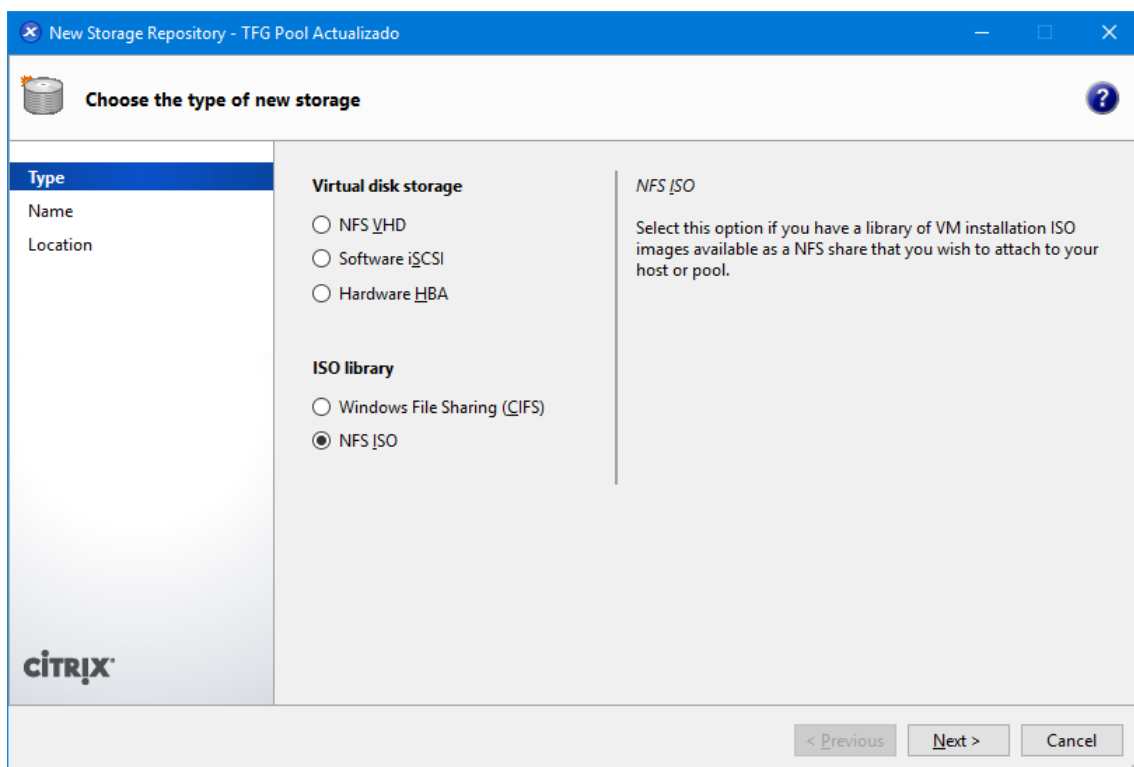


ILUSTRACIÓN 20 - AÑADIR SR (1/3)

---

<sup>18</sup> *Virtual Hard Drive*, en referencia a los discos que utilizarán las máquinas virtuales.



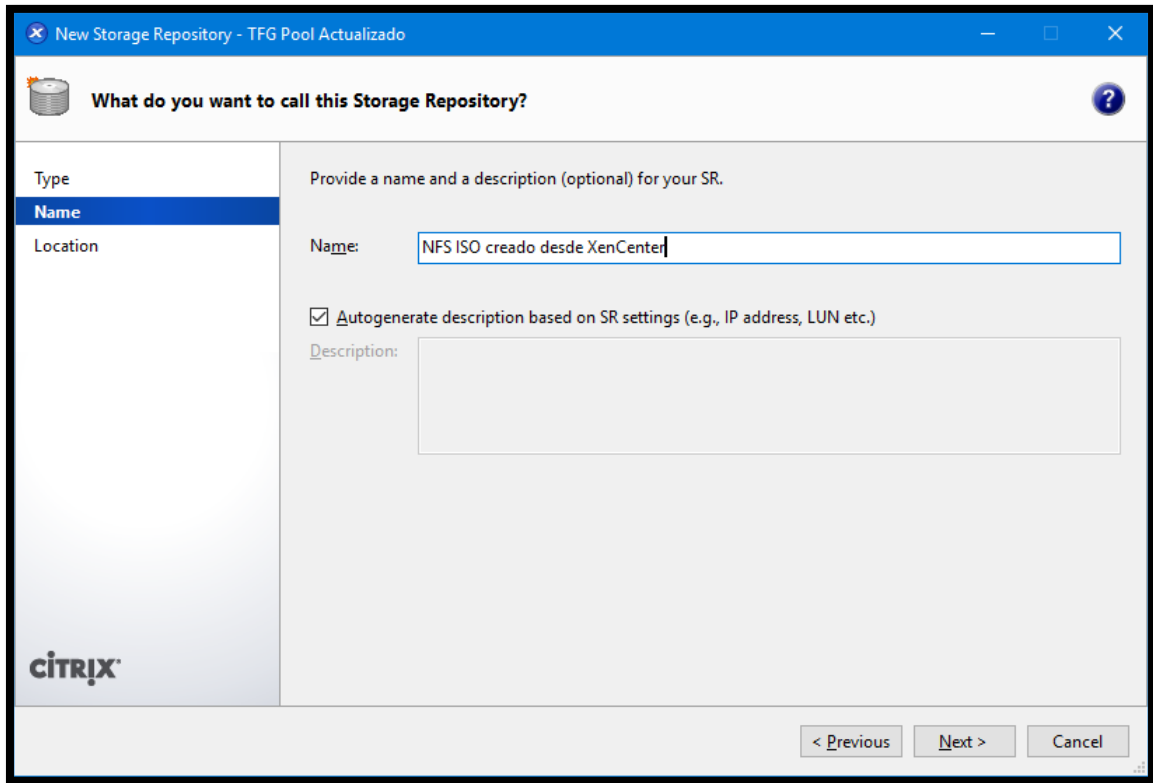


ILUSTRACIÓN 21 - AÑADIR SR (2/3)

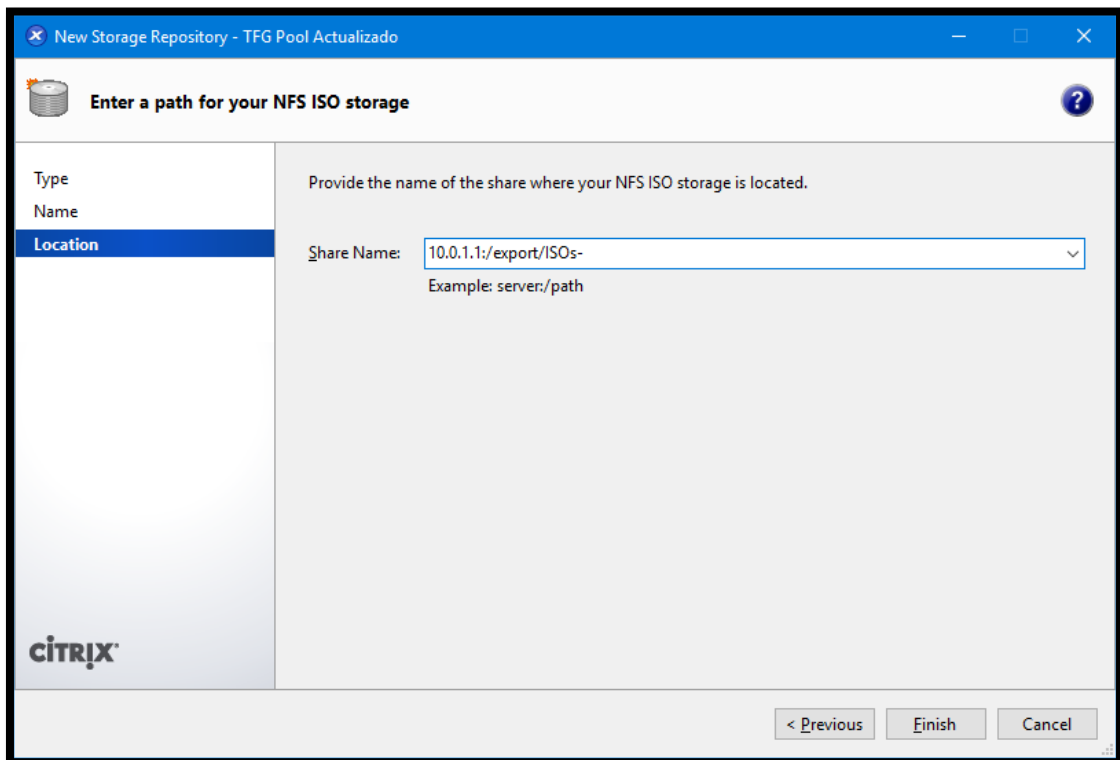


ILUSTRACIÓN 22 -AÑADIR SR (3/3)



Otra opción es crearlo desde la CLI:

```
[root@xs-server-0 ~]# xe sr-create content-type=iso type=nfs shared=true name-label="Creado desde CLI" device-config:server=10.0.1.1 device-config:serverpath=export/ISOs
```

Como se observa, los parámetros toman los mismos valores que hemos introducido en el asistente.

### 5.2.5. Encendido remoto

Para configurar el encendido remoto son necesarias dos acciones: activar desde el BIOS la característica *Wake-on-Lan* y configurar en la pantalla de propiedades del servidor (Menú *Server* -> *Properties* -> *Power on* -> opción *Wake-on-Lan*), como muestra la ilustración 23.

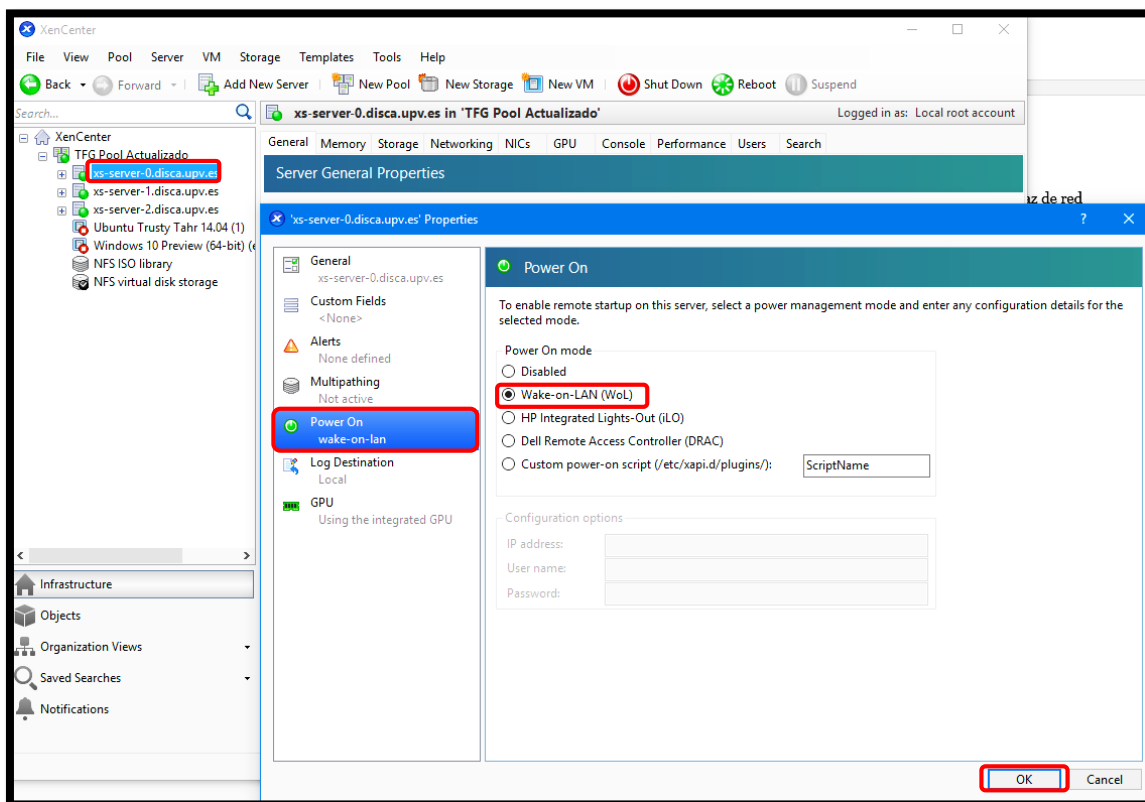


ILUSTRACIÓN 23 - ACTIVACIÓN DE WAKE-ON-LAN

Para encender remotamente un equipo usando la consola *XenCenter* podremos hacer uso del menú *server*, barra de herramientas, etc. Desde *CLI* podremos hacer uso de la siguiente instrucción para encender *xs-server-1* y *xs-server-2*:

```
[root@xs-server-0 ~]# xe host-power-on host=xs-server-1 host=xs-server-2
```

### 5.2.6. Máquinas Virtuales

Las máquinas virtuales son las Instancias de servicio de SERP (la parte activa de la Bolsa de Servicios). Desde la interfaz de *XenCenter* se accede al asistente para crear máquinas virtuales, bien a través del menú, de la barra de herramientas o bien del menú contextual de cualquier servidor del *Resource Pool*, basta con seleccionar *New VM*. En primer término, el asistente ofrece a elegir entre una serie de plantillas de distintos sistemas operativos soportados por *XenServer*, y una vez seleccionado, nos solicitará introducir un nombre la máquina. En las siguientes pantallas elegiremos el soporte de instalación y el *host* anfitrión. Como instalamos desde un *NFS ISO* nos ofrece dejar en sus manos la elección del servidor (al tener almacenamiento compartido y accesible desde cualquier *host* aparece esta opción), encargándose el *Pool* de que el seleccionado cuente con suficientes recursos. En la parte final del asistente se escogerán los detalles referentes a la *CPU*, *RAM*, disco y red. El asistente desde *XenCenter* simplifica mucho esta tarea, ya que desde la *CLI* es necesario introducir un comando distinto para crear, modificar y configurar cada componente.

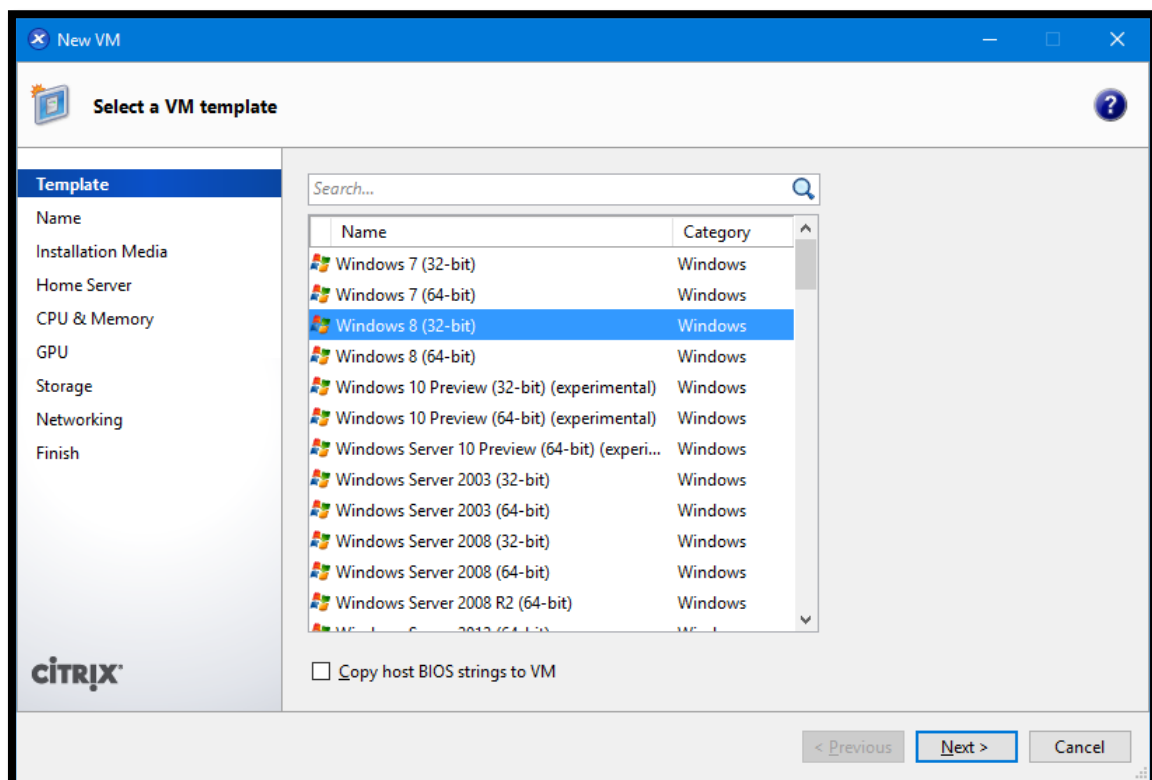


ILUSTRACIÓN 24 - ASISTENTE CREACIÓN MV

Para operar con la *VM* para apagar, reiniciar, suspender, mover, etc. usando la consola *XenCenter* haremos uso del menú *server*, barra de herramientas, etc.

Como se vió en el punto 5.1, desde la *CLI* se pueden hacer las mismas operaciones usando la clase *vm*, y si después de escribir la clase presionamos la tecla tabuladora veremos la lista de comandos de su clase:

```
[root@xs-server-0 ~]# xe vm-
vm-assert-can-be-recovered      vm-destroy                      vm-param-list
vm-call-plugin                  vm-disk-add                     vm-param-remove
vm-cd-add                       vm-disk-list                   vm-param-set
vm-cd-eject                    vm-disk-remove                 vm-pause
vm-cd-insert                    vm-export                      vm-reboot
vm-cd-list                     vm-import                      vm-recover
vm-cd-remove                   vm-install                    vm-reset-powerstate
vm-checkpoint                  vm-is-bios-customized         vm-resume
vm-clone                       vm-list                        vm-retrieve-wlb-
recommendations
vm-compute-maximum-memory      vm-memory-dynamic-range-set   vm-shutdown
vm-compute-memory-overhead     vm-memory-limits-set         vm-snapshot
vm-copy-bios-strings           vm-memory-static-range-set   vm-start
vm-crashdump-list             vm-memory-target-set         vm-suspend
vm-data-source-forget         vm-migrate                    vm-uninstall
vm-data-source-list           vm-param-add                  vm-unpause
vm-data-source-query          vm-param-clear                vm-vcpu-hotplug
vm-data-source-record         vm-param-get                  vm-vif-list
```

Por ejemplo, para migrar la *VM HAProxy* que se está ejecutando sobre *xs-server-1* y enviarla a *xs-server-2* haríamos uso de la siguiente instrucción:

```
[root@xs-server-0 ~]# xe vm-migrate vm=HAProxy host=xs-server-2.disca.upv.es
```

### 5.2.7.Xs-tools

*Xs-tools* es una aplicación que ha de instalarse en las *VM* para obtener un soporte completo. La instalación de la herramienta proporciona una mejora del rendimiento, activa algunas acciones que no están disponibles<sup>19</sup> y mejora la comunicación con *XenCenter* y con la *CLI*. El proceso de instalación consistirá en montar la imagen *ISO xs-tools* en la *VM* e invocar el programa instalador para la plataforma. Desde *CLI*, por ejemplo:

```
[root@xs-server-0 ~]# xe vm-cd-insert cd-name=xs-tools.iso vm=HAProxy
```

Ahora resta ejecutar el proceso de instalación. En el caso que nos ocupa, al ser un *Linux* que solo cuenta con línea de comandos procederemos de la siguiente manera:

```
root@xs-HAProxy:~# mount /dev/sr0 /media/cdrom/
root@xs-HAProxy:~# /media/cdrom/Linux/install.sh
Detected `Ubuntu 14.04.3 LTS' (ubuntu version 14).
The following changes will be made to this Virtual Machine:
* enable arp_notify in sysctl.
* packages to be installed/upgraded:
  - xe-guest-utilities_6.5.0-1420_amd64.deb
Continue? [y/n] y
Seleccionando el paquete xe-guest-utilities previamente no seleccionado.
```

<sup>19</sup> migración de máquinas en determinadas condiciones, reinicios y apagados limpios, aplicación y cambio instantáneo de algunos parámetros sin necesidad de reiniciar, etc.

```
(Leyendo la base de datos ... 118347 ficheros o directorios instalados
actualmente.)
Preparing to unpack .../xe-guest-utilities_6.5.0-1420_amd64.deb ...
Unpacking xe-guest-utilities (6.5.0-1420) ...
Configurando xe-guest-utilities (6.5.0-1420) ...
Mounting xenfs on /proc/xen: OK
Detecting Linux distribution version: OK
Starting xe daemon: OK
Processing triggers for ureadahead (0.100.0-16) ...
You should now reboot this Virtual Machine.
```

Al reiniciar la *VM* ya estará actualizada.

### 5.2.8. Instantáneas y Plantillas

En el capítulo 6 se verá con detalle la creación y configuración de los Servicios y una vez configurados se realizará una instantánea de ellos para posteriormente convertirlos en una plantilla. La plantilla será el Servicio, y una máquina generada a partir de ella será nuestra Instancia de servicio. Centrándonos de nuevo en el Centro de Proceso, para realizar una instantánea de alguna de nuestras máquinas virtuales desde la *CLI*, utilizaremos el comando correspondiente de la clase *vm*, en el ejemplo siguiente usamos *vm-snapshot*. Como siempre utilizaremos los pares argumento=valor que necesite la instrucción, y los podremos consultar de la misma forma que en el caso anterior.

```
[root@xs-server-0 ~]# xe vm-snapshot vm=HAProxy new-name-label="HAProxy con
xstools" new-name-description="Instantánea de la VM HAProxy después de haber
instalado las xstools"
36e8a846-8c4e-212e-91ca-dd83117287b5
[root@xs-server-0 ~]#
```

Si la máquina virtual de la que queremos sacar la instantánea está en ejecución y se desea guardar el estado se puede utilizar el comando *vm-snapshot-with-quiesce*. En cualquier caso, la ejecución nos devolverá el *uuid* de la instantánea creada (lo que es de utilidad para encadenar acciones al realizar un *script*).

Desde la interfaz gráfica lo podemos hacer rápidamente con un *click* derecho (en la *VM*) y eligiendo en el menú desplegable la opción *Take a snapshot...* y seleccionando después el nombre, descripción e indicando si se desea guardar el estado.

En *XenCenter* puede ser muy útil la herramienta gráfica a tal efecto (seleccionando la *VM* y marcando la pestaña *snapshots*). Como se puede ver en la ilustración 25, en esta vista se muestran las instantáneas que se han hecho, y se puede revertir la *VM* a cualquiera de los estados anteriores.

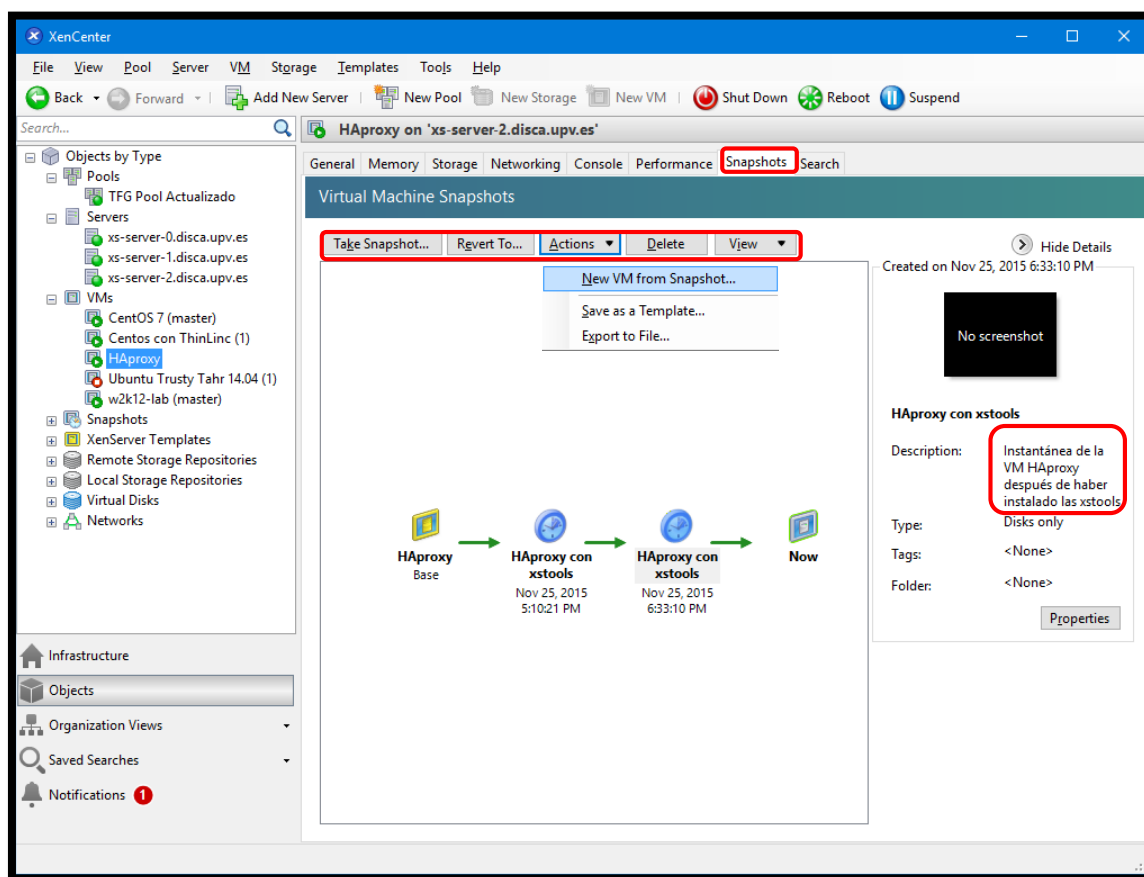


ILUSTRACIÓN 25 - GESTIÓN DE INSTANTÁNEAS

Precisamente esta vista es también la más adecuada para generar una plantilla, puesto que podremos seleccionar una instantánea y convertirla con *Actions* -> *Save as template* o usarla instantáneamente con *New VM from snapshot*, o también exportarla.

Desde la *CLI* el proceso de convertir *snapshot* en *template* es el siguiente:

- 1) Averiguar el *uuid* (esta es una de las operaciones que lo necesita).
- 2) Ejecutar el comando *xe* pertinente con el *uuid* conseguido.

Esto se puede realizar con estas dos instrucciones...

```
[root@xs-server-0 ~]# xe snapshot-list | grep -C1 HAProxy20
uuid ( RO) : 36e8a846-8c4e-212e-91ca-dd83117287b5
  name-label ( RW): HAProxy con xstools
  name-description ( RW): Instantánea de la VM HAProxy después de haber
instalado las xstools
```

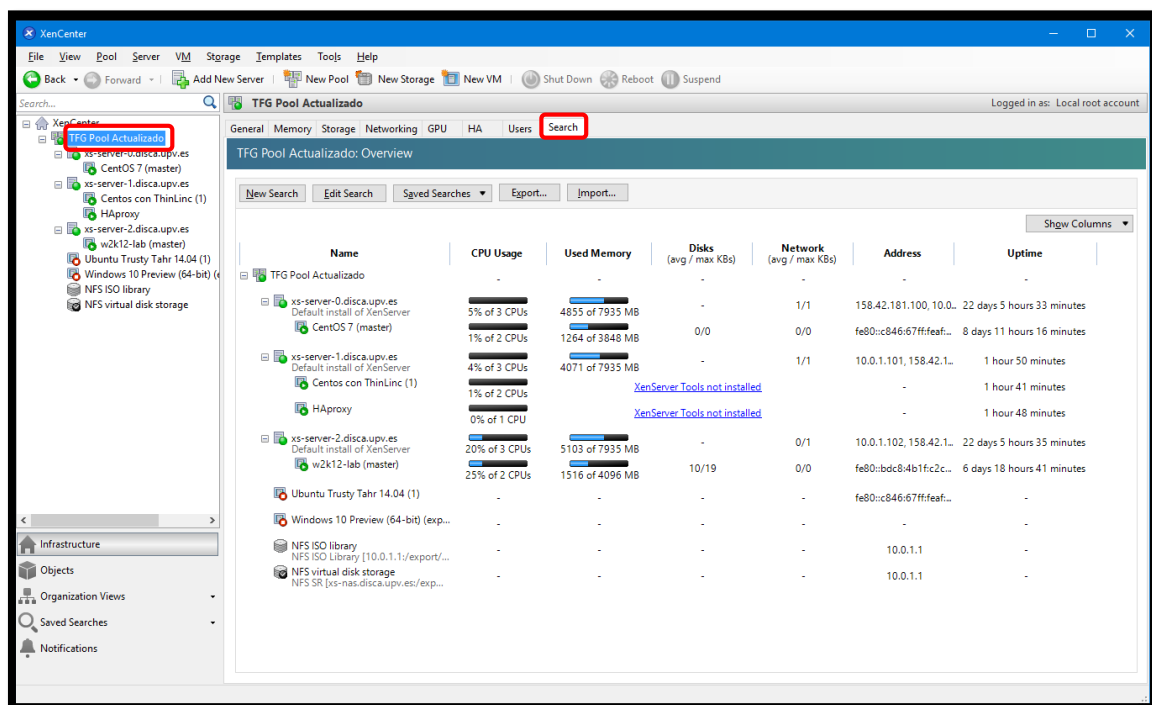
<sup>20</sup> La tubería *grep HAProxy* hace un filtrado de la información que recibe del comando *xe snapshot-list*, mostrando solo las líneas que contienen la cadena *HAProxy*. Con el modificador - *C1* hacemos que nos muestre también las líneas inmediatamente anteriores y posteriores. De esta manera aparecen solo cuatro líneas (*uuid*, *name-label*, *name-description* y una línea en blanco).

```
[root@xs-server-0 ~]# xe snapshot-export-to-template snapshot-uuid=36e8a846-8c4e-212e-91ca-dd83117287b5 filename="ejemplo de plantilla de HAProxy-2"
```

### 5.2.9. Visualización del rendimiento

Con *XenCenter* es posible observar el rendimiento a tres niveles distintos: *Pool*, *Host* y *VM* (ilustraciones 26-28). Como hemos visto en 5.2.7, es necesario tener instaladas las *xs-tools* en las máquinas virtuales para obtener la máxima información.

Estas vistas son configurables y proporcionan información sobre la carga del sistema: *CPU*, *RAM*, discos e interfaces de red en múltiples parámetros. Serán importantes si queremos comprobar que SERP funciona adecuadamente, ya que todo nuestro sistema queda reflejado en esos datos.



The screenshot shows the XenCenter interface for a pool named "TFG Pool Actualizado". The "Search" tab is selected, displaying a table of VMs and their performance metrics. The table includes columns for Name, CPU Usage, Used Memory, Disks, Network, Address, and Uptime. The data is as follows:

| Name  | CPU Usage     | Used Memory     | Disks (avg / max KBs) | Network (avg / max KBs) | Address                 | Uptime                     |
|---|---------------|-----------------|-----------------------|-------------------------|-------------------------|----------------------------|
| TFG Pool Actualizado  | -             | -               | -                     | -                       | -                       | -                          |
| xs-server-0.disca.upv.es<br>Default install of XenServer<br>CentOS 7 (master)       | 5% of 3 CPUs  | 4855 of 7935 MB | -                     | 1/1                     | 158.42.181.100, 10.0... | 22 days 5 hours 33 minutes |
| xs-server-1.disca.upv.es<br>Centos con ThinLinc (1)                                 | 1% of 2 CPUs  | 1264 of 3848 MB | 0/0                   | 0/0                     | fe80::c846:67fffef...   | 8 days 11 hours 16 minutes |
| xs-server-1.disca.upv.es<br>Default install of XenServer<br>Centos con ThinLinc (1) | 4% of 3 CPUs  | 4071 of 7935 MB | -                     | 1/1                     | 10.0.1.101, 158.42.1... | 1 hour 50 minutes          |
| HAProxy   | 1% of 2 CPUs  | -               | -                     | -                       | -                       | 1 hour 41 minutes          |
| HAProxy   | 0% of 1 CPU   | -               | -                     | -                       | -                       | 1 hour 48 minutes          |
| xs-server-2.disca.upv.es<br>Default install of XenServer<br>w2k12-lab (master)      | 20% of 3 CPUs | 5103 of 7935 MB | -                     | 0/1                     | 10.0.1.102, 158.42.1... | 22 days 5 hours 35 minutes |
| w2k12-lab (master)  | 25% of 2 CPUs | 1516 of 4096 MB | 10/19                 | 0/0                     | fe80::bdc8:4b1fc2c...   | 6 days 18 hours 41 minutes |
| Ubuntu Trusty Tahr 14.04 (1)  | -             | -               | -                     | -                       | fe80::c846:67fffef...   | -                          |
| Windows 10 Preview (64-bit) (exp...)  | -             | -               | -                     | -                       | -                       | -                          |
| NFS ISO library   | -             | -               | -                     | -                       | 10.0.1.1                | -                          |
| NFS ISO Library [10.0.1.1/export/...]   | -             | -               | -                     | -                       | 10.0.1.1                | -                          |
| NFS virtual disk storage  | -             | -               | -                     | -                       | 10.0.1.1                | -                          |
| NFS SR [xs-nas.disca.upv.es/exp...]   | -             | -               | -                     | -                       | 10.0.1.1                | -                          |

ILUSTRACIÓN 26 - VISUALIZACIÓN DEL POOL

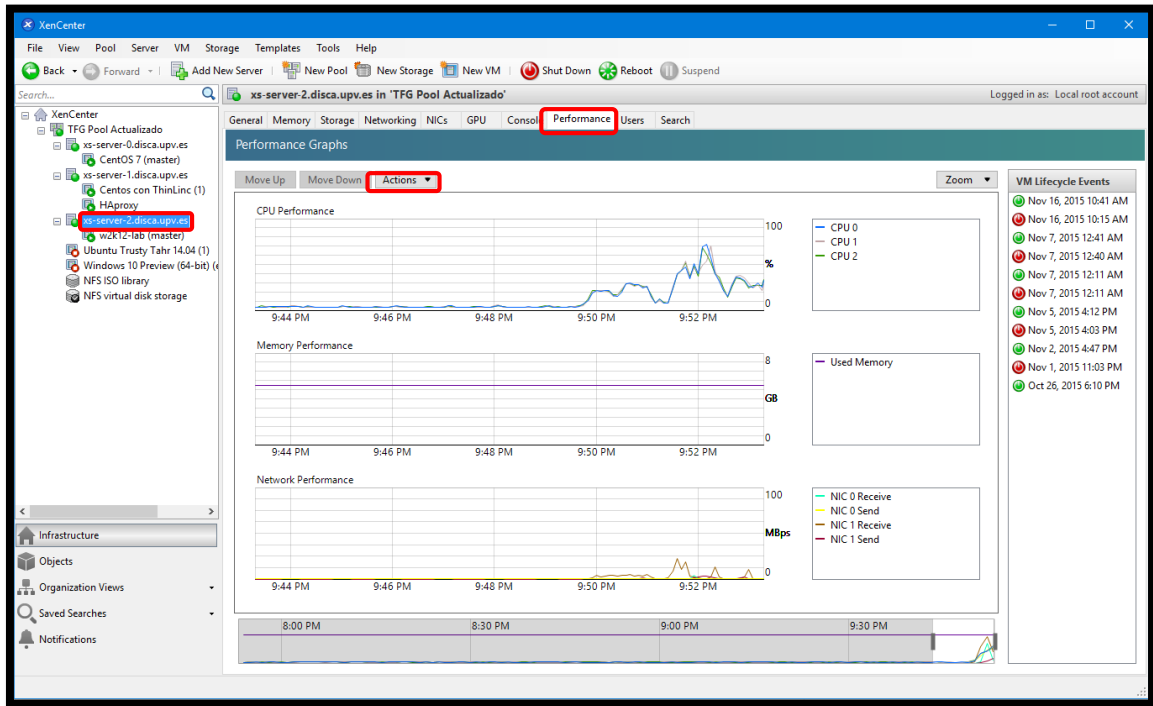


ILUSTRACIÓN 27 - VISUALIZACIÓN DEL HOST

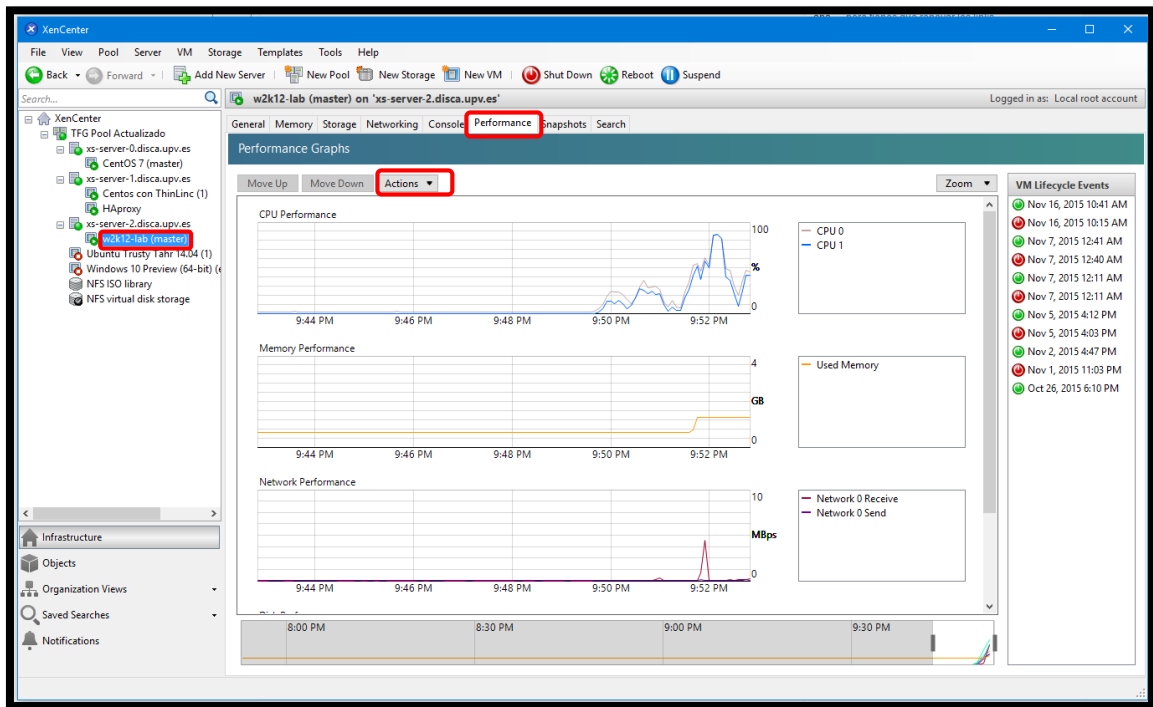


ILUSTRACIÓN 28 - VISUALIZACIÓN DE LA MÁQUINA W2K12-LAB



### 5.2.10. Alta disponibilidad

Uno de los objetivos principales para SERP es ofrecer alta disponibilidad. La utilidad *HA* de *XenServer* permite que, si un servidor está ejecutando instancias, y en un momento dado queda fuera de línea, el resto de servidores del *Pool* restablecerán el servicio de las máquinas virtuales afectadas y las devolverán a ejecución. [12]

Esta característica es posible si se cumplen dos condiciones: las máquinas virtuales son ágiles y existe un espacio libre de al menos 365 MBytes en algún *SR* del *Pool* (ilustración 29). Se considerará como máquina ágil aquella en la que sus discos virtuales se alojen en un *SR* de red (como es el caso de todas nuestras máquinas) y que no tengan ninguna conexión con los *DVD* locales de los servidores (condición que también cumplimos). Una vez configurado el sistema, los servidores comprueban constantemente que los demás se encuentran activos, así como también registran su actividad en el disco creado en el almacenamiento compartido. En la ilustración 30 seleccionamos nuestro *SR NFS VHD* como *heartbeat SR*<sup>21</sup>.

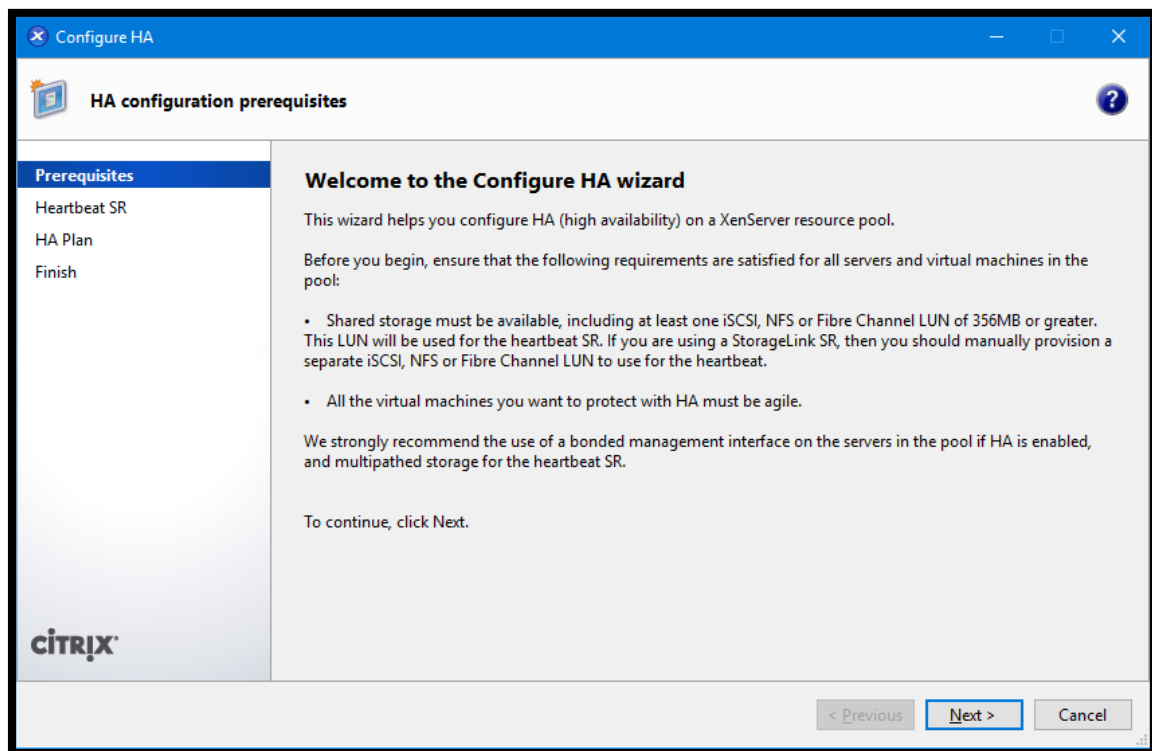


ILUSTRACIÓN 29 - ASISTENTE HA (1/4)

<sup>21</sup> Aclaración: El disco creado para ese fin (*VHD*) estará ubicado en un repositorio de software (*SR*) compartido en red por algún protocolo (*NFS*). El *SR* que se usa para ubicar dicho disco recibe el nombre de *heartbeat SR* (la nomenclatura no facilita en exceso su compresión).

Además de cumplir las condiciones anteriores, hay que tener en cuenta que solo se podrá emplear en condiciones de carga por debajo de un umbral, ya que si el sistema está funcionando al máximo de su capacidad no será posible que los demás servidores recuperen las instancias perdidas. Por este motivo será necesario configurar un mecanismo, para asegurar al menos, las que se consideren imprescindibles. *XenServer* utiliza para este cometido el *HA Plan*.

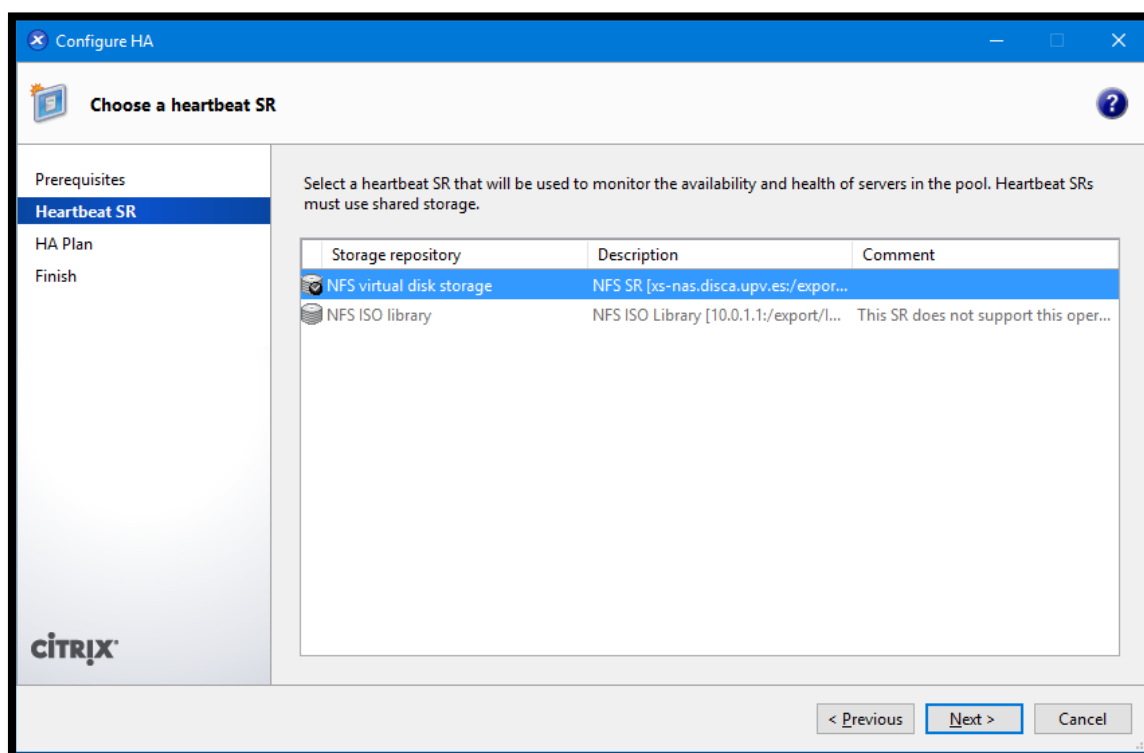


ILUSTRACIÓN 30 - ASISTENTE HA (2/4)

En la ilustración 31 se puede ver la configuración del *HA Plan* del *Pool TFG*. En plan, deberemos asignar prioridades a las máquinas: eligiendo si en caso de fallo deseamos que se reinicie siempre, si queremos que se reinicie si es posible o si por el contrario no queremos que se reinicie. Junto con estas condiciones se seleccionará en el orden para empezar a levantar las máquinas y tiempo que se esperará entre una y otra. Mientras se realiza la elección de los valores para estos parámetros el asistente calculará (teniendo en cuenta las máquinas virtuales que se encuentran actualmente en marcha y donde se encuentran ubicadas) el número máximo de servidores que pueden quedar fuera de línea manteniendo el plan. Para poder aprobar el plan el número que se indique deberá ser menor o igual que el máximo soportado. Una vez que se cumplan las condiciones se continuará el asistente mostrando el resumen de la configuración (ilustración 32) y al finalizarlo reservará el espacio para el *heartbeat SR*. A partir de este momento SERP cumplirá con las condiciones de alta disponibilidad que se hayan planificado.

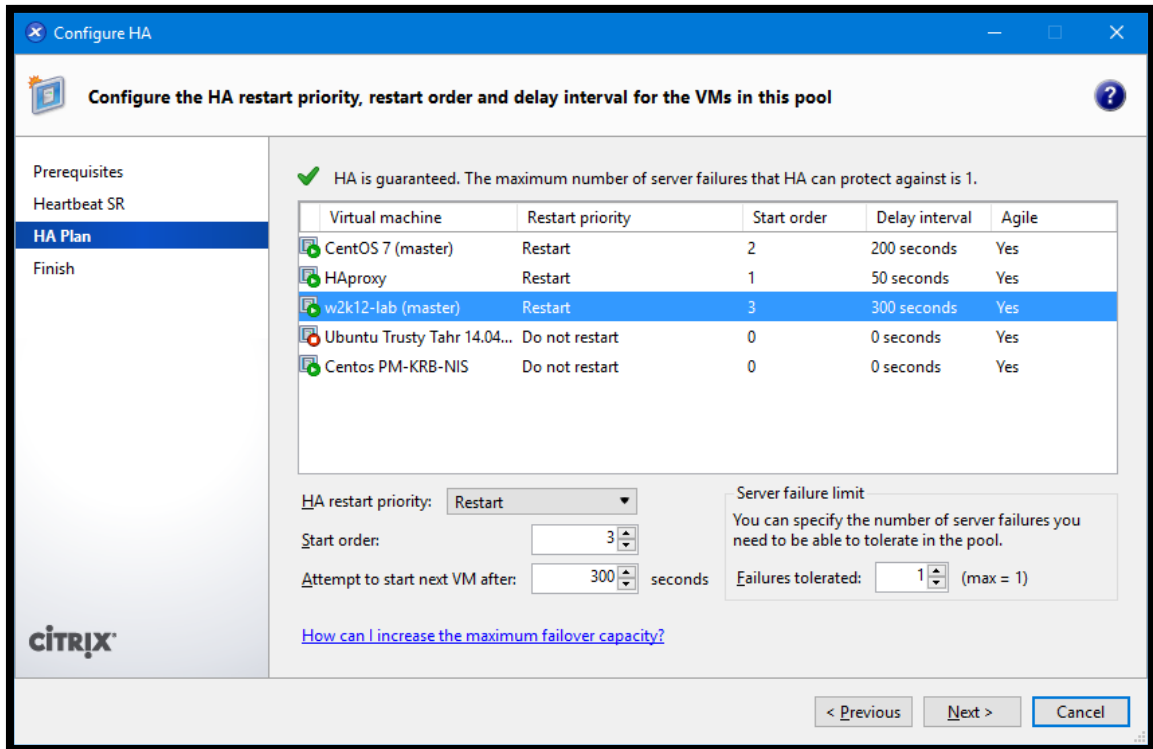


ILUSTRACIÓN 31 - ASISTENTE HA (3/4)

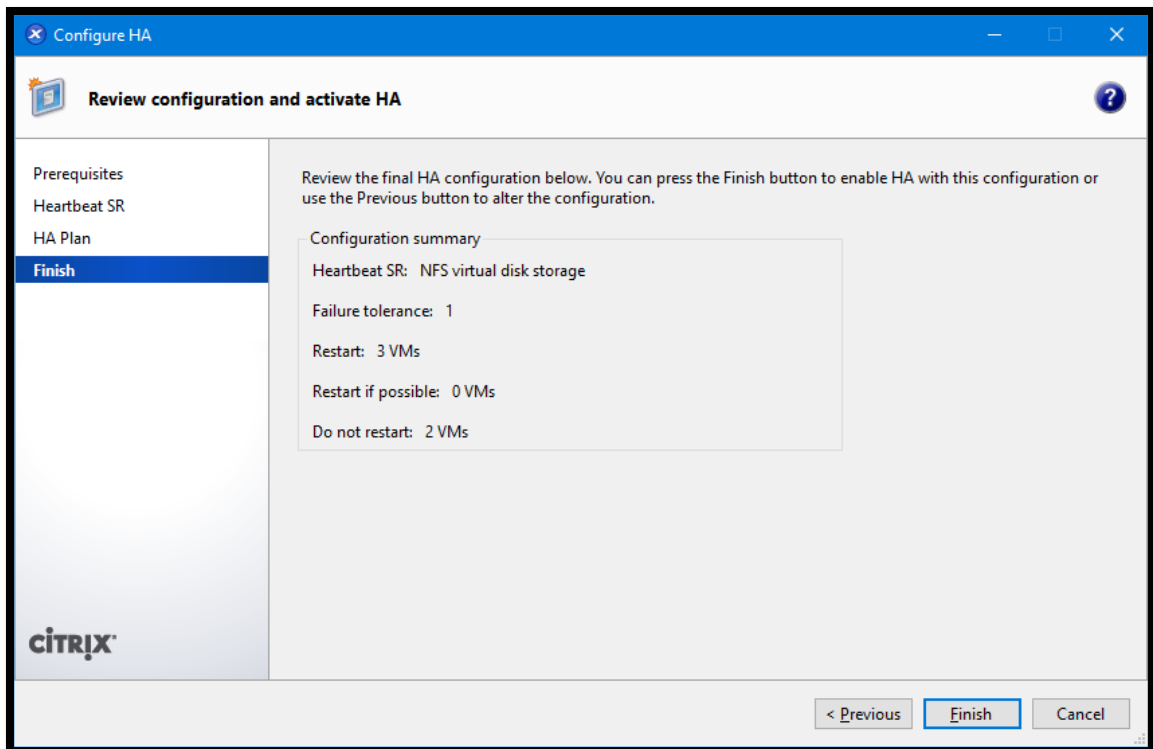


ILUSTRACIÓN 32 - ASISTENTE HA (4/4)



Es importante destacar que el plan es dinámico y evalúa constantemente la carga del sistema, y que no se pueden aplicar características expulsivas para las máquinas virtuales. Esto quiere decir que si, por ejemplo, intentamos arrancar la máquina virtual *Ubuntu* que se encuentra apagada, el sistema de alta disponibilidad recalcularía el plan con esas condiciones (nueva máquina encendida), y si no pudiera garantizar con ello el plan acordado (teniendo en cuenta los *hosts* disponibles y la ubicación de las máquinas virtuales dentro de cada uno de ellos), rechazaría el encendido de la máquina virtual *Ubuntu*, indicando que si se quiere realizar esa acción debemos reconfigurar el plan o desactivar la característica *HA*. Dotar al sistema de alta disponibilidad obligará a disponer de menos recursos.

En este capítulo se ha visto con profundidad el Centro de Proceso: la plataforma *XenServer*. Se ha comenzado mostrando el proceso de instalación que se repetirá en cada servidor, para posteriormente indicar como refinar su configuración gracias a las herramientas *XenCenter* y la *CLI*. Con *XenCenter* y la *CLI* se han integrado los servidores independientes en una unidad coordinada y centralizada, el *Pool TFG*. También se ha explicado la labor fundamental que realiza el Centro de Almacenamiento, que además de ofrecer el soporte para las máquinas virtuales y facilitar las imágenes de instalación, es usado para la mantener la comunicación activa entre los servidores, necesaria para activar la característica *HA* del *Pool TFG* y así alcanzar el objetivo de alta disponibilidad.

## 6. Bolsa de Servicios

El Centro de Proceso y el Centro de Almacenamiento ya están configurados y preparados para ejecutar y respaldar las Instancias de servicio que se encuentren en SERP, pero antes de dar servicio a los usuarios es necesario definir qué Servicio se va a instanciar. En este trabajo se prestan dos servicios: un escritorio de *Windows* y un escritorio de *Linux*. Una vez configurados correctamente la plataforma *XenServer* creará una plantilla de cada uno y estará en disposición de instanciarlas el número de veces apropiado, siempre conforme a sus recursos.

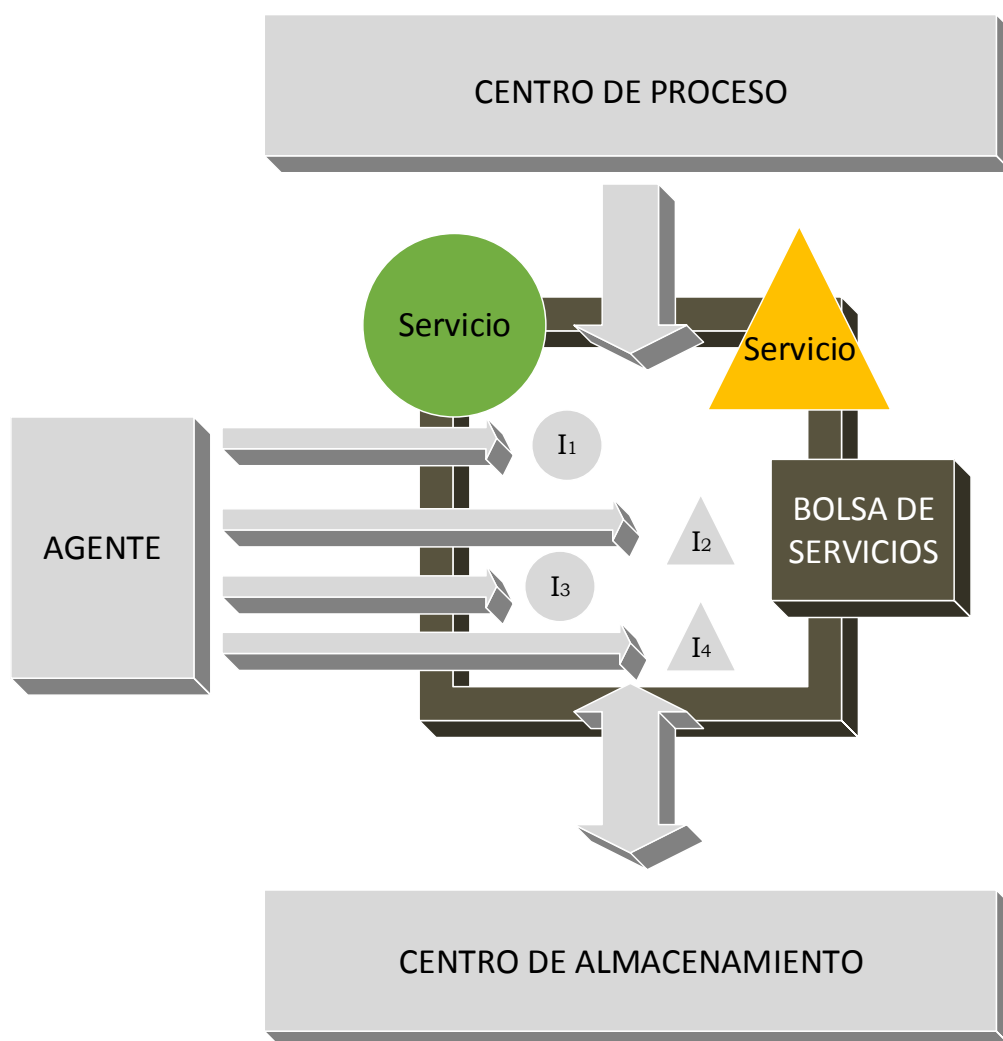


ILUSTRACIÓN 33 - SERP (BOLSA DE SERVICIOS)

En las secciones se 6.1 y 6.2 se explican respectivamente los pasos seguidos para instalar y configurar los servicios de escritorio remoto, tanto en *Windows* como *Linux*. La forma de hacerlo es significativamente distinta: en los equipos *Windows* está todo integrado dentro de la plataforma y solo se necesita instalar y configurar adecuadamente los

servicios, mientras que en los equipos *Linux* se deben tomar una serie de decisiones para componer el servicio.

## 6.1. Configuración de los servicios de escritorio remoto en Windows

Una vez creada la máquina virtual con *Windows server* (que habremos generado mediante alguno de nuestros servidores *xs-server*, por los métodos descritos en el capítulo 5, y que se encontrará alojada ya en nuestro Centro de Almacenamiento) nos dispondremos a configurar el servicio que deseamos que ofrezca. Los pasos previos a la configuración del servicio remoto serán, en primer lugar, instalar todas las actualizaciones recomendadas del sistema operativo, y seguidamente introducir el servidor en el dominio y asignar al grupo de administradores la cuenta del dominio que usemos para configurar el servicio. Para ofrecer el escritorio remoto en nuestros servidores *Windows* utilizamos la aplicación general de administración en *Windows Server 2012*, *Administrador del servidor* (la ilustración 34) e iniciamos el asistente *Agregar roles y características*.

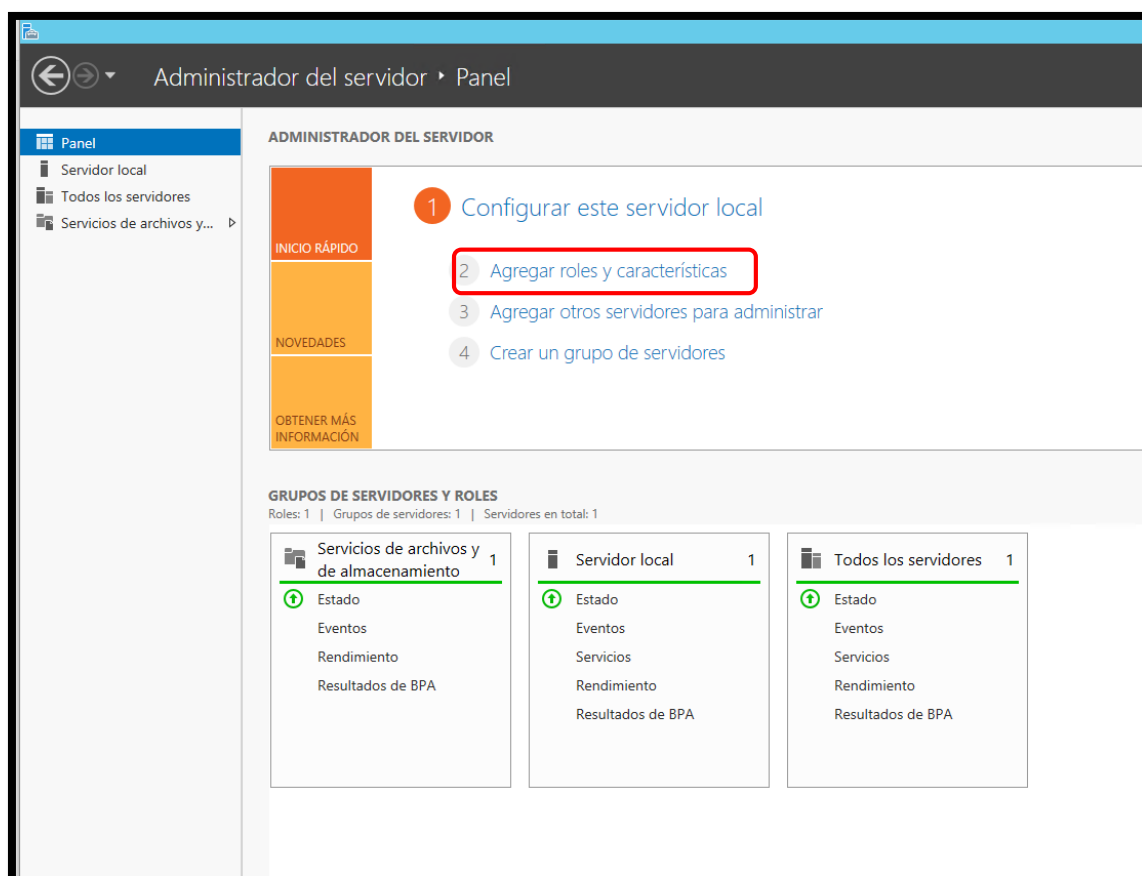


ILUSTRACIÓN 34 - PANEL DEL SERVIDOR W2K12-LAB

Si necesitaríamos hacer un despliegue completo (*Windows* utiliza el término de implementación) elegiríamos la *Instalación de Servicios de Escritorio remoto*, en la que asignaríamos a varios servidores los roles de agente de conexión, acceso web y *host* de sesión, pero dado que lo que deseamos es únicamente el servicio que permita conectar remotamente a los usuarios y encargar al Agente el rol de la conexión, desestimaremos esa solución de implementación y solo elegiremos el *host de escritorio remoto*, que es la parte de la implementación de *Windows* que se necesita. Como se ve en la ilustración 35 elegimos *Instalación basada en características o en roles* y pulsamos *Siguiente*.

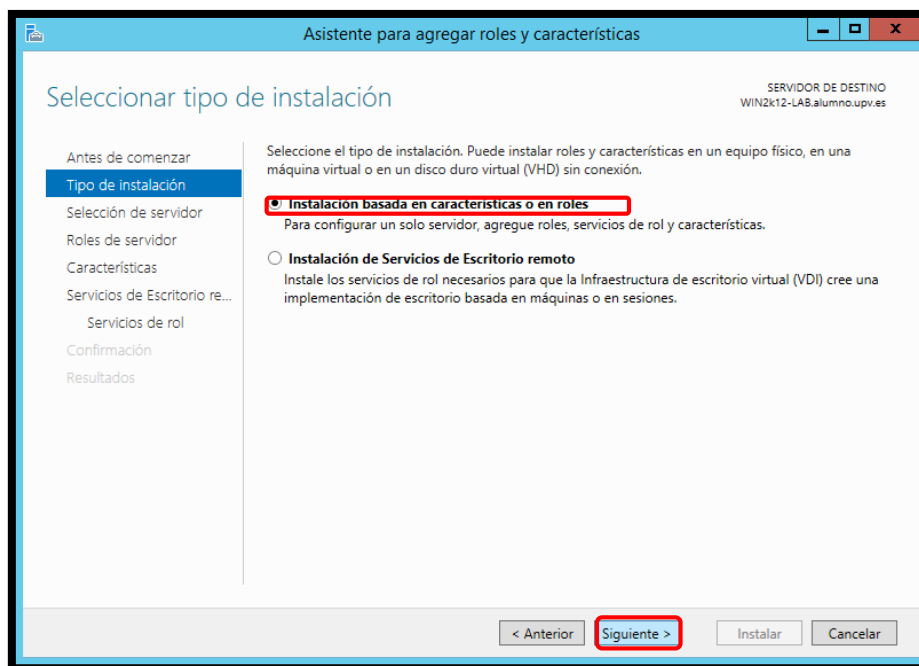


ILUSTRACIÓN 35 - SELECCIÓN BASADA EN ROLES

En la siguiente pantalla del asistente (ilustración 36) elegimos *win2k12-lab* y pulsamos *Siguiente*.

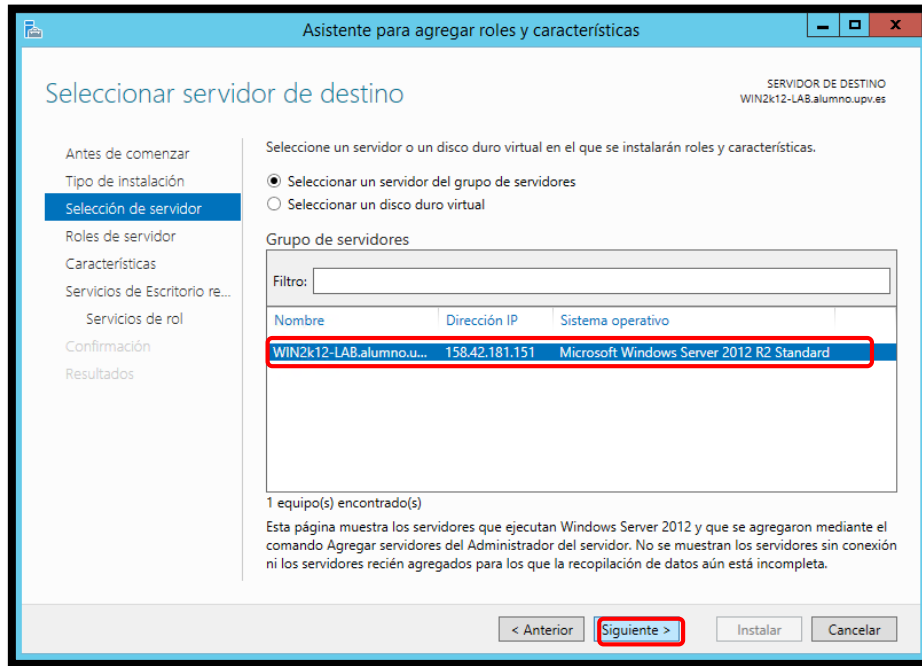


ILUSTRACIÓN 36 - SELECCIÓN DEL SERVIDOR

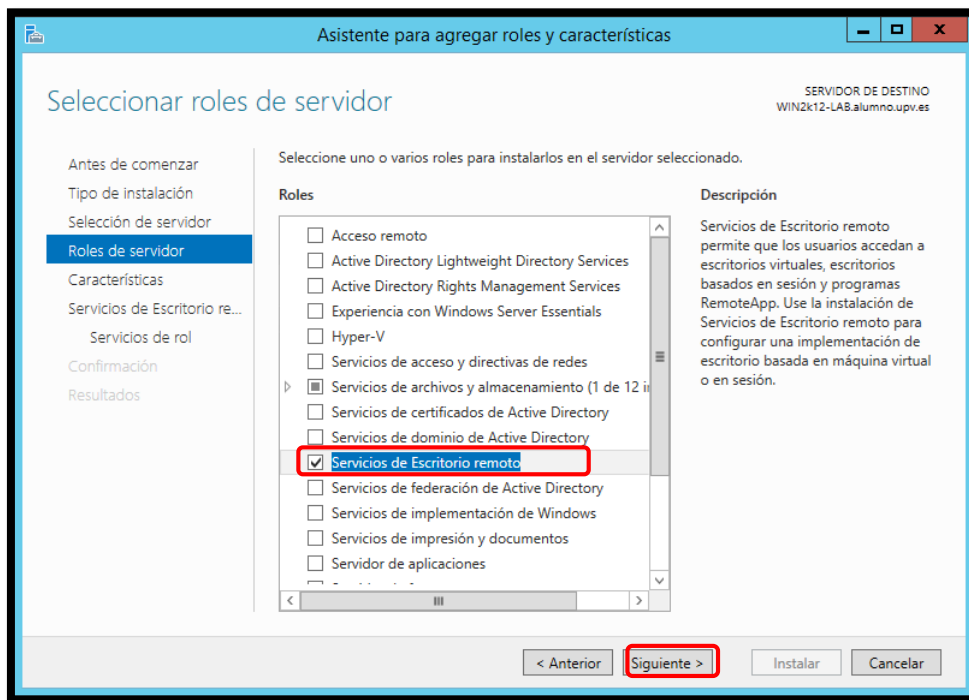


ILUSTRACIÓN 37 - SELECCIÓN DE LOS SERVICIOS DE ESCRITORIO REMOTO

Seleccionamos el rol de *Servicios de Escritorio remoto* (ilustración 37) y pulsamos *Siguiente*.



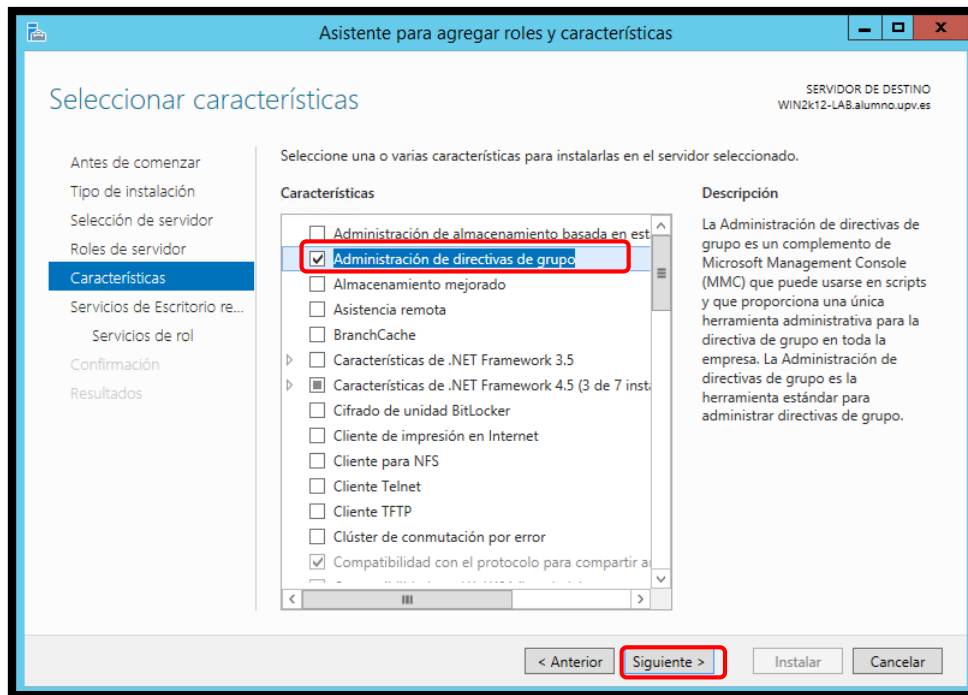


ILUSTRACIÓN 38 - ELECCIÓN DE CARACTERÍSTICAS ADICIONALES

En nuestro caso instalamos la característica de *Administración de directivas de grupo* (ilustración 38) para aplicarle más adelante configuraciones adicionales por políticas de dominio desde el mismo *win2k12-lab*, pero no es necesario ya que esta acción puede realizarse esta desde cualquier otra máquina *Windows* similar.

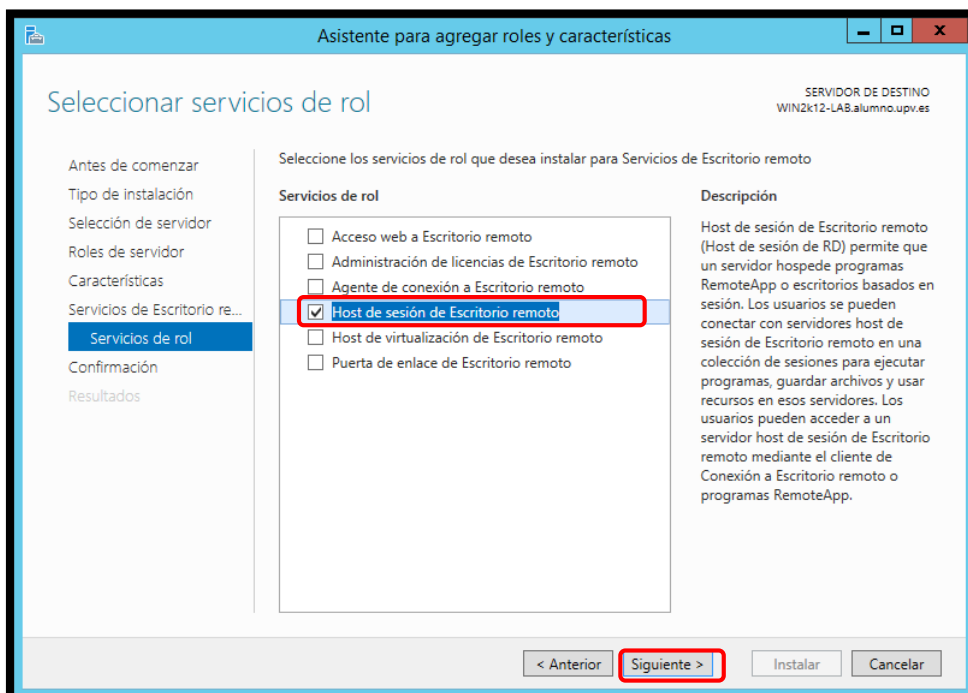


ILUSTRACIÓN 39 - SERVICIOS EXISTENTES EN EL ROL DE ESCRITORIO REMOTO

En la ilustración 39 se aprecia el penúltimo paso del asistente, y en este nos ofrece los seis roles disponibles en los *Servicios de Escritorio remoto*. No vamos a acceder a nuestras *Instancias Windows* a través de un navegador ni vamos a utilizar el servicio de balanceo de carga que ofrece el *Agente de conexión a Escritorio remoto*, por lo que no los instalamos (nos obligaría a instalar bastantes componentes adicionales que supondrían una carga innecesaria). Como tampoco vamos a hacer uso de la virtualización a través de *Hyper-V* (no necesitaremos el *host de virtualización de escritorio remoto*). Por último, el *Administrador de licencias de Escritorio remoto* solo lo utilizaríamos si fuéramos a instalar un servidor de licencias de escritorio remoto o si administráramos otro equipo que las sirviera, y tampoco es nuestro caso. Instalaremos únicamente el *Host de sesión de Escritorio remoto*. Pulsamos *Siguiente*.

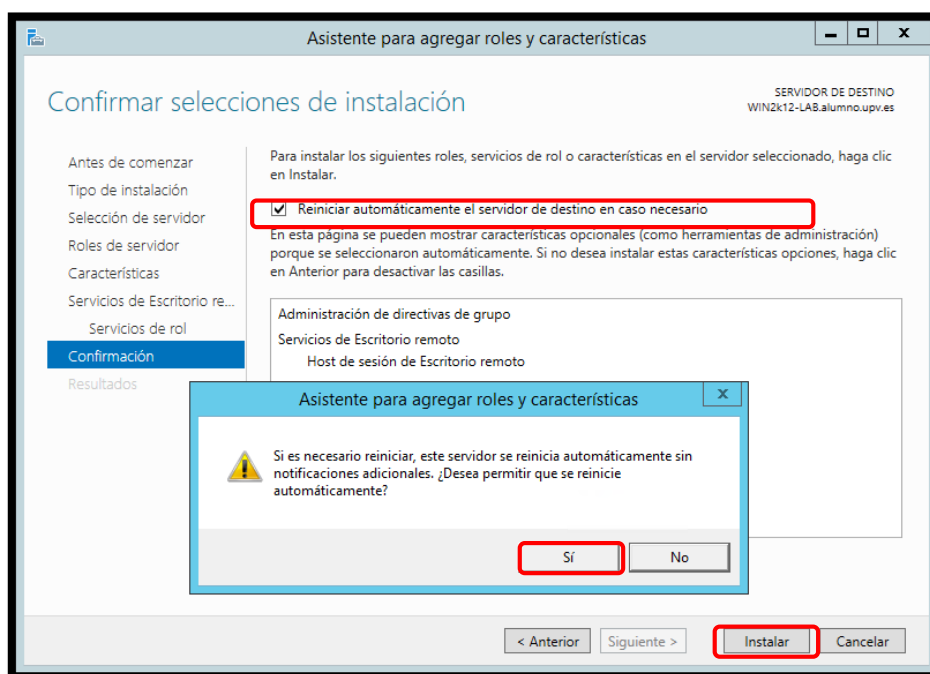


ILUSTRACIÓN 40 - CONFIRMACIÓN DE REINICIO

El servicio no estará disponible hasta el reinicio del sistema, así que podemos marcar la casilla apropiada y ordenar al sistema que en cuanto acabe las operaciones pendientes reinicie el equipo. Pulsamos *Sí* y seguidamente *Instalar* para que comience el proceso (Ilustración 40). En la ilustración 41 se observa el progreso de la instalación. Ya podemos terminar con el asistente pulsando *Cerrar*. Todo este proceso se ha seguido con una cuenta con permisos en el dominio ALUMNO.UPV.ES y permisos de administración en el servidor (cuenta *joaruig1@upvnet.upv.es*).

Terminado el asistente y reiniciada la máquina, para que nuestro servicio funcione como esperamos, deberemos realizar los siguiente tres pasos:

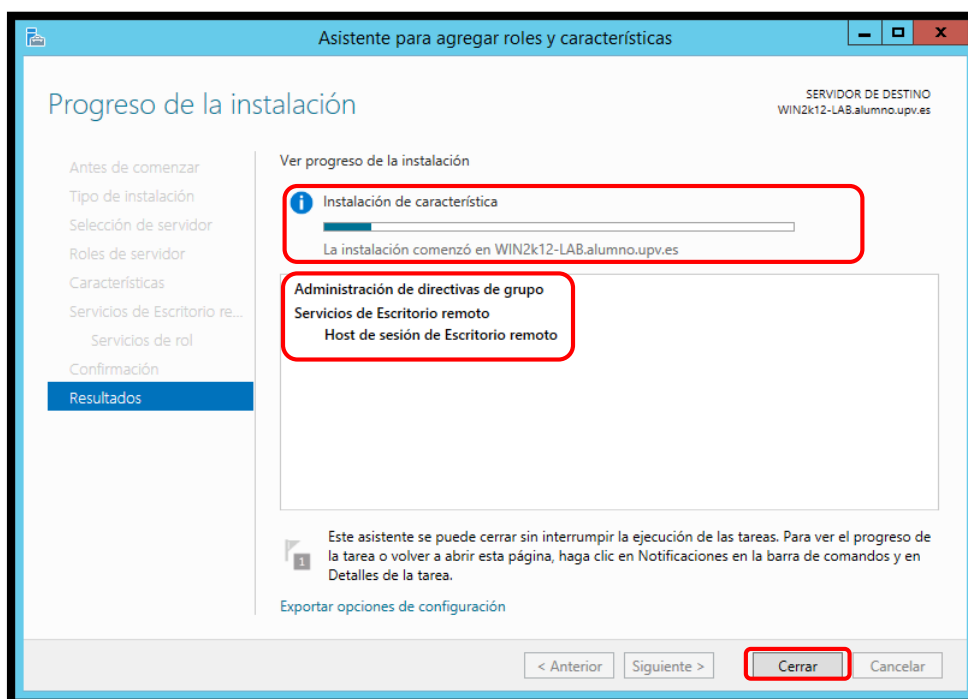


ILUSTRACIÓN 41 - INFORMACIÓN DE RESULTADOS DE INSTALACIÓN

**1.- Conectar nuestro servidor con un servidor de licencias.** El concepto de licencia es el mismo que se introducía en la sección 2.3.5 cuando se hablaba de las licencias de *Hyper-V* (las *CAL*) y se necesita una licencia por dispositivo o por usuario. La gestión del número de licencias disponibles, así como concederlas, depende de un *Servidor de licencias de Escritorio remoto*. Al no disponer de uno propio deberemos indicar a nuestro *Host de sesión* dónde encontrarlo. En la infraestructura de *AD* de la UPV existen varios servidores para ese fin. En nuestro caso elegiremos a *ADUMAR.CC.UPV.ES*, que es un *Servidor de licencias de Escritorio remoto*.

**2.- Seleccionar el modo de licencia.** Deberemos elegir entre *por usuario* o *por dispositivo*. Siguiendo recomendaciones del ASIC<sup>22</sup> elegiremos por dispositivo.

**3.- Seleccionar los usuarios.** Para permitir el inicio de sesión de los grupos o usuarios que determinemos tendremos que autorizarlos mediante la pertenencia al grupo de *Usuarios de Escritorio remoto*.

<sup>22</sup> Área de Sistemas Informáticos y Computación en la Universitat Politècnica de València. Nos indican que para minimizar el uso de *CAL* debemos seleccionar la opción de dispositivos.

Para los dos primeros necesitaremos hacer uso de las directivas de grupo (nos valdrán las locales). Esto lo haremos ejecutando el complemento directivas de grupo local (ilustración 42).



ILUSTRACIÓN 42 – POLÍTICAS DE GRUPO (LÍNEA DE COMANDO)

Como podemos ver en la ilustración 43, recorreremos las ramas:

*Directiva Equipo local -> Configuración del equipo -> Plantillas administrativas -> Componentes de Windows -> Servicios de Escritorio remoto*

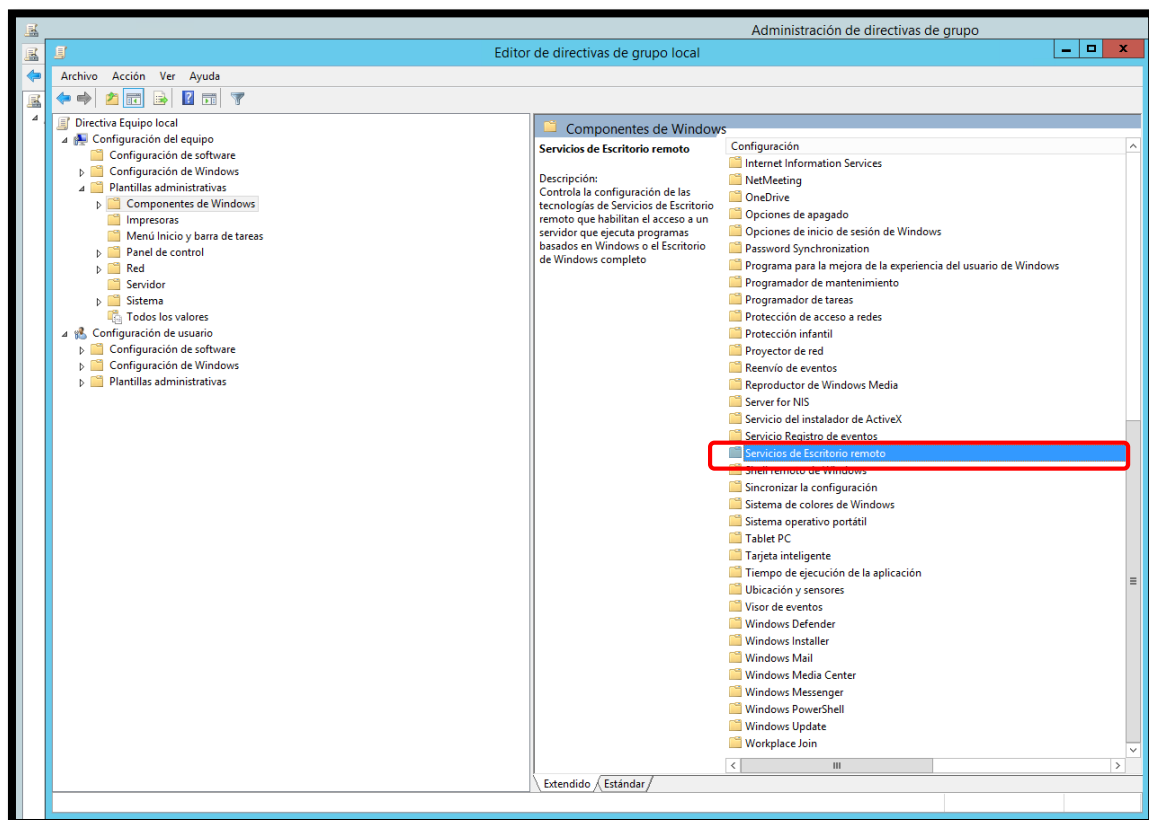


ILUSTRACIÓN 43 - UBICACIÓN EN LAS POLÍTICAS LOCALES

Aquí encontramos tres componentes (entre llaves), de los cuales dos requieren nuestra atención (ilustración 44):

- Usar los servidores de licencias de Escritorio remoto especificados
- Establecer el modo de licencia de Escritorio remoto.

Usaremos ambos para completar la información del servicio (ilustraciones 44 y 45).

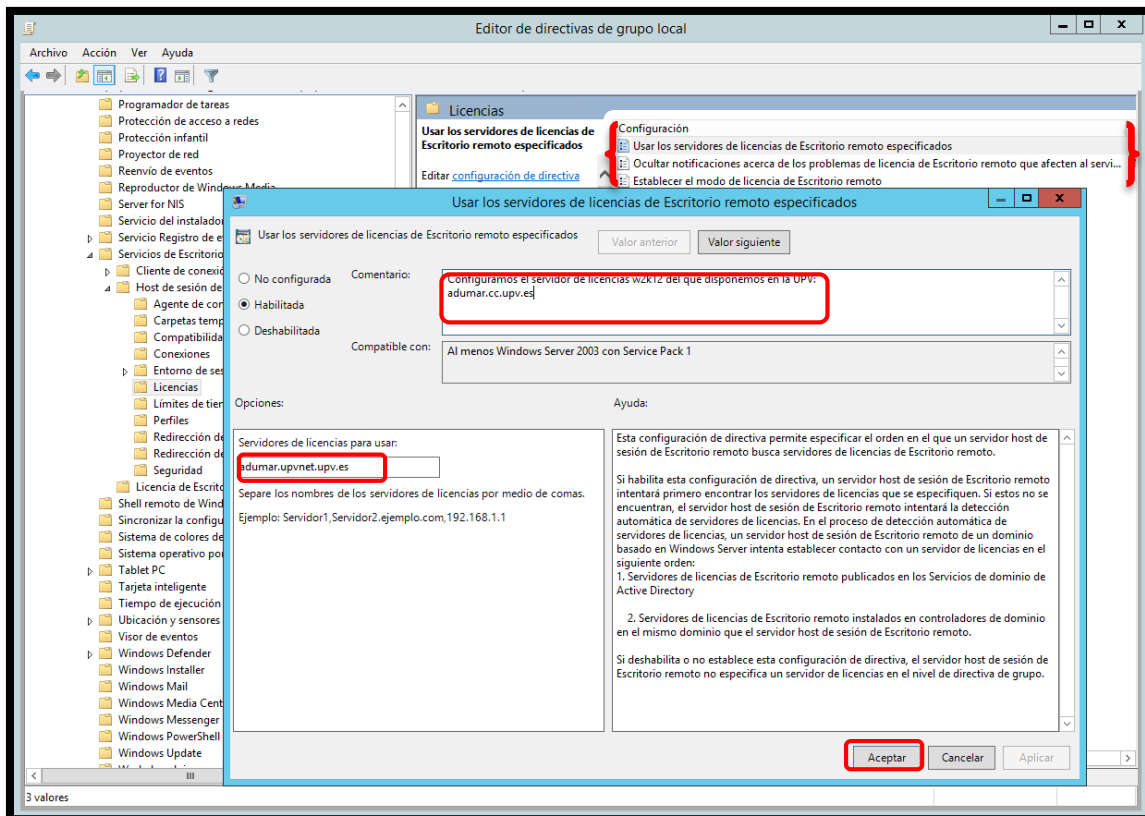


ILUSTRACIÓN 44 - INFORMACIÓN DEL SERVIDOR DE LICENCIAS

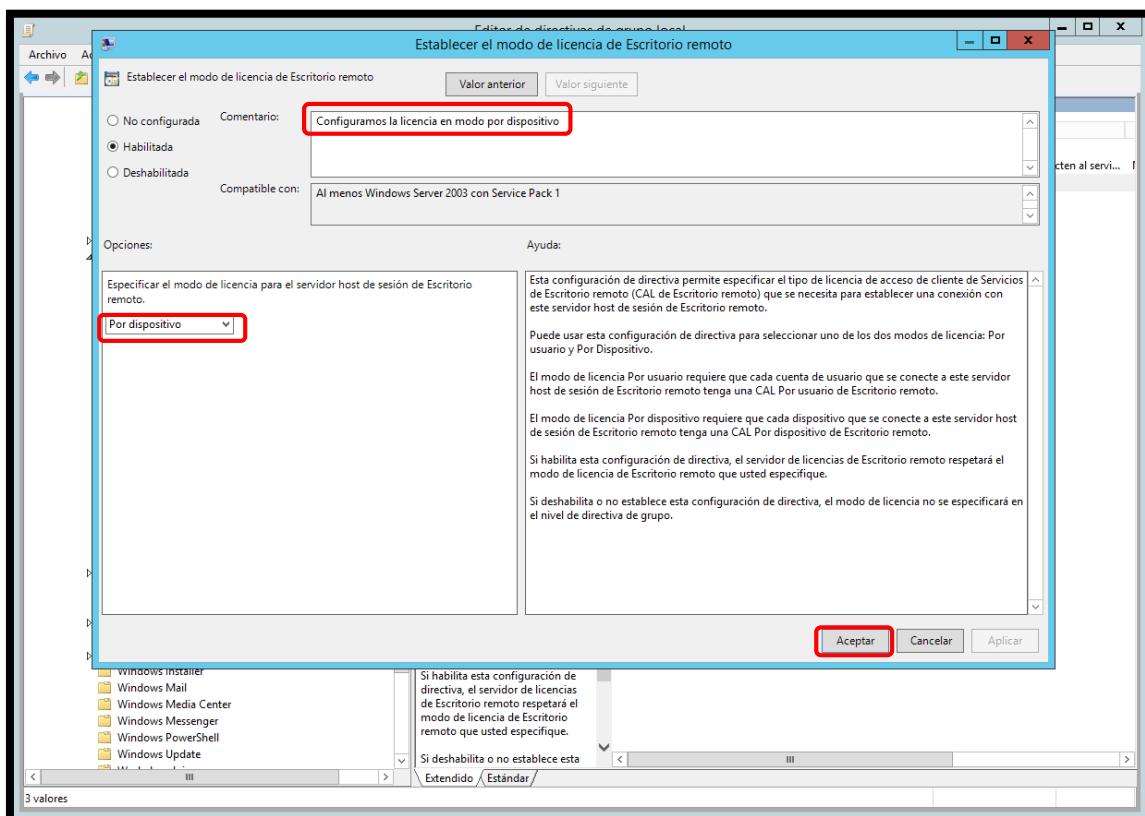


ILUSTRACIÓN 45 - CONFIGURACIÓN DEL TIPO DE LICENCIA



Por último, solo restará introducir los usuarios que deseamos que hagan uso en el grupo adecuado.

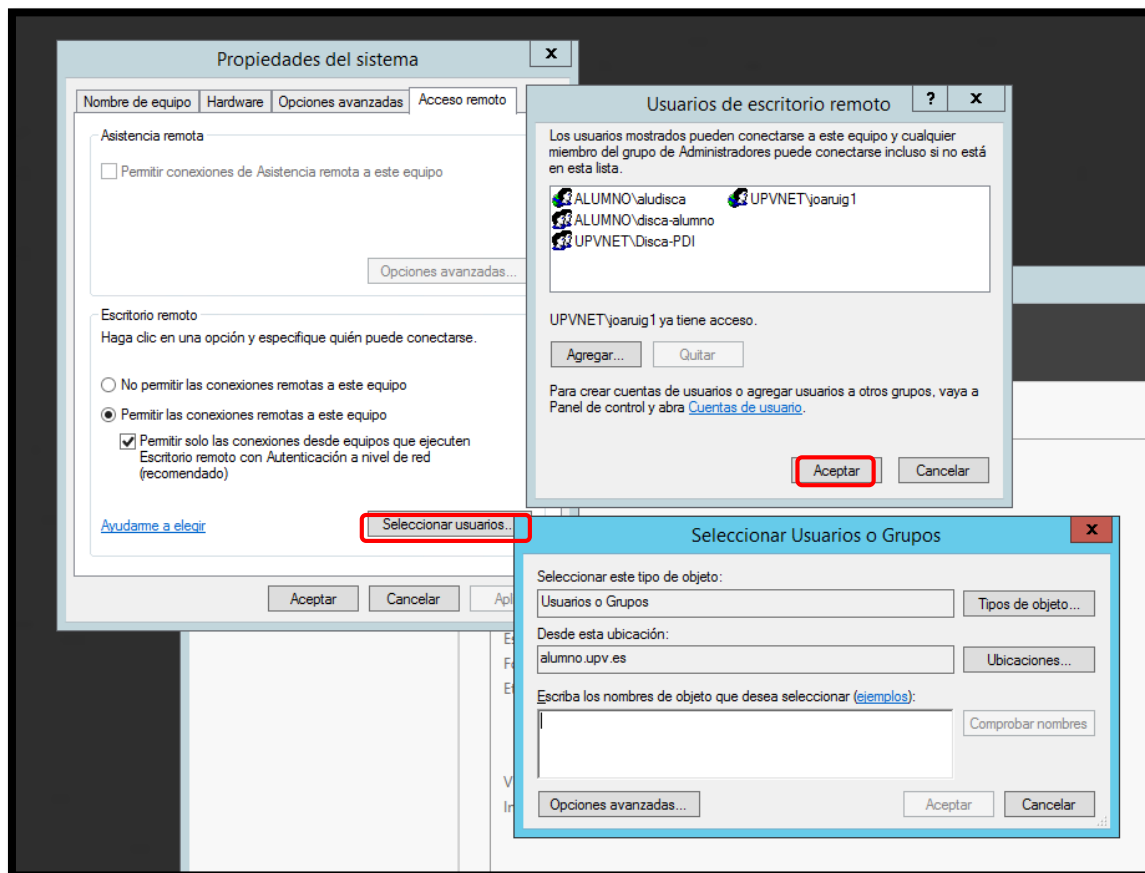


ILUSTRACIÓN 46 - ACTIVACIÓN Y SELECCIÓN DE USUARIOS

La configuración principal ha concluido. Ahora solo nos queda, desde el Centro de Proceso, convertir esta máquina virtual en una plantilla que nos permita replicarla y crear las *Instancias de servicio*. El proceso de configuración de la plantilla (*Servicio*) no acabará aquí; sucesivamente habrá que ir modificando la plantilla para adecuarla a las necesidades de los usuarios, principalmente realizando la instalación de las aplicaciones necesarias.

## 6.2. Configuración de los servicios de escritorio remoto en Linux

Para ofrecer el servicio de escritorio remoto en *Linux* se ha optado por una distribución *CentOS*. *CentOS 7* tiene soporte de *XenServer 6.5* y se ha instalado con el entorno gráfico de *Gnome* en sus versiones 2 y 3 (*Gnome-Shell*) (siguiendo los mismos pasos descritos al principio de la sección 6.1 en lo referente a la generación y almacenamiento de la

máquina virtual). Para proveerlo del servicio de escritorio remoto, en primera instancia se instaló y configuro el servidor *Xrdp*, que tiene la ventaja de que para acceder no hace falta un cliente concreto. Utilizar este servidor (*Xrdp*) habría facilitado a los usuarios realizar el acceso desde sus sistemas operativos con las aplicaciones habituales de escritorio remoto. Sin embargo, se obtuvo un rendimiento gráfico algo deficiente, por lo que se descartó y se buscaron aplicaciones que pudieran ofrecer la misma funcionalidad, pero con mejor desempeño. De entre los posibles candidatos, finalmente se seleccionó *Thinlinc*,

*ThinLinc* es una aplicación cliente y servidor gratuito (el servidor es gratuito para soportar hasta diez sesiones simultáneas). Esto último podría ser un problema, pero por las características de nuestro modelo no es necesario superar ese límite por *Instancia de servicio*, ya que si nuestro servicio necesita soportar más sesiones basta con crear *Instancias de servicio* adicionales. *ThinLinc* es un *framework* construido en base a soluciones de software libre, pero conjugado con otros módulos propietarios. Al igual que la *Implementación de Escritorio remoto* que ofrece *Windows Server 2012*, ofrece una carta de servicios amplia que permiten balanceo de carga, agente de conexión, autenticación sobre *AD*, *LDAP* o *eDirectory*<sup>23</sup> de *Novell* y gestión de impresión entre otras. También, al igual que con *Windows*, no se va a hacer uso de esos servicios: la autenticación se resuelve a nivel del sistema operativo (con los servicios necesarios) y la gestión de la conexión y el balanceo de carga lo realizará el *Agente*.

Para llevar a cabo la integración de nuestro equipo dentro del directorio activo de la UPV necesitaremos hacer uso de *kerberos* para validar, *PAM* para autenticar y *NIS* para proveer de un espacio de trabajo propio. Existen soluciones de terceros, como *Centrify* [13], que podrían virtualmente convertir los equipos *Linux* en *Windows* de cara a la integración de estos en el directorio, pero dado que existe un espacio de almacenamiento *NIS*, vamos a utilizar esta tripleta *KPN*<sup>24</sup>. En las secciones 6.2.1, 6.2.2, 6.2.3 y 6.2.4 explicaremos respectivamente *kerberos*, *PAM*, *NIS* y *Thinlinc*.

### 6.2.1. Configuración *kerberos*

*Kerberos* es un protocolo de autenticación de red que mediante el uso de criptografía proporciona un método seguro para realizar una autenticación entre cliente y servidor.

---

<sup>23</sup> *AD*, *LDAP* o *eDirectory* son implementaciones de un servicio de directorio en una red distribuida.

<sup>24</sup> *Kerberos*, *PAM* y *NIS*.

El mecanismo de autenticación de *Kerberos* se basa en *tickets* para tener acceso a los servicios de red. Estos *tickets* contienen datos cifrados, e incluyen una contraseña cifrada para confirmar la identidad del usuario al servicio solicitado. Todo el proceso de autenticación pasa desapercibido para el usuario.

Cada entidad que utiliza el sistema *Kerberos*, sea un usuario o un servidor de red, es en este sentido un cliente, pues actúa como tal contra el servicio *Kerberos*.

El servicio *kerberos* se distribuye entre los *KDC* (*Key Distribution Center*) y los *TGS*, (*Ticket-Granting Service*). Los *KDC* almacenan las contraseñas de los clientes, así como de cualquier otro servidor que preste en la red un servicio que precise de validación. Los servidores deben registrarse previamente, al igual que los usuarios. El proceso de autenticación de *kerberos* comienza cuando un cliente solicita acceso a un recurso registrado. Una vez que *kerberos* valida al cliente (comprueba que es quien dice ser) el *KDC* le emite un *ticket* especial que le permitirá obtener más *tickets*. El cliente usa el *ticket* obtenido para acceder al *TGS* para que este emita a su vez un *ticket de servicio* para que el cliente pueda acceder al recurso de red solicitado inicialmente. El *ticket* que presenta el cliente al servicio de red solicitado, después de ser procesado, garantiza que tanto el cliente como servidor han sido correctamente identificados.

En un equipo CentOS 7 instalado y actualizado, preparamos nuestro Servicio para que use *kerberos* y los usuarios que se conecten puedan validarse en los servidores de la red UPV. Para hacer esto deberemos instalar los paquetes cliente en nuestra máquina virtual y configurarla adecuadamente. Este proceso lo haremos empleando el gestor de paquetes de la distribución *CentOS 7*, que tiene disponible en su repositorio los paquetes demandados. Para consultar los paquetes de *kerberos* utilizamos la herramienta que utiliza *CentOS*: *yum*.

Para buscar y que nos muestre los paquetes disponibles usamos *yum* junto con *search*.

```
[root@xs-centos7-0~]# yum search kerberos
Complementos cargados:fastestmirror, langpacks
Loading mirror speeds from cached hostfile
 * base: mirror.brutalsys.com
 * extras: mirror.brutalsys.com
 * updates: mirror.brutalsys.com
===== N/S matched: kerberos
=====
python-kerberos.x86_64 : A high-level wrapper for Kerberos (GSSAPI) operations
freeradius-krb5.x86_64 : Kerberos 5 support for freeradius
krb5-devel.i686 : Development files needed to compile Kerberos 5 programs
krb5-devel.x86_64 : Development files needed to compile Kerberos 5 programs
krb5-libs.i686 : The shared libraries used by Kerberos 5
krb5-libs.x86_64 : The shared libraries used by Kerberos 5
krb5-pkinit.x86_64 : The PKINIT module for Kerberos 5
krb5-server.x86_64 : The KDC and related programs for Kerberos 5
krb5-server-ldap.x86_64 : The LDAP storage plugin for the Kerberos 5 KDC
```



```

krb5-workstation.x86_64 : Kerberos 5 programs for use on workstations
mod_auth_kerb.x86_64 : Kerberos authentication module for HTTP
PAM_krb5.i686 : A Pluggable Authentication Module for Kerberos 5
PAM_krb5.x86_64 : A Pluggable Authentication Module for Kerberos 5
python-krbV.x86_64 : Python extension module for Kerberos 5
realmd.x86_64 : Kerberos realm enrollment service
sssd-krb5.x86_64 : The Kerberos authentication back end for the SSSD
sssd-krb5-common.i686 : SSSD helpers needed for Kerberos and GSSAPI
                        : authentication
sssd-krb5-common.x86_64 : SSSD helpers needed for Kerberos and GSSAPI
                        : authentication

```

Observaremos los paquetes de *kerberos* que tenemos disponibles en el repositorio. El paquete que necesitamos es *krb5-workstation*, y para usar *kerberos* de forma conjunta con *PAM* instalamos también el paquete *PAM\_krb5*

```

[root@xs-centos7-0 ~]# yum install krb5-workstation PAM_krb5
=====
Package                Arquitectura  Versión           Repositorio       Tamaño
=====
Instalando:
krb5-workstation      x86_64       1.12.2-15.el7_1  updates          755 k
PAM_krb5              x86_64       2.4.8-4.el7      base              158 k

```

Junto con estos paquetes se instalarán otros para cumplir las dependencias. Una vez instalado hay que completar el fichero de configuración con la información adecuada. La configuración de *kerberos* se realiza en el fichero */etc/krb5.conf* que podemos ver a continuación, añadiendo las líneas resaltadas en azul.

Las líneas añadidas indican a *kerberos* cuáles son los dominios configurados. Toda la información al respecto se puede encontrar en [13].

```

[logging]
default = FILE:/var/log/krb5libs.log
kdc = FILE:/var/log/krb5kdc.log
admin_server = FILE:/var/log/kadmind.log
[libdefaults]
dns_lookup_realm = false
ticket_lifetime = 24h
renew_lifetime = 7d
forwardable = true
rdns = false
default_realm = UPVNET.UPV.ES
default_realm = ALUMNO.UPV.ES
default_ccache_name = KEYRING:persistent:%{uid}
dns_lookup_kdc = false
[realms]
ALUMNO.UPV.ES = {
  kdc = alya.cc.upv.es
  kdc = alsafi.cc.upv.es
  kdc = diadem.cc.upv.es
  kpasswd_server = alya.cc.upv.es
  kpasswd_server = alsafi.cc.upv.es
  kpasswd_server = diadem.cc.upv.es
# admin_server = kerberos.example.com
}
UPVNET.UPV.ES = {
  kdc = juno.cc.upv.es
  kdc = adib.cc.upv.es
  kdc = adara.cc.upv.es

```



```

kpasswd_server = juno.cc.upv.es
kpasswd_server = adib.cc.upv.es
kpasswd_server = adara.cc.upv.es
}
[domain_realm]
.alumno.upv.es = ALUMNO.UPV.ES
.upvnet.upv.es = UPVNET.UPV.ES
# example.com = EXAMPLE.COM

```

Se ha introducido la información básica del dominio: el nombre, tres *KDC* en los que validarán nuestras credenciales, e información acerca del servidor para cambiar de contraseñas (*kpasswd\_server*) que como vemos son los mismos *KDC*.

Para comprobar que funciona (no necesita reiniciar ningún servicio) bastará con usar en línea de comando alguna utilidad como *kinit*.

```

[root@xs-centos7-0 /]# kinit joaruig1
Password for joaruig1@UPVNET.UPV.ES:

```

Si se introduce la contraseña y se pulsa *intro*, simplemente insertará un retorno de carro y volverá a mostrar el *prompt* (en caso de que todo haya ido bien) y el sistema habrá recogido un *ticket* de *kerberos* con nuestro nombre. Esto se puede comprobar con el siguiente comando:

```

[root@xs-centos7-0 /]# klist
Ticket cache: KEYRING:persistent:0:0
Default principal: joaruig1@UPVNET.UPV.ES
Valid starting    Expires          Service principal
26/11/15 16:56:46  27/11/15 02:56:46  krbtgt/UPVNET.UPV.ES@UPVNET.UPV.ES
                renew until 03/12/15 16:56:36

```

Con *klist* se nos mostrará la información del *ticket* o los *tickets* de los que podamos conseguir. Con esto ya habríamos acabado de configurar *kerberos*.

### 6.2.2. Configuración PAM

*PAM*, o *Linux-PAM (Pluggable Authentication Modules for Linux)* es un conjunto de librerías que permiten definir en el sistema cómo las aplicaciones autenticarán a los usuarios. Básicamente, consiste en una pila de condiciones que indican los requisitos y acciones que se deben realizar para que un usuario tenga acceso a una aplicación que requiere autenticación [14]. *PAM* se extiende en varios ámbitos, no solo en la autenticación de los usuarios, pudiendo desarrollar otras funciones previas o posteriores al inicio de sesión o incluso comprobar la robustez de la contraseña. *PAM* está incluido con prácticamente todas las distribuciones de *Linux* y resulta transparente al usuario. En nuestro caso no ha sido necesario realizar directamente modificación alguna, ya que el asistente de configuración del *NIS* se encargará de realizar los cambios pertinentes en sus archivos de configuración.

### 6.2.3. Configuración NIS

*NIS* es un protocolo de servicios de directorio cliente/servidor desarrollado por *Sun Microsystems*, y antes era el más ampliamente utilizado en sistemas *Linux* en red para centralizar y administrar las cuentas de usuarios y proveerlos de servicios comunes, como el *home* de usuario. En nuestro caso contamos con un servicio *NIS* que se usa para trasladar cuentas y grupos que se usan en el *directorio activo* de la UPV y así poder conceder a través de esa identidad permisos a recursos, como puede ser el propio *home* que les sirve. Para poder usar este *NIS* necesitaremos instalar los dos paquetes mostrados (y las dependencias que se generen):

| Package                           | Arquitectura | Versión        | Repositorio | Tamaño |
|-----------------------------------|--------------|----------------|-------------|--------|
| Instalando:                       |              |                |             |        |
| <b>ypbind</b>                     | x86_64       | 3:1.37.1-7.e17 | base        | 62 k   |
| Instalando para las dependencias: |              |                |             |        |
| <b>yp-tools</b>                   | x86_64       | 2.14-3.e17     | base        | 79 k   |
| Resumen de la transacción         |              |                |             |        |

No se va a utilizar el *NIS* directamente ni para validar ni para autenticar, simplemente, si el usuario que intenta acceder existe dentro del directorio del *NIS* quedará englobado en una serie de grupos, y la pertenencia a esos grupos le otorgará acceso a determinados recursos. Por otra parte, a nivel de usuario, tendrá acceso a su *home* personal.

Para hacer funcionar *NIS* en el cliente necesitaremos indicar quién es el servidor y el cual es el dominio *NIS*. Para hacer esto hay que modificar la configuración de varios servicios y ficheros, pero existen varias herramientas que lo pueden hacer por nosotros. En nuestro caso utilizaremos *authconfig-tui*, de la forma que muestran las siguientes ilustraciones.

Desde la línea de comando accederemos a la aplicación:

```
[root@xs-centos7-0 ~]# authconfig-tui
```

Como se puede observar en la ilustración 47, seleccionamos utilizar *NIS* y *kerberos* (las demás opciones estaban marcadas por defecto y no es necesario modificarlas).

El dominio que vamos a usar es *labs.disca.upv.es* y como servidores ponemos las *IP* correspondientes: 158.42.180.[93|94], que son los equipos que realizan este servicio (servidor principal y de respaldo)(ilustración 48) .

Las opciones que aparecen en la ilustración 49 aparecen rellenas. Esto sucede porque en la sección 6.2.1 ya configuramos el archivo */etc/krb5.conf*.

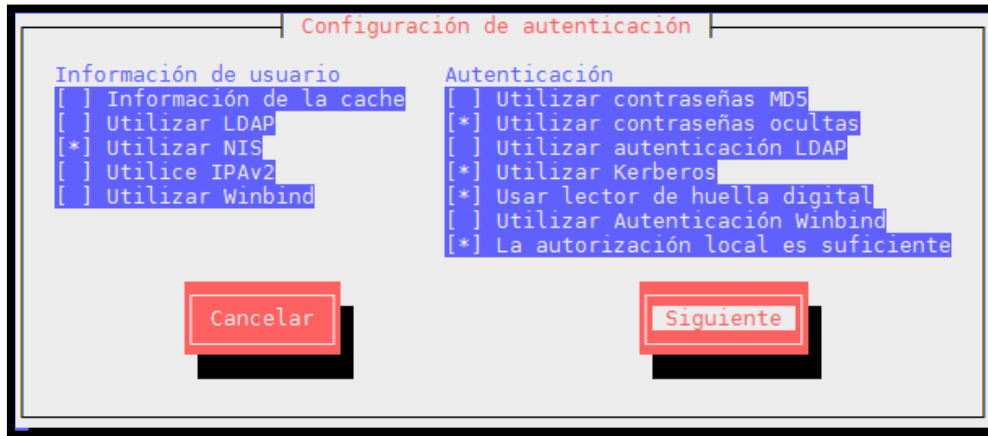


ILUSTRACIÓN 47 - ASISTENTE AUTHCONFIG (1/3)

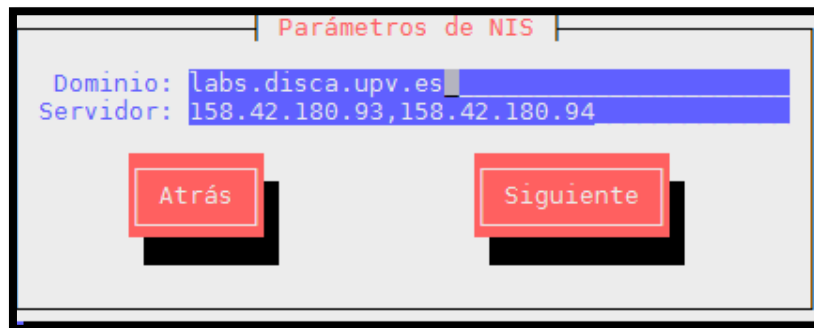


ILUSTRACIÓN 48 - ASISTENTE AUTHCONFIG (2/3)

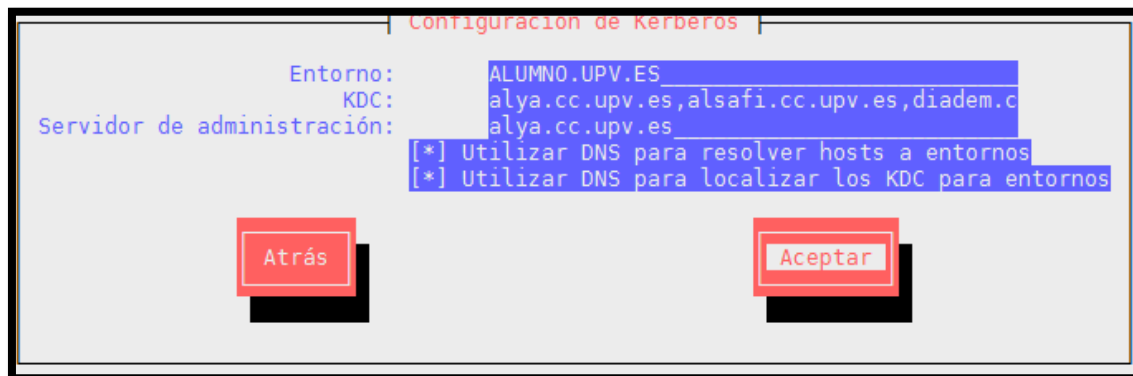


ILUSTRACIÓN 49 - ASISTENTE AUTHCONFIG (3/3)

Para comprobar que la configuración ha sido correcta se hace una solicitud al NIS, por ejemplo, sobre un usuario:

```
[root@xs-centos7-0 ~]# getent passwd joaruig1  
joaruig1:x:1692415552:1692452651:Joaquin Ruiz Gil:/home/UPVNET/joaruig1:/bin/bash
```

Al responder con la información del usuario no hay duda que funciona

Ahora solo queda montar en el sistema el recurso para alojar el *home*. Para hacer esto modificaremos el fichero */etc/fstab* para que contenga la información del dispositivo y el punto de montaje y el fichero */etc/hosts* para usar nombres cortos:

```
[root@xs-centos7-0 ~]# cat /etc/fstab
/dev/mapper/centos_xs--linux--1-root          /          xfs      defaults    0 0
UUID=75673621-aacb-4853-ad4a-5eac97a06ae1    /boot      xfs      defaults    0 0
/dev/mapper/centos_xs--linux--1-swap         swap       swap     defaults    0 0
discadct3:/home                              /home     nfs      defaults,nolock
```

```
[root@xs-centos7-0 ~]# cat /etc/hosts
158.42.180.93 discadct3 discadct3.disca.upv.es
158.42.180.94 discadct4 discadct4.disca.upv.es
127.0.0.1    localhost localhost.localdomain localhost4 localhost4.localdomain4
::1         localhost localhost.localdomain localhost6 localhost6.localdomain6
```

Con esto se da por concluida la configuración de autenticación, validación y soporte de espacio de usuario (*home*).

## 6.2.4. Configuración *Thinlinc*

Tal y como introducimos en el punto 6.2, *ThinLinc* es una aplicación con cliente/servidor que dentro de SERP, y concretamente en el *Servicio de Escritorio Linux*, va a ser integrada para configurar y dar servicio a la parte gráfica de las sesiones.

*ThinLinc* funciona usando el cliente *VNC TigerVNC* con túneles *ssh*, y entrega a cada sesión un servidor (gráfico) dedicado. Según apuntábamos en 6.2, *xrdp* fue desestimado puesto que su desempeño gráfico resultaba deficiente, y entre varias alternativas (*nxFree*, *X2go* y el propio *Thinlinc*) se optó por esta solución. El desempeño gráfico es correcto y no ha presentado ningún problema de estabilidad.

El inconveniente de esta solución, como también se comentó en la sección 6.2, es que necesitará que el usuario se descargue e instale el cliente específico, que se puede descargar desde su página web, así como su documentación [15]. Para poder descargar el software (del servidor) es necesario registrarse. Una vez descargado el software, procedemos a instalar el programa servidor en nuestra máquina.

Para instalar el programa descomprimiremos el archivo .zip en el que viene empaquetado y ejecutaremos el instalador (ilustración 50).

El asistente instalará los paquetes que proporcionan todos los servicios de *ThinLinc* y solicitará confirmación para instalar dependencias y servicios. El programa nos pregunta la función del servidor: *master* o *agent*; elegimos *master* (instala las dos partes). Para funcionar con todos sus servicios pide abrir los puertos 22, 300, 1010, 9000 y 904. El proceso finaliza y el servicio ya está listo.



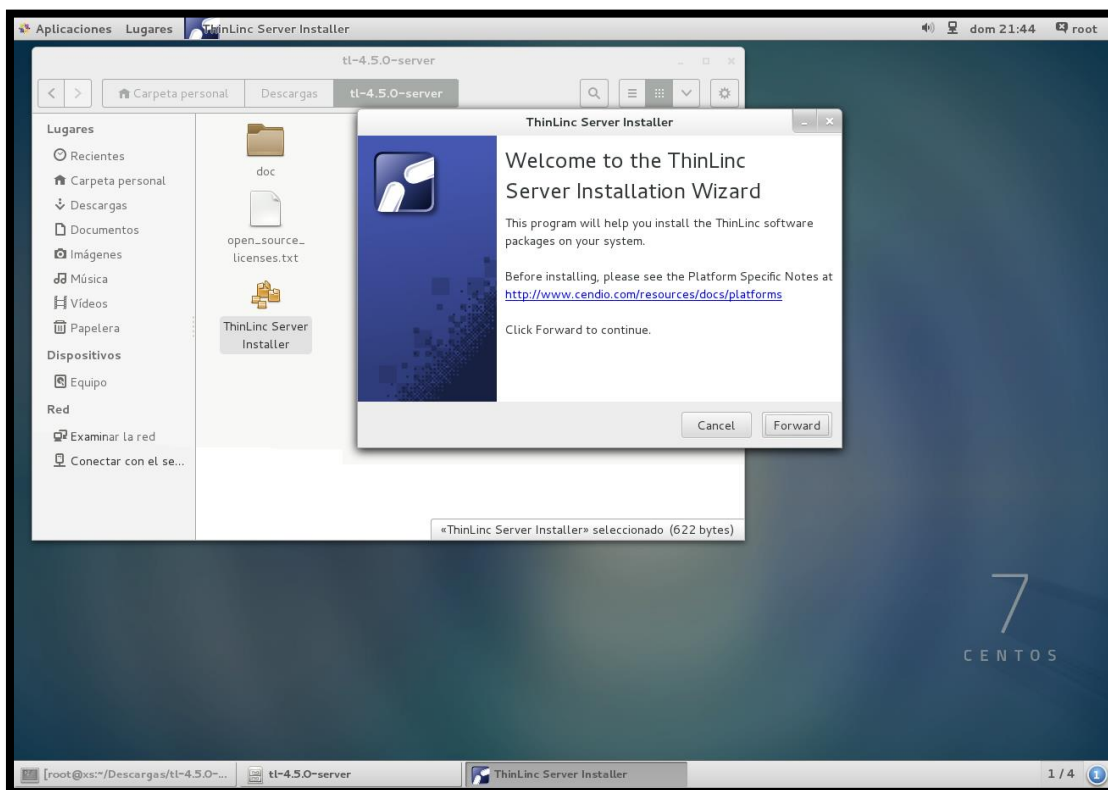


ILUSTRACIÓN 50 - INSTALADOR THINLINC

El asistente instalará los paquetes que proporcionan todos los servicios de *ThinLinc* y solicitará confirmación para instalar dependencias y servicios. El programa nos pregunta la función del servidor: *master* o *agent*; elegimos *master* (instala las dos partes). Para funcionar con todos sus servicios pide abrir los puertos 22, 300, 1010, 9000 y 904. El proceso finaliza y el servicio ya está listo.

Si instalamos el cliente en cualquier equipo podremos conectarnos a nuestro servidor recién configurado (ilustración 51).

En la ilustración 52 se puede ver la sesión iniciada con algunas aplicaciones en marcha.

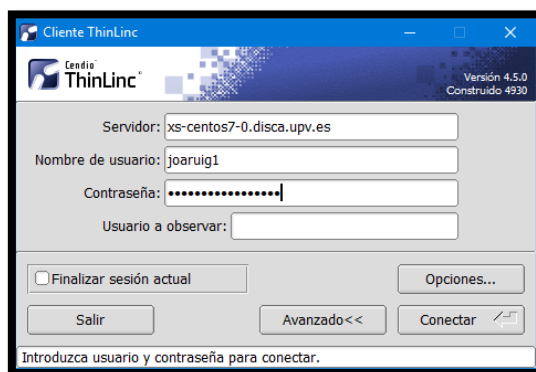


ILUSTRACIÓN 51 - CLIENTE THINLINC. DIALOGO INICIO SESIÓN.

Con esto termina la configuración del servicio de Linux y el capítulo de la Bolsa Servicios. Los dos servicios están configurados adecuadamente, y al igual que comentábamos al final del punto 6.1, será momento de convertir la máquina virtual en plantilla para replicarla y crear el servicio a través de su instanciación. En ese momento SERP ya será completamente funcional, y solo necesitará el apoyo del Agente para atender las peticiones de los usuarios.

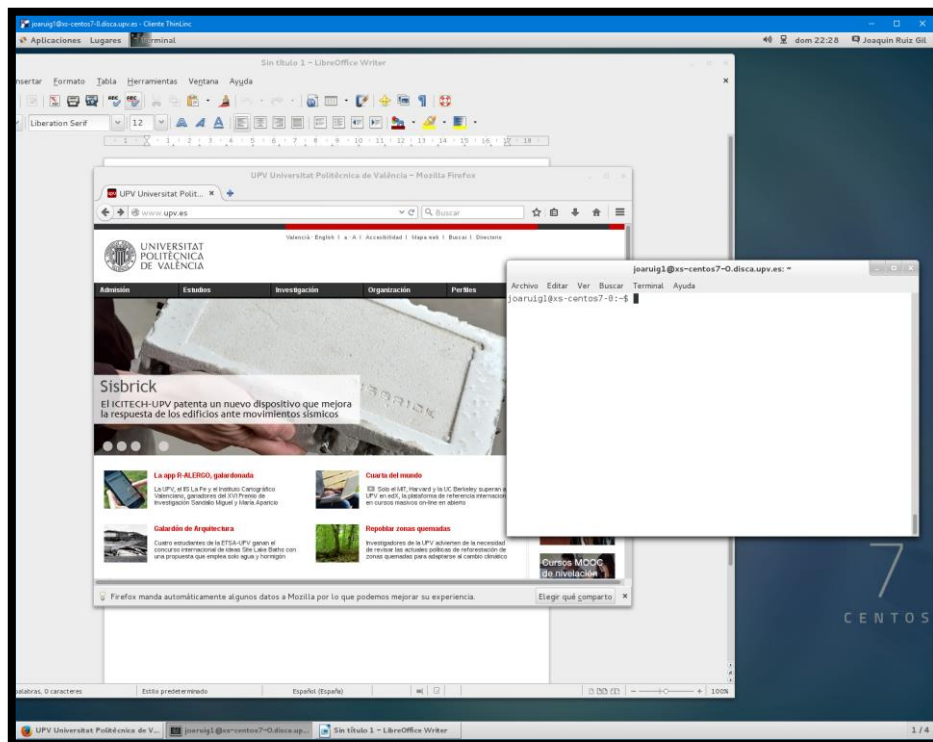


ILUSTRACIÓN 52 - SESIÓN INICIADA CON THINLINC





## 7. Agente

SERP ya está listo y a la espera de recibir peticiones que atender, y solo necesita que los clientes entren en el sistema. El Centro de Proceso puede atender a los clientes, y en su naturaleza existen mecanismos para el balanceo de carga, pero solo lo puede hacer a nivel de Instancias de servicio, pero no a nivel de sesión. Por otra parte, los usuarios necesitan un punto de entrada al sistema, independientemente de quien les vaya a atender deben acceder a alguna dirección única y conocida. Estas serán las funciones del Agente: ser el nombre conocido del sistema al que los usuarios se conecten, y proporcionar un balanceo de carga a nivel de sesión, seleccionando en base a un criterio la instancia de que debe atender la petición. Para todo esto utilizaremos otro servidor virtual, que alojaremos junto con el resto de nuestras máquinas virtuales, en el que instalaremos la aplicación *HAProxy*.

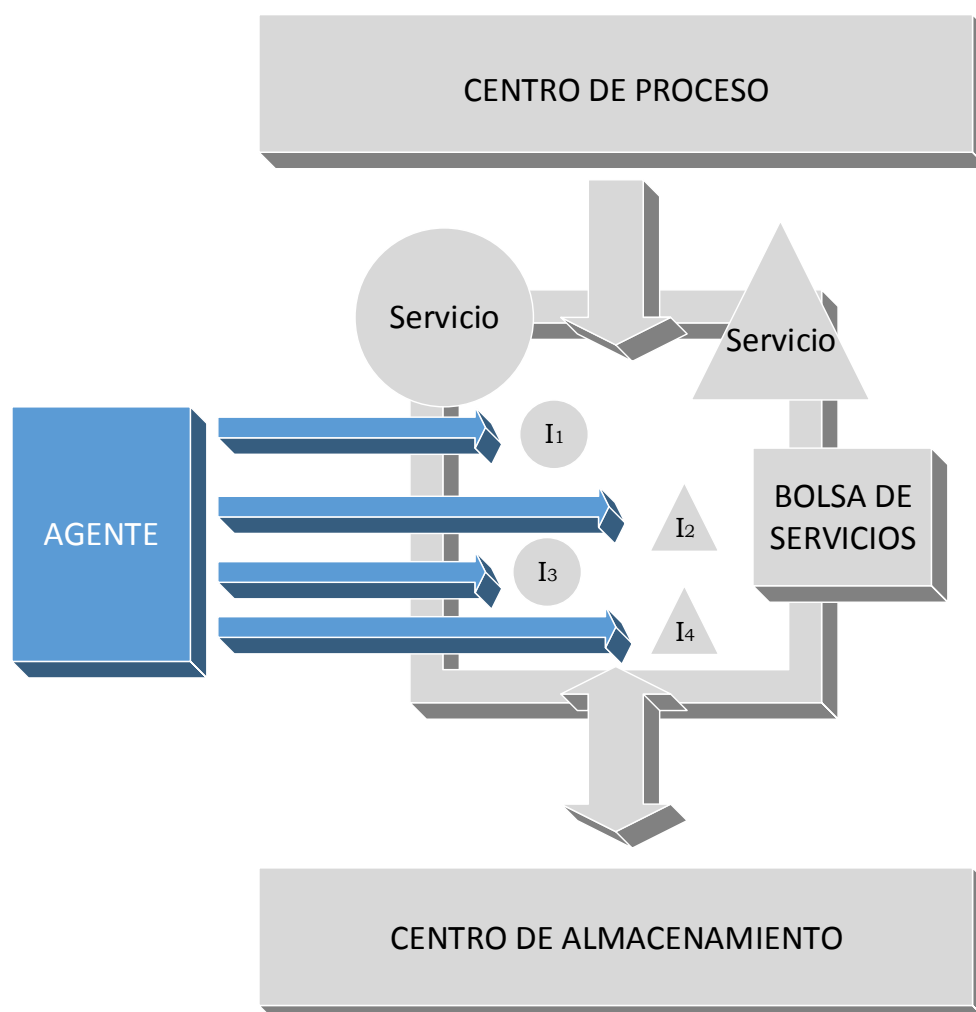


ILUSTRACIÓN 53 - SERP (AGENTE)

## 7.1. HAProxy

*HAProxy* es software libre de código abierto que ofrece un balanceo de carga y alta disponibilidad para aplicaciones basadas en *TCP* y *HTTP*. Su autor [16] enfatiza su velocidad y eficiencia, siendo capaz de realizar cientos de tareas por milisegundo y usando únicamente unos pocos KBytes por sesión.

Utilizaremos a *HAProxy* para realizar la tarea de agente: recogerá cada petición de acceso a SERP y la redirigirá hacia el servicio correspondiente, seleccionando en cada caso la instancia objetivo para realizar el balanceo de carga. Gracias a esto, los usuarios podrán usar un cliente *RDP* para conectar con las instancias de *Windows* o el cliente *ThinLinc* para conectar con *Linux*, pero en cualquier caso solo tendrán que hacer la petición a [SERP.disca.upv.es](http://SERP.disca.upv.es).

*HAProxy* está disponible para *Linux* en todas las grandes distribuciones, y para instalarlo solo tendremos que seleccionar el paquete. En este caso, y como se habrá podido apreciar a lo largo del trabajo, utilizaremos una máquina virtual para su uso, en este caso instalada con Ubuntu server 14.04.

Para instalarlo seleccionamos el paquete con el gestor de paquetes en línea de comando:

```
root@xs-haproxy:~# apt install haproxy
```

Para dejar activo el servicio tras su reinicio hay que modificar el fichero de configuración `/etc/default/haproxy`:

```
# Set ENABLED to 1 if you want the init script to start haproxy.
ENABLED=1
# Add extra flags here.
#EXTRA_OPTS="-de -m 16"
```

Modificaremos el fichero `/etc/haproxy/haproxy.conf` con la información de nuestros servicios y finalmente lo activaremos con la siguiente instrucción:

```
root@xs-haproxy:~# service haproxy start
```

La configuración que se ha realizado para *HAProxy* es la siguiente:

```

log      global
mode     http
option   dontlognull
retries  3

listen MS-RDP :3389

mode tcp
balance leastconn
option tcplog

server w2k12-1 158.42.181.151:3389 check maxconn 5
server w2k12-2 158.42.181.152:3389 check maxconn 5
server w2k12-3 158.42.181.153:3389 check maxconn 5

listen ThinLinc :22

mode tcp
balance source
option tcplog

server CentOS-1 158.42.181.154:22 check maxconn 5
server CentOS-2 158.42.181.155:22 check maxconn 5
server CentOS-3 158.42.181.156:22 check maxconn 5

listen stats :1936
mode http
log global
maxconn 10
timeout client 100s
timeout server 100s
timeout connect 100s
timeout queue 100s
stats enable
stats hide-version
stats refresh 3s
stats show-node
stats auth tfg:tfg2015
stats uri /stats

```

En el fichero de configuración podemos ver tres bloques diferenciados con la palabra *Listen*. Cada uno es una entrada de configuración asociada a un número de puerto. *HAProxy*, cuando reciba una petición desde los puertos 3389 o 22, aplicará reglas a nivel de la cabecera de *TCP* que darán como resultado el reenvío de esas peticiones hacia alguno de los servidores de su lista. En nuestro caso, por ejemplo, si se recibe una petición de escritorio remoto (*TCP3389*), el agente utilizará la regla de balanceo asociada (*balance leastconn*) para determinar cuál será el servidor de destino: *w2k12-1*, *w2k12-2* o *w2k12-3*. De la misma manera, si la petición que se recibe llega al puerto 22, el Agente utilizará la regla *balance source* para determinar a cuál de sus servidores: *CentOS-1*, *CentOS-2* o *CentOS-3*. La última de las entradas permite consultar vía web en el puerto 1936 de *HAProxy* las estadísticas y parte de la configuración, tal y como se puede ver en la ilustración 54. La aplicación web nos muestra agrupados los servicios que “escucha” junto con los servidores candidatos para recibir el tráfico. La información que ofrece es de gran utilidad ya que nos permite ver rápidamente si alguno de los servidores se



encuentra fuera de línea (como en este caso ocurre con *w2k12-3* y *CentOS-3*), o si no están atendiendo las peticiones en el orden que esperábamos.

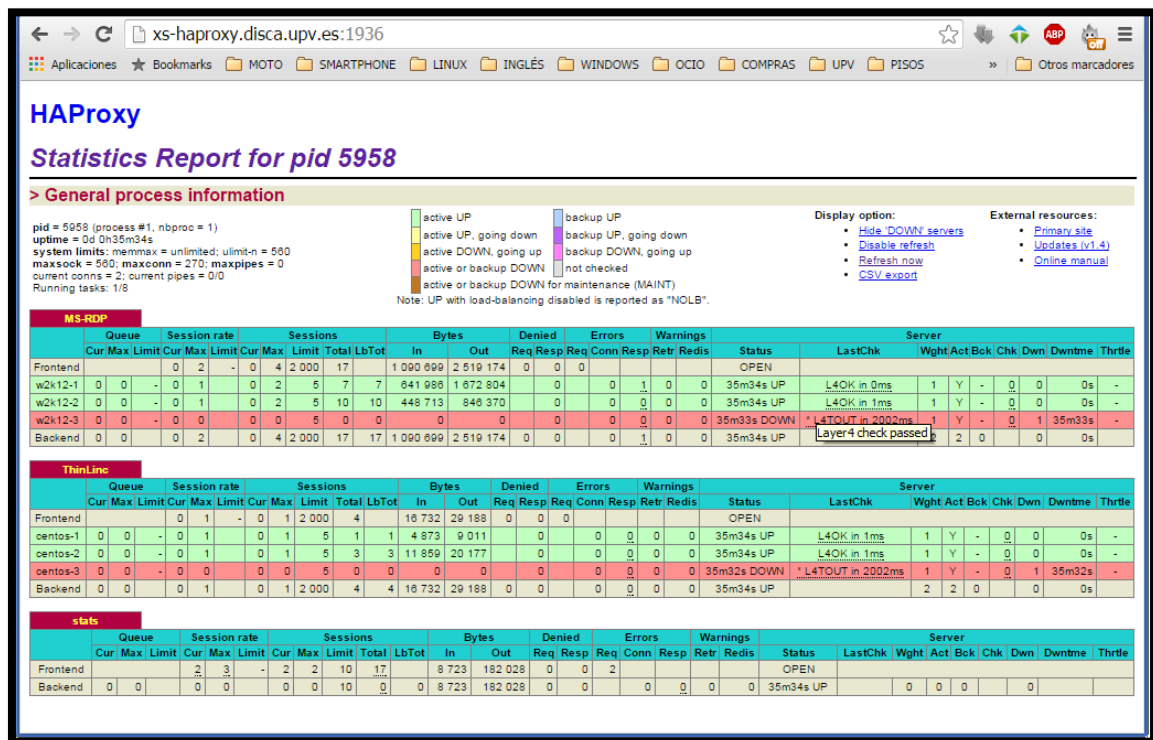


ILUSTRACIÓN 54 - ESTADÍSTICAS HAPROXY

Como hemos visto, los dos servicios no usan el mismo algoritmo de balanceo. Esto es debido al modo en que se establece uno y otro servicio. Con el servicio en *Windows* se puede utilizar el algoritmo más natural y las nuevas conexiones se establecen contra el servidor que este sirviendo menos sesiones. En cambio, con el servicio *Linux* no es posible: *ThinLinc* necesita establecer dos sesiones consecutivas y *HAProxy* lo interpreta como conexiones distintas y las balancea, impidiendo que finalice el protocolo de inicio de sesión. Para evitar esta segregación utilizaremos otro algoritmo que, aunque realiza bien su trabajo, no es el que mejor permitiría aprovechar los recursos. Esta opción determina el destino de una petición mediante tablas de dispersión, otorgando siempre el mismo servidor al mismo cliente (siempre que se conecte con el mismo equipo). Este mecanismo nos asegura que las dos conexiones iniciales de un cliente van a ser entregadas a un mismo servidor, pero como contrapartida no permitirá balancear tan adecuadamente las sesiones. En cualquier caso, el Agente actúa correctamente y SERP está completamente implementado.

## 8. Pruebas de rendimiento

---

SERP ya está completo y funcionando, y es el momento de evaluar en términos de carga balanceo y disponibilidad el servicio que va a ofrecer. Aunque el modelo es eficaz, la implementación hardware es muy modesta, y las prestaciones habrá que valorarlas desde el punto de vista de la eficiencia.

### 8.1. Carga de sesiones

Se han hecho pruebas del consumo de recursos por sesión de usuario:

En los equipos *Windows* una sesión supone una carga mínima aproximada de unos 30 Mbytes por cada sesión iniciada. La carga media de un usuario que abra un navegador, algún programa de la *suite Microsoft Office* y un lector de *pdf* sobrepasa los 300 Mbytes.

En *Linux*, el peso mínimo de una sesión ha resultado de 90 Mbytes. Una carga media con la utilización de programas similares a los usados en *Windows* alcanza los 400 Mbytes.

Desde *Cendio* [15] recomiendan reservar 150-300 Mhz por cada cliente que se conecte con un escritorio *KDE* o *Gnome*, y sobre *Windows* no hay artículo que cuantifique la *CPU* necesaria por sesión. Lo cierto es que el consumo de *CPU* oscilará más rápidamente que la *RAM*, siendo impredecible. Asimismo, la *CPU* que posee un servidor *XenServer* se reparte entre todos sus procesos, por lo que si dos máquinas comparten *host* se verán afectadas mutuamente.

El uso de la red, al igual que la *CPU*, tampoco se puede determinar fácilmente. El ancho de banda necesario para que el cliente tenga su pantalla actualizada en procesos ofimáticos se cifra entre 50 y 100 Kbits, pero las necesidades de cara a la red de almacenamiento dependerán de su interacción con el sistema y de los accesos que necesite realizar a su *VHD*.

Se han llegado a cargar cinco sesiones en cada sistema sin notar una ralentización mayor que la propia de una sesión. Si los picos de trabajo de cada sesión no son coincidentes, puede soportar cinco sesiones simultaneas sin problema. Se han empleado tanto en *Windows* como en *Linux* máquinas con 3'5GB de *RAM* (la cantidad máxima para dejar a *XenServer* con 1GB y que pudieran ser ubicadas dos por cada servidor). Asimismo, hay que tener en cuenta la cantidad de memoria necesaria para el sistema operativo de la máquina virtual, que para los cálculos se ha estimado como 1 GB *RAM* a los sistemas *Linux* y 1'2 GB *RAM* a los sistemas *Windows*. En la ilustración 55 se puede observar el



consumo de una sesión para cada sistema. Se ha determinado el número máximo de sesiones como la división entre la RAM disponible (memoria asignada a la máquina virtual menos la memoria del sistema operativo) y el consumo medio de RAM por sesión.

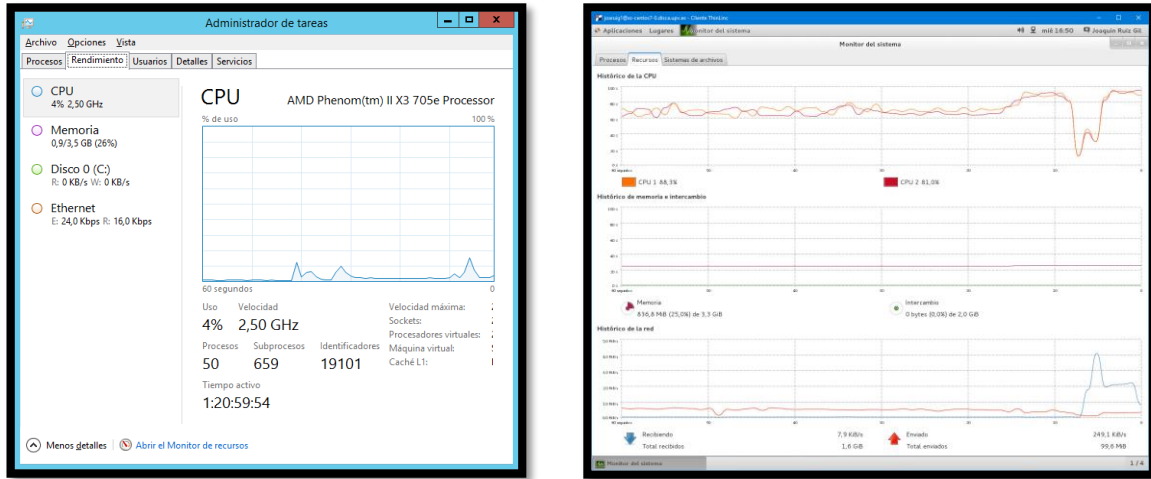


ILUSTRACIÓN 55 - CONSUMO DE SESIONES (WINDOWS/LINUX)

|   | <i>Windows</i>                    | <i>Linux</i>                      |
|---|-----------------------------------|-----------------------------------|
| <i>Consumo sistema</i>                          | <i>1'2 GB</i>                     | <i>1 GB</i>                       |
| <i>Consumo mínimo (por sesión)</i>              | <i>30 MB</i>                      | <i>90 MB</i>                      |
| <i>Consumo medio (por sesión)</i>               | <i>300 MB</i>                     | <i>400MB</i>                      |
| <i>Red</i><br><i>(solo información gráfica)</i> | <i>50-100 Kbits/sec</i>           | <i>50-100 Kbits/sec</i>           |
| <i>USO CPU</i>                                  | <i>Determinado por el usuario</i> | <i>Determinado por el usuario</i> |
| <i>Nº Máx sesiones</i>                          | <i>7</i>                          | <i>6</i>                          |

TABLA 10 - CÁLCULO DE SESIONES

## 8.2. Prueba de balanceo de carga

El test ha sido realizado con la configuración mostrada en la sección 7.1, con algoritmo *leastconn* para instancias *Windows* y algoritmo *source* para instancias *Linux*. Las peticiones se distribuyen entre dos instancias disponibles (dos *Linux* y dos *Windows*).

No hay mucho que poder resaltar de la prueba, ya que el resultado de usar *HAProxy* es transparente salvo en el caso de que no se cumplan condiciones<sup>25</sup> para poder enviar a ningún servidor (en caso *Windows*), o el seleccionado (en el caso de *Linux*). En ambos, *HAProxy* retiene la conexión hasta que finalmente queda disponible un recurso (caso de *Windows*) o el recurso buscado (*Linux*). Gracias a las estadísticas descritas en la ilustración 54, podemos comprobar sin duda dónde van las peticiones de los clientes. El servidor *Agente* cumple completamente y balancea la carga con la configuración de la política sin ninguna pérdida de tiempo.

## 8.3. Prueba de la alta disponibilidad

La prueba de alta disponibilidad se realiza con la característica *HA* habilitada:

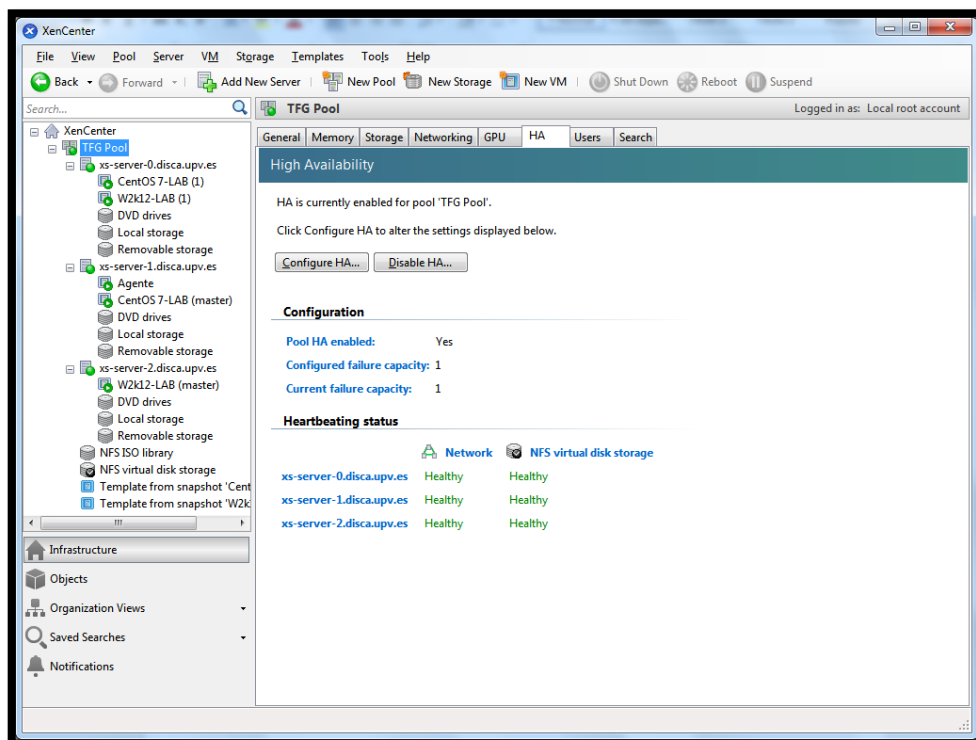


ILUSTRACIÓN 56 - ESCENARIO INICIAL HA (1/2)

<sup>25</sup> Se ha limitado a un máximo de 5 sesiones por instancia.

Como se puede ver en la ilustración 56, los tres servidores están activos y hospedan en total a cinco máquinas virtuales. En ilustración 57 aparece especificado el plan de recuperación.

La máquina virtual *Agente* es la única de las cinco que tiene garantizado el reinicio, y las máquinas virtuales *master* intentarán reactivarse de nuevo si es posible (el nombre de master no tiene nada que ver ninguna función del *Pool*, es un nombre distinto para indicar que son las primeras que intentarán rearrancar si hay oportunidad).

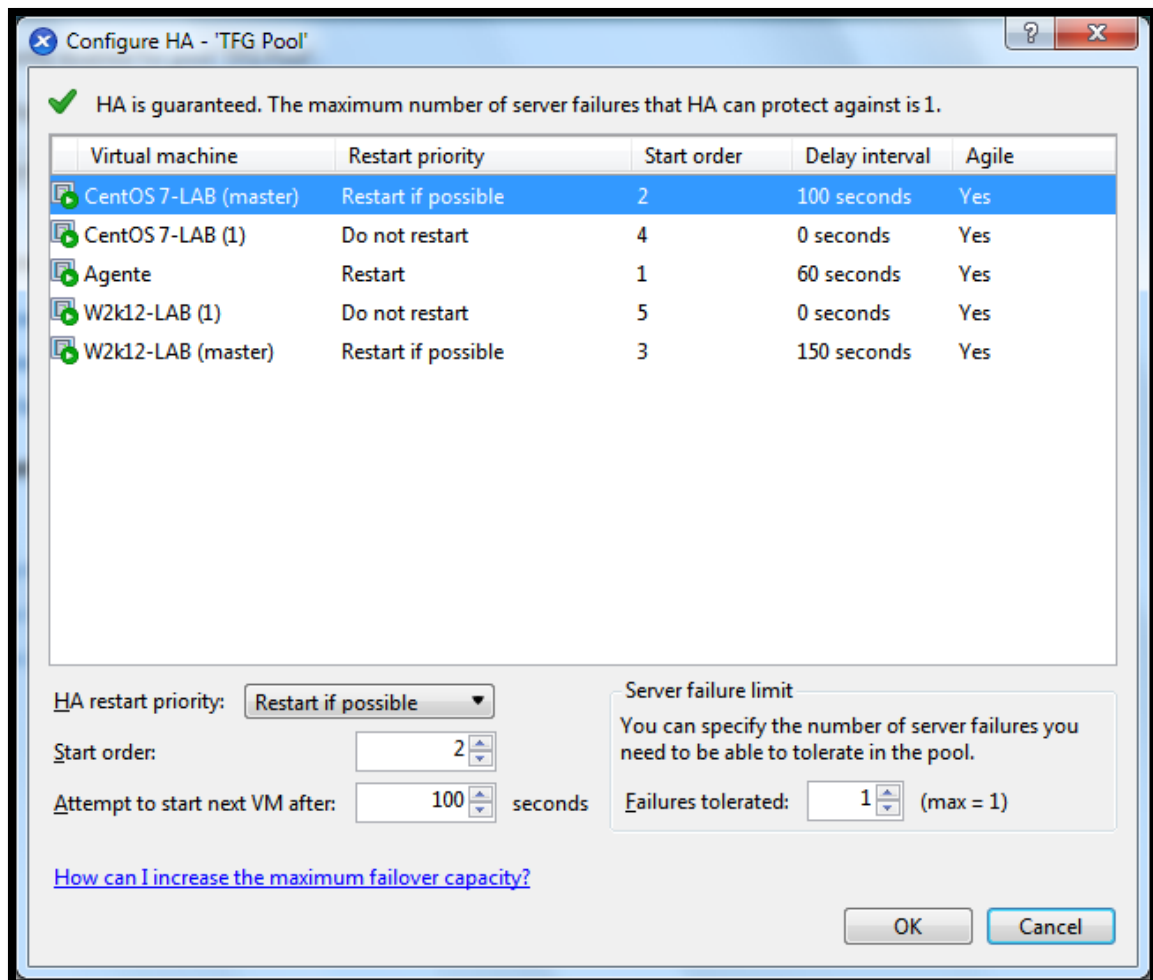
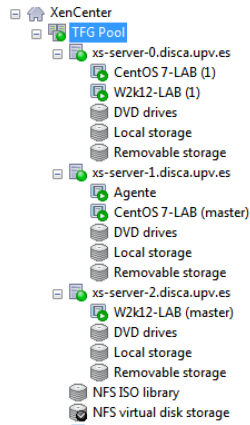


ILUSTRACIÓN 57 - ESCENARIO INICIAL HA (2/2)

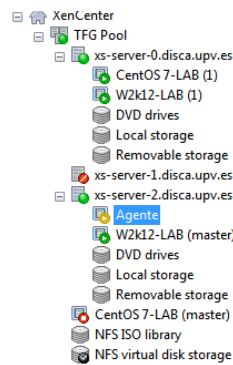
La prueba consistirá en desconectar, en el servidor *xs-server-1*, los dos cables físicos de red. Como vemos en la ilustración 55, tiene alojada la máquina virtual *Agente* y otra máquina que, en condiciones favorables, puede que se reinicie (*CentOS 7-LAB (master)* tiene supeditado su reinicio a los recursos disponibles en el momento de su recuperación: *Restart if possible*).





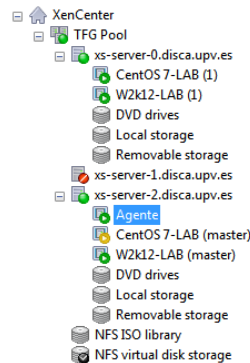
00:00 -> Se desconectan las dos interfaces de red. Desde la consola de *XenCenter* todo parece seguir igual.

01:00 -> El servidor *xs-server-1* comienza a reiniciarse



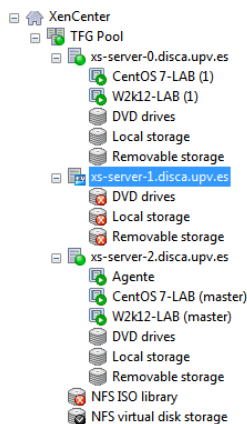
02:00 -> El servidor aparece como desconectado

02:30 -> Se inicia el proceso de recuperación de la máquina virtual *Agente*



03:00 -> *Agente* está activo (arrancando la máquina virtual, en breve podrá dar servicio)

03:00 -> Se inicia el proceso de recuperación de la máquina virtual *master*



04:45 -> *CentOS-LAB (master)* está activa.

TABLA 11 – EVENTOS EN LA PRUEBA HA



Como se puede observar en la tabla 11, se ha registrado el tiempo y los eventos. En el momento inicial se han desconectado los cables de red dejando completamente aislado a *xs-server-1*, y por tanto, desconectado de las máquinas virtuales que servía. Al contener a la máquina virtual *Agente*, todo SERP se ha quedado sin servicio. Aproximadamente un minuto después y de forma automática, en *xs-server-1* comienza el proceso de reinicio.

A los dos minutos, la información que nos ofrece *XenCenter* marca a *xs-server-1* como desconectado, y 30 segundos después comienza en *xs-server-2* (el servidor elegido dinámicamente por el *HA Plan*) la recuperación de la máquina virtual *Agente* (la máquina que tiene garantizado el re arranque).

A los tres minutos *Agente* ha sido puesto en línea por *xs-server-2* (comienza el arranque de su distribución *Linux* y cuando acabe lanzará el servicio de *HAProxy*, por lo que en breves momentos podrá atender peticiones). En este momento *XenServer* evalúa si es posible recuperar la otra máquina virtual y determina que *xs-server-2* puede hospedarla, por lo que comienza su recuperación (comienza la recuperación de *CentOS7-LAB(master)*).

A los cuatro minutos y cuarenta y cinco segundos acaba la recuperación. *Xs-server-0* está en modo de mantenimiento a la espera de que lo revisemos.

El resultado final es satisfactorio: se han recuperado completamente los servicios que ofrecía *xs-server-1*, y por tanto, todo el servicio que ofrecía el sistema está disponible.

## 9. Conclusiones

---

En los últimos años la virtualización se ha implantado de forma definitiva en los sistemas informáticos, ofreciendo muchas soluciones en distintos escenarios, todo gracias a poseer y facilitar características óptimas de servicio. La facilidad de cambio y el aprovechamiento de recursos son fundamentales para crear sistemas eficientes.

Las grandes empresas, apuestan por *PaaS (Platform as a Service)* y ofrecen productos como *AWS (Amazon Web Services)*, *Microsoft Azure*, o *Google CloudPlatform*. Otros grandes competidores como *Oracle* ven en la *PaaS* el futuro de negocio para los próximos 25 años o incluso más, y finalmente, empresas como *VMWare* líderes del sector de la virtualización en la nube privada, encuentran fundamental entrar con productos de integración de nubes pública y privada. Todo esto da a entender la enorme magnitud que tiene y tendrá la virtualización, que incluso podría tener como parangón el hito de la creación de internet.

En este trabajo se presenta SERP, un sistema basado en la implantación de tecnologías de virtualización, que ha sido diseñado e implementado en una plataforma con numerosas herramientas software libre, y conjugándolas ha construido un servicio de escritorio remoto.

Este proyecto ha cumplido con todos sus objetivos propuestos. Primeramente, diseñando un sistema con la tecnología *XenServer*, que ha sido el Centro de Proceso y junto con el Centro de Almacenamiento han dado el soporte físico del servicio. En segundo lugar, con la Bolsa de Software, en la que se ha diseñado y configurado el servicio que ofrece, en este caso el escritorio remoto. Para concluir y mejorar todo el ecosistema se ha integrado con un Agente, que además de facilitar la entrada a los usuarios con un único nombre, ha cumplido con el objetivo de mantener balanceado el sistema, repartiendo la carga entre los recursos disponibles. Finalmente, se ha testado y evaluado, demostrando que es un sistema de alta escalabilidad, alta disponibilidad y balanceado, que posee una rápida recuperación frente a fallos hardware o software y, además, el mantenimiento que se realiza resulta sencillo y eficaz.



## 9.1. Propuestas de mejora

Aun habiendo cumplido con todos los objetivos, el sistema se puede mejorar en los siguientes aspectos:

En el Centro de Proceso las mejoras vendrían acordes con el presupuesto, puesto que la principal mejora sería hardware. Aun así, en el entorno de administración se podrían diseñar guiones que regeneraran las máquinas virtuales periódicamente a partir de plantillas.

El Centro de Almacenamiento se encuentra en la misma tesitura que el Centro de Proceso. De cualquier forma, si se pudiera replicar se podría hacer que trabajara de forma coordinada usando otro sistema de ficheros como *glusterfs* o *ZFS*, lo que mejoraría tanto el rendimiento como la fiabilidad. El sistema de RAID o también debería cambiarse, a ser posible a *RAID 5* o *RAID 6*, o en su caso a *RAID 1*.

El Agente no es capaz de hacer un reparto perfecto de carga con las instancias *Linux*. *HAProxy* permite utilizar un *checking* externo que le permite al cliente modificar el algoritmo. Una solución sería utilizarlo para desactivar el servidor que ha alcanzado su límite y ya no va a aceptar más sesiones. Esto modificaría la función de hash, y *HAProxy* le asignaría otra instancia como destino. En cuanto a los *Windows*, que sí que hacen un reparto ideal de la carga, no son capaces de retomar una sesión iniciada si el cliente se desconecta del servidor. Esto puede solucionarse utilizando otro algoritmo, *rdp-cookie*, que construye un registro con las sesiones que se han dejado abiertas y es capaz de asociar el nombre de usuario con la instancia en la que está la sesión iniciada.

Por último, la Bolsa de Servicios podría tanto ampliar las plataformas ofertadas como refinar la configuración que ofrece. El sistema de directorio *NIS* podría eliminarse en *Linux* en favor de una implementación de *LDAP* o una integración más directa con *AD* por medio de otra aplicación como *Centrify* [13]. Otra perspectiva sería ofrecer directamente aplicaciones virtualizadas en lugar de plataformas enteras.





# Bibliografía

---

- [1] Fundeu, «Fundeu - Recomendaciones,» 7 10 2010. [En línea]. Available: <http://www.fundeu.es/recomendacion/virtualizar-y-virtualizacion-son-terminos-correctos-en-espanol-708/>. [Último acceso: 10 11 2015].
- [2] M. Portnoy, «Essentials: Virtualization essentials,» de *Essentials: Virtualization essentials*, Hoboken, NJ, USA, Sybex, 2012, p. 2.
- [3] B. Sosinsky, «Cloud Computing Bible,» de *Cloud Computing Bible*, Indianapolis, Indiana, USA, Wiley Publishing, Inc., 2011, p. 100.
- [4] W. Keegan, «Embedded Innovator Magazine,» 27 7 2012. [En línea]. Available: <http://embeddedinnovator.com/2012/06/the-rise-of-the-type-zero-hypervisor/>. [Último acceso: 20 11 2015].
- [5] Diandro SL, «norbertogallego.com - Análisis de mercados y empresas de tecnología,» Diandro SL, 25 3 2015. [En línea]. Available: <http://www.norbertogallego.com/vmware-suiza-del-cloud-computing/2015/03/25/>. [Último acceso: 20 11 2015].
- [6] Citrix Systems, Inc., «Citrix product documentation,» 1999-2015. [En línea]. Available: [http://docs.citrix.com/content/dam/docs/en-us/xenserver/xenserver-65/XenServer-6.5.0-Configuration\\_Limits.pdf](http://docs.citrix.com/content/dam/docs/en-us/xenserver/xenserver-65/XenServer-6.5.0-Configuration_Limits.pdf). [Último acceso: 20 11 2015].
- [7] WhatMatrix.com, « WhatMatrix.com,» WhatMatrix.com, 2015. [En línea]. Available: <https://www.whatmatrix.com/comparison/Virtualization>. [Último acceso: 20 11 2015].
- [8] Red Hat, Inc., «redhat,» 2015. [En línea]. Available: <https://www.redhat.com/es/files/resources/en-rh-kvm-kernal-based-virtual-machine.pdf>. [Último acceso: 20 Noviembre 2015].



- [9] Open Media Vault, «openmediavault,» 2015. [En línea]. Available: <http://www.openmediavault.org/>. [Último acceso: 15 11 2015].
- [10] Citrix Systems, Inc., «XenServer. Open Source Virtualization,» Citrix Systems, Inc., 1999-2015. [En línea]. Available: <http://xenserver.org/overview-xenserver-open-source-virtualization/download.html>. [Último acceso: 20 11 2015].
- [11] Citrix Systems, Inc., «XenServer. Open Source Virtualization,» Citrix Systems, Inc., 1999-2015. [En línea]. Available: <http://xenserver.org/overview-xenserver-open-source-virtualization/documentation.html>. [Último acceso: 20 11 2015].
- [12] Citrix Systems, Inc, «Support Citrix,» 11 3 2014. [En línea]. Available: <https://support.citrix.com/servlet/KbServlet/download/21018-102-664364/High%20Availability%20for%20Citrix%20XenServer.pdf>. [Último acceso: 12 11 2015].
- [13] MIT/Kerberos Consortium, «MIT. Massachusetts Institute of Technology,» 1985-2015. [En línea]. Available: [http://web.mit.edu/kerberos/krb5-1.12/doc/admin/conf\\_files/krb5\\_conf.html](http://web.mit.edu/kerberos/krb5-1.12/doc/admin/conf_files/krb5_conf.html). [Último acceso: 25 11 2015].
- [14] I. González Sosa y M. Padrón Martínez, «AMPLIACIÓN DE SISTEMAS OPERATIVOS LINUX + Administración y Seguridad,» 2004. [En línea]. Available: [http://sopa.dis.ulpgc.es/ii-aso/portal\\_aso/lelinux/seguridad/pam/pam\\_doc.pdf](http://sopa.dis.ulpgc.es/ii-aso/portal_aso/lelinux/seguridad/pam/pam_doc.pdf). [Último acceso: 10 11 2015].
- [15] Cendio, «Cendio ThinLinc,» Cendio, 2015. [En línea]. Available: <https://www.cendio.com/resources/docs/adminguide.pdf>. [Último acceso: 29 11 2015].
- [16] Z. H. Shah, «Windows Server 2012 Hyper-V : Deploying the Hyper-V Enterprise Server Virtualization Platform,» de *Windows Server 2012 Hyper-V : Deploying the Hyper-V Enterprise Server Virtualization Platform*, Olton, Birmingham, GBR, Packt Publishing Ltd, 2013, pp. 27-28.



- [17] VMware, Inc, «<https://www.vmware.com/>,» 2015. [En línea]. Available: <https://www.vmware.com/es/products/vsphere/>. [Último acceso: 20 11 2015].
- [18] Server World, «<http://www.server-world.info/en/>,» Server World, 2007-2015. [En línea]. Available: <http://www.server-world.info/en/>. [Último acceso: 20 11 2015].
- [19] W. Tarreau, «HAProxy. The Reliable, High Performance TCP/HTTP Load Balancer,» 2015. [En línea]. Available: <http://www.haproxy.org/>. [Último acceso: 25 11 2015].







# ANEXO 1: Caso concreto: Hyper-V

---

Hyper-V es la plataforma de virtualización presentada e incluida por Microsoft en Windows server 2008 y versiones sucesivas. Hyper-V implementa el aislamiento de las máquinas virtuales en términos de partición. Una partición es una unidad lógica de aislamiento, asistida por el hipervisor, en la que cada MV se ejecuta. Una instancia de hipervisor tiene que tener por lo menos una partición padre, que ejecuta una versión compatible de Windows Server. La pila de virtualización se ejecuta en la partición principal y tiene acceso directo a los dispositivos de hardware. Después, la partición padre puede crear las particiones hijas que albergan el sistema operativo invitado. Una partición hija no tiene acceso al procesador físico, ni al manejo de las interrupciones reales, sino que tiene una visión virtual de su entorno en cuanto a la CPU y la memoria del sistema (dependiendo de los recursos que se le hayan asignado a la MV). El hipervisor gestiona las interrupciones al procesador, y las redirecciona a la partición correspondiente utilizando el controlador SynIC (Synthetic Interrupt Controller). Este proceso de traducción se puede acelerar con el soporte de instrucciones adecuado por parte del procesador.

En lo que respecta al resto de recursos de hardware, las particiones hijas tampoco tienen acceso, pero tienen una vista virtual del mismo como dispositivos virtuales. Las solicitudes a los dispositivos virtuales se redirigen a través del VMBus (Virtual Machine Bus) a los dispositivos en la partición padre, que gestiona las peticiones. El VMBus es un canal lógico que permite comunicar las particiones y a través de él se envían peticiones y respuestas. Las particiones pueden anidarse y por tanto puede darse el caso de que la partición padre que recibe una petición de su partición hija tenga que redirigir la petición a un nivel superior hasta que sea tratada (ninguna partición que tenga padre tiene acceso a los dispositivos físicos). La conexión de VMBus se efectúa entre dos servicios, siendo para los padres Particiones padres VSP (Virtualization Server Provider) y para los hijos VSC (Virtualization Server Client).

Hyper-V es un hipervisor de tipo 1, algo que puede parecer contradictorio cuando tenemos instalado el sistema operativo Windows Server y añadimos la funcionalidad de hipervisor al instalar el rol de Hyper-V, como si fuera una aplicación que ejecutara el sistema operativo. La explicación es que, cuando instalamos el rol y reiniciamos el equipo para completar la instalación se crea una partición padre y el propio hipervisor se introduce entre la capa del sistema y Windows server, que ahora es una partición que trabaja encima del hipervisor y es su partición hija [16].





## ANEXO 2: Características vSphere

---

Como se ha comentado con anterioridad, la plataforma de VMWare es la más extendida y la que ofrece funcionalidades más amplias. En el punto anterior se han mencionado las características que tendrían un uso común en una infraestructura pequeña o mediana, pasando por alto otras más innovadoras. Es importante remarcar esa diferencia, tanto en la gestión como en la capacidad de la plataforma, ya que, si no fuera por su coste, hubiese sido la primera opción para implementar la solución propuesta.

Las características que ofrece vSphere en su versión 6.0 son las siguientes [17]:

- VMware vSphere Virtual Symmetric Multiprocessing: hasta 128 CPU virtuales.
- VMware vSphere Thin Provisioning: asignación dinámica de la capacidad de almacenamiento compartido. Permite implementar una estrategia de almacenamiento en niveles y reducir al mismo tiempo el gasto en almacenamiento hasta un 50 %.
- VMware vSphere vMotion: migración dinámica de máquinas virtuales entre servidores y entre conmutadores virtuales sin interrupción alguna para los usuarios ni pérdidas de servicio.
- VMware vSphere Storage vMotion: migración dinámica de discos de máquinas virtuales sin interrupción alguna para los usuarios.
- VMware vSphere High Availability (HA): en caso de un fallo de hardware o del sistema operativos en cuestión de minutos automatizado de todas las aplicaciones.
- VMware vSphere Fault Tolerance (FT) proporciona disponibilidad continua de todas las aplicaciones en caso de fallo de hardware, sin pérdida de datos ni tiempo de inactividad. Para cargas de trabajo de hasta 2 vCPU.
- VMware vSphere Data Protection es la solución de copia de seguridad y replicación de VMware basada en la tecnología de EMC Avamar. Ofrece copias de seguridad con un almacenamiento eficiente gracias a la funcionalidad patentada de eliminación de duplicados de longitud variable, recuperación rápida y replicación optimizada para WAN de recuperación ante desastres. Ofrece copia de seguridad en disco de máquinas virtuales en el nivel de imagen sin agentes y protección con reconocimiento de aplicaciones para aplicaciones empresariales esenciales junto con replicación de copias de seguridad cifradas con eficiencia de red de área extendida entre sitios.



- VMware vShield Endpoint protege las máquinas virtuales con soluciones descargadas de antivirus y antimalware sin necesidad de agentes dentro de la máquina virtual.
- VMware vSphere Virtual Volumes permite la abstracción para dispositivos de almacenamiento externo (SAN y NAS), de modo que reconozcan la máquina virtual.
- VMware vSphere Storage Policy-Based Management permite la gestión común a través de niveles de almacenamiento y automatización de clase de servicio de almacenamiento dinámico mediante un plano de control basado en políticas.
- VMware vSphere Content Library proporciona una gestión centralizada sencilla y efectiva para plantillas de máquina virtual, dispositivos virtuales, imágenes ISO y scripts. Componentes adicionales disponibles en la edición Enterprise
- VMware vSphere Distributed Resource Scheduler proporciona equilibrio de carga dinámico independiente del hardware y asignación de recursos para máquinas virtuales en clúster. Utiliza la automatización basada en políticas para reducir la complejidad de gestión y reforzar el cumplimiento de los acuerdos de nivel de servicio (SLA).
- VMware vSphere Distributed Power Management automatiza el consumo eficiente de la energía en los clústeres de vSphere Distributed Resource Scheduler optimizando continuamente el consumo de energía de los servidores dentro de cada clúster.
- VMware vSphere Big Data Extensions ejecuta Hadoop en vSphere para lograr una mayor utilización, fiabilidad y agilidad. vSphere Big Data Extensions admite diferentes distribuciones de Hadoop y facilita a TI la implementación, ejecución y gestión de cargas de trabajo Hadoop en una plataforma común. Componentes adicionales disponibles en la edición Enterprise Plus (además de los componentes de la edición Enterprise enumerados anteriormente)
- VMware vSphere Distributed Switch simplifica y mejora la red de máquinas virtuales en entornos de vSphere. Además, permite usar en ellos conmutadores virtuales distribuidos de terceros.
- VMware vSphere Storage I/O Control y VMware vSphere Network I/O Control establecen nuevas prioridades de almacenamiento y calidad de servicio a fin de garantizar el acceso a los recursos de red.
- VMware vSphere Auto Deploy efectúa una implementación rápida de los *hosts* de vSphere adicionales a medida que se necesitan. Cuando vSphere Auto Deploy está en



ejecución, aplica imágenes de actualización, con lo que elimina la aplicación de parches y la consiguiente necesidad de programar el tiempo correspondiente.

- VMware vSphere *Host Profiles* ayuda a los administradores de TI a simplificar la implementación de servidores y el cumplimiento normativo.

- VMware vSphere Storage DRS automatiza el equilibrio de carga mediante el uso de funciones de almacenamiento para determinar la mejor ubicación para que residan los datos de una máquina virtual concreta, tanto cuando se crea, como cuando se utiliza a lo largo del tiempo.

- VMware vSphere Flash Read Cache virtualiza la memoria flash del servidor mediante una capa de caché de lectura de alto rendimiento que reduce drásticamente la latencia de las aplicaciones.

- VMware vSphere Fault Tolerance proporciona disponibilidad continua de todas las aplicaciones en caso de fallo de hardware, sin pérdida de datos ni tiempo de inactividad. Para cargas de trabajo de hasta 4 vCPU.

- VMware vSphere vMotion permite la migración dinámica de máquinas virtuales entre servidores, entre vCenter Servers y a grandes distancias (con tiempo de ida y vuelta de hasta 100 milisegundos) sin interrupción alguna para los usuarios ni pérdidas de servicio. De esta forma, se elimina la necesidad de programar tiempo de inactividad de las aplicaciones para el mantenimiento de servidores.

- El software NVIDIA GRID vGPU ofrece todas las ventajas de los gráficos acelerados por hardware de NVIDIA a las soluciones virtualizadas.



---

