

UNIVERSIDAD CATÓLICA DE LA SANTÍSIMA CONCEPCIÓN  
Facultad de Ingeniería  
Ingeniería Civil Informática



# Servidor de Máquinas Virtuales Didácticas

**Carlos Alberto Albistur Urrutia**

Informe de Proyecto de Título para optar al Título de  
Ingeniero Civil Informático

**Profesor Guía**

Pedro Sebastián Gómez Meneses

Concepción, Noviembre 2016

# Resumen

En el mundo las empresas utilizan múltiples servidores para llevar a cabo sus operaciones, que en general no se manejan con un computador, sino con dos o más por motivos de confiabilidad. Esto se debe a que no se puede garantizar que un sistema operativo funcione correctamente todos los días durante el año. Por esta razón cada servicio se separa y se deja funcionando en un computador distinto, pero esta solución resulta costosa y difícil de administrar por el número de máquinas y el espacio requerido para su funcionamiento (Tanenbaum and García, 2009).

En las universidades como también en las empresas las Tecnologías de Información Comunicacional (TIC) juegan un papel fundamental para su funcionamiento, sin embargo existe una problemática en la adquisición de nuevas tecnologías y en su renovación por su alto costo. Por lo anterior las universidades se ven con dificultades en adquirir nuevos productos informáticos en especial el software docente para que los alumnos afines a la tecnología desarrollen aplicaciones y tengan disponible entornos de prueba para testearlas (Medina and Pérez, 2015). Para esto se ha propuesto el uso de la virtualización, lo que permite que un computador ejecute múltiples máquinas virtuales. Esto resulta beneficioso, ya que si falla una máquina virtual no ocasiona que las otras máquinas dejen de funcionar. Gracias a la virtualización se pueden ejecutar varios servidores en distintas máquinas virtuales, lo cual permite ejecutar servicios a un costo menor mejorando también su administración (Tanenbaum and García, 2009).

Se plantea como solución implementar un servidor de máquinas virtuales utilizando software libre con KVM (Kernel-based Virtual Machine) como servidor en Linux usando QEMU (Quick Emulator) como herramienta de emulación de las máquinas, a las cuales se podrá tener acceso instalando un administrador de conexión a escritorios vir-

---

tuales remotos en las máquinas clientes (Linux-KVM, 2016). Así la universidad podrá instalar sistemas operativos y las aplicaciones que desee sin tener que incurrir en un gasto físico del entorno. También se podrán crear por ejemplo, laboratorios virtuales para que los estudiantes puedan poner en práctica los conocimientos adquiridos en la administración de redes y en la configuración e instalación de servidores en diversos sistemas operativos (Medina and Pérez, 2015).

Además se validará el servidor mediante múltiples pruebas, las cuales miden recursos computacionales. Los recursos en los cuales se enfocan las pruebas son: el rendimiento de la CPU, el rendimiento del uso de la red, el rendimiento de la memoria RAM, el rendimiento de las unidades de almacenamiento. Con estas pruebas se buscará la mejor configuración para distintos escenarios de uso de la máquina anfitriona y de las máquinas virtuales (Langer, French T, 2011).

# Abstract

In the world, the companies use numerous servers to carry out their operations that in general ways are not handled by a simple computer, but two or more for reasons of reliability. This is due to the fact that it cannot be guaranteed that one operating system works properly every single day of the year. For this reason, every service must be separated and it left running in a different computer, however, this kind of solution is very costly and difficult to administrate because of machine quantities and required space for its correct functioning.

In universities and also in the companies, the information and communication technologies (ICT) play a very important role in its functioning, nevertheless there is one problem remain about the acquisition in new technologies and the renovation due to its high cost. For the mentioned reason, it is very difficult for the universities to acquire new data products, especially the teaching software in which the students interrelated with technologies create and develop some applications and of course that students have the opportunity and the availability to test these techs. For this, it has been proposed the use of virtualization, allowing that a computer execute several virtual machines. This results advantageous because if a virtual machine fails, the functioning of the others it will continue. Due to virtualization, we can execute several servers in different virtual machines reducing their costs and improving the administration too.

As a solution, the following proposal has been considered: To implement a virtual machine server using free software with KVM (Kernel-based Virtual Machine) as a server in Linux, using QEMU (Quick Emulator) as an emulation tool of the machines. You can access to these machines through the installation of a connection administrator to remote virtual desktops in the client machines. In this way, the university would be able

---

to install operating systems and applications without any physical use of the computing environment. In addition, can be created virtual labs, in which the students can implement all the knowledge gained in the network management, setting and installation of servers in different operating systems.

Finally, the server will be validated by several test with focus in computing resources. This resources are: CPU performance, network performance, Ram memory performance and storage unit performance. This test will seek the best efficient setting for different scenarios of use in the host machine and the virtual machines.

# Agradecimientos

Quiero agradecer en primer lugar a mi profesor guía, Pedro Gómez por todo el apoyo brindado en el desarrollo de mi proyecto de título, tanto en el ámbito profesional como humano llegando a ser un verdadero amigo.

Agradecer a mis amigos por su apoyo, ayuda y compañía en el desarrollo de mi tesis.

Agradecer también a mi familia, en especial a mis hermanos que siempre me ayudaron en mi vida de universitario.

Por último agradecer y dedicar este logro a mi madre, que siempre estuvo preocupada de mí en el desarrollo de mi vida de estudiante, donde quiera que estés gracias.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1.1. Descripción del Problema . . . . .	1
1.2. Justificación del Problema . . . . .	1
1.3. Delimitación del Problema . . . . .	2
1.4. Objetivo General . . . . .	2
1.5. Objetivos Específicos . . . . .	2
1.6. Metodología . . . . .	3
1.7. Esquema de documento de Tesis . . . . .	4
<b>2. Marco teórico</b>	<b>5</b>
2.1. Virtualización . . . . .	5
2.2. Hipervisor . . . . .	5
2.2.1. Hipervisores tipo 1 . . . . .	6
2.2.2. Hipervisores tipo 2 . . . . .	7
2.3. Paravirtualización . . . . .	8
2.4. KVM . . . . .	8
2.5. QEMU . . . . .	9
2.6. Administradores de conexión a escritorios virtuales remotos . . . . .	9
<b>3. Estado del arte</b>	<b>11</b>
3.1. VirtualBox . . . . .	11
3.2. VMware . . . . .	12
3.3. Citrix XenServer . . . . .	14
3.4. Hyper-V . . . . .	15
3.5. Evaluación y comparación de los Hipervisores . . . . .	16

<b>4. Desarrollo</b>	<b>18</b>
4.1. Evaluación el entorno actual . . . . .	20
4.2. Planificación el entorno virtual . . . . .	22
4.2.1. Diagrama de distribución del entorno virtual . . . . .	25
4.2.2. Diagrama de casos de uso del entorno virtual . . . . .	27
4.2.3. Diagrama de componentes del entorno virtual . . . . .	28
4.3. Construcción del entorno virtual . . . . .	29
4.3.1. Instalación de CentOS 7 . . . . .	29
4.3.2. Configuración de la Red . . . . .	32
4.3.2.1. Archivos de configuración de interfaces . . . . .	32
4.3.2.2. Interfaces Ethernet . . . . .	32
4.3.3. Instalar modo gráfico en Centos 7 . . . . .	34
4.3.4. Instalación Kernel Virtual Machine + QEMU . . . . .	36
4.3.5. Virt-install . . . . .	36
4.3.5.1. Creación de invitados con virt-install . . . . .	36
4.3.6. Libvirt-python . . . . .	38
4.3.7. Virt-manager . . . . .	38
4.3.7.1. Instalación de una máquina virtual con virt-manager .	38
4.3.8. Libvirt-client . . . . .	46
4.4. Configuración el entorno virtual . . . . .	47
4.4.1. Configuración de la red . . . . .	47
4.4.1.1. Creación de un bonding . . . . .	48
4.4.1.2. Traducción de Direcciones de Red (Red NAT) . . . . .	50
4.4.1.3. Gestión de una red virtual NAT . . . . .	51
4.4.1.4. Creación de una red virtual NAT con virt-manager . .	52
4.4.1.5. Conexión de una red virtual a un invitado . . . . .	55
4.4.2. Configuración de almacenamiento . . . . .	57
4.4.2.1. Creación de almacenamiento usando virsh . . . . .	57
4.4.2.2. Crear el archivo de configuración del conjunto de alma-	
cenamiento . . . . .	58
4.4.2.3. Agregar grupo de almacenamiento . . . . .	59
4.4.2.4. Iniciar la agrupación de almacenamiento . . . . .	59
4.4.2.5. Activar inicio automático . . . . .	60

4.4.2.6.	Verificar la configuración de la agrupación de almacenamiento . . . . .	60
4.4.2.7.	Eliminar el archivo de configuración temporal . . . . .	60
4.4.2.8.	Eliminación de una agrupación de almacenamiento mediante virsh . . . . .	61
4.4.2.9.	Grupo de almacenamiento usando Virt Manager . . . . .	61
4.4.3.	Configuración de recursos . . . . .	65
4.4.3.1.	Configuración del dispositivo de la máquina virtual . . . . .	66
4.4.3.2.	Dispositivos PCI . . . . .	66
4.4.3.3.	Asignación de un dispositivo PCI con virsh . . . . .	67
4.4.3.4.	Asignación de un dispositivo PCI con virt-manager . . . . .	70
4.5.	Configuración de la seguridad del entorno virtual . . . . .	72
4.5.1.	Configuración del firewall . . . . .	72
4.5.2.	Firewall modo gráfico en CentOS 7 . . . . .	74
4.5.3.	Fail2ban . . . . .	74
4.5.4.	Instalación Fail2Ban . . . . .	75
4.5.5.	Añadir un archivo para proteger SSH con Fail2Ban . . . . .	77
4.5.6.	Ejecución de servicio Fail2Ban . . . . .	78
4.6.	Creación del entorno virtual . . . . .	79
4.6.1.	Creación de máquina virtual . . . . .	80
4.6.1.1.	Descarga de Ubuntu . . . . .	80
4.6.1.2.	Instalar Ubuntu con virt-manager . . . . .	81
4.6.2.	Clonación de máquinas virtuales . . . . .	89
4.6.3.	Instalación Vídeo QXL driver y SPICE VDAgent en máquina virtual . . . . .	91
4.6.4.	Instalación Virt-viewer e ingreso de forma remota a una máquina virtual . . . . .	92
4.6.5.	Conectar dispositivo de almacenamiento USB de forma remota utilizando Virt-viewer . . . . .	94
<b>5.</b>	<b>Validación</b>	<b>96</b>
5.1.	Monitorización del entorno virtual . . . . .	96
5.1.1.	Herramienta de monitoreo Munin . . . . .	96

5.1.2.	Instalación de Munin . . . . .	97
5.1.3.	Monitoreo de uso de disco con Munin . . . . .	99
5.1.4.	Como usar Munin . . . . .	99
5.1.4.1.	Monitoreo del uso de CPU con Munin . . . . .	100
5.1.4.2.	Monitoreo del uso de disco con Munin . . . . .	102
5.1.4.3.	Monitoreo del uso de red con Munin . . . . .	107
5.1.4.4.	Monitoreo del uso de memoria RAM con Munin . . . . .	108
<b>6.</b>	<b>Conclusiones</b>	<b>110</b>
	<b>Bibliografía</b>	<b>113</b>
	<b>Apéndice A: Glosario</b>	<b>117</b>

# Índice de figuras

2.1. Hipervisor tipo 1 (Tanenbaum and García, 2009). . . . .	6
2.2. Hipervisor tipo 2 (Tanenbaum and García, 2009). . . . .	7
2.3. Logo KVM. . . . .	8
2.4. Logo QEMU. . . . .	9
3.1. Logo Virtualbox. . . . .	11
3.2. Logo VMWare. . . . .	13
3.3. Logo Citrix. . . . .	14
3.4. Logo Hyper-V. . . . .	15
4.1. Centos 7 Logo (CentOS, 2016). . . . .	23
4.2. Lubuntu Logo (Lubuntu, 2016). . . . .	24
4.3. Esquema protocolo Spice (Spice, 2016). . . . .	24
4.4. Diagrama de distribución del servidor (Fuente Propia). . . . .	25
4.5. Diagrama de casos de uso del servidor (Fuente Propia). . . . .	27
4.6. Diagrama de componentes del servidor (Fuente Propia). . . . .	28
4.7. Menú instalación de CentOS 7 (Fuente Propia). . . . .	29
4.8. Selección instalación mínima, instalación CentOS 7. . . . .	30
4.9. Configuración contraseña root y creación de usuarios, instalación CentOS 7 (Fuente Propia). . . . .	31
4.10. Logueo modo root en CentOS 7 (Fuente Propia). . . . .	32
4.11. Interfaz de red ifcfg-eno0 en CentOS 7 (Fuente Propia). . . . .	33
4.12. Configuración de interfaz de red ifcfg-eno0 en CentOS 7 (Fuente Propia). . . . .	33
4.13. Reiniciar red en CentOS 7 (Fuente Propia). . . . .	34
4.14. Descargar modo gráfico GUI para CentOS 7 (Fuente Propia). . . . .	34
4.15. Selección target modo gráfico para CentOS 7 (Fuente Propia). . . . .	35

4.16. Modo gráfico GUI en CentOS 7 (Fuente Propia). . . . .	35
4.17. Aplicación Virt Manager (Fuente Propia). . . . .	39
4.18. Botón para crear máquina virtual con Virt Manager (Fuente Propia). . . . .	40
4.19. Selección tipo de instalación Virt Manager (Fuente Propia). . . . .	41
4.20. Selección tipo de imagen ISO Virt Manager (Fuente Propia). . . . .	41
4.21. Asignación de memoria RAM y cantidad de CPU con Virt Manager (Fuente Propia). . . . .	43
4.22. Asignación de almacenamiento con Virt Manager (Fuente Propia). . . . .	44
4.23. Asignación de grupo de almacenamiento Virt Manager (Fuente Propia). . . . .	45
4.24. Asignación de nombre de almacenamiento Virt Manager (Fuente Propia). . . . .	45
4.25. Asignación de nombre a máquina virtual Virt Manager (Fuente Propia). . . . .	46
4.26. Grupos de paquetes de virtualización (Laura Novich, 2015) (Fuente Propia). . . . .	47
4.27. Creación interfaz bond en línea de comandos en CentOS 7 (Fuente Propia). . . . .	48
4.28. Configuración interfaz ifcfg-eno1 modo comando en CentOS 7 (Fuente Propia). . . . .	49
4.29. Configuración interfaz ifcfg-eno2 modo comando en CentOS 7 (Fuente Propia). . . . .	49
4.30. Interruptor de red virtual utilizando NAT con dos invitados (Fuente Propia). . . . .	50
4.31. Selección de los detalles de conexión (Fuente Propia). . . . .	51
4.32. Configuración de red virtual (Fuente Propia). . . . .	51
4.33. Configuración de red virtual (Fuente Propia). . . . .	52
4.34. Nombrar su red virtual (Fuente Propia). . . . .	52
4.35. Elegir un espacio de direcciones IPv4 (Fuente Propia). . . . .	53
4.36. Elegir un espacio de direcciones IPv6 (Fuente Propia). . . . .	53
4.37. Conexión a una red física (Fuente Propia). . . . .	54
4.38. Red 1 creada (Fuente Propia). . . . .	54
4.39. Selección de una máquina virtual (Fuente Propia). . . . .	55
4.40. Visualización de los detalles de la máquina virtual (Fuente Propia). . . . .	55
4.41. La ventana Detalles de la máquina virtual. . . . .	56
4.42. Selección de red en la ventana Añadir Nuevo Hardware Virtual (Fuente Propia). . . . .	56

4.43. Nueva red mostrada en la lista de hardware de invitados (Fuente Propia).	57
4.44. Detalles de conexión con Virt Manager (Fuente Propia).	62
4.45. Ventana de almacenamiento Virt Manager (Fuente Propia).	62
4.46. Nombre y tipo del grupo de almacenamiento con Virt Manager (Fuente Propia).	63
4.47. Ruta de la agrupación de almacenamiento (Fuente Propia).	63
4.48. Confirmación de la lista de almacenamiento con Virt Manager (Fuente Propia).	64
4.49. Eliminar grupo de almacenamiento (Fuente Propia).	65
4.50. Asignar un nuevo dispositivo a máquina virtual con virt-manager (Fuente Propia).	70
4.51. El asistente para agregar nuevo hardware virtual (Fuente Propia).	71
4.52. Ventana de información de hardware de la máquina virtual (Fuente Propia).	71
4.53. Detalles zona por defecto firewall (Fuente Propia).	72
4.54. Firewall aplicación en CentOS 7 (Fuente Propia).	74
4.55. Archivo jail.local del servidor (Fuente Propia).	76
4.56. Archido jail.local del servidor (Fuente Propia).	77
4.57. Registro de fallas de intentos de inicio con contraseña (Fuente Propia).	78
4.58. Seleccion de ISO distribución Lubuntu modo grafico (Fuente Propia).	80
4.59. Inicio aplicación virt-manager (Fuente Propia).	81
4.60. Crear una nueva máquina virtual virt-manager (Fuente Propia).	81
4.61. Medio de instalación ISO o CDRROM, virt-manager (Fuente Propia).	81
4.62. Buscar imagen ISO, virt-manager (Fuente Propia).	82
4.63. Seleccionar imagen ISO, virt-manager (Fuente Propia).	82
4.64. Buscar el sistema operativo, virt-manager (Fuente Propia).	83
4.65. Seleccionar el sistema operativo, virt-manager (Fuente Propia).	83
4.66. Asignación de memoria RAM y cantidad CPUs virtuales con virt-manager (Fuente Propia).	84
4.67. Asignar espacio de almacenamiento con virt-manager (Fuente Propia).	84
4.68. Seleccionar grupo de almacenamiento con virt-manager (Fuente Propia).	85
4.69. Escribir nombre de espacio de almacenamiento con virt-manager (Fuente Propia).	85

4.70. Seleccionar el volumen de almacenamiento con virt-manager (Fuente Propia). . . . .	86
4.71. Dirección de volumen de almacenamiento con virt-manager (Fuente Propia). . . . .	86
4.72. Seleccionar red NAT para nueva máquina virtual con virt-manager (Fuente Propia). . . . .	87
4.73. Configurar video QXL con virt-manager (Fuente Propia). . . . .	87
4.74. Iniciar e instalar máquina virtual Lubuntu con virt-manager (Fuente Propia). . . . .	88
4.75. Abrir virt-manager. . . . .	89
4.76. Opción de clonación de máquina virtual con virt-manager (Fuente Propia). . . . .	89
4.77. Nombrar nueva máquina clonada con virt-manager (Fuente Propia). . . . .	90
4.78. Virt Manager creando clon de máquina virtual (Fuente Propia). . . . .	90
4.79. Instalar QXL driver en linea comando Lubuntu (Fuente Propia). . . . .	91
4.80. Instalar QXL driver en linea comando Lubuntu (Fuente Propia). . . . .	91
4.81. Sección descargas, pagina oficial de SPICE (Fuente Propia). . . . .	92
4.82. Conectarse a una máquina virtual por medio de virt-viewer (Fuente Propia). . . . .	93
4.83. Escribir contraseña de máquina virtual en virt-viewer (Fuente Propia). . . . .	93
4.84. Máquina virtual Lubuntu accedida mediante virt-viewer (Fuente Propia). . . . .	94
4.85. USB device selection desde el menú de virt-viewer (Fuente Propia). . . . .	94
4.86. Selección de dispositivo USB a conectar de forma remota con virt-viewer (Fuente Propia). . . . .	95
5.1. Logo Munin (Munin, 2016). . . . .	97
5.2. Ingresando a Munin desde el navegador (Fuente Propia). . . . .	99
5.3. Aplicación Munin (Fuente Propia) . . . . .	99
5.4. Munin, uso de CPU en el mes de Octubre (Fuente Propia). . . . .	100
5.5. Uso de comando top (Fuente Propia). . . . .	101
5.6. Tiempo de saturación para disco sda (Fuente Propia). . . . .	102
5.7. Tiempo de saturación para disco sdb (Fuente Propia). . . . .	103
5.8. Grafico saturación sda (Fuente Propia). . . . .	104
5.9. Grafico saturación sdb (Fuente Propia). . . . .	105

5.10. Trafico de red en bond0 (Fuente Propia). . . . .	107
5.11. Uso de memoria RAM del servidor (Fuente Propia). . . . .	108

# Índice de tablas

3.1. Características de XenServer y KVM (Rayamajhi, 2016). . . . .	16
3.2. Características de VirtualBox y VMware (Rayamajhi, 2016). . . . .	16
3.3. Características de Microsoft Hyper-V (Collins, 2017). . . . .	17

# Capítulo 1

## Introducción

### 1.1. Descripción del Problema

Este proyecto está enfocado en la implementación de un servidor de máquinas virtuales didácticas para la Universidad Católica de la Santísima Concepción con el fin de que alumnos adquieran competencias en configuración, administración, operación y desarrollo en sistemas computacionales.

El desafío de este proyecto es el estudio e instalación de KVM (Kernel-based Virtual Machine) usando QEMU (Quick Emulator) como herramienta de emulación de máquinas virtuales y todas las herramientas asociadas para mejorar su rendimiento como servidor de virtualización en Linux. También instalar, configurar y poblar el servidor con máquinas virtuales. Culminando con la validación del servidor a través de un set de pruebas para medir los tiempos de respuestas, capacidad de procesamiento y usos de recursos computacionales.

### 1.2. Justificación del Problema

- Los recursos computacionales de los computadores de los laboratorios son restringidos, no tienen permisos de súper usuario ni modo administrador, dificultando el trabajo en la instalación de software y con esto limitando el aprendizaje de los estudiantes.

- Para la universidad es una oportunidad de bajar costos en la adquisición de equipamiento de alta gama y también de su renovación.
- Se pueden crear laboratorios por medio de la virtualización simplificando la administración en comparación a tener laboratorios con máquinas físicas
- La universidad brindara a los estudiantes un entorno en donde pueden adquirir el aprendizaje interactuando con entornos reales virtualizados como tecnología de punta.

### **1.3. Delimitación del Problema**

- La implementación del servidor debe ser efectuada dentro de un semestre académico.
- Se limitara a usar software libre en su implementación.
- Se utilizara CentOS como anfitrión en el servidor.
- El servidor debe estar en la misma subred de los laboratorios.
- Los testeos del servidor se van a aplicar a dos asignaturas, sistemas operativos y redes de computadores.

### **1.4. Objetivo General**

Implementar un servidor de máquinas virtuales para el departamento de informática de la Universidad Católica de la Santísima Concepción.

### **1.5. Objetivos Específicos**

- Estudiar la virtualización de plataformas computacionales, herramientas de emulación y estudiar un administrador de conexión a escritorios virtuales remotos.
- Definir los requisitos del sistema.
- Diseñar el servidor de máquinas virtuales.

- Instalar el servidor.
- Testear y monitorear el servidor.

## 1.6. Metodología

- Para estudiar la virtualización de plataformas computacionales, herramientas de emulación y estudiar un administrador de conexión a escritorios virtuales se acudió a bibliografía, se busco información a través de Google Scholar y se ordeno la información con mapas mentales.
- En el diseño del servidor de máquinas virtuales se ocuparon diseños UML utilizando diagramas de casos de usos, diagrama de componentes y diagrama de distribución.
- En la implementación del servidor se utilizo un método propuesto por Eric Siebert usado en la implementación de un servidor VMware. El cual define lo siguiente:
  - Evaluación del entorno actual.
  - Planificación del entorno virtual.
  - Construcción del entorno virtual.
  - Configuración del entorno virtual.
  - Configuración de la seguridad del entorno virtual.
  - Creación del entorno virtual.
  - Monitorización del entorno virtual.
  - Mantención del entorno virtual.
  - Creación de una copia de seguridad del entorno virtual.
  - Solución de problemas del entorno virtual.

## 1.7. Esquema de documento de Tesis

- **Capítulo I:** En este capítulo se describe el problema de la virtualización del ambiente físico, se justifica, delimita y se plantea la metodología de trabajo para ejecutar el proyecto. También se establecen los objetivos específicos y objetivo general del proyecto.
- **Capítulo II:** En este capítulo se da a conocer el marco teórico del proyecto. Se definen conceptos importantes para entender la virtualización. Conceptos como hipervisor, paravirtualización, también se explica que es un administrador de escritorio virtual remoto.
- **Capítulo III:** En este capítulo se dio a conocer el estado del arte a través de ejemplos para los tipos de hipervisores 1 y 2 respectivamente. Se dan a conocer sus características, sus ventajas y se finaliza comparándolos en una tabla.
- **Capítulo IV:** En este capítulo se procedió con la ejecución del proyecto con un método. Método propuesto por Eric Siebert para virtualizar ambientes físicos. Se ejecutaron las siguientes etapas.
  1. Evaluación del entorno actual.
  2. Planificación del entorno virtual.
  3. Construcción del entorno virtual.
  4. Configuración del entorno virtual.
  5. Configuración de la seguridad del entorno virtual.
  6. Creación del entorno virtual.
  7. Monitorización del entorno virtual.
  8. Mantención del entorno virtual.
  9. Creación de una copia de seguridad del entorno virtual.
  10. Solución de problemas del entorno virtual.
- **Capítulo V:** En este capítulo se instaló y configuró una herramienta de monitoreo para el servidor “Munin” con la cual se analizó el uso de CPU, el uso de disco, el uso de la red y el uso de la memoria RAM por medio de gráficos.

# Capítulo 2

## Marco teórico

En este capítulo se explicaran conceptos y utilidades informáticas necesarias relacionadas con la virtualización antes de la realización del proyecto con el propósito de introducirnos en el ambiente virtual que se instalara.

### 2.1. Virtualización

La virtualización se define como la ejecución de varios sistemas operativos simultáneos en forma aislada en un solo sistema. En general utilizan un hipervisor que es una capa de software o subsistema que ejerce control sobre el hardware que permite la ejecución múltiples sistemas operativos llamados invitados en una sola máquina física llamada anfitrión (Red Hat, 2016a).

### 2.2. Hipervisor

Para la virtualización de una máquina virtual existen dos formas de llevarlo a cabo, una se denomina hipervisor de tipo 1 y la otra hipervisor de tipo 2. Para ambas el hardware debe funcionar de igual manera que una máquina real y se debe poder instalar sobre ella cualquier sistema operativo (Tanenbaum and García, 2009).

### 2.2.1. Hipervisores tipo 1

Una primera forma para virtualizar una máquina virtual se denomina hipervisor de tipo 1, en donde el sistema operativo es el único proceso que se ejecuta en modo kernel. Su función es hacer copias de su propio hardware a lo cual se denomina máquina virtual. Su ejecución funciona de una forma semejante a los procesos de un sistema operativo normal (Tanenbaum and García, 2009). Hipervisores tipo 1 conocidos son VMware ESXi, Xen, Citrix XenServer y Microsoft Hyper-V Server.

Podemos ver un esquema del hipervisor tipo 1 en la Figura 2.1

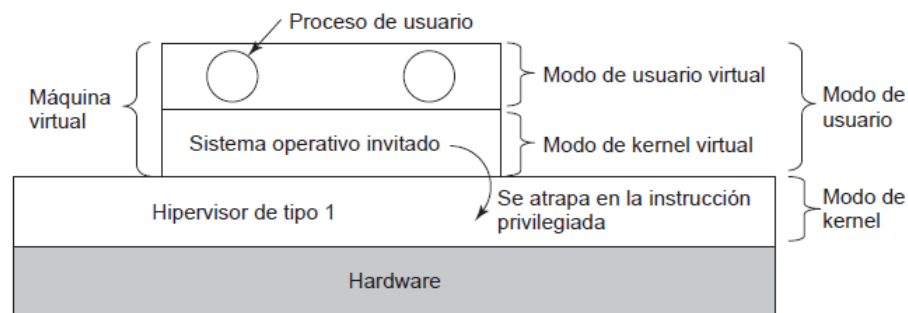


Figura 2.1: Hipervisor tipo 1 (Tanenbaum and García, 2009).

Entre la virtualización de tipo 1 se pueden distinguir a su vez dos subtipos:

- **Arquitectura monolítica**

La arquitectura monolítica es compleja, ya que cuando una máquina virtual realiza una llamada al Hardware, el hardware virtualizado la redirecciona hacia los drivers en el hipervisor, el cual enruta esa llamada al recurso real. Un ejemplo de esto es VMware ESX (Data Keeper, 2016). El proceso que sigue una llamada a hardware en un sistema virtualizado usando un hipervisor de tipo monolítico es la siguiente:

1. El hardware emulado debe interceptar la llamada.
2. El Monitor de Máquinas Virtuales (VMM) redirige las llamadas hacia los drivers de dispositivo que operan dentro del hipervisor, lo cual requiere de varios cambios de contexto en el código de la llamada.
3. Los drivers del hipervisor enrutan la llamada hacia el dispositivo físico.

Este funcionamiento implica desarrollar drivers específicos para el hipervisor de cada componente hardware.

- Arquitectura micro-kernelizada** Esta arquitectura es más sencilla porque las máquinas virtuales no requieren de drivers específicos accediendo directamente al hardware mediante los drivers usados por el hipervisor, de esta manera el hipervisor será como una capa transparente dedicada a separar y administrar las distintas máquinas virtuales. De esta forma se puede aumentar el rendimiento mediante la reducción de código intermediario y el número de cambios de contexto, sino que también se puede aumentar la estabilidad de los sistemas, al existir menos componentes, disminuyendo la superficie de trabajo del hipervisor (Data Keeper, 2016).

### 2.2.2. Hipervisores tipo 2

Una segunda forma para virtualizar una máquina virtual se denomina hipervisor de tipo 2 en donde el hipervisor funciona encima de un sistema operativo anfitrión, en esta forma el sistema operativo funciona como un programa de usuario, en el cual las líneas de código se procesan y se sitúan en el cache para su ejecución (Tanenbaum and García, 2009). Un hipervisor tipo 2 conocido es VirtualBox. Podemos ver un esquema del hipervisor tipo 2 en la Figura 2.2.

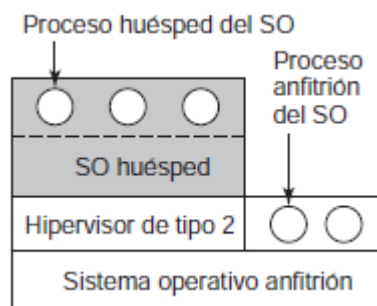


Figura 2.2: Hipervisor tipo 2 (Tanenbaum and García, 2009).

## 2.3. Paravirtualización

Otra forma de virtualizar un sistema operativo es la paravirtualización, la cual consiste en hacer modificaciones al código fuente del sistema operativo invitado de tal manera que no efectúe llamadas sensibles si no que efectúe llamadas al hipervisor. Utiliza una interfaz en donde se registran las operaciones a las cuales pueden acceder o llamar los sistemas operativos invitados, este interfaz forma una API (Interfaz de programación de aplicaciones). De esta manera el hipervisor se transforma en un microkernel. Los hipervisores utilizan instrucciones sensibles por motivo de falta de acceso al código fuente del sistema operativo invitado en el caso de Windows o por las diversas distribuciones de Linux (Tanenbaum and García, 2009).

## 2.4. KVM

KVM (Máquina virtual basada en el kernel) es un hipervisor tipo 1 que se usa para lograr una virtualización completa para Linux en arquitecturas x86 y posee extensiones de virtualización para Intel VT y AMD-V. Su estructura consta de un módulo de núcleo, `kvm.ko`, que brinda infraestructura de virtualización de núcleo y un módulo específico de procesador, `KVM-intel.ko` o `KVM-amd.ko`.

Cuando se utiliza KVM, se logran ejecutar múltiples máquinas virtuales que se ejecutan sin hacer modificaciones a imágenes de sistemas operativos de Linux o Windows. Cada máquina virtual tiene un hardware virtualizado para si misma, es decir una tarjeta de red, disco, adaptador de gráficos, etc.

KVM es un software de código abierto. El componente de espacio de usuario de KVM se incluye en la línea principal de QEMU, a partir de la versión 1,3 (Linux-KVM, 2016).



Figura 2.3: Logo KVM.

## 2.5. QEMU

QEMU es un emulador y una máquina de virtualización que permite la creación de máquinas virtuales dentro de un sistema operativo. En el caso de utilizar QEMU como emulador de máquina virtual puede ejecutar sistemas operativos y programas en una máquina diferente. Si se utiliza en modo kernel se ejecutara con el hipervisor Xen o KVM. QEMU si se utiliza con KVM, puede utilizar arquitecturas x86, PowerPC, servidor embebido y huéspedes S390 (QEMU, 2016).



Figura 2.4: Logo QEMU.

## 2.6. Administradores de conexión a escritorios virtuales remotos

Administrador de conexión a escritorio remoto (Remote Desktop Connection) es la tecnología que permite a un equipo (cliente) conectarse a equipo remoto (anfitrión). Estos equipos pueden se pueden geográficamente en lugares distintos (Cavsi, 2016).

Funciona bajo el modelo cliente servidor. Las personas pueden conectarse a su equipo de trabajo utilizando un computador en casa y con esto lograr tener acceso a todos los programas, archivos y recursos de red, como si estuvieran delante del computador en el trabajo (Cavsi, 2016).

Un administrador de conexión a escritorio remoto se utiliza por los administradores de servidores o redes para acceder a equipos o servidores que se encuentran en otras ciudades o países permitiendo mantener centralizado el control en corporaciones o empresas (Cavsi, 2016).

Principales características :

1. Permite a un equipo conectarse al escritorio de otro equipo que esté conectado en la misma red o por medio de Internet (Microsoft, 2016a).
2. Utiliza el modelo cliente servidor (Spice, 2016).

# Capítulo 3

## Estado del arte

### 3.1. VirtualBox

VirtualBox es una solución de virtualización con sistemas operativos invitados, es un hipervisor tipo 2. Fue desarrollado por Innotek, Sun Microsystems y la comunidad Linux. Puede ejecutar máquinas virtuales de 32 y 64 bits Linux, Microsoft Windows, Solaris, BSD, o IBM OS/2 en hosts Microsoft Windows, Mac OS, Linux y OpenSolaris (VirtualBox, 2016).



Figura 3.1: Logo Virtualbox.

Las características que ofrece VirtualBox son útiles para distintos escenarios:

1. **La ejecución de múltiples sistemas operativos en forma simultánea:**  
VirtualBox permite ejecutar más de un sistema operativo al mismo tiempo. De esta manera, se puede ejecutar el software escrito para un sistema operativo en otro sin tener que realizar un reinicio en el sistema para utilizarlo.

2. **Los proveedores de software pueden usar máquinas virtuales para enviar configuraciones de software enteras:** Por ejemplo, la instalación de un servidor de correo en una máquina real puede ser una tarea complicada. Con VirtualBox, una configuración compleja puede ser embalado de manera fácil en una máquina virtual.

3. **Pruebas y recuperación de desastres:** Ya instalada, una máquina virtual y sus discos duros virtuales pueden considerarse como un contenedor que puede ser arbitrariamente congelado, despertado, copiado y transportado entre los hosts.

Además de eso, con el uso de otra de las características de VirtualBox conocida como instantáneas, se puede guardar un estado particular de una máquina virtual y volver a ese estado.

Cualquier número de instantáneas se pueden crear además de eliminar, lo que le permite viajar hacia atrás y adelante en el tiempo de la máquina virtual.

4. **Infraestructura de consolidación:** La virtualización puede reducir significativamente los costos de hardware y de la electricidad. Ya que la mayor parte del tiempo, las computadoras hoy en día sólo utilizan una fracción de su potencial. Por esta razón una gran cantidad de recursos de hardware, así como la electricidad se pierde de esta manera. Por eso en vez de ejecutar muchos equipos físicos que se utilizan sólo en parte, se puede empacar muchas máquinas virtuales en unos anfitriones y equilibrar las cargas entre ellos (VirtualBox, 2016).

## 3.2. VMware

VMware fue uno de los primeros en ofrecer soluciones para sistemas operativos Windows como Linux. La empresa tiene soluciones de virtualización en todos los niveles y para todas las necesidades.

Está compuesto de un sistema operativo autónomo que suministra el entorno de gestión (hipervisor tipo 1), administración y ejecución al software hipervisor; y de los servicios y servidores que permiten la interacción con el software de gestión y administración de las máquinas virtuales (VMware, 2016).



Figura 3.2: Logo VMWare.

Permite ejecutar varios sistemas operativos dentro de un mismo hardware de manera simultánea, de esta forma se logra un mayor aprovechamiento de recursos. Todas las máquinas virtuales están conectadas a la red, no hay necesidad de reiniciar las máquinas si se quiere ejecutar algunas de las máquinas virtuales. La mayor parte de las instrucciones se ejecutan directamente sobre el hardware físico. También permite importar máquinas físicas para su utilización como máquinas virtuales. Dependiendo de su versión éste puede ser un hipervisor de virtualización completa o paravirtualización (VMware, 2016). Posee algunas versiones tales como:

- **VMware ESX Server:** Es un hipervisor instalado directamente en el hardware de esta manera se crea una capa de virtualización entre el hardware y el sistema operativo.
- **VMware Vsphere:** Transforma los centros de datos en infraestructura de Cloud Computing, aprovechando así las capacidades de la virtualización.
- **VMware Player:** Solo permite la ejecución de máquinas virtuales ya creadas con software VMware.
- **VMWare Server:** Permite la ejecución concurrente de máquinas virtuales, además de la administración remota de los mismos.
- **VMware WorkStation:** Permite la virtualización de escritorio a un nivel superior, permitiendo una compatibilidad de aplicaciones entre sistemas operativos.

### 3.3. Citrix XenServer

Citrix XenServer es un hipervisor tipo 1 que proporciona una gestión eficiente de Windows y Linux en el uso de máquinas virtuales y ofrece una plataforma extremadamente económica para aplicaciones, escritorios, y la consolidación de servidores. Citrix continúa creando nuevas características, fue el primer hipervisor en integrar gráficos virtualizados con NVIDIA GRID vGPU, y la introducción en memoria caché de lectura para palear el almacenamiento de E/S que aumenta el rendimiento (Citrix, 2016).



Figura 3.3: Logo Citrix.

Características que solo ofrece XenServer:

1. Soporta la virtualización de GPU Intel GVT-g, una GPU incorporada a la CPU que no requiere hardware adicional, y es una oferta que VMware no admite hoy.
2. Se integra con la concesión de licencias de XenApp/XenDesktop; no se requieren licencias de XenServer para apoyar los proyectos de virtualización de aplicaciones y de escritorio de Citrix.
3. Ofrece GPU de paso a través de virtualización y implementaciones de GPU que son superiores a las implementaciones basadas en interceptación basados en software como VMware vSGA.
4. Optimiza el escritorio de Citrix, los gráficos y las cargas de trabajo en la nube, como XenApp y XenDesktop, a un precio mucho más competitivo.

### 3.4. Hyper-V

El hipervisor Hyper-V permite la creación y administración de un entorno informático virtualizado utilizando tecnología de virtualización integrada en Windows Server. Con la instalación de Hyper-V, se instalan los componentes necesarios y también, si lo desea, herramientas de administración. Los componentes que se requieren incluyen el hipervisor de Windows, el servicio Administración de máquinas virtuales de Hyper-V, el proveedor de WMI de virtualización y otros componentes de virtualización, como el bus de máquina virtual (VMbus), el proveedor de servicios de virtualización (VSP) y el controlador de infraestructura virtual (VID) (Microsoft, 2016b).



Figura 3.4: Logo Hyper-V.

Las herramientas de administración de Hyper-V se componen de lo siguiente:

- **Herramientas de administración basadas en la interfaz gráfica de usuario:** Contienen un administrador de Hyper-V, un complemento Microsoft Management Console (MMC) y un administrador de conexión a un escritorio virtual, que da acceso a la salida de vídeo de una máquina virtual para poder interactuar con esta.
- **Cmdlets específicos de Hyper-V para PowerShell-Windows Server 2012:** Cmdlets es la unidad de funcionalidad más pequeña del Shell, incluye un módulo de Hyper-V que entrega acceso de la línea de comandos a toda la funcionalidad disponible en la GUI (Interfaz gráfica de usuario), también acceso a la funcionalidad no disponible en ella.

Si se usa el administrador del servidor para instalar Hyper-V, se incluirán herramientas de administración, a menos que las excluya expresamente. Si usa Windows PowerShell para instalar Hyper-V, las herramientas de administración no se incluirán de manera predeterminada (Microsoft, 2016b).

### 3.5. Evaluación y comparación de los Hipervisores

Un método de comparación es el método de virtualización. Cada hipervisor está diseñado concentrándose en alguna tarea específica a realizar (Rayamajhi, 2016), ver tabla 3.1 , tabla 3.2 y tabla 3.3.

Tabla 3.1: Características de XenServer y KVM (Rayamajhi, 2016).

	XenServer	KVM
Paravirtualización	Si	No
Virtualización completa	Si	Si
Host PC	x86, x86-64, IA64	x86, x86-64, IA64, PPC
Guest PC	x86, x86-64, IA64	x86, x86-64, IA64, PPC
Host OS	Linux, UNIX	Linux
Guest OS	Windows, Linux, UNIX	Windows, Linux, UNIX
VT x/AMD v	Opt	Req
Corcs Admitidos	128	16
Memoria Soportada	1TB	1TB
Aceleración 3D	Xen-GL	VMGL
Live Migration	Si	Si
Licencia	GPL	GPL
Costo	Pagado	Gratuito

Tabla 3.2: Características de VirtualBox y VMware (Rayamajhi, 2016).

	VirtualBox	VMware
Paravirtualización	No	No
Virtualización completa	Si	Si
Host PC	x86, x86-64	x86, x86-64
Guest PC	x86, x86-64	x86, x86-64
Host OS	Windows, Linux, UNIX	Propietario Linux
Guest OS	Windows, Linux, UNIX	Windows, Linux, UNIX
VT x/AMD v	Opt	Opt
Corcs Admitidos	32	8
Memoria Soportada	16GB	61GB
Aceleración 3D	Open-GL	Open-GL, DirectX
Live Migration	Si	Si
Licencia	GPL	Propietario
Costo	Gratuito	Pagado

Tabla 3.3: Características de Microsoft Hyper-V (Collins, 2017).

	Hyper-V
Paravirtualización	Si
Virtualización completa	Si
Host PC	x86, x86-64
Guest PC	x86, x86-64
Host OS	Windows
Guest OS	Windows, Linux, UNIX
VT x/AMD v	Opt
Corcs Admitidos	64
Memoria Soportada	1TB
Aceleración 3D	RemoteFX
Live Migration	Si
Licencia	OEM, OLP
Costo	Pagado

# Capítulo 4

## Desarrollo

La virtualización ha obtenido popularidad en los últimos años y se utiliza en la mayoría de los centros de datos. En la gran mayoría de las empresas, no se trata solo de cuándo van a virtualizar su infraestructura, sino de cuánto van a virtualizar (Siebert, 2009).

La virtualización tiene beneficios sobre los servidores físicos tradicionales, y la tecnología está en constante evolución y mejora, por esto la decisión de si la virtualización es el camino, resulta más fácil. De los tantos proveedores que ofrecen hipervisores de virtualización, como VMware, Microsoft, Citrix y Red Hat los proveedores de tecnología física de Hardware, como AMD e Intel, están cambiando sus productos para optimizarlos para trabajar con hosts virtuales. Mas aun, un sin numero de proveedores han escrito aplicaciones específicas de virtualización y modificaciones al hardware para trabajar en entornos virtuales, porque reconocen que la virtualización llevo para quedarse. La virtualización se ha hecho tan popular hoy en día que casi todos los proveedores de software la soportan. También añadir que la mayoría de los vendedores han cambiado sus políticas de licencias para que sean amigables y así ejecutarse en máquinas virtuales (Siebert, 2009).

Para implementar el servidor se necesita una metodología o set de pasos para llevarlo a cabo, para esto se ha escogido una referencia bibliográfica *VMware VI3 Implementation and Administration* propuesto por **Eric Siebert** que ha estado trabajando con computadores por más de 25 años. En los últimos 15 años, su enfoque primordial ha

sido la administración de entornos de servidores Windows y el soporte de aplicaciones empresariales como servidores web, correo electrónico y de base de datos. Hace cuatro años, descubrió la tecnología de virtualización en la forma de VMware ESX y desde ahí se ha enganchado, estando innumerables horas aprendiendo el producto por dentro y por fuera. Ahora pasa mucho de su tiempo ayudando a otros en foros de apoyo de la comunidad de VMware, logrando un estatus de Gurú (Siebert, 2009).

El libro documenta todas las fases de la implementación de un proyecto de virtualización utilizando *VMware Virtual Infrastructure 3*, desde las etapas iniciales de planificación hasta el diseño, construcción, configuración, mantenimiento, solución de problemas y más (Siebert, 2009). Pero en la implementación de este servidor se usará el *hipervisor KVM* con *CentOS 7* como sistema operativo anfitrión. Solo se usará esta guía para tener un camino de ruta a seguir. Se utilizarán diversas herramientas de manera análoga a lo utilizado por VMware para cumplir con las diferentes etapas en la implementación de un servidor proporcionados por esta guía. Teniendo esto en claro, a continuación se presentará el método que se utilizará. En este proyecto solo se cubrirán las primeras 7 etapas para instalar servidor.

1. Evaluación del entorno actual.
2. Planificación del entorno virtual.
3. Construcción del entorno virtual.
4. Configuración del entorno virtual.
5. Configuración de la seguridad del entorno virtual.
6. Creación del entorno virtual.
7. Monitorización del entorno virtual.
8. Mantención del entorno virtual.
9. Creación de una copia de seguridad del entorno virtual.
10. Solución de problemas del entorno virtual.

## 4.1. Evaluación el entorno actual

Casi siempre al realizar proyectos de virtualización se requiere realizar migraciones desde servidores físicos a máquinas virtuales. Por esta razón, es necesario hacer un estudio sobre el escenario que requiere la virtualización para así adquirir un servidor que tenga las características adecuadas para virtualizar tal escenario (Siebert, 2009).

En nuestra situación, nuestro escenario consta de dos laboratorios, con denominación 1 y 5 respectivamente.

El laboratorio 1 consta de dos grupos, el primer grupo tiene una cantidad de 18 máquinas con las siguientes características:

- Procesador: Intel Core I7.
- Placa Madre: ASUS P8B75-M LX PLUS.
- Memoria Ram: 8GB.
- Tarjeta Video: Geforce GT 630 2GB.
- Disco Duro: 500GB.

El segundo grupo consta de 14 máquinas, las cuales tienen las siguientes características.

- Procesador: Intel Core I5.
- Placa Madre: Intel DH61CR.
- Memoria Ram: 4GB.
- Tarjeta Video: NO TIENE.
- Disco Duro: 500GB.

En efecto el laboratorio 1 dispone de 32 máquinas físicas divididas en dos grupos.

El laboratorio 5 consta de 24 máquinas con las siguientes características:

- Procesador: Intel Core I5.
- Notebook Samsung: Modelo NP470R5E.
- Memoria Ram: 8GB.
- Disco Duro: 1TB.

Ahora, lo que se virtualizara son 24 equipos capaces de ejecutar los siguientes programas:

- RPC-BIN.
- libncurses5-dev.
- Tomcat 8.
- Atom.
- Sublime-Text.
- Netbeans.
- Eclipse.
- GCC.
- JAVA.

En el listado de programas y aplicaciones anterior los que requieren mayor capacidad de recursos para ser ejecutados en un equipo son el IDE Netbeans y IDE Eclipse (Eclipse, 2016). Netbeans necesita como mínimo 512 MB de memoria RAM para su ejecución y en cuanto a velocidad de procesamiento refiriéndose a arquitecturas Intel se necesita mínimo un procesador Intel Pentium III o algun equivalente a 800 MHz. Para Eclipse se necesita 2 GB de memoria RAM minimo y una velocidad de procesamiento igual o superior a 800 Mhz (Oracle, 2016).

Para virtualizar el escenario requerido que consta de 24 máquinas se debe considerar que se dispone de un servidor *HPE Proliant DL160 Generation 9*.

## 4.2. Planificación el entorno virtual

Para asegurar una virtualización exitosa del ambiente que se quiere virtualizar, es necesario estudiar la mejor herramienta de virtualización, para ello se debe seleccionar un hipervisor (Siebert, 2009). Para este proyecto se utilizo KVM como hipervisor ya que el proyecto esta orientado al software libre.

También hay que estudiar el ambiente físico que se requiere virtualizar y con esto adquirir un servidor acorde a los recursos que utiliza el escenario. Recursos como la memoria RAM, velocidad de procesamiento, uso de la red y espacio de almacenamiento (Siebert, 2009).

Para el proyecto se dispuso de un servidor *HPE Proliant DL160 Generation 9* el cual posee las siguientes características:

- Dos procesadores Intel(R) Xeon(R) CPU E5-2603 v3 @ 1.60GHz.
- 64 GB de memoria RAM.
- OS Type 64-bit.
- Disco 2,0 TB.
- 2 Tarjetas de red.

Se debió analizar el servidor con el cual se dispuso para verificar si cumplía con los requisitos necesarios para la virtualización.

En cuanto a memoria RAM, se necesitaron 24 máquinas a virtualizar con 2GB RAM cada una. El servidor dispone de 62,5 GB de memoria RAM reales en el servidor, de dividió por 2048 MB y nos quedo una cifra de 31 máquinas aproximadamente. Se asigno al servidor 2048 MB de RAM así quedo una cifra de 30 máquinas disponibles a virtualizar. En cuanto a las unidades físicas, se asignaron 40 GB de memoria de almacenamiento por máquina virtual, quiere decir que 24 máquinas ocuparan espacio de almacenamiento de 960 GB aproximadamente restando la del SO anfitrión, lo cual es posible porque el servidor dispone de 2 TB de espacio de almacenamiento.

Como el servidor dispone de 2 tarjetas de red se implemento un channel bonding, una interfaz de red virtual que consiste en simular un dispositivo de red con un gran ancho de banda por medio de la unión de varias tarjetas de red independientes, permitiendo a las aplicaciones ver solo una interfaz de red. El channel bonding permite un mayor ancho de banda ya que suma los anchos de banda de las tarjetas de red, también permite un balanceo de carga en el trafico de red en las tarjetas de red y por ultimo si falla una tarjeta de red permite redirigir el tráfico de red solo por las tarjetas que estén en funcionamiento (Estréllate y Arde, 2016).

Para el servidor se procedió a instalar la distribución CentOS 7 Linux ya que es derivado de las fuentes de Red Hat Enterprise Linux (RHEL) además que es un proyecto de software libre impulsada por la comunidad que tiene por objetivo brindar una plataforma robusta a las comunidades de código abierto para construir y desarrollar (CentOS, 2016).



Figura 4.1: Centos 7 Logo (CentOS, 2016).

En las máquinas virtuales se instaló el sistema operativo Lubuntu, una variante de Ubuntu. Se escogió esta distribución porque es rápida y ligera. Utiliza el escritorio LXDE (Lightweight X11 Desktop Environment). Esta distribución está diseñada para sistemas que demanden pocos recursos (Distrowatch, 2016).



Figura 4.2: Lubuntu Logo (Lubuntu, 2016).

Para acceder a una máquina virtual se instaló un administrador de escritorio virtual remoto. En este caso se utilizó SPICE cuya función es ofrecer una solución completa de código abierto para el acceso remoto a máquinas virtuales la cual permite reproducir videos, grabar audio y conectar dispositivos USB (Spice, 2016).

Spice se divide en 4 componentes, protocolo, cliente, servidor y visitantes. El servidor Spice es la biblioteca que utiliza el hipervisor para dar acceso a las máquinas virtuales a través del protocolo SPICE (Spice, 2016).

Para hacer al protocolo SPICE funcional por el lado de los huéspedes en las máquinas virtuales en contenido de software se debe instalar el conductor QXL y SPICE VDAgent (Spice, 2016).

Esquema protocolo Spice en la Figura 4.3.

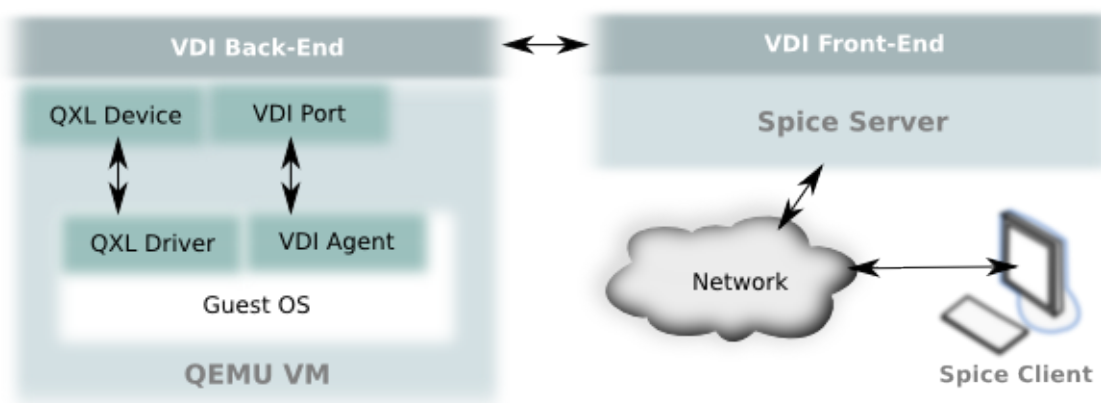


Figura 4.3: Esquema protocolo Spice (Spice, 2016).

### 4.2.1. Diagrama de distribución del entorno virtual

En UML existe la posibilidad de enfocarse solo en partes de todos los elementos que componen un sistema, en una parte tenemos la conceptualización, construcción de las soluciones en software y modelos abstractos, en la otra parte, esta la distribución de los equipos y el hardware que hará que el sistema funcione (Solvetic, 2016).

En los diagramas de distribución el enfoque se centra en mostrar la estructura física representándola en diagramas, de tal manera de establecer todos los mapas necesarios para documentar los requerimientos del sistema (Solvetic, 2016). Por esta razón en la implementación del servidor se ocupó un diagrama de distribución el cual se puede ver en la Figura 4.4.

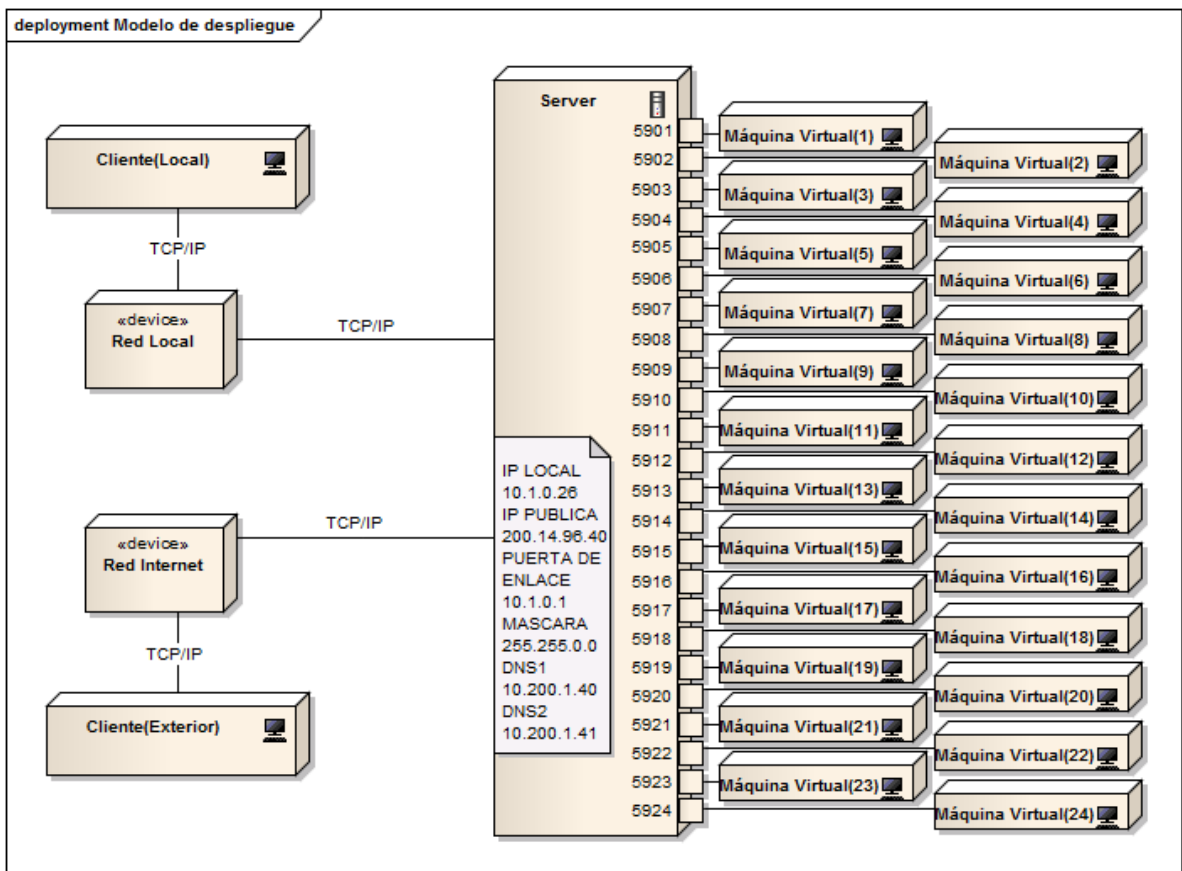


Figura 4.4: Diagrama de distribución del servidor (Fuente Propia).

Se puede ver en el medio al servidor que maneja a 24 máquinas virtuales a su izquierda, estas máquinas se van a crear con el hipervisor KVM. Cada máquina será asignada a un puerto del servidor, desde el puerto 5900 hasta el puerto 5924.

Se definió anteriormente que para poder acceder a una máquina virtual de forma remota se utilizaría el protocolo SPICE que es una solución completa de código abierto para el acceso remoto a máquinas virtuales. Quienes tendrán acceso al servidor serán clientes locales que estén en la misma sub-red del servidor y clientes exteriores en otras redes, redes exteriores por medio de Internet. Por lo tanto el servidor dispondrá de una IP pública y una IP local para el acceso.

### 4.2.2. Diagrama de casos de uso del entorno virtual

El diagrama de casos de uso muestra la forma en como uno o varios Clientes (Actores) operan con el sistema en funcionamiento, también la forma, el tipo de orden en que los elementos conviven (casos de uso) (Salinas, 2016). De esta manera se pudo comprender mejor el funcionamiento del servidor, diagrama de casos de uso del servidor ver en la Figura 4.5.

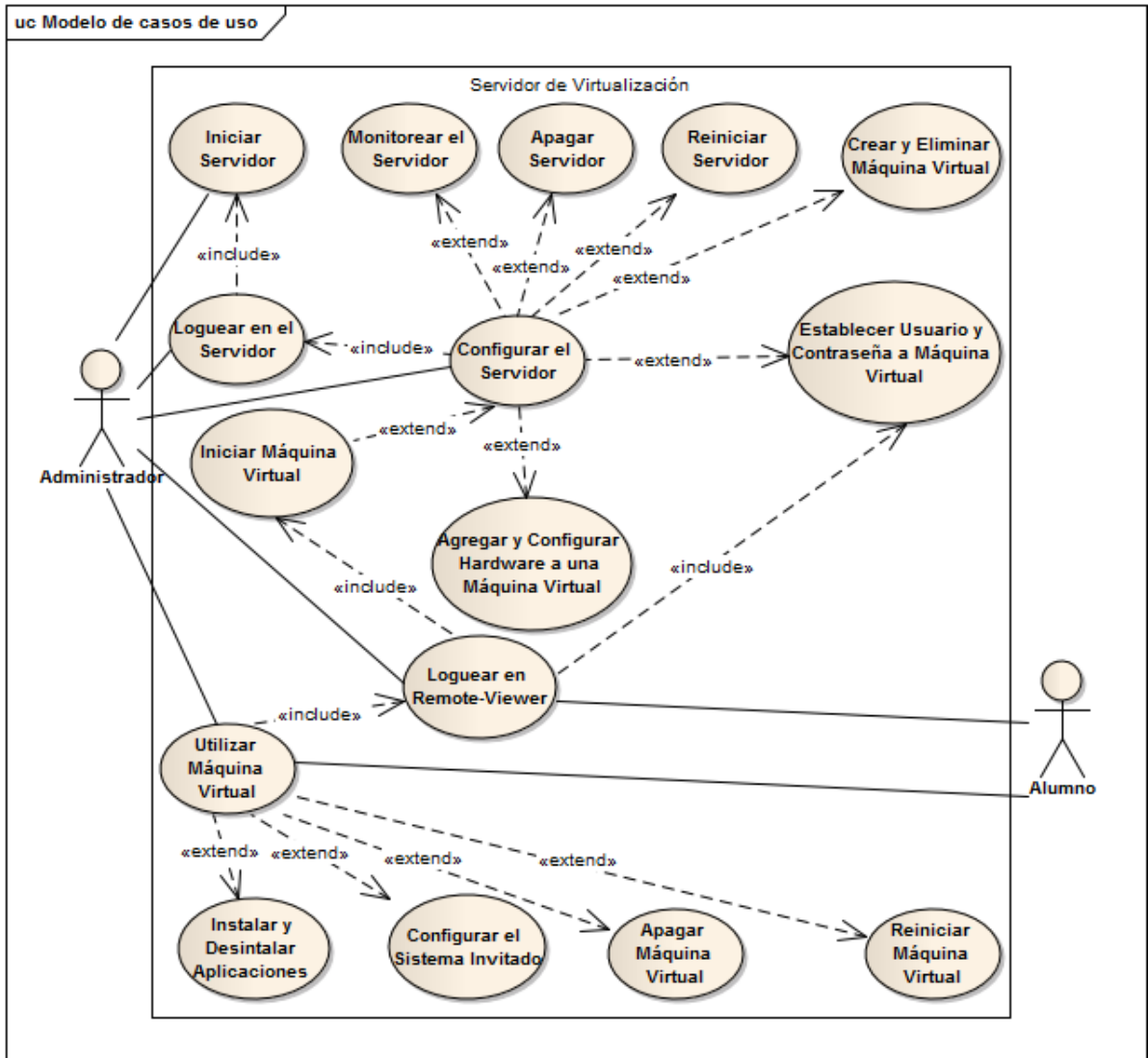


Figura 4.5: Diagrama de casos de uso del servidor (Fuente Propia).

En el diagrama podemos ver a dos actores, el *Administrador* del servidor y el *Alumno*. El administrador puede tener acceso al servidor y a una máquina virtual,

puede configurar ambos si están operativos. En cambio el *Alumno* solo puede utilizar y configurar una máquina virtual siempre y cuando este operativa.

### 4.2.3. Diagrama de componentes del entorno virtual

Un diagrama de componentes muestra las piezas del software y controladores embebidos entre otros. Dicho de otra forma, muestra los componentes que conformarán un sistema (Sparx, 2016). Se uso para mostrar como se relacionan los componentes de software en el servidor. Puede ver el diagrama de componentes del servidor en la siguiente Figura 4.6.

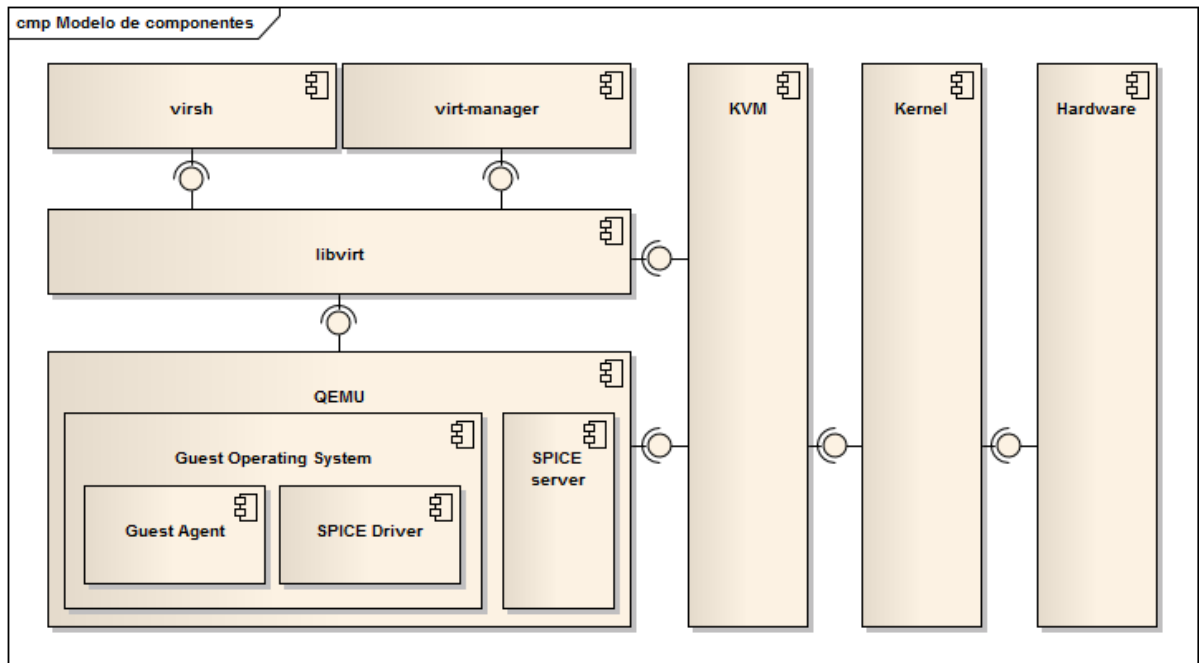


Figura 4.6: Diagrama de componentes del servidor (Fuente Propia).

Se puede observar a QEMU que tiene la responsabilidad de emular las máquinas virtuales integrando a SPICE Driver y SPICE server, con el fin de lograr que máquinas virtuales sean accesibles en forma remota. También se ve a libvirt que es una API que interactuando con virt-manager, virsh y con QEMU. Sirve para administrar las máquinas, su creación, eliminación, etc.

## 4.3. Construcción del entorno virtual

En esta etapa corresponde preparar todo lo necesario para la instalación del servidor, sistema operativo CentOS 7 en nuestro caso, el hipervisor KVM, sistema operativo Lubuntu en las máquinas invitadas y los programas que se utilizaran en ellas.

### 4.3.1. Instalación de CentOS 7

Se descargó de CentOS 7 desde el sitio oficial, en su versión mínima en formato ISO.

Después de descargado CentOS 7 versión mínima se procedió a crear un USB bootable con el sistema operativo descargado y luego se ejecutó en el servidor el dispositivo USB e instaló CentOS 7.

Al arrancar el servidor desde el dispositivo USB aparece lo siguiente, ver Figura 4.7.



Figura 4.7: Menú instalación de CentOS 7 (Fuente Propia).

En la ventana se ejecutó la instalación de CentOS 7 y luego se configuró el idioma, teclado y zona horaria

Luego apareció una ventana verificando la instalación mínima de CentOS 7, ver Figura 4.8.

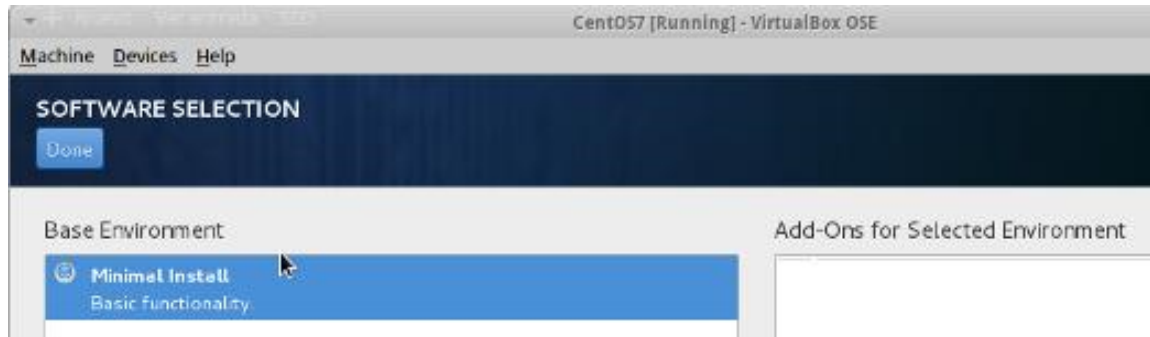


Figura 4.8: Selección instalación mínima, instalación CentOS 7.

El servidor dispone de dos discos duros que en total tienen una capacidad de almacenamiento de 1 TB. Se ocuparon los dos discos duros en la instalación de tal manera de disponer a todo el espacio disponible.

Se estableció la contraseña root y añadió un usuario. En este caso se estableció la contraseña para root y agrego el usuario UCSC, ver Figura 4.9.

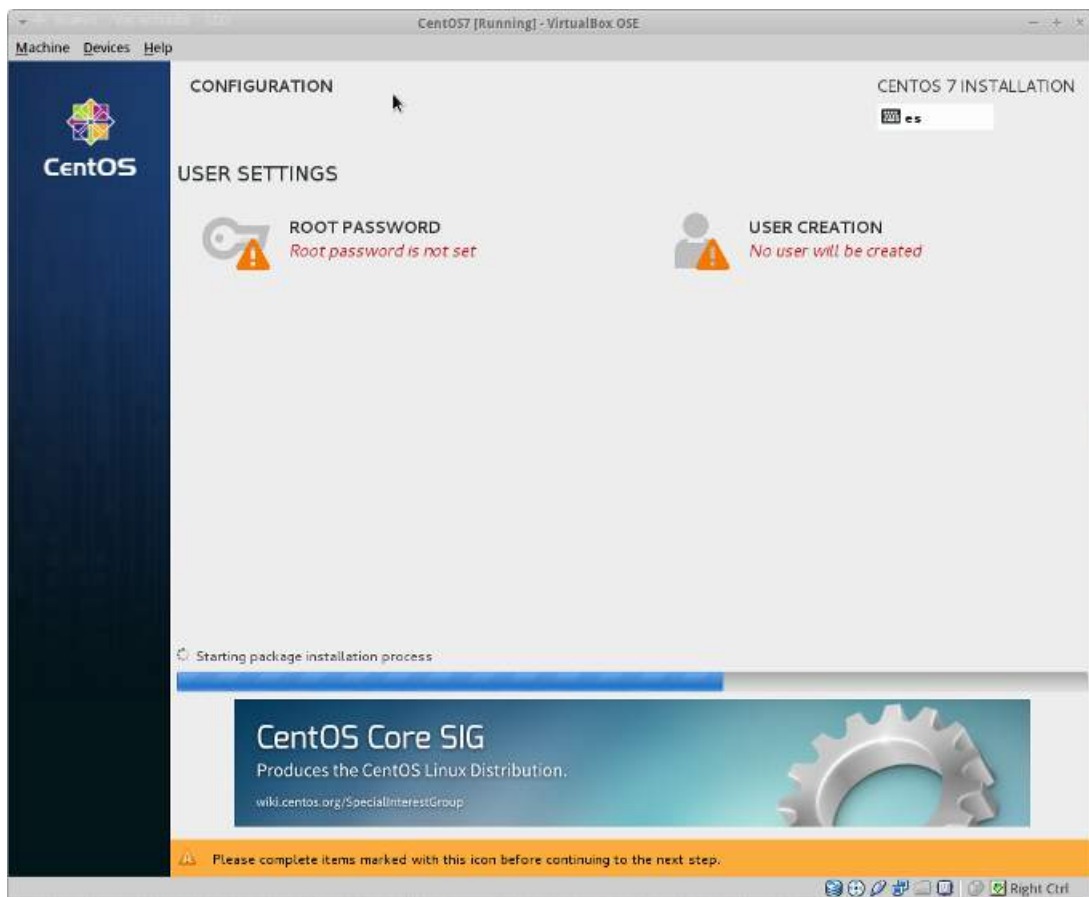


Figura 4.9: Configuración contraseña root y creación de usuarios, instalación CentOS 7 (Fuente Propia).

### 4.3.2. Configuración de la Red

Después de la instalación, se ejecutó el sistema e inicio sesión como root, ver Figura 4.10.

```
CentOS Linux 7 (Core)
Kernel 3.10.0-327.28.3.el7.x86_64 on an x86_64

localhost login: root
Password:
Last login: Thu Oct 27 10:49:07 on tty1
[root@localhost ~]# _
```

Figura 4.10: Logueo modo root en CentOS 7 (Fuente Propia).

#### 4.3.2.1. Archivos de configuración de interfaces

Estos archivos controlan las interfaces de software para dispositivos de red individuales al iniciar el sistema para así determinar que interfaces tiene y su manera de configurar. La forma de nombrar estos archivos generalmente hace referencia al tipo de dispositivo que controla el archivo de configuración de esta forma **ifcfg-name** (Red Hat, 2016b).

#### 4.3.2.2. Interfaces Ethernet

Un archivo de interfaz de red en CentOS 7 se puede encontrar en */etc/sysconfig/network-scripts/ifcfg-eth0*, este archivo controla el primer Ethernet tarjeta de interfaz de red o NIC en el sistema. En sistemas con varias tarjetas de red, existen varios archivos (donde X corresponde a una interfaz). Cada dispositivo tiene su propio archivo de configuración, un administrador puede controlar la forma en que funciona cada interfaz individual editando tal archivo **ifcfg-ethX** (Red Hat, 2016b).

Para configurar la red, se editó el archivo de la interfaz de red. Primero se dirigió al directorio y buscó el archivo que identifica a la interfaz, ver Figura 4.11.

```
CentOS Linux 7 (Core)
Kernel 3.10.0-327.28.3.el7.x86_64 on an x86_64

localhost login: root
Password:
Last login: Thu Oct 27 19:22:52 on tty1
[root@localhost ~]# cd /etc/sysconfig/network-scripts
[root@localhost network-scripts]# ls
ifcfg-eno0      ifdown-post    ifup-eth       ifup-sit
ifcfg-eno1      ifdown-ppp     ifup-ib        ifup-Team
ifcfg-lo        ifdown-routes  ifup-ippp      ifup-TeamPort
ifdown         ifdown-sit     ifup-ipv6      ifup-tunnel
ifdown-bnep     ifdown-Team    ifup-isdn      ifup-wireless
ifdown-eth      ifdown-TeamPort ifup-plip      init.ipv6-global
ifdown-ib       ifdown-tunnel  ifup-plusb     network-functions
ifdown-ippp     ifup           ifup-post      network-functions-ipv6
ifdown-ipv6     ifup-aliases  ifup-ppp
ifdown-isdn     ifup-bnep      ifup-routes
[root@localhost network-scripts]# _
```

Figura 4.11: Interfaz de red *ifcfg-eno0* en CentOS 7 (Fuente Propia).

Después se abrió el archivo *ifcfg-eno0* y se agregaron las líneas de la dirección que contienen el IP, DNS, puerta de enlace y la máscara de red, ver Figura 4.12.

```
TYPE=Ethernet
BOOTPROTO=static
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=enp0s3
UUID=6bcc122f-3f72-476b-bcc9-d194aa749a8a
DEVICE=enp0s3
ONBOOT=yes

IPADDR=192.168.0.8
NETMASK=255.255.255.0
GATEWAY=192.168.0.1
DNS1=200.73.120.16
DNS2=179.60.91.66

"ifcfg-eno0" 22L, 379C
```

Figura 4.12: Configuración de interfaz de red *ifcfg-eno0* en CentOS 7 (Fuente Propia).

Después de editado el archivo se guardó y se reiniciaron los servicios de Internet, ver Figura 4.13.

```
[root@localhost network-scripts]# service network restart
Restarting network (via systemctl): [ OK ]
[root@localhost network-scripts]# _
```

Figura 4.13: Reiniciar red en CentOS 7 (Fuente Propia).

Así se finalizo la configuración de la tarjeta de red y se pudo tener acceso a Internet por parte del servidor.

### 4.3.3. Instalar modo gráfico en Centos 7

Al instalar CentOS 7 en su versión mínima para servidor, el modo gráfico predeterminado es la consola de comandos, si se quiere instalar modo gráfico en CentOS 7 se deberá descargar e instalar, en este caso se utilizó Gnome GUI (IT'zGeek, 2016).

Para descargar Gnome GUI para CentOS 7 se escribió lo siguiente en la consola de comandos ver Figura 4.14.

```
CentOS Linux 7 (Core)
Kernel 3.10.0-327.28.3.el7.x86_64 on an x86_64

localhost login: root
Password:
Last login: Fri Oct 28 16:23:45 on tty1
[root@localhost ~]# yum groupinstall "GNOME Desktop" "Graphical Administration Tools" _
```

Figura 4.14: Descargar modo gráfico GUI para CentOS 7 (Fuente Propia).

Después de la instalación se debió habilitar la interfaz de usuario al arrancar el sistema. Para habilitarla se debió ejecutar el siguiente comando ver Figura 4.15.

```
CentOS Linux 7 (Core)
Kernel 3.10.0-327.28.3.el7.x86_64 on an x86_64

localhost login: root
Password:
Last login: Fri Oct 28 16:23:45 on tty1
[root@localhost ~]# ln -sf /lib/systemd/system/runlevel15.target /etc/systemd/system/default.target_
```

Figura 4.15: Selección target modo gráfico para CentOS 7 (Fuente Propia).

Al arrancar el equipo se vio el escritorio ver Figura 4.16.

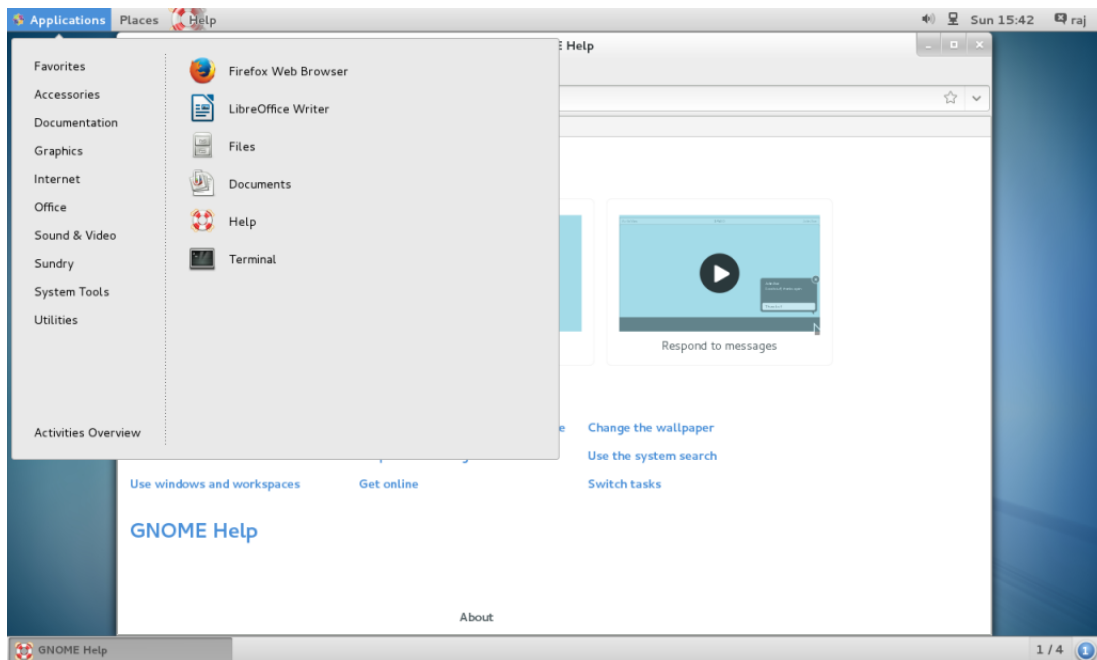


Figura 4.16: Modo gráfico GUI en CentOS 7 (Fuente Propia).

#### 4.3.4. Instalación Kernel Virtual Machine + QEMU

Para la instalación del hipervisor en el servidor se utilizó una referencia bibliográfica *Red Hat Enterprise Linux 7 Virtualization Deployment and Administration Guide*, dicha guía cubre cómo configurar una máquina física host de Red Hat Enterprise Linux 7. La guía fue de utilidad ya que el anfitrión CentOS 7 es un derivado de las fuentes de Red Hat Enterprise Linux (RHEL). La guía cubre cómo instalar y configurar máquinas virtuales invitadas usando el hipervisor KVM. Incluye la configuración de dispositivos PCI, redes, almacenamiento, administración de dispositivos y máquinas virtuales invitadas (Laura Novich, 2015).

Para utilizar la virtualización con CentOS 7 lo mínimo que se requiere es `qemu-kvm`, `qemu-img` y además los paquetes de `libvirt`. Estos paquetes proporcionan el emulador KVM a nivel de usuario, gestor de imágenes de disco, y herramientas de gestión de la virtualización en el sistema de CentOS 7.

El paquete `libvirt` proporciona las bibliotecas de servidor por lado del anfitrión para interactuar con el hipervisor y el sistema de host y el demonio `libvirtd` se encarga de las llamadas a las bibliotecas, gestiona las máquinas virtuales y controla el hipervisor.

Para instalar estos paquetes QEMU, KVM y `libvirt`, se ejecuto el siguiente comando, ver Figura.

```
# yum install qemu-kvm qemu-img libvirt
```

Hay paquetes adicionales de gestión para la virtualización disponibles que se recomiendan cuando se utiliza la virtualización.

#### 4.3.5. Virt-install

Este paquete proporciona **`virt-install`** para la creación de máquinas virtuales en la línea de comando.

##### 4.3.5.1. Creación de invitados con `virt-install`

Con el comando **`virt-install`** se pueden crear máquinas virtuales desde la línea de comandos. Para usarlo se necesitan privilegios de root. La herramienta `virt-install` ofre-

ce una serie de opciones que se pueden ejecutar en la línea de comandos.

Los requisitos mínimos para crear un invitado en línea de comando son **-name**, **-ram**, almacenamiento de invitados (**-disk**, **-filesystem** o **-nodisks**), y una opción de instalación

Para ver una lista completa de opciones, ejecute el siguiente comando.

```
# virt-install -help
```

En el siguiente ejemplo se creara un invitado de Red Hat Enterprise Linux 6.

```
# virt-install
-name=guest1-rhe16-64
-disk path=/var/lib/libvirt/images/guest1-rhe16-
64.dsk,size=8,sparse=false,cache=none
-graphics spice
-vcpus=2 -ram=2048
-location=http://example1.com/installation_tree/RHEL6.4-Server-x86_64/os
-network bridge=br0
-os-type=linux
-os-variant=rhe16
```

Las opciones utilizadas en este ejemplo son las siguientes:

- **-name**  
El nombre de la máquina virtual.
- **-disk**  
Especifica los detalles de configuración de almacenamiento para la máquina virtual.
- **-graphics**  
Especifica el tipo de herramienta gráfica que se utilizará para la instalación de una máquina virtual.
- **-vcpus**  
Número de vCPUs que se asignarán al invitado

- **-ram**  
La cantidad de memoria (RAM) a asignar al huésped, en MB.
- **-location**  
La ubicación del medio de instalación.
- **-network bridge**  
Especifica el puente de red que se utilizará para la instalación, debe configurarse antes ejecutando virt-install.
- **-os-type**  
El tipo de sistema operativo invitado.
- **-os-variant**  
Otra opción importante en virt-install, utilizada para optimizar aún más la configuración de invitados. El uso de esta opción puede reducir el tiempo de instalación y mejorar el rendimiento.

### 4.3.6. Libvirt-python

El paquete libvirt-python contiene un módulo que permite a las aplicaciones escritas en Python lenguaje de programación, utilizar la interfaz proporcionada por la API libvirt.

### 4.3.7. Virt-manager

Virtual Machine Manager, virt-manager, proporciona una herramienta gráfica para administrar máquinas virtuales. Utiliza la biblioteca libvirt-client como la API de administración.

#### 4.3.7.1. Instalación de una máquina virtual con virt-manager

El asistente virt-manager desglosa el proceso de creación de la máquina virtual en cinco pasos:

1. Elegir el hipervisor y el tipo de instalación.
2. Ubicación y configuración de los medios de instalación.

3. Configuración de las opciones de memoria y CPU.
4. Configuración del almacenamiento de la máquina virtual.
5. Configuración del nombre de la máquina virtual, la configuración de red, la arquitectura y otros ajustes de hardware.

En este ejemplo se creará una máquina virtual invitada de Red Hat Enterprise Linux 7 con una imagen de DVD o de DVD de instalación almacenada localmente.

Red Hat Enterprise Linux 7 requiere al menos 1 GB de espacio de almacenamiento. Sin embargo, Red Hat recomienda al menos 5 GB de espacio de almacenamiento para una instalación de Red Hat Enterprise Linux 7.

1. **Abra virt-manager e inicie el asistente.**

Abra virt-manager ejecutando el comando `virt-manager` como root o dirigiéndose a *Aplicaciones -> Herramientas del sistema -> Virtual Machine Manager*, ver Figura 4.17.

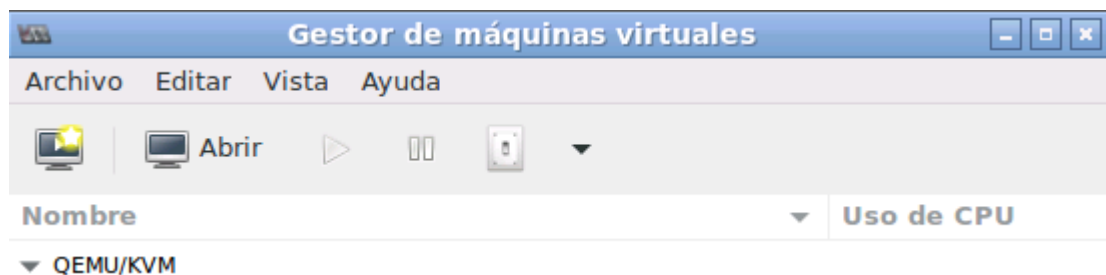


Figura 4.17: Aplicación Virt Manager (Fuente Propia).

## 2. Seleccione el botón crear una nueva máquina virtual.

Ver Figura 4.18.

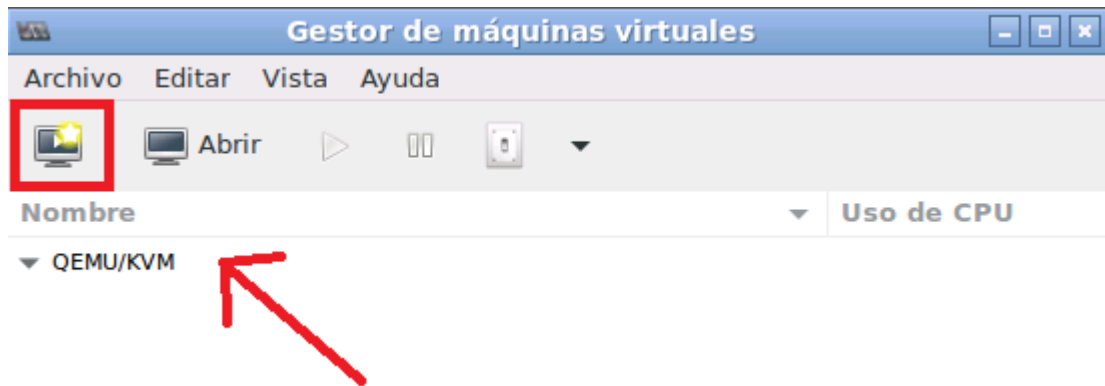


Figura 4.18: Botón para crear máquina virtual con Virt Manager (Fuente Propia).

## 3. Especificar el tipo de instalación.

- Medio de instalación local: Este método utiliza un CD-ROM, DVD o una imagen de un disco de instalación.
- Instalación por red: Este método implica el uso de Red Hat Enterprise Linux o Fedora. El árbol de instalación debe ser accesible a través de HTTP, FTP o NFS.
- Arranque por red: Este método utiliza un servidor PXE (Preboot eXecution Environment) para instalar la máquina virtual invitada. Para instalar a través de arranque de red, el invitado debe tener una dirección IP enrutable o compartir dispositivo de red. Una vez que se hayan completado todos los pasos, se enviará una solicitud DHCP y si se encuentra un servidor PXE válido, se iniciarán los procesos de instalación.
- Importar imagen de disco existente: Este método permite crear una nueva máquina virtual invitada e importar un disco (Contiene un sistema operativo pre-instalado y de arranque).

Opciones del tipo de instalación, ver Figura 4.19.

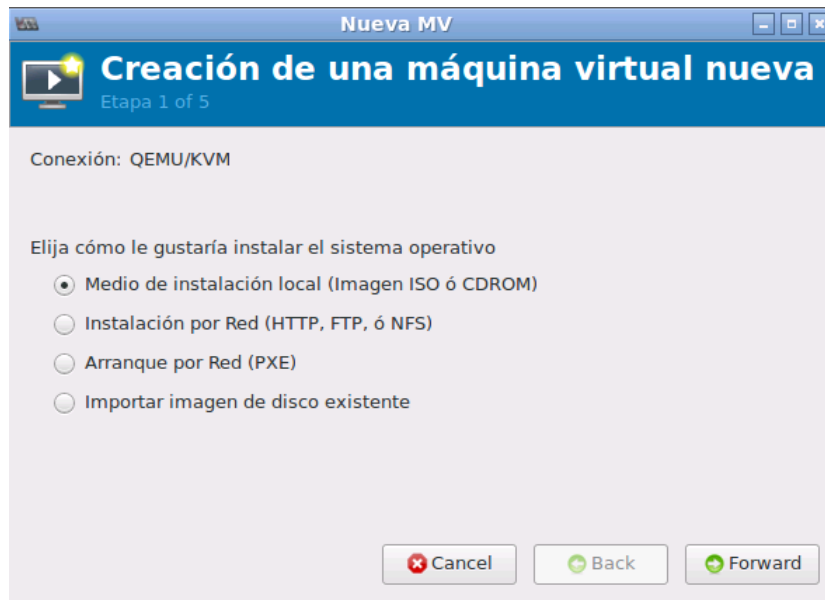


Figura 4.19: Selección tipo de instalación Virt Manager (Fuente Propia).

#### 4. Seleccione el medio de instalación local.

Si ha seleccionado medio de instalación local (imagen ISO o CDROM), especifique el medio de instalación local deseado, ver Figura 4.20.

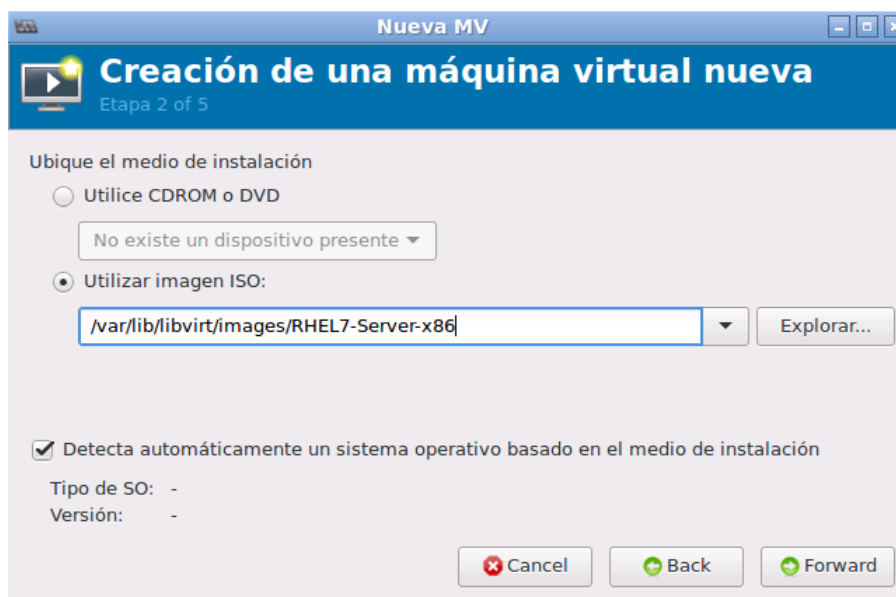


Figura 4.20: Selección tipo de imagen ISO Virt Manager (Fuente Propia).

- a) Se desea instalar desde un CD-ROM o DVD, seleccione el botón de opción *Utilice CD-ROM o DVD* y seleccione la unidad de disco adecuada en la lista desplegable.
- b) Se desea instalar desde una imagen ISO, seleccione *Utilizar imagen ISO* y haga clic en *Explorar*, para abrir la ventana de volumen de selección de medios. Después seleccione la imagen de instalación que desea utilizar y haga clic en *Elegir volumen*. Si no se muestran imágenes en la ventana de selección de medios, haga clic en el botón *Explorar localmente* para explorar la máquina host en busca de la imagen de instalación o la unidad DVD que contiene el disco de instalación. Una vez encontrada deberá seleccionar la imagen de instalación o la unidad DVD y proceder a hacer clic en *Open*.

Para los archivos de imagen ISO y las imágenes de almacenamiento de invitados, la ubicación recomendada para utilizar es `/var/lib/libvirt/images/`.

A continuación, deberá configurar el tipo de sistema operativo y la versión de la instalación. Asegúrese de que ha seleccionado el tipo de sistema operativo adecuado para su máquina virtual. Esto lo puede especificar manualmente o seleccionando la casilla de verificación automática.

#### 5. Para la configuración de la memoria RAM y las CPU virtuales.

Se debe especificar el número de CPUs y la cantidad de memoria RAM que se asignarán a la máquina virtual. Estos valores afectan el rendimiento del anfitrión y del huésped.

Las máquinas virtuales requieren suficiente memoria RAM para funcionar de manera eficiente y eficaz. Red Hat admite un mínimo de 512 MB de RAM para una máquina virtual. También recomienda al menos 1024MB de RAM para cada núcleo lógico.

Se debe asignar suficientes CPU virtuales para la máquina virtual. Si la máquina virtual ejecuta un multithreaded aplicación, asigne el número de CPUs virtuales que la máquina virtual invitada requerirá para funcionar eficientemente.

No puede asignar más CPU virtuales que procesadores físicos disponible en el sistema host. El número de CPUs virtuales disponibles se anota en el campo *CPUs* hasta X disponible.

Puede ver la asignación de la memoria RAM y el numero de CPU para una máquina virtual en la Figura 4.21.

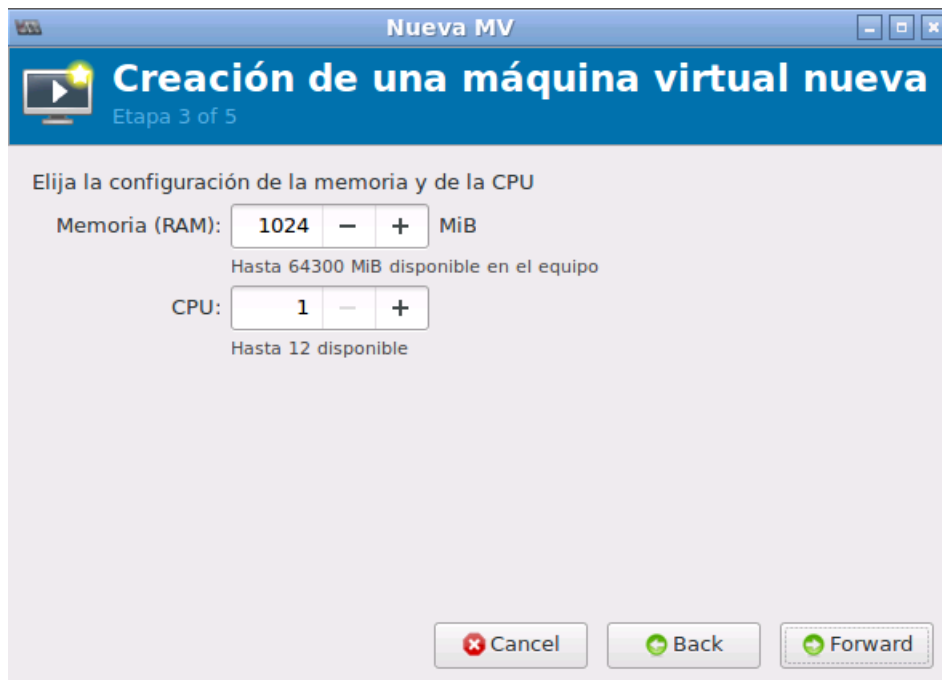


Figura 4.21: Asignación de memoria RAM y cantidad de CPU con Virt Manager (Fuente Propia).

## 6. Configuración de almacenamiento.

Aquí se debe habilitar y asignar espacio de almacenamiento suficiente para la máquina virtual y las aplicaciones que requiera, ver Figura 4.22.

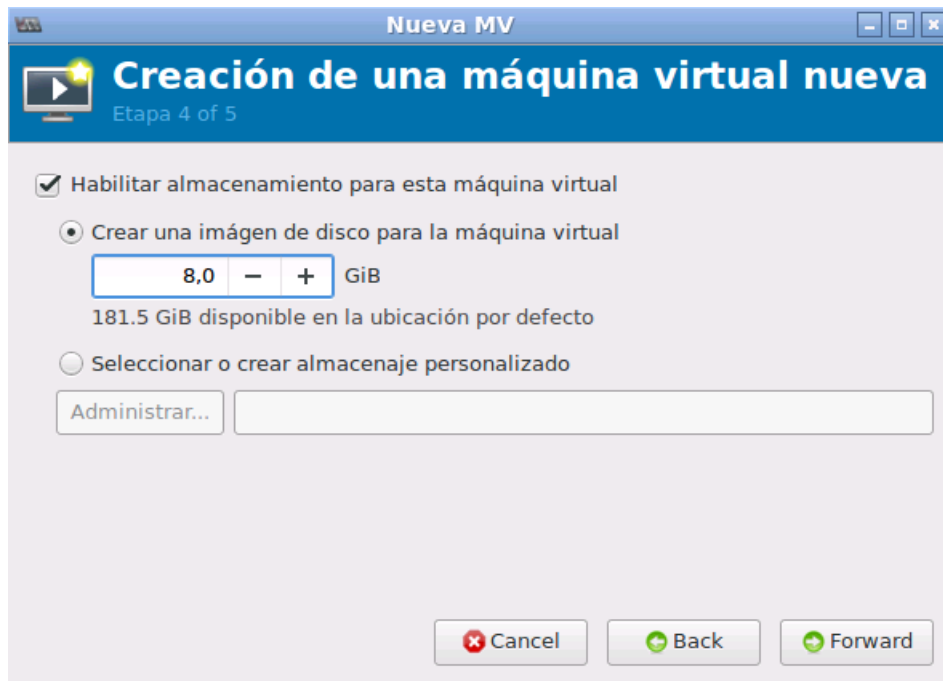


Figura 4.22: Asignación de almacenamiento con Virt Manager (Fuente Propia).

*a)* **Con el almacenamiento local predeterminado.**

Seleccione la opción crear una imagen de disco en el disco duro de la computadora para crear una imagen basada en archivos en el grupo de almacenamiento predeterminado, `/var/lib/libvirt/images/` directorio. Introduzca el tamaño de la imagen del disco que desea crear.

*b)* **Con un grupo de almacenamiento.**

Acá podrá seleccionar o crear un nuevo grupo de almacenamiento, ver Figura 4.23.

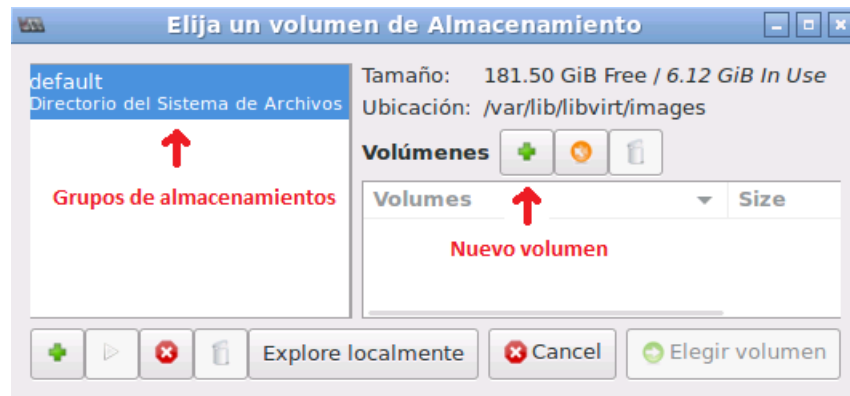


Figura 4.23: Asignación de grupo de almacenamiento Virt Manager (Fuente Propia).

- 1) Seleccione un grupo de almacenamiento en la lista grupos de almacenamiento ver Figura 4.23.
- 2) Haga clic en el botón *Nuevo volumen* para crear un nuevo volumen de almacenamiento, ver Figura 4.23. Aparecerá la pantalla añadir un volumen de almacenamiento, ver Figura 4.24.

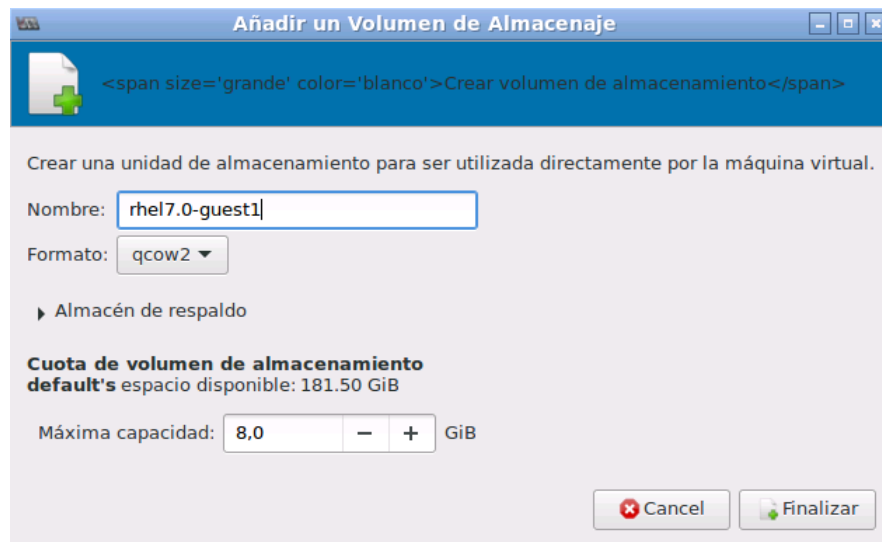


Figura 4.24: Asignación de nombre de almacenamiento Virt Manager (Fuente Propia).

## 7. Nombre y configuración final.

Deberá asignar un nombre a la máquina virtual y finalizar la instalación, ver Figura 4.25.

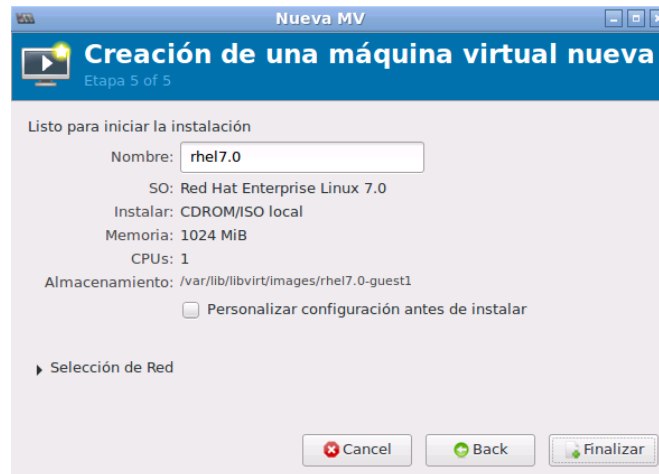


Figura 4.25: Asignación de nombre a máquina virtual Virt Manager (Fuente Propia).

### 4.3.8. Libvirt-client

El paquete libvirt-client proporciona las API del lado del cliente y bibliotecas para acceder a libvirt en servidores. El paquete libvirt-client incluye la herramienta de línea de comandos de virsh para gestionar y controlar las máquinas virtuales e hipervisores desde la línea de comandos o una virtualización.

Para instalar todos estos paquetes de virtualización recomendados se ejecuto siguiente comando.

```
# yum install virt-install libvirt-python virt-manager python-virtinst
libvirt-client
```

Los paquetes de virtualización también pueden instalarse desde grupos de paquetes. En la siguiente tabla se describen los grupos de paquetes de virtualización y lo que ofrecen, ver Figura 4.26.

Package Group	Description	Mandatory Packages	Optional Packages
<b>Virtualization Hypervisor</b>	Smallest possible virtualization host installation	libvirt, qemu-kvm	qemu-kvm-tools
<b>Virtualization Client</b>	Clients for installing and managing virtualization instances	gnome-boxes, virt-install, virt-manager, virt-viewer	virt-top, libguestfs-tools, libguestfs-tools-c
<b>Virtualization Platform</b>	Provides an interface for accessing and controlling virtual machines and containers	libvirt, libvirt-client, virt-who	fence-virt-libvirt, fence-virt-multicast, fence-virt-serial, libvirt-cim, libvirt-java, libvirt-snmp, perl-Sys-Virt
<b>Virtualization Tools</b>	Tools for offline virtual image management	libguestfs	libguestfs-java, libguestfs-tools, libguestfs-tools-c

Figura 4.26: Grupos de paquetes de virtualización (Laura Novich, 2015) (Fuente Propia).

Para instalar un grupo de paquetes, debe ejecutar en la línea de comandos, `yum groupinstall <grupo de paquetes>`. Por ejemplo, se instaló el grupo de paquetes *Virtualización Tools*.

```
# yum groupinstall "Virtualization Tools"
```

## 4.4. Configuración el entorno virtual

Una vez instalados los diversos componentes del nuevo entorno, se debió configurar para preparar las máquinas virtuales. Los pasos más importantes implican la configuración de la red virtual, almacenamiento y recursos.

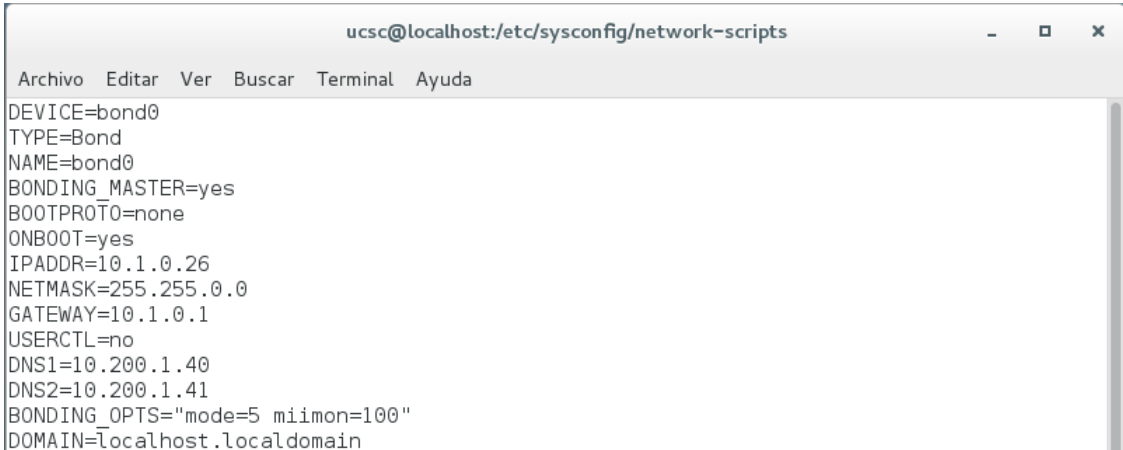
### 4.4.1. Configuración de la red

Se pueden vincular múltiples interfaces de red en un solo canal usando el bonding-módulo del núcleo y una interfaz de red especial llamado interfaz de unión de canales. Esta permite que dos o más interfaces de red actuar como uno, aumentando al mismo tiempo el ancho de banda (Red Hat, 2016c).

#### 4.4.1.1. Creación de un bonding

Para crear una interfaz bonding, debe crear un archivo en el directorio `/etc/sysconfig/network-scripts/` denominado `ifcfg-bondN`, reemplazando N por el número de la interfaz, como 0. El contenido del archivo puede ser idéntico a cualquier tipo de interfaz que se está uniendo, como una interfaz Ethernet. La única diferencia es que la directiva `DEVICE` es `bondN`, reemplazando N por el número de la interfaz. La directiva `NM_CONTROLLED` se puede agregar para evitar que NetworkManager configure este dispositivo.

Se configuró la interfaz `ifcfg-bond0` para el servidor, ver Figura 4.27.



```

ucsc@localhost:/etc/sysconfig/network-scripts
Archivo Editar Ver Buscar Terminal Ayuda
DEVICE=bond0
TYPE=Bond
NAME=bond0
BONDING_MASTER=yes
BOOTPROTO=none
ONBOOT=yes
IPADDR=10.1.0.26
NETMASK=255.255.0.0
GATEWAY=10.1.0.1
USERCTL=no
DNS1=10.200.1.40
DNS2=10.200.1.41
BONDING_OPTS="mode=5 miimon=100"
DOMAIN=localhost.localdomain

```

Figura 4.27: Creación interfaz bond en línea de comandos en CentOS 7 (Fuente Propia).

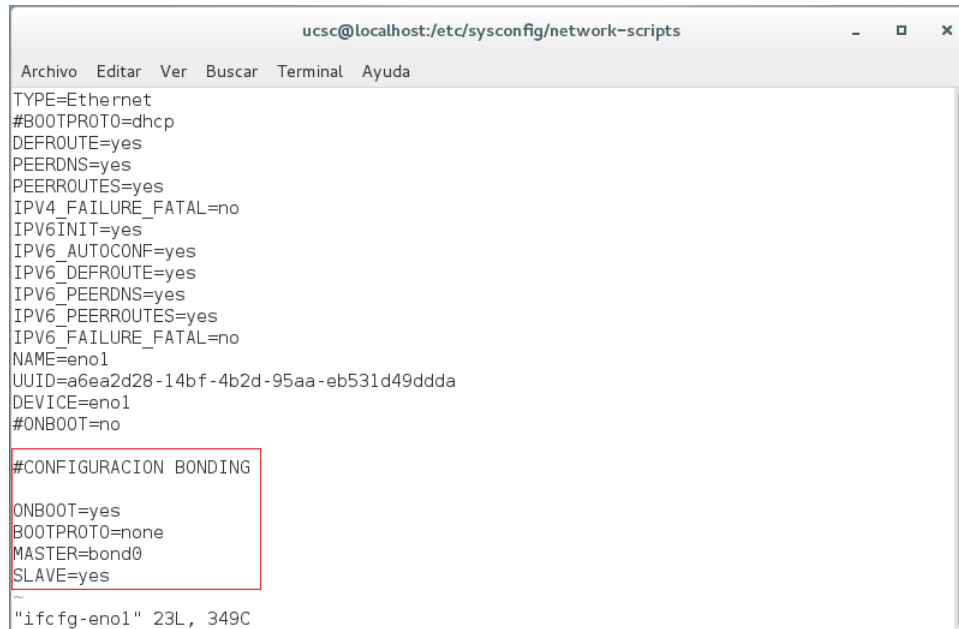
La dirección MAC del enlace se tomará de la primera interfaz a ser esclavizada. También se puede especificar utilizando la directiva `HWADDR` si es necesario. Si desea que NetworkManager controle esta interfaz, elimine la directiva `NM_CONTROLLED = NO`, o establezca en `yes`, y agregue `TYPE = Bond` y `BONDING_MASTER = YES`.

Una vez creada la interfaz, las interfaces de red a enlazar deben configurarse agregando las directivas `MASTER` y `SLAVE` a sus archivos de configuración. Los archivos de configuración para cada una de las interfaces son casi idénticos (Red Hat, 2016c).

Recordar que el servidor dispone de dos interfaces de red.

1. **ifcfg-eno1.**
2. **ifcfg-eno2.**

1. Configuración creada para interfaz **ifcfg-eno1** ver Figura 4.28.



```
ucsc@localhost:/etc/sysconfig/network-scripts
Archivo Editar Ver Buscar Terminal Ayuda
TYPE=Ethernet
#BOOTPROTO=dhcp
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=eno1
UUID=a6ea2d28-14bf-4b2d-95aa-eb531d49ddda
DEVICE=eno1
#ONBOOT=no

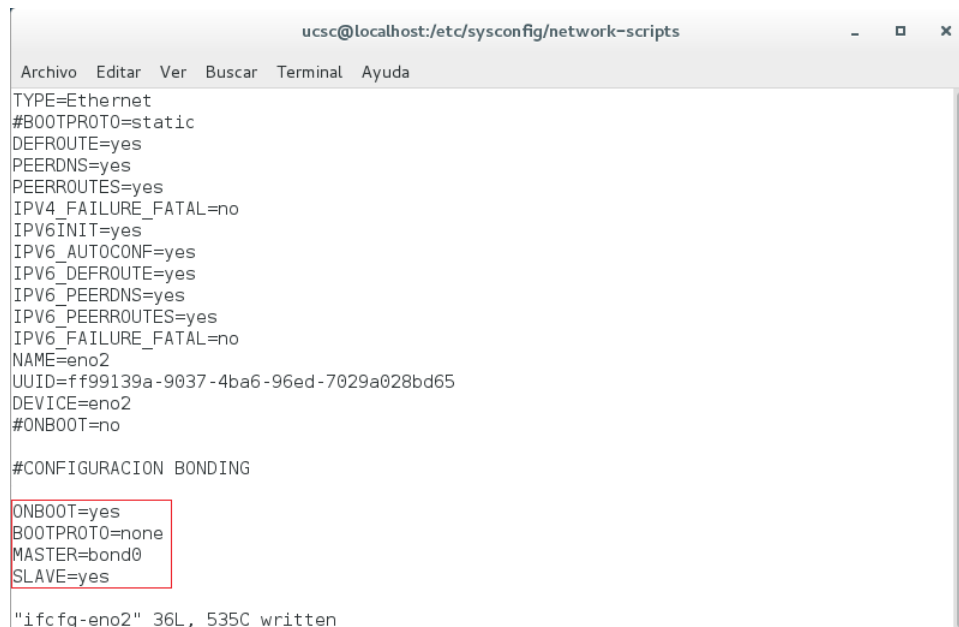
#CONFIGURACION BONDING

ONBOOT=yes
BOOTPROTO=none
MASTER=bond0
SLAVE=yes

~
"ifcfg-eno1" 23L, 349C
```

Figura 4.28: Configuración interfaz ifcfg-eno1 modo comando en CentOS 7 (Fuente Propia).

2. Configuración creada para interfaz **ifcfg-eno2** ver Figura 4.29.



```
ucsc@localhost:/etc/sysconfig/network-scripts
Archivo Editar Ver Buscar Terminal Ayuda
TYPE=Ethernet
#BOOTPROTO=static
DEFROUTE=yes
PEERDNS=yes
PEERROUTES=yes
IPV4_FAILURE_FATAL=no
IPV6INIT=yes
IPV6_AUTOCONF=yes
IPV6_DEFROUTE=yes
IPV6_PEERDNS=yes
IPV6_PEERROUTES=yes
IPV6_FAILURE_FATAL=no
NAME=eno2
UUID=ff99139a-9037-4ba6-96ed-7029a028bd65
DEVICE=eno2
#ONBOOT=no

#CONFIGURACION BONDING

ONBOOT=yes
BOOTPROTO=none
MASTER=bond0
SLAVE=yes

~
"ifcfg-eno2" 36L, 535C written
```

Figura 4.29: Configuración interfaz ifcfg-eno2 modo comando en CentOS 7 (Fuente Propia).

Una vez que las interfaces fueron configuradas, se reinició el servicio de red como root, con el siguiente comando.

```
# service network restart
```

#### 4.4.1.2. Traducción de Direcciones de Red (Red NAT)

De forma predeterminada, los conmutadores de red virtual funcionan en modo NAT. Usan IP enmascarada que permite a los huéspedes conectados utilizar la dirección IP de la máquina física anfitriona para comunicarse con cualquier red externa. De forma predeterminada, los equipos que se colocan externamente a la máquina física de host, no pueden comunicarse con los invitados cuando el conmutador de red virtual está operando en modo NAT, como se muestra en el siguiente diagrama, ver Figura 4.30.

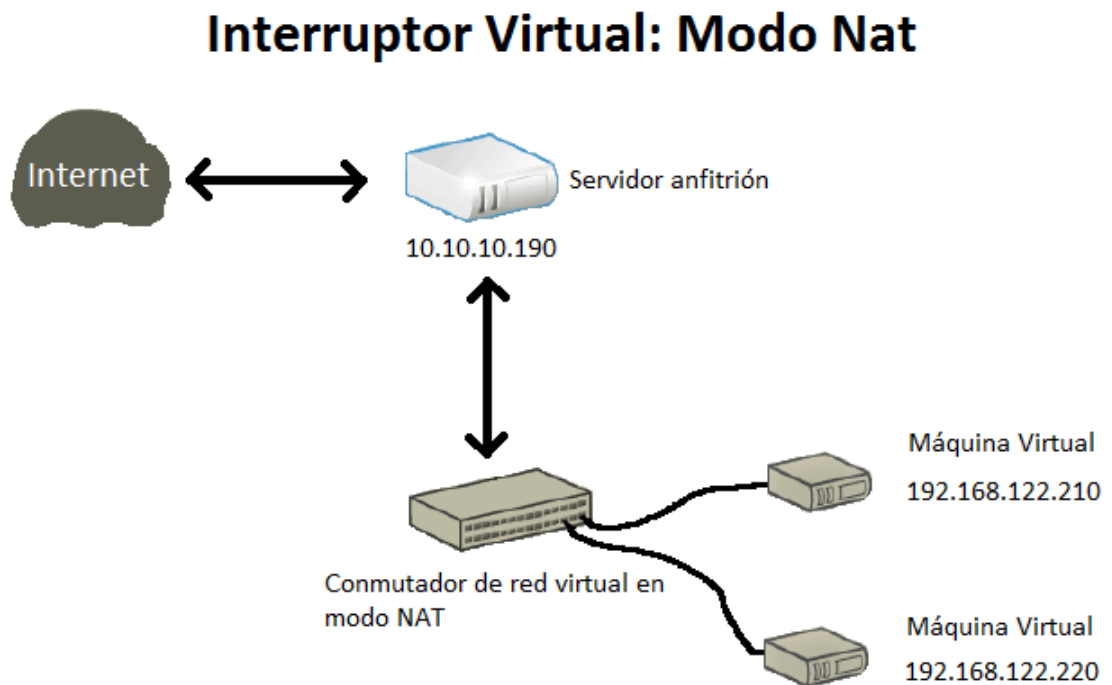


Figura 4.30: Interruptor de red virtual utilizando NAT con dos invitados (Fuente Propia).

#### 4.4.1.3. Gestión de una red virtual NAT

1. En el menú *Editar*, se seleccionó *Detalles de la conexión*, ver Figura 4.31.

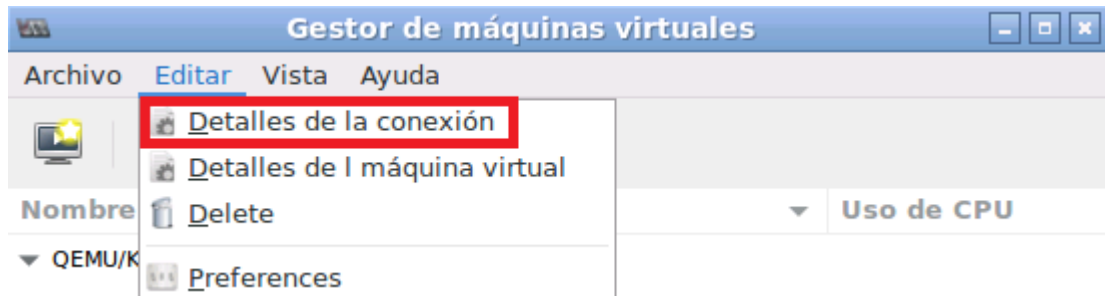


Figura 4.31: Selección de los detalles de conexión (Fuente Propia).

2. Se abrió el menú *Detalles de conexión*. Se procedió a hacer clic en la pestaña *Redes virtuales*, ver Figura 4.32.

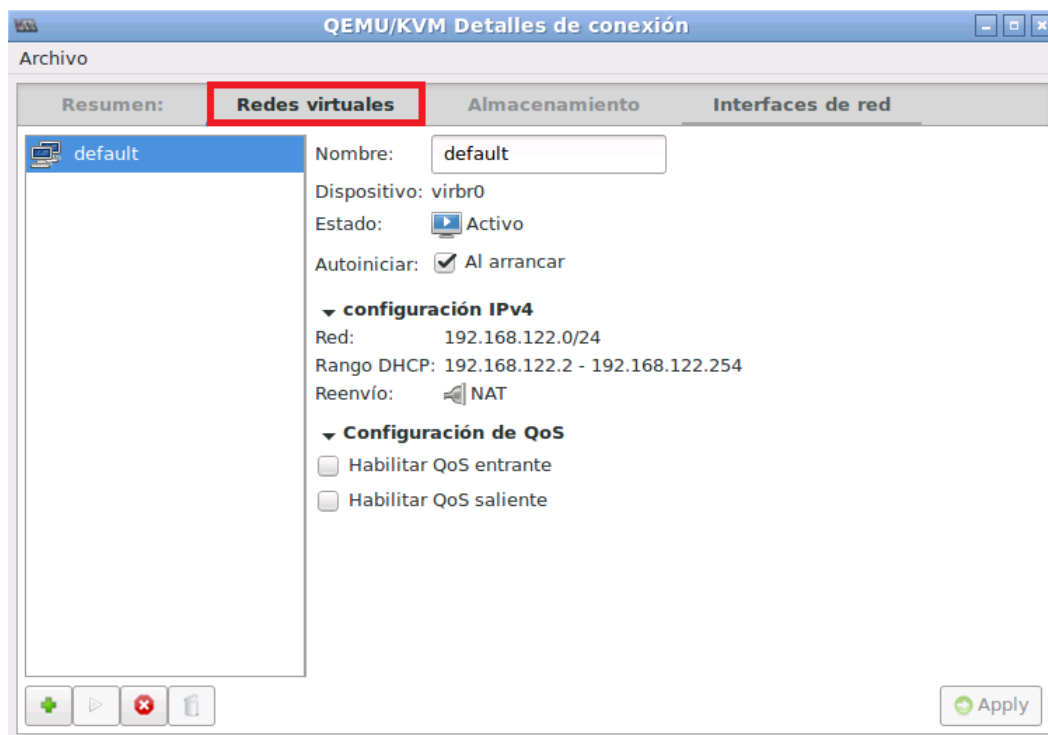


Figura 4.32: Configuración de red virtual (Fuente Propia).

3. Todas las redes virtuales disponibles se enumeran en la caja de la izquierda del menú. Puede editar la configuración de una red virtual seleccionándola en este cuadro y editándola como mejor le parezca.

#### 4.4.1.4. Creación de una red virtual NAT con virt-manager

1. Se abrió la pestaña *Redes virtuales* desde el menú Detalles de la conexión. Luego se hizo clic en el botón *Añadir red*, identificado con un signo más (+), ver Figura 4.33.



Figura 4.33: Configuración de red virtual (Fuente Propia).

2. Se introdujo un nombre para la red virtual y luego se hizo clic en *Forward*, ver Figura 4.34.

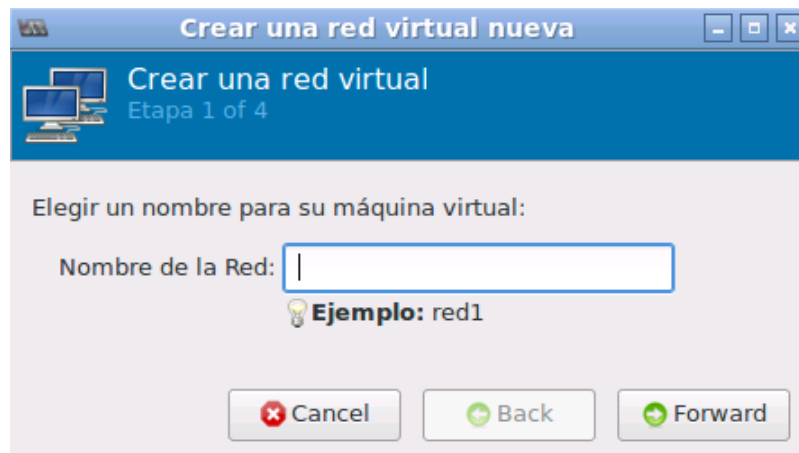


Figura 4.34: Nombrar su red virtual (Fuente Propia).

3. Se introdujo un espacio de direcciones IPv4 para su red virtual y definió el rango DHCP especificando un intervalo de inicio y fin de direcciones IP, luego se hizo clic en *Forward*, ver Figura 4.35.

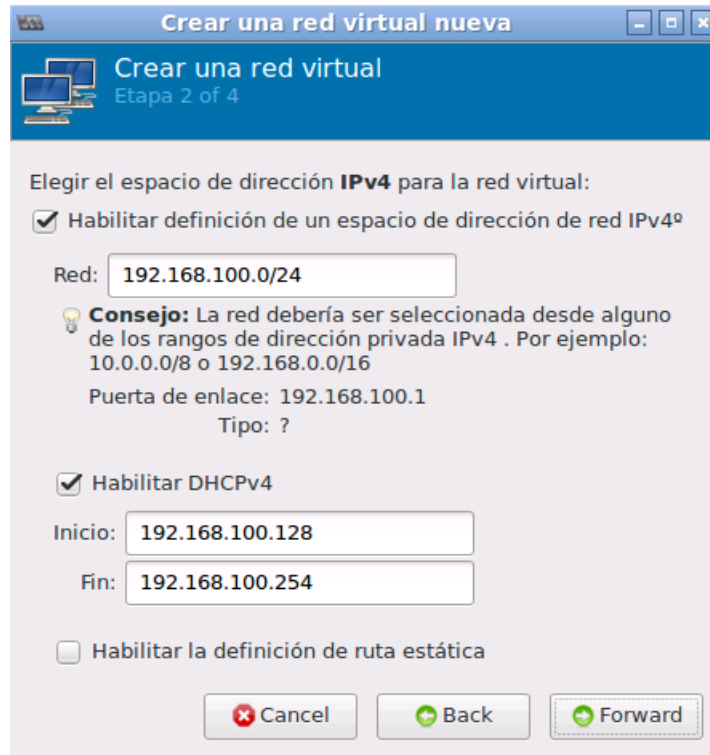


Figura 4.35: Elegir un espacio de direcciones IPv4 (Fuente Propia).

4. También si lo desea, puede habilitar un espacio de direcciones IPv6 para su red virtual y definir el rango DHCP especificando un intervalo de inicio y fin de direcciones IP, luego hacer clic en *Forward*, ver Figura 4.36.

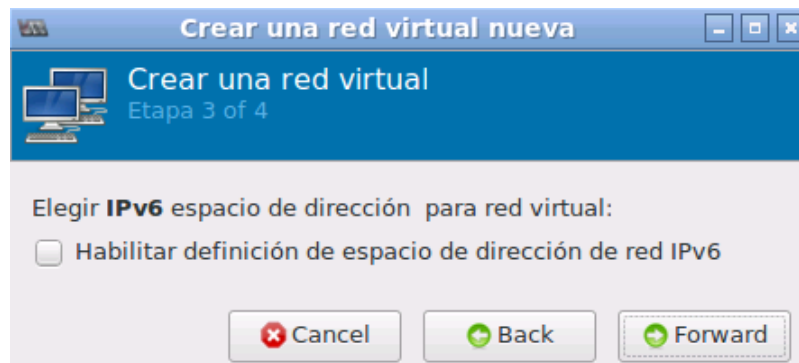


Figura 4.36: Elegir un espacio de direcciones IPv6 (Fuente Propia).

5. Deberá seleccionar cómo se conectara la red virtual a la red física, ver Figura 4.37.

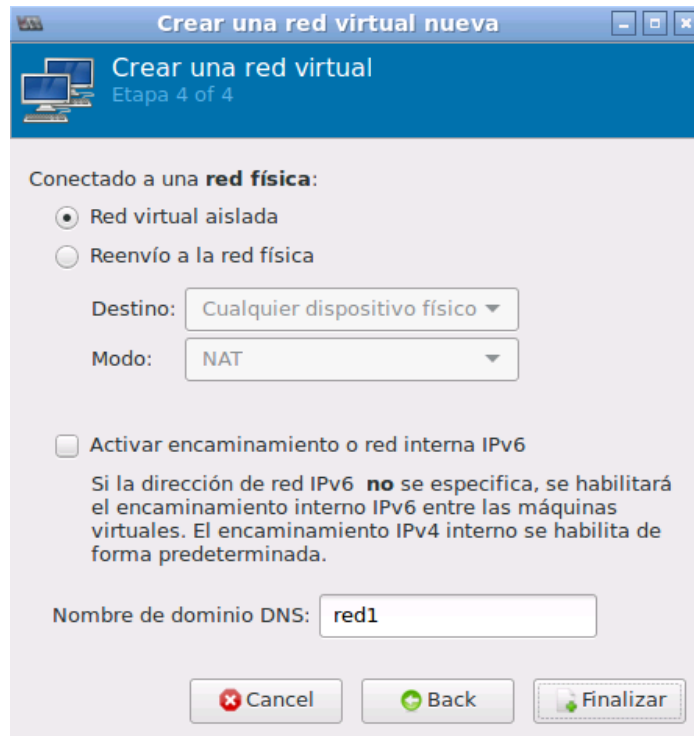


Figura 4.37: Conexión a una red física (Fuente Propia).

Si selecciono *Reenvió a red física*, deberá elegir si el destino debe ser cualquier dispositivo físico o un dispositivo físico específico. También deberá seleccionar si el modo será NAT o enrutado.

6. Ahora estará listo para la creación de la red. Para comprobar la configuración de la red, se hizo clic en *Finalizar*, ver Figura 4.38.

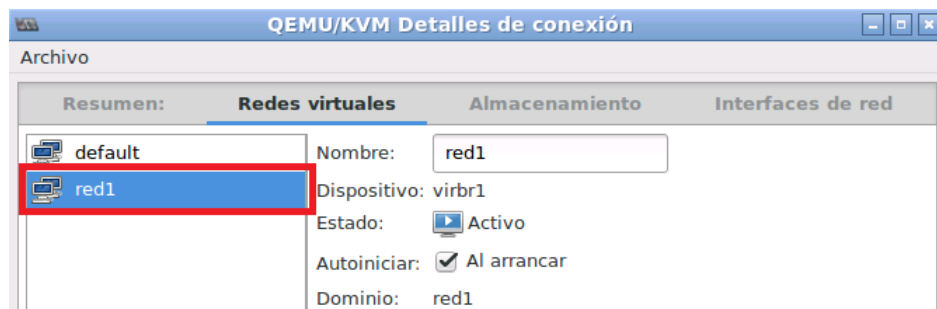


Figura 4.38: Red 1 creada (Fuente Propia).

La nueva red virtual ya está disponible en la pestaña *Redes virtuales* de la ventana.

#### 4.4.1.5. Conexión de una red virtual a un invitado

1. En la ventana de Virtual Machine Manager, deberá seleccionar el invitado al cual le asignara la red, ver Figura 4.39.

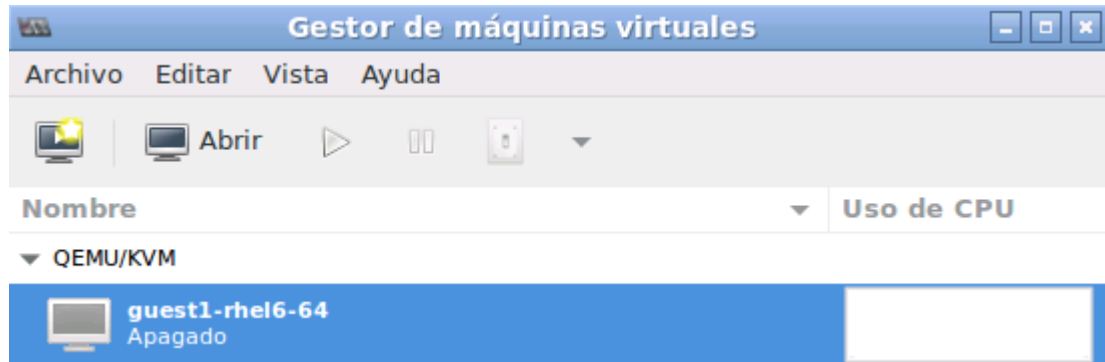


Figura 4.39: Selección de una máquina virtual (Fuente Propia).

2. En el menú *Editar* de Virtual Machine Manager, seleccione *Detalles de la máquina virtual*, ver Figura 4.40.

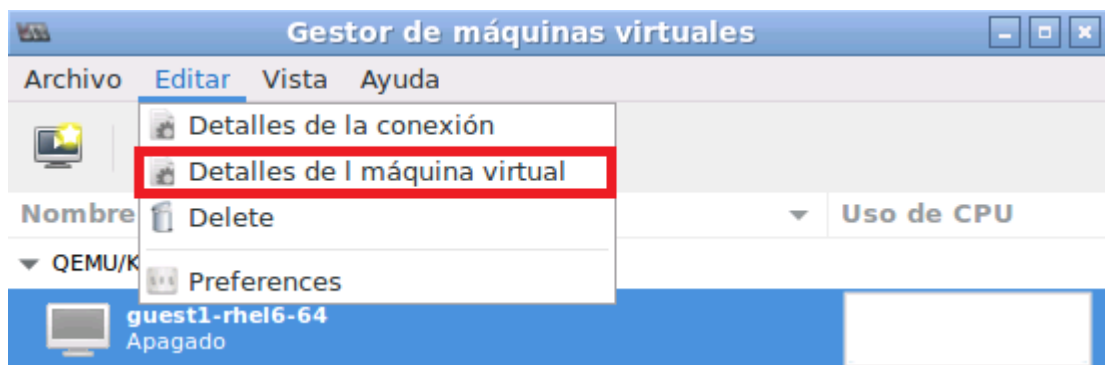


Figura 4.40: Visualización de los detalles de la máquina virtual (Fuente Propia).

3. Deberá dirigirse a *Vista->Detalles*. Después en la ventana emergente deberá hacer clic en el botón *Agregar hardware*, ver Figura 4.41.

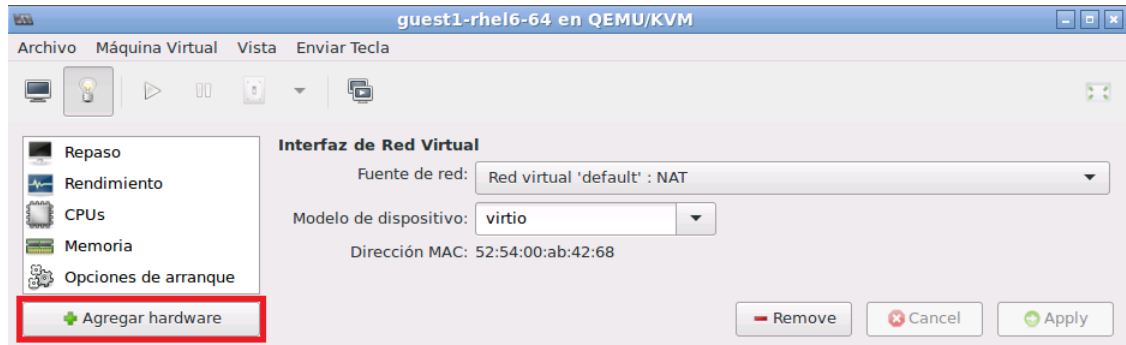


Figura 4.41: La ventana Detalles de la máquina virtual.

4. En la ventana Agregar nuevo hardware virtual, deberá seleccionar *Network* en el panel izquierdo y luego en *Fuente de la red* seleccionar el nombre de la red (red1 en este caso) luego hacer clic en *Finalizar*, ver Figura 4.42.

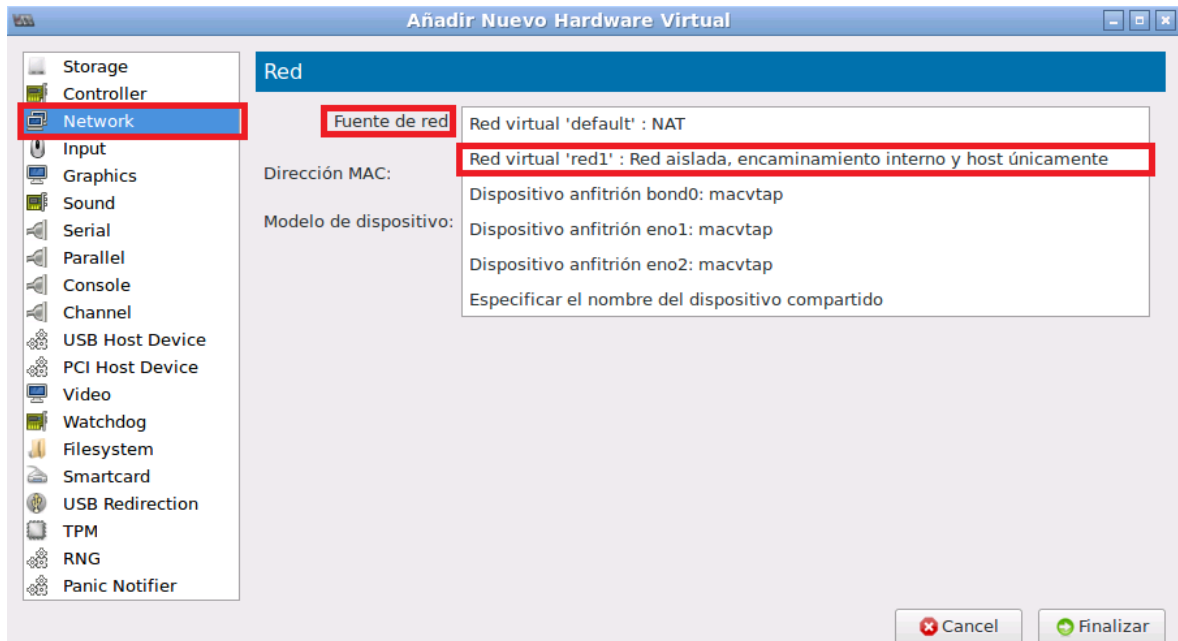


Figura 4.42: Selección de red en la ventana Añadir Nuevo Hardware Virtual (Fuente Propia).

5. La nueva red se muestra ahora como una interfaz de red virtual que se presentará al invitado al iniciar, ver Figura 4.43.

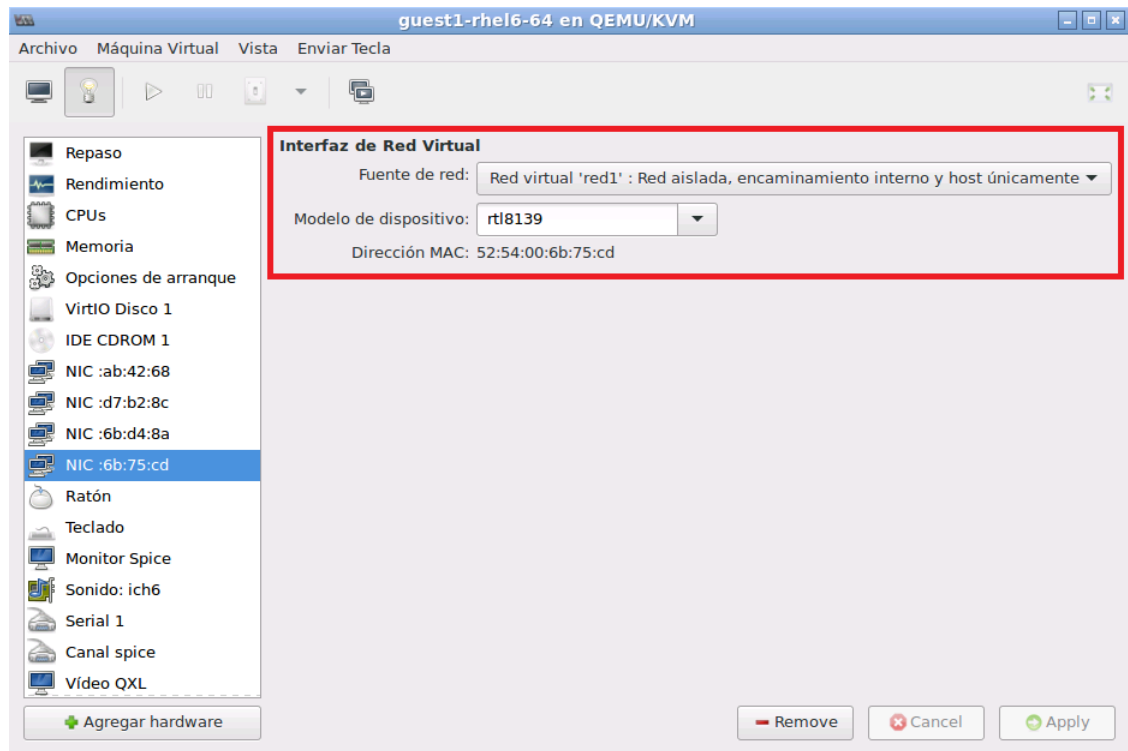


Figura 4.43: Nueva red mostrada en la lista de hardware de invitados (Fuente Propia).

## 4.4.2. Configuración de almacenamiento

Esta sección cubre la creación de dispositivos de almacenamiento basados en disco para máquinas virtuales invitadas.

### 4.4.2.1. Creación de almacenamiento usando virsh

Este procedimiento crea una nueva agrupación de almacenamiento que utiliza un dispositivo de disco con el comando virsh.

La dedicación de un disco a una agrupación de almacenamiento reformateará y borrará todos los datos almacenados actualmente en el dispositivo de disco, ver en el siguiente comando.

```
# parted ldev/sdb
GNU Parted 2.1
Using ldev/sdb
Welcome to GNU Parted! Type 'help' to view a list of commands.
(parted) mklabel
New disk label type? gpt
(parted) quit
Information: You may need to update letc/fstab.
#
```

#### 4.4.2.2. Crear el archivo de configuración del conjunto de almacenamiento

El procedimiento consiste en crear un archivo de texto XML temporal que contenga la información de agrupación de almacenamiento del dispositivo.

El archivo debe tener el formato que se muestra a continuación y contener los siguientes campos:

- **<name>guest \_\_images\_\_disk</name>**: El parámetro name determina el nombre del grupo de almacenamiento. Aquí se utiliza el nombre `guest__images__disk`.
- **<device path = '/dev/sdb'/>**: El parámetro device con el atributo path especifica la ruta del dispositivo de almacenamiento. Este ejemplo utiliza el dispositivo `/dev/sdb`.
- **<target> <path>/dev</path></target>**: El parámetro de destino del sistema de archivos con el subparámetro path determina la ruta en el sistema de archivos de la máquina física host para adjuntar volúmenes creados con este grupo de almacenamiento.
- **<format type= 'gpt'/>**: El parámetro format especifica el tipo de tabla de particiones.

Tendrá que crear el archivo XML para el dispositivo de agrupación de almacenamiento con un editor de texto.

```
<pool type='disk'>
  <name>guest_images_disk</name>
  <source>
    <device path='/dev/sdb' />
    <format type='gpt' />
  </source>
  <target>
    <path>/dev</path>
  </target>
</pool>
```

#### 4.4.2.3. Agregar grupo de almacenamiento

Deberá agregar la definición del conjunto de almacenamiento mediante el comando `virsh pool-define` con el archivo de configuración XML creado en el paso anterior en la línea de comandos.

```
# virsh pool-define ~/guest_images_disk.xml
Pool guest_images_disk defined from /root/guest_images_disk.xml
# virsh pool-list --all
Name                State      Autostart
-----
default             active    yes
guest_images_disk   inactive  no
```

#### 4.4.2.4. Iniciar la agrupación de almacenamiento

Deberá iniciar la agrupación de almacenamiento con el comando `virsh pool-start`. Comprobar que el grupo se ha iniciado con el comando `virsh pool-list --all` en la línea de comandos.

```
# virsh pool-start guest_images_disk
Pool guest_images_disk started
# virsh pool-list --all
Name                State      Autostart
-----
default             active    yes
guest_images_disk   active    no
```

#### 4.4.2.5. Activar inicio automático

Deberá activar el inicio automático de la agrupación de almacenamiento. Autostart configura el servicio libvirt para iniciar el grupo de almacenamiento cuando se inicia el servicio.

```
# virsh pool-autostart guest_images_disk
Pool guest_images_disk marked as autostarted
# virsh pool-list --all
Name                State      Autostart
-----
default             active    yes
guest_images_disk   active    yes
```

#### 4.4.2.6. Verificar la configuración de la agrupación de almacenamiento

Deberá comprobar que el grupo de almacenamiento se creó correctamente, que los tamaños se informaron correctamente y que los informes de estado se ejecutaron, mediante el siguiente comando.

```
# virsh pool-info guest_images_disk
Name:          guest_images_disk
UUID:          551a67c8-5f2a-012c-3844-df29b167431c
State:         running
Capacity:     465.76 GB
Allocation:    0.00
Available:    465.76 GB
# ls -la /dev/sdb

brw-rw----. 1 root disk 8, 16 May 30 14:08 /dev/sdb
# virsh vol-list guest_images_disk
Name                Path
-----
```

#### 4.4.2.7. Eliminar el archivo de configuración temporal

Se ya no es necesario se puede eliminar el archivo de configuración XML del conjunto de almacenamiento temporal con el siguiente comando.

```
# rm ~/guest_images_disk.xml
```

#### 4.4.2.8. Eliminación de una agrupación de almacenamiento mediante virsh

A continuación se muestra cómo eliminar una agrupación de almacenamiento mediante virsh:

1. Para evitar problemas con otras máquinas virtuales invitadas que utilizan la misma agrupación, es mejor detener la agrupación de almacenamiento y liberar todos los recursos en uso por ella con el siguiente comando.

```
# virsh pool-destroy guest_images_disk
```

2. Eliminar la definición de la agrupación de almacenamiento con el siguiente comando.

```
# virsh pool-undefine guest_images_disk
```

#### 4.4.2.9. Grupo de almacenamiento usando Virt Manager

Esta sección cubre la creación de dispositivos de almacenamiento basados en disco para máquinas virtuales invitadas, utilizando Virt Manager. Eso sirve para administrar donde se almacenaran los datos de las máquinas virtuales.

##### **Creación de un grupo de almacenamiento usando a virt-manager:**

1. **Deberá abrir la configuración del grupo de almacenamiento:** Este procedimiento crea un nuevo grupo de almacenamiento que utiliza una partición de un dispositivo de almacenamiento, también se pueden utilizar directorios, entre otras.

- a) Deberá abrir en el menú *Editar* y seleccionar *Detalles de conexión*  
Ver Figura 4.44.

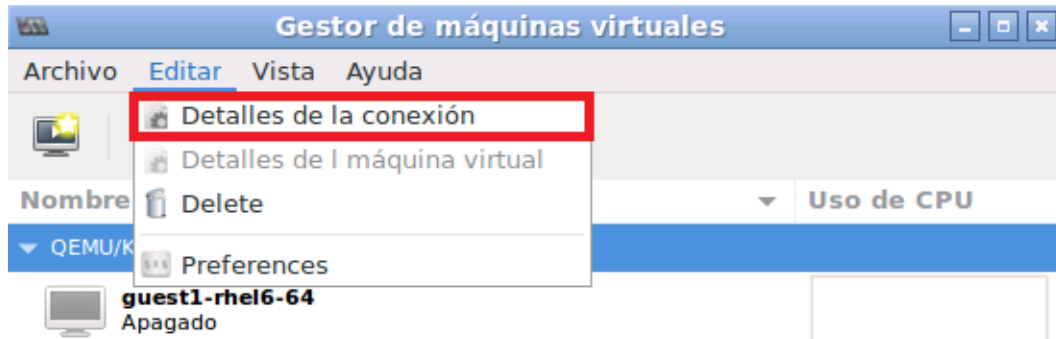


Figura 4.44: Detalles de conexión con Virt Manager (Fuente Propia).

- b) Deberá hacer clic en la pestaña *Almacenamiento* de la ventana  
*Detalles de conexión*  
Ver Figura 4.45.

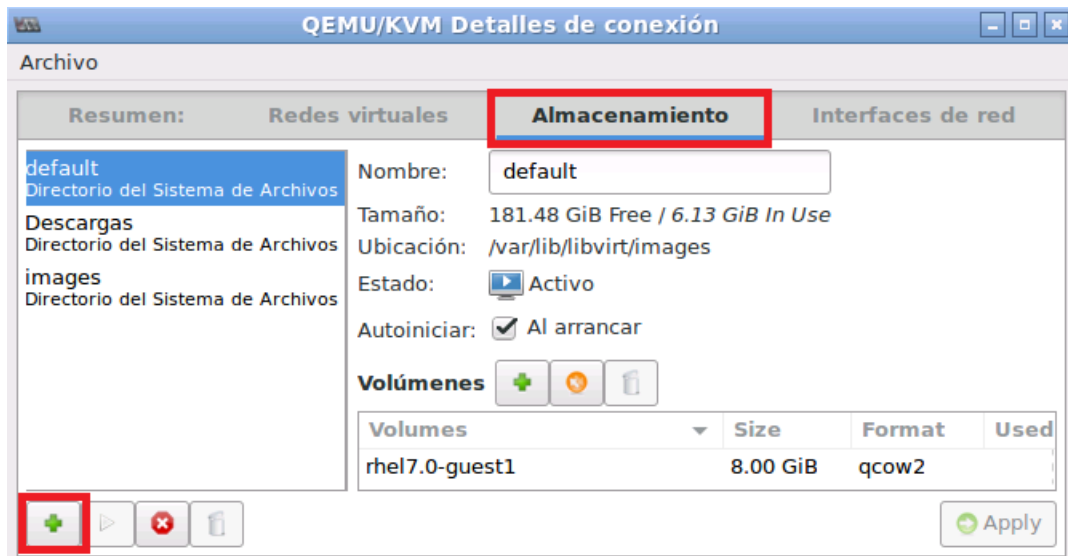


Figura 4.45: Ventana de almacenamiento Virt Manager (Fuente Propia).

## 2. Crear la nueva agrupación de almacenamiento

- a) **Agregar un nuevo grupo de almacenamiento (parte 1):** Deberá presionar el botón + (el botón agregar grupo), ver Figura 4.45. Aparecerá el asistente para agregar una nueva agrupación de almacenamiento. Deberá elegir

un nombre para el grupo. Este ejemplo utiliza el nombre `guest_images_fs`. Deberá cambiar el tipo a `dir`: Directorio del Sistema de Archivos, ver Figura 4.46.

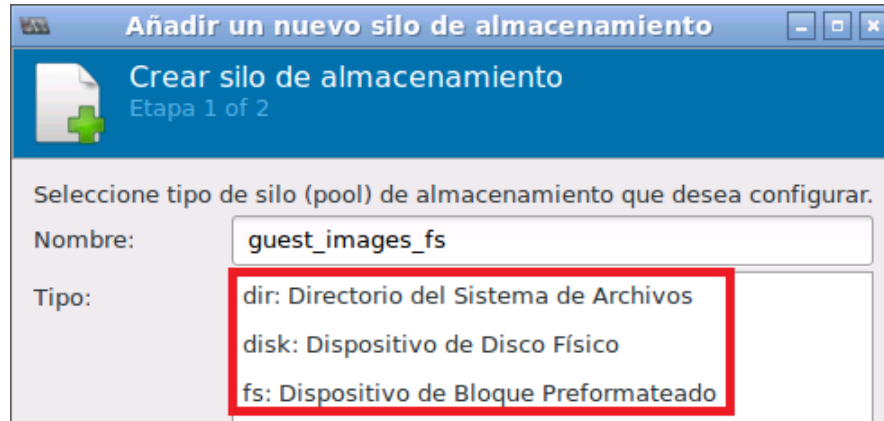


Figura 4.46: Nombre y tipo del grupo de almacenamiento con Virt Manager (Fuente Propia).

- b) **Agregar un nuevo grupo de almacenamiento (parte 2):** Acá podrá verificar la ruta del grupo, si no es correcta deberá cambiar el campo *Ruta de destino*, ver Figura 4.47.

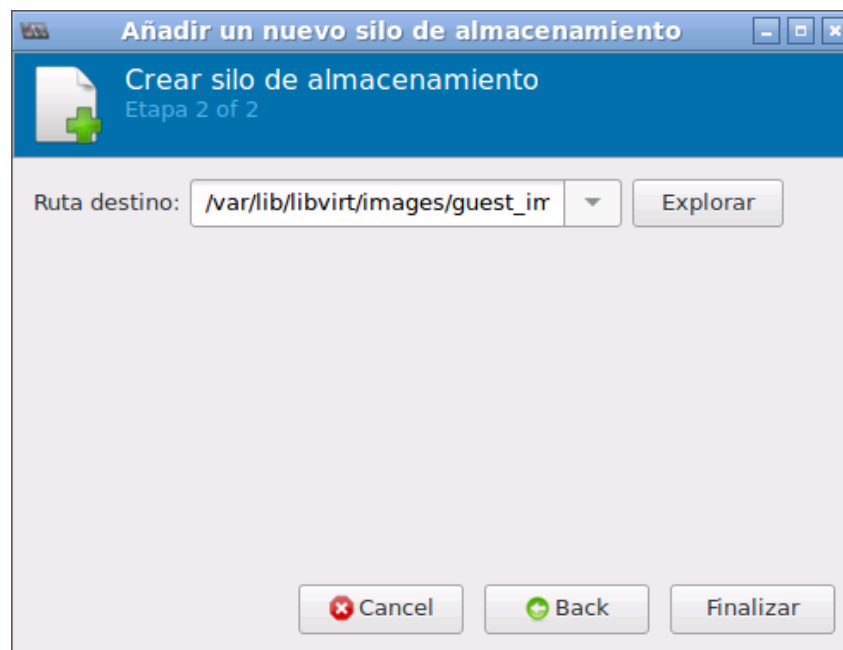


Figura 4.47: Ruta de la agrupación de almacenamiento (Fuente Propia).

3. **Verificar la nueva agrupación de almacenamiento:** El nuevo grupo de almacenamiento aparecerá en la lista de almacenamiento a la izquierda después de unos segundos. Deberá verificar que el campo *Estado* informa del nuevo grupo de almacenamiento como *Activo*. En el campo *Autoiniciar*, deberá hacer clic en la casilla *Al arrancar*. Esto asegurará que el dispositivo de almacenamiento se inicie cada vez que se inicia el servicio libvirt, ver Figura 4.48.

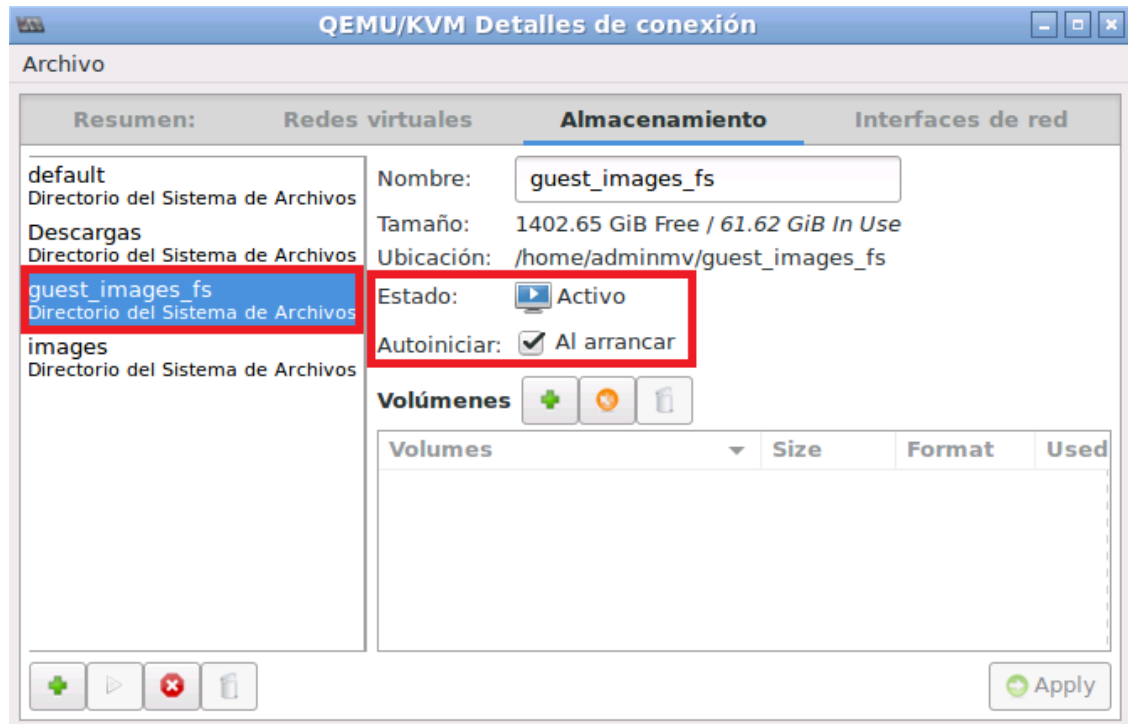


Figura 4.48: Confirmación de la lista de almacenamiento con Virt Manager (Fuente Propia).

#### 4. Eliminación de una agrupación de almacenamiento mediante virt-manager

- a) Para evitar cualquier problema con otras máquinas virtuales invitadas que utilizan la misma agrupación, es mejor detener la agrupación de almacenamiento y liberar todos los recursos en uso por ella. Para ello, deberá seleccionar la agrupación de almacenamiento que desea detener y hacer clic en el icono X rojo situado en la parte inferior de la ventana Almacenamiento, ver Figura 4.49.

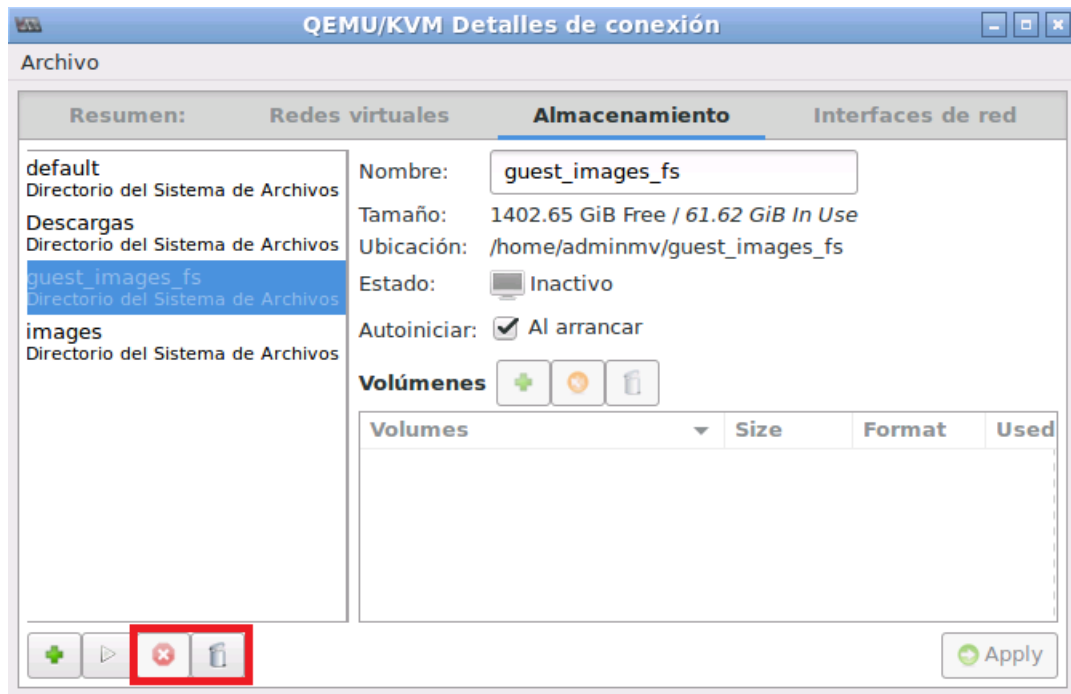


Figura 4.49: Eliminar grupo de almacenamiento (Fuente Propia).

- b) Para eliminar el grupo de almacenamiento deberá hacer clic en el icono *Papelera*. Este icono sólo se habilitara si se detiene la agrupación de almacenamiento primero.

### 4.4.3. Configuración de recursos

En las máquinas físicas existen dispositivos físicos que se pueden insertar y remover como por ejemplo discos duros, tarjetas de red, tarjetas RAM, dispositivos USB, DVD, etc. Estos dispositivos son fabricados por diversos fabricantes cumpliendo todos la misma función. En el mundo de la virtualización también existen dispositivos, pero estos emulan el funcionamiento de los dispositivos físicos, a estos dispositivos se les denomina dispositivos virtuales. Dentro de los dispositivos virtuales existen los dispositivos emulados, dispositivos virtio y dispositivos asignados.

#### 4.4.3.1. Configuración del dispositivo de la máquina virtual

1. Los *dispositivos emulados* son dispositivos puramente virtuales que imitan el hardware real, permitiendo que los sistemas operativos invitados no modificados trabajen con ellos usando sus controladores estándar.
2. Los *dispositivos virtio* son dispositivos puramente virtuales diseñados para funcionar de forma óptima en una máquina virtual. Los dispositivos virtio son similares a los dispositivos emulados, sin embargo, las máquinas virtuales que no son de Linux no incluyen los controladores que requieren de forma predeterminada. El software de administración de virtualización como Virtual Machine Manager y el hipervisor de virtualización KVM instalan estos automáticamente para los sistemas operativos invitados no Linux.
3. Los *dispositivos asignados* son dispositivos físicos expuestos a la máquina virtual. Este método también se conoce como "paso". La asignación de dispositivos permite a las máquinas virtuales un acceso exclusivo a los dispositivos PCI, permitiendo que aparezcan y se comporten como si fueran físicamente conectados al sistema operativo invitado (Laura Novich, 2015).

#### 4.4.3.2. Dispositivos PCI

La asignación de dispositivos PCI sólo está disponible en plataformas de hardware que admitan Intel VT-d o AMD IOMMU. Estas especificaciones Intel VT-d o AMD IOMMU deben estar habilitadas en la BIOS para que la asignación de dispositivos PCI funcione (Laura Novich, 2015).

Esta sección tiene por objetivo dar un ejemplo sobre como asignar un dispositivo en modo texto, a través de la línea de comandos.

##### 1. Habilitar las especificaciones Intel VT-d

Las especificaciones Intel VT-d proporcionan soporte de hardware para asignar directamente un dispositivo físico a una máquina virtual. Estas especificaciones son necesarias para utilizar la asignación de dispositivos PCI.

Las especificaciones Intel VT-d deben estar habilitadas en la BIOS. Algunos fabricantes del sistema desactivan estas especificaciones de forma predeterminada. Los términos utilizados para referirse a estas especificaciones pueden diferir entre los fabricantes.

## 2. Activar Intel VT-d en el kernel

Deberá activar Intel VT-d en el kernel agregando el parámetro `intel_iommu = pt` al final de la línea `GRUB_CMD LINUX_LINUX`, dentro de las comillas, en el archivo `/etc/sysconfig/grub`. El siguiente ejemplo es un archivo grub modificado con Intel VT-d activado.

```
GRUB_CMDLINE_LINUX="rd.lvm.lv=vg_VolGroup00/LogVol01
vconsole.font=latarcyrheb-sun16 rd.lvm.lv=vg_VolGroup_1/root
vconsole.keymap=us $([ -x /usr/sbin/rhcrashkernel-param ] &&
/usr/sbin/
rhcrashkernel-param || :) rhgb quiet intel_iommu=pt"
```

## 3. Regenerar archivo de configuración

Se deberá regenerar el grub para que la activación haga efecto, ejecutando.

```
grub2-mkconfig -o /boot/grub2/grub.cfg
```

## 4. Listo para usar

Deberá reiniciar el sistema para habilitar los cambios. Su sistema es ahora capaz de asignar de dispositivos PCI.

### 4.4.3.3. Asignación de un dispositivo PCI con virsh

Estos pasos cubren la asignación de un dispositivo PCI a una máquina virtual en un hipervisor KVM en modo texto desde la línea de comandos.

Se vera un ejemplo que utiliza un controlador de red PCI con el código de identificador PCI, `pci_0000_01_00_0` y una máquina huésped completamente virtualizada denominada `guest1-rhel7-64`.

## 1. Identificar el dispositivo

En primer lugar, deberá identificar el dispositivo PCI designado para la asignación de dispositivos a la máquina virtual. Deberá utilizar el comando *lspci* para enumerar los dispositivos PCI disponibles. También podrá refinar la salida de *lspci* agregando al comando *grep*.

```
# lspci | grep Ethernet
00:19.0 Ethernet controller: Intel Corporation 82567LM-2 Gigabit
Network Connection
01:00.0 Ethernet controller: Intel Corporation 82576 Gigabit
Network Connection (rev 01)
01:00.1 Ethernet controller: Intel Corporation 82576 Gigabit
Network Connection (rev 01)
```

Este controlador Ethernet se muestra con el identificador corto *00:19.0*. Se necesita averiguar el identificador completo utilizado por *virsh* para asignar este dispositivo PCI a una máquina virtual.

Por lo tanto, deberá utilizar el comando *virsh nodedev-list* para enumerar todos los dispositivos de un tipo particular PCI que están conectados a la máquina host. Luego deberá observar la salida de la cadena que se asigna al identificador corto del dispositivo que desea utilizar.

Este ejemplo muestra la cadena que se asigna al controlador Ethernet con el identificador corto *00:19.0* con el comando *virsh nodedev-list*.

```
# virsh nodedev-list --cap pci
pci_0000_00_00_0
pci_0000_00_01_0
pci_0000_00_03_0
pci_0000_00_07_0
pci_0000_00_10_0
pci_0000_00_10_1
pci_0000_00_14_0
pci_0000_00_14_1
pci_0000_00_14_2
pci_0000_00_14_3
pci_0000_00_19_0
pci_0000_00_1a_0
pci_0000_00_1a_1
pci_0000_00_1a_2
pci_0000_00_1a_7
```

Deberá registrar el número de dispositivo PCI que se asigna al dispositivo que desea utilizar; esto es necesario para pasos siguientes.

## 2. Revisar la información del dispositivo

La información sobre el dominio, el bus y la función están disponibles en la salida del comando `virsh nodedev-dumpxml`.

```
# virsh nodedev-dumpxml pci_0000_00_19_0
<device>
  <name>pci_0000_00_19_0</name>
  <parent>computer</parent>
  <driver>
    <name>e1000e</name>
  </driver>
  <capability type='pci'>
    <domain>0</domain>
    <bus>0</bus>
    <slot>25</slot>
    <function>0</function>
    <product id='0x1502'>82579LM Gigabit Network
Connection</product>
    <vendor id='0x8086'>Intel Corporation</vendor>
    <iommuGroup number='7'>
      <address domain='0x0000' bus='0x00' slot='0x19'
function='0x0' />
    </iommuGroup>
  </capability>
</device>
```

## 3. Determinar los detalles de configuración necesarios

Deberá consultar la salida del comando `virsh nodedev-dumpxml pci_0000_00_19_0` para los valores requeridos para el archivo de configuración.

El dispositivo de ejemplo tiene los siguientes valores que se usan en la configuración.

```
bus='0'
slot='25'
function='0'
```

## 4. Agregar detalles de configuración

Deberá ejecutar `virsh edit`, especificar el nombre de la máquina virtual y deberá agregar una entrada de dispositivo en la sección `<source>` para asignar el dispositivo PCI a la máquina virtual invitada.

```
# virsh edit guest1-rhel7-64
<hostdev mode='subsystem' type='pci' managed='yes'>
  <source>
    <address domain='0' bus='0' slot='25' function='0' />
  </source>
</hostdev>
```

## 5. Iniciar la máquina virtual

Por último deberá iniciar la máquina virtual para observar el dispositivo con el siguiente comando.

```
# virsh start guest1-rhel7-64
```

### 4.4.3.4. Asignación de un dispositivo PCI con virt-manager

Los dispositivos PCI se pueden agregar a máquinas virtuales invitadas mediante la herramienta de administración gráfica. El siguiente procedimiento agrega un controlador Gigabit Ethernet a una máquina virtual invitada.

#### 1. Abra la configuración de hardware

Deberá abrir la máquina virtual invitada y hacer clic en el botón *Agregar hardware* para agregar un nuevo dispositivo a la máquina virtual, ver Figura 4.50.



Figura 4.50: Asignar un nuevo dispositivo a máquina virtual con virt-manager (Fuente Propia).

## 2. Seleccione un dispositivo PCI

Deberá seleccionar PCI Host Device en la lista Hardware de la izquierda. Proceda seleccionando un dispositivo PCI no utilizado. Tenga en cuenta que la selección de dispositivos PCI actualmente en uso por otro invitado provocara errores. En este ejemplo, se utiliza un dispositivo de red `0000:05:00:1`. Deberá hacer clic en *Finalizar* para terminar la configuración, ver Figura 4.51.

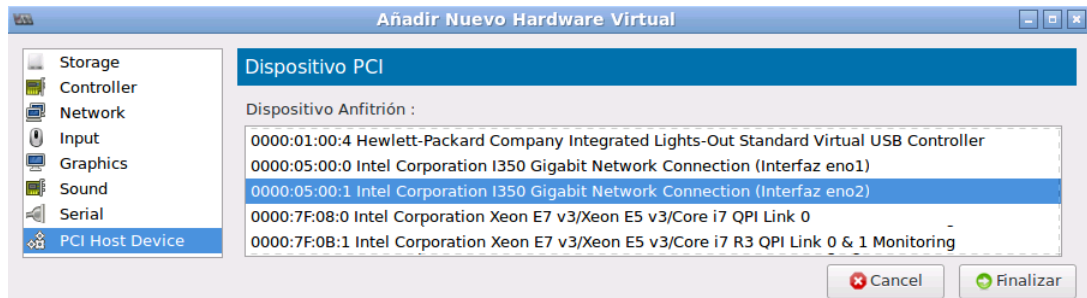


Figura 4.51: El asistente para agregar nuevo hardware virtual (Fuente Propia).

## 3. Agregar el nuevo dispositivo

La instalación está completa y la máquina virtual invitada ahora tiene acceso directo al dispositivo PCI, ver Figura 4.51.



Figura 4.52: Ventana de información de hardware de la máquina virtual (Fuente Propia).

## 4.5. Configuración de la seguridad del entorno virtual

Teniendo todo preparado para empezar a poblar, antes es necesario verificar que el entorno virtual es seguro. Primero se chequeó que el firewall del servidor funciona correctamente, también se instaló una aplicación llamada *fail2ban* la cual protege al servidor de ataques por fuerza bruta.

### 4.5.1. Configuración del firewall

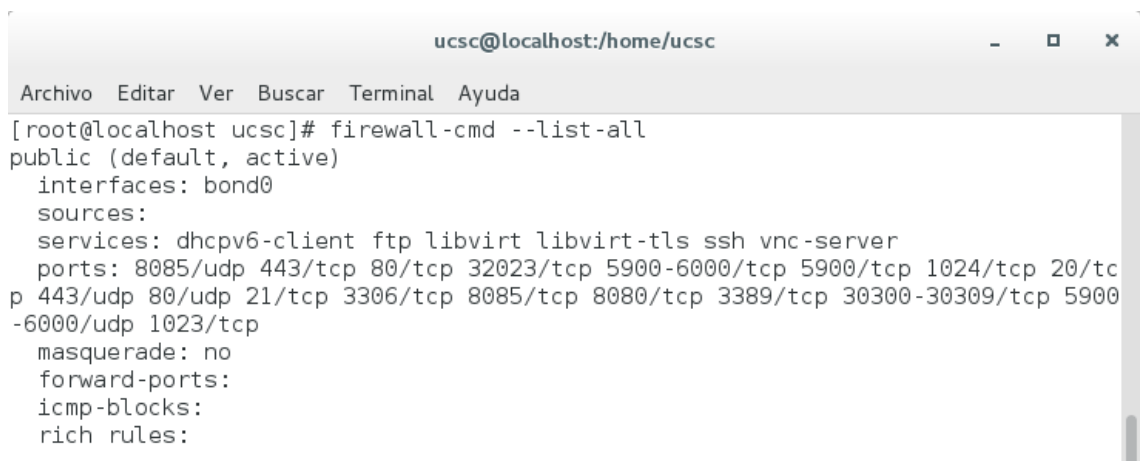
1. Activar/Habilitar el firewall:

```
systemctl start firewalld
systemctl enable firewalld
```

2. Para saber que zonas por defecto están habilitadas:

```
firewall-cmd --get-default-zone
```

3. Devuelve la zona *public* por defecto y se aplica a la interfaz *Bond0*, ver Figura 4.53.



```
ucsc@localhost:/home/ucsc
Archivo Editar Ver Buscar Terminal Ayuda
[root@localhost ucsc]# firewall-cmd --list-all
public (default, active)
  interfaces: bond0
  sources:
  services: dhcpv6-client ftp libvirt libvirt-tls ssh vnc-server
  ports: 8085/udp 443/tcp 80/tcp 32023/tcp 5900-6000/tcp 5900/tcp 1024/tcp 20/tcp
  443/udp 80/udp 21/tcp 3306/tcp 8085/tcp 8080/tcp 3389/tcp 30300-30309/tcp 5900
  -6000/udp 1023/tcp
  masquerade: no
  forward-ports:
  icmp-blocks:
  rich rules:
```

Figura 4.53: Detalles zona por defecto firewall (Fuente Propia).

4. Para mostrar detalladas de una zona en específico por ejemplo *external*:

```
firewall-cmd --list-service --zone=external
```

### 5. Filtro de servicios:

El firewall puede filtrar varios servicios, por ejemplo ssh (puerto 22), http (80) y mysql (3306). Para listar los servicios definidos por defecto.

```
firewall-cmd -get-services
```

Para adicionar mas servicios, se puede adicionar un nuevo archivo xml en la siguiente dirección:

```
ls /usr/lib/firewalld/services
```

Para adicionar o remover los servicios a nuestra zona, por ejemplo servicio http:

```
firewall-cmd -add-service=http
```

```
firewall-cmd -remove-service=http
```

Si se reinicia la máquina, el servicio que se adiciona desaparece. Entonces para hacerlo permanente se necesita adicionar al comando la palabra *permanent* seguido de un reload del servicio:

```
firewall-cmd -add-service=http -permanent
```

```
firewall-cmd -reload
```

### 6. Agregar y quitar puertos:

Adicionar o remover un puerto por ejemplo TCP 465, se lleva a cabo de la siguiente manera.

```
firewall-cmd -add-port=465/tcp
```

```
firewall-cmd -remove-port=465/tcp
```

Si se quiere hacer permanente el puerto.

```
firewall-cmd -add-port=465/tcp -permanent
```

### 4.5.2. Firewall modo gráfico en CentOS 7

CentOS 7 en modo gráfico configura el firewall en una aplicación que tiene dos modos, *Ejecución* y *Permanente*, por lo tanto si desea agregar puertos a una zona que deben siempre estar activados deberá usar el modo permanente. En el proyecto se agregaron los puertos en modo gráfico, Firewall en Figura 4.54.

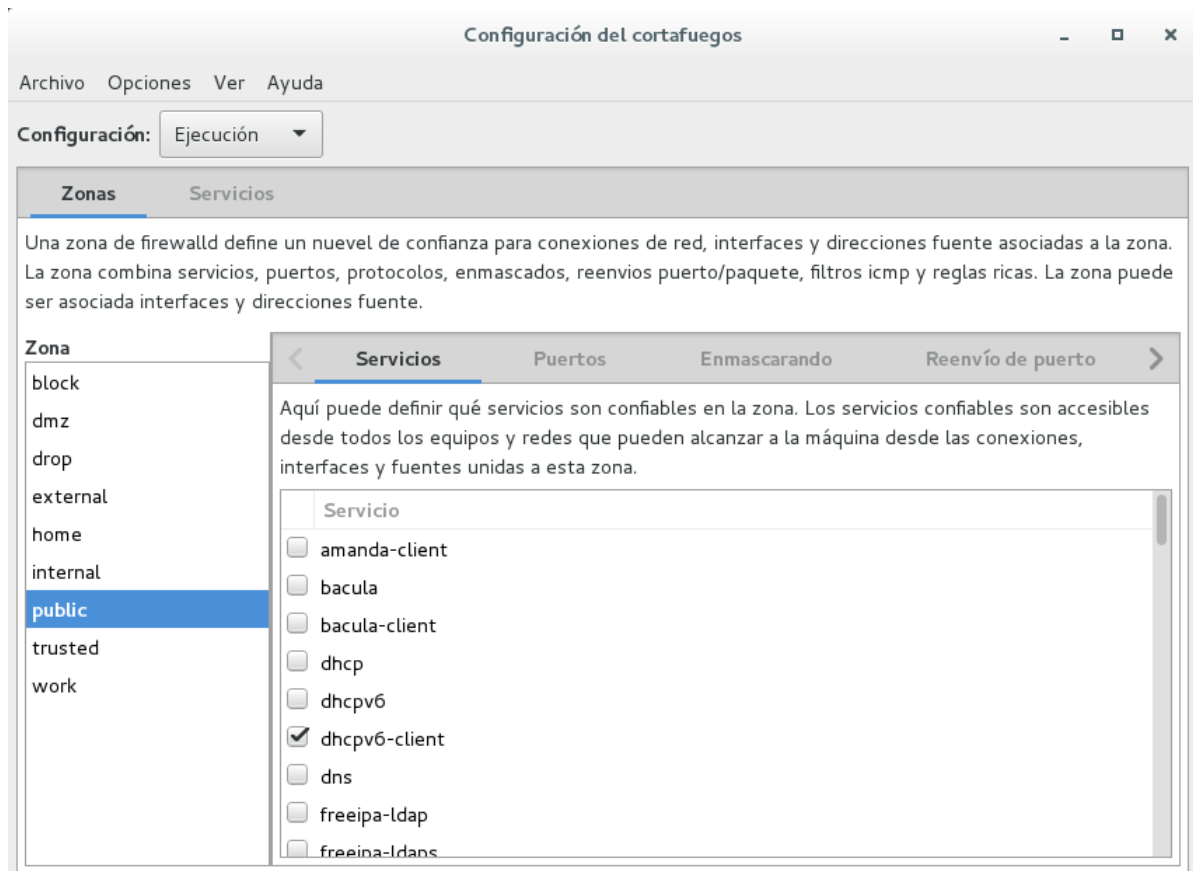


Figura 4.54: Firewall aplicación en CentOS 7 (Fuente Propia).

### 4.5.3. Fail2ban

Una gran parte de servidores basados en Linux tienen conexión SSH a través del puerto 22 con fines de administración remota. Es un puerto bien conocido, por lo mismo, a menudo recibe ataques de fuerza bruta. Fail2ban es un software que escanea archivos de registro con motivo de ver intentos de inicio de sesión con fuerza bruta en tiempo real y prohíbe a los atacantes por medio de firewall o iptables. Fail2ban distingue los

intentos de violación de seguridad al servidor en el marco de tiempo establecido por el administrador y bloquea las direcciones IP que muestran signos de ataques de fuerza bruta. Su ejecución es en segundo plano y explora archivos de registro con el fin de detectar patrones de inicio de sesión inusuales (Howtforge, 2016).

La universidad exige la instalación de este programa en los servidores, por esta razón se instalara en este servidor.

#### 4.5.4. Instalación Fail2Ban

En la instalación de Fail2Ban en CentOS 7, hubo que instalar paquetes adicionales EPEL para Enterprise Linux. EPEL tiene paquetes adicionales para todas las versiones de CentOS, uno de ellos es Fail2Ban (Howtforge, 2016).

1. Se ejecutó los siguientes comandos en modo root:

```
yum install epel-release
```

```
yum install fail2ban fail2ban-systemd
```

2. Se actualizaron las políticas de SELinux:

```
yum update -y selinux-policy*
```

3. Configurar Fail2Ban:

Una vez instalado, se configuró y personalizó el software con un archivo de configuración *jail.local*. El archivo *jail.local* anula el archivo *jail.conf*, se utilizó para hacer su actualización de configuración personalizada segura.

Se hizo una copia del archivo *jail.conf* y se guardó con el nombre *jail.local*:

```
cp -pf /etc/fail2ban/jail.conf /etc/fail2ban/jail.local
```

4. Se abrió el archivo *jail.local* para su edición con *nano* usando el siguiente comando.

```
nano /etc/fail2ban/jail.local
```

El código del archivo consiste en muchas líneas de código que se ejecutan para la prohibición de una o varias direcciones IP.

Ejemplo de archivo *jail.local* en la Figura 4.55.

```
ucsc@localhost:/home/ucsc
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 2.3.1  Fichero: /etc/fail2ban/jail.local

[DEFAULT]

#
# MISCELLANEOUS OPTIONS
#

# "ignoreip" can be an IP address, a CIDR mask or a DNS host. Fail2ban will not
# ban a host which matches an address in this list. Several addresses can be
# defined using space separator.
ignoreip = 127.0.0.1/8

# External command that will take an tagged arguments to ignore, e.g. <ip>,
# and return true if the IP is to be ignored. False otherwise.
#
# ignorecommand = /path/to/command <ip>
ignorecommand =

# "bantime" is the number of seconds that a host is banned.
bantime = 600

^G Ver ayuda  ^O Guardar  ^R Leer Fich  ^Y Pág Ant  ^K CortarTxt  ^C Pos actual
^X Salir      ^J Justificar  ^W Buscar    ^V Pág Sig  ^U PegarTxt  ^T Ortografía
```

Figura 4.55: Archivo *jail.local* del servidor (Fuente Propia).

**Ignoreip:** se utiliza para establecer la lista de IPs que no serán prohibidas. La lista se administra con un separador de espacio. Este parámetro es utilizado para asignar una dirección IP personal si entra al servidor desde con dirección IP fija.

**Bantime:** Este parámetro se utiliza para definir la duración en segundos para los que tiene que ser prohibido un host.

**Findtime:** Este parámetro se utiliza para verificar si un host debe ser prohibido, cuando el anfitrión genera maxretry en su última findtime, está prohibido.

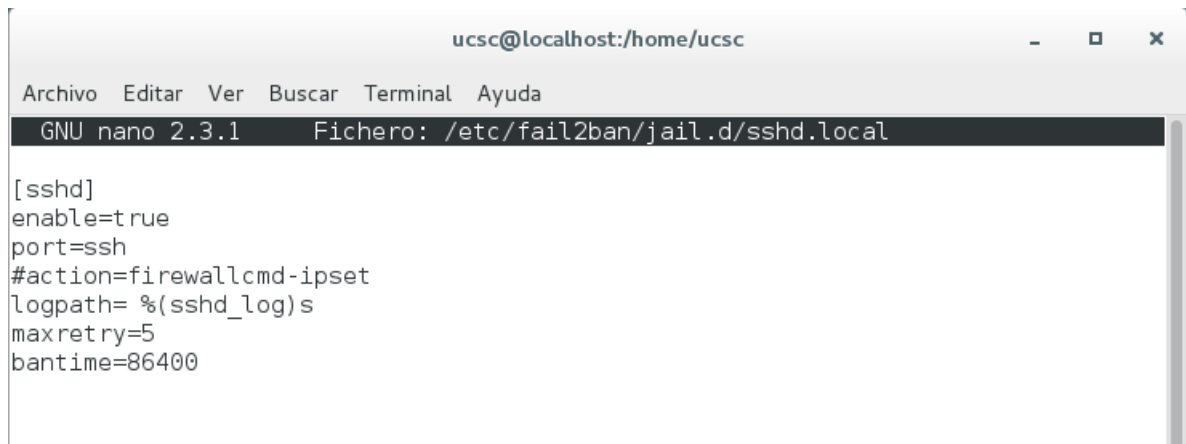
**Maxretry:** Este parámetro se utiliza para establecer el límite en números de reintentos para un huésped, si sobrepasa el límite, el anfitrión lo prohíbe.

### 4.5.5. Añadir un archivo para proteger SSH con Fail2Ban

1. Se creó un nuevo archivo con el editor *nano*, en este caso *sshd.local*:

```
nano /etc/fail2ban/jail.d/sshd.local
```

Para el archivo anterior, se debieron agregar las siguientes líneas de código, ver la Figura 4.56.



```
ucsc@localhost:/home/ucsc
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.3.1 Fichero: /etc/fail2ban/jail.d/sshd.local
[sshd]
enable=true
port=ssh
#action=firewallcmd-ipset
logpath= %(sshd_log)s
maxretry=5
bantime=86400
```

Figura 4.56: Archido jail.local del servidor (Fuente Propia).

Parámetro **enabled** se asigna en *true*, con el propósito de entregar protección, para desactivarla, se asigna en *false*. El parámetro de filtro comprueba el archivo de configuración *sshd*, ubicado en la ruta */etc/fail2ban/filter.d/sshd.conf*.

El parámetro **action** se utiliza para derivar la dirección de IP que se prohibirá, hace uso del filtro proveniente de la ruta */etc/fail2ban/action.d/firewallcmd-ipset.conf*.

**Port** se utiliza para cambiar el puerto. Cuando utiliza el puerto 22, no hay necesidad de cambiarlo.

**Logpath** proporciona la ruta del archivo de registro. Este archivo se escanea por Fail2Ban.

**Maxretry** es utilizado para asignar el límite máximo para entradas de acceso fallidos.

**Bantime**: Este parámetro se utiliza para definir la duración en segundos para los que tiene que ser prohibido un host.

### 4.5.6. Ejecución de servicio Fail2Ban

1. Para iniciar y permitir Fail2Ban se ejecute en el arranque del servidor se teclearon los siguientes comandos:

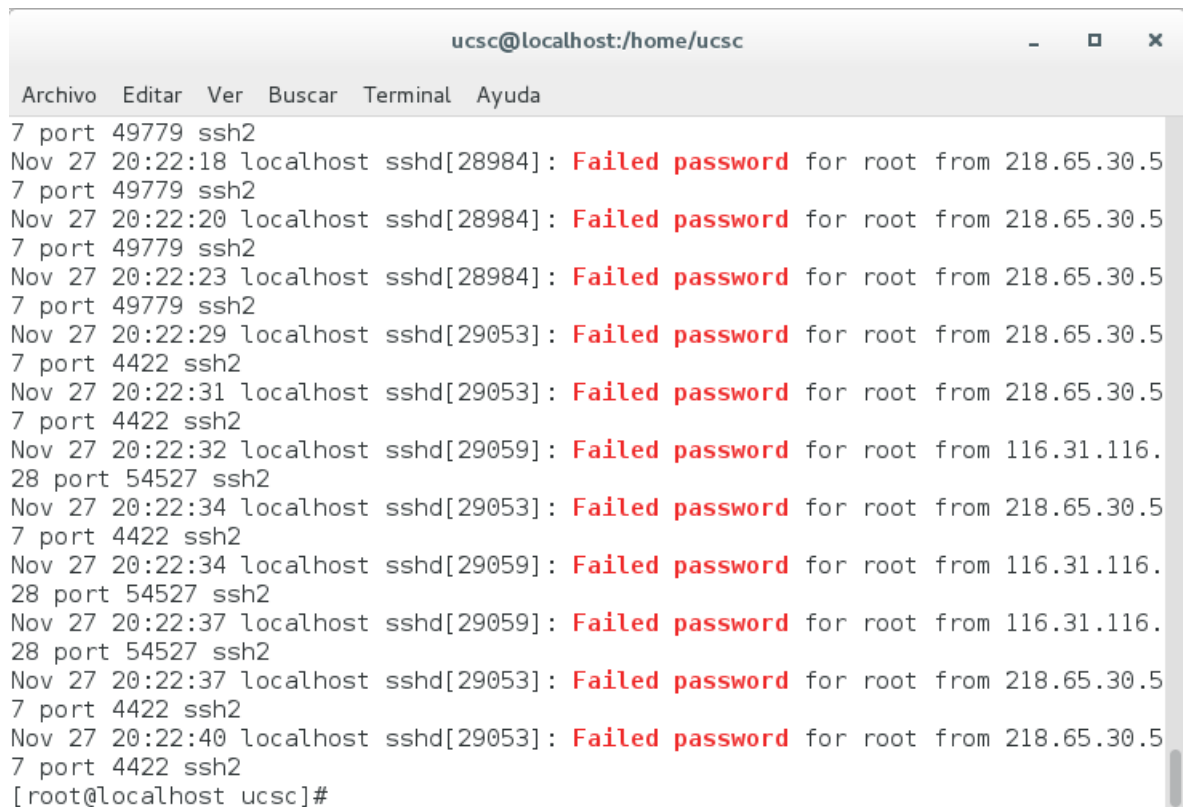
```
systemctl enable fail2ban
```

```
systemctl start fail2ban
```

2. Para el seguimiento de las entradas de inicio de sesión fallidos a través del puerto de SSH se utilizó el siguiente comando:

```
cat /var/log/secure | grep 'Failed password'
```

Con el comando anterior se obtuvo una lista de los intentos fallidos de contraseña para diferentes direcciones IP. El formato de los resultados sera de la siguiente forma para el servidor, ver la Figura 4.57.



The screenshot shows a terminal window titled 'ucsc@localhost:/home/ucsc'. The terminal output displays a list of failed password attempts for SSH. Each entry includes the port number, the date and time, the local host, the sshd process ID, the message 'Failed password', the user 'root', and the source IP address. The attempts are as follows:

```

7 port 49779 ssh2
Nov 27 20:22:18 localhost sshd[28984]: Failed password for root from 218.65.30.5
7 port 49779 ssh2
Nov 27 20:22:20 localhost sshd[28984]: Failed password for root from 218.65.30.5
7 port 49779 ssh2
Nov 27 20:22:23 localhost sshd[28984]: Failed password for root from 218.65.30.5
7 port 49779 ssh2
Nov 27 20:22:29 localhost sshd[29053]: Failed password for root from 218.65.30.5
7 port 4422 ssh2
Nov 27 20:22:31 localhost sshd[29053]: Failed password for root from 218.65.30.5
7 port 4422 ssh2
Nov 27 20:22:32 localhost sshd[29059]: Failed password for root from 116.31.116.28
28 port 54527 ssh2
Nov 27 20:22:34 localhost sshd[29053]: Failed password for root from 218.65.30.5
7 port 4422 ssh2
Nov 27 20:22:34 localhost sshd[29059]: Failed password for root from 116.31.116.28
28 port 54527 ssh2
Nov 27 20:22:37 localhost sshd[29059]: Failed password for root from 116.31.116.28
28 port 54527 ssh2
Nov 27 20:22:37 localhost sshd[29053]: Failed password for root from 218.65.30.5
7 port 4422 ssh2
Nov 27 20:22:40 localhost sshd[29053]: Failed password for root from 218.65.30.5
7 port 4422 ssh2
[root@localhost ucsc]#

```

Figura 4.57: Registro de fallas de intentos de inicio con contraseña (Fuente Propia).

## 4.6. Creación del entorno virtual

En esta sección se procedió a instalar las máquinas virtuales en el servidor, procedimiento al cual se le denomina *Creación el entorno virtual*.

En este Capítulo en la sección 3 *Construcción del entorno virtual*, se presento a Virt Manager explicando que es un programa con el cual se pueden crear y administrar máquinas virtuales, culminando con un ejemplo de instalación. También en este Capítulo en la sección 2 *Planificación del entorno actual*, se especifico que se instalaran 24 máquinas virtuales con el sistema operativo Lubuntu, pues es una sistema que ocupa pocos recursos.

Antes de empezar con la creación de máquinas virtuales con el sistema operativo Lubuntu, deberá asegurarse de conocer los requisitos sobre los recursos necesarios para ellas. Cuando proceda con eso deberá establecer los requisitos que las aplicaciones necesitan para funcionar correctamente en las máquinas virtuales. En lugar de dar a una máquina virtual dos CPU virtuales porque eso es lo que tenía en un servidor físico, puede probar dándole uno y ver cómo funciona. Con la memoria RAM, deberá asignar sólo lo que una máquina virtual va a utilizar, puede observar las estadísticas de la memoria del servidor para identificar la cantidad de memoria RAM que se esta utilizando en el tope de uso y con ello asignar o disminuir memoria RAM a la máquina virtual en consecuencia. Para la creación de discos virtuales, si una máquina física tiene un disco de 72 GB porque era la unidad más pequeña que podía obtener y está utilizando sólo 8 GB de espacio en disco en él, deberá crear discos virtuales más pequeños. Hacer esto ahorrará espacio de disco valioso y costoso. Si se necesita más espacio de almacenamiento más tarde puede agregar más recursos a una máquina virtual según sea necesario, una de las ventajas de administrar un entorno virtual (Siebert, 2009).

A continuación se mostrara como se creo una máquina virtual para el servidor. Antes mencionar que en este Capítulo, en la sección 2 *Planificación del entorno virtual* se estableció que para una correcta ejecución, sin problemas de una máquina virtual, las aplicaciones necesitan como mínimo 2 GB RAM y 40 GB de disco de almacenamiento para funcionar correctamente.

### 4.6.1. Creación de máquina virtual

En la *Planificación del entorno virtual*, se estableció instalar el sistema operativo Lubuntu en las máquinas virtuales, entonces lo primero que se hizo fue descargar la imagen ISO directamente del sitio web de la distribución.

#### 4.6.1.1. Descarga de Lubuntu

Lubuntu descargado, ver Figura 4.58.

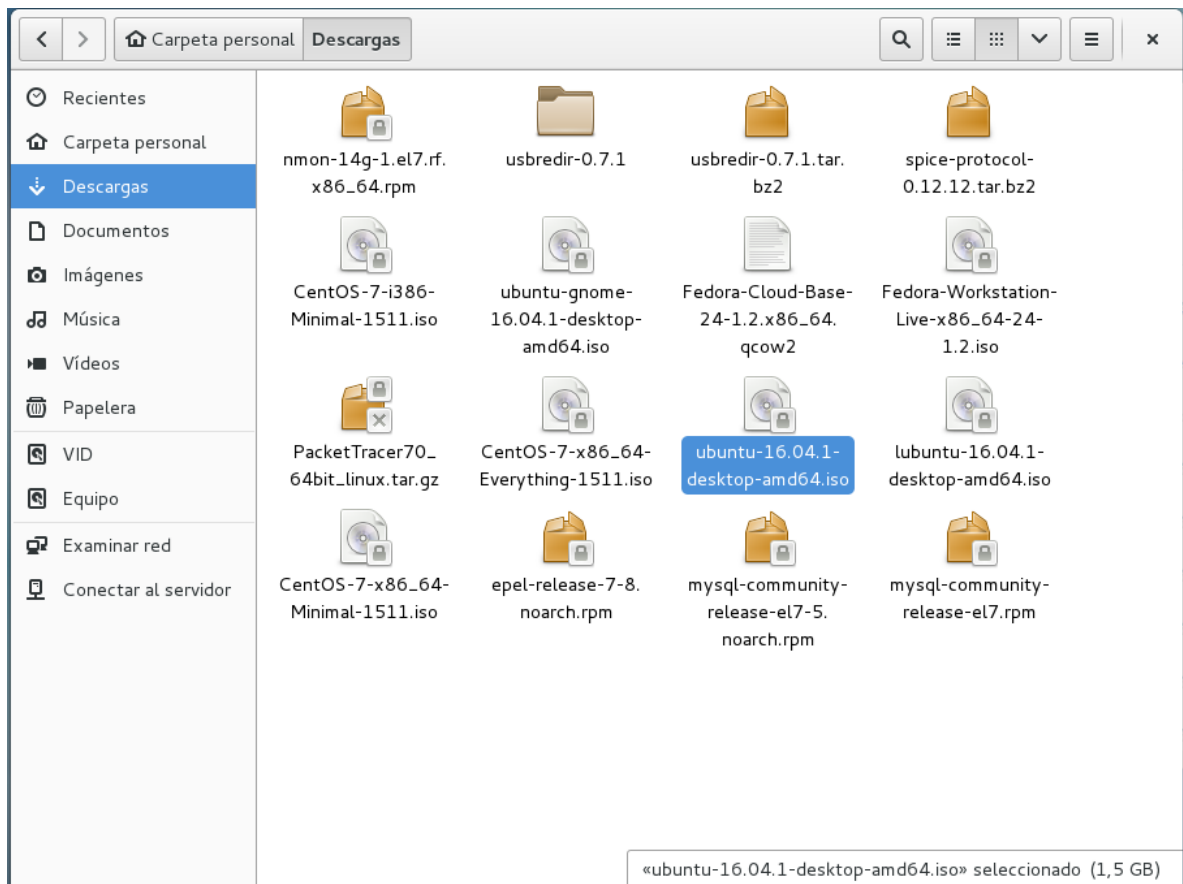


Figura 4.58: Selección de ISO distribución Lubuntu modo gráfico (Fuente Propia).

Esta imagen ISO de la distribución Lubuntu se usó para la instalación de la máquina virtual.

#### 4.6.1.2. Instalar Lubuntu con virt-manager

Se instaló la distribución Lubuntu en modo gráfico con virt-manager.

1. Se procedió a abrir la aplicación *Virt Manager*, ver Figura 4.59.

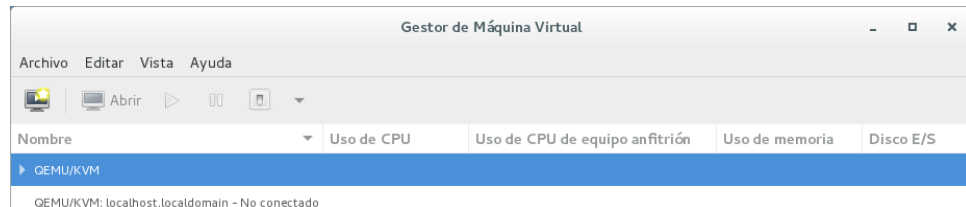


Figura 4.59: Inicio aplicación virt-manager (Fuente Propia).

2. Se debió dirigir a *Archivo, Nueva máquina virtual*, ver Figura 4.60.

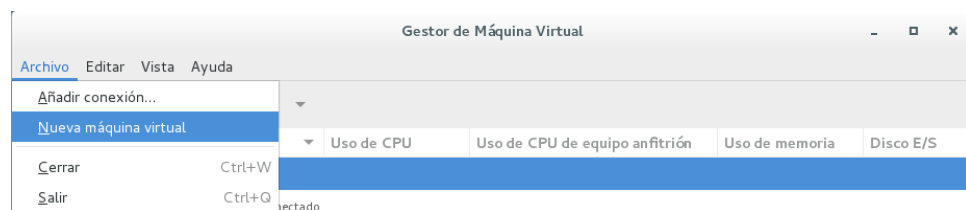


Figura 4.60: Crear una nueva máquina virtual virt-manager (Fuente Propia).

3. En la opción de instalación, se seleccionó la primera opción, que esta por defecto y se presionó *Adelante*, ver Figura 4.61.

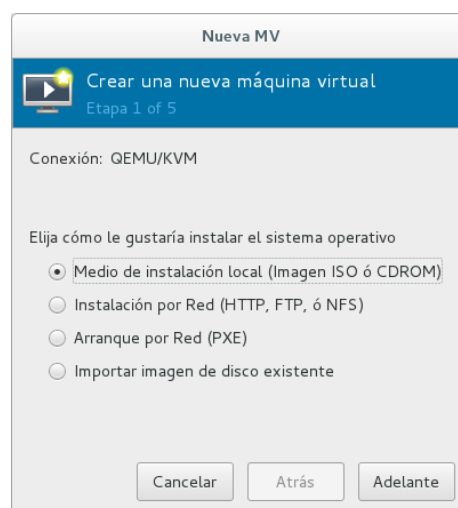


Figura 4.61: Medio de instalación ISO o CDROM, virt-manager (Fuente Propia).

- Se seleccionó *Utilizar imagen ISO* y se presiono *Explorar* para buscar la ISO de Lubuntu descargada, ver Figura 4.62.

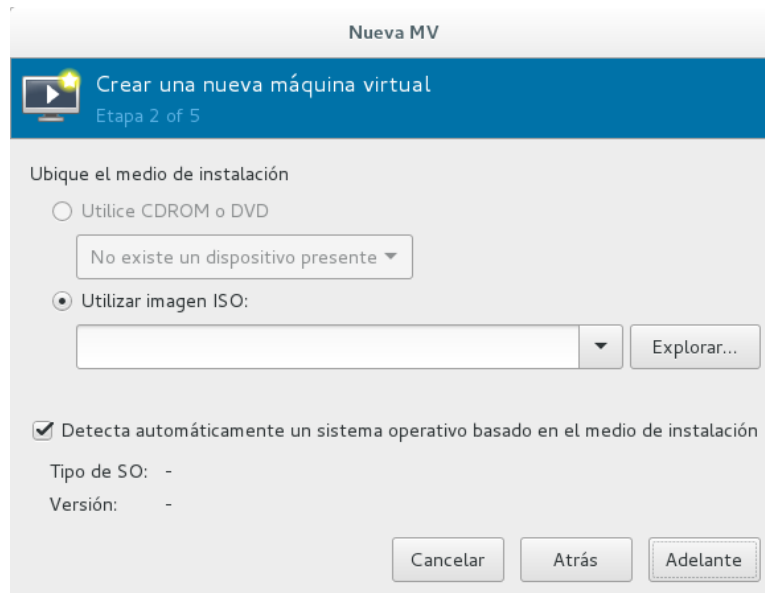


Figura 4.62: Buscar imagen ISO, virt-manager (Fuente Propia).

- Se posicionó a la izquierda donde dice *Descargas* y se buscó la ISO con la distribución Lubuntu, y se presionó *Elegir volumen*, ver Figura 4.63.

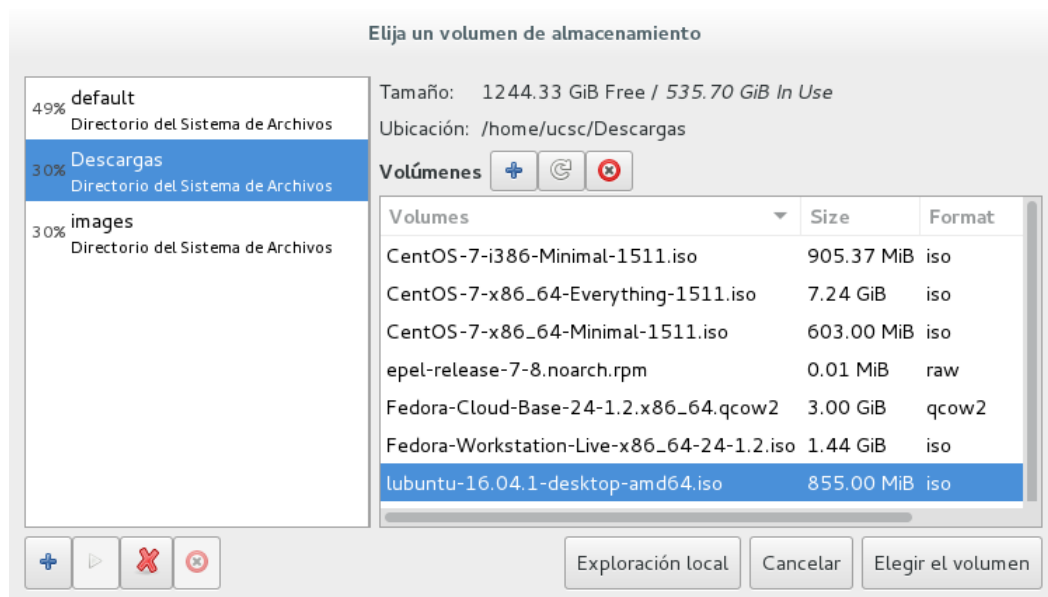


Figura 4.63: Seleccionar imagen ISO, virt-manager (Fuente Propia).

- Se debió desactivar la opción *Detectar automáticamente el sistema operativo* y se seleccionó el sistema operativo *Ubuntu Vivid Varvet*, pues Lubuntu no aparece entre las opciones y es un derivado de Ubuntu, ver Figura 4.64.

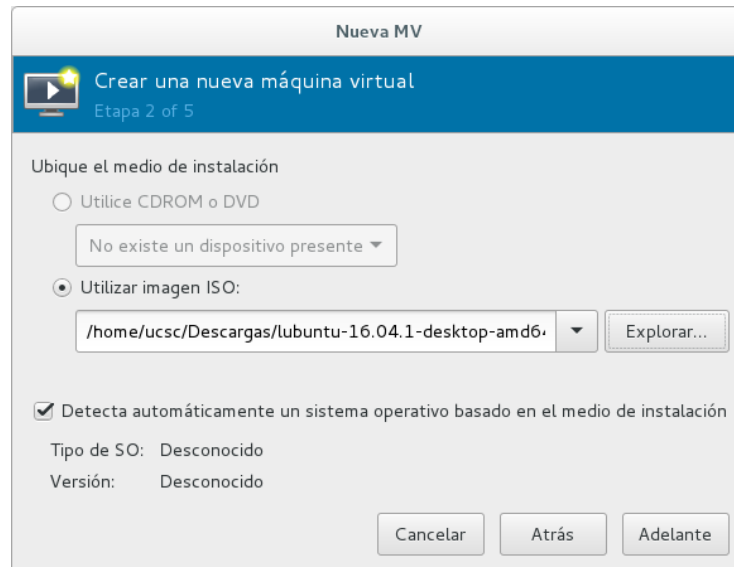


Figura 4.64: Buscar el sistema operativo, virt-manager (Fuente Propia).

Se seleccionó el sistema operativo *Ubuntu Vivid Varvet* y presionar *Adelante*, ver Figura 4.65.

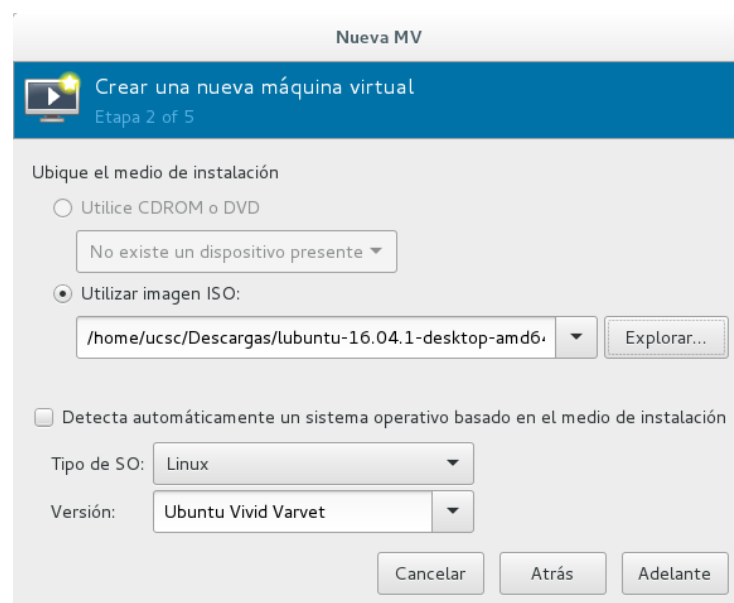


Figura 4.65: Seleccionar el sistema operativo, virt-manager (Fuente Propia).

- Se asignó la memoria RAM para la máquina virtual y la cantidad de CPUs virtuales. Se estableció en la planificación del entorno virtual en la sección 2 que una máquina virtual tendría 2 GB de RAM y 1 CPU asignada, procediendo con aquello se seleccionó *Adelante*, ver Figura 4.66.



Figura 4.66: Asignación de memoria RAM y cantidad CPUs virtuales con virt-manager (Fuente Propia).

- Se eligió el espacio de almacenamiento y en donde se instaló la máquina virtual, se seleccionó donde dice *Elija Administrado*, ver Figura 4.67.



Figura 4.67: Asignar espacio de almacenamiento con virt-manager (Fuente Propia).

9. A la izquierda se seleccionó el grupo de almacenamiento *images* y se presionó el botón + y se agregó un nuevo espacio de almacenamiento, ver Figura 4.68.

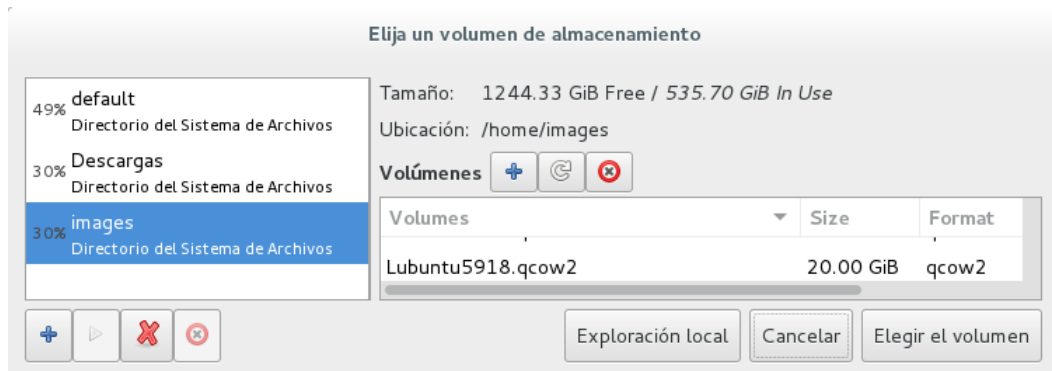


Figura 4.68: Seleccionar grupo de almacenamiento con virt-manager (Fuente Propia).

10. Se debió escribir el nombre del espacio de almacenamiento, se puede nombrar como quiera, pero en este caso se nombró Lubuntu, ver Figura 4.69.

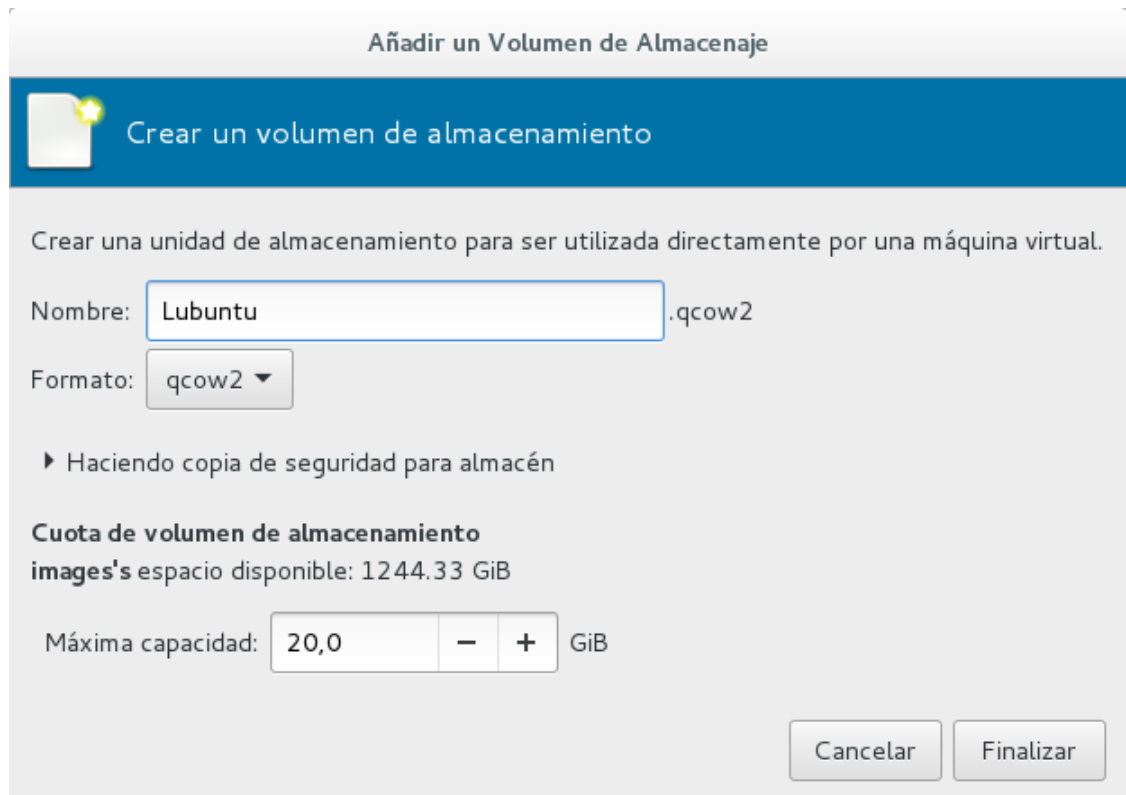


Figura 4.69: Escribir nombre de espacio de almacenamiento con virt-manager (Fuente Propia).

11. Una vez creado el espacio de almacenamiento se apretó el botón *Elegir Volumen*, ver Figura 4.70.

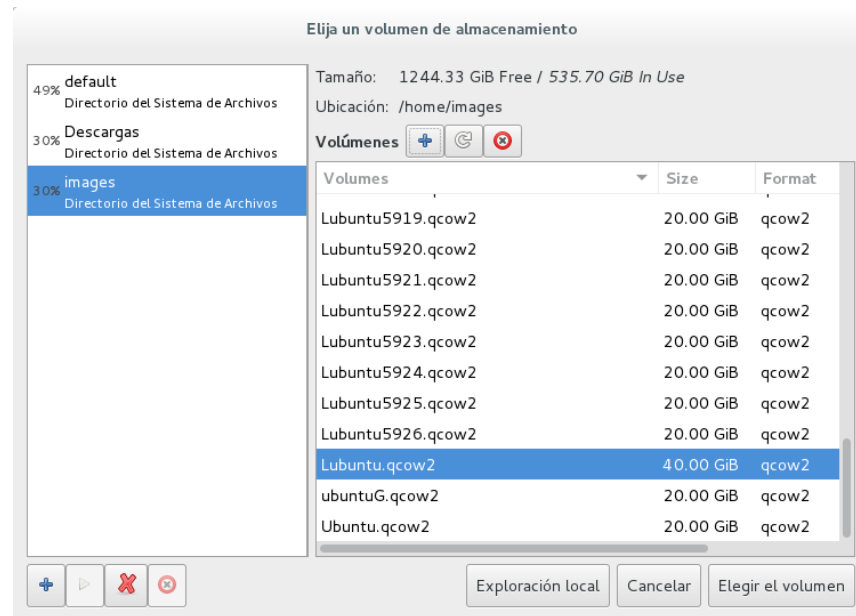


Figura 4.70: Seleccionar el volumen de almacenamiento con virt-manager (Fuente Propia).

12. Una vez creado el espacio de almacenamiento y seleccionado apareció la ruta de acceso, se presionó el botón *Adelante*, ver Figura 4.71.

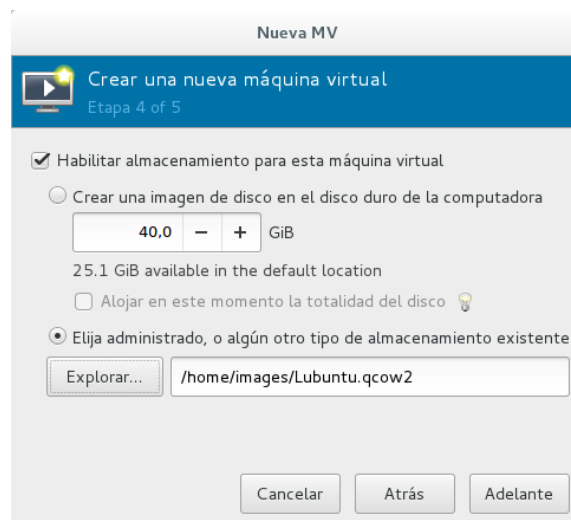


Figura 4.71: Dirección de volumen de almacenamiento con virt-manager (Fuente Propia).

13. Se seleccionó *Personalizar configuración antes de instalar*, además se seleccionó una red NAT que esta por default y se presionó el botón *Finalizar*, ver Figura 4.72.

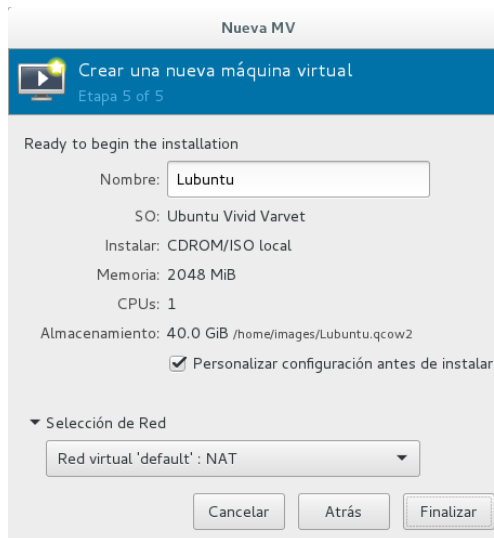


Figura 4.72: Seleccionar red NAT para nueva máquina virtual con virt-manager (Fuente Propia).

14. Antes de finalizar se configuro el software de vídeo. Se seleccionó a la izquierda *Video VGA* y entre las opciones que aparecieron se escogió *QXL*, se presionó botón *Aplicar* y finalmente presionó el tick verde que dice *Iniciar Instalación*, ver Figura 4.73.

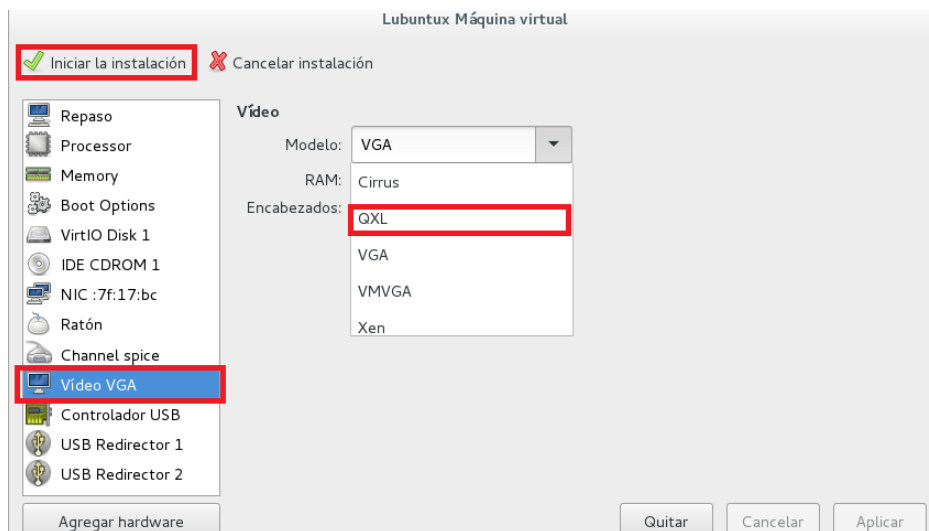


Figura 4.73: Configurar video QXL con virt-manager (Fuente Propia).

15. Se inicio la máquina virtual con la distribución Lubuntu, se procedió con su instalación hasta finalizar, ver Figura 4.74.

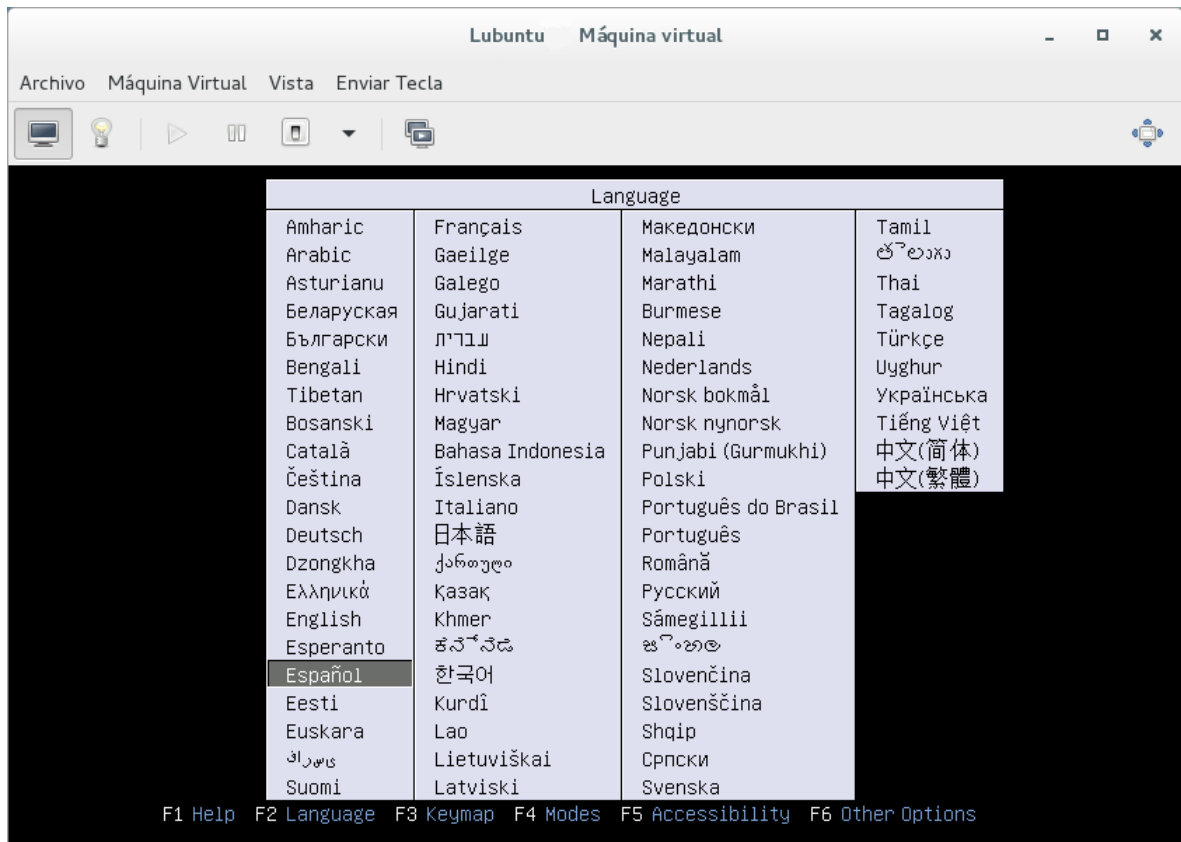


Figura 4.74: Iniciar e instalar máquina virtual Lubuntu con virt-manager (Fuente Pro-pia).

### 4.6.2. Clonación de máquinas virtuales

En esta sección se ocupó la primera máquina virtual instalada en la sección anterior para hacer copias de ellas, hasta obtener las 24 máquinas que se necesitan instalar de acuerdo con la planificación del proyecto en la sección 2.

1. Primero se debió abrir virt-manager, ver Figura 4.75.

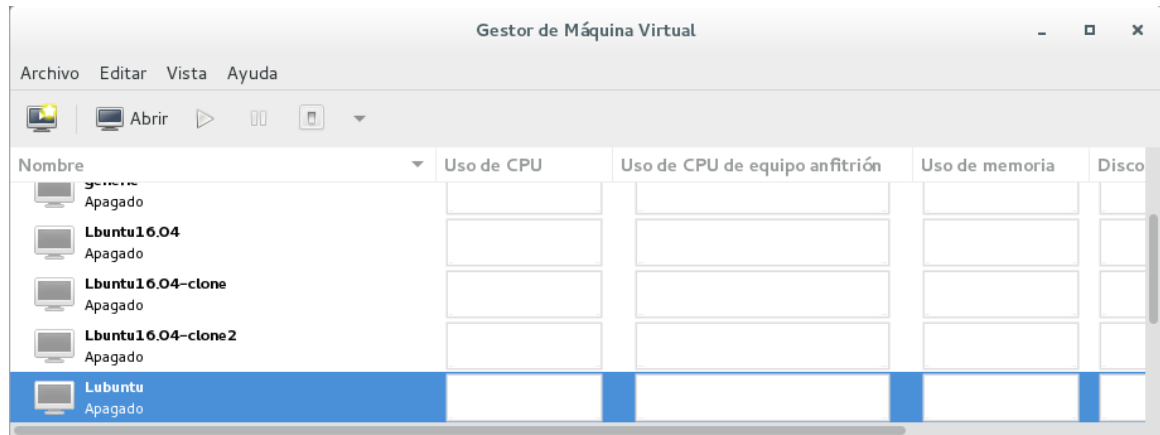


Figura 4.75: Abrir virt-manager.

2. Se hizo click derecho sobre la máquina virtual Lubuntu creada en la sección anterior y se presionó *Clonar*, ver Figura 4.76.

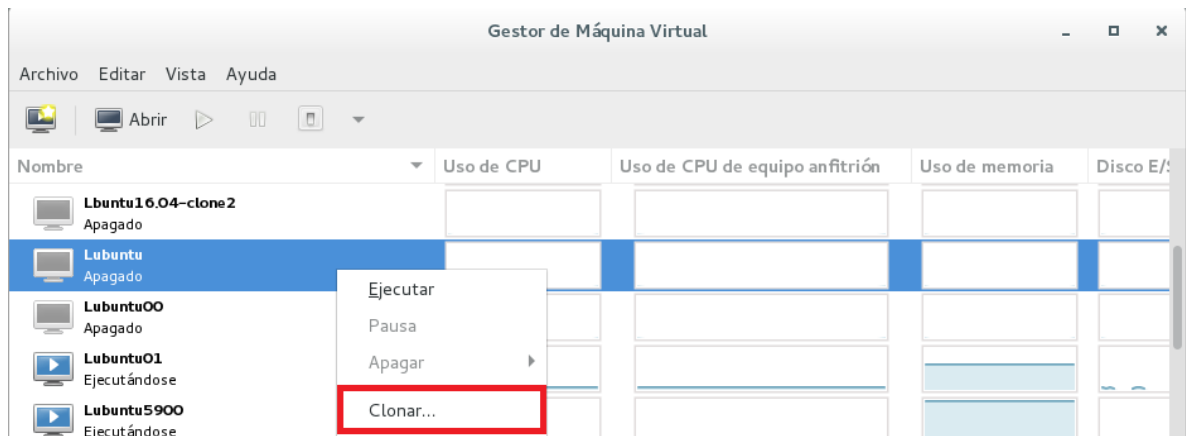


Figura 4.76: Opción de clonación de máquina virtual con virt-manager (Fuente Propia).

3. Se abrió una ventana en donde se puso nombre a la nueva máquina virtual y también se asignó espacio de almacenamiento, una vez hecho esto se presiono el botón *Clonar*, ver Figura 4.77.

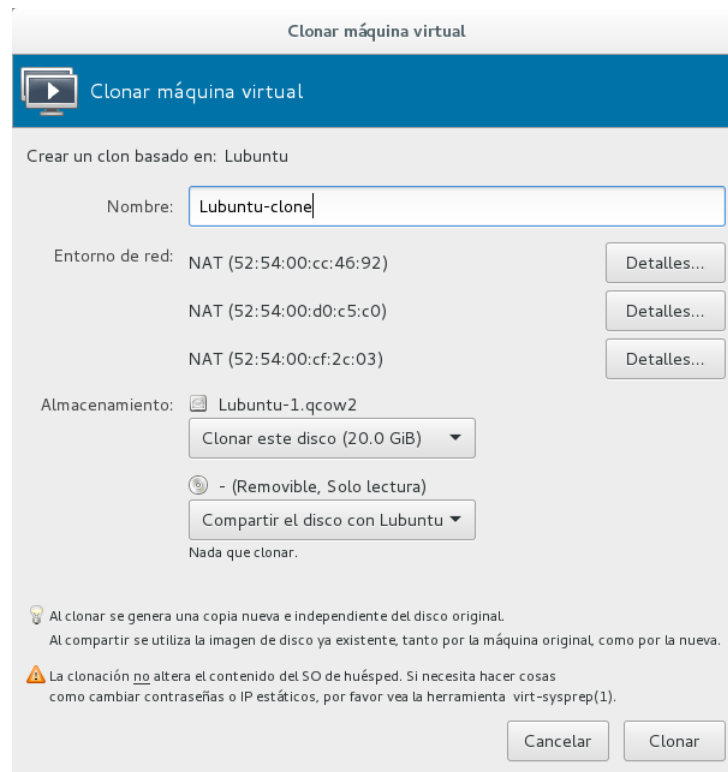


Figura 4.77: Nombrar nueva máquina clonada con virt-manager (Fuente Propia).

4. Finalmente se creó la máquina virtual, ver Figura 4.78.

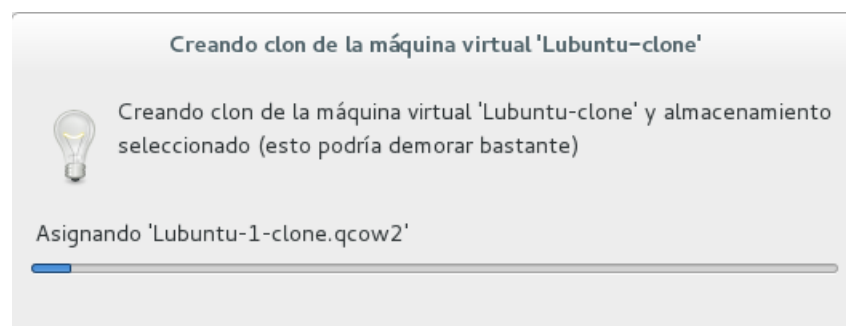
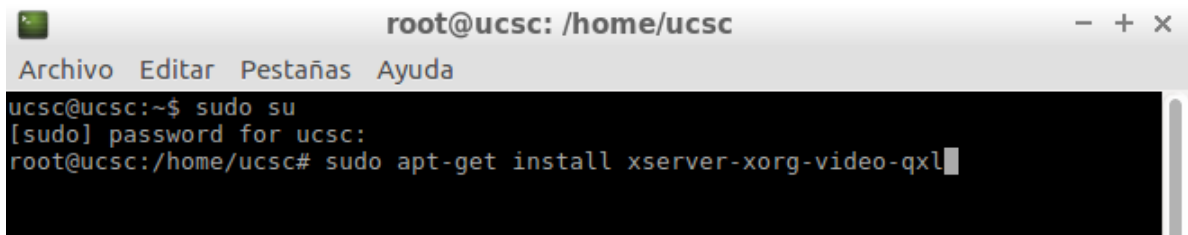


Figura 4.78: Virt Manager creando clon de máquina virtual (Fuente Propia).

### 4.6.3. Instalación Vídeo QXL driver y SPICE VDAgent en máquina virtual

Después de la instalación de una máquina virtual, se instalaron varios controladores y demonios en el invitado para proporcionar una integración mejorada y un rendimiento óptimo de SPICE.

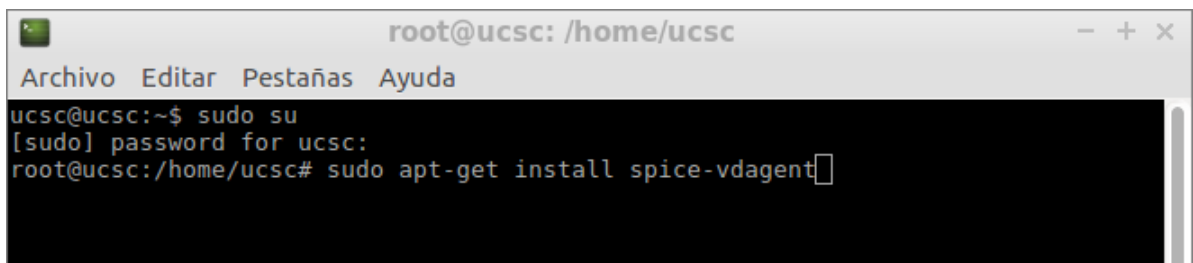
Para la instalación de QXL driver, se debió iniciar la consola de Lubuntu y escribir lo siguiente ver, Figura 4.79.



```
root@ucsc: /home/ucsc
Archivo Editar Pestañas Ayuda
ucsc@ucsc:~$ sudo su
[sudo] password for ucsc:
root@ucsc:/home/ucsc# sudo apt-get install xserver-xorg-video-qxl
```

Figura 4.79: Instalar QXL driver en linea comando Lubuntu (Fuente Propia).

Para instalar SPICE VDAgent, con la consola de Lubuntu se escribio lo siguiente ver, Figura 4.80.



```
root@ucsc: /home/ucsc
Archivo Editar Pestañas Ayuda
ucsc@ucsc:~$ sudo su
[sudo] password for ucsc:
root@ucsc:/home/ucsc# sudo apt-get install spice-vdagent
```

Figura 4.80: Instalar QXL driver en linea comando Lubuntu (Fuente Propia).

#### 4.6.4. Instalación Virt-viewer e ingreso de forma remota a una máquina virtual

Para conectarse a una máquina virtual usando SPICE, se necesita instalar una aplicación en la máquina cliente (Spice, 2016). Esta aplicación se encuentra en el sitio web de SPICE, en la sección de descargas. Se debió descargar la aplicación cliente *virt-viewer* apropiada de acuerdo al sistema operativo instalado en la máquina cliente. Sitio de descargas, ver Figura 4.81.

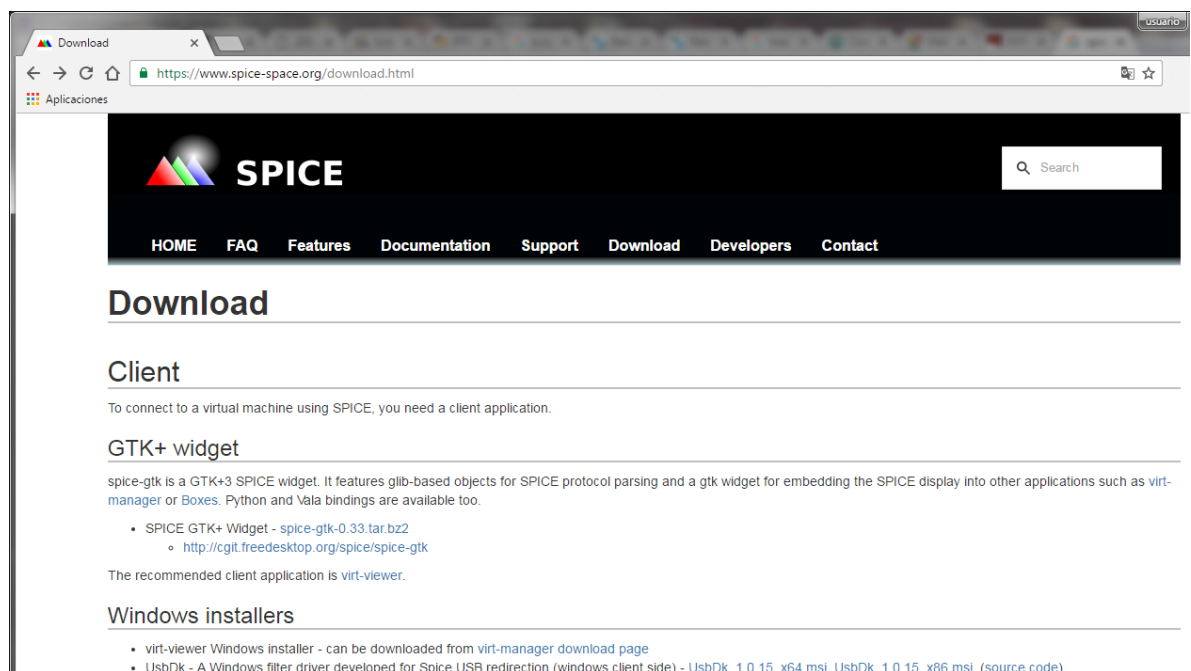


Figura 4.81: Sección descargas, página oficial de SPICE (Fuente Propia).

Una vez instalado *virt-viewer*, se abrió la aplicación y se accedió a la máquina virtual, indicando primero el protocolo, la dirección IP y por último el puerto donde conectarse.

En este ejemplo se accedió a una máquina virtual en el puerto 5999, por medio de la IP pública del servidor, ver Figura 4.82.

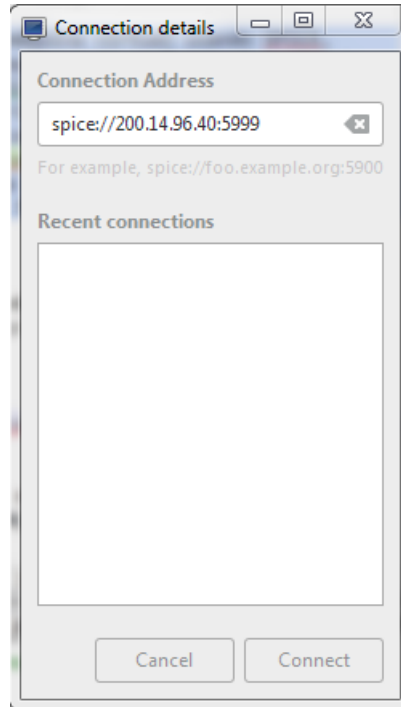


Figura 4.82: Conectarse a una máquina virtual por medio de virt-viewer (Fuente Propia).

Se escribió la contraseña correspondiente a la máquina virtual, proporcionada por el administrador, ver Figura 4.83.

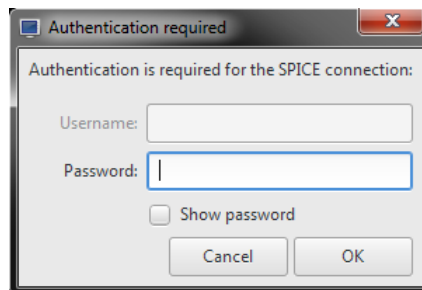


Figura 4.83: Escribir contraseña de máquina virtual en virt-viewer (Fuente Propia).

Después se pudo acceder y ocupar la máquina virtual ver, Figura 4.84.



Figura 4.84: Máquina virtual Lubuntu accedida mediante virt-viewer (Fuente Propia).

#### 4.6.5. Conectar dispositivo de almacenamiento USB de forma remota utilizando Virt-viewer

Para conectar un dispositivo de almacenamiento USB de forma remota, se abrió *virt-viewer* y se conecto a una máquina virtual. Luego desde el menú de *virt-viewer* se accedió a *File->USB device selection*, ver Figura 4.85.

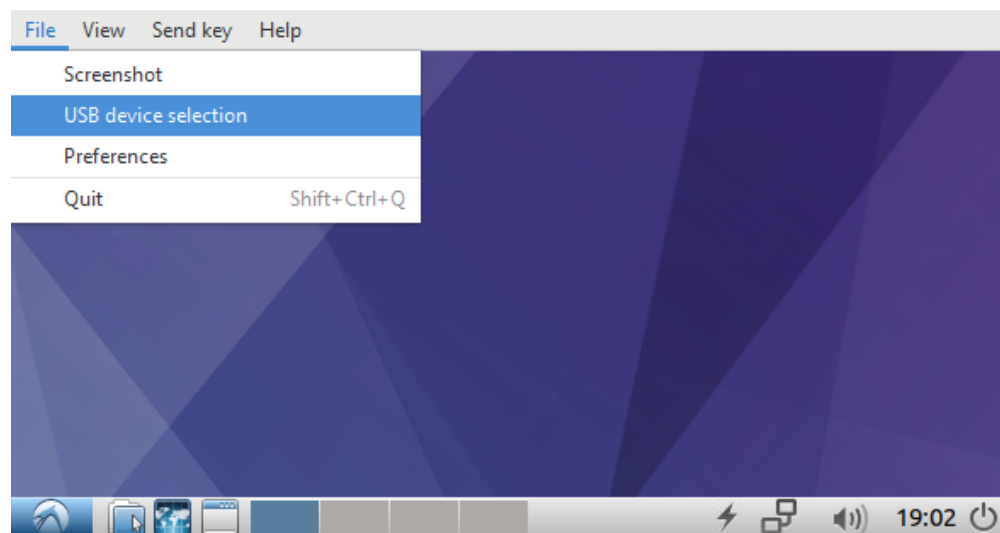


Figura 4.85: USB device selection desde el menú de virt-viewer (Fuente Propia).

Después se escogió el dispositivo USB que conectara de forma remota a la máquina virtual, ver Figura 4.86.

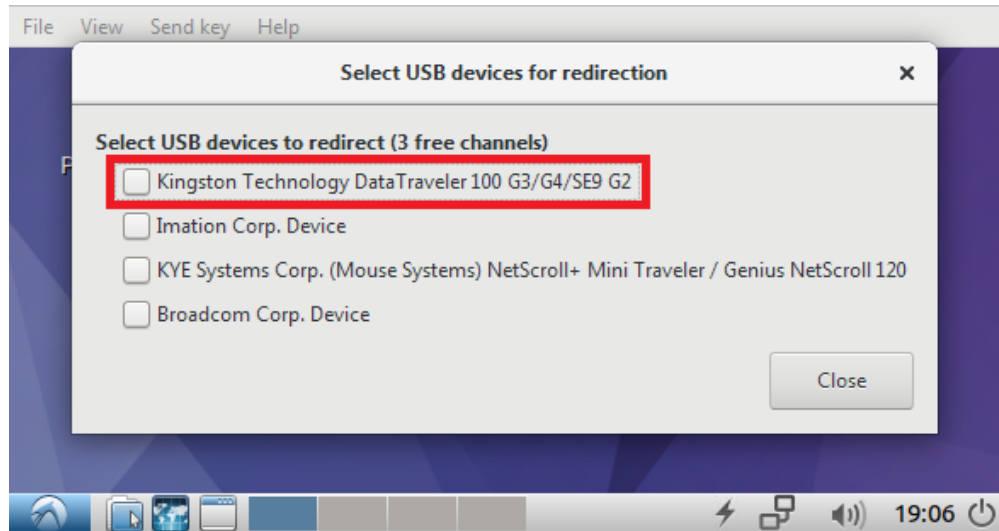


Figura 4.86: Selección de dispositivo USB a conectar de forma remota con virt-viewer (Fuente Propia).

# Capítulo 5

## Validación

### 5.1. Monitorización del entorno virtual

Ahora que el entorno virtual esta configurado y poblado, se tendrá que vigilar el rendimiento para ver si existen problemas. Por lo tanto, se deberá configurar la supervisión en el entorno y comprender los datos que se están midiendo para eliminar los cuellos de botella y problemas. Para esto deben instalar herramientas de supervisión (Siebert, 2009).

Es hora de poner a prueba el servidor. De tal manera de verificar su correcto funcionamiento. Para esto el servidor de máquinas virtuales sera probado en 2 cursos, el primero *Sistemas Distribuidos* que necesita 6 máquinas virtuales operativas y *Redes de Datos* que necesitara 20 máquinas. Se hará una prueba de 1 mes de uso y las mediciones se harán a toda hora en el servidor. Para las pruebas se instalara Munin, herramienta para el monitorio de servidores.

#### 5.1.1. Herramienta de monitoreo Munin

Munin es una herramienta de monitorización de recursos en red que ayuda a analizar las tendencias de recursos y problemas. Está diseñado para ser Plug and Play. Una instalación predeterminada proporciona una gran cantidad de gráficos con un trabajo de configuración mínimo (RoseHosting, 2016).

Munin monitorea varios recursos del sistema, pero en este documento se centrara en el uso de la CPU, memoria RAM, uso de la red y el uso de disco.



Figura 5.1: Logo Munin (Munin, 2016).

### 5.1.2. Instalación de Munin

1. Se accedió al servidor via SSH en modo ROOT.

```
# ssh root@server_ip
```

2. Se introdujo el comando siguiente para comprobar si tiene la versión correcta de CentOS instalado en su máquina.

```
# cat /etc/redhat-release
```

3. Mostró la salida siguiente.

```
CentOS Linux release 7.2.1511 (Core)
```

4. Se verificó que el servidor está totalmente actualizado.

```
# yum update
```

5. Completada la actualización, se instaló el repositorio EPEL.

```
# yum install epel-release
```

6. Se instaló Munin y Apache con el comando.

```
# yum install munin munin-node httpd -y
```

7. Completada la instalación, se permitió que Munin inicie en el arranque.

```
# systemctl start munin-node
```

```
# systemctl enable munin-node
```

8. Se verificó que el archivo `/etc/munin/munin.conf` tiene habilitadas las siguientes líneas, con un editor de texto.

```
[localhost]
address 127.0.0.1
use_node_name yes
```

9. A continuación se utilizó `htpasswd`, utilidad que creó un nombre de usuario y contraseña para la autenticación básica de protección para Munin.

```
# htpasswd /etc/munin/munin-htpasswd admin
```

10. Se solicitó, como se muestra a continuación, la contraseña para el usuario.

```
New password: Re-type new password: Adding password for user admin
```

11. Se añadió el nombre del servidor al archivo `munin-node.conf`, se hizo con un editor de texto.

```
# nano /etc/munin/munin-node.conf
```

12. Se añadió el nombre del host al final del archivo, en este servidor el host es *localhost.localdomain*.

```
host_name vps.rosehosting.com
```

13. Se abrió la configuración Munin para Apache y añadió las líneas siguientes en este archivo `/etc/httpd/conf.d/munin.conf`.

```
# nano /etc/httpd/conf.d/munin.conf
Order Deny,Allow
Deny from all
Allow from 127.0.0.1 your_public_IP
```

14. Se reinició Apache para que los cambios surtan efecto.

```
# systemctl restart httpd
```

### 5.1.3. Monitoreo de uso de disco con Munin

En la sección anterior se vio como se instalaba la herramienta Munin en el servidor, ahora con Munin operativo se analizara el uso de la CPU, memoria RAM, uso de la red y el uso de disco.

### 5.1.4. Como usar Munin

Para entrar a Munin se puede acceder desde cualquier navegador. Escribiendo la dirección IP local o publica. Se escribió la contraseña para el usuario *admin* que es el administrador. Esto se definió en el paso 10 en la instalación de Munin, ver Figura 5.2.

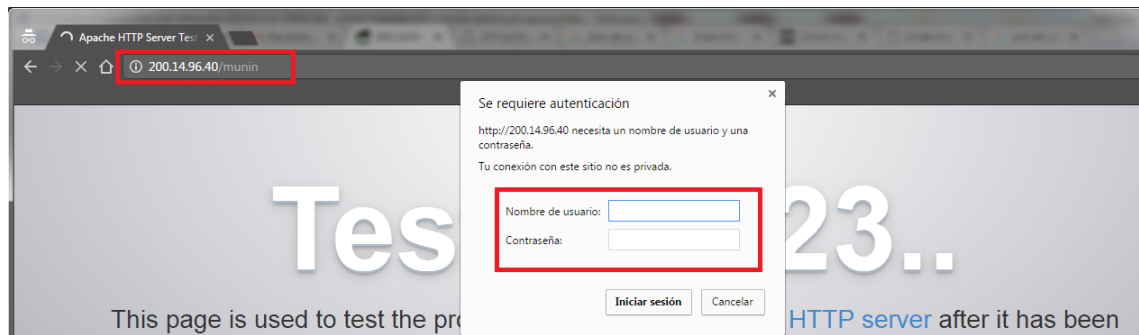


Figura 5.2: Ingresando a Munin desde el navegador (Fuente Propia).

Una vez dentro usted se vio a la izquierda en la sección *problemas* el registro del servidor cuando tiene alguno. También se observaron estadísticas de los recursos del servidor en la sección *categoría*, eligiendo el recurso a monitorear, ver Figura 5.3.

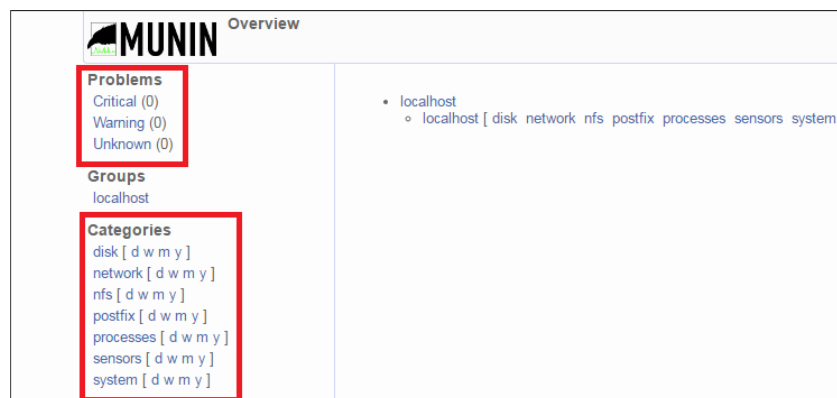


Figura 5.3: Aplicación Munin (Fuente Propia)

#### 5.1.4.1. Monitoreo del uso de CPU con Munin

Para el monitoreo del servidor se centro un periodo de tiempo de 1 mes, para ser claros se medio el mes de octubre del 2016.

El uso de CPU durante el mes de octubre del 2016 se resume en el siguiente gráfico. Este gráfico muestra cómo se gasta el tiempo de CPU, ver Figura 5.4.

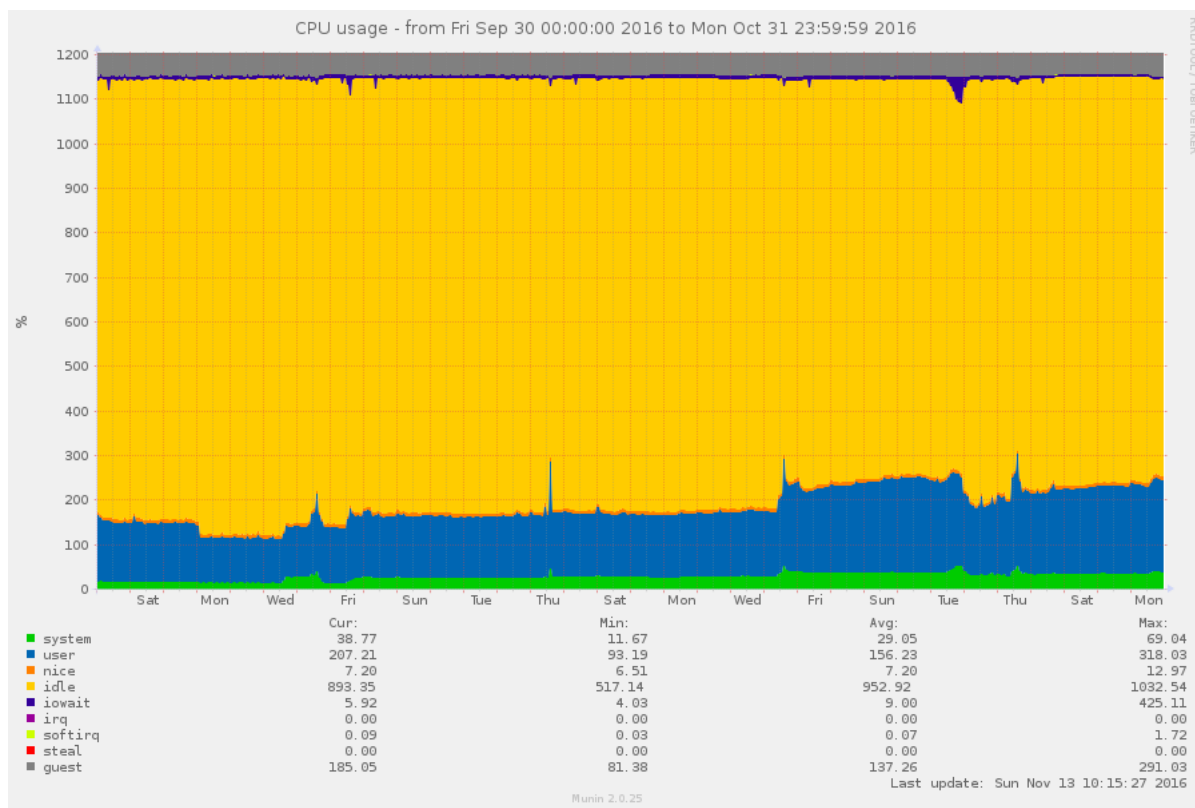


Figura 5.4: Munin, uso de CPU en el mes de Octubre (Fuente Propia).

Observando el gráfico vemos que en el eje de las abscisas se encuentran los días correspondientes al mes de Octubre y en el eje de las ordenadas los núcleos del procesador, el servidor consta de dos procesadores con 6 núcleos cada uno. Cada núcleo se representa con un porcentaje de cien por ciento, por lo tanto habiendo doce núcleos tienen una representación del mil docientos por ciento.

Analizando el gráfico, se ve de color amarillo, representando la mayoría del porcentaje a la variable *idle* que corresponde al tiempo de inactividad de la CPU, de esto se puede decir el servidor no presenta problemas en el tiempo de uso de la CPU ya que los procesadores están la mayoría del tiempo sin uso.

De color azul tenemos a la variable **user** que corresponde a los procesos de usuario y este se define como el tiempo de CPU gastado por programas y demonios normales. Los procesos de usuario son los que mas trabajan en el servidor llegando a mostrar topes que bordean el 320%, en algunas ocasiones esto se puede explicar ya que en los días jueves los estudiantes ocupan las máquinas todos al mismo tiempo en los laboratorios de la universidad y justamente es ahí donde se llega a tales topes.

En la siguiente imagen Figura 5.5, ocupando el comando **top** se puede observar que los procesos de usuario son los que ocupan los primeros lugares.

```

ucsc@localhost:/home/ucsc
Archivo Editar Ver Buscar Terminal Ayuda
top - 14:45:18 up 3 days, 15:01, 4 users, load average: 1,39, 1,80, 1,70
Tasks: 423 total, 3 running, 420 sleeping, 0 stopped, 0 zombie
%Cpu(s): 13,2 us, 3,7 sy, 0,0 ni, 82,3 id, 0,8 wa, 0,0 hi, 0,0 si, 0,0 st
KiB Mem : 65546820 total, 269628 free, 46531224 used, 18745968 buff/cache
KiB Swap: 32899068 total, 32527204 free, 371864 used, 18393396 avail Mem

  PID USER      PR  NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 3225 qemu      20   0 2925924 2,074g 11424 S  49,8  3,3   2322:22 qemu-kvm
 6915 qemu      20   0 2740880 2,082g 11460 S  38,4  3,3   1394:58 qemu-kvm
22449 munin    20   0 290612 21304 2432 R  14,4  0,0    0:02.12 /usr/share+
 8678 ucsc      20   0 1572048 313992 26340 S  12,8  0,5   527:50.29 virt-manag+
14575 qemu      20   0 2807084 2,114g 11440 S  10,2  3,4   556:23.62 qemu-kvm
 4681 ucsc      20   0 1908000 263808 43776 S   6,9  0,4   69:44.10 gnome-shell
 1741 root      20   0 1221996 36852 13472 S   4,9  0,1   207:41.04 libvirtd
  917 root      20   0   4376    600    512 S   4,6  0,0    28:57.56 rngd
 1678 qemu      20   0 2758936 1,304g 11344 S   2,3  2,1    98:48.72 qemu-kvm
 1962 qemu      20   0 2735844 1,299g 11856 S   2,3  2,1    55:31.54 qemu-kvm
 2271 qemu      20   0 2775324 2,056g 11416 S   2,3  3,3   153:48.15 qemu-kvm
14915 qemu      20   0 2767128 1,404g 11396 S   2,3  2,2   101:45.45 qemu-kvm
15185 qemu      20   0 2736404 1,354g 11276 S   2,3  2,2   100:26.89 qemu-kvm
1117 qemu     20   0 2847072 2,178g 11492 R   2,0  3,5   117:00.53 qemu-kvm
 2536 qemu      20   0 2757912 1,354g 11348 S   2,0  2,2    99:11.08 qemu-kvm
 7232 qemu      20   0 2736412 1,367g 11364 S   2,0  2,2    97:11.50 qemu-kvm
 9775 qemu      20   0 2735384 1,305g 11348 S   2,0  2,1   102:23.73 qemu-kvm

```

Figura 5.5: Uso de comando top (Fuente Propia).

Podemos ver al usuario **QEMU** utilizando el proceso *qemu-kvm* que se repite constantemente, pues el servidor utiliza este proceso para emular las máquinas virtuales.

Después tenemos de color plomo a la variable **guest** que corresponde al tiempo dedicado a la ejecución de un CPU virtual para sistemas operativos invitados bajo el control del kernel de Linux. Esta variable se mantiene constante pues como lo dice es el tiempo que se requiere para emular un CPU virtual para la ejecución de las máquinas virtuales, depende mas que nada de la capacidad del procesador. Ocupa un porcentaje del 100%, se podría decir que ocupa un núcleo para tener a todas las máquinas operativas.

Por ultimo tenemos de color verde a la variable **system** que corresponde al tiempo de CPU empleado por el núcleo a las actividades del sistema del servidor, ocupa un porcentaje de 50% teniendo a un núcleo del procesador a mitad de uso.

En conclusión se puede decir que el tiempo de gasto de CPU es bueno, ya que en el análisis la variable *idle* que representa el tiempo inactividad de la CPU tiene un promedio del 950% y un mínimo de 517% o sea en todo momento existe tiempo de inactividad en el uso de la CPU, incluso en los momentos que tiene mayor actividad el servidor.

#### 5.1.4.2. Monitoreo del uso de disco con Munin

El uso de disco durante el mes de octubre del 2016 se midió en términos de la utilización del dispositivo de almacenamiento. El servidor dispone de dos unidades de almacenamiento sda y sdb.

Este gráfico muestra la utilización del dispositivo *sda* en porcentaje, ver Figura 5.6.

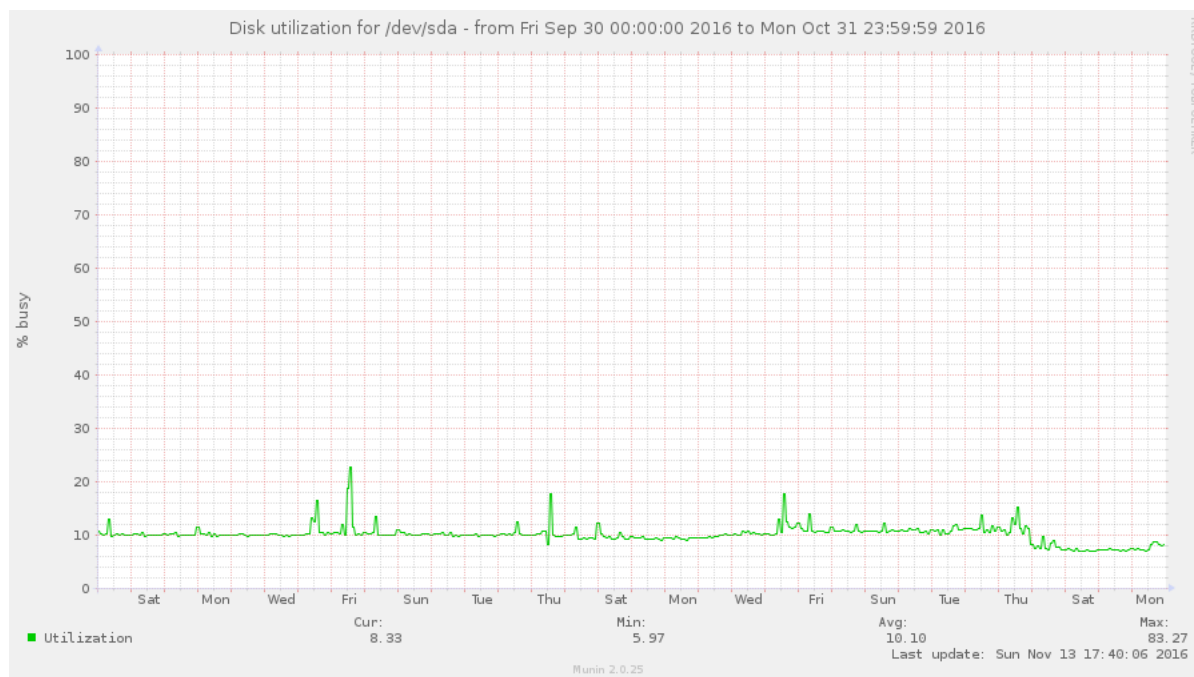


Figura 5.6: Tiempo de saturación para disco sda (Fuente Propia).

Observando el gráfico vemos que en el eje de las abscisas se encuentran los días correspondientes al mes de Octubre y en el eje de las ordenadas el tiempo de saturación del disco en términos de porcentaje.

La variable *utilization* de color verde significa que si el tiempo necesario para la E/S está cerca de 1000 msec para un determinado segundo, el aparato es prácticamente del 100% saturado.

Se puede observar que la saturación llega hasta un máximo del 21 % y presenta los mayores niveles de saturación los días jueves, ya que el curso que opera las máquinas virtuales ocupa ese día para trabajar en los laboratorios de la universidad. En promedio el porcentaje de saturación durante el mes de Octubre es del 10 % lo cual es un buen resultado.

El gráfico correspondiente a la unidad de almacenamiento, ver Figura 5.7.

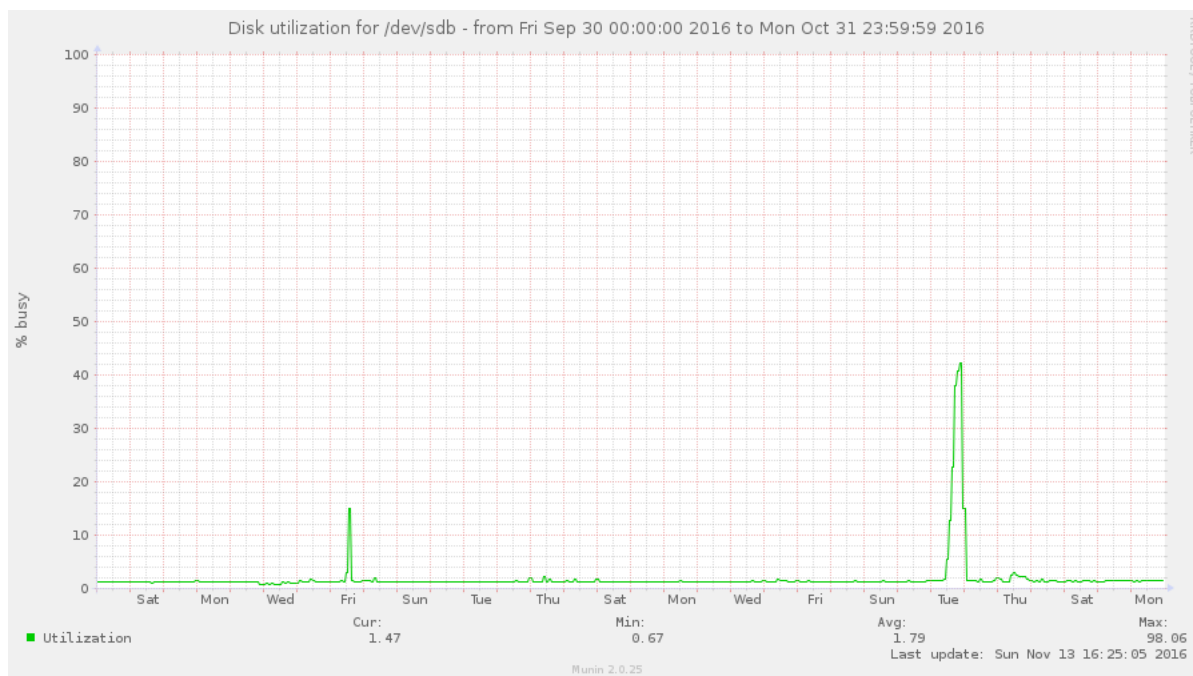


Figura 5.7: Tiempo de saturación para disco sdb (Fuente Propia).

Observando el gráfico vemos que en el eje de las abscisas se encuentran los días correspondientes al mes de Octubre y en el eje de las ordenadas el tiempo de saturación del disco en términos de porcentaje.

La variable *utilization* de color verde significa que si el tiempo necesario para la E/S está cerca de 1000 msec para un determinado segundo, el aparato es prácticamente del 100 % saturado.

En el disco *sdb* la situación de saturación alcanza un tope del 41 % bastante mas considerable que el disco *sda*, aun así esto no representa un valor crítico. En promedio la saturación es del 2 %, lo cual es bueno.

De la misma forma que el disco anterior, este gráfico muestra el tiempo de espera promedio y latencia para las diferentes categorías de operaciones del disco *sdb*. Los tiempos que incluyen los tiempos de cola indican cómo esta ocupado el sistema. Si el tiempo de espera llega a 1 segundo el sistema de E/S esta 100 % ocupado, ver Figura 5.8.

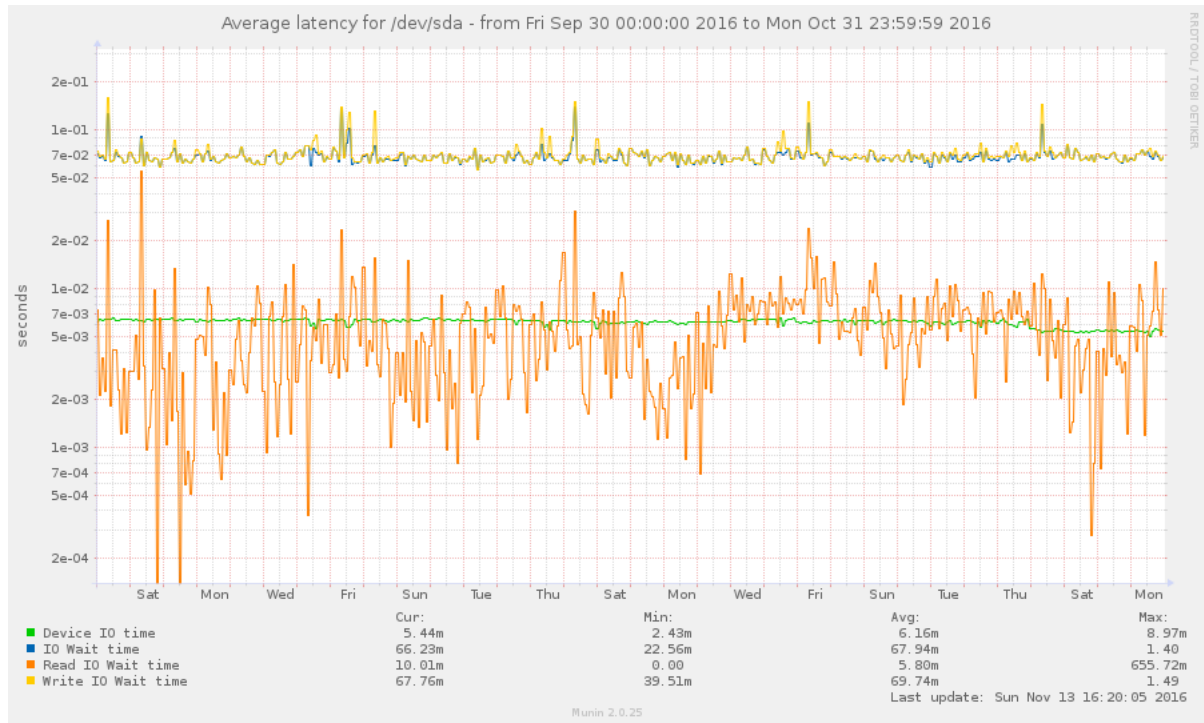


Figura 5.8: Gráfico saturación sda (Fuente Propia).

De color verde tenemos a la variable *Device IO TIME* que corresponde al tiempo medio en que una operación de E/S toma en el dispositivo de bloque sin incluir los tiempos de cola, sólo el tiempo de ida y vuelta para la solicitud de disco. Este tiempo para todo el mes de Octubre esta por debajo de los 0.01 segundos por lo cual el tiempo esta bastante bien (valores cercanos 1 son malos).

En color azul tenemos a la variable *IO Wait time* que corresponde al tiempo medio de espera para una operación de E/S desde el inicio de la solicitud (incluidos tiempos de cola). Mirando el gráfico el tiempo transcurrido para el mes de Octubre esta por debajo de los 0.05 segundos por lo tanto también es un buen valor (valores cercanos a 1 son malos).

De color naranja esta la variable *Read IO Wait time* que corresponde al tiempo medio de espera para una operación de E/S pero de lectura, desde el inicio de la solicitud hasta el final (incluye tiempos de cola y otros). Mirando el gráfico el tiempo esta muy por debajo de 0.01 segundos que es el pico mas alto visible, lo cual es bueno.

En color amarillo esta la variable *Write IO Wait Time* que es el tiempo promedio de espera para una operación de E/S pero de escritura desde el inicio de solicitud (incluye tiempos de cola y otros), en general el tiempo durante el mes de Octubre esta muy por debajo de los 0.05 segundos que es el tope mas alto, lo cual es bueno.

Este gráfico muestra el tiempo de espera promedio y latencia para las diferentes categorías de operaciones del disco *sdb*, ver Figura 5.9.

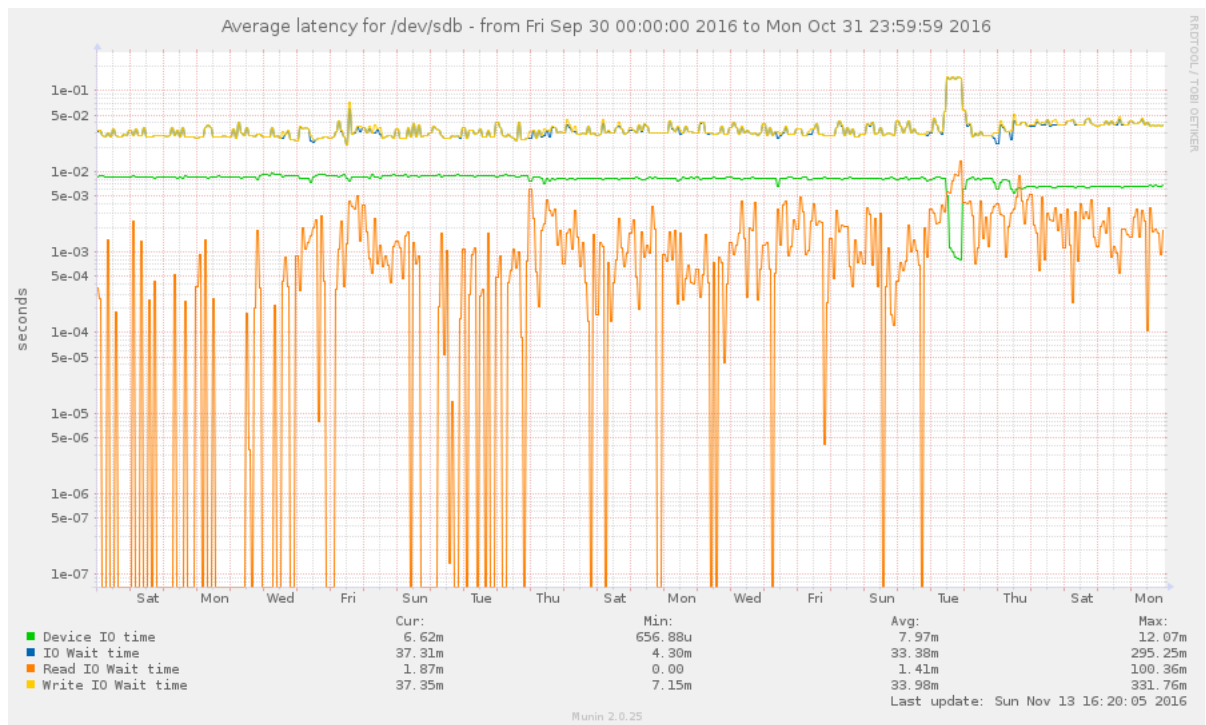


Figura 5.9: Gráfico saturación sdb (Fuente Propia).

De color verde tenemos a la variable *Device IO TIME* que corresponde al tiempo medio en que una operación de E/S toma en el dispositivo de bloque sin incluir los tiempos de cola, sólo el tiempo de ida y vuelta para la solicitud de disco. Este tiempo para todo el mes de Octubre esta por debajo de los 0.007 segundos por lo cual el tiempo

esta bastante bien(valores cercanos 1 son malos).

En color azul tenemos a la variable ***IO Wait time*** que corresponde al tiempo medio de espera para una operación de E/S desde el inicio de la solicitud (incluidos tiempos de cola). Mirando el gráfico el tiempo transcurrido para el mes de Octubre esta por debajo de los 0.2 segundos por lo tanto también es una buen valor(valores cercanos a 1 son malos).

De color naranja esta la variable ***Read IO Wait time*** que corresponde al tiempo medio de espera para una operación de E/S pero de lectura, desde el inicio de la solicitud hasta el final(incluye tiempos de cola y otros). Mirando el gráfico el tiempo esta muy por debajo de 0.07 segundos que es el pico mas alto visible, lo cual es bueno.

En color amarillo esta la variable ***Write IO Wait Time*** que es el tiempo promedio de espera para una operación de E/S pero de escritura desde el inicio de solicitud (incluye tiempos de cola y otros), en general el tiempo durante el mes de Octubre esta muy por debajo de los 0.02 segundos que es el pico mas alto, lo cual es bueno.

### 5.1.4.3. Monitoreo del uso de red con Munin

Ahora corresponde monitorear el uso de la red del servidor. Se dispuso a observar el mes de Octubre.

Este gráfico muestra el tráfico de la interfaz de red *bond0*. Tenga en cuenta que el tráfico en el gráfico se muestra en bits por segundo y no en bytes por segundo. Cuando es negativo (-) significa que los datos entran y cuando es positivo (+) que los datos salen, ver Figura 5.10.

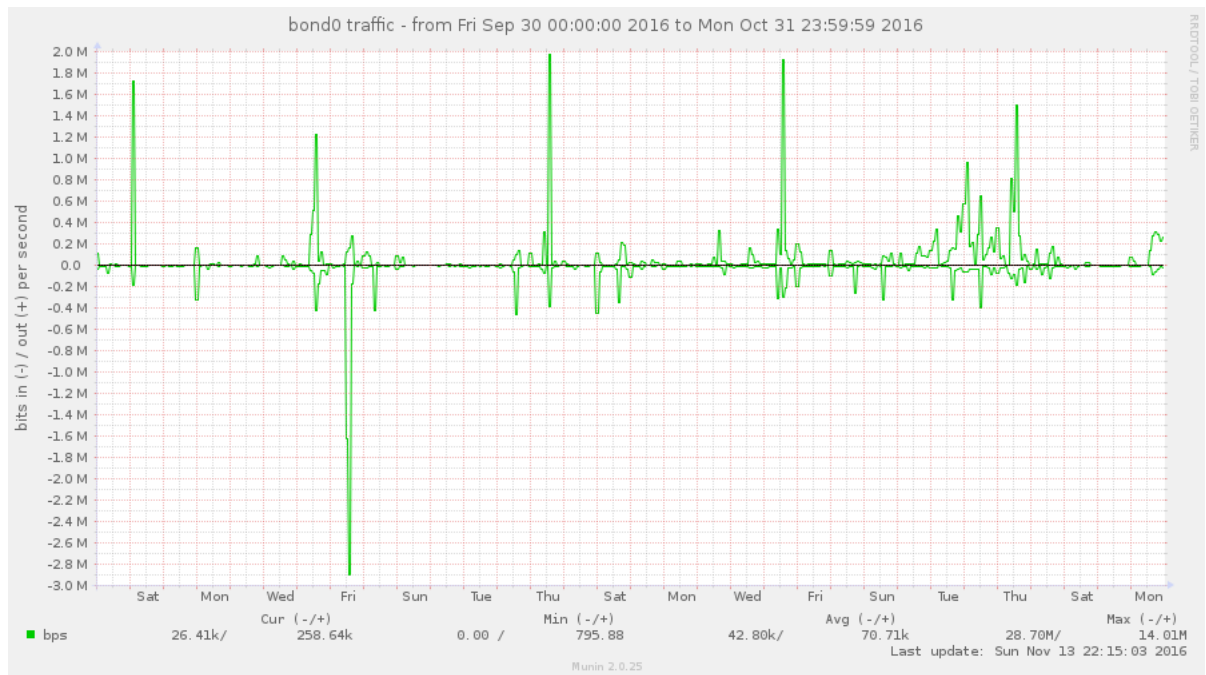


Figura 5.10: Tráfico de red en bond0 (Fuente Propia).

Mirando el gráfico se observa a la variable de color verde **bps** la cual representa el tráfico de la interfaz *bond0*. Su velocidad máxima es de 1000 Mb/s (bits por segundo).

Se puede observar que el tráfico de red tiene topes entre -28.70Mb de entrada y 14Mb de salida. Los topes de entrada y salida se marcan los días jueves por la mayor demanda de máquinas virtuales en los laboratorios. En efecto se puede analizar el tráfico dado la velocidad de la red de 1000 Mb/s y la situación de los topes entre -28.70Mb y 14Mb que el tráfico de la red es normal y buena para la interfaz *bond0*.

#### 5.1.4.4. Monitoreo del uso de memoria RAM con Munin

Ahora corresponde monitorear el uso de la memoria RAM en el servidor. También se observó el mes de Octubre del 2016.

Este gráfico muestra para que se utiliza la memoria RAM en el servidor, ver Figura 5.11.

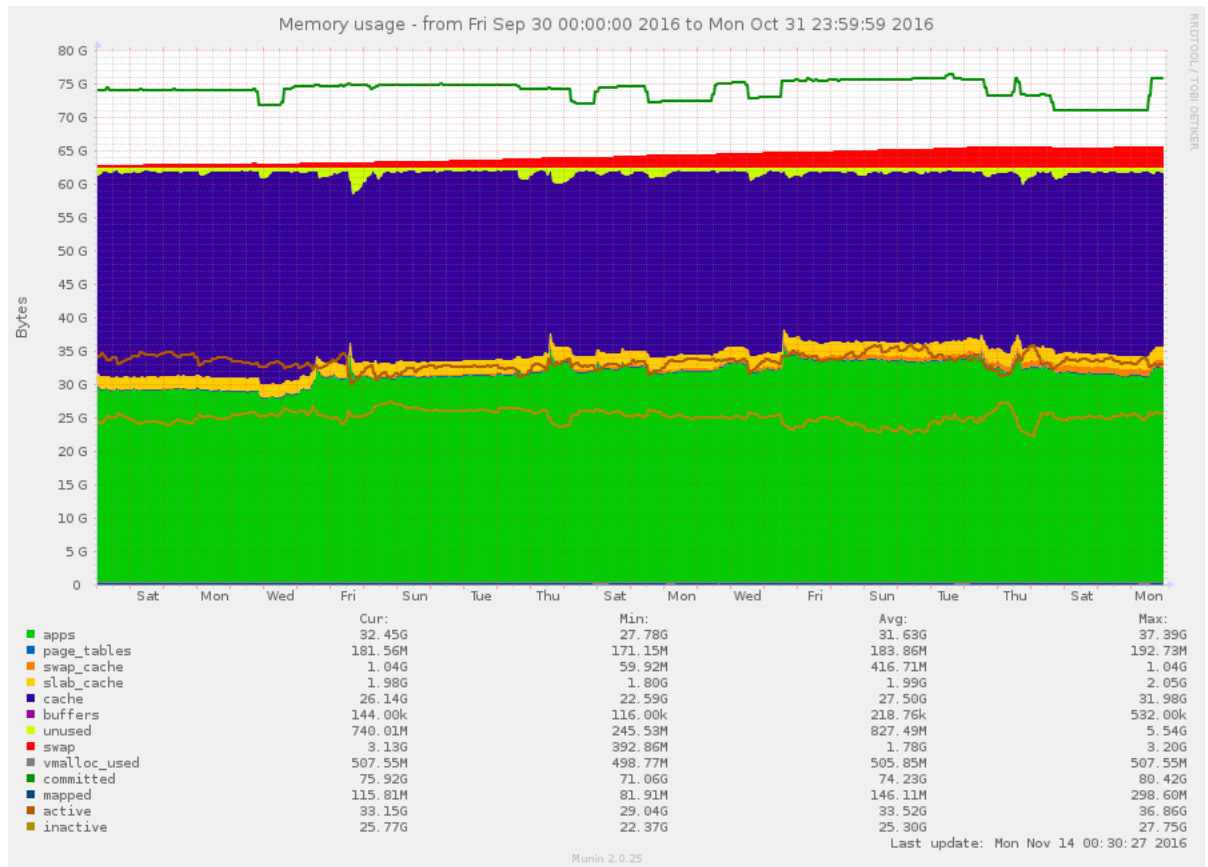


Figura 5.11: Uso de memoria RAM del servidor (Fuente Propia).

En color verde tenemos a **apps** que corresponde a la memoria RAM que utilizan las aplicaciones de usuario. Como vimos en el monitoreo del uso de CPU, el mayor porcentaje de uso de CPU corresponde a procesos de usuario específicamente a **qemu-kvm** por lo tanto el resultado era esperable. Ocupan 30GB de memoria RAM las apps aproximadamente.

En color lila oscuro tenemos a **cache** que corresponde a la memoria cache, que se define como la memoria de acceso rápido del servidor. Sirve para acelerar el intercambio de datos.

El servidor ocupa 30GB aproximadamente de memoria para el uso de cache.

De color amarillo tenemos a ***slab\_cache*** que corresponde a la memoria que es utilizada por el kernel. El kernel ocupa 2.5 GB de RAM en el servidor aproximadamente.

En color rojo esta ***swap*** que corresponde a la memoria *swap* y esta corresponde a una partición en el disco que se necesita para manejar la memoria “virtual” que no cabe en nuestra memoria RAM “física” y que no estamos utilizando en ese instante. En la instalación del servidor se designo 34GB para la memoria *swap*, si se hace uso de esta memoria en el servidor la memoria RAM estaría completamente siendo utilizada. El uso de la memoria *swap* en el servidor es de aproximadamente 3.2GB en periodos largos de 9 días observando el gráfico. Lo ideal es que nunca se utilice la memoria *swap*, por lo tanto se puede concluir que el uso de la memoria RAM esta saturada, se debe aumentar la memoria RAM para mejorar el rendimiento del servidor.

# Capítulo 6

## Conclusiones

En la implementación del servidor la ruta establecida para virtualizar el ambiente físico la constituyó la referencia bibliográfica, *VMware VI3 Implementation and Administration*, método propuesto por *Eric Siebert*.

En el proyecto se sentó una meta “*Implementar una servidor de máquinas virtuales*”, la cual se lleva a cabo con el cumplimiento de objetivos específicos.

### Los resultados de cada objetivo:

El primer objetivo del proyecto fue “*Estudiar la virtualización de plataformas computacionales, herramientas de emulación y estudiar un administrador de conexión a escritorios virtuales remotos*”. Objetivo el cual se cumple en el Capítulo 2 y 3 del documento, correspondiendo al *marco teórico y estado del arte*. Información utilizada como insumo en el Capítulo de *desarrollo*.

El segundo objetivo del proyecto fue “*Definir los requisitos del sistema*”, este objetivo se presenta en el desarrollo del proyecto en el Capítulo 4, en la sección 1 y 2 respectivamente. Correspondiendo a la “*Evaluación el entorno actual*” y la “*Planificación el entorno virtual*”.

Conclusiones importantes a mencionar que se desprenden del segundo objetivo:

- Para asegurar una virtualización exitosa se debe estudiar el mejor hipervisor y

escoger el que se ajuste mejor a las necesidades requeridas.

- Es bueno realizar un análisis de los recursos computacionales del ambiente físico antes de la virtualización, así se evita un gasto desproporcionado en la adquisición de la máquina servidora.
- Es necesario analizar los componentes de software, programas y aplicaciones a virtualizar con tal de distribuir mejor los recursos en el servidor.

El tercer objetivo del proyecto fue *“Diseñar el servidor de máquinas virtuales”*, este objetivo se discute en el Capítulo 4 en la sección 2 *“Planificación del entorno virtual”*. Se presentaron diagramas UML, diagrama de distribución, casos de uso y componentes, con los cuales se diseñó el servidor.

El cuarto objetivo del proyecto fue la *“Instalación el servidor”*, este objetivo se presenta en el Capítulo 4 en la sección 3, 4, 5 y 6 respectivamente. Dichas secciones corresponden a *“Construcción el entorno virtual”*, *“Configuración el entorno virtual”*, *“Configuración de la seguridad del entorno virtual”* y *“Creación del entorno virtual”*. En resumen lo principal de estas secciones fue la instalación del sistema operativo CentOS 7, la instalación del hipervisor KVM, la instalación de un administrador de conexión a un escritorio virtual específicamente SPICE, la configuración de la red, la seguridad del servidor y la configuración e instalación de máquinas virtuales en el servidor.

El quinto objetivo del proyecto fue *“Testear y monitorear el servidor”*, este objetivo se demuestra en el Capítulo 5 correspondiendo a la *“Validación”*, sección 1 y corresponde a *“Monitorización del entorno virtual”*. En esta sección se instaló una herramienta de monitoreo *Munin*, la cual se utilizó para recopilar datos en un periodo de tiempo de 1 mes (Octubre del 2016). Se monitoreó el uso de CPU, uso de disco, uso de memoria RAM y el uso de la red.

Conclusiones y recomendaciones importantes a mencionar que se desprenden del quinto objetivo:

- Para visualizar y corregir posibles problemas en el entorno virtual es necesario utilizar herramientas de monitoreo que entreguen resultados del rendimiento de

recursos computacionales como memoria RAM, uso de disco, velocidad de la red y uso de CPU.

- El servidor necesita expandir su memoria RAM ya que se detecto que usa recurrentemente la memoria swap, con un máximo de 3.2GB.
- También se recomienda aumentar la velocidad de procesamiento del servidor, ya que cuando las máquinas virtuales trabajan al mismo tiempo se producen cuellos de botella.
- La Universidad podría adquirir un servidor de alta capacidad para ofrecer servicios de virtualización, de forma que los recursos de las máquinas virtuales no sean limitados.
- La Universidad podría disponer de servidores virtuales para que docentes y alumnos puedan por ejemplo montar paginas web o cualquier tipo de servicio requerido.
- Se recomienda a la Universidad invertir en virtualización para ir a la vanguardia y no retrasarse en tecnologías de información.

# Bibliografía

- Cavsi (2016). *¿Qué es Conexión a Escritorio remoto?* URL: <http://www.cavsi.com/preguntasrespuestas/que-es-conexion-a-escritorio-remoto/> (Accedido el 17-11-2016).
- CentOS (2016). *CentOS Linux*. URL: <https://www.centos.org/about/> (Accedido el 19-05-2016).
- Citrix (2016). *Only XenServer*. URL: <https://www.citrix.com/products/xenserver/compare.html> (Accedido el 18-11-2016).
- Collins, T. (2017). *Hyper-V vs. VMware: Which Is Best?* URL: <https://www.atlantech.net/blog/hyper-v-vs.-vmware-which-is-best> (Accedido el 10-04-2017).
- Data Keeper (2016). *¿Qué son los Hipervisores?* URL: <http://www.datakeeper.es/?p=716> (Accedido el 06-07-2016).
- Distrowatch (2016). *Lubuntu*. URL: <https://distrowatch.com/table.php?distribution=lubuntu> (Accedido el 28-05-2016).
- Eclipse (2016). *Notas de Lanzamiento del IDE NetBeans 6.9.1*. URL: [https://netbeans.org/community/releases/69/relnotes\\_es.html](https://netbeans.org/community/releases/69/relnotes_es.html) (Accedido el 13-10-2016).
- Estréllate y Arde (2016). *Unión de interfaces de red (channel bonding) en Linux*. URL: <http://www.estrellateyarde.org/configurar-la-red-en-linux/union-de-interfaces-de-red-channel-bonding-en-linux> (Accedido el 28-05-2016).

- Howtoforge (2016). *How to install Fail2Ban on CentOS 7*. URL: <https://www.howtoforge.com/tutorial/how-to-install-fail2ban-on-centos/> (Accedido el 9-11-2016).
- IT'zGeek (2016). *Install Gnome GUI on CentOS 7 / RHEL 7*. URL: <http://www.itzgeek.com/how-tos/linux/centos-how-tos/install-gnome-gui-on-centos-7-rhel-7.html> (Accedido el 28-05-2016).
- Langer, French T (2011). Virtual machine performance benchmarking. *Journal of Digital Imaging*, 24(5):883–889.
- Laura Novich, Dayle Parker, S. R. T. R. (2015). *Red Hat Enterprise Linux 7 Virtualization Deployment and Administration Guide*. Copyright©2015 Red Hat, Inc.
- Linux-KVM (2016). *Kernel Virtual Machine*. URL: <http://www.linux-kvm.org> (Accedido el 25-05-2016).
- Lubuntu (2016). URL: <http://lubuntu.me/> (Accedido el 10-11-2016).
- Medina, D. S. and Pérez, M. O. R. (2015). Virtualización del datacenter, una necesidad en la universidad del siglo XXI. *Revista Contribuciones a las Ciencias Sociales*, (29).
- Microsoft (2016a). *Conectarse a otro equipo mediante Conexión a Escritorio remoto*. URL: <http://windows.microsoft.com/es-xl/windows/connect-using-remote-desktop-connection#connect-using-remote-desktop-connection=windows-7> (Accedido el 27-05-2016).
- Microsoft (2016b). *Información general sobre Hyper-V*. URL: [https://msdn.microsoft.com/es-es/library/hh831531\(v=ws.11\).aspx](https://msdn.microsoft.com/es-es/library/hh831531(v=ws.11).aspx) (Accedido el 17-11-2016).
- Munin (2016). *About Munin*. URL: <http://munin-monitoring.org/> (Accedido el 12-11-2016).
- Oracle (2016). *Oracle Enterprise Pack for Eclipse 12c (12.2.1.3.1) System Requirements*. URL: <http://www.oracle.com/technetwork/developer-tools/eclipse/system-requirements-101778.html> (Accedido el 11-10-2016).

- QEMU (2016). URL: [http://wiki.qemu.org/Main\\_Page](http://wiki.qemu.org/Main_Page) (Accedido el 25-05-2016).
- Rayamajhi, Khan, S. (2016). Relative performance analysis of major virtual. *International Journal of Scientific & Engineering Research*, 6(1):1326–1330.
- Red Hat (2016a). *GENERAL INTRODUCTION TO VIRTUALIZATION*. URL: [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/7/html/Virtualization\\_Getting\\_Started\\_Guide/chap-Virtualization\\_Getting\\_Started-What\\_Is\\_It.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Virtualization_Getting_Started_Guide/chap-Virtualization_Getting_Started-What_Is_It.html) (Accedido el 25-05-2016).
- Red Hat (2016b). *Interface Configuration Files*. URL: [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/6/html/Deployment\\_Guide/s1-networkscripts-interfaces.html#s2-networkscripts-interfaces-eth0](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/6/html/Deployment_Guide/s1-networkscripts-interfaces.html#s2-networkscripts-interfaces-eth0) (Accedido el 28-05-2016).
- Red Hat (2016c). *USING THE COMMAND LINE INTERFACE (CLI)*. URL: [https://access.redhat.com/documentation/en-US/Red\\_Hat\\_Enterprise\\_Linux/7/html/Networking\\_Guide/sec-Network\\_Bonding\\_Using\\_the\\_Command\\_Line\\_Interface.html](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/Networking_Guide/sec-Network_Bonding_Using_the_Command_Line_Interface.html) (Accedido el 28-05-2016).
- RoseHosting (2016). *Install Munin on CentOS 7*. URL: <https://www.rosehosting.com/blog/install-munin-on-centos-7/> (Accedido el 11-11-2016).
- Salinas, P. (2016). *Casos de Uso (Use Case)*. URL: <https://users.dcc.uchile.cl/~psalinas/uml/casosuso.html> (Accedido el 18-11-2016).
- Siebert, E. (2009). *VMware VI3 implementation and administration*. Prentice Hall Professional.
- Solvetic (2016). *Diagrama de Distribución*. URL: <http://www.solvetic.com/tutoriales/article/555-uml-diagrama-de-distribucion/> (Accedido el 18-11-2016).
- Sparx (2016). *Diagrama de Componentes UML 2*. URL: [http://www.sparxsystems.com.ar/resources/tutorial/uml2\\_componentdiagram.html](http://www.sparxsystems.com.ar/resources/tutorial/uml2_componentdiagram.html) (Accedido el 6-11-2016).

Spice (2016). URL: <http://www.spice-space.org/> (Accedido el 21-05-2016).

Tanenbaum, A. S. and García, R. E. (2009). *Sistemas operativos modernos*. Pearson Educación.

VirtualBox (2016). URL: <https://www.virtualbox.org/> (Accedido el 23-05-2016).

VMware (2016). URL: <http://www.vmware.com/> (Accedido el 20-05-2016).

# Apéndice A: Glosario

## 1. **Anfitrión**

En ambientes de virtualización se refiere al servicio principal que corre sobre el servidor físico.

## 2. **Bonding**

Es una interfaz de red virtual que consiste en simular un dispositivo de red con un gran ancho de banda por medio de la unión de varias tarjetas de red independientes.

## 3. **GPU**

Unidad de Procesamiento Gráfico (GPU). Se trata de un procesador que se dedica exclusivamente al procesamiento de gráficos. Lo que hace la GPU es aligerar de trabajo a la CPU, sobre todo a la hora de abrir juegos o aplicaciones con gráficos interactivos 3D.

## 4. **Huésped**

En ambientes de virtualización se refiere al servicio cliente que corre sobre cada máquina virtual.

## 5. **Hipervisor**

Un hipervisor es un software que permite utilizar distintos sistemas operativos sobre un mismo computador. Se encarga de gestionar las máquinas virtuales que funcionan sobre el.

## 6. **Invitado**

Es la máquina virtual que se crea y a la que se le asignan recursos para funcionar.

## 7. **KVM**

Es un hipervisor que permite ejecutar máquinas virtuales de forma completa a

través de imágenes de disco con sistemas operativos sin modificar, en donde cada máquina virtual tiene su propio hardware virtualizado.

8. **Máquina Virtual**

Computador que esta formado con recursos computacionales virtualizados. Se comporta de forma idéntica a un computador físico.

9. **Paravirtualización**

Técnica de virtualización en donde la máquina virtual no simula obligatoriamente un hardware, sino que tiene una librería de funciones (API) que solo pueden usarse en sistemas operativos modificados.

10. **QEMU**

Es un emulador y una máquina de virtualización que permite la creación de máquinas virtuales dentro de un sistema operativo.

11. **Servidor**

En informática, un servidor es un computador que forma parte de una red que suministra servicios a otros computadores denominados clientes.