



UNIVERSIDAD INDUSTRIAL DE SANTANDER  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
GRUPO SIMON DE INVESTIGACIONES EN MODELADO Y SIMULACION.



IMPLEMENTACIÓN DE SERVICIOS, POLÍTICAS DE ADMINISTRACIÓN Y  
SEGURIDAD PARA EL SERVIDOR DEL GRUPO SIMON DE  
INVESTIGACIONES DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.

BELKY DIMARY ALVARADO RICO

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FÍSICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA-SANTANDER

2014



UNIVERSIDAD INDUSTRIAL DE SANTANDER  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
GRUPO SIMON DE INVESTIGACIONES EN MODELADO Y SIMULACION.



IMPLEMENTACIÓN DE SERVICIOS, POLÍTICAS DE ADMINISTRACIÓN Y  
SEGURIDAD PARA EL SERVIDOR DEL GRUPO SIMON DE  
INVESTIGACIONES DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER.

PROYECTO DE GRADO PARA OBTENER EL TÍTULO DE INGENIERA DE  
SISTEMAS

AUTORA:  
BELKY DIMARY ALVARADO RICO

DIRECCIÓN :  
HUGO HERNANDO ANDRADE SOSA  
Director Grupo SIMON de Investigaciones -UIS

CODIRECCIÓN:  
GIOVANNI LÓPEZ MOLINA                      LUIS EDUARDO GUERRA GONZÁLEZ  
Ing. de Sistemas y Msc en Ingeniería                      Ingeniero de Sistemas

UNIVERSIDAD INDUSTRIAL DE SANTANDER  
FACULTAD DE INGENIERÍAS FISICO-MECÁNICAS  
ESCUELA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA  
BUCARAMANGA-SANTANDER

2014

# Índice general

<b>Introducción</b>	<b>12</b>
<b>1. Presentación del Proyecto</b>	<b>13</b>
1.1. Antecedentes . . . . .	13
1.2. Descripción del Proyecto . . . . .	14
1.2.1. Planteamiento del Problema . . . . .	14
1.2.2. Objetivos . . . . .	16
1.2.2.1. Objetivo General . . . . .	16
1.2.2.2. Objetivos Específicos . . . . .	16
1.3. Metodología . . . . .	17
<b>2. Fundamentos de la Virtualización</b>	<b>18</b>
2.1. Definición . . . . .	18
2.2. Ventajas de la Virtualización . . . . .	20
2.3. Desventajas de la Virtualización . . . . .	21
2.4. Características de las Máquinas Virtuales . . . . .	21
2.5. Tipos de Virtualización . . . . .	22
2.5.1. Emulación . . . . .	22
2.5.2. Para-virtualización . . . . .	23
2.5.3. Virtualización completa . . . . .	23
2.6. Herramientas de Virtualización Libres . . . . .	24
<b>3. Biblioteca de Infraestructura de Tecnologías de la Información</b>	<b>27</b>
3.1. Definición y Conceptos de ITIL . . . . .	28
3.2. El Ciclo de Vida ITIL . . . . .	29
3.3. Fases y Procesos abarcados en el Proyecto . . . . .	32
<b>4. Implementación del Ciclo de Vida ITIL para el Servidor del Grupo</b>	<b>34</b>
4.1. Fase 1: Estrategia del Servicio . . . . .	34
4.1.1. Proceso de la Estrategia del Servicio . . . . .	35
4.1.1.1. Gestión de la Cartera de servicios . . . . .	35
4.1.2. Implementación de la Estrategia del Servicio . . . . .	36
4.1.2.1. Servicios servidor Grupo Simon de Investigaciones . . . . .	36
4.1.2.2. Requerimientos de usuarios servidor Grupo Simon . . . . .	39
4.2. Fase 2: Diseño del Servicio . . . . .	45
4.2.1. Procesos del diseño del Servicio . . . . .	45
4.2.1.1. Gestión del Catálogo de Servicios . . . . .	45
4.2.1.2. Gestión de la Capacidad. . . . .	46

4.2.1.3.	Gestión de la Disponibilidad . . . . .	46
4.2.1.4.	Gestión de la Seguridad de la Información. . . . .	47
4.2.2.	Implementación del Diseño del Servicio . . . . .	48
4.2.2.1.	Arquitectura de Virtualización Completa . . . . .	48
4.2.2.2.	Diagrama de red y mecanismo de despliegue . . . . .	51
4.2.2.3.	Especificaciones técnicas servidor Grupo Simon . . . . .	52
4.2.2.4.	Sistema Operativo <i>Debian 7.0 Wheezy</i> . . . . .	53
4.2.2.5.	Hipervisor KVM . . . . .	54
4.2.2.6.	Arquitectura de virtualización de KVM . . . . .	55
4.2.2.7.	Máquinas Virtuales KVM . . . . .	59
4.2.2.8.	Proxy inverso . . . . .	59
4.2.2.9.	Introducción y requisitos previos. . . . .	60
4.2.2.10.	Catálogo de Servicios servidor Grupo Simon . . . . .	62
4.2.2.11.	Políticas de administración y seguridad . . . . .	68
4.3.	Fase 3: Transición del Servicio . . . . .	76
4.3.1.	Proceso de la Transición del servicio . . . . .	76
4.3.1.1.	Validación y pruebas . . . . .	76
4.3.2.	Implementación de la Transición del Servicio . . . . .	77
4.3.2.1.	Prueba de soporte de virtualización KVM . . . . .	77
4.3.2.2.	Prueba de Funcionalidad <i>Virtual Machine Manager</i> . . . . .	78
4.3.2.3.	Pruebas de Conectividad y funcionamiento de las MV. . . . .	81
4.3.2.4.	Prueba de Accesos no autorizados al <i>Host</i> . . . . .	82
4.4.	Fase 4: Operación del Servicio . . . . .	85
4.4.1.	Procesos de la Operación del servicio . . . . .	85
4.4.1.1.	Gestión de Eventos . . . . .	85
4.4.1.2.	Gestión de Problemas . . . . .	85
4.4.1.3.	Gestión de Acceso a los Servicios TI . . . . .	85
4.4.2.	Implementación de la Operación del Servicio . . . . .	86
4.4.2.1.	Monitorización de la Infraestructura virtual . . . . .	86
4.5.	Fase 4: Mejora continua del Servicio . . . . .	91
<b>5.</b>	<b>Conclusiones</b>	<b>92</b>
<b>6.</b>	<b>Recomendaciones</b>	<b>93</b>
	<b>Referencias</b>	<b>94</b>

Anexos	98
A. Herramientas de administración de las Máquinas Virtuales	99
B. Instalación del sistema operativo base en el host	104
C. Instalación y configuración de <i>KVM</i> en el servidor	111
D. Instalación de <i>VirtualBox</i> y el sistema de escritorio	115
E. Instalación y configuración de <i>virt-manager</i>	121
F. Despliegue de las Máquinas Virtuales	126
G. Instalación <i>proxy inverso en el servidor host</i>	134
H. Instalación de la herramienta <i>Fail2ban</i>	136
I. Instalación y configuración de <i>bacula</i>	139
J. Instalación y configuración de <i>Nagios3</i>	145

# Índice de figuras

2.1. Arquitectura de Virtualización. . . . .	19
3.1. Ciclo de Vida ITIL . . . . .	30
3.2. Fases del Ciclo de Vida ITIL con sus Procesos y Funciones . . . . .	31
3.3. Procesos y Funciones del Ciclo ITIL abarcados en el proyecto. . . . .	32
4.1. Arquitectura de Virtualización Completa. . . . .	49
4.2. Diagrama de red y mecanismo de despliegue. . . . .	51
4.3. Arquitectura de virtualización KVM . . . . .	56
4.4. Arquitectura de virtualización KVM libvirt. . . . .	57
4.5. Nivel de soporte Virtualización KVM . . . . .	78
4.6. Administrador de Máquina Virtual . . . . .	79
4.7. Rendimiento CPU y Memoria: Máquina Host. . . . .	79
4.8. Rendimiento CPU y Memoria: SimonMV1 . . . . .	80
4.9. Monitor del Sistema: Recursos . . . . .	80
4.10. Prueba: Conexión ssh al Servidor Host . . . . .	81
4.11. Prueba: Instalación mysql, Phpmyadmin . . . . .	82
4.12. Prueba: Conexión por Phpmyadmin . . . . .	82
4.13. Prueba: Conexión ssh contraseña segura . . . . .	83
4.14. Prueba: Conexión ssh contraseña errónea . . . . .	83
4.15. Prueba: Intento de acceso Ip bloqueada . . . . .	84
4.16. Diagrama infraestructura virtual . . . . .	87
4.17. Ingreso a la herramienta nagios3 . . . . .	87
4.18. Area de trabajo de nagios3 . . . . .	88
4.19. Detectar errores del sistema con nagios3. . . . .	88
4.20. Posibles alertas críticas del sistema. . . . .	89
4.21. Estructura Monitorizada de nagios3. . . . .	89
4.22. Equipos y servicios monitorizados y sus estados . . . . .	90
4.23. Estado de los servicios. . . . .	90
B.1. Pantalla de arranque Debian Wheezy . . . . .	105
B.2. Configuración de red . . . . .	106
B.3. Elección del método del particionado . . . . .	106
B.4. Esquema – Particionado de discos . . . . .	109
B.5. Utilidades estándar necesarias . . . . .	110
C.1. Nivel de soporte de CPU para Virtualización KVM . . . . .	112
C.2. Configuración de red servidor Grupo SIMON . . . . .	113
C.3. Configuración archivo <i>/etc/hosts</i> . . . . .	114

D.1.	Pantalla de bienvenida de <i>VirtualBox</i> . . . . .	116
D.2.	Nombre de la Estación de trabajo y Sistema Operativo . . . . .	116
D.3.	Ubicación del archivo y tamaño . . . . .	117
D.4.	Personalizar Estación de trabajo . . . . .	117
D.5.	Adjuntar la imagen de <i>Debian 7 ISO</i> . . . . .	118
D.6.	Pantalla de Bienvenida de Instalación de <i>Debian 7</i> . . . . .	118
D.7.	Particionado de Discos . . . . .	119
D.8.	Pantalla seleccionar programas . . . . .	120
E.1.	Elección de Synaptic - Gestor de paquetes . . . . .	121
E.2.	Elección de virt-manager para su instalación . . . . .	122
E.3.	Elección de ssh-askpass para su instalación . . . . .	123
E.4.	Descarga de paquetes . . . . .	123
E.5.	Configuración del archivo <i>/etc/hosts</i> . . . . .	124
E.6.	Copiar clave SSH pública . . . . .	125
F.1.	Ingreso al administrador de Máquina Virtual . . . . .	126
F.2.	Administrador de Máquina Virtual . . . . .	127
F.3.	Detalles de la conexión . . . . .	127
F.4.	Administrador de Máquina Virtual . . . . .	128
F.5.	Conexión remota al Host . . . . .	129
F.6.	Creación de una nueva máquina virtual . . . . .	129
F.7.	Asignar nombre y medio de Instalación a la MV . . . . .	130
F.8.	Ubicación del medio de instalación y SO . . . . .	130
F.9.	Asignar la cantidad de memoria y CPUs a la MV . . . . .	131
F.10.	Selección de la cantidad de almacenamiento en disco . . . . .	131
F.11.	Selección de configuración de red y KVM . . . . .	132
F.12.	Autenticación para el ingreso a la nueva MV . . . . .	132
F.13.	Instalación del Sistema Operativo de la nueva MV . . . . .	133
H.1.	Configuración de los parámetros Ips, tiempos e intentos . . . . .	137
H.2.	Configuración del parámetro SSH . . . . .	138
I.1.	Creación de la Base de Datos bacula . . . . .	141
I.2.	Configuración del directorio de bacula . . . . .	142
I.3.	Configuración del directorio de bacula . . . . .	143
I.4.	Reiniciar servicios con bacula . . . . .	144
J.1.	Ingreso por la Web a nagios3 . . . . .	146
J.2.	Area de trabajo de nagios3 . . . . .	146

# Índice de tablas

4.1. Servicios servidor del Grupo Simon. . . . .	36
4.2. Requisitos de instalación de las aplicaciones. . . . .	39
4.3. Requerimientos tecnológicos servidor Simon. . . . .	39
4.4. Acceso al servidor y condiciones requeridas. . . . .	40
4.5. Espacio de almacenamiento de las aplicaciones . . . . .	41
4.6. Recomendaciones y estrategias de seguridad. . . . .	41
4.7. Sugerencias en general. . . . .	43
4.8. Especificaciones técnicas servidor <i>Hp Proliant</i> . . . . .	52

# Resumen

**TÍTULO:** IMPLEMENTACIÓN DE SERVICIOS, POLÍTICAS DE ADMINISTRACIÓN Y SEGURIDAD PARA EL SERVIDOR DEL GRUPO SIMON DE INVESTIGACIONES DE LA UNIVERSIDAD INDUSTRIAL DE SANTANDER \*

**AUTORA:** BELKY DIMARY ALVARADO RICO\*\*

**PALABRAS CLAVE:** Virtualización, KVM, ITIL.

**DESCRIPCIÓN:** El Grupo Simon de Investigaciones de la Universidad Industrial de Santander dedicado a la investigación y desarrollo, principalmente bajo el enfoque del pensamiento sistémico, realiza actividades y trabajos de investigación sobre el modelado y la simulación de sistemas, desarrollo de software, desarrollo de proyectos de investigación y extensión. Con recursos de un proyecto de investigación cofinanciado por Colciencias, el Grupo adquirió un servidor para el alojamiento de sus datos y aplicaciones. Este proyecto presenta el diseño y la implementación de servicios, políticas de administración y seguridad para el servidor del Grupo, a través de la implementación de una infraestructura de virtualización donde se ejecutaron varias máquinas virtuales a cargo de diferentes servicios y aplicaciones, ésta infraestructura garantiza el resguardo, la disponibilidad y el despliegue de los trabajos desarrollados y por desarrollar en el Grupo. La guía metodológica que orientó esta propuesta se basa en el enfoque del ciclo de vida ITIL para la Gestión de Servicios de Tecnologías de la Información, estas prácticas gestionan la entrega de servicios de alta calidad y fueron aplicadas e interpretadas de tal manera que se ajustaran a las necesidades y requerimientos de los miembros del Grupo Simon. La opción de virtualización permitió establecer los servicios necesarios para dar respuesta a las necesidades de gestión de datos y aplicaciones desarrolladas por los miembros del Grupo.

---

\*Trabajo de Grado en la Modalidad de Investigación.

\*\*Facultad de Ingenierías Físico Mecánicas. Escuela de Ingeniería de Sistemas e Informática.  
Director MSc. Hugo Hernando Andrade Sosa.

# Abstract

**TITLE:** SERVICE DEPLOYMENT, SECURITY AND MANAGEMENT POLICIES FOR THE SIMON'S RESEARCH GROUP SERVER OF THE INDUSTRIAL UNIVERSITY OF SANTANDER\*.

**AUTHOR:** BELKY DIMARY ALVARADO RICO\*\*

**KEYWORDS:** Virtualization, KVM, ITIL.

**DESCRIPTION:** The Simon Research Group, of the Industrial University of Santander, is dedicated to the investigation and development, mainly under the system dynamics approach, conducting researches on modeling and simulation systems, software development, as well as research and extension project development. Using financial assets, derived from a research project co-funded by Colciencias, a server has been acquired by Simon in order to provide secure and efficient hosting for the applications developed within the group. This project supports the design and implementation of services, administration and security policies for the group's hosting server, through the implementation of a virtualization infrastructure where several virtual machines were deployed to ensure the effective protection, availability and support of several developed and developing applications created by members of the research group. The methodological guide in order to carry out this proposal is based on the ITIL lifecycle approach for Service Management of Information Technology and the technological tool that could be applied to improve services and provide enhanced solutions for the Simon Research Group is virtualization. The use of virtualization technologies allows the establishment of the necessary services to attend the needs of database management and application hosting for the members of the group.

---

\*Undergraduate final project, research modality.

\*\*Physico-Mechanical Engineering Faculty. Systems Engineering and Computer Science School. Director MSc. Hugo Hernando Andrade Sosa.

# Introducción

El Grupo Simon<sup>1</sup> de Investigaciones de la Universidad Industrial de Santander, dedicado a la investigación y desarrollo en el modelado y simulación de sistemas, principalmente bajo el enfoque del pensamiento sistémico, realiza actividades y trabajos de investigación sobre el modelado y la simulación de sistemas, desarrollo de software, desarrollo de proyectos de investigación y extensión. El Grupo ha adquirido un nuevo servidor para el alojamiento de sus aplicaciones, se hace necesario contar con una organización adecuada para el resguardo de los datos y garantizar la disponibilidad, optimización y seguridad de los mismos. Además, por la dinámica de trabajo de investigación, en el Grupo se desarrollan aplicaciones utilizando diferentes tecnologías que demandan un uso dinámico de los recursos del servidor.

Este trabajo busca llevar a cabo la implementación de servicios, mantenimiento y administración del servidor del Grupo Simon, implementando una infraestructura de virtualización que garantice el resguardo, la disponibilidad y el despliegue de las aplicaciones desarrolladas y por desarrollar, dando apoyo a los proyectos y trabajos realizados por los miembros del Grupo. La propuesta de este trabajo se llevará a cabo bajo la Metodología del Ciclo de Vida del Servicio que ofrecen las buenas prácticas de ITIL<sup>2</sup> para la Gestión de Servicios de TIC, estas prácticas gestionan la entrega de servicios de Tecnologías de la Información de alta calidad, entregando a los usuarios un buen servicio y seguridad en sus datos.

---

<sup>1</sup>Grupo de investigación de la Universidad Industrial de Santander adscrito a la Escuela de Ingeniería de Sistemas e Informática de la facultad de Ingenierías Físico Mecánicas.

<http://simon.uis.edu.co/joomla/home/>

<sup>2</sup>La Biblioteca de Infraestructura de Tecnologías de Información, es un conjunto de conceptos y prácticas para la gestión de servicios de tecnologías de la información, el desarrollo de tecnologías de la información y las operaciones relacionadas con la misma.

[http://es.wikipedia.org/wiki/Information\\_Technology\\_Infrastructure\\_Library](http://es.wikipedia.org/wiki/Information_Technology_Infrastructure_Library)

# Capítulo 1

## Presentación del Proyecto

### 1.1. Antecedentes

En la Escuela de Ingeniería de Sistemas e informática de la Universidad Industrial de Santander, Grupos interesados en temas como la seguridad, administración gestión de recursos informáticos, desarrollo de aplicaciones software, entre otras, participan activamente en el desarrollo de técnicas que permiten proveer y mejorar servicios a los usuarios. Algunos trabajos realizados por estos equipos de trabajo (estudiantes de pregrado) son nombrados a continuación:

- "Modelo y prototipo de servicios de Computación en la Nube para estudiantes y profesores de la escuela de ingeniería de sistemas e informática de la Universidad Industrial de Santander". Proyecto de Grado elaborado por EMMANUELL DÍAZ CARREÑO. Escuela de Ingeniería de Sistemas, Universidad Industrial de Santander. 2012.
- "Instalación, administración, configuración e implantación de servidores Linux con énfasis en el desarrollo de un modelo administrativo y la creación de un prototipo de clúster de alta disponibilidad". Proyecto de Grado elaborado por ALEXANDER BARBOSA AYALA y ELKIN DARÍO MUÑOZ DUARTE. Escuela de Ingeniería de Sistemas, Universidad Industrial de Santander. 2012.
- "Administración, mantenimiento, configuración y monitoreo de los equipos servidores del grupo GID-CONUSS con énfasis en el análisis y reestructuración de los modelos de alta disponibilidad y computación en la nube". Proyecto de Grado elaborado por JOHN EDINSON LIZARAZO TORRES y DIEGO ALBERTO NOGUERA GIRALDO. Escuela de Ingeniería de Sistemas, Universidad Industrial de Santander. 2013.

## 1.2. Descripción del Proyecto

### 1.2.1. Planteamiento del Problema

En las investigaciones y trabajos realizados por el Grupo Simon y en el desarrollo de sus aplicaciones software se ha determinado la necesidad de soportar múltiples tecnologías que permitan la implementación de los productos tecnológicos fruto de cada uno de los proyectos realizados, por esta razón se adquirió un Servidor con el cual se pretende suplir los diversos requerimientos y necesidades de cada aplicación. Entre los proyectos que soportará este equipo de cómputo se destacan proyectos de investigación y extensión; trabajos pregrado algunos de los cuales brindan apoyo a proyectos de investigación; trabajos de investigación en maestría, entre otras. Tanto la temática, el funcionamiento y la tecnología varían en cada uno de los proyectos y aplicaciones. Algunos de estos trabajos son nombrados a continuación:

- SÍDEF-MOVIL<sup>3</sup>: "Sistema de Información Educación y Comunicación en enfermedades infecciosas que cursan con fiebres agudas". Proyecto dedicado al sistema de salud de primer orden que busca proporcionar al paciente, médico y sistema de salud la recolección, procesamiento y administración de datos en cada caso donde se presenten fiebres inespecíficas de origen infeccioso. SÍDEF-MOVIL Centraliza y analiza esta información con el fin de identificar epidemias en una fase temprana, establecer sistemas de alerta y tendencias de la enfermedad. Proyecto terminado y cofinanciado por Colciencias - -2013.
- "Mantenimiento y Evaluación del Sistema de Información: Educación y Comunicación en enfermedades infecciosas que cursan con fiebres agudas". Proyecto en curso.
- "Lineamientos para la formación inicial de docentes en el uso pedagógico de las TIC orientado al mejoramiento y a la innovación educativa. Trabajo en conjunto con Ministerio de Educación Nacional". Proyecto Terminado - -2012.
- ECOGRANJA: "Ambiente para el aprendizaje y la toma de decisiones en las granjas agropecuarias, basado en la dinámica de sistemas y soportado por tecnologías Web". Proyecto terminado - -2012.
- EVOLUCIÓN WEB: "Herramienta para simular modelos dinámico-sistémicos en la Web". Proyecto en Curso.
- "Propuesta para la difusión del modelado y la simulación, basado en objetos y reglas, en la educación básica primaria colombiana". Proyecto en curso.

---

<sup>3</sup><http://simon.uis.edu.co/SaludMovil>

- SMART GRID<sup>4</sup>: "Ambiente software integrado por un juego para teléfonos móviles, un sitio Web y una aplicación para computador personal, para el aprendizaje y la toma de decisiones en redes inteligentes de SMART GRID". Proyecto en curso.
- PESCO CONOCIMIENTO-SIMON: "Ambiente Software para el Aprendizaje y la Toma de Decisiones". El ambiente de aprendizaje está destinado a la toma de decisiones en un juego alojado en un dispositivo móvil, sustentada en un conocimiento previo adquirido con la ayuda de una aplicación de escritorio, la cual permite experimentar y analizar un posible comportamiento del juego. Esta decisión en el juego afectará directamente el estado de un mercado accesible en todo momento y que se encuentra alojado en un sitio web, que además lleva un indicador de rendimiento en el juego. Proyecto terminado - -2012.

Cada aplicación desarrollada presenta ciertas necesidades de almacenamiento y de cómputo por lo cual se hace necesario definir e implementar políticas de administración y seguridad del servidor para su correcto funcionamiento y posterior mantenimiento. Se propone un proyecto de grado, para la realización de estas tareas fundamentales en el trabajo continuo del Grupo. Este proyecto contribuye como requisito para optar por el título de Ingeniera de sistemas. El enfoque que guía la realización de este proyecto busca hacer transparente los requisitos tecnológicos de cada tipo de aplicación, permitiendo a los usuarios la libertad de escoger las tecnologías más pertinentes para lograr sus objetivos y así facilitar el desarrollo y la implementación de cada aplicativo. Se hace necesario configurar, poner en marcha un servidor que ejecute varias máquinas virtuales cada una de ellas con tecnologías independientes, aisladas unas de las otras donde la ejecución de una de ellas no afecte el rendimiento de las otras o del sistema en general.

---

<sup>4</sup><http://simon.uis.edu.co/avasmart>

## **1.2.2. Objetivos**

### **1.2.2.1. Objetivo General**

Definir e implementar políticas de administración y seguridad para el servidor del Grupo Simon de investigaciones que permitan dar soporte a las aplicaciones existentes y en desarrollo, en los procesos de investigación, docencia, extensión y difusión para el Grupo.

### **1.2.2.2. Objetivos Específicos**

- Definir los requerimientos, arquitectura y un esquema de servicios que soporte las diferentes tecnologías que utilizan las aplicaciones del Grupo actuales y en desarrollo.
- Implementar la arquitectura e instalar los servicios del servidor buscando la administración eficiente de los recursos, usuarios y permisos del sistema.
- Realizar la migración de los aplicativos implantados en el servidor actual al servidor nuevo.
- Definir lineamientos de seguridad informática del servidor del Grupo e implementar los mecanismos de gestión de la seguridad tanto para los sitios alojados en el servidor, el sistema operativo, la red y usuarios.
- Elaborar una guía básica de políticas de administración y seguridad del servidor del Grupo.

### 1.3. Metodología

Hoy en día las Tecnologías de la Información (TIC) juegan un papel importante dentro de las organizaciones, para poder llevar a cabo un buen trabajo dentro de las mismas, se deben garantizar aspectos tales como el desarrollo, diseño, y la seguridad para poder entregar eficientes servicios a los usuarios. Se pretende con el desarrollo de este proyecto, implementar una infraestructura de virtualización que logre poner a disposición de los miembros del Grupo Simon una herramienta para mejorar el rendimiento de los recursos, cumpliendo con los requerimientos de cada usuario.

Para proporcionar una infraestructura estable y que ofrezca los servicios requeridos, se plantea para el desarrollo de los objetivos de este trabajo emplear como guía metodológica el enfoque basado en el Ciclo de Vida ITIL.

ITIL apuntan a facilitar la entrega de servicios de TI de alta calidad, es por ello que adoptar estas prácticas satisfacen la calidad y seguridad de la información. El planteamiento que hace ITIL no es una norma a seguir, es una herramienta que la organización adapta de la manera que mejor le convenga. La flexibilidad del marco de ITIL es uno de los factores determinantes de su éxito. ITIL debe implementarse como parte de una metodología empresarial versátil que abarque todos los procesos de la organización y que al mismo tiempo permita perfeccionar los procesos y la eficacia de la Gestión de Servicios[8].

# Capítulo 2

## Fundamentos de la Virtualización

Se elige la virtualización de servidores como la herramienta indispensable para el desarrollo de esta propuesta ya que proporciona las utilidades necesarias para hacer un uso más eficiente de los recursos del servidor, permitiendo ejecutar varios sistemas operativos y aplicaciones simultáneamente dentro de una misma máquina física, esto reduce significativamente costos de mantenimiento, administración, costos de infraestructura física, energía entre otras y satisface los requerimientos que demandan la atención de la problemática a tender.

En este capítulo se definirá detalladamente el concepto de Virtualización, sus ventajas y desventajas, las características de las máquinas virtuales, los tipos de arquitecturas de virtualización y se explicará resumidamente las herramientas libres que existen para virtualizar un entorno Linux.

### 2.1. Definición

La virtualización se define como la técnica que permite, la ejecución de múltiples máquinas virtuales, aisladas, independientes, con su propio sistema operativo a cargo de servicios y aplicaciones sobre una única máquina física[5]. Las máquinas virtuales se comunican con la máquina real por medio de una capa de virtualización ó hipervisor VMM (*Virtual Machine Monitor*), compartiendo los recursos principales de la máquina aftrion, CPU, Memoria, Dispositivos de Red y Almacenamiento. El software de virtualización planifica la ejecución de las máquinas virtuales y gestiona el acceso compartido a los recursos hardware reales disponibles.

El Hypervisor o VMM, crea una capa entre el hardware de la máquina física (*host*) y el sistema operativo de la máquina virtual (*guest*), esta capa de virtualización gestiona, maneja y arbitra los cuatro recursos de la máquina host, es así como reparte dinámicamente dichos recursos entre todas las máquinas y de este modo permite tener varias máquinas virtuales ejecutándose sobre el mismo ordenador físico, ver *figura 2.1*<sup>5</sup>

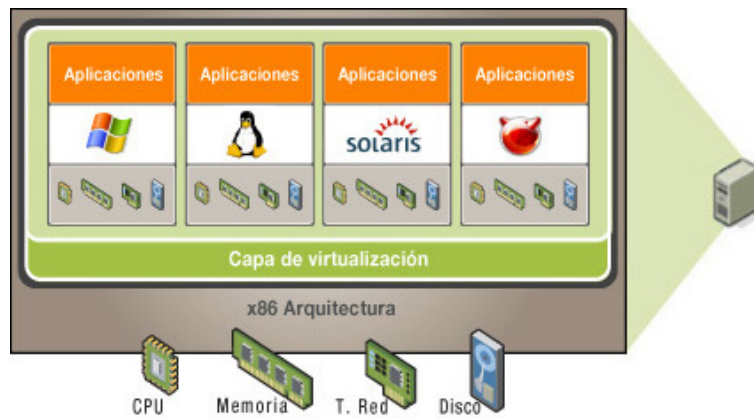


Figura 2.1: Arquitectura de Virtualización.  
Fuente: Fuente Digital<sup>5</sup>

La virtualización permite:

- Almacenar varios servidores en una sola computadora física, optimizando el uso de los recursos computacionales.
- Utilizar servidores virtuales para cada servicio que se desea implementar en la red, independientemente del hardware disponible.
- Facilitar la realización de copias de respaldo, así como su restablecimiento.
- Mejorar aspectos de seguridad, ya que cada servidor es independiente y aislado los unos de los otros, ocupándose cada uno de diferentes tareas.
- Facilitar la migración entre distintos equipos.

---

<sup>5</sup>Fuente Digital:  
<http://www.niux.com.ar/usos-virtualizacion-y-para-que-sirve-la-virtualizacion/>

## 2.2. Ventajas de la Virtualización

- **Reducción de costos:** Uno de los beneficios que trae la Virtualización con respecto a los costos es la consolidación de servidores, al consolidar servidores se tiene un cambio transformador en la disminución de los costos de energía y refrigeración. Según Tony Iams, analista senior de DH Brown Associates Inc. en Port Chester, Nueva York, "Los servidores de muchas empresas suelen ejecutar en un 15-20 % de su capacidad, lo que no puede ser una relación sostenible en el entorno económico actual". Las empresas están cada vez más interesadas en la consolidación de servidores como un medio para reducir los costos innecesarios y maximizar el retorno de la inversión. De 518 encuestados en un estudio de investigación de Gartner Group, el 6 % había llevado a cabo un proyecto de consolidación de servidores, el 61 % realiza actualmente una parte, y el 28 % planea hacerlo en el futuro inmediato[2]. Otra ventaja en cuanto a costos, es que la virtualización permite que los usuarios puedan utilizar múltiples aplicaciones en un solo servidor físico, por medio de la implementación de máquinas virtuales generando una reducción de infraestructura de cómputo y por ende un beneficio económico elevado. El tiempo es otro aspecto importante para reducir los costos, la implementación de servidores en un ambiente virtual es menor que en un ambiente físico.
- **Recuperación ante desastres:** Utilizar un sistema de servidores virtuales es muy útil en caso de desastres. Las máquinas virtuales están configuradas para comportasen en forma de ficheros, es decir que se pueden copiar, mover de una máquina física a otra, de esta forma es muy fácil realizar el backup de dicho fichero y recuperar la máquina virtual inmediatamente, solamente con copiar el fichero sobre una máquina física que soporte la solución de virtualización.
- **Alta disponibilidad:** Este concepto hace referencia a la probabilidad que un servicio funcione adecuadamente y que esté disponible para el usuario las veinticuatro horas del día. Cuando una máquina virtual se caiga se pueda levantar otra con las mismas características en otra máquina física, sin pérdida del servicio[26].
- **Administración:** La virtualización facilita la administración de servidores. Todas las herramientas de virtualización ya sean comerciales o de código abierto, poseen programas que facilitan la administración de máquinas virtuales. La gestión de las máquinas virtuales se puede hacer de forma centralizada, con lo que desde una ubicación física central se podrán gestionar todas y cada una de las máquinas virtuales que pueden estar implementadas en una compañía a lo largo del mundo[26]. las máquinas virtuales se administran de forma independiente, lo cual facilita las labores de mantenimiento, es más sencillo identificar cierto problema de una máquina sin necesidad de inspeccionar las otras. La administración de la máquina física es muy sencilla, ya que solo tiene instalado el sistema operativo y la aplicación de virtualización. La administración del hardware de las máquinas virtuales es independiente del hardware real, si se desea realizar alguna modifi-

cación, o cambio lo único que se requiere es apagar cada máquina y copiarla al nuevo servidor y luego arrancarla de nuevo. Al implementar una infraestructura virtual los usuarios pueden solicitar nuevos servicios TI u obtenerlos rápidamente sin que sea necesario adquirir nuevos servidores.

## 2.3. Desventajas de la Virtualización

- Crear máquinas virtuales innecesarias implica un coste en ocupación de recursos principalmente en el espacio del disco y la memoria RAM.
- Si por algún motivo se daña el disco duro de la máquina física, se perderán todas las máquinas virtuales. Por este motivo es necesario el uso del RAID.
- Un sistema virtualizado no alcanzará el mismo rendimiento que un sistema directamente instalado ya que la solución de virtualización introduce una capa intermedia en la gestión del hardware por lo que el rendimiento de las máquinas virtuales es menor.
- En caso de robo, incendio, cualquier evento que ocurra con el hardware, afectará toda la máquina, por ello es necesario hacer respaldos de las máquinas virtuales frecuentemente.
- No es posible utilizar hardware que no esté gestionado o soportado por la solución de virtualización.

## 2.4. Características de las Máquinas Virtuales

Una máquina virtual es un contenedor de software perfectamente aislado que puede ejecutar sus propio sistema operativo y aplicaciones como si fuera un ordenador físico, ya que se comporta exactamente igual, contiene su propia CPU, RAM, Disco duro y Tarjetas de red virtuales. El propósito de una máquina virtual es mejorar la distribución de los recursos para múltiples usuarios y el rendimiento del equipo en términos de utilización de recursos y flexibilidad de aplicación[26]. A diferencia de una máquina física, una virtual se compone exclusivamente de software y poseen las siguientes características:

**Compatibilidad:** Las máquinas virtuales son compatibles con sistemas operativos x86 estándar como Windows y Linux, y con los controladores de hardware y aplicaciones creadas para esos sistemas operativos. Una máquina virtual posee todos los componentes que se encuentran en un servidor físico. Cualquier aplicación que pueda ejecutarse en los servidores físicos, también se ejecutará en las máquinas virtuales.

**Aislamiento:** Las máquinas virtuales pueden compartir los recursos físicos de un mismo ordenador pero están aisladas al igual que una máquina física. Cada máquina tiene su sistema operativo, configuración de datos, configuración de red, CPU, RAM y discos independiente. Al estar aisladas se facilita el nivel de flexibilidad que lo proporciona un entorno físico no virtualizado[26].

**Encapsulamiento:** Las máquinas virtuales se gestionan como archivos. Una máquina virtual se encapsula en una colección de archivos, y al igual que los archivos, las máquinas virtuales se pueden copiar, mover, distribuir, es decir, son portátiles y fáciles de administrar[26].

**Independencia de hardware:** Una máquina virtual es totalmente independiente del hardware físico. La máquina virtual tiene todos los componentes de una máquina física pero éstos son virtuales (placa base virtual, tarjeta VGA virtual, controlador de tarjeta de red virtual, etc.) es decir que una máquina virtual puede ejecutarse en una máquina real con tarjeta de red de marca x, pero siempre verá una tarjeta de red virtual. Esto es útil porque permite que una máquina virtual se pueda mover de una máquina real a otra, sin realizar cambios en los controladores del dispositivo aunque las dos máquinas reales sean de dos fabricantes distintos. La independencia del hardware proporciona más portabilidad y flexibilidad.

## 2.5. Tipos de Virtualización

El tipo de virtualización más implementado en sistemas es la creación de entornos con máquinas virtuales. Antes de ser virtualizada una máquina el sistema operativo se encarga de gestionar el hardware y después de ser virtualizada una capa de virtualización es insertada entre el hardware y el sistema operativo, en tal caso, la capa de virtualización es responsable de convertir las porciones de hardware real en hardware virtual, por lo tanto, diferentes sistemas operativos tal como Linux y Windows pueden correr en la máquina física a la vez.

Existen tres tipos comunes de virtualización son: Emulación, Para-virtualización y Virtualización completa.

### 2.5.1. Emulación

Se basa en crear máquinas virtuales en el host anfitrión que emulan completamente el comportamiento de la plataforma hardware completa, permitiendo la ejecución de los sistemas operativos sin modificar. El software llamado emulador puede hacer creer a los programas y sistemas operativos diseñados para una arquitectura concreta que son ejecutados sobre ella realmente; ésta es la arquitectura emulada[27]. Una de las

ventajas es que se pueden ejecutar sistemas operativos completamente diferentes del anfitrión. El inconveniente de este modelo de virtualización es que la simulación es muy lenta y es la más costosa.

### 2.5.2. Para-virtualización

Con este tipo de virtualización las máquinas virtuales que se crean son modificadas para que interactúen con el virtualizador y sean conscientes que existe dicha capa de virtualización. La Para-virtualización implica ejecutar sistemas operativos guests sobre otro sistema operativo que actúa como hypervisor (host). Los guests tienen que comunicarse con el hipervisor para lograr la virtualización. Como es necesario que se modifiquen los kernels del host y el guest, esta aproximación a la virtualización sólo se puede utilizar en kernels de código libre. Igualmente esta limitación explica que la paravirtualización no permite virtualizar sistemas Windows[24].

Una de las ventajas de la paravirtualización es que presta buen rendimiento y la posibilidad de ejecutar distintos sistemas operativos como guests, también se obtiene todas las ventajas de la virtualización, reducción de costos de energía entre otras. Una desventaja es el trabajo de modificar los sistemas operativos guest para que funcionen con este esquema.

### 2.5.3. Virtualización completa

Este tipo de Virtualización permite la creación de diferentes máquinas virtuales con distintos sistemas operativos que media entre el hardware y las máquinas virtuales. A diferencia de la Paravirtualización, este tipo de virtualización permite ejecutar un sistema operativo guest sin ninguna modificación en él. Todas y cada una de las instrucciones y accesos a los dispositivos de hardware tienen que ser emulados por un hipervisor. Las máquinas virtuales que se ejecutan con virtualización completa no son conscientes que tienen esta capa de virtualización que las separa del hardware.

La arquitectura del modelo de Virtualización Completa consta de los siguientes elementos[27]:

- Hardware la máquina anfitriona que incluya un procesador con soporte a tecnología de virtualización, como Intel VT o AMD-V.
- Un hipervisor ejecutado directamente sobre el hardware disponible, y en el que se han realizado diversas modificaciones (emulación hardware) para el manejo y administración de la infraestructura virtual. Sin embargo, también es posible encontrar arquitecturas de virtualización completa en las que el hipervisor, llamado en este caso hosted, está integrado con un sistema operativo determinado; este

tipo es usado en escenarios en los que los requisitos son de menor importancia o el entorno es menos exigente.

- La presencia de una máquina virtual con carácter administrativo, aunque también puede que sea una consola que disponga de este control administrativo.

## 2.6. Herramientas de Virtualización Libres

Algunas de las Herramientas de Virtualización libres serán descritas a continuación.

### Bochs

*Definición:* Es un emulador de arquitecturas basadas en x86, funciona en múltiples plataformas y en cualquier sistema anfitrión. Incluye emulación de la CPU Intel x86, dispositivos de E/S comunes y un BIOS[1].

*Ventaja:* *Bochs* funciona en la mayoría de sistemas operativos; Linux, Microsoft Windows. *Bochs* es capaz de emular un PC completo incluyendo los periféricos.

*Desventaja:* *Bochs* es muy poco extendido por su rendimiento bajo, carece de una interfaz gráfica amigable y sencilla.

*Tipo de virtualización:* Permite virtualización mediante emulación.

### VirtualBox

*Definición:* Oracle VM VirtualBox es una solución de virtualización de código abierto para arquitecturas X86 que funcionan sobre múltiples sistemas operativos[28].

*Ventajas:* Permite crear, administrar y configurar máquinas virtuales de 32 y 64 bits, sobre las que se puede instalar cualquier sistema operativo que funcione en la arquitectura virtualizada; cada máquina virtual puede arrancarse, suspenderse y pararse de modo independiente y como los ficheros que definen la máquina virtual son los mismos en todos los sistemas anfitriones es fácil mover las máquinas virtuales entre distintos equipos[9]. *VirtualBox* Puede ser ejecutado en Windows, Linux, Macintosh, Solaris/OpenSolaris, Mac OSX, OS/2 Warp. Otra característica de VirtualBox es que permite el acceso a máquinas virtuales de forma remota, usando VNC (en la edición libre), por medio del Remote Desktop Protocol (RDP).

*Desventaja:* No tiene un API o utilidad de línea de comandos para manipular el estado de las máquinas virtuales.

*Tipo de virtualización:* Permite virtualización completa.

## QEMU

*Definición:* Es un emulador similar a Boch, emula un equipo entero. QEMU tiene dos modos de funcionamiento, uno como un emulador de otro hardware y otro como Virtualizador.

En el primer modo, *QEMU* emula todos los componentes físicos de un ordenador real (procesador, memoria, buses, periféricos...). La CPU emulada puede ser diferente de la CPU física que se utiliza para ejecutar QEMU, de esta manera es posible ejecutar en el emulador programas para una arquitectura diferente. Se utiliza traducción dinámica para aumentar la velocidad de emulación[25].

Como Virtualizador, este modo depende de un componente llamado *acelerador QEMU* o bien *KQEMU*, que es un driver que permite ejecutar el código de la máquina huésped directamente en la CPU física. De esta manera se alcanza una velocidad de ejecución similar a la velocidad de ejecución nativa. En este caso la máquina hospedada y la CPU física deben ser x86[25].

Para las arquitecturas x86 *QEMU* soporta el uso de un módulo de aceleración para sistemas anfitriones Linux y Windows que permite que parte del código que se ejecuta en los sistemas invitados sea ejecutado directamente por la CPU real, haciendo que el *QEMU* funcione como un sistema de virtualización nativa en lugar de como un emulador. diferentes[6].

*Ventaja:* Capacidades de virtualización dentro de un sistema operativo, GNU/Linux, Windows. Puede ejecutar cualquier tipo de arquitectura (x86, x86-64). Está licenciado en parte por la LGPL.

*Desventaja:* No dispone de GUI.

*Tipo de virtualización:* Permite virtualización mediante emulación.

## KVM

*Definición:* Kernel Virtual Machine (KVM) es una solución de virtualización completa en la que se utiliza el núcleo de Linux como hipervisor, de manera que tanto el control de los dispositivos reales como la planificación de tareas y la gestión de memoria del sistema anfitrión las hace el núcleo de Linux[6]. Con KVM se pueden ejecutar múltiples máquinas virtuales sin modificar las imágenes Linux o Windows. Cada máquina virtual tiene un hardware virtualizado privado: una tarjeta de red, disco, tarjeta gráfica[21].

*Ventaja:* KVM está formado por un módulo del kernel de Linux denominado *kvm.ko* y un núcleo de herramientas que se ejecutan en el espacio de usuario. *kvm.ko* permite realizar las funciones de hipervisor tratando a las máquinas virtuales como si fuesen un proceso más del sistema host, esto permite que la virtualización se beneficie de las mejoras de rendimiento (planificación de procesos o

la gestión de la memoria) que van surgiendo en el continuo desarrollo del kernel. KVM necesita ejecutarse en procesadores de la familia x86 que tengan soporte hardware para la virtualización (Intel VT o AMD-V) para poder proporcionar una virtualización completa[21].

*Tipo de virtualización:* Permite virtualización completa.

## Xen

*Definición:* Xen es una solución de para-virtualización que implementa un hipervisor que se ejecuta en el nivel más privilegiado de la máquina y se hace cargo de la planificación de tareas y la gestión de memoria, delegando la gestión de la Entrada/Salida en un invitado privilegiado (llamado domain 0 o dom0 en Xen) que arranca siempre que lanzamos el hipervisor[6].

*Ventaja:* Xen es usualmente utilizado como una plataforma de virtualización de servidores, que se ejecutan en servidores sin interfaz gráfica y controlados a través de la red. Algunos usuarios avanzados también corren Xen en sus entornos de escritorio[31]. Xen es un hipervisor para uso en IA-32, x86-64, es un hipervisor de código abierto, fue desarrollado por la Universidad de Cambridge, proporciona un entorno virtual situado entre el hardware y el sistema operativo. Los componentes centrales de un sistema Xen son el hipervisor, el núcleo y las aplicaciones. Al igual que los sistemas de virtualización, muchos sistemas operativos invitados pueden ejecutar en la parte superior del hipervisor. Sin embargo, no todos los sistemas operativos invitados se crean del mismo modo, y hay uno determinado que controla los demás.

*Desventaja:* Cuando Xen se emplea en una CPU que no soporta virtualización a nivel de hardware es necesario modificar el código del sistema operativo que se vaya a ejecutar sobre él, por lo que no es posible ejecutar sistemas como Windows Xp en una CPU que no proporcione soporte hardware a la virtualización[25]. El hipervisor de Xen nunca será integrado con el código del núcleo de Linux.

*Tipo de virtualización:* Permite paravirtualización.

## Capítulo 3

# Biblioteca de Infraestructura de Tecnologías de la Información

En múltiples definiciones que existen sobre las Tecnologías de Información (TIC,) muchas concuerdan en que las TIC son definidas como la integración de la computación, el procesamiento de datos y las redes de telecomunicaciones, para el almacenamiento, manejo y recuperación de la información, donde sus principales componentes son: el hardware, el software, los datos, la infraestructura, el factor humano, los mecanismos de intercambio de la información, recursos financieros, entre otros.

Hoy en día las condiciones tecnológicas de las organizaciones hacen necesaria la optimización de los recursos buscando eficiencia, eficacia y calidad en la prestación de un servicio. Las TIC son la herramienta clave y necesaria para el desarrollo y mantenimiento de cualquier tipo de organización. A través del uso de las TIC, la organización logra importantes mejoras para la toma de decisiones que aportan al crecimiento de la misma, la reducción de costos y entrega al usuario de un mejor servicio.

Para llevar a cabo los objetivos de esta propuesta se toma como guía metodológica a ITIL. (*Information Technology Infrastructure Library*), conjunto de buenas prácticas orientadas hacia la prestación de servicios de alta calidad en el área de las tecnologías de la información. Es evidente en este proyecto la implementación de la virtualización como un servicio ya que representa una alternativa necesaria y eficaz para dar respuesta a las necesidades del Grupo Simon y mejorar los servicios que se ofrecen y ofrecerán.

En este capítulo se describirán las buenas prácticas ITIL y el Ciclo de Vida ITIL, haciendo énfasis en que ITIL descompone cada una de las fases del ciclo de vida del servicio en procesos de gestión, que son una serie de acciones conectadas con el fin de lograr un objetivo, se describirán las fases y los procesos abarcados en este proyecto.

### 3.1. Definición y Conceptos de ITIL

La *Biblioteca de Infraestructura de Tecnologías de la Información*, abreviada ITIL, definida como un conjunto de buenas prácticas, está destinada a facilitar y mejorar la gestión y provisión de servicios de Tecnologías de la Información. ITIL es de libre utilización, y es la organización quien decide cómo implementarlas de la forma en que más le convenga dependiendo de sus necesidades y sus requerimientos.

Los orígenes de ITIL se remontan a la década de los 80 cuando el gobierno británico preocupado por la calidad de los servicios TI de los que dependía la administración, solicitó a una de sus agencias, la *CCTA* (acrónimo de *Central Computer and Telecommunications Agency*), para que desarrollara un estándar para la provisión eficiente de servicios de Tecnologías de la Información. En la actualidad es la *OGC* (*Office of Government Commerce*) el organismo encargado de velar por este estándar y la responsable de la última versión de ITIL (v3) que data del año 2007[17].

Según ITIL *La Gestión de Servicios de TI (ITSM)*, consiste en dejar a disposición los servicios de Tecnologías de la Información que cumplan con las expectativas de la organización que las aplique y garantiza que estos servicios se lleven a cabo de una forma efectiva y eficiente. En otras palabras, la Gestión de servicios es la habilidad orientada hacia la transformación de los recursos en servicios de valor, optimizando el uso de las capacidades organizativas durante el ciclo de vida de los servicios.

Una correcta gestión de servicio requerirá:

- Conocer las necesidades del cliente.
- Estimar la capacidad y recursos necesarios para la prestación del servicio
- Establecer los niveles de calidad del servicio
- Supervisar la prestación del servicio
- Establecer mecanismos de mejora y evolución del servicio.

Otro concepto importante que ITIL ofrece es el concepto de *Servicio* como un medio para entregar valor a los clientes facilitándoles un resultado deseado sin la necesidad de que estos asuman los costes y riesgos específicos asociados. Es decir que el objetivo de un servicio consiste en satisfacer una necesidad, sin asumir directamente las capacidades y recursos necesarios para ello[14].

ITIL también involucra tres conceptos fundamentales: *Función*, *Proceso* y *Rol*. La *Función* se considera como la unidad especializada en la realización de alguna actividad y es la responsable de su resultado. Las funciones incorporan todos los recursos y capacidades necesarias para el correcto desarrollo de dicha actividad. El *Proceso* es un

conjunto de actividades interrelacionadas orientada hacia el estricto cumplimiento de un objetivo. El *Rol* involucra un conjunto de actividades y responsabilidades para una persona o equipo.

## 3.2. El Ciclo de Vida ITIL

Como se describió anteriormente ITIL es una librería de buenas prácticas que tiene como objetivo asegurar la calidad en la gestión de los servicios y mejorar la calidad de los procesos en la organización pero no proporciona ninguna pauta a seguir y la mejor manera de hacer uso de estas prácticas es interpretarlas y aplicarlas de la forma que mejor se ajuste a las necesidades y requerimientos de la organización.

El principio en que se basa ITIL es en el Ciclo de vida de los Servicios, que tiene como objetivo ofrecer una visión global de la vida de un servicio, es decir, las diferentes fases por las que pasa un servicio desde su creación (*estrategia del servicio, diseño del servicio*) pasando por su implementación en la infraestructura (*transición del servicio*) hasta su cumplimiento (*operación del servicio*).

El ciclo de Vida del Servicio consta de cinco fases:

- 1 Estrategia del servicio TI
- 2 Diseño del servicio TI
- 3 Transición del Servicio TI
- 4 Operación del servicio TI
- 5 Mejora continua del servicio TI

Como lo muestra la *figura 3.1*<sup>6</sup>, la *Estrategia del Servicio* es el eje en torno al que giran todas las demás fases del Ciclo de Vida del Servicio, es la fase de definición de políticas y se fundamenta en las habilidades objetivos y metas del negocio para lograr servicios efectivos. Es en la Estrategia del Servicio donde se determinan las necesidades, prioridades e importancia relativa de los servicios para entregar los resultados deseados. Las fases de *Diseño*, *Transición* y *Operación del Servicio* (etapas giratorias del ciclo) ponen en práctica esta estrategia a través de ajustes y cambios. La fase de *Mejora Continua del Servicio*, que consiste en el aprendizaje y mejora, abarca todas las fases del ciclo e inicia los proyectos y programas de mejora, asignándoles prioridades en función de los objetivos estratégicos de la organización.

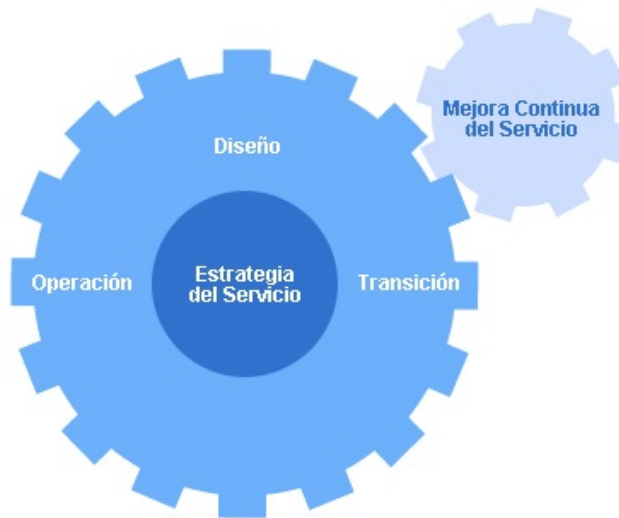


Figura 3.1: Ciclo de Vida ITIL  
Fuente: Fuente Digital<sup>6</sup>

---

<sup>6</sup>Fuente Digital: El ciclo de vida de los servicios TI:  
[http://itilv3.osiatis.es/ciclo\\_vida\\_servicios\\_TI.php](http://itilv3.osiatis.es/ciclo_vida_servicios_TI.php)

ITIL descompone cada una de las cinco fases del Ciclo de vida del Servicio en procesos de Gestión<sup>7</sup>. Un *proceso* es una serie de acciones o actividades conectadas con el fin de satisfacer un propósito y lograr un objetivo. En la *figura 3.2* se muestra las fases y los procesos del Ciclo de vida del servicio considerados por ITIL.

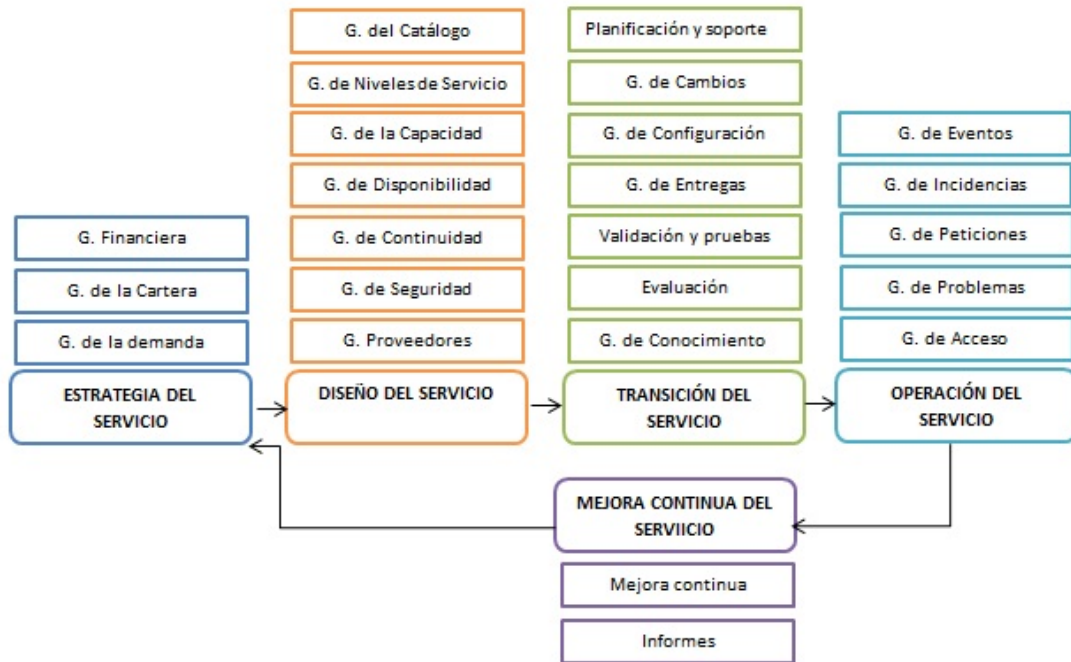


Figura 3.2: Fases del Ciclo de Vida ITIL con sus Procesos y Funciones  
Fuente: Autora

<sup>7</sup>Procesos descritos en la *figura 3.2* modificados por la autora, tomando como referencia el diagrama ITIL de las fases del Ciclo de Vida con sus procesos y funciones más destacados.  
[http://itilv3.osiatis.es/apendice\\_itilv2\\_itilv3.php](http://itilv3.osiatis.es/apendice_itilv2_itilv3.php)

### 3.3. Fases y Procesos abarcados en el Proyecto

A continuación se definirá breve cada uno de los procesos asociado directamente a cada fase del ciclo abarcados en el desarrollo de esta propuesta *figura 3.3* y posteriormente serán detallados el capítulo siguiente.

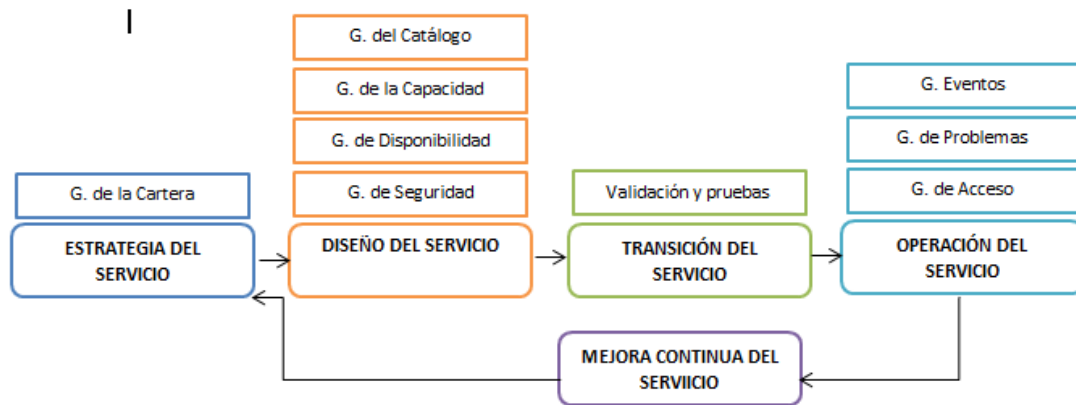


Figura 3.3: Procesos y Funciones del Ciclo ITIL abarcados en el proyecto.

Fuente: Autora

#### Fase 1. Estrategia del Servicio

*Gestión de la Cartera de Servicios:* Consiste en definir una estrategia de servicio que sirva para generar el máximo valor controlando riesgos y costes. En este proceso, se conocen todos los servicios que presta la organización, así como su estado, recursos, generando como consecuencia, el conocimiento de los servicios ante lo miembros de la Organización.

#### Fase 2. Diseño del Servicio

*Gestión del Catálogo de Servicios:* Consiste en ofrecer la información detallada de todos los servicios que la organización presta y los recursos asignados para ello. El Catálogo de Servicios proporciona una guía a los usuarios a la hora de elegir un servicio que se adapte a sus necesidades.

*Gestión de la Capacidad:* Responsable de garantizar que la organización TI dispone de la capacidad suficiente para prestar los servicios acordados.

*Gestión de la Disponibilidad:* Responsable de garantizar que los servicios funcionen ininterrumpidamente y de manera fiable.

*Gestión de la Seguridad en la Información:* Responsable de establecer las políticas de

integridad, confidencialidad y disponibilidad de la información.

### **Fase 3. Transición del Servicio**

*Validación y pruebas:* responsable de garantizar que los servicios cumplen los requisitos preestablecidos antes de su paso al entorno de producción.

### **Fase 4. Operación del Servicio**

*Gestión de Eventos:* Es la encargada de monitorizar el rendimiento de la infraestructura TI para la prevención de errores o interrupciones en el servicio.

*Gestión de Problemas:* Responsable de analizar y ofrecer soluciones a aquellos incidentes que por su frecuencia o impacto degradan la calidad del servicio.

*Gestión de Acceso a los Servicios TI:* Responsable de garantizar que sólo las personas con los permisos adecuados pueda acceder a la información de carácter restringido.

### **Fase 5. Mejora Continua del Servicio**

Esta fase provee una forma para que la organización pueda identificar y administrar adecuadamente las mejoras, de los servicios y mejoras en todas las fases del ciclo de vida del servicio.

# Capítulo 4

## Implementación del Ciclo de Vida ITIL para el Servidor del Grupo

En este capítulo se describirá el proceso de implementación del Ciclo de Vida ITIL para la Gestión del Servicio que ofrecerá el servidor del Grupo Simon. En cada fase se desglosarán los procesos involucrados en la propuesta y se definirán una serie de actividades que ayudarán al desarrollo de cada objetivo.

### 4.1. Fase 1: Estrategia del Servicio

La Estrategia del Servicio es definida por ITIL como la fase central al concepto de ciclo de vida del servicio y tiene como principal objetivo establecer el valor del servicio. El *servicio* es definido, como un medio de aportar valor al cliente sin que éste deba asumir los riesgos y costes específicos de su prestación. Por otra parte el valor, es el producto de la utilidad y garantía del servicio[10].

La *Utilidad* establece que el servicio:

- Cumpla los requisitos señalados por el cliente.
- Aumente el rendimiento y productividad.
- Resulte en un beneficio para el cliente, disminuyendo directamente los costes o contribuyendo a aumentar los ingresos.

La *Garantía* implica que el servicio:

- Estará disponible cuando se le necesite.
- Estará correctamente dimensionado para cumplir sus objetivos.
- Sea Seguro.
- Dispondrá de mecanismos de respaldo que permitirá su continuidad.

Con estas definiciones se concluye que el propósito de esta fase consiste en definir qué servicios deben ser prestados y por qué han de ser prestados desde la perspectiva del cliente. Con el fin de proporcionar un servicio aceptable resulta indispensable

conocer las necesidades de los usuarios para dar una respuesta optima a esas necesidades, esto ayuda a medir la capacidad y los recursos necesarios para prestar un servicio.

Para que una organización TI pueda ofrecer valor en forma de servicios debe hacer buen uso de los recursos y capacidades necesarias para prestar los servicios ofrecidos teniendo en cuenta los costes y riesgos asociados[9]. ITIL define Los *recursos* como la "materia prima" necesaria para la prestación del servicio e incluye: el capital, las infraestructuras, aplicaciones e información. Las *capacidades* representan las habilidades desarrolladas a lo largo del tiempo para transformar los recursos en valor a través de la gestión, la organización, los procesos y el conocimiento. También se encuentra el *personal* que es en sí mismo un recurso que aporta entre otras capacidades su profesionalidad creatividad y capacidad de liderazgo.

### **4.1.1. Proceso de la Estrategia del Servicio**

#### **4.1.1.1. Gestión de la Cartera de servicios**

Por medio de la Cartera de servicios se conocen todos los servicios que presta la organización, así como su estado, los recursos asignados a cada uno de ellos, las tendencias, bajo ciertos parámetros para la generación de nuevos servicios, siempre sujetos a requisitos de calidad, con los costos asociados respectivamente.

Es de gran importancia mencionar que la Cartera de servicios es la fase que alimenta el resto del Ciclo de vida ITIL ya que tiene toda la información necesaria para orientar cualquier tipo de actividad, es la encargada de decidir la estrategia a seguir para dar servicio a los clientes.

Las principales actividades de la Cartera de Servicios se resumen en conocer las necesidades del usuario, analizar los servicios y recursos necesarios para suministrar los servicios siempre buscando un rendimiento eficiente.

### 4.1.2. Implementación de la Estrategia del Servicio

Para el desarrollo de esta fase, se proponen las siguientes actividades:

*Actividad 1:* Definir de forma detallada los servicios (las aplicaciones) que ofrece el Grupo Simon de Investigaciones por parte del servidor y los recursos tecnológicos que brindan apoyo a las mismas.

*Actividad 2:* Analizar y definir de forma detallada los requerimientos de los usuarios del servidor del Grupo Simon de Investigaciones.

#### Desarrollo

*Actividad 1:* Definir de forma detallada los servicios (las aplicaciones) que ofrece el Grupo Simon de Investigaciones por parte del servidor y los recursos tecnológicos que brindan apoyo a las mismas.

#### 4.1.2.1. Servicios servidor Grupo Simon de Investigaciones

Tabla 4.1: Servicios servidor del Grupo Simon.

Fuente: Autora.

Aplicación	SIDEF-MOVIL
Definición	<i>"Sistema de información educación y comunicación en enfermedades infecciosas que cursan con fiebres agudas"</i> . Proyecto dedicado al sistema de salud de primer orden que busca proporcionar al paciente, médico y sistema de salud la recolección, procesamiento y administración de datos en cada caso donde se presenten fiebres inespecíficas de origen infeccioso. <i>SIDEF-MOVIL</i> , centraliza y analiza la información con el fin de identificar epidemias en una fase temprana, establecer sistemas de alerta y tendencias de la enfermedad.
Aplicación compuesta de:	Base de datos Aplicación Web Aplicación para dispositivos Móviles
Tecnologías de Desarrollo	Java Development Kit 1.6 Java Runtime Environment 1.6 Eclipse Java EE IDE for Web Developers 3.5 SQL

Alojamiento Web	PHP Motor de base de datos MySQL Servidor Web Apache Tomcat Java Development Kit 1.6 Java Runtime Environment.
<b>Aplicación</b>	<b>ECOGRANJA</b>
Definición	<i>"Ambiente para el aprendizaje y la toma de decisiones en las granjas agropecuarias, basado en la dinámica de sistemas y soportado por Tecnologías Web".</i>
Aplicación compuesta de:	Base de datos Ecogranja: <i>"Ambiente de aprendizaje y toma de decisiones en la Web.</i> Evolución web: <i>"Ambiente para el modelado y simulación de dinámica de sistemas en la Web"</i>
Tecnologías de Desarrollo	Estándares Web HTML, XHTML, XML. Virtual Hosts. Web Services SQL a través de ORM (Object Relational Mapping). Shells.
Alojamiento Web	PHP Servidor Web HTTP: Apache 2. Ruby y Javascript del lado servidor. Base de datos: PostgreSQL. Utilitarios: OpenSSH, RVM (Ruby versión Manager). Ruby onRails. Passenger.
<b>Aplicación</b>	<b>EVOLUCIÓN WEB</b>
Definición	<i>"Ambiente para el modelado y simulación de dinámica de sistemas en la Web".</i>
Tecnologías de Desarrollo	Estándares Web HTML, XHTML, XML. Virtual Hosts. Web Services SQL a través de ORM (Object Relational Mapping). Shells.
Alojamiento Web	PHP Servidor Web HTTP: Apache 2. Ruby y Javascript del lado servidor. Base de datos: PostgreSQL. Utilitarios: OpenSSH, RVM (Ruby versión Manager). Ruby onRails. Passenger

<b>Aplicación</b>	<b>SMART GRID</b>
Definición	<i>"Ambiente software integrado por un juego para teléfonos móviles, un sitio Web y una aplicación para computador personal, para el aprendizaje y la toma de decisiones en redes inteligentes de SMART GRID".</i>
Aplicación compuesta de	Base de datos Gestor de contenido web Aplicativo servlet de java
Alojamiento	MYSQL PHP Joomla APACHE PHPMYADMIN
<b>Aplicación</b>	<b>PESCO CONOCIMIENTO-SIMON</b>
Definición	<i>"Ambiente Software para el Aprendizaje y la Toma de Decisiones".</i> El ambiente de aprendizaje está destinado a la toma de decisiones en un juego alojado en un dispositivo móvil, sustentada en un conocimiento previo adquirido con la ayuda de una aplicación de escritorio, la cual permite experimentar y analizar un posible comportamiento del juego. La decisión en el juego afectará directamente el estado de un mercado accesible en todo momento y que se encuentra alojado en un sitio web, que además lleva un indicador de rendimiento en el juego.
Aplicación compuesta de	Aplicaciones en la web joomla Bases de datos
Alojamiento Web	PHP MYSQL APACHE Joomla
<b>Aplicación</b>	<b>Sitio Web Simon</b>
Definición	Página Web donde se aloja la información General del Grupo Simon
Aplicación compuesta de	Aplicaciones en la web Bases de datos
Alojamiento Web	PHP MYSQL APACHE Joomla

*Actividad 2:* Analizar y definir de forma detallada los requerimientos de los usuarios del servidor del Grupo Simon de Investigaciones.

#### 4.1.2.2. Requerimientos de usuarios servidor Grupo Simon

Tabla 4.2: Requisitos de instalación de las aplicaciones.

Fuente: Autora.

Requerimientos	Aplicación
MYSQL, PHP, JBoss (Para el aplicativo en java).	SMARTGRID
Java Development Kit 1.6, Java Runtime Environment 1.6, Eclipse Java EE IDE for Web Developers 3.5 o superior, SQL, Java.	SIDEF-MOVIL
Estándares Web HTML, XHTML, XML, Virtual Hosts, Web Services SQL a través de ORM (Objecto Relational Mapping), Shells.	ECOGRANJA
Java, PHP, Mysql.	Otras aplicaciones

Tabla 4.3: Requerimientos tecnológicos servidor Simon.

Fuente: Autora.

Requerimientos	Aplicación
Motor de base de datos MySQL 5.5 o superior, Servidor Web Apache Tomcat 5.5 o superior, Java Development Kit 1.6 o JDK6, Java Runtime Environment 1.6 o JRE6.	SIDEF-MOVIL
Tanto ECOGRANJA como Evolución Web, requieren de las mismas tecnologías web de parte del servidor como los son: <ul style="list-style-type: none"> <li>■ Servidor Web HTTP: Apache 2. <ul style="list-style-type: none"> <li>● Lenguajes de programación del lado servidor: Ruby y Javascript del lado servidor.</li> </ul> </li> <li>■ Base de datos: PosgreSQL.</li> <li>■ Utilitarios: OpenSSH, RVM (Ruby versión Manager), Ruby onRails, Passenger.</li> </ul>	ECOGRANJA
Motor de base de datos MySQL 5.5 o superior, Servidor Web Apache Tomcat 5.5 o superior, PHP 5.0 o superior	Otras aplicaciones

Tabla 4.4: Acceso al servidor y condiciones requeridas.

Fuente: Autora.

Requerimientos	Aplicación
<ul style="list-style-type: none"> <li>■ Se desea tener un espacio de dominio donde se pueda instalar una versión de joomla (2.5 o superior) con conexión a una base de datos mysql y que se pueda administrar remotamente (tanto el sitio, como la base de datos).</li> <li>■ Un dominio <i>simon.uis.edu.co/nombrepryecto</i>.</li> <li>■ Usuario con acceso remoto a disco para administrar ficheros que se requieran para la gestión de conexión y otros.</li> <li>■ Usuario para acceso a una base de datos mysql y vincularla con joomla. Además de poder realizar los cambios que se deseen a esta base de datos. De ser posible que el acceso al dominio apunte a un fichero "index.php" o "index.html" del espacio en disco otorgado.</li> <li>■ La instalación de joomla deseada puede ser realizada remotamente por nosotros una vez se tenga la información del acceso al equipo y a la base de datos.</li> <li>■ Acceso para poder iniciar o detener el aplicativo servlet de java en cualquier momento para un mantenimiento óptimo.</li> </ul>	SMARTGRID
Se requiere un acceso SSH y una cuenta root para la instalación de los aplicativos.	ECOGRANJA
<ul style="list-style-type: none"> <li>■ Permisos para subir y modificar archivos en el área de trabajo.</li> <li>■ Usuario con privilegios para la base de datos, asociados a la base de datos de cada una de las aplicaciones.</li> </ul>	Otras aplicaciones

Tabla 4.5: Espacio de almacenamiento de las aplicaciones

Fuente: Autora

<b>Requerimientos</b>	<b>Aplicación</b>
0.5 GB.	SMARTGRID
500 MB.	SIDEF-MOVIL
EcoGranja ocupa 5MB aproximadamente y el espacio de datos cliente se aconseja reservar un total de 1GB. Evolución web ocupa 10 MB aproximadamente y el espacio de datos cliente se aconseja reservar un total de 2GB.	ECOGRANJA
En conjunto las aplicaciones requieren 20GB.	Otras aplicaciones

Tabla 4.6: Recomendaciones y estrategias de seguridad.

Fuente: Autora.

<b>Requerimientos</b>	<b>Aplicación</b>
Definición clara de roles	SMARTGRID
<ul style="list-style-type: none"> <li>▪ Buena configuración del Apache, incluyendo una lista blanca.</li> <li>▪ Límite de peticiones cliente.</li> <li>▪ Cantidad límite de conexiones.</li> <li>▪ Comunicación encriptada.</li> <li>▪ Una base de datos independiente con seguridad especializada (en otra máquina virtual).</li> <li>▪ Copias de seguridad automatizadas.</li> </ul>	ECOGRANJA

<ul style="list-style-type: none"> <li>▪ Hacer una revisión de los puertos para dejar abiertos solo los que se estén utilizando.</li> <li>▪ Los Accesos vía sftp y ssh deben hacerse por una lista blanca.</li> <li>▪ Implementar bloqueos a usuarios que fallen al ingresar las contraseñas.</li> <li>▪ Usar IPTABLES.</li> <li>▪ No permitir accesos con root vía ftp.</li> <li>▪ Restringir el acceso de los usuarios a su área de trabajo por el ftp.</li> <li>▪ Limitar el acceso por ventana de comando lo más posible.</li> <li>▪ Ningún usuario puede cambiar la configuración del equipo, solo el root.</li> <li>▪ Definir una política de contraseñas seguras.</li> <li>▪ Uso de sudo para registrar las fuentes de los posibles cambios en el sistema.</li> <li>▪ Uso adecuado de los hosts para los diferentes sitios.</li> <li>▪ Uso de una conexión de la red Renata.</li> </ul>	<p>Otras aplicaciones</p>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------

Tabla 4.7: Sugerencias en general.

Fuente:Autora.

<b>Requerimientos</b>	<b>Aplicación</b>
<p>La consideración de qué usuarios pueden realizar una supervisión autónoma de su proyecto, para evitar de esta forma trabas en la implementación y control del proyecto.</p>	<p>SMARTGRID</p>
<ul style="list-style-type: none"> <li>▪ Que tenga UPS para caídas de energía (en cuanto a nivel de hardware).</li> <li>▪ Discos duros de respaldo (en cuanto a nivel de hardware).</li> <li>▪ Conexión de la mayor velocidad y calidad posible (en cuanto a nivel red).</li> <li>▪ Que tenga el soporte para servidores virtuales (VPS) (en cuanto a nivel de software).</li> <li>▪ Que permita una gestión física de subdominios para cada VPS.</li> <li>▪ Conexión remota.</li> <li>▪ Interfaz gráfica y web de administración remota.</li> <li>▪ Un excelente sistema de Logs.</li> </ul>	<p>ECOGRANJA</p>

<ul style="list-style-type: none"> <li>■ Implementar un servicio de alojamiento en el servidor del grupo, con seguridad de acceso ( no permitir ejecución).</li> <li>■ Definir las políticas de administración que garanticen el mantenimiento y servicios del servidor 24/7.</li> <li>■ Realizar backup automáticos de Base de datos diarios.</li> <li>■ Realizar Backup automáticos de aplicaciones semanales.</li> <li>■ Definir un proceso de montaje de las aplicaciones en el servidor.</li> <li>■ Solicitar a los proyectos un manual de instalación y problemas frecuentes de cada una de las aplicaciones.</li> <li>■ Asegurar el reinicio automático de todos los servicios y aplicaciones, una cuando se reinicie el equipo (por falla de luz por ejemplo).</li> <li>■ Uso de la conexión de la red renata.</li> </ul>	<p>Otras aplicaciones</p>
-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------

## 4.2. Fase 2: Diseño del Servicio

Esta fase es la encargada de desarrollar nuevos servicios o modificar los ya existentes, esto incluye; arquitecturas, procesos, políticas, documentación, siempre asegurando que se cumplan los requisitos de los usuarios o clientes. Los servicios a diseñar se deben adecuar a las necesidades, siendo eficientes en cuanto a costos y rentabilidad y cumplan con los estándares de calidad, aportando valor en forma de servicios.

El diseño del Servicio debe tener en cuenta tanto los requisitos del servicio como los recursos y capacidades disponibles en la organización TI[11], ofreciendo garantía y funcionalidad en el servicio prestado.

ITIL contempla ciertos aspectos esenciales en el Diseño del Servicio como lo son:

- Diseño de soluciones de servicio: Debe incluir de forma estructurada todos los elementos clave del nuevo o modificado servicio como requisitos del servicio, estudio de recursos o capacidades involucradas.
- Diseño de la Cartera de Servicios: Debe incluir información sobre todos los servicios ofrecidos, capacidades y recursos utilizados.
- Diseño de la arquitectura del servicio: La arquitectura debe tener en cuenta todos los elementos necesarios para la Gestión del Servicio, debe ofrecer una guía para el diseño y evolución del servicio teniendo en cuenta: La alineación entre la tecnología y el negocio, la infraestructura TI necesaria, la Gestión de las aplicaciones, la Gestión de los datos y la información, los Planes de Despliegue del servicio.

### 4.2.1. Procesos del diseño del Servicio

Los procesos de la fase del Diseño del Servicio abarcados en este proyecto son los siguientes:

- 1 Gestión del Catálogo del Servicio.
- 2 Gestión de la Capacidad.
- 3 Gestión de la Disponibilidad.
- 4 Gestión de la Seguridad de la Información.

#### 4.2.1.1. Gestión del Catálogo de Servicios

El Catálogo del Servicio describe en forma detallada la información de todos los servicios que se ofrecen en la organización y los recursos asignados. El Catálogo es indispensable ya que proporciona a los usuarios una guía a la hora de elegir un servicio que se adapte a sus necesidades y debe especificar para cada servicio

información tal como: nombre y descripción, propietario del servicio, cliente, fechas de versión y revisión, condiciones de prestación del servicio, precios, cambios y excepciones.

#### **4.2.1.2. Gestión de la Capacidad.**

El objetivo de este proceso es poner a disposición de clientes, usuarios y del propio departamento TI los recursos informáticos necesarios, asegurando que se dispondrá de la capacidad necesaria para llevar a cabo sus tareas y todo ello sin incurrir en costes desproporcionados[12].

La Gestión de la Capacidad es la encargada de que todos los servicios TI se vean respaldados por una capacidad de proceso y almacenamiento suficiente y correctamente dimensionada. Si en una organización TI hay una correcta Gestión de la Capacidad, el aprovechamiento de los recursos es mayor, generando un buen servicio.

Entre las responsabilidades de la Gestión de la Capacidad se encuentran[12]:

- Asegurar que se cubren las necesidades de capacidad TI tanto presentes como futuras.
- Controlar el rendimiento de la infraestructura TI.
- Desarrollar planes de capacidad asociados a los niveles de servicio acordados.
- Gestionar y racionalizar la demanda de servicios TI.

Si la Gestión de la capacidad es adecuada se evita realizar inversiones innecesarias en tecnologías que en muchos casos se vuelven obsoletas, se racionaliza la adquisición y el mantenimiento de infraestructura TI logrando mejoras financieras y de rendimiento operativo y también se asignan recursos adecuados de hardware, software y personal a cada servicio y aplicación.

#### **4.2.1.3. Gestión de la Disponibilidad**

La Gestión de la Disponibilidad es la encargada de asegurar que los servicios TI estén disponibles y funcionen sin interrupciones y de manera confiable siempre que los clientes y usuarios hagan uso de ellos.

Entre las actividades que se realizan en la Gestión de la Disponibilidad se encuentran la determinación de los requisitos de disponibilidad, diagnósticos sobre disponibilidad, informes, evaluar el impacto de las políticas de seguridad en la disponibilidad.

#### 4.2.1.4. Gestión de la Seguridad de la Información.

La Gestión de la Seguridad es el proceso encargado de asegurar que la información sea correcta y completa, esté siempre a disposición de la organización y sea utilizada sólo por aquellos que tienen autorización para hacerlo. La Gestión de la Seguridad debe así mismo tener en cuenta los riesgos generales a los que está expuesta la infraestructura TI.

La correcta gestión de la información debe apoyarse en tres pilares fundamentales[13]:

- Confidencialidad: la información debe ser sólo accesible a sus destinatarios predefinidos.
- Integridad: la información debe ser correcta y completa.
- Disponibilidad: debemos de tener acceso a la información cuando la necesitamos.

La Gestión de la seguridad debe establecer una política de seguridad en colaboración con los usuarios para minimizar los riesgos de seguridad que impidan la continuidad del servicio y también de establecer protocolos de seguridad que aseguren que la información esté accesible cuando es necesaria para aquellos que tengan autorización de utilizarla. También es responsabilidad de la Gestión de la Seguridad, asignar los recursos necesarios para garantizar la seguridad de los servicios, establecer las medidas necesarias y protocolos de acceso a la información.

La política de seguridad determina los protocolos de acceso a la información. Por otra parte, establece los procedimientos para el análisis de riesgos y requerimientos de recursos (personal, software y hardware), auditorías y generación de reportes.

Por otra parte es responsabilidad de la Gestión de Seguridad coordinar la implementación de los protocolos y medidas de seguridad establecidas en la Política y el Plan de Seguridad, asignar los recursos necesarios, instalar y mantener las herramientas de hardware y software necesarias para garantizar la seguridad y establecer las políticas y protocolos de acceso a la información.

Resulta indispensable evaluar el cumplimiento de las medidas de seguridad y resultados, es recomendable que estas evaluaciones se complementen con auditorías de seguridad externas o internas realizadas por personal independiente de la Gestión de la Seguridad, estas evaluaciones o auditorías deben valorar el rendimiento y proponer mejoras.

### 4.2.2. Implementación del Diseño del Servicio

Para la fase de Diseño se proponen las siguientes actividades:

*Actividad 1:* Definir la arquitectura tecnológica que albergará los servicios del servidor del Grupo Simon.

*Actividad 2:* Diseñar y definir el diagrama de red y el mecanismo de despliegue de la infraestructura virtual y los elementos que la componen.

*Actividad 3:* Implementar la arquitectura elegida e instalar los servicios del servidor gestionando y dimensionando adecuadamente los recursos, buscando un rendimiento óptimo de la infraestructura

*Actividad 4:* Elaborar un Catálogo de servicios para los miembros del Grupo Simon, con el fin de que conozcan las aplicaciones y recursos disponibles, y elijan los servicios que más les convenga para el alojamiento de sus aplicaciones.

*Actividad 5:* Diseñar las políticas de administración y seguridad necesarias para el buen desempeño y rendimiento de la Infraestructura.

*Actividad 6:* Diseñar un documento interno donde se especifique la administración eficiente del servidor del Grupo Simon, incluyendo las políticas de administración y seguridad, recomendaciones generales.

#### Desarrollo

*Actividad 1:* Definir la arquitectura tecnológica que albergará los servicios del servidor del Grupo Simon.

##### 4.2.2.1. Arquitectura de Virtualización Completa

Para el desarrollo del proyecto es fundamental la elección de una Arquitectura de Virtualización de servidores adecuada, donde el uso de los recursos tecnológicos sea óptimo y eficiente. En el *capítulo 2* se introdujeron los tipos de arquitecturas de virtualización, y se eligió la arquitectura de virtualización completa para el desarrollo del proyecto. La virtualización completa permite alojar sistemas operativos invitados no modificables, ya que es necesario un hipervisor que permite compartir el hardware real de la máquina física, así mismo aísla las máquinas virtuales entre sí y del sistema operativo, al estar aisladas una caída o falla de una de ellas no afectará el rendimiento de las otras máquinas y del servidor.

En la *figura 4.1*<sup>8</sup> podemos ver los elementos que componen la virtualización completa.

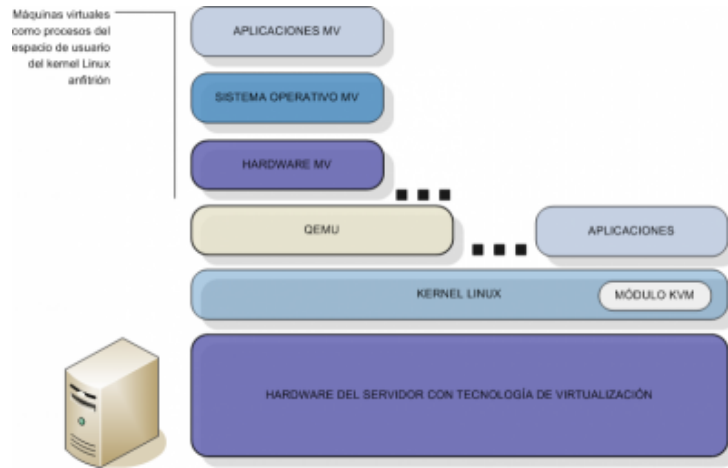


Figura 4.1: Arquitectura de Virtualización Completa.  
Fuente: Fuente Digital<sup>8</sup>

La Virtualización Completa a comparación con otros tipos de arquitecturas, es la más flexible, consiste fundamentalmente en la abstracción de un sistema hardware completo permitiendo que diversos sistemas operativos corran sobre una misma máquina física. En este tipo de virtualización las máquinas virtuales comparten los recursos físicos de la máquina anfitrión y cada una de ellas son sistemas totalmente independientes.

La arquitectura de virtualización completa consta de los siguientes elementos:

- **Hardware:** Se debe disponer siempre de un servidor físico sobre el cual se implementará la infraestructura de virtualización.
- **Sistema operativo Base:** Sobre la máquina física corre un sistema operativo que actúa como anfitrión.
- **Hipervisor:** Es el elemento virtualizador, este corre sobre el sistema operativo anfitrión y es quien aportará los recursos hardware de los que harán uso las máquinas virtuales.
- **Máquinas virtuales:** Creadas, configuradas, inicializadas y vigiladas por la capa de virtualización, las máquinas virtuales actúan como equipos totalmente independientes y disponen de su propio hardware virtual, sistema operativo y aplicaciones.

<sup>8</sup>Imagen tomada de la página web <http://www.adminso.es/index.php/KVM> Administración de sistemas operativos.

A continuación se definirá como gestiona la virtualización completa los recursos que proporciona a los guests:

- **CPU:** En la gestión de la CPU en la virtualización completa, el planificador del hipervisor del sistema, gestiona en que momento un guest puede utilizar la CPU. En caso de que haya más de un núcleo en la CPU se puede reservar uno o más núcleos para un guest concreto, tarea que realiza de forma transparente el hipervisor[26].
- **Memoria:** La gestión de la memoria también es muy parecida a la que se hace en un entorno sin virtualizar. Cuando se ejecuta un guest, el hipervisor crea un espacio de direcciones contiguo para esta. Este espacio de direcciones tiene las mismas propiedades que el espacio de memoria virtual de un sistema no virtualizado. De esta forma, este tipo de gestión permite al hipervisor ejecutar simultáneamente múltiples máquinas protegiendo la memoria de cada una de ellas[26].

Actividad 2: Diseñar y definir el diagrama de red y el mecanismo de despliegue de la infraestructura virtual y los elementos que la componen.

#### 4.2.2.2. Diagrama de red y mecanismo de despliegue

Dando respuesta a los requerimientos planteados por los miembros del Grupo, se propone el siguiente esquema de red *Figura 4.2*.

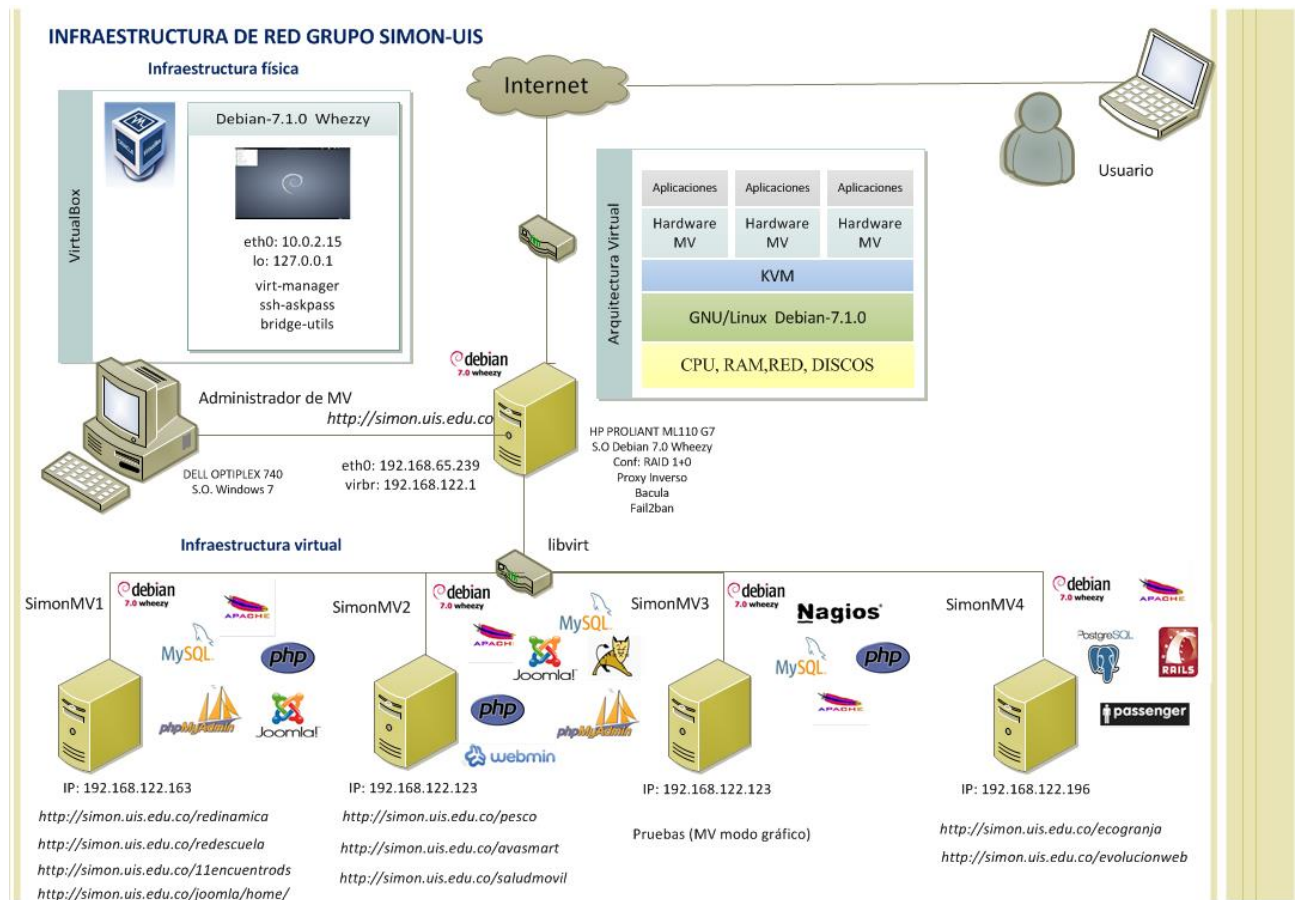


Figura 4.2: Diagrama de red y mecanismo de despliegue.

Fuente: Autora

Como se describe en la figura anterior, el diagrama de red está compuesto por la infraestructura física conformada por el servidor que cumple la función de host y un equipo alternativo donde se lleva a cabo la administración de las Máquinas Virtuales del host y la infraestructura virtual conformada por el sistema operativo host, el hipervisor, las máquinas virtuales y las herramientas tecnológicas usadas para el correcto funcionamiento de la infraestructura. A continuación se muestran con más detalles estos componentes.

## Infraestructura Física

Como se puede apreciar en la *Figura 4.2*, la infraestructura física está conformada por un servidor Hp Proliant ML110 G7, y un equipo Dell OpTipler 740, descritos a continuación:

**Servidor Hp Proliant ML110 G7:** Este servidor fue adquirido por el Grupo Simon con recursos de una investigación financiada por *Colciencias* para el resguardo de sus aplicaciones y datos. Su dirección IP está registrada con el dominio *http://simon.uis.edu.co*, y será configurado de tal forma que pueda redirigir las peticiones de la forma *http://simon.uis.edu.co/nombreaplicacion*. En el servidor *Proliant* correrá un Sistema Operativo *Debian 7.1.0- Wheezy* que actuará como sistema anfitrión y que proporcionará el entorno de virtualización. En la *tabla 4.8* se muestran las Especificaciones técnicas del servidor *Proliant*.

### 4.2.2.3. Especificaciones técnicas servidor Grupo Simon

Tabla 4.8: Especificaciones técnicas servidor *Hp Proliant*

Fuente: Autora

Característica	Descripción
Factor de forma	Micro torre -4U
Dimensiones (Ancho x Profundo x Altura)	17.5 cm x 47.3 cm x 36.7 cm
Peso	26.8 kg
Localización	Europea
Número de procesadores	1
Tipo de procesador	Intel Xeon E3-1220 (4 núcleos, 3,10 GHz, 8 MB , 80 W)
Características principales del procesador	Tecnología HyperThreading, Intel Turbo Boost Technology
Tecnología multipolar	Quad-core
Núcleo de procesador disponible	4 o 2
Memoria (máximo)	16 GB
Tipo de memoria	UDIMM DDR3
Tipo de controlador de almacenamiento	1 x Serial ATA integrado
Tipo de controlador interfaz	Serial ATA-300
Nombre de la controladora de almacenamiento	Smart Array B110i
Nivel RAID	RAID 0, RAID 1, RAID 10
Tipo de monitor	Ninguno

Disco duro	1 x 250 GB estándar Serial ATA-300 7200 rpm
Conexión de redes	Adaptador de red-PCI Express 2.0 x1-integrado
Controladora(s) Ethernet	HP NC112i
Protocolo de interconexión de datos	Ethernet, Fast Ethernet, Gigabit Ethernet
Protocolo de gestión remota	IPMI 2.0
Gestión de infraestructura	Estándar iLO, Insight Control

**Equipo Dell OpTipler 740:** Computador encargado de proporcionar una interfaz gráfica amigable para la administración, creación, configuración de las máquinas virtuales alojadas en el servidor *Proliant*, su administración remota se lleva a cabo por medio de un hipervisor *VirtualBox* que sobre el se encuentra alojado un sistema operativo *Debian 7*.

### Infraestructura Virtual

La infraestructura virtual estará conformada por un sistema operativo anfitrión *Debian 7.1.0- Wheezy*, sobre el sistema operativo correrá *KVM*, el elemento virtualizador, que aportará los recursos hardware de los que harán uso las máquinas virtuales. Las máquinas virtuales dispondrán cada una de un sistema operativo *Debian 7.1.0- Wheezy* de 32 bits y recursos tecnológicos para soportar distintas aplicaciones, realizadas en el por los miembros del Grupo. Sobre el sistema operativo anfitrión de la máquina física se configurará un *proxy inverso*, encargado de redirigir las peticiones de los usuarios que desean acceder a un sitio web alojado en algunas de las máquinas virtuales.

A continuación se proporcionará un resumen detallado de cada una de estas tecnologías.

#### 4.2.2.4. Sistema Operativo *Debian 7.0 Wheezy*

Sobre la máquina física corre un sistema operativo que actúa como anfitrión. *Debian Wheezy* es la versión 7 de la distribución GNU/Linux Debian[4]. *Debian* es una de las distribuciones linux más completas ya que dispone de versiones para varios entornos informáticos, todo ello basado en el software libre (en especial del proyecto GNU). Debian es una de las distribuciones GNU/Linux más maduras y estables, el detalle más importante de Debian es que tiene más paquetes de software en sus repositorios oficiales y por lo tanto se dice que es la más completa, con más de 14.500 paquetes fuente, el software disponible puede satisfacer casi cualquier necesidad que se tenga ya sea en casa o en la empresa[7].

*Debian 7.0 Wheezy* nueva versión estable 7.0 de *Debian*, incluye varias funcionalidades interesantes como, soporte multiarquitectura, que permitirá a los usuarios de *Debian* instalar paquetes para múltiples arquitecturas en la misma máquina. Esto significa que ya es posible, por primera vez, instalar software para 32 bits y 64 bits en el mismo sistema y tener todas las dependencias relevantes resueltas automáticamente[23].

*Debian* es un sistema operativo que se puede utilizar para distintos casos: desde sistemas de escritorio, servidores de desarrollo hasta clústeres de alta disponibilidad; para servidores de bases de datos, servidores web o de almacenamiento. Se ha hecho un gran esfuerzo para asegurar la calidad por medio de pruebas automáticas de instalación y actualización para todos los paquetes en el archivo de *Debian*, asegurando de esta forma que *Wheezy* cumple las altas expectativas que tienen los usuarios de una versión estable de *Debian*. Es tan sólido como una roca y ha sido probado de manera rigurosa[23].

#### 4.2.2.5. Hipervisor KVM

Es el elemento virtualizador, integrado sobre un Sistema Operativo en nuestro caso *Debian 7.1.0- Wheezy*. Para la implementación de la infraestructura propuesta se elige *KVM*, encargado de proporcionar el entorno de virtualización completa y proporcionará los recursos hardware de los que harán uso las máquinas virtuales, se escoge *KVM* por la facilidad que proporciona el venir integrado como módulo del *kernel Linux* y la capacidad de poder ejecutar sistemas operativos de diferente naturaleza. *KVM* permitirá ejecutar, crear, monitorizar máquinas virtuales invitadas independientes.

*KVM (Kernel-based Virtual Machine* traducido *máquina virtual basada en el núcleo*) es uno de los últimos hipervisores en hacer su entrada en el mercado de la virtualización, es en su totalidad un software de código abierto, que se integra en un sistema operativo existente y proporciona una solución de virtualización completa para Linux en hardware x86 que contiene extensiones de virtualización Intel VT o AMD-V, por lo que es necesario disponer de un equipo con procesador x86 o x86\_64, con soporte para virtualización. *KVM* permite ejecutar como huésped máquinas virtuales utilizando cualquier sistema operativo GNU/Linux ( como *debian*, *ubuntu*, *RHEL*, *Centos*, *Fedora*, *opensuse*, *mandriva*), *BDS*, *Solaris*, *MAC* y *Windows* (32-bit y 64-bit), sin tenerles que hacer modificaciones. Cada máquina virtual tiene su propio hardware virtualizado: una tarjeta de red, discos duros, tarjeta gráfica[19].

Algunas de las principales ventajas de *KVM* se resumen a continuación:

- *KVM* viene incluido como un módulo del Kernel, por lo tanto no es necesario instalar ningún software específico, ya que lo único que se necesita es tener una máquina con el procesador adecuado, instala un Linux basado en el kernel y se dispondrá automáticamente de los servicios de virtualización.

- Ha sido la primera tecnología de virtualización que ha pasado a formar parte del núcleo de Linux.
- KVM permite a los hipervisores un rendimiento mayor en modo virtualización completa.
- KVM es el hipervisor del kernel de Linux, por lo tanto el host tiene que ser Linux, pero puede tener de guests a Linux, Windows, Solaris y BSD.
- KVM es el sistema de virtualización "oficial" de Debian y Ubuntu, funciona con todas las versiones del kernel Linux a partir de 2.6.20 [3].
- KVM se completa con la versión de QEMU que proporciona toda la funcionalidad de Hardware virtual para los sistemas virtualizados.
- No se modifica el kernel de GNU/Linux.
- KVM tiene administración vía Web, gráfica y consola por medio de un usuario mortal.
- Maneja tres tipos de configuración de red : Bridge, Route, NAT.
- Migración en caliente de máquinas virtuales.
- Permite ejecutar múltiples máquinas virtuales cada una con su propia instancia.

Una limitación para implementar KVM es que se necesita un procesador con soporte para virtualización por hardware, como son los procesadores con tecnologías AMD-V o Intel-VT.

#### 4.2.2.6. Arquitectura de virtualización de KVM

La integración de las tecnologías de virtualización en el nivel de procesador tanto Intel (Intel-VT) como AMD (AMD V) ha permitido a la virtualización ser implementada en el nivel del núcleo Linux, generando un mayor rendimiento y seguridad. Como se nombró anteriormente *KVM* es implementado como un módulo del *kernel*, al cargarse *KVM*, Linux se convierte en un hipervisor, permitiendo soportar huéspedes que pueden ejecutar tanto sistemas Windows y Linux. Los sistemas operativos de las máquinas virtuales invitadas se ejecutan en el espacio de usuario de núcleo Linux anfitrión y se comunican a través de un proceso *QEMU* con el módulo *KVM* que ha sido integrado en el kernel para así obtener acceso al hardware virtualizado. El módulo de *KVM* introduce un nuevo modo de ejecución en el núcleo de Linux adicional a los modos *kernel* y *user* que de por sí aporta el kernel de forma estandarizada llamado *guest*[3]. Las máquina virtuales se implementan como un proceso de Linux proporcionando aislamiento de cada guest, control y aprovechamiento en los recursos hardware, ya que

cualquier dispositivo compatible con Linux pueden ser utilizados por *KVM*.

Para proporcionar el hardware disponible de cada máquina virtual como Disco Duro, Tarjetas de Red, Unidades de CD, *KVM* hace uso de *QEMU* figura 4.3<sup>9</sup> *QEMU* *QEMU* es un emulador de máquinas y virtualizador genérico open source, cuando se usa como emulador de máquinas, *QEMU* puede ejecutar sistemas operativos y programas compilados para una arquitectura (por ejemplo un sistema ARM) en una máquina diferente (por ejemplo una PC)[30]. Cuando se utiliza como virtualizador, *QEMU* ejecuta el código guest directamente en el CPU host. *QEMU* soporta virtualización cuando se ejecuta a través del módulo del kernel *KVM* en Linux[30].

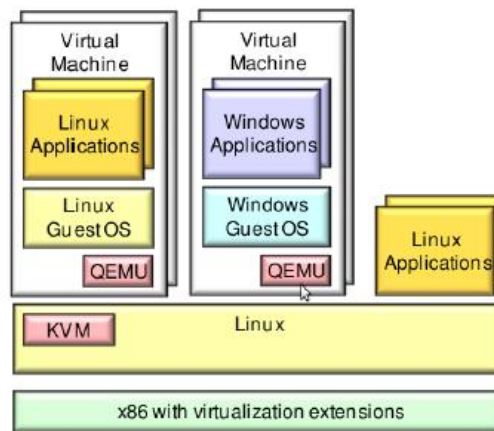


Figura 4.3: Arquitectura de virtualización KVM

Fuente: Fuente Digital<sup>9</sup>

Otros paquetes de *KVM* son nombradas a continuación.

El paquete *kvm* como se describió anteriormente, este paquete contiene el módulo de kernel de *KVM* que proporciona el hipervisor *KVM*, sobre el kernel de sistema operativo predeterminado.

En *Qemu/KVM* es posible crear máquinas virtuales fundamentalmente de dos formas diferentes: Utilizando la herramienta de forma gráfica **Virtual Machine Manager** (*virt-manager*) ó en línea de comandos **virt-install** (*virtinst*).

*KVM* está formado por un núcleo con el nombre de *kvm.ko*, que proporciona la infraestructura básica de virtualización y un módulo específico de procesador,

<sup>9</sup>Tomada del material: *Virtualización de Servidores KVM: Kernel-based Virtual Machine por Gonzalo Nazareno*. <http://www.gonzalonazareno.org/cloud/material/KVM.pdf>

*kvm-intel.ko* ó también *kvm-amd.ko*.

El paquete *python-virtinst* Proporciona el comando *virt-install* para crear máquinas virtuales.

El paquete *virt-install* es un script de python desarrollado por RedHat para la provisión desatendida de guests, se puede utilizar este comando para crear huéspedes virtualizados desde la línea de comandos.

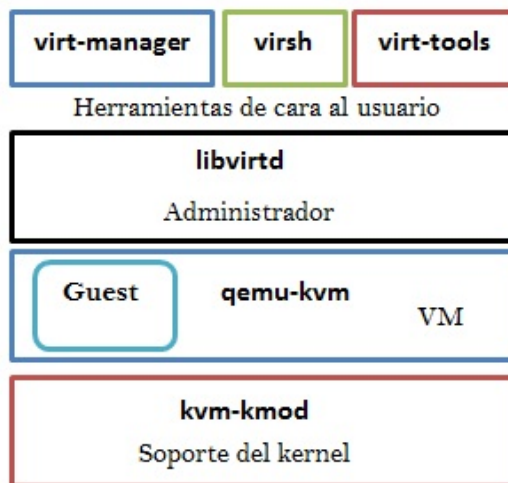


Figura 4.4: Arquitectura de virtualización KVM libvirt.

Fuente: Autora

El paquete *libvirt* es una interfaz de aplicación de programadores multiplataforma (API, por las iniciales en inglés *Application Programmers Interface*) que permite interactuar con los sistemas del hipervisor y del equipo. *libvirt* administra los sistemas y controla el hipervisor, ofrece la herramienta de línea de comando *virsh* para administrar y controlar huéspedes virtualizados e hipervisores ya sea desde la línea de comando o desde una shell de virtualización especial[22]. Muchas herramientas utilizadas de alto nivel como *virsh* ó *virt-manager* han sido construidas sobre la librería base *libvirt*. La librería *libvirt*, por defecto crea para las máquinas virtuales una red privada en el equipo host. Esta red privada utiliza la dirección de subred 192.168.x.x.

El paquete *virt-manager*, es Herramienta de Gestión gráfica para las máquinas virtuales de un host. Es una aplicación escrita en Python y distribuida bajo licencia GPL. Cuando se ejecuta *virt-manager* se puede acceder a un listado de dominios actualmente activos en la plataforma, así como su rendimiento y consumo de recursos como procesador y memoria. Proporciona un asistente gráfico para la creación de dominios o

la configuración y edición de los que ya existen. Incluye un asistente VNC embebido de forma que se puede acceder a las consolas gráficas de los dominios de una forma sencilla.

A continuación se muestran algunas funciones de *virt-manager*[20]:

- Conexión con múltiples servidores de máquinas virtuales: KVM y XEN.
- Gestión de redes virtuales.
- Permite el acceso a la consola gráfica de la Máquina virtual.
- Muestra estadísticas de rendimiento.
- Permite conectarse a hipervisores remotos.
- Permite consultar las máquinas virtuales disponibles en un servidor junto con su estado y los recursos que utilizan.
- Ejecuta y detiene máquinas virtuales.
- Consulta la información y estado del servidor de máquinas virtuales.
- Administra la información de las redes virtuales y los almacenamientos.
- Información de las máquinas virtuales.
- Permite crear nuevas máquinas virtuales.
- Permite remover máquinas virtuales disponibles.
- Permite conectarse con la consola gráfica de las máquinas virtuales en ejecución a través de VNC.

El paquete *virsh* es la principal interfaz de gestión de máquinas virtuales *Qemu/KVM* mediante línea de comandos. *virsh* permite iniciar, pausar y apagar máquinas virtuales, también permite otras funciones como gestión de la red, trabaja sobre el conjunto de herramientas libvirt para interactuar con las herramientas de virtualización más usadas: *Qemu/KVM* y *Xen*.

#### 4.2.2.7. Máquinas Virtuales KVM

Creadas, configuradas, inicializadas y vigiladas por la capa de virtualización, las máquinas virtuales actúan como equipos totalmente independientes y disponen de su propio hardware virtual, sistema operativo y aplicaciones. Como se muestra en el esquema de red cada máquina virtual dispondrá de un sistema operativo *Debian 7.1.0-Wheezy* como sistema invitado y serán configuradas con distintas tecnologías para soportar múltiples aplicaciones.

Para una administración eficiente de las máquinas ocuparemos dos herramienta bajo la librería *libvirt*, la herramienta *virsh*, usada para la administración en consola y *virt-manager* para la administración en modo gráfica. En el documento [ANEXO A], se describirán los dos modos de administración gráfica y consola para las máquinas virtuales.

#### 4.2.2.8. Proxy inverso

Para redirigir las solicitudes de los usuarios a cualquier aplicación alojada dentro de una de las máquinas virtuales de la infraestructura, se utiliza un *Proxy inverso*. Un proxy inverso es una tecnología que centraliza el tráfico entrante de internet hacia nuestra red local. Un proxy es un punto intermedio entre una computadora conectada a internet y el servidor que está accediendo. El método consiste en que cuando una solicitud llega al proxy, este la lee y se conecta al servidor, reúne la información y genera una respuesta. La solicitud nunca llega hasta el servidor directamente, sin embargo para el usuario este proceso es completamente transparente ya que cree que está conectado directamente con el servidor destino.

Otro aspecto importante del *Proxy inverso* es que ayuda a proteger los recursos de la red de amenazas exteriores, es útil por varios motivos posibles como seguridad, rendimiento y anonimato. Como se muestra en la figura *Figura 4.2* las máquinas virtuales están configuradas bajo el dominio *http://simon.uis.edu.co* y al implementar el *Proxy inverso* se redirigen las peticiones de los usuarios a *http://simon.uis.edu.co/nombreakaplicacion*.

*Actividad 3:* Implementar la arquitectura elegida e instalar los servicios del servidor gestionando y dimensionando adecuadamente los recursos, buscando un rendimiento óptimo de la infraestructura.

Para la implementación de la infraestructura elegida se van a seguir los siguientes items<sup>10</sup>:

- 1 Introducción y requisitos previos.
- 2 Instalación del sistema operativo base – *Debian 7.1.0 - Wheezy* en el servidor *Proliant ML110 G7*.
- 3 Instalación y configuración de *KVM* en el servidor.
- 4 Instalación de *VirtualBox* y el sistema de escritorio de la estación de trabajo de gestión, equipo *Dell OpTipler 740*.
- 5 Instalación y configuración de *virt-manager* en la estación de trabajo.
- 6 Despliegue de las Máquinas Virtuales y montaje de los recursos tecnológicos y las aplicaciones en a cada máquina.
- 7 Configuración del servidor *proxy inverso*

#### **4.2.2.9. Introducción y requisitos previos.**

Para comenzar, el primer requisito es disponer de un equipo que actúe como servidor host. La máquina debe admitir extensiones de virtualización de hardware AMD (AMD-V) o Intel(VT-x).

Estas son algunas de las especificaciones mínimas para implementar una infraestructura virtual en un servidor.

- CPU AMD o Intel de doble núcleo con soporte de virtualización de hardware.
- 4 GB o más de RAM..
- 10/100 Ethernet apoyo.
- 7200 RPM IDE de disco duro con 80 GB de espacio en disco.
- Unidad de CD-ROM (puede ser USB / externa)

---

<sup>10</sup>Algunos de los pasos para la implementación de estas herramientas fueron tomadas del tutorial Manual de Instalación de KVM que mostrará cómo configurar el servidor con virtualización KVM: <http://www.palegray.net/tutorials/debian-6-kvm-howto>

- Unidades de disco duro SATA Fast, 500 GB o más de espacio de disco de hardware RAID-1 o RAID-10, para máquinas virtuales.

El servidor host utilizado para el desarrollo de este proyecto es un HP Proliant ML110 G7, tiene un procesador Intel Xeon E3-1220 de cuatro núcleos, 16 GB de Memoria máxima, tecnología multipolar Quad-core, un disco duro SATA de 100 para el sistema operativo host y dos discos duros de 500 GB SATA configurado en RAID-1+0 por software. para ver más especificaciones del servidor host, se muestran en el *Tabla 4.8: Especificaciones técnicas servidor Hp Proliant ML110 G7.*

Los pasos siguientes para la implementación de la infraestructura se muestran como documentos Anexos. Instalación del sistema operativo base – *Debian 7.1.0 - Wheezy* en el servidor, documento [ANEXO B]. Instalación y configuración de *KVM* en el servidor, documento [ANEXO C]. Instalación de *VirtualBox* y el sistema de escritorio en la estación de trabajo de gestión, equipo *Dell OpTipler 740*, documento [ANEXO D]. Instalación y configuración de *virt-manager* en la estación de trabajo, documento [ANEXO E]. Despliegue de las Máquinas Virtuales y montaje de los recursos tecnológicos y las aplicaciones en a cada máquina documento [ANEXO F] y en el documento [ANEXO G] se describen los pasos para la configuración del *proxy inverso* en nuestra máquina anfitrión, teniendo en cuenta que las solicitudes se redirigen sobre el dominio <http://simon.wis.edu.co/nombreaplicacion>.

*Actividad 4:* Elaborar un Catálogo de servicios para los miembros del Grupo Simon, con el fin de que conozcan las aplicaciones y recursos disponibles, y elijan los servicios que más les convenga para el alojamiento de sus aplicaciones.

#### 4.2.2.10. Catálogo de Servicios servidor Grupo Simon

El Catálogo de Servicios, es un documento donde se describen detalladamente los servicios que presta una organización y los recursos asignados, este sirve de guía a los usuarios a la hora de elegir las tecnologías que más se adapten a sus necesidades. El Catálogo de Servicios puesto a disposición de los miembros del grupo SIMON describe en primera instancia las *Características del Sistema Anfitrión*, luego las *Tecnologías de Virtualización del Sistema Anfitrión*, *Características del equipo alterno* y por último describe las *Características propias de las Máquinas Virtuales* como se muestra a continuación.

<b>Características del Sistema Anfitrión: HP Proliant ML 110.</b>	
Memoria	16 GB.
Disco duro	1x250 GB. estándar Serial ATA-300
Configuración de RAID	RAID 1+0
Sistema Operativo Base	<i>debian-7.1.0 Whezzy</i> – modo consola, de GNU/Linux, adaptación AMD64.
Conexión de redes	Adaptador de red-PCI Express 2.0x1 integrado
Configuración de red	Interfaz eth0: 192.168.65.239(IP), 255.255.255.0(mascara de red). Virbr0: 192.168.122.1(IP), 255.255.255.0(mascara de red)
Dominio	<i>http://simon.uis.edu.co/</i>
Protocolo de interconexión de datos	Ethernet, Fast Ethernet, Gigabit Ethernet
Elemento virtualizador	KVM
Tecnología para el redireccionamiento IP	Proxy inverso
Bacukps automáticos	Bacula
Sistema de monitorización de la infraestructura virtual	Nagios3

<b>Tecnologías de Virtualización del Sistema Anfitrión</b>	
<i>qemu-kvm</i>	Hipervisor KVM, que permite ejecutar máquinas virtuales utilizando imágenes de disco que contienen sistemas operativos sin modificar. Cada máquina virtual tiene su propio hardware virtualizado: una tarjeta de red, discos duros, tarjeta gráfica, etc. KVM proporciona la infraestructura de virtualización y posee un componente <i>kvm-ko</i> incluido en el núcleo de Linux y es en su totalidad software libre. KVM también requiere un QEMU un emulador que proporciona toda la funcionalidad de hardware virtual para los sistemas virtualizados.
<i>libvirt-bin</i>	Herramientas para la administración de <i>KVM</i> y <i>libvirt</i> . Libvirt es una herramienta para la gestión y administración de la virtualización de plataforma.
<i>python-virtinst</i>	Paquete que proporciona el comando <i>virt-install</i> para crear máquinas virtuales.
<i>virt-install</i>	Script de python desarrollado por RedHat para la provisión desatendida de guests, se puede utilizar este comando para crear huéspedes virtualizados desde la línea de comandos.
<i>bridge-utils</i>	Paquete encargado de la configuración de la interfaz de red sobre la que circula el tráfico desde internet a la red interna. Permite crear un Puente entre las tarjetas de red física y las virtuales.
<i>screen</i>	herramienta útil de administración en modo texto. Permite tener varias consolas abiertas en una terminal.
<i>debconf</i>	Sistema de configuración de paquetes de Debian
<i>virt-manager</i>	Herramienta de Gestión gráfica para las máquinas virtuales de un host. Es una aplicación escrita en Python y distribuida bajo licencia GPL. Incluye un asistente VNC embebido de forma que se puede acceder a las consolas gráficas de los dominios de una forma sencilla.
<i>virsh</i>	Es la principal interfaz de gestión de máquinas virtuales <i>Qemu/KVM</i> mediante línea de comandos. <i>virsh</i> permite iniciar, pausar y apagar máquinas virtuales, también permite otras funciones como gestión de la red, de los <i>pools</i> de almacenamiento, de volúmenes entre otros. Trabaja sobre el conjunto de herramientas libvirt para interactuar con las herramientas de virtualización más usadas: <i>Qemu/KVM</i> y <i>Xen</i> .

<b>Características del equipo alterno Dell Optiplex 740 (Administrador)</b>	
<i>VirtualBox</i>	Se utilizará VirtualBox para albergar un sistema de escritorio dedicado <i>Debian-7.1.0 Whezzy</i> – modo consola, responsable de la administración de las máquinas virtuales que se ejecutan en el Servidor <i>HP Proliant</i> .
<i>AdministradorSimon</i>	Nombre descriptivo de la estación de trabajo para la administración de las Máquinas Virtuales
Sistema Operativo Base	<i>Debian-7.1.0 Whezzy</i> – modo gráfico.
Tamaño de memoria	512 MB
Sistema Operativo Base	<i>debian-7.1.0 Whezzy</i> – modo consola, de GNU/Linux, adaptación AMD64.
CPU.	1
Configuración de red	Interfaz eth0: 10.0.2.15(IP), 255.255.255.0(mascara de red). lo: 127.0.0.1(IP), 255.0.0.0(mascara de red)
<i>Virt-manager</i>	Herramienta de gestión gráfica de las Máquinas Virtuales.
<i>ssh-askpass</i>	Herramienta para conexión al Servidor
<i>bridge-utils</i>	Permite crear un Puente entre las tarjetas de red física y las virtuales.

<b>Características Máquina Virtual 1: SimonMV1</b>	
Sistema Operativo	debian-7.1.0 Wheezy
Tipo de Configuración	modo consola
CPUs	2
Memoria	512 MB.
Configuración de red	Interfaz eth0: 192.168.122.163(IP), 255.255.255.0(mascara de red). lo: 127.0.0.1(IP), 255.0.0.0(mascara de red)
Dominios a cargo	<i>http://simon.uis.edu.co/joomla/home/</i> <i>http://simon.uis.edu.co/11encuentros</i> <i>http://simon.uis.edu.co/redescuela</i> <i>http://simon.uis.edu.co/redinamica</i>
Aplicaciones	Página Web del Grupo SIMON. Página Web del Encuentro Colombiano de Dinámica de Sistemas. Página Web Red Escuela Página Web Red dinámica.
Recursos tecnológicos	Apache2 Mysql PHP5 PHPmyadmin Joomla

<b>Características Máquina Virtual 2: SimonMV2</b>	
Sistema Operativo	debian-7.1.0 Whezzy
Tipo de Configuración	modo consola
CPUs	2
Memoria	512 MB.
Configuración de red	Interfaz eth0: 192.168.122.123(IP), 255.255.255.0(mascara de red). lo: 127.0.0.1(IP), 255.0.0.0(mascara de red)
Dominios a cargo	<i>http://simon.uis.edu.co/pesco</i> <i>http://simon.uis.edu.co/avasmart</i> <i>http://simon.uis.edu.co/saludmovil</i>
Aplicaciones	Ambiente software, para el aprendizaje y la toma de decisiones, PESCO CONOCIMIENTO-SIMON Ambiente software, para el aprendizaje y la toma de decisiones en redes inteligentes de SMART GRID. Sistema de Información Educación y Comunicación en enfermedades infecciosas que cursan con fiebres agudas, SIDEF-MOVIL
Recursos tecnológicos	Apache2 mysql-server, mysql-client phpmyadmin PHP5 Joomla-3.0.3 WebMin

<b>Características Máquina Virtual 3: SimonMV3</b>	
Sistema Operativo	debian-7.1.0 Whezzy.
Tipo de Configuración	modo gráfico, Máquina virtual para pruebas internas dentro de la red <i>libvirt</i> . Gracias a su modo de gráfico esta máquina también aloja el sistema de monitorización de la infraestructura virtual <i>nagios3</i> .
CPUs	2
Memoria	512 MB.
Configuración de red	Interfaz eth0: 192.168.122.169(IP), 255.255.255.0(mascara de red). lo: 127.0.0.1(IP), 255.0.0.0(mascara de red)
Recursos tecnológicos	Apache2 Mysql PHP5 apache nagios3

<b>Características Máquina Virtual 4: SimonMV4</b>	
Sistema Operativo	debian-7.1.0 Whezzy
Tipo de Configuración	modo consola
CPUs	2
Memoria	512 MB.
Configuración de red	Interfaz eth0: 192.168.122.196(IP), 255.255.255.0(mascara de red). lo: 127.0.0.1(IP), 255.0.0.0(mascara de red)
Dominios a cargo	<i>http://simon.wis.edu.co/ecogranja</i> <i>http://simon.wis.edu.co/evolución Web</i>
Aplicaciones	Ambiente para el aprendizaje y la toma de decisiones en granjas agropecuarias, basado en la dinámica de sistemas y soportado por tecnologías Web Herramienta de modelado y simulación en la Web – Evolución Web
Recursos tecnológicos	PosgreSQL Ruby y Javascript del lado servidor OpenSSH RVM (Ruby versión Manager) Ruby onRails Passenger Apache 2

*Actividad 5:* Diseñar las políticas de administración y seguridad necesarias para el buen desempeño y rendimiento de la Infraestructura.

#### 4.2.2.11. Políticas de administración y seguridad

A continuación se describirán las políticas de Administración y Seguridad recomendadas para el control y el el buen uso de los recursos del servidor del Grupo Simon. Para la definición de estas políticas se han tenido en cuenta los conceptos de confidencialidad, disponibilidad e integridad, que nos ofrecen ITIL<sup>11</sup>.

<b>Política 1: Control de puertos de comunicación del Sistema</b>	
Objetivo	Controlar el acceso físico y lógico al sistema, asegurando que los puertos no requeridos estén desactivados o removidos.
Descripción	Mientras menos puertos se tengan abiertos menos posibilidad hay de que los intrusos puedan acceder al servidor.
Procedimiento	Para ver los puertos usados en el sistema se usa la herramienta <i>nmap</i> , que permite el escaneo de los puertos de la máquina y establece si un puerto está abierto, cerrado o protegido por un cortafuegos. Así, es capaz de identificar máquinas dentro de una red, determinar qué servicios utiliza dicha máquina, definir cuál es su sistema operativo e incluso devolver cierta información sobre el hardware de la máquina. En el documento <i>Bitácora interna de administración del Servidor del Grupo Simon</i> se encuentra la descripción detallada de los puertos habilitados del sistema, y la forma de como desabilitarlos cuando no están en uso.

---

<sup>11</sup>La confidencialidad implica que la información solo sea accedida por los usuarios autorizados. La disponibilidad se refiere a la necesidad de ofrecer un servicio sin interrupciones y que puedas ser accedido en cualquier momento y desde cualquier lugar. La integridad consiste en la necesidad de mantener la información inalterada

<b>Política 2: Bloqueos de IPs por intentos de accesos fallidos al sistema</b>	
Objetivo	Asegurar la protección de la infraestructura de soporte por medio de bloqueos a intentos fallidos de accesos al sistema.
Descripción	Para evitar que algunos usuarios no autorizados ingresen al sistema, es necesario de una herramienta que permita el bloqueo de dichas Ips.
Procedimiento	Para llevar a cabo esta política se instala y configura la Herramienta <i>Fail2ban</i> , escrita en Python, usada para la prevención de intrusiones en el sistema. <i>Fail2ban</i> bloquea las direcciones de Internet de donde se hayan originado varios intentos fallidos de acceso con contraseñas inválidas y si los encuentra rechazará esos intentos y bloqueará (con reglas iptables ) a las IP orígenes que intentan estos accesos por fuerza bruta al sistema. El procedimiento de configuración de esta práctica se encuentra descrito en el documento [ANEXO H] y con más detalles en el documento <i>Bitácora interna de administración del Servidor del Grupo Simon</i>

<b>Política 3: Control de contraseñas seguras</b>	
Objetivo	Asegurar el acceso del usuario autorizado y evitar el acceso a los no autorizados, por medio de contraseñas seguras.
Descripción	Para poder asegurar el correcto funcionamiento del sistema host es necesario que se disponga de mecanismos para la protección y aseguramiento de los datos de cada usuario. Solo el superusuario <i>root</i> es la entidad responsable del funcionamiento y el control total del sistema host, no suele tener restricción de seguridad, al no tener restricción y tener acceso universal del sistema, esto puede generar consecuencias graves ya que puede realizar cualquier operación, facilitando la posibilidad de descargar o ejecutar software dañino, o cambiar alguna configuración generando problemas graves.

<p>Procedimiento</p>	<p>Para evitar en lo posible errores o daños por causa de accesos no autorizados se recomienda que para la mayoría de las actividades se deba usar una cuenta de usuario diferente a la del <i>root</i>, disponer de un diferente usuario y su contraseña.</p> <p>Tanto la contraseña del super usuario y el usuario deben cumplir con ciertos requisitos como lo son:</p> <ul style="list-style-type: none"> <li>■ Cualquier contraseña deben ser compleja e ir combinada con letras mayúsculas, minúsculas, números, caracteres especiales.</li> <li>■ Mantener confidenciales cualquier tipo de clave.</li> <li>■ No usar nombres comunes o nombre propios, ni tampoco secuencias contiguas de números o letras.</li> <li>■ Las contraseñas no deben ser de longitud corta, se recomienda usar contraseña de un mínimo de 8 caracteres.</li> <li>■ Si se mantiene un registro de claves del sistema, este debe mantenerse guardado de manera segura.</li> <li>■ Cuando exista un indicio o sospechas de posibles peligros y alteraciones al sistema se recomienda un cambios de clave.</li> <li>■ Las contraseñas se deben cambiar periódicamente.</li> <li>■ Cuando el usuario termine sus labores debe asegurarse que ha cerrado las sesiones activas.</li> <li>■ No utilizar la misma contraseña para las MV y/o servidor físico.</li> <li>■ Cada máquina virtual debe tener sus propias contraseñas tanto del usuario como del superusuario (siempre y cuando se cumpla con los requisitos establecidos anteriormente.)</li> </ul>
----------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<b>Política 4: Aseguramiento de la infraestructura contra amenazas externas</b>	
Objetivo	Evitar el acceso no autorizado a los servicios de toda la red.
Descripción	El tráfico en Internet generalmente hace que las redes sean vulnerables a cualquier tipo de ataque o amenaza. La mejor forma de proteger la red es mediante bloqueos a las solicitudes entrantes. Para asegurar la infraestructura se configura un <i>servidor proxy inverso</i> encargado de atender el tráfico entrante ofreciendo una única puerta de entrada a la red protegiéndola de amenazas exteriores. Un proxy inverso, es un servidor proxy que centraliza el tráfico entrante de internet hacia la red local. Un proxy es un punto intermedio entre una computadora conectada al internet y el servidor que está accediendo.
Procedimiento	El método del <i>proxy inverso</i> consiste en que cuando una solicitud llega al proxy, este la lee y se conecta al servidor, reúne la información y genera una respuesta, la solicitud nunca llega hasta el servidor directamente, sin embargo para el usuario este proceso es completamente transparente ya que cree que está conectado con el servidor destino. La finalidad de un proxy inverso es ayudar a proteger los recursos de la red de amenazas exteriores e interceptar las conexiones con la Web, es útil por varios motivos posibles como seguridad, rendimiento y anonimato. La configuración del <i>proxy inverso</i> se encuentra en el documento [ANEXO G] y mas detalladamente en el documento <i>Bitácora interna de administración del Servidor del Grupo Simon</i>

<b>Política 5: Aseguramiento de la disponibilidad continua de los datos</b>	
Objetivo	Asegurar la disponibilidad continua de los datos en caso de falla.
Descripción	Se configura un RAID (Conjunto Redundante de Discos Independientes). Un RAID es un sistema de almacenamiento de datos que usa múltiples unidades de almacenamiento de datos (discos duros), entre los que se distribuyen o replican los datos, para conseguir mayor protección y un mejor rendimiento, por lo tanto un RAID es una técnica que permiten duplicar la información almacenada en los discos duros para evitar pérdida de datos en caso de que uno de ellos sufra algún problema. Un RAID combina varios discos duros en una sola unidad lógica, tanto el usuario como el Sistema Operativo solo ven un disco duro, pero en realidad la información es almacenada en todos los discos. El uso de múltiples discos duros con información duplicada y distribuida posee las siguientes ventajas: mayor integridad, mayor seguridad en la información (al dañarse un disco duro, la información sigue estando en forma correcta duplicada en otro disco), mayor tolerancia a fallos, mayor rendimiento, mayor capacidad, mayor velocidad en lectura y escritura.
Procedimiento	Se configura un RAID 1+0, construido a partir de cuatro particiones RAID. El esquema consiste en dos pares de particiones, cada una de las cuales refleja a la otra, los datos son distribuidos sobre ambas particiones. El sistema RAID evitará en lo más posible la pérdida de datos de la siguiente manera: Los discos optimizados para RAID poseen circuitos integrados que detectan si el disco está fallando, de ser así este circuito se encargará de sacar la información redundante de los discos buenos y regenerar la información del disco dañado, de forma automática, la unidad continúa funcionando, sin pérdida de tiempo. RAID 1+0 : Es una combinación de un RAID 1 un RAID 0, dispone de la redundancia y protección de datos del RAID 1 y el nivel de rendimiento del RAID 0. El RAID 1+0 duplica en espejo los datos y después distribuye el resultado por los discos. (Se requiere un mínimo de 4 discos.)

<b>Política 6: Respaldo o Back-Up de los datos del sistema.</b>	
Objetivo	Mantener la integridad y disponibilidad de la información del sistema, por medio de respaldos Back-Up.
Descripción	Se deberán establecer los procedimientos de rutina para implementar los respaldos a los datos del sistema y a sí mismo se deberán proporcionar medios de respaldo adecuados para asegurar que toda la información esencial y software se pueda recuperar después de una falla.
Procedimiento	Para cubrir eficientemente las necesidades de respaldo de la infraestructura virtual, se configura la herramienta <i>Bacula</i> . <i>Bacula</i> ofrece un sistema completo, distribuido y libre encargado de secuenciar los trabajos de copias y gestionar el medio de almacenamiento de las mismas. <i>Bacula</i> es un sistema de creación de copias de seguridad capaz de cubrir eficientemente las necesidades de respaldo de equipos bajo una red IP. <i>Bacula</i> también facilita hallar y recuperar archivos dañados. El procedimiento de configuración de <i>Bacula</i> se encuentra descrito en el documento [ANEXO I] y con más detalles en el documento <i>Bitácora interna de administración del Servidor del Grupo Simon</i>

<b>Política 7: Monitoreo y revisión de los servicios de la infraestructura.</b>	
Objetivo	Monitorear y revisar regularmente las actividades de monitoreo de la infraestructura para prevenir fallas y errores y asegurar el desempeño requerido del sistema.
Descripción	Se deberán establecer los procedimientos de monitoreo para asegurar y, cuando sea necesario, mejorar la eficiencia y la disponibilidad de los recursos de la infraestructura y detectar las actividades no autorizadas que puedan causar problemas graves.
Procedimiento	Para las actividades de monitoreo de la infraestructura, se configura el sistema de monitorización <i>Nagios3</i> , encargado de monitorear los hosts y servicios de la infraestructura que se especifiquen, revisando regularmente que los servicios estén funcionando y disponibles, enviando alertas de cuando algo sale mal y cuando esta bien. <i>Nagios3</i> es una herramienta licenciada bajo GNU. En el documento [ANEXO J] se detalla más esta herramienta y su respectiva configuración también se encuentra descrita en el documento <i>Bitácora interna de administración del Servidor del Grupo Simon</i>

*Actividad 6:* Diseñar un documento interno donde se especifique la administración eficiente del servidor del Grupo SIMON, incluyendo políticas y recomendaciones generales.

En el documento *Bitácora interna de administración del Servidor del Grupo Simon* se proporcionará la guía de administración eficiente de la Infraestructura, incluye, las políticas de administración y seguridad del servidor y describe el proceso de implementación de cada política, los modos de uso de las herramientas, recomendaciones generales, entre otras. Este documento es de uso exclusivo del administrador del Grupo.

## 4.3. Fase 3: Transición del Servicio

La fase de Transición del Servicio, consiste en integrar y poner a disposición de los clientes y usuarios autorizados, para su uso, todos aquellos servicios definidos en la fase de Diseño, garantizando que estos servicios cumplan con los requisitos definidos en la fase de la Estrategia, buscando mejorar la satisfacción del cliente con respecto a los servicios prestados[18]. Es indispensable que durante esta fase todo lo que se realice esté planeado para asegurar la adecuación de los servicios, considerando medidas de retorno en casos necesarios, concluyendo todo el proceso y generando la documentación correspondiente.

### 4.3.1. Proceso de la Transición del servicio

El proceso definido en la fase de la Transición del servicio abarcado en este proyecto es la Validación y pruebas.

#### 4.3.1.1. Validación y pruebas

Este proceso consiste en garantizar que los servicios cumplan los requisitos acordados por el usuario y que estos servicios no generen error inesperado cuando estén operando. El objetivo del proceso consiste en detectar y prevenir aquellos errores causados por incompatibilidades imprevistas, y verificar que se cumplen los niveles de utilidad y garantía establecidos.

En este proceso, se validan los paquetes de servicios, se definen los modelos de pruebas, se construye un escenario de pruebas, y se accede a los elementos a probar, se aceptan los datos y elaborando los informes y registrando los errores generados. Finalmente, se limpia el entorno de pruebas y se cierra el proceso.

Para la realización de pruebas se incluyen las siguientes actividades:

- Pruebas del correcto funcionamiento de la versión.
- Pruebas de los procedimientos automáticos o manuales de instalación.
- Prueba para medir la utilidad del servicio.
- Prueba de carácter funcional

### 4.3.2. Implementación de la Transición del Servicio

Se proponen para esta fase realizar las siguientes actividades:

*Actividad 1:* Migrar los aplicativos implantados en el servidor actual al nuevo servidor.

*Actividad 2:* Realizar las pruebas necesarias de funcionalidad y utilidad de la infraestructura, para verificar el desempeño de la misma.

#### Desarrollo

*Actividad 1:* Migrar los aplicativos implantados en el servidor actual al nuevo servidor.

Se realiza la migración de los aplicativos implantados en el servidor actual al nuevo servidor. Esta migración está escrita en el documento "Bitacora de administración servidor Grupo Simon"

*Actividad 2:* Realizar las pruebas de funcionalidad de las herramientas necesarias para verificar el buen desempeño de la infraestructura.

Las pruebas garantizan que los servicios están acoplados a lo que el usuario espera, cumpliendo los requisitos y se ajustan a la disponibilidad, garantía, continuidad y seguridad exigida. El objetivo principal de este proceso es del proceso es comprobar que la entrega de los servicios proporciona los resultados esperados los usuarios y los interesados. A continuación se describen las pruebas realizadas y los resultados obtenidos.

#### 4.3.2.1. Prueba de soporte de virtualización KVM

Para saber si la CPU del equipo host *Hp Proliant* soporta virtualización y es compatible con KVM, se ejecuta el siguiente comando una vez instalado el sistema operativo base e iniciado sesión como usuario *root* al sistema.

```
egrep '(vmx|svm)' -color=always /proc/cpuinfo
```

Se deben ver varias líneas de salida, como se muestra en la *figura 4.5*, si por el contrario no se hubiera observado ningún mensaje en absoluto, la CPU no soportaría la virtualización.

```

Terminal (como superusuario)
Archivo Editar Ver Buscar Terminal Ayuda
root@simon:/home/simon# egrep '(vmx|svm)' --color=always /proc/cpuinfo
flags              : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe nx rdtscp lm constant_
tsc arch_perfmon pebs bts xtopology nonstop_tsc aperfmperf pni pclmulqdq dtes64
monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic po
pcent tsc_deadline_timer aes xsave avx lahf_lm ida arat xsaveopt pln pts dtherm t
pr_shadow vnmi flexpriority ept vpid
flags              : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe nx rdtscp lm constant_
tsc arch_perfmon pebs bts xtopology nonstop_tsc aperfmperf pni pclmulqdq dtes64
monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic po
pcent tsc_deadline_timer aes xsave avx lahf_lm ida arat xsaveopt pln pts dtherm t
pr_shadow vnmi flexpriority ept vpid
flags              : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe nx rdtscp lm constant_
tsc arch_perfmon pebs bts xtopology nonstop_tsc aperfmperf pni pclmulqdq dtes64
monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic po
pcent tsc_deadline_timer aes xsave avx lahf_lm ida arat xsaveopt pln pts dtherm t
pr_shadow vnmi flexpriority ept vpid
root@simon:/home/simon#

```

Figura 4.5: Nivel de soporte Virtualización KVM

Fuente: Autora

#### 4.3.2.2. Prueba de Funcionalidad *Virtual Machine Manager*

La aplicación *virt-manager* es una interfaz de usuario de escritorio para la gestión de máquinas virtuales a través de libvirt. Se dirige principalmente a las máquinas virtuales KVM, pero también maneja Xen y LXC (contenedores de Linux). Se presenta una visión resumida de los dominios de funcionamiento, sus estadísticas de utilización de presentaciones en vivo y recursos[29], como se muestra en la *figura. virt-manager* presenta una consola gráfica completa y permiten la creación de nuevos dominios, configuración y el ajuste de la asignación de recursos de un dominio y hardware virtual[29].

*Virtual Machine Manager* tiene la capacidad para albergar varios sistemas operativos al mismo tiempo en distintas máquinas virtuales, y permite la convivencia de las máquinas sin interferir en rendimiento entre ellas. Como podemos ver en la *figura 4.6* se tienen cuatro máquinas virtuales independientes configuradas cada una con diversas tecnologías a cargo de aplicaciones. En las siguientes *figuras* se muestran detalladamente el rendimiento del sistema tanto de la la máquina host como las máquinas virtuales

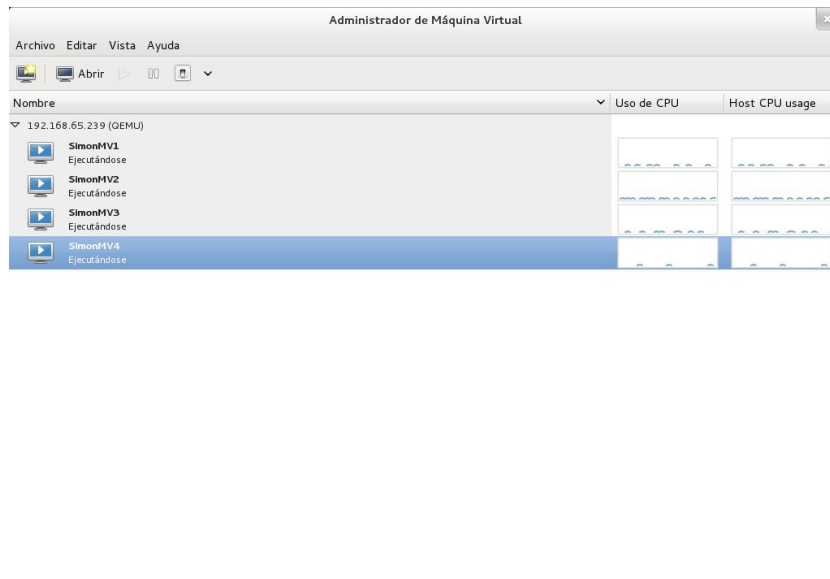


Figura 4.6: Administrador de Máquina Virtual  
Fuente: Autora

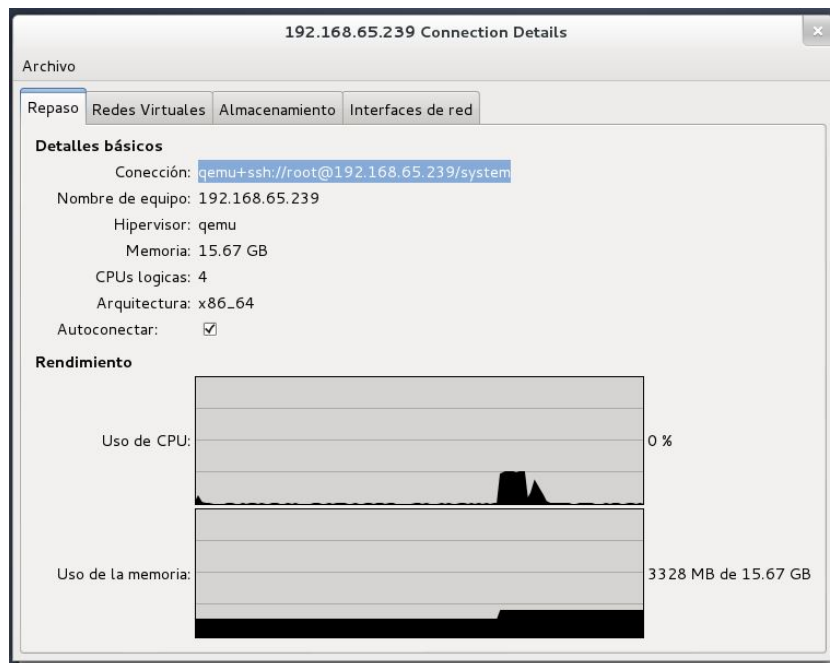


Figura 4.7: Rendimiento CPU y Memoria: Máquina Host.  
Fuente: Autora

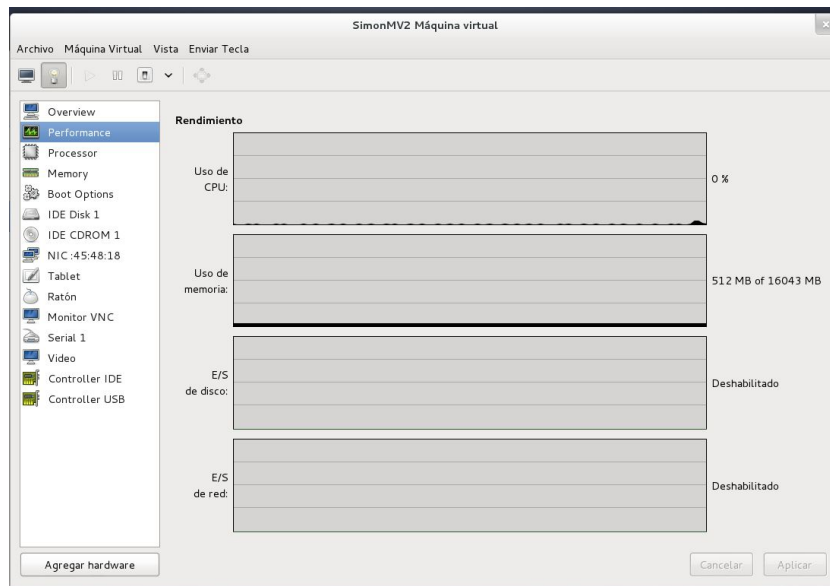


Figura 4.8: Rendimiento CPU y Memoria: SimonMV1  
Fuente: Autora



Figura 4.9: Monitor del Sistema: Recursos  
Fuente: Autora

Por otra parte hay que hacer énfasis que el consumo de memoria del sistema dependerá del número de máquinas virtuales que tengamos ejecutando en ese momento y que entre menos máquinas operen, el rendimiento es mayor. Como se puede observar la velocidad de ejecución del sistema es adecuada, la ejecución de las máquinas virtuales no provocan retrasos en el sistema operativo host.

#### 4.3.2.3. Pruebas de Conectividad y funcionamiento de las MV.

Antes del montaje de los servicios a las correspondientes máquinas virtuales se implementaron máquinas de pruebas en las que se llevaron a cabo pruebas de conexión. Primero se crea una máquina virtual con las distribución Debian 7, en esta máquina virtual se prueba conectividad por el protocolo SSH a la máquina host, esta conexión se muestra en la *figura 4.10*.

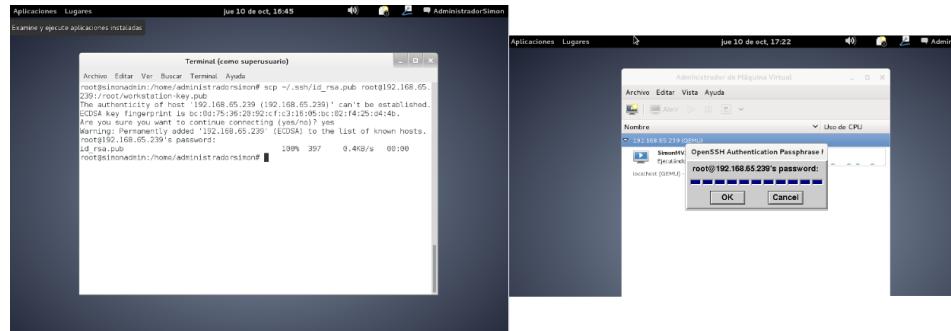


Figura 4.10: Prueba: Conexión ssh al Servidor Host  
Fuente: Autora

También se instalan algunos servicios como Mysql, PHP5 y Phpmyadmin para ver el funcionamiento de la máquina de prueba esto se resume en las siguientes *figuras*. así mismo se hicieron pruebas de guardado, suspensión, eliminación de las máquinas virtuales. Se especifica que para eliminar una máquina se debe primero apagar.

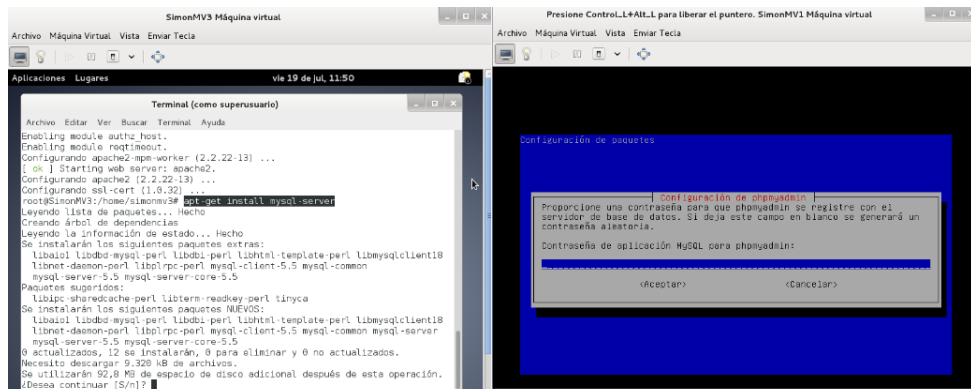


Figura 4.11: Prueba: Instalación mysql, Phpmyadmin  
Fuente: Autora

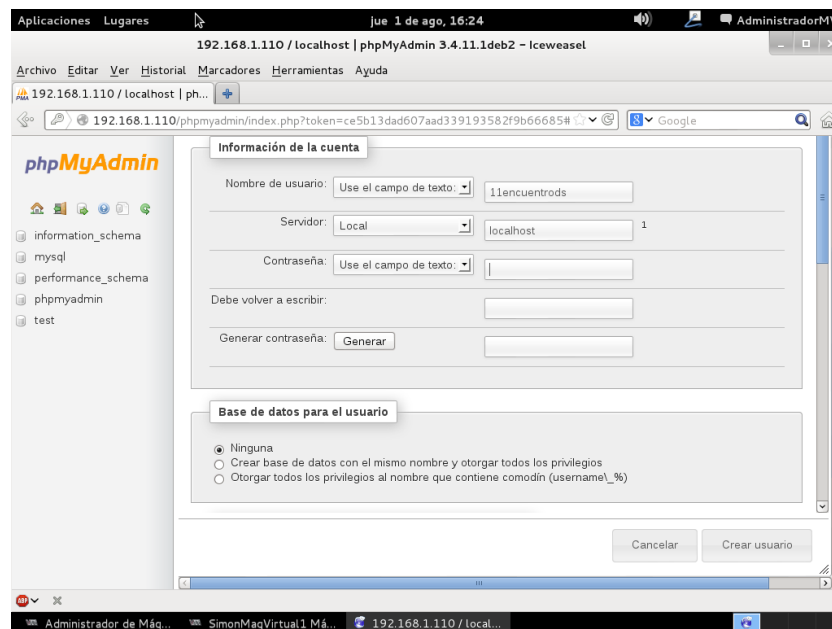


Figura 4.12: Prueba: Conexión por Phpmyadmin  
Fuente: Autora

#### 4.3.2.4. Prueba de Accesos no autorizados al *Host*

Se configura en el *host* la aplicación *fail2ban* para la prevenir accesos no autorizados por *ssh* al sistema físico. Esta herramienta bloquea las conexiones remotas que intentan accesos *ssh* por fuerza bruta al sistema.

La prueba de acceso se hace por medio de un *cliente ssh putty* instalado en un equipo

fuera de la red local. la *figura 4.13* muestra el acceso con contraseña correcta y la *figura 4.14* muestra el acceso denegado al sistema con contraseñas erróneas.

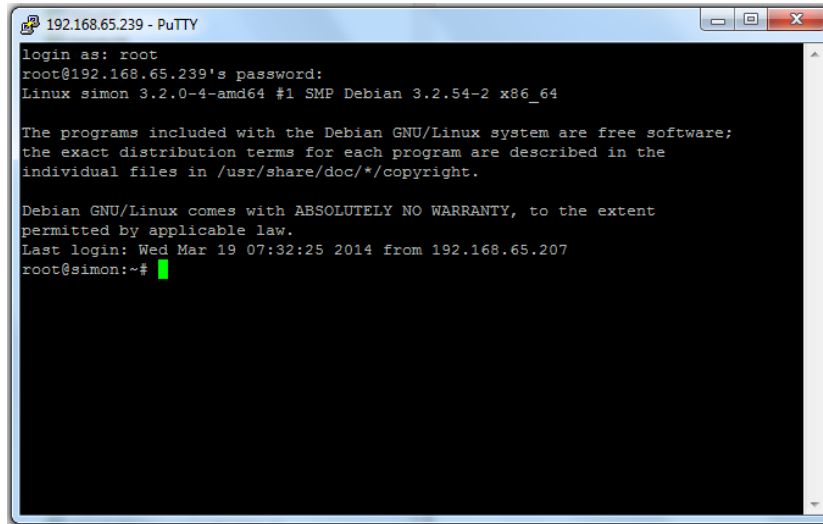


Figura 4.13: Prueba: Conexión ssh contraseña segura  
Fuente: Autora

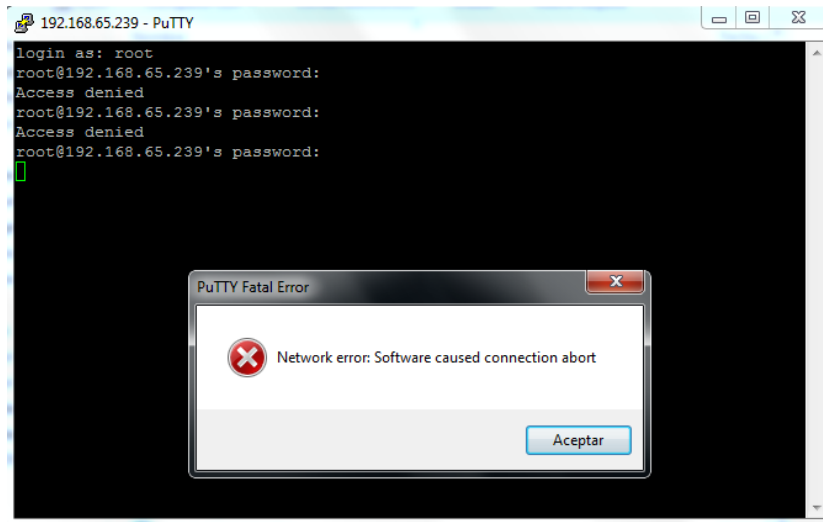


Figura 4.14: Prueba: Conexión ssh contraseña errónea  
Fuente: Autora

En el archivo de configuración de la herramienta *fail2ban*, se debe especificar la Ip a asegurar, el tiempo que tendrá que esperar el usuario que superó la cantidad de logueos para desbloquear su IP (tiempo dado en segundos) y el número de intentos. la siguiente *figura 4.15* muestra el mensaje si el usuario desea volver a intentar un acceso y no se ha desbloqueado su IP.

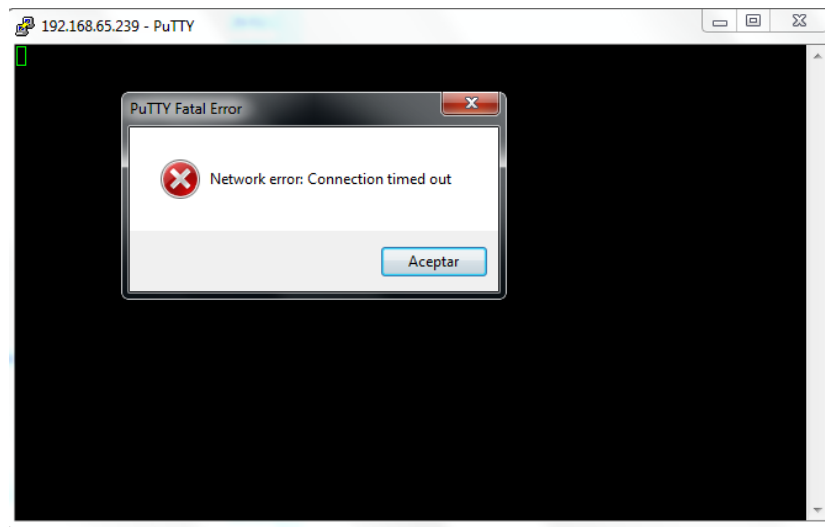


Figura 4.15: Prueba: Intento de acceso Ip bloqueada  
Fuente: Autora

## **4.4. Fase 4: Operación del Servicio**

La Fase de Operación del Servicio corresponde a la coordinación e implementación de las actividades y funciones necesarias para la prestación de los servicios definidos e implementados en las fases anteriores, servicios ofrecidos a proporcionar el soporte a los usuarios y de gestionar la infraestructura tecnológica necesaria para la prestación del servicio[15].

### **4.4.1. Procesos de la Operación del servicio**

Los procesos definidos en la fase de la Operación del Servicio abarcado en este proyecto son:

- 1 Gestión de Eventos.
- 2 Gestión de Problemas.
- 3 Gestión de Acceso a los Servicios TI.

#### **4.4.1.1. Gestión de Eventos**

La Gestión de Eventos es la responsable de monitorizar todos los eventos que acontezcan en la infraestructura TI con el objetivo de asegurar su correcto funcionamiento y ayudar a prever incidencias futuras[16]. Una vez que los servicios de la infraestructura estén operando se hace necesario monitorizarlos, esto ayuda a la detección temprana de incidentes, detecta, notifica y clasifica a los eventos determinando su impacto.

#### **4.4.1.2. Gestión de Problemas**

Proceso responsable de analizar y ofrecer soluciones a aquellos incidentes que por su frecuencia o impacto degradan la calidad del servicio[16]. Algunas de las funciones de la Gestión de Problemas son las de investigar las causas posibles de las alteraciones de un servicio, determinar las posibles soluciones, proponer cambios para restablecer el servicio, realizar revisiones.

La gestión de problemas genera un aumento en la calidad de los servicios, permite solucionar incidentes y los minimiza, generando disponibilidad continua de los servicios.

#### **4.4.1.3. Gestión de Acceso a los Servicios TI**

Proceso responsable de garantizar que sólo las personas con los permisos adecuados pueda acceder a la información de carácter restringido[16] y denegar a los usuarios no autorizados el acceso. La Gestión de Acceso pone en práctica las políticas y acciones

establecidas en la Gestión de la seguridad y la Gestión de disponibilidad.

La Gestión de Acceso a los servicios, proporciona una serie de ventajas a la organización TI: mayor garantía de confidencialidad de la información, gracias a un acceso controlado a los servicios, mayor respuesta efectiva a los usuarios del sistema, menor probabilidad de errores, capacidad de monitorización de servicios.

#### 4.4.2. Implementación de la Operación del Servicio

Se proponen para esta fase el desarrollo de las siguientes actividades:

*Actividad 1:* Monitorizar la Infraestructura virtual con el fin de detectar incidentes y si los hay hacer un estudio de las posibles soluciones.

##### Desarrollo

*Actividad 1:* Monitorizar la Infraestructura virtual con el fin de detectar incidentes y si los hay hacer un estudio de las posibles soluciones.

##### 4.4.2.1. Monitorización de la Infraestructura virtual

Para llevar a cabo esta actividad, se realizará la monitorización de la infraestructura por medio de la herramienta *nagios3* de libre utilización, que permite llevar un control completo de la disponibilidad de los servicios y recursos del sistema que se especifiquen en su archivo de configuración, para ver más detalles, instalación y configuración de la herramienta diríjase al documento [ANEXO J].

Como se ha indicado en apartados anteriores, la infraestructura virtual cuenta con cuatro máquinas virtuales, cada una con un sistema operativo base Debian 7, sus recursos hardware virtuales CPU, memoria, disco, a cargo de diversos recursos tecnológicos y aplicaciones. Como se muestra en la *figura 4.16*, en la máquina virtual *SimonMV3*, se encuentra configurada *nagios3* y desde ahí se monitorizara toda la infraestructura virtual.

Después de configurado *nagios3*, en un navegador web accedemos a *http://localhost/nagios3*, se debe introducir la cuneta de usuario y la contraseña establecidas, como se muestra en la *figura 4.17*, e ingresamos al área de trabajo, *figura 4.18*.

Ingresamos como usuario *root* a la consola de comando y ejecutamos el siguiente comando para verificar que no existan errores en el sistema.

```
# nagios3 -v /etc/nagios3/nagios.cfg
```

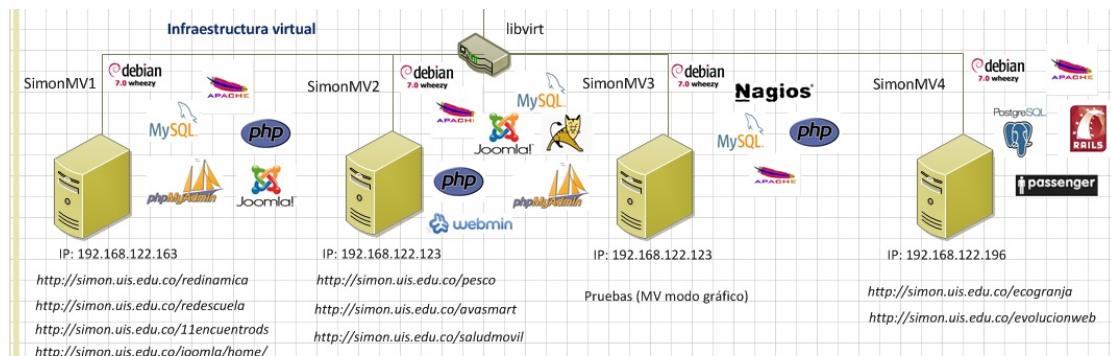


Figura 4.16: Diagrama infraestructura virtual

Fuente: Autora

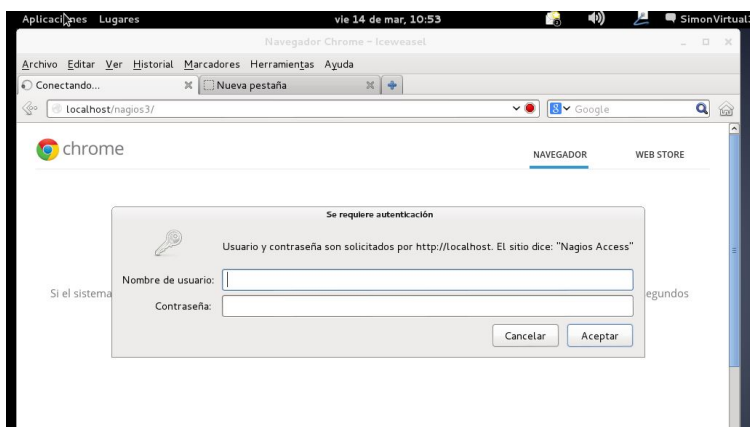


Figura 4.17: Ingreso a la herramienta nagios3

Fuente: Autora

Los resultados los podemos apreciar en la *figura 4.19*. Como podemos ver no se detectaron errores graves.

Para ver de una forma más general el estado de la red monitorizada, vamos al menú del área de trabajo de *nagios3*, y pinchamos en *Tactical Overview* ver *figura 4.20*. En la parte que se subraya en *Unhanded problems* aparecen los posibles alertas críticas, en este caso no tenemos, si lo fuera aparecerían en rojo y la cantidad.

Para ver la estructura monitorizada pinchamos *Map*, como nos muestra la *figura 4.21*, vemos la máquina localhos encargada de albergar el sistema de monitorización, y los equipos a monitorizar.

En la opción *Hosts* podemos ver la cantidad de equipos monitorizados, sus estados,

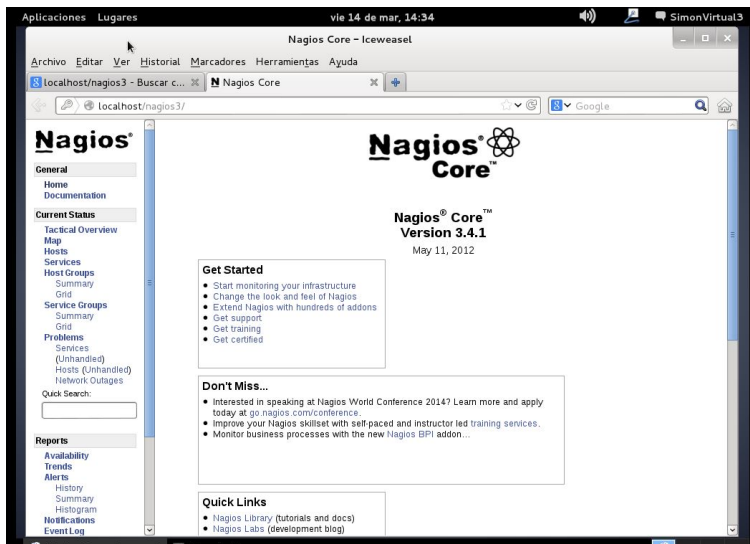


Figura 4.18: Area de trabajo de nagios3  
Fuente: Autora

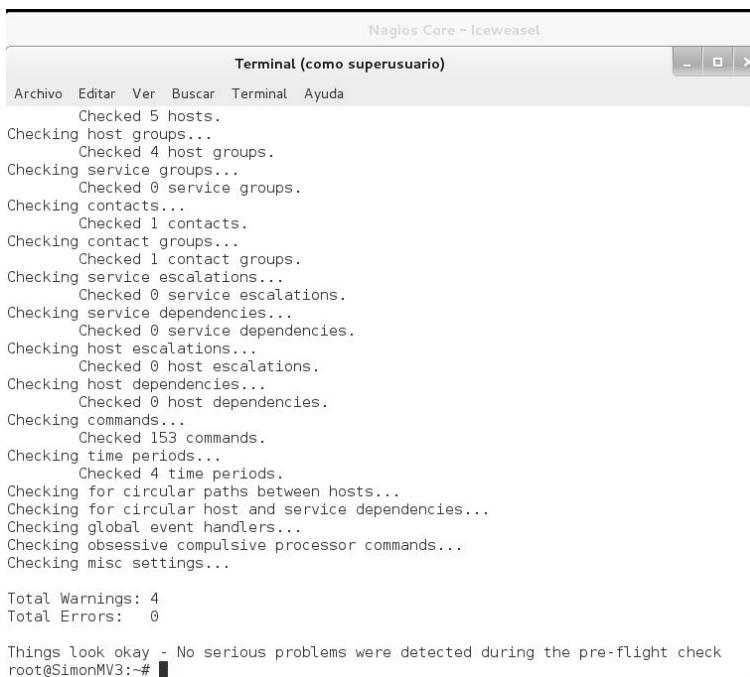


Figura 4.19: Detectar errores del sistema con nagios3.  
Fuente: Autora

los servicios predefinidos, y sus posibles problemas *figura 4.22*. Como se aprecia en la infraestructura la máquinas están en encendido y no se muestran problemas críticos.

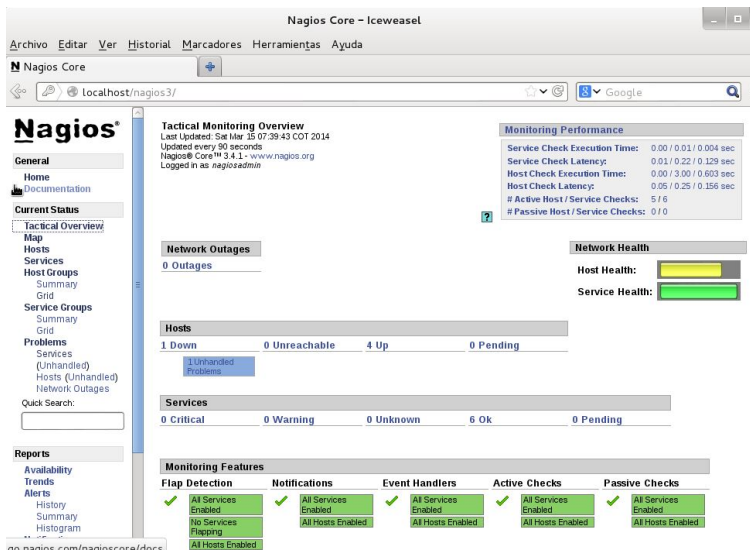


Figura 4.20: Posibles alertas críticas del sistema.  
Fuente: Autora

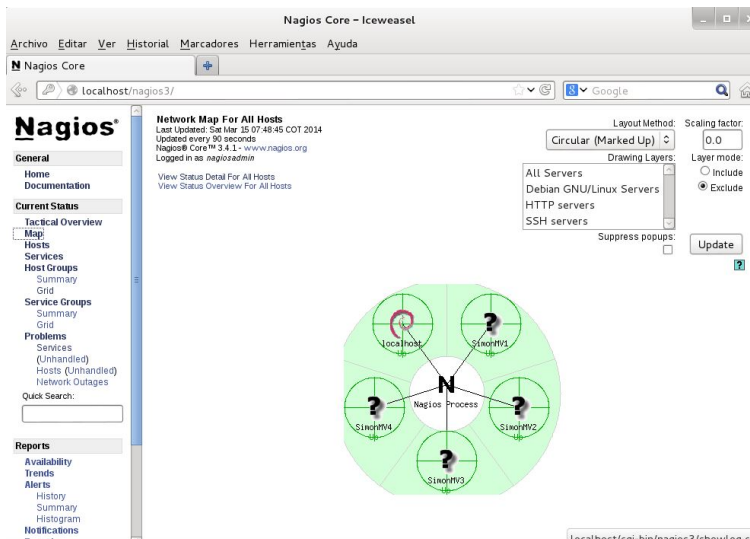


Figura 4.21: Estructura Monitorizada de nagios3.  
Fuente: Autora

Si existiera una alerta crítica en algún servicio, este se mostraría en rojo en la tabla ver figura *figura 4.23*.

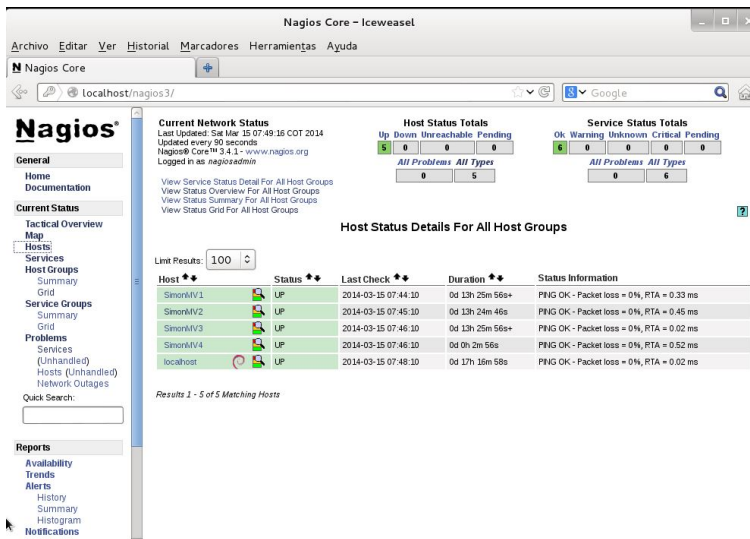


Figura 4.22: Equipos y servicios monitorizados y sus estados  
Fuente: Autora

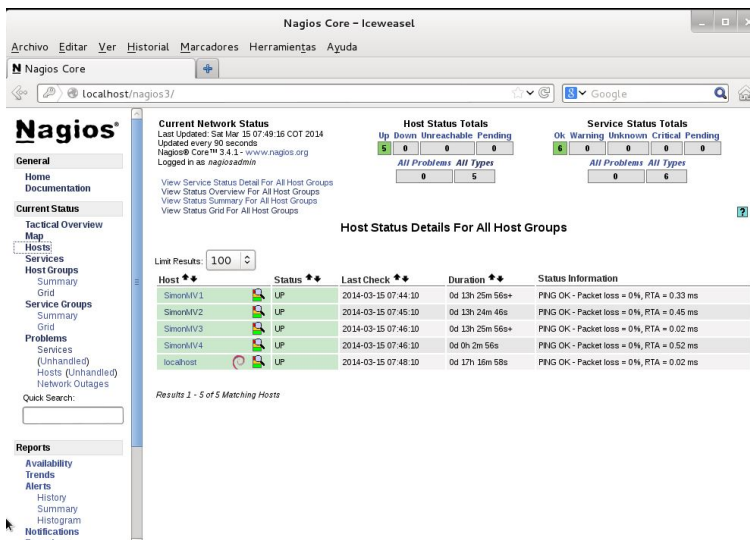


Figura 4.23: Estado de los servicios.  
Fuente: Autora

Como se puede observar *nagios3* es una herramienta fácil de usar y los parámetros de monitorización cambian dependiendo del sistema operativo, en este caso todas las máquinas están sobre *Debian 7*, por lo tanto el proceso es el mismo. Para monitorizar equipos máquinas virtuales Linux, se necesita la Ip de cada máquina, se edita el archivo de configuración de *nagios* y se añade la máquina a monitorizar, enseguida se reinicia el sistema, se accede al administrador web de *nagios*.

## 4.5. Fase 4: Mejora continua del Servicio

Esta fase provee una forma para que la organización pueda identificar y administrar adecuadamente las mejoras, de los servicios y mejoras en todas las fases del ciclo de vida del servicio. La Mejora continua del servicio TI tiene como objetivos, recomendar mejoras para todos los procesos y actividades involucradas, monitorear y analizar parámetros funcionales, estudiar mejoras para los servicios en en cuanto al soporte de las fases de estrategia y diseño.

Para llevar a cabo esta fase se recomienda estar haciendo revisiones periódicas de los recursos de la infraestructura por medio de las etapas de Gestión de eventos y Gestión de problemas descritas en la fase anterior. Dependiendo de las debilidades o niveles críticos encontrados, se debe hacer un estudio e implementar las mejoras y restaurar los servicios.

# Capítulo 5

## Conclusiones

En la realización de este proyecto, surgió un espacio de aprendizaje y de formación, que facilitó el desarrollo de los objetivos planteados y la construcción de conceptos sobre virtualización, Ciclo de vida ITIL y gestión de servicios de TI. Este ejercicio profesional permitió aportar al grupo una infraestructura de virtualización de servidores estable que ofrece los servicios requeridos por los miembros del Grupo Simon, garantizando la disponibilidad de los datos y las utilidades necesarias para hacer un uso eficiente de los recursos del servidor.

Gracias al tipo de virtualización completa se implementaron varias máquinas virtuales independientes, con su propio sistema operativo (Debian 7.1.0- Wheezy), cada máquina está a cargo de diversos recursos tecnológicos y aplicaciones; reduciendo costos de mantenimiento, energía y administración.

Esta experiencia permite ratificar que el aislamiento de las máquinas virtuales es una gran ventaja ya que ayuda a mejorar aspectos de seguridad y disponibilidad, si una máquina falla no afecta el rendimiento de las demás o del sistema en general.

Esta experiencia permitió verificar que el software libre facilita el proceso de virtualización con efectividad y eficiencia y además no genera costos de licenciamiento para su implementación.

Terminado el proyecto se puede afirmar que la adopción, para el desarrollo de esta propuesta del Ciclo de vida ITIL para la Gestión de servicios de TI, prácticas y orientaciones que fueron usadas e interpretadas y aplicadas de forma que se ajustaran a las necesidades y requerimientos de los usuarios del servidor del Grupo Simon, fue muy apropiadas y facilitó el cumplimiento de los objetivos del proyecto.

KVM resultó ser un componente indispensable y fundamental para el desarrollo de la virtualización completa, se eligió por la facilidad que proporciona estar integrado como módulo del kernel de Linux y la capacidad de ejecutar sistemas operativos de diferente naturaleza. KVM permitirá ejecutar, crear, monitorizar máquinas virtuales invitadas independientes y es el encargado de proporcionar los recursos hardware de los que hacen uso las máquinas virtuales. Al estar integrado como módulo del Kernel, no es necesario instalar ningún software específico para su uso.

# Capítulo 6

## Recomendaciones

Se recomienda cumplir con las políticas de administración y seguridad definidas por este proyecto; el cumplimiento de estas permitirán llevar un mejor control del uso de los recursos del servidor y mayor seguridad en las aplicaciones y datos de los usuarios.

Para una mejor disposición de los datos se recomienda el traslado del servidor al Centro de tecnologías de la información y la comunicación (CENTIC) para garantizar mejores condiciones físicas y eléctricas para el correcto funcionamiento del servidor.

Por la manera en que esta implementada la infraestructura de virtualización de servidores es necesario que exista un administrador al interior del Grupo, quien sea el responsable de gestionar y ofrecer los servicios del servidor.

Se recomienda mantener aseguradas y actualizadas las herramientas de la infraestructura, las aplicaciones y el kernel de linux ya que esto permite tener acceso a las nuevas funcionalidad del sistema y reparar cualquier error o vulnerabilidad que se haya encontrado.

Se recomienda monitorear y revisar regularmente los servicios establecidos para prevenir fallas y errores y asegurar el desempeño requerido del sistema, usando la herramienta nagios3.

Se debe tener presente que está a disposición de los miembros del grupo Simon, un catálogo de Servicios, donde se describen detalladamente los recursos que ofrece el Grupo a nivel de servidor de aplicaciones, este documento sirve de guía a la hora de elegir las tecnologías que más se adapten a las necesidades de los usuarios.

# Referencias

- [1] bochs: think inside the bochs. (En línea), citado en septiembre de 2013.  
<http://bochs.sourceforge.net/>
- [2] Consolidación de Servidores. What is Server Consolidation? (En línea), citado en agosto de 2013.  
[http://www.vmware.com/files/pdf/WhitePaper\\_ReducePowerConsumption.pdf](http://www.vmware.com/files/pdf/WhitePaper_ReducePowerConsumption.pdf)
- [3] CURRAN, CHRISTOPHER, *Fedora13 Manual de virtualización*, La guía definitiva de virtualización en Fedora, 0.Ed, 2010. (En línea), citado en diciembre de 2013.  
[http://docs.fedoraproject.org/es-ES/Fedora/13/pdf/Virtualization\\_Guide/Fedora-13-Virtualization\\_Guide-es-ES.pdf](http://docs.fedoraproject.org/es-ES/Fedora/13/pdf/Virtualization_Guide/Fedora-13-Virtualization_Guide-es-ES.pdf)
- [4] Debian Wheezy. Wikipedia, *Artículo principal: Debian*. (En línea), citado en agosto de 2013.  
[http://es.wikipedia.org/wiki/Debian\\_Wheezy](http://es.wikipedia.org/wiki/Debian_Wheezy)
- [5] Fuente Digital: Qué es Virtualización?. (En línea), citado en octubre de 2013.  
<http://uiaberto.wordpress.com/2010/09/26/%C2%BFque-es-virtualizacion/>
- [6] Herramientas de virtualización libres para sistemas GNU/Linux. (En línea), citado en septiembre de 2013.  
[http://www.uv.es/sto/charlas/2010\\_CIM/hvl-cim-2010-slides.pdf](http://www.uv.es/sto/charlas/2010_CIM/hvl-cim-2010-slides.pdf)
- [7] HERTZOG, RAPHAEL ; MAS,ROLAND, THE DEBIAN ADMINISTRATOR'S HANDBOOK, Copyright 2012 Freexian SARL. (En línea), citado en noviembre de 2013.  
<http://debian-handbook.info/download/stable/debian-handbook.pdf>
- [8] ITIL para las pymes: Documento de posición sobre buenas prácticas. itSMF IT Service Management, Versión 2. (En línea), citado en agosto de 2013.  
<http://documents.bmc.com/products/documents/32/94/63294/63294.pdf>
- [9] *ITIL V3: Gestión de Servicios TI*. Activos del servicio. (En línea), citado en agosto de 2013.  
[http://itilv3.osiatis.es/estrategia\\_servicios\\_TI/<introduccion\\_objetivos\\_activos\\_servicio.php](http://itilv3.osiatis.es/estrategia_servicios_TI/<introduccion_objetivos_activos_servicio.php)
- [10] *ITIL V3: Gestión de Servicios TI*. Creación de valor. (En línea), citado en agosto de 2013.

[http://itilv3.osiatis.es/estrategia\\_servicios\\_TI/introduccion\\_objetivos\\_creacion\\_valor.php](http://itilv3.osiatis.es/estrategia_servicios_TI/introduccion_objetivos_creacion_valor.php)

[11] *ITIL V3: Gestión de Servicios TI*. Diseño de los Servicios TI. (En línea), citado en septiembre de 2013.

[http://itilv3.osiatis.es/disenos\\_servicios\\_TI.php](http://itilv3.osiatis.es/disenos_servicios_TI.php)

[12] *ITIL V3: Gestión de Servicios TI*. Gestión de la Capacidad. (En línea), citado en septiembre de 2013.

[http://itilv3.osiatis.es/disenos\\_servicios\\_TI/gestion\\_capacidad/introduccion\\_objetivos.php](http://itilv3.osiatis.es/disenos_servicios_TI/gestion_capacidad/introduccion_objetivos.php)

[13] *ITIL V3: Gestión de Servicios TI*. Gestión de la Seguridad de la Información. (En línea), citado en septiembre de 2013.

[http://itilv3.osiatis.es/disenos\\_servicios\\_TI/gestion\\_seguridad\\_informacion.php](http://itilv3.osiatis.es/disenos_servicios_TI/gestion_seguridad_informacion.php)

[14] *ITIL V3: Gestión de Servicios TI*. Gestión de Servicios TI. (En línea), citado en agosto de 2013.

[http://itilv3.osiatis.es/gestion\\_servicios\\_ti.php](http://itilv3.osiatis.es/gestion_servicios_ti.php)

[15] *ITIL V3: Gestión de Servicios TI*. Operación del Servicio. (En línea), citado en diciembre de 2013.

[http://itilv3.osiatis.es/operacion\\_servicios\\_TI.php](http://itilv3.osiatis.es/operacion_servicios_TI.php)

[16] *ITIL V3: Gestión de Servicios TI*. Procesos. (En línea), citado en diciembre de 2013.

[http://itilv3.osiatis.es/operacion\\_servicios\\_TI/procesos.php](http://itilv3.osiatis.es/operacion_servicios_TI/procesos.php)

[17] *ITIL V3: Gestión de Servicios TI*. Qué es ITIL V3? (En línea), citado en agosto de 2013.

<http://itilv3.osiatis.es/itil.php>

[18] *ITIL V3: Gestión de Servicios TI*. Transición de los Servicios TI. (En línea), citado en noviembre de 2013.

[http://itilv3.osiatis.es/transicion\\_servicios\\_TI.php](http://itilv3.osiatis.es/transicion_servicios_TI.php)

[19] KVM. *Definición de KVM*. (En línea), citado en noviembre de 2013.

[http://es.wikipedia.org/wiki/Kernel-based\\_Virtual\\_Machine](http://es.wikipedia.org/wiki/Kernel-based_Virtual_Machine)

[20] KVM. Geeks y Linux Atelier. KVM o Xen: Escogiendo una plataforma de virtualización. (En línea), citado en noviembre de 2013.

<http://glatelier.wordpress.com/2010/07/21/kvm-o-xen-escogiendo-una-plataforma-de->

*virtualizacion-parte-1-2/*

[21] KVM: Kernel Based Virtual Machine. (En línea), citado en septiembre de 2013.  
*[http://www.linux-kvm.org/page/Main\\_Page](http://www.linux-kvm.org/page/Main_Page)*

[22] Linux para todos, Kernel Based Virtual Machine (KVM). (En línea), citado en noviembre de 2013.  
*[http://www.linuxparatodos.net/web/comunidad/base-de-conocimiento/-/wiki/Base+de+Conocimiento/Kernel+Based+Virtual+Machine+\(KVM\)](http://www.linuxparatodos.net/web/comunidad/base-de-conocimiento/-/wiki/Base+de+Conocimiento/Kernel+Based+Virtual+Machine+(KVM))*

[23] MAS,ROLAND,*Debian 7.0 "Wheezy" publicado.* ((En línea), citado en noviembre de 2013.  
*<http://www.debian.org/News/2013/20130504>*

[24] PERALTA,XAVIER; VERDEJO,GABRIEL, Virtualización de servidores del RDLab, Ingeniería técnica en Informática de sistemas, Junio 2012. (En línea), citado en agosto de 2013.  
*<http://rdlab.lsi.upc.edu/docu/PFC-XavierPeralta2012.pdf>*

[25] qemu: modos de funcionamiento. (En línea), citado en septiembre de 2013.  
*<http://elpuig.xeill.net/Members/vcarceler/articulos/qemu>*

[26] RAMOS,ANTONIO; GARCÍA-MORÁN,JEAN; PICOUTO,FERNANDO; GRIJALBA,JACINTO; MAYAN,MAIKEL; GARCÍA,ÁNGEL; INZA,EDUARDO; BARBERO,CARLOS. Instala, administra, securiza y virtualiza entornos Linux. 1a.Ed. Alfaomega Grupo Editor, S.A. de C.V. México DF, Enero 2009.

[27] VILLAR,EUGENIO; GÓMEZ,JULIO, Virtualización de plataforma. (En línea), citado en agosto de 2013.  
*[http://www.adminso.es/images/a/a2/Eugenio\\_cap2.pdf](http://www.adminso.es/images/a/a2/Eugenio_cap2.pdf)*

[28] VirtualBox: Welcome to VirtualBox.org!. (En línea), citado en septiembre de 2013.  
*<https://www.virtualbox.org/>*

[29] Virtual Machine Manager, Manager virtual machines with virt-manager. (En línea), citado en diciembre de 2013.  
*<http://virt-manager.org/>*

[30] Virtualización con KVM en Linux: cómo compilar e instalar QEMU+libvirt+virt-manager desde los fuentes. (En línea), citado en noviembre de 2013.  
*<http://www.linuxito.com/gnu-linux/nivel-alto/316-virtualizacion-con-kvm-en-linux-como-compilar-e-instalar-qemu-libvirt-virt-manager-desde-los-fuentes>*

[31] XEN: Qué es?. (En línea), citado en septiembre de 2013  
*[http://wiki.xen.org/wiki/Xen\\_Overview/es](http://wiki.xen.org/wiki/Xen_Overview/es)*

# Anexos

# Anexo A

## Herramientas de administración de las Máquinas Virtuales

La administración de Máquinas Virtuales, consiste en instalar, eliminar, ver el estado y las características, modificar recursos, configurar la red de cada una de las Máquinas virtuales dentro de KVM. Para una administración eficiente ocuparemos dos herramientas bajo la librería *libvirt*, la herramienta *virsh* y *virt-manager*.

### Administración vía consola de las máquinas virtuales (*virsh*<sup>1</sup>)

*virsh* es una herramienta de gestión de máquinas virtuales mediante línea de comandos, funciona de modo de solo lectura para el usuario con permisos al grupo *libvirt* y el usuario administrador es el único que puede modificar los datos.

Conexión a *virsh*: El siguiente comando permite conectarnos a *virsh*, y dentro de *virsh* podemos administrar las máquinas virtuales.

```
administrador@simon: $ virsh -c qemu:///system
```

Instalación de las Máquinas Virtuales: Por otra parte si deseamos hacer la instalación desde consola de una máquina virtual se tendrá que ejecutar el comando *virt-install*, el tipo de conexión y el parámetro así:

```
virt-install [conexión][opciones]
```

Nota:El tipo de conexión siempre va ser:

```
- -connect qemu:/system
```

los son los parámetros: siguientes

```
-n Nombre la máquina virtual.
```

---

<sup>1</sup>Para el desarrollo de esta guía se toma como referencia la fuente digital *Linux para todos: Kernel Based Virtual Machine (KVM)* tomada de la página web: [http://www.linuxparatodos.net/web/comunidad/base-de-conocimiento/-/wiki/Base+de+Conocimiento/Kernel+Based+Virtual+Machine+\(KVM\)](http://www.linuxparatodos.net/web/comunidad/base-de-conocimiento/-/wiki/Base+de+Conocimiento/Kernel+Based+Virtual+Machine+(KVM))

<code>-r</code>	Memoria RAM virtual.
<code>-f</code>	Ruta del Disco Duro Virtual.
<code>-s</code>	Tamaño del disco duro.
<code>-c</code>	Unidad DVD/CDRom o imagen ISO.
<code>-l</code>	Instalación vía red por medio de http, ftp o nfs.
<code>-vnc</code>	Conexión tipo VNC.
<code>-vcpus</code>	Número de CPU virtuales.
<code>-network=br0</code>	Modo de conexión a la red física en puente.
<code>-w network=default</code>	Modo de conexión a la red virtual en NAT.

Visualización de una MV: Después de tener instalada la máquina virtual la herramienta *virt-viewer* proporciona la visualización en modo gráfico, editando el siguiente comando:

```
administrador@simon: $ virt-viewer -c qemu:///system simonmv1
```

Conexión al hipervisor: Si se desea conectar al hipervisor para poder ejecutar comandos de administración en consola, se escribe el siguiente comando:

```
administrador@simon: $ virsh -c qemu:///system
```

Desde luego nos queda algo parecido a:

```
virsh #
```

Ya desde *virsh* podemos ejecutar los siguientes comandos de administración de las máquinas virtuales.

Listar MV encendidas:

```
virsh # list
```

Encender automáticamente la MV (simonmv1)

```
virsh # autostart simonmv1
```

Desactivar el encendido automático de la MV (simonmv1)

```
virsh # autostart - --disable simonmv1
```

Encender manualmente la MV: comando usado cuando la máquina virtual no inicia en automático.

```
virsh # start simonmv1
```

Conocer la información de la MV: Se obtiene la información de Ver cuantos CPU virtuales tiene la máquina virtual. ( nombre, CPU, Memoria, estado,etc.)

```
virsh # dominfo simonmv1
```

Saber el estado de las MV:(encendido, apagado, pausado, etc.)

```
virsh # domstate simonmv1
```

Ver el nombre la máquina física

```
virsh # hostname
```

Editar Información de la MV:

```
virsh # edit simonmv1
```

Suspender MV:

```
virsh # suspend simonmv1
```

Reaunudar MV suspendida:

```
virsh # shutdown simonmv1
```

Apagar MV:

```
virsh # reboot simonmv1
```

Matar máquina virtual:

```
virsh # destroy simonmv1
```

Mostrar información relacionada de la máquina física: (CPU, memoria, etc.)

```
virsh # nodeinfo
```

Capacidad de la Máquina física: Muestra la capacidad de las plataforma en KVM.

```
virsh # capabilities
```

Ver cuantos CPU virtuales tiene la máquina virtual

```
virsh # vcpuinfo simonmv1
```

Consola de Máquina virtual:

```
virsh # ttyconsole simonmv1
```

Redes virtuales configuradas:

```
virsh # net-list
```

Información de redes virtuales:

```
virsh # net-dumpxml default
```

Ayuda en linea:

```
virsh # help vol-list
```

Salir del administrador:

```
virsh # quit
```

### **Administración gráfica de las MV (*virt-manager*)**

*virt-manager*, es la herramienta en modo gráfico que permite la administración de las Máquinas Virtuales. Esta herramienta permite realizar las siguientes características<sup>2</sup>:

- Instalación de las máquinas virtuales.
- Administración de las máquinas.
- Recursos de las máquinas virtuales
- Uso de CPU, Memoria de la máquina física.
- Permite iniciar, pausar, reiniciar, apagar y forzar el apagado sencillamente.

---

<sup>2</sup>Tomada de la fuente digital *Linux para todos: Kernel Based Virtual Machine (KVM)* tomada de la página web:  
[http://www.linuxparatodos.net/web/comunidad/base-de-conocimiento/-/wiki/Base+de+Conocimiento/Kernel+Based+Virtual+Machine+\(KVM\)](http://www.linuxparatodos.net/web/comunidad/base-de-conocimiento/-/wiki/Base+de+Conocimiento/Kernel+Based+Virtual+Machine+(KVM))

- Permite ver el estado actual de las máquinas virtuales. Número de proceso, estado, uso de CPU, procesadores virtuales y cantidad de memoria ocupada.
- Borrado de las máquinas.
- Configuración del almacenaje de los discos duros de las máquinas virtuales.
- Configuración de red interna por medio de DHCP.
- Tiempo de refresco de la información.

Para conocer más acerca de la instalación y configuración de *virt-manager* en el documento [ANEXO E] se describen los pasos y en el documento [ANEXO F] se describe el despliegue de las máquinas virtuales con esta herramienta.

# Anexo B

## Instalación del sistema operativo base en el host

Los pasos para la intalación del sistema operativo en el sistema anfitrión son los siguientes:

1. Descargar la imagen de *Debian Wheezy* AMD 64 bits en la página oficial de Debian, <http://www.debian.org/>, en el enlace *imágenes ISO de CD, AMD64* y se elige *debian-7.1.0-amd64-netinst.iso*
2. Grabar la imagen en un CD en blanco.
3. Arrancar el servidor con el CD de la imagen del Sistema Operativo.
4. Aparecerá una pantalla como la que se muestra en la *Figura B.1*, el programa *debian installer* se encarga de esta tarea. La instalación del sistema base es en modo texto, ya que en cuanto a consumo de recursos resulta más eficiente trabajar en este entorno que uno gráfico, es más sencillo escribir una línea de comandos, que manejar una completa herramienta gráfica, la consola de texto siempre va a estar funcionando.
5. Realizar la configuración básica del servidor, el programa de instalación inicia en ingles este paso consiste en seleccionar el lenguaje que se usará para el proceso de instalación. *Español*
6. Seleccionar la Localización. *Colombia*
7. Elegir el mapa de teclado. *Ingles Estados Unidos*
8. Detectar el hardware, este paso es automático el instalador detecta el hardware e intenta identificar el dispositivo CD-ROM a utilizar para acceder a su contenido, luego carga los módulos correspondientes a los componentes del hardware y monta el CD, luego lo lee y el instalador descarga todos los archivos necesarios para continuar con su trabajo.
9. Introducir el nombre que identificará el sistema en la red. *Grupo SIMON*.



Figura B.1: Pantalla de arranque Debian Wheezy  
Fuente: Autora

10. Escribir el nombre de dominio del sistema, como se muestra en la *Figura B.2 uis.edu.co*.
11. Definir una contraseña para el root o superusuario, esta cuenta es reservada para el administrador del equipo.
12. Elegir un nombre de usuario para la cuenta.
13. Definir una contraseña para el nuevo usuario.
14. Seleccionar la zona horaria.
15. A continuación se iniciará el particionado de discos. El particionado consiste en dividir el espacio disponible en los discos duros, cada subdivisión de los mismos es llamada una partición. En la *Figura B.3*, se muestra la guía para el particionado, elegimos la última opción *manual*, ya que vamos a trabajar junto a otras particiones preexistentes y esta opción provee mayor flexibilidad, permitiéndole al usuario seleccionar el propósito y tamaño de cada partición.
16. A continuación se muestra el resumen de las particiones y punto de montaje <sup>1</sup> que

---

<sup>1</sup>El punto de montaje es el árbol de directorios que albergará el contenido del sistema de archivos en la partición seleccionada.

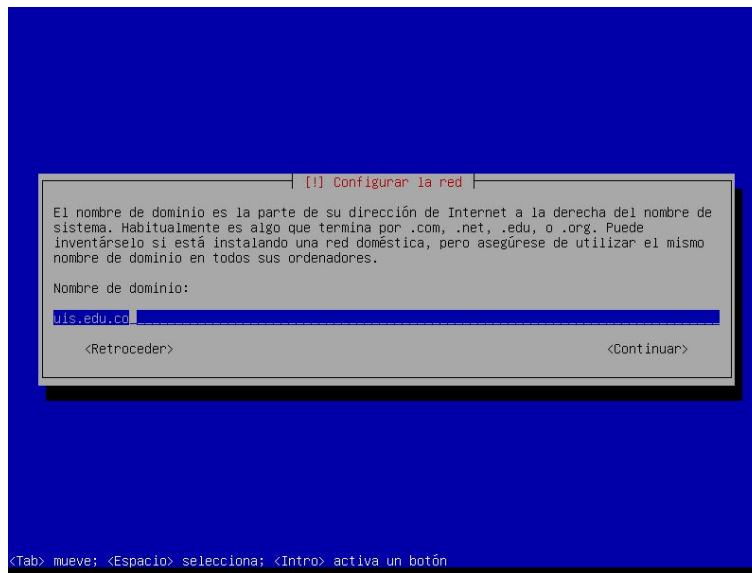


Figura B.2: Configuración de red  
Fuente: Autora

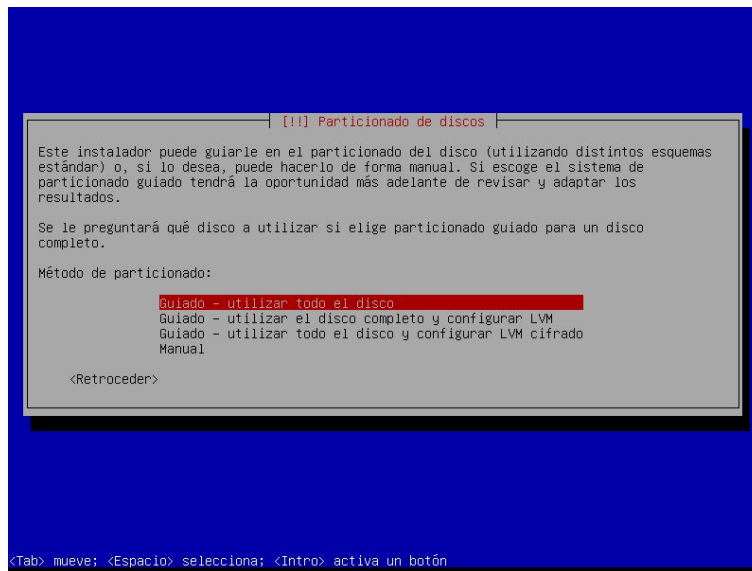


Figura B.3: Elección del método del particionado  
Fuente: Autora

se tiene configurado acualmente. Se configurará un RAID 1+0, construido a partir de cuatro particiones RAID, consiste en dos pares de particiones, cada una de las cuales refleja a la otra. Los datos son distribuidos sobre ambas particiones.

17. Seleccionar la primera partición para modificar sus valores.
18. Luego se crea una tabla de particiones vacía para el dispositivo.
19. Se selecciona *configurar RAID por Software*.
20. En la siguiente ventana se afirma que se *desea escribir los cambios en los dispositivos de almacenamiento y configurar RAID*.
21. Se abre el menú de configuración de RAID por software o MD y se elige la opción *crear dispositivo MD*.
22. A continuación se elige el tipo de dispositivo RAID por software a crear *RAID 10*.
23. Digitar el número de dispositivos activos para el array RAID10: *4*(*Los cuatro discos*), como es un raid anidado, necesitamos de los 4 discos duros.
24. Digitar el numero de dispositivos que queremos fuera del raid, número de dispositivos libres para array RAID10: *0* (*cuantos no va a utilizar*) en este caso como no tenemos mas discos duros.
25. A continuación se muestra una pantalla para seleccionar las particiones que queremos en el RAID, seleccionamos las 4 creadas anteriormente.
26. Luego se guardan los cambios realizados y configuración del RAID y se selecciona *Terminar*.
27. A continuación se muestra el resumen de las particiones y puntos de montaje creados y se comienza a editar las particiones. Se selecciona la primera partición.
28. Seleccionar *utilizar como* y se da click en sistema de *fichero ext3 transaccional*, Se selecciona como punto de montaje / - *Sistema de fichero raíz*<sup>2</sup>, luego se da click en terminar de definir la partición. Y continuar.
29. Se hace lo mismo con las particiones */home*<sup>3</sup> y */vm*<sup>4</sup>, y quedan listas las particiones con su punto de montaje definido.

A continuación se muestra el resumen de la tabla de particiones y sus puntos de montaje. ( Particionado de discos)

---

<sup>2</sup>la raíz de la partición que contendrá el sistema Debian

<sup>3</sup>Una partición montada en */home/* generalmente está destinada a contener la información de los usuarios.

<sup>4</sup>Partición destinada al almacenamiento de las máquinas virtuales

Dispositivo RAID10#0-100.GB Dispositivo RAID por software

#1	100.0 GB	f	ext4	/
	512.0 B		inútil	

Dispositivo RAID10#1-2.0 TB Dispositivo RAID por software

#1	2.0 TB	f	ext4	/home
	512.0 B	B	inútil	

Dispositivo RAID10#2-1.9 TB Dispositivo RAID por software

#1	1.9 TB	f	ext4	/vm
	512.0 B		inútil	

SCSI1 (0,0,0) (sda)-2.0 TB ATA ST2000DM001-9YN1

#1	100.0 MB	B	f	ext4	/boot
#2	50.0 GB		K	raid	
#3	1.0 TB		K	raid	
#4	950.3 GB		K	raid	
	8.2 KB			ESPACIO LIBRE	

SCSI2 (0,0,0) (sdb)-2.0 TB ATA ST2000DM001-9YN1

#1	100.0 MB				
#2	50.0 GB		K	raid	
#3	1.0 TB		K	raid	
#4	950.3 GB		K	raid	
	8.2 KB			ESPACIO LIBRE	

```

SCSI3 (0,0,0) (sdc)-2.0 TB ATA ST2000DM001-9YN1
#1 100.0 MB
#2 50.0 GB K raid
#3 1.0 TB K raid
#4 950.3 GB K raid
8.2 KB ESPACIO LIBRE
SCSI4 (0,0,0) (sdd)-2.0 TB ATA ST2000DM001-9YN1
#1 100.0 MB
#2 50.0 GB K raid
#3 1.0 TB K raid
#4 950.3 GB K raid
8.2KB ESPACIO LIBRE

```

Figura B.4: Esquema – Particionado de discos  
Fuente: Autora

30. A continuación se instalarán los paquetes del sistema base , las herramientas que administran los paquetes Debian y también los programas necesarios para iniciar el sistema.

31. En seguida se configura el gestor de paquetes *apt*, (paso automático).

32. Elegir una réplica de red, es útil para obtener paquetes de la red, primero se selecciona un país y luego una réplica disponible en dicho país.

33. A continuación se Elije las utilidades estándar necesarias, en la *Figura B.5*, podemos ver diez tareas de las cuales se seleccionan: *Servidor SSH* y *Utilidades estándar del Sistema*.

34. Cargar automáticamente el *gestor de arranque GRUB*. El gestor de arranque es el primer programa iniciado por el BIOS. Este programa carga el núcleo Linux a la memoria y luego lo ejecuta.

35. Luego se finaliza la instalación y el sistema es reiniciado.

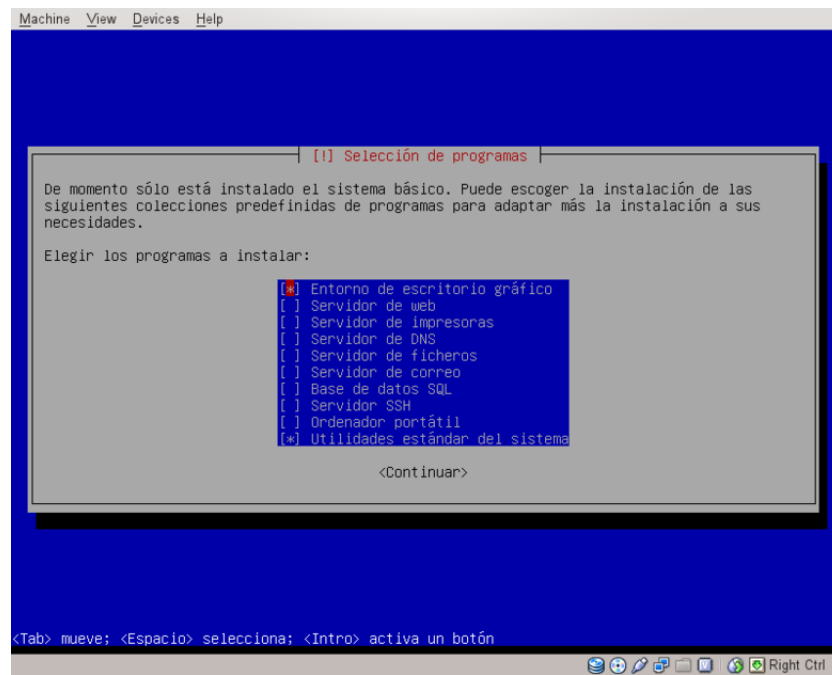


Figura B.5: Utilidades estándar necesarias  
Fuente: Autora

# Anexo C

## Instalación y configuración de *KVM* en el servidor

A continuación se nombran los pasos para la instalación<sup>1</sup> KVM.

1. Iniciar sesión como *root* en el sistema Debian 7 recién instalado, con la contraseña que ha asignado durante el proceso de instalación.

2. Asegurar que el sistema es compatible con KVM, para ello edite el siguiente comando:

```
egrep '(vmx|svm)' -color=always /proc/cpuinfo
```

como resultado se deben ver unas líneas de salida como las que muestra la *Figura C.1*, es decir que tenemos el soporte para realizar la virtualización KVM, por el contrario si no se observa ningún mensaje en absoluto, la CPU no soporta la virtualización de hardware.

3. Actualizar los repositorios y paquetes del sistema. La lista de nuevos paquetes se descargan desde los repositorios del *sources.list*. con el siguiente comando:

```
apt-get update
```

4. Instalar las actualizaciones del sistema pendientes con el comando:

```
apt-get dist-upgrade
```

5. Instalar el modulo para el kernel en el cual permitirá tener máquinas virtuales huéspedes. El sistema pide insertar el disco etiquetado.

```
apt-get install qemu-kvm
```

6. Instalar la librería encargada de virtualizar las máquinas huésped.

```
apt-get install libvirt-bin
```

7. Instalar el paquete *bridge-utils*, para configurar la interfaz de red sobre la que circula el tráfico desde internet a la red interna, es decir permitirá crear un puente entre las tarjetas de red física y las virtuales.

---

<sup>1</sup>Para la respectiva instalación de KVM se siguieron los pasos de la guía *Palegray.net. Install and Configure KVM and on the Server del hipervisor* tomada de la página: <http://www.palegray.net/tutorials/debian-6-kvm-howto/configure-kvm>

```

Terminal (como superusuario)
Archivo Editar Ver Buscar Terminal Ayuda
root@simon:/home/simon# egrep '(vmx|svm)' --color=always /proc/cpuinfo
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe nx rdtscp lm constant_
tsc arch_perfmon pebs bts xtopology nonstop_tsc aperfmperf pni pclmulqdq dtes64
monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic po
pcnt tsc_deadline_timer aes xsave avx lahf_lm ida arat xsaveopt pln pts dtherm t
pr_shadow vnmi flexpriority ept vpid
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe nx rdtscp lm constant_
tsc arch_perfmon pebs bts xtopology nonstop_tsc aperfmperf pni pclmulqdq dtes64
monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic po
pcnt tsc_deadline_timer aes xsave avx lahf_lm ida arat xsaveopt pln pts dtherm t
pr_shadow vnmi flexpriority ept vpid
flags           : fpu vme de pse tsc msr pae mce cx8 apic sep mtrr pge mca cmov
pat pse36 clflush dts acpi mmx fxsr sse sse2 ss ht tm pbe nx rdtscp lm constant_
tsc arch_perfmon pebs bts xtopology nonstop_tsc aperfmperf pni pclmulqdq dtes64
monitor ds_cpl vmx smx est tm2 ssse3 cx16 xtpr pdcm pcid sse4_1 sse4_2 x2apic po
pcnt tsc_deadline_timer aes xsave avx lahf_lm ida arat xsaveopt pln pts dtherm t
pr_shadow vnmi flexpriority ept vpid
root@simon:/home/simon#

```

Figura C.1: Nivel de soporte de CPU para Virtualización KVM  
Fuente: Autora

```
apt-get install bridge-utils
```

8. Instalar el paquete *screen*, herramienta útil de administración en modo texto, ya que permite tener varias consolas abiertas en una terminal.

```
apt-get install screen
```

9. Configurar el teclado con el siguiente comando:

```
apt-get install console-data
```

10. El siguiente paquete permite configurar y manipular la consola de Linux (es decir, pantalla y teclado).

```
apt-get install console-tools
```

11. Instalar el sistema de configuración de paquetes de Debian, con el siguiente comando:

```
apt-get install debconf
```

12. Agregar la cuenta de usuario *simon* al grupo *libvirt*<sup>2</sup>, ejecutar el siguiente comando:

<sup>2</sup>Libvirt es una API común que ayuda a gestionar las plataformas de virtualización. A través de esta sola API, se tendrá una comunicación con una gran cantidad de hipervisores como VirtualBox, KVM, Xen, LXC, OpenVZ, hipervisores basados en VMware y Microsoft Hyper-V. Libvirt también proporciona para la gestión de la red y el disco, la autenticación y control de acceso, entre otras. <http://thejaswi.info/tech/blog/2012/06/25/libvirt-and-kvm/>

```
adduser simon libvirt
```

13. Ajustar el parámetro *swappiness*<sup>3</sup> del sistema.

```
sysctl vm.swappiness=0  
echo "vm.swappiness=0">>/etc/sysctl.conf
```

14. El siguiente paso es asignar estáticamente la dirección IP del servidor y crear un puente de red para máquinas virtuales, se debe editar el archivo */etc/network/interfaces* como se muestra en la *Figura C.2*.

```
# The loopback network interface  
auto lo  
iface lo inet loopback  
  
# The primary network interface  
auto eth0  
iface eth0 inet manual  
  
# Network bridge  
auto br0  
iface br0 inet static  
address 192.168.65.239  
network 192.168.65.0  
netmask 255.255.255.0  
broadcast 192.168.65.255  
gateway 192.168.65.1  
bridge_ports eth0  
bridge_stp off  
bridge_fd 0  
bridge_maxwait 0
```

Figura C.2: Configuración de red servidor Grupo SIMON  
Fuente: Autora

15. Una vez que haya realizado los cambios necesarios, se debe reiniciar la creación de redes en el sistema con el siguiente comando:

```
/etc/init.d/networking restart
```

16. Cambiar los valores del archivo */etc/hosts* como se muestra en la *Figura C.3*. 17. Se debe reiniciar el sistema con el siguiente comando:

---

<sup>3</sup>Es una propiedad del Núcleo Linux que permite establecer un balance entre el uso del espacio de intercambio (swap), y la Memoria de acceso aleatorio (RAM). El *swappiness* puede tomar valores desde el 0 hasta el 100. Si se establece 0 el núcleo intentará no hacer intercambio, mientras que si se establece 100 el sistema intentará mantener la Memoria de acceso aleatorio lo más libre posible haciendo intercambio. <http://en.wikipedia.org/wiki/Swappiness>

```
127.0.0.1      localhost
192.168.65.239  simon.uis.edu.co      simon

# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
ff02::1 ip6-allnodes
Ff02::2 ip6-allrouters
```

Figura C.3: Configuración archivo */etc/hosts*

Fuente: Autora

`reboot`

18. Una vez que el sistema vuelve a iniciar, se ingresa a la sesión como usuario *root* y ejecute el siguiente comando para crear las claves SSH del host.

`ssh-keygen -t rsa`

19. Descargar una imagen ISO de *Debian-7.1.0*<sup>4</sup> para el servidor KVM. Será utilizado más tarde para crear su las máquinas invitadas. Emita el siguiente comando dentro del fichero */var/lib/libvirt/images* para descargar la imagen ISO de Debian 7.

```
wget http://cdimage.debian.org/debian-cd/7.1.0/amd64/iso-cd/debian-7.1.0-  
amd64-netinst.iso
```

---

<sup>4</sup>*debian-7.1.0-amd64-i386-netinst*, descargado de la página <http://www.debian.org/>, enlace *Imágenes ISO de CD*, en la pestaña *Descargar mediante HTTP/FTP*, y luego en el enlace *amd64*

# Anexo D

## Instalación de *VirtualBox* y el sistema de escritorio

Para la administración en modo gráfico de las máquinas virtuales alojadas en el servidor HP Proliant ML 110, se instalará el hipervisor *VirtualBox*<sup>1</sup> y sobre el se configurará un sistema de escritorio dedicado que tendrá instalado un sistema operativo *Debian 7 - Wheezy*, a continuación se enumeran los pasos a seguir<sup>2</sup>.

1. Descargar el paquete de instalación del *VirtualBox* que coincida con el sistema operativo (*Windows XP*) de la estación de trabajo. En nuestro caso se instala *VirtualBox 4.2.12* de la página oficial <http://www.virtualbox.org/wiki/Downloads>.
2. Realizar la instalación respectiva del *VirtualBox*.
3. Instalar un sistema de escritorio *Debian 7.1.0 - Wheezy en VirtualBox*. Sobre este Sistema de escritorio se administrarán las Máquinas Virtuales alojadas en el Servidor. En la *Figura D.1*, se muestra la pantalla de bienvenida de *VirtualBox*, dar click en el icono *New* para continuar.
4. Escribir un nombre descriptivo para la estación de trabajo *Debian 7*, seleccionar *Linux* para el sistema operativo, y *Debian* para la versión, como se muestra en la *Figura D.2*.
5. Asignar una cantidad razonable de memoria para la máquina virtual. Se asigna una potencia de *512 MB*.
6. Crear un nuevo disco duro de la máquina virtual, especificando el almacenamiento de tamaño fijo y la asignación de aproximadamente *8 GB* de espacio para ello *Figura D.3*.

---

<sup>1</sup>Es un producto de virtualización de Oracle que permite a los usuarios ejecutar fácilmente múltiples sistemas operativos simultáneamente y está disponible de forma gratuita

<sup>2</sup>Para la respectiva instalación y configuración de *VirtualBox* se siguieron los pasos de la guía *Palegray.net. Install VirtualBox on the Management Workstation* tomada de la página: <http://www.palegray.net/tutorials/debian-6-kvm-howto/install-virtualbox>

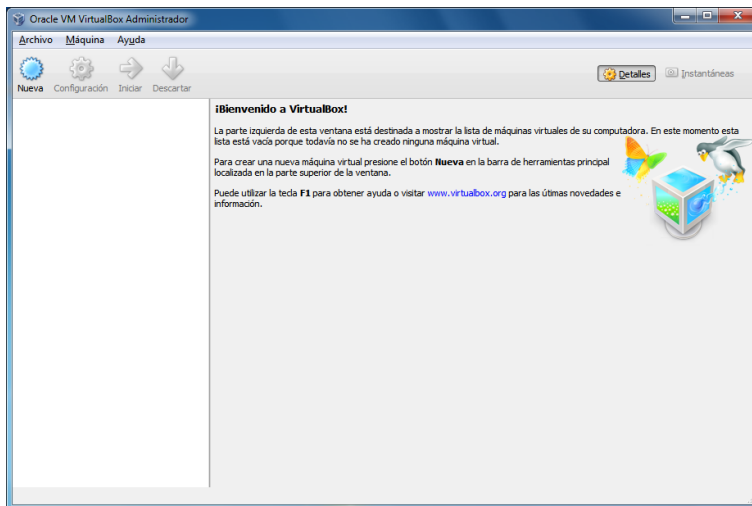


Figura D.1: Pantalla de bienvenida de *VirtualBox*  
Fuente: Autora

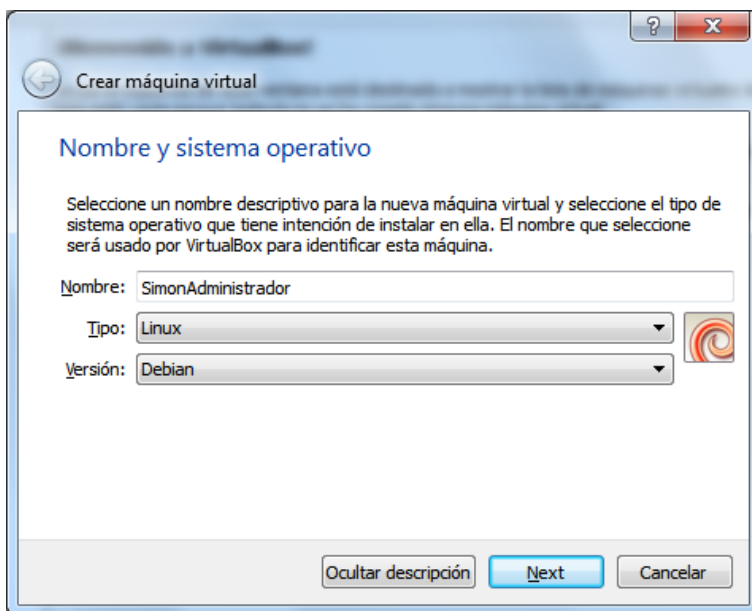


Figura D.2: Nombre de la Estación de trabajo y Sistema Operativo  
Fuente: Autora

7. Una vez que la máquina virtual se ha creado, se le presentará con una información general que describe su configuración inicial, como se muestra en la *Figura D.4*. Haga click en el icono *Configuración* para personalizar la estación de trabajo.

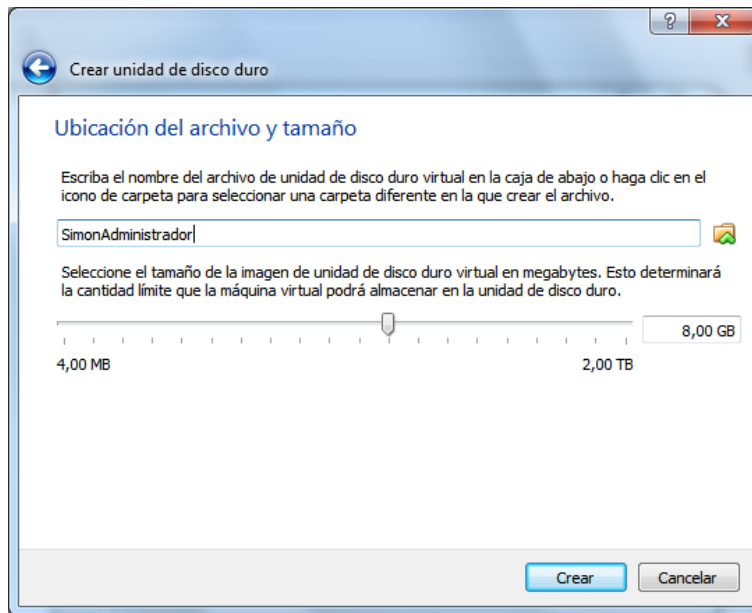


Figura D.3: Ubicación del archivo y tamaño  
Fuente: Autora

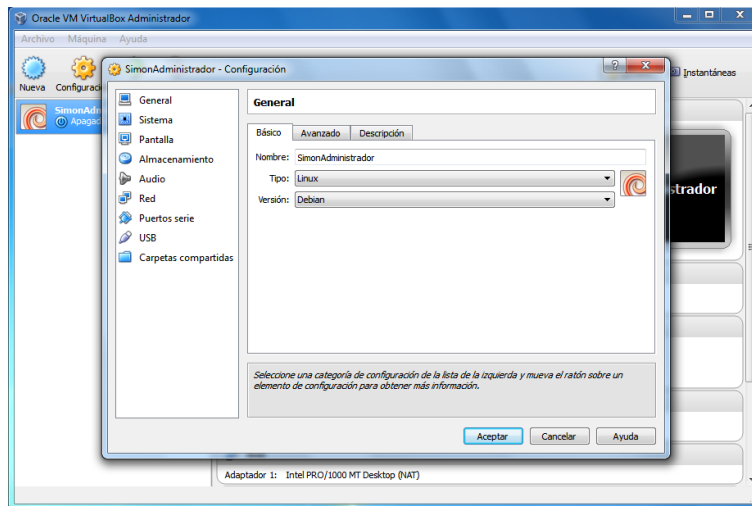


Figura D.4: Personalizar Estación de trabajo  
Fuente: Autora

8. En la pestaña *Almacenamiento*, se debe adjuntar la imagen previamente descargada de *Debian 7 ISO* en el dispositivo de la unidad de CD / DVD, como se muestra en la *Figura D.5*

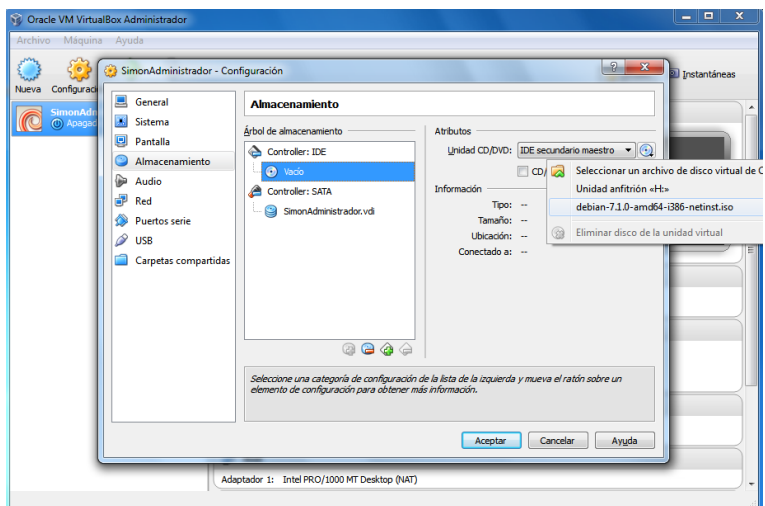


Figura D.5: Adjuntar la imagen de *Debian 7 ISO*  
Fuente: Autora

9. Una vez finalizado todos los ajustes, se debe arrancar Máquina Virtual, en el icono *Iniciar* (flecha). A continuación se verá la pantalla de bienvenida de instalación de Debian 7, como se muestra en la siguiente *Figura D.6*



Figura D.6: Pantalla de Bienvenida de Instalación de *Debian 7*  
Fuente: Autora

10. El tipo de instalación que se realizará en la estación de trabajo es de tipo gráfico. Se inicia eligiendo el lenguaje, la ubicación, se configura el mapa de teclado, la red, los usuarios y contraseñas. El método de particionado en este caso es *Guiado - utilizar todo el disco* como se muestra en la siguiente *Figura D.7*. Luego se eli-

ge el disco a particionar y el esquema de particionado: *Todos los ficheros en una partición*. A continuación se finaliza el particionado y se escriben los cambios en el disco.

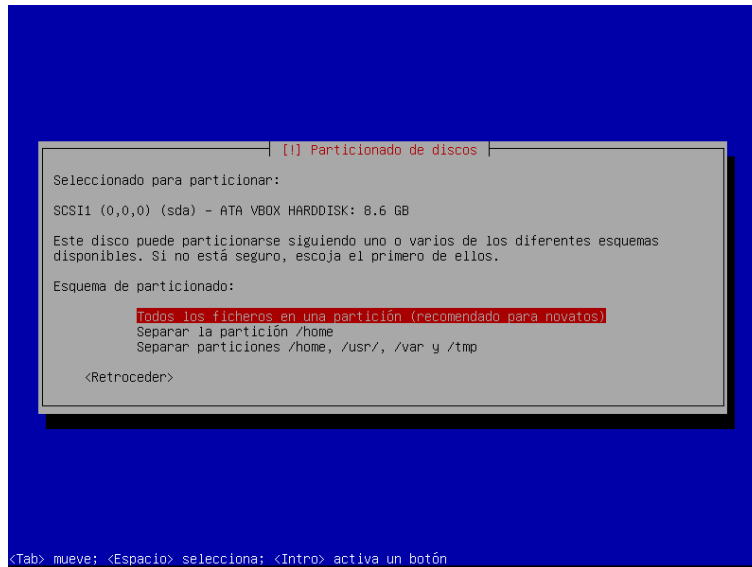


Figura D.7: Particionado de Discos  
Fuente: Autora

11. Configurar el gestor de paquetes se elige una réplica de Debian.
12. Seleccionar los programas a instalar, *Figura D.8*. Como se dijo anteriormente es necesario que el administrador de las Máquinas virtuales sea configurado en modo gráfico, en esta ocasión también se elige la primera opción.
13. Instalar el cargador del GRUB, y con esto concluye la instalación del sistema de base para el entorno de estación de trabajo de gestión KVM. Al final del proceso de instalación, se reiniciará la máquina virtual.

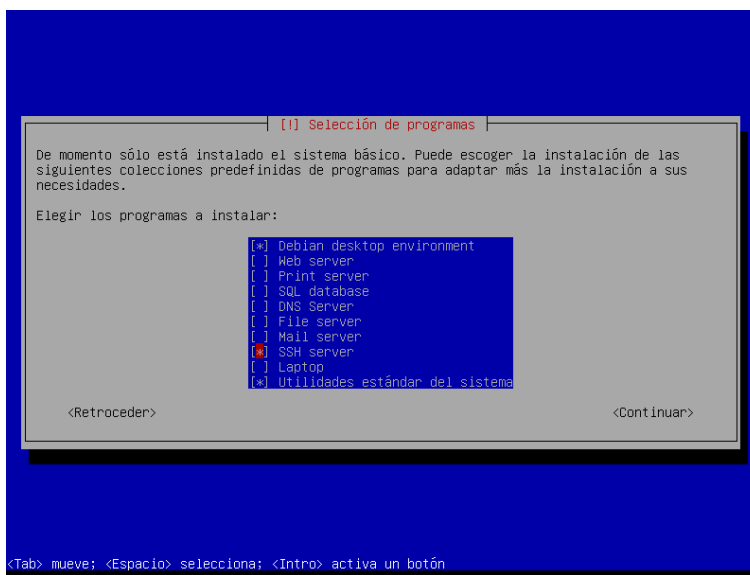


Figura D.8: Pantalla seleccionar programas  
Fuente: Autora

# Anexo E

## Instalación y configuración de *virt-manager*

1. Después de que Debian 7 reinicia la Máquina en VirtualBox, se observará la pantalla de arranque inicial. Se debe Iniciar sesión en la cuenta de usuario creada durante la instalación y seguir con los siguientes pasos<sup>1</sup>.

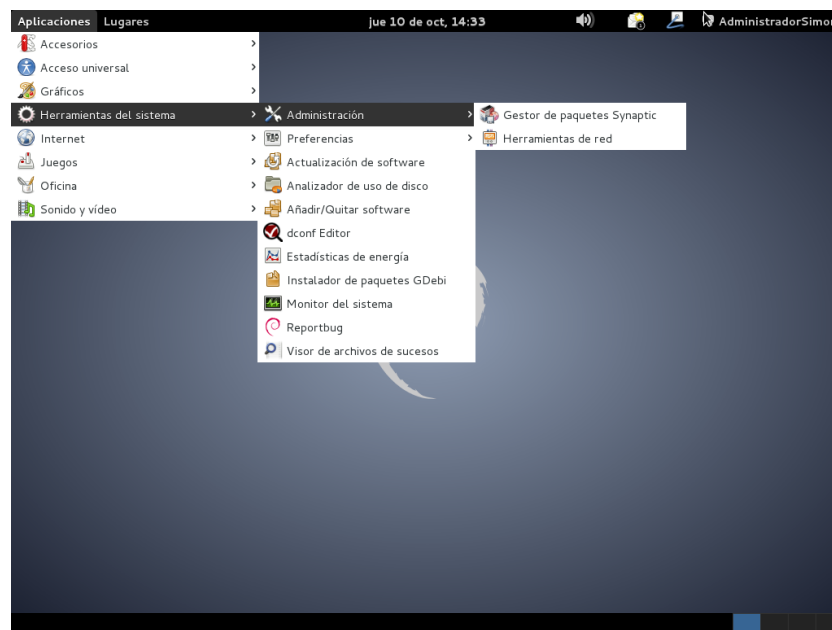


Figura E.1: Elección de Synaptic - Gestor de paquetes  
Fuente: Autora

<sup>1</sup>Se toma como referencia la fuente digital *palegray.net: Install and Configure virt-manager on the Workstation*, tomada de la página:  
<http://www.palegray.net/tutorials/debian-6-kvm-howto/configure-virt-manager>

2. Iniciar el *Synaptic*<sup>2</sup> gestor de paquetes en el menú del sistema, *Figura E.1*.
3. Hacer clic en el botón *Buscar*, introduzca *virt-manager* en el campo de búsqueda y seleccionar el paquete *virt-manager* para la instalación, *Figura E.2*.

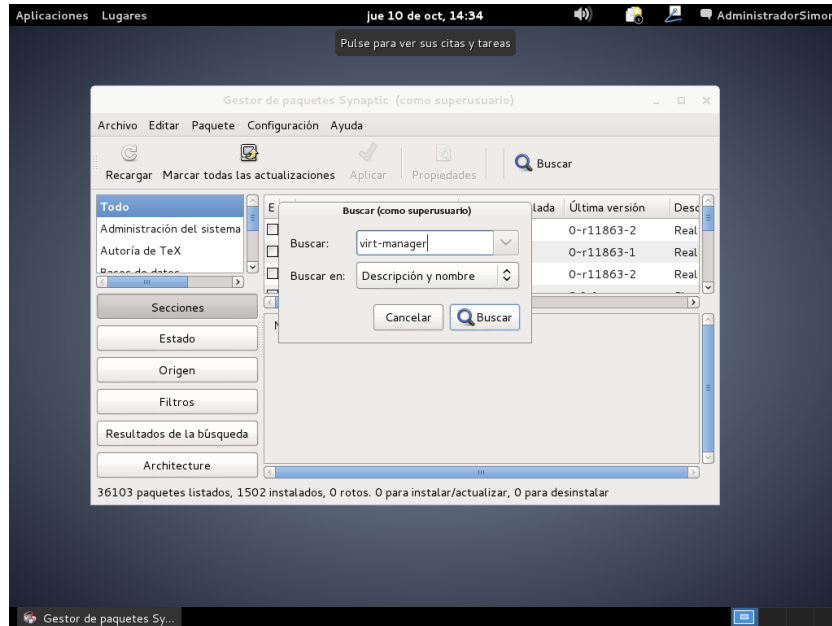


Figura E.2: Elección de *virt-manager* para su instalación  
Fuente: Autora

4. Realizar una nueva búsqueda para el paquete *ssh-askpass* y para su respectiva instalación, *Figura E.3*.
5. Hacer clic en el botón *Aplicar* para iniciar el proceso de descarga e instalación, *Figura E.4*.

<sup>2</sup>Synaptic es un programa muy potente para instalar o eliminar aplicaciones en Debian GNU/Linux, con un entorno gráfico amigable. Synaptic tiene el control de los paquetes (aplicaciones) instalados en el sistema, sus dependencias y desde aquí se seleccionan los diferentes paquetes a instalar según las necesidades del usuario. Dispone de un avanzado filtro de búsqueda, es capaz de reparar las dependencias rotas de paquetes y permite deshacer y rehacer las últimas selecciones de paquetes. Generalmente se utiliza Synaptic para sistemas basados en paquetes .deb (Debian y derivados) pero también puede ser usado en sistemas basados en paquetes RPM (Red Hat y derivados). <http://usuariodebian.blogspot.com/2011/12/synaptic-gestor-de-paquetes.html>

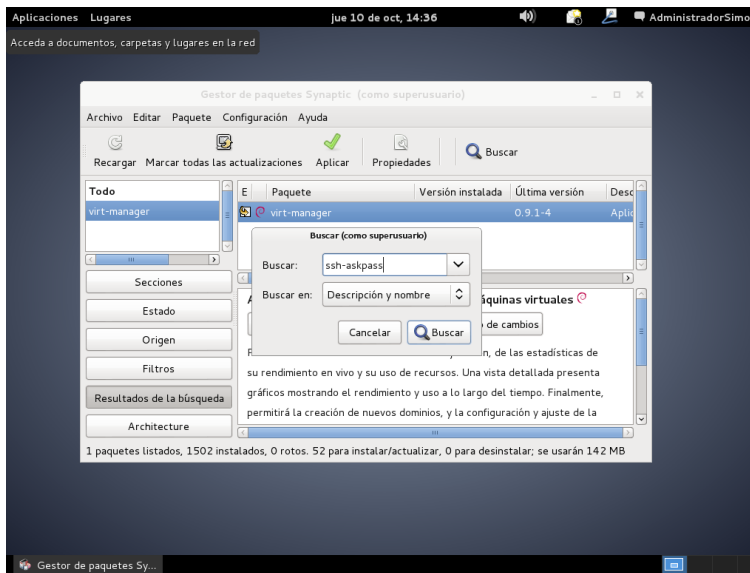


Figura E.3: Elección de ssh-askpass para su instalación  
Fuente: Autora

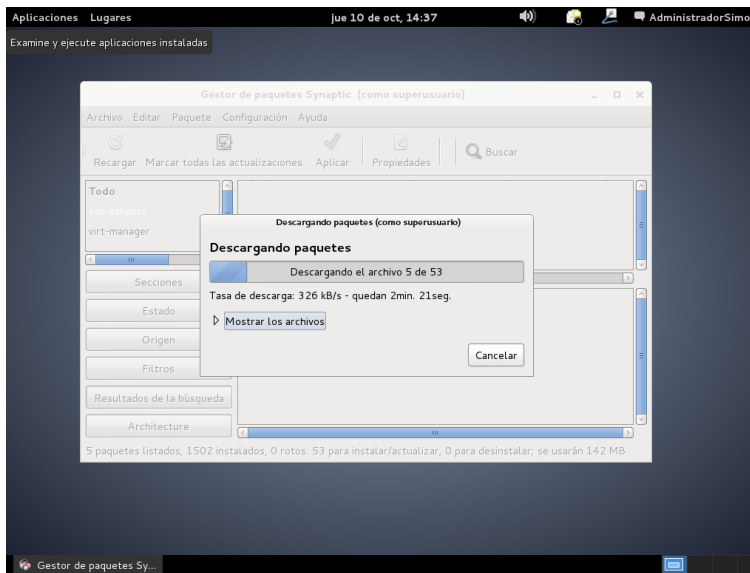


Figura E.4: Descarga de paquetes  
Fuente: Autora

6. En el menú *Sistema*, hacer click en *Administración*, seguido de *Usuarios y grupos* para abrir *la herramienta de administración del grupo*.

7. En el cuadro de diálogo que aparece, hacer click en el botón *Administrar Grupos*.

8. Se mostrará una lista de los grupos. Seleccionar el grupo *libvirt* y hacer click en el botón *Propiedades*.
9. Marcar la casilla junto a su cuenta de usuario en el *Grupo de los Miembros*, lista que se muestra en el cuadro de diálogo de propiedades.
10. Reiniciar el sistema para que tome los cambios.
11. Iniciar sesión de nuevo e ingresar a la consola de comando y editar el archivo */etc/hosts*. Al igual que con todas las operaciones que requieren privilegios de administrador, se tendrá que introducir la contraseña *root*. El archivo debe quedar configurado como se muestra en la *Figura E.5*.

```
127.0.0.1      localhost
192.168.65.239  simon.uis.edu.co      simon

# The following lines are desirable for IPv6 capable hosts
::1          localhost ip6-localhost ip6-loopback
ff02::1     ip6-allnodes
Ff02::2     ip6-allrouters
```

Figura E.5: Configuración del archivo */etc/hosts*  
Fuente: Autora

12. Generar un par de claves SSH. con el siguiente comando: (se deben aceptar todos los valores predeterminados cuando se soliciten)

```
ssh-keygen -t rsa
```

13. Emitir los siguientes comandos para copiar clave SSH en la estación de trabajo al archivo *authorized\_keys2* del servidor KVM. Sustituya la dirección IP del servidor KVM como se muestra en la *Figura E.6*.

Con esto se concluye los pasos de configuración para la estación de trabajo de gestión KVM. A continuación, se explicará como implementar la primera máquina virtual.

```
scp ~/.ssh/id_rsa.pub root@192.168.65.239:/root/workstation-  
key.pub  
ssh root@192.168.65.239| "cat ~/workstation-key.pub >>  
~/.ssh/authorized_keys2"
```

Figura E.6: Copiar clave SSH pública  
Fuente: Autora

# Anexo F

## Despliegue de las Máquinas Virtuales

1. Para empezar se debe seleccionar *Administrador de Máquina Virtual* en el menú *Aplicaciones y Herramientas del sistema*, como se muestra en la *Figura F.1*, y muestra la siguiente pantalla, *Figura F.2*.

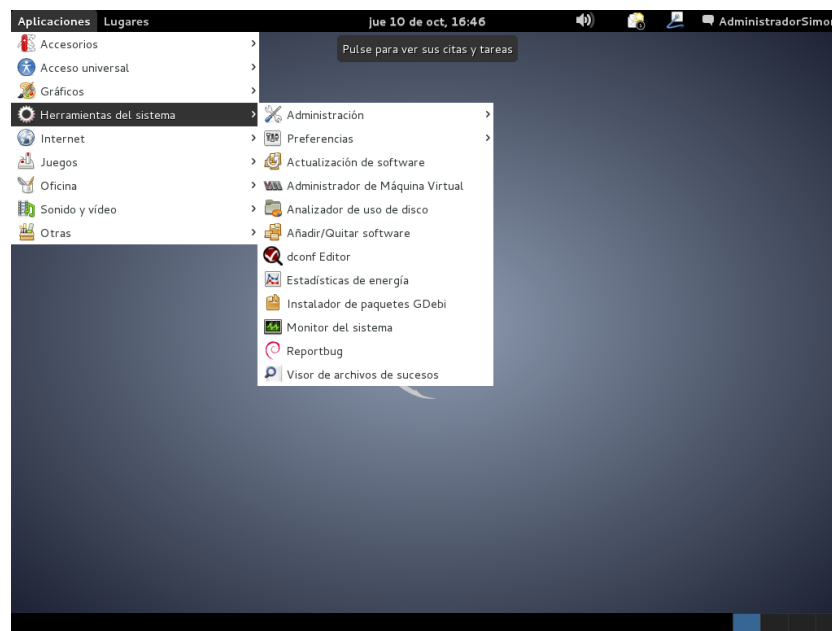


Figura F.1: Ingreso al administrador de Máquina Virtual

Fuente: Autora

2. Seleccionar *Detalles de la conexión* en el menú *Editar*, *Figura F.3*.

3. Desmarcar la casilla *Autoconectar* en la pestaña *Repaso*, de forma predeterminada, el Administrador de máquina virtual intentará conectarse a la máquina local, pero esto no es necesario, *Figura F.3*.

4. De vuelta a la pantalla de *Aplicaciones*, seleccionar *Agregar conexión* desde el menú *Archivo*, *Figura F.4*

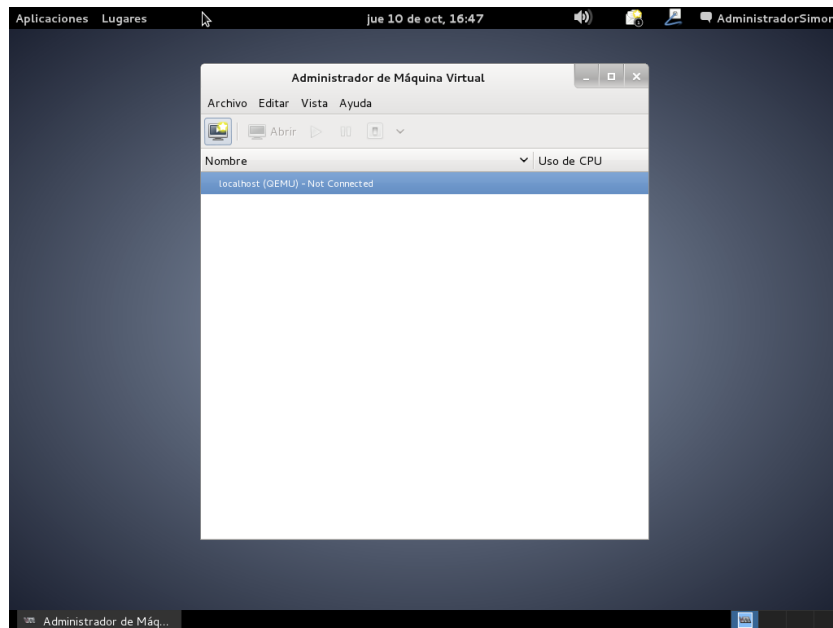


Figura F.2: Administrador de Máquina Virtual  
Fuente: Autora

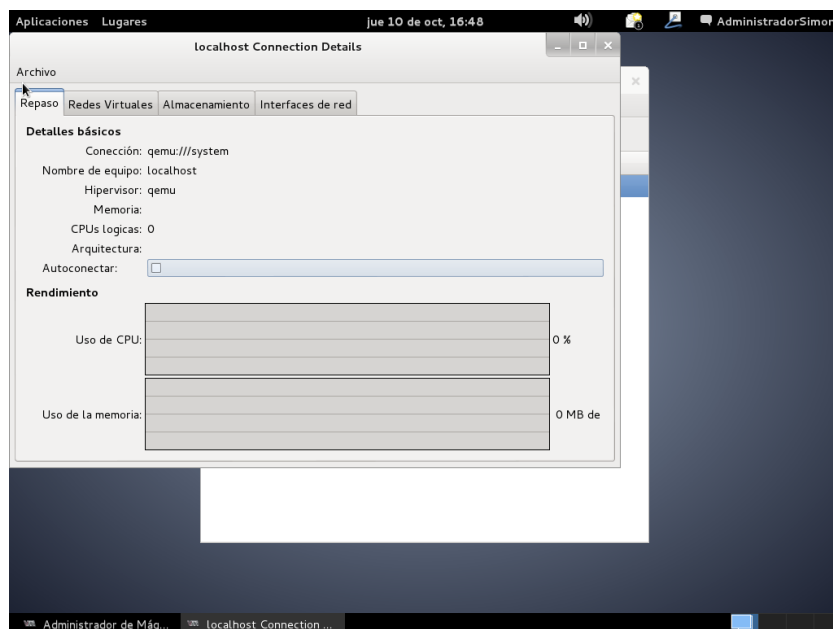


Figura F.3: Detalles de la conexión  
Fuente: Autora

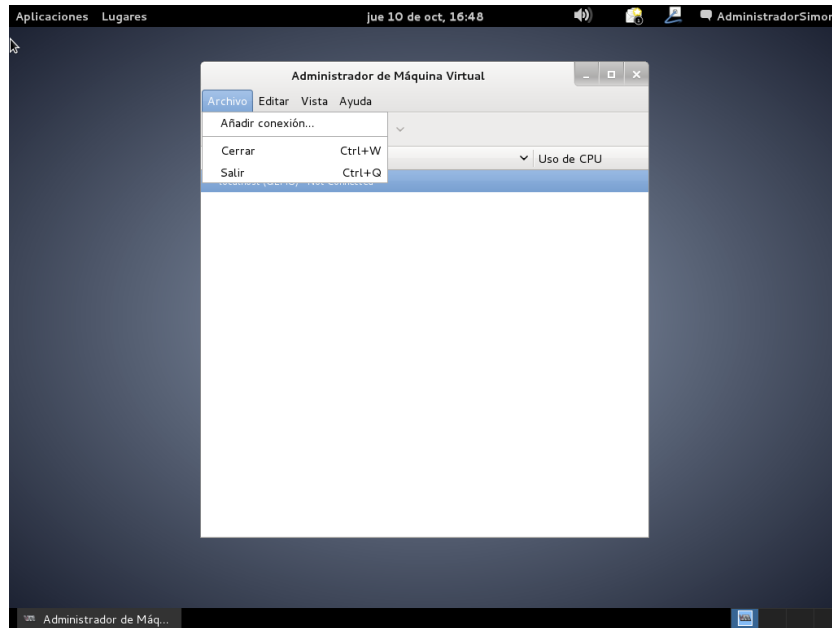


Figura F.4: Administrador de Máquina Virtual

Fuente: Autora

5. Seleccionar *QEMU/KVM* para la opción *hipervisor*, *Conectar a equipo remoto* a través del método *SSH*. Para lograr la conexión, escriba el nombre de host del servidor KVM en el campo *Nombre de host* o la dirección IP *192.168.65.239*, como es el caso, *Figura F.5*. Luego marque la casilla *Autoconectar* y haga clic en *Conectar* para continuar.
6. Hacer clic derecho sobre la entrada de servidor KVM en la lista de hosts y seleccionar *Nuevo* para comenzar a crear una nueva máquina virtual, *Figura F.6*.
7. A continuación inicia la creación de las Máquinas Virtuales. Asignar un nombre a la máquina virtual y elegir el medio de Instalación del Sistema Operativo, *Figura F.7*.
8. Seleccionar la imagen ISO de *Debian 7* que previamente se ha descargado en el servidor *Figura F.8*. Seleccionar *Linux* en la lista de *Tipo de OS* y seleccione *Debian Wheezy* en *Versión*.
9. Seleccionar la cantidad de memoria *RAM* y el número de *CPUs* virtuales que desea asignar a la máquina virtual, *Figura F.9*, y haga clic en *Siguiente* para continuar.

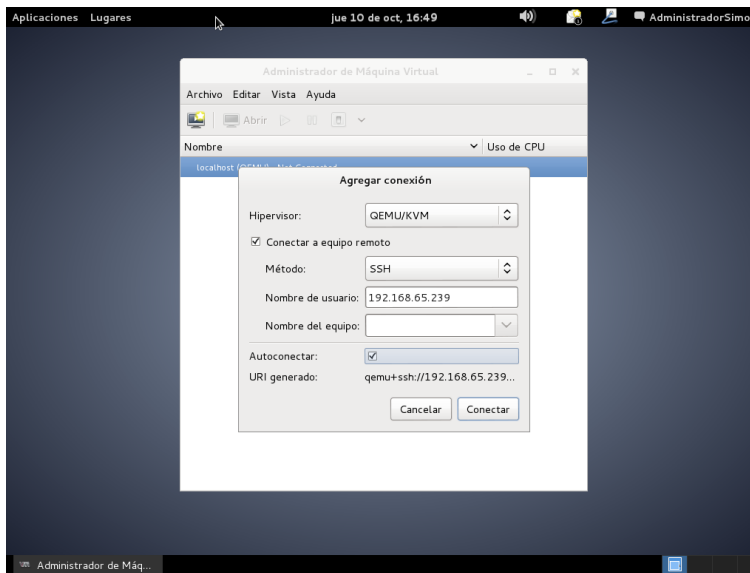


Figura F.5: Conexión remota al Host  
Fuente: Autora

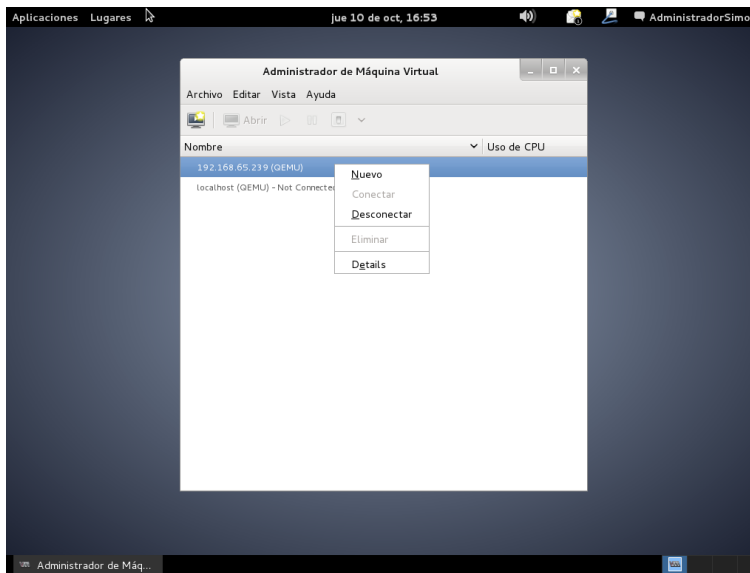


Figura F.6: Creación de una nueva máquina virtual

10. Seleccionar la cantidad de espacio de almacenamiento en disco que se desee asignar en el disco duro virtual de la Máquina . De manera predeterminada, se creará un archivo de imagen de disco virtual para el máquina en `/var/lib/libvirt/images`. En este caso se opta por asignar el almacenamiento en la partición de disco `/vm/discosvirtuales`, *Figura F.10*.

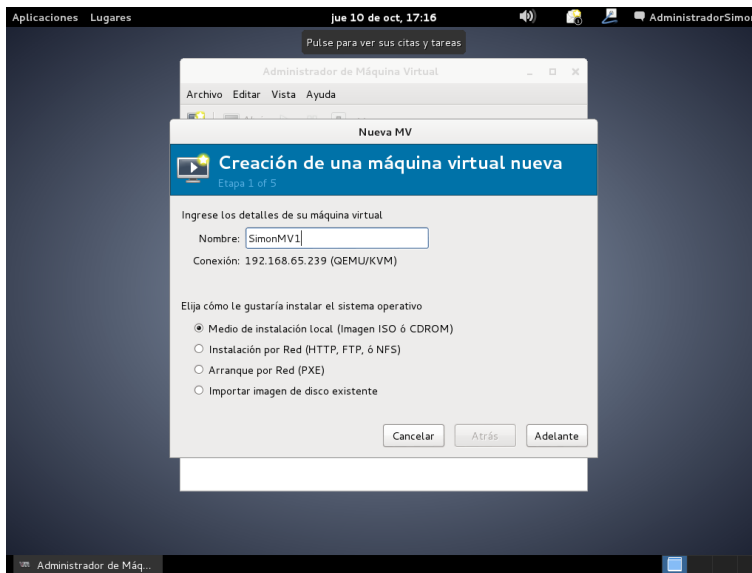


Figura F.7: Asignar nombre y medio de Instalación a la MV  
Fuente: Autora

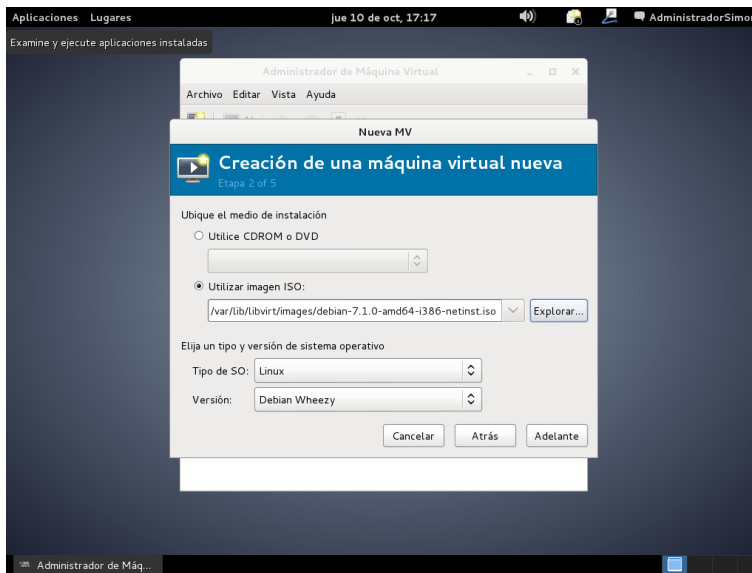


Figura F.8: Ubicación del medio de instalación y SO  
Fuente: Autora

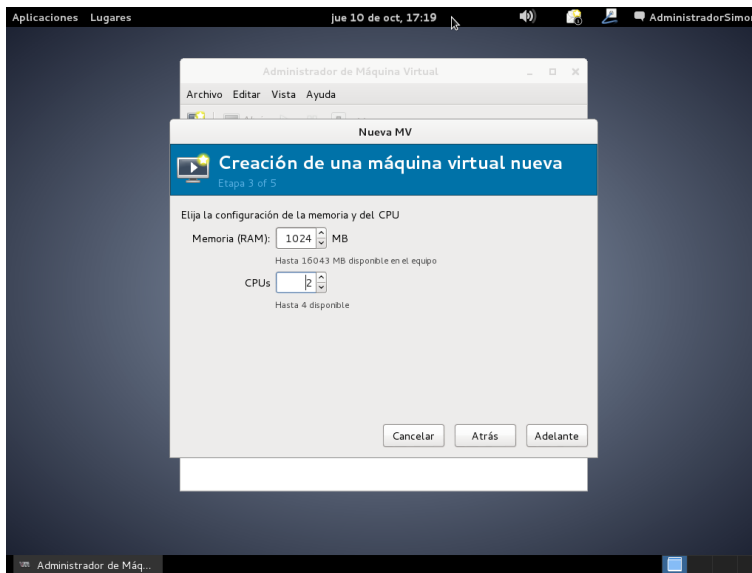


Figura F.9: Asignar la cantidad de memoria y CPUs a la MV  
Fuente: Autora

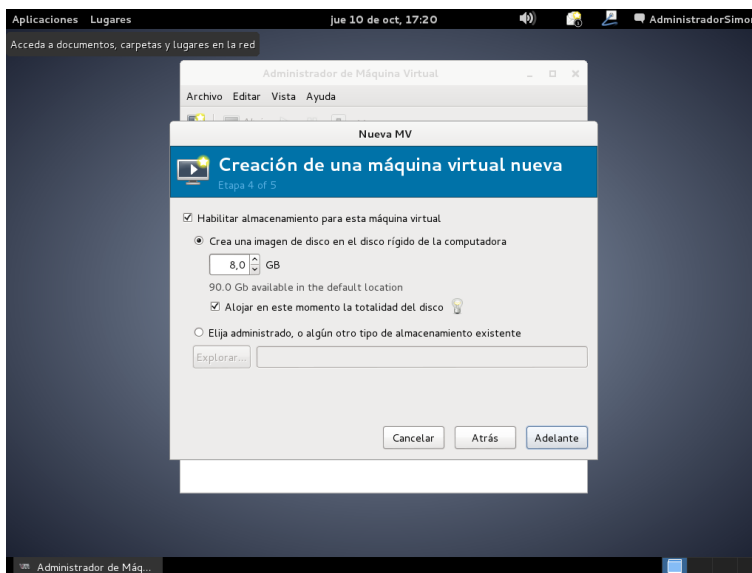


Figura F.10: Selección de la cantidad de almacenamiento en disco  
Fuente: Autora

11. A continuación se debe expandir la sección *Opciones avanzadas* en la pantalla de resumen y se debe seleccionar la configuración br0 en el campo *Nombre del Puente*, y a continuación seleccionar el *Tipo de Virtualización, KVM*, Figura F.11. Hacer clic en *Finalizar* para inicializar y arrancar la máquina virtual invitada.

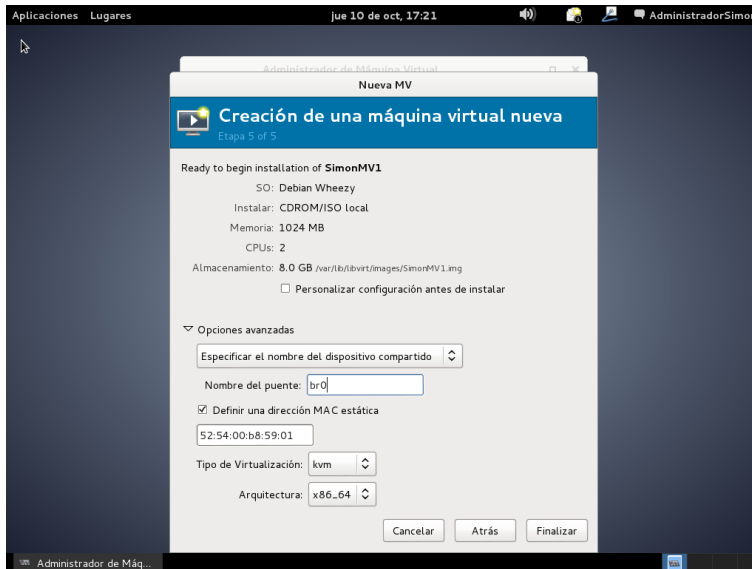


Figura F.11: Selección de configuración de red y KVM  
Fuente: Autora

12. A continuación se pedirá la clave del root de la máquina para arrancar, *Figura F.12*.

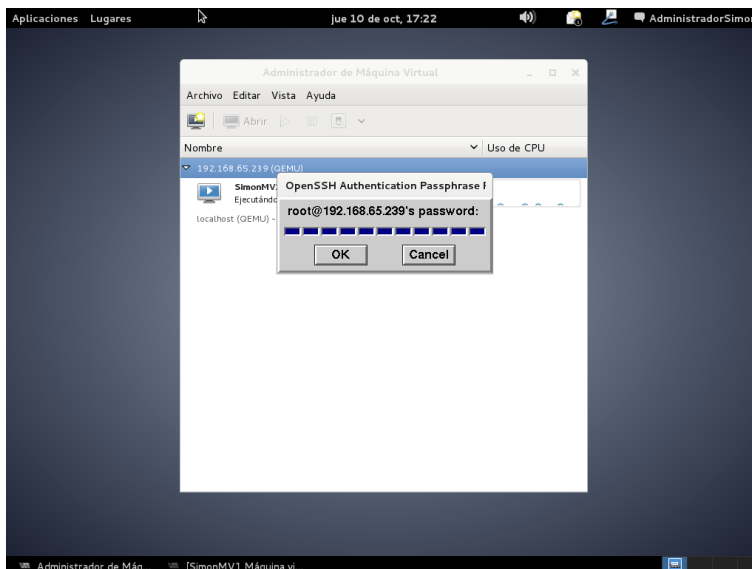


Figura F.12: Autenticación para el ingreso a la nueva MV  
Fuente: Autora

13. Una vez que la Máquina invitada ha arrancado por primera vez, se procederá instalar el sistema operativo invitado como se hizo en la máquina Administrador de VirtualBox, *Figura F.13*.

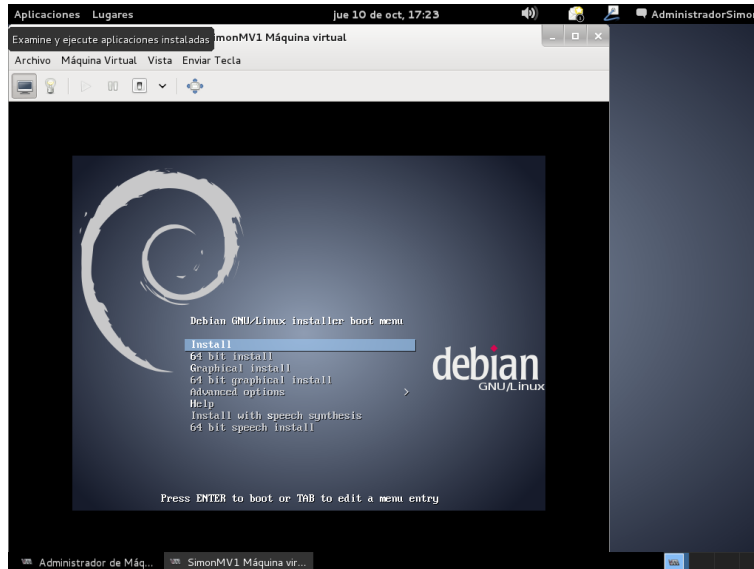


Figura F.13: Instalación del Sistema Operativo de la nueva MV  
Fuente: Autora

14. El siguiente paso es el montaje de los recursos tecnológicos en cada máquina para el soporte y mantenimiento de cada aplicativo. El proceso de montaje de los recursos tecnológicos de cada máquina se encuentra descrito en el documento *Bitácora interna servidor Grupo Simon*

# Anexo G

## Instalación *proxy inverso* en el *servidor host*

El *proxy inverso*, se instala en el servidor host, ingresando como usuario *root* y siguiendo los siguientes pasos:

1. Se deben actualizar los repositorios de *debian*, con el comando:

```
# apt-get update
```

2. Actualizar los paquetes disponibles y sus versiones del sistema.

```
# apt-get upgrade
```

3. Instalar apache2.

```
# apt-get install apache2
```

4. Añadir el módulo de proxy de apache2

```
# apt-get install libapache2-mod-proxy-html
```

5. Configurar Apache2 para que utilice el módulo *proxy*. Es necesario indicarle a *apache* la carga de librerías del mdulo, si no se le indica no aceptará las directivas para configurar el *proxy*. Para ello se edita el archivo `httpd.conf` ingresando con:

```
# nano etc/apache2/httpd.conf
```

Modificando el archivo se debe mostrar de la siguiente manera:

```
LoadModule proxy_module /usr/lib/apache2/modules/mod_proxy.so
LoadModule proxy_connect_module
/usr/lib/apache2/modules/mod_proxy_connect.$
LoadModule proxy_ftp_module /usr/lib/apache2/modules/mod_proxy_ftp.so
LoadModule proxy_http_module /usr/lib/apache2/modules/mod_proxy_http.so
```

6. El siguiente paso es habilitar los módulos necesarios de apache.

```
# a2enmod proxy proxy_http
```

7. Reiniciar apache2.

```
# /etc/init.d/apache2 restart
```

8. Lo que se quiere es lograr que apache se comporte cómo un proxy para cualquier Virtual Hosts configurado, para ello se añaden en el archivo de configuración de apache `/etc/apache2/httpd.conf` las directivas correspondientes. En el documento *Bitácora interna Servidor Grupo Simon*, se especifica más detalladamente la configuración del *proxy inverso*.

# Anexo H

## Instalación de la herramienta *Fail2ban*

*Fail2ban* es una aplicación para la prevención de intrusos en un sistema, bloquea las conexiones que intentan accesos por fuerza bruta al sistema. Se usará esta herramienta para evitar accesos fallidos al sistema por ssh. La instalación y configuración se hará en el el sistema host y los siguientes son los pasos a realizar:

1. Ingresar a la consola de comando como *root* e instalar *fail2ban*, con el siguiente comando:

```
# apt-get install fail2ban
```

2. Luego de instalado se procede a configurar el de *Fail2ban* ingresando con el comando `nano /etc/fail2ban/jail.local`, aquí se establecen los parámetros como la dirección IP a proteger, es decir la del Host, el numero máximo de intentos que se dese que *Fail2ban* pase por alto, es este caso se coloco un máximo de 4 fallos y el tiempo de bloqueo de la IP que intenta acceder erróneamente de 900 segundos. La *Figura H.1* muestra ésta configuración.

3. Este paso consiste en configurar (unas líneas más abajo), los parámetros de conexión por SSH como se muestra en la jaula de la *Figura H.2*.

4. Una vez terminado de configurar el archivo de *Fail2ban*, se reinicia el servicio con el comando:

```
# service fail2ban restart
```

5. Por último se comprueba que el funcionamiento sea correcto con el comando:

```
#service fail2ban status
```

```
Terminal
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 2.2.6      Fichero: /etc/fail2ban/jail.conf      Modificado

# Fail2Ban configuration file.
#
# This file was composed for Debian systems from the original one
# provided now under /usr/share/doc/fail2ban/examples/jail.conf
# for additional examples.
#
# To avoid merges during upgrades DO NOT MODIFY THIS FILE
# and rather provide your changes in /etc/fail2ban/jail.local
#
# Author: Yaroslav O. Halchenko <debian@onerussian.com>
#
# $Revision$
#

# The DEFAULT allows a global definition of the options. They can be overridden
# in each jail afterwards.

[DEFAULT]

# "ignoreip" can be an IP address, a CIDR mask or a DNS host
ignoreip = 192.168.65.239
bantime  = 900
maxretry = 3

# "backend" specifies the backend used to get files modification. Available
# options are "gamin", "polling" and "auto".
# yoh: For some reason Debian shipped python-gamin didn't work as expected
#      This issue left ToDo, so polling is default backend for now
backend = auto

#
# Destination email address used solely for the interpolations in
# jail.{conf,local} configuration files.

^G Ver ayuda      ^O Guardar      ^L Leer Fich     ^P Pág Ant      ^X CortarTxt
^Y Salir         ^J Justificar   ^N Buscar       ^R Pág Sig     ^U PegarTxt
^_ Pos actual    ^T Ortografía
```

Figura H.1: Configuración de los parámetros Ips, tiempos e intentos  
Fuente: Autora



```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
GNU nano 2.2.6 Fichero: /etc/fail2ban/jail.conf Modificado

#
# in /etc/fail2ban/jail.local.
#
# Optionally you may override any other parameter (e.g. banaction,
# action, port, logpath, etc) in that section within jail.local

[ssh]
enabled = true
port = ssh
filter = sshd
logpath = /var/log/auth.log
maxretry = 4

[dropbear]
enabled = false
port = ssh
filter = sshd
logpath = /var/log/dropbear
maxretry = 4

[apache]
enabled = true
port = http,https
filter = apache-auth
logpath = /var/log/apache*/error.log
maxretry = 4

# Generic filter for pam. Has to be used with action which bans all ports

^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 ^I Ortografía
```

Figura H.2: Configuración del parámetro SSH  
Fuente: Autora

# Anexo I

## Instalación y configuración de *bacula*

*Bacula* es una herramienta libre, para la creación de copias de respaldo de equipos bajo redes IP. *Bacula* se encarga de gestionar los trabajos de copia y el medio de almacenamiento.

Los elementos necesarios para que *Bacula* funcione son<sup>1</sup>

- *Bacula-director*: Es el demonio que gestiona la lógica de los procesos de backup y los demás servicios. En el archivo de configuración de este demonio se especifica dónde y cómo acceder al resto de demonios y los recursos, la contraseña para el acceso mediante *bacula-console* y los trabajos o *Jobs*.
- *Bacula-storage daemon*: Este demonio es el encargado de manejar los dispositivos de almacenamiento; el fichero de configuración de este servicio define tanto los dispositivos de almacenamiento que maneja, como que directores pueden utilizarlo. *Bacula-file daemon*: Mediante este demonio *Bacula* obtiene los ficheros que necesita respaldar, este es el componente que hay que instalar en las máquinas que necesiten respaldo. Este archivo de configuración es el más simple de todos, simplemente especifica qué directores pueden realizarle peticiones.
- Para poder interactuar con el servicio de backup, necesitaremos un cliente: *Bacula-console*

Los pasos para la instalación y configuración de *Bacula* se describen a continuación <sup>2</sup>

1. Se deben actualizar los repositorios de *debian*, con el comando:

```
# apt-get update
```

2. Actualizar los paquetes disponibles y sus versiones del sistema.

```
# apt-get upgrade
```

3. Instalar *apache2*.

---

<sup>1</sup>Tomado de la página web:

<http://www.vafe.com.ar/2008/09/18/bacula-y-los-1000-y-un-servidores/>

<sup>2</sup>se toman como referencia las fuentes digitales:

*Bacula en Debian*: <http://linuxtargz.blogspot.com/2009/07/bacula-en-debian.html> y

*Instalar y configurar Bacula en un VPS*: <http://www.digitalvalley.com/blog/instalar-y-configurar-bacula-en-un-vps/>

```
# apt-get install apache2
```

#### 4. Instalar MySQL

```
# apt-get install mysql-server mysql-client
```

Durante el proceso de instalación nos pedirá digitar varias veces la contraseña para el usuario *root* de *MySQL*

#### 5. Instalar PHP.

```
# apt-get install php5
```

#### 6. Instalar los paquetes necesarios de cliente y servidor Bacula en el servidor.

```
# apt-get install bacula-client bacula-console bacula-director-mysql  
bacula-sd
```

Se puede comprobar que en el archivo *init.d*, se tienen tres servicios nuevos: *bacula-dir*, *bacula-sd* y *bacula-fd*. Pero antes de iniciarlos se debe crear la base de datos de *Bacula* que es el siguiente paso.

#### 7. Configurar la base de datos que *bacula-director* va a usar, ingresar a MySQL con el siguiente comando:

```
# mysql -u root -p
```

Luego se crea la base de datos con:

```
mysql>create database bacula;
```

```
mysql>exit
```

La *Figura I.1* describe el proceso.

#### 8. Dar los permisos necesarios al archivo *bacula* con los comandos :

```
# chown -R bacula:bacula /bacula
```

```
# chmod -R 700 /bacula
```

9. El siguiente paso consiste en la configuración de *Bacula*. Los archivos de configuración se encuentran en */etc/bacula*. Lo primero que se configura es el directorio de *Bacula*, ingresando con el comando:

```
# nano /etc/bacula/bacula-dir.conf
```

```

Terminal
Archivo Editar Ver Buscar Terminal Ayuda
0)
root@simon:~# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 47
Server version: 5.5.35-0+wheezy1 (Debian)

Copyright (c) 2000, 2013, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> select user,host,password from mysql.user;
+-----+-----+-----+
| user          | host          | password          |
+-----+-----+-----+
| root          | localhost    | *B2E3E85820C5C0FC3D548B81894D4A49AD3ED809 |
| root          | simon        | *B2E3E85820C5C0FC3D548B81894D4A49AD3ED809 |
| root          | 127.0.0.1    | *B2E3E85820C5C0FC3D548B81894D4A49AD3ED809 |
| root          | ::1          | *B2E3E85820C5C0FC3D548B81894D4A49AD3ED809 |
|              | localhost    |                  |
|              | simon        |                  |
| debian-sys-maint | localhost    | *D9EE2074DBF4107DD25AB8A8C7DD798E739685C7 |
| bacula        | localhost    | *B2E3E85820C5C0FC3D548B81894D4A49AD3ED809 |
+-----+-----+-----+
8 rows in set (0.00 sec)

mysql>

```

Figura I.1: Creación de la Base de Datos bacula  
Fuente: Autora

Unas pocas secciones deben ser modificadas para que funcione el backup. Primero se busca la sección *Standard Restores template*, en la zona de configuración de *Job* y se cambia el valor de *Where* por la ubicación de la carpeta de restauración que automáticamente se crea al instalar la herramienta, ubicación */bacula/restore*. Luego, se busca la sección *List of files to be backed up*, debajo de *FileSet* y se añadimos una *Gzi* a la hora de comprimir los archivos, la *Figura I.2* muestra la configuración.

Por otra parte por medio del parámetro *File*, se indica aquello de lo que queremos hacer copia de seguridad. Se pueden tener más de un parámetro en la configuración, en este caso, se le indica que se quiere hacer copia de todo el directorio root (/). La *Figura I.3* muestra la configuración

10. El siguiente paso es la configuración de *bacula-sd.conf*, qué es donde se define el área donde *Bacula* almacena sus copias de seguridad. Ingresamos al archivo de configuración con:

```
# nano /etc/bacula/bacula-sd.conf
```

En el parámetro *Archive Device* se debe indicar la localización del backup, en este caso */bacula/backup*. Se guarda el archivo y se procede al siguiente paso.

11. Para asegurar que no existan errores en la sintaxis de los archivos modificados, se deben ejecutar los siguientes comandos:

```
# bacula-dir -tc /etc/bacula/bacula-dir.conf
```

```

#
# Standard Restore template, to be changed by Console program
# Only one such job is needed for all Jobs/Clients/Storage ...
#
Job {
  Name = "RestoreFiles"
  Type = Restore
  Client=simon-fd
  FileSet="Full Set"
  Storage = File
  Pool = Default
  Messages = Standard
  Where = /bacula/store
}

# List of files to be backed up
FileSet {
  Name = "Full Set"
  Include {
    Options {
      signature = MD5
      compression = GZIP
    }
  }
}

# Put your list of files here, preceded by 'File =', one per line
# or include an external list with:
#
#   File = <file-name>
#
# Note: / backs up everything on the root partition.
# if you have other partitions such as /usr or /home
# you will probably want to add them too.
#

```

Figura I.2: Configuración del directorio de bacula  
Fuente: Autora

```
# bacula-sd -tc /etc/bacula/bacula-sd.conf
```

El primer comando es para revisar el directorio de configuración de *Bacula* y el segundo el área de almacenamiento del backup. Si al ejecutar ambos comando no se obtiene una respuesta indica que los directorios no presentan errores.

12. Arrancar los servicios con los comandos:

```
# /etc/init.d/bacula-dir start
```

```
# /etc/init.d/bacula-sd start
```

13. Si todo esta bien a la hora de reiniciar los servicios de *Bacula*, estos no tendrán errores, la *Figura I.3* describe esta prueba.

```
Terminal
Archivo  Editar  Ver  Buscar  Terminal  Ayuda
GNU nano 2.2.6  Fichero: bacula-dir.conf

Name = "Full Set"
Include {
  Options {
    signature = MD5
    Compression = GZIP
  }
#
# Put your list of files here, preceded by 'File =', one per line
# or include an external list with:
#
# File = <file-name>
#
# Note: / backs up everything on the root partition.
# if you have other partitions such as /usr or /home
# you will probably want to add them too.
#
# By default this is defined to point to the Bacula binary
# directory to give a reasonable FileSet to backup to
# disk storage during initial testing.
#
File = /
}

#
# If you backup the root directory, the following two excluded
# files can be useful
#
Exclude {
  File = /var/lib/bacula
  File = /nonexistent/path/to/file/archive/dir
  File = /proc
  File = /tmp
  File = /.journal
  File = /.fsck
#
# Ver ayuda  ^O Guardar  ^R Leer Fich  ^Y Pág Ant  ^K CortarTxt  ^C Pos actual
# Salir      ^J Justificar ^W Buscar    ^V Pág Sig   ^U PegarTxt   ^T Ortografía
```

Figura I.3: Configuración del directorio de bacula  
Fuente: Autora

```
Terminal
Archivo Editar Ver Buscar Terminal Ayuda
mysql> show tables;
+-----+
| Tables_in_bacula |
+-----+
| BaseFiles          |
| CDImages           |
| Client             |
| Counters           |
| Device             |
| File               |
| FileSet            |
| Filename           |
| Job                |
| JobHisto           |
| JobMedia           |
| Location           |
| LocationLog        |
| Log                |
| Media              |
| MediaType          |
| Path               |
| PathHierarchy      |
+-----+
root@simon:/etc/bacula# nano bacula-sd.conf
root@simon:/etc/bacula# nano bacula-sd.conf
root@simon:/etc/bacula# cd
root@simon:~# bacula-dir -tc /etc/bacula/bacula-dir.conf
root@simon:~# bacula-sd -tc /etc/bacula/bacula-sd.conf
root@simon:~# service bacula-sd restart
[ ok ] Stopping Bacula Storage daemon...: bacula-sd.
[ ok ] Starting Bacula Storage daemon...: bacula-sd.
root@simon:~# service bacula-director restart
[ ok ] Stopping Bacula Director...: bacula-dir.
[ ok ] Starting Bacula Director...: bacula-dir.
root@simon:~#
```

Figura I.4: Reiniciar servicios con bacula  
Fuente: Autora

# Anexo J

## Instalación y configuración de *Nagios3*

*Nagios3* es un sistema de monitorización libre. Monitorea los hosts y servicios que se especifiquen, alertando cuando algo sale mal y nuevamente cuando esta bien<sup>1</sup>.

La instalación y configuración de *Nagios3* se realizarán en la máquina virtual *SimonMV3* de la infraestructura virtual, máquina en modo gráfico. Los pasos son los siguientes:

1. Ingresar a la consola de comandos como usuario *root* y lo primero que se debe instalar es *apache2*, con el siguiente comando

```
# aptitude install apache2
```

2. Instalar las librerías de *apache2*

```
# aptitude install libapache2-mod-php5
```

3. Instalar el siguiente paquete *build-essential*

```
# aptitude install build-essential
```

4. Instalar *nagios3*

```
# aptitude install nagios3
```

Durante la instalación se debe se debe introducir la contraseña del usuario *nagios3* para ingresar a su área de trabajo por la web, por defecto es el usuario es *nagiosadmin*.

5. El siguiente paso es abrir el navegador web y colocar en la barra de dirección <http://localhost/nagios3>. En seguida nos muestra la siguiente ventana *Figura J.1*. Luego de la autenticación se muestra el área de trabajo de *nagios3*, *Figura J.2*.

---

<sup>1</sup>Tomado de NAGIOS EN SERVIDOR Debian: Nagios como sistema de monitorización

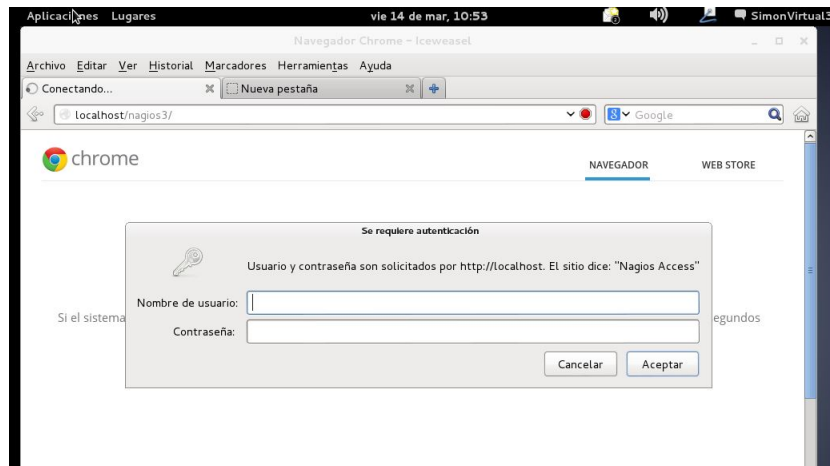


Figura J.1: Ingreso por la Web a nagios3  
Fuente: Autora

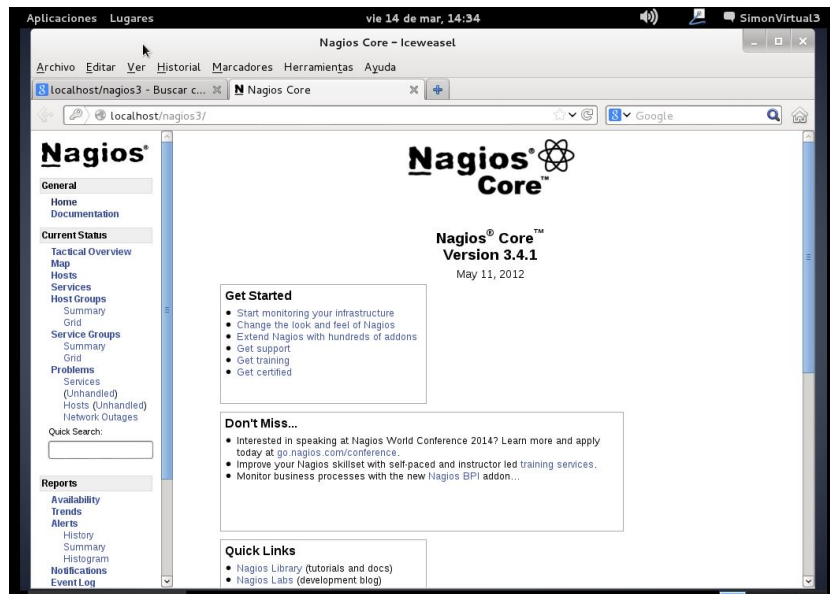


Figura J.2: Area de trabajo de nagios3  
Fuente: Autora