**Universidad Michoacana de San Nicolás de Hidalgo**

**Facultad de Ingeniería Eléctrica**

# Conexión de dispositivos de red reales con dispositivos virtuales y acceso remoto

**TESIS**

**Que para obtener el Título de**
**INGENIERO EN COMPUTACIÓN**

**Presenta**
**JULIETA GARDUÑO HERNÁNDEZ**

**Asesor de Tesis**

**M.I. SAMUEL PÉREZ AGUILAR**

**Morelia, Michoacán, Febrero de 2017**

# Agradecimientos


**A DIOS**

*Por haberme dado la sabiduría y fortaleza necesarias para afrontar cada obstáculo de la vida de manera satisfactoria, así mismo por proveerme de salud para lograr el objetivo desde que decidí seguir con una preparación académica y llegar a la meta de haber terminado, y por haberme elegido a unos padres maravillosos.*

**A MIS PADRES**

*Juana Hernández Valdés y Héctor Garduño López. El más profundo y sincero agradecimiento les corresponde a ellos que desde que tengo uso de razón han dedicado su vida para lograr que no me falta nada, ellos son la pieza fundamental de que hoy este realizando este proyecto de vida que es terminar una carrera universitaria, a quienes les reconozco lo que soy hoy en día, y la más fuerte inspiración en mi vida para seguir preparándome cada día más con el objetivo de darles una satisfacción, por todo el apoyo económico, pero primordialmente moral que incondicionalmente nos han dado a mis hermanos y a mí logrando mantener la familia que hasta hoy somos.*

**A MIS HERMANOS**

*Héctor, Liliana (qepd), Mayolo, Mauricio y Juan de Dios, con quienes he convivido y pasado momentos inolvidables y de quienes he aprendido varias cosas para lograr un objetivo, y para quienes busco ser un ejemplo positivo para que ellos también siempre busquen un futuro mejor.*

**A MI ASESOR**

*Samuel Pérez Aguilar que me apoyó de una manera excelente para realización de este proyecto, quien siempre se prestó respetuoso y accesible cuando lo requería, por transmitir conocimientos amplios y por motivarme para seguir preparándome cada día más.*

# Dedicatoria

A mis padres Juana y Héctor

A mis hermanos Héctor, Liliana (q.e.p.d.), Mayolo, Mauricio y Juan de Dios.

# *ÍNDICE*

## CAPITULO 3. VIRTUALIZACIÓN CON KVM Y ACCESO REMOTO CON DISTINTOS CLIENTES

## CAPÍTULO 4. SOFTWARE DE ADMINISTRACIÓN DE USUARIOS

# Lista de Figuras

# Lista de Símbolos y Abreviaciones

| | |
|---|---|
| **AFS** | Andrew File System |
| **AMD** | Advanced Micro Devices |
| **AoE** | ATA over Ethernet |
| **APT** | Advanced Packaging Tool |
| **BBS** | Bulletin Board System |
| **BIOS** | Basic Input/Output System |
| **BSD** | Berkeley Software Distribution |
| **CCNA** | Cisco Certified Network Associate |
| **CD-ROM** | Compact Disc, read-only-memory |
| **CLR** | Common Language Runtime |
| **CPU** | Central Processing Unit |
| **DHCP** | Dynamic Host Configuration Protocol |
| **DNS** | Domain Name System |
| **DVD** | Digital Video Disc |
| **E/S** | Entrada/Salida |
| **GFS** | Global Forecast System |
| **GPL** | General Public License |
| **HTTP** | Hypertext Transfer Protocol |
| **IBM** | International Business Machines |
| **IDC** | International Data Corporation |
| **IDE** | Integrated Development Environment |
| **IEEE** | Institute of Electrical and Electronics Engineers |
| **IP** | Internet Protocol |
| **iSCSI** | Internet Small Computer System Interface |
| **ISO** | International Organization for Standardization |
| **LAN** | Local Area Network |
| **LVM** | Logical Volume Manager |
| **MAC** | Media Access Control |
| **MIT** | Massachusetts Institute of Technology |
| **MV** | Máquina Virtual |
| **NAS** | Network Attached Storage |
| **NAT** | Network Address Translation |
| **NDG** | Natal Digital Group |
| **NFS** | Network File System |
| **OSI** | Open System Interconnection |
| **PC** | Personal Computer |
| **PHP** | Hypertext Pre-processor |
| **QoS** | Calidad del Servicio |
| **RAID** | Redundant Array of Independent Disks |

| | |
|---|---|
| **RAM** | Random-access memory |
| **RFB** | Remote Framebuffer |
| **RSPAN** | Remoted Switched Port Analyzer |
| **SAN** | Storage Area Network |
| **SATA** | Serial Advanced Technology Attachment |
| **SCP** | Security Copy |
| **SCSI** | Small Computers System Interface |
| **SFTP** | Secure File Transfer Protocol |
| **SGBD** | Sistema de Gestión de Base de Datos |
| **SHA** | Secure Hash Algorithm |
| **SPAN** | Switched Port Analyzer |
| **SSH** | Secure Shell |
| **SSL** | Secure Sockets Layer |
| **TCP/IP** | Transmission Control Protocol |
| **TFTP** | Trivial file transfer Protocol |
| **TI** | Tecnologías de Información |
| **USB** | Universal Serial Bus |
| **vHBA** | virtual Host Bus Adapter |
| **vNIC** | virtual Network Interface Card |
| **VPN** | Virtual Private Network |
| **VPS** | Virtual Private Server |
| **X11** | X Windows System |

# RESUMEN

Los equipos de laboratorio utilizados para desarrollar habilidades de diseño, configuración, diagnóstico y corrección de errores en el ámbito de la redes de datos juegan un papel muy importante para quienes pretenden una certificación en la industria de las redes de computadoras.

Como es conocido, las prácticas que se realizan dentro de los laboratorios de redes son realizadas de manera presencial, es decir, se debe estar frente a los equipos para en su caso aplicar o modificar la configuración a los equipos finales y a los equipos de capa 2 y 3 como lo son los switches y routers, para los cuales, el acceso es por medio de los puertos de consola.

El objetivo general del proyecto no es eliminar la interacción practica con los equipos físicos mediante simuladores, sino que a través de un sistema de software se le permita al usuario final, en este caso el estudiante, acceder al equipo real del laboratorio de forma remota por las noches, los fines de semana o en aquellas ocasiones cuando no es posible acceder a las instalaciones de la universidad, logrando con esto un mejor aprovechamiento de los recursos materiales y del tiempo, todo lo anterior con las consideraciones de facilidad y seguridad requeridas. Para encontrar una solución a lo anterior se realizó una valoración de las diferentes herramientas, con el fin de determinar cuál era la mejor combinación que se adapte a las necesidades requeridas para este caso, considerando a los usuarios, el equipo y el software disponible. Esta determinación se logró tras realizar las pruebas necesarias para tener el resultado final esperado, el cual fue el permitir la comunicación de forma remota.

El software involucrado en la solución comprende tanto herramientas de virtualización como de acceso remoto y software de gestión de usuarios. Este último tiene como propósito habilitar a los usuarios el acceso al hardware mediante la autenticación respectiva y la reservación del uso del equipo.

**Palabras clave:** Virtualización, acceso remoto, KVM, software de administración de usuarios, aplicación Guacamole.

## Abstract

*Laboratory equipment used to develop design, configuration, diagnostics and error correction skills in the field of data networks play a very important role for those seeking certification in the computer network industry.*

*As it is known, the practices that are carried out within the network laboratories are carried out in person, that is to say, it must be in front of the equipment in order to apply or modify the configuration to the final equipment and to the layer 2 equipment and 3 as are the switches and routers, for which, access is via the console ports.*

*The general objective of the project is not to eliminate the practical interaction with the physical equipment through simulators, but through a software system the end user, in this case the student, is allowed to access the real equipment of the laboratory remotely by nights, weekends or in those occasions when it is not possible to accede to the facilities of the university, obtaining with this a better use of the material resources and of the time, all the previous with*

*the considerations of facility and security required. In order to find a solution to the above, a valuation of the different tools was done, in order to determine the best combination that suits the needs required for this case, considering the users, equipment and software available. This determination was achieved after performing the necessary tests to have the expected final result, which was to allow communication remotely.*

*The software involved in the solution comprises both virtualization and remote access tools and user management software. The latter purpose is to enable users to access the hardware through the respective authentication and reservation of the use of the equipment.*

**Keywords**: *Virtualization, remote access, KVM, user management software, Guacamole application.*

# *Capítulo 1. Introducción*

El contenido de este capítulo habla sobre las causas que motivaron a realizar este proyecto de tesis, especificando el problema principal a resolver, denotando con ello el objetivo y justificación de este trabajo. Se da una reseña sobre los antecedentes de acceso remoto, así como de los inicios de la virtualización y su avance en la actualidad.

# 1. Introducción

Dentro de las academias de redes Cisco, conforme transcurre el tiempo, los dispositivos y los cables se desgastan debido al uso que se les da. Es por eso que hace algunos años un grupo de desarrolladores diseñaron una plataforma que brinda el soporte para mantener los equipos en un ambiente controlado, es decir, se pueden realizar prácticas sin necesidad de modificar las conexiones físicas, pero sí utilizando los equipos reales.

NDG (Natal Digital Group) Netlab fue pensado para implantarse particularmente en las academias de redes de datos, y se ha extendido su uso hasta empresas que trabajan con equipos de redes y que se controlan remotamente ya sea para fines educativos o de entrenamiento.

La Figura 1 muestra la interacción entre el equipo Netlab, los estudiantes y profesores de forma remota a través de internet [NDG 2016].

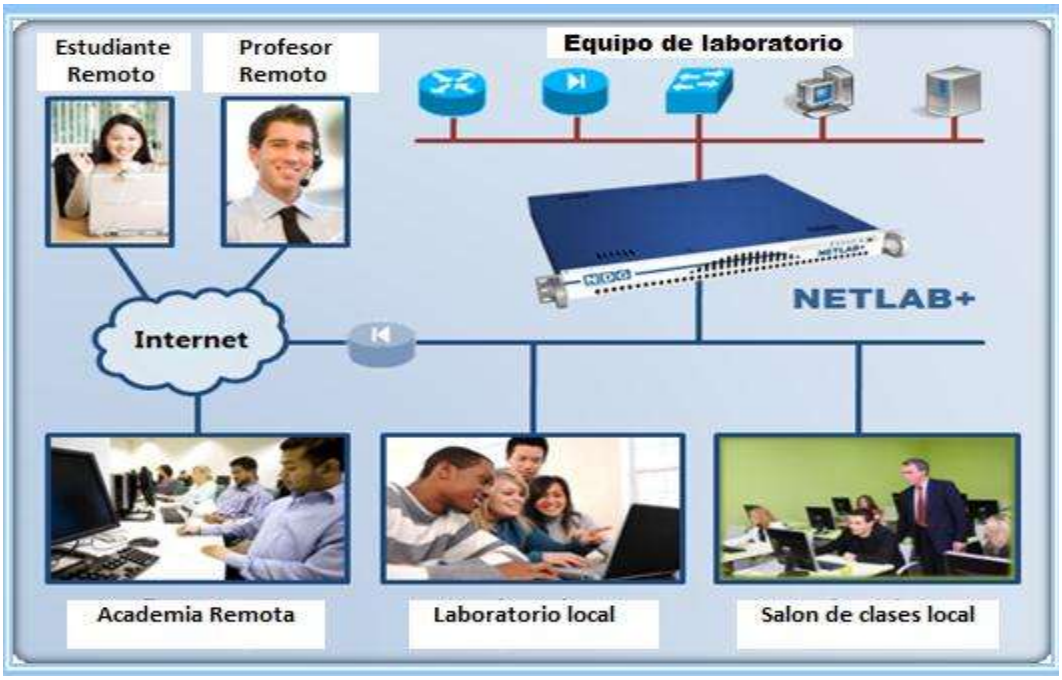


*Figura 1. Entorno Netlab*

Netlab es un equipo compuesto por una unidad de rack que se conecta a otros equipos de red auxiliares, a través de los cuales controla grupos de equipos de redes como lo son switches y routers, permitiendo a los usuarios del sistema acceder a una topología específica de laboratorio que se configura mediante una interfaz web.

Netlab ejecuta un software que considera la existencia de perfiles de instructor y estudiantes. La arquitectura de este sistema consiste en un servidor que controla físicamente los equipos de laboratorio de redes, y que cuenta con acciones de encendido, apagado y reinicio de los dispositivos cuando un usuario lo solicita de forma remota.

Los dispositivos de laboratorio son controlados mediante las salidas de consola conectadas a una interfaz que envía lo que sale de la consola a un proceso web, este también recibe lo que el usuario solicita y lo envía a la consola del dispositivo requerido. Un usuario del sistema puede reservar tiempo de uso de los equipo de laboratorio y hacer prácticas como si realmente

estuvieran trabajando con los equipos físicos.

Los equipos del laboratorio son agrupados en pequeños grupos, los cuales son armados y registrados por el administrador de la plataforma. Los laboratorios pueden ser de propósito general para realizar prácticas de CCNA (Cisco Certified Network Associate – Academia de Certificación de Redes Cisco) las cuales se basan en 3 routers y 3 switches, estos pueden ser utilizados para realizar prácticas tanto de enrutamiento como de conmutación a nivel de CCNA. El uso de esta tecnología trae consigo ventajas y desventajas, presentadas a continuación:

**Ventajas**

- Reducción en el desgaste de los equipos de laboratorio.
- Realización de prácticas de laboratorio con equipo real y no simulado como cuando se usa Cisco Packet Tracer, ya que puede ocurrir que algún comando no se pueda realizar o que el resultado no sea el esperado.
- Interfaz amigable para el usuario.
- Permite guardar configuraciones de los equipos.
- Se puede realizar una acción para varios dispositivos, por ejemplo, el apagado o reinicio de todos los dispositivos.
- Recuperación rápida y fácil de contraseña.
- Cargar la última configuración utilizada.
- Control de cantidad de horas usadas por usuario y cancelar reservas abusivas de algún estudiante.

**Desventajas**

- Licenciamiento cerrado. Existen tres tipos de licencias la más económica es el Netlab AE usada solo para actividades relacionadas a los cursos de este fabricante. Esta licencia no permite modificar el sistema operativo ya que se corre el riesgo de perderla. Esta licencia obliga a dar un mantenimiento anual que corresponde a $2,000.00 dólares, así mismo, se debe dar permiso a la empresa para que sus empleados corroboren que no se han hecho modificaciones al sistema.
- No existe suficiente documentación acerca de esta tecnología, debido a que la licencia obliga a no crear material (videos o documentación) de apoyo.

## 1.1 Antecedentes

El acceso remoto es una tecnología la cual permite a un usuario en la red trabajar en una computadora a distancia a partir de otro equipo remoto mediante el uso de un escritorio gráfico. Los eventos principales tales como las pulsaciones de teclas y movimientos del mouse, son transmitidos a la computadora sobre la cual se está trabajando de forma remota donde estos eventos son procesados como si se tratasen de eventos locales.

En el pasado, el acceso remoto a una computadora era una función realizada por las primeras computadoras, la cuales, tenían un numero de terminales de texto unidos a éstas por medio de cables.

Un terminal de texto o consola es un dispositivo electrónico de hardware, el cual es usado para mostrar los resultados arrojados por una computadora.

En un principio un terminal de consola era un dispositivo barato pero con la desventaja de

que eran muy lentos para la recepción de datos, pero debido a la rápida evolución de la tecnología y a que fueron introducidos los monitores de computadora, los terminales mejoraron su forma de interacción con el usuario.

Un ejemplo de terminal de consola son los sistemas de tiempo compartido que fueron desarrollados a finales de 1957, para (tal como el nombre lo indica) compartir recursos de la computadora entre múltiples usuarios.

Las consolas de texto constan de un terminal con una conexión serial para comunicarse con la computadora, un teclado para la entrada de datos y una pantalla para mostrar la información, la cual es vista como un arreglo de caracteres sin la introducción de gráficos.

A inicios de su creación, las consolas de texto estaban conectadas con las computadoras mediante un puerto serial. Posteriormente los equipos contaron con consolas de sistema por el cual, los usuarios daban instrucciones a un programa mediante una línea de texto, también contaban con programas emuladores de terminal, dicho programa simula una terminal de texto en un dispositivo de visualización.

Debido a los cambios que surgieron con la evolución de las redes de telecomunicaciones, en este caso el desarrollo de los ya mencionados emuladores de terminal, se puede hablar de dos de las primeras tecnologías de acceso remoto, las cuales son telnet y ssh, desarrollados en ambientes Unix.

### 1.1.1 Telnet

Telnet es un protocolo de red el cual permite manipular una computadora de forma remota como si se estuviera físicamente en dicho equipo. Este programa es ejecutado del lado del cliente

y para que la conexión con el servidor se realice satisfactoriamente es necesario que la máquina cuente con un programa que reciba las conexiones.

El funcionamiento de Telnet básicamente consiste en acceder únicamente de forma terminal, sin gráficos, de un equipo de cómputo a otro.

Inicialmente Telnet se usó para dar solución a problemas sin estar físicamente frente a la computadora, así como la extracción de datos. El principal inconveniente que surgió durante su uso fue la nula seguridad que se tiene, ya que los nombres de usuario y contraseñas viajan por la red en texto plano, es decir sin cifrar, lo cual permite que un intruso mediante un analizador de tráfico obtenga la información de manera fácil y así acceder a los equipos de cómputo.

Actualmente, Telnet es usado para acceder a los BBS (Bulletin Board System – Sistema de Tablón de Anuncios) el cual, es un software que permite a usuarios conectarse a la red ya sea por medio de internet o por una línea telefónica, y además una terminal para descargar programas, leer noticias, intercambiar información, entre otros [Telnet.org 2015].

### 1.1.2. SSH

SSH es un protocolo de red el cual, al igual que Telnet son utilizados para acceder de forma remota a una computadora con la diferencia que, SSH (Secure Shell – Interprete de ordenes seguro) utiliza mecanismos de cifrados de datos, para evitar que personas ajenas a la información puedan acceder a ella.

### 1.1.3. Virtualización de equipos

En los años 90’s las interfaces de usuario sufren una revolución a partir de la cual surgen dos nuevas tecnologías: los terminales gráficos que tan solo son una evolución de los terminales de texto anteriores, así como los escritorios gráficos realizados por Apple Macintosh y Microsoft.

El primer entorno remoto con el que se contó, fue el escritorio remoto X-Window System el cual es un software que fue desarrollado a mediados del año 1980 en el MIT (Massachusetts Institute of Technology – Instituto Tecnológico de Massachusetts) para dotar de una interfaz gráfica a los sistemas Unix. Este protocolo permite la interacción gráfica en red entre un usuario y una o más computadoras haciendo transparente la red para éste. Generalmente se refiere a la versión 11 de este protocolo, X11, el que está en uso actualmente.

X11 es el encargado de mostrar la información gráfica de forma totalmente independiente del sistema operativo. [Microsoft 2016].

El software X11 usa el modelo cliente-servidor, donde, un servidor se comunica con varios clientes, el servidor accede a sus peticiones y devuelve la entrada del usuario.

La tecnología de acceso remoto se basa en los eventos comunes de una computadora tales como presionar una tecla y los movimientos del ratón son transmitidos a un servidor central donde la aplicación los procesa como si estos fueran parte de eventos que suceden dentro del equipo en específico.

El protocolo de comunicaciones de acceso remoto varía dependiendo del programa que se use. Por ejemplo:

- ICA (Independent Computing Architecture – Arquitectura de Computo

Independiente), utilizado por MetaFrame.

- RDP (Remote Desktop Protocol – Protocolo de Escritorio Remoto), utilizado por Terminal Services.
- AIP (Adaptive Internet Protocol – Protocolo de Internet Adaptado), utilizado por Secure Global Desktop.
- VNC (Virtual Network Computing – Computación Virtual en Red), utilizado por el producto del mismo nombre.
- X11, utilizado por X-Windows.

La virtualización, comenzó a desarrollarse en la década de los sesenta para crear particiones de grandes piezas de hardware mainframe y lograr una mejor utilización.

La virtualización es el proceso de presentar un subconjunto de recursos físicos agrupados de forma lógica, de tal forma que se obtengan beneficios sobre la configuración original [Academia.edu 2014].

IBM (International Business Machines Corp.) implementó la virtualización por primera vez hace más de 30 años para particionar de manera lógica computadoras mainframe en máquinas virtuales separadas. Estas particiones permitieron que los mainframes ejecutaran múltiples tareas a la vez, es decir, que ejecutaran varias aplicaciones y procesos al mismo tiempo. Debido a que los mainframes constituían recursos costosos en ese momento, fueron diseñados para permitir la creación de particiones como un modo de aprovechar totalmente la inversión [IBM 2015].

La virtualización se abandonó durante las décadas de los ochenta y noventa, cuando aparecieron las de cliente-servidor, los servidores y los escritorios x86 poco costosos los cuales,

abrieron paso a la computación distribuida.

Con la aparición de Windows y Linux como sistemas operativos para servidores en los noventa, se establecieron los servidores x86 como los más predominantes dentro del campo.

El crecimiento en las implementaciones de servidores y escritorios x86 generó nuevos desafíos operacionales y de infraestructura de TI. Estos desafíos incluyen:

**Poca utilización de la infraestructura**. Las implementaciones estándar de servidores x86 alcanzan un promedio de uso de solo el 10% al 15% de la capacidad total, según IDC (International Data Corporation- Corporativo de datos Internacional). Las organizaciones acostumbran ejecutar una aplicación por servidor para evitar que la vulnerabilidad de una aplicación afecte la disponibilidad de otra aplicación del mismo servidor.

**Mayores costos de infraestructura física.** Los gastos operacionales que sustentan el crecimiento de la infraestructura física han ido aumentando continuamente. La infraestructura de computación debe estar en funcionamiento en todo momento, lo que implica un costo por consumo de energía, refrigeración e instalaciones.

**Mayores costos de administración de TI**. A medida que los entornos de computación se vuelven más complejos, el nivel de capacitación y experiencia requerida para el personal de administración de infraestructura y los costos asociados a tal personal aumentan. Las organizaciones invierten tiempo y dinero de manera desproporcionada en tareas manuales asociadas al mantenimiento de servidores, y eso requiere más personal.

**Insuficiencia de conmutación de recuperación y protección contra desastres.** Las organizaciones se ven cada vez más afectadas por el tiempo fuera de servicio de aplicaciones fundamentales de servidor y por la falta de accesibilidad de escritorios fundamentales de usuario final. La amenaza de los ataques a la seguridad, los desastres naturales, las pandemias y el terrorismo han puesto en manifiesto la importancia de una planificación para la continuidad del negocio relacionada con servidores y escritorios.

**Demasiado mantenimiento para los escritorios de usuario final.** La administración y la seguridad de los escritorios empresariales generan varios desafíos. Controlar un entorno distribuido de escritorio y promover políticas de administración, acceso y seguridad sin afectar el trabajo eficaz de los usuarios son tareas complejas y muy costosas. Es necesario llevar a cabo actualizaciones y aplicar parches continuamente en los entornos de escritorio, para eliminar las vulnerabilidades de seguridad. [Mera O.S 2012].

En 1999, Vmware inició la virtualización de sistemas x86 para enfrentar muchos de estos desafíos y para transformar los sistemas x86 en una infraestructura de hardware compartida de uso general que ofreciera un aislamiento total, movilidad y alternativas de sistemas operativos para los entornos de aplicaciones. [Timetoast 2014].

A continuación se presenta un cronograma acerca de la evolución de las empresas desarrolladoras de software de virtualización:

- **1998:** Se funda la empresa Vmware.
- **1999:** Vmware lanza su primer producto, Vmware Workstation
- **2003:** Se lanza la primera versión de Xen.

- **2005:** Intel introduce su tecnología VT-x (Vanderpool) en arquitecturas x86.
- **2006:** AMD (Advanced Micro Devices – Microdispositivos Avanzados) introduce su tecnología AMD-V (Pacifica).
- **2007:** KVM (Kernel-based Virtual Machine – Máquina Virtual basada en núcleo) se integra en la rama oficial del kernel de Linux 2.6.20.
- **2007:** VirtualBox OSE (Open Source Edition) se libera como software libre.
- **2008:** Qumranet, la empresa detrás de KVM, es comprada por Red Hat. Innotek, la empresa detrás de Virtual Box, es comprada por Sun Microsystems. Vmware decide convertir Vmware ESXi en freeware. Microsoft lanza la versión final de Hyper-V.
- **2010:** Virtual Box pasa a llamarse Oracle VM VirtualBox.
- **2011:** Se empieza a incluir ciertas partes de Xen en la rama oficial del kernel de Linux 2.6.37. Integración completa en la versión 3.0.

## 1.2 Objetivo

El propósito con el cual se realiza este proyecto de tesis es el de contar con un sistema que permita a los usuarios del Laboratorio de Redes y Comunicaciones de la Facultad de Ingeniería Eléctrica de la U.M.S.N.H. conectarse de forma remota a partir de sus propios equipos a los equipos de red que se encuentran en el laboratorio, esto con el fin de que se le permita tener un medio para realizar prácticas con los equipos cuando no lo pueda realizar de manera presencial, así mismo, desarrollar un software de gestión que permita dar acceso a los usuarios cuando estos lo requieran con los requerimientos de seguridad necesarios.

## 1.3 Justificación

El impacto tecnológico que se ha dado debido a la evolución de las Telecomunicaciones ha generado el interés de estudiantes por adentrarse al mundo de las Redes de computadoras, por tal motivo, se pensó realizar este proyecto de tesis, con el cual, se busca proporcionar una herramienta a los usuarios del Laboratorio de Redes y Comunicaciones con la cual no se vean limitadas las oportunidades para realizar trabajos relacionados a la asignatura de Redes de Computadoras.

El estudiante de las materias de Redes de Computadoras cuenta con programas de simulación de redes con los cuales se crean entornos de trabajo para realizar prácticas de acuerdo a lo que se imparte en la asignatura, pero los resultados que se obtiene con este software nunca se comparan con los que puede proporcionar el trabajar con equipos reales.

Un obstáculo eminente que se tiene para la realización de prácticas en el entorno del laboratorio es el tiempo, es decir, no se puede coincidir con un horario por parte del estudiante con la disponibilidad del laboratorio, así mismo, cuando las instalaciones universitarias por diferentes cuestiones ya sea en fines de semana, en días oficiales de suspensión o cuando termina el horario de trabajo dentro de la institución, se encuentran cerradas impidiendo el acceso de los usuarios. Por ello es necesario contar con un software de gestión, que incluya el sistema de acceso, con el cual se maneje el control para permitir el ingreso en determinado horario a los usuarios que reservaron tiempo para la realización de una práctica.

## 1.4 Metodología

**Tipo de investigación**

El presente trabajo de investigación está basado principalmente en investigaciones teóricas, de campo, y bibliográfica.

**Investigación Teórica:** El trabajo comienza con la recolección de datos, que permite tener un entorno claro sobre los antecedentes que se tienen del trabajo a realizar, para sustentarlo en situaciones ya probadas, calificar y seleccionar herramientas que apoyen a la solución del problema.

**Investigación de campo:** La investigación se desarrolla directamente en el laboratorio de Redes y comunicaciones, probando los dispositivos de red con los que se cuenta, observando los pros y contras en cada etapa de desarrollo.

**Investigación bibliográfica:** Se sustentó la base teórica de la investigación, mediante consultas a fuentes bibliográficas, textos, revistas, apuntes, documentos varios, así como también fuentes informáticas de Internet.

El método de investigación sobre el cual se basa este proyecto es el Método hipotético-deductivo, sobre el cual, como primera instancia se tiene la observación de un problema al que se enfrenta el alumno de Redes de Computadoras, **"no contar con herramientas que sustente el aprendizaje y aproveche en la mayor proporción posible los equipos de laboratorio con el que se cuenta".** Enseguida se planteó una hipótesis a partir de lo anterior, definir explícitamente

el problema que se tiene donde es de claro conocimiento por qué se enfrenta estos problemas donde se busca mejorar la disponibilidad del equipo de Laboratorio de Redes, aplicando herramientas de acceso remoto y virtualización, así como, incrementar la posibilidad de desarrollo de habilidades en los usuarios del laboratorio. Posteriormente, una vez que se da preferencia a una opción de solución se considera las consecuencias, y por último se hace el desarrollo donde se comparan los enunciados deducidos con la experiencia.

## 1.5 Contenido de la tesis

El contenido del proyecto es expuesto en 6 capítulos los cuales se describen a continuación.

En el primer capítulo se define los antecedentes históricos del tema a desarrollar. En este caso se da una semblanza de las primeras herramientas tecnológicas de acceso remoto, su evolución y su aplicación en el tiempo actual, así mismo con el software de virtualización. Se define claramente el objetivo del problema que se quiere solucionar, se explica la meta a la cual se quiere llegar al termino del proyecto. Así mismo se presenta la justificación o motivación que conllevó a elegir el proyecto de tesis, además se da a conocer la metodología de investigación que mejor se adecua a este trabajo. Se describe el método de investigación “Hipotético – Deductivo” por ser el método que más se adecua para la solución del problema.

En el segundo capítulo aborda de forma general el problema. Se dan a conocer de formas más específicas las herramientas de acceso remoto y de virtualización con las que se cuenta

actualmente, que además de estar dentro de las más populares, son las que mejor se adaptan y cumplen las expectativas de solución.

En el tercer capítulo se describe explícitamente la alternativa de solución elegida de acuerdo a las ventajas que ésta ofrece. Se trata de la herramienta KVM, de la cual se explica el funcionamiento y utilización de los protocolos VNC y RDP.

El cuarto capítulo contiene la solución de forma detallada de cómo se llegó al resultado final. Se dan detalles sobre la utilización del sistema, la creación de las máquinas virtuales. Se habla de cómo se relaciona este dicho sistema con el software de administración de usuarios el cual, permite el acceso o denegación del servicio.

En el capítulo quinto se presentan las conclusiones observadas durante el desarrollo, el enfoque que se le dará al sistema, la forma como se aplicará así como, los trabajos futuros que se vislumbran como factibles.

El capítulo 6 se enumeran las fuentes bibliográficas que sustentan la investigación presentada en este proyecto de tesis.

## 1.6 Comentarios finales

Conforme transcurre el tiempo la comunicación entre personas se ha convertido en una necesidad muy grande, ya sea dentro del campo laboral, estudiantil y en la vida cotidiana. En la mayoría de las veces esta comunicación se ve limitada por distancias y tiempos.

La causa principal de desarrollo de este proyecto es la limitante en cuanto a tiempo, de que un estudiante de la Facultad de Ingeniería Eléctrica de la UMSNH presenta cuando requiere utilizar los equipos del laboratorio de redes para llevar a cabo prácticas de CCNA, debido a que el laboratorio no está disponible a cualquier hora o día. Es por eso que se pensó en este proyecto de tesis cuyo objetivo principal es que al alumno se le permita el acceso a los equipos del laboratorio desde su hogar y pueda realizar sus prácticas.

# *Capítulo 2. Virtualización*

En este capítulo se dan detalles sobre lo que trata la tecnología de virtualización y como facilitan la comunicación a distancia. Se explica detalladamente el concepto de virtualización y se da a conocer la importancia que este tiene para que se relacionen la máquina virtual con la maquina real. Se explican los diferentes tipos de virtualización disponibles, para diferentes necesidades del usuario: Virtualización de plataforma, virtualización de recursos, virtualización de aplicaciones y virtualización a nivel de escritorio.

## 2.1 Introducción

El hardware actual de las computadoras x86 fue diseñado para ejecutar un solo sistema operativo y una sola aplicación, lo que significa que la mayoría de las máquinas estén subutilizadas.

La virtualización permite ejecutar varias máquinas virtuales en una sola máquina física, y compartir los recursos de esa computadora física entre varios entornos. [Informaticaitc 2015].

La Figura 2 ilustra el escenario de virtualización, donde, teniendo un equipo real se puede tener 2 máquinas virtuales con el uso de una aplicación de virtualización. Sin embargo estos recursos compartidos son limitados, es decir, no se comparten todos los recursos del sistema, esencialmente se comparten solo los más básicos.

Básicamente la idea de virtualización, expresándose de una forma simple, consiste en crear una computadora virtual dentro de una computadora real, de tal forma que dentro de la

computadora virtual sea posible correr un sistema operativo, el cual se dice que es "engañado", debido a que este considera que se está ejecutando dentro de otra computadora real.

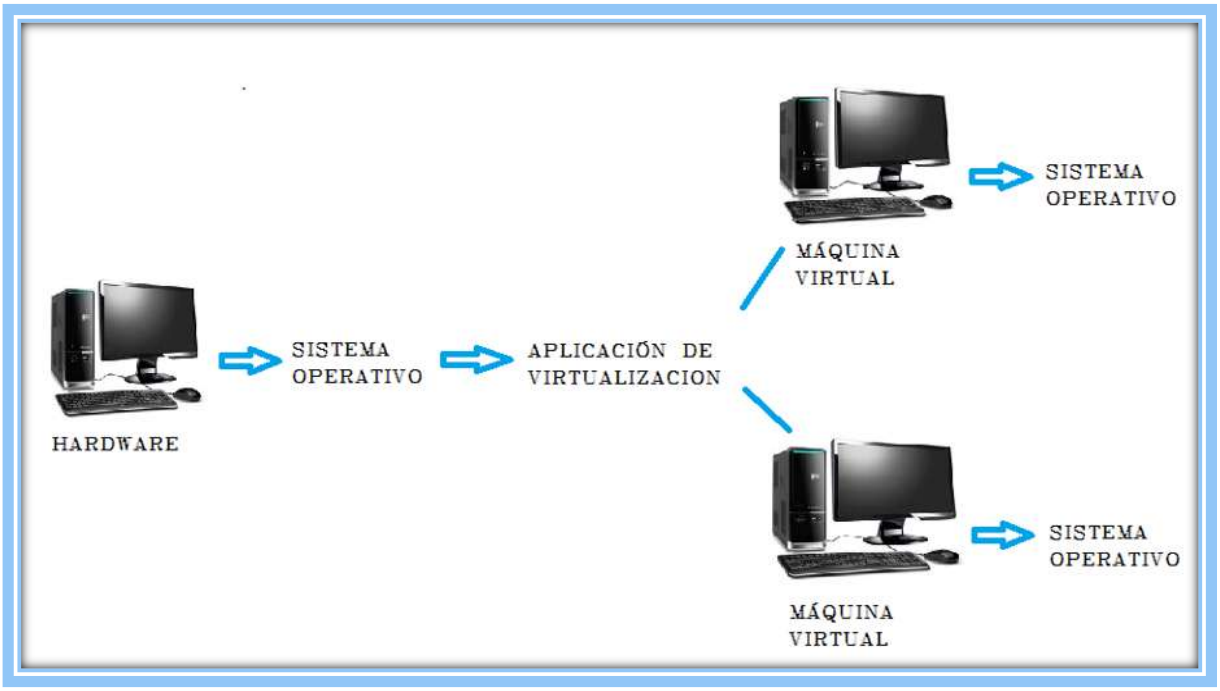


*Figura 2. Escenario de virtualización*

Hablar de virtualización, conlleva a mencionar 2 términos importantes: **sistema anfitrión y sistema invitado.**

Como se puede observar en la Figura 3 un equipo con sistema operativo Linux actúa como el equipo anfitrión alojando la máquina virtual o sistema invitado también con sistema operativo Windows.

El sistema anfitrión es el sistema operativo de la computadora real, en la cual, se instala el software de virtualización que determina la forma de compartir recursos entre esta y la computadora virtual.

El sistema invitado es el sistema operativo que se encuentra instalado dentro de la computadora virtual que ha sido creada dentro del equipo real a partir de la plataforma de

virtualización, el cual recibe los recursos para que funcione.


*Figura. 3. Maquina real ejecutando Linux conteniendo una Máquina virtual que ejecuta Windows.*

Una máquina virtual es un software el cual simula a una computadora real y ejecuta programas como las que ésta puede realizar.

Los procesos que la máquina virtual puede realizar están limitados por los recursos con los cuales esta cuenta.

Un uso común que se hace con una máquina virtual es el de probar un sistema operativo dentro de otro distinto, sin la necesidad de instalarlo en la computadora real.

El uso que se le da a la máquina virtual para este proyecto de tesis es el de simular

dispositivos finales en una topología de red típica, sobre la cual, los usuarios realizarán actividades de configuración de protocolos de redes.

Actualmente existen dos tipos de máquinas virtuales: Máquinas virtuales de sistema y Máquinas virtuales de proceso.

- **Máquinas virtuales de sistema:** También llamada Máquina virtual de hardware, permite a la maquina real (con la ayuda de un software hypervisor, el cual se describe posteriormente), crear dentro de ella varias máquinas virtuales incluso con diferentes sistemas operativos (por ejemplo Linux y Windows).
- **Máquina virtual de proceso:** También conocida como Máquina virtual de aplicación, es ejecutada como un proceso normal dentro de otro proceso normal y único dentro del sistema operativo, es iniciada cuando se manda llamar el proceso y finalizada cuando este es interrumpido, permite su ejecución sin importar la plataforma de hardware y el sistema operativo. La Máquina virtual de Java es un ejemplo de este tipo.

El concepto de virtualización de hardware ha sido ya utilizado desde la década de los 60´s, pero ahora con los nuevos procesadores que han sido creados tanto por AMD (AMD Virtualization) como por Intel cuya extensión de virtualización es llamada IVT (Intel Virtualization Technology) que está disponible en los procesadores de última generación y en algunos procesadores anteriores, los cuales con el desarrollo de plataformas de virtualización la gran parte de ellas de software libre, han hecho que la virtualización retome importancia en la era actual.

Para la construcción de una máquina virtual (como se ha venido mencionando), es necesario compartirle memoria RAM, procesadores y espacio en disco duro, estos proporcionados por el equipo anfitrión, después de esto será obligatorio instalarle un sistema operativo.

Hoy en día virtualizar es una buena opción ya que los equipos con los que actualmente se cuenta, ya sean servidores o PC portátiles o de escritorio, no se utiliza al máximo su potencial en procesamiento de datos ni la capacidad del disco ni de memoria. Además de ésta existen otras tantas ventajas las cuales llevan a pensar en los beneficios que se pueden alcanzar [Capacity 2014], entre las más destacadas son:

- **Reducción de costes:** Se puede reducir en gran medida el gasto económico en Hardware, esto debido que se cuenta con poco equipo el cual comparte sus diferentes recursos entre las máquinas virtuales, así también, no es requerido el mantenimiento y se reduce el consumo eléctrico.
- **Disponibilidad:** Las plataformas de virtualización tienen la posibilidad de configurar opciones de alta disponibilidad para recuperación de fallos, esto es, configurar los sistemas para en el caso de que una máquina virtual falle, de inmediato se cree una máquina virtual nueva en el mismo estado, esto sin que el usuario se entere.
- **Flexibilidad:** Se permite crear tantas máquinas virtuales como el hardware del equipo real lo permita, proveyendo de características de almacenamiento, de memoria y procesamiento. Así mismo, permite que estos recursos puedan ser modificados cuando se requiera, es decir, se puede pedir más espacio en memoria RAM.

- **Migración de equipo:** Si se cuenta con varios ordenadores los cuales alojan en ellos máquinas virtuales es posible moverlas entre los distintos equipos de cómputo, esta característica es importante cuando se requiere una actividad de mantenimiento en alguno de los equipos.
- **Independencia del hardware:** Las máquinas virtuales únicamente requieren de un entorno en el cual puedan ejecutarse, dicho entorno no depende de la marca y el hardware del ordenador real, esto es, se puede tener una máquina virtual dentro de un equipo con arquitectura IBM, se podría mover a uno con hardware HP, sin que la máquina requiera una configuración.
- **Menor consumo energético-Menor disipación de calor:** Al tener un número bajo de equipo funcionando es menor el consumo de energía lo cual también disminuye la generación de calor, manteniendo una temperatura controlada.
- **Aislamiento:** Las máquinas virtuales son independientes entre sí y el hypervisor; el cual es el encargado de manejar los recursos del sistema real exportándolos a las máquinas virtuales. Esta independencia evita que cualquier fallo en una máquina virtual afecte a otras.
- **Seguridad:** Para tener acceso a una determinada máquina virtual es necesario tener permisos de administrador, por lo tanto un ataque de seguridad solo afectaría a esa máquina en específico sin tocar las demás.

Es importante mencionar que así como la virtualización trae consigo varias ventajas mencionadas anteriormente también tiene repercusiones descritas a continuación:

- **Rendimiento inferior:** Un equipo virtualizado no dará los mismos resultados de rendimiento como los que se obtienen al trabajar con equipo real. La ejecución de los programas no tendrán una velocidad de ejecución ideal.
- **Condiciones de hardware:** Debido a que la virtualización es lograda gracias a un sistema anfitrión y a que este comparte sus recursos con los sistemas invitados, el sistema anfitrión requiere ser un equipo potente para poder brindar los servicios a los equipos virtuales, y que no se tenga un rendimiento deficiente por parte de ambos sistemas.
- **Sistema operativo anfitrión:** Si el equipo real sufre un daño irremediable todas las máquinas virtuales contenidas en él también se verán afectadas por este hecho.
- **Desaprovechamiento de recursos:** Se debe tener cuidado en el número de máquinas virtuales que se crearán dentro del sistema, ya que equipos innecesarios están ocupando recursos del sistema anfitrión, principalmente en el procesamiento de datos, la memoria RAM (Random Access Memory – Memoria de acceso aleatorio) así como espacio en disco.

Como se puede apreciar la virtualización provee ventajas que son de importancia para aplicarlas dentro del proyecto, aunque también es pertinente poner atención en los problemas que ésta puede causar cuando se está diseñando un entorno que implique el uso de esta tecnología. Centrando estas ventajas y desventajas dentro del proyecto de esta tesis, se puede determinar que las que benefician a este son principalmente, la flexibilidad y la independencia del hardware, el uso de recursos que utilizarán las máquinas virtuales no serán demasiados, sólo lo indispensable para realizar las configuraciones requeridas tratándose a estas máquinas virtuales como

dispositivo de usuario final, de igual manera para configurar el protocolo TCP/IP.

El aspecto importante a tomar en cuenta es que será necesario tener el debido mantenimiento con el equipo que aloje a las máquinas virtuales, esto para que siempre estén disponibles, y se encuentren en excelente funcionamiento.

Se tiene contemplado únicamente crear las máquinas virtuales necesarias y exactas para que funcionen dentro de la topología que se requiera para cada práctica, evitando con esto tener equipos sin utilizar.

## 2.2 Hypervisor

El Hypervisor, es un software también llamado monitor de máquina virtual, se trata del núcleo central de algunas de las tecnológicas de virtualización.

Los hypervisores son aplicaciones que presentan a los sistemas operativos virtualizados una plataforma operativa virtual, a la vez que se encarga de ocultar a dicho sistema virtual las características físicas reales del equipo sobre el que operan. También son los encargados de monitorizar la ejecución de los sistemas operativos invitados, con ellos es posible conseguir que múltiples sistemas operativos compitan por el acceso simultáneo a los recursos de hardware de una máquina virtual de forma eficaz. [Data Keeper 2015].

Existen tres tipos de hypervisores con características específicas los cuales se describen a continuación:

### 2.2.1. Hypervisor de tipo 1

El Hypervisor de tipo 1 también llamado nativo, unhosted o bare-metal, se ilustra en la

Figura 4, donde se observa que en este tipo el hypervisor se ejecuta directamente sobre el hardware físico.

El VMM (Virtual Machine Monitor – Manejador de máquina virtual) el cual es el nombre con el cual también se le nombra al hypervisor, se encarga de los sistemas operativos invitados y todos los accesos directos a hardware son controlados por él.

Los programas diseñados para el hypervisor tipo 1 son Microsoft Hyper-V, Citrix Xen Server y VMWare ESX-Server.

Este tipo de hypervisor se clasifica en 2 tipos:

- ***Monolíticos***

  Emulan hardware para las máquinas virtuales, las cuales cuando realizan una petición a dicho hardware emulado, este hardware emulado la intercepta y el hypervisor redirige estas peticiones hacia los drives de dispositivo que opera dentro de este, para finalmente enrutarse hacia al dispositivo real.

- ***Microkernel***

  El hypervisor es una capa de software sencilla cuya funcionalidad radica en particionar el sistema físico entre los diversos sistemas virtualizados evitando la necesidad de drivers para acceder al sistema.

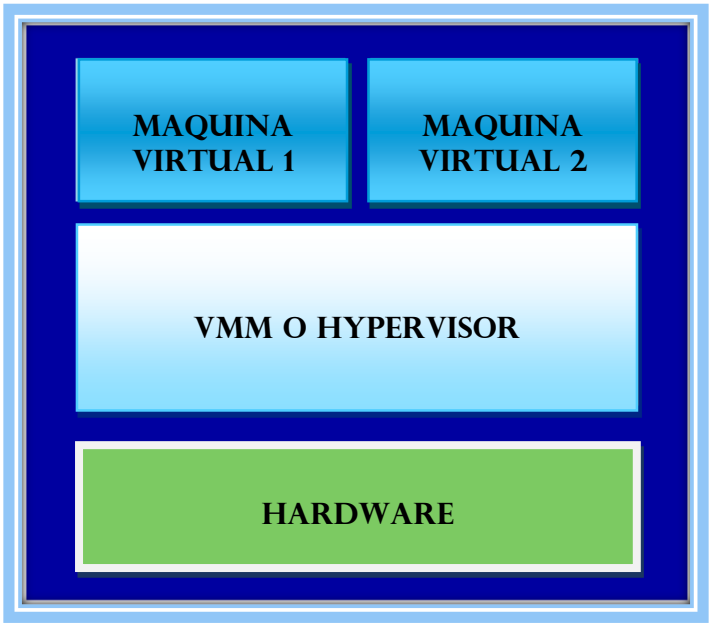


*Figura 4. Hypervisor de tipo 1*

### 2.2.2. Hypervisor de tipo 2

El hypervisor de tipo 2 también conocido como hosted, se ejecuta en el contexto de un sistema operativo completo, que se carga antes que el hypervisor, las máquinas virtuales se ejecutan en un tercer nivel, por encima del hypervisor como se puede apreciar en la figura 5. Ejemplos de este, se tienen en: Virtual PC, Virtual Server, VMWare Workstation, KVM.

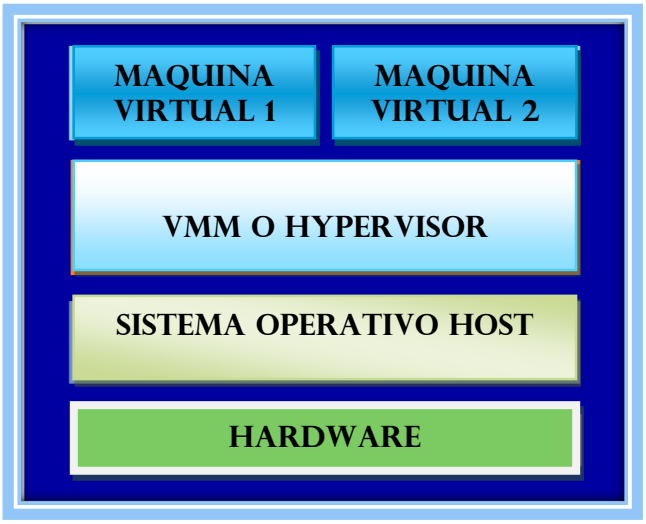


*Figura 5. Hypervisor de tipo 2.*

### 2.2.3. Hypervisor híbrido

El hypervisor híbrido y el sistema operativo anfitrión interactúan directamente con el hardware físico, las maquinas son ejecutadas en tercer nivel con respecto al hardware, por encima del hypervisor, aunque también interactúan con el sistema operativo anfitrión así como se ilustra en la Figura 6, de este tipo se puede mencionar Parallels y VirtualBox. [Geeks 2007].

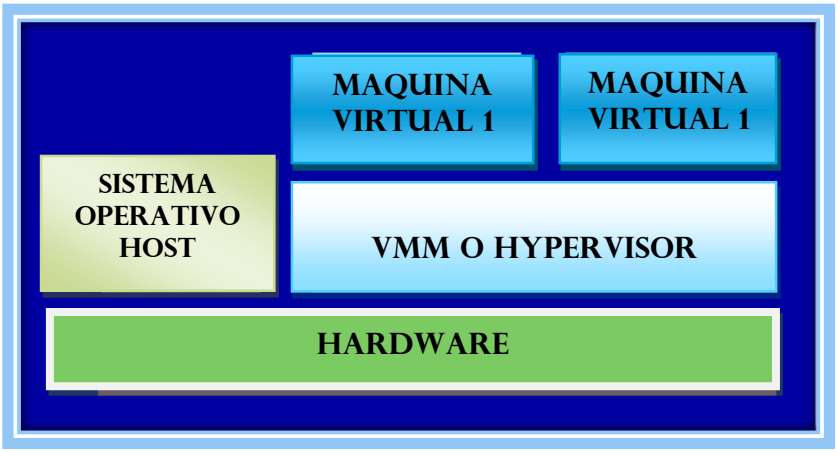


*Figura 6. Hypervisor Híbrido*

## 2.3 Tipos de virtualización

Existen cuatro tipos de virtualización, cada uno con características específicas. Estos tipos son: Virtualización de plataforma, virtualización de recursos, virtualización de aplicaciones y virtualización de escritorio, las cuales son descritas a continuación. [Wikispaces 2016].

### 2.3.1 Virtualización de plataforma

La virtualización de plataforma, como se observa en la Figura 7, se basa en la abstracción de recursos de una máquina real para que esta los comparta con máquinas virtuales, esto es gestionado y asignado a partir de un software llamado hypervisor el cual es una capa que divide el entorno real con el virtual.

El sistema operativo de la máquina virtual creada dentro del sistema anfitrión asume que está siendo ejecutada a partir de un hardware real, de igual forma las distintas máquinas virtuales que pueden ser creadas no pueden interactuar una con la otra debido a la ventaja de asilamiento que se mencionó anteriormente.

La virtualización de plataforma está clasificada a su vez en los siguientes tipos: sistema operativo invitado, emulación, virtualización completa, paravirtualización, virtualización a nivel de sistema operativo y virtualización a nivel de kernel.

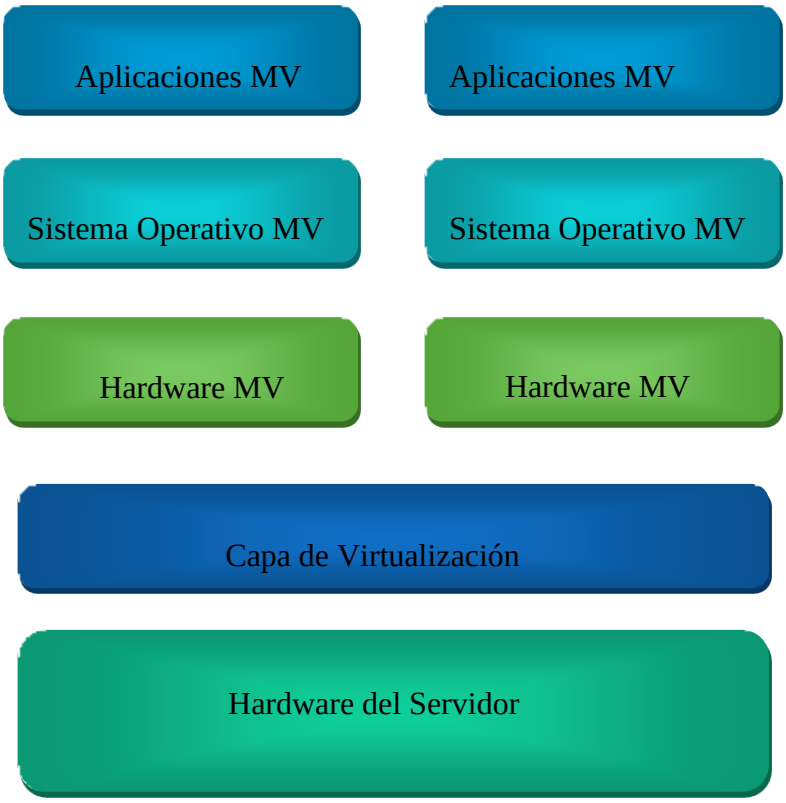


*Figura 7. Diagrama de arquitectura de la virtualización de plataforma*

#### 2.3.1.1 Sistema operativo invitado

La virtualización de sistema operativo invitado necesita contar con el software hypervisor, el cual como ya se ha mencionado es el encargado de crear, configurar, gestionar, administrar y mantener las máquinas virtuales, las cuales son ejecutadas dentro del sistema operativo de un equipo real. Este software también es el encargado de controlar la forma de compartición de recursos, esta estructura es mostrada en la Figura 8.

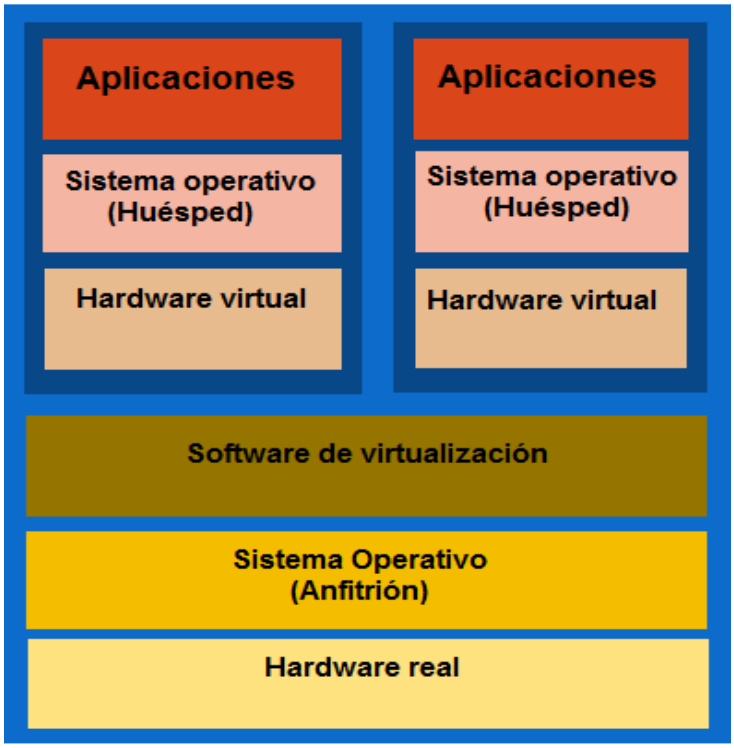


*Figura 8. Entorno de virtualización con sistema operativo invitado*

A continuación se describen herramientas que manejan este tipo de virtualización:

### *VirtualBox*

VirtualBox es un software libre dentro de la versión Open Source Edition, ya que también cuenta con una edición comercial, desarrollado por Innotek, Sun Microsystems y Linux sobre el cual se pueden ejecutar máquinas virtuales de 32 y 64 bits en Linux, Windows, Solaris, BSD, o IBM OS/2 en hosts Microsoft Windows, Mac OS, Linux y OpenSolaris.

VirtualBox es una solución de virtualización bastante completa y actual, al tratarse de una herramienta altamente usada, ya que trabaja del lado del usuario. Soporta tanto tecnología Intel VT como AMD-V. Proporciona tanto interfaces gráficas como línea de comandos para crear y

trabajar las máquinas virtuales. Dispone de herramientas para la creación de nuevas interfaces. Permite que las diferentes máquinas virtuales puedan compartir carpetas, es posible la toma instantánea de ciertos estados del sistema, logrando regresar a anteriores si se requiere. Es posible configurar diversas formas de conexiones de red entre el host y las máquinas virtuales.

### *Parrallels Desktop*

Parallels Desktop es un software para virtualización de hardware para Mac con procesadores Intel, permite trabajar con los sistemas operativos Linux y Windows. Este funciona mediante el mapeo de recursos de la maquina real directamente a los recursos del hardware de la máquina virtual.

Parallels Desktop es un software flexible, que permite acceder a la consola gráfica de la máquina virtual de forma remota, permite crear infraestructuras de múltiples sistemas operativos de cualquier complejidad sin costos adicionales de hardware. Ejecuta sistemas operativos de antiguas generaciones junto con equipo de fabricación reciente. Permite el desarrollo y evaluación de aplicaciones multicapa y configuración de red sin poner en riesgo lo ya existente. No es necesario reiniciar el ordenador o realizar algún tipo de particionamiento y permite la compartición de carpetas.

### *Vmware Player*

Vmware Player es una herramienta de virtualización libre que permite correr máquinas virtuales sobre las plataformas de Windows o Linux. Provee una interfaz de usuario de fácil

entendimiento, es posible tener aislamiento entre máquinas, es posible acceder a los dispositivos del equipo real, permite la modificación de memoria, cuenta con capacidades de funcionamiento de red, cuenta con soporte para SO de 64 bits, detecta inmediatamente el hardware del ordenador real.

### *Microsoft Virtual PC*

Microsoft Virtual PC es un software gestor de virtualización desarrollado por Connectix y comprado por Microsoft para la creación de dispositivos virtuales de uso libre, este no emula el procesador sino que deja que el mismo ejecute las instrucciones en el entorno virtual. La instalación de esta herramienta es fácil de realizar, los usuarios pueden tener acceso a los dispositivos USB que dependen del host, incluyendo dispositivos como impresoras, tarjetas de memoria, discos, entre otros. Permite el manejo de redes.

### *Vmware Workstation*

Vmware Workstation fue desarrollado y vendido por Vmware, Inc., se trata de un hypervisor que se ejecuta dentro de los ordenadores de 64 bits y que permite crear múltiples máquinas virtuales, a cada una de ellas se le permite poder ejecutar cualquier tipo de sistema operativo. Tiene soporte para la construcción de redes, permite compartir unidades de disco y puertos USB (Bus Universal en Serie-Universal Serie Bus).

### 2.3.1.2. Emulador

Para el tipo de virtualización Emulador, la máquina virtual creada dentro del sistema anfitrión simula por completo el hardware que necesita ésta para funcionar ya sea el procesador, memoria, discos, tal como se ilustra en la Figura 9. Permite la creación de cualquier tipo de máquina siendo ésta diferente con la que se está trabajando, ya sean esto máquinas antiguas o equipos que están por salir al mercado.

La virtualización Emulador es la más costosa y la menos eficiente ya que obliga a emular completamente el hardware a simular. Implicando con esto que cada instrucción que sea ejecutada dentro de la máquina virtual, sea traducida al equipo real haciendo con esto que la ejecución sea lenta. Ejemplos de este se tiene: Bochs, Qemu y Hercules.

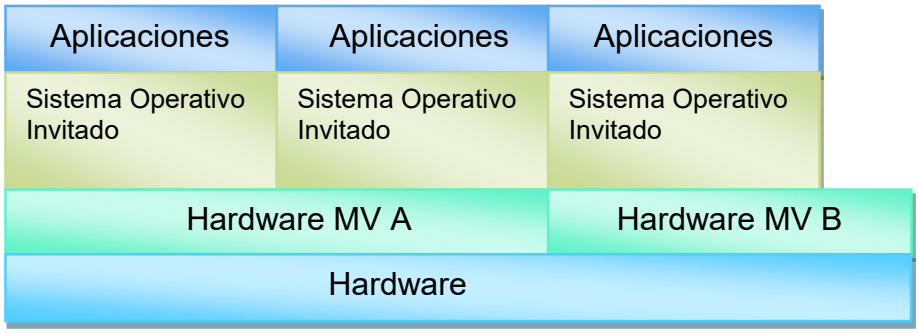


*Figura 9. Arquitectura de Emulador*

***Bochs***

Bochs, se trata de un emulador de código abierto escrito en C++ por Kevin Lawton y distribuido bajo la licencia GPL (General Public License- Licencia General Pública), es posible

ejecutarse en múltiples plataformas, permite emular una computadora con arquitectura Intel x86, dispositivos de E/S así como el BIOS. En la actualidad puede emular arquitectura 386, 486, Pentium en sus primeras 4 versiones hasta computadoras con x86-64. Es posible ejecutar diferentes sistemas operativos, la interfaz de usuario no es fácil de usar.

### *Hercules*

El emulador Hercules, permite que el software diseñado para las computadoras mainframe de IBM puedan ser ejecutadas dentro de cualquier otra arquitectura de hardware, permite al usuario el ejecutar el software dentro de sus propios equipos, sean estos de cualquier sistema operativo.

### *Qemu*

En apartados posteriores se hablará del emulador Qemu, que también participa dentro de la virtualización completa por ahora se explicarán únicamente las características que este tiene como emulador.

Qemu fue escrito por Fabrice Bellard como software libre disponible bajo la licencia GPL. Simula unidades centrales de proceso por medio de traducción binaria ejecutando diferentes modelos de dispositivos, permitiéndole con eso que se pueda ejecutar en diferentes sistemas operativos sin que estos sean modificados. Permite guardar y restaurar estados de la máquina virtual con todos los programas en ejecución, soporta emulación de diferentes arquitecturas, así mismo permite la interacción del hardware del equipo físico, tales como unidades de CD-ROM,

discos duros, tarjetas de red y dispositivos USB. Permite emular tarjetas de red que comparten la conectividad del sistema principal al hacer la traducción de direcciones de red permitiendo a la máquina virtual estar dentro de la misma red que la maquina real, contiene servicios que permiten la comunicación entre sistemas reales y sistemas virtualizados.

### 2.3.1.3. Virtualización completa

El tipo de virtualización completa, hace uso del hypervisor o monitor de máquina virtual, como se describe en la Figura 10, se encuentra como intermediario entre el sistema invitado y el sistema anfitrión, este hypervisor se encarga de emular un sistema operativo.

La ventaja principal de la virtualización completa es que los sistemas operativos del equipo anfitrión no requieren ser modificados, el único requisito es que este sistema operativo soporte virtualización.

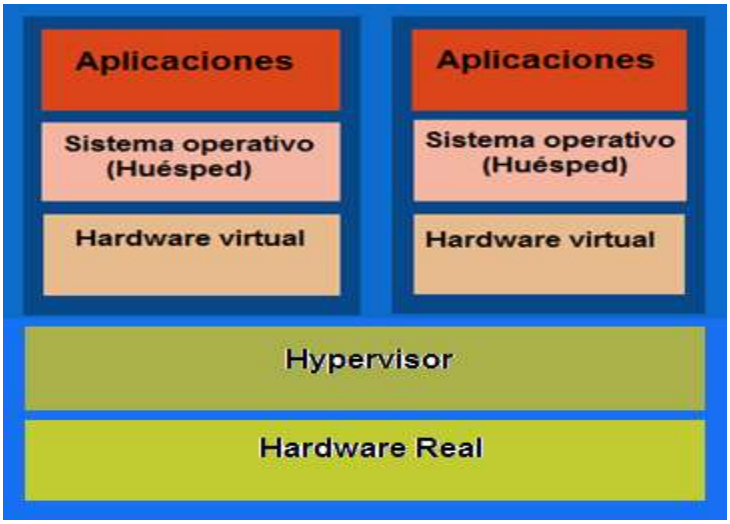


*Figura 10. Virtualización completa*

Algunas herramientas que manejan la virtualización completa es Vmware, ya mencionada

anteriormente. Xen, es otro que también es utilizado para el uso de paravirtualización el cual se abordará posteriormente. Oracle VM, Virtual Server, Hyper-V, son otros ejemplos.

### *Oracle VM*

Oracle VM es un software de código abierto el cual presenta una consola de administración web con interfaces gráficas de fácil uso para la creación y administración de equipo virtual que corren sobre sistemas x86. Cada máquina virtual creada con este hypervisor cuenta con su propio CPU, interfaces de red, almacenamiento y principalmente el sistema operativo, brindando así las herramientas para crear, clonar, compartir, configurar y migrar máquinas virtuales. Los principales aspectos que lo caracterizan son:

- Bajo uso de hardware, de energía y de espacio
- Desempeño eficiente
- Instalación rápida y fácil
- Soporte para Linux y Windows

### *Virtual Server*

Virtual Server es una aplicación que hace posible la creación de máquinas virtuales dentro de los equipos Windows XP y Windows Server 2003. Desarrollado por Connectix y transferido a Microsoft. Las máquinas virtuales son creadas y gestionadas a partir de la interfaz web de IIS. Se instala en un servidor dedicado a ello teniendo las máquinas virtuales disponibles desde cualquier PC, esto es, se pueda acceder a ellas desde cualquier computadora real por vía internet.

***Hyper-V***

Hyper-V es una herramienta de virtualización la cual incorpora una arquitectura de hypervisor basada en un microkernel. Contiene un conjunto de herramientas de gestión con los cuales los usuarios pueden disponer de diferentes herramientas que le permiten la gestión de los recursos virtuales así como de los físicos.

Hyper-V permite el traspaso de máquinas virtuales en ejecución de un anfitrión a otro con mínimas perdidas del servicio. Es posible la creación de un cluster para organizar todas las máquinas virtuales ejecutadas dentro del host. Facilita la importación o exportación de la máquina virtual con el fin de hacer una copia de seguridad y permitir clonar las características de una determinada máquina virtual en otra distinta.

### 2.3.1.4. Paravirtualización

La paravirtualización surgió con la necesidad de mejorar la eficiencia de las máquinas virtuales y garantizar el rendimiento de las mismas, basándose en que dichos sistemas huésped deben estar basados en sistemas operativos especialmente modificados para ejecutarse sobre un hypervisor, logrando con esto que este no realice todas las instrucciones sino que los sistemas huésped y anfitrión colaboran con él, como lo muestra la Figura 11.

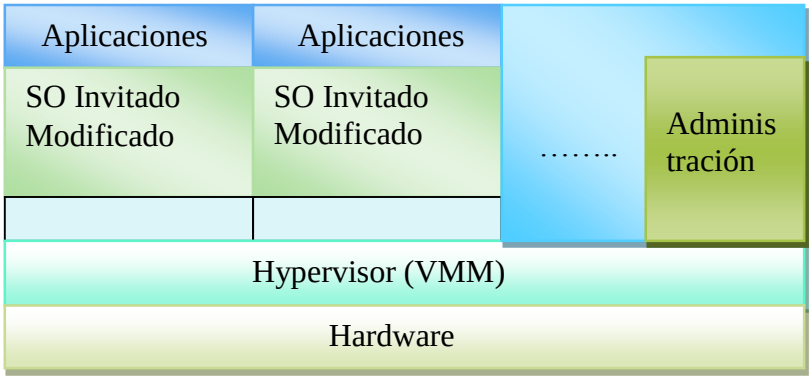


*Figura 11. Arquitectura de paravirtualización*

La herramienta más significativa para realizar paravirtualización es Xen, el cual se trata de un software libre desarrollado en la Universidad de Cambridge en 2003, utilizado por diferentes empresas y que se encuentra en constante competencia con KVM por alcanzar un mejor rendimiento.

Xen proporciona una plataforma de virtualización en la que es posible la ejecución en paralelo de varias máquinas virtuales (llamados dominios para la tecnología Xen) en una sola máquina física anfitriona que proporciona los recursos a dichos dominios, el hypervisor es ejecutado directamente sobre el hardware.

Como ya se mencionó anteriormente la naturaleza de la virtualización obliga a que los sistemas operativos huésped sean modificados, solo se pueden ejecutar sistemas modificables como lo son Linux, Minix, NetBSD, OpenSolaris.

Cuando es iniciado Xen primero se arranca el hypervisor que a su vez inicia un dominio llamado `Dom0`, sobre el cual corre el sistema operativo anfitrión, mismo que tiene privilegios para acceder a los recursos del hardware del servidor, a los dominios no privilegiados se les llama dom Us. Cada dominio interactúa con el hypervisor a través de una llamada conocida

como hypercall. A continuación se mencionan algunas características principales de Xen:

- Soporta arquitecturas x86, x86_64 e ia64
- Rendimiento eficiente en los dominios invitados incluso para aplicaciones con grandes cargas para el procesador.
- Fuerte aislamiento entre los dominios invitados, logrando con ello una mejor política de seguridad dentro de su virtualización.
- Permite restaurar dominios.
- Soporte Hardware Virtual Machine para ejecutar instancias de sistemas operativos no modificados en arquitecturas Intel VT y AMD-V.
- Escalable al permitir gran cantidad de dominios invitados.

### 2.3.1.5. Virtualización a nivel del sistema operativo

En la virtualización a nivel de sistema operativo solo se ejecuta el núcleo del anfitrión creando entornos de ejecución que las aplicaciones ven como máquinas virtuales. En este tipo no hace falta emular el hardware a bajo nivel, puesto que el sistema operativo controla los dispositivos físicos, a su vez incluye herramientas para contar con dispositivos virtuales dentro de cada entorno de ejecución.

La idea principal de la virtualización a nivel del sistema operativo es que los programas se ejecutan en un entorno que hace creer a las aplicaciones que se encuentran en un sistema independiente cuando la realidad es que comparten recursos con otras máquinas virtuales, a pesar de que el sistema las organice para evitar que interfieran entre sí.

La virtualización a nivel de sistema operativo es económica, ya que no necesita apoyo del

hardware ni tratar al código a bajo nivel, aun así el inconveniente que tiene es que solo permite ejecutar entornos virtuales para el propio ordenador y sistema operativo contenidos dentro de un solo núcleo, y si se llegase a tener una falla todas las maquinas serían afectadas. Las herramientas de software libre que se mencionan para este tipo de virtualización son Linux-Vserver y OpenVZ.

### *Linux-Vserver*

Linux-Vserver es un sistema de virtualización que se implementa a partir de una serie de parches que se realizan sobre el núcleo de Linux. Incluye apoyo en el núcleo para crear y mantener múltiples entornos de usuario independientes llamados VPS (Virtual Private Server – Servidor Virtual Privado), sin que se tenga una interferencia entre ellos. Para independizar los espacios de usuarios se define el concepto de contexto que es un contenedor de procesos relacionados con un único VPS, cuando se inicia el sistema se define un contexto por defecto que es el que emplean todos los procesos que pertenecen al sistema anfitrión.

Debido a que Linux-Vserver no tiene dependencia con el ordenador anfitrión es posible manejar diferentes arquitecturas de procesadores. El inconveniente que presenta es que no gestiona de forma adecuada el uso compartido de recursos virtuales como las tarjetas de red virtuales, ya que usa los recursos del anfitrión sin aislarlos de los que usa la máquina virtual.

### *OpenVZ*

OpenVZ es una herramienta similar a LinuxVServer con la diferencia de que incluye capacidades y herramientas de administración más sofisticadas que LinuxVServer. Permite la

existencia de múltiples entornos virtuales aislados dentro del mismo núcleo. Proporciona mecanismos para limitar y garantizar la disponibilidad de recursos dentro de la computadora real.

OpenVZ permite llevar a cabo procesos de congelación del estado de los servidores privados virtuales, pudiendo salvar en disco, para después si es requerido restaurar. Los entornos para los cuales es más utilizado es para la consolidación de servidores, escenarios que requieren alto grado de seguridad, hosting, desarrollo de software y prueba.

### 2.3.1.6. Virtualización a nivel de kernel

La virtualización a nivel de kernel no requiere el hypervisor sino que es el kernel de Linux el que ejecuta las máquinas virtuales de igual forma a como lo hacen con otros procesos de usuario. Tanto las librerías como las aplicaciones y sistemas operativos que corren dentro de las máquinas virtuales deben estar compilados para el mismo hardware y conjunto de instrucciones que el kernel del sistema anfitrión que las ejecuta.

Un ejemplo de la virtualización a nivel de kernel es KVM, el cual será la herramienta de solución para el proyecto y sobre el cual, se abordaran sus características en el capítulo 3 sub-tema 3.1, donde se da a conocer porque es la herramienta ideal para este caso.

#### *User-mode Linux*

Este módulo no requiere software administrativo de manera separada para ejecutar o administrar las máquinas virtuales, sino que pueden ser ejecutadas directamente desde la línea de comandos. Permite al sistema operativo Linux ejecutar otros sistemas operativos Linux en el espacio de usuario, como si se tratase de procesos en el sistema operativo anfitrión por lo que

deben estar compilados.

Los principales usos que ha tenido User-mode Linux son para alojar servidores virtuales así como entornos de supervisión de determinados proyectos y procesos.

### 2.3.2. Virtualización de recursos

Para la virtualización de recursos se abstrae un recurso individual de un ordenador, como puede ser la conexión a red, el almacenamiento principal o elementos de entrada y salida. Dentro de este tipo se pueden manejar ejemplos como el uso de memoria virtual, los sistemas RAID (Redundant Array of Independent Disks – Conjunto Redundante de Discos Independientes), LVM (Logical Volume Manager – Administrador de recursos Lógicos), NAS (Network Attached Storage – Almacenamiento conectado en red) o la virtualización de red.

A continuación se citan diferentes modelos de este tipo de virtualización.

- **Encapsulación**

  Se trata de la ocultación de la complejidad y características del recurso creando una interfaz simplificada.

- **Memoria virtual**

  Permite aumentar al sistema que dispone de mayor cantidad de memoria principal y que se compone de segmentos contiguos de esta, en este caso el recurso individual que es abstraído es la memoria y disco. Por ejemplo la swap usada por todos los sistemas operativos Unix, o las técnicas de paginado de memoria usadas en sistemas operativos Microsoft.

- **Virtualización de almacenamiento**

Es la abstracción completa del almacenamiento lógico sobre el físico, independiente de los dispositivos de hardware. Ejemplos de esto se tiene RAID, LVM, SAN (Storage Area Network – Red de Área de Almacenamiento), NAS, NFS (Network File System – Sistema de Archivos de Red), AFS (Andrew File System – Sistema de Archivos Andrew), GFS (Global File System – Sistema Global de Archivos), iSCSI (Internet Small Computer System Interface – Interfaz de Sistema para pequeñas computadoras en Internet), AoE (ATA over Ethernet – Ethernet sobre ATA).

- **Virtualización de red**

  Consiste en la creación de un espacio de direcciones de red virtualizado dentro de otro o entre subredes. Por ejemplo OpenVPN y OpenSwarm, los cuales permiten la creación de VPNs (Virtual Private Network – Red Privada Virtual).

- **Unión de interfaces de red**

  Combinación de varios enlaces de red para ser usados como un único enlace de mayor ancho de banda, el recurso abstraído son los enlaces de red, soluciones para esto se tiene; vHBA (virtual Host Bus Adapter – Adaptador de Bus de Host virtual) y vNIC (virtual Network Interface Card – Tarjeta de Interfaz de Red virtual).

- **Virtualización de entrada y salida**

  Abstracción de los protocolos de capas superiores de las conexiones físicas o del transporte físico como lo son las conexiones de entrada/salida y transporte, ejemplos: Xsigo systems, 3Leaf Systems y Cisco Systems Brocade.

### 2.3.3. Virtualización de aplicaciones

La virtualización de aplicaciones es usada para permitir a las aplicaciones proveer características como portabilidad o compatibilidad. El usuario es capaz de ejecutar en su ordenador una aplicación que realmente no está instalada en su equipo, esta aplicación se descargará bajo demanda desde un servidor en la red que suministrará el paquete que contiene la aplicación y todo el entorno y configuraciones necesarias para su ejecución, la aplicación se ejecutará en el sistema anfitrión en un entorno virtual protegido sin que se modifique nada dentro de él. Las aplicaciones pueden ser ejecutadas en sistemas operativos para los cuales no fueron implementadas.

Un ejemplo de virtualización de aplicaciones es Wine que permite la ejecución de aplicaciones de Windows sobre Linux mediante técnicas de simulación. Otro ejemplo también es la máquina virtual de Java y CLR.

### 2.3.4. Virtualización de escritorio

La virtualización de escritorio consiste en la manipulación de forma remota del escritorio, las aplicaciones, archivos y datos que están dentro de él que se encuentran separados de la máquina física, almacenado en un servidor central remoto en lugar de en el disco duro del ordenador anfitrión.

El escritorio del usuario es encapsulado y entregado creando máquinas virtuales, de esta forma el usuario puede tener acceso de forma remota a su escritorio desde otros dispositivos, en este caso el recurso abstraído es el almacenamiento físico del escritorio del usuario.

## 2.4. Comentarios Finales

En este capítulo se explicó más detalladamente la tecnología de virtualización, los tipos de esta que existen así como las diferentes herramientas usadas para manejar cada tipo de virtualización.

La virtualización de plataforma se explica más detalladamente debido a que se divide en otros tipos de virtualización.

Un aspecto importante del cual también se habló, es sobre el hypervisor el cual es la parte principal que integra las tecnologías de virtualización, enlazando la tecnología real con la tecnología virtual.

Se menciona la herramienta que se usa para este proyecto, misma que en el siguiente capítulo se aborda completamente.

# *Capítulo 3. Virtualización con KVM y Acceso remoto con distintos clientes*

En este capítulo se explicarán más profundamente los aspectos que describen la herramienta de virtualización KVM, también se indica cómo se lleva a cabo el proceso de instalación de esta. Se explica los pasos a seguir para llevar a cabo la virtualización y comunicación entre usuarios que manejen sistema operativo Windows y usuarios de Linux.

## 3.1. Introducción

En el capítulo anterior se describieron algunas de las herramientas de virtualización más populares hoy en día, exponiendo las características que distinguen a cada una, dando a conocer las ventajas y desventajas con las que cuentan, esto con la finalidad de decidir cual herramienta es la apropiada y provee resultados más encaminados a lo que se pretende resolver con este trabajo de tesis.

Tras el análisis de cada herramienta de virtualización, se llega a la conclusión de utilizar la herramienta KVM, la cual no se eligió por tratarse de la mejor en su tipo, sino porque analizando las características con las que cuenta, son viables en cuanto a este proyecto siendo la principal de ellas, los beneficios que recibe de Linux, el cual es el principal sistema operativo con que el alumno de la Facultad de la U.M.S.N.H. cuenta.

Otras de las ventajas que trae consigo KVM son: rendimiento y escalabilidad, seguridad, gestión de memoria y almacenamiento las cuales se explicaran más a detalle en este capítulo, así como el proceso de instalación.

KVM (Máquina virtual basada en el Kernel) es una solución de virtualización completa

para Linux en hardware x86 que contiene extensiones de virtualización (Intel VT o AMD-V). Se compone de un módulo cargable del núcleo, `kvm.ko`, que proporciona la infraestructura básica de virtualización y un módulo específico de procesador, `kvm-intel.ko o kvm-amd.ko.` [IBM 2011].

KVM también requiere un QEMU modificado aunque se está trabajando para conseguir los cambios necesarios de acuerdo a las características de la que demanda la Tecnología de virtualización actuales; las cuales incluyen, mejoras en la interfaz para el usuario.

Con KVM se pueden ejecutar múltiples máquinas virtuales que se ejecutan sin modificar imágenes de Linux o Windows. Cada máquina virtual tiene un hardware virtualizado privado: una tarjeta de red, disco, tarjeta gráfica, etc.

El componente de kernel de KVM está incluido en Linux, a partir del kernel 2.6.20., y se trata de un software de código abierto [KVM 2016].

KVM cuenta con las siguientes características dentro de su módulo:

- Módulo del kernel (`kvm.ko`): Proporciona el Core (microprocesador) de la infraestructura de virtualización.
- Módulo específico para cada procesador como puede ser `kvm-intel.ko` para procesadores Intel o `kvm-amd.ko` para procesadores AMD.
- `Libvirt`: API desarrollada por RedHat. Brinda una capa de abstracción y un protocolo, los cuales permiten al usuario crear y administrar máquinas virtualizadas independiente del virtualizador utilizado y sistema operativo, logrando con esto una virtualización fácil y homogénea.
- `Virt-manager`: Es una GUI (Interfaz gráfica de usuario) para la gestión de

máquinas virtuales a través de `libvirt`. Dirigida principalmente a las máquinas manipuladas por KVM, también trabaja sobre las plataformas Xen y LXC. Presenta una ejecución resumida de los dominios en ejecución así como estadísticas de la utilización de recursos. Asistentes para la creación de nuevos dominios y configuración y ajuste de la asignación de recursos de un dominio y hardware virtual. Contiene un cliente VNC.

- `Virt-install`: es una herramienta que hace uso de la línea de comandos proporcionando una manera fácil de prestación de sistemas operativos en máquinas virtuales.
- `Ubuntu-vm-builder`: Es una herramienta desarrollada por Canonical para crear máquinas virtuales, instalando el paquete `python-vm-builder` que es un script que automatiza la creación de una VM basada en Linux por línea de comandos.
- `Tigervnc /xtightvncviewer`: Es un cliente VNC que permite conectarse a la consola "física" de las máquinas virtuales de forma remota.
- Libmmigration: Permite mover una ejecución de la máquina virtual o aplicación entre diferentes máquinas físicas sin desconectar el cliente o la aplicación.
- OpenNebula: Es una interfaz web para la administración de una infraestructura cloud (computación en la nube) entre hosts KVM.

KVM es un hypervisor de tipo 1 el cual proporciona virtualización completa para el sistema operativo. Como se observa en la Figura 12, este agrega soporte a huéspedes de multiprocesamiento simétrico soportando características importantes como la migración de sistemas operativos.

KVM es implementado, tal como se ha mencionado, como un módulo que es parte del núcleo de Linux, con lo cual se logra que Linux actúe como hypervisor al cargarse únicamente un módulo. Esta tecnología de virtualización se implementa en forma de dos componentes: Modulo cargable de KVM; que al ser instalado en el núcleo Linux, se encarga de gestionar el hardware de virtualización, el otro componente importante es una versión modificada de QEMU el cual es el encargado de proporcionar la emulación de la plataforma de PC, este componente es ejecutado como un proceso de espacio de usuario, que en coordinación con el núcleo, se encargan de gestionar las solicitudes de los sistemas operativos huéspedes.

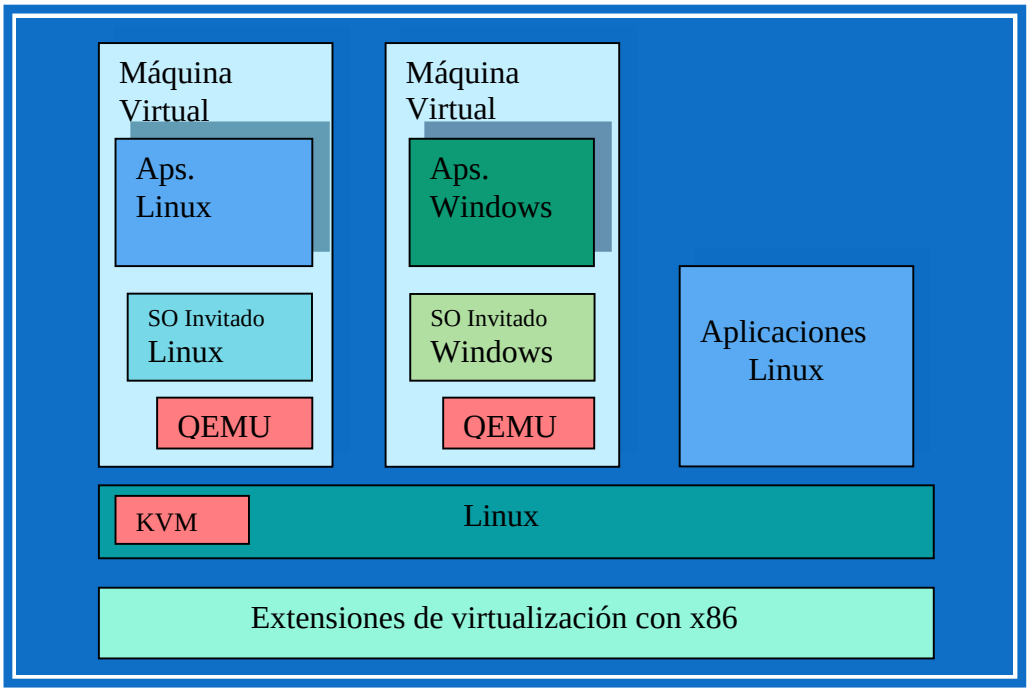


*Figura 12. Arquitectura de hypervisor KVM*

KVM ve a las máquinas virtuales como procesos por lo que cada una puede beneficiarse de todas las características del kernel de Linux, tal como, hardware, seguridad, almacenamiento, etc.

## 3.2 Historia de KVM

KVM se inició como un proyecto de Open Source por la empresa Qumranet, el cual inicialmente fue pensado como una solución de Infraestructura de escritorio virtual para clientes Windows.

Qumranet fue adquirida por Red Hat en 2008, la cual ha buscado ser la mejor opción de virtualización dentro de las opciones de código libre, realizando por ello, el cambio de Xen por KVM dentro de Red Hat Enterprise Linux 6.

Por su parte, IBM ha contribuido en áreas como la gestión de memoria, mejoras en el rendimiento y el subsistema de E/S virtual, ofreciéndolo como solución en el contenido de su software para servidores.

KVM aprovecha todo el entorno Linux conservando el mismo rendimiento y escalabilidad de este, puesto que cualquier mejora o modificación que se realice sobre Linux esta afecta al módulo de virtualización. En contraparte mientras más pesado se vuelva Linux, menos recursos disponibles deja para el resto de las máquinas virtuales [Suite 101 2016].

Los desarrolladores de KVM están enfocados en optimizar la ejecución de procesos que se derivan por la ejecución de las máquinas virtuales, como se mencionó anteriormente, los cambios que sucedan dentro del kernel de Linux también se aplican a KVM, de los cuales se puede mencionar:

- Planificación, control de recursos y gestión de memoria: Las máquinas virtuales son vistas como procesos.
- Almacenamiento: Los discos de las máquinas virtuales son tratados como otro fichero o dispositivo de Linux, pudiendo utilizarse cualquier tipo de almacenamiento

soportado por Linux, tales como: discos locales, sistemas NAS, iSCSI, SAN, entre otros.

- Soporte de hardware: KVM recibe los beneficios de Linux, permitiendo que cualquier dispositivo soportado por Linux también lo sea para KVM, QEMU es el responsable de proporcionar soporte para dispositivos de E/S con las que las máquinas virtuales cuentan.
- Seguridad: Cada máquina virtual es vista como un proceso de Linux, por lo que KVM logra aprovecharse de los protocolos de seguridad proporcionados tanto por SELinux como AppArmor. SELinux es un módulo de seguridad para el kernel Linux que proporciona el mecanismo para soportar políticas de seguridad para el control de acceso, incluyendo controles de acceso obligatorios como los del Departamento de Defensa de Estados Unidos. Se trata de un conjunto de modificaciones del núcleo y herramientas de usuario que pueden ser agregadas a diversas distribuciones Linux. Su arquitectura se enfoca en separar las decisiones de las aplicaciones de seguridad de las políticas de seguridad mismas y racionalizar la cantidad de software encargado de las aplicaciones de seguridad. SELinux ha sido integrado a la rama principal del núcleo Linux desde la versión 2.6, el 8 de agosto de 2003 [NSA 2016]. AppArmor es un programa de seguridad para Linux, lanzado bajo la licencia GPL. Actualmente se encarga de mantenerlo la empresa Novell. AppArmor permite al administrador del sistema asociar a cada programa un perfil de seguridad que restrinja las capacidades de ese programa. Complementa el modelo tradicional de control de acceso discrecional de Unix (DAC) proporcionando el control de acceso obligatorio (MAC). Además de la especificación manual de perfiles, AppArmor incluye un modo de

aprendizaje, en el que las violaciones del perfil son registradas pero no prevenidas. Este registro puede utilizarse para crear un perfil basado en el comportamiento típico del programa [OpenSuse 2016].

Tanto SELinux como AppArmor proporcionan sandboxes (mecanismo para ejecutar programas con seguridad y de manera separada, usado comúnmente para ejecutar código nuevo o de dudosa confiabilidad), logrando con eso que si un proceso se daña en este caso una máquina virtual, esta no afecta el desempeño de las otras ejecutadas dentro del mismo sistema físico. Estas arquitecturas de seguridad mencionadas permiten separar a las máquinas virtuales entre sí, y a su vez, a éstas del hypervisor.

Es de suma importancia considerar que el hardware donde se pretenda realizar virtualización, cuente con las tecnologías Intel VT-x o AMD-V, cada una asociada a su tipo de procesador. Dichas tecnologías ayudan al hypervisor, en caso de que sea necesario realizar virtualización completa; permitiendo la ejecución de sistemas operativos huésped, sin la necesidad de que el sistema operativo físico sufra alguna modificación. Estas tecnologías evitan que el sistema huésped tome control directo de recursos como la memoria y el procesador. Todo esto con el fin de ofrecer una implementación fácil y un rendimiento mayor.

Durante el desarrollo del proyecto se tenía un problema, el cual indicaba que los módulos de KVM no habían sido cargados, llegando a la conclusión de que el soporte de virtualización no está activado. Esto no resolvió el problema ya que el BIOS del equipo de pruebas no contaba con dicha opción por lo cual fue necesario actualizar la versión de BIOS. Este aspecto es de relevante importancia ya que de no ser así KVM no podrá trabajar de manera eficiente, dependiendo del BIOS con el que cuente el sistema.

La Figura 13 muestra en el recuadro rojo, que la tecnología de virtualización esta activada.


*Figura 13. Activación de la tecnología de virtualización desde el BIOS del sistema.*

Como se describió anteriormente, en un inicio KVM se diseñó pensando en la virtualización sobre entornos de Windows, pero en el año 2009 Red Hat firma un acuerdo con Microsoft en el cual se estipula que los sistemas operativos podrán ser soportados por hypervisores ajenos a su propia compañía, como entornos de sistemas invitados.

Para sistemas Windows, KVM funciona sobre las siguientes versiones: Windows Server (2008/2008R2/2003/2000), Windows 8, Windows 7, Windows Vista, Windows XP, Windows NT.

Así mismo, como se han presentado las características acompañando implícitamente a las ventajas que provee el uso de este método de virtualización, se menciona a continuación las desventajas que presenta KVM:

- Proyecto de desarrollo joven, sin demasiada información al respecto.
- No existen herramientas sofisticadas para la gestión de equipos reales así como para la gestión de máquinas virtuales.

- KVM debe mejorar en aspectos como soporte de redes virtuales, soporte de almacenamiento virtual, seguridad, alta disponibilidad, tolerancia a fallos, gestión de energía.

## 3.3 Instalación de KVM en Linux Ubuntu 14.04

Como primer paso para la instalación de KVM es necesario verificar si el procesador de la maquina física a utilizar cuenta con soporte para virtualización, esto probado con el comando que se muestra en la Figura 14 [Ubuntu 2016].

```
# grep -c '(vmx|svm)' /proc/cpuinfo
1
#
```

*Figura 14. Comando para verificar soporte de virtualización en el equipo.*

En el caso de que el resultado arrojado sea 0 indica que no existe soporte para virtualización, por el contrario, si el resultado indica 1 o un número mayor, significa que dicho soporte sí existe, pero el hecho de probarlo de esta forma, no garantiza que la tecnología este activa en el BIOS del sistema.

Es recomendable, aunque no necesario, que el procesador del sistema Linux sobre el cual se está trabajando sea de 64 bits, ya que en un sistema de 32 bits no es posible asignar más de 2GB de memoria a una máquina virtual. Así mismo, dentro de sistemas de 32 bits no es posible ejecutar huéspedes de 64 bits, dicha característica es corroborada con el comando mostrado en la Figura 15.

```
~$ uname -m
x64
~$
```

*Figura 15. Comando para verificar tipo de procesador en el equipo.*

En la máquina sobre la cual se realizó la prueba el resultado arrojado fue x64, lo cual quiere decir que la arquitectura del procesador es de 64 bits, aunque, para efectos de este proyecto no se consideró como relevante este concepto ya que los requerimientos de las máquinas virtuales no excederán 2GB de memoria.

Como siguiente paso se instalan los paquetes básicos desde la terminal de Ubuntu Linux mediante el comando mostrado en la Figura 16.

```
# apt-get install qemu-kvm libvirt-bin bridge-utils
…
#
```

*Figura 16. Comando para instalar paquetes básicos de KVM en el equipo.*

`qemu-kvm`: Instala el sistema KVM dentro del cual también es incluido el modulo del kernel.

`libvirt-bin`: Provee un conjunto de herramientas que mantendrán la interacción con el hypervisor.

`bridge-utils`**:** Contiene herramientas para la configuración de bridges Ethernet.

Es importante tener en cuenta que los usuarios que estarán gestionando las máquinas virtuales tendrán que estar registrados dentro de las librerías `libvirtd` y `kvm`; ya que de no ser así no podrá tener control sobre estas.

Es posible instalar otros paquetes adicionales para proveer más beneficios en la virtualización, esto se realiza tecleando el comando ilustrado en la Figura 17:

```
 # apt-get install virtinst virt-manager ubuntu-vm-builder virt-viewer
……
 #
```

*Figura 17. Comando para instalar paquetes adicionales de virtualización de KVM en el equipo.*

`virtinst`: proporciona herramientas en la línea de comandos para la creación y clonado de máquinas virtuales.

`virt-manager`**:** Interfaz gráfica para la gestión de máquinas virtuales.

`ubuntu-vm-builder`: Scripts para la automatización de la creación de máquinas virtuales basadas en Ubuntu.

`virt-viewer`**:** Permite la conexión a la consola de la máquina virtual a través del protocolo VNC.

Para corroborar si lo que se ha instalado funciona de manera correcta se teclea el comando mostrado en la Figura 18.

```
 # virsh -c qemu:/// system list

 Id Nombre                 Estado
----------------------------------
 #
```

*Figura 18. Comando para verificar correcta instalación de KVM en el equipo.*

Si arroja un error durante la instalación será necesario revisar los permisos en las carpetas `/var/run/libvirt/libvirt-sock` asi como de `/dev/kvm`. Si los permisos son cambiados será necesario reiniciar KVM mediante los siguientes comandos mostrados en la Figura 19:

```
# rmmod kvm_intel
#
# rmmod kvm
#
# modprobe -a kvm_intel
#
# modprobe -a kvm
#
```

*Figura 19. Comando para reiniciar el equipo una vez cambiados los permisos.*

Existen diferentes formas para crear las máquinas virtuales sobre KVM, pudiendo manejarse los comandos `virt-install`, `ubuntu-vm- builder` o `virsh`. Para efecto de este proyecto se optó por usar VMM (Virtual Machine Manager) porque está contenido dentro del paquete `virt-manager` que es una herramienta gráfica para la gestión de máquinas virtuales.

VMM proporciona características significativas como: apoyo al desarrollo para el proyecto de tesis, debido a que permite gestionar de forma completa la utilización de las máquinas virtuales con procesos como encender y apagar, guardar, restaurar y suspender. Proporciona la posibilidad de gestionar pools (Conjunto de recursos inicializados que se mantienen listos para su uso, en lugar de ser asignados y destruidos una vez que ya no son usados) de almacenamiento, gestiona las redes virtuales, permite el acceso a la consola grafica de la máquina virtual y muestra estadísticas de rendimiento de la misma [Virt-manager 2015]. De las características mencionadas anteriormente, las que interesan dentro del desarrollo de la tesis son las que implican la gestión del ciclo de vida de la máquina virtual, como crear la máquina virtual, apagarla, prenderla o guardarla y permitir el acceso a la consola.

A continuación se detalla los pasos a seguir para crear una máquina virtual usando virt-manager.

**Paso 1:**

Se abre el VMM el cual inicializa la pantalla mostrada en la Figura 20. En esta parte se crea la máquina virtual, se le coloca un nombre, además de seleccionar la forma en que se le instalará el sistema operativo, ya sea mediante medios de instalación local, instalación en red o arranque de red.

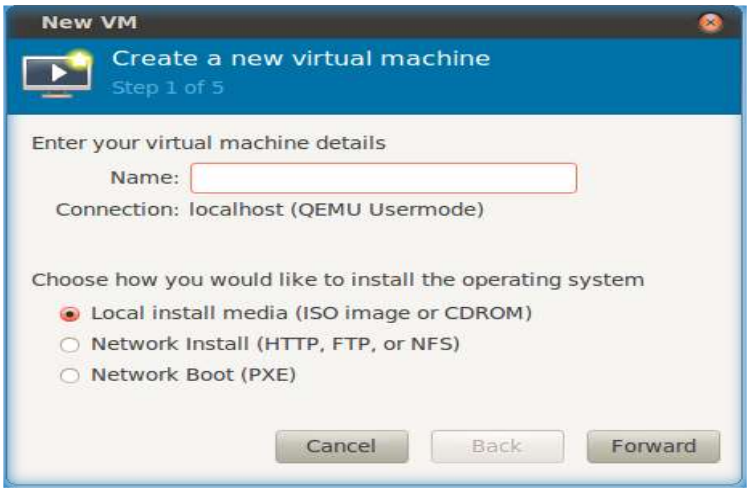


*Figura 20. Nombramiento de máquina virtual*

**Paso 2:**

Al teclear siguiente, la pantalla mostrada es la que se aprecia en la Figura 21, aquí se selecciona el medio de almacenamiento donde se encuentra la imagen ISO del sistema operativo a instalar, así mismo se especifica el tipo y versión del mismo.


*Figura 21. Cargar imagen para el sistema operativo*

**Paso 3:**

Asignar la memoria RAM y el número de CPU's con los que contará la máquina virtual como se muestra en la Figura 22.

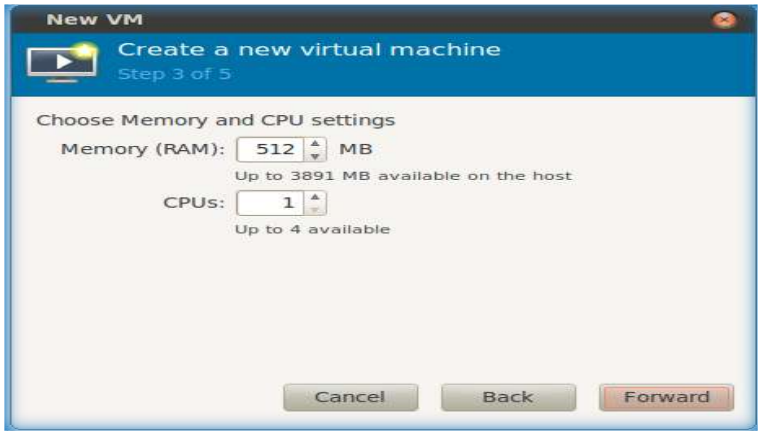


*Figura 22. Asignación de memoria RAM*

**Paso 4:**

En este apartado se permite crear una nueva imagen de almacenamiento o utilizar una existente, se puede configurar el tamaño y asignar el espacio que se desee para el disco duro virtual, como se muestra en la Figura 23, donde se asignó de 8GB.

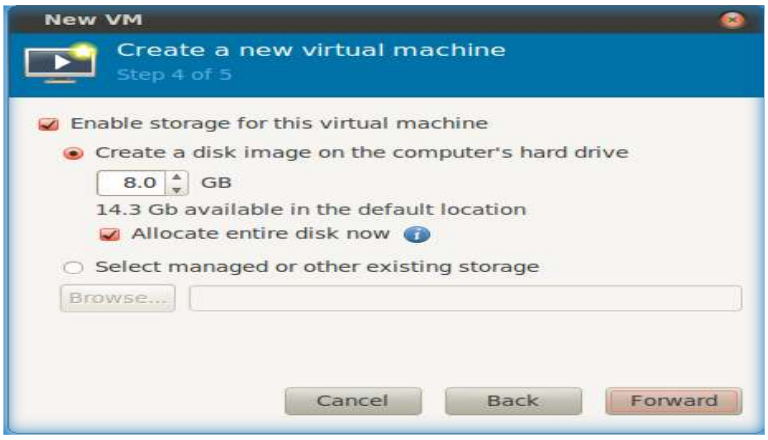


*Figura 23. Nueva imagen de almacenamiento*

**Paso 5:**

La figura 24 muestra el último paso a realizar donde se indica el tipo de virtualización a usar, dando como opciones VMWare y VirtualBox, así como KVM el cual fue el que se eligió como se explicó anteriormente, y el tipo de arquitectura que tendrá el procesador.

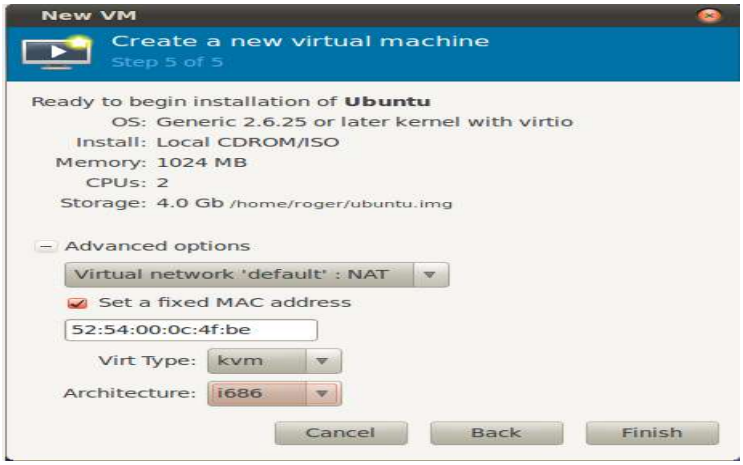


*Figura 24. Indicación del tipo virtualización*

En este punto del proyecto ya se tienen funcionando las máquinas virtuales dentro del equipo físico, ahora el siguiente paso a seguir, es conectar dichas máquinas virtuales dentro de

una red, para lo cual enseguida se citan aspectos importantes que involucra las redes virtuales dentro del hypervisor KVM.

## 3.4. Redes Virtuales en KVM

Para implementar redes virtuales es posible hacer uso de la librería `libvirt`, descrita en puntos anteriores, donde se utiliza el concepto de Switch virtual.

Un Switch virtual es una implementación en software que hace referencia a un Switch real el cual puede conectar máquinas virtuales, el tráfico de red de una máquina virtual es redirigido a través de este Switch [IT From All Angles 2015].

Al instalar y configurar `libvirt`, la red virtual por defecto se denomina default, que a su vez, es el Switch virtual de KVM, siendo este, el soporte para crear redes virtuales, representado por la interfaz `virbr0`.

Para el desarrollo de este proyecto no se utiliza el Swtich de KVM debido a que no permite la creación de VLAN, lo cual es requerido en este caso, tampoco permite el port mirroring ni el Port channel, estos dos últimos no son de mucha importancia para la finalidad de este proyecto, pero aun así se mencionan sus características en este apartado.

La Figura 25, muestra la forma como se relaciona el Servidor, el Switch virtual y este a su vez con las máquinas virtuales. Como se puede apreciar se tiene instalado el Switch virtual dentro del servidor que es el equipo físico donde se alojan las máquinas virtuales.

El Switch virtual cuenta con interfaces las cuales son conectadas a cada máquina virtual que se tenga en uso.

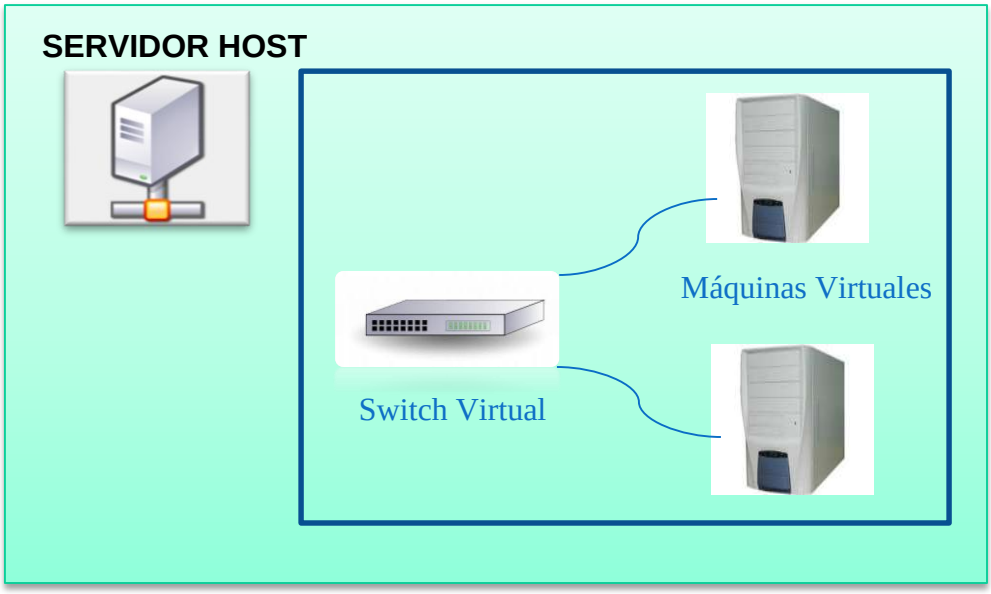


*Figura 25. Relación switch real y switch virtual*

Por defecto, los Switches virtuales operan en modo NAT, es decir, en IP masquerading, el cual se refiere al hecho de que las máquinas virtuales cuando se requiere que se conecten al exterior lo hacen mediante la dirección IP del sistema anfitrión. Así mismo las máquinas del exterior no pueden conectarse con las máquinas virtuales como se muestra en la Figura 26.

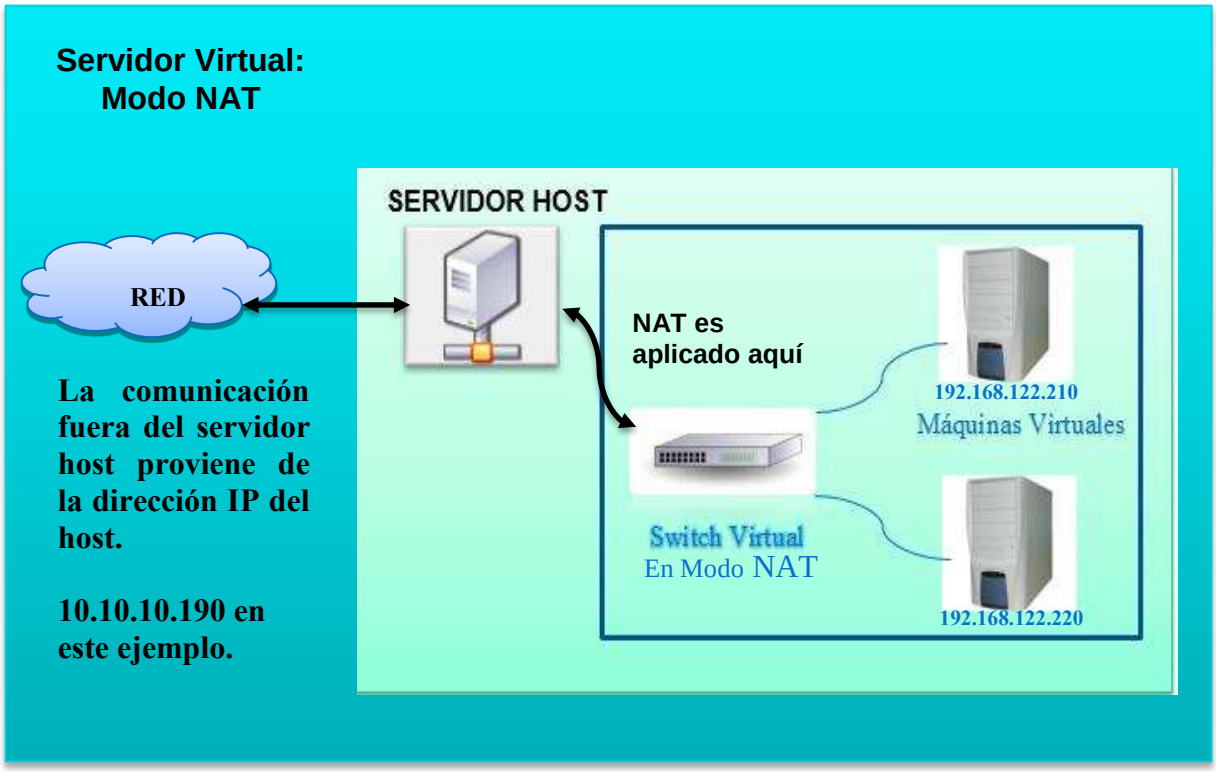


*Figura 26. Operación del switch en modo real*

Para configurar el protocolo TCP/IP dentro de las máquinas virtuales, se permite realizarlo mediante el protocolo DHCP, creando un pool de direcciones en el Switch virtual. En este caso `libvirt` hace uso de un programa `dnsmasq`, el cual es un servidor ligero que proporciona servicios de DNS, DHCP y TFTP para pequeñas redes con lo cual permite a `libvirt` iniciar y configurar de forma automática una instancia de este por cada Switch configurado, su funcionamiento se muestra en la Figura 27.

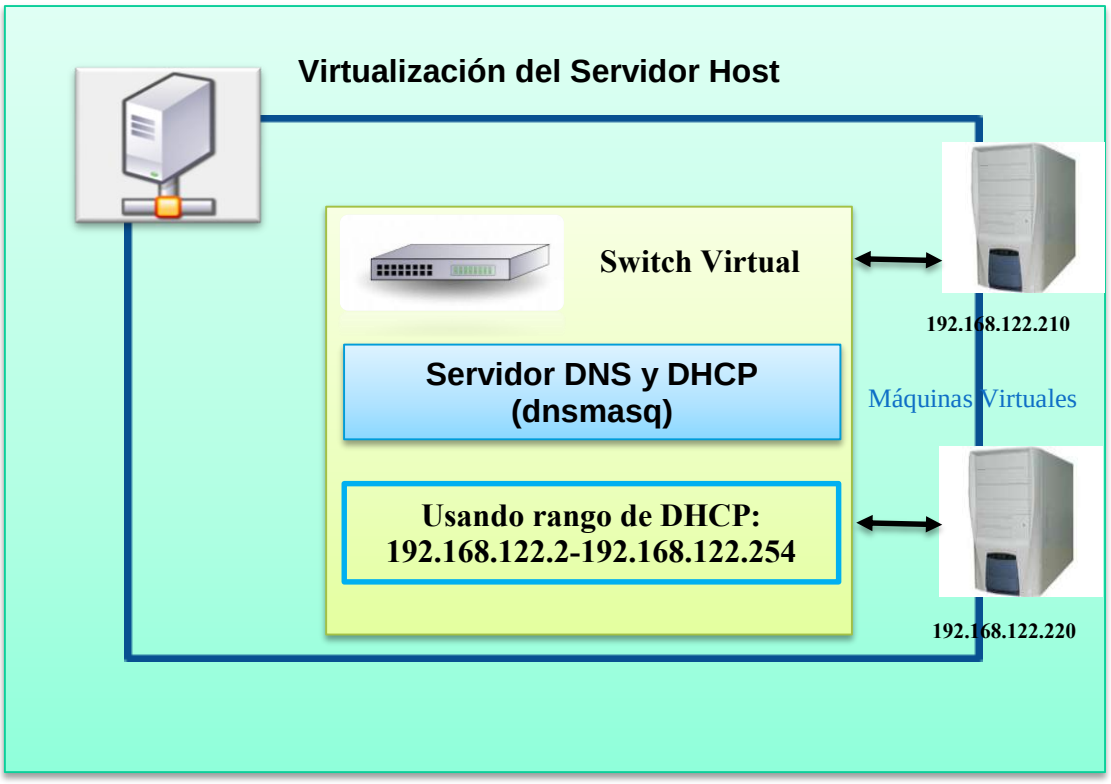


*Figura 27. Switch con virtualización DNS, DHCP y TFTP*

La Figura 28 muestra otro modo por el cual los Switches virtuales pueden trabajar, es en Routed mode, mediante el cual el Switch virtual se conecta a la LAN física del anfitrión pasando el tráfico entre este y las máquinas virtuales sin hacer uso de NAT, ya que será dentro del Switch donde se llevaran a cabo procesos de enrutamiento, por tal motivo en este caso el Switch opera en nivel 3 del modelo OSI, colocando cada máquina virtual dentro de su propia subred, permitiendo a las máquinas externas conectarse a las máquinas virtuales mediante reglas definidas de enrutamiento.

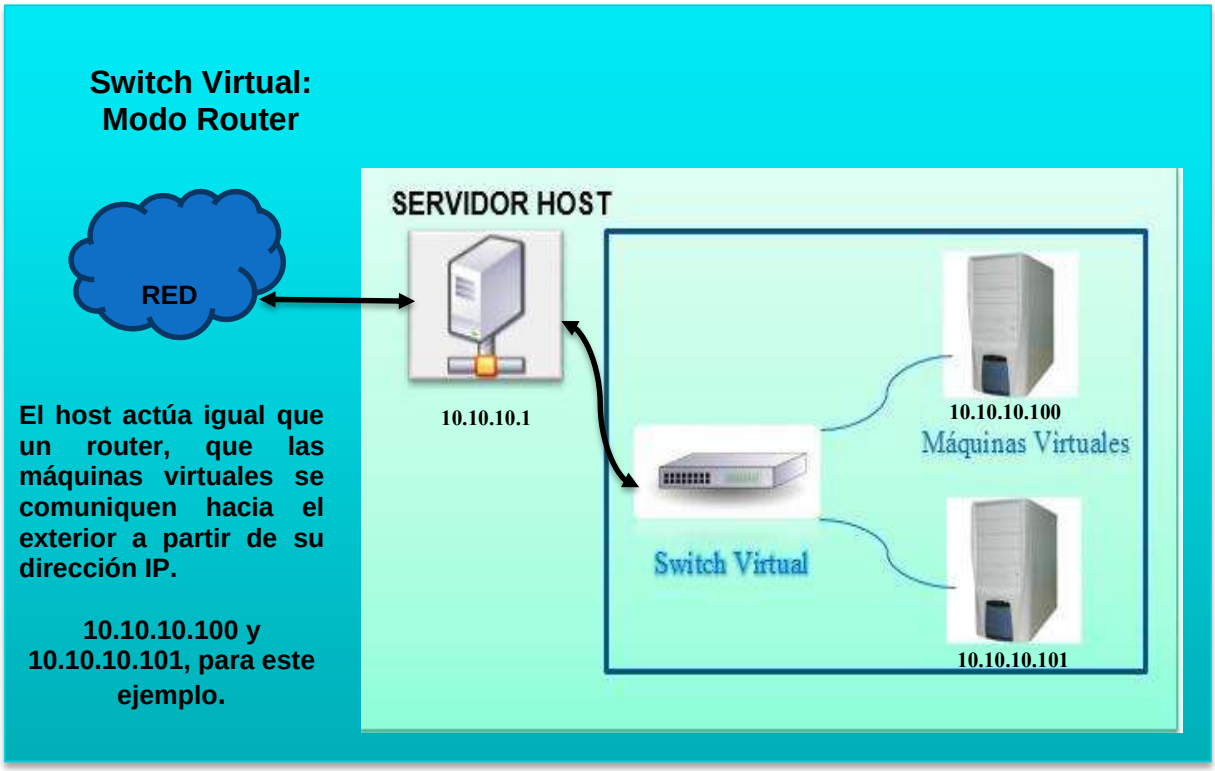


*Figura 28. Switch en modo router*

Los Switches virtuales son implementados dentro de los hypervisores 80x86. Como se ha apreciado, cada máquina virtual tiene al menos una tarjeta de interfaz de red virtual llamadas vNICs, las cuales son compartidas con las tarjetas de interfaz de red virtual de los hosts físicos a través del Switch virtual, sin embargo, a pesar de las importantes características que presenta dentro del hypervisor KVM, no son las que se requieren para la realización del proyecto debido a los siguientes aspectos negativos:

- No permiten la creación de VLAN las cuales son necesarias para el desarrollo de este proyecto y en el transcurso de este capítulo se abordara el tema.

- No es posible el port mirroring el cual es utilizado con un Switch de red para enviar copias de paquetes de red vistos en un puerto de Switch a una conexión de red monitoreada en otro puerto del Switch.
- No existe soporte de Multi-Link Trunking, el cual es una arquitectura para conectar swtiches, dentro de la cual se permite la agrupación de varios enlaces Ethernet para crear un enlace lógico de mayor ancho de banda.

El principal aspecto por el cual no se optó por utilizar el Switch virtual que por defecto tiene KVM, es porque no se tiene la posibilidad de trabajar con VLAN's, lo cual representa un gran problema ya que trabajar este tipo de subredes es una parte fundamental para el proyecto, puesto que cada máquina virtual formará parte de una red independiente.

La solución que se estudió para ser utilizada dentro de este proyecto es la instalación y el manejo de OpenvSwitch el cual es un Switch virtual, pero con características extras, entre ellas el manejo de VLAN.

## 3.5. OpenvSwitch

Como se ha venido describiendo durante el desarrollo, la virtualización es un concepto interesante dentro de las tecnologías de las comunicaciones, debido a que ofrece bastantes beneficios positivos, es por eso que esta tecnología llegó para quedarse principalmente con hypervisores como Vmware, Xen y KVM. [Open vSwitch 2014].

A pesar de las ventajas que rodea a la virtualización, como se mencionó hasta ahora, ningún Switch virtual de alguno de estos hypervisores ha sido capaz de desarrollar escenarios complejos.

Open vSwitch es capaz de soportar grandes flujos de datos, permite implementar VLAN, así como trunking.

Uno de los principales retos de la virtualización es la implementación de una estructura de red completa y funcional.

Open vSwtich es soportado por KVM, Xen, VirtualBox y Vmware. La Figura 29 muestra las empresas que han contribuido en el desarrollo y manteamiento de este.


*Figura 29. Colaboradores de Open vSwitch*

Open Vswitch se basa en el proyecto OpenFlow de la Universidad de Standford el cual es un nuevo estándar abierto diseñado para apoyar la gestión de los switches y routers con software abierto. De las características que Open vSwitch ofrece se citan las siguientes:

- Funcionalidad completa de capa 2.
- Soporte de NetFlow, sFlow, SPAN y RSPAN.
- VLAN 802.1Q con enlaces troncales.
- QoS.
- Agregación de puerto.
- GRE tunneling
- Compatibilidad con el código fuente en Linux.

### 3.5.1. Soporte de NetFlow, sFlow, SPAN, RSPAN

NetFlow es un protocolo de red desarrollado por cisco Systems para recolectar información sobre tráfico IP, se ha convertido en un estándar de la industria para monitorización de tráfico de red.

sFlow es un estándar para el monitoreo de dispositivos de red a través de routers, switches u otros dispositivos relacionados con la interconexión de redes.

Por su parte, SPAN permite tomar el tráfico que atraviesa por un puerto, grupo de puertos o VLAN de un switch copiando el contenido en el puerto destino, que se refiere al puerto donde será instalada la estación de monitoreo [Mis Libros De Networking 2007].

En lo que concierne a RSPAM permite que el tráfico generado en varios swtiches pueda ser reenviado al puerto al que se ha conectado la estación de monitoreo.

### 3.5.2. VLAN 802.1Q con enlaces troncales

Una Virtual Local Área Network (VLAN) se refiere a un conjunto de dispositivos que se encuentran en cualquier ubicación y son agrupados de manera lógica y no física, pero se comunican como si en realidad estuvieran conectadas dentro de un mismo segmento.

La Figura 30 muestra 3 VLAN: la primera corresponden a las maquinas 1 y 5, la segunda a la 2,3 y 8 y la tercera a la 4,6 y 9.

Las ventajas que proporciona trabajar con VLANs son principalmente:

- Mayor flexibilidad y mejor gestión de recursos al facilitar el cambio y mantenimiento de los dispositivos de red.

- Facilidad para localizar y solucionar problemas de forma aislada sin afectar el resto de la red.
- Mejora la calidad de seguridad, debido a la separación de dispositivos en diferentes VLANs.
- Control de tráfico de broadcast

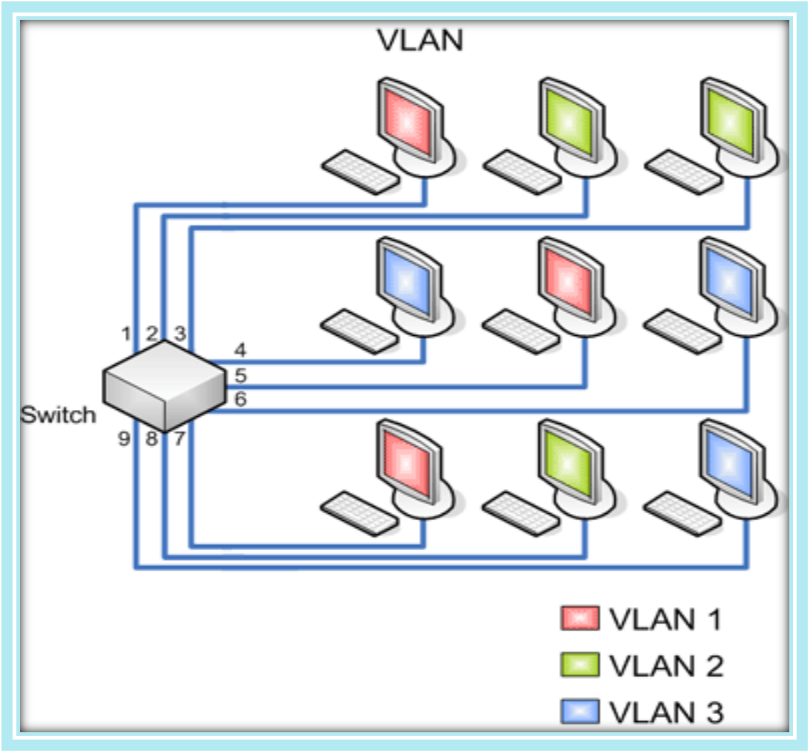


*Figura 30. VLAN*

El protocolo IEEE 802.1Q (IEEE 802.Q,2014) es una modificación del estándar Ethernet, el cual, fue un proyecto del grupo de trabajo 802 de IEEE para desarrollar un mecanismo que permita a múltiples redes interconectadas con puentes o swtiches compartir transparentemente el mismo medio físico sin problemas de interferencia entre las redes que comparten el medio (Trunking). Es también el nombre actual del estándar establecido en este proyecto y se usa para

definir el protocolo de encapsulamiento usado para implementar este mecanismo en redes Ethernet.

El protocolo 802.1Q permite identificar a una trama como proveniente de un equipo conectado a una red determinada. Una trama perteneciente a una VLAN solo se va a distribuir a los equipos que pertenezcan a su misma VLAN, de forma que se separan dominios de broadcast.

El formato de la trama para el protocolo 802.1Q propone añadir 4 bytes al encabezado Ethernet original en lugar de encapsular la trama original, el valor del campo EtherType se cambia a 0x8100 para marcar el cambio en el formato de la trama [802.1Q 2014].

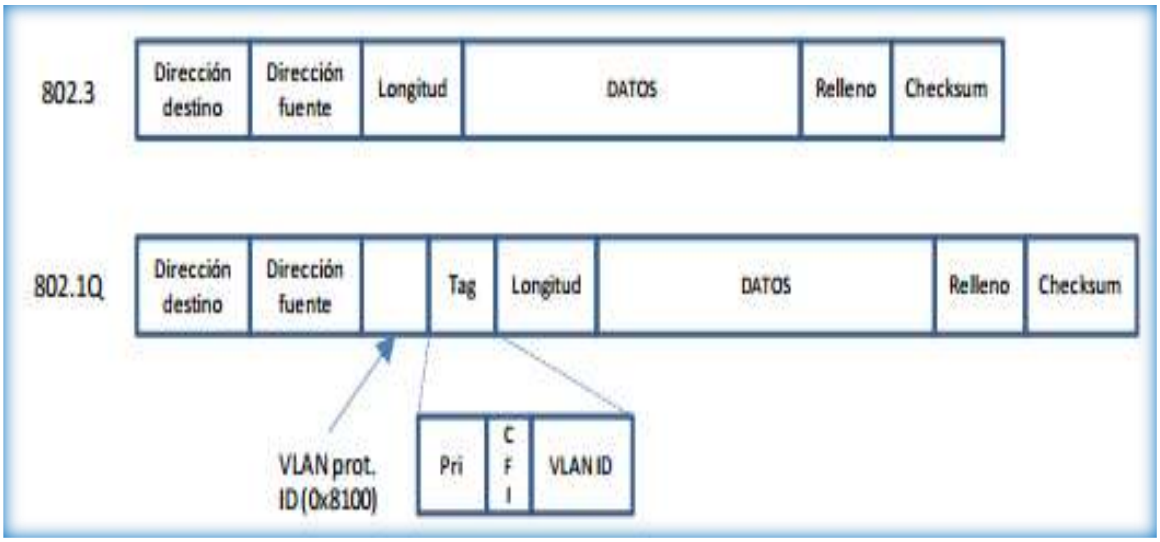


*Figura 31. Trama 802.3 y 802.1Q.*

Como se puede observar en la Figura 31, la VLAN tag se inserta en la trama Ethernet entre el campo de la dirección origen y la longitud. Los primeros 2 bytes de VLAN tag es el Tag Type del 802.1Q y siempre está puesto a 0x8100. Con lo que respecta a los últimos 2 bytes, los 3 primeros bits son el campo Priority Field que son usados para asignar nivel de prioridad a los datos. El siguiente bit corresponde a Canonical Format Indicator usado para indicar la presencia

de un campo Routing Information, y los últimos 12 bits es el identificador de VLAN que identifica de forma única a la VLAN a la cual pertenece la trama Ethernet.

Dentro de los switches se pueden crear dos tipos de puertos: puertos de acceso y puertos troncales. Los primeros son conectados a las estaciones de trabajo, mapean el puerto a una VLAN programada, cuando entra una trama Ethernet se agrega el TAG de 802.1Q y cuando sale una trama del mismo tipo se le quita el TAG para que llegue a la estación con el formato Ethernet original.

### 3.5.3. QoS

Para los administradores de redes un aspecto importante a considerar es el restringir el ancho de banda de huéspedes virtuales de forma independiente, en particular cuando los diferentes clientes hacen uso del mismo entorno virtual.

Open vSwtich permite definir niveles de calidad de servicios como el ancho de banda anteriormente mencionado, la disponibilidad del medio, latencia, ratio de error y priorizar el tráfico de red.

### 3.5.4. Agregado de puertos y GRE tunneling

Al agregado de puertos, los desarrolladores del kernel Linux lo llaman proceso de agrupación, debido a que se le proporciona al administrador la capacidad de combinar múltiples puertos físicos como un único puerto lógico, lo cual sirve para equilibrar la carga y garantizar un alto grado de disponibilidad. Así mismo permite la creación de un túnel GRE (Generic Routing

Encapsulation – Encapsulación de enrutamiento genérico) (protocolo para el establecimiento de túneles a través de Internet), entre múltiples sistemas.

El agregado de puertos emplea protocolos de enrutamiento especializados para que obtengan el camino óptimo entre los extremos de las comunicaciones, permitiendo así, una secuencia en el envío de paquetes así como la creación de túneles sobre redes de alta velocidad, al mismo tiempo que establece políticas de enrutamiento y de seguridad.

Después de analizar los beneficios que aporta el uso de Open vSwtich se puede concluir que se está hablando de un proyecto interesante y potente en su tipo, aunque aún con todo esto, no se cuenta con suficiente información sobre él.

Se espera que las principales distribuciones decidan incorporar Open vSwitch dentro de sus propias herramientas como Libvirt en el caso de KVM, ya que esto eliminaría la necesidad de utilizar el daemon de compatibilidad de puente.

A continuación se proporcionan las acciones necesarias para instalar Open vSwitch dentro del equipo anfitrión.

## 3.6. Instalación de Open vSwitch en Ubuntu 14.04

Se comprueba si existen actualizaciones dentro del sistema, esto garantiza que el sistema este al día y al momento que se hace la instalación de los paquetes necesarios no existan problemas si se llegara a requerir algún componente. Enseguida se hace la instalación de los paquetes necesarios para la instalación de diversos componentes de Open vSwitch [Ovs-discuss 2014]. Esto se realiza mediante los siguientes comandos mostrados en la Figura 32 respectivamente para cada acción mencionada anteriormente.

```
# apt-get update && apt get dist-upgrade
……
# apt-get install openvswtich-controller openvswitch-brcompat
openvswitch-switch openvswitch-datapath-source
……
#
```

*Figura 32.Comandos para instalación de paquetes de componentes de Open vSwitch.*

A continuación se elimina el puente por defecto que trae `libvirt`, el cual es el puente `virbr0`. Esto asegura que la porción de compatibilidad del puente de Open vSwitch pueda cargarse sin que se desate algún problema. Para realizar los pasos anteriores es necesario detener la librería `libvirt-bin` así como también el servicio `quemu-kvm`, para evitar que `libvirt` cargue el puente que trae por defecto, lo cual se hace introduciendo los siguientes comandos:

```
# virsh net-destroy default
La red default ha sido destruida
# virsh net-autostart – disable default
#
# service libvirt-bin stop
  libvirt-bin stop
#
# service qemu-kvm stop
#
```

*Figura 33.Comandos para detener librerías libvirt-bin y quemu-kvm.*

Enseguida, se modifica el archivo `/etc/modprobe.d/bridge.conf`, si no existe dicho archivo este puede ser creado, en él se escribe la línea "`blacklist bridge`" esto indicará al kernel que no vuelva a cargar el módulo bridge que ya se eliminó colocándolo en una lista negra, todo elemento que se encuentra dentro de ella no será cargado. Una vez que se realizó el paso anterior, será necesario activar el modo de compatibilidad del puente para Open vSwitch esto se realiza editando el archivo de configuración que se encuentra dentro del

directorio `/etc/default/openvswitch-swtich` cambiando la línea `#BRCOMPAT=no`, se quita el símbolo de comentario y se activa con la palabra yes, `BRCOMPAT=yes`, hasta este punto ya no se está haciendo uso del puente por defecto que trae consigo KVM. Una vez concluido el proceso anterior, es posible iniciar el Open vSwitch tecleando en la terminal el comando mostrado en la Figura 34.

```
# service openvswtich-swtich start
  unrecognized service
#
```

*Figura 34.Comandos para iniciar OpenvSwitch.*

En el caso del proyecto y en otros casos de estudio que se investigó se observó el problema que una vez realizada esta acción, el sistema indicó que el modulo no está en el formato correcto o no está construido para el kernel utilizado. Para solucionar este problema es necesario instalar los módulos necesarios mediante el comando mostrado en la Figura 35.

```
# module-assistant auto-install openvswitch-datapath
……
#
```

*Figura 35.Comando para instalar módulos faltantes en el kernel de Linux.*

A continuación se reinician los servicios de Open vSwitch con los comandos mostrados en la Figura 36.

```
#service openvswitch-switch restart
#
#service openvswtich-controller restart
#
#reboot
#
```

*Figura 36.Comandos para reiniciar Open vSwitch.*

Después de teclear el comando reboot es probable que los módulos no hayan sido cargados, esto se verificará tecleando el comando mostrado en la Figura 37.

```
# lsmod | grep brcom
brcompat_mod 13512 0
openvswitch_mod 83993 2 brcompat_mod
#
```

*Figura 37.Comandos para verificar la carga de módulos Open vSwitch.*

Para determinar en qué estado se encuentran los componentes de Open vSwitch, se comprueba con el comando mostrado en la Figura 38.

```
# service openvswitch-Switch
ovsdb-server is running with pid 1885
ovs-vswitchd is running with pid 1908
ovs-brcompatd is running with pid 2096
#
```

*Figura 38.Comando para verificar estado de componentes de Open vSwitch.*

Si el resultado no es como las tres últimas líneas mostradas en la Figura 38, significa que los módulos del Switch virtual no están cargados dentro del kernel por lo que es necesario inicializar este servicio mediante el comando mostrado en la Figura 39.

```
#service openvswitch-switch restart
*service openvswitch-switch restart
* Killing ovs-brcompatd (2096)
* Killing ovs-vswitchd (1908)
* Killing ovsdb-server (1885)
* Starting ovsdb-server
* Configuring Open vSwitch system IDs
* Starting ovs-vswitchd
* Starting ovs-brcompatd
* iptables already has a rule for gre, not explicitly enabling
#
```

*Figura 39.Comando para inicializar Open vSwitch.*

Cabe mencionar que teclear el comando anterior no es una solución óptima ya que se tendría que estar realizando esta misma acción cada que se reinicia el sistema, es por eso que dentro del archivo `/etc/init/failsafe.conf` se modificará el tiempo en que se carga Open vSwitch poniéndolo en los últimos eventos de carga de módulos. Estos son los pasos que se siguen para tener funcionando OpenvSwitch dentro del sistema, ahora la acción siguiente es conectar las máquinas virtuales con el Switch virtual, lo cual es una parte medular para seguir adelante con el proyecto.

## 3.7. Creación de VLANs y conexión con el equipo virtual

Para realizar esta parte se contemplaron dos posibles soluciones las cuales a continuación se explicará en qué consisten ambas para después exponer por qué se eligió una de ellas. Cabe mencionar que para realizar esto, se consideraron diferentes aspectos, la potencialidad que ofrece Open vSwitch, así como el que ofreciera eficiencia y factibilidad para realizar con éxito este paso. La primera solución que se investigó fue crear las VLAN con el Open vSwitch y la librería `libvirt`.

En el apartado 3.7, ya se explicó la forma como trabaja la librería `libvirt` para la creación de redes virtuales, entonces, cabe mencionar que se requiere conocer el puerto al cual está conectado un dominio en particular, es decir; la máquina virtual que es el sistema invitado, por ejemplo se puede configurar un puerto el cual será troncal como el comando mostrado en la Figura 40.

```
# ovs-vsctl set port <port name> trunks=10,11,12
#
```

*Figura 40.Comando para configurar puerto troncal en Open vSwitch.*

En la etiqueta `<port name>` se coloca el puerto que conecta a la máquina virtual con el equipo real, un ejemplo es colocar `eth0` el cual es el puerto Ethernet integrado de la máquina real. Con esta configuración las etiquetas VLAN permitirían el paso para las VLAN 10, 11,12, asumiendo con esto que el sistema operativo instalado en el dominio tiene soporte para VLAN. Con el comando similar al anterior si se conoce el puerto al que está conectado un dominio determinado, se puede configurar ese puerto como puerto de acceso de la VLAN, tal como se muestra a continuación en la sentencia mostrada en la Figura 41. Esto hace al dominio un miembro de la VLAN 15.

```
#ovs-vsctl set port <port name> tag=15
#
```

*Figura 41.Comando para configurar puerto de acceso de la VLAN.*

La acción mencionada en la Figura 41 es correcto y funciona de forma adecuada, pero se presenta un primer inconveniente la información de la VLAN no está contenida en la configuración del domino, esto se guarda dentro del Open vSwitch el cual está unido a un puerto efímero, es decir, esta configuración dura poco, debido a que cuando el domino se apague, el puerto y la configuración asociada desaparecen y cuando se vuelva a iniciar el sistema esto no se encontrará cargado.

Para mantener la configuración actualizada es necesario tener en cuenta dos aspectos importantes, el primero definir una red virtual con `libvirt` que contenga las definiciones necesarias de `portgroup` y en segundo lugar incluir la referencia del `portgroup` a la red virtual en la configuración de las máquinas virtuales.

El portgroup consiste en agregar varios puertos bajo una configuración común y proporcionar un punto de anclaje para las máquinas virtuales que se conectan a redes etiquetadas, cada portgroup se identifica mediante una etiqueta de red, el cual es único para el host.

Para configurar la red virtual es necesario trabajar manualmente sobre un documento XML para después ser importado a libvirt con el comando virsh net-define <XML name>. Ejemplo de esto es el mostrado en la Figura 42 donde se muestran la etiqueta <bridge name> que define el nombre del dispositivo de puente que se utiliza para construir la red virtual. Las máquinas virtuales se conectaran a este dispositivo para que puedan comunicarse entre sí, así como <portgroup name> para definir las VLAN ya que esto es una parte importante para lograr la configuración requerida. [Libvirt 2015].

```
<network>
<name>ovs-network</name>
<forward mode='bridge'/>
<bridge name='ovsbr0'/>
<virtualport type='openvswitch'/>
<portgroup name='vlan-01'
default='yes'>
</portgroup>
<portgroup name='vlan-02'>
<vlan>
<tag id='2'/>
</vlan>
</portgroup>
<portgroup name='vlan-03'>
<vlan>
<tag id='3'/>
</vlan>
</portgroup>
<portgroup name='vlan-all'>
<vlan trunk='yes'>
<tag id='2'/>
<tag id='3'/>
</vlan>
</portgroup>
</network>
```

*Figura 42.Codigo para configurar la red virtual.*

Para la configuración del dominio también es necesario configurar otra parte del documento XML, donde la configuración importante es `<source network>` especificando el nombre de la red que se ha creado así como el nombre del `portgroup` a la que este sistema virtual deberá adjuntarse. Esta configuración se muestra en la Figura 43.

```
<interface type=’network’>
MAC> address=´'11:22:33:44:55:66'´
<source network='ovs-network'
portgroup='vlan-02'/>
</Interface>
```

*Figura 43.Codigo para configurar el dominio de la red virtual.*

Con el código mostrado en la Figura 43, se garantiza que dicha configuración no sea borrada la próxima vez que el sistema se apague sino que será de forma permanente. Esto es un beneficio, el problema radica en que siempre se deberá tener conocimiento a qué puerto del Open vSwtich el equipo invitado se encuentra conectado, es complicado lograr la correspondencia de los puertos con los sistemas invitados. También es difícil tratar con los archivos `XML` de `libvirt` para poder realizar la conexión, debido a estas desventajas se expone otra solución que para efectos de este proyecto será la opción que se usará. Se trata de trabajar con fake bridges (puentes falsos).

Los fake bridges actúan como un puente normal, con la diferencia de que se encuentran relacionados a una VLAN ID en particular [Rafael Bonifaz 2011].

Para este proyecto, se contará con un bridge padre el cual será el descrito anteriormente en el apartado de la creación de VLAN con ayuda de `libvirt`, los puentes falsos serán hijos de

este, ya que estarán unidos a él. Una vez que se tiene el bridge padre se pueden crear los bridges falsos lo cual se hace con el comando mostrado en la Figura 44.

```
#ovs-vsctl add br <fake bridge> <parent bridge> <VLAN>
#
```

*Figura 44. Comando para crear bridges falsos.*

Para la realización del proyecto se creó el bridge padre, el cual tiene 3 fake bridges, cada uno conectado a una máquina virtual; logrando con esto, tener la arquitectura mostrada en la Figura 45.

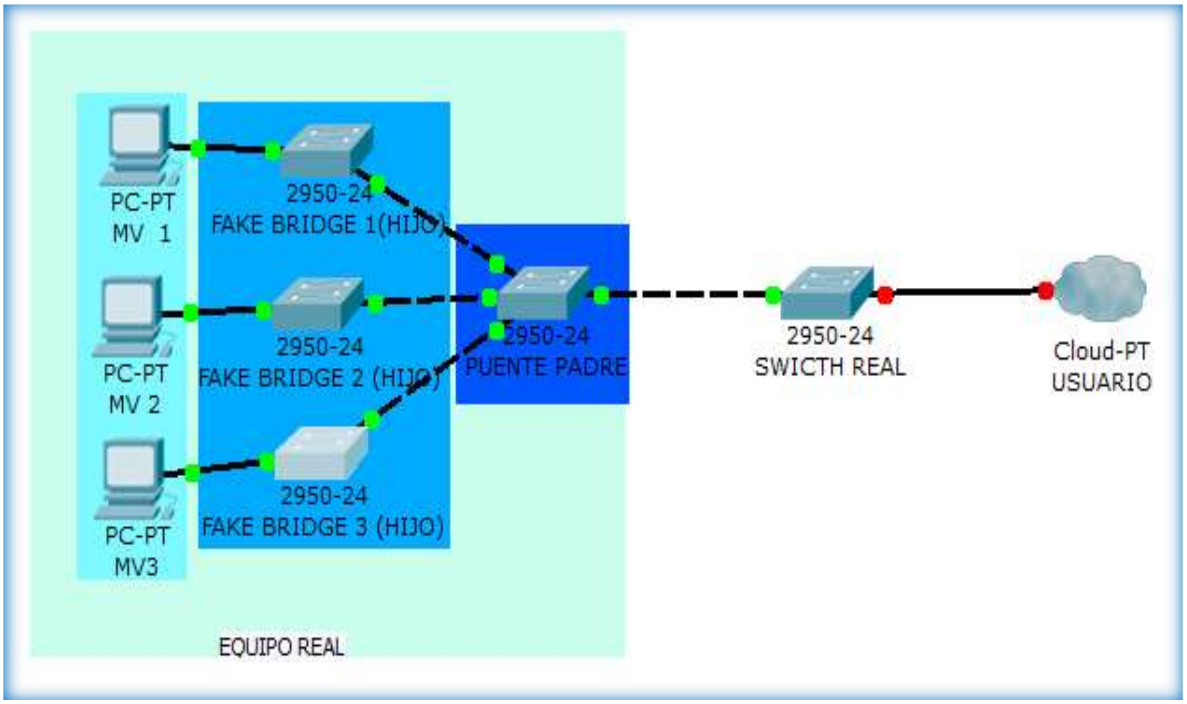


*Figura 45. Escenario del proyecto*

Con esta parte del proyecto ya se tiene el sistema virtual funcionando, habiendo logrado la conectividad de las máquinas virtuales dentro del equipo real, el paso siguiente es tener el acceso remoto a dicho equipo, es decir, lograr acceder a él desde fuera de la universidad.

## 3.8. Acceso remoto al equipo real

En este proceso de desarrollo para el proyecto, el objetivo general es poder acceder al equipo que aloja las máquinas virtuales de forma remota, para ello fue necesario tomar varias consideraciones para lograr el cometido, ya que no es un paso sencillo porque la máquina de interés no es accesible desde internet.

Para lograr el acceso a las máquinas virtuales desde fuera de la institución, se requiere realizar varios pasos y mecanismos, así como probar herramientas de acceso remoto ya existentes y determinar cuál de ellas es más adecuada para poder interactuar con las máquinas virtuales y que el usuario final, es decir, el alumno logre realizar configuraciones TCP/IP.

## 3.9. Aspectos importantes de acceso remoto

Un acceso remoto es poder acceder desde una computadora a un recurso ubicado físicamente en otra computadora que se encuentra geográficamente en otro lugar, a través de una red local o externa (como Internet).

Para realizar el acceso remoto es necesario tomar en cuenta los protocolos implicados, los cuales darán el conjunto de reglas para que una máquina se comunique con otra mediante el intercambio de mensajes.

Para lograr el acceso remoto a las máquinas virtuales es necesario considerar el uso de programas los cuales son una herramienta indispensable para que las computadoras reciban y envíen los datos, lo anterior realizado con las medidas de seguridad necesarias que garanticen la integridad de los datos. Para este caso los recursos a los cuales se requiere acceder a las máquinas virtuales. Para ello se hará uso de dos protocolos VNC y RDP de los cuales a continuación se mencionan las características de cada uno de ellos, se hace énfasis en las características que apoyan el desarrollo del proyecto [Soluciones IT 2008].

## 3.10 VNC (Virtual Network Computing)

Las características principales que describen el protocolo VNC se presentan a continuación:

- Software libre.
- Desarrollo en los laboratorio AT&T.
- Estructura cliente servidor, permitiendo tomar el control del ordenador servidor de forma remota a través del ordenador cliente.
- Sin restricciones en el sistema operativo del ordenador-servidor con respecto al cliente, es decir, se permite compartir la pantalla de una máquina con cualquier sistema operativo que soporte VNC conectándose desde otro dispositivo que disponga de un cliente VNC.
- El programa servidor puede funcionar como servidor HTTP para mostrar la pantalla compartida en un navegador con soporte de Java. El usuario no necesita tener

instalado un cliente VNC ya que éste es descargado por el navegador de forma inmediata.

- Con soporte para Windows cuando se requiere conectarse con dispositivos con sistemas operativos de esta arquitectura.
- El servicio consta de 3 partes: un cliente, un servidor y un protocolo de comunicación. El servidor es el programa que comparte la pantalla permitiendo tomar el control de la misma al cliente que es el programa que lo controla e interactúa con él. La pantalla es transmitida gracias a el protocolo RFB el cual basado en una primitiva gráfica del servidor al cliente, el cual coloca datos basándose en las coordenadas X,Y, transmitiendo mensajes de eventos desde el cliente al servidor.
- VNC no es un protocolo seguro ya que las contraseñas son enviadas en texto plano. Sin embargo VNC puede ser tuneleado (este término se explicará posteriormente), a través de una conexión SSH o VPN los cuales agregan seguridad extra a los datos.
- VNC por defecto usa puerto TCP 5900+N,5 donde N es el número de la pantalla (por lo general: 0 para una pantalla física).

## 3.11. RDP (Remote Desktop Protocol)

Las características importantes del protocolo RDP se describen a continuación:

- Protocolo propietario desarrollado por Microsoft que permite la comunicación en la ejecución de una aplicación entre un terminal y un servidor Windows.

- La información recibida que genera el servidor es convertida a un formato propio RDP y enviada a través de la red al terminal que interpreta la información contenida en el paquete del protocolo para reconstruir la imagen a mostrar en la pantalla terminal, utiliza el puerto por defecto TCP 3389.
- Permite uso de colores de 8, 16, 24 y 32 bits.
- Permite seguridad a nivel de la capa de transporte.
- Redireccionamiento de audio permite ejecutar un programa del lado del servidor y escuchar por medio del ordenador local.
- Permite utilizar ficheros locales en una ventana remota.
- Permite al usuario utilizar su impresora local al estar conectado al sistema remoto.

Con lo anterior expuesto, las características que interesan para el desarrollo de la tesis principalmente es tener acceso remoto de un cliente a un servidor ya que ésta es relevante para llevar a cabo los objetivos establecidos.

Una función importante del protocolo VNC es que no importa el tipo de plataforma donde éste se desarrolle ya que se tendrán usuarios que utilicen sistemas operativos Windows o Linux.

## 3.12. CLIENTE SSL/SSH VNC Viewer

El cliente SSL/SSH VNC Viewer se utilizará para realizar conexiones tanto con dispositivos Windows como Linux como se muestra en la Figura 46. Algunas características que rodean a este cliente son las siguientes:

- Agrega seguridad de cifrado para las conexiones VNC.
- Proporciona una interfaz gráfica de usuario para Windows, Mac OS X y Unix.

- Soporte SSL para las conexiones que utilizan el programa stunnel incluido.
- Posibilidad de guardar y cargar perfiles de VNC para los diferentes hosts.
- Crea certificados SSL y claves privadas.
- Conexiones a través de SSL y SSH.
- Establece redirecciones de puerto SSH adicionales deseadas.

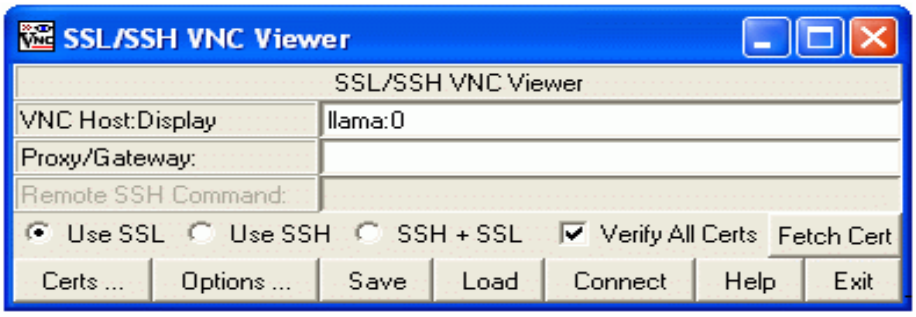


*Figura 46. Cliente SSL/SSH VNC Viewer*

## 3.13. PuTTy

PuTTy es un cliente que trabaja con dispositivos Windows y Unix. Se pueden realizar conexiones con SSH, Telnet, rlogin y TCP raw, el cual es de licencia libre. Entre las características más sobresalientes sobre este cliente se puede mencionar:

- Almacenamiento de host y facilidad de restaurar para uso posterior.
- Control sobre la clave de cifrado SSH y la versión de protocolo.
- Clientes de línea de comandos SCP y SFTP.
- Control sobre el redireccionamiento de puertos con SSH.
- Emuladores de terminal.

- Soporte Ipv6.
- Soporte de autenticación de clave pública.
- Conexiones de puerto serie local.

Para lograr tener acceso al servidor que aloja las máquinas virtuales se utilizaran túneles ssh los cuales se describen a continuación.

Un túnel SSH es una conexión que utiliza el protocolo SSH, el cual encripta los datos que transmite, transforma un puerto “X” en el equipo real en el puerto “Y” de otra máquina, para el caso del proyecto el servidor de las máquinas virtuales. Todo esto se realiza pasando por un tercer dispositivo.

Cuando se requiere conectarse a una maquina directamente se realiza lo mostrado en la Figura 47, ya que en este caso no se tienen prohibiciones o restricciones de acceso.

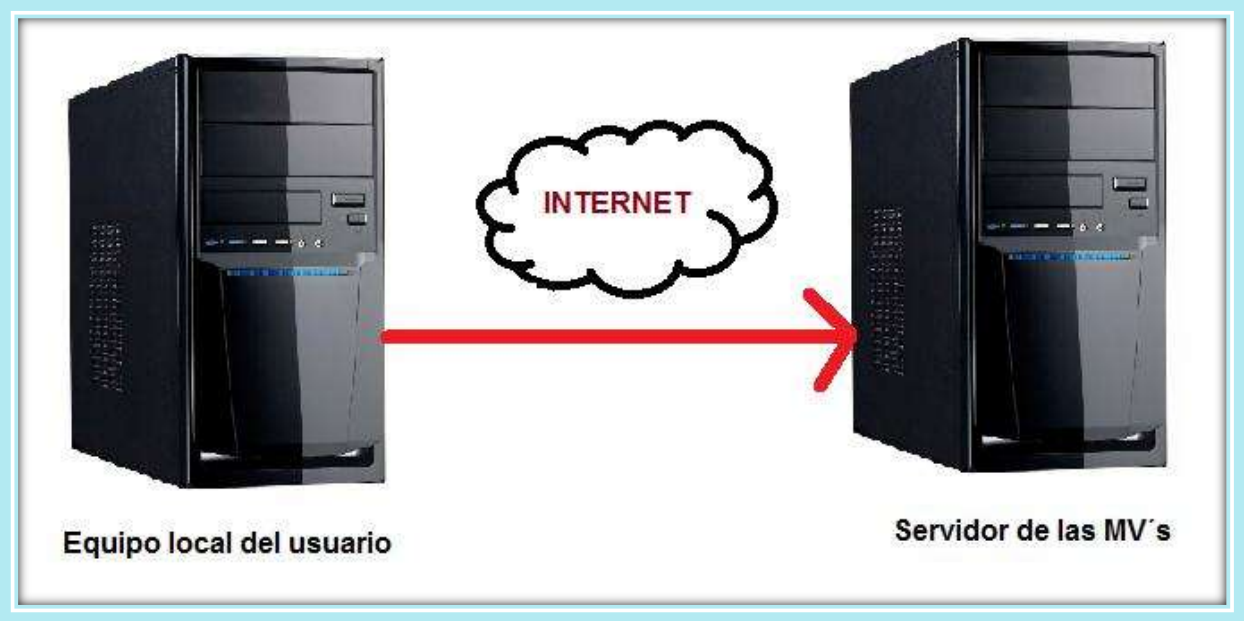


*Figura. 47. Conexión típica de dos equipos de cómputo*

En este caso, en el Centro de Cómputo Universitario de la U.M.S.N.H. filtra y prohíbe el acceso de otros equipos ajenos a la institución, pero este tercer dispositivo sí tiene acceso al servidor donde se encuentran alojadas las máquinas virtuales por lo tanto se crea un túnel como se muestra en la Figura 48 donde la conexión indicada mediante la flecha azul representa las conexiones permitidas.

La flecha verde es el Túnel SSH desde la computadora local hasta la computadora no accesible por internet a la cual es requerido conectarse.

Por último, la flecha roja indica la conexión de un puerto local del equipo real, dicha conexión es equivalente a la mostrada en la Figura 47.

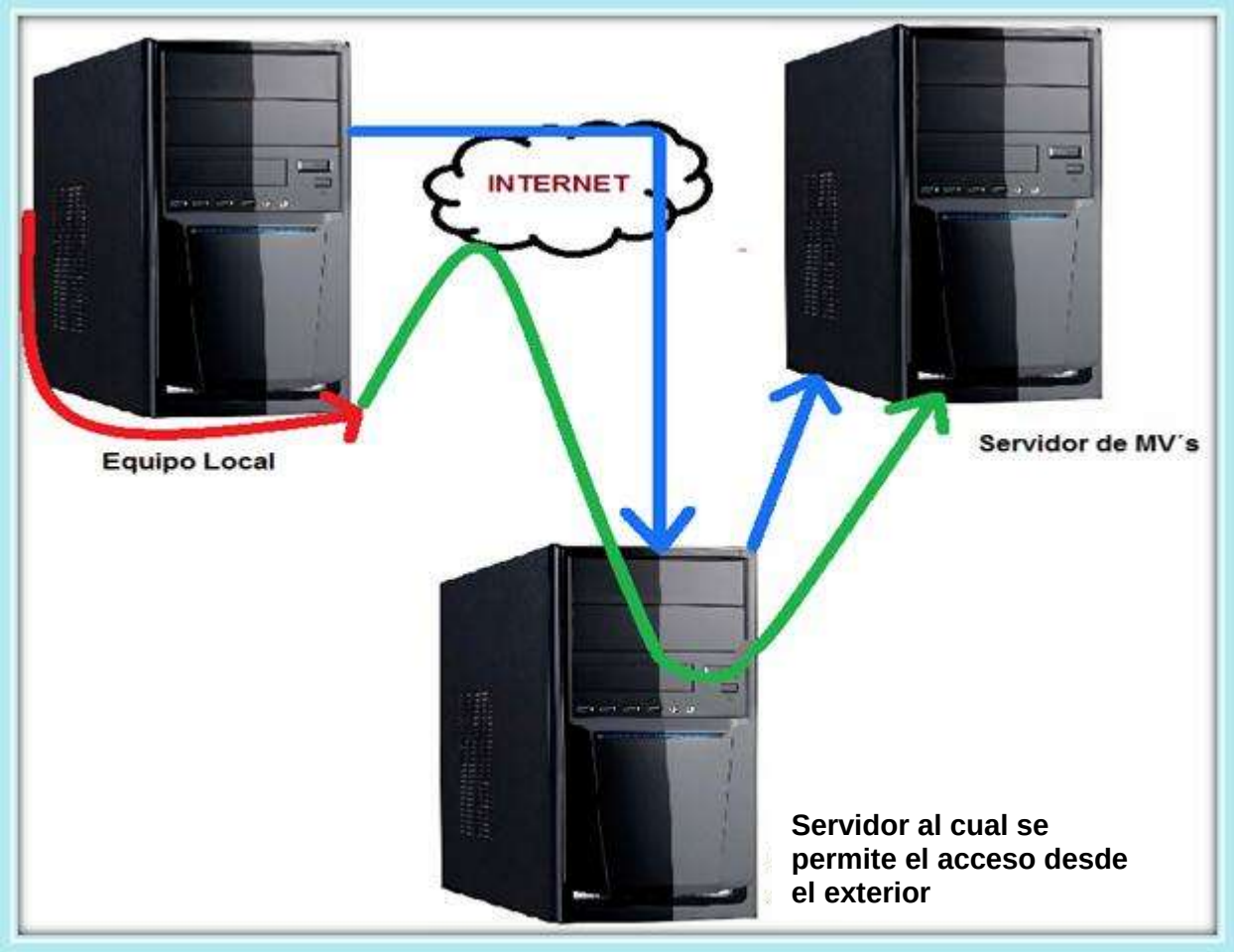


*Figura 48. Túnel SSH*

Para la lograr la comunicación es necesario teclear el comando mostrado en la Figura 49 en el equipo local para poder conectarse al servidor al cual se tiene acceso.

```
# ssh -L <puerto local>:<equipo remoto>:<puerto remoto> <equipo al que se tiene acceso>
```

*Figura 49. Comando para acceder al servidor desde el exterior.*

Esto significa que el puerto local del equipo del usuario es transformado en el puerto remoto que es el puerto ssh.

Se aborda el tema de túnel ssh debido a que continuación se muestran las pruebas que se realizaron para probar el acceso remoto donde se hace uso de estos.

## 3.14. Pruebas

La Figura 50 ilustra la forma como el usuario desde fuera de la universidad accede a las máquinas virtuales.

Se conecta al servidor local de la FIE con IP 148.216.17.24 que se encuentra en la Intranet 148.216.17.0/24 el cual solo es accesible desde internet con el usuario del alumno.

Se crea un puente (como se indica posteriormente en los apartados 3.14.1 y 3.14.2) a partir de 148.216.17.24 para poder tener acceso hacia el servidor servicios.fie.umich.mx con IP 148.216.17.28, el cual pertenece a la misma Intranet, pero no se puede acceder desde Internet.

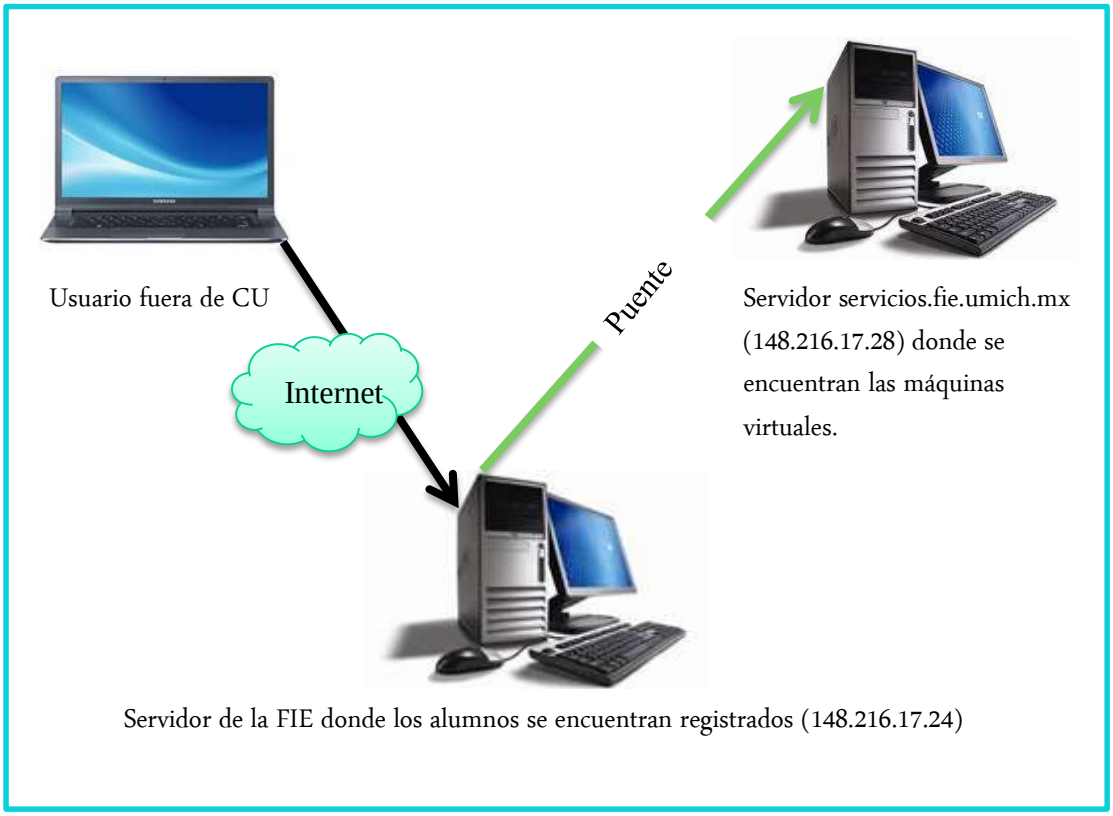


*Figura 50. Escenario para acceder desde fuera de la universidad a las máquinas virtuales.*

A continuación se muestran las pruebas realizadas tanto con usuarios Linux como con usuarios de Windows.

### 3.14.1. Prueba 1: Usuario Linux con cliente SSL/SSH VNC usando un proxy

El objetivo de esta prueba es la de verificar la conexión de un usuario de su equipo que cuenta con sistema operativo Linux al servidor donde se encuentran alojadas las máquinas virtuales mediante el cliente SSL/SSH.

Para efectos de este proyecto, se nombra Julieta al alumno que requiere el acceso a las máquinas virtuales. Enseguida son descritos los pasos para lograr este cometido desde el cliente.

## Paso 1.

Editar el archivo `etc/ssh/ssh_config`, este archivo es el principal de configuración de los clientes OpenSSH utilizado desde el equipo anfitrión local. Dentro de él se escribe tal como lo muestra la Figura 51, donde la opción `ProxyCommand` permite conectar a un servidor SSH a través de un proxy. En este caso se expone que el host 148.216.17.183 podrá ser accedido a través del servidor servicios.fie.umich.mx.

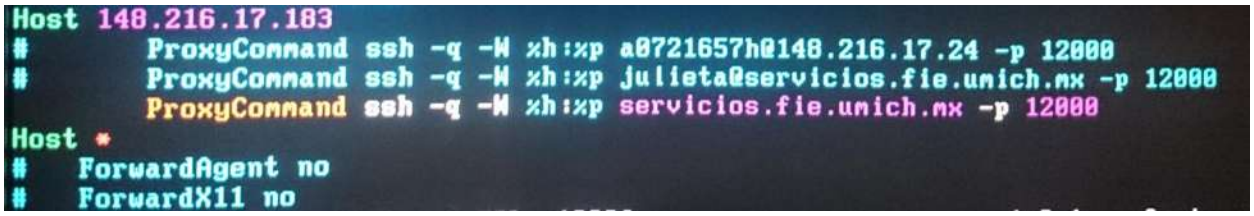


*Figura 51. Configuración del archivo etc/ssh/ssh_config.*

El host es la IP de la máquina virtual a la cual se requiere conectar, la cual es 148.216.17.183, ésta máquina está alojada en el servidor servicios.fie.umich.mx con IP 148.216.17.28. Para la cual, con el uso de la primera entrada de `ProxyCommand` permite la conexión al servidor con IP 148.216.17.24, donde la alumna Julieta con matricula a0721657h está registrada, esto a través del puerto 1200. Ya estando logeado en ese sitio el siguiente salto es hacia el servidor servicios.fie.umich.mx donde se encuentra la máquina virtual, dentro de este servidor el usuario debe estar registrado para lograr el acceso, esto se realiza con el segundo comando `ProxyCommand`.

El último comando `ProxyCommand` permite el acceso al servidor donde se encuentra la máquina virtual con IP 148.216.17.183.

### Paso 2.

Como lo indica la Figura 52, dentro del cliente SSL/SSH VNC se selecciona la casilla ssh ya que se le está indicando que se conecte por medio de este protocolo y en Proxy/Gateway se escribe la IP de la máquina virtual, la cual es 148.216.17.183 y en la casilla VNC se marca el host local el cual es 127.0.0.1:5911.

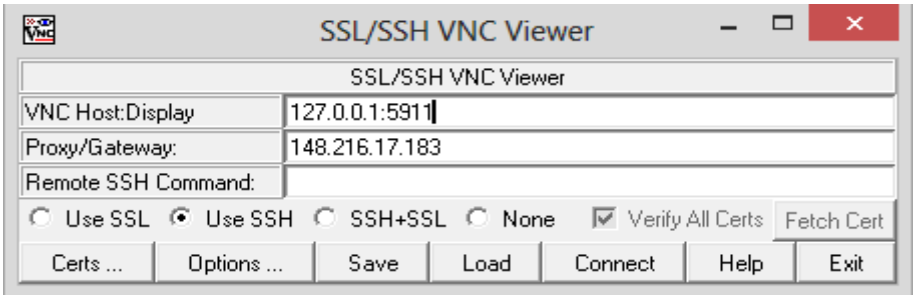


*Figura 52. Configuración dentro del cliente SSL/SSH VNC.*

## Resultados

Una vez realizado estos dos pasos se pedirán 3 contraseñas, la primera para entrar a **a0721657h@148.216.17.24** donde la alumno Julieta se encuentra registrado como usuario de la Facultad, la segunda para **julieta@servicios.fie.umich.mx** donde la alumna de la materia de Laboratorio de Redes de Computadoras deberá estar registrado y la última para acceder a la máquina virtual que se encuentra en el servidor **servicios.fie.umich.mx**.

La Figura 53 muestra la máquina virtual a la cual se accedió.

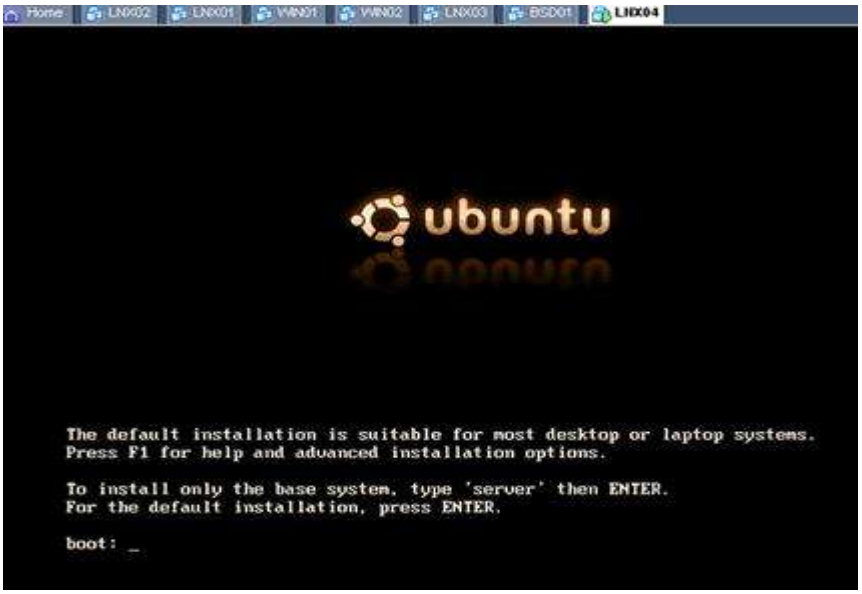


*Figura 53. Visualización de la máquina virtual en Ubuntu*

## 3.14.2. PRUEBA 2: USUARIO WINDOWS CON CLIENTE SSL/SSH VNC USANDO UN PROXY

En esta sección se requiere lograr el acceso de un usuario que se quiere conectar mediante su equipo con sistema operativo Windows al servidor donde se encuentran las máquinas virtuales con las cuales realizará las configuraciones correspondientes, usando igualmente como en el objetivo anterior el cliente SSL/SSH VNC. En este caso se hará uso de un Túnel SSH el cual permitirá acceder a la máquina virtual desde fuera de la Universidad. Enseguida se describen los pasos necesarios a seguir para realizar esta prueba dentro del equipo del alumno.

## Paso 1.

Especificar el destino al cual se requiere conectar que en este caso es primero a servicios.fie.umich.mx donde se encuentra la máquina virtual mediante el puerto 12000 como lo muestra la Figura 54.

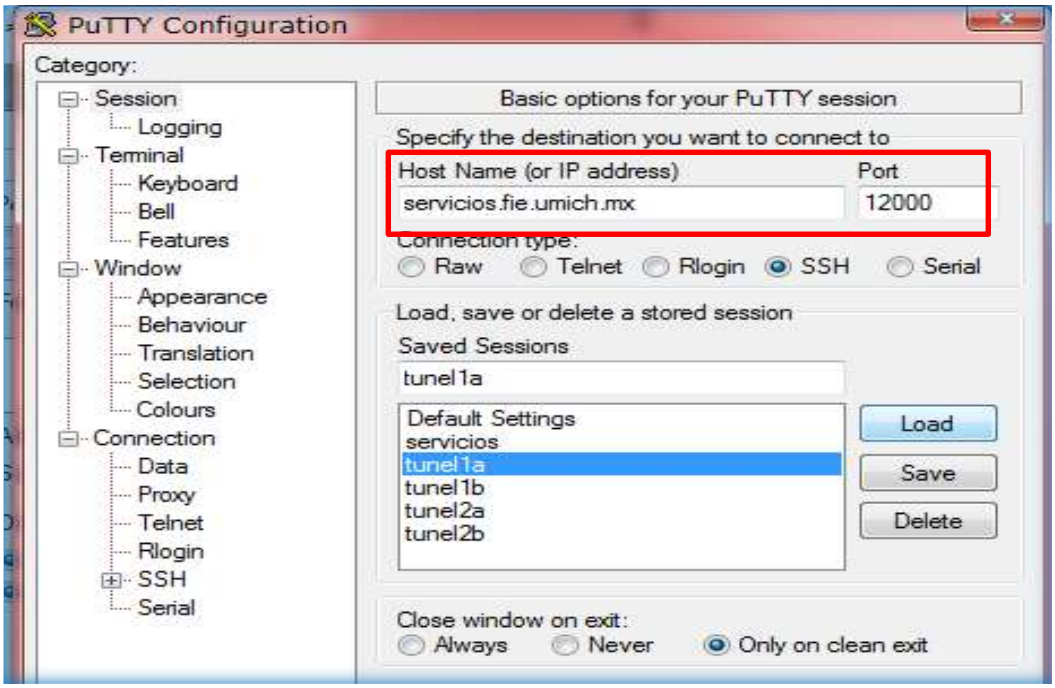


*Figura 54. PuTTY*

## Paso 2.

Se crea el primer túnel el cual será el primer salto para lograr la conexión mediante PuTTY. Para realizar esto en la opción de  Categoría de PuTTY se selecciona la opción de Túnel, y se escribe la dirección de la máquina virtual a la cual se requiere el acceso indicando el puerto 22 de SSH, como lo muestra la Figura 55.

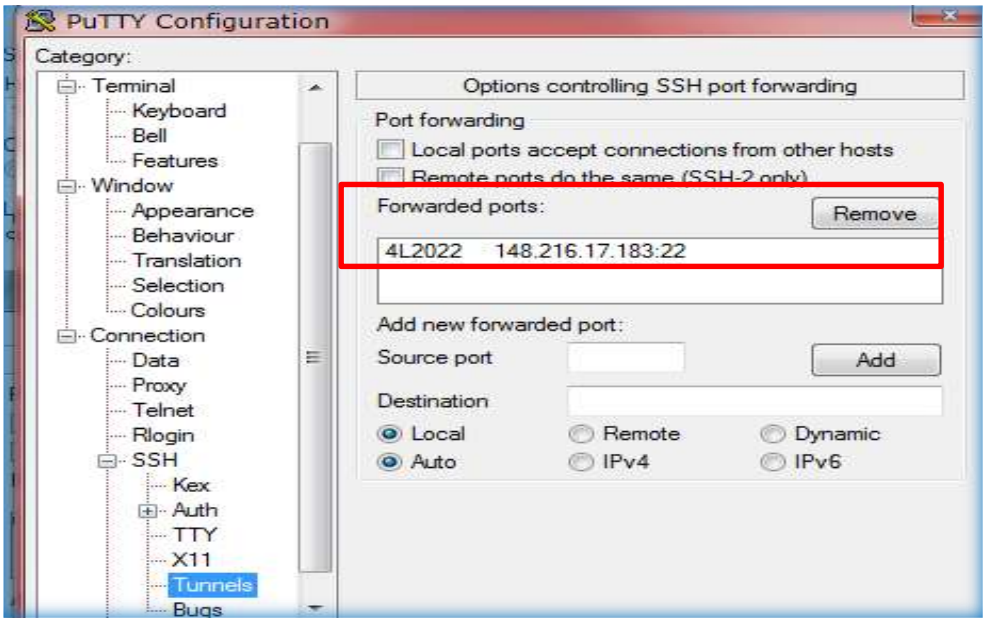


*Figura 55. Primer túnel para el acceso*

## Paso 3.

Ahora se crea el segundo salto donde se especifica ahora el host, es decir, la máquina del alumno como lo muestra la Figura 56.

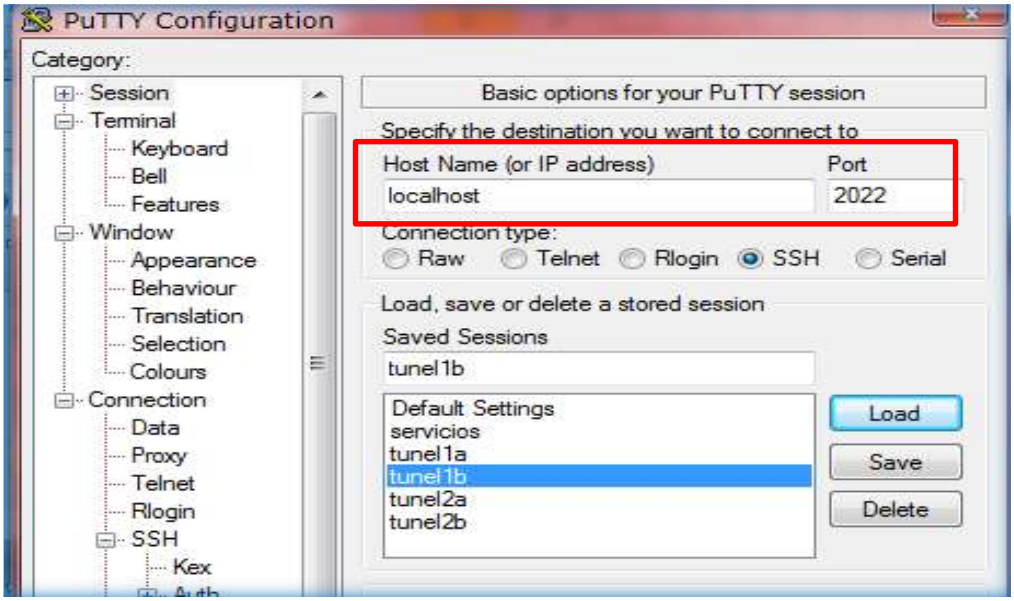


*Figura 56. Segundo salto para la conexión*

## Paso 4.

Crear el túnel del último salto donde se especifica la dirección de la máquina del alumno como lo muestra la Figura 57.

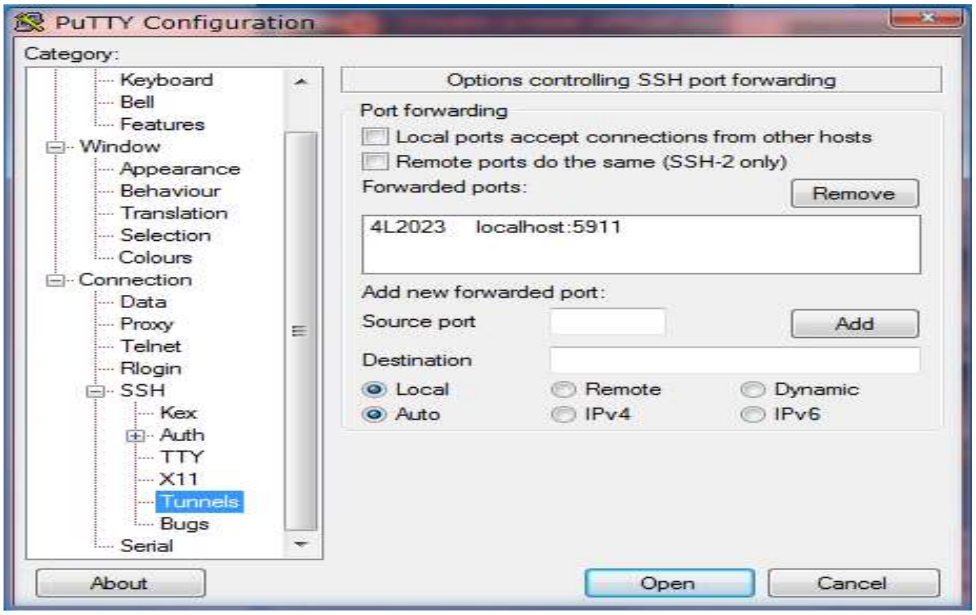


*Figura 57. Segundo túnel.*

Se hace uso del cliente SSL/SSH VNC para introducir los siguientes datos dentro de la casilla VNC Host:Display : localhost:5911 como lo ilustra la Figura 58 para lograr el acceso.

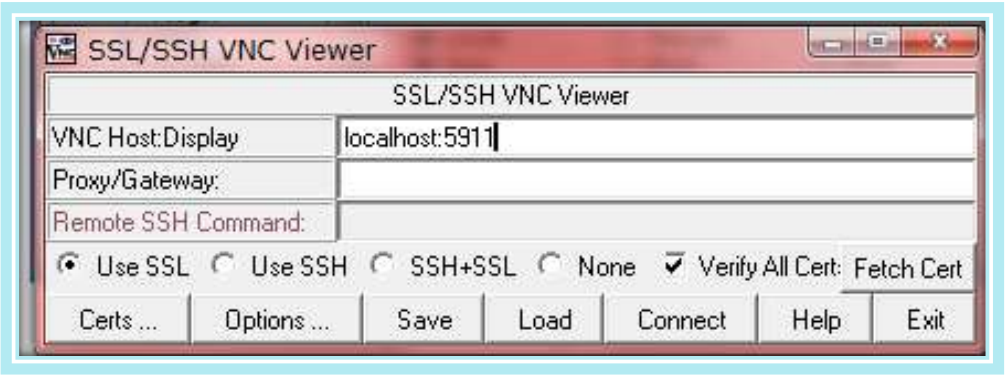


*Figura 58. Conexión al servidor de las MV´s.*

Una vez realizado el paso anterior se puede tener el acceso a la máquina virtual como lo muestra la Figura 59.

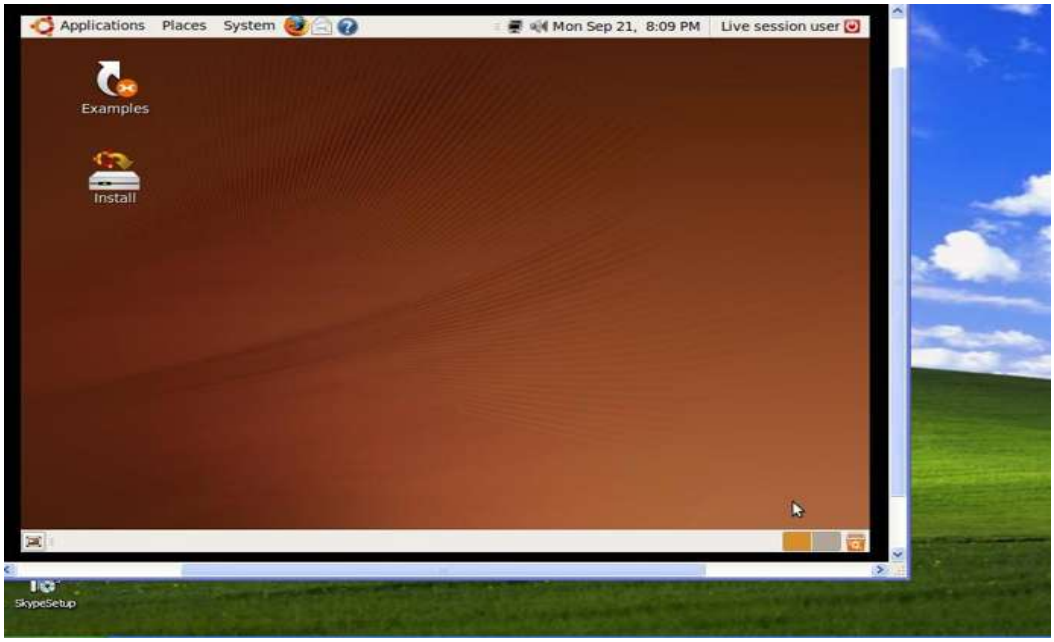


*Figura 59. Acceso a máquina virtual usando Windows.*

Es importante mencionar que las configuraciones para usuarios Linux será necesario realizarlas cada vez que se requiera conectar a la máquina virtual, y para usuarios Windows es posible guardar los Túneles para la conexión en PuTTY.

## 3.15. Comentarios Finales

En este capítulo se abordó el tema principal del que trata este trabajo de tesis, ya que se explicó la solución del problema que en un principio se planteó. Se describió el funcionamiento e instalación de la herramienta de virtualización KVM dentro del entorno Linux. Mediante esta herramienta y el uso de la aplicación de acceso remoto VNC Viewer se probó la conexión entre

el usuario que se encuentra en su hogar, con la máquina virtual alojada en servicios.fie.umich.mx, que se encuentra en la intranet de la UMSNH.

Se describe la manera de establecer una conexión ssh, tanto en sistema operativo Linux como Windows.

# *Capítulo 4. Software de Gestión de Usuarios*

En este capítulo se hablará del Sistema de Gestión de Usuarios el cual se encargara del registro de usuarios, registrar las solicitudes de agendar una práctica, así como permitir el acceso a la herramienta Guacamole donde se encontraran disponibles las máquinas virtuales con las que se trabajará, ya sea con sistema operativo Linux o Windows.

## 4.1. Descripción del Objetivo

Una vez que se ha logrado tener la conexión de forma remota al equipo que tiene las máquinas virtuales de interés, el proceso siguiente es administrar o gestionar la forma en que los usuarios van a poder acceder a dicho sistema para poder realizar el cometido, que como se ha venido explicando, es la configuración de dispositivos de red finales (máquinas virtuales).

Se creó una pequeña base de datos en el manejador de base de datos MySQL, a partir del modelo entidad-relación mostrado en la Figura 60, conformado por dos tablas una llamada `alumno` y la otra `agenda_practica`; misma que resulta accedida y afectada, mediante scripts desarrollados en los lenguajes orientados a eventos como lo es HTTP y PHP.

Para registrar usuarios, los datos necesarios requeridos son el nombre y apellidos del usuario, su matrícula, el nombre de usuario y contraseña, cada usuario tendrá su propio identificador.

La otra tabla creada es para reservar un tiempo en el laboratorio para lo cual será requerido el nombre del usuario, así como la fecha y hora inicial y final para cuando requiera la utilización del equipo.

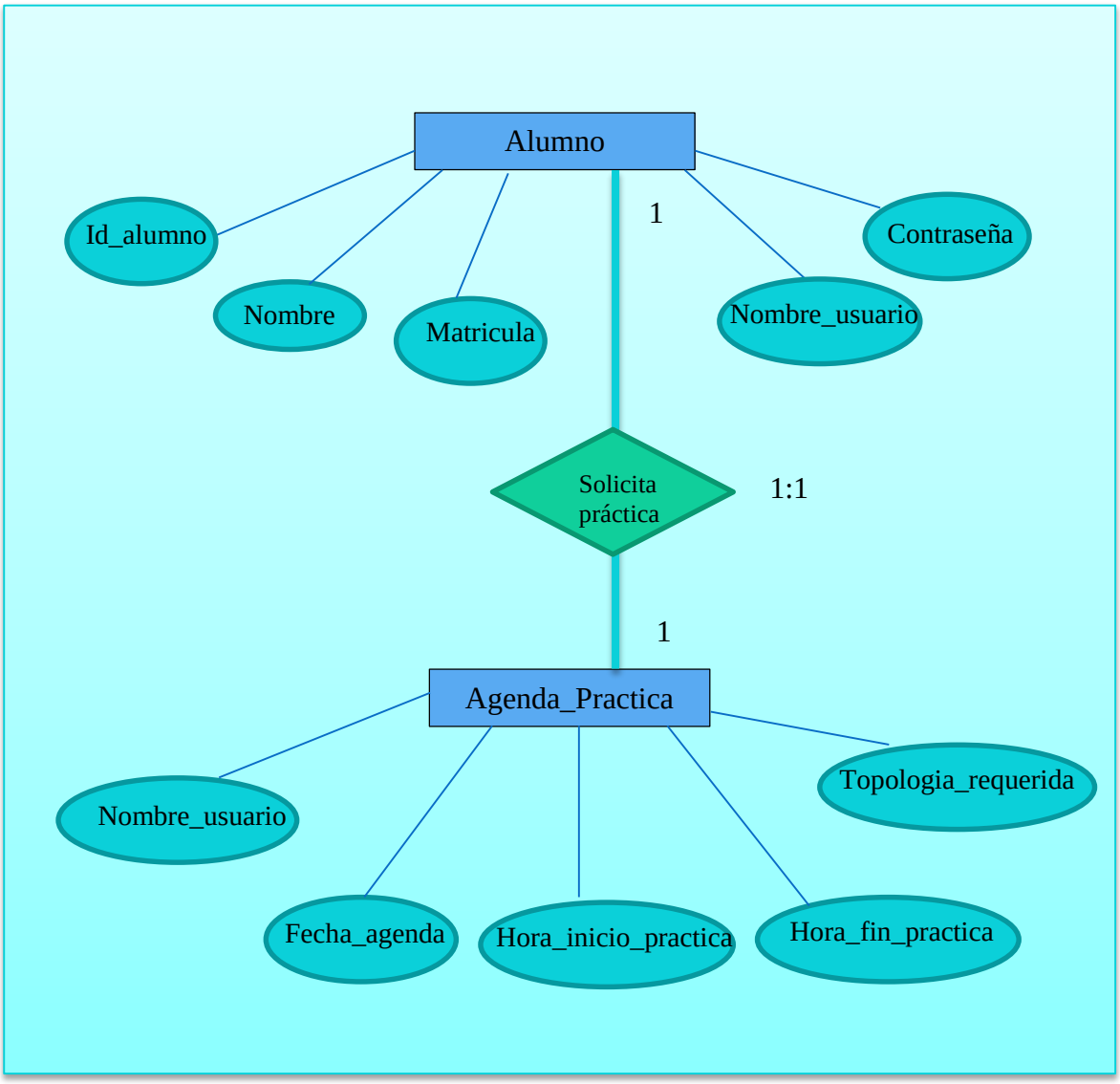


*Figura 60. Modelo Entidad-Relación.*

Enseguida se muestran vistas de las páginas que conforman el sistema de administración de usuario.

## 4.2. Autenticación de usuarios

Como punto inicial, se tiene la página donde el usuario se autentifica con su usuario y contraseña para poder acceder a la página principal del sistema, como se muestra en la Figura 61.


*Figura 61. Autenticación de usuarios*

## 4.3. Página principal del sistema

Una vez que el usuario se encuentra dentro del sistema, se muestra la página (Figura 62), donde éste puede realizar diferentes acciones, con las restricciones necesarias. La cuales, están determinadas dependiendo si el usuario autentificado se trata de un alumno o del administrador del sistema. En esta página se aprecian las reservaciones del laboratorio, así como enlaces para registrar usuarios, registrar reservaciones del laboratorio, mostrar las reservaciones del usuario

que en ese momento este dentro del sistema y por ultimo una pestaña que permite cargar topologías al servidor (definidas por el profesor de la materia de Redes de Computadoras), las cuales mostraran las conexiones disponibles a los usuarios para la realización de las practicas.


*Figura 62. Página principal*

## 4.4 Registro de usuarios

Esta página (Figura 63), permite el registro de los usuarios, ya sean alumnos o encargado del laboratorio a los cuales se les permitirá el acceso al sistema. Para ello son requeridos los datos más significativos del usuario como lo son; el nombre completo, matricula, nombre de usuario y contraseña. Así mismo se indica si el usuario que se está registrando será administrador o no del sistema, esto con el fin de restringir determinadas acciones dentro del mismo.

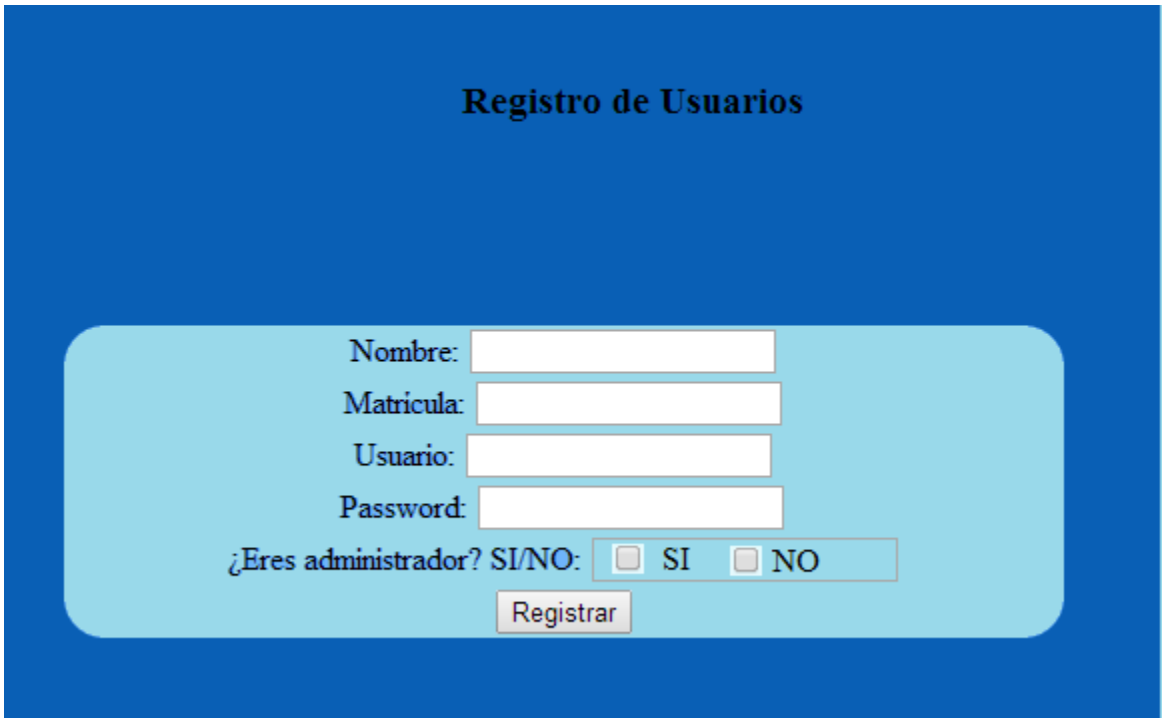


*Figura 63. Registro de Usuarios*

## 4.5. Agendar práctica

En esta sección se permite el registro de un usuario para realizar una práctica. Para lo cual es requerido: el nombre del usuario, el día que desea realizar la práctica, así como la hora de inicio y terminación, para determinar el periodo de tiempo que el laboratorio estará siendo ocupado; como se ilustra en la Figura 64.

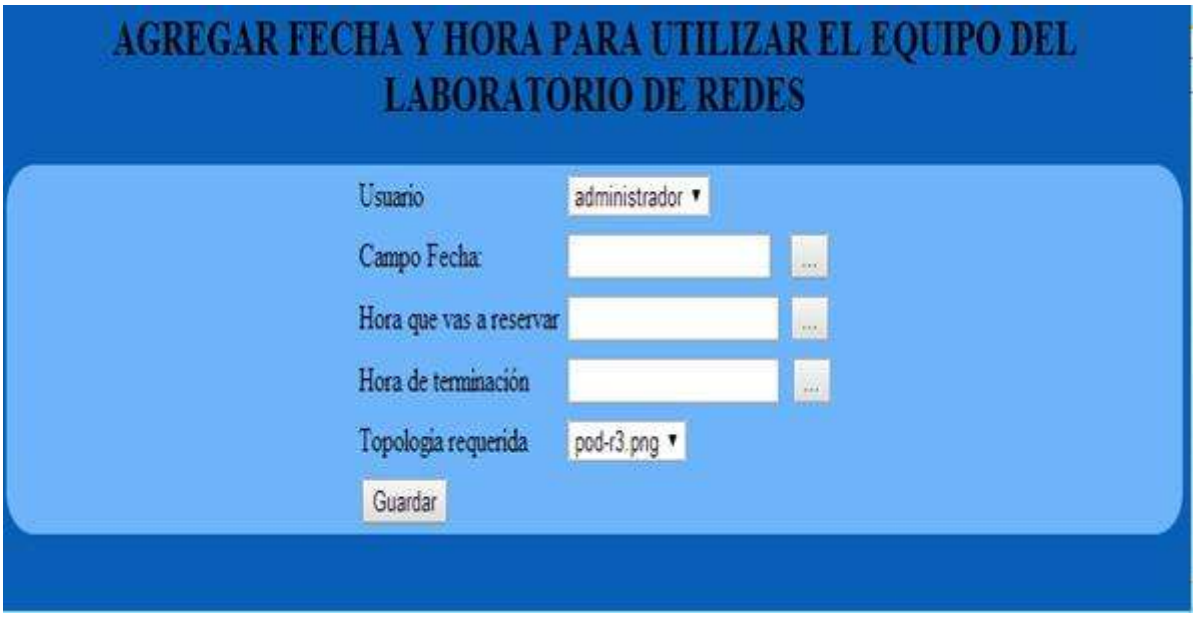


*Figura 64. Reservar laboratorio*

## 4.6. Inspección de reservaciones de forma individual

En esta sección el usuario que está conectado al sistema en ese momento, puede revisar las reservaciones del laboratorio que él ha solicitado; donde el sistema mostrará la fecha que el alumno solicitó así como la hora y la topología establecida como se ilustra en la Figura 65.

**Mis Reservaciones del Laboratorio de redes**

| | | | | | |
|---|---|---|---|---|---|
| 1 | Luis Perez | 2014/02/28 | 11:10:00 | 11:12 | pod-r3.png |
| 2 | Luis Perez | 2014/02/28 | 11:25:00 | 11:35 | pod-r3.png |
| 3 | Luis Perez | 2014/02/28 | 12:01:00 | 12:20 | pod-r3.png |
| 4 | Luis Perez | 2014/03/03 | 20:33:00 | 21:10 | pod-r3.png |
| 5 | Luis Perez | 2014/03/12 | 19:01:00 | 21:10 | pod-r3.png |

*Figura 65. Información de reservaciones de un usuario activo*

## 4.7. Exposición de topologías disponibles

Dentro de la sección topologías disponibles, el usuario puede observar (Figura 66) las topologías que el laboratorio tiene disponibles en ese momento, las cuales fueron cargadas previamente por el administrador del sistema.

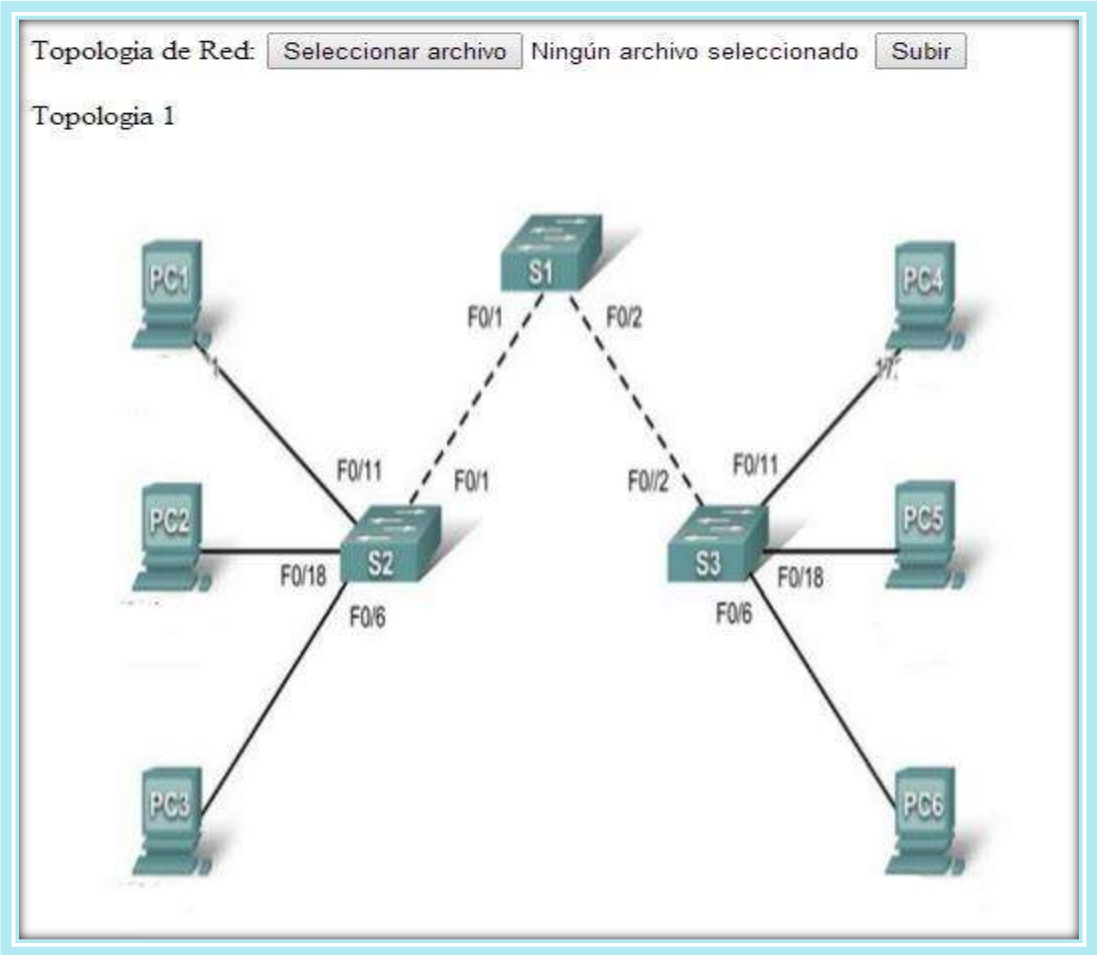


*Figura 66. Topologías disponibles*

## 4.8. Unión del Sistema de administración de usuarios con la aplicación guacamole

### 4.8.1. Software Guacamole

Guacamole es una aplicación de software libre que está basada en web HTML 5, la cual permite a los usuarios acceder a escritorios remotos, tanto de Windows como de Linux, por medio del navegador; evitándose hacer uso de plugins u de otro tipo de software de conexión remota del lado del cliente, siendo esto la principal ventaja que caracteriza este software sobre la mayoría de soluciones de acceso remoto comúnmente usadas [Apache Guacamole 2016].

Guacamole no es una aplicación web independiente, sino que está compuesto de varias partes. Dicha aplicación web está diseñada para ser simple y contener solo componentes esenciales y que al mismo tiempo sea funcional.

La Figura 67, muestra la arquitectura de la aplicación Guacamole, la cual está compuesta por un demonio llamado `guacd` (detallado en el apartado 4.8.1.2) que se comunica con la máquina del alumno en lo que a este proyecto respecta, mediante un protocolo, ya sea RDP o VNC.

Por otra parte, guacd se conecta con un contenedor de servlet (Clases Java, utilizadas para ampliar las capacidades de un servidor, capaces de responder a cualquier tipo de solicitudes, de tal manera que pueden ser vistos como applets de Java que se ejecutan en servidores en vez de navegadores web), que permite que se muestre en la pantalla del usuario, el escritorio de la máquina virtual.

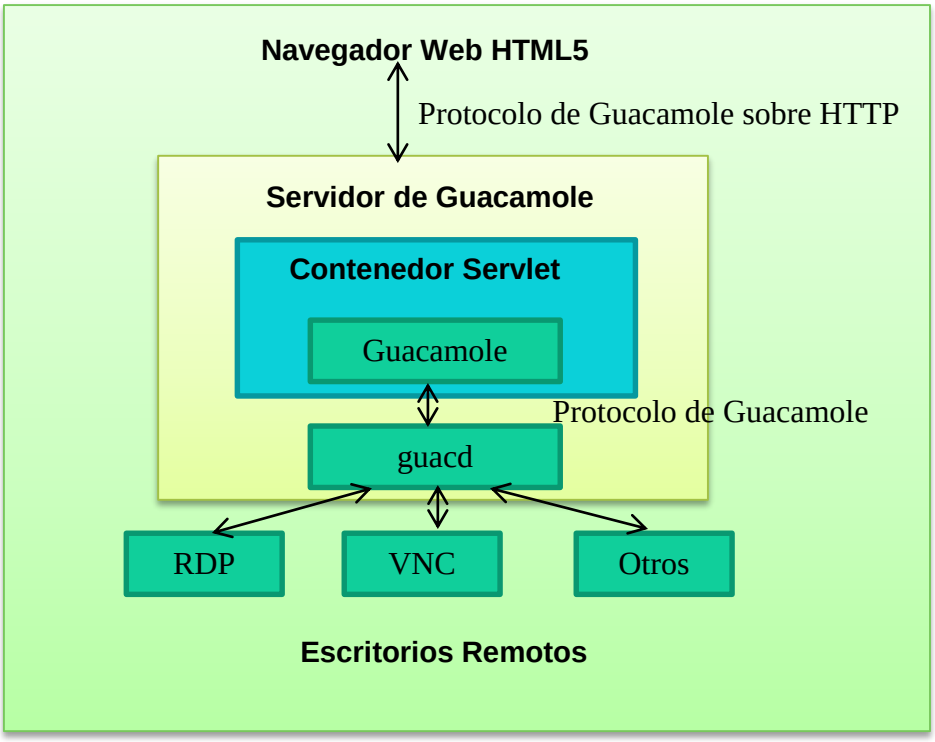


*Figura 67. Arquitectura de Guacamole*

### 4.8.1.1. El protocolo Guacamole

La aplicación web no entiende cualquier protocolo de escritorio remoto, no contiene soporte para VNC, RDP u otro protocolo soportado por Guacamole. Esta aplicación solo entiende el protocolo Guacamole, el cual representa la visualización remota y transporte de eventos. Un protocolo con esas propiedades tendría las mismas capacidades que un protocolo de escritorio remoto, pero el protocolo Guacamole y los principios de diseño detrás de un protocolo de escritorio remoto son diferentes, ya que Guacamole no está destinado a poner en práctica las características de un entorno de escritorio específico.

Guacamole como una pantalla remota y el protocolo de interacción, implementa un superconjunto de los protocolos de escritorio remotos existentes, añadiendo soporte para un protocolo de escritorio remoto en particular (como RDP) para Guacamole. Por lo tanto consiste en escribir un capa intermedia llamada `guacd` que traduce el contenido del protocolo de escritorio remoto y el protocolo Guacamole.

### 4.8.1.2. guacd

Guacd es la parte principal de Guacamole, ya que se encarga de la compatibilidad con los protocolos de escritorio remoto (llamados plugins Cliente), conectándolos a escritorios remotos basados en las instrucciones recibidas de la aplicación web. Se trata de un servicio ó demonio que se instala junto con Guacamole y es ejecutado en segundo plano.

Guacd escucha conexiones TCP de la aplicación web, pero no entiende ningún protocolo específico de escritorio remoto. Sino que únicamente implementa lo necesario del protocolo Guacamole para determinar que necesita el apoyo del protocolo para ser cargado. Una vez cargado el plug-in del cliente, se ejecuta de forma independiente y tiene el control de la comunicación entre él mismo y la aplicación web hasta que se termina la sesión.

### 4.8.1.3. La aplicación web

Es con la que el usuario interactúa, no implementa ningún protocolo únicamente es basada en `guacd`, proporcionando solo una interfaz web y la capa de autenticación.

Ahora que se tiene conocimiento del funcionamiento de la aplicación Guacamole, a continuación se explicará la forma en que ésta es relacionada con el sistema de administración de usuario previamente presentado.

La principal función del sistema de administración de usuario es la de registrar usuarios y reservaciones del equipo del laboratorio. Dicho sistema permite al alumno revisar el día y hora que él ha solicitado hacer uso del laboratorio, como se aprecia en la Figura 68, así mismo el sistema proporciona al alumno una contraseña la cual le permitirá entrar a Guacamole, donde estarán disponibles las conexiones de la topología que el alumno solicitó. Cuando se haya llegado el tiempo de que el usuario vaya a utilizar el laboratorio, el sistema informará que a ese usuario se le ha dado una contraseña como se observa en la Figura 69 en el recuadro rojo.

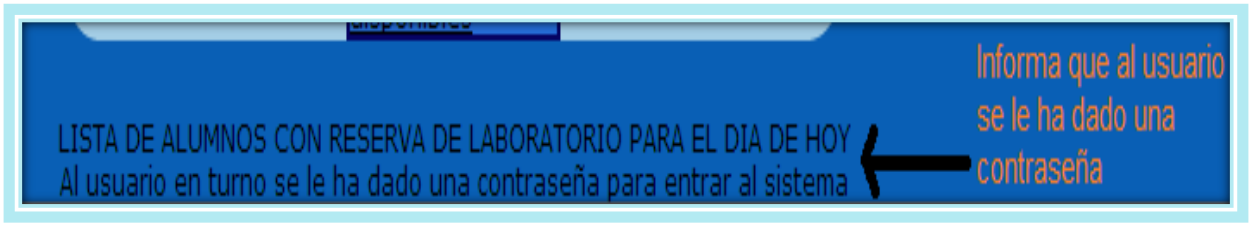


*Figura 68. Información al sistema de que se ha asignado una contraseña*

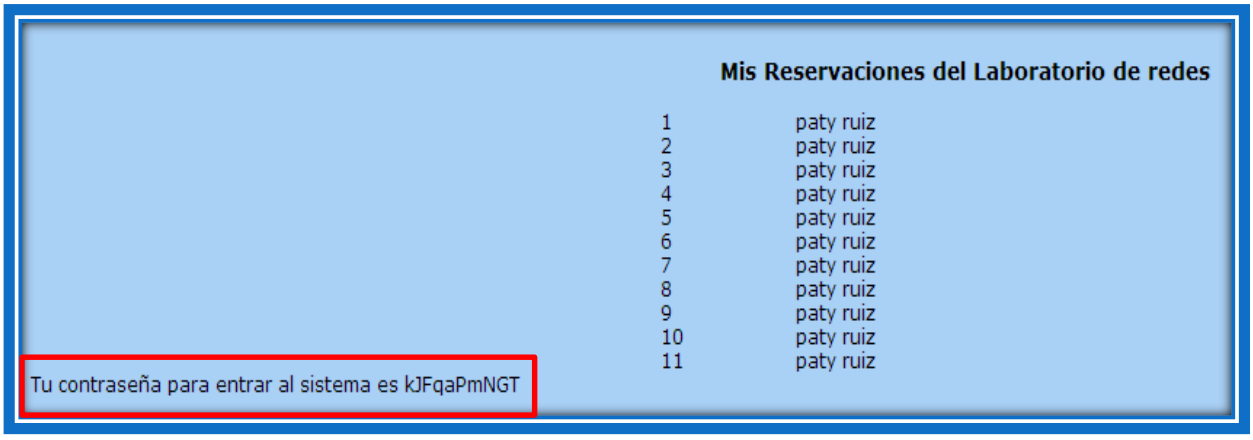


*Figura 69. Contraseña asignada al usuario para entrar a Guacamole*

Guacamole permite la autenticación de usuarios mediante MySQL; así mismo permite la modificación de datos que respaldan el módulo de autenticación MySQL, de forma manual mediante la ejecución de sentencias SQL contra la base de datos. Esta característica de modificación de datos manualmente es un argumento importante para esta tesis, ya que se requiere que por cada usuario registrado en el sistema de reservaciones, ese mismo usuario sea registrado en Guacamole.

## 4.9. Usuarios en Gucamole

Cada usuario en Guacamole tiene una entrada correspondiente en la tabla `guacamole_user`. Cada usuario le corresponde un único nombre de usuario con su correspondiente contraseña salt, esta contraseña salt es dividida en dos columnas, una que contiene la contraseña `salt` y la otra que contiene la contraseña `hash` con SHA-256.

La tabla `guacamole_user` contiene las siguientes columnas:

- `user_id`

  Es el número entero único asociado a cada usuario, este valor es generado de forma automática después de que se inserta un nuevo usuario en esta tabla.

- `username`

  Se refiere al nombre único asociado con cada usuario, este valor se debe especificar manualmente y debe ser diferente de cualquier nombre de usuario existente en la tabla.

- `password_hash`

Es el resultado de aplicar hash a la contraseña del usuario concatenándolo con el contenido de password_salt utilizando SHA. La contraseña salt se añade a la contraseña antes de hash.

- `password_salt`

  Es un valor aleatorio de 32 bytes. Cuando se crea un nuevo usuario desde la interfaz web, este valor es generado aleatoriamente utilizando un generador de números aleatorios.

Para la inserción de un usuario de forma manual, el comando `insert` se realiza como la muestra la Figura 70. Enseguida, en la Figura 71 se muestra parte del código introducido en el sistema para que se inserte un usuario una vez que se ha llegado la hora de que se le permita acceder al sistema. Ambos códigos se describen más adelante en esta sección.

```
--Generar contraseña salt
$sql="SET @salt=UNHEX(SHA2(UUID(), 256))";
--Insertar usuario
INSERT INTO guacamole_user (nombre de usuario,
password_salt, password_hash) VALUES ('nick', @
sal, UNHEX (SHA2 (CONCATHEX (@ sal)),256)));
```

*Figura 70. Inserción manual de usuarios en Guacamole*

```
1. $sql="select user_id, username from guacamole.guacamole_user where
   username='$nick'";
2. $res=mysql_query($sql);
3. $fila=mysql_fetch_array($res);
4. $user_id=$fila['user_id'];
5. if(mysql_fetch_array($res)){
6.    sql="SET @salt=UNHEX(SHA2(UUID(), 256))";
7.    $res=mysql_query($sql);
8. $sql="update guacamole.guacamole_user set
   password_salt=@salt,password_hash=UNHEX(SHA2(CONCAT('passwordGuacamole',
   HEX(@salt)), 256)) where username='$nick'";
9. $res=mysql_query($sql);
10.                          }
11.  else {
12.     $sql="SET @salt=UNHEX(SHA2(UUID(), 256))";
13.     $res=mysql_query($sql);
14.     $sql="insert into guacamole.guacamole_user (username, password_salt,
   password_hash) VALUES ('$nick', @salt, UNHEX(SHA2(CONCAT('$hola', HEX(@salt)),
   256)));";
```

*Figura 71. Inserción de usuarios manual dentro de Guacamole usando PHP y MySQL*

En la primera línea del código mostrado en la Figura 71 se comprueba si el usuario que desea acceder a Guacamole ya se encuentra registrado, mediante la sentencia `select` se realiza la búsqueda dentro de la tabla `username` en la columna `user_id` del sistema Guacamole.

La variable `nick` es el nombre de la celda que se da para registrar el nombre del usuario en el sistema administrador de usuarios.

En la línea 2, la variable `$res` guarda la consulta realizada en la base de datos de `guacamole_user` mediante la sentencia `mysql_query`.

Con la sentencia `mysql_fetch_array` se extraen los datos de la consulta y se guardan en un array, descrito en las líneas 3 y 4. Si se encuentra un resultado entonces no se insertará este usuario sino que solo se cambiara su contraseña para entrar a Guacamole, la cual es la contraseña que al usuario se le da una vez que se ha llegado el momento de iniciar su reservación del laboratorio, como se indica en el código de la línea 8.

Mediante la sentencia `update` se actualiza las columnas `password_salt` **y** `password_hash` en la base de datos `guacamole_user`. Si por el contrario el usuario aún no se encuentra dentro de la tabla de usuarios de Guacamole, entonces éste es insertado y así mismo la contraseña asignada acorde a las líneas 13 y 14, donde mediante la sentencia `insert` se actualiza la base de datos guacamole en la tabla `guacamole_user` las columnas `username, pasword_hash` **y** `pasword_salt,` con los datos del nuevo usuario registrado.

Así también, es necesario modificar los datos de otras dos tablas de la base de datos de Guacamole de manera manual, con el fin de que el usuario pueda tener acceso a la aplicación y estando ahí pueda comenzar con la práctica del laboratorio. Una de ellas es la tabla de permisos de conexión `guacamole_connection_permission,` la cual permite a un usuario específico realizar una operación específica dentro de una conexión existente. Esta tabla contiene los siguientes parámetros:

- `user_id` **:** Este valor es el asociado a un usuario específico.
- `connection_id`**:** Este valor asocia la conexión entre la máquina virtual y el usuario que la usa.

- `permission`: Indica el permiso que será concedido al usuario. Los permisos pueden ser los siguientes; administrar: que permite la posibilidad de agregar o quitar permisos que afectan la conexión; leer: permite consultar los datos asociados a la conexión; actualizar: otorga la capacidad de actualizar los datos asociados con la conexión; y eliminar: permite la eliminación de esta.

Otra tabla de la cual es necesario modificar los datos es la tabla de permisos de conexión `guacamole_connection_group_permission`, la cual permite que un usuario que se encuentra registrado dentro de ella, pueda realizar actividades, tales como administrar, leer, cargar o borrar una conexión. Esta tabla contiene los siguientes campos:

- `user_id` : Este valor es el asociado a un usuario específico.
- `connection__group_id`: Este campo asocia un grupo de conexión afectados por un permiso en específico.
- `permission`: Indica el permiso que será concedido al usuario, los cuales pueden ser de administrar, leer, cargar o borrar una conexión.

El código mostrado en la Figura 72 asigna permisos de conexión de grupo y los permisos de conexiones al usuario al cual se le ha llegado la hora de inicio de práctica.

En la línea 12 mediante la sentencia `select` se busca al usuario que hará uso del sistema en ese momento, la consulta se realiza dentro de la tabla `guacamole_user` y se guarda en un array en este caso nombrado **como** `$fila`.

La variable `$user_id` se le asigna este array el cual contendrá el nombre del usuario que esta por realizar su práctica. En la línea 19 se inserta este usuario dentro de la tabla `guacamole_coneccion_group_permission`, el cual pertenecerá al grupo 1 de conexión y solo tendrá el permiso de leer. En la línea 22 se inserta el usuario dentro de la tabla `guacamole_connection_permission` el cual mediante el constructor `foreach` asignara la conexión de las máquinas virtuales 2,3,4,5,9,10,11,12 con el usuario que esta hacienda uso del sistema, mismo que también solo tendrá el permiso de lectura.

```
15.$res=mysql_query($sql);
16.$sql="select user_id, username from guacamole.guacamole_user where
   username='$nick'";
17.$res=mysql_query($sql);
18.$fila=mysql_fetch_array($res);
19.$user_id=$fila['user_id'];
20.$q="insert into guacamole.guacamole_connection_group_permission
   values ('$user_id', 1, 'READ')";
21.$res=mysql_query($q);
22.foreach(array(2,3,4,5,9,10,11,12) as $conn){
23.$q="insert into guacamole.guacamole_connection_permission values
   ('$user_id', '$conn', 'READ')";
24.$res=mysql_query($q);
25.$sql="select user_id, username from
   guacamole.guacamole_user where username='$nick'";
26.$res=mysql_query($sql);
27.$fila=mysql_fetch_array($res);
28.$user_id=$fila['user_id'];
29.$q="insert into guacamole.guacamole_connection_group_permission
   values ('$user_id', 1, 'READ')";
30.$res=mysql_query($q);
31.foreach(array(2,3,4,5,9,10,11,12) as $conn){
32.$q="insert into guacamole.guacamole_connection_permission values
   ('$user_id', '$conn', 'READ')";
33.$res=mysql_query($q);
```

*Figura 72. Asignación de permisos de conexión*

Una vez que se ha llegado la fecha y hora para comenzar la práctica que el usuario solicitó (como se mostró anteriormente el sistema de gestión de usuarios), éste proporcionará su usuario y contraseña para tener acceso a la plataforma de Guacamole (ver Figura 73).

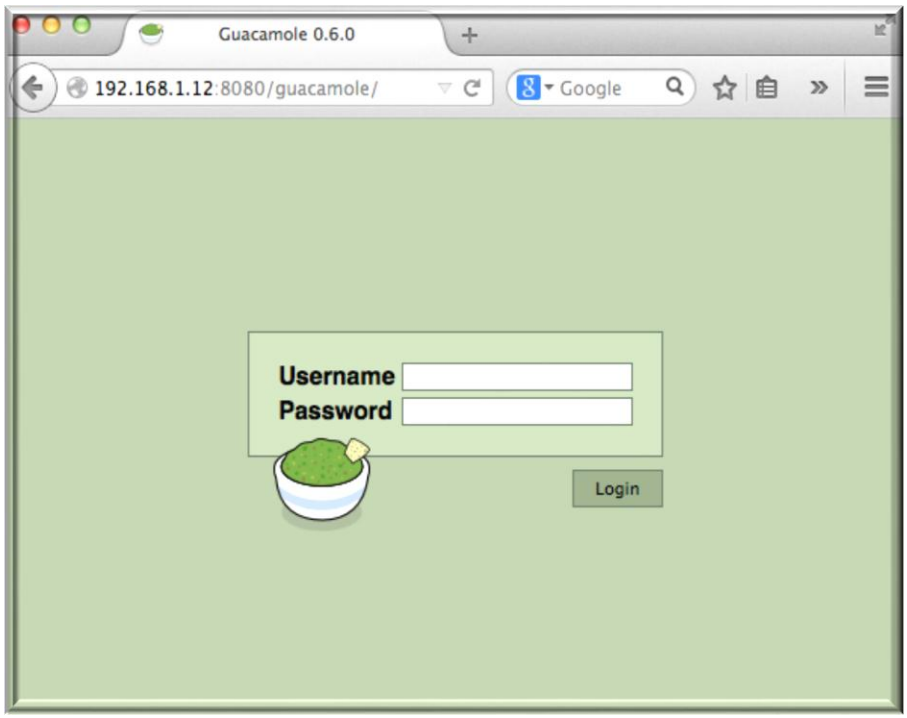


*Figura 73. Login de Guacamole*

Ya estando dentro del sistema Guacamole, éste mostrará las conexiones disponibles, ya sea para una máquina con entorno Windows o Linux; para que el usuario pueda comenzar con las actividades de configuración.

La Figura 74 muestra la pantalla que muestra guacamole una vez que se ha iniciado sesión.

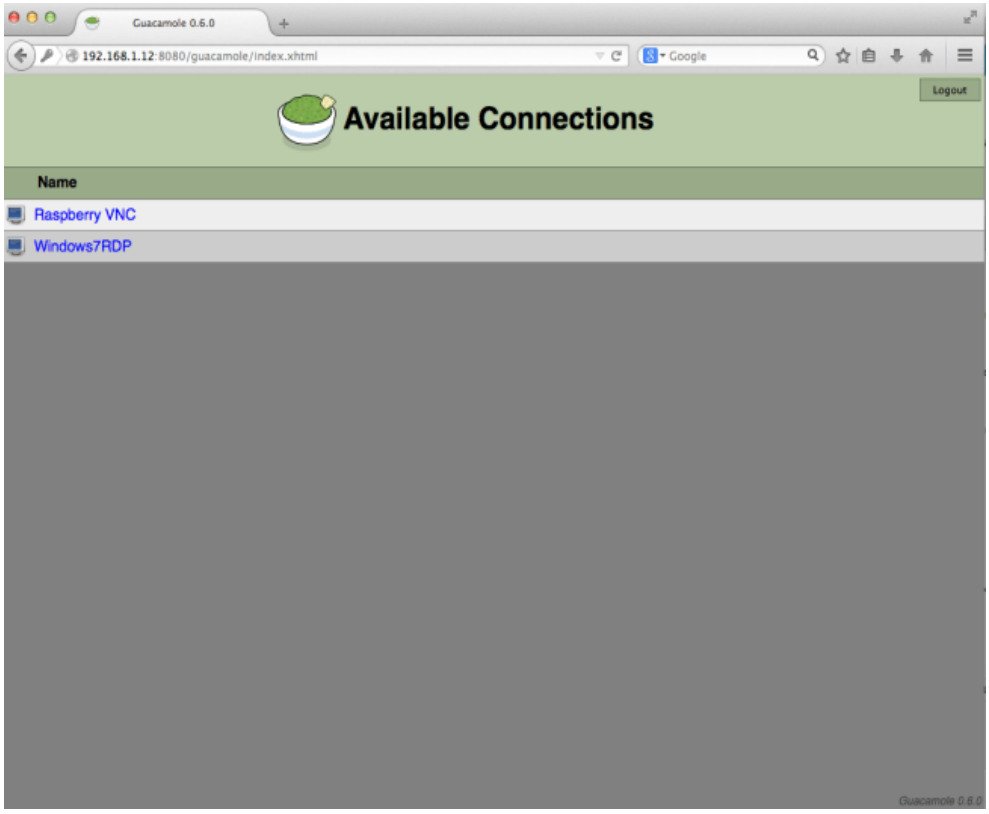


*Figura 74. Pantalla de inicio a Guacamole*

## 4.10 Comentarios Finales

En este capítulo se explicó la manera en que se diseñó el Sistema de Gestión de Usuarios, el cual permitirá el acceso al sistema a un usuario que se encuentre registrado para realizar prácticas de laboratorio específicas.

El sistema permite el registro de usuarios, reservar tiempo para poder practicar, así como mostrar las topologías que se tienen disponibles.

También se dio una pequeña explicación acerca de la aplicación Guacamole, la cual es un software de escritorio remoto tanto Linux como Windows.

# *Capítulo 5. Conclusiones y Trabajos futuros*

## 5.1. Conclusiones

Hoy en día, las tecnologías de la información y comunicación son de relevante importancia para lograr que personas de una determinada zona se puedan conectar o comunicar de manera satisfactoria con otras, dentro de otra zona geográficamente diferente.

Las grandes empresas buscan la mayor eficiencia sin que sus gastos aumenten a medida que su tecnología es eficiente. Es por eso, que los temas tratados y manejados dentro de este proyecto de tesis, incluyen tecnologías actuales, como es el caso de la virtualización, la cual, a pesar de que su desarrollo data desde hace más de 50 años, no se cuenta con la documentación necesaria para abordar el tema. Aún con esta desventaja, se tiene contemplado que dentro de un par de años más esta tecnología este más de moda, debido a las grandes funcionalidades que esta ofrece.

Otro tema del cual se abordó en este trabajo, es el acceso remoto, se investigó y estudiaron algunas de las diferentes herramientas que permiten tener acceso a un equipo de cómputo sin tener que estar presentes de manera física frente a él.

Este proyecto de tesis surge de la necesidad de mantener la relación de aprendizaje teórico y práctico, la cual, debido a diferentes circunstancias como ya se ha mencionado, no es posible para los usuarios que no tienen la posibilidad de practicar con equipo físico. Por lo tanto se planteó un mecanismo mediante el cual los usuarios no restringieran la posibilidad de realizar prácticas con los dispositivos disponibles en el laboratorio.

En un principio como bien se planteó, no se tenía contemplado trabajar con máquinas virtuales, pero debido a que serían necesarias 2 interfaces de red dentro de los equipos físicos, no resulta económicamente conveniente.

Por otra parte, que los usuarios no sabrían cual interfaz sería con la cual desarrollar su práctica y podrían confundirse con la interfaz mediante la cual se tiene acceso al sistema de forma remota.

En ese momento también se pensó en, qué tipo de usuario utilizaría este sistema, es decir, si se trataría de un alumno que maneje Linux o un alumno que trabaje mayormente con Windows. Es por eso que se realizaron los dos tipos de pruebas para conectar con cada sistema operativo ya mencionados.

El software de gestión de usuario es de relevante importancia, porque permite tener un control sobre el sistema y sobre los usuarios que requieran el servicio. Así como las reservaciones que este ha hecho del laboratorio ya que una vez que se tienen registrados dentro de este sistema será posible dar acceso a la aplicación de Guacamole que es donde se encuentran las conexiones de topologías disponibles.

## 5.2. Trabajos futuros

Las actividades planeadas para realizar posteriormente son:

Realizar más pruebas mediante otros clientes SSH para los diferentes sistemas operativos que se están manejando.

Ampliar la base de datos para considerar varios aspectos más, como lo son, no permitir que un usuario quiera practicar más de una vez el mismo día, acciones a realizar cuando la hora de

reserva coincide con una existente, entre otros varios. Todo esto para lograr que la base de datos, así como el sistema en general, esten desarrollados de forma completa con los menos errores y problemas posibles.

Desarrollar un servidor de consolas es un proceso que tiene mayor relación con personas que estén dentro de la rama de electrónica. Para realizar el sistema de forma completa, es necesario contar con dicho dispositivo, ya que no es suficiente tener en funcionamiento un sistema que solo provea el acceso a los dispositivos reales. Por tal motivo es necesario contar con este servidor, y una vez creado, buscar la manera de mediante el software ya creado también permitir el acceso hacia él.

# *Capítulo 6. Bibliografía*

## 6.1 Referencias bibliográficas

**[Academia.edu 2014]**

Academia.edu. *Definición de virtualización*.15 de mayo de 2014. http://www.academia.edu/9522025/UNIVERSIDAD_NACIONAL_MAYOR_UNMSM_2014_-II_INDICE_Introduccion_3_Definici%C3%B3n_de_virtualizaci%C3%B3n_3

**[Apache Guacamole 2016]**

Apache Guacamole. *Guacamole Manual*. 15 de agosto de 2014. https://guacamole.incubator.apache.org/doc/gug/

[**Capacity 2014]**

Capacity. *¿Qué es la virtualización y cuáles son sus beneficios?* 23 de abril de 2013. http://blog.capacityacademy.com.

**[Data Keeper 2015]**

Data Keeper. *¿Qué son los hypervisores?* 8 de junio de 2014. http//www.datakeeper.es/?p=716

**[Geeks 2007]**

Geeks. *Modelos de virtualización*. Consultado el 14 de mayo de 2013, de http://geeks.ms/eliasmereb/2007/07/09/modelos-de-virtualizacin/

**[IBM 2011].**

IBM. *Anatomy of a Linux hypervisor*. 17 de mayo de 2013. http://www.ibm.com/developerworks/ssa/linux/library/l-kvm-virtual-server/

**[IBM 2015]**

IBM. *La anatomía del hypervisor Linux*. 25 de Abril 2013. http://www.ibm.com.

**[Informaticaitc 2015]**

Informaticaitc. *Virtualización*. 3 de junio de 2014. http://informatictoday.blogspot.mx/2013/07/virtualizacion.html

**[IT From All Angles 2012]**

IT From All Angles. *Linux-KVM: Ubuntu 12.04 with Openvswitch*. 04 de junio de 2013. http://blog.itfromallangles.com

**[KVM 2016]**

KVM. *Kernel Virtual Machine*. 24 de junio de 2014. http://www.linux-kvm.org/page/Main_Page

**[Libvirt 2015]**

Libvirt. *Network XML Format*. 25 de julio de 2014. https://libvirt.org/formatnetwork.html

**[Mera O.S 2012]**

*Análisis técnico experimental para la implementación de una solución sobre servidores Blade con la utilización de máquinas virtuales en un entorno vSphere, aplicado a la empresa Montamant***.** Tesis de licenciatura publicada, Universidad Politécnica Salesiana, Quito, Ecuador.

**[Microsoft 2016]**

Microsoft. *Como utilizar la característica de Escritorio remoto de Windows XP Professional.* 18 de julio de 2014. https://support.microsoft.com/es-es/kb/315328/es

**[Mis Libros De Networking, 2007]**

Mis libros Networking. *Funcionalidades SPAN para monitoreo de tráfico a través de swtiches*. 05 de junio de 2013. http://librosnetworking.blogspot.mx.

**[NDG 2016]**

NDG. *NETLAB+Overview.* 24 de mayo de 2014. http://www.netdevgroup.com/products/.

**[NSA 2016]**

NSA. *Security-Enhanced Linux*. 28 de junio de 2014. http://www.nsa.gov/what-we-do/research/selinux/

**[OpenSuse 2016]**

OpenSuse. AppArmor – openSUSE. 28 de junio de 2014. https://es.opensuse.org/AppArmor

**[Open vSwitch 2014]**

Open vSwitch. *Production Quality, Multilayer Open Virtual Switch*. 25 de junio de 2014. http://openvswitch.org

**[Ovs-discuss 2014]**

Ovs-discuss. *Installing Open vSwitch on Ubuntu 12.04*. 06 de junio de 2013. http://openvswitch,org.

**[Rafael Bonifaz 2011]**

Rafael Bonifaz. *Túneles SSH*. 02 de junio de 2013. http://rafael.bonifaz.ec.

**[Soluciones IT 2008]**

Soluciones IT. *Acceso Remoto*. 23 de junio de 2014. https://solucionesit.wikispaces.com/-/Srvs015/AccesoRemoto/Acceso+Remoto

**[Suite 101 2016]**

Suite 101. *Comprendiendo Kernel Virtual Machine (KVM).* 23 de mayo de 2013. http://suite101.net.

**[Telnet.org 2015]**

Telnet.org. *Places to Telnet.* 15 de junio de 2015. http://www.telnet.org/htm/places.htm.

**[Timetoast 2014]**

Timetoast. *Historia de la virtualización*. 25 de abril de 2014. http://www.timetoast.com/timelines/historia-de-la-virtualizacion--8

**[Ubuntu 2016]**

Ubuntu. *KVM/Installation*. 21 de mayo 2013. http://help.ubuntu.com

**[Virt-manager 2015]**

Virt-manager. *Manage virtual machines with virt-manager*. 25 de mayo de 2013. http://virt-manager.org

**[Wikispaces 2016]**

Wikispaces. *Comparación de herramientas de virtualización de sistemas operativos*. 15 de mayo de 2013. http://nglnlx.wikispaces.com.

**[802.1Q 2014]**

802.1Q. *802.1Q – Virtual LANs*. 30 de julio de 2014. http://www.ieee802.org/1/pages/802.1Q.html